对象存储

API 文档

产品文档



腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

- API 文档
  - 简介
  - 公共请求头部
  - 公共响应头部
  - 错误码
  - 请求签名
  - 操作列表
  - Service 接口
    - GET Service（List Buckets）
  - Bucket 接口
    - 基本操作
      - PUT Bucket
      - GET Bucket（List Objects）
      - HEAD Bucket
      - DELETE Bucket
      - GET Bucket Object Versions
    - 访问控制（acl）
      - PUT Bucket acl
      - GET Bucket acl
    - 跨域资源共享（cors）
      - PUT Bucket cors
      - GET Bucket cors
      - DELETE Bucket cors
    - 生命周期（lifecycle）
      - PUT Bucket lifecycle
      - GET Bucket lifecycle
      - DELETE Bucket lifecycle
    - 存储桶策略（policy）
      - PUT Bucket policy
      - GET Bucket policy
      - DELETE Bucket policy
    - 防盗链（referer）
      - PUT Bucket Referer
      - GET Bucket Referer
    - 标签（tagging）

版权所有：腾讯云计算（北京）有限责任公司

- PUT Bucket tagging
- GET Bucket tagging
- DELETE Bucket tagging

静态网站（website）

- PUT Bucket website
- GET Bucket website
- DELETE Bucket website

智能分层（IntelligentTiering）

- PUT Bucket IntelligentTiering
- GET Bucket IntelligentTiering

清单（inventory）

- PUT Bucket inventory
- POST Bucket inventory
- GET Bucket inventory
- List Bucket Inventory Configurations
- DELETE Bucket inventory

版本控制（versioning）

- PUT Bucket versioning
- GET Bucket versioning

存储桶复制（replication）

- PUT Bucket replication
- GET Bucket replication
- DELETE Bucket replication

日志管理（logging）

- PUT Bucket logging
- GET Bucket logging

全球加速（Accelerate）

- PUT Bucket Accelerate
- GET Bucket Accelerate

存储桶加密（encryption）

- PUT Bucket encryption
- GET Bucket encryption
- DELETE Bucket encryption

自定义域名（domain）

- PUT Bucket domain
- GET Bucket domain
- DELETE Bucket domain

回源（Origin）

版权所有：腾讯云计算（北京）有限责任公司

- GET Bucket Origin
- PUT Bucket Origin
- DELETE Bucket Origin

Object 接口

基本操作

- PUT Object
- PUT Object - Copy
- POST Object
- GET Object
- HEAD Object
- DELETE Object
- DELETE Multiple Objects
- OPTIONS Object
- POST Object restore
- APPEND Object
- SELECT Object Content

访问控制（acl）

- PUT Object acl
- GET Object acl

分块上传

- Initiate Multipart Upload
- Upload Part
- Upload Part - Copy
- Complete Multipart Upload
- Abort Multipart Upload
- List Multipart Uploads
- List Parts

对象标签

- PUT Object tagging
- GET Object tagging
- DELETE Object tagging

批量处理接口

- CreateJob
- DescribeJob
- ListJobs
- UpdateJobPriority
- UpdateJobStatus

公共元素

版权所有：腾讯云计算（北京）有限责任公司

- 错误响应
- 数据处理接口
  - 图片处理
    - 基础图片处理
      - 缩放
      - 裁剪
      - 旋转
      - 格式转换
      - 质量变换
      - 高斯模糊
      - 亮度
      - 对比度
      - 锐化
      - 灰度图
      - 图片水印
      - 文字水印
      - 获取图片基本信息
      - 获取图片 EXIF
      - 获取图片主色调
      - 去除元信息
      - 快速缩略模板
      - 限制图片大小
      - 管道操作符
    - 图片高级压缩
    - 图片处理机制介绍
    - 图片压缩
      - WebP 压缩
      - HEIF 压缩
      - TPG 压缩
      - AVIF 压缩
      - SVG 压缩
    - 盲水印
  - AI 内容识别
    - 队列接口
      - 查询队列
      - 更新队列
    - 服务配置
      - 开通 AI 内容识别

版权所有：腾讯云计算（北京）有限责任公司

查询 AI 内容识别
媒体处理
媒体 bucket 接口
查询媒体处理开通状态
开通媒体处理
媒体截图接口
获取媒体文件截图
获取媒体文件截图（万象域名）
媒体信息接口
获取媒体文件信息
私有 M3U8 接口
获取私有M3U8
处理方式接口
文档处理
查询文档预览开通状态
文档转码
同步请求接口
异步处理任务接口
提交文档转码任务
查询指定的文档转码任务
拉取符合条件的文档转码任务
异步处理队列接口
查询文档转码队列
更新文档转码队列
支持的功能和效果
文档转 HTML
快速入门
自定义配置概述
配置项说明
高级功能
支持的字体和动画
更新日志
文件处理
文件哈希值计算
哈希值计算同步请求
提交哈希值计算任务
查询哈希值计算结果
文件解压

版权所有：腾讯云计算（北京）有限责任公司

提交文件解压任务
查询文件解压结果
多文件打包压缩
提交多文件打包压缩任务
查询多文件打包压缩结果
任务与工作流
公共请求头部
公共响应头部
错误码
工作流接口
创建工作流
查询工作流
删除工作流
更新工作流
测试工作流
工作流回调
工作流实例
获取工作流实例详情
获取工作流实例详情列表
任务接口
媒体处理
取消媒体处理任务
CancelMediaJob
查询媒体处理任务
DescribeMediaJob
DescribeMediaJobs
媒体处理任务回调
视频转动图
音视频拼接
添加数字水印
提取数字水印
获取媒体信息
降噪
视频质量评分
SDRtoHDR
转封装（音视频分段）
智能封面
截帧

版权所有：腾讯云计算（北京）有限责任公司

流分离
超分辨率
音视频转码
语音合成
精彩集锦
视频增强
视频标签
人声分离
图片处理
提交图片处理任务
CreatePicJobs
取消图片处理任务
CancelPicJob
查询图片处理任务
DescribePicJob
DescribePicJobs
图片处理回调
多任务处理
提交多任务处理
MultiJobs
AI 内容识别
提交内容识别任务
提交翻译任务
提交分词任务
取消内容识别任务
查询内容识别任务
查询内容识别任务
内容识别任务回调
翻译任务回调
分词任务回调
同步媒体处理
媒体截图接口
获取媒体文件截图
模板接口
媒体处理
创建媒体处理模板
创建动图模板
创建拼接模板

版权所有：腾讯云计算（北京）有限责任公司

创建极速高清转码模板
创建截图模板
创建超分辨率模板
创建音视频转码模板
创建转码 pro 模板
创建语音合成模板
创建精彩集锦模板
创建视频增强模板
创建人声分离模板
创建水印模板
创建智能封面模板
删除媒体处理模板
DeleteMediaTemplate
查询媒体处理模板
DescribeMediaTemplates
更新媒体处理模板
更新动图模板
更新拼接模板
更新极速高清转码模板
更新截图模板
更新超分辨率模板
更新音视频转码模板
更新转码 pro 模板
更新语音合成模板
更新精彩集锦模板
更新视频增强模板
更新人声分离模板
更新水印模板
更新智能封面模板
AI 内容识别
创建 AI 内容识别模板
创建语音识别模板
查询 AI 内容识别模板
DescribeTemplates
更新 AI 内容识别模板
更新语音识别模板
批量任务接口
触发任务（独立节点）

版权所有：腾讯云计算（北京）有限责任公司

触发任务（工作流）
取消任务
查询任务
批量拉取任务
回调内容
工作流回调内容
任务回调内容
附录
结构体
内容审核接口
图片审核
图片单次审核
图片批量审核
查询图片审核任务结果
图片审核回调内容
图片审核结果反馈
视频审核
提交视频审核任务
查询视频审核任务结果
视频审核回调内容
音频审核
提交音频审核任务
查询音频审核任务结果
音频审核回调内容
文本审核
文本内容同步审核
文档审核
提交文档审核任务
查询文档审核任务结果
文档审核回调内容
云查毒接口
提交病毒检测任务

版权所有：腾讯云计算（北京）有限责任公司

API 文档

简介

最近更新时间：2023-12-20 11:04:19

腾讯云对象存储 COS 使用 XML API，这是一种轻量级的、无连接状态的接口，调用此接口您可以直接通过 HTTP/HTTPS 发出请求和接受响应，实现与腾讯云对象存储后台的交互操作。
由于使用了不同的数据传输框架，对象存储 COS 提供了独立于云 API 的接口和独立的 SDK，您可直接前往 COS 的 API 操作列表 了解详情，或前往 COS 的 SDK 列表 下载您需要的 SDK。云 API 的指南和对应的 SDK 不包含对象存储 COS 的操作功能。

注意

如您已开始使用腾讯云 COS API，即代表您已阅读并同意 《腾讯云服务协议》 和 《腾讯云对象存储服务等级协议》。
COS 的可用地域（Region）的详细信息请查阅 地域和访问域名 文档。
在使用 API 或 SDK 发起请求前，建议您阅读 创建请求概述 文档了解发起访问的域名、安全鉴权概念以及内外网访问检查等信息。
其他腾讯云产品的 API 格式，请参见 API 中心介绍。

API 概览

COS 支持的 API 接口请参见 操作列表。

术语信息

使用 API 接口时会出现一些主要概念和术语，请见下表：

名称	描述
APPID	开发者访问 COS 服务时拥有的用户维度唯一资源标识，用以标识资源，可在 API 密钥管理 页面获取
SecretId	开发者拥有的项目身份识别 ID，用于身份认证，可在 API 密钥管理 页面获取
SecretKey	开发者拥有的项目身份密钥，可在 API 密钥管理 页面获取
Bucket	存储桶，COS 中用于存储数据的容器。有关存储桶的进一步说明，请参见 存储桶概述 文档

版权所有：腾讯云计算（北京）有限责任公司

BucketName-APPID	存储桶名称格式，用户在使用 API、SDK 时，需要按照此格式填写存储桶名称。例如 examplebucket-1250000000，含义为该存储桶 examplebucket 归属于 APPID 为 1250000000的用户
Object	对象，COS 中存储的具体文件，是存储的基本实体
ObjectKey	对象键，对象（Object）在存储桶（Bucket）中的唯一标识。有关对象与对象键的进一步说明，请参见 对象概述 文档
Region	地域信息，枚举值可参见 可用地域 文档，例如：ap-beijing、ap-hongkong、eu-frankfurt 等
ACL	访问控制列表（Access Control List），是指特定 Bucket 或 Object 的访问控制信息列表
CORS	跨域资源共享（Cross-Origin Resource Sharing），指发起请求的资源所在域不同于该请求所指向资源所在的域的 HTTP 请求
Multipart Uploads	分块上传，腾讯云 COS 服务为上传文件提供的一种分块上传模式
Object Content	Object Content 是上传文件的二进制内容

快速入门

要使用腾讯云对象存储 API，需要先执行以下步骤：

1. 在腾讯云 对象存储控制台 开通腾讯云对象存储（COS）服务。
2. 在腾讯云 对象存储控制台 创建一个 Bucket。
3. 在访问管理控制台中的 API 密钥管理 页面里获取 APPID，并创建 SecretId、SecretKey。
4. 编写一个请求签名算法程序（或使用任何一种服务端 SDK），详情请参见 请求签名 文档。
5. 计算签名，调用 API 执行操作。

版权所有：腾讯云计算（北京）有限责任公司

公共请求头部

最近更新时间：2023-12-20 11:07:26

描述

此篇文档将为您介绍在使用 API 时候会使用到的公共请求头部（Request Header），下文提到的头部在之后的具体 API 文档中不再赘述。

请求头部列表

Header 名称	描述	类型	是否必选
Authorization	携带鉴权信息，用以验证请求合法性的签名信息。针对公有读的对象可不携带此头部，如通过请求参数传递鉴权信息也无需携带此头部，详情请参见 请求签名 文档。	string	是。针对公有读的对象或通过请求参数传递鉴权信息，此头部为可选项。
Content-Length	RFC 2616中定义的 HTTP 请求内容长度（字节）。	integer	针对 PUT 和 POST 请求（指定 Transfer-Encoding 请求头部的 PUT Object 请求除外），此头部是必选项。 针对 GET、HEAD、DELETE 和 OPTIONS 请求，不能指定此头部。
Content-Type	RFC 2616中定义的 HTTP 请求内容类型（MIME）例如application/xml或image/jpeg。	string	针对有请求体的 PUT 和 POST 请求，此头部是必选项。 针对 GET、HEAD、DELETE 和 OPTIONS 请求，不能指定此头部。
Content-MD5	RFC 1864中定义的请求体内容的16字节二进制 MD5 哈希值的 Base64 编码形式，用于完整性检查，验证请求体在传输过程中是否发生变化，最终的取值长度应为24个字符，请注意在编写代码时使用正确的方法和参数，例如ZzD3iDJdrMAAb00lgLLeig==。	string	针对有请求体的 PUT 和 POST 请求（除 POST Object），此头部是必选项。 针对 GET、HEAD、DELETE 和 OPTIONS 请求，不能指定此头部。
Date	RFC 1123中定义的 GMT 格式当前时间，例如Wed, 29 May 2019 04:10:12 GMT。	string	否。

版权所有：腾讯云计算（北京）有限责任公司

Host	请求的主机，形式为<BucketName-APPID>.cos.<Region>.myqcloud.com	string	是。
x-cos-security-token	使用临时安全凭证时需要传入的安全令牌字段，详情请参见 临时安全凭证 相关说明。	string	否。当使用临时密钥并通过 Authorization 携带鉴权信息时，此头部为必选项。

服务端加密专用头部

对于支持服务端加密（Server Side Encryption，SSE）的接口，根据不同的加密方式适用如下请求头部，请参阅具体的接口文档确定是否适用 SSE。下列头部的是否必选仅针对使用 SSE 的场景，如果请求不支持 SSE 的接口或不使用 SSE，则以下头部均无需携带。详情请参见 服务端加密概述。

SSE-COS

Header 名称	描述	类型	是否必选
x-cos-server-side-encryption	服务端加密算法，使用 SSE-COS 时指定为：AES256	string	上传或复制对象（包括简单上传/复制与分块上传/复制）时必选，下载对象时不能指定此头部

SSE-KMS

Header 名称	描述	类型	是否必选
x-cos-server-side-encryption	服务端加密算法，使用 SSE-KMS 时指定为：cos/kms	string	上传或复制对象（包括简单上传/复制与分块上传/复制）时必选，下载对象时不能指定此头部
x-cos-server-side-encryption-cos-kms-key-id	当 x-cos-server-side-encryption 值为 cos/kms 时，用于指定 KMS 的用户主密钥 CMK，如不指定，则使用 COS 默认创建的 CMK，更多详细信息可参见 SSE-KMS 加密	string	否
x-cos-server-side-encryption-context	当 x-cos-server-side-encryption 值为 cos/kms 时，用于指定加密上下文，值为 JSON 格式加密上下文键值对的 Base64 编码。	string	否

版权所有：腾讯云计算（北京）有限责任公司

	例 如 `eyJhIjoiYXNkZmEiLCJiIjoiMTIzMzIxIn0=`		

SSE-C

Header 名称	描述	类型	是否必选
x-cos-server-side-encryption-customer-algorithm	服务端加密算法，目前仅支持 AES256	string	是
x-cos-server-side-encryption-customer-key	服务端加密密钥的 Base64 编码， 例如 `MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=`	string	是
x-cos-server-side-encryption-customer-key-MD5	服务端加密密钥的 MD5 哈希值，使用 Base64 编码， 例如 `U5L61r7jcwdNvT7frmUG8g==`	string	是

公共响应头部

最近更新时间：2023-12-20 11:05:48

描述

此篇文档将为您介绍在使用 API 时会出现的公共响应头部（Response Header），下文提到的头部在后续的具体 API 文档中不再赘述。

响应头部列表

Header 名称	描述	类型
Content-Length	RFC 2616 中定义的 HTTP 响应内容长度（字节）。	string
Content-Type	RFC 2616 中定义的 HTTP 响应内容类型（MIME）。	string
Connection	RFC 2616 中定义，表明响应完成后是否会关闭网络连接。枚举值：keep-alive，close。	enum
Date	RFC 1123 中定义的 GMT 格式服务端响应时间，例如Wed, 29 May 2019 04:10:12 GMT。	string
ETag	ETag 全称为 Entity Tag，是对象被创建时标识对象内容的信息标签，可用于检查对象的内容是否发生变化，例如"8e0b617ca298a564c3331da28dcb50df"。此头部并不一定返回对象的 MD5 值，而是根据对象上传和加密方式而有所不同。如果您需要校验上传到云端的文件和本地文件的一致性，我们推荐您使用 x-cos-hash-crc64ecma 头部中的 CRC64编码进行校验。	string
Last-Modified	对象的最近一次上传的时间，例如Fri, 10 Apr 2020 18:17:25 GMT。	string
Server	接收请求并返回响应的服务器的名称，默认值：tencent-cos。	string
Transfer-Encoding	RFC 2616 中定义的传输编码格式。	string
x-cos-hash-crc64ecma	对象的 CRC64 值，详情请参见 CRC64 校验 文档。	integer
x-cos-request-id	每次请求发送时，服务端将会自动为请求生成一个 ID。	string
x-cos-trace-id	每次请求出错时，服务端将会自动为这个错误生成一个 ID。仅当请求出错	string

版权所有：腾讯云计算（北京）有限责任公司

	时才会在响应中包含此头部。	

服务端加密专用头部

对于支持服务端加密（Server Side Encryption，SSE）且在请求中使用了服务端加密的接口，将根据具体的加密类型返回以下响应头，详情请参见 服务端加密概述。

SSE-COS

Header 名称	描述	类型
x-cos-server-side-encryption	当上传对象时使用了 SSE-COS，或者下载经过 SSE-COS 加密的对象时，请求的响应均会返回此头部，用于表明对象上传时使用的服务端加密算法。	string

SSE-KMS

Header 名称	描述	类型
x-cos-server-side-encryption	当上传对象时使用了 SSE-KMS，或者下载经过 SSE-KMS 加密的对象时，请求的响应均会返回此头部，用于表明对象上传时使用的服务端加密算法。	string
x-cos-server-side-encryption-cos-kms-key-id	返回 KMS（密钥管理系统）的用户主密钥 CMK。如未指定，则返回 COS 默认创建的 CMK。	string

SSE-C

Header 名称	描述	类型
x-cos-server-side-encryption-customer-algorithm	当上传对象时使用了 SSE-C，或者下载经过 SSE-C 加密的对象时，请求的响应均会返回此头部，用于表明对象上传时使用的服务端加密算法。	string
x-cos-server-side-encryption-customer-key-MD5	对象上传时使用的服务端加密密钥的 MD5 哈希值，使用 Base64 编码，例如 `U5L61r7jcwdNvT7frmUG8g==` 。	string

错误码

最近更新时间：2023-12-20 14:50:18

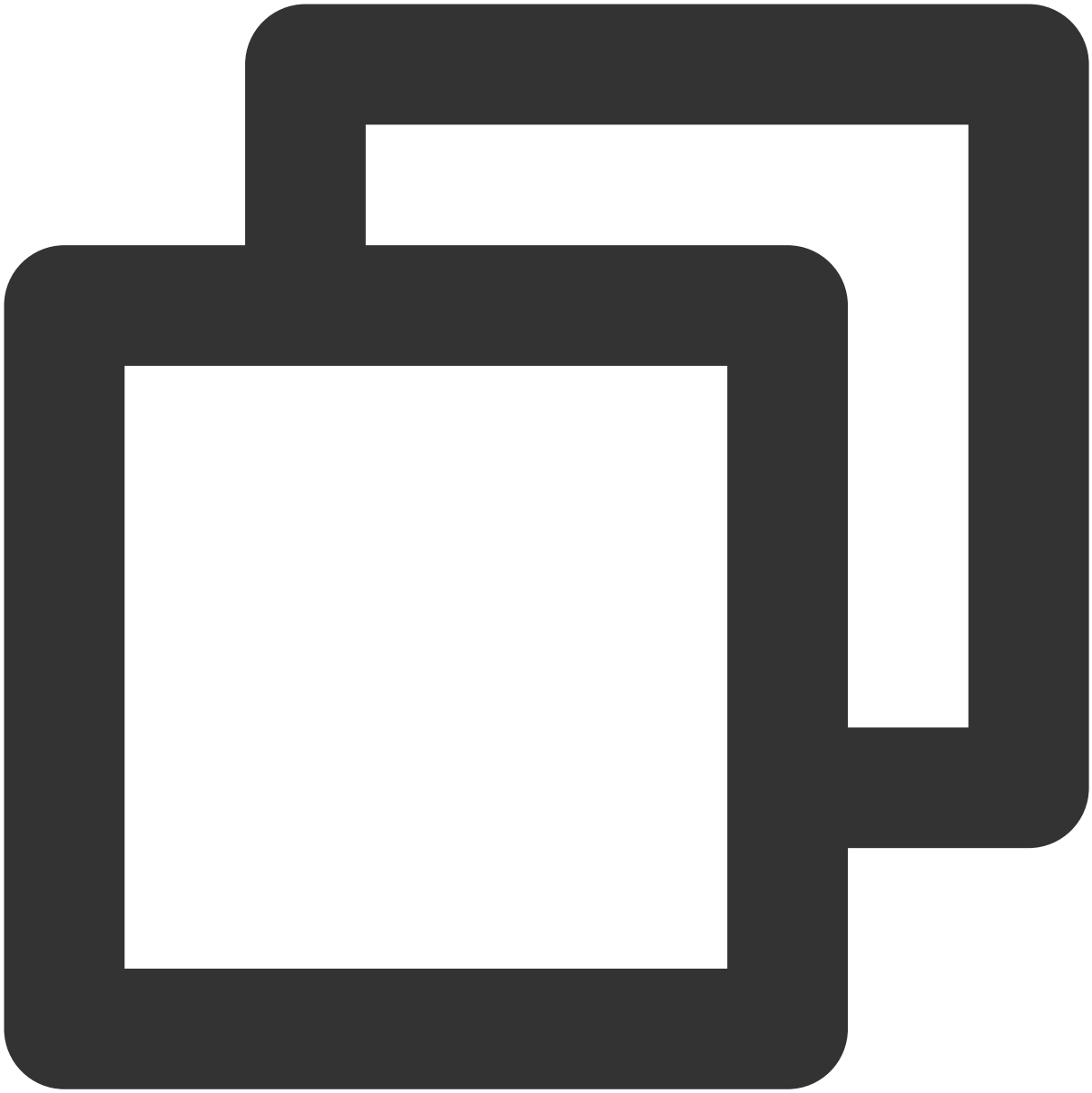
概述

本文将为您介绍请求出错时返回的错误码和对应错误信息。

错误响应

Content-Type：application/xml
对应 HTTP 状态码：3XX，4XX，5XX。对于 PUT Object - Copy 接口，即使 HTTP 状态码为200也有可能在响应体中包含错误。

响应体



```
<?xml version='1.0' encoding='utf-8' ?>
<Error>
            <Code>string</Code>
            <Message>string</Message>
            <Resource>string</Resource>
            <RequestId>string</RequestId>
            <TraceId>string</TraceId>
</Error>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
Error	无	包含所有的错误信息	Container

Container 节点 Error 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Error	错误码，用来定位唯一的错误条件和确定错误场景，具体错误码见下文	string
Message	Error	具体的错误信息	string
Resource	Error	请求的资源，存储桶地址或对象地址	string
RequestId	Error	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助 COS 定位问题	string
TraceId	Error	每次请求出错时，服务端将会自动为这个错误生成一个 ID，遇到问题时，该 ID 能更快地协助 COS 定位问题	string

错误码列表

4XX 类型错误

HTTP 状态码	错误码	描述
400 Bad Request	ActionAccelerateNotSupported	加速域名不支持该操作
400 Bad Request	AttachmentFull	ACL 和 Policy 数量到达上限。详情请参见 规格与限制
400 Bad Request	BadDigest	提供的 Content-MD5 值与服务端收到的请求体的 MD5 哈希值不一致
400 Bad Request	BadRequest	参数错误
400 Bad Request	BucketAccelerateNotEnabled	该存储桶未启用加速域名
400 Bad Request	BucketNameTooLong	存储桶名称过长。详情请参见 存储桶命名规范

400 Bad Request	BucketVersionNotOpen	存储桶未启用版本控制
400 Bad Request	DNSRecordVerifyFailed	DNS 记录验证失败，请添加 CNAME 或 TXT 记录。DNS 记录可能需要最多10分钟生效
400 Bad Request	EntitySizeNotMatch	请求体大小与 Content-Length 请求头不符
400 Bad Request	EntityTooLarge	上传的对象大小超过规定的最大值。详情请参见 规格与限制
400 Bad Request	EntityTooSmall	上传的对象大小不足规定的最小值，常见于分块上传。详情请参见 规格与限制
400 Bad Request	ExpiredToken	临时密钥 Token 已过期
400 Bad Request	ImageResolutionExceed	图片分辨率超出限制或动图帧数过多
400 Bad Request	ImageTooLarge	图片超过限制大小
400 Bad Request	IncompleteBody	请求体大小小于 Content-Length 请求头
400 Bad Request	IncorrectNumberOfFilesInPostRequest	POST Object 请求每次只允许上传一个对象
400 Bad Request	InvalidArgument	请求参数不合法，请确认是否允许携带该请求参数
400 Bad Request	InvalidBucketName	存储桶名称不合法。详情请参见存储桶命名规范
400 Bad Request	InvalidCopySource	复制对象源不合法
400 Bad Request	InvalidDelimiter	分隔符（delimiter）参数不合法，分隔符只能为一个字符
400 Bad Request	InvalidDigest	给定的 Content-MD5 值不合法
400 Bad Request	InvalidImageFormat	图片格式不合法
400 Bad Request	InvalidImageSource	图片源不合法
400 Bad Request	InvalidLocationConstraint	指定的 location 不合法
400 Bad Request	InvalidObjectName	对象名称不合法。详情请参见 对象键
400 Bad Request	InvalidPart	分块缺失

版权所有：腾讯云计算（北京）有限责任公司

400 Bad Request	InvalidPartOrder	分块的编号不连续
400 Bad Request	InvalidPicOperations	Pic-Operations 请求头不合法
400 Bad Request	InvalidPolicyDocument	POST Object 请求中的策略（Policy）不合法
400 Bad Request	InvalidRegionName	不合法的地域名。详情请参见 地域和访问域名
400 Bad Request	InvalidRequest	请求不合法
400 Bad Request	InvalidSHA1Digest	请求内容 SHA1 校验不合法
400 Bad Request	InvalidTag	存储桶标签不合法。详情请参见 存储桶标签概述
400 Bad Request	InvalidTargetBucketForLogging	用于存放日志的目标存储桶不合法，目标存储桶必须与当前存储桶在同一个地域
400 Bad Request	InvalidUploadStatus	当启用版本控制时不能使用 JSON API 上传对象，请使用 XML API
400 Bad Request	InvalidURI	URI 不合法
400 Bad Request	InventoryFull	清单任务数量已达到限制。清单任务上限1000条
400 Bad Request	JsonAPINotSupportOnMAZBucket	JSON API 不支持操作多 AZ 存储桶，请使用 XML API
400 Bad Request	KeyTooLong	对象键过长。详情请参见 对象键
400 Bad Request	KmsException	密钥管理服务异常
400 Bad Request	KmsKeyDisabled	提供的密钥已被禁用
400 Bad Request	KmsKeyNotExist	提供的密钥不存在
400 Bad Request	ListPartUploadIdIsEmpty	UploadId 为空
400 Bad Request	LoggingConfExists	日志配置已存在
400 Bad Request	LoggingPrefixInvalid	日志前缀参数不合法
400 Bad Request	MalformedPolicy	策略格式不合法
400 Bad Request	MalformedPOSTRequest	该 POST 请求的请求体内容不合法

版权所有：腾讯云计算（北京）有限责任公司

400 Bad Request	MalformedXML	请求体的 XML 格式不符合 XML 语法
400 Bad Request	MAZOperationNotSupportOnOAZBucket	单可用区（单 AZ）存储桶不支持多可用区（多 AZ）操作。关于多 AZ 特性及使用限制的说明请参见 多 AZ 特性概述 The multiple availability zones operation is not supported by single availability zone bucket
400 Bad Request	MissingRequestBodyError	请求体缺失
400 Bad Request	MultiAZFeatureNotSupport	当前地域不支持多可用区
400 Bad Request	MultiBucketNotSupport	跨地域复制只能设置一个目的存储桶
400 Bad Request	NotifyRuleEventConflict	通知规则 Event 冲突
400 Bad Request	NotifyRulePrefixConflict	通知规则 Prefix 冲突
400 Bad Request	NotifyRuleSuffixConflict	通知规则 Suffix 冲突
400 Bad Request	NotSupportedStorageClass	指定的 存储类型 不支持
400 Bad Request	OAZOperationNotSupportOnMAZBucket	多可用区存储桶不支持单可用区操作。关于多 AZ 特性及使用限制的说明请参见 多 AZ 特性概述 The single availability zone operation is not supported by multiple availability zones bucket
400 Bad Request	PolicyFull	ACL 和 Policy 数量到达上限。详情请参见 规格与限制
400 Bad Request	PolicyVersionFull	Policy 版本数量到达上限
400 Bad Request	RequestTimeout	请求超时
400 Bad Request	SsecDecryptHeaderInvalid	源文件使用 SSE-C 加密，需要在请求头中提供相同的密钥
400 Bad Request	SSEContentNotSupported	加密方式不支持
400 Bad Request	SSEHeaderNotAllowed	该操作不支持指定的服务端加密头部
400 Bad Request	TargetBucketNameInvalid	目标存储桶名称不合法。详情请参见存储桶 命名规范
400 Bad Request	TooManyBuckets	存储桶数目达到上限200

版权所有：腾讯云计算（北京）有限责任公司

400 Bad Request	UnexpectedContent	请求不支持相关内容
400 Bad Request	UserCnameInvalid	用户指定的 CNAME 不存在或不合法
400 Bad Request	UserNetworkTooSlow	用户的网络速度过慢
400 Bad Request	VerifyAlgorithmNotSupported	校验算法不支持
400 Bad Request	WebsiteURLInvalid	自定义域名 URL 不合法
400 Bad Request	XMLSizeLimit	XML 长度超过限制
402 Payment Required	PaymentRequired	超出费用配额限制
403 Forbidden	AccessDenied	签名或者权限不正确，拒绝访问。请参考 请求签名 和 ACL 概述
403 Forbidden	AccessForbidden	跨域资源共享（CORS）请求被拒绝，请检查请求方法或请求头部中的 Origin、Access-Control-Request-Method、Access-Control-Request-Headers 是否在配置的 CORS 白名单中
403 Forbidden	InvalidAccessKeyId	SecretID 不存在
403 Forbidden	InvalidObjectState	对象存储类型与操作请求冲突
403 Forbidden	NoProcessAuthority	没有处理图片的权限
403 Forbidden	RequestTimeTooSkewed	本地时间与服务器时间相差过大，超过15分钟
403 Forbidden	Request has expired	发起请求的时间超过了签名的有效时间，或者本地系统时间和所在时区的时间不一致。详情请参见 常见问题
403 Forbidden	SignatureDoesNotMatch	客户端计算的签名与 COS 服务端计算的签名不一致。请参阅 请求签名 文档并借助COS 请求签名工具 检查自行实现的签名过程
403 Forbidden	UserNotSourceBucketOwner	当前用户不是源存储桶的拥有者
403 Forbidden	UserNotTargetBucketOwner	当前用户不是目标存储桶的拥有者
404 Not Found	InventoryConfigurationNotFoundError	清单配置未找到
404 Not Found	NoBucketQuotaPolicy	存储桶配额策略不存在

版权所有：腾讯云计算（北京）有限责任公司

404 Not Found	NoSuchBucket	指定的存储桶不存在
404 Not Found	NoSuchCopySource	复制对象源不存在
404 Not Found	NoSuchCORSConfiguration	指定的跨域资源共享配置不存在
404 Not Found	NoSuchEncryptionConfiguration	指定的存储桶加密配置不存在
404 Not Found	NoSuchJob	指定的批量处理任务不存在
404 Not Found	NoSuchKey	指定的对象键不存在。检查对象是否存在请使用 HEAD Object 接口
404 Not Found	NoSuchLifecycleConfiguration	指定的生命周期配置不存在
404 Not Found	NoSuchObjectLockConfiguration	指定的对象锁定配置不存在
404 Not Found	NoSuchPolicyVersion	指定的策略版本不存在
404 Not Found	NoSuchTagSet	指定的标签集合不存在
404 Not Found	NoSuchUpload	分块上传时指定的 UploadId 不存在
404 Not Found	NoSuchVersion	指定版本不存在
404 Not Found	NoSuchWebsiteConfiguration	指定的静态网站配置不存在
404 Not Found	OriginConfigurationNotFoundError	回源配置未找到
404 Not Found	ReplicationConfigurationNotFoundError	跨地域复制配置未找到
405 Method Not Allowed	MethodNotAllowed	此资源不支持该 HTTP 方法
405 Method Not Allowed	RestoreNonArchiveObject	不允许对非归档对象进行回热（恢复）
405 Method Not Allowed	UploadIdNotSupported	指定的 UploadId 由 JSON API 生成，不能使用 XML API 操作
409 Conflict	AppendPositionErr	Append 操作时，对象长度和 Position 不一致
409 Conflict	BucketAlreadyExists	指定的存储桶已存在
409 Conflict	BucketAlreadyOwnedByYou	指定的存储桶已存在且由当前帐户创建
409 Conflict	BucketLocked	存储桶已锁定，不能操作跨地域复制或生命周期

版权所有：腾讯云计算（北京）有限责任公司

409 Conflict	BucketNotEmpty	存储桶不为空
409 Conflict	DomainConfigConflict	域名配置冲突，请删除冲突的记录
409 Conflict	InvalidBucketState	存储桶状态与操作请求冲突，例如版本控制管理与跨地域复制的冲突
409 Conflict	InvalidLockedTime	对象锁定时间无效
409 Conflict	ObjectLocked	对象已锁定，不允许覆盖上传或删除该对象，也不允许通过复制修改该对象的元数据
409 Conflict	InvalidObjectState	对象状态与操作请求冲突
409 Conflict	PathConflict	存在同名对象的毫秒级并发冲突
409 Conflict	QuotaConflict	配额冲突
409 Conflict	QuotaOperationConflict	该存储桶的状态不适用于该操作
409 Conflict	RecordAlreadyExist	DNS 记录冲突，请删除冲突的记录，或添加 CNAME/TXT 记录
409 Conflict	RestoreAlreadyInProgress	该对象正在回热（恢复）中
409 Conflict	UploadConflict	同一个对象键频繁上传会导致冲突
409 ObjectNotAppendable	ObjectNotAppendable	指定的对象不能追加
411 Length Required	MissingContentLength	Content-Length 请求头部缺失
412 Precondition Failed	PreconditionFailed	前置条件不匹配
416 Requested Range Not Satisfiable	InvalidRange	请求的对象范围不合法
451 Unavailable For Legal Reasons	DomainAuditFailed	域名未备案
451 Unavailable For Legal Reasons	UnavailableForLegalReasons	当前服务不可用，请检查账号是否欠费，若无欠费可通过 内容合规 查看是否存在违规内容

5XX 类型错误

HTTP 状态码	错误码	描述
500 Internal Server Error	InternalError	服务端内部错误
500 Internal Server Error	KmsInternalException	查询密钥管理服务时发生服务端内部错误
501 Not Implemented	NotImplemented	请求尚未实现
503 Service Unavailable	KmsFreqControl	请降低使用密钥管理服务的请求的访问频率
503 Service Unavailable	ServiceUnavailable	服务暂不可用，请重试
503 Service Unavailable	SlowDown	请降低访问频率

数据处理相关错误码

图片下载时处理错误码

错误码	说明	处理措施
-103	无效的请求；请求报文无法识别。	请检查请求内容。
-104	无效的 APPID，URL 中包含的 APPID 无效，或者域名没有和 APPID 绑定。	请检查 APPID 是否正确。
-105	无效的样式名，URL 中指定的样式名或者别名没有配置。	请检查样式配置或样式名内容。
-106	无效的 URL，URL 格式不符合格式要求。	请检查 URL 格式。
-107	无效的 Host 头域。	请检查 Host 是否正确。
-108	无效的 Referer。	请检查相关 Referer 配置。
-109	无效的样式名 ID。没有找到该样式名对应的图片。可能是上传该图片后，新增的样式名， 因此图片上传时不能生成该样式名对应的数据。	-
-110	该图片在黑名单中。	-
-113	该文档的类型不支持	文档预览目前所支持的输入文件类型，请参见 规则与限制
-120	回源到源站获取数据时，源站返回的数据有异常，无法正常获取到图片数据。	请检查源站数据。
-124	下载偏移错误。HTTP 请求的 Range 断点续传偏移量可能设置错误。	请检查断点续传偏移量的相关配置。

版权所有：腾讯云计算（北京）有限责任公司

-154	原图保护机制禁止该 URL 下载。	请使用样式访问。
-156	强制执行302流程。	-
-162	业务配置为欠费，禁止下载图片。	请及时充值。
-163	业务配置不存在。	请检查相关配置项。
-164	热图下载，限制。	请降低请求频率，或 提交工单 联系我们处理。
-165	攻击性下载请求，限制。	-
-166	消息格式错误。	-
-167	文件太大，无法支持图片下载。	请降低图片大小。
-168	文件太大，无法支持图片下载。	请降低图片大小。
-441	图片格式不符，无法进行压缩。	请使用数据万象支持的图片格式进行图片处理操作。具体支持格式为 jpg、bmp、gif、png、webp。
-442	图片超过限制大小。	请将图片长宽限制在9999像素以内。
-443	图片格式不满足处理限制。	请使用支持的图片格式进行处理，详情请参见 规则与限制
-447	图片分辨率超出限制或动图帧数过多。	请将图片长宽限制在9999像素以内（动图请限制帧数）。
-901	压缩超时。	-
-902	回源超时，镜像存储功能把请求转发到开发商源站，但没有收到响应，超时了。	请检查源站数据或回源配置。
-3006	该文档被加密，无法解析。	-
-3008	该文件为空文件，无法解析。	-
-3011	该文档的类型不支持。	文档预览目前所支持的输入文件类型，请参见 规则与限制
-3015	请求的页码不存在。	-
-3075	该文档超过100MB，无法解析。	输入文件大小需限制在100MB之内
-5062	图片涉嫌违禁。	请检查图片内容是否违规。

版权所有：腾讯云计算（北京）有限责任公司

-6101	图片不存在、图片没有上传或者图片已经删除。	请检查请求源数据。
-29033	下载请求无有效的 range。	-
-29034	下载的 offset 大于文件 size。	-
-29214	触发频控。	请降低请求频率，或 提交工单 联系我们处理。
-40168	原图为空数据。	请检查请求源数据。
-46152	非法的 bucketName。	请检查存储桶名称是否正确。
-46154	非法的 APPID。	请检查 APPID 是否正确。
-46614	文件尚未上传完成，禁止下载。	请等待文件上传成功后请求。
-46617	命中黑名单。	请检查图片内容是否违规。
-46618	签名校验失败。	请检查签名是否准确。
-46619	签名过期。	请更新签名。
-46620	Bucket 被封禁。	您的 Bucket 已被封禁，请 提交工单 联系我们处理。
-46621	文件被封禁。	您的资源已被封禁，请 提交工单 联系我们处理。
-46627	用户目前在黑名单中。	您的账号已被封禁，请 提交工单 联系我们处理。
-46628	文件不存在。	请确认请求下载的文件是否真实存在。
-46642	APPID 不存在。	请检查 APPID 是否正确。
-46646	文件已封禁。	文件已被封禁，请检查文件内容是否违规。
-46661	无权访问	请联系资源所有者授予权限。

图片持久化处理错误码

错误码	说明	处理措施
-60998	图片处理参数无效。	请检查图片处理规则是否正确。
-60997	图片处理规则过多。	持久化处理当前最多支持五条规则，请控制在五条以内。

-60996	文件太大，无法处理。	请将文件控制在32MB以内。
-60987	图片获取失败。	请检查图片 URL。
-60983	无效的 Host。	请检查 Host。
-60972	无效地域。	请检查请求地域是否支持数据万象服务。数据万象现有地域详见地域与域名。
-60967	图片样式不存在。	请检查图片样式。
-60957	样式数量超过上限。	当前样式数量上限为100条，如需更多样式请 提交工单 联系我们。
-60955	无效的 URL。	请检查 URL 是否正确。
-60950	未指定文件。	请在请求中指定需要处理的文件。
-60949	请求频率过高。	请放缓请求频率。如有需要请 提交工单 联系我们提升频率限制。
-60948	账号涉嫌违规。	您的 APPID 已被封禁，请 提交工单 联系我们处理。
-60938	签名信息不完整。	请检查签名信息中的必选项。
-60936	请求被拒绝。	AccessDenied。
-60932	签名错误。	签名无效，请检查签名信息是否匹配。
-62999	无效的图片格式。	请检查图片格式是否正确。

文档预览错误码

错误码	说明	处理措施
-3006	文档为加密文件	建议使用已解密的文档。
-3008	文档长度为0	请确认文档是否正常。
-3011	文档类型不支持	请确认文档是否在支持范围内，支持的文档类型详情请参见 规则与限制。
-3064	文件解析失败	文档解析失败，请检查资源是否损坏。
-3070	文件处理超时失败	任务处理超时失败，建议调整单次任务生成范围。
-3075	文档大小超过限制	文档资源限制，需要调整文档大小。

内容审核错误码

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明	处理措施
-61011	文件不存在。	请检查文件是否存在。
-65007	文件内容错误，解码失败。	请检查文件格式是否正确。
-65010	下载文件失败。	请检查文件链接是否正确，私有读文件需要携带签名。
-65011	文件过大，无法处理。	当前不支持审核过大的文件。
-65012	图片分辨率过小，无法处理。	当前不支持分辨率过小的图片。
-65013	不符合要求（例如不支持识别的文件后缀、文件过大等），未进行处理。	请检查文件是否符合规则。
-65014	没有数据万象角色 CI_QCSRole。	请绑定数据万象角色。
-65015	未启用审核场景。	请使用至少一种审核场景进行审核。
-65016	使用的 BizType 不存在。	请填写正确的 BizType。
其他	内部错误	无。

版权所有：腾讯云计算（北京）有限责任公司

请求签名

最近更新时间：2023-01-29 15:08:38

注意：

- 此文档仅适用于 COS XML 版本。
- 此文档不适用于 POST Object 的 HTTP 请求。

简介

使用腾讯云对象存储（Cloud Object Storage，COS）时，可通过 RESTful API 对 COS 发起 HTTP 匿名请求或 HTTP 签名请求，对于签名请求，COS 服务器端将会进行对请求发起者的身份验证。

- 匿名请求：HTTP 请求不携带任何身份标识和鉴权信息，通过 RESTful API 进行 HTTP 请求操作。
- 签名请求：HTTP 请求时携带签名，COS 服务器端收到消息后，进行身份验证，验证成功则可接受并执行请求，否则将会返回错误信息并丢弃此请求。

COS 基于密钥 HMAC（Hash Message Authentication Code）的自定义方案进行身份验证。

SDK 签名实现

COS SDK 已经实现签名，在使用 SDK 发起请求或通过 SDK 获取签名时无需关注签名问题。如果您需要自行实现签名或了解各语言的具体实现，可参考各语言 SDK 的签名实现文件：

SDK	签名实现文件
Android SDK	COSXmlSigner.java
C SDK	cos_auth.c
C++ SDK	auth_tool.cpp
.NET(C#) SDK	IQCloudSigner.cs (class CosXmlSigner)
Go SDK	auth.go
iOS SDK	QCloudAuthentationV5Creator.m

SDK	签名实现文件
Java SDK	COSSigner.java
JavaScript SDK	util.js (getAuth)
Node.js SDK	util.js (getAuth)
PHP SDK	Signature.php
Python SDK	cos_auth.py
小程序 SDK	util.js (getAuth)

生成签名链接

COS 目前支持的各语言 SDK 都提供了生成签名链接的功能，生成的链接 URL 中附带有一定有效期的签名，且该签名支持 PUT 请求或 GET 请求，因此您可以直接使用该链接上传或下载对象而不需要额外生成签名。

- 在生成用于上传的签名链接时，还可以指定 Content-Type 或 Content-MD5 等头部，以便限制上传的媒体类型或限制上传的内容必须为指定内容。有关上传时的请求头部设置，请参阅 PUT Object 文档。
- 在生成用于下载的签名链接时，还可以指定 response-xxx 请求参数，以便在下载时临时修改响应中的相关头部。有关下载时的请求参数设置，请参阅 GET Object 文档。

各语言 SDK 生成签名链接的详细介绍，请参见下列各语言 SDK 文档：

说明：

- 建议用户使用临时密钥生成预签名，通过临时授权的方式进一步提高预签名上传、下载等请求的安全性。申请临时密钥时，请遵循 最小权限指引原则，防止泄漏目标存储桶或对象之外的资源。
- 如果您一定要使用永久密钥来生成预签名，建议永久密钥的权限范围仅限于上传或下载操作，以规避风险。

SDK	预签名文档
Android SDK	生成预签名链接
C SDK	预签名 URL
C++ SDK	预签名 URL

SDK	预签名文档
.NET(C#) SDK	生成预签名链接
Go SDK	预签名 URL
iOS SDK	生成预签名链接
Java SDK	预签名 URL
JavaScript SDK	预签名 URL
Node.js SDK	预签名 URL
PHP SDK	预签名 URL
Python SDK	预签名 URL
小程序 SDK	预签名 URL

签名使用场景

在 COS 服务使用的场景中，对于需要对外发布类的数据，通常可将对象设置为公有读私有写。即所有人可查看，通过 ACL 策略指定账号可写入。此时，可将 ACL 策略与 API 请求签名相结合，对访问进行身份验证，并对操作进行权限和有效期的控制。

> 注意：
> 本文所描述的 API 请求签名，如果您使用 SDK 进行开发，则已包含在内。**仅在您希望通过原始 API 进行二次开发时，需要根据本文所描述步骤进行操作**。

在以上场景中，可对 API 请求进行多方面的安全防护：

1. **请求者身份验证**。通过访问者唯一 ID 和密钥确定请求者身份。
2. **防止传输数据篡改**。对数据签名并检验，保障传输内容完整性。
3. **防止签名被盗用**。对签名设置时效，避免签名盗用并重复使用。

准备工作

1. APPID、SecretId 和 SecretKey。
   在访问管理控制台的 API 密钥管理 页面中获取。
2. 确定开发语言：
   支持但不限于 Java、PHP、.NET、C++、Node.js、Python，根据不同的开发语言，确定对应的 HMAC-SHA1、SHA1 和 UrlEncode 函数。
   其中，HMAC-SHA1 和 SHA1 函数以 UTF-8 编码字符串为输入，以16进制小写字符串为输出。UrlEncode 基于 UTF-8 编码，此外对于 ASCII 范围内的可打印字符，下列特殊符号也应被编码：

字符	十进制	十六进制	字符	十进制	十六进制
(空格)	32	20	;	59	3B
!	33	21	<	60	3C
"	34	22	=	61	3D
#	35	23	>	62	3E
$	36	24	?	63	3F
%	37	25	@	64	40
&	38	26	[	91	5B
'	39	27	\	92	5C
(	40	28	]	93	5D
)	41	29	^	94	5E
*	42	2A	`	96	60
+	43	2B	{	123	7B
,	44	2C	|	124	7C
/	47	2F	}	125	7D
:	58	3A	无	无	无

签名步骤

步骤1：生成 KeyTime

1. 获取当前时间对应的 Unix 时间戳 StartTimestamp，Unix 时间戳是从 UTC（协调世界时，或 GMT 格林威治时间）1970年1月1日0时0分0秒（北京时间 1970年1月1日8时0分0秒）起至现在的总秒数。
2. 根据上述时间戳和期望的签名有效时长算出签名过期时间对应的 Unix 时间戳 EndTimestamp。
3. 拼接签名有效时间，格式为 `StartTimestamp;EndTimestamp`，即为 KeyTime。例如：`1557902800;1557910000`。

步骤2：生成 SignKey

使用 HMAC-SHA1 以 SecretKey 为密钥，以 KeyTime 为消息，计算消息摘要（哈希值，16进制小写形式），即为 SignKey，例如：`eb2519b498b02ac213cb1f3d1a3d27a3b3c9bc5f`。

步骤3：生成 UrlParamList 和 HttpParameters

1. 遍历 HTTP 请求参数，生成 key 到 value 的映射 Map 及 key 的列表 KeyList：

- key 使用 UrlEncode 编码并转换为小写形式。
- value 使用 UrlEncode 编码。若无 value 的参数，则认为 value 为空字符串。例如请求路径为 `/?acl`，则认为是 `/?acl=`。

> 说明：
> HTTP 请求参数，即请求路径中 `?` 以后的部分，例如请求路径为 `/?versions&prefix=example-folder%2F&delimiter=%2F&max-keys=10`，则请求参数为 `versions&prefix=example-folder%2F&delimiter=%2F&max-keys=10`。

2. 将 KeyList 按照字典序排序。
3. 按照 KeyList 的顺序拼接 Map 中的每一个键值对，格式为 `key1=value1&key2=value2&key3=value3`，即为 HttpParameters。
4. 按照 KeyList 的顺序拼接 KeyList 中的每一项，格式为 `key1;key2;key3`，即为 UrlParamList。

示例：

- 示例一：
  请求路径：`/?prefix=example-folder%2F&delimiter=%2F&max-keys=10`
  UrlParamList：`delimiter;max-keys;prefix`
  HttpParameters：`delimiter=%2F&max-keys=10&prefix=example-folder%2F`

> 注意：
> 请求路径中的请求参数在实际发送请求时也会进行 UrlEncode，因此要注意不要重复执行 UrlEncode。

- 示例二：
  请求路径： `/exampleobject?acl`
  UrlParamList： `acl`
  HttpParameters： `acl=`

步骤4：生成 HeaderList 和 HttpHeaders

1. 遍历 HTTP 请求头部，生成 key 到 value 的映射 Map 及 key 的列表 KeyList，key 使用 UrlEncode 编码并转换为小写形式，value 使用 UrlEncode 编码。
2. 将 KeyList 按照字典序排序。
3. 按照 KeyList 的顺序拼接 Map 中的每一个键值对，格式为 `key1=value1&key2=value2&key3=value3` ，即为 HttpHeaders。
4. 按照 KeyList 的顺序拼接 KeyList 中的每一项，格式为 `key1;key2;key3` ，即为 HeaderList。

示例：

请求头：

```
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Date: Thu, 16 May 2019 03:15:06 GMT
x-cos-acl: private
x-cos-grant-read: uin="100000000011"
```

计算得到：

- HeaderList = `date;host;x-cos-acl;x-cos-grant-read`
- HttpHeaders = `date=Thu%2C%2016%20May%202019%2003%3A15%3A06%20GMT&host=examplebucket-1250000000.cos.ap-shanghai.myqcloud.com&x-cos-acl=private&x-cos-grant-read=uin%3D%22100000000011%22`

步骤5：生成 HttpString

根据 HTTP 方法（HttpMethod）、HTTP 请求路径（UriPathname）、HttpParameters 和 HttpHeaders 生成 HttpString，格式为 `HttpMethod\nUriPathname\nHttpParameters\nHttpHeaders\n` 。

其中：

- HttpMethod 转换为小写，例如 get 或 put。
- UriPathname 为请求路径，例如 `/` 或 `/exampleobject` 。
- `\n` 为换行符。如果其中有字符串为空，前后的换行符需要保留，例如 `get\n/exampleobject\n\n\n` 。

步骤6：生成 StringToSign

根据 KeyTime 和 HttpString 生成 StringToSign，格式为 `sha1\nKeyTime\nSHA1(HttpString)\n` 。
其中：

- sha1 为固定字符串。
- `\n` 为换行符。
- SHA1(HttpString) 为使用 SHA1 对 HttpString 计算的消息摘要，16进制小写形式，例如： `54ecfe22f59d3514fdc764b87a32d8133ea611e6` 。

步骤7：生成 Signature

使用 HMAC-SHA1 以 SignKey 为密钥（字符串形式，非原始二进制），以 StringToSign 为消息，计算消息摘要，即为 Signature，例如： `01681b8c9d798a678e43b685a9f1bba0f6c01234` 。

步骤8：生成签名

根据 SecretId、KeyTime、HeaderList、UrlParamList 和 Signature 生成签名，格式为：

```
q-sign-algorithm=sha1
&q-ak=SecretId
&q-sign-time=KeyTime
&q-key-time=KeyTime
&q-header-list=HeaderList
&q-url-param-list=UrlParamList
&q-signature=Signature
```

> 注意：
> 上述格式中的换行仅用于更好的阅读，实际格式并不包含换行。

签名使用

通过 RESTful API 对 COS 发起的 HTTP 签名请求，可以通过以下几种方式传递签名：

1. 通过标准的 HTTP Authorization 头，例如 `Authorization: q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753;1557996953&...&q-signature=...`
2. 作为 HTTP 请求参数，请注意 UrlEncode，例如 `/exampleobject?q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753%3B1557996953&...&q-signature=...`

说明：
上述示例中使用 `...` 省略了部分具体签名内容。

使用临时安全凭证（临时密钥）

如果在计算签名时使用了临时安全凭证，那么在发送请求时还应传入安全令牌字段 x-cos-security-token，根据签名使用方式不同，安全令牌字段的传入方式也有所不同：

1. 当通过标准的 HTTP Authorization 头传入签名时，应同时通过 x-cos-security-token 请求头部传入安全令牌字段，例如：

```
Authorization: q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753;1557996953
&...&q-signature=...
x-cos-security-token: ...
```

2. 当通过 HTTP 请求参数传入签名时，应同时通过 x-cos-security-token 请求参数传入安全令牌字段，例如：

```
/exampleobject?q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753%3B15579969
53&...&q-signature=...&x-cos-security-token=...
```

说明：
上述示例中使用 `...` 省略了部分具体签名和安全访问令牌内容。

代码示例

伪代码

```
KeyTime = [Now];[Expires]
SignKey = HMAC-SHA1([SecretKey], KeyTime)
HttpString = [HttpMethod]\n[HttpURI]\n[HttpParameters]\n[HttpHeaders]\n
StringToSign = sha1\nKeyTime\nSHA1(HttpString)\n
Signature = HMAC-SHA1(SignKey, StringToSign)
```

版权所有：腾讯云计算（北京）有限责任公司

消息摘要算法示例

不同语言如何调用 HMAC-SHA1 可以参考下面的示例：

PHP

```
$sha1HttpString = sha1('ExampleHttpString');
$signKey = hash_hmac('sha1', 'ExampleKeyTime', 'YourSecretKey');
```

Java

```
import org.apache.commons.codec.digest.DigestUtils;
import org.apache.commons.codec.digest.HmacUtils;
String sha1HttpString = DigestUtils.sha1Hex("ExampleHttpString");
String signKey = HmacUtils.hmacSha1Hex("YourSecretKey", "ExampleKeyTime");
```

Python

```
import hmac
import hashlib
sha1_http_string = hashlib.sha1('ExampleHttpString'.encode('utf-8')).hexdigest()
sign_key = hmac.new('YourSecretKey'.encode('utf-8'), 'ExampleKeyTime'.encode('utf
-8'), hashlib.sha1).hexdigest()
```

Node.js

```
var crypto = require('crypto');
var sha1HttpString = crypto.createHash('sha1').update('ExampleHttpString').digest
('hex');
var signKey = crypto.createHmac('sha1', 'YourSecretKey').update('ExampleKeyTime'
).digest('hex');
```

Go

```
import (
"crypto/hmac"
"crypto/sha1"
)
h := sha1.New()
h.Write([]byte("ExampleHttpString"))
sha1HttpString := h.Sum(nil)
var hashFunc = sha1.New
h = hmac.New(hashFunc, []byte("YourSecretKey"))
```

版权所有：腾讯云计算（北京）有限责任公司

```
h.Write([]byte("ExampleKeyTime"))
signKey := h.Sum(nil)
```

实际案例

准备工作

登录访问管理控制台的 API 密钥管理 页面获取其 APPID、SecretId 和 SecretKey，举例如下：

APPID	SecretId	SecretKey
1250000000	AKXXXXXXXXXXXXXXXXXXX	BQXXXXXXXXXXXXXXXXXXX

上传对象

原始请求

```
PUT /exampleobject(%E8%85%BE%E8%AE%AF%E4%BA%91) HTTP/1.1
Date: Thu, 16 May 2019 06:45:51 GMT
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Content-Type: text/plain
Content-Length: 13
Content-MD5: mQ/fVh815F3k6TAUm8m0eg==
x-cos-acl: private
x-cos-grant-read: uin="100000000011"
ObjectContent
```

中间变量

- **KeyTime** = `1557989151;1557996351`
- **SignKey** = `eb2519b498b02ac213cb1f3d1a3d27a3b3c9bc5f`
- **UrlParamList** = `(empty string)`
- **HttpParameters** = `(empty string)`
- **HeaderList** = `content-length;content-md5;content-type;date;host;x-cos-acl;x-cos-grant-read`
- **HttpHeaders** = `content-length=13&content-md5=mQ%2FfVh815F3k6TAUm8m0eg%3D%3D&content-type=text%2Fplain&date=Thu%2C%2016%20May%202019%2006%3A45%3A51%20GMT&host=examplebucket-1250000000.cos.ap-beijing.myqcloud.com&x-cos-acl=private&x-cos-grant-read=uin%3D%22100000000011%22`

- **HttpString** = `put\n/exampleobject(腾讯云)\n\ncontent-length=13&content-md5=mQ%2FfVh815F3k6TAUm8m0eg%3D%3D&content-type=text%2Fplain&date=Thu%2C%2016%20May%202019%2006%3A45%3A51%20GMT&host=examplebucket-1250000000.cos.ap-beijing.myqcloud.com&x-cos-acl=private&x-cos-grant-read=uin%3D%22100000000011%22\n`
- **StringToSign** = `sha1\n1557989151;1557996351\n8b2751e77f43a0995d6e9eb9477f4b685cca4172\n`
- **Signature** = `3b8851a11a569213c17ba8fa7dcf2abec6931234`

其中，(empty string) 代表长度为0的空字符串，`\n` 代表换行符。

签名后的请求

```
PUT /exampleobject(%E8%85%BE%E8%AE%AF%E4%BA%91) HTTP/1.1
Date: Thu, 16 May 2019 06:45:51 GMT
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Content-Type: text/plain
Content-Length: 13
Content-MD5: mQ/fVh815F3k6TAUm8m0eg==
x-cos-acl: private
x-cos-grant-read: uin="100000000011"
Authorization: q-sign-algorithm=sha1&q-ak=AKIDQjz3ltompVjBni5LitkWHFlFpwkn****&q-sign-time=1557989151;1557996351&q-key-time=1557989151;1557996351&q-header-list=content-length;content-md5;content-type;date;host;x-cos-acl;x-cos-grant-read&q-url-param-list=&q-signature=3b8851a11a569213c17ba8fa7dcf2abec693****
ObjectContent
```

下载对象

原始请求

```
GET /exampleobject(%E8%85%BE%E8%AE%AF%E4%BA%91)?response-content-type=application%2Foctet-stream&response-cache-control=max-age%3D600 HTTP/1.1
Date: Thu, 16 May 2019 06:55:53 GMT
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

中间变量

- **KeyTime** = `1557989753;1557996953`
- **SignKey** = `937914bf490e9e8c189836aad2052e4feeb35eaf`
- **UrlParamList** = `response-cache-control;response-content-type`

- **HttpParameters** = `response-cache-control=max-age%3D600&response-content-type=application%2Foctet-stream`
- **HeaderList** = `date;host`
- **HttpHeaders** = `date=Thu%2C%2016%20May%202019%2006%3A55%3A53%20GMT&host=examplebucket-1250000000.cos.ap-beijing.myqcloud.com`
- **HttpString** = `get\n/exampleobject(腾讯云)\nresponse-cache-control=max-age%3D600&response-content-type=application%2Foctet-stream\ndate=Thu%2C%2016%20May%202019%2006%3A55%3A53%20GMT&host=examplebucket-1250000000.cos.ap-beijing.myqcloud.com\n`
- **StringToSign** = `sha1\n1557989753;1557996953\n54ecfe22f59d3514fdc764b87a32d8133ea611e6\n`
- **Signature** = `01681b8c9d798a678e43b685a9f1bba0f6c01234`

其中，`\n` 代表换行符。

签名后的请求

```
GET /exampleobject(%E8%85%BE%E8%AE%AF%E4%BA%91)?response-content-type=applicatio
n%2Foctet-stream&response-cache-control=max-age%3D600 HTTP/1.1
Date: Thu, 16 May 2019 06:55:53 GMT
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDQjz3ltompVjBni5LitkWHFlFpwkn****&q-
sign-time=1557989753;1557996953&q-key-time=1557989753;1557996953&q-header-list=da
te;host&q-url-param-list=response-cache-control;response-content-type&q-signature
=01681b8c9d798a678e43b685a9f1bba0f6c0****
```

操作列表

最近更新时间：2023-02-23 16:43:51

腾讯云对象存储服务（COS）相关接口及说明如下：

Service 接口

API	操作名	操作描述
GET Service（List Buckets）	查询存储桶列表	查询指定账号下所有存储桶列表

Bucket 接口

基本操作接口

API	操作名	操作描述
PUT Bucket	创建存储桶	在指定账号下创建一个存储桶
GET Bucket（List Objects）	查询对象列表	查询存储桶下的部分或者全部对象
HEAD Bucket	检索存储桶及其权限	确认存储桶是否存在且是否有权限访问
DELETE Bucket	删除存储桶	删除指定账号下的空存储桶
GET Bucket Object versions	查询对象版本	查询存储桶下的部分或者全部对象及其历史版本信息

访问控制（acl）接口

API	操作名	操作描述
PUT Bucket acl	设置存储桶 ACL	设置指定存储桶访问权限控制列表
GET Bucket acl	查询存储桶 ACL	查询存储桶的访问控制列表

跨域资源共享（cors）接口

API	操作名	操作描述
PUT Bucket cors	设置跨域配置	设置存储桶的跨域访问权限

版权所有：腾讯云计算（北京）有限责任公司

API	操作名	操作描述
GET Bucket cors	查询跨域配置	查询存储桶的跨域访问配置信息
DELETE Bucket cors	删除跨域配置	删除存储桶的跨域访问配置信息

生命周期（lifecycle）接口

API	操作名	操作描述
PUT Bucket lifecycle	设置生命周期	设置存储桶生命周期管理的配置
GET Bucket lifecycle	查询生命周期	查询存储桶生命周期管理的配置
DELETE Bucket lifecycle	删除生命周期	删除存储桶生命周期管理的配置

存储桶策略（policy）接口

API	操作名	操作描述
PUT Bucket policy	设置存储桶策略	设置指定存储桶的权限策略
GET Bucket policy	查询存储桶策略	查询指定存储桶的权限策略
DELETE Bucket policy	删除存储桶策略	删除指定存储桶的权限策略

防盗链（referer）接口

API	操作名	操作描述
PUT Bucket referer	设置存储桶 referer	设置存储桶 Referer 白名单或者黑名单
GET Bucket referer	查询存储桶 referer	查询存储桶 Referer 白名单或者黑名单

存储桶标签（tagging）接口

API	操作名	操作描述
PUT Bucket tagging	设置存储桶标签	为已存在的存储桶设置标签
GET Bucket tagging	查询存储桶标签	查询指定存储桶下已有的存储桶标签
DELETE Bucket tagging	删除存储桶标签	删除指定的存储桶标签

静态网站（website）接口

版权所有：腾讯云计算（北京）有限责任公司

API	操作名	操作描述
PUT Bucket website	设置静态网站	为存储桶配置静态网站
GET Bucket website	查询静态网站配置	查询与存储桶关联的静态网站配置信息
DELETE Bucket website	删除静态网站配置	删除指定存储桶的静态网站配置信息

智能分层（IntelligentTiering）接口

API	操作名	操作描述
PUT Bucket IntelligentTiering	设置智能分层配置	启用存储桶的智能分层存储配置
GET Bucket IntelligentTiering	查询智能分层配置	查询存储桶的智能分层配置信息

清单（inventory）接口

API	操作名	操作描述
PUT Bucket inventory	设置清单任务	在存储桶中创建清单任务
POST Bucket inventory	创建一次性清单任务	对一个存储桶创建一个一次性清单任务
GET Bucket inventory	查询清单任务	查询存储桶的指定清单配置信息
List Bucket Inventory Configurations	查询所有清单	查询存储桶的所有清单任务
DELETE Bucket inventory	删除清单任务	删除存储桶中指定的清单任务

版本控制（versioning）接口

API	操作名	操作描述
PUT Bucket versioning	设置版本控制	启用或者暂停存储桶的版本控制功能
GET Bucket versioning	查询版本控制	查询存储桶的版本控制信息

存储桶复制（replication）接口

API	操作名	操作描述
PUT Bucket replication	设置存储桶复制	对已启用版本控制的存储桶配置存储桶复制规则
GET Bucket replication	查询存储桶复制	查询存储桶的存储桶复制配置信息

版权所有：腾讯云计算（北京）有限责任公司

API	操作名	操作描述
DELETE Bucket replication	删除存储桶复制	删除存储桶的存储桶复制配置信息

日志管理（logging）接口

API	操作名	操作描述
PUT Bucket logging	设置日志管理	为源存储桶开启日志记录
GET Bucket logging	查询日志管理	查询源存储桶的日志配置信息

全球加速（accelerate）接口

API	操作名	操作描述
PUT Bucket Accelerate	设置全球加速	启用或暂停存储桶的全球加速功能
GET Bucket Accelerate	查询全球加速	查询存储桶的全球加速功能配置信息

存储桶加密（encryption）接口

API	操作名	操作描述
PUT Bucket encryption	设置存储桶加密	设置指定存储桶下的默认加密配置
GET Bucket encryption	查询存储桶加密	查询指定存储桶下的默认加密配置
DELETE Bucket encryption	删除存储桶加密	删除指定存储桶下的默认加密配置

对象锁定（ObjectLock）接口

API	操作名	操作描述
PUT Bucket ObjectLockConfiguration	设置对象锁定	为已存在的存储桶设置对象锁定
GET Bucket ObjectLockConfiguration	查询对象锁定	查询已生效的对象锁定配置
GET Object retention	查询对象锁定的到期日期	查询对象锁定的到期日期

Object 接口

基本操作接口

API	操作名	操作描述
PUT Object	简单上传对象	上传一个对象至存储桶
PUT Object - Copy	设置对象复制	复制文件到目标路径
POST Object	表单上传对象	使用表单请求上传对象
GET Object	下载对象	下载一个对象至本地
HEAD Object	查询对象元数据	查询对象的元数据信息
DELETE Object	删除单个对象	在存储桶中删除指定对象
DELETE Multiple Objects	删除多个对象	在存储桶中批量删除对象
OPTIONS Object	预请求跨域配置	用预请求来确认是否可以发送真正的跨域请求
POST Object restore	恢复归档对象	将归档类型的对象取回访问
SELECT Object Content	检索对象内容	检索指定对象的内容

访问控制接口

API	操作名	操作描述
PUT Object acl	设置对象 ACL	设置存储桶中某个对象的访问控制列表
GET Object acl	查询对象 ACL	查询对象的访问控制列表

对象标签接口

API	操作名	操作描述
PUT Object tagging	设置对象标签	为已上传的对象设置标签
GET Object tagging	查询对象标签	查询指定对象下已有的对象标签
DELETE Object tagging	删除对象标签	删除指定对象下已有的对象标签

分块上传接口

API	操作名	操作描述
Initiate Multipart Upload	初始化分块上传	初始化分块上传任务
Upload Part	上传分块	分块上传文件

版权所有：腾讯云计算（北京）有限责任公司

API	操作名	操作描述
Upload Part - Copy	复制分块	将其他对象复制为一个分块
Complete Multipart Upload	完成分块上传	完成整个文件的分块上传
Abort Multipart Upload	终止分块上传	终止一个分块上传操作并删除已上传的块
List Multipart Uploads	查询分块上传	查询正在进行中的分块上传信息
List Parts	查询已上传块	查询特定分块上传操作中的已上传的块

批量处理（batch）接口

API	操作名	操作描述
CreateJob	创建任务	用于在存储桶中创建批量处理任务
DescribeJob	描述任务	用于获取已创建的批量处理任务的参数和任务执行状态
ListJobs	查询任务	用于列出已创建的批量处理任务
UpdateJobPriority	更新任务优先级	用于更新已创建任务的优先级
UpdateJobStatus	更新任务状态	用于更新已创建任务的状态
公共元素	—	批量处理功能的公共元素
错误响应	—	批量处理功能的错误响应

数据处理接口

数据处理接口

数据处理接口包括图片处理、AI 内容识别、媒体处理、文档处理、文件处理等接口类别，详情请参见 数据处理接口。

内容审核接口

内容审核接口包括图片审核、视频审核、音频审核、文本审核、文档审核、网页审核、直播审核等接口类别，详情请参见 内容审核接口。

任务与工作流接口

任务与工作流接口包括工作流接口、工作流实例、任务接口、模板接口、批量任务接口等接口类别，详情请参见 任务与工作流接口。

云查毒接口

API	操作描述
提交病毒检测任务	对云上的文件进行文件病毒（例如木马病毒、蠕虫病毒等）检测
查询病毒检测任务结果	用于查询一个病毒检测任务的状态或结果

Service 接口

GET Service（List Buckets）

最近更新时间：2023-12-20 14:55:05

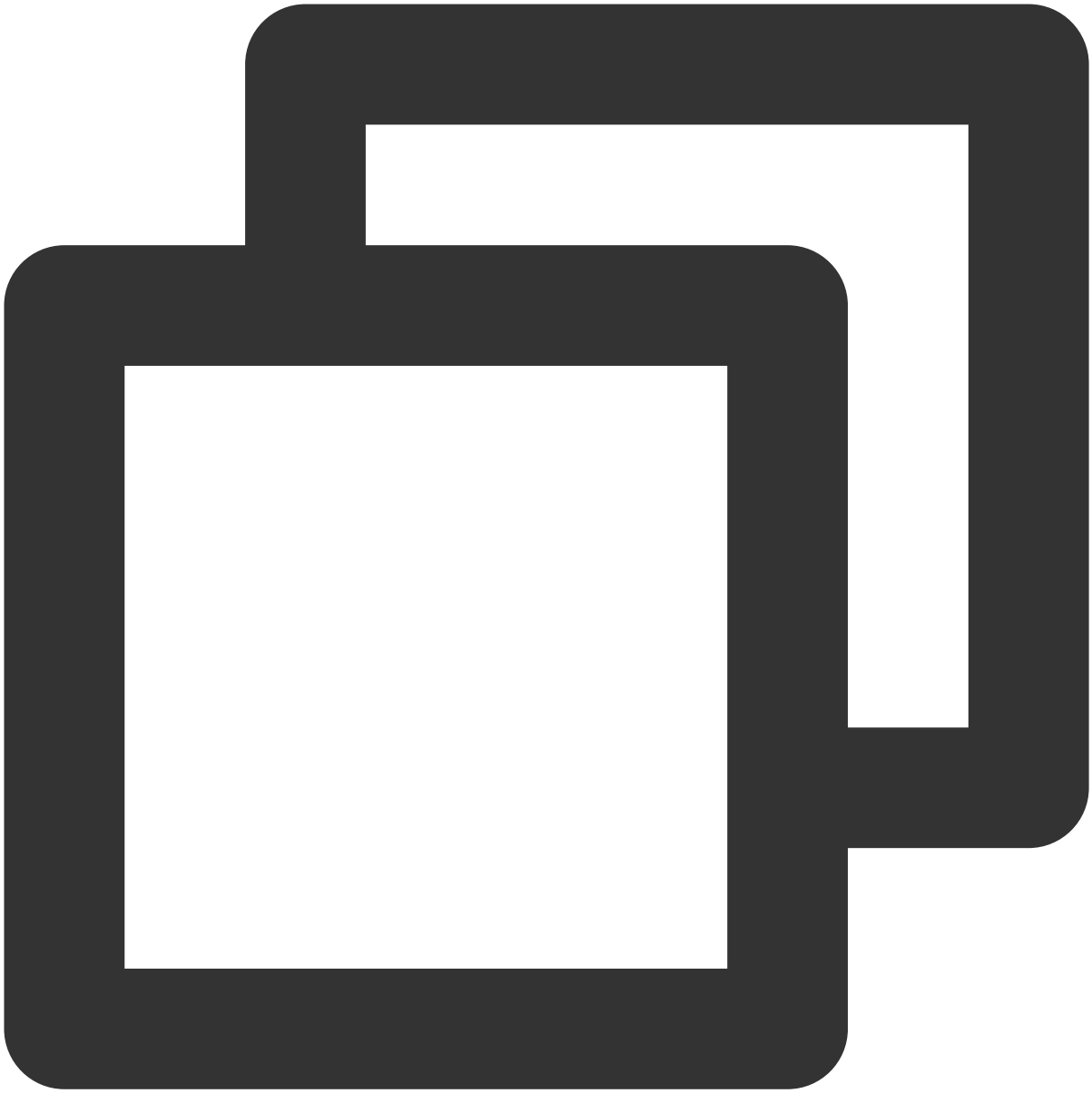
功能描述

GET Service 接口是用来查询请求者名下的所有存储桶列表或特定地域下的存储桶列表。

请求

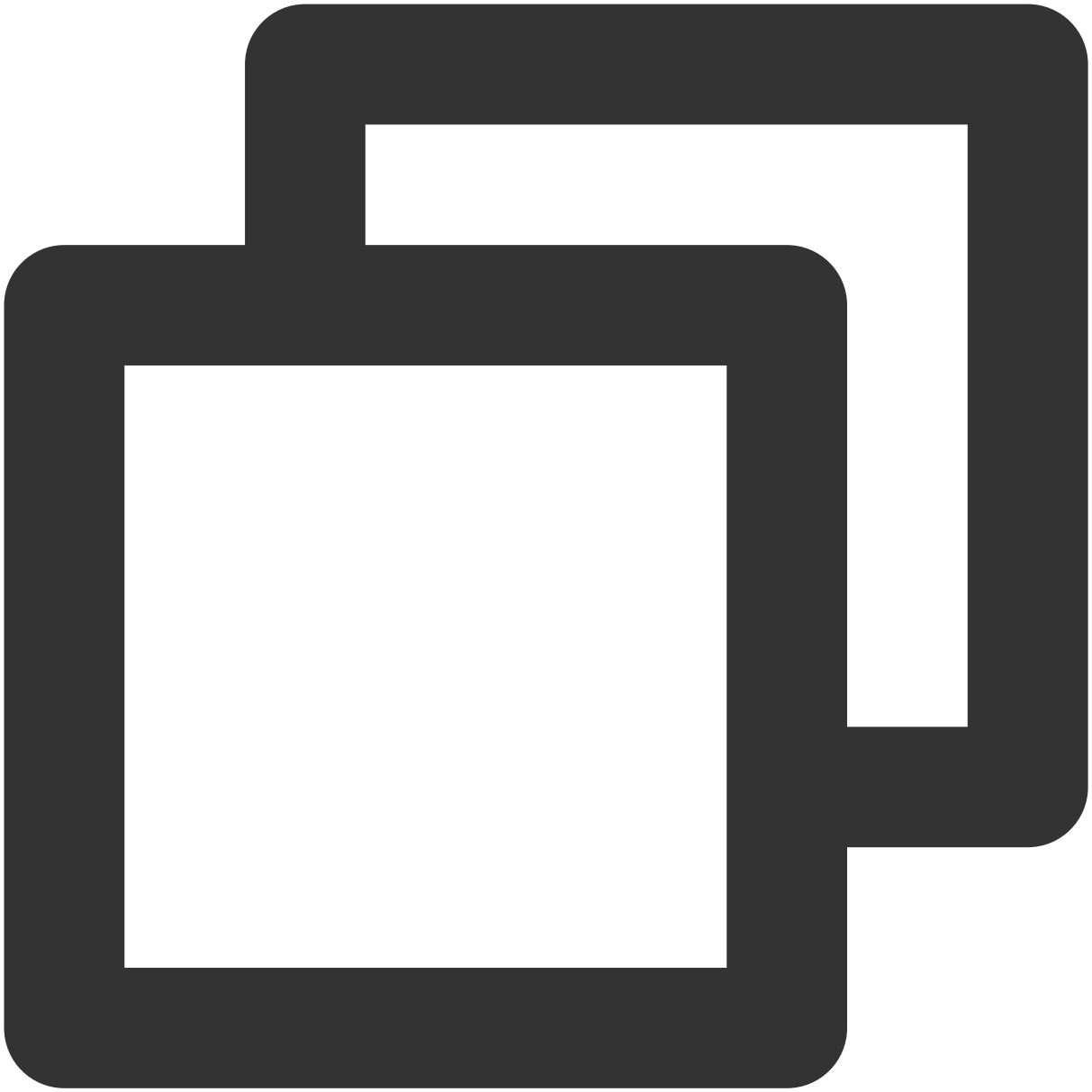
请求示例

示例一



```
GET / HTTP/1.1
Host: service.cos.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

示例二



```
GET / HTTP/1.1
Host: cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

Host: 查询全部存储桶列表指定为 `service.cos.myqcloud.com`，查询特定地域下的存储桶列表指定为 cos.<Region>.myqcloud.com，其中 <Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

请求参数

GetService 支持通过请求参数根据存储桶标签、地域、创建时间过滤存储桶。其中，根据存储桶标签过滤存储桶，仅支持传入一个存储桶标签。当存储桶包含多个存储桶标签时，只要其中任意一个为用户指定的存储桶标签，就会返回该存储桶。

名称	描述	类型	是否必选
tagkey	支持根据存储桶标签过滤存储桶，仅支持传入一个存储桶标签，tagkey 用于传入标签键	string	否
tagvalue	支持根据存储桶标签过滤存储桶，仅支持传入一个存储桶标签，tagvalue 用于传入标签值	string	否
region	支持根据地域过滤存储桶，例如 `region=ap-beijing`，COS 支持的地域可参考 地域和访问域名	string	否
create-time	GMT 时间戳，和 range 参数一起使用，支持根据创建时间过滤存储桶，例如 `create-time=1642662645`	Timestamp	否
range	和 create-time 参数一起使用，支持根据创建时间过滤存储桶，支持枚举值 lt（创建时间早于 create-time）、gt（创建时间晚于 create-time）、lte（创建时间早于或等于 create-time）、gte（创建时间晚于或等于create-time）	string	否

当存储桶标签授权情况与 GetService 授权不同时，GetService 请求的鉴权和返回情况如下。

存储桶标签授权情况	GetService 授权情况	GetService 请求	请求返回
主账号通过存储桶标签授权，授权了子账号存储桶标签 tagA 的资源操作权限。	没有授权 GetService 权限	携带存储桶标签参数 tagA	包含存储桶标签 tagA 的存储桶列表
		不携带存储桶标签参数	Access Denied
	额外授权 GetService 权限	携带存储桶标签参数 tagA	包含存储桶标签 tagA 的存储桶列表
		不携带存储桶标签参数	所有存储桶列表
主账号没有通过存储桶标签授权，没有授权子账号存储桶标签 tagA 的资源操作权限。	没有授权 GetService 权限	携带存储桶标签参数 tagA	Access Denied
		不携带存储桶标签参数	Access Denied

	额外授权 GetService 权限	携带存储桶标签参数 tagA	包含存储桶标签 tagA 的存储桶列表
		不携带存储桶标签参数	所有存储桶列表

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

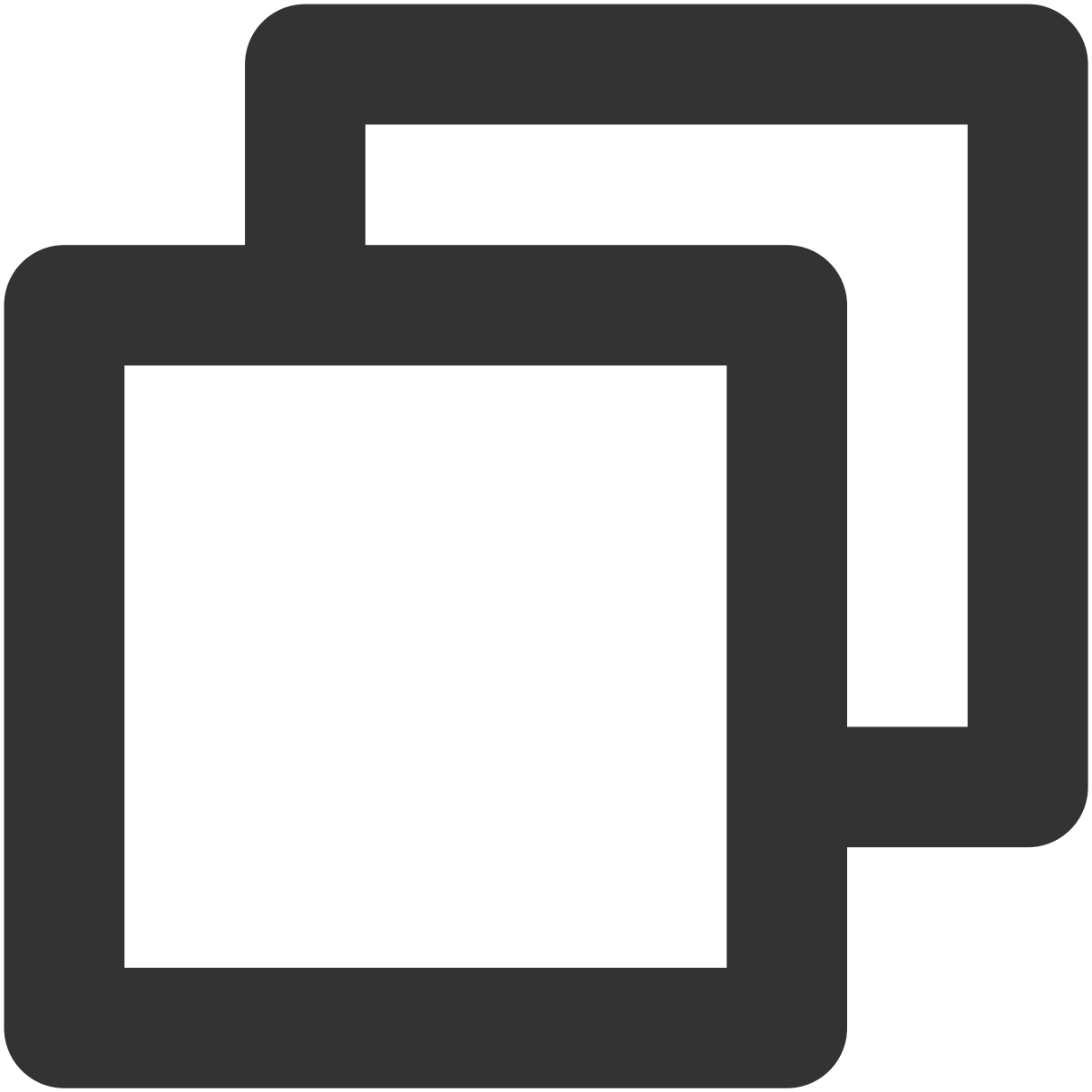
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含所有或特定地域下的存储桶列表。



```
<ListAllMyBucketsResult>
    <Owner>
        <ID>string</ID>
        <DisplayName>string</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>string</Name>
            <Location>Enum</Location>
            <CreationDate>date</CreationDate>
        </Bucket>
```

```
        <Bucket>
            <Name>string</Name>
            <Location>Enum</Location>
            <CreationDate>date</CreationDate>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
ListAllMyBucketsResult	无	保存 GET Service 结果的所有信息	Container

Container 节点 ListAllMyBucketsResult 的内容：

节点名称（关键字）	父节点	描述	类型
Owner	ListAllMyBucketsResult	存储桶持有者信息	Container
Buckets	ListAllMyBucketsResult	存储桶列表	Container

Container 节点 Owner 的内容：

节点名称（关键字）	父节点	描述
ID	ListAllMyBucketsResult.Owner	存储桶持有者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:uin/[OwnerUin]` 例如 `qcs::cam::uin/100000000001:uin/10000000000`
DisplayName	ListAllMyBucketsResult.Owner	存储桶持有者的名字

Container 节点 Buckets 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	ListAllMyBucketsResult.Buckets	存储桶信息	Container

Container 节点 Buckets.Bucket 的内容：

节点名称（关键字）	父节点	描述	类型
Name	ListAllMyBucketsResult.Buckets.Bucket	存储桶的名称，格式为	string

		`<BucketName-APPID>` 例如 `examplebucket-1250000000`	
Location	ListAllMyBucketsResult.Buckets.Bucket	存储桶所在地域，枚举值请参见 地域和访问域名 文档 例如 ap-beijing，ap-hongkong，eu-frankfurt 等	Enum
CreationDate	ListAllMyBucketsResult.Buckets.Bucket	存储桶的创建时间，为 ISO8601 格式，例如 2019-05-24T10:56:40Z	date

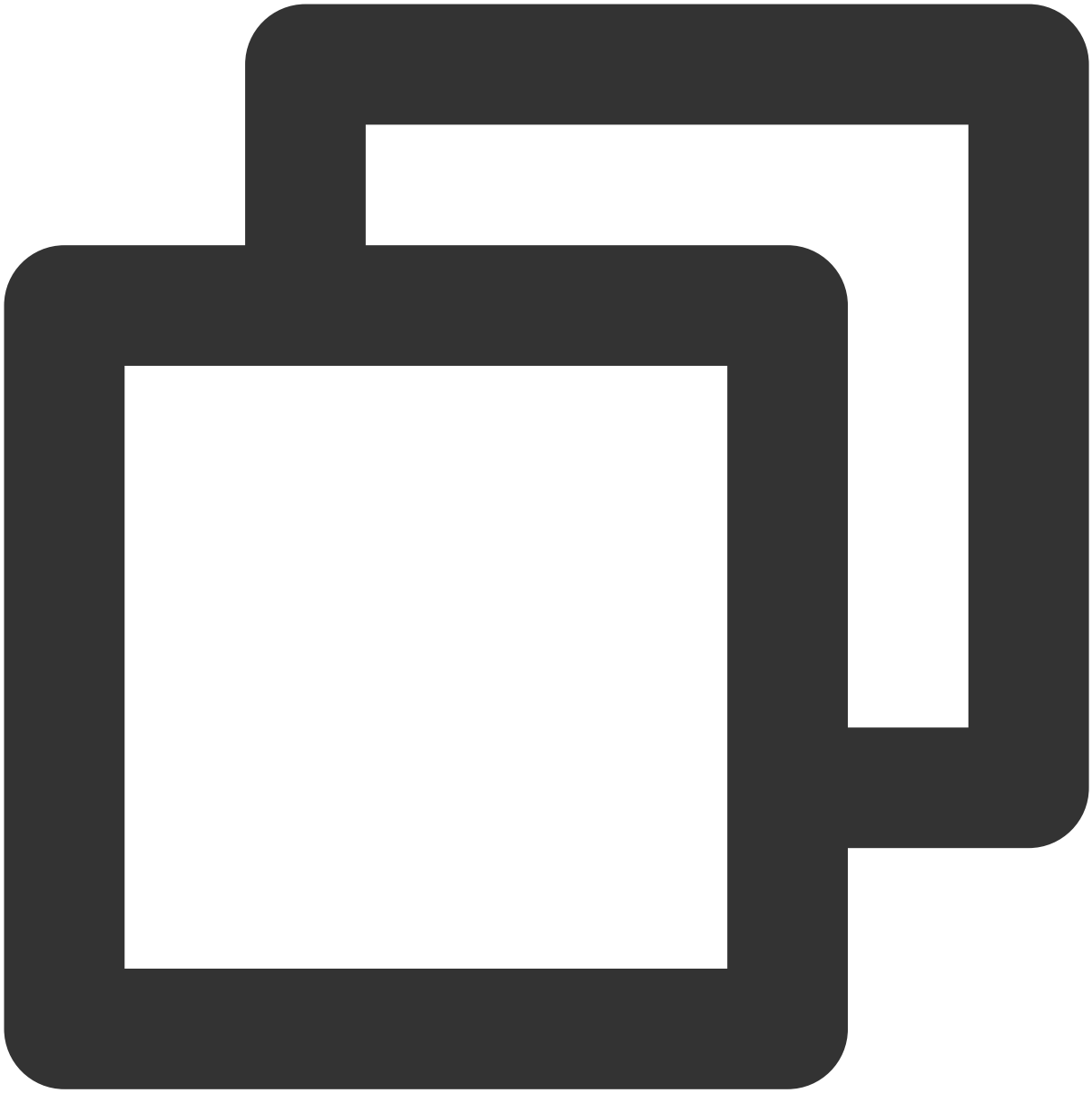
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

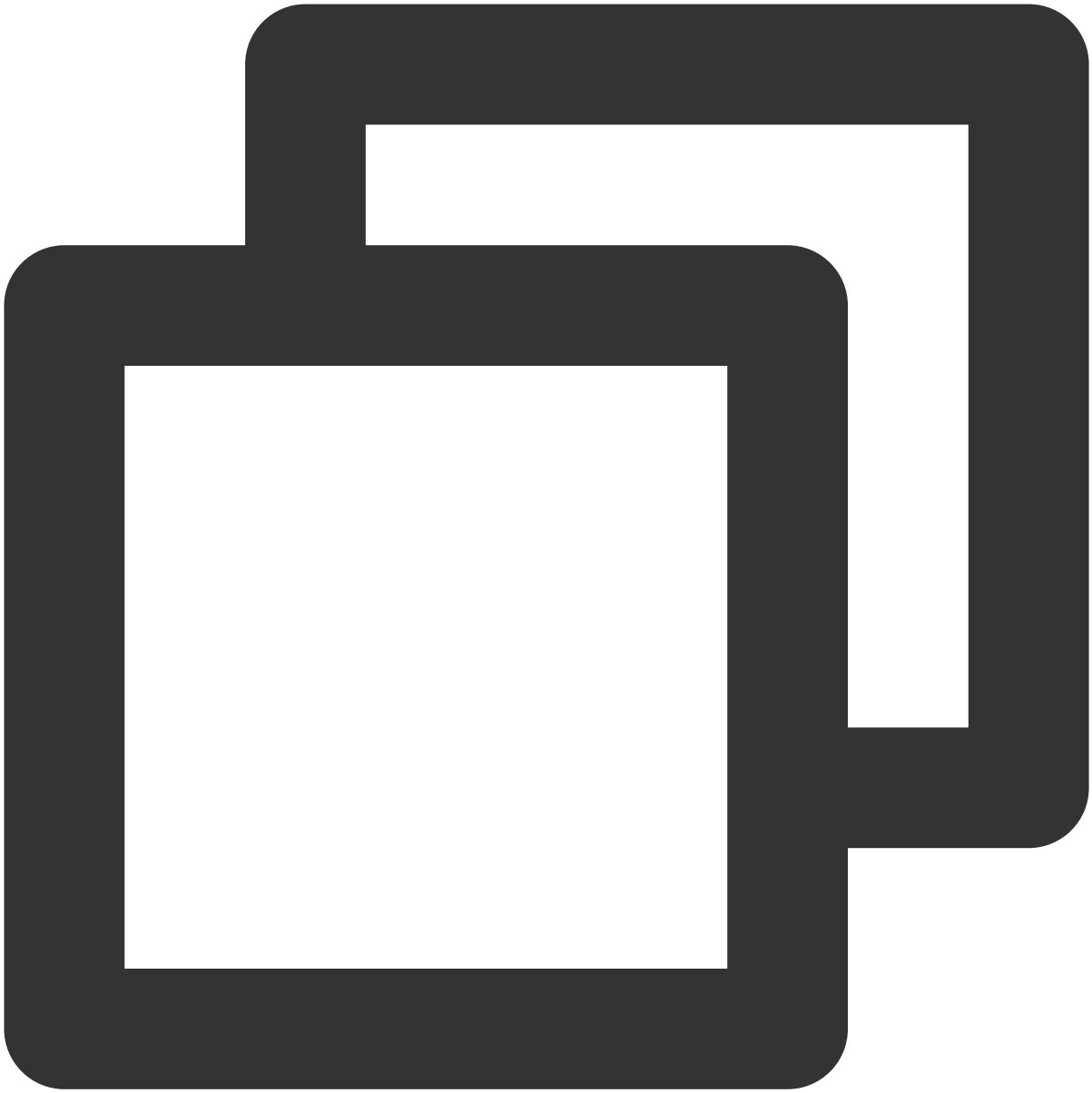
案例一：查询所有存储桶列表

请求



```
GET / HTTP/1.1
Host: service.cos.myqcloud.com
Date: Fri, 24 May 2019 11:59:50 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

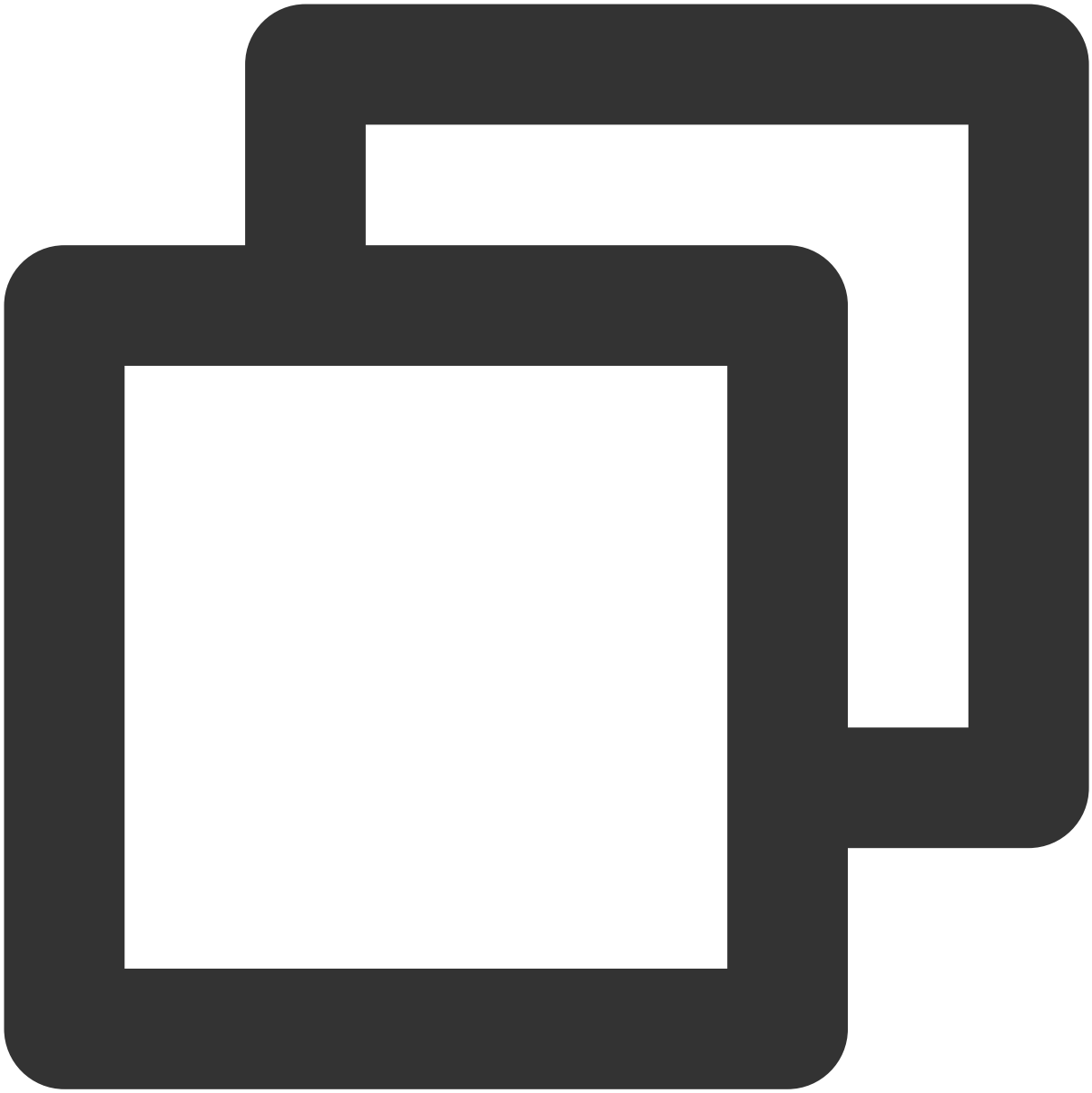


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 805
Connection: close
Date: Fri, 24 May 2019 11:59:51 GMT
Server: tencent-cos
x-cos-request-id: NWNlN2RjYjdfOGFiMjM1MGFfNTVjMl8zMmI1****

<ListAllMyBucketsResult>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>100000000001</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>examplebucket1-1250000000</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2019-05-24T11:49:50Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>examplebucket2-1250000000</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2019-05-24T11:51:50Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>examplebucket3-1250000000</Name>
            <Location>eu-frankfurt</Location>
            <CreationDate>2019-05-24T11:53:50Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>examplebucket4-1250000000</Name>
            <Location>eu-frankfurt</Location>
            <CreationDate>2019-05-24T11:55:50Z</CreationDate>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

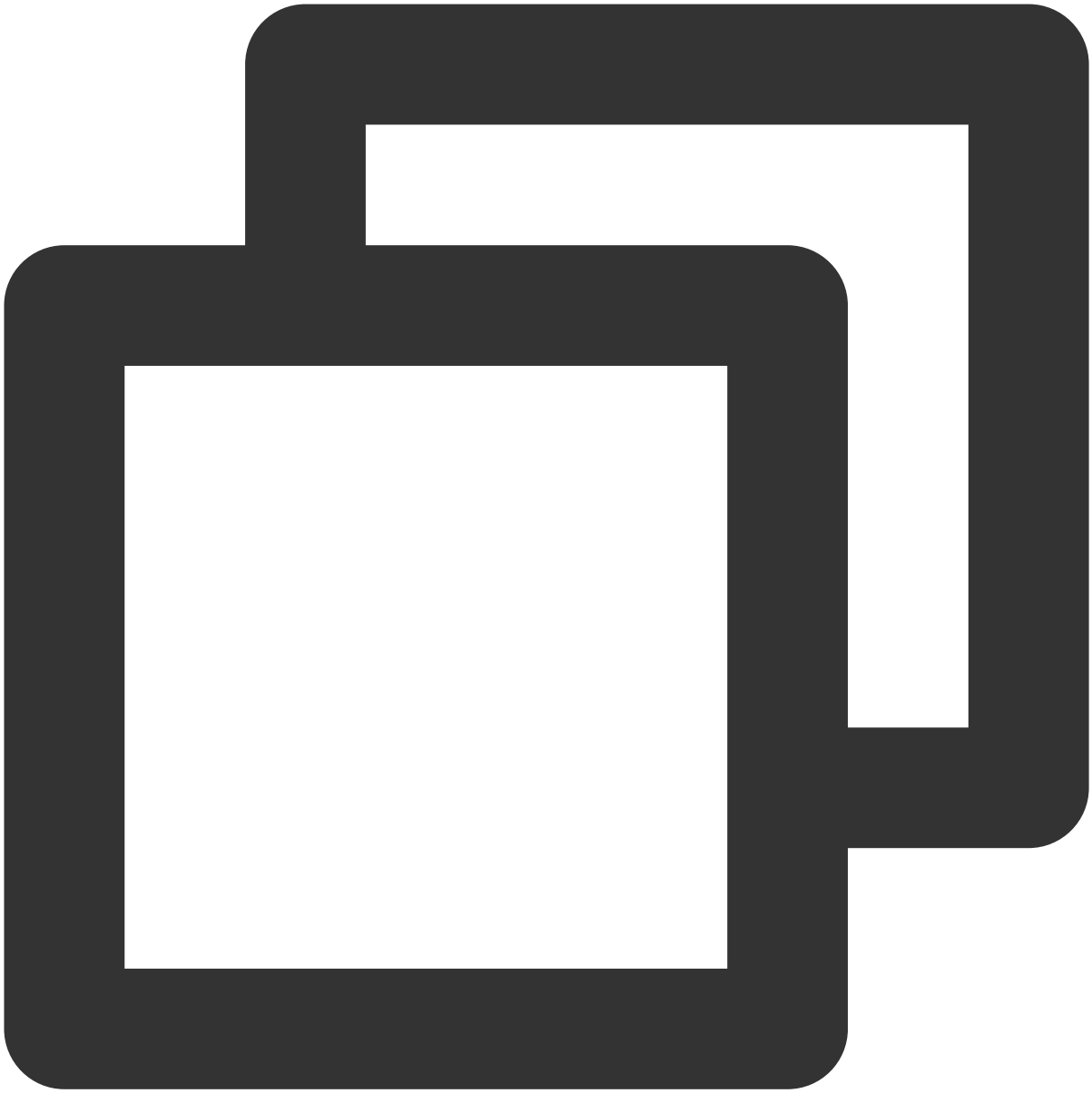
案例二：查询特定地域下的存储桶列表（通过域名筛选）

请求



```
GET / HTTP/1.1
Host: cos.ap-beijing.myqcloud.com
Date: Fri, 24 May 2019 11:59:51 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

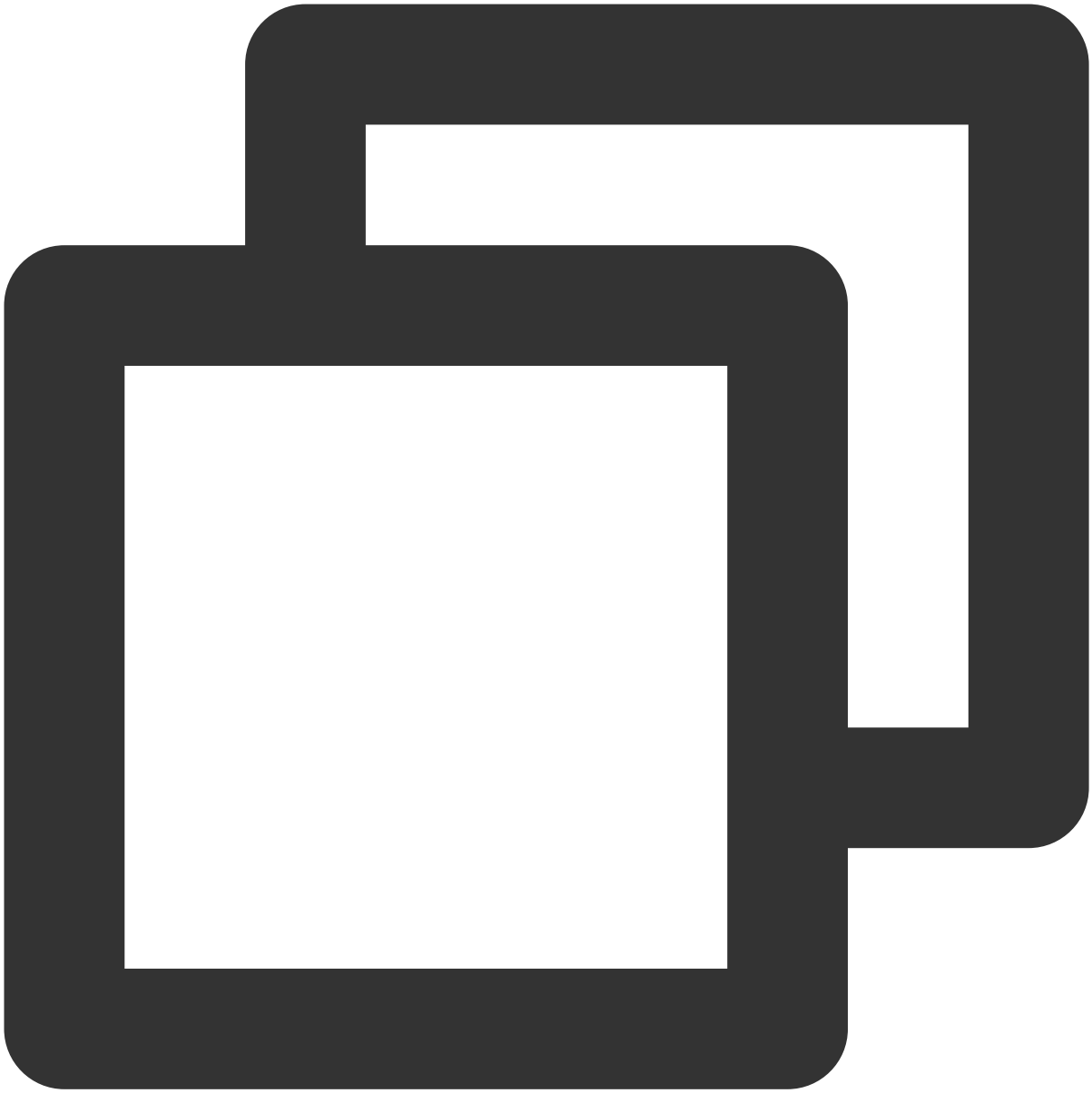


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 495
Connection: close
Date: Fri, 24 May 2019 11:59:51 GMT
Server: tencent-cos
x-cos-request-id: NWNlN2RjYjdfZjhjODBiMDlfOWNlNF9hYzc2****

<ListAllMyBucketsResult>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>100000000001</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>examplebucket1-1250000000</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2019-05-24T11:49:50Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>examplebucket2-1250000000</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2019-05-24T11:51:50Z</CreationDate>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

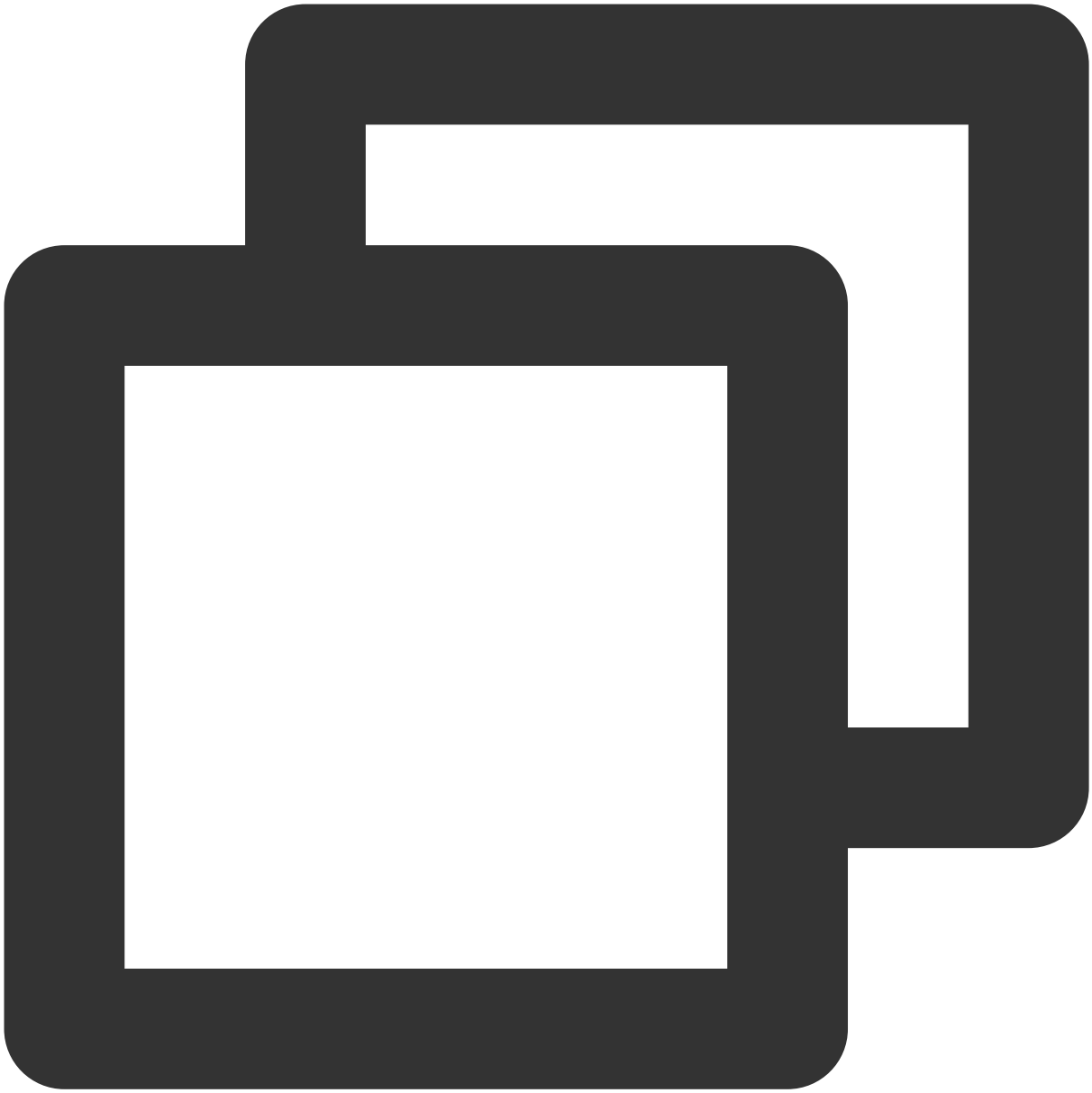
案例三：查询特定地域下的存储桶列表（根据请求参数筛选）

请求



```
GET /?region=ap-beijing HTTP/1.1
Host: service.cos.myqcloud.com
Date: Fri, 24 May 2019 11:59:51 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

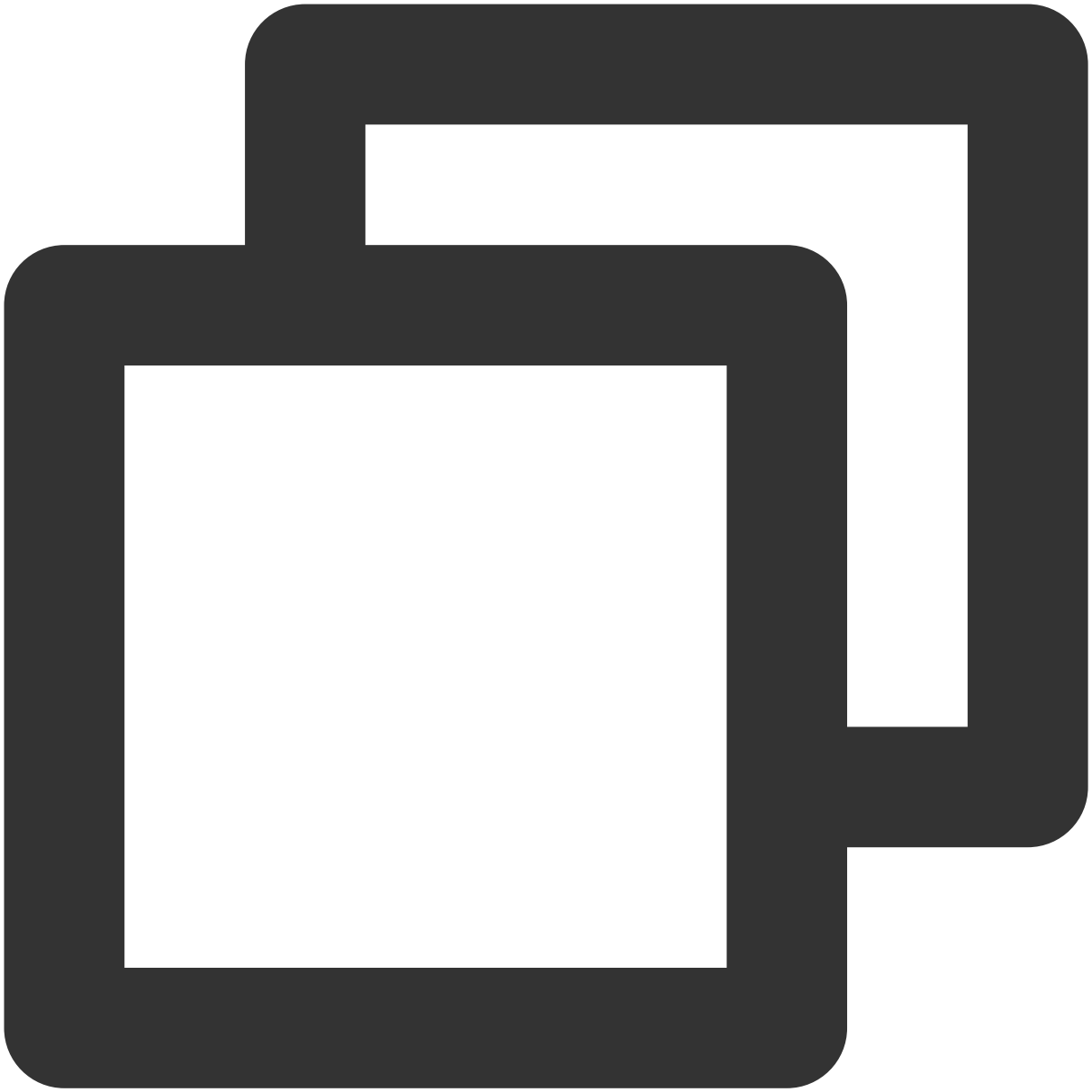


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 495
Connection: close
Date: Fri, 24 May 2019 11:59:51 GMT
Server: tencent-cos
x-cos-request-id: NWNlN2RjYjdfZjhjODBiMDlfOWNlNF9hYzc2****

<ListAllMyBucketsResult>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>100000000001</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>examplebucket1-</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2019-05-24T11:49:50Z</CreationDate>
        </Bucket>
        <Bucket>
            <Name>examplebucket2-1250000000</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2019-05-24T11:51:50Z</CreationDate>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

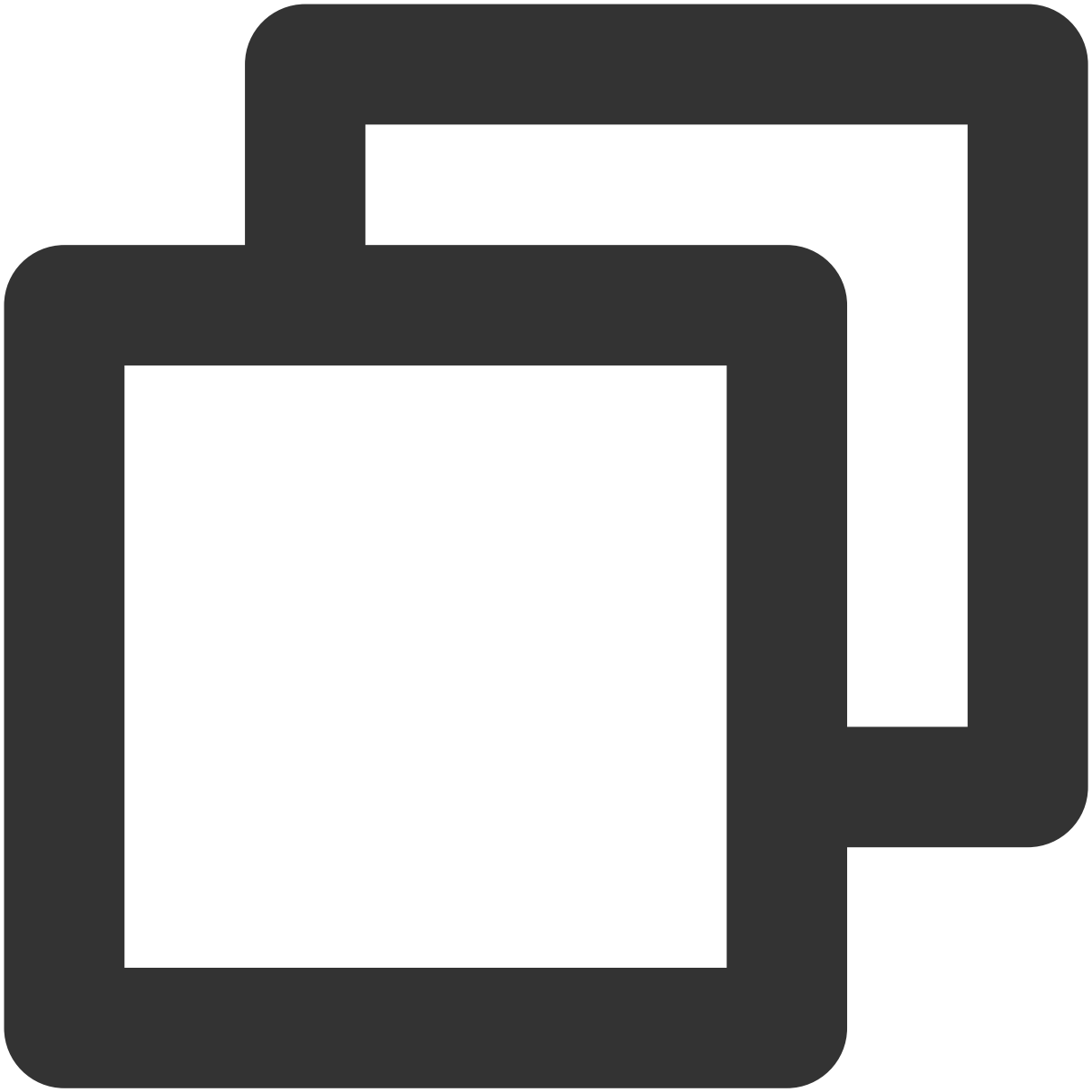
案例四：根据指定标签过滤存储桶列表

存储桶 examplebucket-1250000000 的标签为



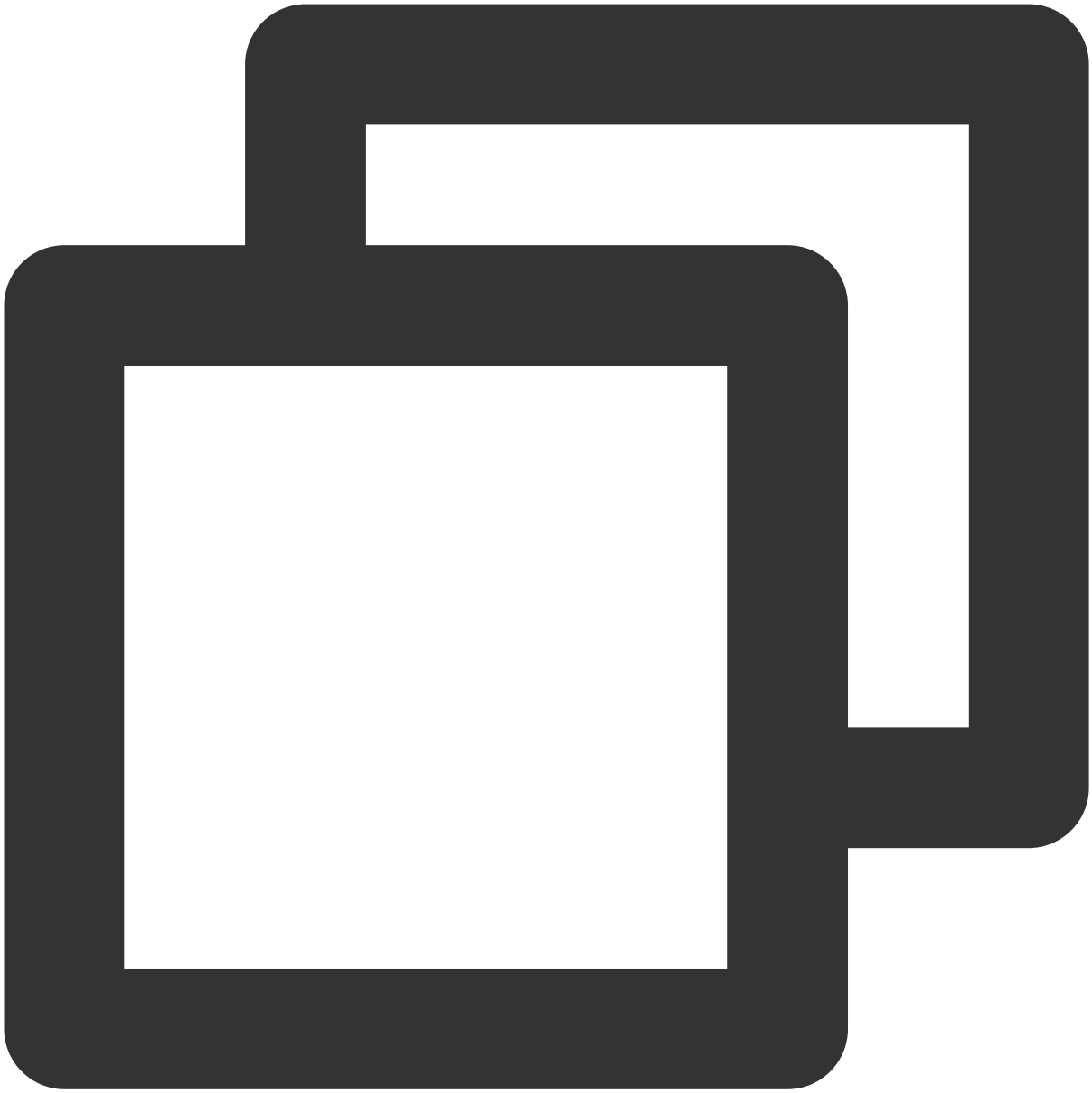
```
<key1, value1>
```

存储桶 examplebucket1-1250000000 的标签为



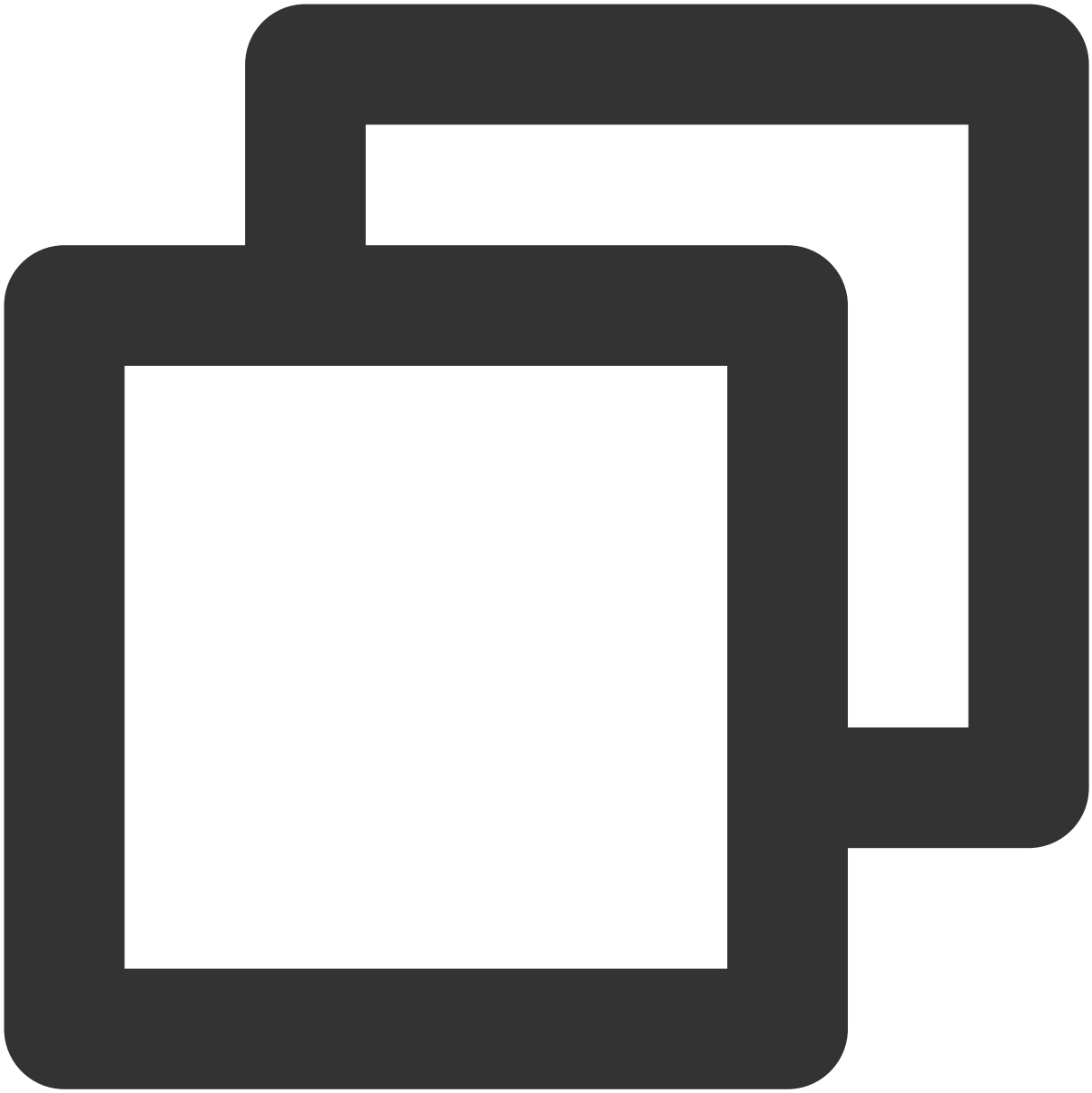
```
<key1, value1> 和 <key2, value2>
```

请求



```
GET /?tagkey=key1&tagvalue=value1 HTTP/1.1
Host: service.cos.myqcloud.com
Date: Fri, 24 May 2019 11:59:51 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



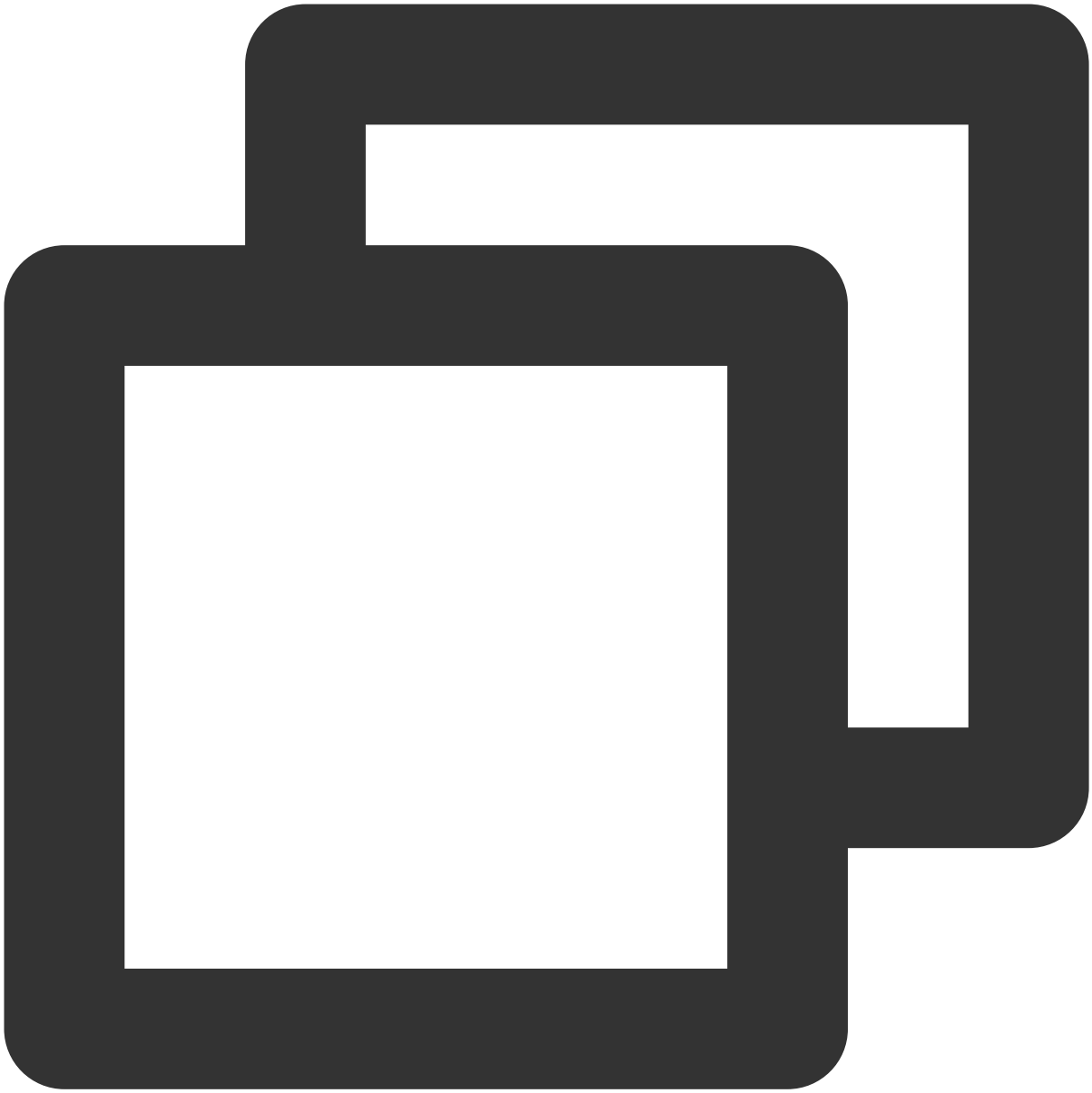
```
HTTP/1.1 200 OK
Content-Length: 378
Content-Type: application/xml
Server: tencent-cos
Connection: keep-alive
Date: Thu, 20 Oct 2022 07:29:40 GMT
x-cos-request-id: NjM1MGY4ZTRfMWViMjM1MGFfYjg3MV8xNjdk****

<ListAllMyBucketsResult>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>100000000001</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>examplebucket-1250000000 </Name>
            <Location>ap-guangzhou</Location>
            <CreationDate>2022-04-11T03:01:49Z</CreationDate>
            <BucketType>cos</BucketType>
        </Bucket>
    </Buckets>
    <Buckets>
        <Bucket>
            <Name>examplebucket1-1250000000 </Name>
            <Location>ap-guangzhou</Location>
            <CreationDate>2022-04-12T03:01:49Z</CreationDate>
            <BucketType>cos</BucketType>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

案例五：根据创建时间过滤存储桶列表

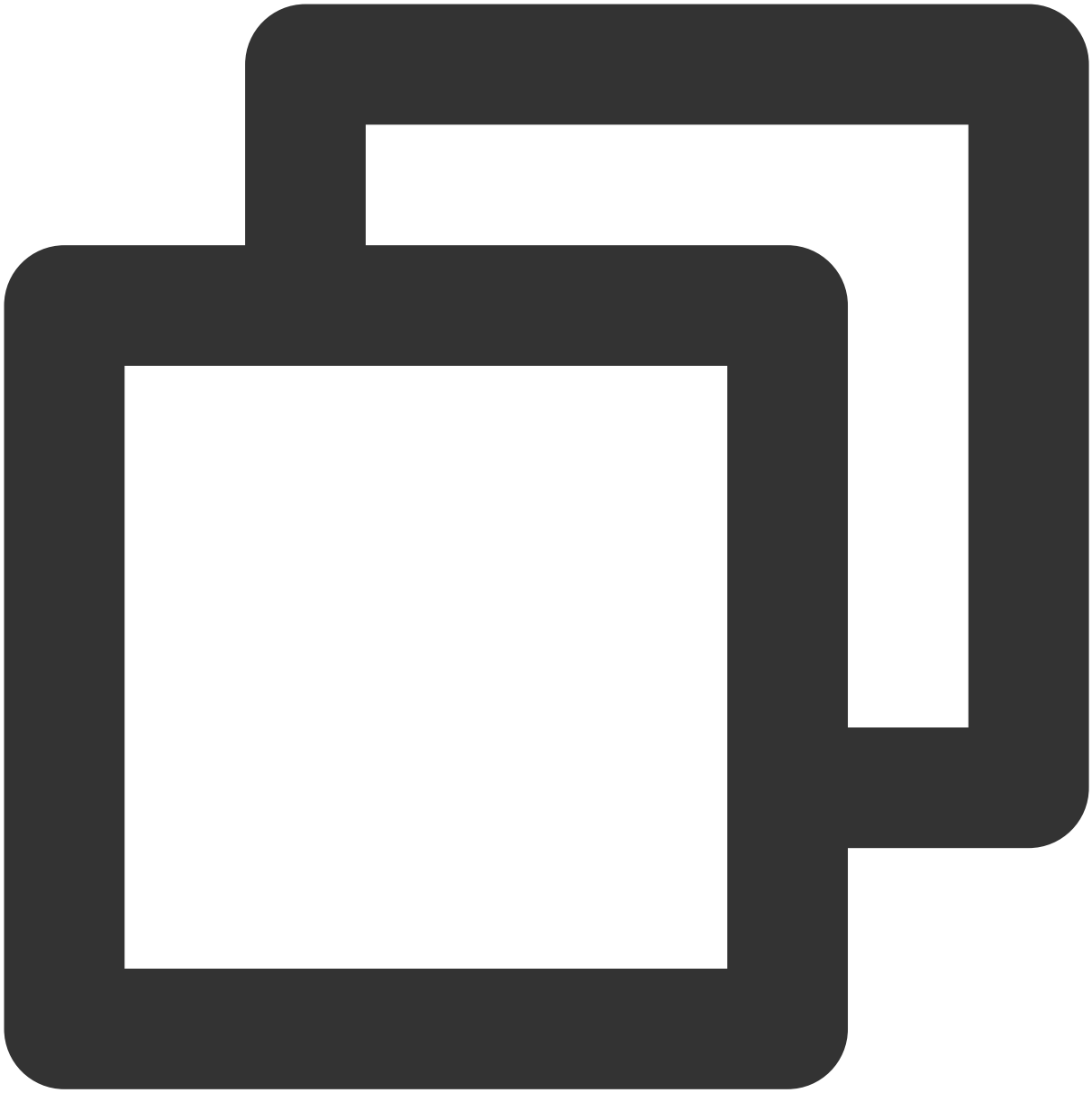
列出创建时间早于 `2022-1-20 15:10:45` 的存储桶。

请求



```
GET /?range=lt&create-time=1642662645 HTTP/1.1
Host: service.cos.myqcloud.com
User-Agent: curl/7.64.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDYv3vWrwkHXVDfqk*****&q-sign-time=1667
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 5566
Connection: keep-alive
Date: Thu, 03 Nov 2022 04:14:23 GMT
Server: tencent-cos
x-cos-request-id: NjM2MzQwMWZfMmJiMjM1MGFfYTc******

<ListAllMyBucketsResult>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>100000000001</DisplayName>
    </Owner>
    <Buckets>
        <Bucket>
            <Name>examplebucket-1250000000</Name>
            <Location>ap-beijing</Location>
            <CreationDate>2021-11-23T03:02:12Z</CreationDate>
            <BucketType>cos</BucketType>
        </Bucket>
    </Buckets>
</ListAllMyBucketsResult>
```

Bucket 接口
基本操作
PUT Bucket

最近更新时间：2023-12-20 16:21:42

功能描述

PUT Bucket 接口请求可以在指定账号下创建一个存储桶。该 API 接口不支持匿名请求，您需要使用带 Authorization 签名认证的请求才能创建新的 Bucket 。创建存储桶的用户默认成为存储桶的持有者。

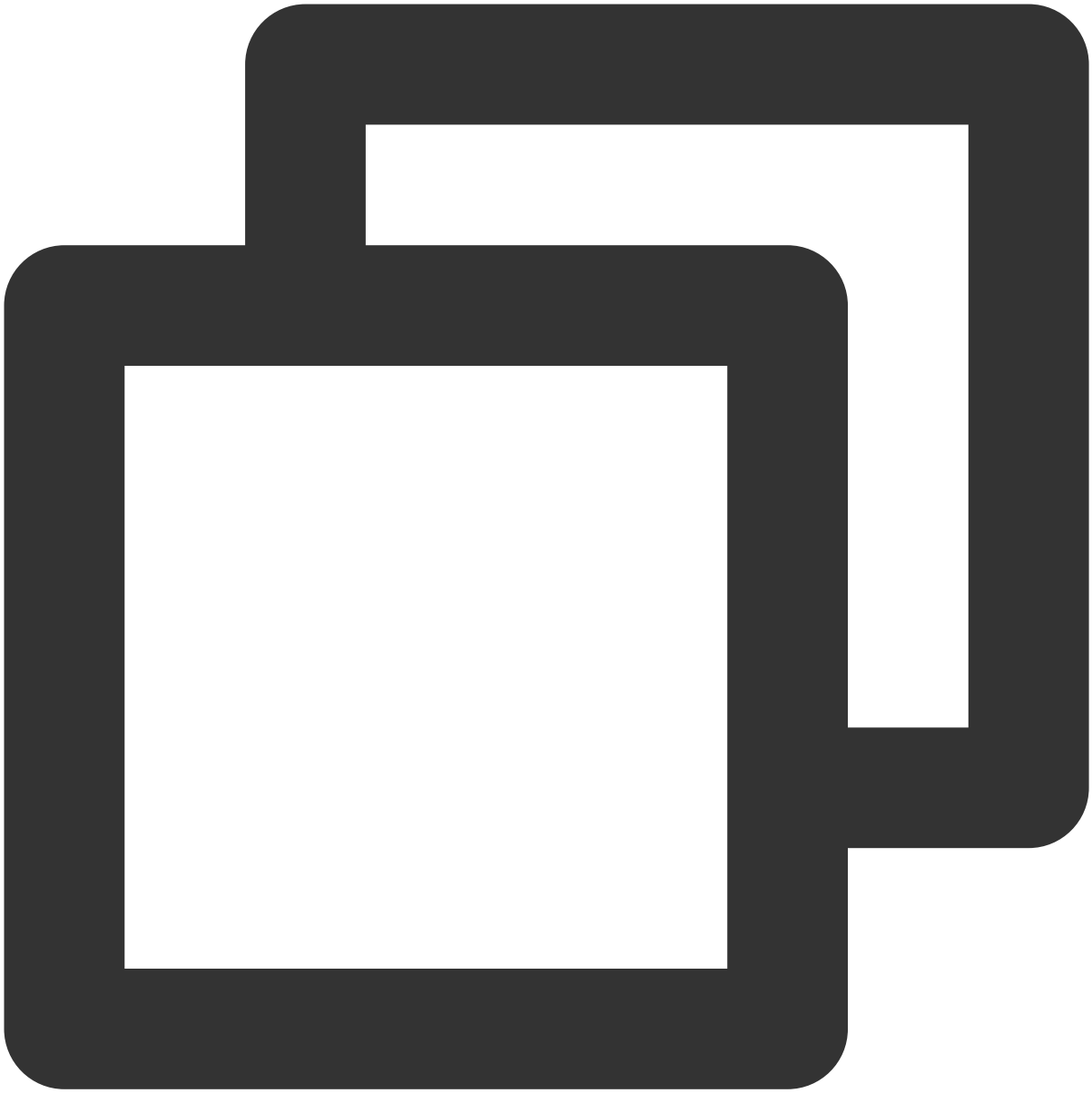
说明

创建存储桶时，如果没有指定访问权限，则默认使用私有读写（private）权限。
如需创建多 AZ 存储桶，那么应当通过请求体指示存储桶配置，否则无需传入请求体。

请求

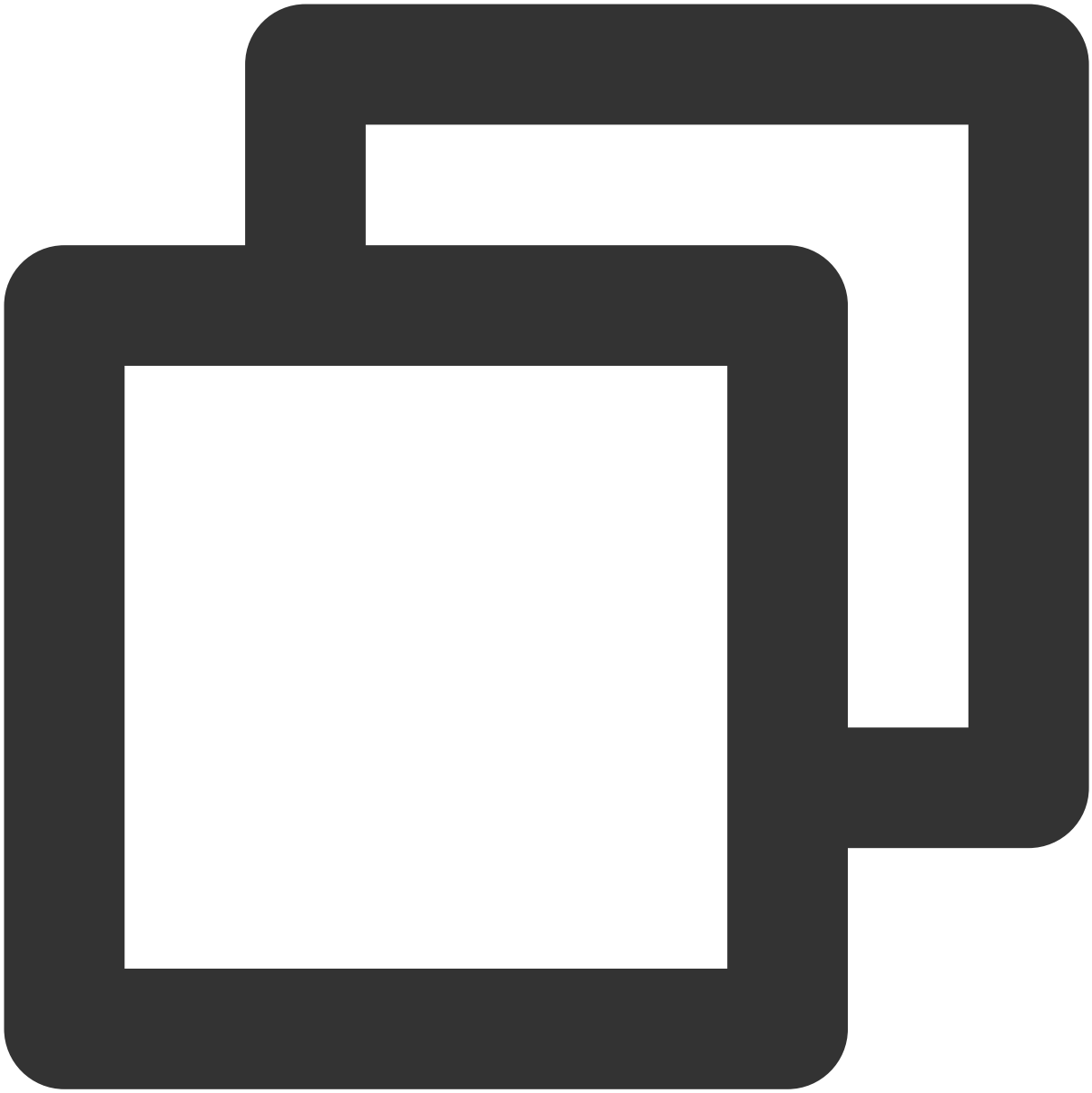
请求示例

示例一



```
PUT / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: 0
Authorization: Auth String
```

示例二



```
PUT / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
x-cos-tagging	在创建存储桶的同时，为存储桶添加标签，最多可设置50个标签。例如 `key1=value1&key2=value2` 。	string	否

访问控制列表（ACL）相关头部

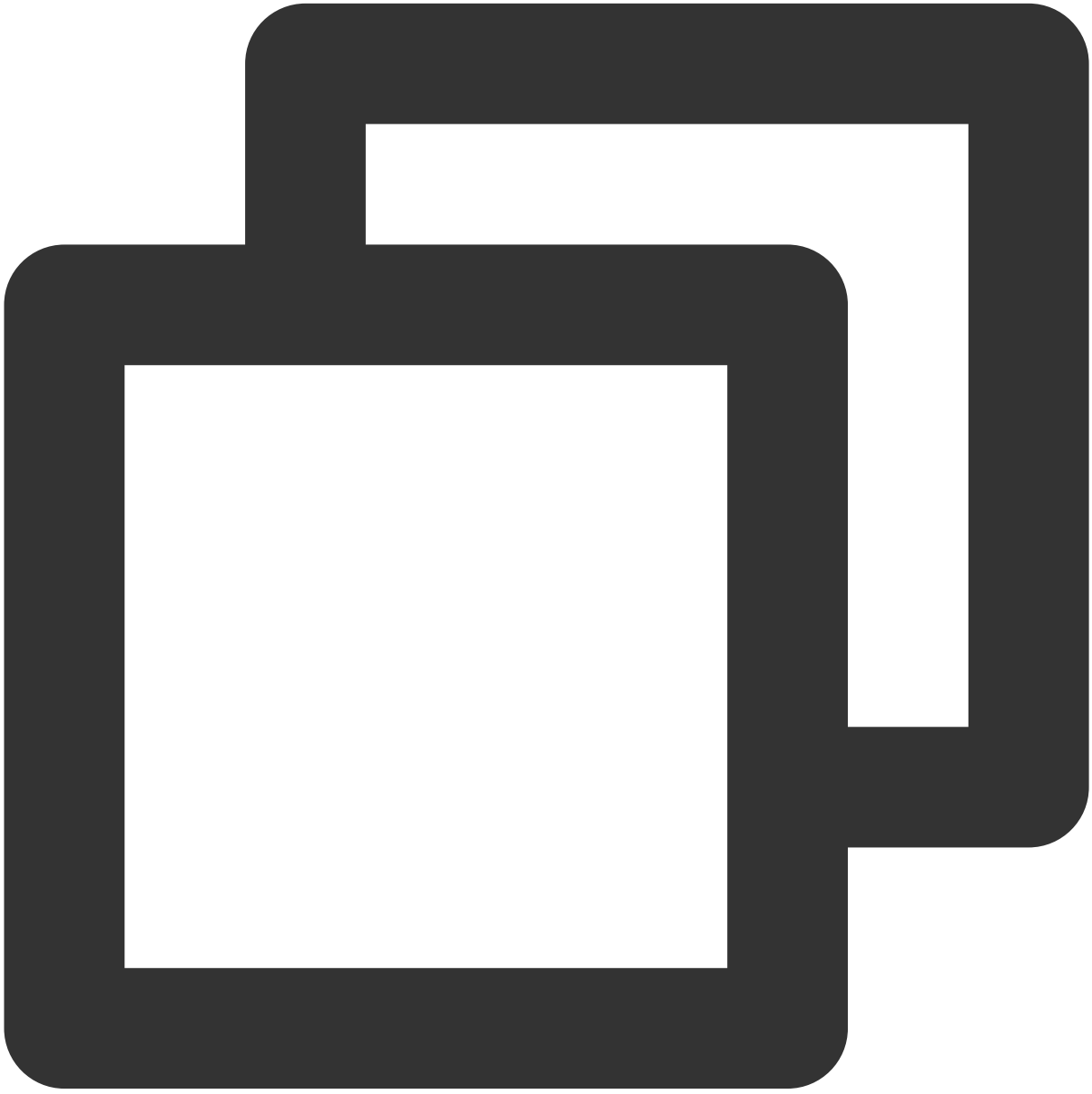
在创建存储桶时可以通过指定下列请求头部来设置存储桶的访问权限：

名称	描述	类型	是否必选
x-cos-acl	定义存储桶的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中存储桶的预设 ACL 部分，如 private，public-read 等，默认为 private	Enum	否
x-cos-grant-read	赋予被授权者读取存储桶的权限，格式为 id="[OwnerUin]"，如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write	赋予被授权者写入存储桶的权限，格式为 id="[OwnerUin]"，如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取存储桶的访问控制列表（ACL）和存储桶策略（Policy）的权限，格式为 id="[OwnerUin]"，如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入存储桶的访问控制列表（ACL）和存储桶策略（Policy）的权限，格式为 id="[OwnerUin]"，如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否

x-cos-grant-full-control	赋予被授权者操作存储桶的所有权限，格式为 id="[OwnerUin]"，如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否

请求体

仅当需要创建多 AZ 存储桶时提交 **application/xml** 请求数据，包含创建存储桶的配置信息，否则无需传入请求体。



```
<CreateBucketConfiguration>
    <BucketAZConfig>string</BucketAZConfig>
</CreateBucketConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
CreateBucketConfiguration	无	包含 PUT Bucket 操作的所有请求信息	Container	否

Container 节点 CreateBucketConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
BucketAZConfig	CreateBucketConfiguration	存储桶 AZ 配置，指定为 MAZ 以创建多 AZ 存储桶。	string	是

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

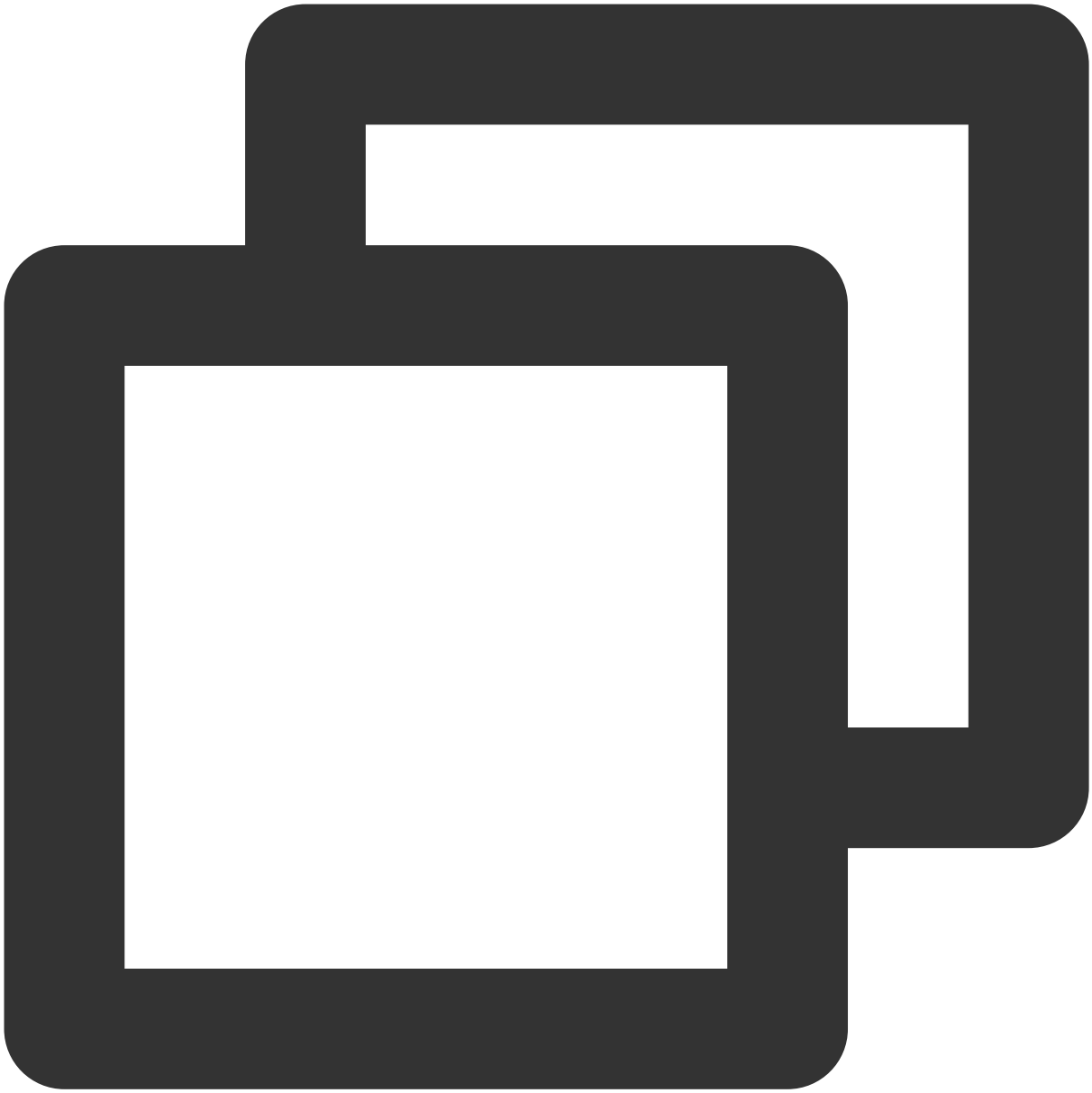
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

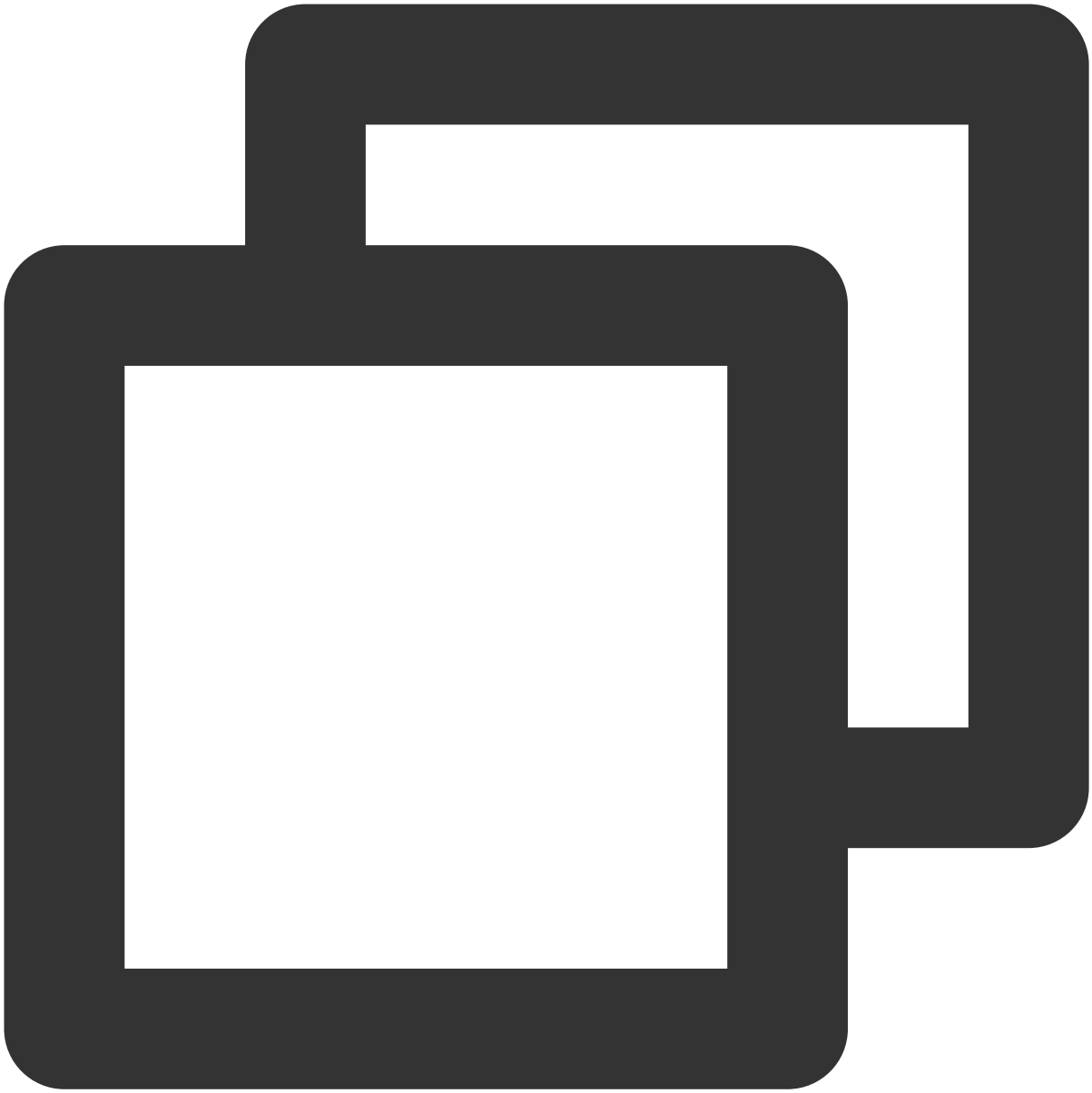
案例一：简单案例（单 AZ 存储桶）

请求



```
PUT / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sun, 26 May 2019 14:51:38 GMT
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

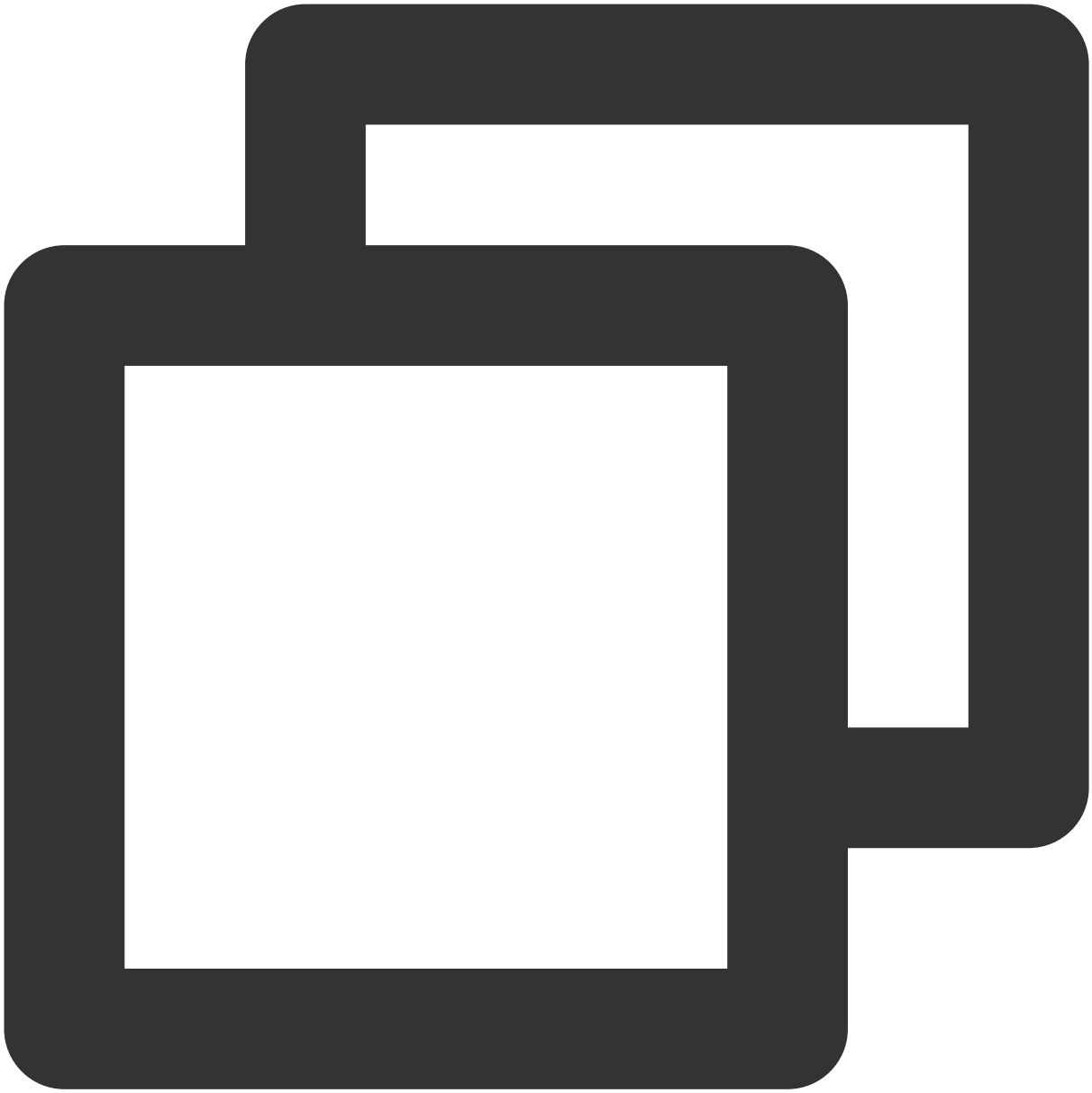
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Sun, 26 May 2019 14:51:37 GMT
Server: tencent-cos
x-cos-request-id: NWNlYWE3ZjlfZDQyNzVkNjRfMzg1N18yNzFh****
```

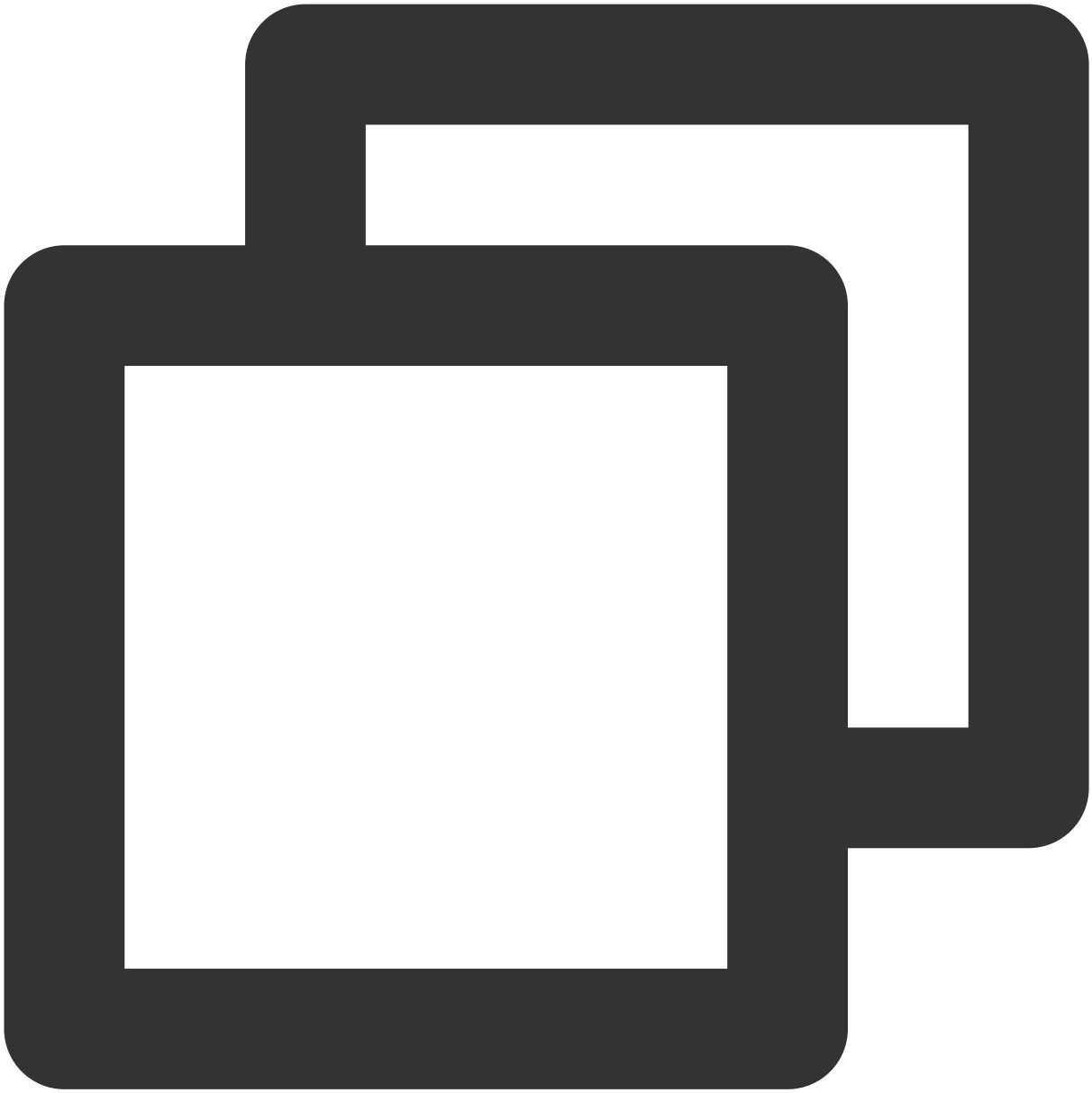
案例二：指定公有读并授权特定用户读取权限和写入对象

请求



```
PUT / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 14 Jun 2019 13:48:59 GMT
x-cos-acl: public-read
x-cos-grant-write: id="100000000002"
x-cos-grant-read-acp: id="100000000002"
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

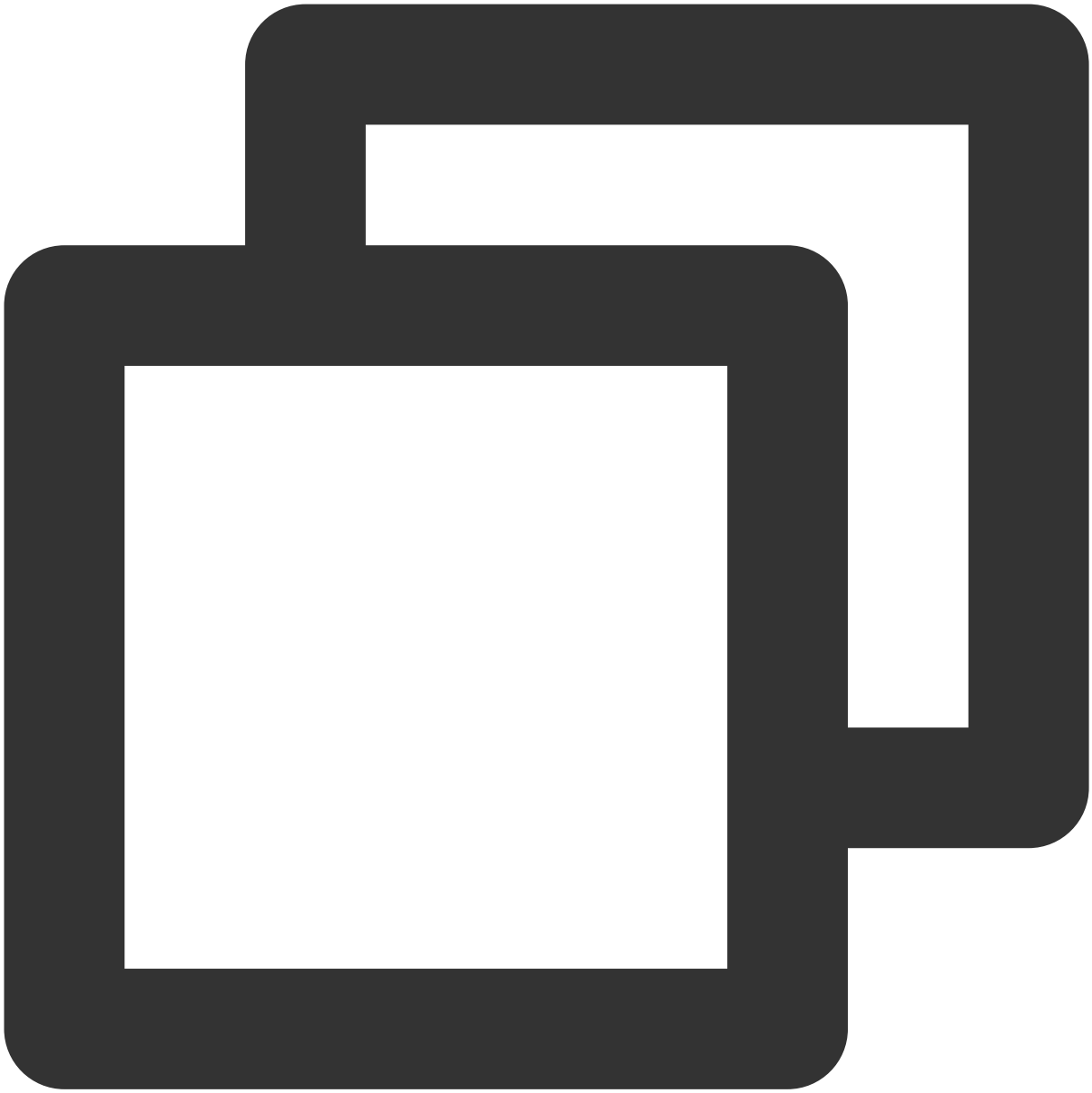
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 14 Jun 2019 13:49:00 GMT
Server: tencent-cos
x-cos-request-id: NWQwM2E1Y2NfZjBhODBiMDlfOTM1YV83NDRi****
```

案例三：创建多 AZ 存储桶

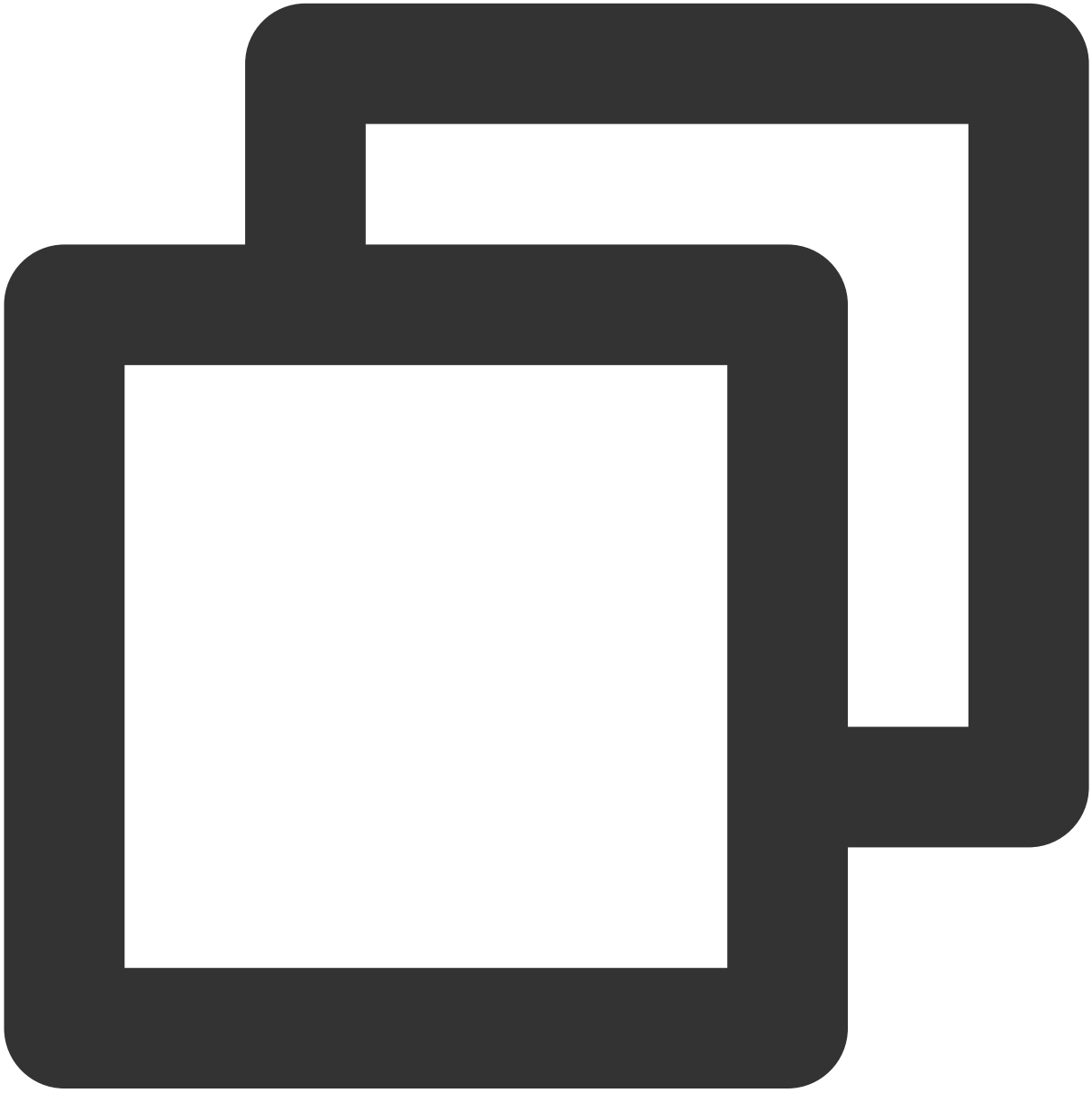
请求



```
PUT / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jun 2020 06:06:09 GMT
Content-Type: application/xml
Content-Length: 96
Content-MD5: R1ES/YbddhKJK/wcN+f4yg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

```
<CreateBucketConfiguration>
    <BucketAZConfig>MAZ</BucketAZConfig>
</CreateBucketConfiguration>
```

响应



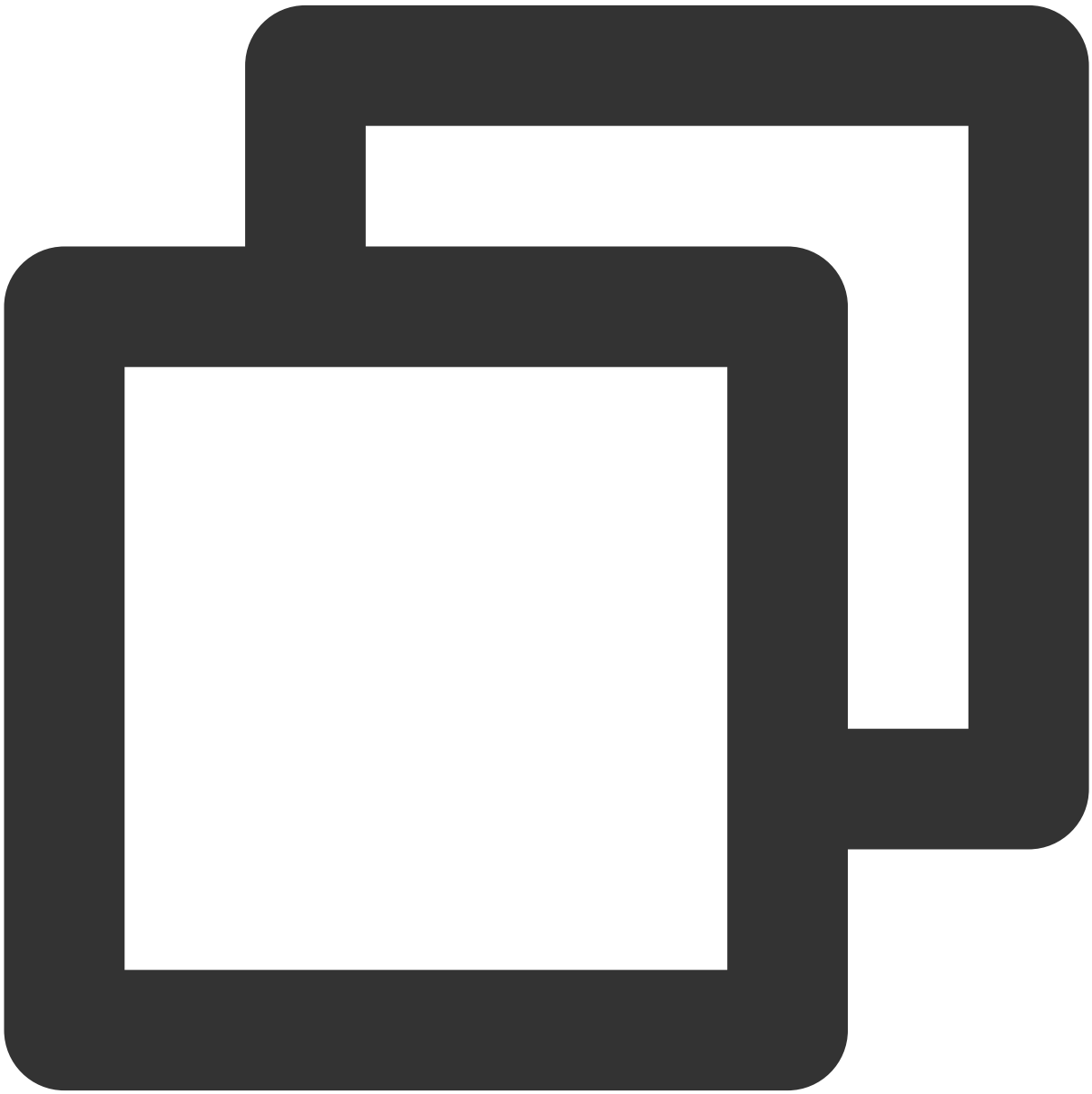
```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Thu, 04 Jun 2020 06:06:10 GMT
Server: tencent-cos
```

```
x-cos-request-id: NWVkODhmNTFfM2JiODJhMDlfMjg4NmFfMzA5ZmE2****
```

案例四：创建存储桶同时添加存储桶标签

以下请求示例，用于在北京地域创建存储桶 `examplebucket-1250000000`，同时为存储桶添加两个存储桶标签 `<a,a>` 和 `<b,b>`。

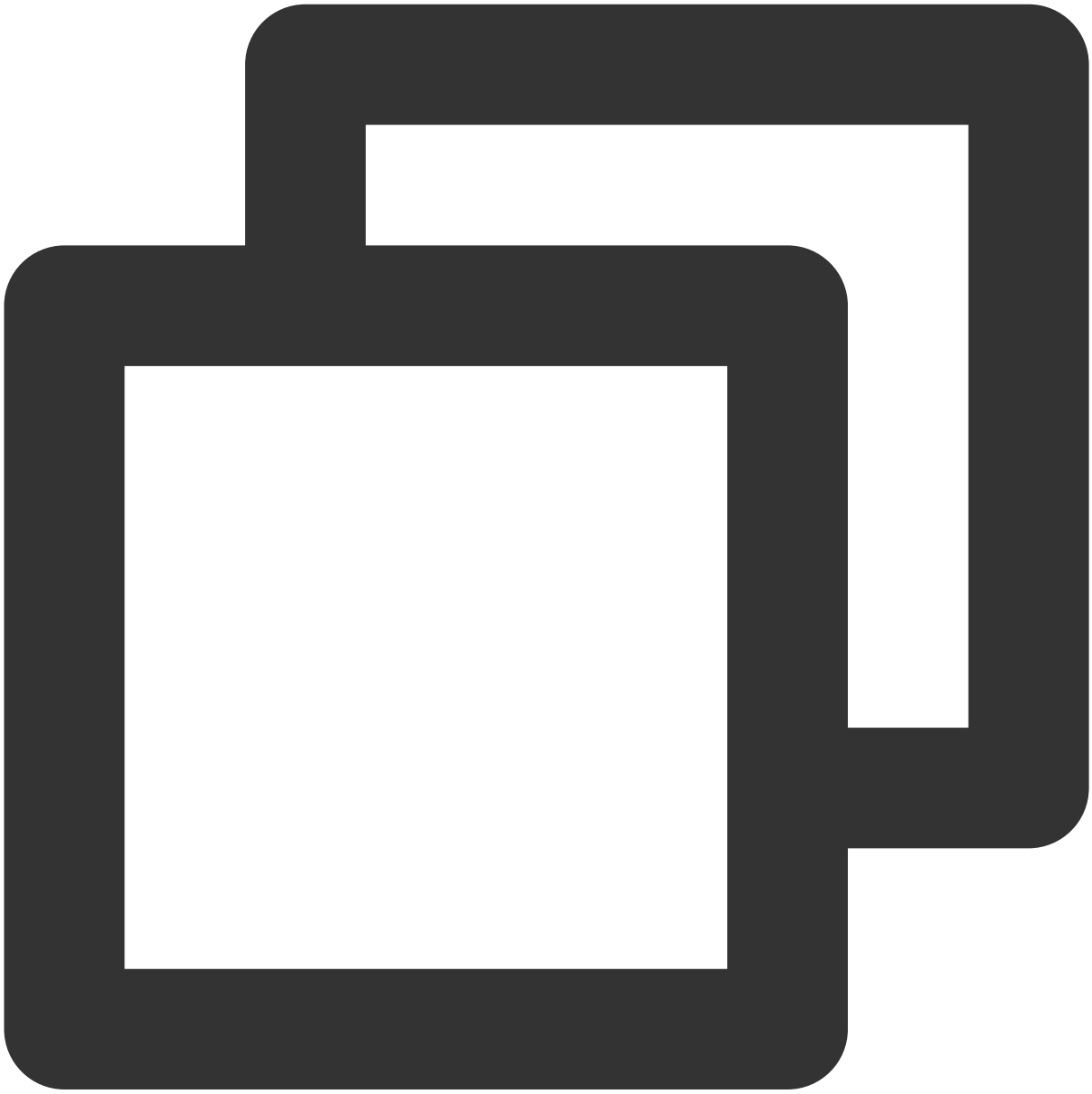
请求



```
PUT / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
x-cos-tagging: a=a&b=b
Authorization: q-sign-algorithm=sha1&q-ak=AKIDYv3vWrwkHXVDf******&q-sign-time=16674
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Thu, 03 Nov 2022 03:43:30 GMT
Server: tencent-cos
x-cos-request-id: NjM2MzM4ZTJfZDRiNTE0M********
```

版权所有：腾讯云计算（北京）有限责任公司

GET Bucket（List Objects）

最近更新时间：2023-12-20 18:07:41

功能描述

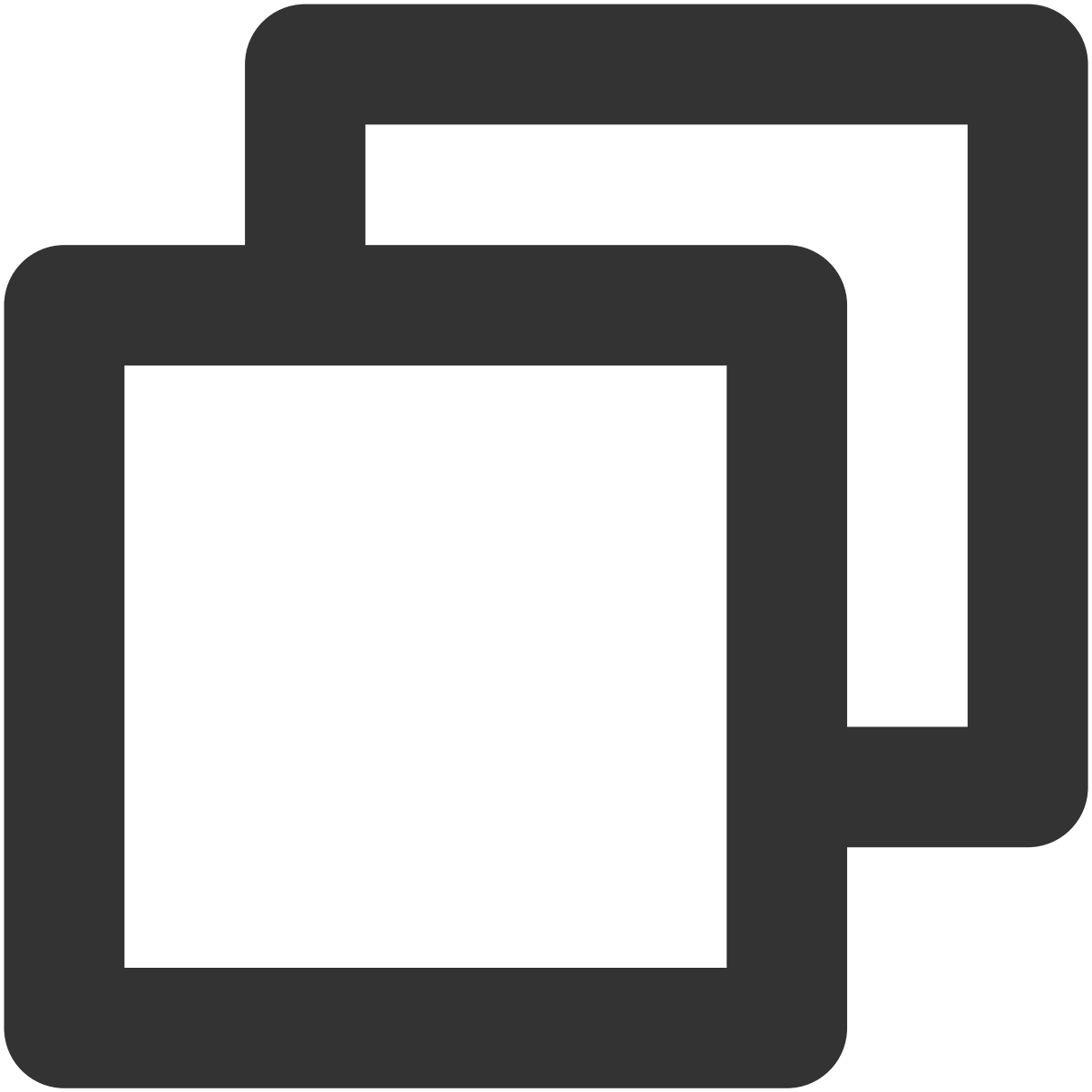
GET Bucket 请求等同于 List Objects 请求，可以列出该存储桶内的部分或者全部对象。该 API 的请求者需要对存储桶有读取权限。

说明

如果您往存储桶中上传了一个对象，并立即调用 GET Bucket 接口，由于此接口的最终一致性特性，返回的结果中可能不会包含您刚刚上传的对象。

请求

请求示例



```
GET / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

名称	描述	类型	是否必选
prefix	对象键匹配前缀，限定响应中只包含指定前缀的对象键	string	否
delimiter	一个字符的分隔符，用于对对象键进行分组。所有对象键中从 prefix 或从头（如未指定 prefix）到首个 delimiter 之间相同的部分将作为 CommonPrefixes 下的一个 Prefix 节点。被分组的对象键不再出现在后续对象列表中，具体场景和用法可参考下面的实际案例	string	否
encoding-type	规定返回值的编码方式，可选值：url，代表返回的对象键为 URL 编码（百分号编码）后的值，例如“腾讯云”将被编码为 `%E8%85%BE%E8%AE%AF%E4%BA%91`	string	否
marker	起始对象键标记，从该标记之后（不含）按照 UTF-8 字典序返回对象键条目	string	否
max-keys	单次返回最大的条目数量，默认值为1000，最大为1000 **注意：**该参数会限制每一次 List 操作返回的最大条目数，COS 在每次 List 操作中将返回不超过 max-keys 所设定数值的条目（即 CommonPrefixes 和 Contents 的总和），如果单次响应中未列出所有对象，COS 会返回 NextMarker 节点，其值作为您下次 List 请求的 marker 参数，以便您列出后续对象	integer	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

响应

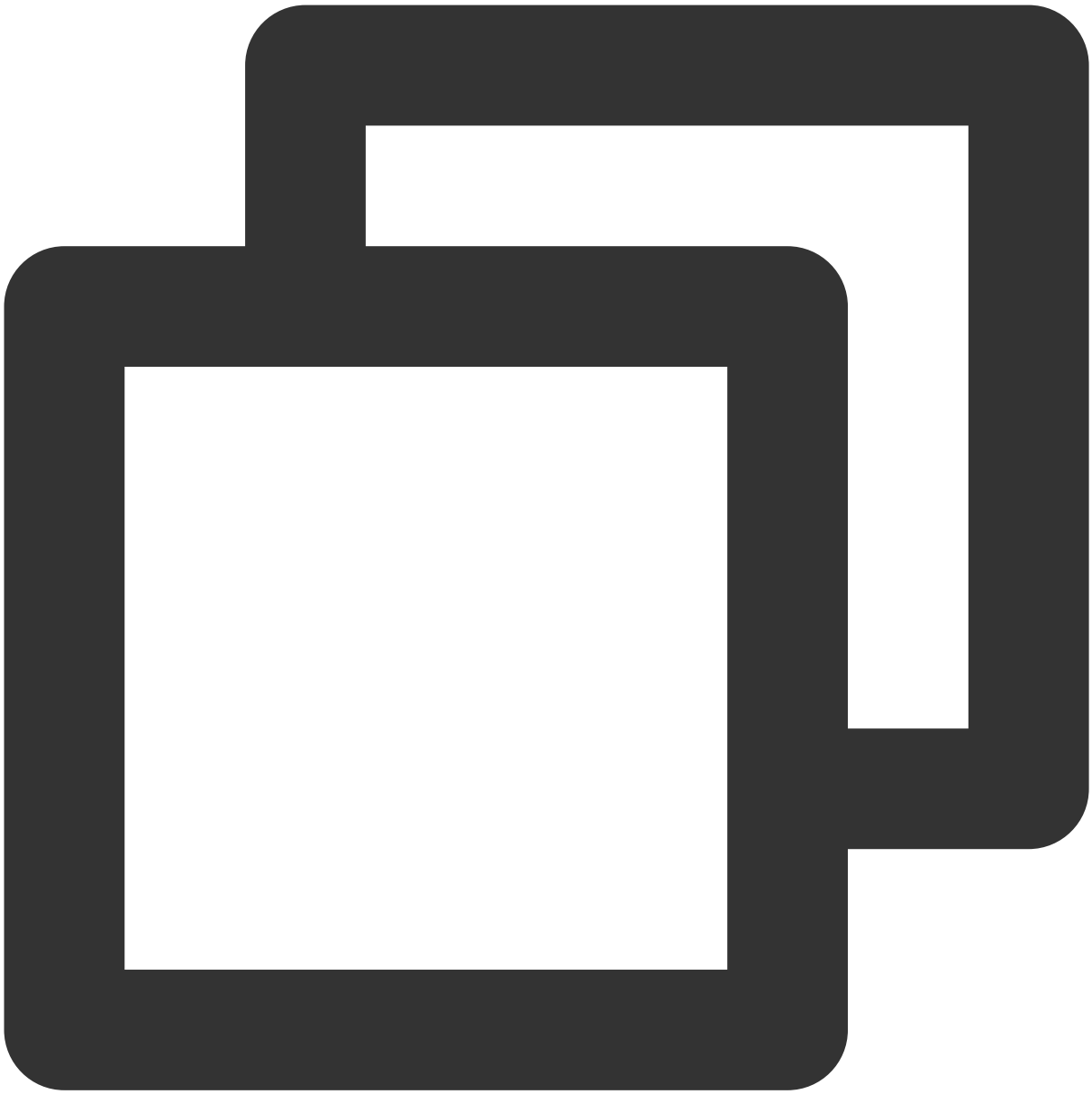
响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

名称	描述	类型
x-cos-bucket-region	存储桶所在地域。枚举值请参见 地域和访问域名 文档，例如 ap-beijing，ap-hongkong，eu-frankfurt 等	Enum

响应体

查询成功，返回 **application/xml** 数据，包含存储桶中的对象信息。不同场景下的响应体请参见下方的实际案例。



```
<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
            <Name>string</Name>
            <EncodingType>string</EncodingType>
            <Prefix>string</Prefix>
            <Marker>string</Marker>
            <MaxKeys>integer</MaxKeys>
            <Delimiter>string</Delimiter>
            <IsTruncated>boolean</IsTruncated>
```

```
            <NextMarker>string</NextMarker>
            <CommonPrefixes>
                <Prefix>string</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>string</Prefix>
            </CommonPrefixes>
            <Contents>
                <Key>string</Key>
                <LastModified>date</LastModified>
                <ETag>string</ETag>
                <Size>integer</Size>
                <Owner>
                    <ID>string</ID>
                    <DisplayName>string</DisplayName>
                </Owner>
                <StorageClass>Enum</StorageClass>
                <StorageTier>Enum</StorageTier>
                <RestoreStatus>Enum</RestoreStatus>
            </Contents>
            <Contents>
                <Key>string</Key>
                <LastModified>date</LastModified>
                <ETag>string</ETag>
                <Size>integer</Size>
                <Owner>
                    <ID>string</ID>
                    <DisplayName>string</DisplayName>
                </Owner>
                <StorageClass>Enum</StorageClass>
                <StorageTier>Enum</StorageTier>
                <RestoreStatus>Enum</RestoreStatus>
            </Contents>
</ListBucketResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
ListBucketResult	无	保存 GET Bucket 结果的所有信息	Container

Container 节点 ListBucketResult 的内容：

节点名称（关键字）	父节点	描述	类型
Name	ListBucketResult	存储桶的名称，格式为 `<BucketName-APPID>` ，	string

		例如 `examplebucket-1250000000`	
EncodingType	ListBucketResult	编码格式，对应请求中的 encoding-type 参数，且仅当请求中指定了 encoding-type 参数才会返回该节点	string
Prefix	ListBucketResult	对象键匹配前缀，对应请求中的 prefix 参数	string
Marker	ListBucketResult	起始对象键标记，从该标记之后（不含）按照 UTF-8 字典序返回对象键条目，对应请求中的 marker 参数	string
MaxKeys	ListBucketResult	单次响应返回结果的最大条目数量，对应请求中的 max-keys 参数	integer
Delimiter	ListBucketResult	分隔符，对应请求中的 delimiter 参数，且仅当请求中指定了 delimiter 参数才会返回该节点	string
IsTruncated	ListBucketResult	响应条目是否被截断，布尔值，例如 true 或 false	boolean
NextMarker	ListBucketResult	仅当响应条目有截断（IsTruncated 为 true）才会返回该节点，该节点的值为当前响应条目中的最后一个对象键，当需要继续请求后续条目时，将该节点的值作为下一次请求的 marker 参数传入	string
CommonPrefixes	ListBucketResult	从 prefix 或从头（如未指定 prefix）到首个 delimiter 之间相同的部分，定义为 Common Prefix。仅当请求中指定了 delimiter 参数才有可能返回该节点	Container
Contents	ListBucketResult	对象条目	Container

Container 节点 CommonPrefixes 的内容：

节点名称（关键字）	父节点	描述	类型
Prefix	ListBucketResult.CommonPrefixes	单条 Common Prefix 的前缀	string

Container 节点 Contents 的内容：

节点名称（关键字）	父节点	描述	类型
Key	ListBucketResult.Contents	对象键	string
LastModified	ListBucketResult.Contents	对象最后修改时间，为 ISO8601 格式，例如 2019-05-24T10:56:40Z	date
ETag	ListBucketResult.Contents	对象的实体标签（Entity Tag），是对象被创建时标识对象内容的信息标签，可用于检查对象的	string

版权所有：腾讯云计算（北京）有限责任公司

		内容是否发生变化， 例 如“8e0b617ca298a564c3331da28dcb50df”， 此头部并不一定返回对象的 MD5 值，而是根据对象上传和加密方式而有所不同	
Size	ListBucketResult.Contents	对象大小，单位为 Byte	integer
Owner	ListBucketResult.Contents	对象持有者信息	Container
StorageClass	ListBucketResult.Contents	对象存储类型。枚举值请参见 存储类型 文档，例如 STANDARD_IA，ARCHIVE	Enum
StorageTier	ListBucketResult.Contents	当对象存储类型为智能分层存储时，指示对象当前所处的存储层，枚举值：FREQUENT（标准层），INFREQUENT（低频层）。仅当 StorageClass 为 INTELLIGENT_TIERING（智能分层）时才会返回该节点	Enum
RestoreStatus	ListBucketResult.Contents	当对象存储类型为归档存储或深度归档存储时，指示对象的回热状态，支持枚举值 ONGOING、DONE、FAILED。当未向对象发起回热请求时，该字段为空；当发起回热请求后，回热进行中为 ONGING，回热已完成为 DONE，回热失败为 FAILED。仅当 StorageClass 为 ARCHIVE （归档）或 DEEP_ARCHIVE（深度归档）时才会返回该节点	Enum

Container 节点 Contents.Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	ListBucketResult.Contents.Owner	对象持有者的 APPID	string
DisplayName	ListBucketResult.Contents.Owner	对象持有者的名称	string

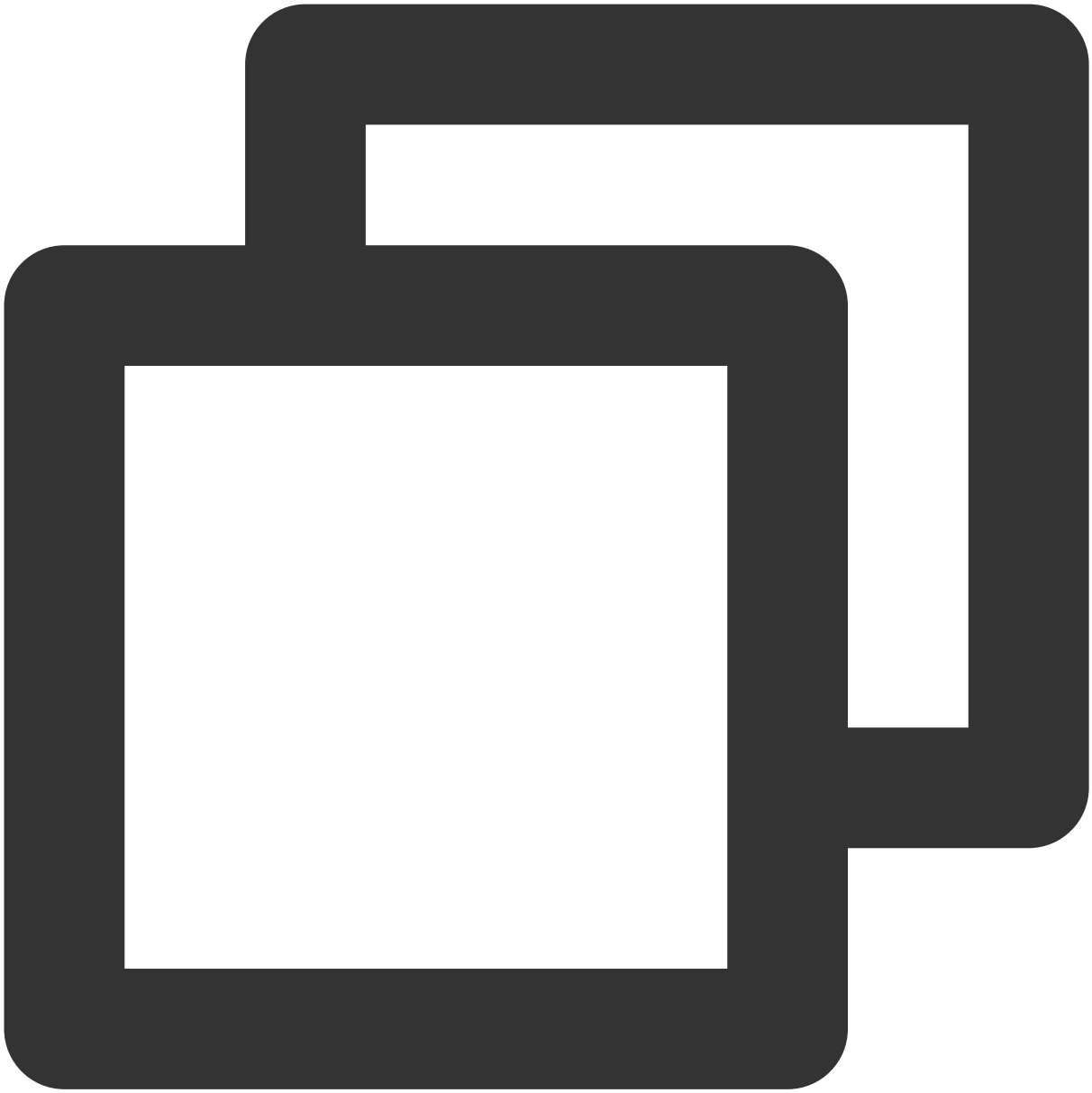
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

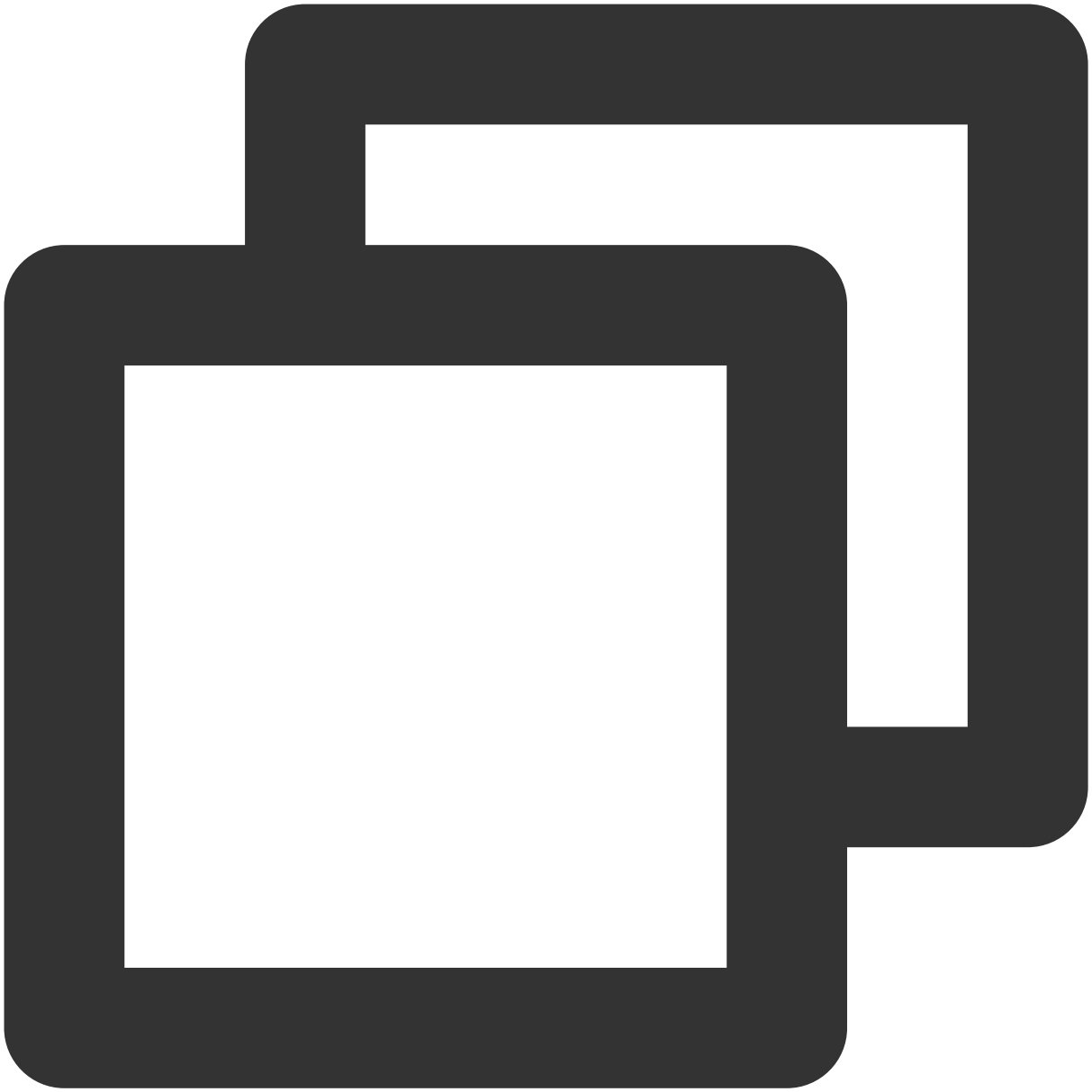
案例一：简单案例

请求



```
GET / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 03:37:41 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 2931
Connection: close
Date: Thu, 10 Dec 2020 03:37:41 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZkMTk4MDVfNjViODJhMDlfNDZkYl8xNzU0****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```

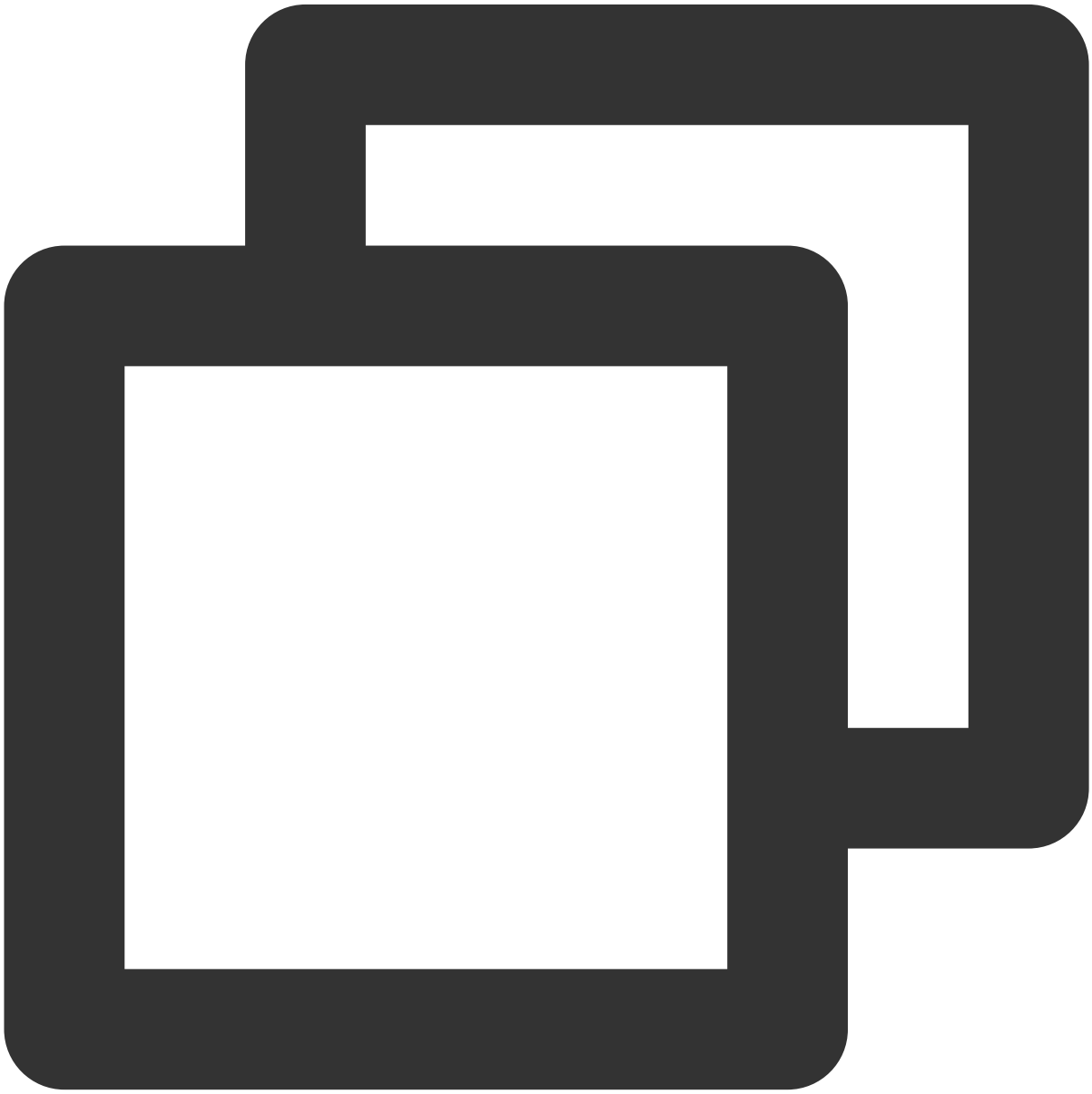
版权所有：腾讯云计算（北京）有限责任公司

```
        <Name>examplebucket-1250000000</Name>
        <Prefix/>
        <Marker/>
        <MaxKeys>1000</MaxKeys>
        <IsTruncated>false</IsTruncated>
        <Contents>
            <Key>example-folder-1/example-object-1.jpg</Key>
            <LastModified>2020-12-10T03:37:30.000Z</LastModified>
            <ETag>&amp;quot;f173c1199e3d3b53dd91223cae16fb42&amp;quot;</ETag>
            <Size>37</Size>
            <Owner>
                      <ID>1250000000</ID>
                      <DisplayName>1250000000</DisplayName>
            </Owner>
            <StorageClass>STANDARD</StorageClass>
        </Contents>
        <Contents>
            <Key>example-folder-1/example-object-2.jpg</Key>
            <LastModified>2020-12-10T03:37:30.000Z</LastModified>
            <ETag>&amp;quot;c9d28698978bb6fef6c1ed1c439a17d3&amp;quot;</ETag>
            <Size>37</Size>
            <Owner>
                      <ID>1250000000</ID>
                      <DisplayName>1250000000</DisplayName>
            </Owner>
                  <StorageClass>INTELLIGENT_TIERING</StorageClass>
                  <StorageTier>FREQUENT</StorageTier>
        </Contents>
                  ...
        <Contents>
            <Key>example-object-2.jpg</Key>
            <LastModified>2020-12-10T03:37:30.000Z</LastModified>
            <ETag>&amp;quot;51370fc64b79d0d3c7c609635be1c41f&amp;quot;</ETag>
            <Size>20</Size>
            <Owner>
                      <ID>1250000000</ID>
                      <DisplayName>1250000000</DisplayName>
            </Owner>
            <StorageClass>STANDARD_IA</StorageClass>
        </Contents>
        <Contents>
            <Key>example-object-3.jpg</Key>
            <LastModified>2020-12-10T04:37:30.000Z</LastModified>
            <ETag>&amp;quot;51370fc64b79d0d3c7c609635be1c41f&amp;quot;</ETag>
            <Size>20</Size>
            <Owner>
                      <ID>1250000000</ID>
```

```
                    <DisplayName>1250000000</DisplayName>
            </Owner>
            <StorageClass>ARCHIVE</StorageClass>
            <RestoreStatus>DONE</RestoreStatus>
        </Contents>
</ListBucketResult>
```

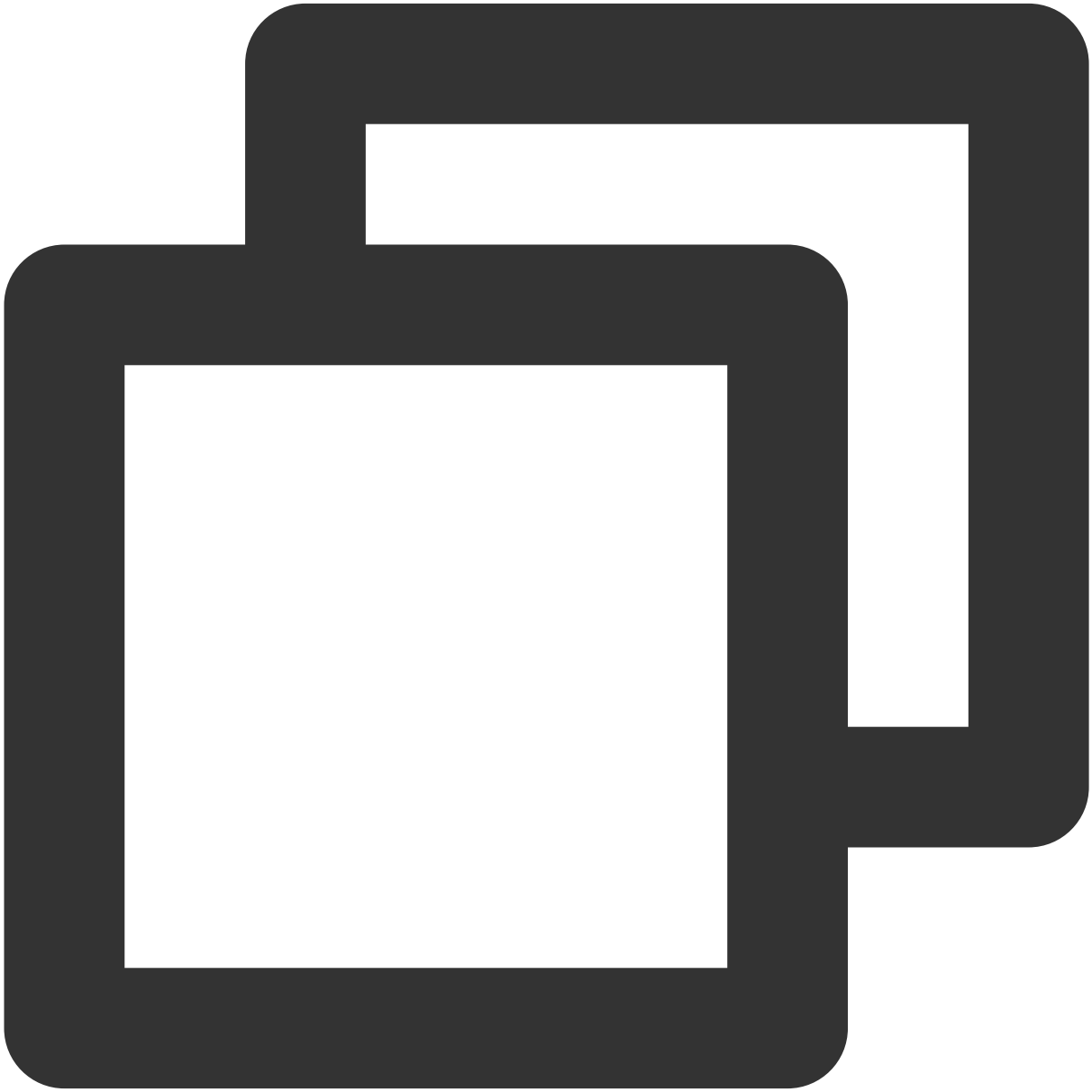
案例二：带 encoding-type 参数（对象键使用 URL 编码）

请求



```
GET /?encoding-type=url HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 08 Dec 2020 13:54:16 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



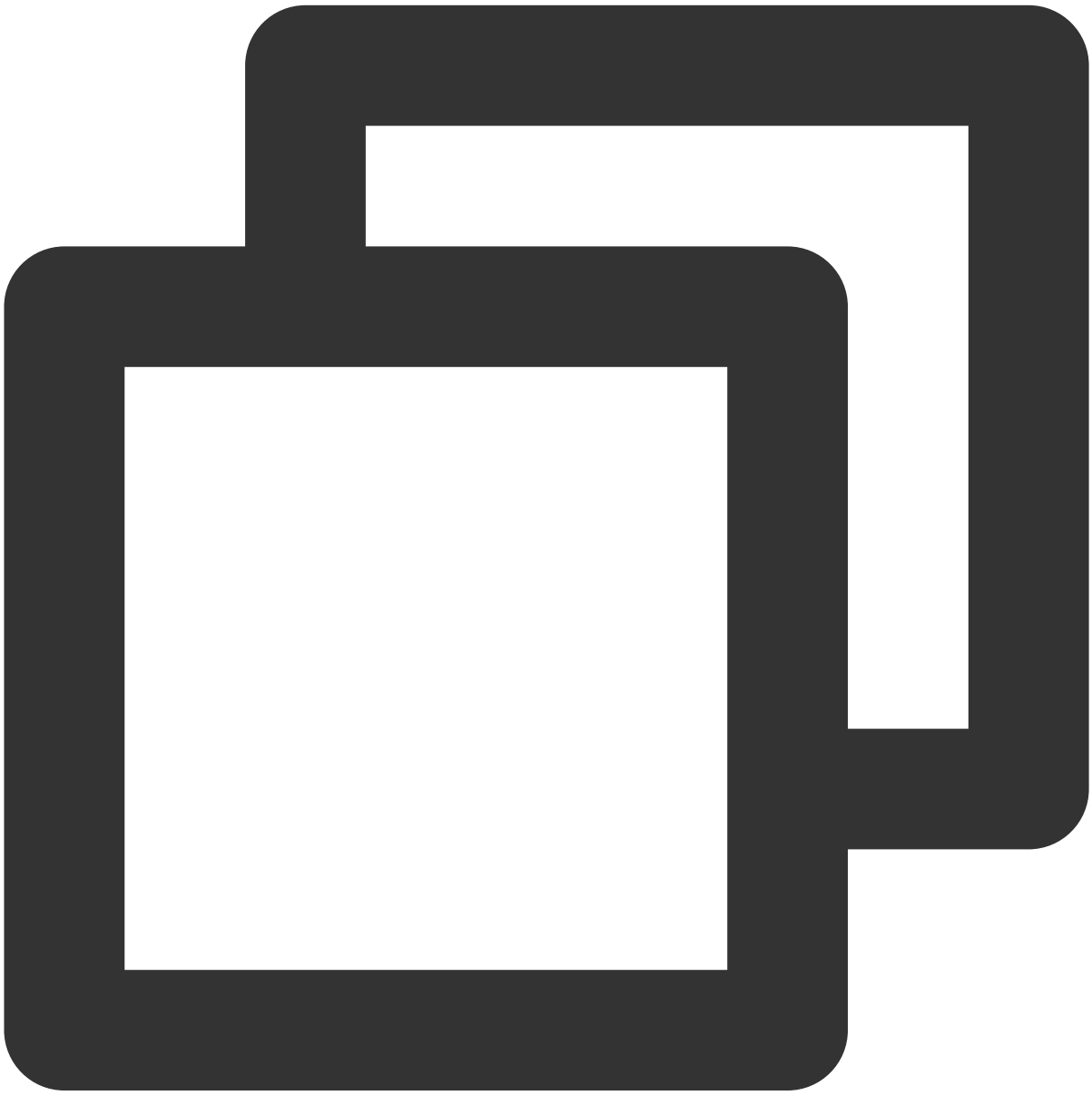
```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 1220
Connection: close
Date: Wed, 09 Dec 2020 15:28:29 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZkMGVkMWRfZjdjNzJhMDlfMjJhYTlfYmJk****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
      <Name>examplebucket-1250000000</Name>
      <EncodingType>url</EncodingType>
      <Prefix/>
      <Marker/>
      <MaxKeys>1000</MaxKeys>
      <IsTruncated>false</IsTruncated>
      <Contents>
                <Key>Tencent%20Cloud.jpg</Key>
                <LastModified>2020-12-09T15:28:19.000Z</LastModified>
                <ETag>&quot;ee8de918d05640145b18f70f4c3aa602&quot;</ETag>
                <Size>16</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
      </Contents>
      <Contents>
                <Key>%E7%85%A7%E7%89%87/2020%E5%B9%B4/IMG0001.jpg</Key>
                <LastModified>2020-12-09T15:28:19.000Z</LastModified>
                <ETag>&quot;ee8de918d05640145b18f70f4c3aa602&quot;</ETag>
                <Size>16</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
      </Contents>
      <Contents>
                <Key>%E8%85%BE%E8%AE%AF%E4%BA%91.jpg</Key>
                <LastModified>2020-12-09T15:28:19.000Z</LastModified>
                <ETag>&quot;ee8de918d05640145b18f70f4c3aa602&quot;</ETag>
                <Size>16</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
```

```
            </Contents>
</ListBucketResult>
```

案例三：带 delimiter 参数（列出根目录下的对象和子目录）

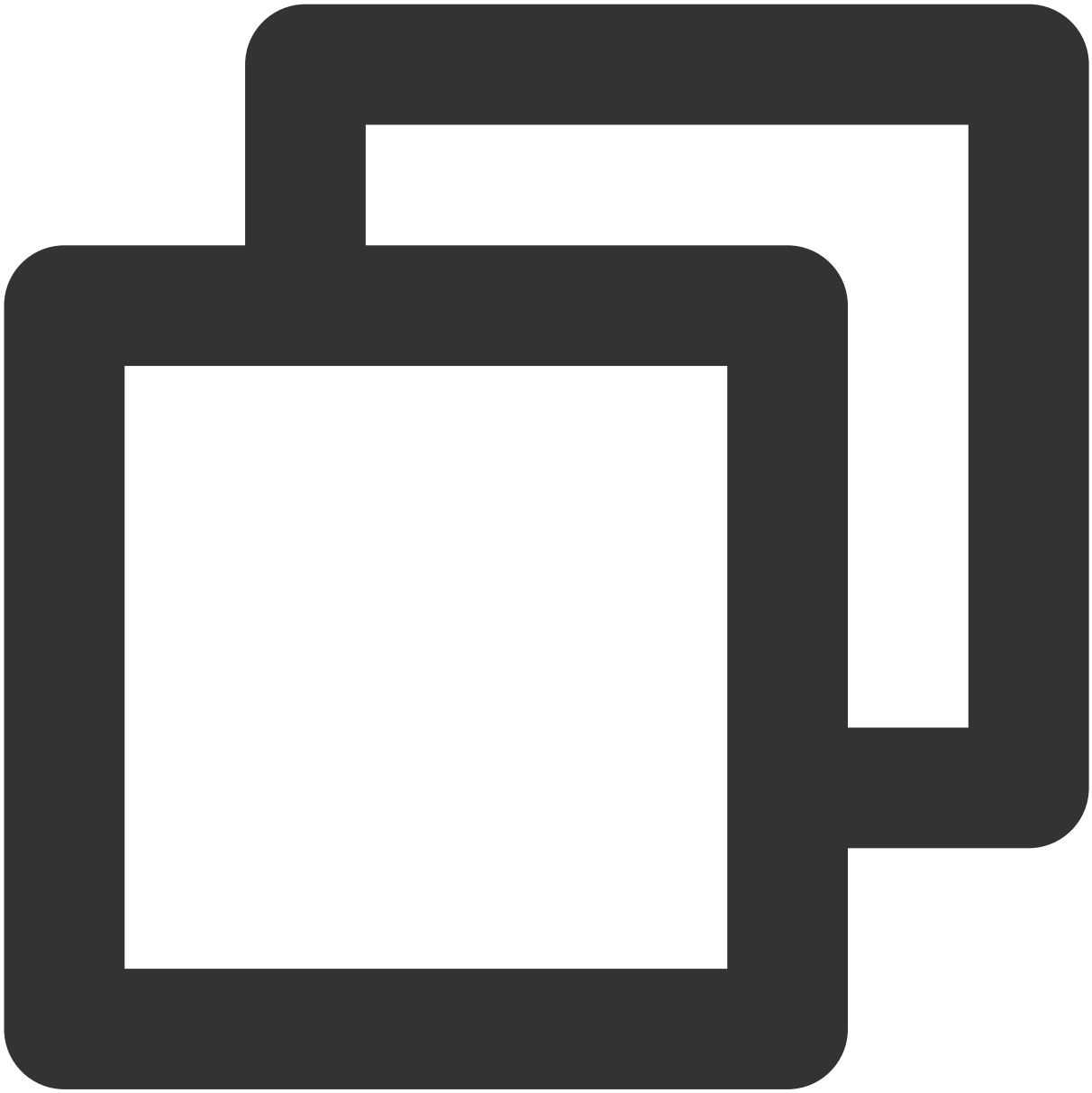
请求



```
GET /?delimiter=%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 03:37:41 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
```

```
Connection: close
```

响应

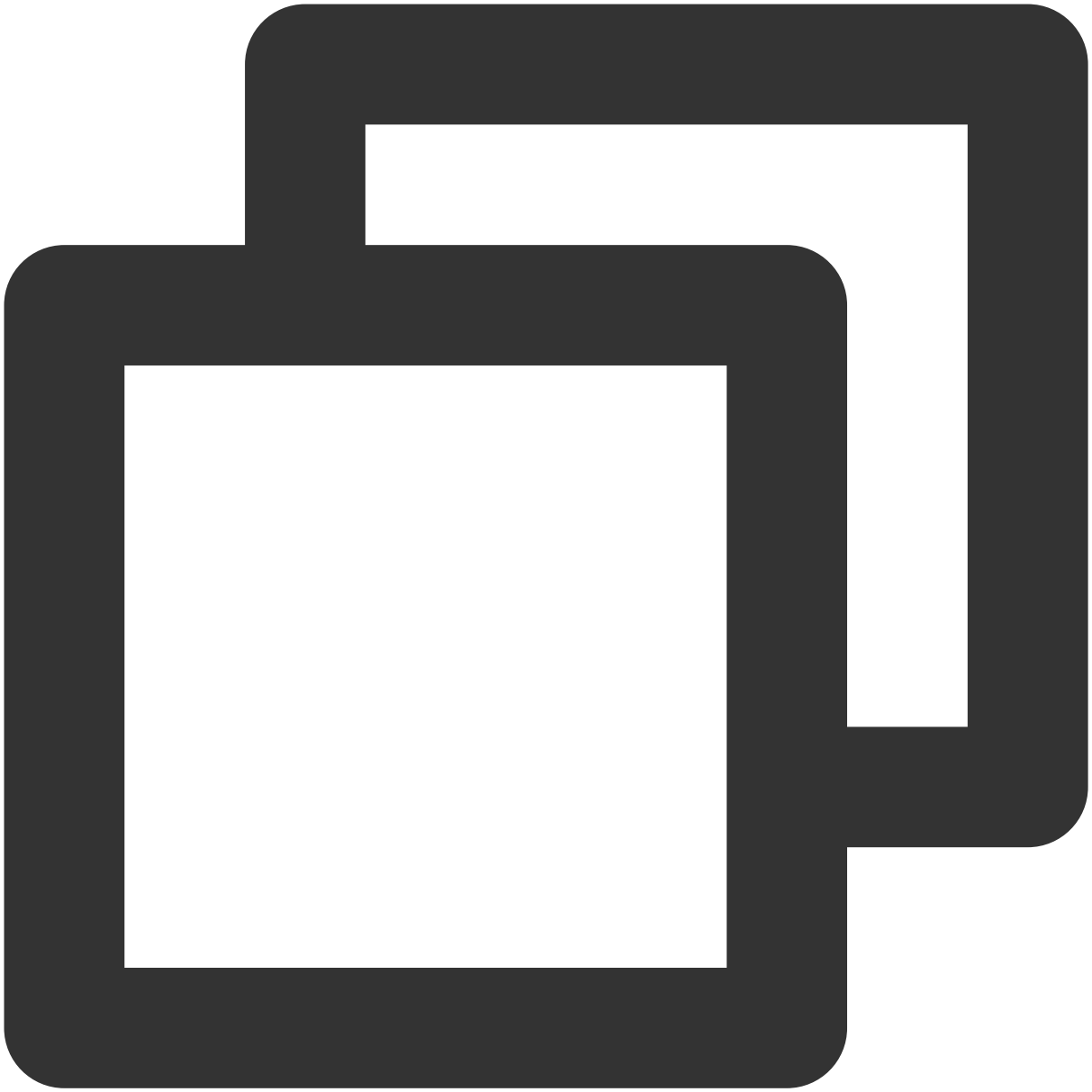


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1011
Connection: close
Date: Thu, 10 Dec 2020 03:37:41 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
```

```
x-cos-request-id: NWZkMTk4MDVfNjRiMDJhMDlfOTZjZV8xOTgw****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <Marker/>
            <MaxKeys>1000</MaxKeys>
            <Delimiter>/</Delimiter>
            <IsTruncated>false</IsTruncated>
            <CommonPrefixes>
                <Prefix>example-folder-1/</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>example-folder-2/</Prefix>
            </CommonPrefixes>
            <Contents>
                <Key>example-object-1.jpg</Key>
                <LastModified>2020-12-10T03:37:30.000Z</LastModified>
                <ETag>&quot;0f0cd12c48979d1bf3f95255a36cb861&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
            <Contents>
                <Key>example-object-2.jpg</Key>
                <LastModified>2020-12-10T03:37:30.000Z</LastModified>
                <ETag>&quot;51370fc64b79d0d3c7c609635be1c41f&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD_IA</StorageClass>
            </Contents>
</ListBucketResult>
```

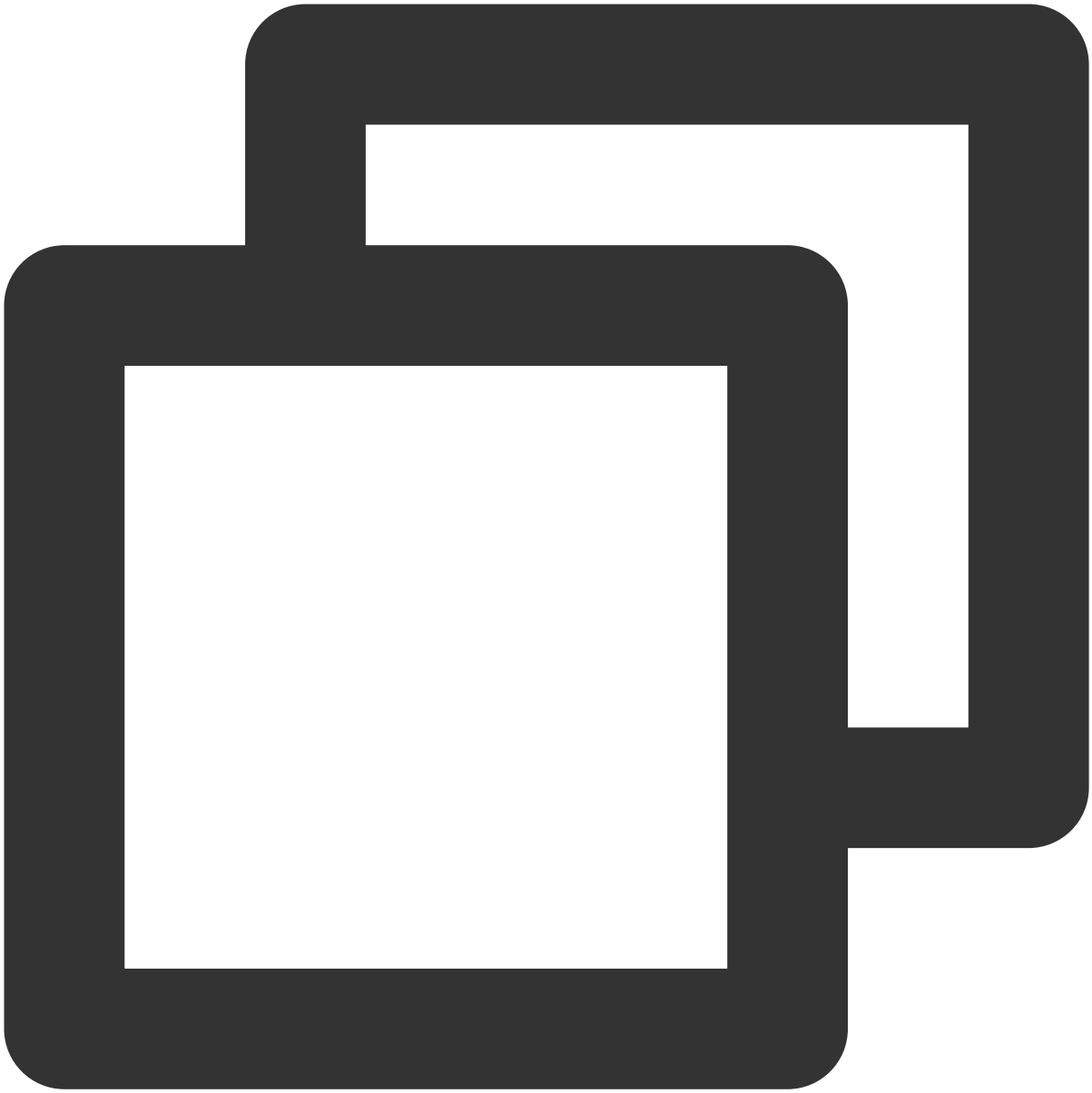
案例四：带 prefix 和 delimiter 参数（列出指定子目录下的对象和子目录）

请求



```
GET /?prefix=example-folder-1%2F&delimiter=%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 03:37:41 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



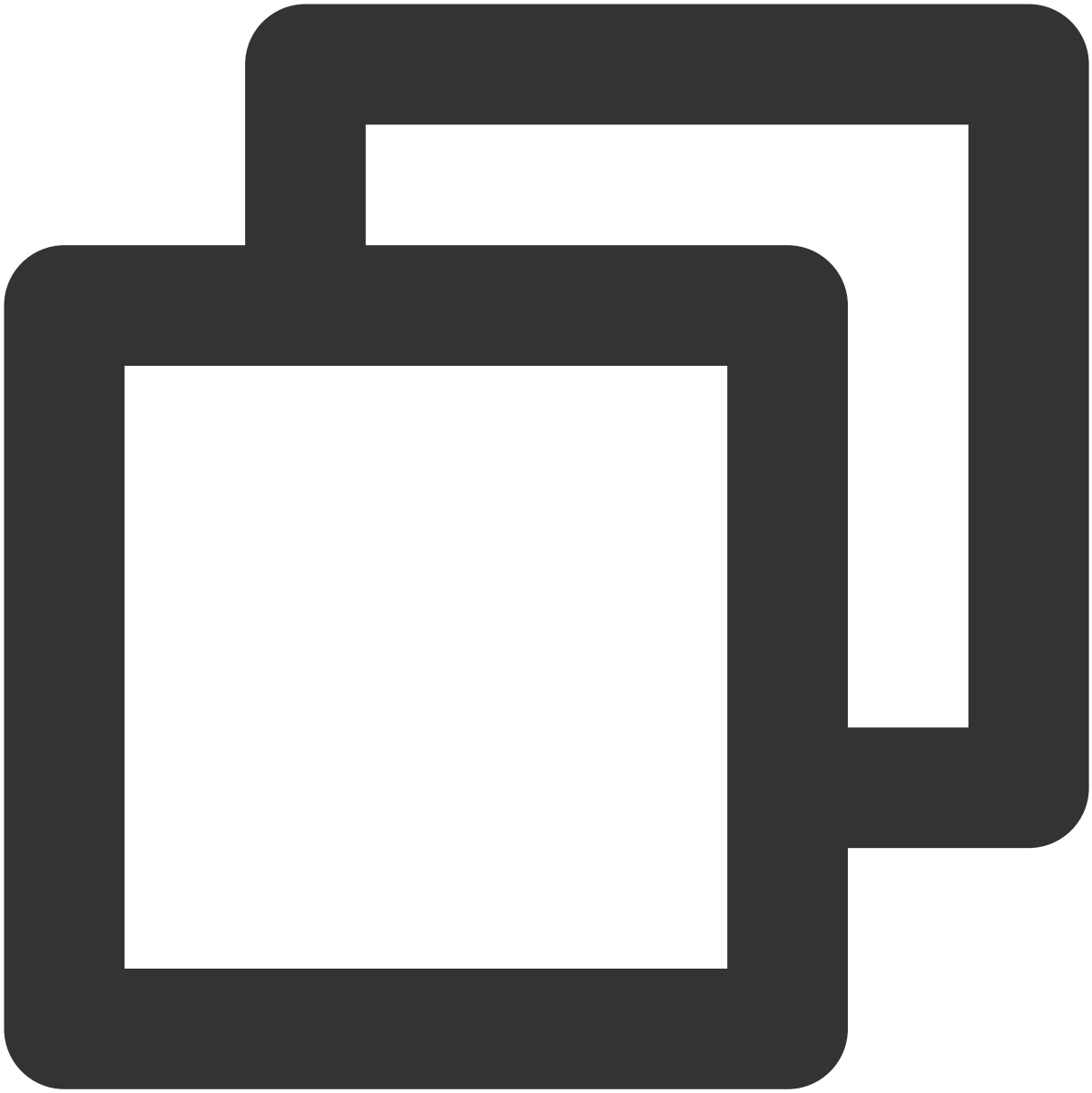
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1142
Connection: close
Date: Thu, 10 Dec 2020 03:37:41 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZkMTk4MDVfZGZjNzJhMDlfMzJiMjRfMTY0****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Name>examplebucket-1250000000</Name>
            <Prefix>example-folder-1/</Prefix>
            <Marker/>
            <MaxKeys>1000</MaxKeys>
            <Delimiter>/</Delimiter>
            <IsTruncated>false</IsTruncated>
            <CommonPrefixes>
                <Prefix>example-folder-1/sub-folder-1/</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>example-folder-1/sub-folder-2/</Prefix>
            </CommonPrefixes>
            <Contents>
                <Key>example-folder-1/example-object-1.jpg</Key>
                <LastModified>2020-12-10T03:37:30.000Z</LastModified>
                <ETag>&quot;f173c1199e3d3b53dd91223cae16fb42&quot;</ETag>
                <Size>37</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
            <Contents>
                <Key>example-folder-1/example-object-2.jpg</Key>
                <LastModified>2020-12-10T03:37:30.000Z</LastModified>
                <ETag>&quot;c9d28698978bb6fef6c1ed1c439a17d3&quot;</ETag>
                <Size>37</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>INTELLIGENT_TIERING</StorageClass>
                <StorageTier>FREQUENT</StorageTier>
            </Contents>
</ListBucketResult>
```

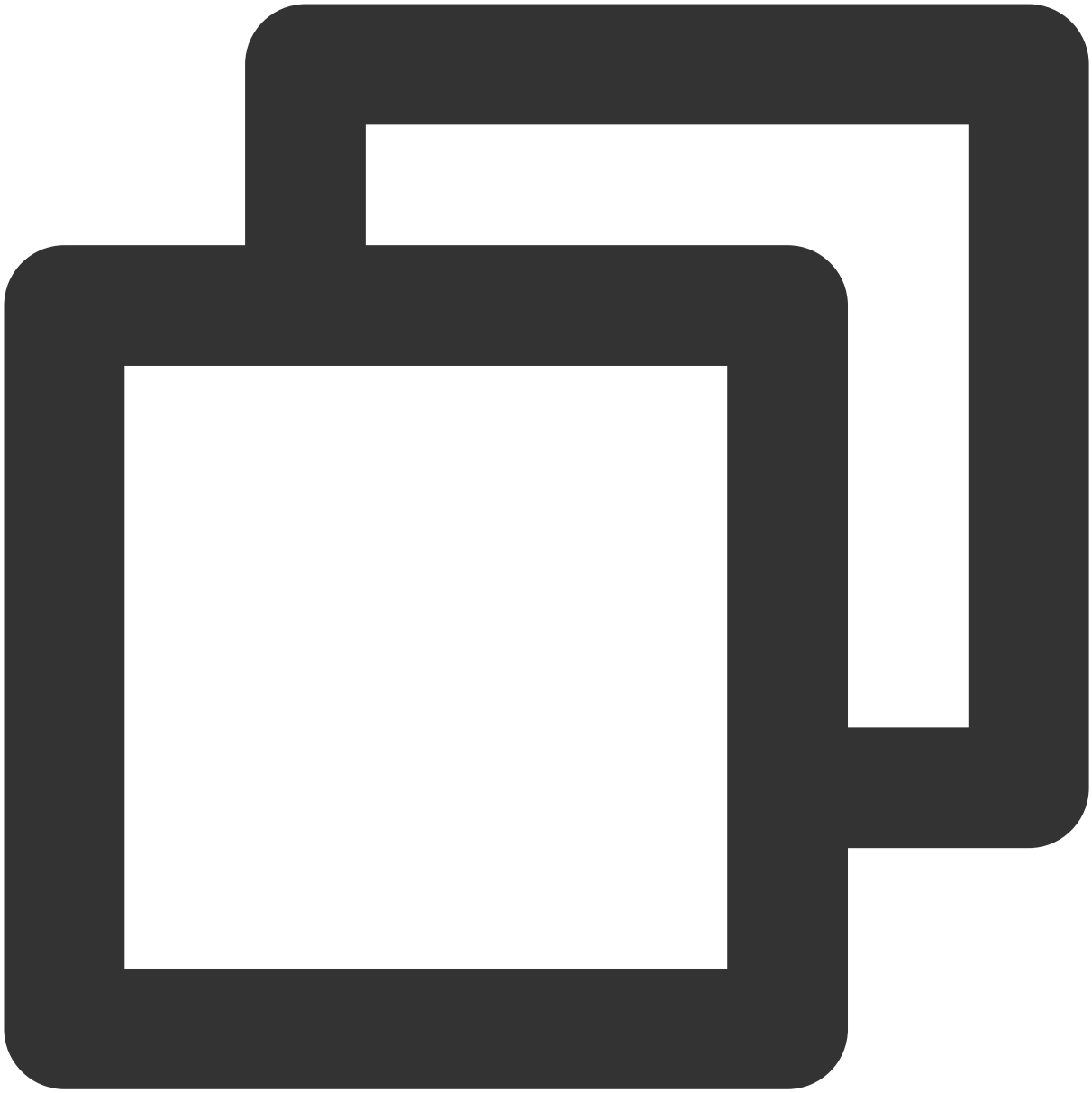
案例五：需分页时获取第一页（案例中限制了 max-keys，如不限制则默认为 1000）

请求



```
GET /?max-keys=3 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 08 Dec 2020 12:05:10 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

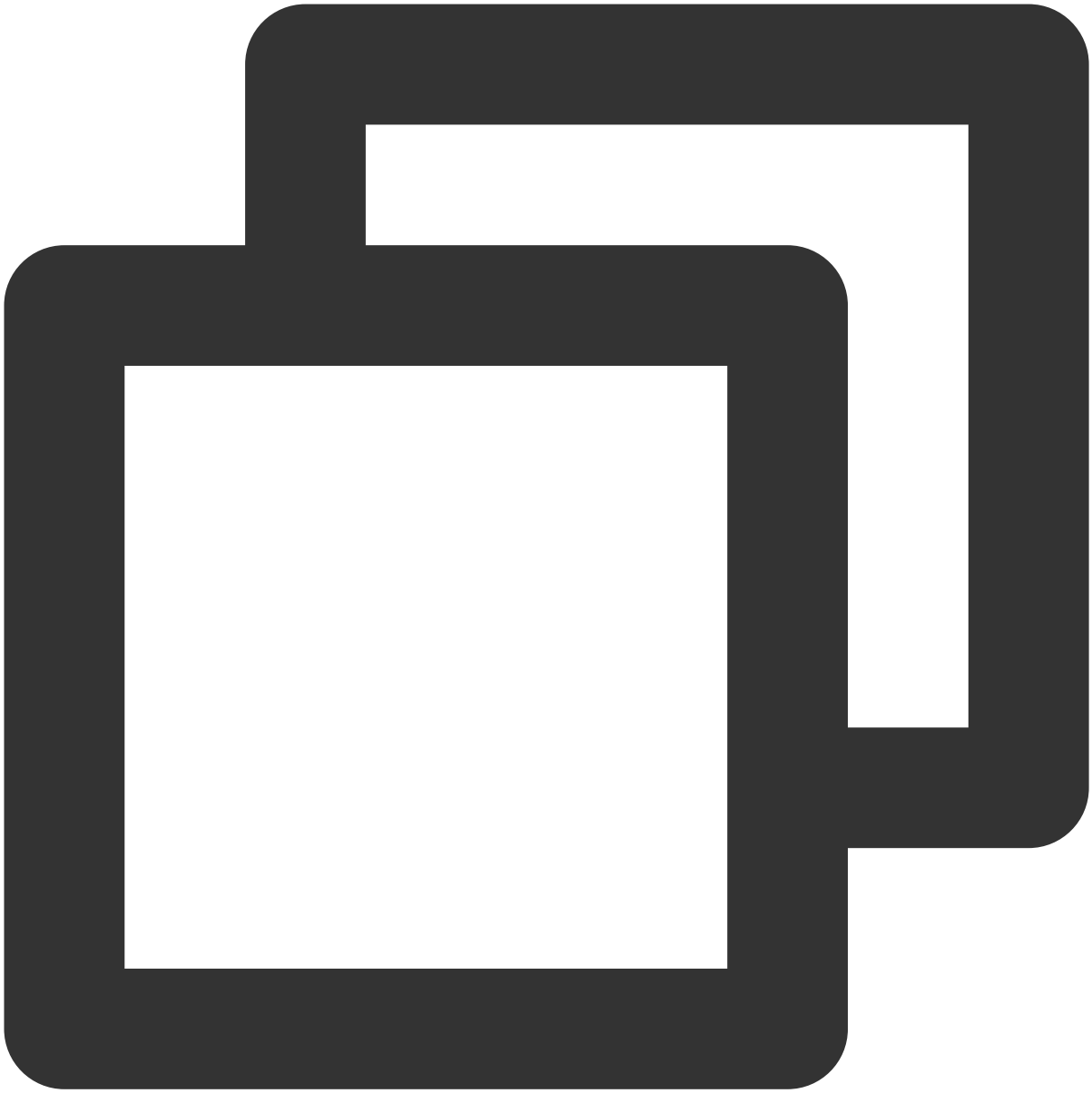


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1195
Connection: close
Date: Tue, 08 Dec 2020 12:05:10 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZjZjZiZjZfZGRjODJhMDlfMWFjZDVfMTlmZTY5****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <Marker/>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>true</IsTruncated>
            <NextMarker>example-object-3.jpg</NextMarker>
            <Contents>
                <Key>example-object-1.jpg</Key>
                <LastModified>2020-12-08T12:05:00.000Z</LastModified>
                <ETag>&quot;0f0cd12c48979d1bf3f95255a36cb861&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
            <Contents>
                <Key>example-object-2.jpg</Key>
                <LastModified>2020-12-08T12:05:00.000Z</LastModified>
                <ETag>&quot;51370fc64b79d0d3c7c609635be1c41f&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
            <Contents>
                <Key>example-object-3.jpg</Key>
                <LastModified>2020-12-08T12:05:00.000Z</LastModified>
                <ETag>&quot;b2f1d893c5fde000ee8ea6eca18ed81f&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
</ListBucketResult>
```

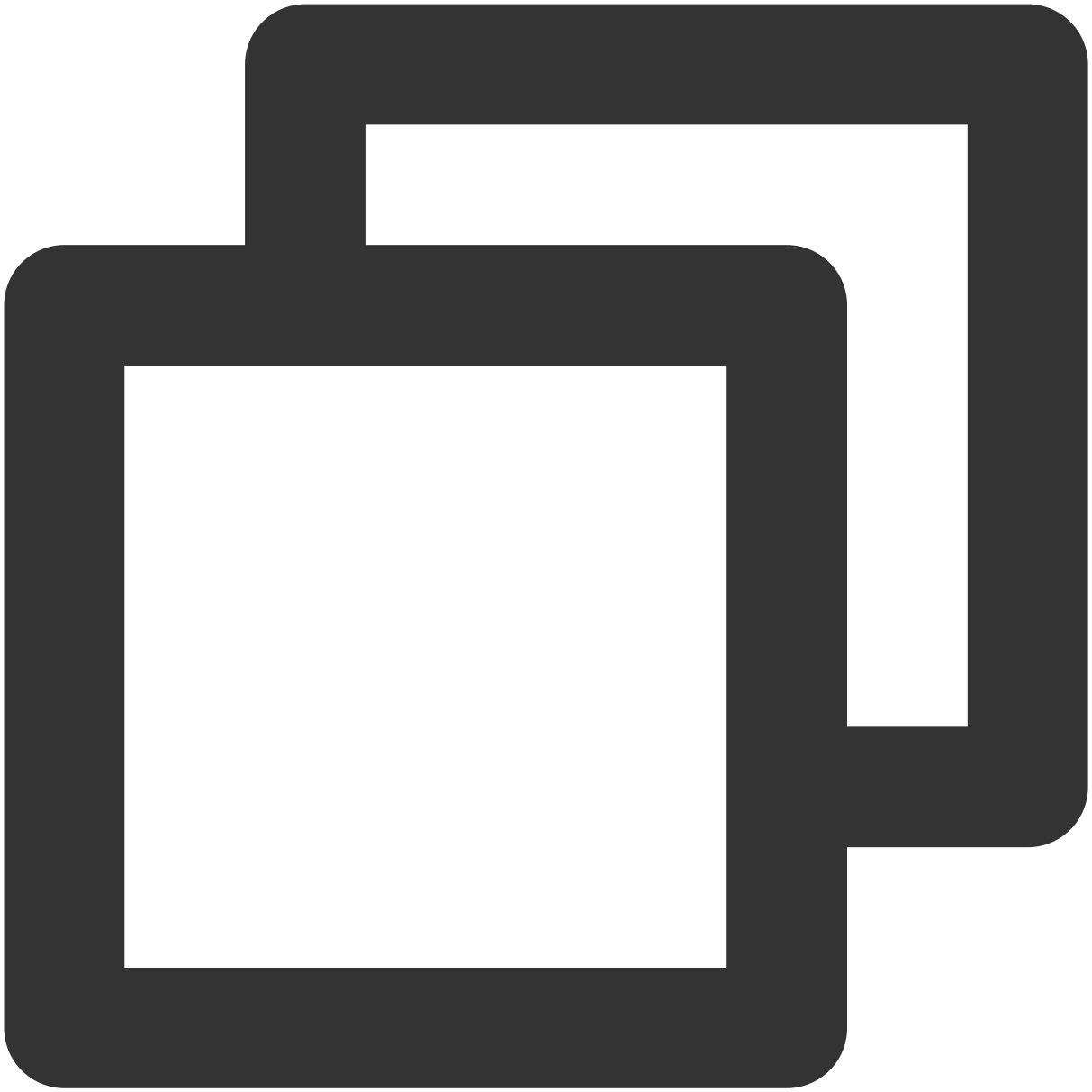
案例六：需分页时获取后续页（续接案例五）

请求



```
GET /?max-keys=3&marker=example-object-3.jpg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 08 Dec 2020 12:05:11 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



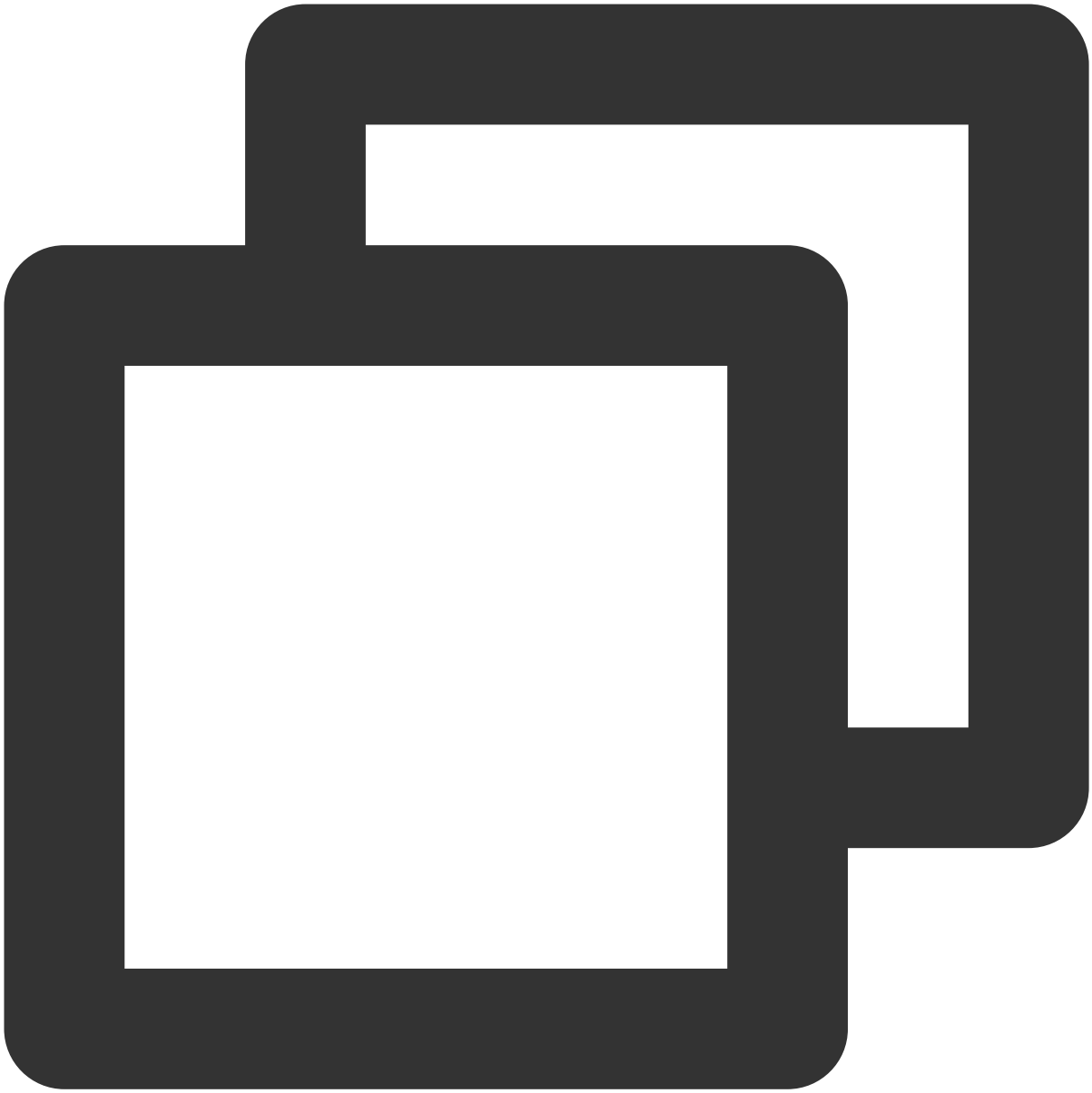
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 859
Connection: close
Date: Tue, 08 Dec 2020 12:05:11 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZjZjZiZjdfMjRhZjJhMDlfMjc2NV8xYmE2****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <Marker>example-object-3.jpg</Marker>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>false</IsTruncated>
            <Contents>
                <Key>example-object-4.jpg</Key>
                <LastModified>2020-12-08T12:05:00.000Z</LastModified>
                <ETag>&amp;quot;e9ec8bcb980d2e4d8526c346eb3b2585&amp;quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
            <Contents>
                <Key>example-object-5.jpg</Key>
                <LastModified>2020-12-08T12:05:00.000Z</LastModified>
                <ETag>&amp;quot;201669a14bdf051d8a9d6f9828d3f4c4&amp;quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
</ListBucketResult>
```

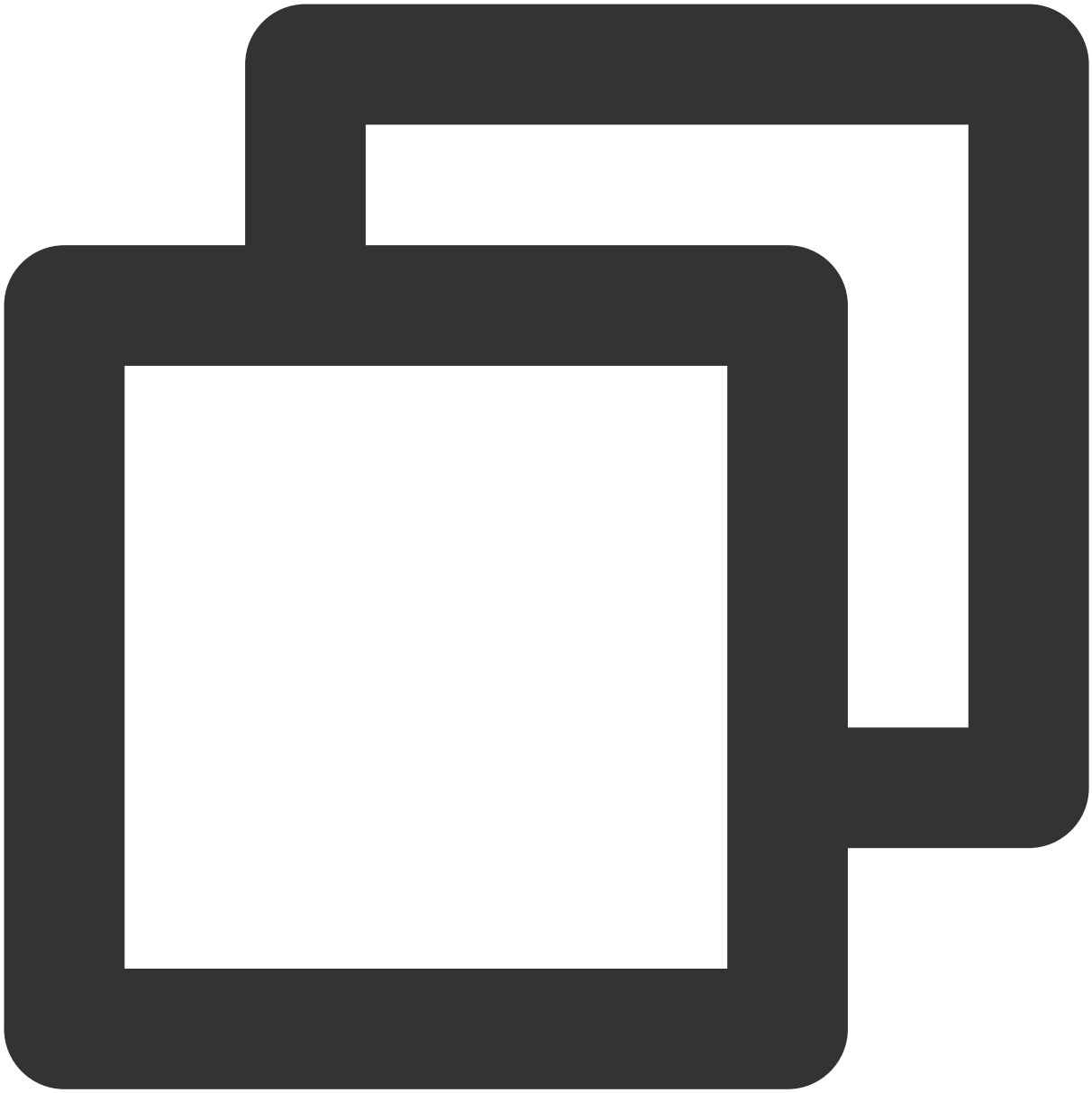
案例七：需分页时获取第一页（带 delimiter 参数，CommonPrefixes 与 Contents 合计条目数不超过 max-keys 指定的值）

请求



```
GET /?delimiter=%2F&max-keys=3 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 09 Dec 2020 03:22:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

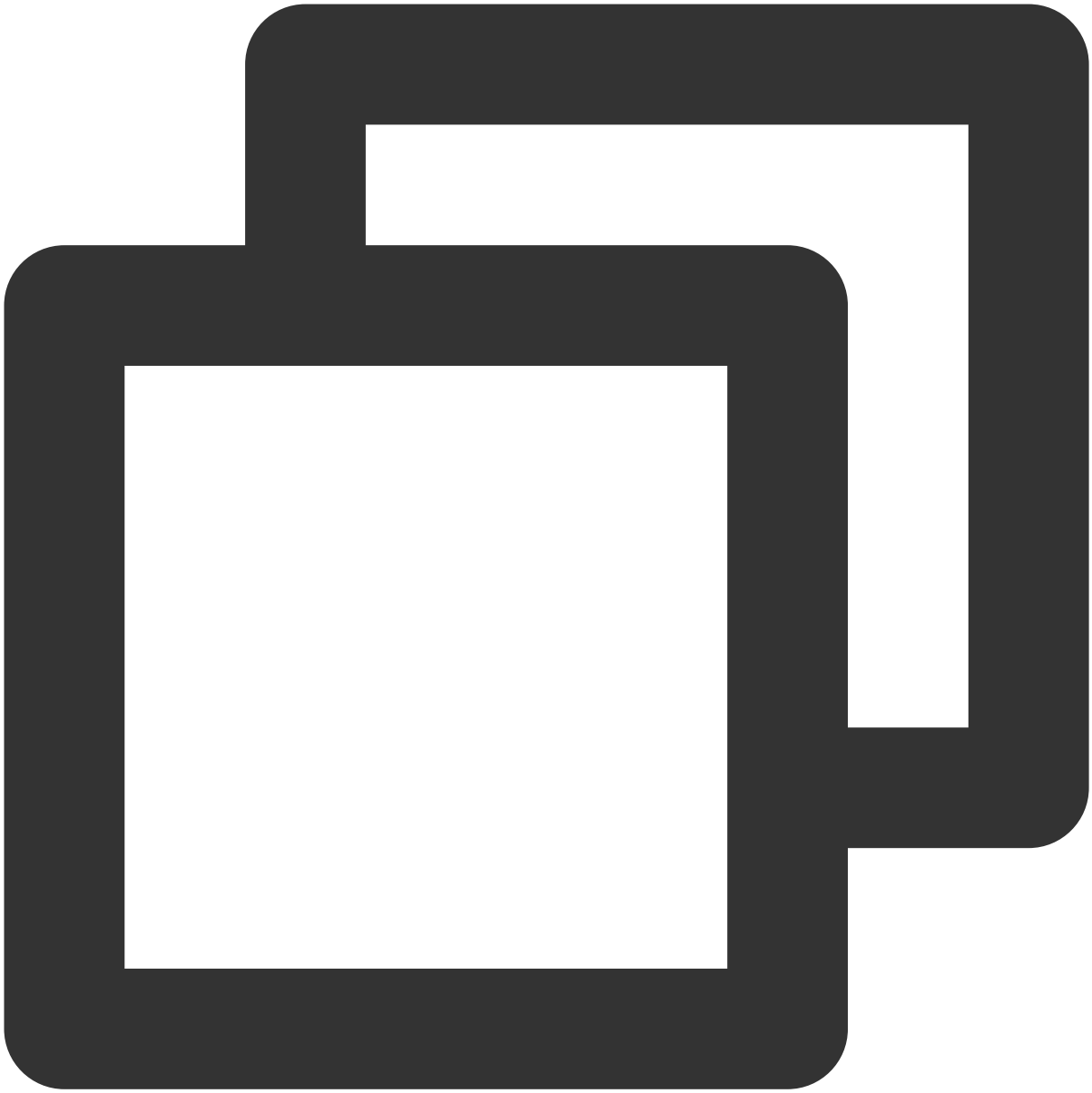


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 486
Connection: close
Date: Wed, 09 Dec 2020 03:22:38 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZkMDQyZmVfYTJjMjJhMDlfYmQwOF8xYjkw****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <Marker/>
            <MaxKeys>3</MaxKeys>
            <Delimiter>/</Delimiter>
            <IsTruncated>true</IsTruncated>
            <NextMarker>example-folder-3/</NextMarker>
            <CommonPrefixes>
                <Prefix>example-folder-1/</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>example-folder-2/</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>example-folder-3/</Prefix>
            </CommonPrefixes>
</ListBucketResult>
```

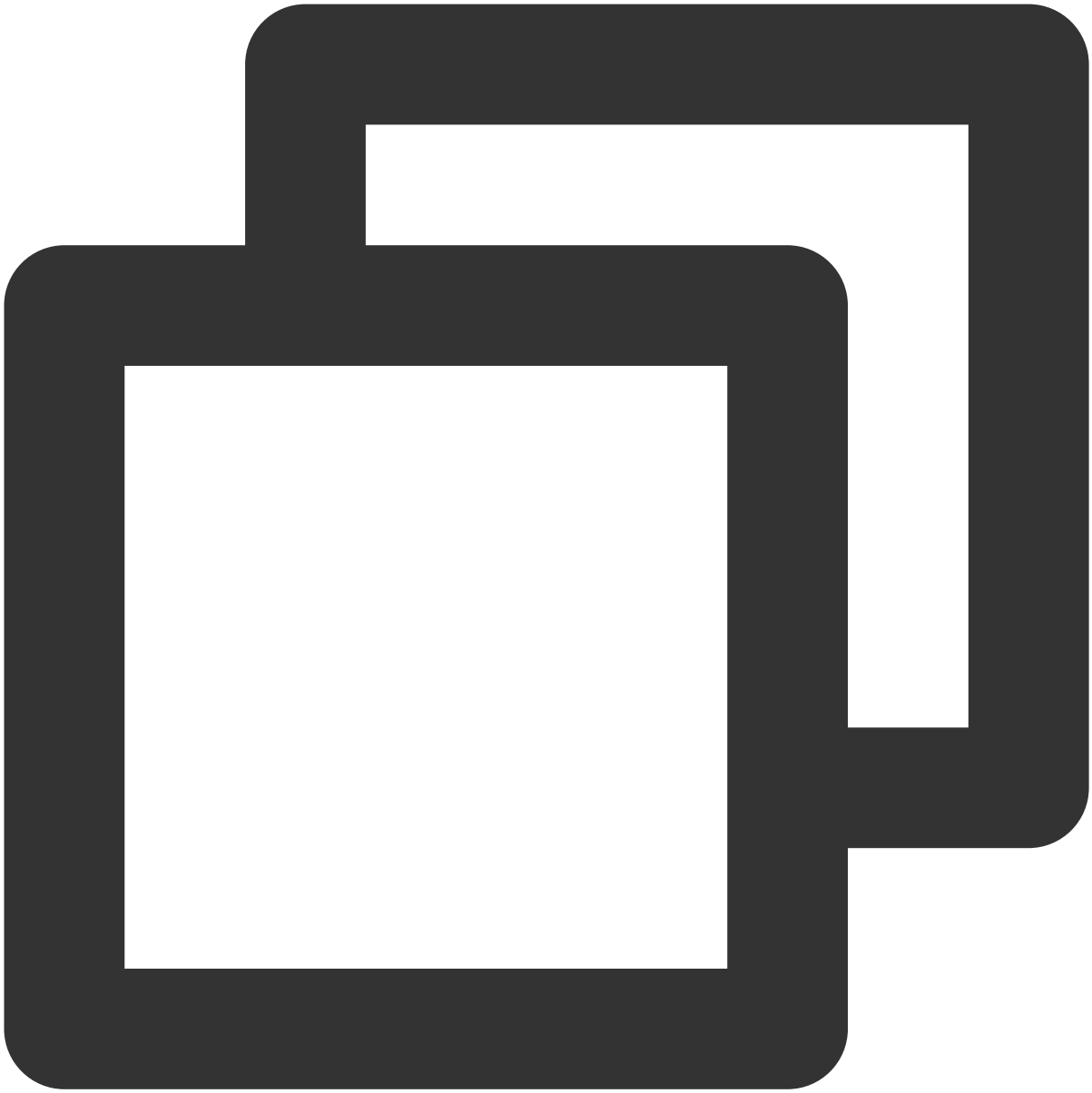
案例八：需分页时获取后续页（带 delimiter 参数，续接案例七）

请求



```
GET /?delimiter=%2F&max-keys=3&marker=example-folder-3%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 09 Dec 2020 03:22:39 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 956
Connection: close
Date: Wed, 09 Dec 2020 03:22:39 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWZkMDQyZmZfZmNhODBiMDlfZmM4N19jYmQz****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <Marker>example-folder-3/</Marker>
            <MaxKeys>3</MaxKeys>
            <Delimiter>/</Delimiter>
            <IsTruncated>false</IsTruncated>
            <CommonPrefixes>
                <Prefix>example-folder-4/</Prefix>
            </CommonPrefixes>
            <Contents>
                <Key>example-object-1.jpg</Key>
                <LastModified>2020-12-09T03:22:28.000Z</LastModified>
                <ETag>&quot;0f0cd12c48979d1bf3f95255a36cb861&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
            <Contents>
                <Key>example-object-2.jpg</Key>
                <LastModified>2020-12-09T03:22:28.000Z</LastModified>
                <ETag>&quot;51370fc64b79d0d3c7c609635be1c41f&quot;</ETag>
                <Size>20</Size>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
            </Contents>
</ListBucketResult>
```

HEAD Bucket

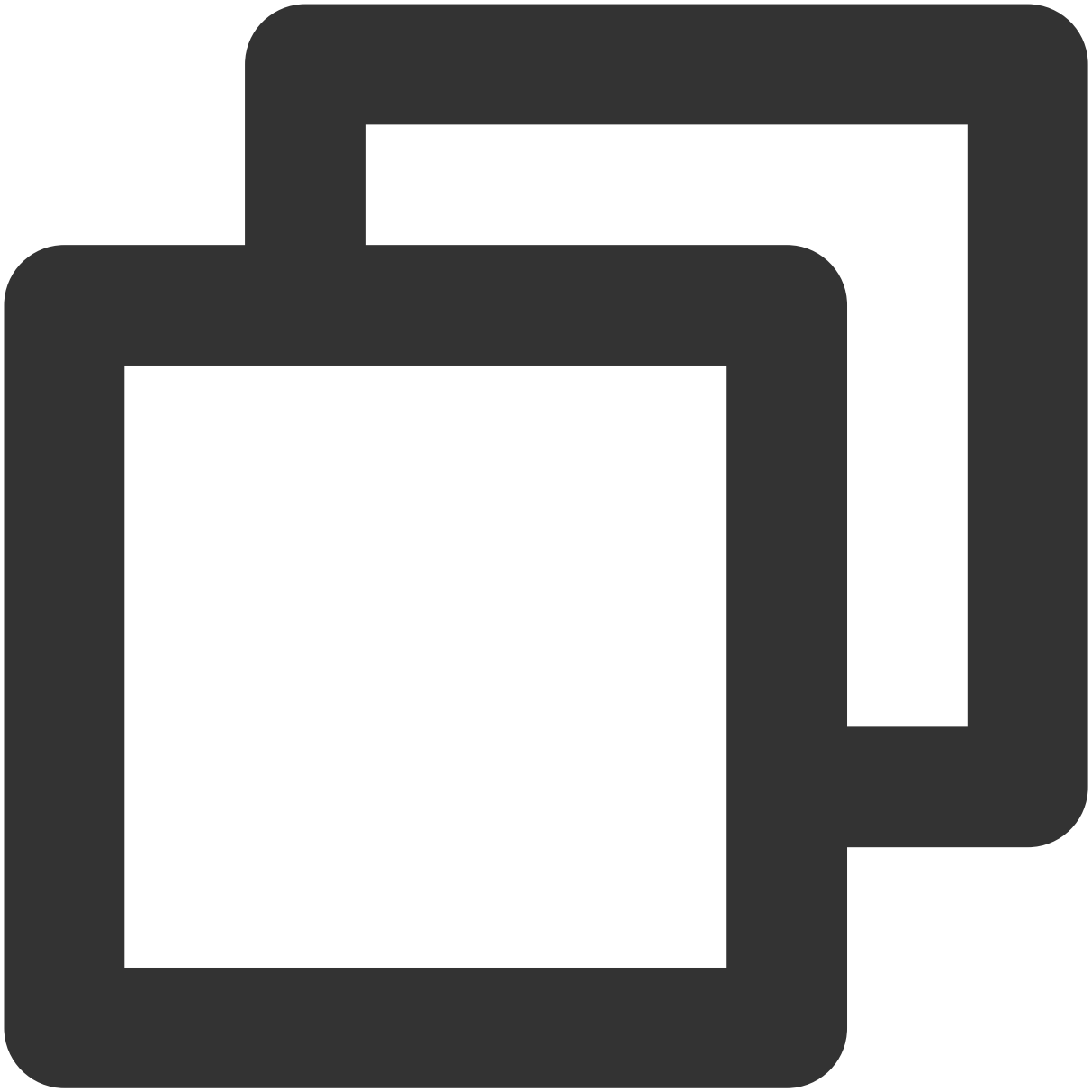
最近更新时间：2023-12-20 18:07:00

功能描述

HEAD Bucket 请求可以确认该存储桶是否存在，是否有权限访问。有以下几种情况：
存储桶存在且有读取权限，返回 HTTP 状态码为200。
无存储桶读取权限，返回 HTTP 状态码为403。
存储桶不存在，返回 HTTP 状态码为404。

请求

请求示例



```
HEAD / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

名称	描述	类型
x-cos-bucket-az-type	存储桶 AZ 类型，当存储桶为多 AZ 存储桶时返回此头部，值固定为 MAZ。	Enum
x-cos-bucket-region	存储桶所在地域。枚举值请参见 地域和访问域名 文档，例如 ap-beijing，ap-hongkong，eu-frankfurt 等	Enum

响应体

此接口响应体为空。

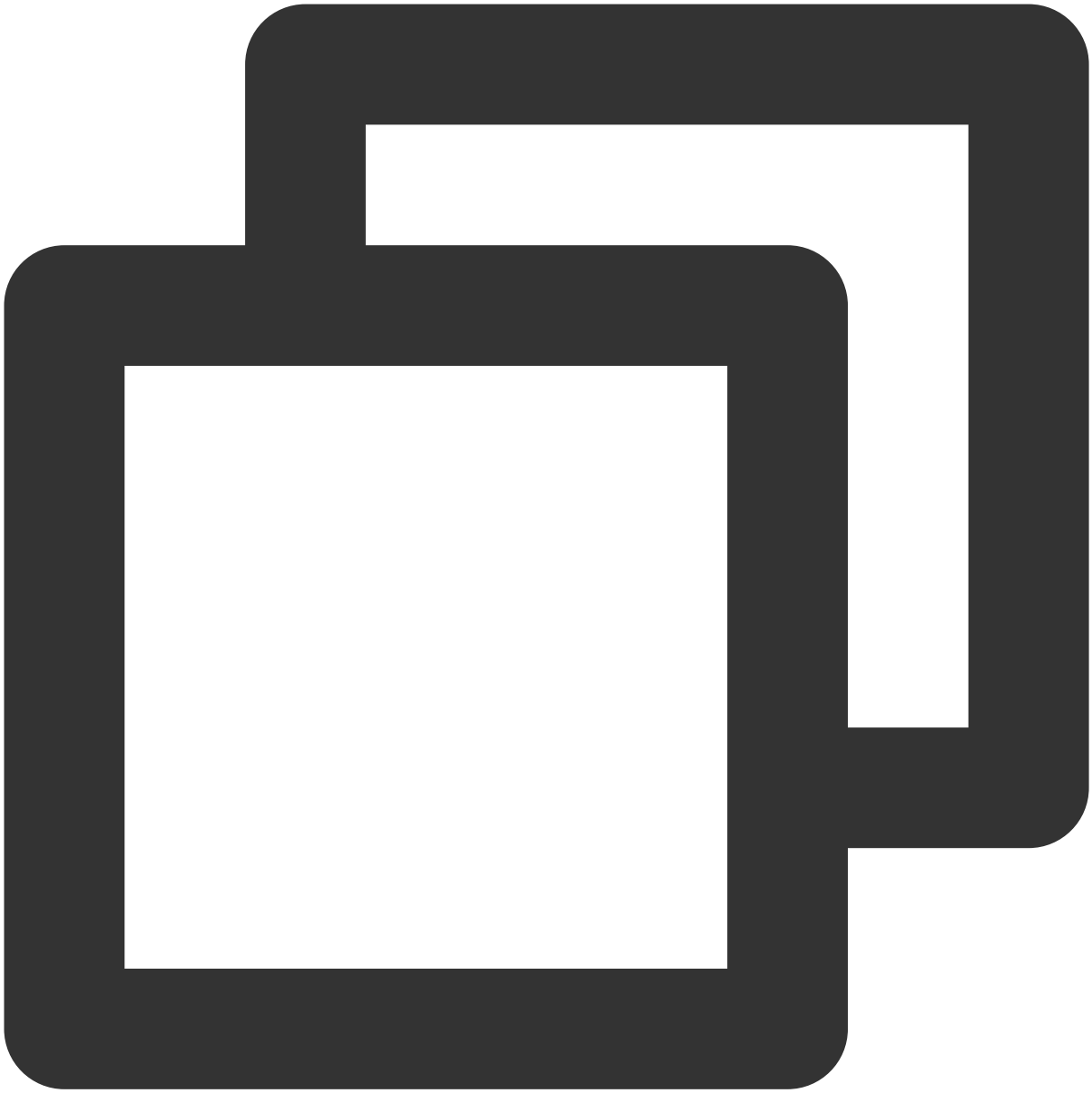
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

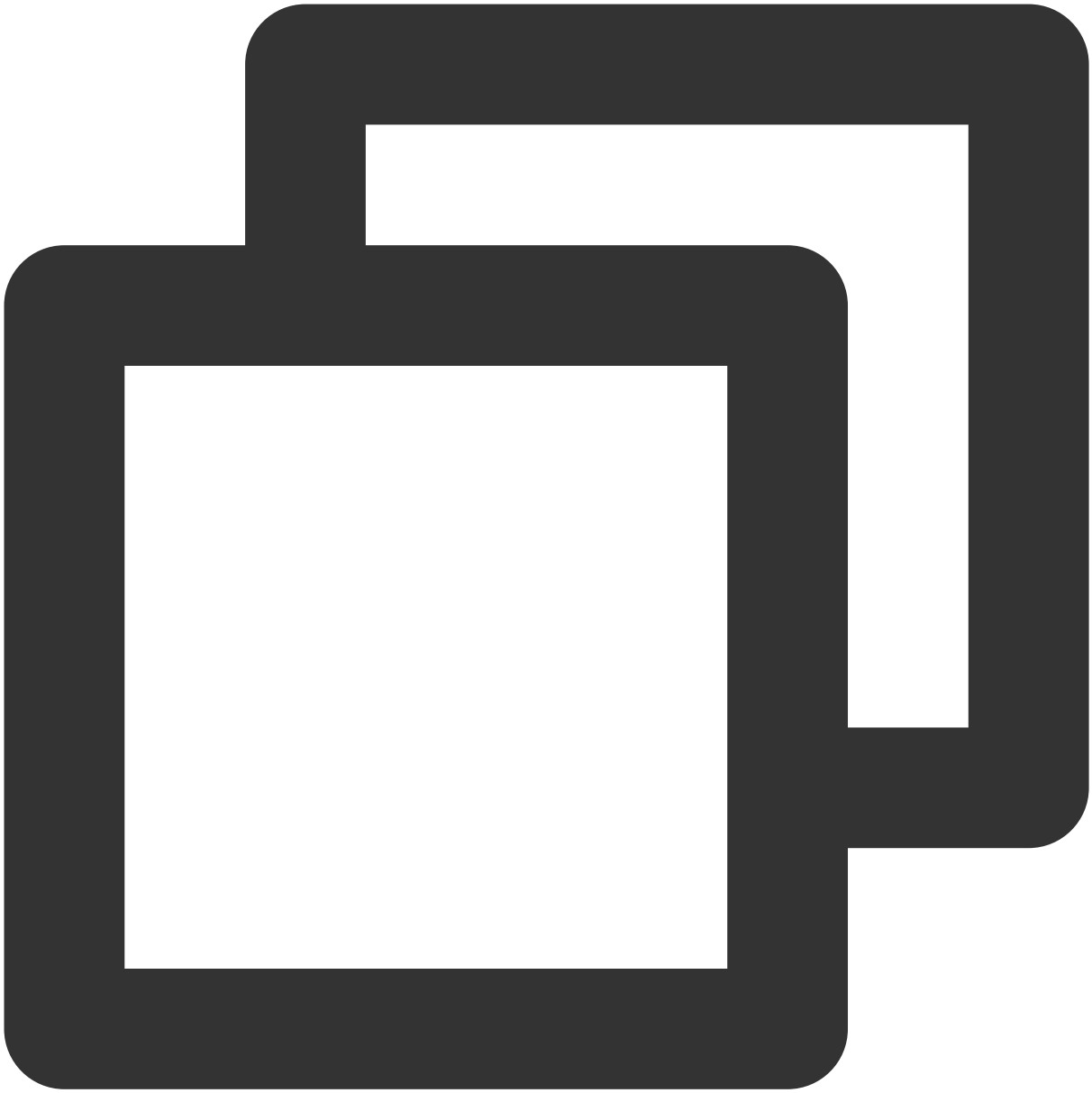
案例一：单 AZ 存储桶

请求



```
HEAD / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 28 May 2019 03:16:12 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

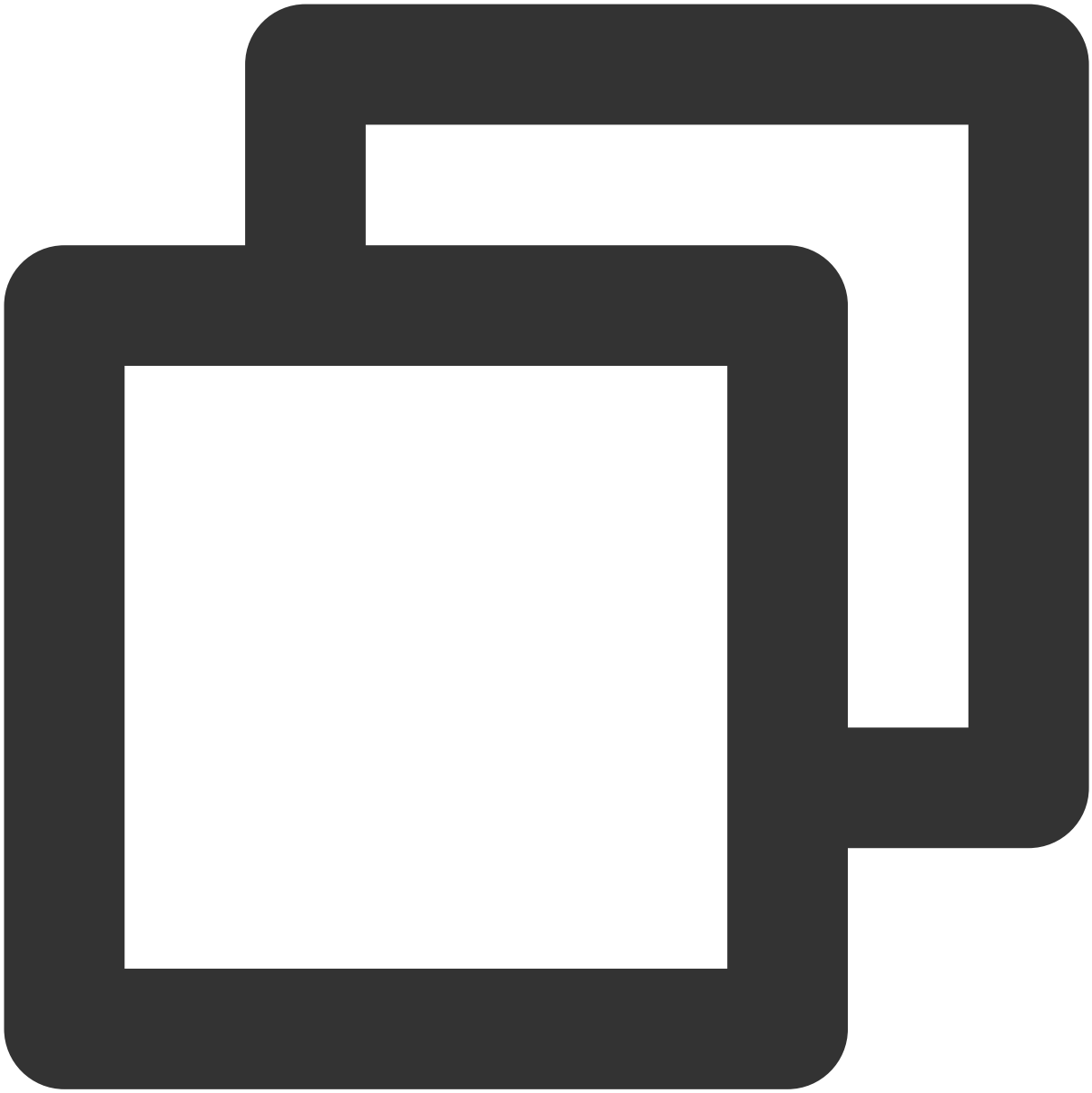
响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: close
Date: Tue, 28 May 2019 03:16:12 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWNlY2E3ZmNfZjhjMDBiMDlfMTBjOWRfZDcz****
```

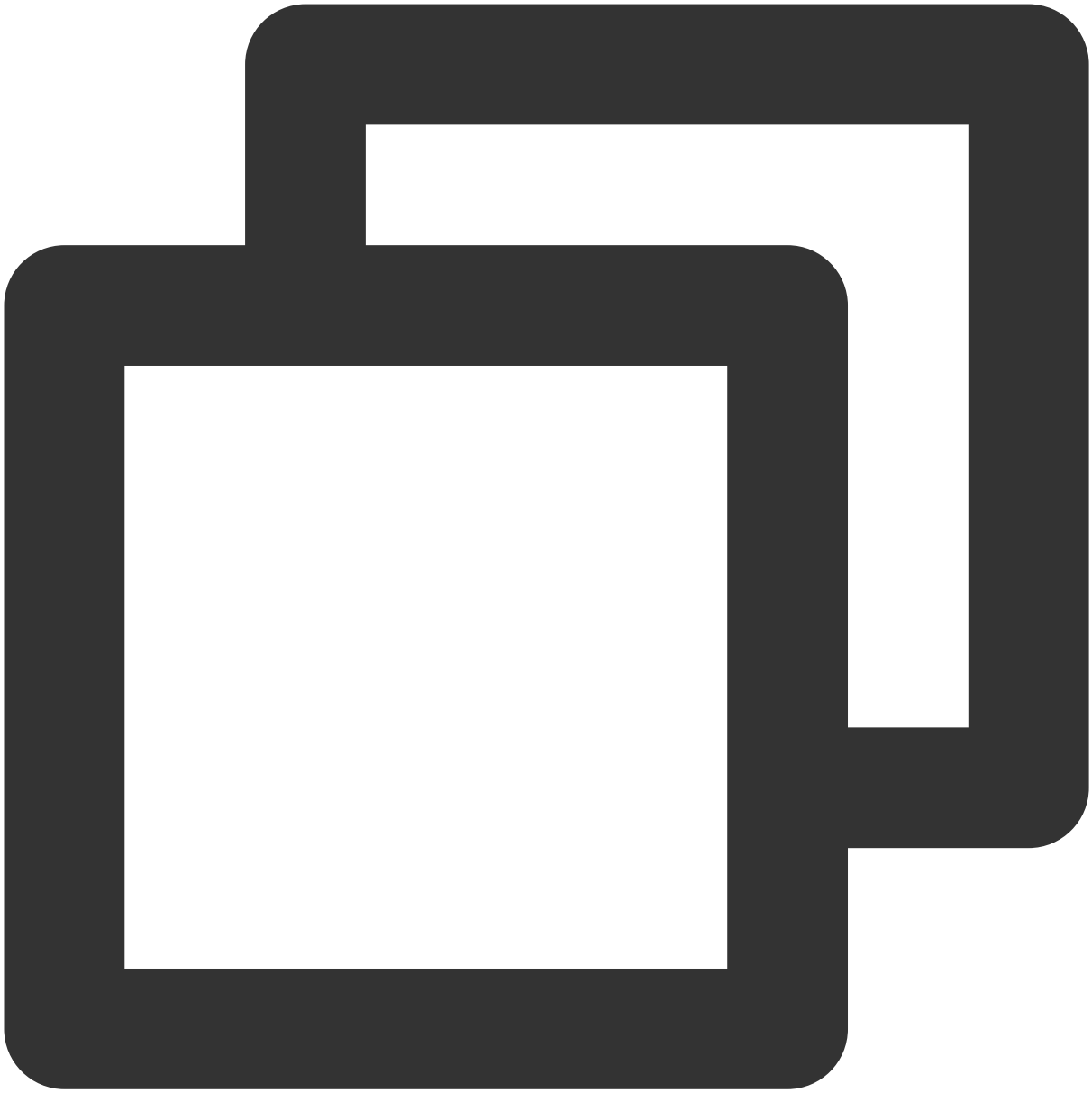
案例二：多 AZ 存储桶

请求



```
HEAD / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jun 2020 06:43:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: close
Date: Thu, 04 Jun 2020 06:43:03 GMT
Server: tencent-cos
x-cos-bucket-az-type: MAZ
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWVkODk3ZjdfZjI4NWQ2NF9hODhkXzRhNmQ0****
```

DELETE Bucket

最近更新时间：2023-12-20 16:45:49

功能描述

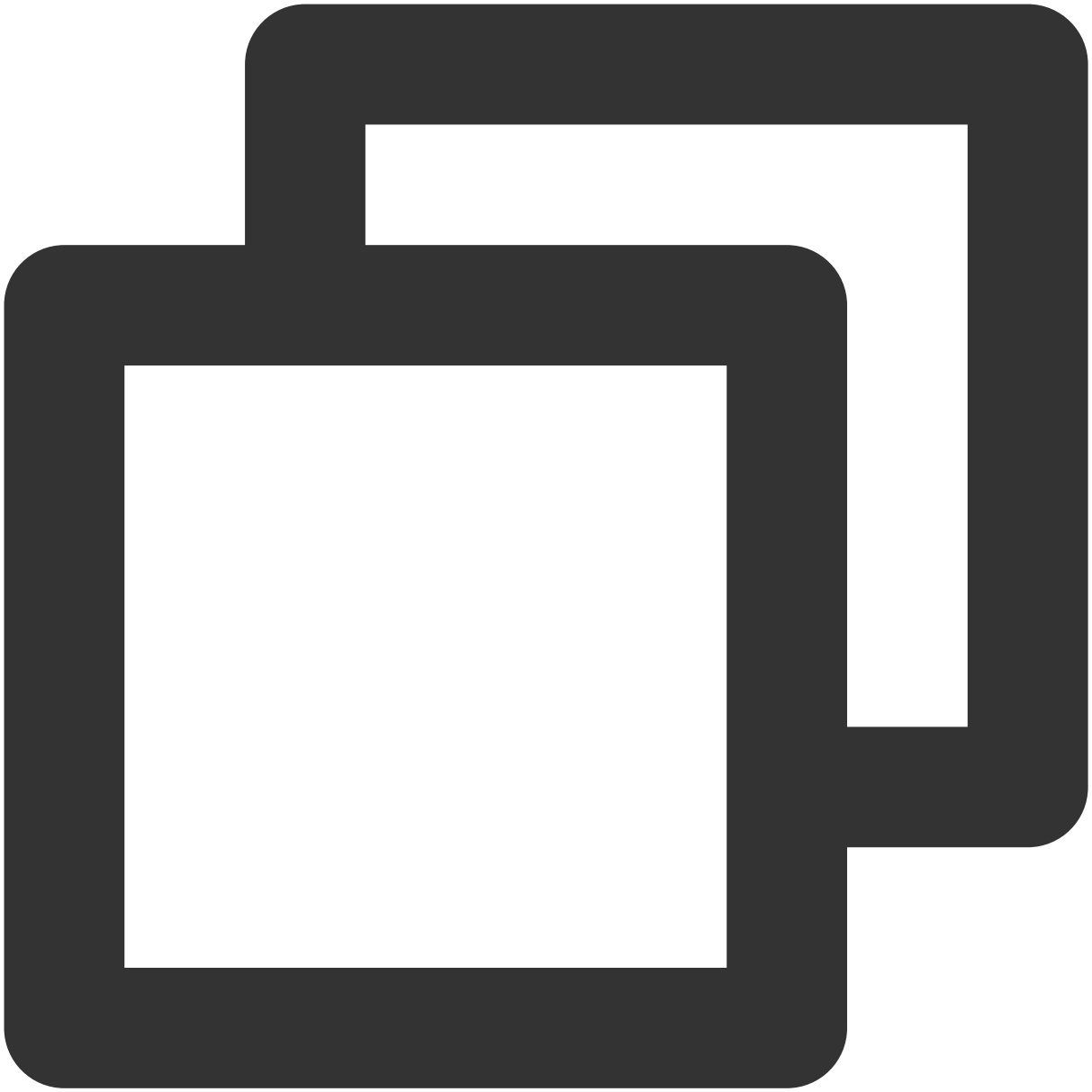
DELETE Bucket 请求用于删除指定的存储桶。该 API 的请求者需要对存储桶有写入权限。

注意

删除存储桶前，请确保存储桶内的数据和未完成上传的分块数据已全部清空，否则会无法删除存储桶。

请求

请求示例



```
DELETE / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

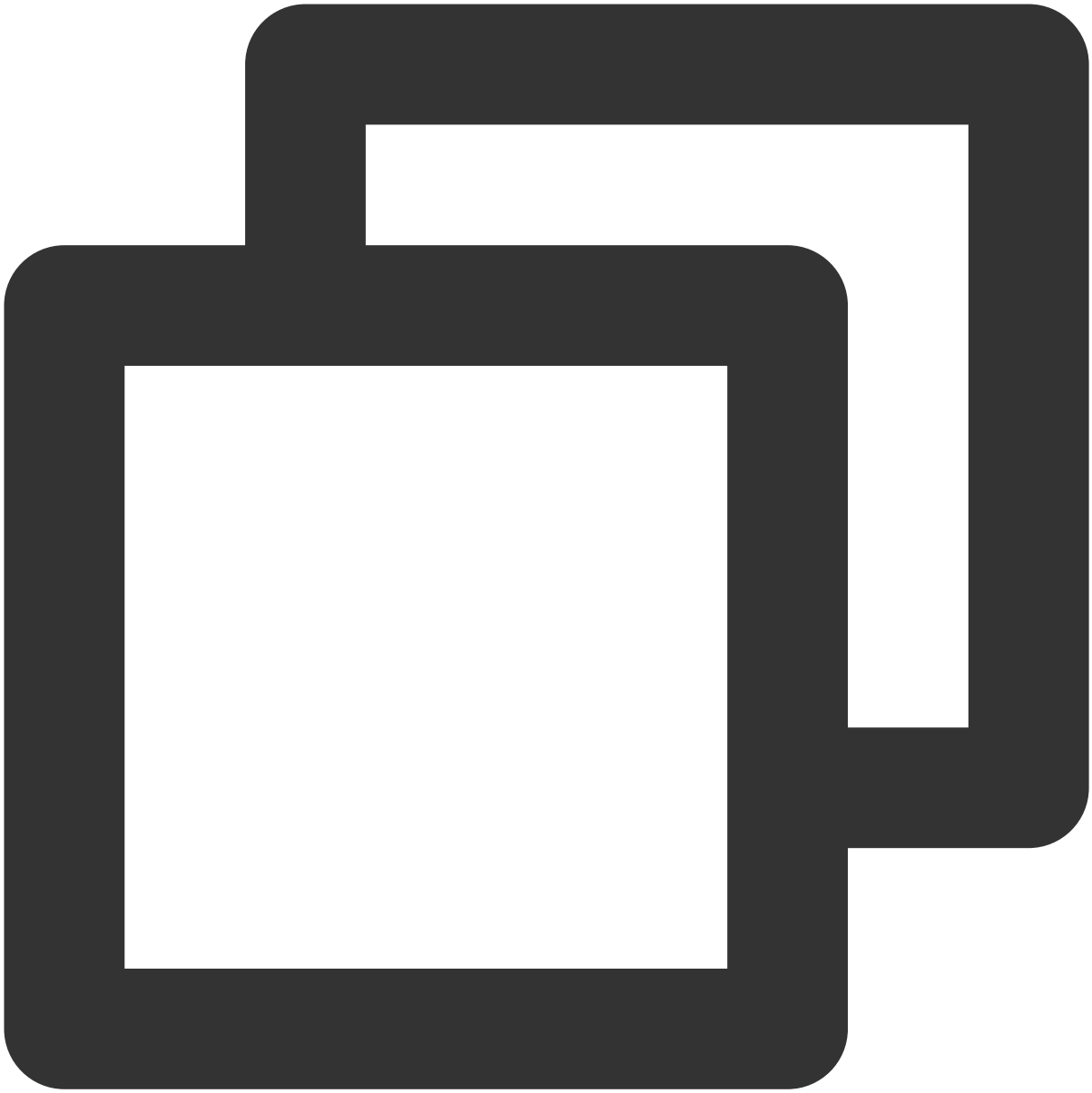
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

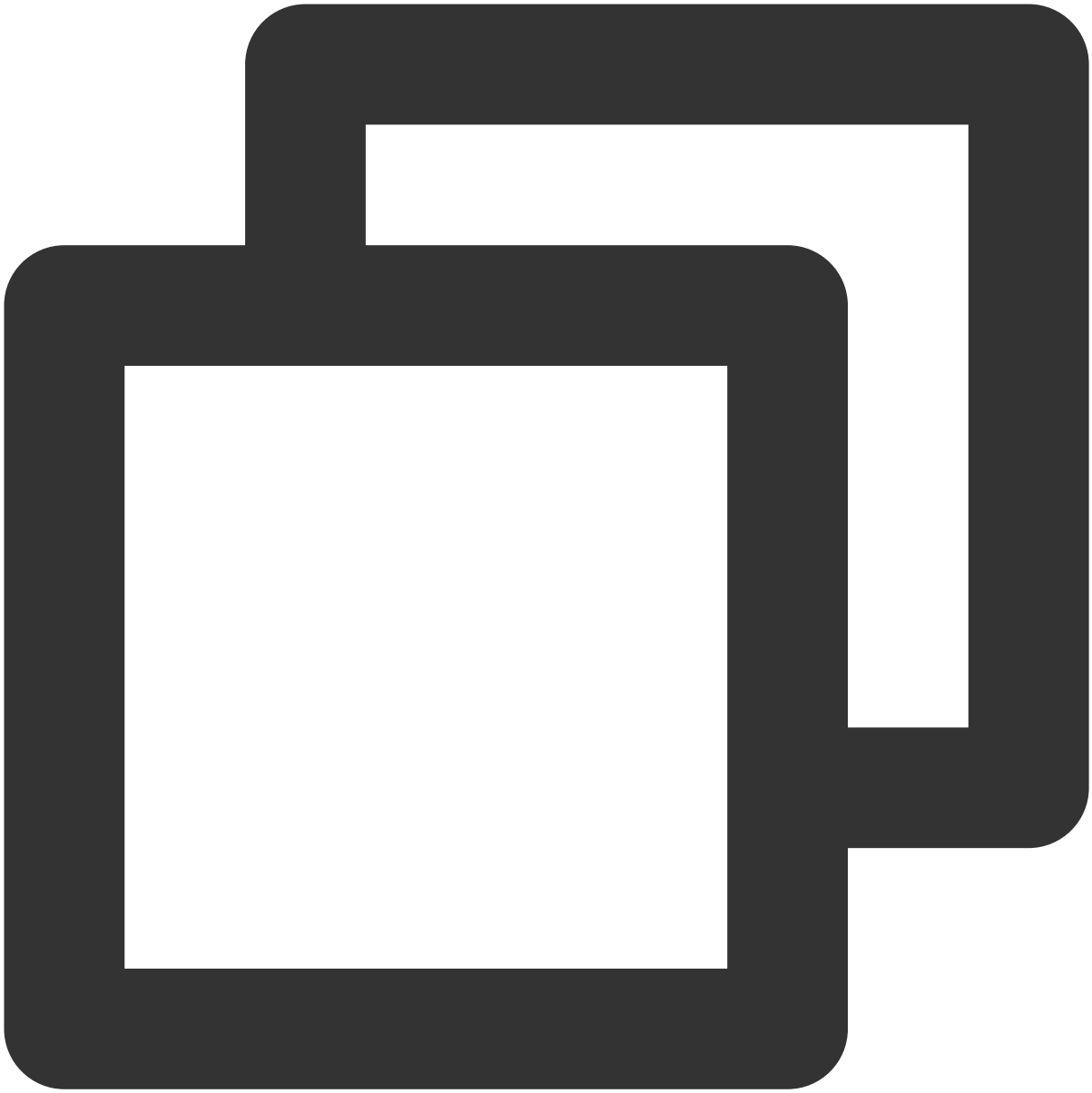
实际案例

请求



```
DELETE / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 28 May 2019 03:19:13 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
Date: Tue, 28 May 2019 03:19:14 GMT
Server: tencent-cos
x-cos-request-id: NWNlY2E4YjFfNjljMDBiMDlfMmNiZTlfZGE0****
```

GET Bucket Object Versions

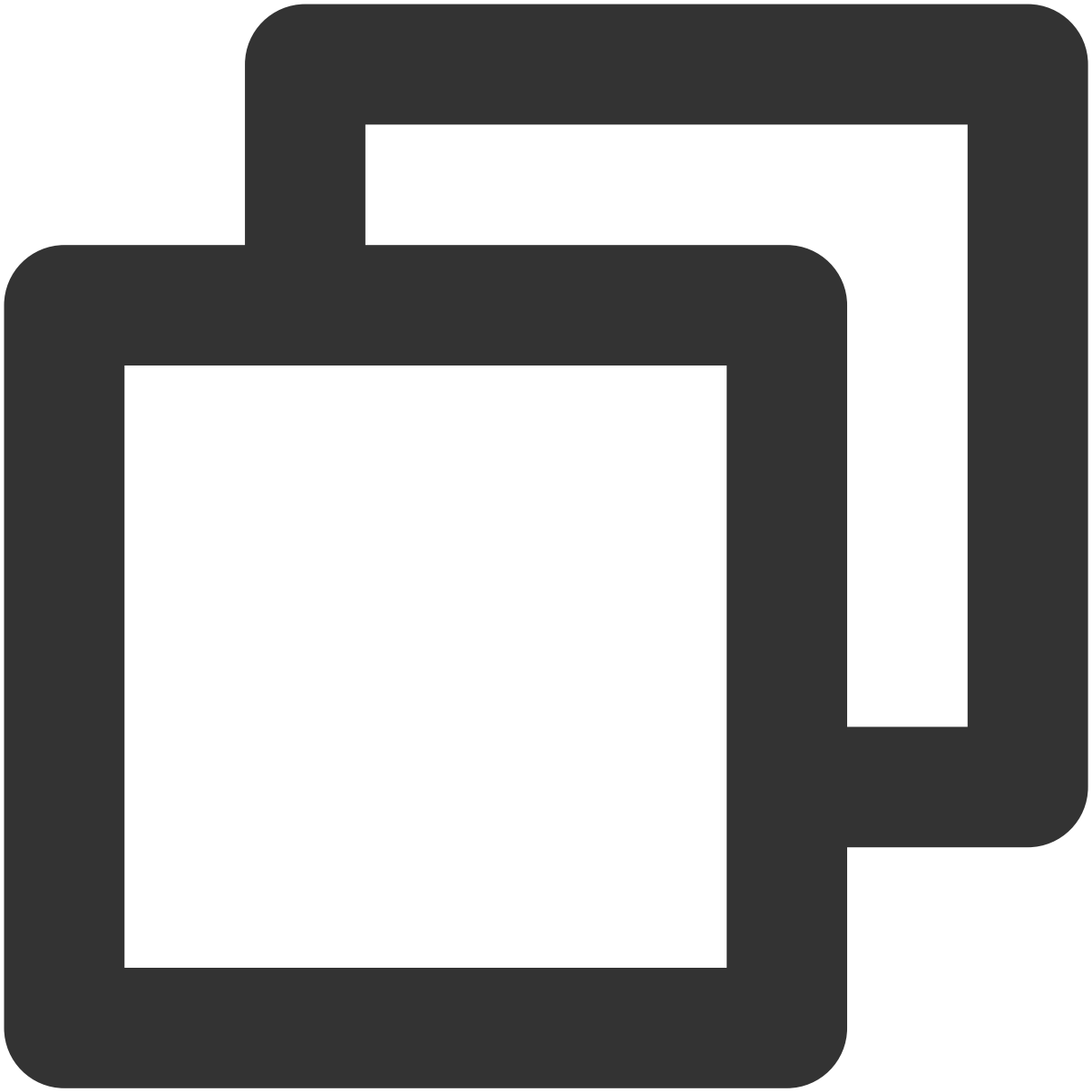
最近更新时间：2023-12-20 18:06:00

功能描述

GET Bucket Object versions 接口用于拉取存储桶内的所有对象及其历史版本信息，您可以通过指定参数筛选出存储桶内部分对象及其历史版本信息。该 API 的请求者需要对存储桶有读取权限。

请求

请求示例



```
GET /?versions HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

名称	描述	类型	是否必选
prefix	对象键匹配前缀，限定响应中只包含指定前缀的对象键	string	否
delimiter	一个字符的分隔符，用于对对象键进行分组。所有对象键中从 prefix 或从头（如未指定 prefix）到首个 delimiter 之间相同的部分将作为 CommonPrefixes 下的一个 Prefix 节点。被分组的对象键不再出现在后续对象列表中，具体场景和用法可参考下面的实际案例	string	否
encoding-type	规定返回值的编码方式，可选值：url，代表返回的对象键为 URL 编码（百分号编码）后的值，例如“腾讯云”将被编码为 `%E8%85%BE%E8%AE%AF%E4%BA%91`	string	否
key-marker	起始对象键标记，从该标记之后（不含）按照 UTF-8 字典序返回对象版本条目	string	否
version-id-marker	起始版本 ID 标记，从该标记之后（不含）返回对象版本条目，如果上一次 List 结果的 NextVersionIdMarker 为空，则该参数也指定为空	string	否
max-keys	单次返回最大的条目数量，默认值为1000，最大为1000 **注意：**该参数会限制每一次 List 操作返回的最大条目数，COS 在每次 List 操作中将返回不超过 max-keys 所设定数值的条目（CommonPrefixes、Version 和 DeleteMarker 合计），如果单次响应中未列出所有对象，COS 会返回 NextKeyMarker 和 NextVersionIdMarker 节点，其值分别为您下次 List 请求的 key-marker 和 version-id-marker 参数，以便您列出后续版本	integer	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

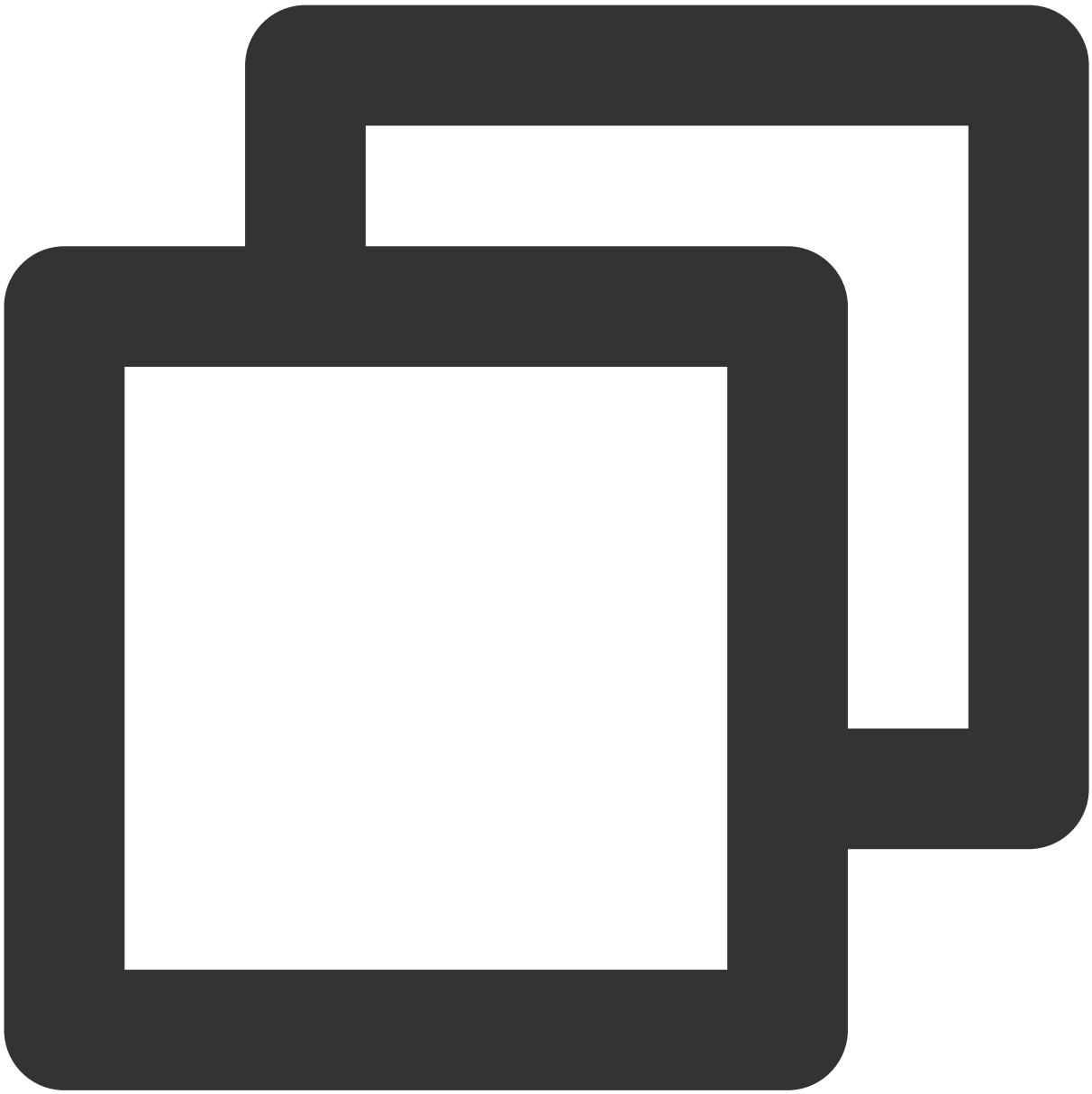
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含存储桶中的对象版本信息。不同场景下的响应体请参见下方的实际案例。



```
<ListVersionsResult>
        <EncodingType>string</EncodingType>
        <Name>string</Name>
        <Prefix>string</Prefix>
        <KeyMarker>string</KeyMarker>
        <VersionIdMarker>string</VersionIdMarker>
        <MaxKeys>integer</MaxKeys>
        <IsTruncated>boolean</IsTruncated>
```

```
        <NextKeyMarker>string</NextKeyMarker>
        <NextVersionIdMarker>string</NextVersionIdMarker>
        <Delimiter>string</Delimiter>
        <CommonPrefixes>
            <Prefix>string</Prefix>
        </CommonPrefixes>
        <CommonPrefixes>
            <Prefix>string</Prefix>
        </CommonPrefixes>
        <Version>
            <Key>string</Key>
            <VersionId>string</VersionId>
            <IsLatest>boolean</IsLatest>
            <LastModified>date</LastModified>
            <ETag>string</ETag>
            <Size>integer</Size>
            <StorageClass>Enum</StorageClass>
            <StorageTier>Enum</StorageTier>
            <Owner>
                <ID>string</ID>
                <DisplayName>string</DisplayName>
            </Owner>
        </Version>
        <DeleteMarker>
            <Key>string</Key>
            <VersionId>string</VersionId>
            <IsLatest>boolean</IsLatest>
            <LastModified>date</LastModified>
            <Owner>
                <ID>string</ID>
                <DisplayName>string</DisplayName>
            </Owner>
        </DeleteMarker>
        <Version>
            <Key>string</Key>
            <VersionId>string</VersionId>
            <IsLatest>boolean</IsLatest>
            <LastModified>date</LastModified>
            <ETag>string</ETag>
            <Size>integer</Size>
            <StorageClass>Enum</StorageClass>
            <StorageTier>Enum</StorageTier>
            <Owner>
                <ID>string</ID>
                <DisplayName>string</DisplayName>
            </Owner>
        </Version>
```

```
        <DeleteMarker>
            <Key>string</Key>
            <VersionId>string</VersionId>
            <IsLatest>boolean</IsLatest>
            <LastModified>date</LastModified>
            <Owner>
                <ID>string</ID>
                <DisplayName>string</DisplayName>
            </Owner>
        </DeleteMarker>
</ListVersionsResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
ListVersionsResult	无	保存 GET Bucket Object versions 结果的所有信息	Container

Container 节点 ListVersionsResult 的内容：

节点名称（关键字）	父节点	描述	类型
EncodingType	ListVersionsResult	编码格式，对应请求中的 encoding-type 参数，且仅当请求中指定了 encoding-type 参数才会返回该节点	string
Name	ListVersionsResult	存储桶的名称，格式为 `<BucketName-APPID>` ，例如 `examplebucket-1250000000`	string
Prefix	ListVersionsResult	对象键匹配前缀，对应请求中的 prefix 参数	string
KeyMarker	ListVersionsResult	起始对象键标记，从该标记之后（不含）按照 UTF-8 字典序返回对象版本条目，对应请求中的 key-marker 参数	string
VersionIdMarker	ListVersionsResult	起始版本 ID 标记，从该标记之后（不含）返回对象版本条目，对应请求中的 version-id-marker 参数	string
MaxKeys	ListVersionsResult	单次响应返回结果的最大条目数量，对应请求中的 max-keys 参数	integer
IsTruncated	ListVersionsResult	响应条目是否被截断，布尔值，例如 true 或 false	boolean
NextKeyMarker	ListVersionsResult	仅当响应条目有截断（IsTruncated 为 true）才会返回该节点，该节点的值为当前响应条目中的最	string

版权所有：腾讯云计算（北京）有限责任公司

		后一个对象键，当需要继续请求后续条目时，将该节点的值作为下一次请求的 key-marker 参数传入	
NextVersionIdMarker	ListVersionsResult	仅当响应条目有截断（IsTruncated 为 true）才会返回该节点，该节点的值为当前响应条目中的最后一个对象的版本 ID，当需要继续请求后续条目时，将该节点的值作为下一次请求的 version-id-marker 参数传入。该节点的值可能为空，此时下一次请求的 version-id-marker 参数也需要指定为空。	string
Delimiter	ListVersionsResult	分隔符，对应请求中的 delimiter 参数，且仅当请求中指定了 delimiter 参数才会返回该节点	string
CommonPrefixes	ListVersionsResult	从 prefix 或从头（如未指定 prefix）到首个 delimiter 之间相同的部分，定义为 Common Prefix。仅当请求中指定了 delimiter 参数才有可能返回该节点	Container
Version	ListVersionsResult	对象版本条目	Container
DeleteMarker	ListVersionsResult	对象删除标记条目	Container

Container 节点 CommonPrefixes 的内容：

节点名称（关键字）	父节点	描述	类型
Prefix	ListVersionsResult.CommonPrefixes	单条 Common Prefix 的前缀	string

Container 节点 Version 的内容：

节点名称（关键字）	父节点	描述	类型
Key	ListVersionsResult.Version	对象键	string
VersionId	ListVersionsResult.Version	对象的版本 ID 当未启用版本控制时，该节点的值为空字符串 当启用版本控制时，启用版本控制之前的对象，其版本 ID 为 null 当暂停版本控制时，新上传的对象其版本 ID 为 null，且同一个对象最多只存在一个版本 ID 为 null 的对象版本	string
IsLatest	ListVersionsResult.Version	当前版本是否为该对象的最新版本	boolean

LastModified	ListVersionsResult.Version	当前版本的最后修改时间，为 ISO8601 格式，例如2019-05-24T10:56:40Z	date
ETag	ListVersionsResult.Version	对象的实体标签（Entity Tag），是对象被创建时标识对象内容的信息标签，可用于检查对象的内容是否发生变化 例如"8e0b617ca298a564c3331da28dcb50df"。此头部并不一定返回对象的 MD5 值，而是根据对象上传和加密方式而有所不同	string
Size	ListVersionsResult.Version	对象大小，单位为 Byte	integer
StorageClass	ListVersionsResult.Version	对象存储类型。枚举值请参见 存储类型 文档，例如 STANDARD_IA，ARCHIVE	Enum
StorageTier	ListVersionsResult.Version	当对象存储类型为智能分层存储时，指示对象当前所处的存储层，枚举值：FREQUENT（标准层），INFREQUENT（低频层）。仅当 StorageClass 为 INTELLIGENT_TIERING（智能分层）时才会返回该节点	Enum
Owner	ListVersionsResult.Version	对象持有者信息	Container

Container 节点 Version.Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	ListVersionsResult.Version.Owner	对象持有者的 APPID	string
DisplayName	ListVersionsResult.Version.Owner	对象持有者的名称	string

Container 节点 DeleteMarker 的内容：

节点名称（关键字）	父节点	描述	类型
Key	ListVersionsResult.DeleteMarker	对象键	string
VersionId	ListVersionsResult.DeleteMarker	对象的删除标记的版本 ID	string
IsLatest	ListVersionsResult.DeleteMarker	当前删除标记是否为该对象的最新版本	boolean
LastModified	ListVersionsResult.DeleteMarker	当前删除标记的删除时间，为 ISO8601 格式，例如2019-05-24T10:56:40Z	date
Owner	ListVersionsResult.DeleteMarker	对象持有者信息	Container

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 DeleteMarker.Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	ListVersionsResult.DeleteMarker.Owner	对象持有者的 APPID	string
DisplayName	ListVersionsResult.DeleteMarker.Owner	对象持有者的名称	string

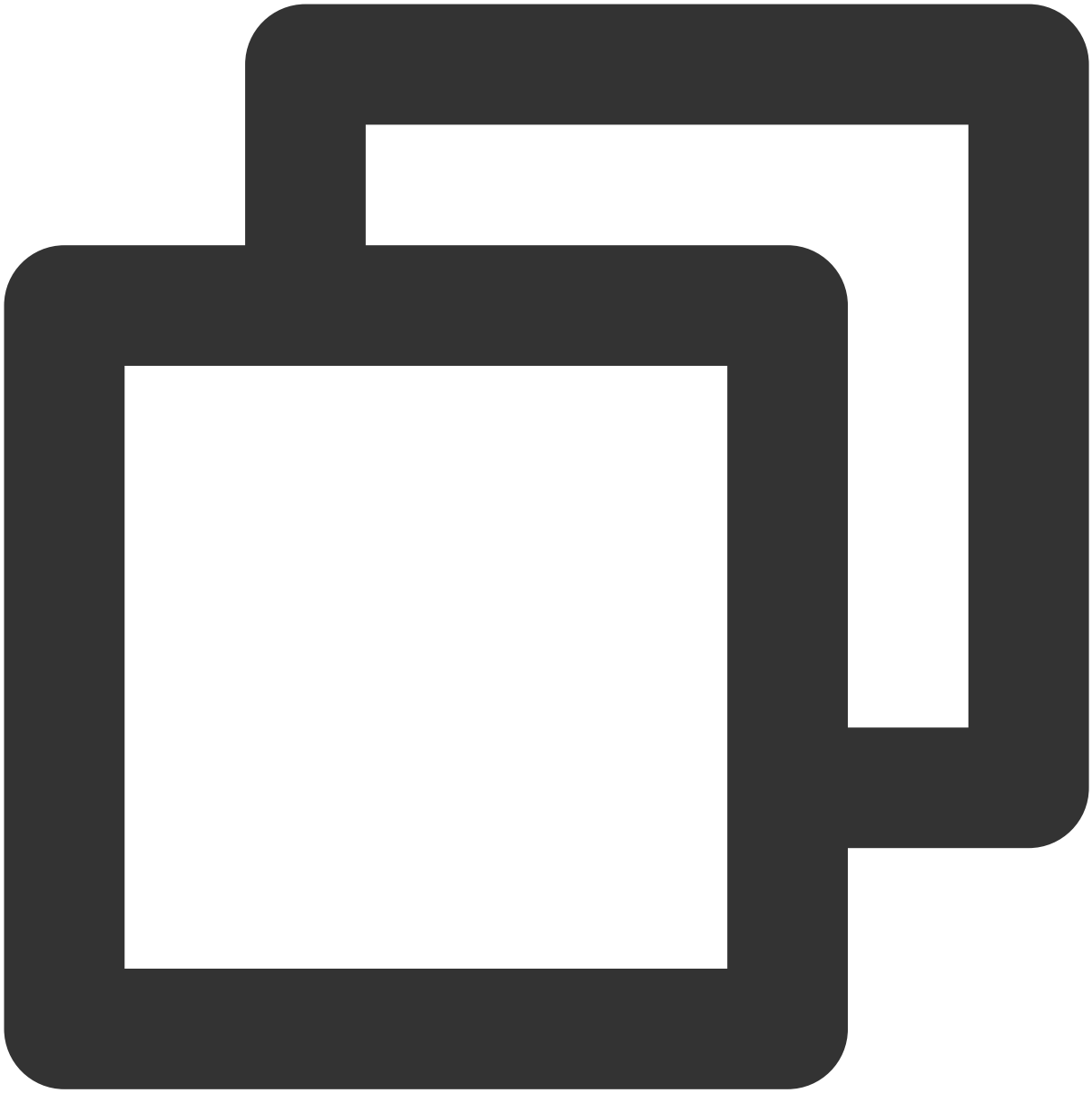
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

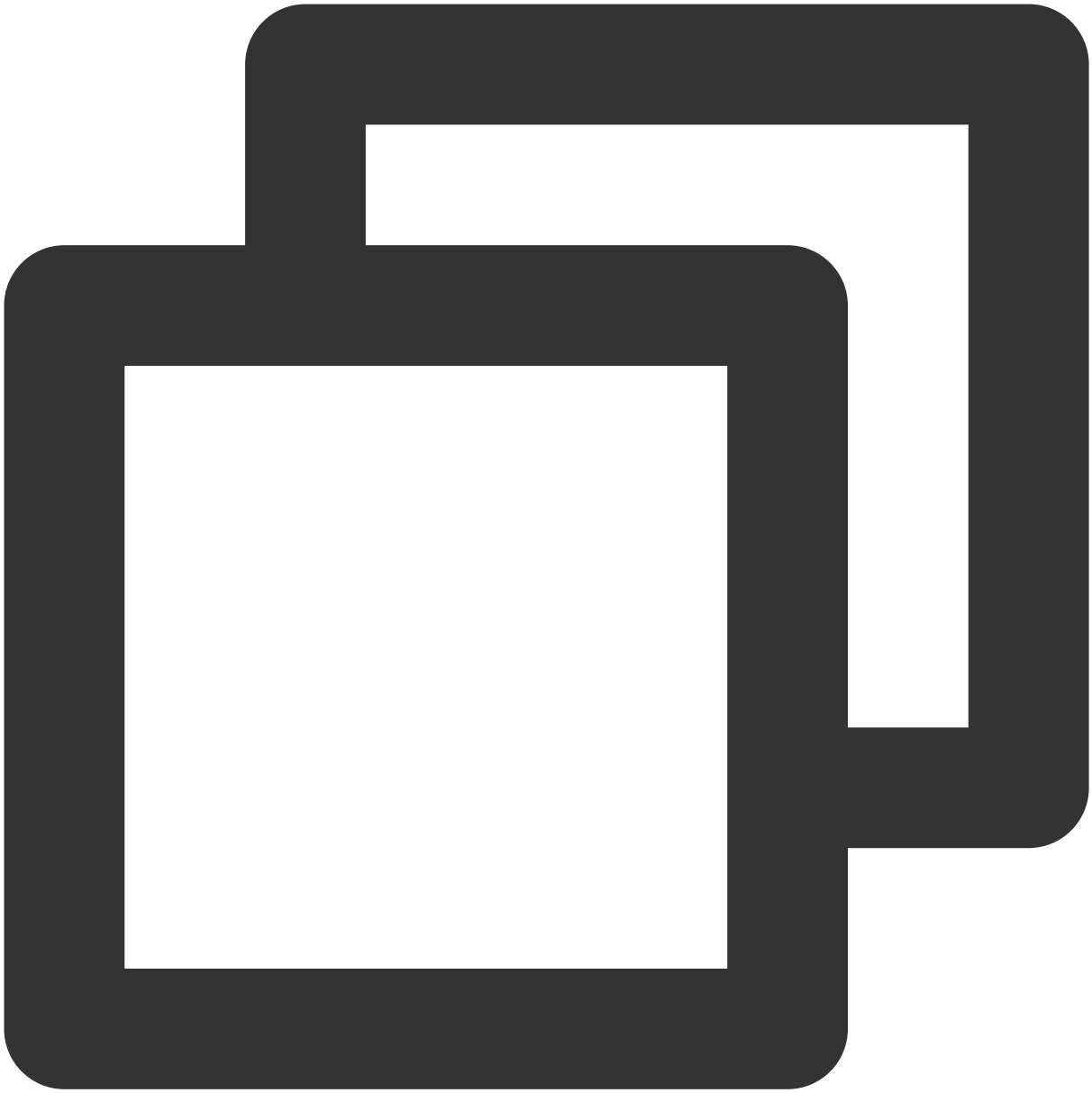
案例一：未启用版本控制

请求



```
GET /?versions HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 03:35:34 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

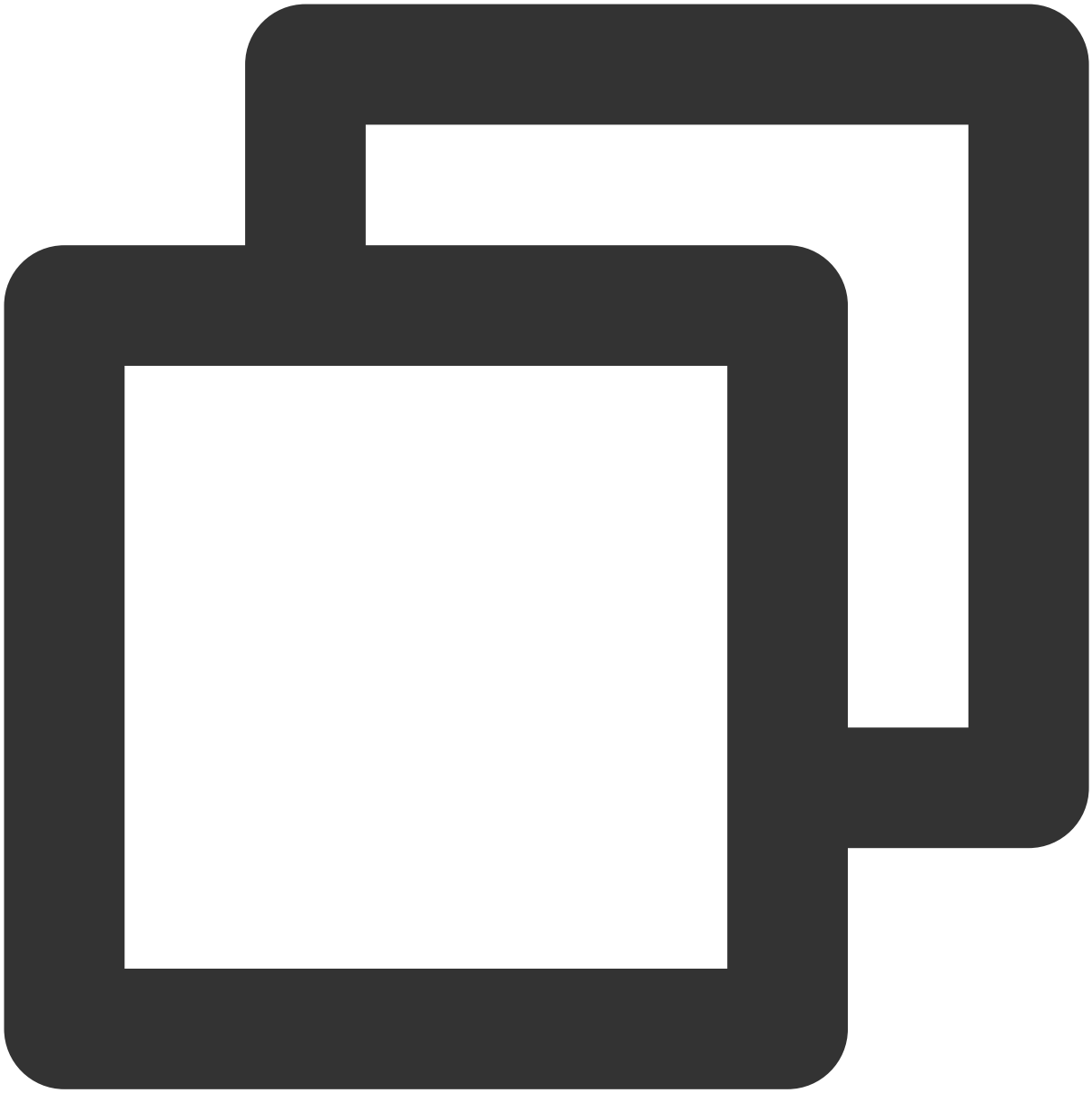


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 952
Connection: close
Date: Thu, 10 Dec 2020 03:35:34 GMT
Server: tencent-cos
x-cos-request-id: NWZkMTk3ODZfZDUyNzVkNjRfNDgxYl8xNjU5****



<ListVersionsResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <KeyMarker/>
            <VersionIdMarker/>
            <MaxKeys>1000</MaxKeys>
            <IsTruncated>false</IsTruncated>
            <Version>
                <Key>example-object-1.jpg</Key>
                <VersionId/>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-10T03:35:24.000Z</LastModified>
                <ETag>&amp;quot;0f0cd12c48979d1bf3f95255a36cb861&amp;quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <Version>
                <Key>example-object-2.jpg</Key>
                <VersionId/>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-10T03:35:24.000Z</LastModified>
                <ETag>&amp;quot;dcffaafe67632b2bd2dd0b9456eafca7&amp;quot;</ETag>
                <Size>23</Size>
                <StorageClass>INTELLIGENT_TIERING</StorageClass>
                <StorageTier>FREQUENT</StorageTier>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

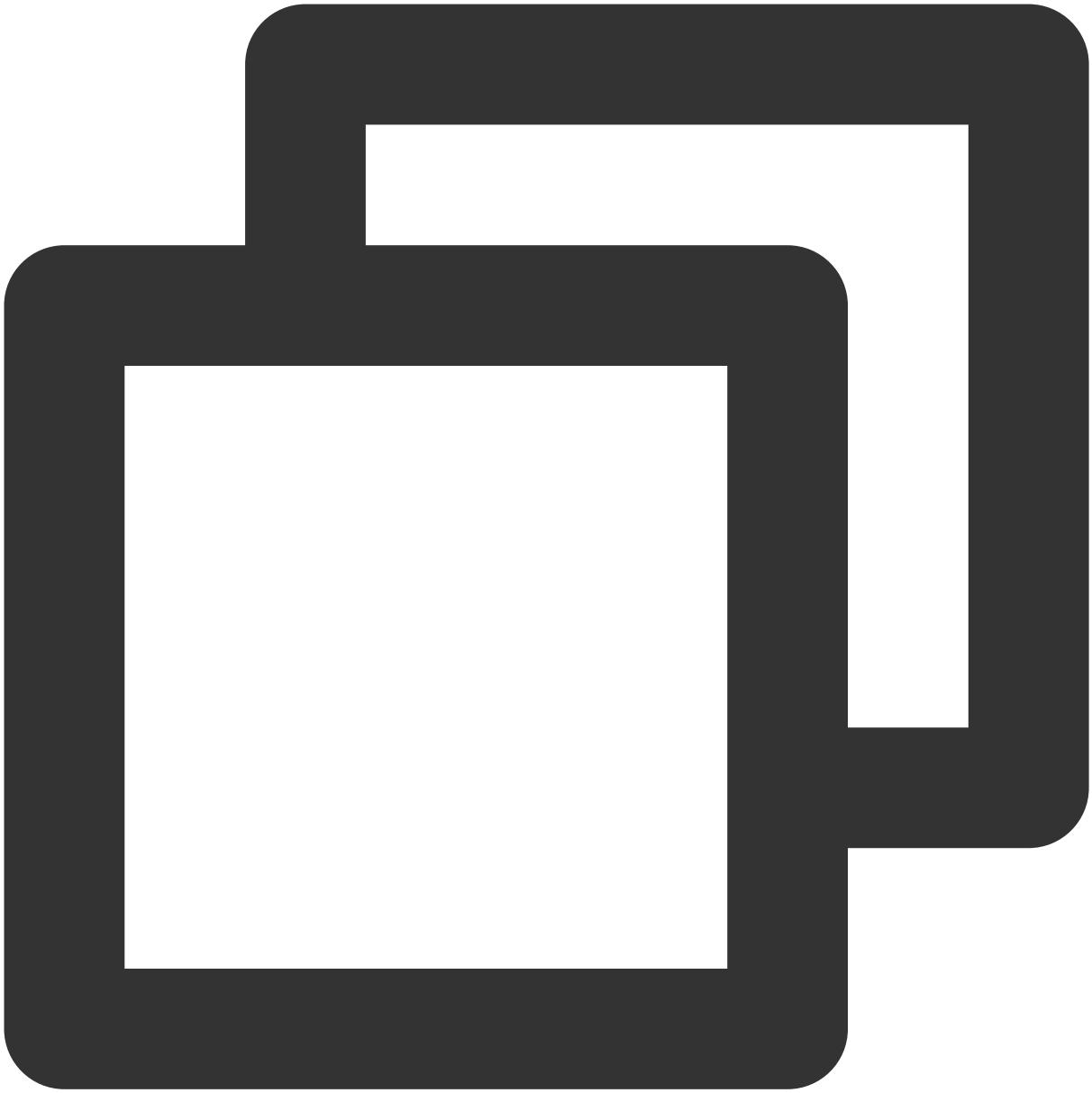
案例二：启用版本控制（续接案例一，在启用版本控制前上传的对象，其版本 ID 为 null）

请求



```
GET /?versions HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 03:36:05 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 2070
Connection: close
Date: Thu, 10 Dec 2020 03:36:05 GMT
Server: tencent-cos
x-cos-request-id: NWZkMTk3YTVfYjFiODJhMDlfNTg0MDZfMTdj****



<ListVersionsResult>
```

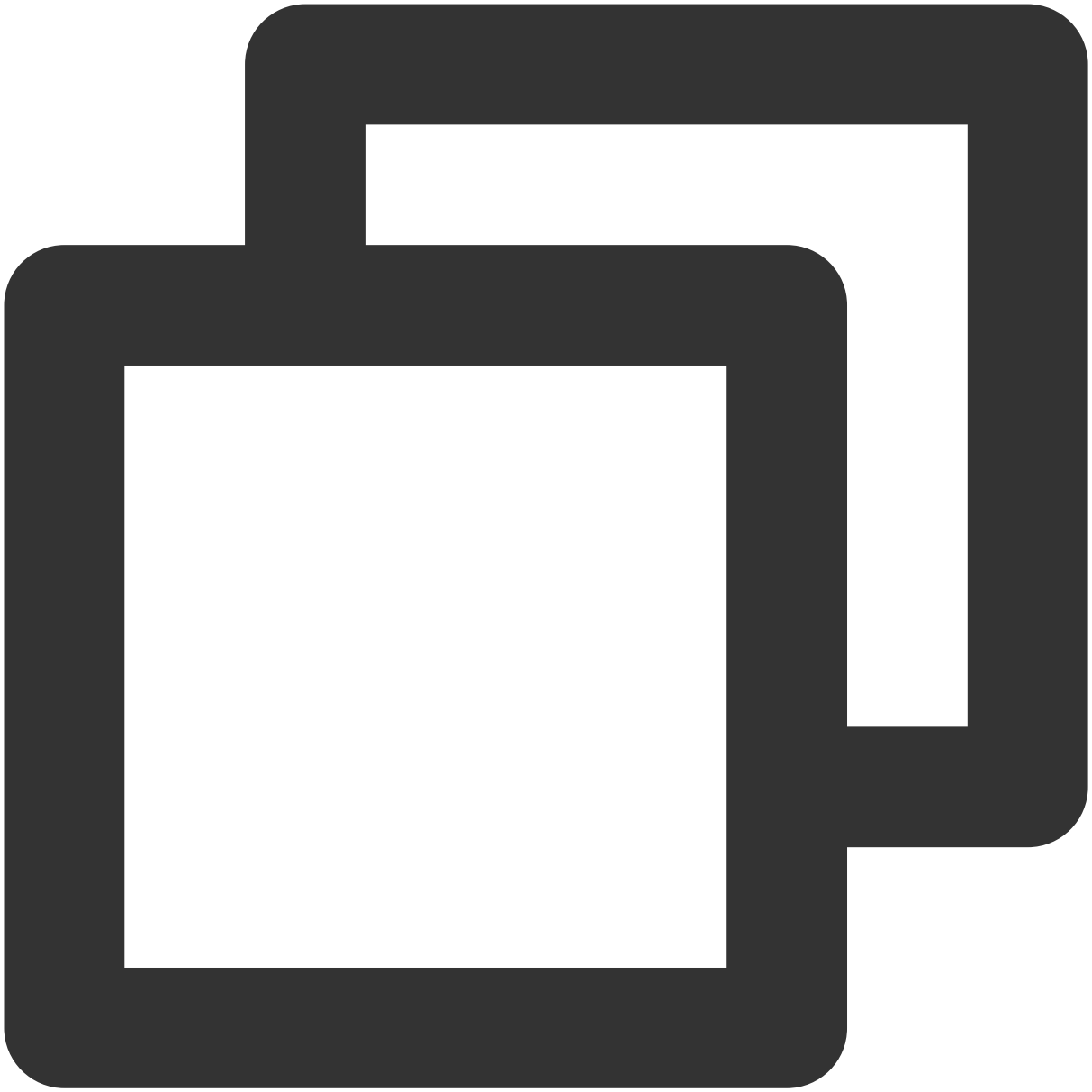
```
<Name>examplebucket-1250000000</Name>
<Prefix/>
<KeyMarker/>
<VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Version>
    <Key>example-object-1.jpg</Key>
    <VersionId>null</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T03:35:24.000Z</LastModified>
    <ETag>&amp;quot;0f0cd12c48979d1bf3f95255a36cb861&amp;quot;</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-object-2.jpg</Key>
    <VersionId>MTg0NDUxMzY1MDIzNjQ3NTcwMDk</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T03:35:44.000Z</LastModified>
    <ETag>&amp;quot;51ffadb19b3bf062ecd0c6f044a4d4ce&amp;quot;</ETag>
    <Size>23</Size>
    <StorageClass>STANDARD_IA</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-object-2.jpg</Key>
    <VersionId>null</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2020-12-10T03:35:24.000Z</LastModified>
    <ETag>&amp;quot;dcffaafe67632b2bd2dd0b9456eafca7&amp;quot;</ETag>
    <Size>23</Size>
    <StorageClass>INTELLIGENT_TIERING</StorageClass>
    <StorageTier>FREQUENT</StorageTier>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<DeleteMarker>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                <Key>example-object-3.jpg</Key>
                <VersionId>MTg0NDUxMzY1MDIzNTQ1ODM2NjE</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-10T03:35:54.000Z</LastModified>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </DeleteMarker>
            <Version>
                <Key>example-object-3.jpg</Key>
                <VersionId>MTg0NDUxMzY1MDIzNjQ3NjM2MDY</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-10T03:35:44.000Z</LastModified>
                <ETag>&amp;quot;b2f1d893c5fde000ee8ea6eca18ed81f&amp;quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

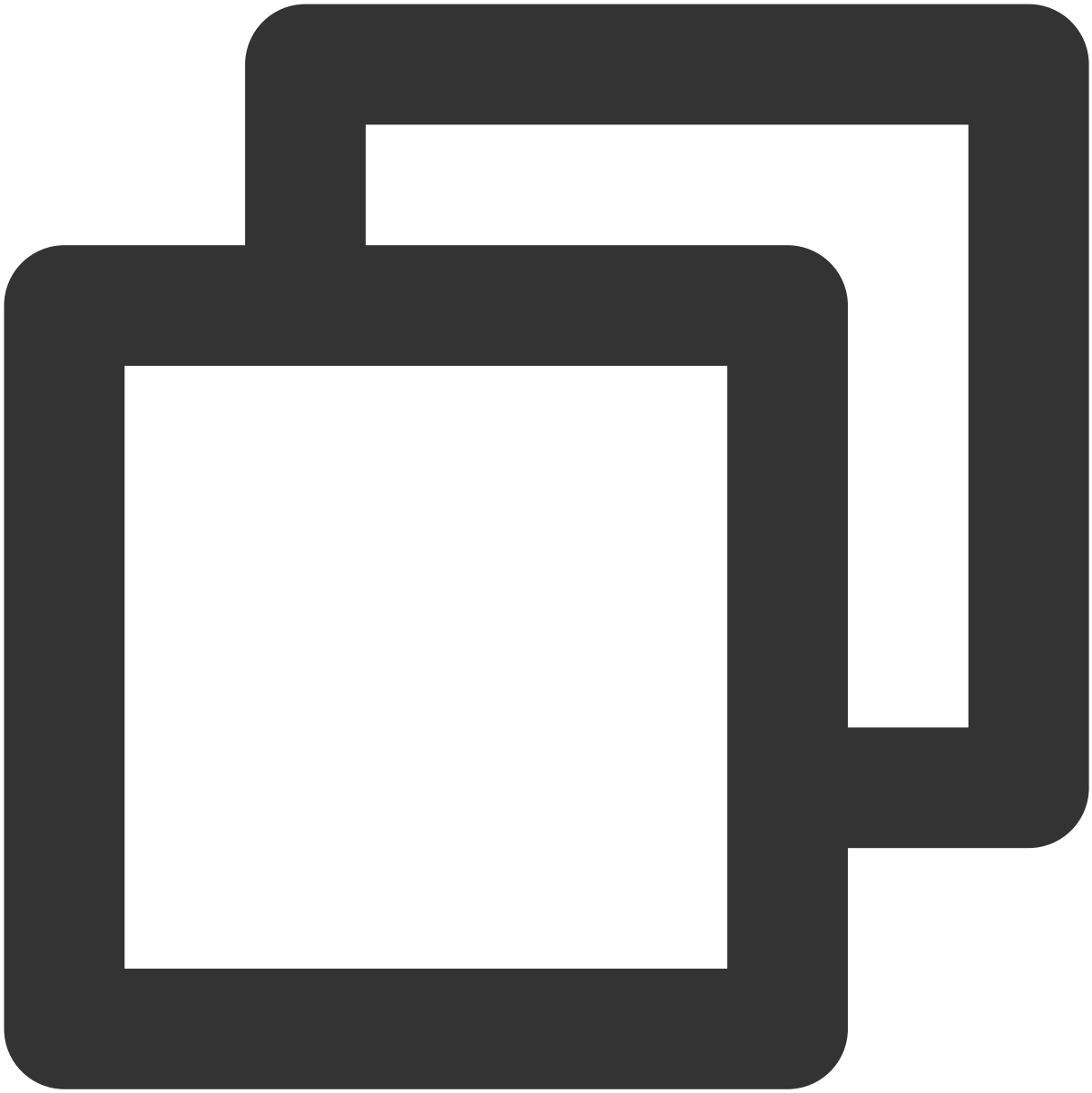
案例三：暂停版本控制（续接案例二，暂停版本控制后新上传的对象其版本 ID 为 null，且同一个对象最多只存在一个版本 ID 为 null 的对象版本）

请求



```
GET /?versions HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 03:36:25 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



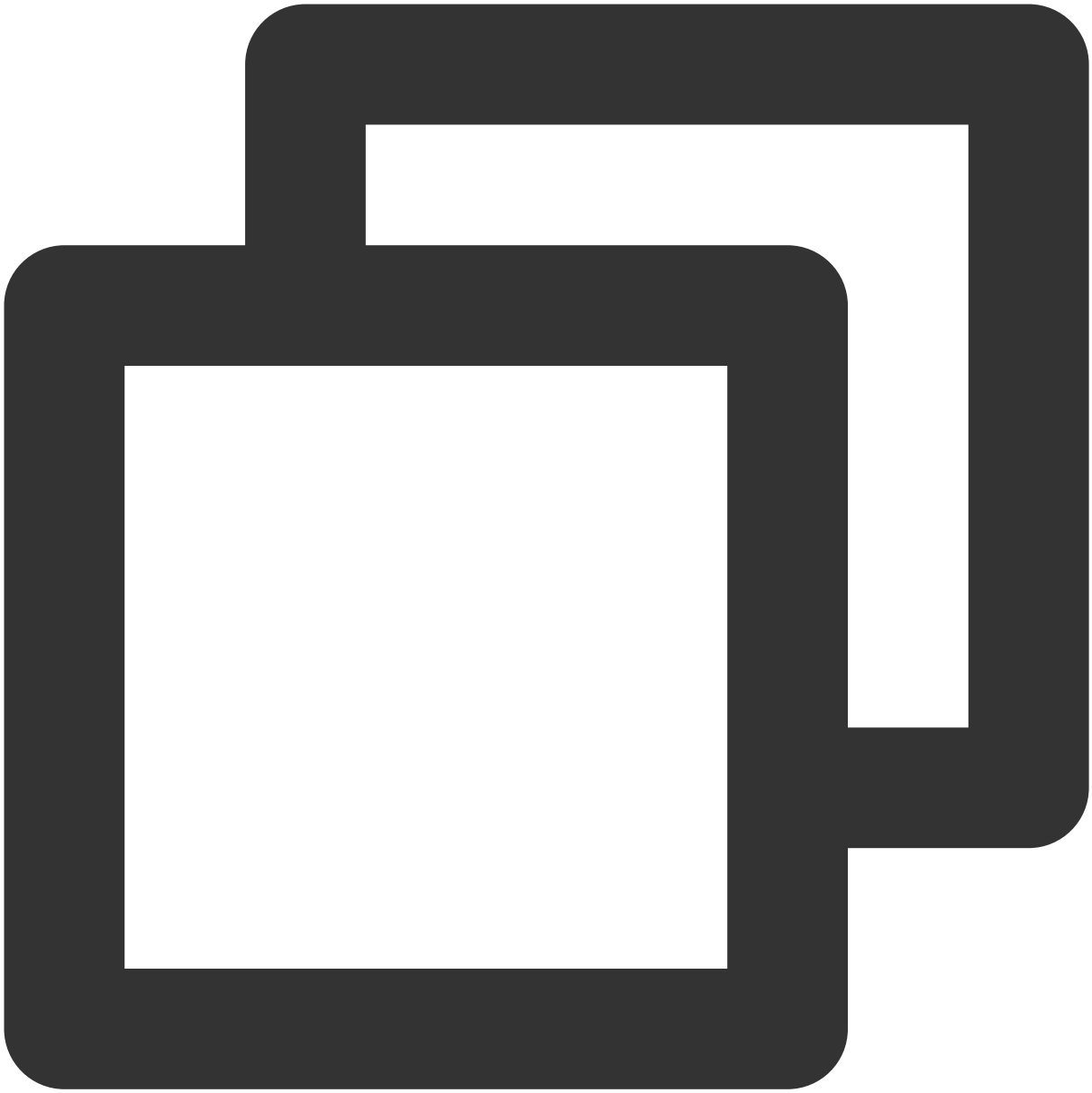
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 2396
Connection: close
Date: Thu, 10 Dec 2020 03:36:25 GMT
Server: tencent-cos
x-cos-request-id: NWZkMTk3YjlfNDhhOTBiMDlfMTYzNTZfMTIw****



<ListVersionsResult>
```

```
<Name>examplebucket-1250000000</Name>
<Prefix/>
<KeyMarker/>
<VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Version>
    <Key>example-object-1.jpg</Key>
    <VersionId>null</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T03:35:24.000Z</LastModified>
    <ETag>&amp;quot;0f0cd12c48979d1bf3f95255a36cb861&amp;quot;</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-object-2.jpg</Key>
    <VersionId>null</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T03:36:15.000Z</LastModified>
    <ETag>&amp;quot;51ffadb19b3bf062ecd0c6f044a4d4ce&amp;quot;</ETag>
    <Size>23</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-object-2.jpg</Key>
    <VersionId>MTg0NDUxMzY1MDIzNjQ3NTcwMDk</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2020-12-10T03:35:44.000Z</LastModified>
    <ETag>&amp;quot;51ffadb19b3bf062ecd0c6f044a4d4ce&amp;quot;</ETag>
    <Size>23</Size>
    <StorageClass>STANDARD_IA</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-object-3.jpg</Key>
```

```
                <VersionId>null</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-10T03:36:15.000Z</LastModified>
                <ETag>&amp;quot;b2f1d893c5fde000ee8ea6eca18ed81f&amp;quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <DeleteMarker>
                <Key>example-object-3.jpg</Key>
                <VersionId>MTg0NDUxMzY1MDIzNTQ1ODM2NjE</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-10T03:35:54.000Z</LastModified>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </DeleteMarker>
            <Version>
                <Key>example-object-3.jpg</Key>
                <VersionId>MTg0NDUxMzY1MDIzNjQ3NjM2MDY</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-10T03:35:44.000Z</LastModified>
                <ETag>&amp;quot;b2f1d893c5fde000ee8ea6eca18ed81f&amp;quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

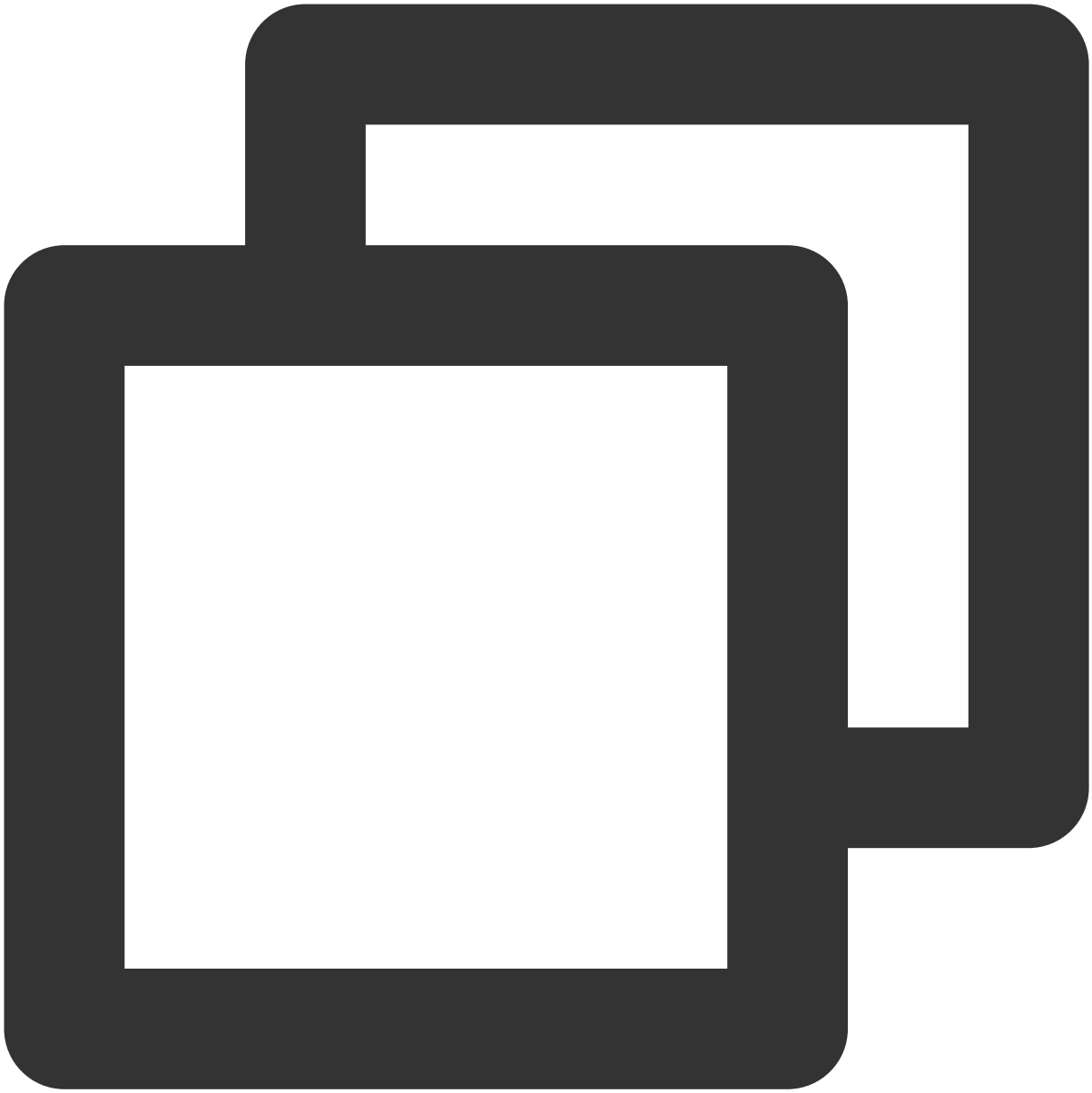
案例四：带 encoding-type 参数（对象键使用 URL 编码）

请求



```
GET /?versions&encoding-type=url HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 04:54:22 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



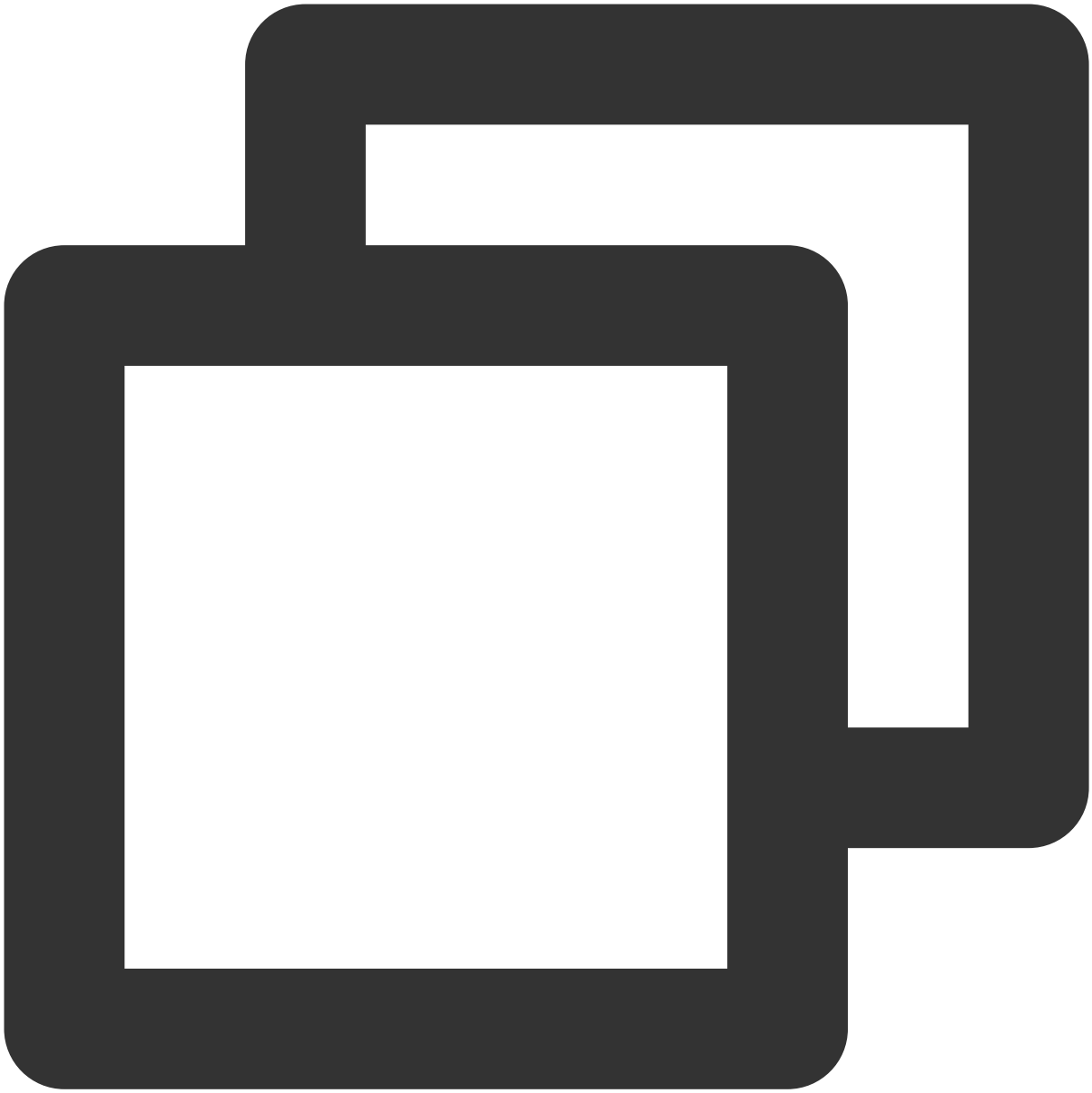
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 2167
Connection: close
Date: Thu, 10 Dec 2020 04:54:22 GMT
Server: tencent-cos
x-cos-request-id: NWZkMWE5ZmVfZTdjODJhMDlfMzgxZl8xMzQ1****



<ListVersionsResult>
```

```
<EncodingType>url</EncodingType>
<Name>examplebucket-1250000000</Name>
<Prefix/>
<KeyMarker/>
<VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Version>
    <Key>Tencent%20Cloud.jpg</Key>
    <VersionId>MTg0NDUxMzY0OTc2NjcxNzE4NjA</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T04:54:02.000Z</LastModified>
    <ETag>&amp;quot;ee8de918d05640145b18f70f4c3aa602&amp;quot;</ETag>
    <Size>16</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<DeleteMarker>
    <Key>%E7%85%A7%E7%89%87/2020%E5%B9%B4/IMG0001.jpg</Key>
    <VersionId>MTg0NDUxMzY0OTc2NTY5MDU3OTc</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T04:54:12.000Z</LastModified>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</DeleteMarker>
<Version>
    <Key>%E7%85%A7%E7%89%87/2020%E5%B9%B4/IMG0001.jpg</Key>
    <VersionId>MTg0NDUxMzY0OTc2NjcwNjMwMDQ</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2020-12-10T04:54:02.000Z</LastModified>
    <ETag>&amp;quot;ee8de918d05640145b18f70f4c3aa602&amp;quot;</ETag>
    <Size>16</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>%E8%85%BE%E8%AE%AF%E4%BA%91.jpg</Key>
    <VersionId>MTg0NDUxMzY0OTc2NTY5MDA4NzU</VersionId>
    <IsLatest>true</IsLatest>
```

```
                <LastModified>2020-12-10T04:54:12.000Z</LastModified>
                <ETag>&amp;quot;8bb76a43f75d3b96442c470f8ec23b89&amp;quot;</ETag>
                <Size>16</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <Version>
                <Key>%E8%85%BE%E8%AE%AF%E4%BA%91.jpg</Key>
                <VersionId>MTg0NDUxMzY0OTc2NjcxNzA2ODg</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-10T04:54:02.000Z</LastModified>
                <ETag>&amp;quot;dcc880c5eba9e002bc4567c733b0e63e&amp;quot;</ETag>
                <Size>13</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

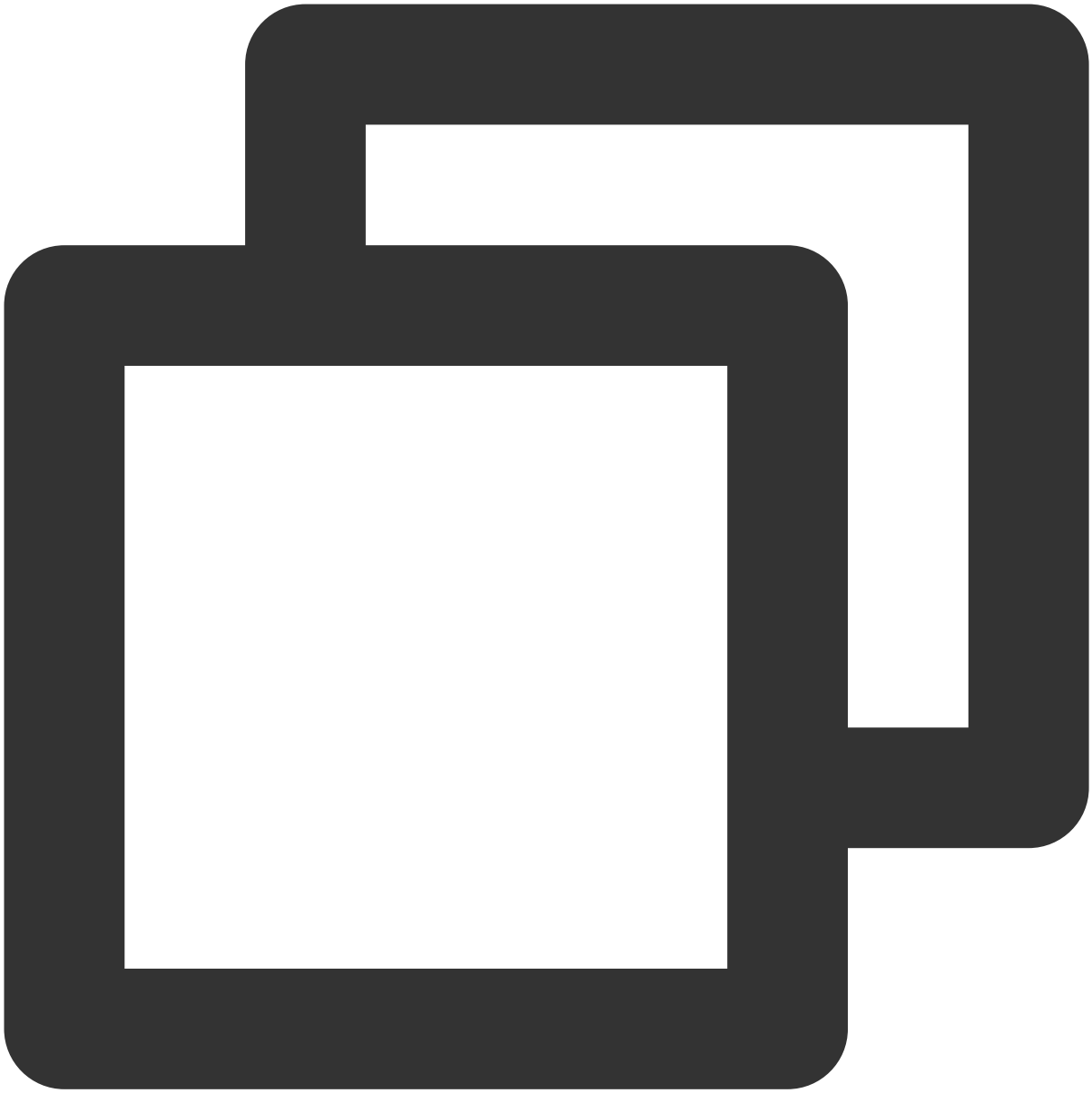
案例五：带 delimiter 参数（列出根目录下的对象版本和子目录）

请求



```
GET /?versions&delimiter=%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 07:04:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



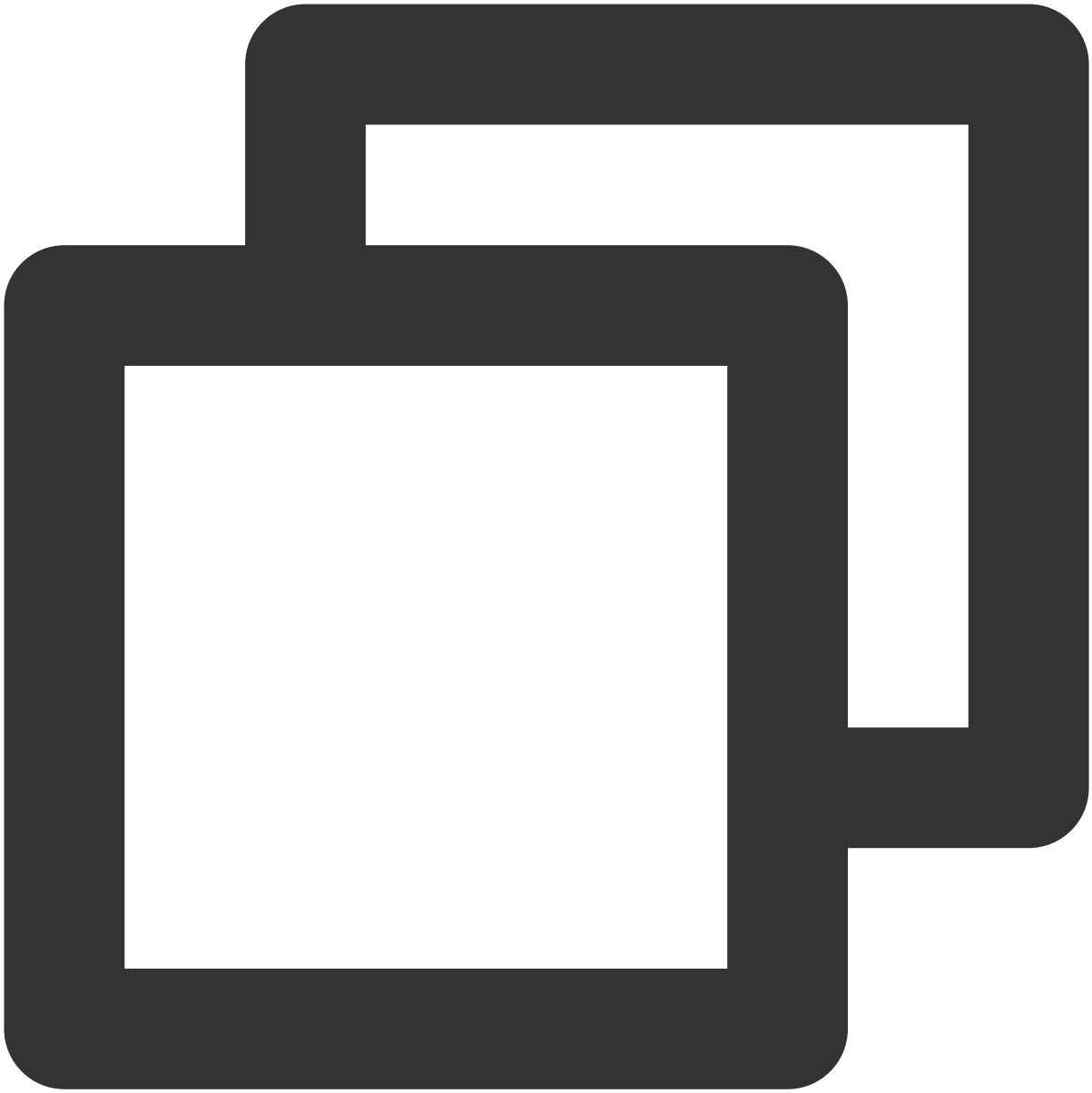
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1893
Connection: close
Date: Thu, 10 Dec 2020 07:04:03 GMT
Server: tencent-cos
x-cos-request-id: NWZkMWM4NjNfNzFjODJhMDlfMjlhZTRfMTg5****



<ListVersionsResult>
```

```
<Name>examplebucket-1250000000</Name>
<Prefix/>
<KeyMarker/>
<VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Delimiter>/</Delimiter>
<CommonPrefixes>
    <Prefix>example-folder-1/</Prefix>
</CommonPrefixes>
<CommonPrefixes>
    <Prefix>example-folder-2/</Prefix>
</CommonPrefixes>
<DeleteMarker>
    <Key>example-object-1.jpg</Key>
    <VersionId>MTg0NDUxMzY0ODk4NzYzNTE4Nzk</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T07:03:53.000Z</LastModified>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</DeleteMarker>
<Version>
    <Key>example-object-1.jpg</Key>
    <VersionId>MTg0NDUxMzY0ODk4ODY2NTcxMzY</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2020-12-10T07:03:42.000Z</LastModified>
    <ETag>&amp;quot;0f0cd12c48979d1bf3f95255a36cb861&amp;quot;</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
            <ID>1250000000</ID>
            <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-object-2.jpg</Key>
    <VersionId>MTg0NDUxMzY0ODk4NzYzODYyODA</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T07:03:53.000Z</LastModified>
    <ETag>&amp;quot;51ffadb19b3bf062ecd0c6f044a4d4ce&amp;quot;</ETag>
    <Size>23</Size>
    <StorageClass>INTELLIGENT_TIERING</StorageClass>
    <StorageTier>FREQUENT</StorageTier>
    <Owner>
            <ID>1250000000</ID>
```

```
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <Version>
                <Key>example-object-2.jpg</Key>
                <VersionId>MTg0NDUxMzY0ODk4ODY2NTA0Mjk</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-10T07:03:42.000Z</LastModified>
                <ETag>&amp;quot;dcffaafe67632b2bd2dd0b9456eafca7&amp;quot;</ETag>
                <Size>23</Size>
                <StorageClass>STANDARD_IA</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

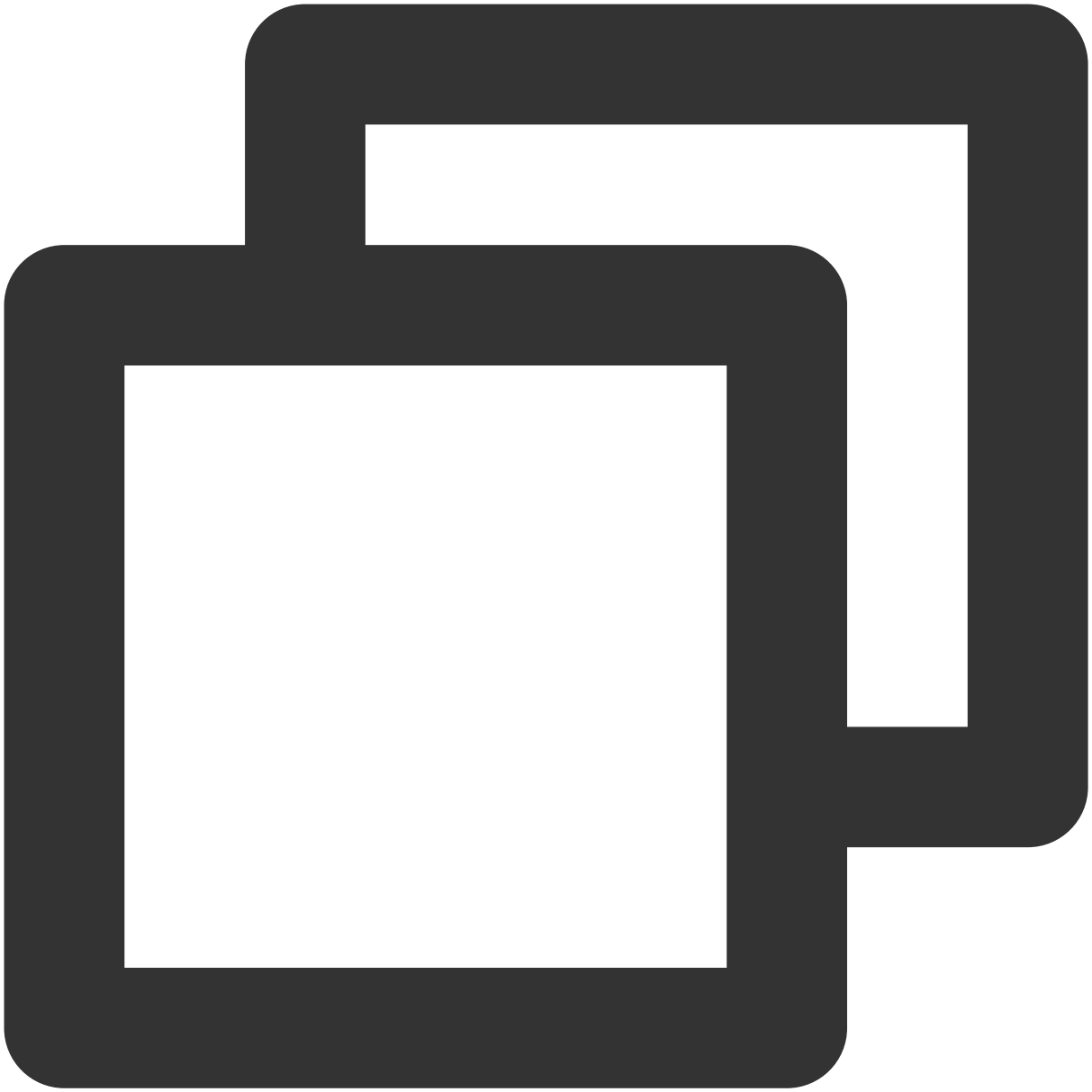
案例六：带 prefix 和 delimiter 参数（列出指定子目录下的对象版本和子目录）

请求



```
GET /?versions&prefix=example-folder-1%2F&delimiter=%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 10 Dec 2020 07:04:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



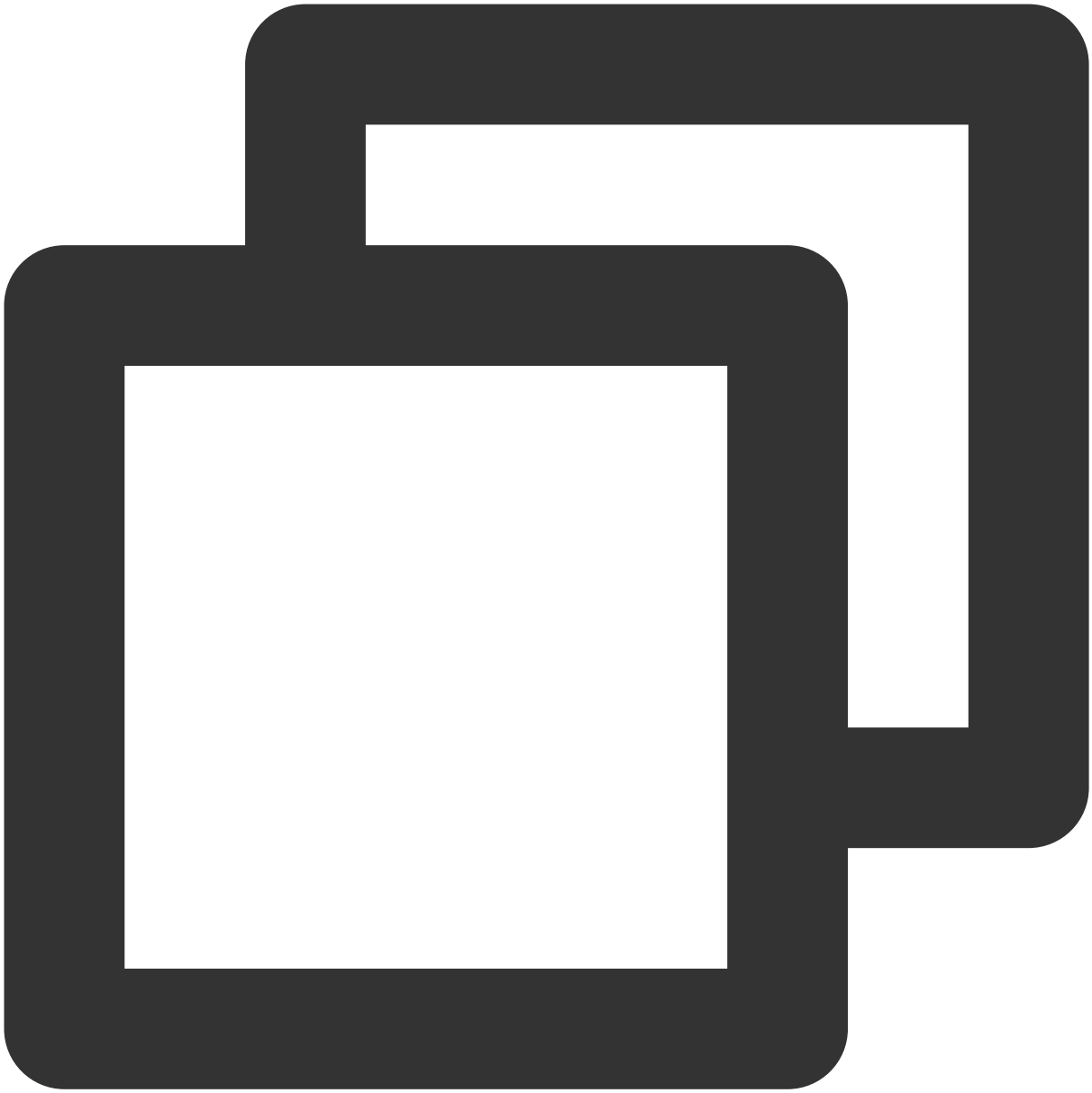
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1960
Connection: close
Date: Thu, 10 Dec 2020 07:04:03 GMT
Server: tencent-cos
x-cos-request-id: NWZkMWM4NjNfZWVjODJhMDlfNTNmN18xNWNj****



<ListVersionsResult>
```

```
<Name>examplebucket-1250000000</Name>
<Prefix>example-folder-1/</Prefix>
<KeyMarker/>
<VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Delimiter>/</Delimiter>
<CommonPrefixes>
    <Prefix>example-folder-1/sub-folder-1/</Prefix>
</CommonPrefixes>
<CommonPrefixes>
    <Prefix>example-folder-1/sub-folder-2/</Prefix>
</CommonPrefixes>
<DeleteMarker>
    <Key>example-folder-1/example-object-1.jpg</Key>
    <VersionId>MTg0NDUxMzY0ODk4NzYzNjE4NzA</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T07:03:53.000Z</LastModified>
    <Owner>
        <ID>1250000000</ID>
        <DisplayName>1250000000</DisplayName>
    </Owner>
</DeleteMarker>
<Version>
    <Key>example-folder-1/example-object-1.jpg</Key>
    <VersionId>MTg0NDUxMzY0ODk4ODY2NTAzOTY</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2020-12-10T07:03:42.000Z</LastModified>
    <ETag>&amp;quot;f173c1199e3d3b53dd91223cae16fb42&amp;quot;</ETag>
    <Size>37</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
        <ID>1250000000</ID>
        <DisplayName>1250000000</DisplayName>
    </Owner>
</Version>
<Version>
    <Key>example-folder-1/example-object-2.jpg</Key>
    <VersionId>MTg0NDUxMzY0ODk4NzYzNzc2ODY</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2020-12-10T07:03:53.000Z</LastModified>
    <ETag>&amp;quot;f52cc0bc2042d201b852385927bcab95&amp;quot;</ETag>
    <Size>40</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
        <ID>1250000000</ID>
        <DisplayName>1250000000</DisplayName>
```

```
                </Owner>
            </Version>
            <Version>
                <Key>example-folder-1/example-object-2.jpg</Key>
                <VersionId>MTg0NDUxMzY0ODk4ODY2NjQ2NDc</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-10T07:03:42.000Z</LastModified>
                <ETag>&amp;quot;7b3be6c39746a970d628e6ab9f250342&amp;quot;</ETag>
                <Size>40</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

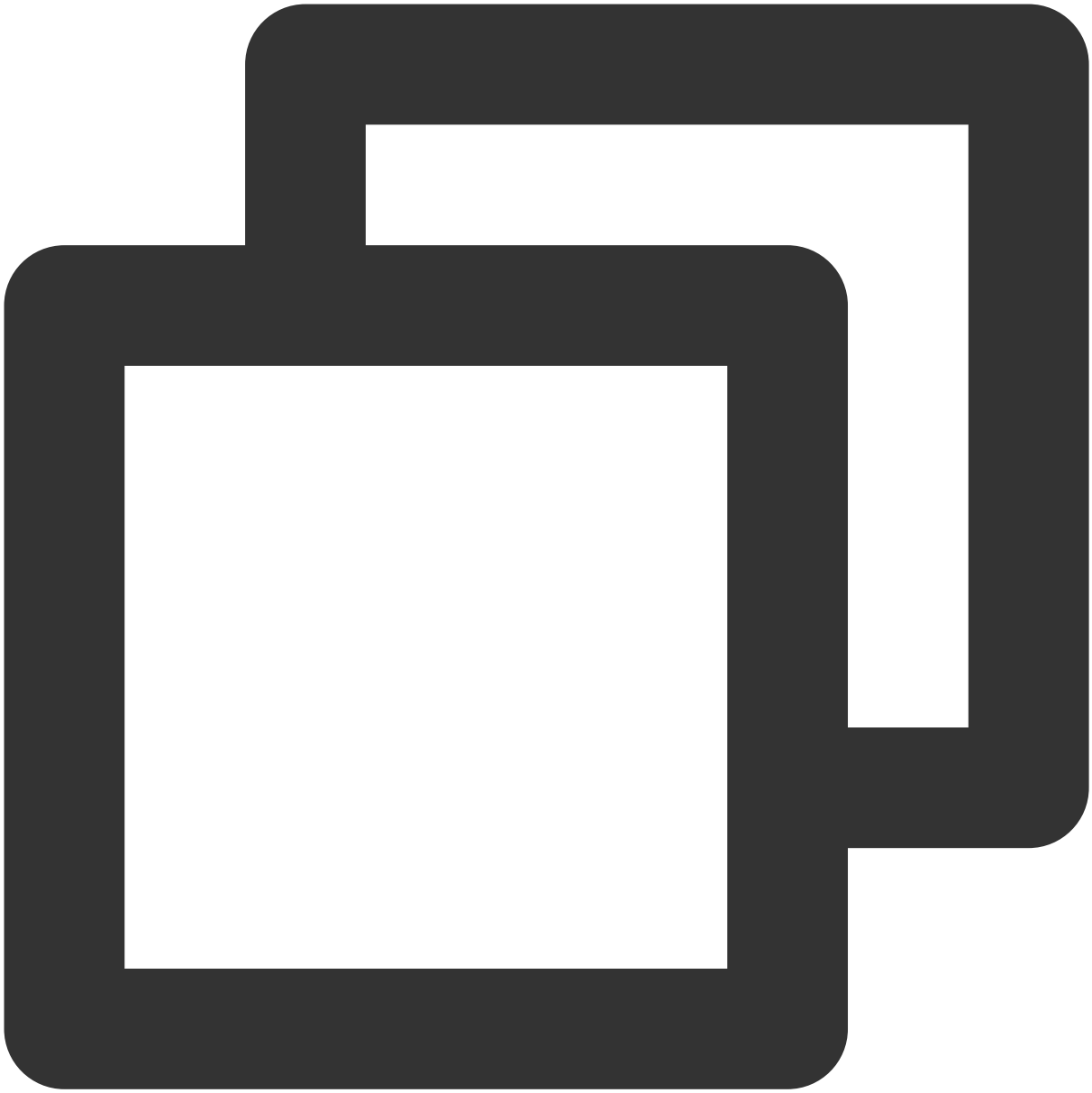
案例七：需分页时获取第一页（案例中限制了 max-keys，如不限制则默认为 1000，Version 与 DeleteMarker 合计条目数不超过 max-keys 指定的值）

请求



```
GET /?versions&max-keys=3 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 08 Dec 2020 12:46:40 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

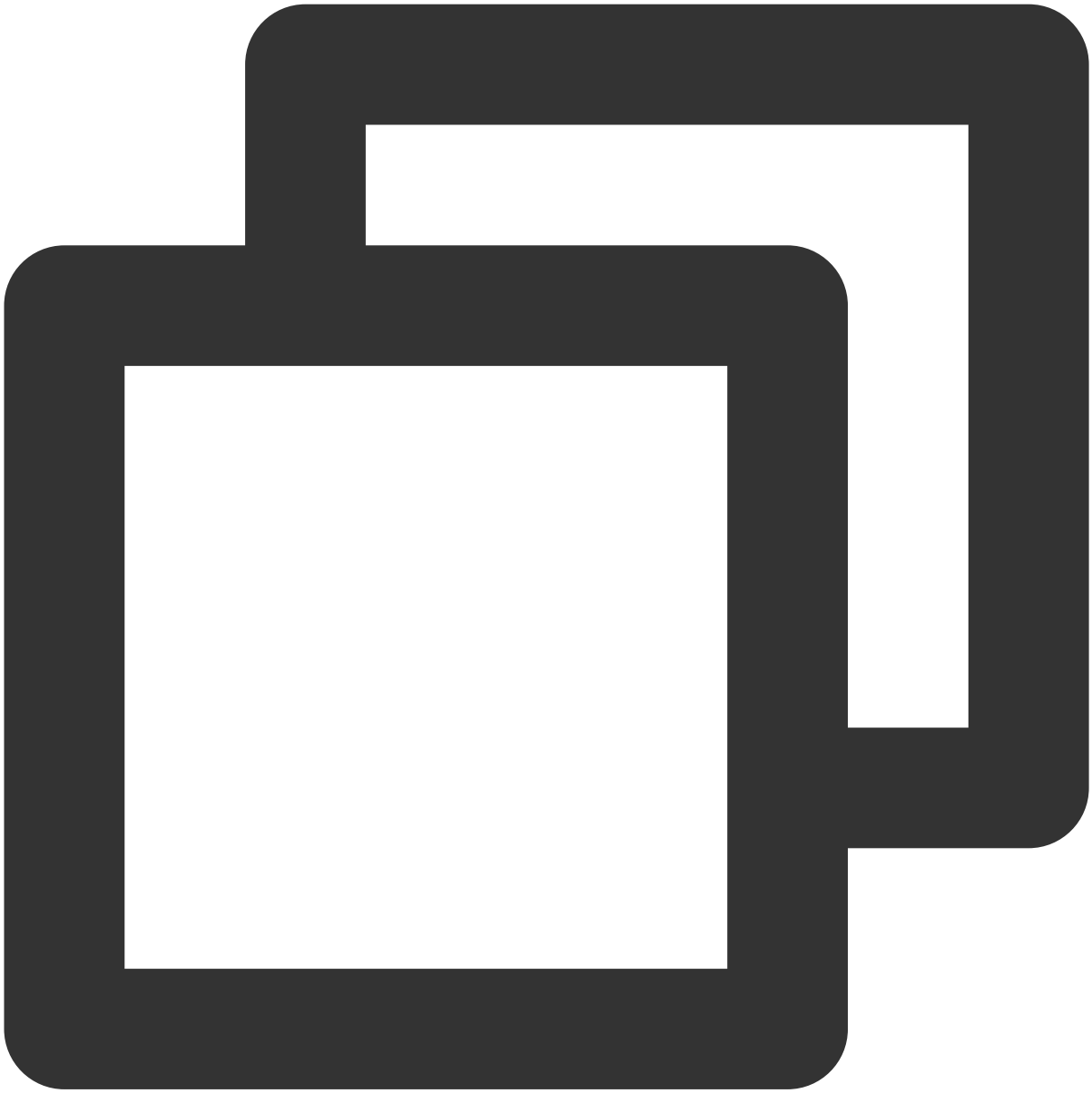


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1390
Connection: close
Date: Tue, 08 Dec 2020 12:46:40 GMT
Server: tencent-cos
x-cos-request-id: NWZjZjc1YjBfYjBhODBiMDlfZDU5Yl9jYjBl****


<ListVersionsResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <KeyMarker/>
            <VersionIdMarker/>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>true</IsTruncated>
            <NextKeyMarker>example-object-2.jpg</NextKeyMarker>
            <NextVersionIdMarker>MTg0NDUxMzY2NDIxMjk3MzMwOTM</NextVersionIdMarker>
            <Version>
                <Key>example-object-1.jpg</Key>
                <VersionId>MTg0NDUxMzY2NDIxNTAwOTk4NzA</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-08T12:45:59.000Z</LastModified>
                <ETag>&quot;15f0f671f04af108023b5603bea2bfda&quot;</ETag>
                <Size>23</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <Version>
                <Key>example-object-1.jpg</Key>
                <VersionId>MTg0NDUxMzY2NDIxNjAyODcyNzk</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-08T12:45:49.000Z</LastModified>
                <ETag>&quot;601389434817e2781d8efb35c0e44717&quot;</ETag>
                <Size>23</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <DeleteMarker>
                <Key>example-object-2.jpg</Key>
                <VersionId>MTg0NDUxMzY2NDIxMjk3MzMwOTM</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-08T12:46:19.000Z</LastModified>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </DeleteMarker>
</ListVersionsResult>
```

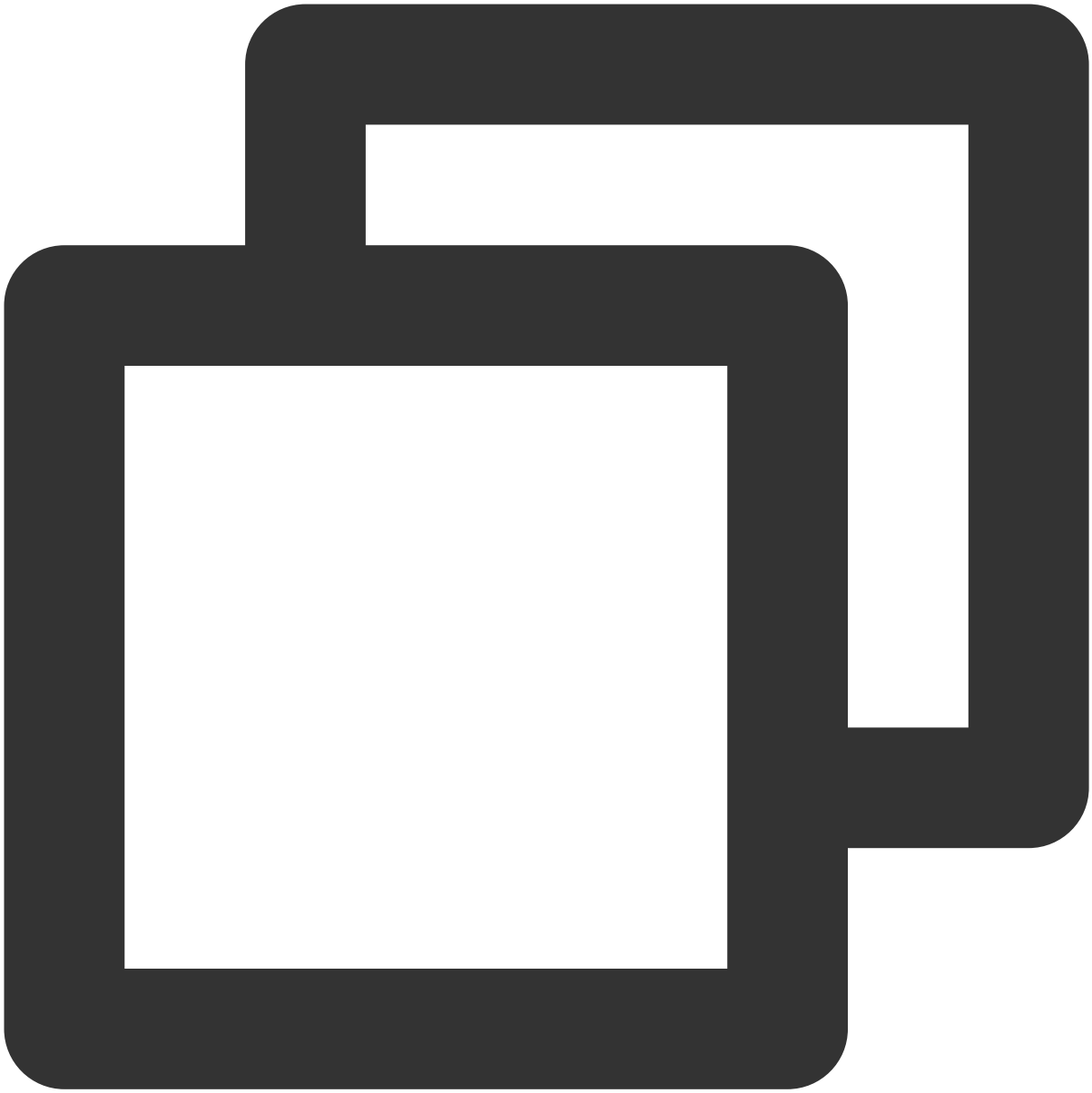
案例八：需分页时获取后续页（续接案例七，使用 key-marker 和 version-id-marker 请求参数）

请求



```
GET /?versions&max-keys=3&key-marker=example-object-2.jpg&version-id-marker=MTg0NDU
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 08 Dec 2020 12:46:41 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

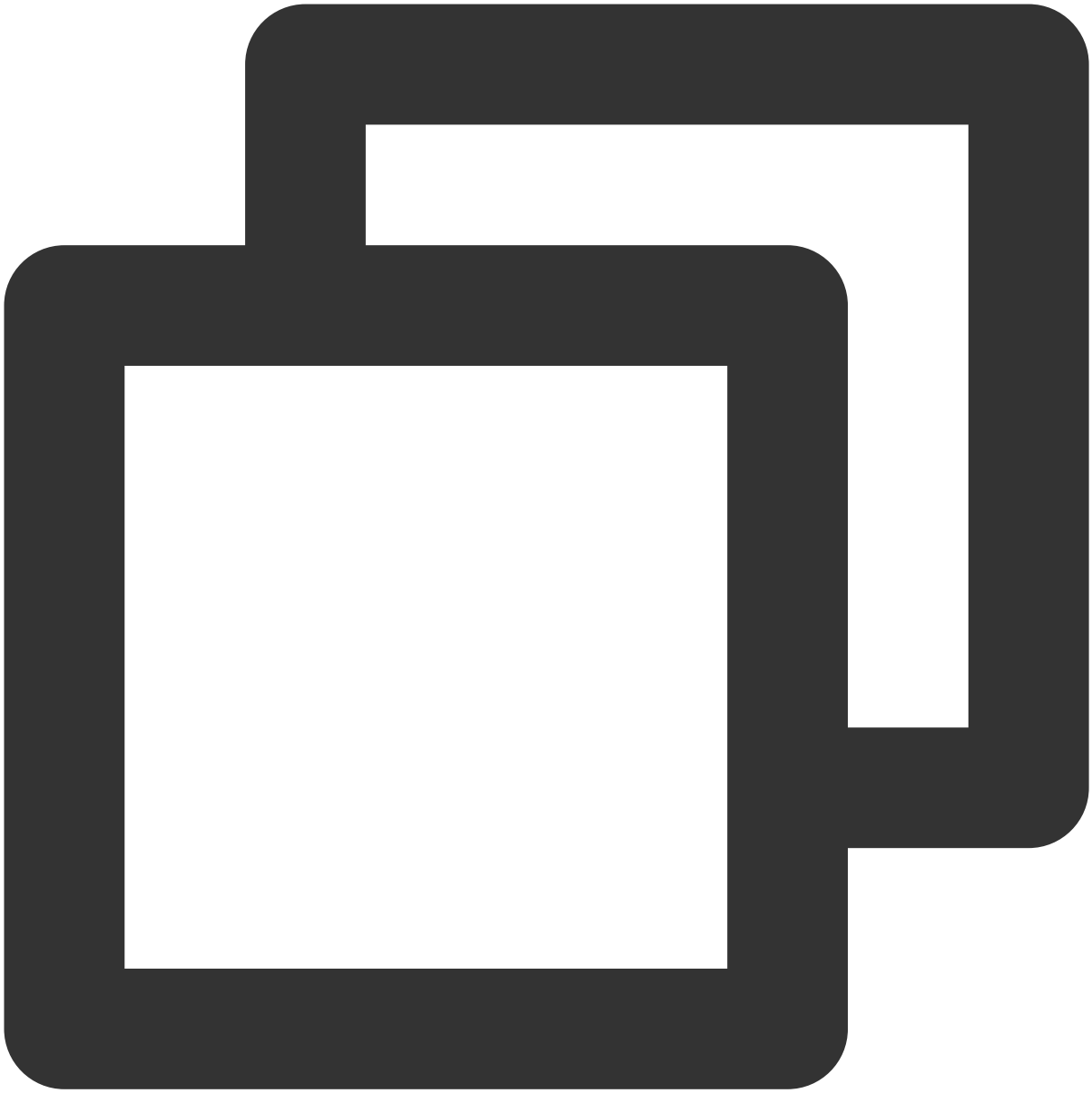


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1052
Connection: close
Date: Tue, 08 Dec 2020 12:46:41 GMT
Server: tencent-cos
x-cos-request-id: NWZjZjc1YjFfNjljMDBiMDlfNjQwOV8xYjM2****



<ListVersionsResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <KeyMarker>example-object-2.jpg</KeyMarker>
            <VersionIdMarker>MTg0NDUxMzY2NDIxMjk3MzMwOTM</VersionIdMarker>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>false</IsTruncated>
            <Version>
                <Key>example-object-2.jpg</Key>
                <VersionId>MTg0NDUxMzY2NDIxMzk5MDk1NzU</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-08T12:46:09.000Z</LastModified>
                <ETag>&quot;51370fc64b79d0d3c7c609635be1c41f&quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <Version>
                <Key>example-object-3.jpg</Key>
                <VersionId>MTg0NDUxMzY2NDIxMTk1NTEyNDI</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-08T12:46:30.000Z</LastModified>
                <ETag>&quot;b2f1d893c5fde000ee8ea6eca18ed81f&quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

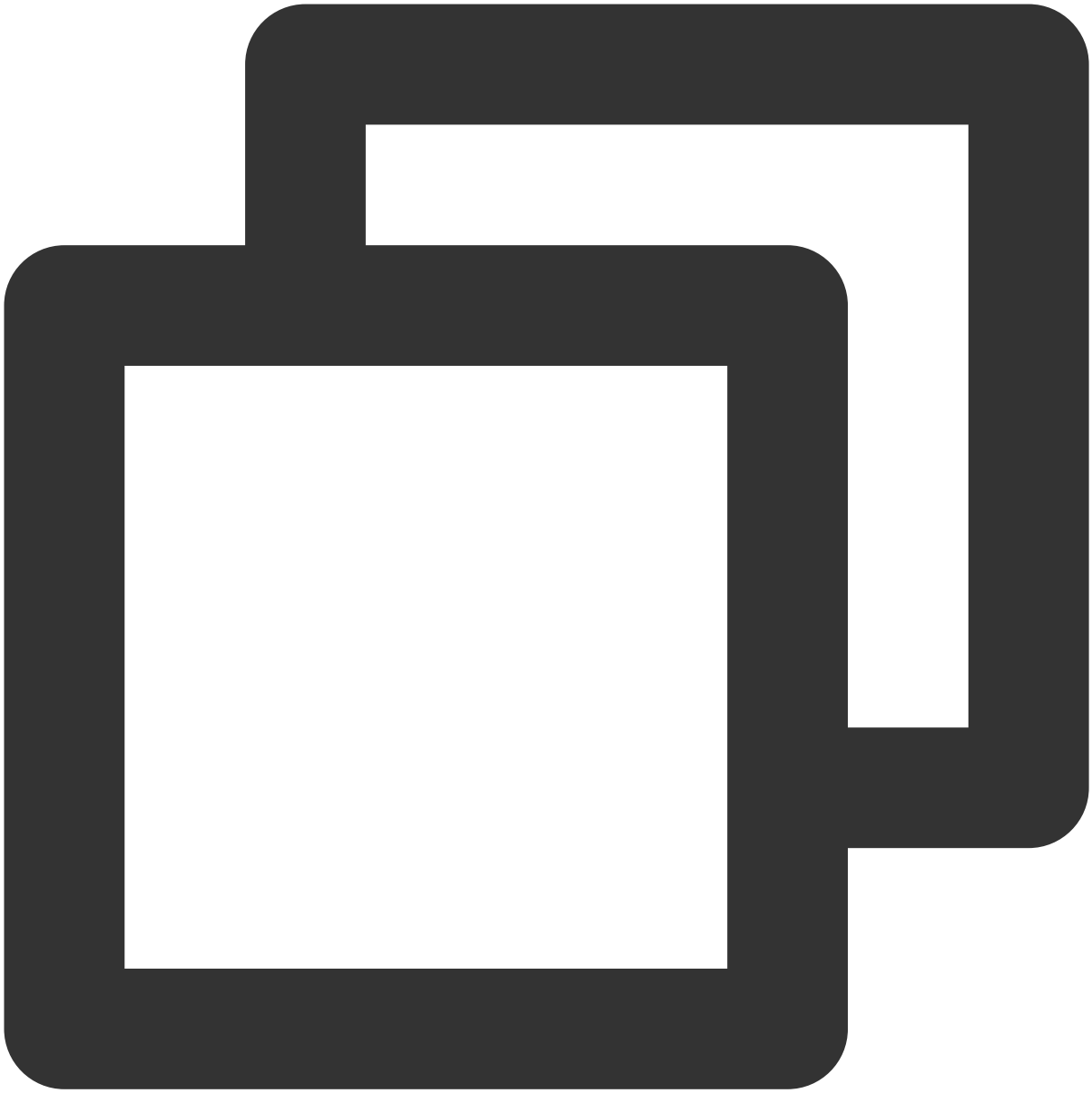
案例九：使用 key-marker 指定起始对象（续接案例七，但仅使用 key-marker 请求参数而不使用 version-id-marker 请求参数）

请求



```
GET /?versions&max-keys=3&key-marker=example-object-2.jpg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 08 Dec 2020 12:46:41 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

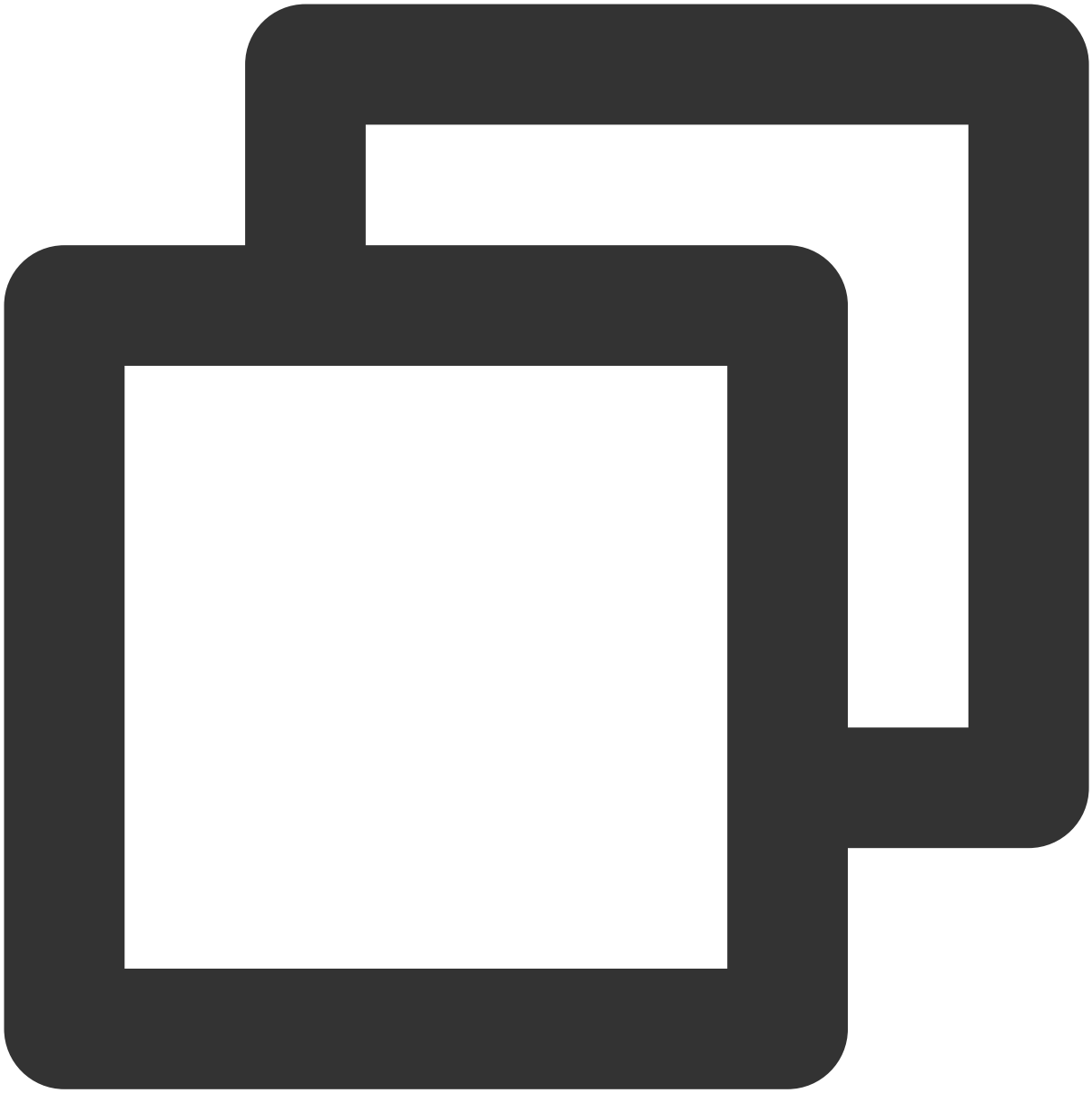


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 610
Connection: close
Date: Tue, 08 Dec 2020 12:46:42 GMT
Server: tencent-cos
x-cos-request-id: NWZjZjc1YjJfZGZjMTBiMDlfNzAyYl8xMzkx****



<ListVersionsResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <KeyMarker>example-object-2.jpg</KeyMarker>
            <VersionIdMarker/>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>false</IsTruncated>
            <Version>
                <Key>example-object-3.jpg</Key>
                <VersionId>MTg0NDUxMzY2NDIxMTk1NTEyNDI</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-08T12:46:30.000Z</LastModified>
                <ETag>&amp;quot;b2f1d893c5fde000ee8ea6eca18ed81f&amp;quot;</ETag>
                <Size>20</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID>1250000000</ID>
                        <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
</ListVersionsResult>
```

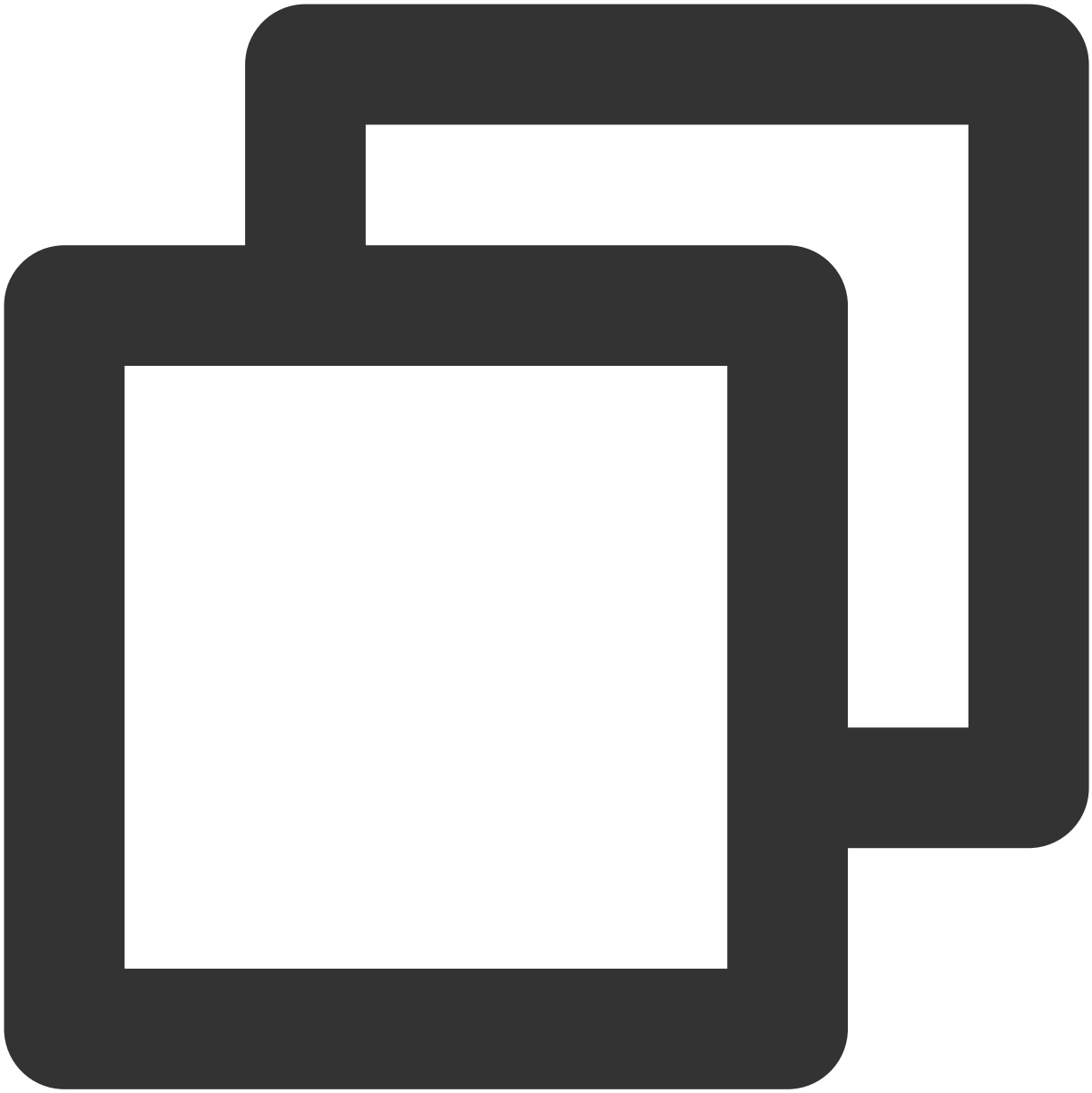
案例十：需分页时获取第一页（带 delimiter 参数，CommonPrefixes、Version 与 DeleteMarker 合计条目数不超过 max-keys 指定的值）

请求



```
GET /?versions&delimiter=%2F&max-keys=3 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 09 Dec 2020 13:40:24 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

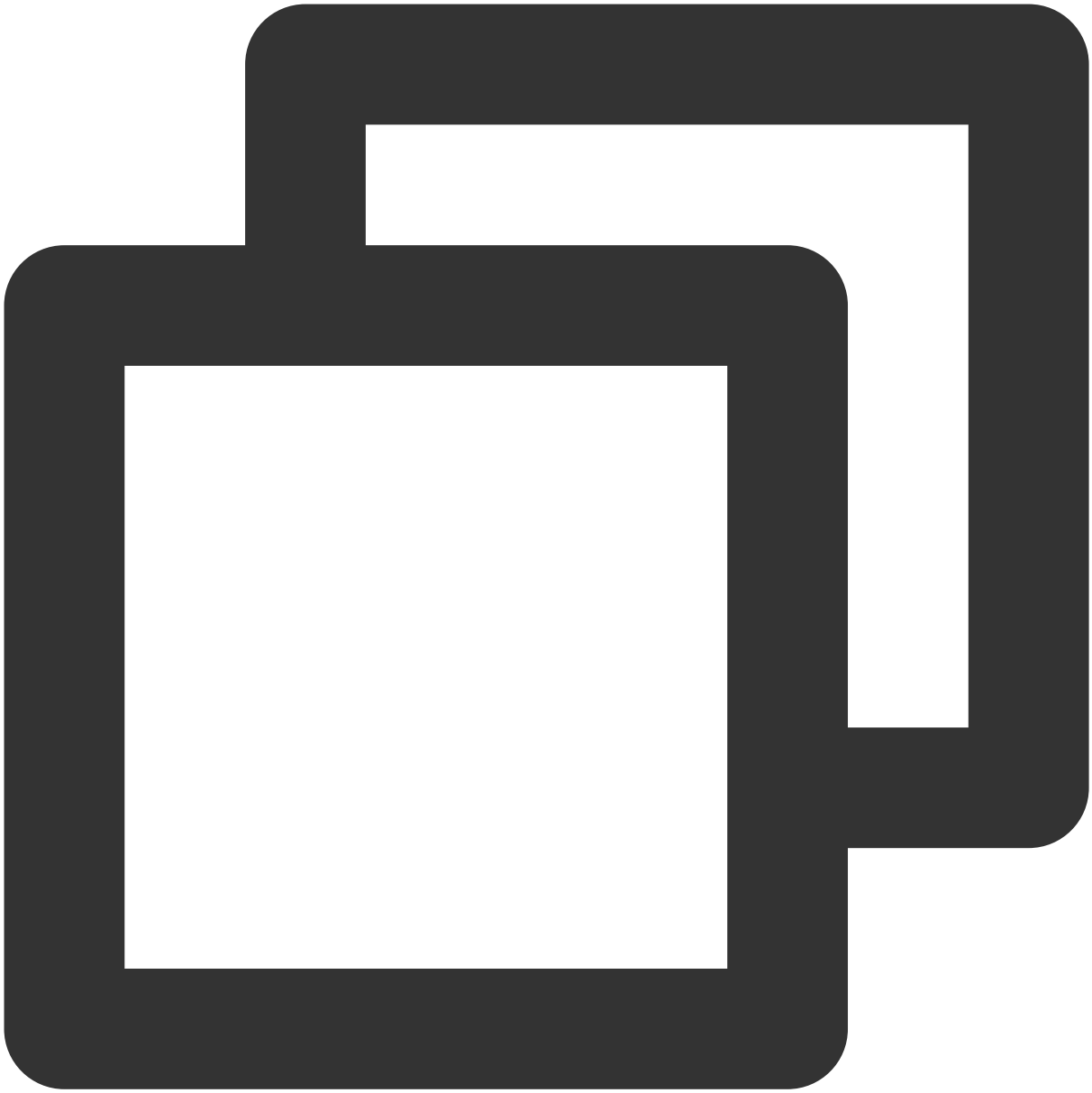


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 503
Connection: close
Date: Wed, 09 Dec 2020 13:40:24 GMT
Server: tencent-cos
x-cos-request-id: NWZkMGQzYzhfYmIwMmEwOV83OGU4XzZj****

<ListVersionsResult>
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
```

```
            <KeyMarker/>
            <VersionIdMarker/>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>true</IsTruncated>
            <NextKeyMarker>example-folder-3/</NextKeyMarker>
            <NextVersionIdMarker/>
            <Delimiter>/</Delimiter>
            <CommonPrefixes>
                <Prefix>example-folder-1/</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>example-folder-2/</Prefix>
            </CommonPrefixes>
            <CommonPrefixes>
                <Prefix>example-folder-3/</Prefix>
            </CommonPrefixes>
</ListVersionsResult>
```

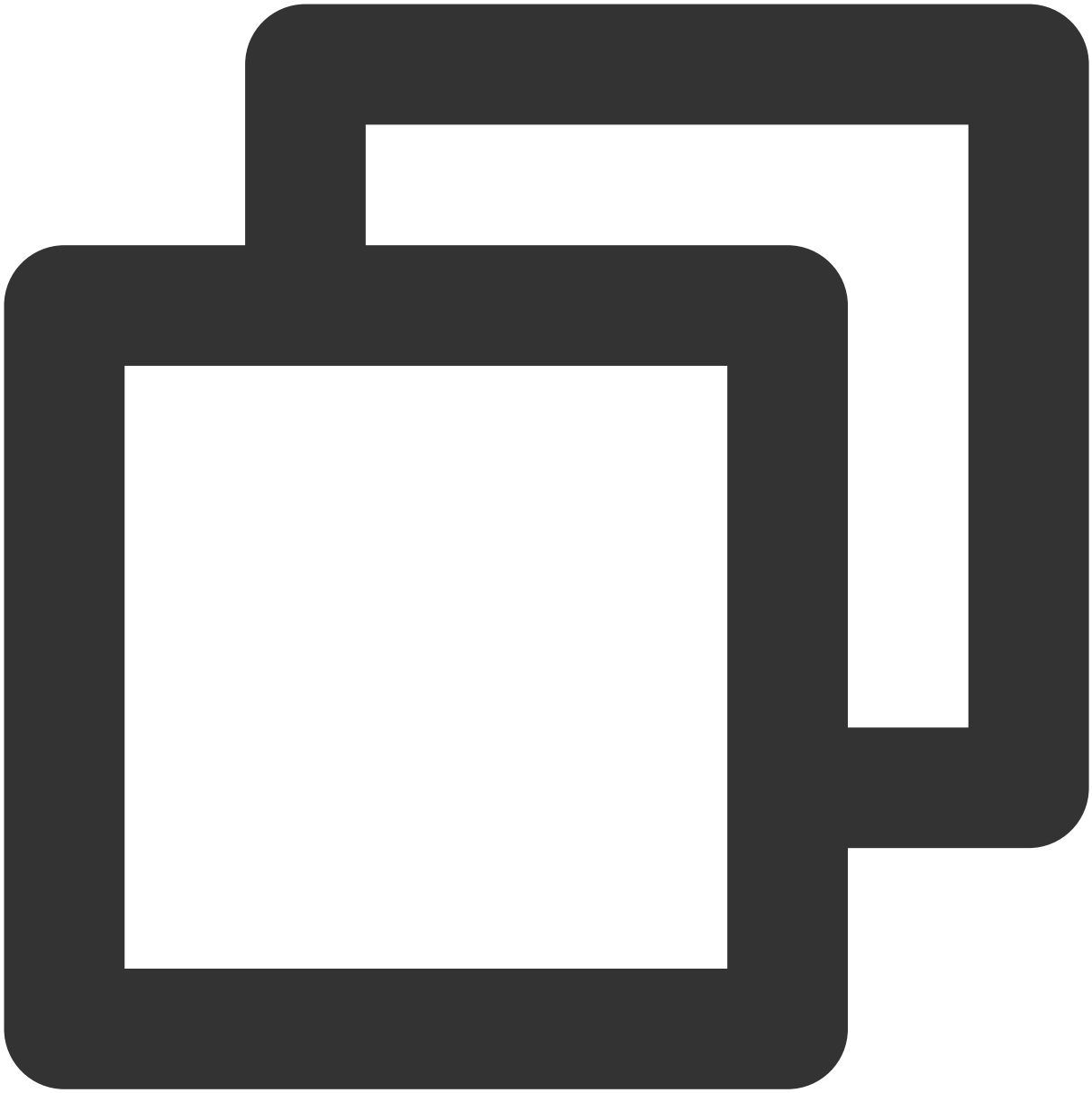
案例十一：需分页时获取后续页（带 delimiter 参数，续接案例十，因为第一页返回的 NextVersionIdMarker 为空，所以本次请求的 version-id-marker 参数也指定为空）

请求



```
GET /?versions&delimiter=%2F&max-keys=3&key-marker=example-folder-3%2F&version-id-m
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 09 Dec 2020 14:02:32 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1115
Connection: close
Date: Wed, 09 Dec 2020 14:02:32 GMT
Server: tencent-cos
x-cos-request-id: NWZkMGQ4ZjhfZmVhODBiMDlfMTI3NzVfMTBj****



<ListVersionsResult>
```

```
            <Name>examplebucket-1250000000</Name>
            <Prefix/>
            <KeyMarker>example-folder-3/</KeyMarker>
            <VersionIdMarker/>
            <MaxKeys>3</MaxKeys>
            <IsTruncated>true</IsTruncated>
            <NextKeyMarker>example-object.jpg</NextKeyMarker>
            <NextVersionIdMarker>MTg0NDUxMzY1NTExNzc4NTYzMTk</NextVersionIdMarker>
            <Delimiter>/</Delimiter>
            <CommonPrefixes>
                <Prefix>example-folder-4/</Prefix>
            </CommonPrefixes>
            <Version>
                <Key>example-object.jpg</Key>
                <VersionId>MTg0NDUxMzY1NTExNjc2OTk4MjE</VersionId>
                <IsLatest>true</IsLatest>
                <LastModified>2020-12-09T14:02:21.000Z</LastModified>
                <ETag>&amp;quot;3979f6bb23f827d258be64cc0d8df5fb&amp;quot;</ETag>
                <Size>21</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </Version>
            <DeleteMarker>
                <Key>example-object.jpg</Key>
                <VersionId>MTg0NDUxMzY1NTExNzc4NTYzMTk</VersionId>
                <IsLatest>false</IsLatest>
                <LastModified>2020-12-09T14:02:11.000Z</LastModified>
                <Owner>
                    <ID>1250000000</ID>
                    <DisplayName>1250000000</DisplayName>
                </Owner>
            </DeleteMarker>
</ListVersionsResult>
```

版权所有：腾讯云计算（北京）有限责任公司

访问控制（acl）

PUT Bucket acl

最近更新时间：2023-12-20 18:01:44

功能描述

PUT Bucket acl 接口用来写入存储桶的访问控制列表（ACL），您可以通过请求头 `x-cos-acl` 和 `x-cos-grant-*` 传入 ACL 信息，或者通过请求体以 XML 格式传入 ACL 信息。

注意

通过请求头设置 ACL 和通过请求体设置 ACL 两种方式只能选择其中一种。

PUT Bucket acl 是一个覆盖操作，传入新的 ACL 将覆盖原有 ACL。

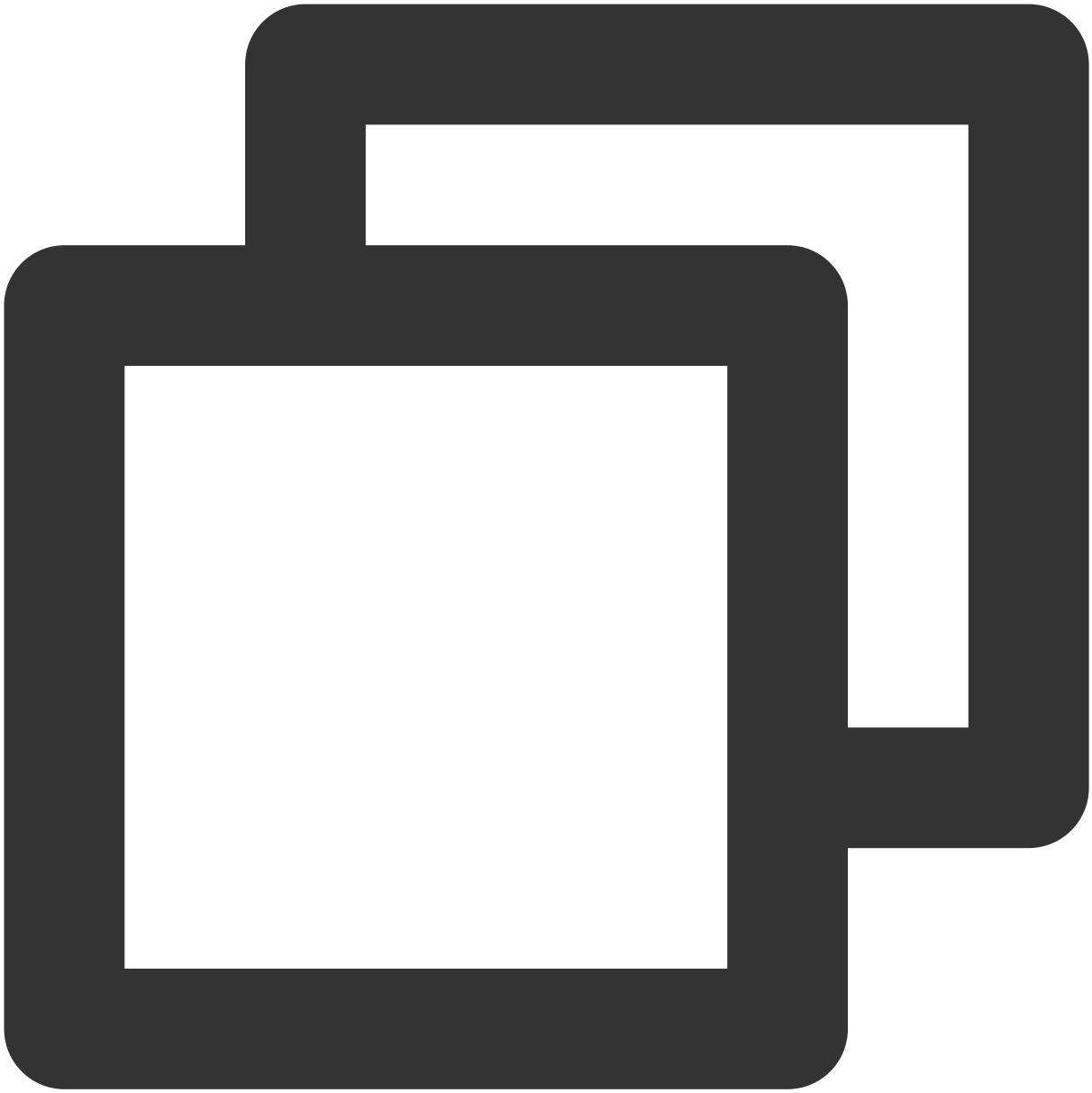
支持对腾讯云 CAM 主账号、匿名用户、子用户授予权限，如需授予用户组权限请参见 为用户组添加/解除策略。有关 ACL 的详细说明，请参见 ACL 概述。

该 API 的请求者需要对存储桶有写入 ACL 权限。

请求

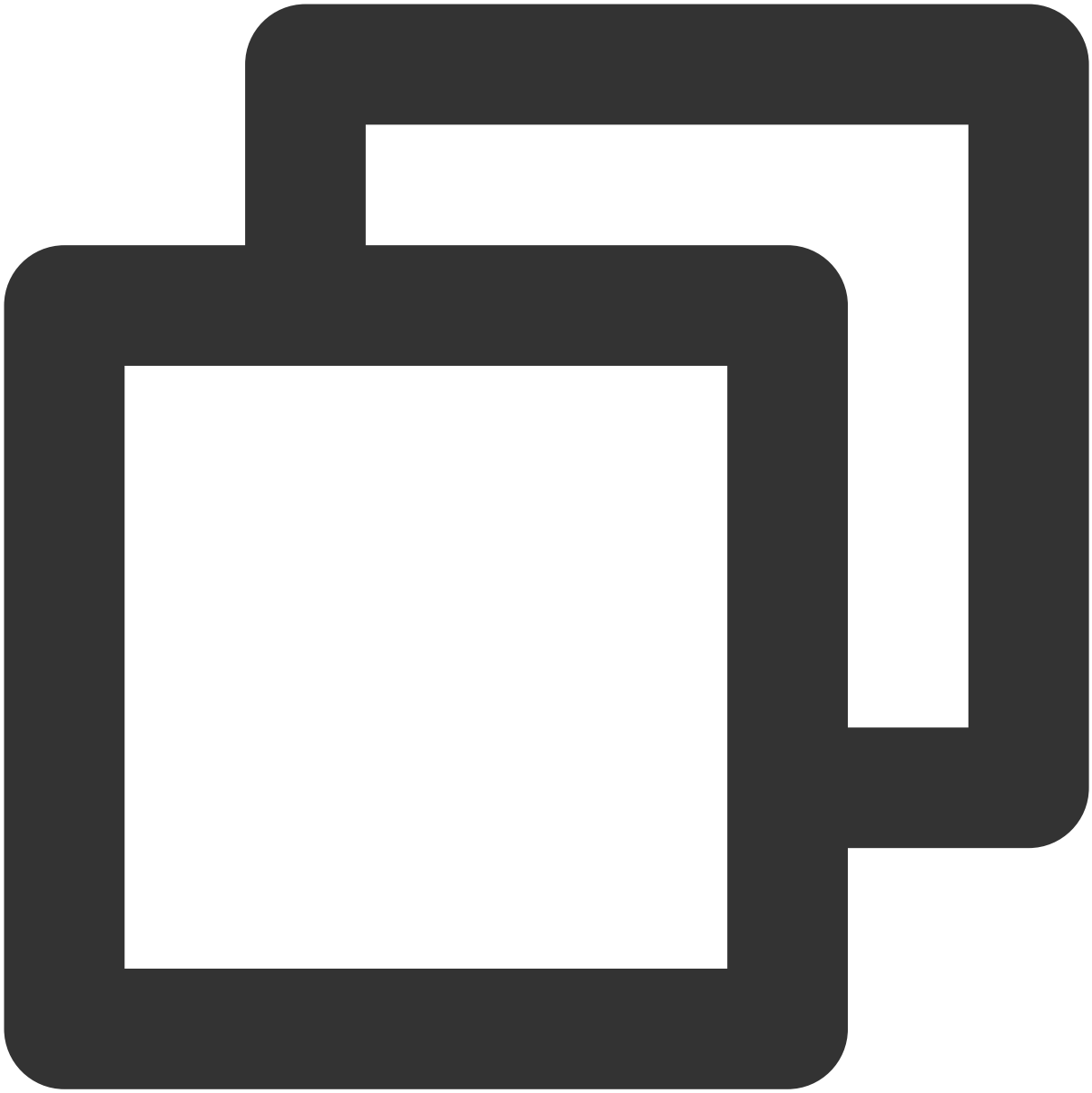
请求示例

示例一



```
PUT /?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: 0
Authorization: Auth String
```

示例二



```
PUT /?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

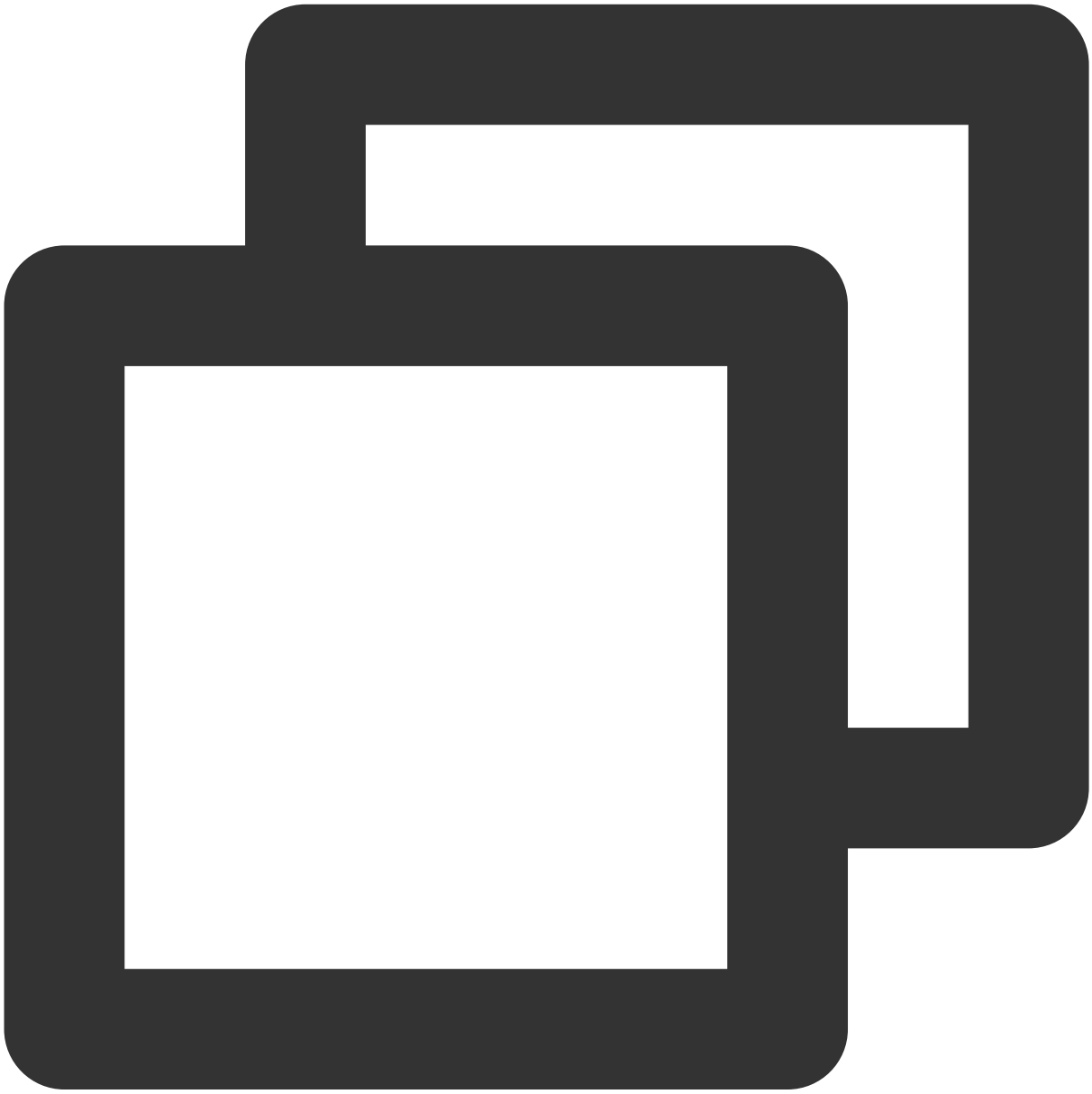
此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
x-cos-acl	定义存储桶的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中存储桶的预设 ACL 部分，例如 private, public-read 等，默认为 private	Enum	否
x-cos-grant-read	赋予被授权者读取存储桶的权限。若被授权者为主账号，格式为 id="[OwnerUin]"，如 id="100000000001"；若被授权者为子账号，格式为 id="[OwnerUin/GrantsUin]"，如 id="100000000001/100000000011"。可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write	赋予被授权者写入存储桶的权限。若被授权者为主账号，格式为 id="[OwnerUin]"，如 id="100000000001"；若被授权者为子账号，格式为 id="[OwnerUin/GrantsUin]"，如 id="100000000001/100000000011"。可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取存储桶的访问控制列表（ACL）的权限。若被授权者为主账号，格式为 id="[OwnerUin]"，如 id="100000000001"；若被授权者为子账号，格式为 id="[OwnerUin/GrantsUin]"，如 id="100000000001/100000000011"。可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入存储桶的访问控制列表（ACL）的权限。若被授权者为主账号，格式为 id="[OwnerUin]"，如 id="100000000001"；若被授权者为子账号，格式为 id="[OwnerUin/GrantsUin]"，如 id="100000000001/100000000011"。可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-full-control	赋予被授权者操作存储桶的所有权限。若被授权者为主账号，格式为 id="[OwnerUin]"，如 id="100000000001"；若被授权者为子账号，格式为 id="[OwnerUin/GrantsUin]"，如	string	否

	id="100000000001/100000000011"。可使用半角逗号（,）分隔多组被授权者，如 `id="100000000001",id="100000000002"`		

请求体

提交 **application/xml** 请求数据，包含存储桶所有者和完整的授权信息。



```
<AccessControlPolicy>
    <Owner>
        <ID>string</ID>
    </Owner>
```

```
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <URI>string</URI>
            </Grantee>
            <Permission>Enum</Permission>
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>string</ID>
            </Grantee>
            <Permission>Enum</Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
AccessControlPolicy	无	包含 PUT Bucket acl 操作的所有请求信息	Container	是

Container 节点 AccessControlPolicy 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Owner	AccessControlPolicy	存储桶持有者信息	Container	是
AccessControlList	AccessControlPolicy	被授权者信息与权限信息	Container	是

Container 节点 Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	AccessControlPolicy.Owner	存储桶持有者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:uin/[OwnerUin]`，如 `qcs::cam::uin/100000000001:uin/100000000001`	string

Container 节点 AccessControlList 的内容：

节点名称（关键	父节点	描述	类型	是否必选

版权所有：腾讯云计算（北京）有限责任公司

字）				
Grant	AccessControlPolicy.AccessControlList	单个授权信息，一个 AccessControlList 最多只能拥有100条 Grant	Container	是

Container 节点 AccessControlList.Grant 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Grantee	AccessControlPolicy.AccessControlList.Grant	被授权者信息，`xsi:type` 可指定为 Group 或 CanonicalUser，当指定为 Group 时子节点包括且仅允许包括 URI，当指定为 CanonicalUser 时子节点包括且仅允许包括 ID	Container	是
Permission	AccessControlPolicy.AccessControlList.Grant	授予的权限信息。枚举值请参见 ACL 概述 文档中存储桶的操作部分，如 WRITE，FULL_CONTROL 等	Enum	是

Container 节点 AccessControlList.Grant.Grantee 的内容：

节点名称（关键字）	父节点	描述
URI	AccessControlPolicy.AccessControlList.Grant.Grantee	预设用户组。请参见 ACL 概述 文档中预设用户 `http://cam.qcloud.com/groups/glob` `http://cam.qcloud.com/groups/glob`
ID	AccessControlPolicy.AccessControlList.Grant.Grantee	被授权者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:uin/[Own` `qcs::cam::uin/100000000001:uin/10`

版权所有：腾讯云计算（北京）有限责任公司

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

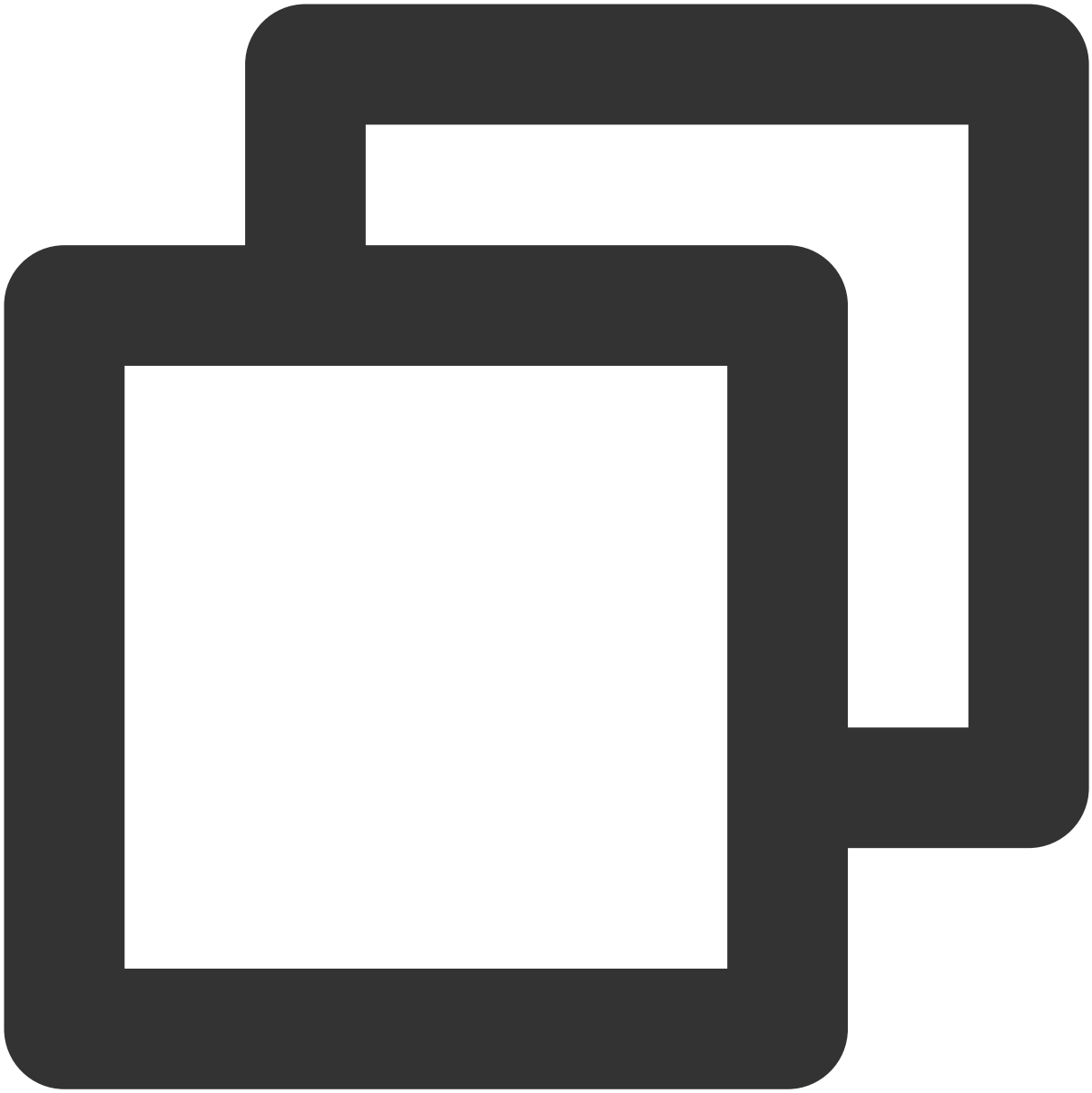
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

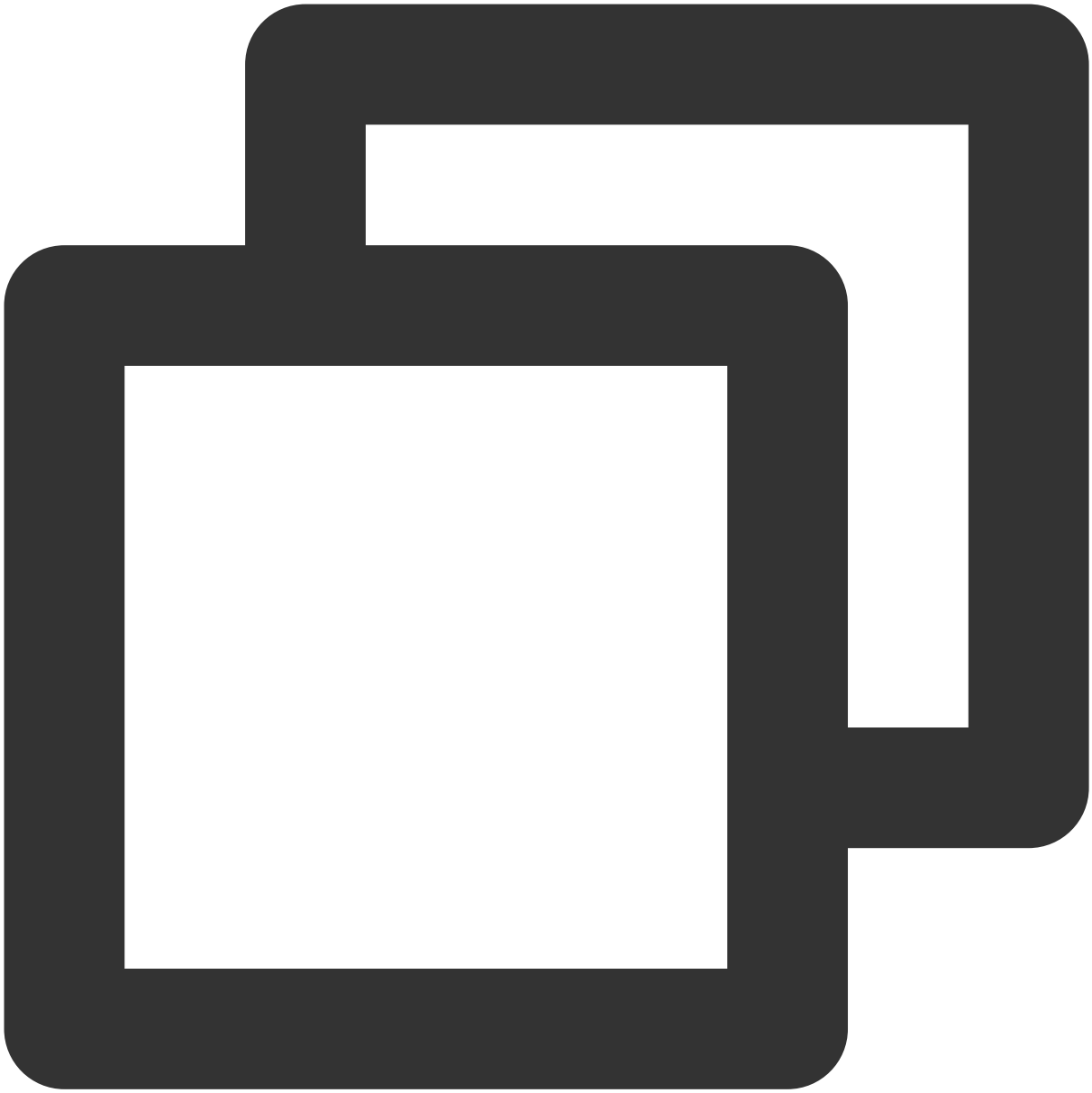
案例一：通过请求头设置 ACL

请求



```
PUT /?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:30:12 GMT
x-cos-acl: public-read
x-cos-grant-write: id="100000000002"
x-cos-grant-read-acp: id="100000000002"
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

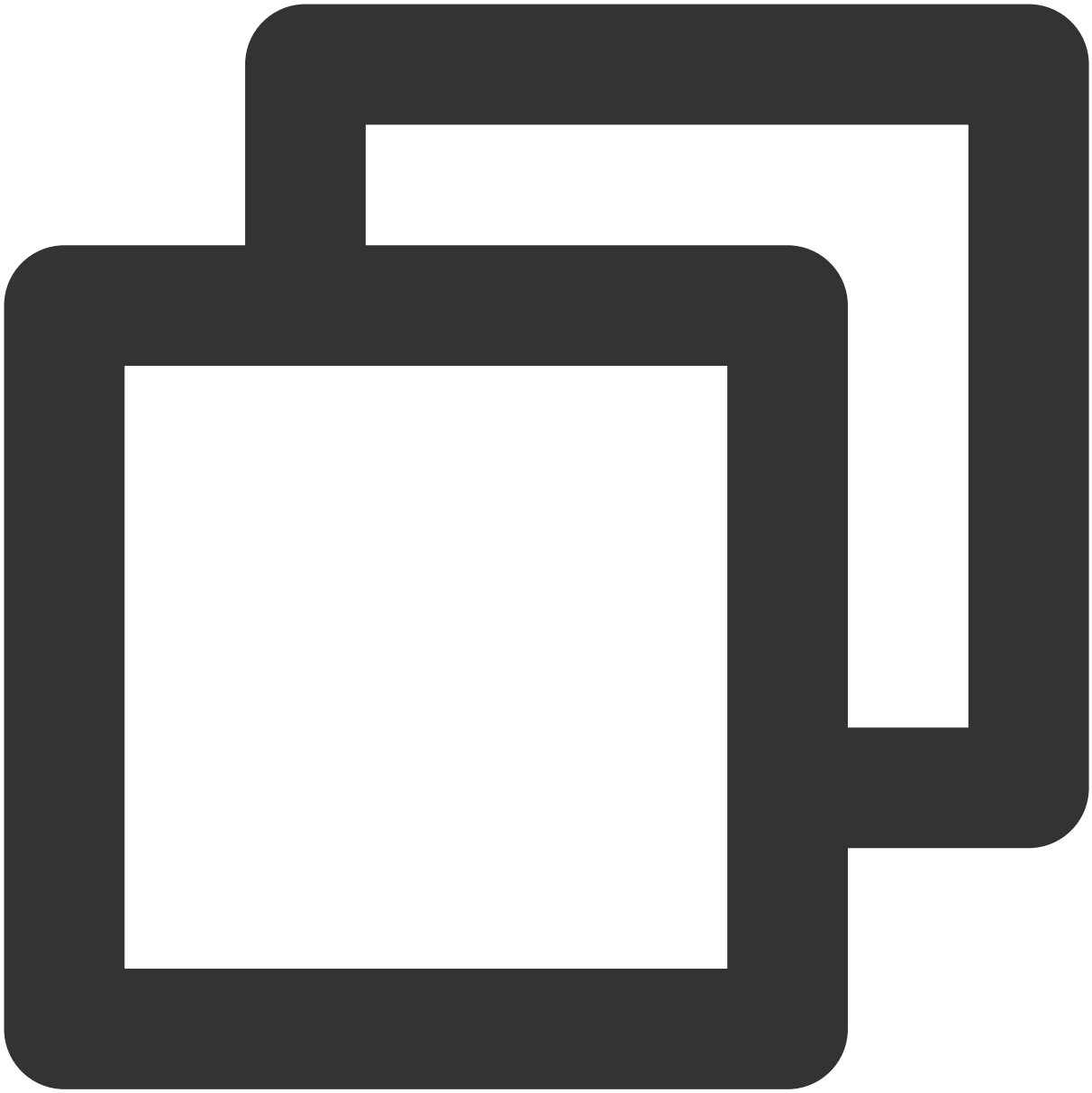
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Mon, 17 Jun 2019 08:30:13 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzRmOTRfODhjMjJhMDlfMWRlYl81Mzc0****
```

案例二：通过请求体设置 ACL

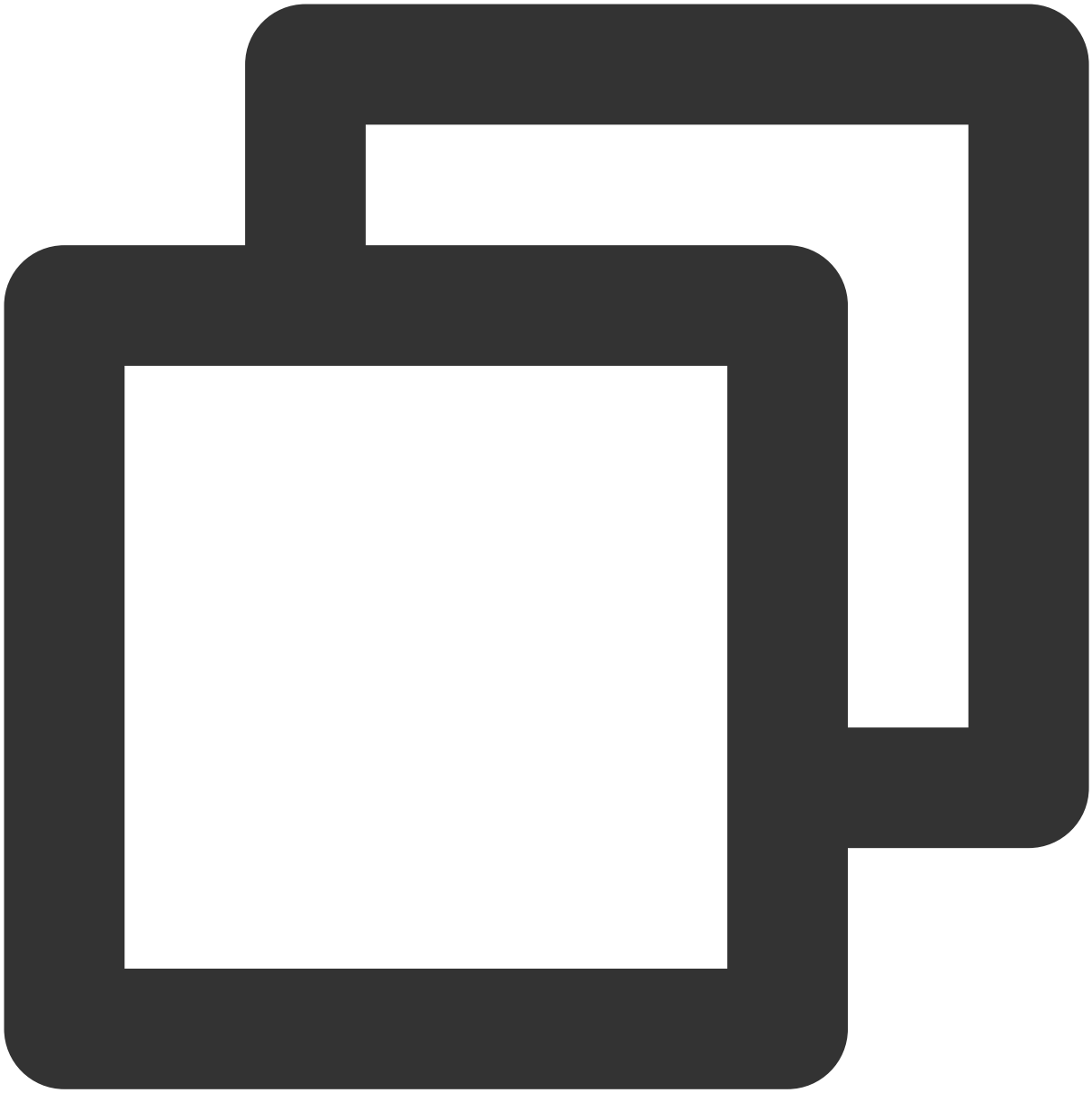
请求



```
PUT /?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:30:13 GMT
Content-Type: application/xml
Content-Length: 812
Content-MD5: 1qS+8SqnivarcO6Z11R0nw==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

```
<AccessControlPolicy>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
    </Owner>
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
            </Grantee>
            <Permission>READ</Permission>
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
            </Grantee>
            <Permission>WRITE</Permission>
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
            </Grantee>
            <Permission>READ_ACP</Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Mon, 17 Jun 2019 08:30:13 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzRmOTVfMzBjMDJhMDlfOTM3MF8yNzdj****
```

GET Bucket acl

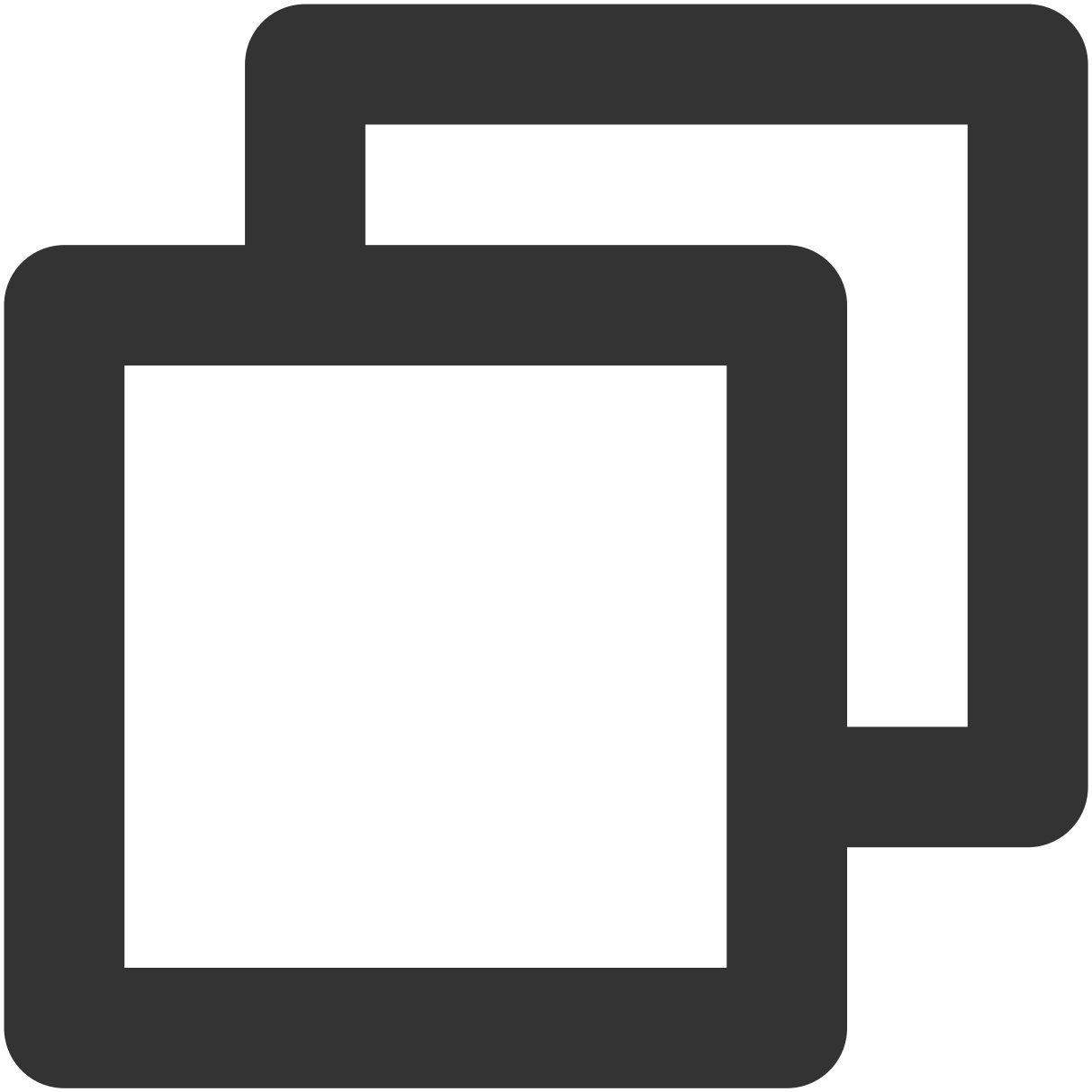
最近更新时间：2023-12-20 18:16:44

功能描述

GET Bucket acl 接口用来获取存储桶的访问控制列表（ACL）。该 API 的请求者需要对存储桶有读取 ACL 权限。

请求

请求示例



```
GET /?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

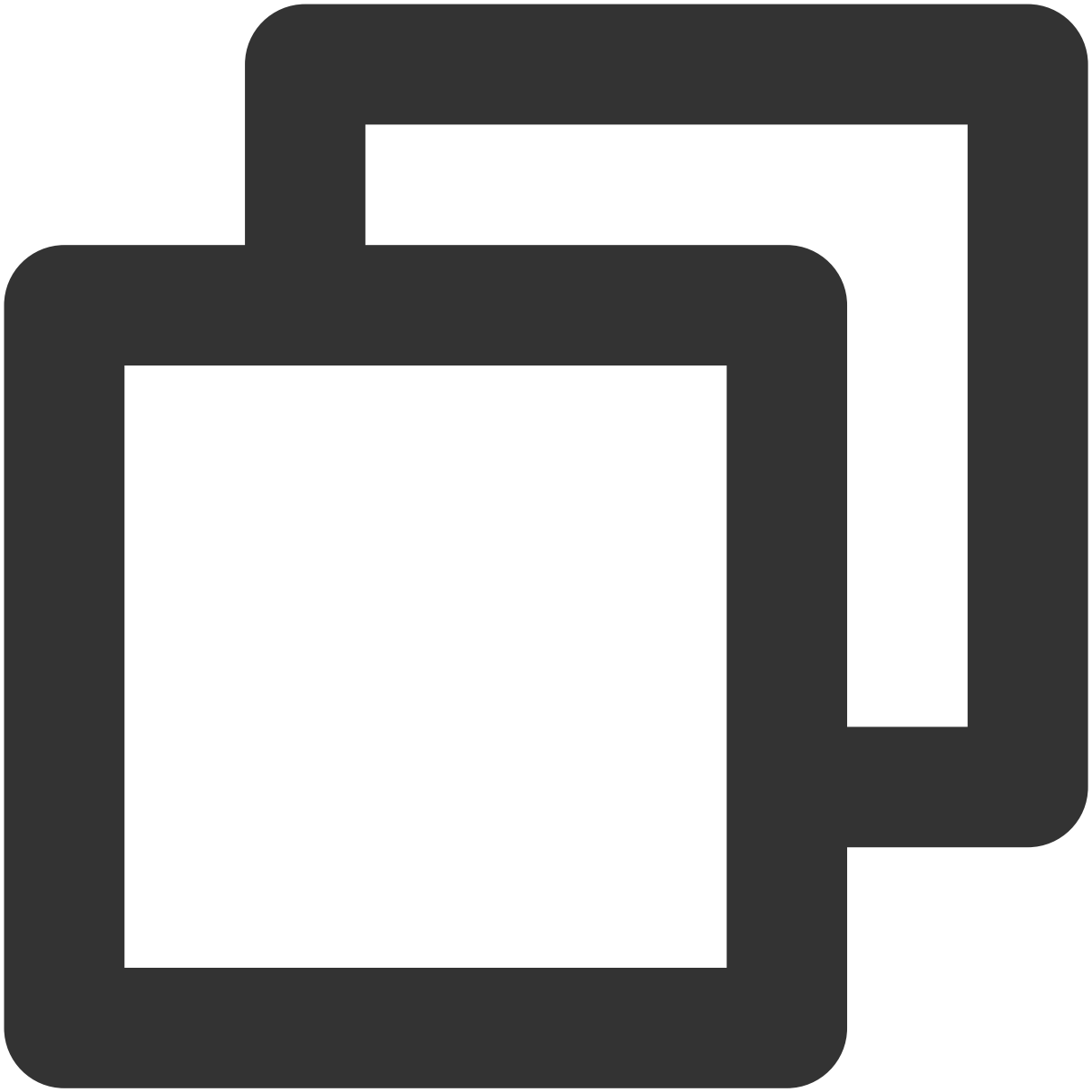
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含存储桶所有者和完整的授权信息。



```
<AccessControlPolicy>
    <Owner>
        <ID>string</ID>
        <DisplayName>string</DisplayName>
    </Owner>
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <URI>string</URI>
            </Grantee>
            <Permission>Enum</Permission>
```

```
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>string</ID>
                <DisplayName>string</DisplayName>
            </Grantee>
            <Permission>Enum</Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
AccessControlPolicy	无	保存 GET Bucket acl 结果的所有信息	Container

Container 节点 AccessControlPolicy 的内容：

节点名称（关键字）	父节点	描述	类型
Owner	AccessControlPolicy	存储桶持有者信息	Container
AccessControlList	AccessControlPolicy	被授权者信息与权限信息	Container

Container 节点 Owner 的内容：

节点名称（关键字）	父节点	描述
ID	AccessControlPolicy.Owner	存储桶持有者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:uin/[OwnerUin]`，如 `qcs::cam::uin/100000000001:uin/100000000001`
DisplayName	AccessControlPolicy.Owner	存储桶持有者的名字

Container 节点 AccessControlList 的内容：

节点名称（关键字）	父节点	描述	类型
Grant	AccessControlPolicy.AccessControlList	单个授权信息	Container

Container 节点 AccessControlList.Grant 的内容：

节点名称	父节点	描述	类型

（关键字）			
Grantee	AccessControlPolicy.AccessControlList.Grant	被授权者信息，`xsi:type` 为 Group 或 CanonicalUser，当为 Group 时子节点包括且仅包括 URI，当指定为 CanonicalUser 时子节点包括且仅包括 ID 和 DisplayName	Container
Permission	AccessControlPolicy.AccessControlList.Grant	授予的权限信息，枚举值请参见 ACL 概述 文档中存储桶的操作部分，如 WRITE，FULL_CONTROL 等	Enum

Container 节点 AccessControlList.Grant.Grantee 的内容：

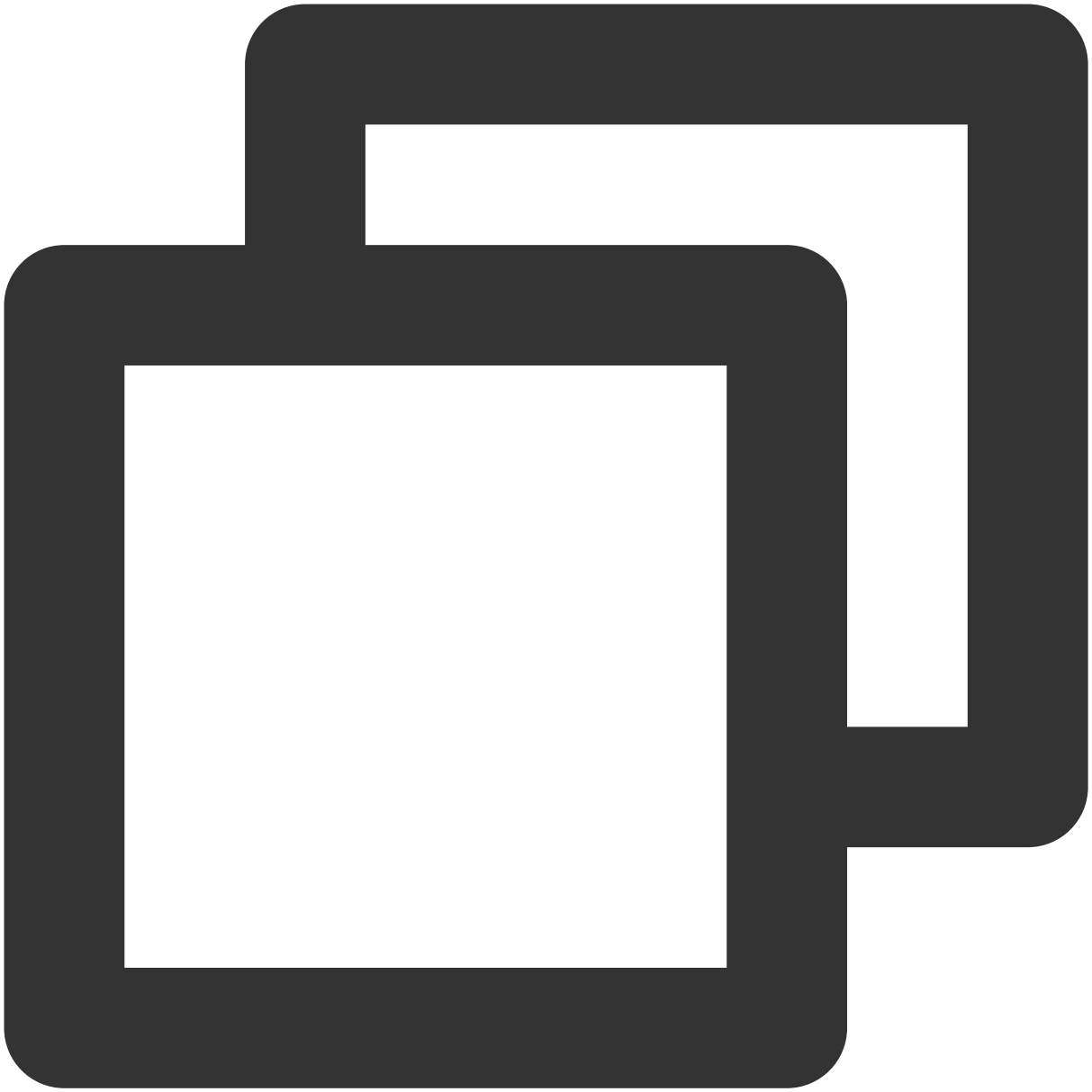
节点名称（关键字）	父节点	描述
URI	AccessControlPolicy.AccessControlList.Grant.Grantee	预设用户组，请参见 ACL 概述 文档中 `http://cam.qcloud.com/group` `http://cam.qcloud.com/group`
ID	AccessControlPolicy.AccessControlList.Grant.Grantee	被授权者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:ui` `qcs::cam::uin/100000000001:`
DisplayName	AccessControlPolicy.AccessControlList.Grant.Grantee	被授权者的名字

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

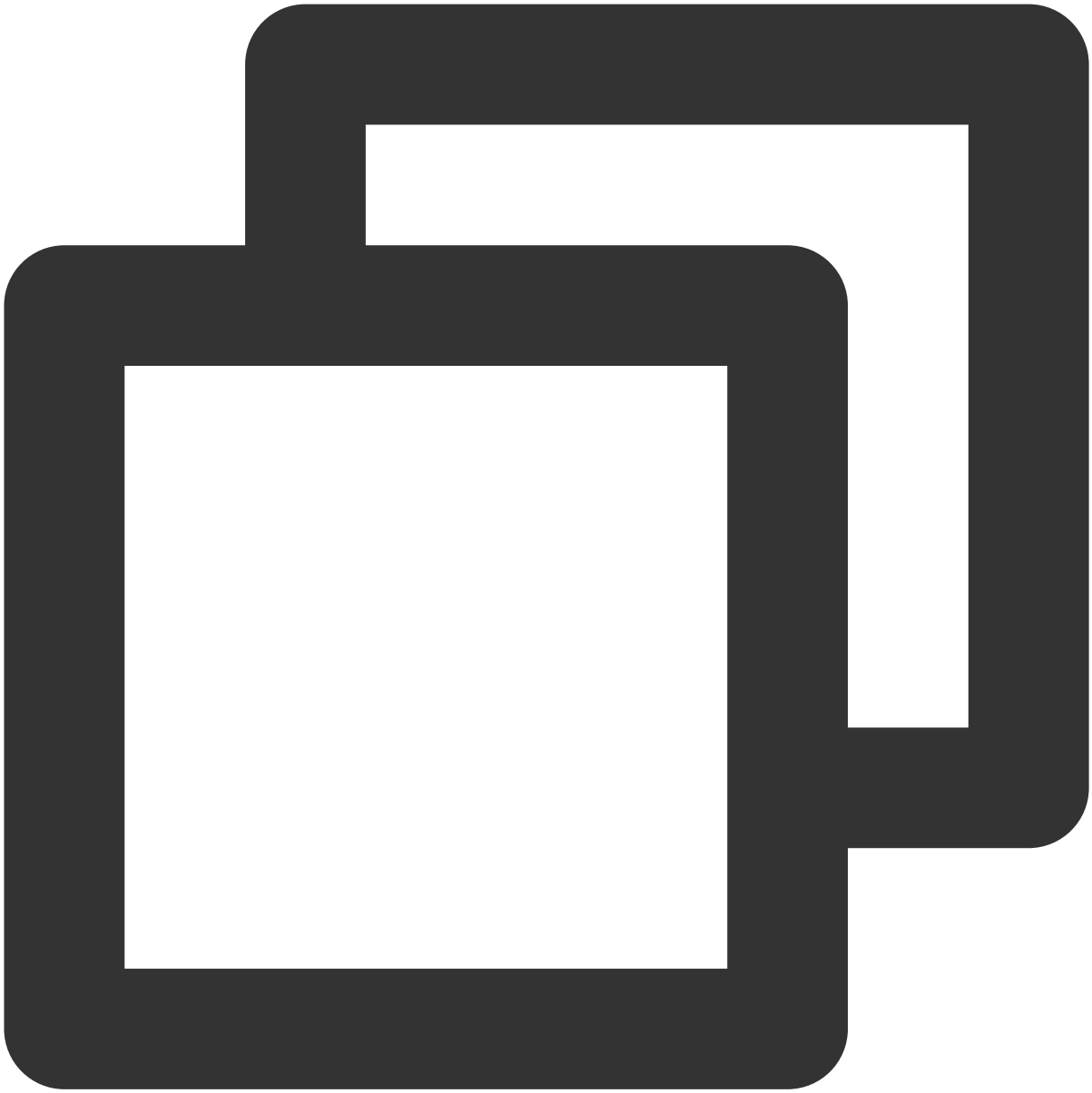
实际案例

请求



```
GET /?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1035
Connection: close
Date: Mon, 17 Jun 2019 08:37:36 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****

<AccessControlPolicy>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
    </Owner>
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
            </Grantee>
            <Permission>READ</Permission>
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
                <DisplayName>qcs::cam::uin/100000000002:uin/100000000002</DisplayNa
            </Grantee>
            <Permission>WRITE</Permission>
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
                <DisplayName>qcs::cam::uin/100000000002:uin/100000000002</DisplayNa
            </Grantee>
            <Permission>READ_ACP</Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
```

版权所有：腾讯云计算（北京）有限责任公司

跨域资源共享（cors）

PUT Bucket cors

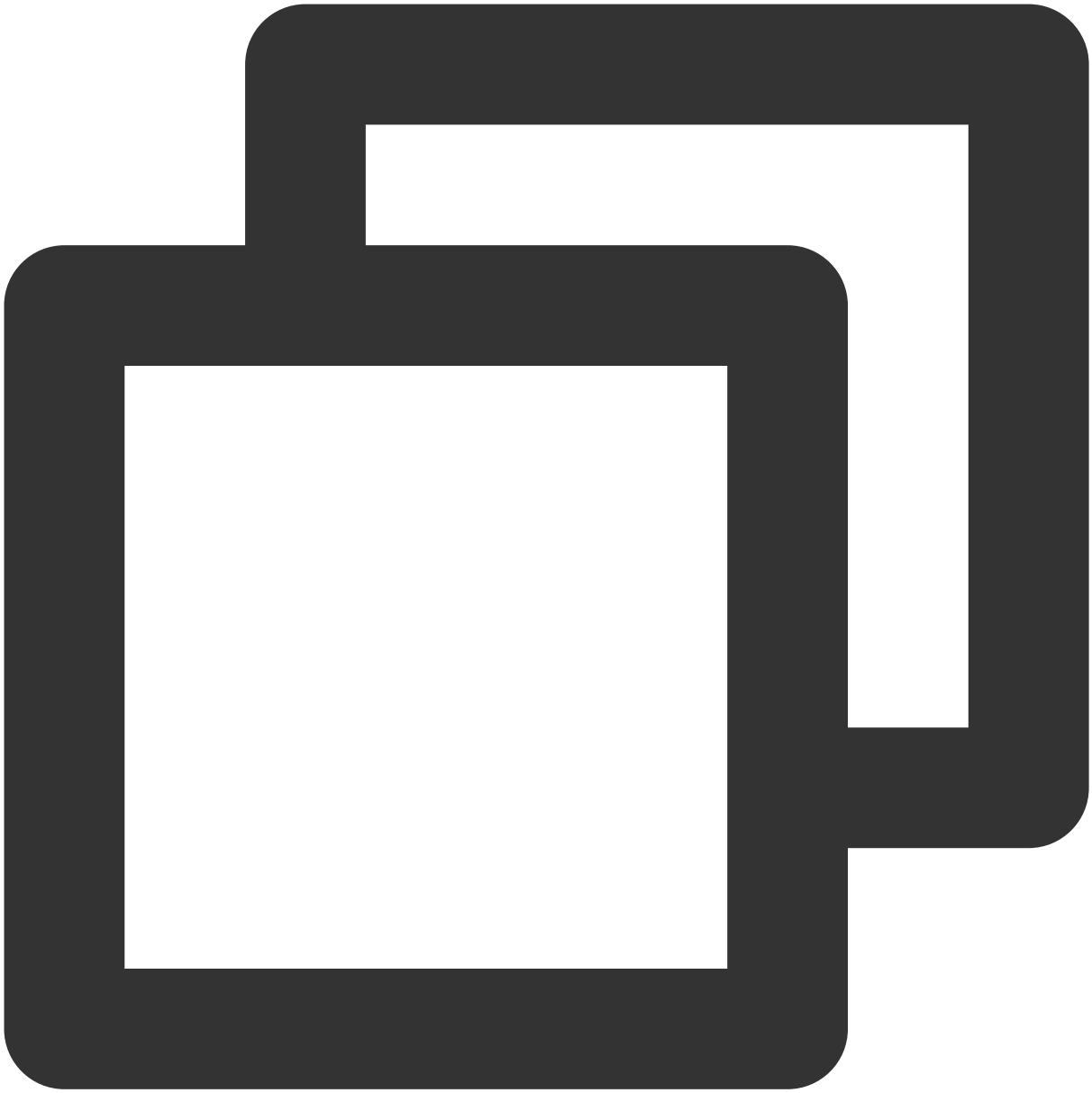
最近更新时间：2023-12-21 11:33:19

功能描述

PUT Bucket cors 请求用于为存储桶配置跨域资源共享（CORS）访问控制，您可以通过传入 XML 格式的配置文件来实现配置，文件大小限制为64KB。

请求

请求示例



```
PUT /?cors HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

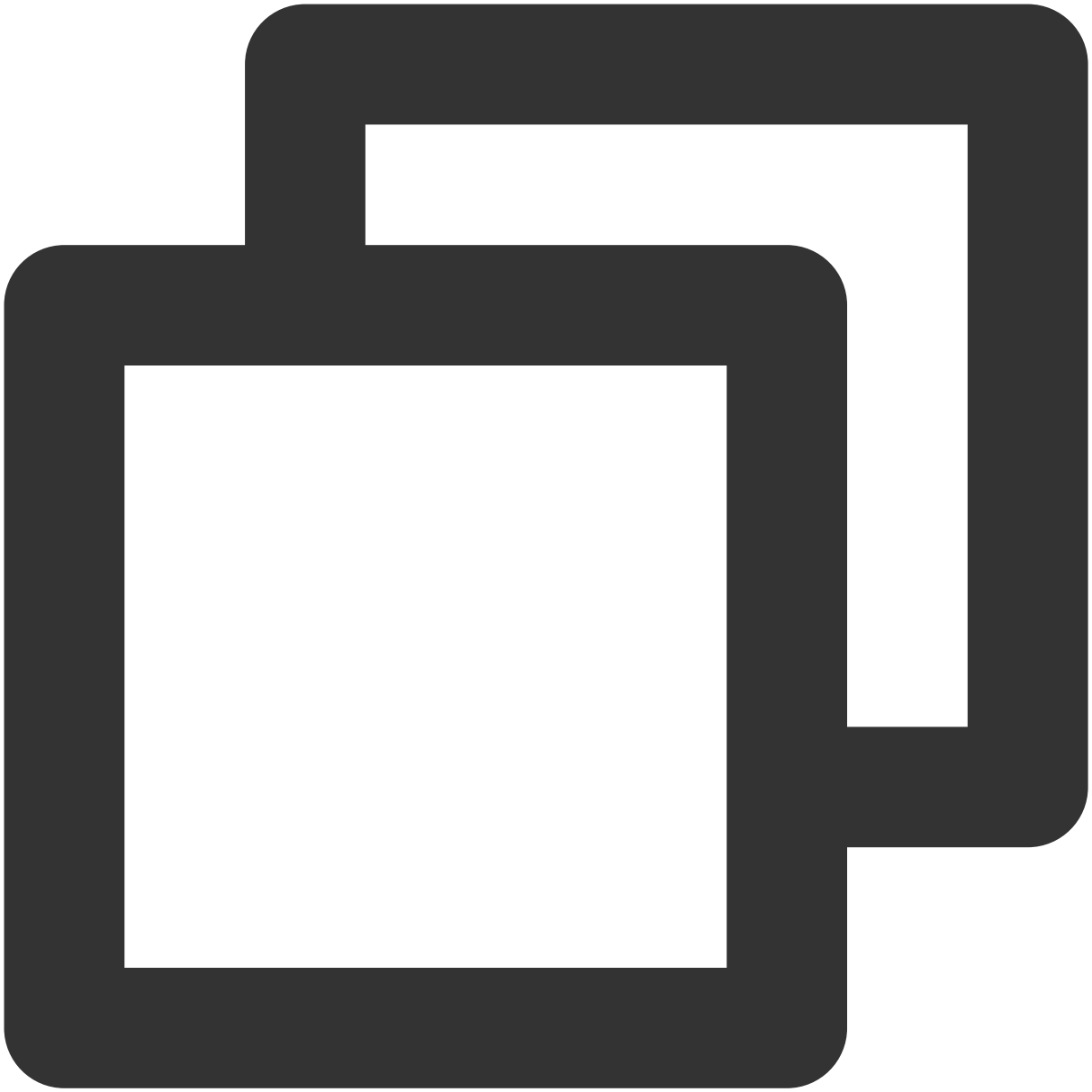
此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

提交 **application/xml** 请求数据，包含完整的存储桶跨域资源共享（CORS）配置信息。



```
<CORSConfiguration>
    <CORSRule>
        <AllowedOrigin>string</AllowedOrigin>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedHeader>string</AllowedHeader>
        <AllowedHeader>string</AllowedHeader>
        <ExposeHeader>string</ExposeHeader>
        <ExposeHeader>string</ExposeHeader>
        <MaxAgeSeconds>integer</MaxAgeSeconds>
    </CORSRule>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <CORSRule>
        <ID>string</ID>
        <AllowedOrigin>string</AllowedOrigin>
        <AllowedOrigin>string</AllowedOrigin>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedHeader>string</AllowedHeader>
        <ExposeHeader>string</ExposeHeader>
        <ExposeHeader>string</ExposeHeader>
        <MaxAgeSeconds>integer</MaxAgeSeconds>
    </CORSRule>
</CORSConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
CORSConfiguration	无	包含 PUT Bucket cors 操作的所有请求信息	Container	否

Container 节点 CORSConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
CORSRule	CORSConfiguration	说明单条跨域资源共享（CORS）配置的所有信息，最多可以包含100条 CORSRule	Container	是

Container 节点 CORSRule 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
AllowedOrigin	CORSConfiguration.CORSRule	允许的访问来源，单条 CORSRule 可以配置多个 AllowedOrigin。 配置支持 `*`，表示全部域名都允许，但不推荐。 支持单个具体域名，例如 `http://www.example.com`。 支持 `*` 通配符，通配符可出现在任何位置，包括协议、域名和端口，可匹配0个或多个字符，但是只能有一个 `*`。请谨慎使用通配符，因为可能意外匹配到非预期的来源	string	是

		注意不要遗漏协议名 http 或 https，若端口不是默认的80(http)或443(https)，还需要带上端口，例如 `https://example.com:8443` 。		
AllowedMethod	CORSConfiguration.CORSRule	允许的 HTTP 操作方法（Method），对应 CORS 请求响应中的 Access-Control-Allow-Methods 头部，单条 CORSRule 可以配置多个 AllowedMethod。枚举值：PUT、GET、POST、DELETE、HEAD。	enum	是
AllowedHeader	CORSConfiguration.CORSRule	在发送预检（OPTIONS）请求时，浏览器会告知服务端接下来的正式请求将使用的自定义 HTTP 请求头部，此配置用于指定允许浏览器发送 CORS 请求时携带的自定义 HTTP 请求头部，不区分英文大小写，单条 CORSRule 可以配置多个 AllowedHeader。 可以配置 `*` ，代表允许所有头部，为了避免遗漏，推荐配置为 `*` 。 如果不配置为 `*` ，那么在预检（OPTIONS）请求中 Access-Control-Request-Headers 头部出现的每个 Header，都必须在 AllowedHeader 中有对应项。	string	是
ExposeHeader	CORSConfiguration.CORSRule	允许浏览器获取的 CORS 请求响应中的头部，不区分大小写，单条 CORSRule 可以配置多个 ExposeHeader。 默认情况下浏览器只能访问简单响应头部：Cache-Control、Content-Type、Expires、Last-Modified，如果需要访问其他响应头部，需要添加 ExposeHeader 配置。 不支持配置为 `*` ，必须明确配置具体的 Header。 根据浏览器的实际需求确定，默认推荐填写 ETag，可参考各 API 文档的响应头部分及 公共响应头部 文档。	string	是
MaxAgeSeconds	CORSConfiguration.CORSRule	跨域资源共享配置的有效时间，单位为秒，在有效时间内，浏览器无须为同一请求再次发起预检（OPTIONS）	integer	是

版权所有：腾讯云计算（北京）有限责任公司

		请求，对应 CORS 请求响应中的 Access-Control-Max-Age 头部，单条 CORSRule 只能配置一个 MaxAgeSeconds。		
ID	CORSConfiguration.CORSRule	单条 CORSRule 配置的 ID，用于在 GET Bucket cors 时查找指定 CORSRule。可选填，单条 CORSRule 最多配置一个 ID。	string	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

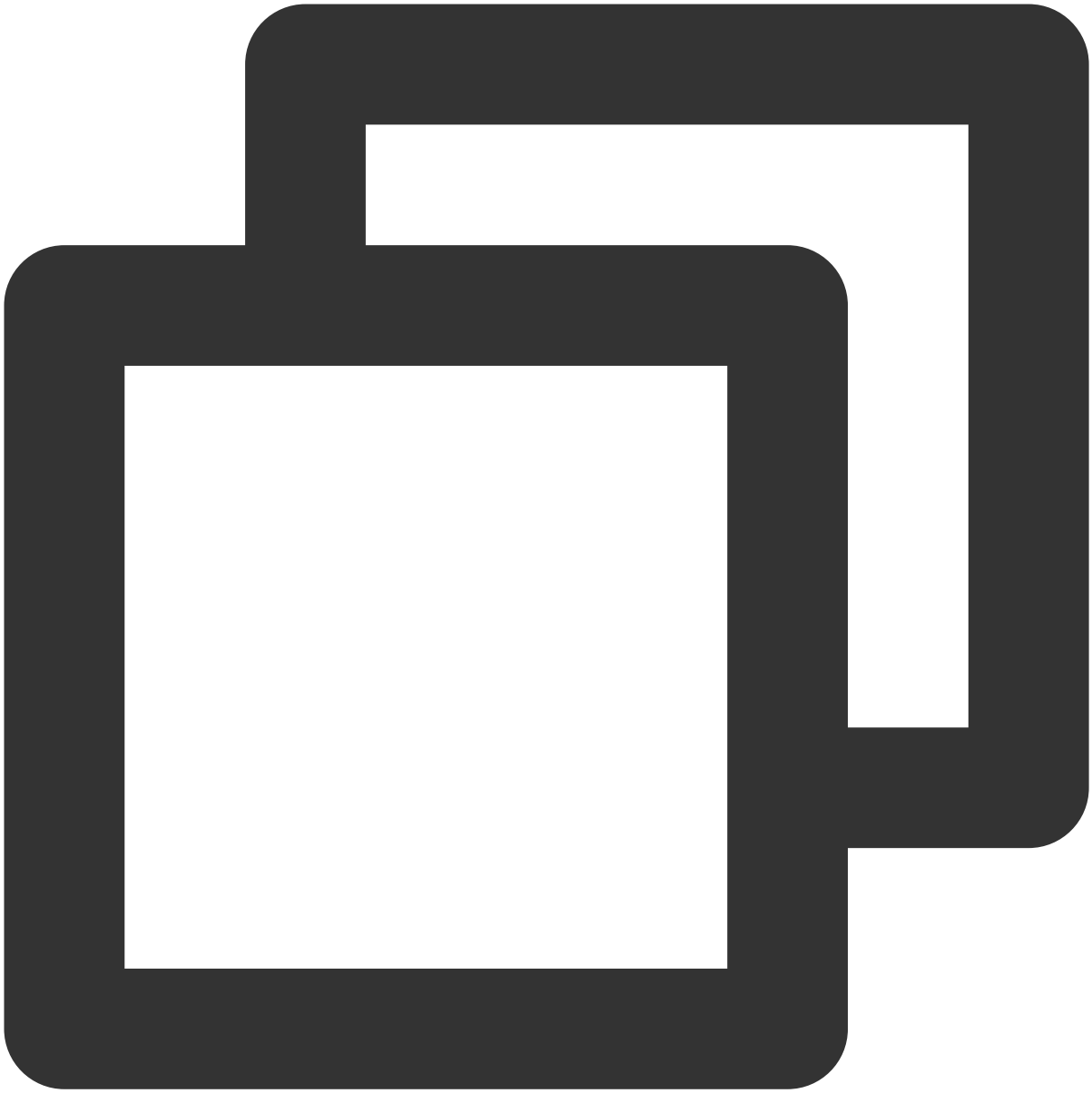
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

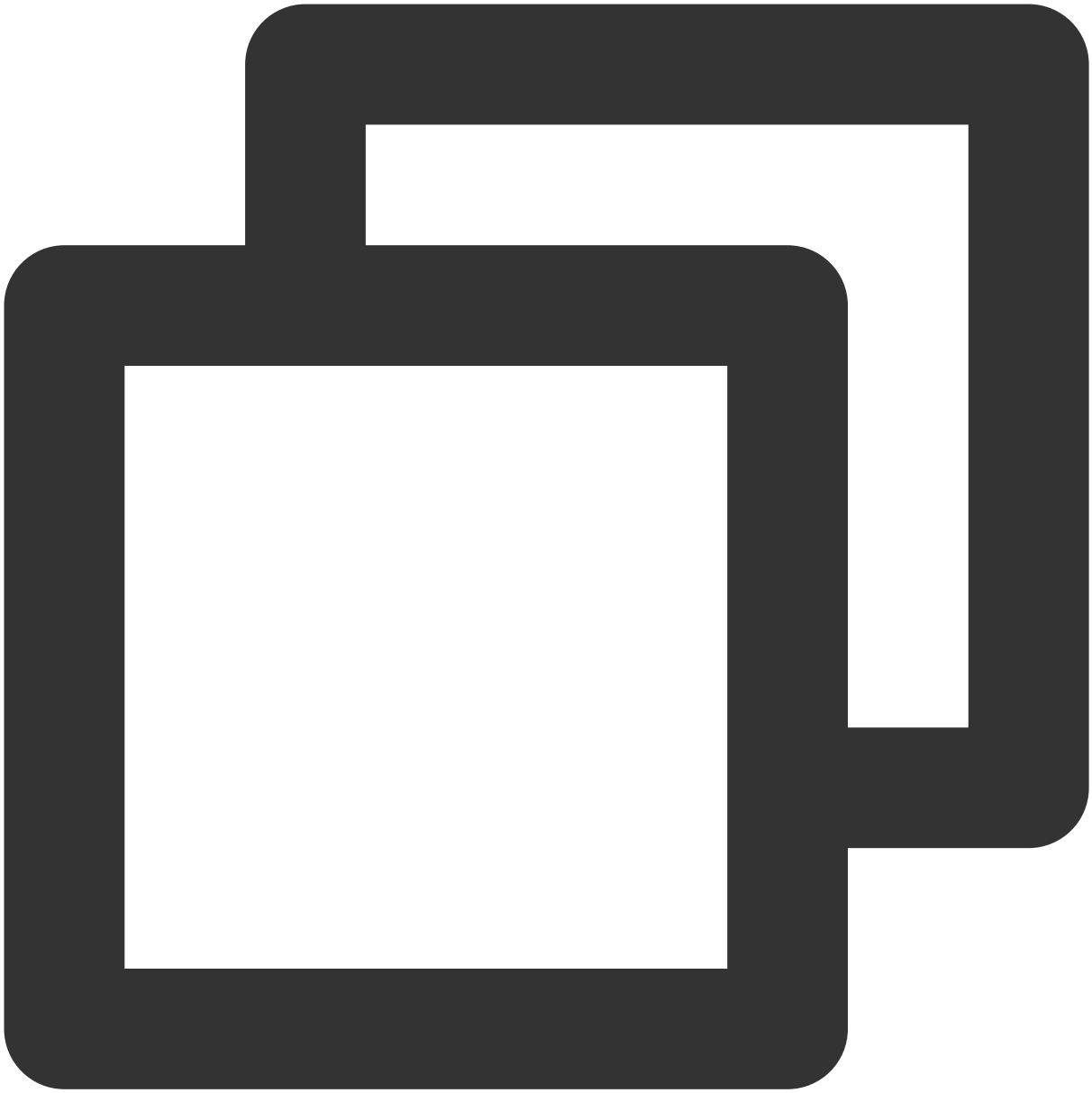
请求



```
PUT /?cors HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 11:15:01 GMT
Content-Type: application/xml
Content-Length: 1185
Content-MD5: ZNkhBxyjkaZcs1j7/cIE2A==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CORSConfiguration>
    <CORSRule>
```

```
        <AllowedOrigin>*</AllowedOrigin>
        <AllowedMethod>GET</AllowedMethod>
        <AllowedMethod>HEAD</AllowedMethod>
        <AllowedHeader>Range</AllowedHeader>
        <AllowedHeader>x-cos-server-side-encryption-customer-algorithm</AllowedHead
        <AllowedHeader>x-cos-server-side-encryption-customer-key</AllowedHeader>
        <AllowedHeader>x-cos-server-side-encryption-customer-key-MD5</AllowedHeader
        <ExposeHeader>Content-Length</ExposeHeader>
        <ExposeHeader>ETag</ExposeHeader>
        <ExposeHeader>x-cos-meta-author</ExposeHeader>
        <MaxAgeSeconds>600</MaxAgeSeconds>
    </CORSRule>
    <CORSRule>
        <ID>example-id</ID>
        <AllowedOrigin>https://example.com</AllowedOrigin>
        <AllowedOrigin>https://example.net</AllowedOrigin>
        <AllowedMethod>PUT</AllowedMethod>
        <AllowedMethod>GET</AllowedMethod>
        ...
        <AllowedMethod>HEAD</AllowedMethod>
        <AllowedHeader>*</AllowedHeader>
        <ExposeHeader>Content-Length</ExposeHeader>
        <ExposeHeader>ETag</ExposeHeader>
        <ExposeHeader>x-cos-meta-author</ExposeHeader>
        <MaxAgeSeconds>600</MaxAgeSeconds>
    </CORSRule>
</CORSConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Thu, 09 Jul 2020 11:15:01 GMT
Server: tencent-cos
x-cos-request-id: NWYwNmZjMzVfMzFiYjBiMDlfZjgzYV8xZDky****
```

GET Bucket cors

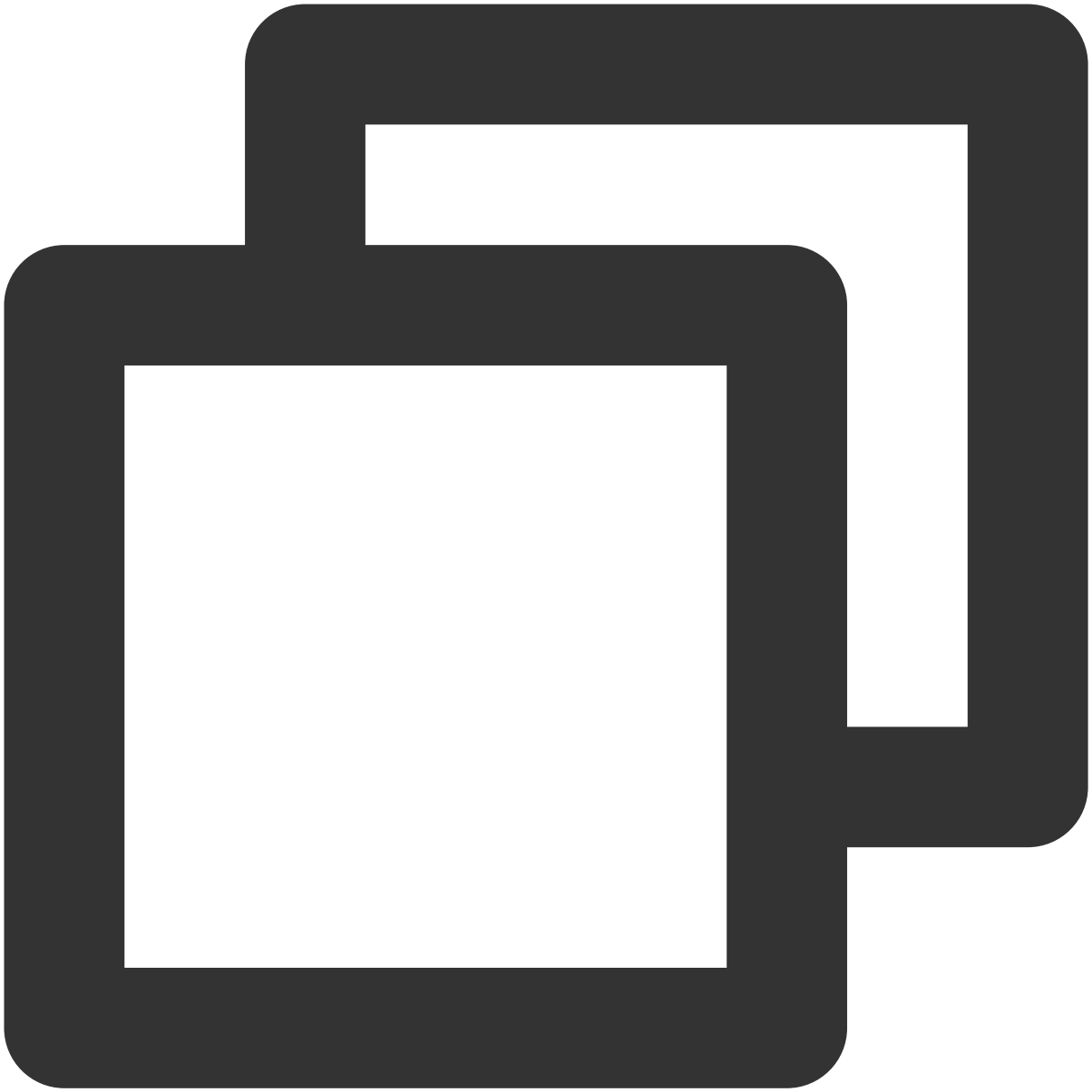
最近更新时间：2023-12-21 11:33:39

功能描述

GET Bucket cors 请求用于查询存储桶的跨域资源共享（CORS）访问控制。

请求

请求示例



```
GET /?cors HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

响应

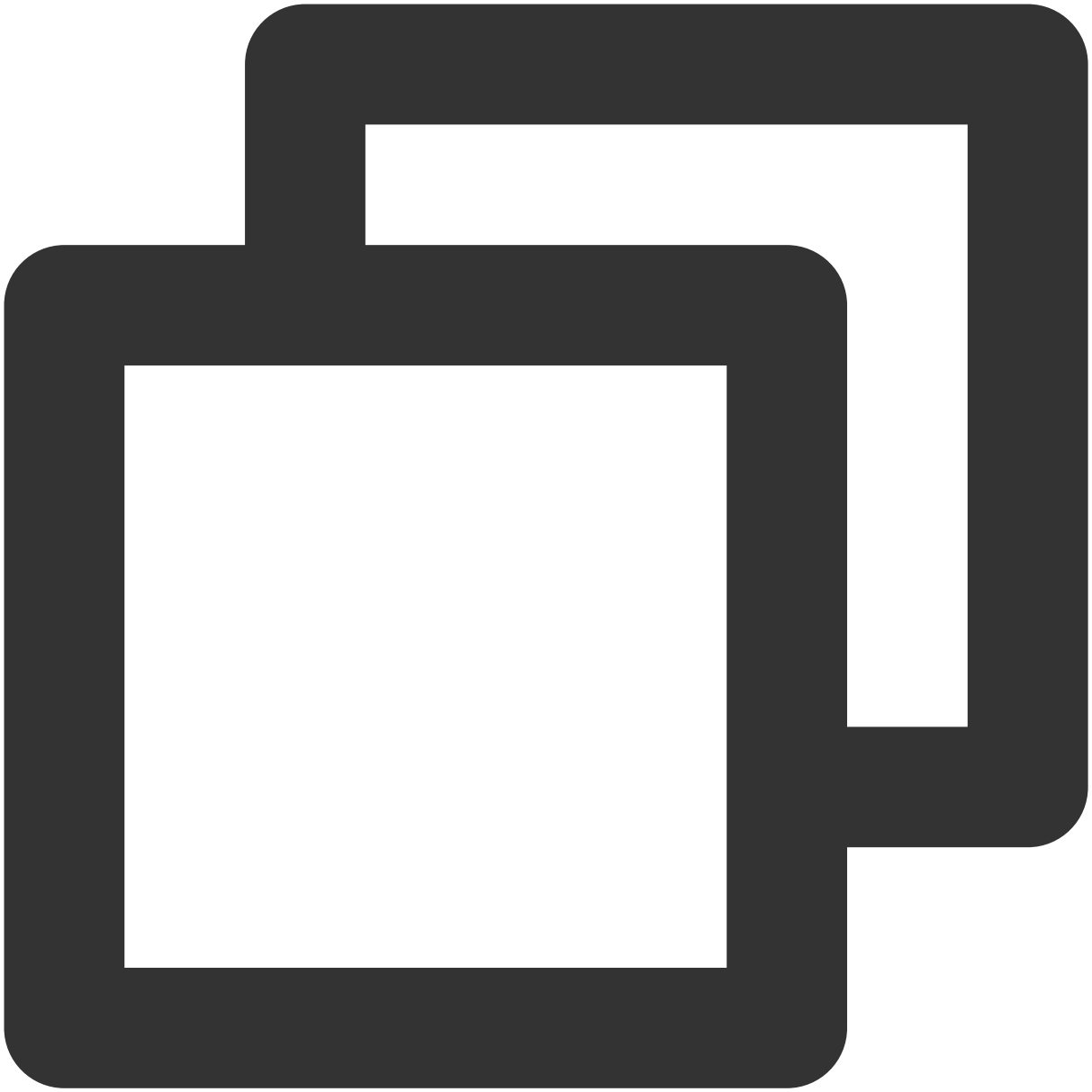
响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含完整的存储桶跨域资源共享（CORS）配置信息。

版权所有：腾讯云计算（北京）有限责任公司



```
<?xml version='1.0' encoding='utf-8' ?>
<CORSConfiguration>
    <CORSRule>
        <AllowedOrigin>string</AllowedOrigin>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedHeader>string</AllowedHeader>
        <AllowedHeader>string</AllowedHeader>
        <ExposeHeader>string</ExposeHeader>
        <ExposeHeader>string</ExposeHeader>
        <MaxAgeSeconds>integer</MaxAgeSeconds>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </CORSRule>
    <CORSRule>
        <ID>string</ID>
        <AllowedOrigin>string</AllowedOrigin>
        <AllowedOrigin>string</AllowedOrigin>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedMethod>enum</AllowedMethod>
        <AllowedHeader>string</AllowedHeader>
        <ExposeHeader>string</ExposeHeader>
        <ExposeHeader>string</ExposeHeader>
        <MaxAgeSeconds>integer</MaxAgeSeconds>
    </CORSRule>
</CORSConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
CORSConfiguration	无	保存 GET Bucket cors 结果的所有信息	Container

Container 节点 CORSConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型
CORSRule	CORSConfiguration	说明单条跨域资源共享（CORS）配置的所有信息	Container

Container 节点 CORSRule 的内容：

节点名称（关键字）	父节点	描述	类型
AllowedOrigin	CORSConfiguration.CORSRule	允许的访问来源，可能为 `*` 或包含 `*` 的通配格式，单条 CORSRule 可以配置多个 AllowedOrigin	string
AllowedMethod	CORSConfiguration.CORSRule	允许的 HTTP 操作方法（Method），对应 CORS 请求响应中的 Access-Control-Allow-Methods 头部，单条 CORSRule 可以配置多个 AllowedMethod。枚举值：PUT、GET、POST、DELETE、HEAD	enum
AllowedHeader	CORSConfiguration.CORSRule	允许浏览器发送 CORS 请求时携带的自定义 HTTP 请求头部，不区分英文大小写，可能为 `*`，单条 CORSRule 可以配置多个 AllowedHeader。	string

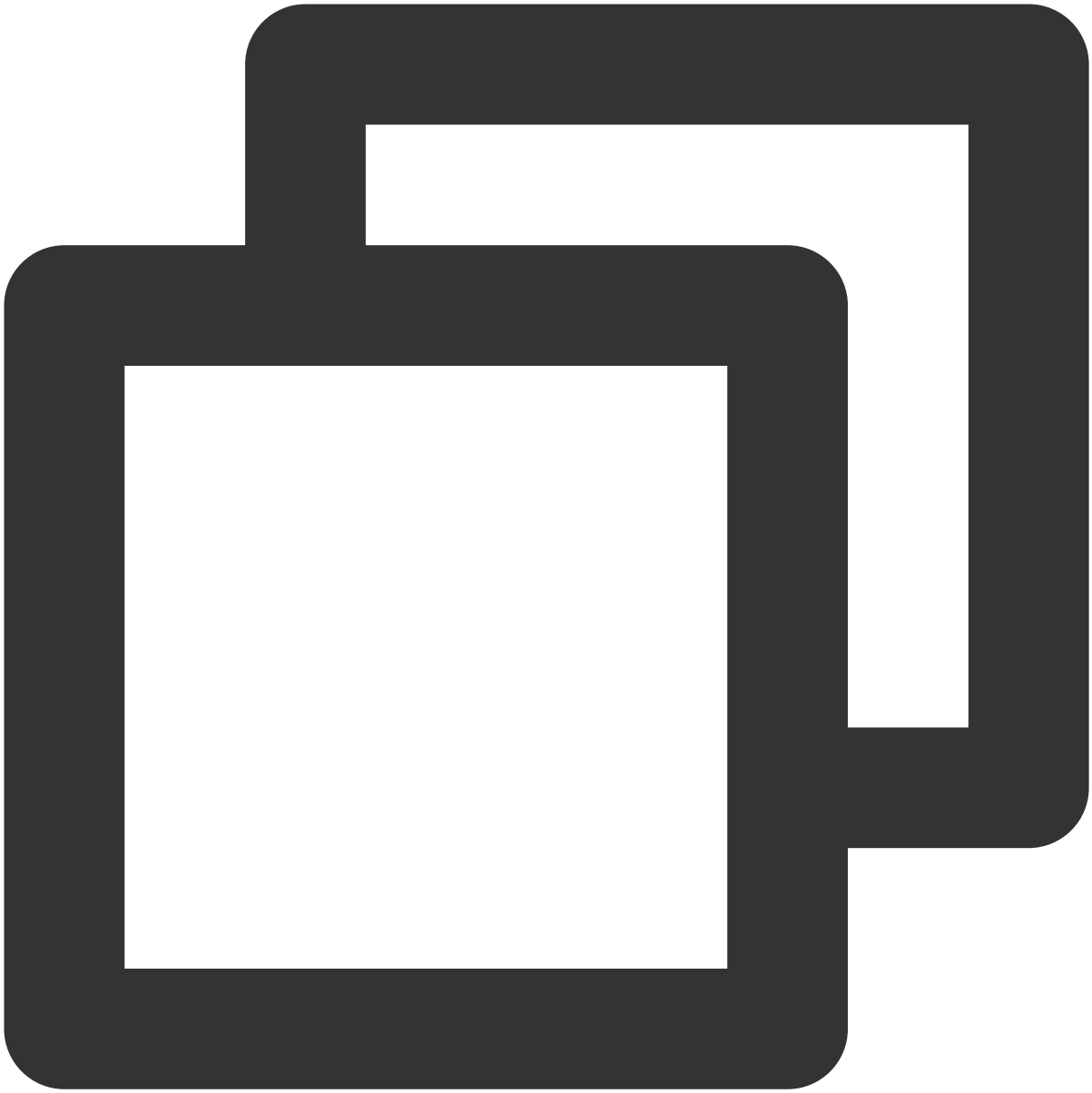
ExposeHeader	CORSConfiguration.CORSRule	允许浏览器获取的 CORS 请求响应中的头部，不区分英文大小写，单条 CORSRule 可以配置多个 ExposeHeader。	string
MaxAgeSeconds	CORSConfiguration.CORSRule	跨域资源共享配置的有效时间，单位为秒，对应 CORS 请求响应中的 Access-Control-Max-Age 头部，单条 CORSRule 只能配置一个 MaxAgeSeconds	integer
ID	CORSConfiguration.CORSRule	单条 CORSRule 配置的 ID，该节点的存在与否取决于使用 PUT Bucket cors 设置存储桶跨域资源共享配置时是否指定了 ID，单条 CORSRule 最多配置一个 ID	string

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

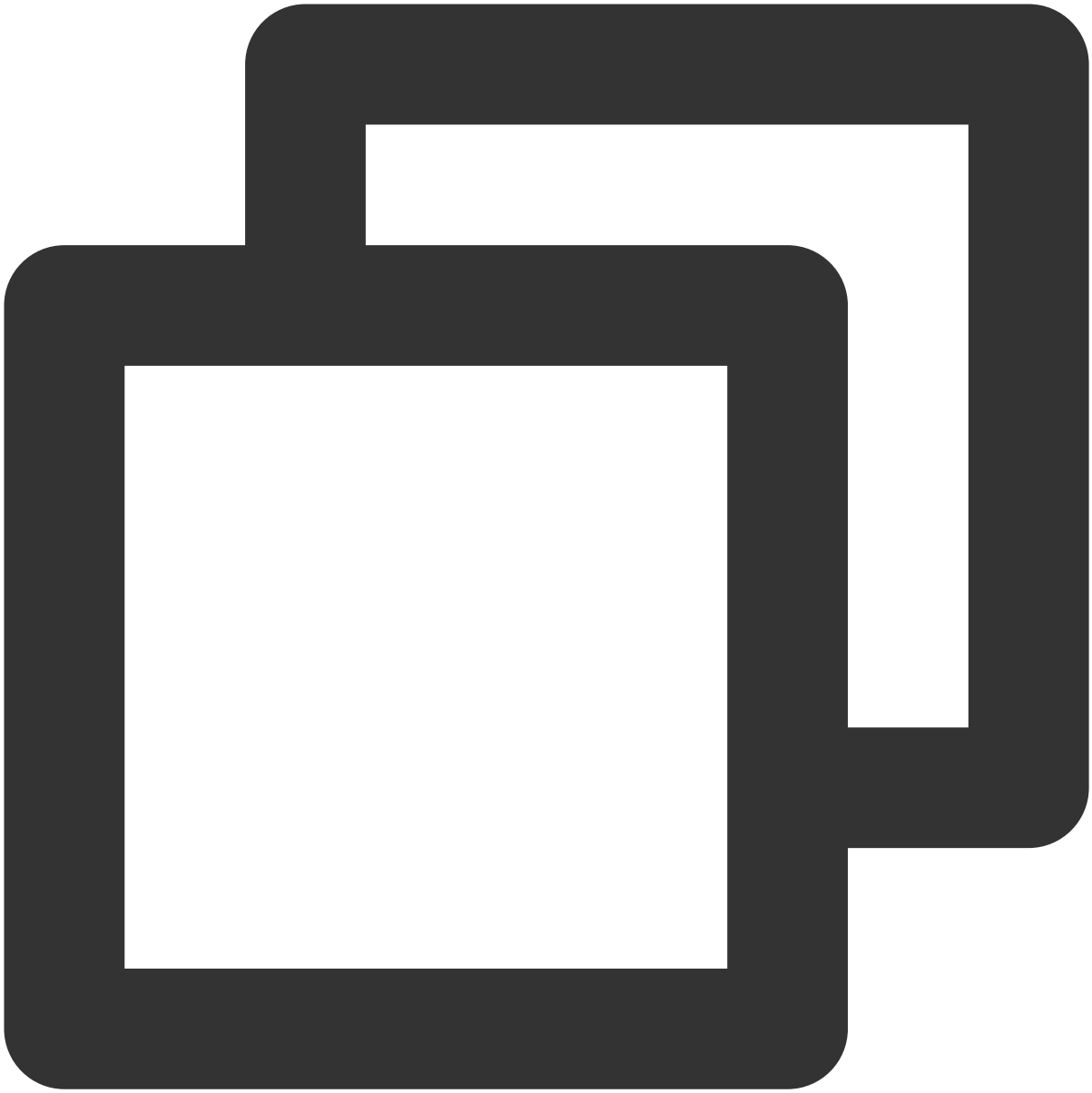
实际案例

请求



```
GET /?cors HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 11:15:12 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1196
Connection: close
Date: Thu, 09 Jul 2020 11:15:12 GMT
Server: tencent-cos
x-cos-request-id: NWYwNmZjNDBfN2ViMTJhMDlfZDNjOV8xYjdk****

<?xml version='1.0' encoding='utf-8' ?>
<CORSConfiguration>
    <CORSRule>
```

```
        <AllowedOrigin>*</AllowedOrigin>
        <AllowedMethod>GET</AllowedMethod>
        <AllowedMethod>HEAD</AllowedMethod>
        <AllowedHeader>Range</AllowedHeader>
        <AllowedHeader>x-cos-server-side-encryption-customer-algorithm</AllowedHead
        <AllowedHeader>x-cos-server-side-encryption-customer-key</AllowedHeader>
        <AllowedHeader>x-cos-server-side-encryption-customer-key-MD5</AllowedHeader
        <ExposeHeader>Content-Length</ExposeHeader>
        <ExposeHeader>ETag</ExposeHeader>
        <ExposeHeader>x-cos-meta-author</ExposeHeader>
        <MaxAgeSeconds>600</MaxAgeSeconds>
    </CORSRule>
    <CORSRule>
        <ID>example-id</ID>
        <AllowedOrigin>https://example.com</AllowedOrigin>
        <AllowedOrigin>https://example-1.com</AllowedOrigin>
        <AllowedMethod>PUT</AllowedMethod>
        <AllowedMethod>GET</AllowedMethod>
        ...
        <AllowedMethod>HEAD</AllowedMethod>
        <AllowedHeader>*</AllowedHeader>
        <ExposeHeader>Content-Length</ExposeHeader>
        <ExposeHeader>ETag</ExposeHeader>
        <ExposeHeader>x-cos-meta-author</ExposeHeader>
        <MaxAgeSeconds>600</MaxAgeSeconds>
    </CORSRule>
</CORSConfiguration>
```

版权所有：腾讯云计算（北京）有限责任公司

DELETE Bucket cors

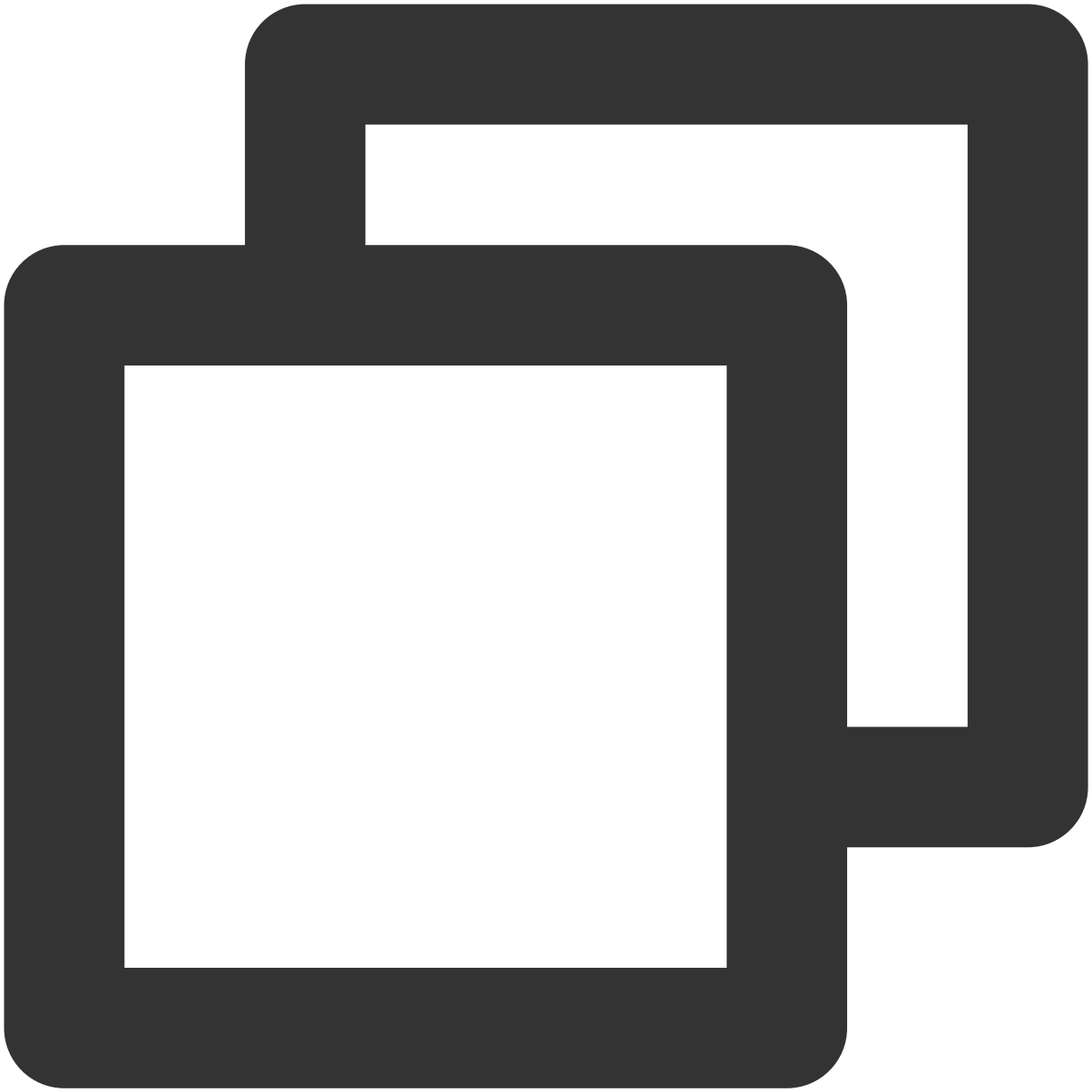
最近更新时间：2023-12-21 11:33:55

功能描述

DELETE Bucket cors 请求用于删除存储桶的跨域资源共享（CORS）访问控制。

请求

请求示例



```
DELETE /?cors HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

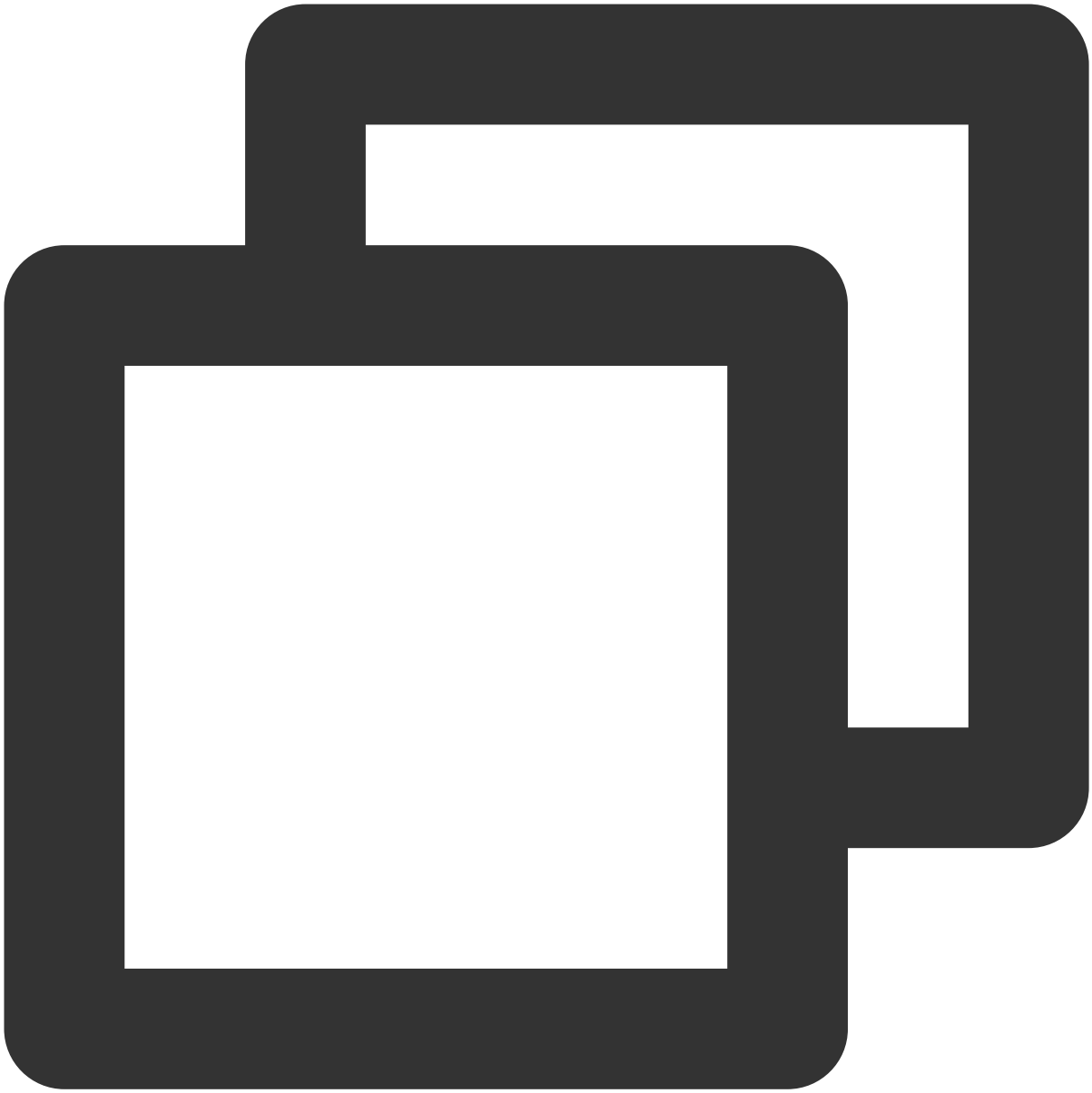
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

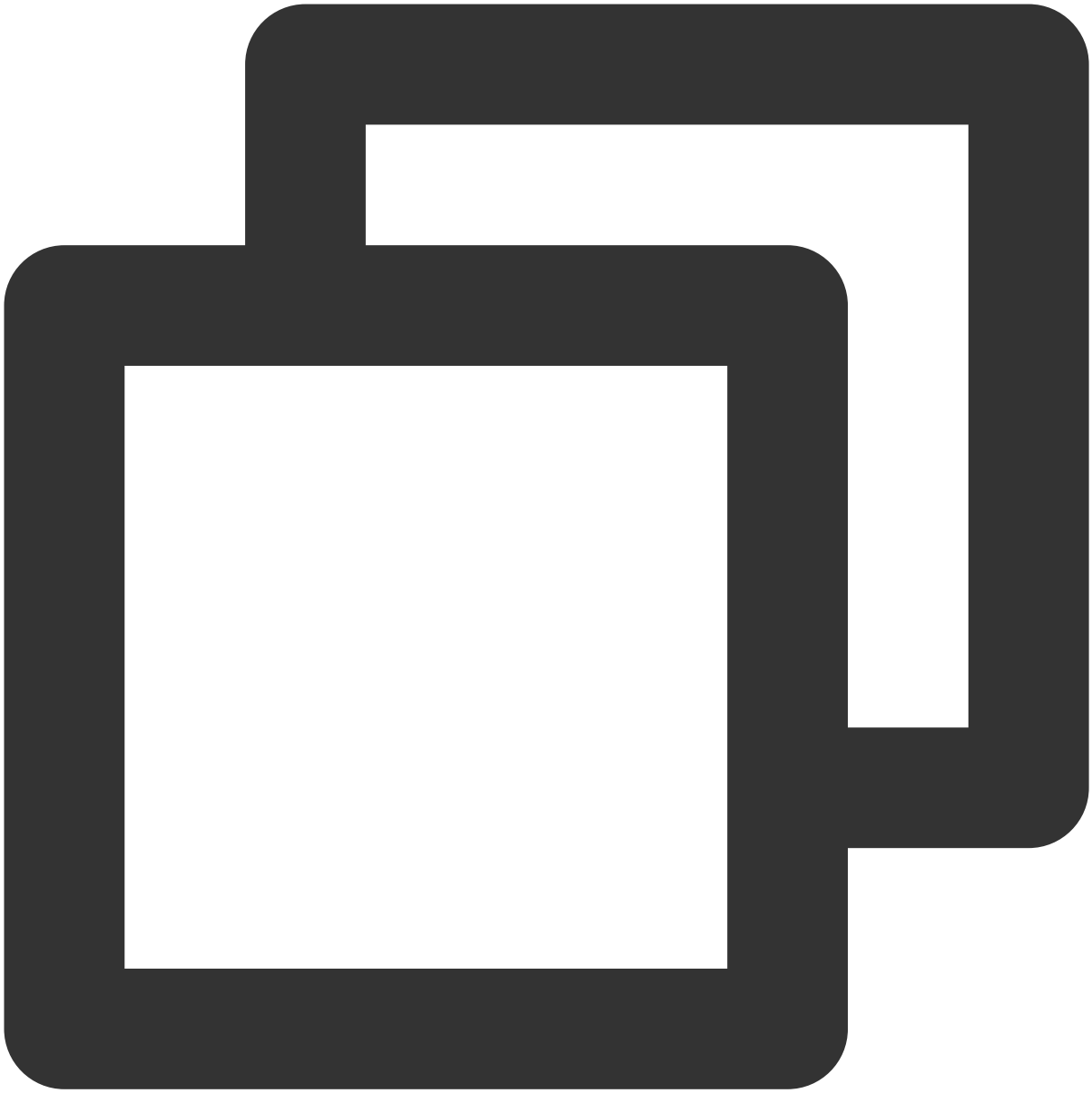
实际案例

请求



```
DELETE /?cors HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 11:52:50 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 204 No Content
Connection: close
Date: Thu, 09 Jul 2020 11:52:50 GMT
Server: tencent-cos
x-cos-request-id: NWYwNzA1MTJfNjRiODJhMDlfMWMyMzlfMWMyNzhl****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODczNTBmNjMwZmQ0MTZkMjg0Njl
Content-Length: 0
```

生命周期（lifecycle）
PUT Bucket lifecycle

最近更新时间：2023-12-21 11:40:42

功能描述

COS 支持用户以生命周期配置的方式来管理 Bucket 中 Object 的生命周期。生命周期配置包含一个或多个将应用于一组对象规则的规则集 (其中每个规则为 COS 定义一个操作)。
这些操作分为以下两种：
转换操作：定义对象转换为另一个存储类的时间。例如，您可以选择在对象创建30天后将其转换为低频存储（STANDARD_IA，适用于不常访问）存储类别。同时也支持将数据沉降到智能分层存储（INTELLIGENT_TIERING，访问模式不固定）和归档存储（ARCHIVE，成本更低）。具体参数查看请求示例说明中 Transition 项。
过期操作：指定 Object 的过期时间。COS 将会自动为用户删除过期的 Object。

细节分析

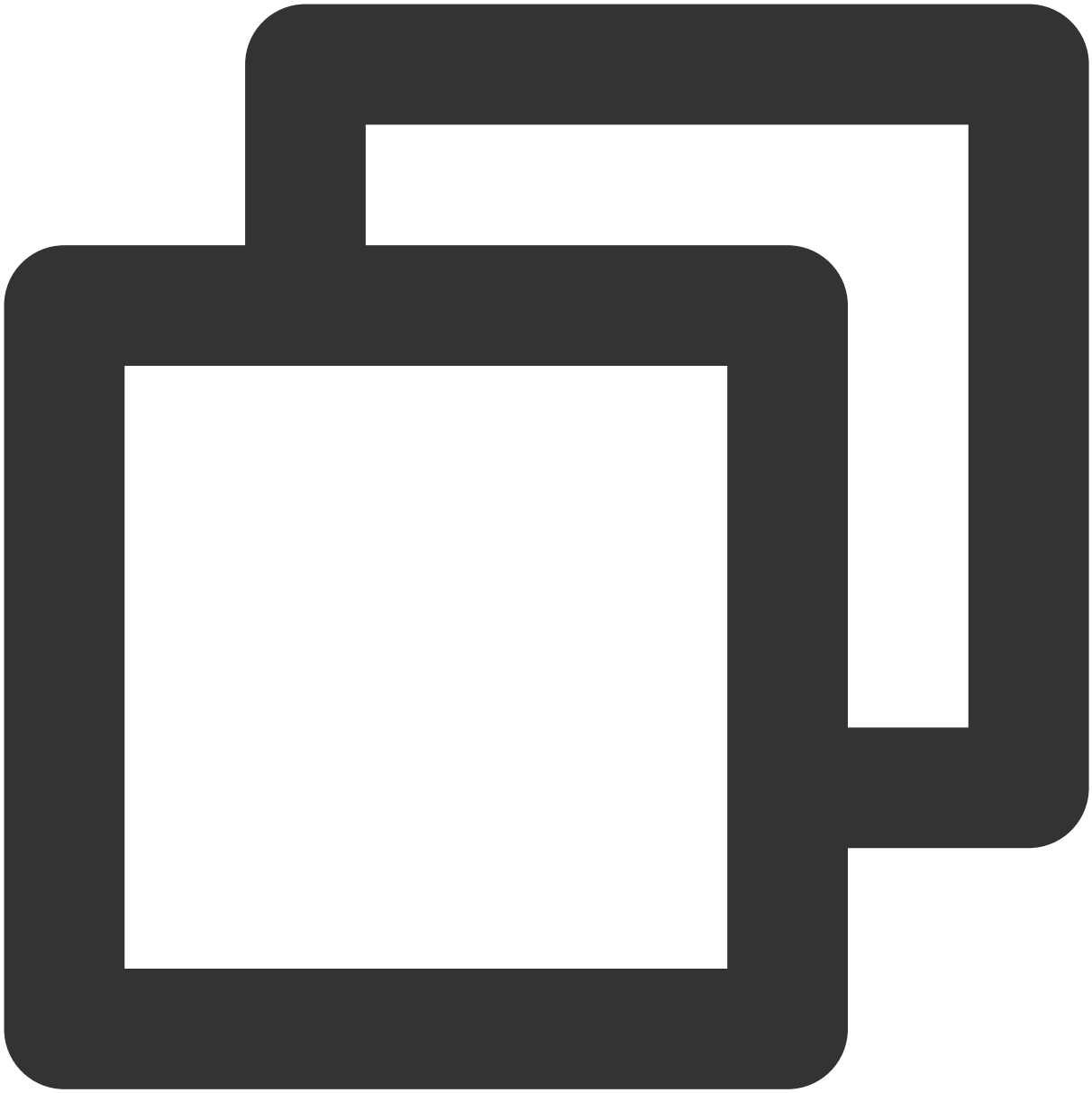
PUT Bucket lifecycle 用于为 Bucket 创建一个新的生命周期配置。

注意

如果该 Bucket 已配置生命周期，使用该接口创建新的配置的同时则会覆盖原有的配置。
同一条生命周期规则中不可同时支持 Days 和 Date 参数，请分成两条规则分别传入，具体请参见下文 实际案例。
开启了 多 AZ 配置的存储桶，不支持将多 AZ 存储类型沉降到单 AZ 存储类型。
每个存储桶最多可添加1000条生命周期规则。

请求

请求示例



```
PUT /?lifecycle HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Content-Length: length
Date: GMT Date
Authorization: Auth String
Content-MD5: MD5
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

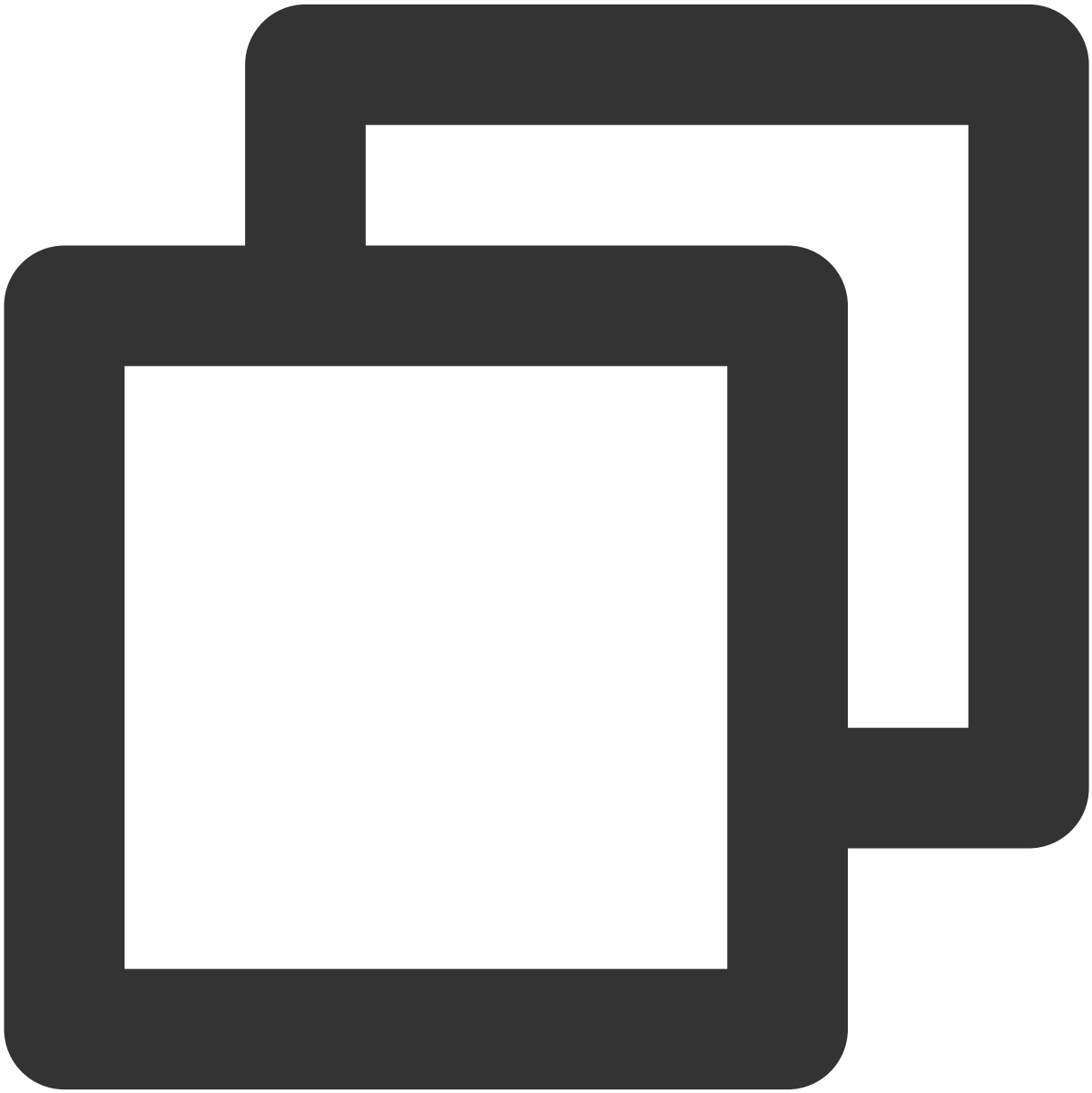
<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该 API 接口请求的请求体具体节点内容为：



```
<LifecycleConfiguration>
```

```
    <Rule>
        <ID></ID>
        <Filter>
            <And>
                <Prefix></Prefix>
                <Tag>
                    <Key></Key>
                    <Value></Value>
                </Tag>
            </And>
        </Filter>
        <Status></Status>
        <Transition>
            <Days></Days>
            <StorageClass></StorageClass>
        </Transition>
        <NoncurrentVersionExpiration>
            <NoncurrentDays></NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
    <Rule>
        <ID></ID>
        <Filter>
            <Prefix></Prefix>
        </Filter>
        <Status></Status>
        <Transition>
            <Days></Days>
            <StorageClass></StorageClass>
        </Transition>
        <NoncurrentVersionTransition>
            <NoncurrentDays></NoncurrentDays>
            <StorageClass></StorageClass>
        </NoncurrentVersionTransition>
    </Rule>
    <Rule>
        <ID></ID>
        <Filter>
            <Prefix></Prefix>
        </Filter>
        <Status></Status>
        <Expiration>
            <ExpiredObjectDeleteMarker></ExpiredObjectDeleteMarker>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NoncurrentDays></NoncurrentDays>
        </NoncurrentVersionExpiration>
```

```
        <AbortIncompleteMultipartUpload>
            <DaysAfterInitiation></DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

具体内容描述如下：

节点名称（关键字）	父节点	描述
LifecycleConfiguration	无	生命周期配置
Rule	LifecycleConfiguration	规则描述
ID	LifecycleConfiguration.Rule	用于唯一地标超过255个字
Filter	LifecycleConfiguration.Rule	Filter 用于描述集合
And	LifecycleConfiguration.Rule.Filter	对象筛选器中需要指定多种此元素，例如和 Tag 筛选，Tag 筛选。
Prefix	LifecycleConfiguration.Rule.Filter.And	指定规则所适缀的对象受该多只能有一个
Tag	LifecycleConfiguration.Rule.Filter.And	标签集合，最
Key	LifecycleConfiguration.Rule.Filter.And.Tag	标签的 Key，节，支持英文格、加号、减号、点号、冒
Value	LifecycleConfiguration.Rule.Filter.And.Tag	标签的 Value节，支持英文加号、减号、号、冒号、斜
Status	LifecycleConfiguration.Rule	指明规则是否Enabled，Dis

Expiration	LifecycleConfiguration.Rule	规则过期属性
Transition	LifecycleConfiguration.Rule	规则转换属性 时进行存储类 存储类型
Days	LifecycleConfiguration.Rule.Transition 或 Expiration	指明规则对应 的修改日期过 如果是 Trans 是非负整数 如果是 Expira 为正整数，最
ExpiredObjectDeleteMarker	LifecycleConfiguration.Rule.Expiration	删除过期对象 true，false
AbortIncompleteMultipartUpload	LifecycleConfiguration.Rule	设置允许分片 长时间
DaysAfterInitiation	LifecycleConfiguration.Rule.AbortIncompleteMultipartUpload	指明分片上传 须完成上传
NoncurrentVersionExpiration	LifecycleConfiguration.Rule	指明非当前版
NoncurrentVersionTransition	LifecycleConfiguration.Rule	指明非当前版 储类型的转换
NoncurrentDays	LifecycleConfiguration.Rule.NoncurrentVersionExpiration或 NoncurrentVersionTransition	指明规则对应 非当前版本多 如果是 Trans 是非负整数 如果是 Expira 为正整数，最
StorageClass	LifecycleConfiguration.Rule.Transition 或 NoncurrentVersionTransition	指定 Object 注 枚举值：STA MAZ_STAND INTELLIGEN MAZ_INTELL ARCHIVE，D 关于存储类型 储类型概述

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

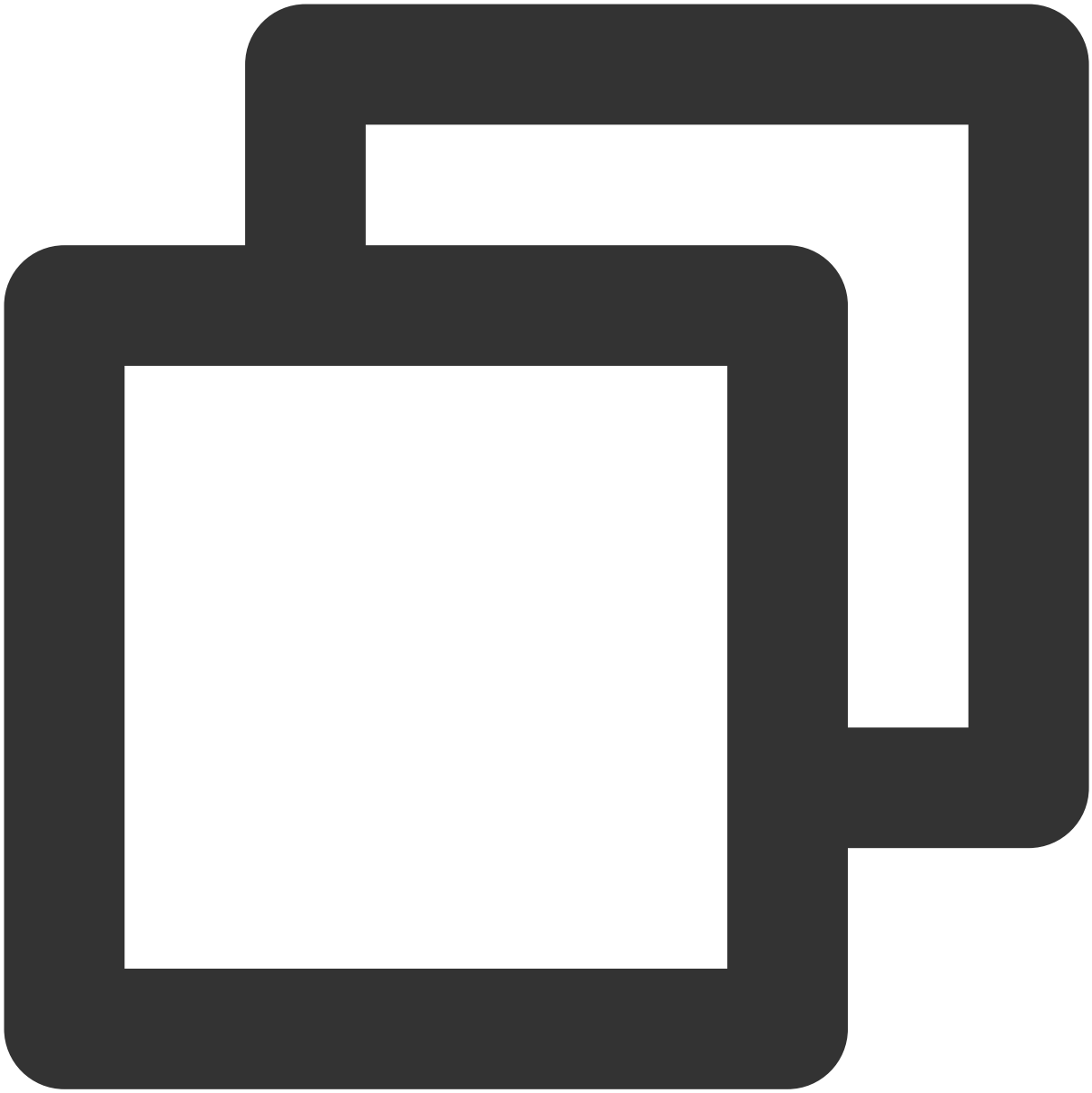
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

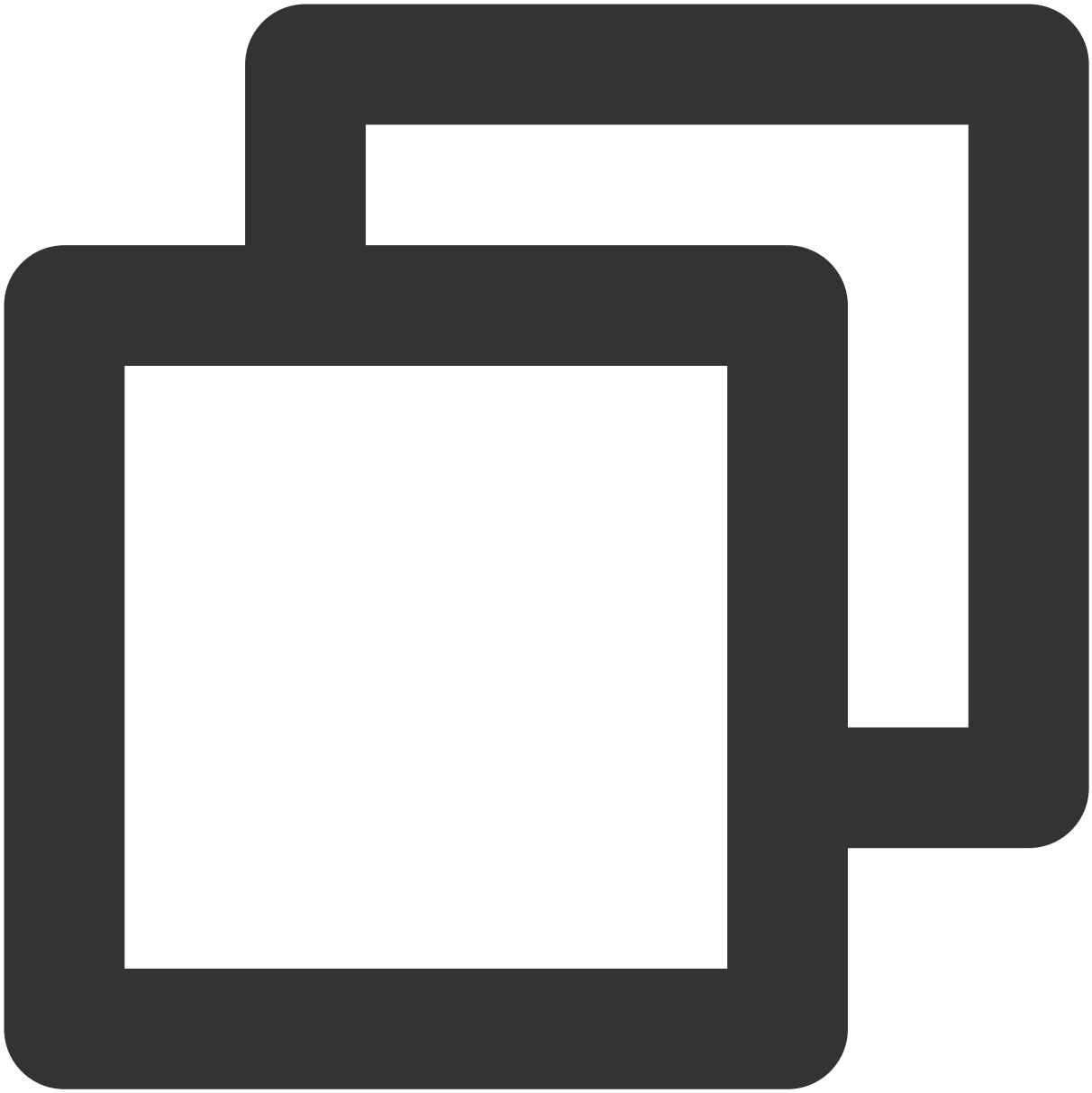
请求



```
PUT /?lifecycle HTTP/1.1
Host:examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 11:59:33 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Content-MD5: LcNUuow8OSZMrEDnvndw1Q==
Content-Length: 348
Content-Type: application/x-www-form-urlencoded



<LifecycleConfiguration>
```

```
        <Rule>
              <ID>id1</ID>
              <Filter>
                    <Prefix>documents/</Prefix>
              </Filter>
              <Status>Enabled</Status>
              <Transition>
                    <Days>100</Days>
                    <StorageClass>ARCHIVE</StorageClass>
              </Transition>
        </Rule>
        <Rule>
              <ID>id2</ID>
              <Filter>
                    <Prefix>logs/</Prefix>
              </Filter>
              <Status>Enabled</Status>
              <Transition>
                    <Days>10</Days>
                    <StorageClass>STANDARD_IA</StorageClass>
              </Transition>
        </Rule>
</LifecycleConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Wed, 16 Aug 2017 11:59:33 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDMzYTRfMjQ4OGY3Xzc3NGRf****
```

GET Bucket lifecycle

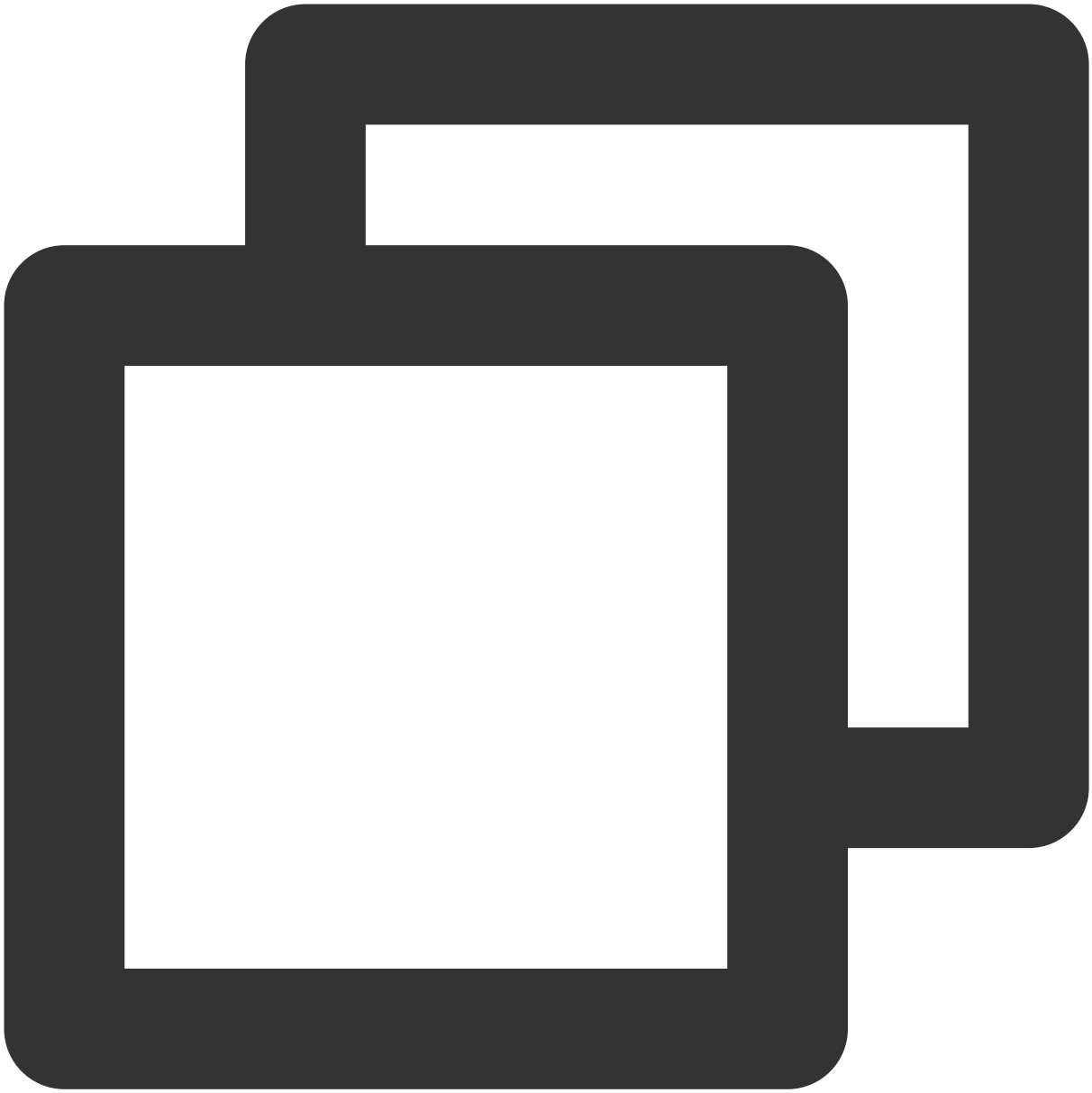
最近更新时间：2023-12-21 11:40:46

功能描述

GET Bucket lifecycle 用于查询 Bucket 的生命周期配置。如果该 Bucket 没有配置生命周期规则会返回 NoSuchLifecycleConfiguration。

请求

请求示例



```
GET /?lifecycle HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

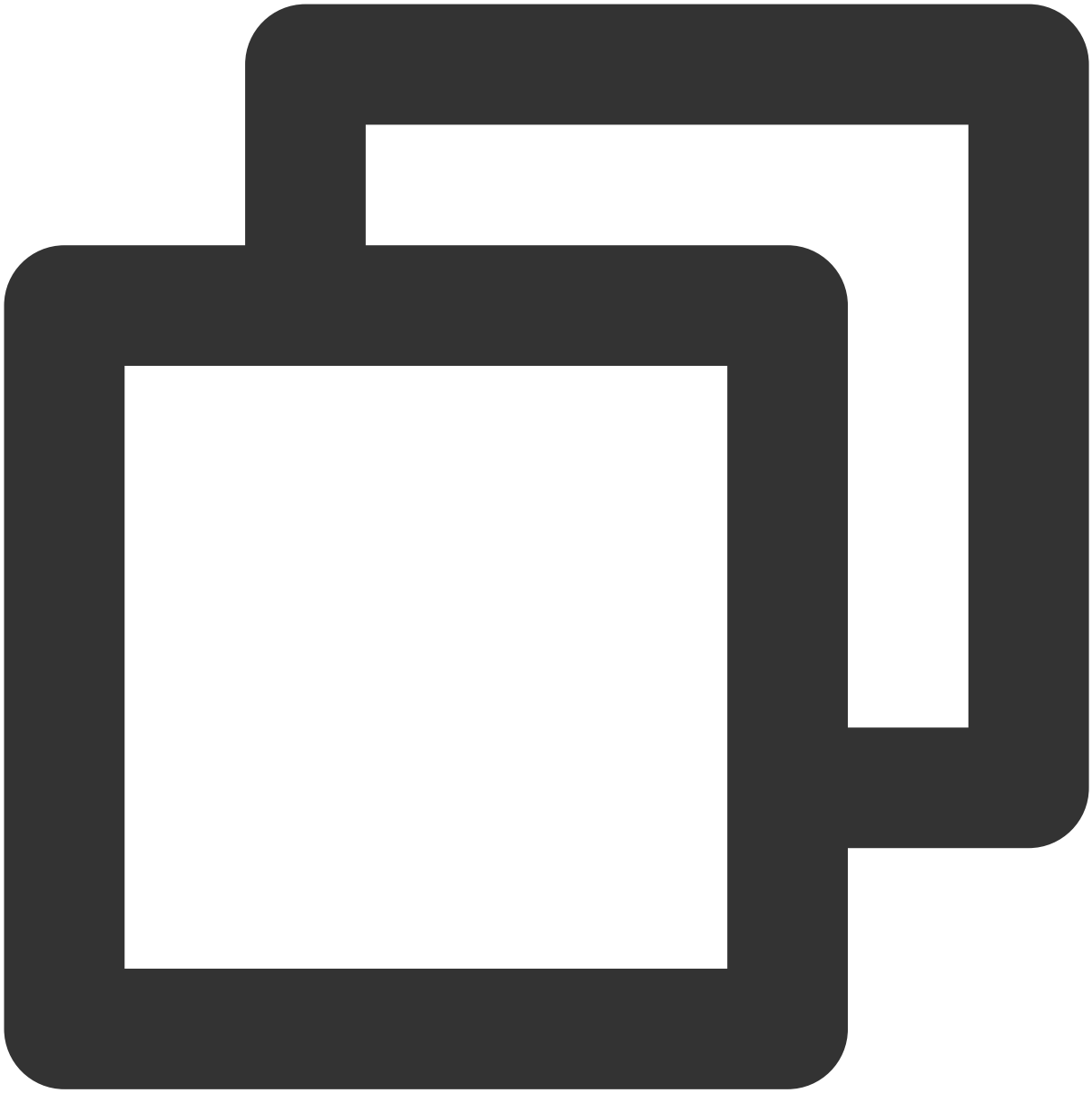
响应体中各个元素的内容及含义与 PUT Buket lifecycle 时的请求体一致。详情请参见 PUT Bucket lifecycle 文档中的请求体节点描述内容。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

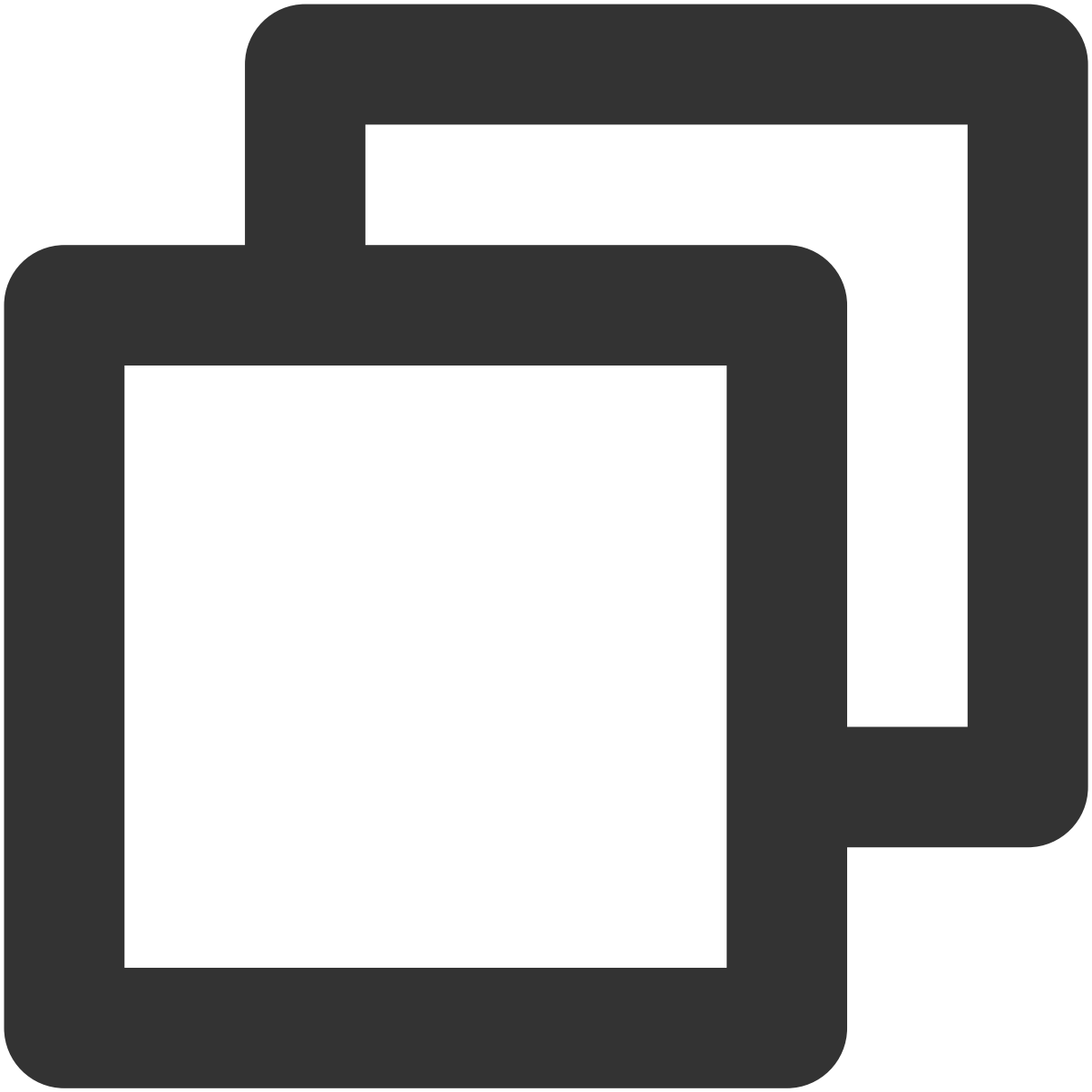
实际案例

请求



```
GET /?lifecycle HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 12:23:54 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 312
Date: Wed, 16 Aug 2017 12:23:54 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDM5NWFfMjQ4OGY3Xzc3NGRf****

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
```

```
        <Prefix>documents/</Prefix>
      </Filter>
      <Status>Enabled</Status>
      <Transition>
        <Days>100</Days>
        <StorageClass>STANDARD_IA</StorageClass>
      </Transition>
    </Rule>
    <Rule>
      <ID>id2</ID>
      <Filter>
        <Prefix>logs/</Prefix>
      </Filter>
      <Status>Enabled</Status>
      <Expiration>
        <Days>10</Days>
      </Expiration>
    </Rule>
</LifecycleConfiguration>
```

DELETE Bucket lifecycle

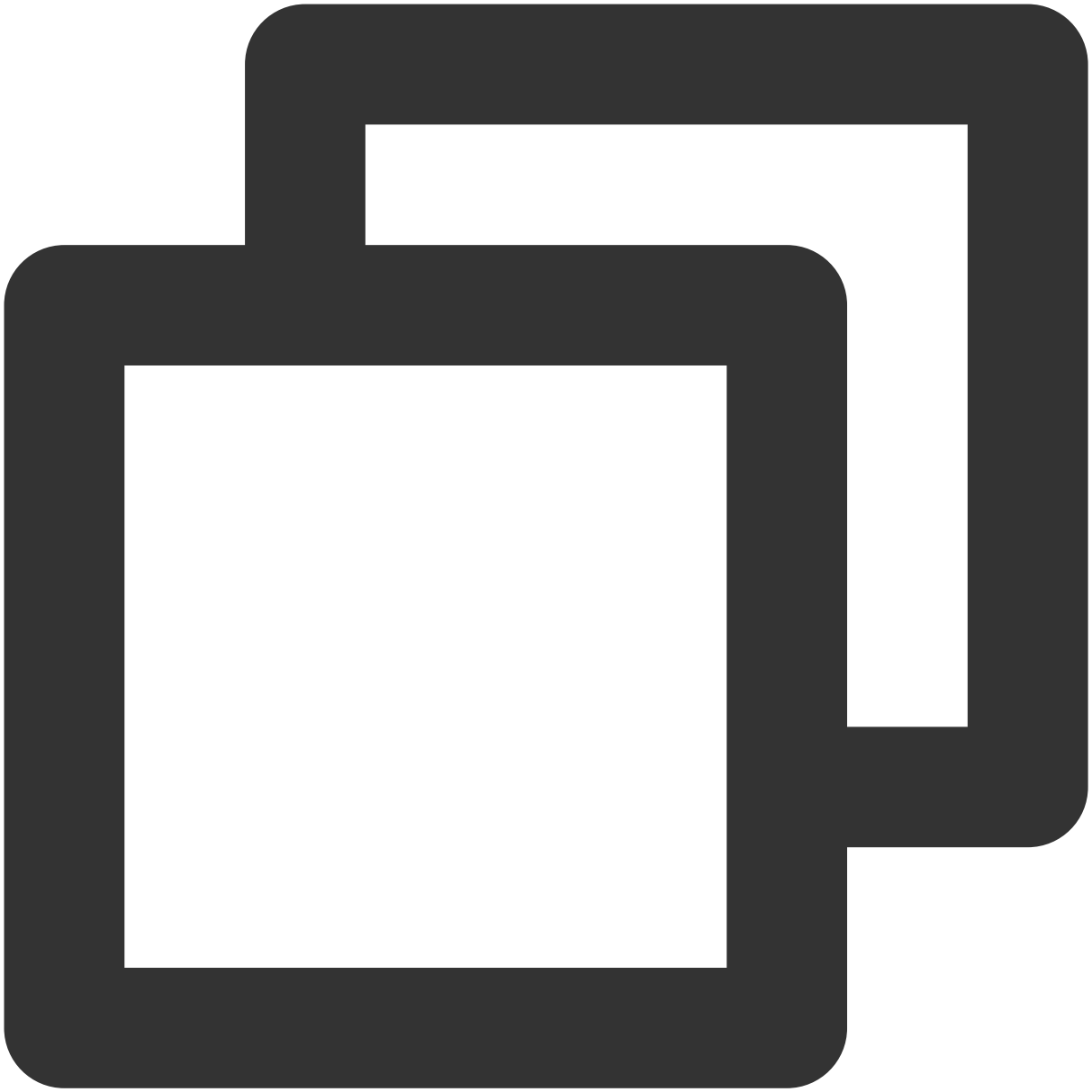
最近更新时间：2023-12-21 11:41:12

功能描述

DELETE Bucket lifecycle 用于删除存储桶的生命周期配置。如果该 Bucket 未配置生命周期规则，将返回 NoSuchLifecycleConfiguration。

请求

请求示例



```
DELETE /?lifecycle HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

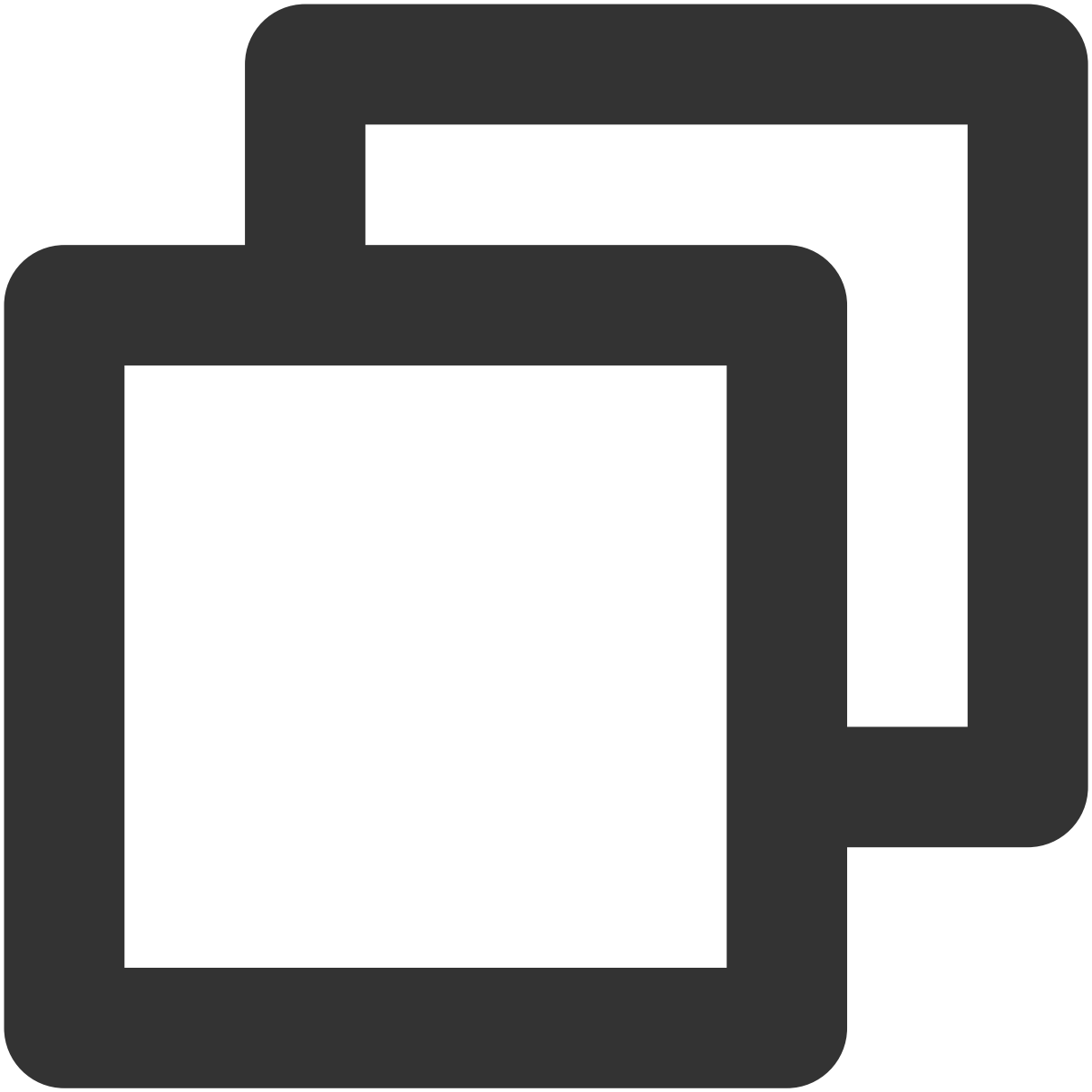
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

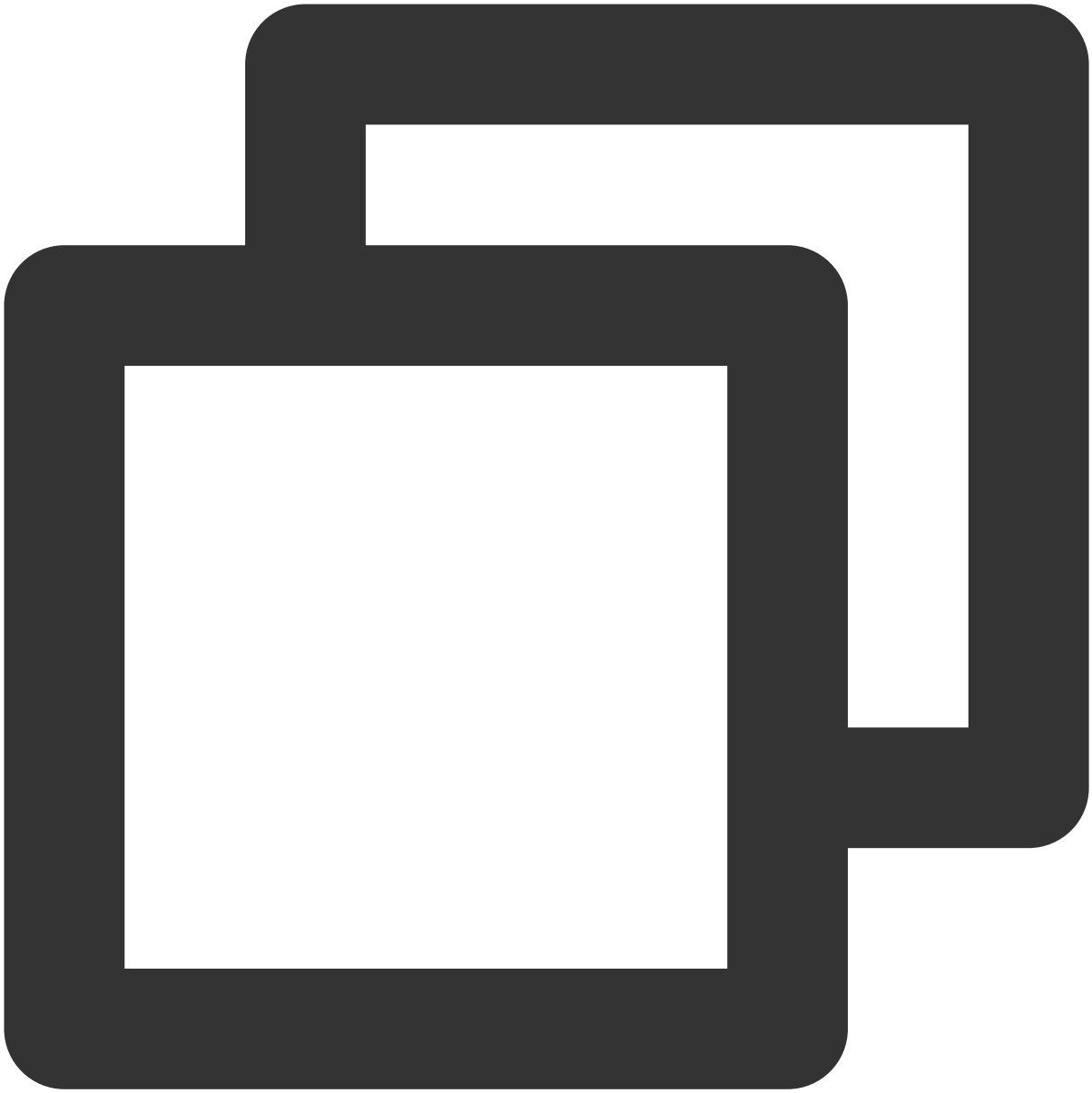
实际案例

请求



```
DELETE /?lifecycle HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 12:59:09 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98JM&q-sig
```

响应



```
HTTP /1.1 204 No Content
Content-Type: application/xml
Date: Wed, 16 Aug 2017 12:59:09 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDQxOWNfMjQ4OGY3Xzc3NGRf****
```

存储桶策略（policy）

PUT Bucket policy

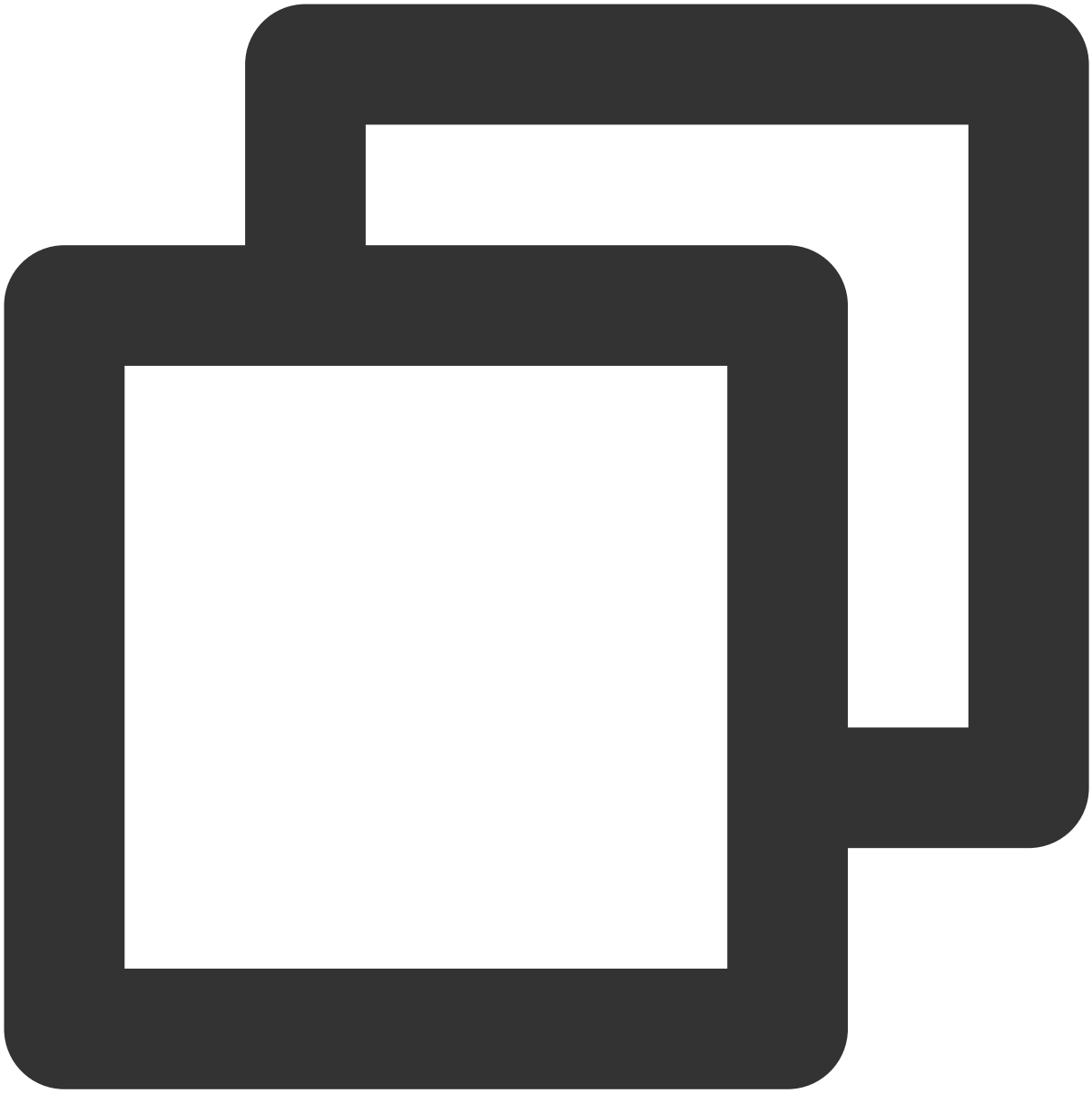
最近更新时间：2023-12-21 11:35:46

功能描述

PUT Bucket policy 请求可以向 Bucket 写入权限策略，当存储桶已存在权限策略时，该请求上传的策略将覆盖原有的权限策略。

请求

请求示例



```
PUT /?policy HTTP/1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: date
Content-Type:application/json
Content-MD5:MD5
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

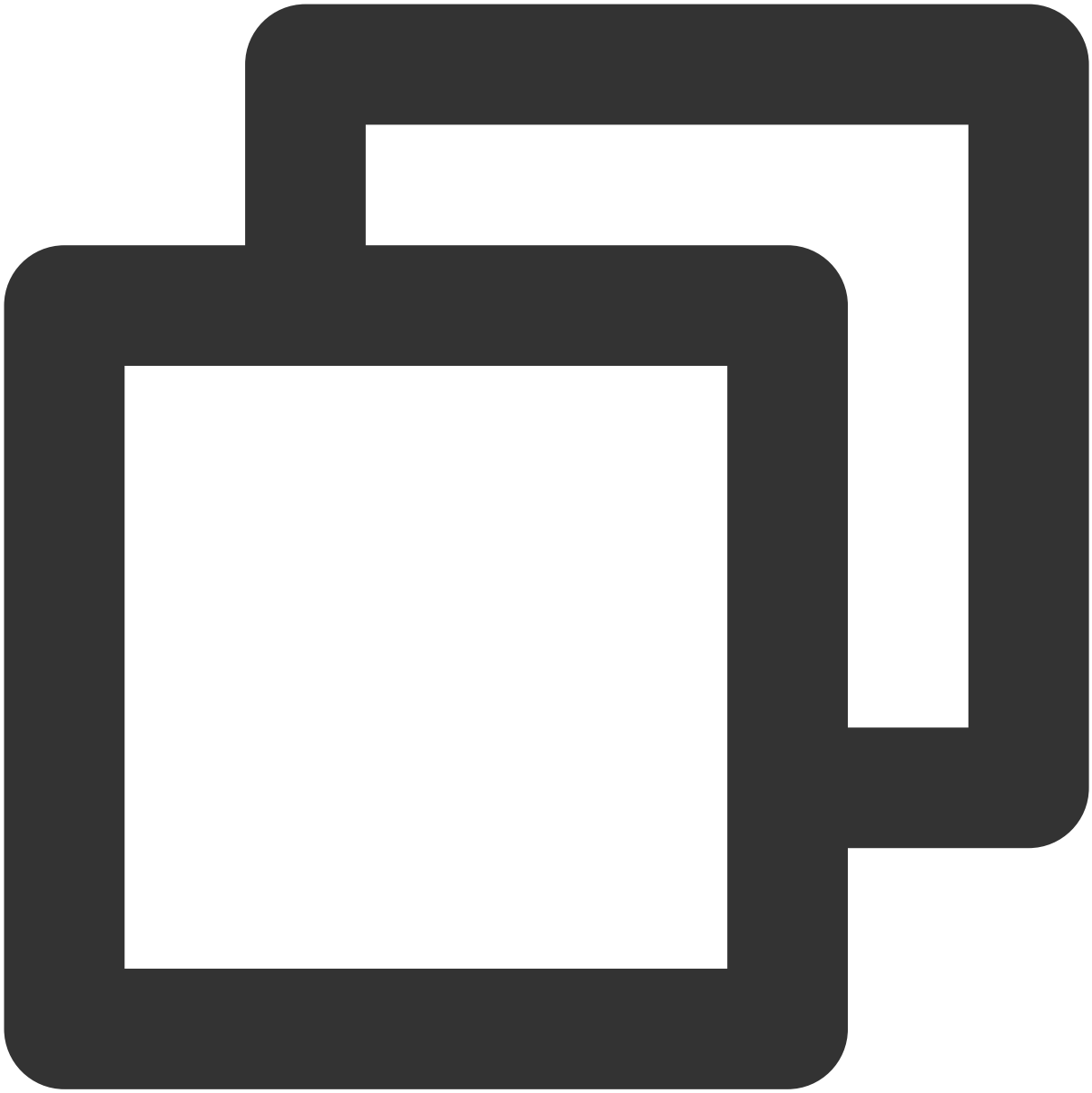
<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

下面的示例表示给主账号100000000001下的子账号100000000011授权，以允许访问存储桶名为 examplebucket-1250000000 中的对象列表。关于访问策略中的元素介绍，请参见 访问策略语言概述，以及授权策略示例请参见 COS API 授权策略使用指引。



```
{
  "Statement": [
    {
      "Principal": {
        "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000011"
        ]
      },
      "Effect": "allow",
      "Action": [
        "name/cos:GetBucket"
```

```
        ],
        "Resource": [
          "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
        ]
      }
    ],
    "version": "2.0"
}
```

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

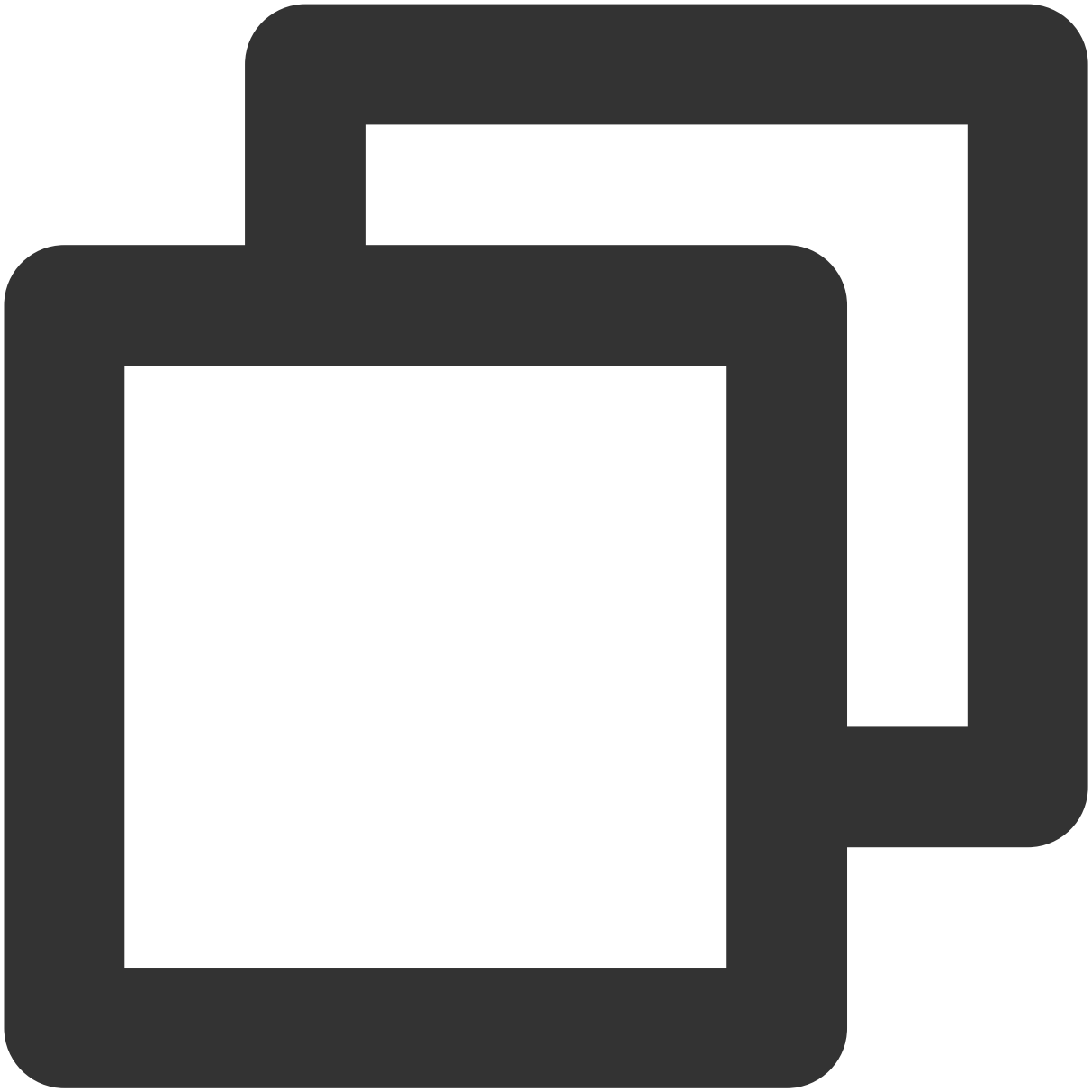
该请求的响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

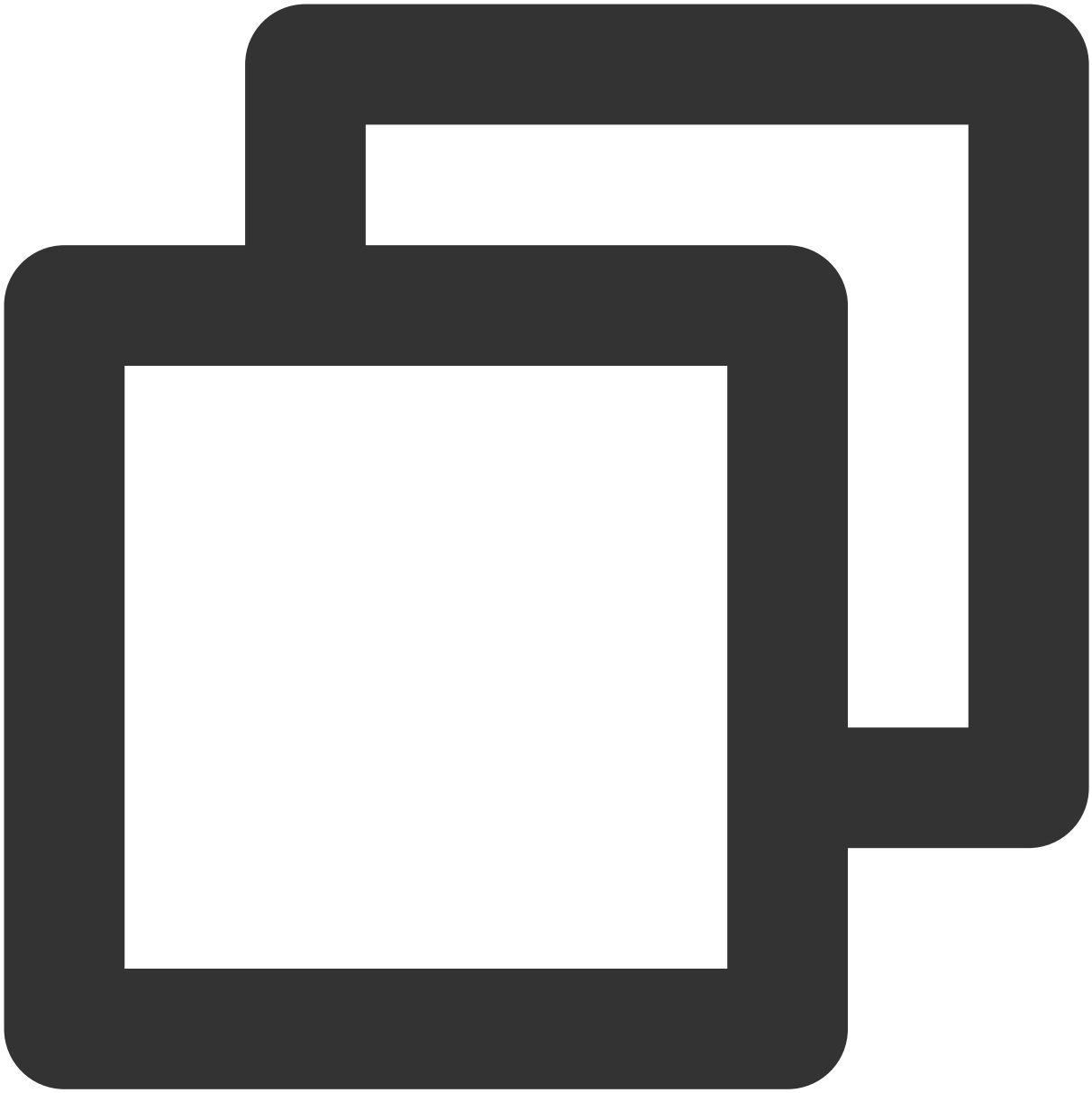
请求



```
PUT /?policy HTTP/1.1
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-sig
Content-Type: application/json
Content-Length: 233

{
  "Statement": [
    {
      "Principal": {
        "qcs": [
```

```
            "qcs::cam::uin/100000000001:uin/100000000001"
          ]
        },
        "Effect": "allow",
        "Action": [
          "name/cos:GetBucket"
        ],
        "Resource": [
          "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
        ]
      }
    ],
    "version": "2.0"
}
```

响应



```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu Jan 19 16:19:22 2017
Server: tencent-cos
x-cos-request-id: NTg4MDc2OGFfNDUyMDRlXzc3NTlf****
```

GET Bucket policy

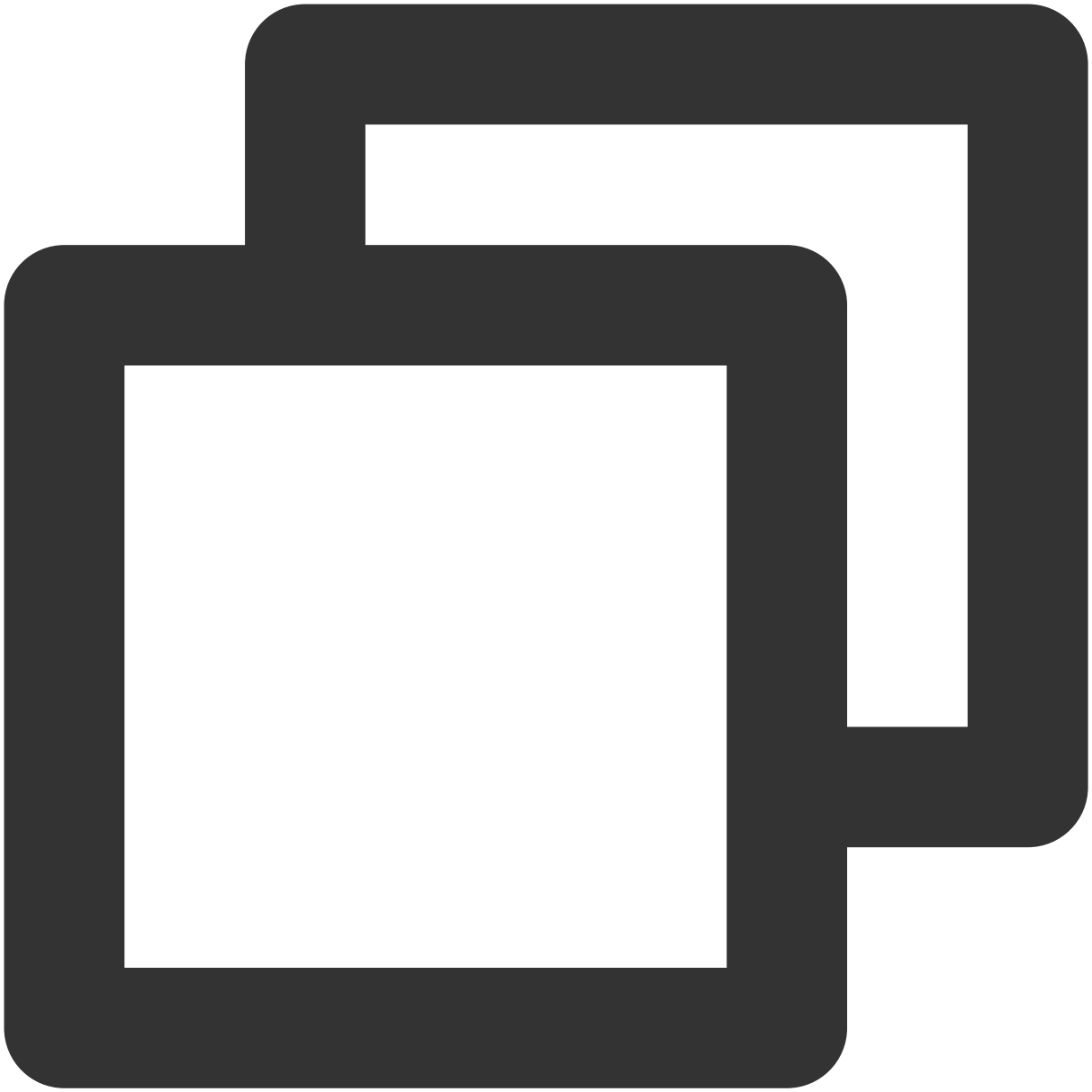
最近更新时间：2023-12-21 11:35:50

功能描述

GET Bucket policy 请求可以向 Bucket 读取权限策略。

请求

请求示例



```
GET /?policy HTTP/1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

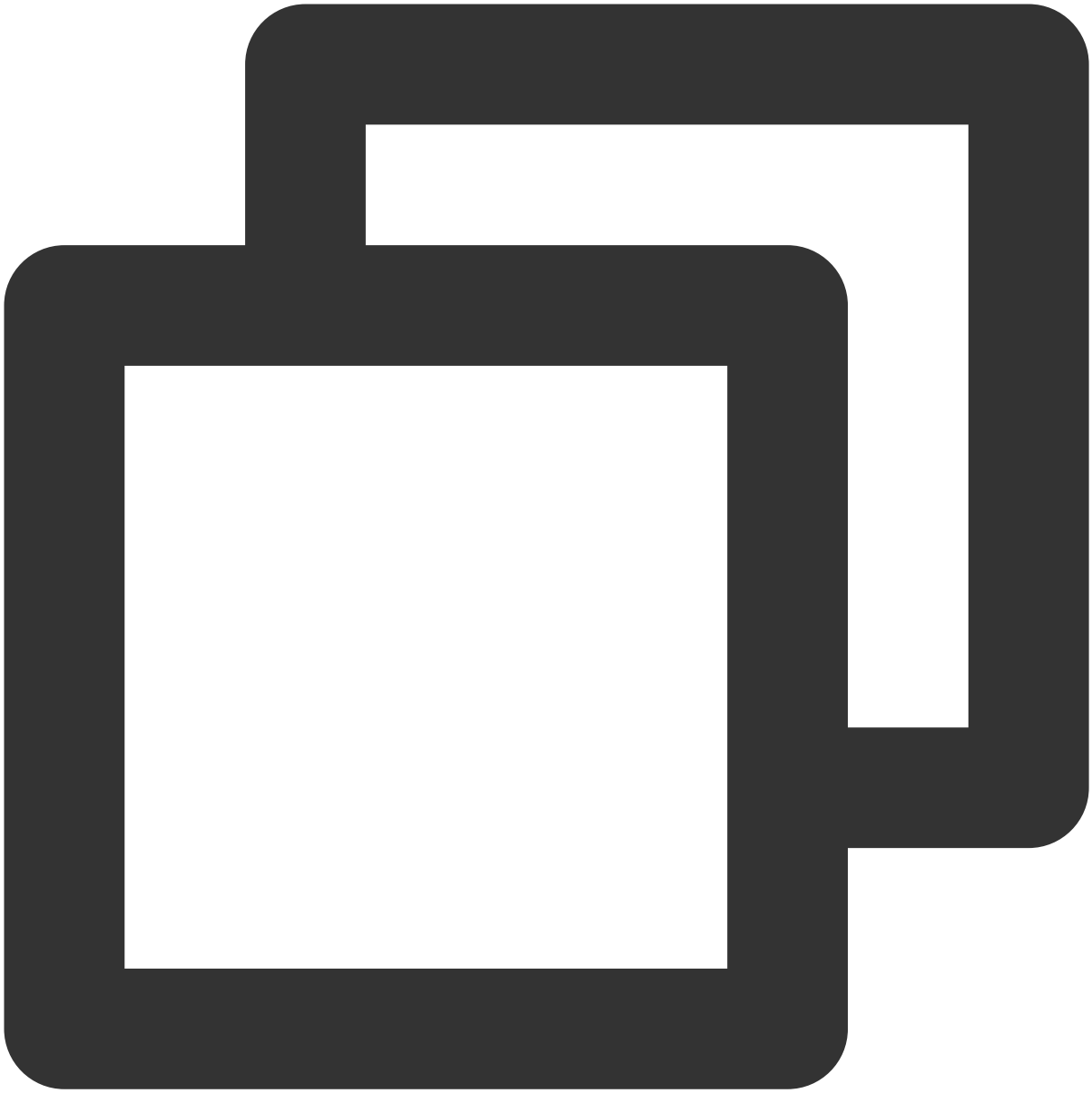
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

响应返回权限策略。



```
{
  "Statement": [
    {
      "Principal": {
        "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000001"
        ]
      },
      "Effect": "allow",
      "Action": [
        "name/cos:GetBucket"
```

```
        ],
        "Resource": [
          "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
        ]
      }
    ],
    "version": "2.0"
  }
```

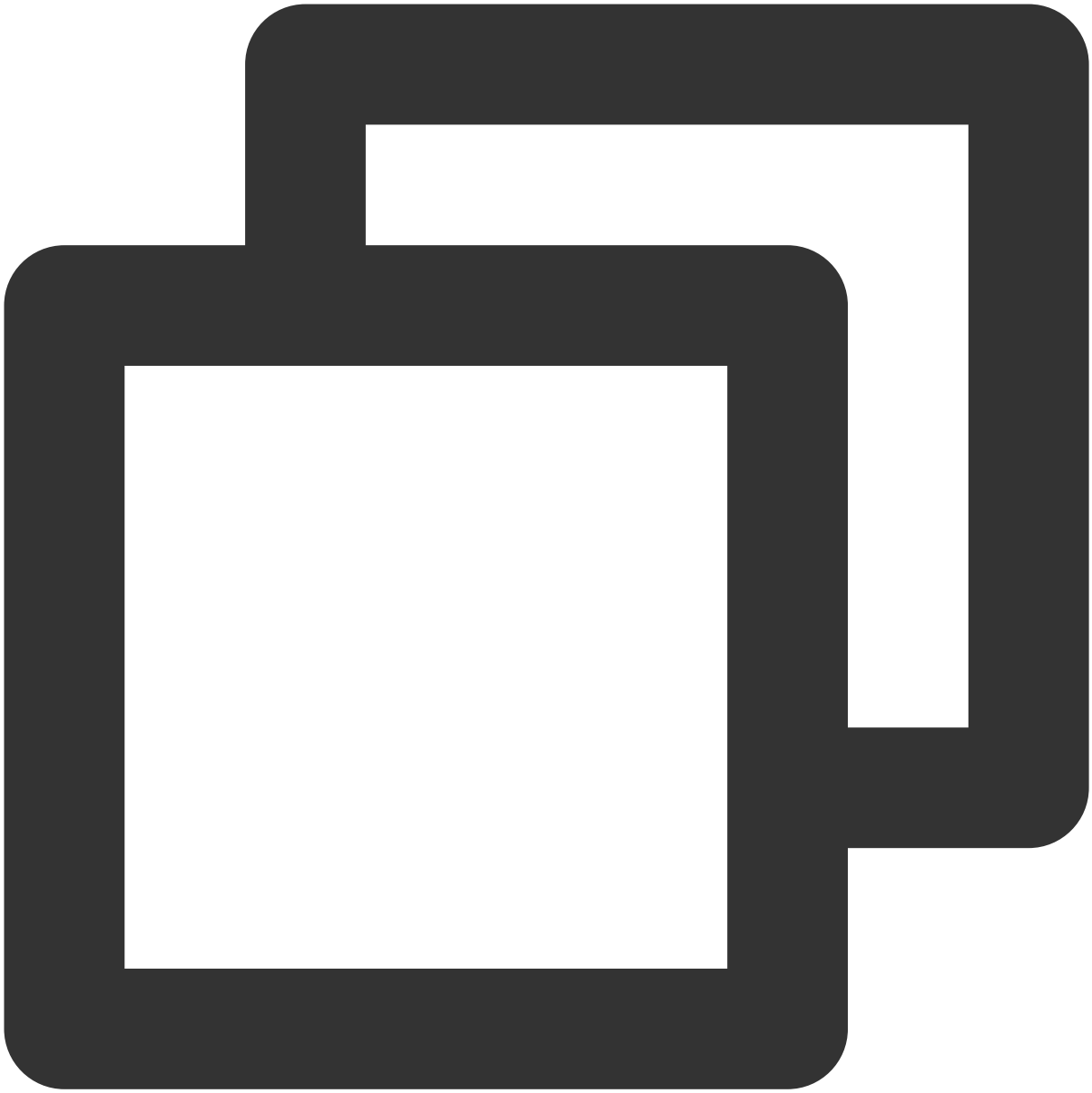
关于访问策略中的元素介绍，请参见 访问策略语言概述。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

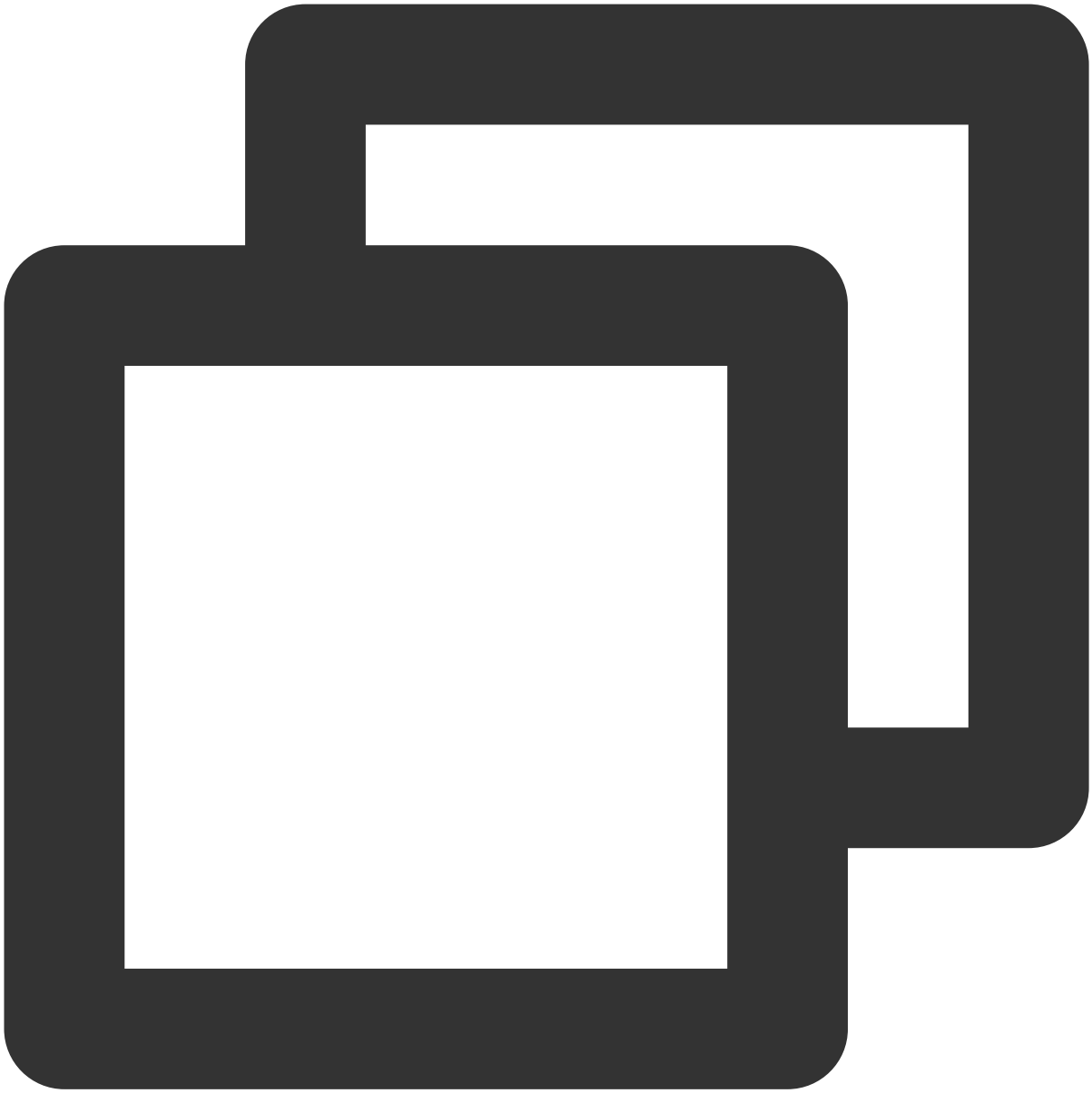
实际案例

请求



```
GET /?policy HTTP/1.1
Host:examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-sig
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 237
Connection: keep-alive
Date: Thu Jan 19 16:21:46 2017
Server: tencent-cos
x-cos-request-id: NTg4MDc3MWFfOWIxZjRlXzZmNDVf****

{
  "Statement": [
    {
```

```
      "Principal": {
        "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000001"
        ]
      },
      "Effect": "allow",
      "Action": [
        "name/cos:GetBucket"
      ],
      "Resource": [
        "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
      ]
    }
  ],
  "version": "2.0"
}
```

DELETE Bucket policy

最近更新时间：2023-12-21 11:38:00

功能描述

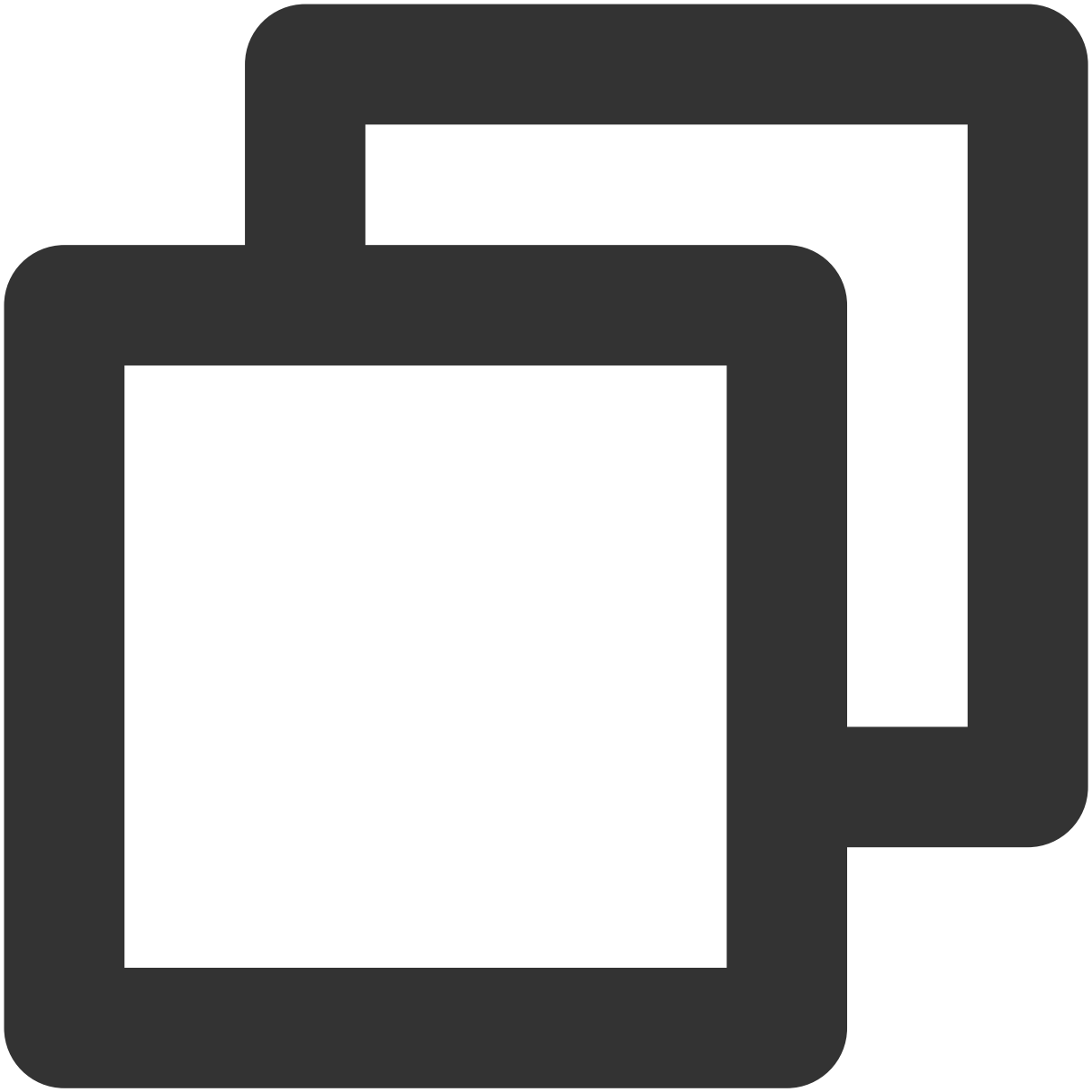
DELETE Bucket policy 请求可以向存储桶删除权限策略。

注意

只有存储桶所有者有权限发起该请求。如果权限策略不存在，将返回204 No Content。

请求

请求示例



```
DELETE /?policy HTTP/1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

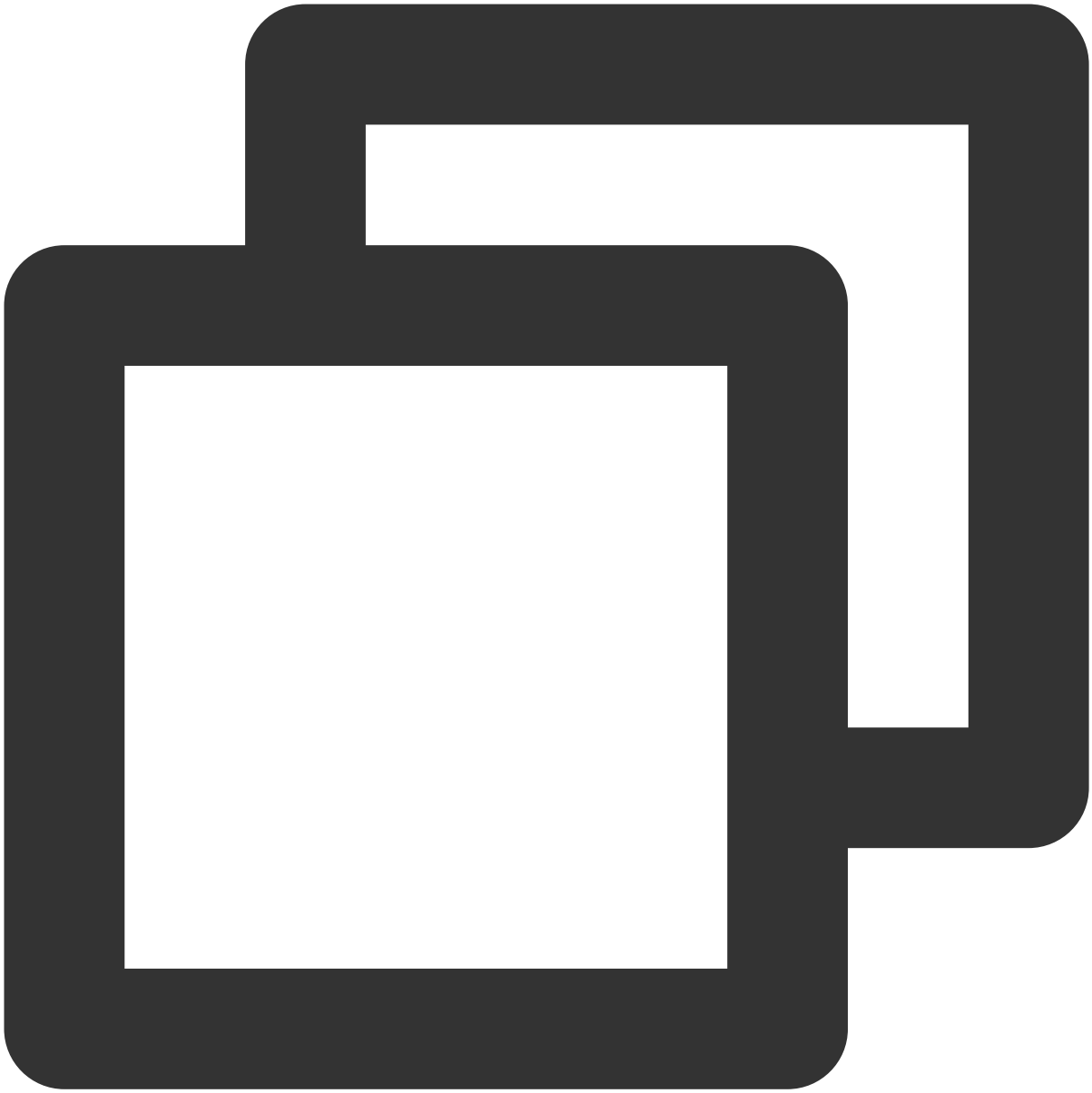
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

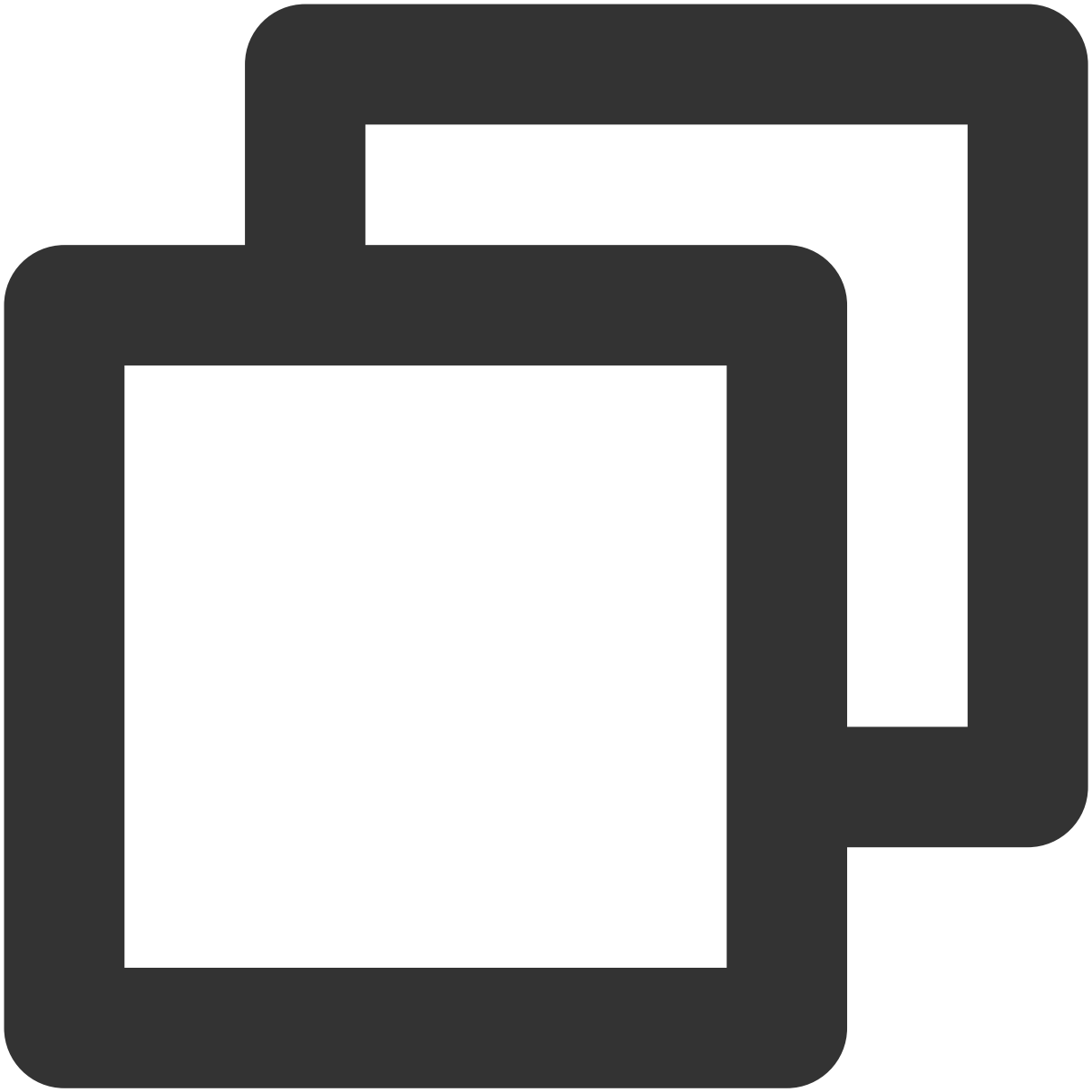
实际案例

请求



```
DELETE /?policy HTTP/1.1
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-sig
```

响应



```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu Jan 19 16:30:21 2017
Server: tencent-cos
x-cos-request-id: NTg4MDc5MWRfNDQyMDRlXzNiMDRf****
```

防盗链（referer）

PUT Bucket Referer

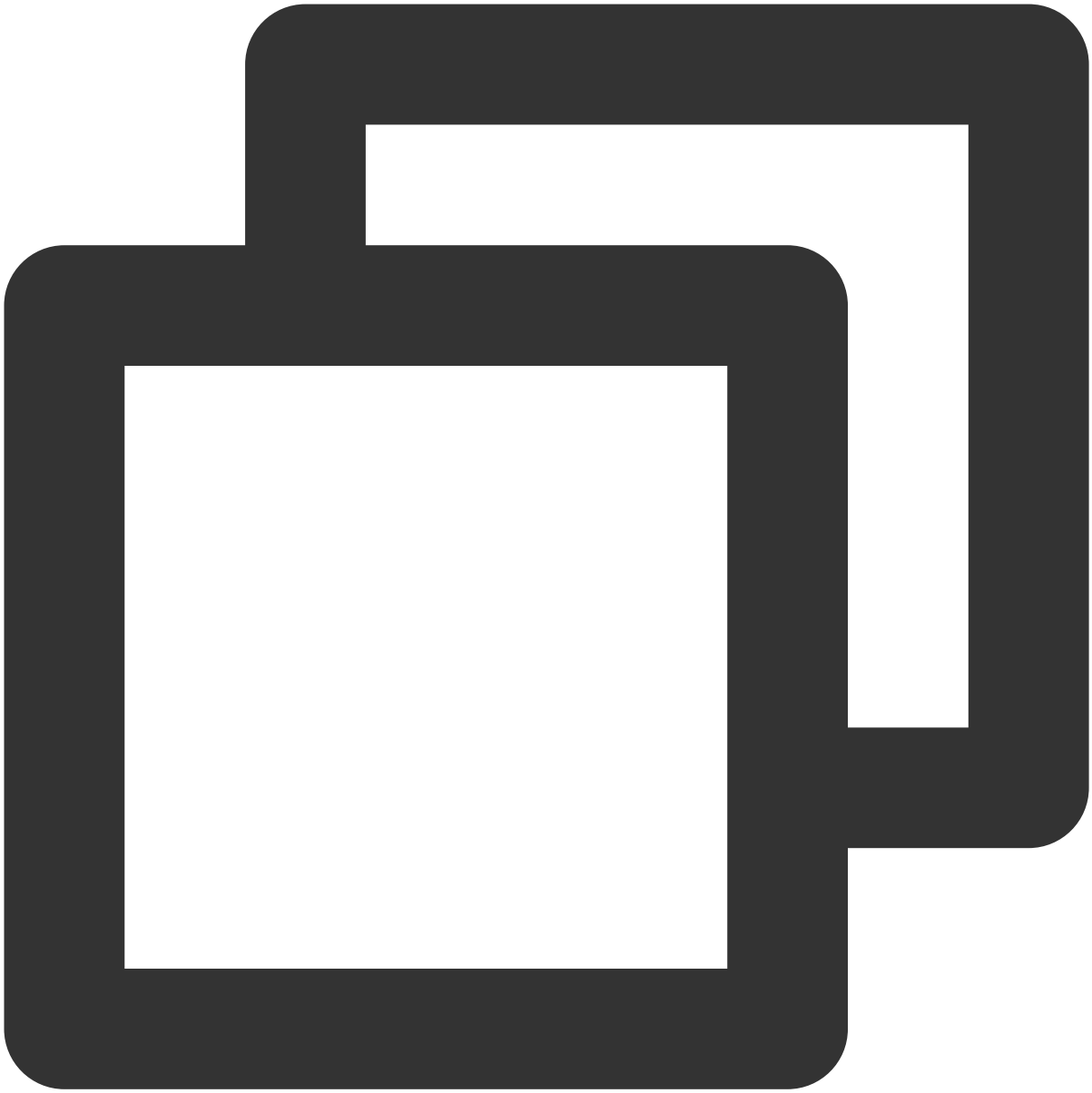
最近更新时间：2023-12-21 11:43:28

功能描述

PUT Bucket referer 接口用于为存储桶设置 Referer 白名单或者黑名单。

请求

请求示例



```
PUT /?referer HTTP 1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date:GMT Date
Authorization: Auth String
Content-Length:length
Content-MD5:MD5
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

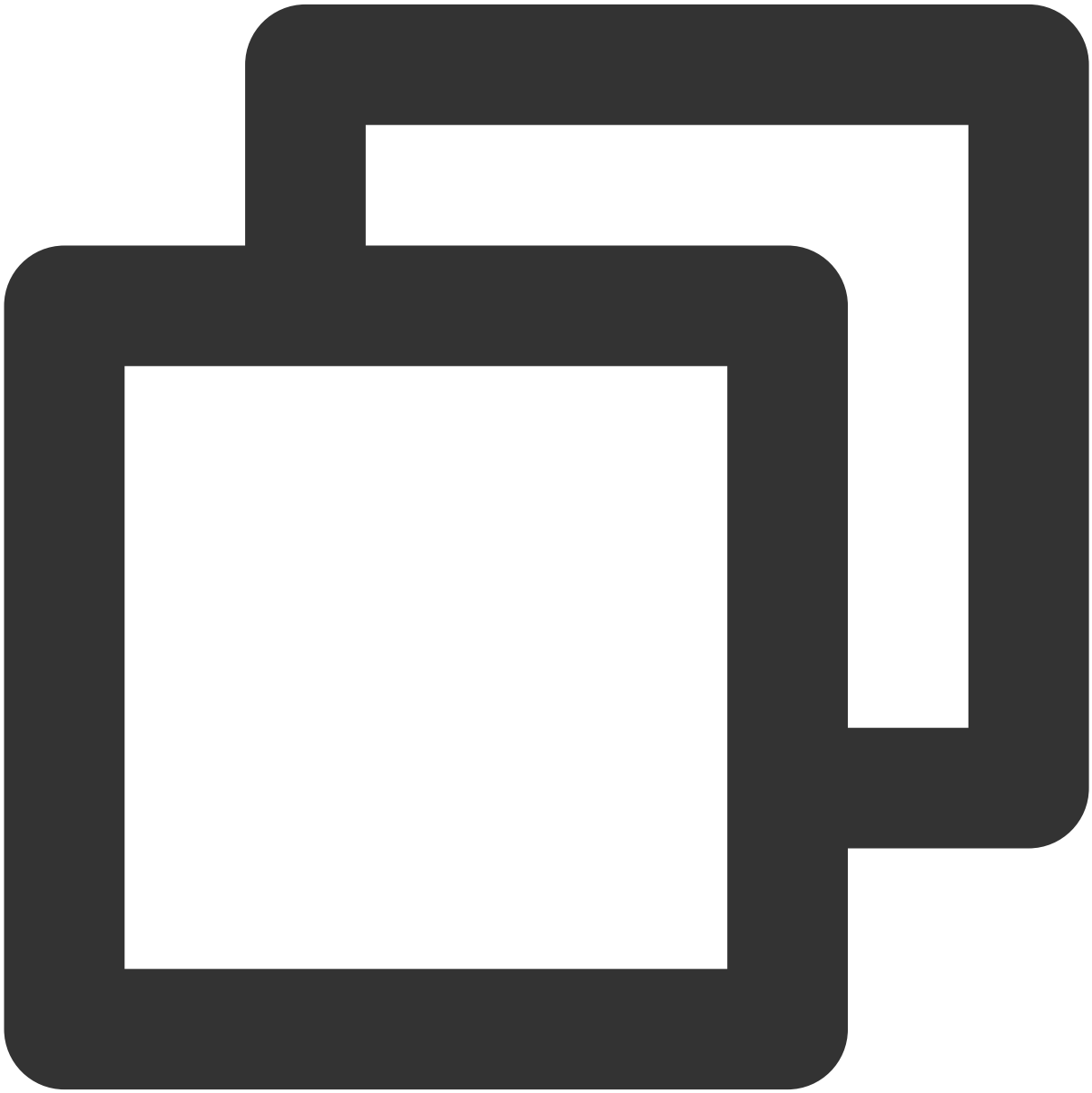
<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体具体节点内容为：



```
<RefererConfiguration>
```

```
    <Status>Enabled</Status>
    <RefererType>White-List</RefererType>
    <DomainList>
      <Domain>*.qq.com</Domain>
      <Domain>*.qcloud.com</Domain>
    </DomainList>
    <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
  </RefererConfiguration>
```

具体内容描述如下：

名称	父节点	描述
RefererConfiguration	无	防盗链配置信息
Status	RefererConfiguration	是否开启防盗链，枚举值：Enabled、Disabled
RefererType	RefererConfiguration	防盗链类型，枚举值：Black-List、White-List
DomainList	RefererConfiguration	生效域名列表， 支持多个域名且为前缀匹配， 支持带端口的域名和 IP， 支持通配符 `*` ，做二级域名或多级域名的通配
Domain	DomainList	单条生效域名 例 如 `www.qq.com/example` ， `192.168.1.2:8080` ， `*.qq.com`
EmptyReferConfiguration	RefererConfiguration	是否允许空 Referer 访问，枚举值：Allow、Deny，默认值为 Deny

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

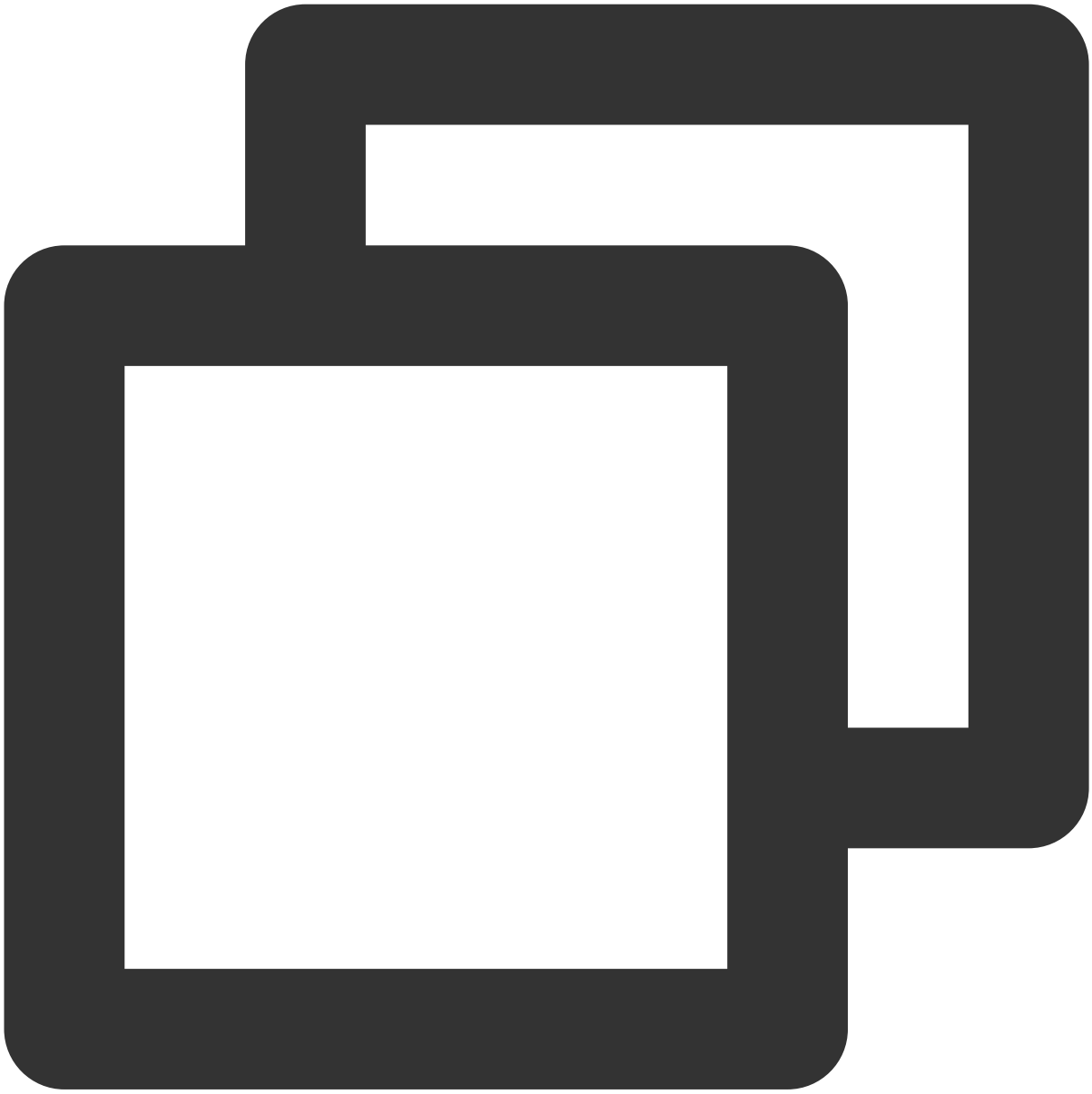
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

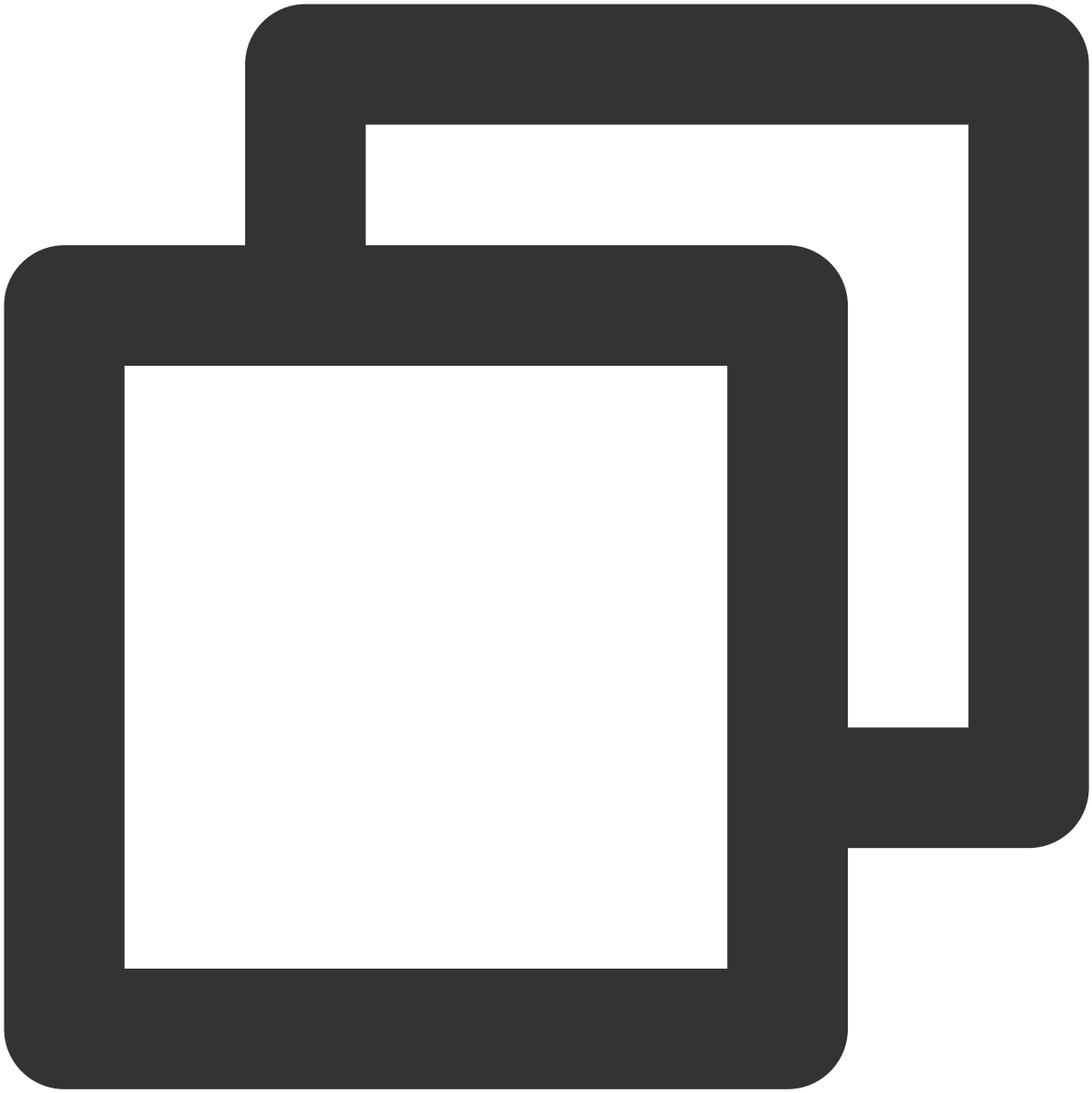
请求



```
PUT /?referer HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 25 Feb 2017 04:10:22 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Content-MD5: kOz7g54LMJZjxKs070V9jQ==
Content-Type: application/xml
```

```
<RefererConfiguration>
  <Status>Enabled</Status>
  <RefererType>White-List</RefererType>
  <DomainList>
    <Domain>*.qq.com</Domain>
    <Domain>*.qcloud.com</Domain>
  </DomainList>
  <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
</RefererConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Fri, 25 Feb 2017 04:10:22 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjFjMmJfOWIxZjRlXzZmNDhf****
```

GET Bucket Referer

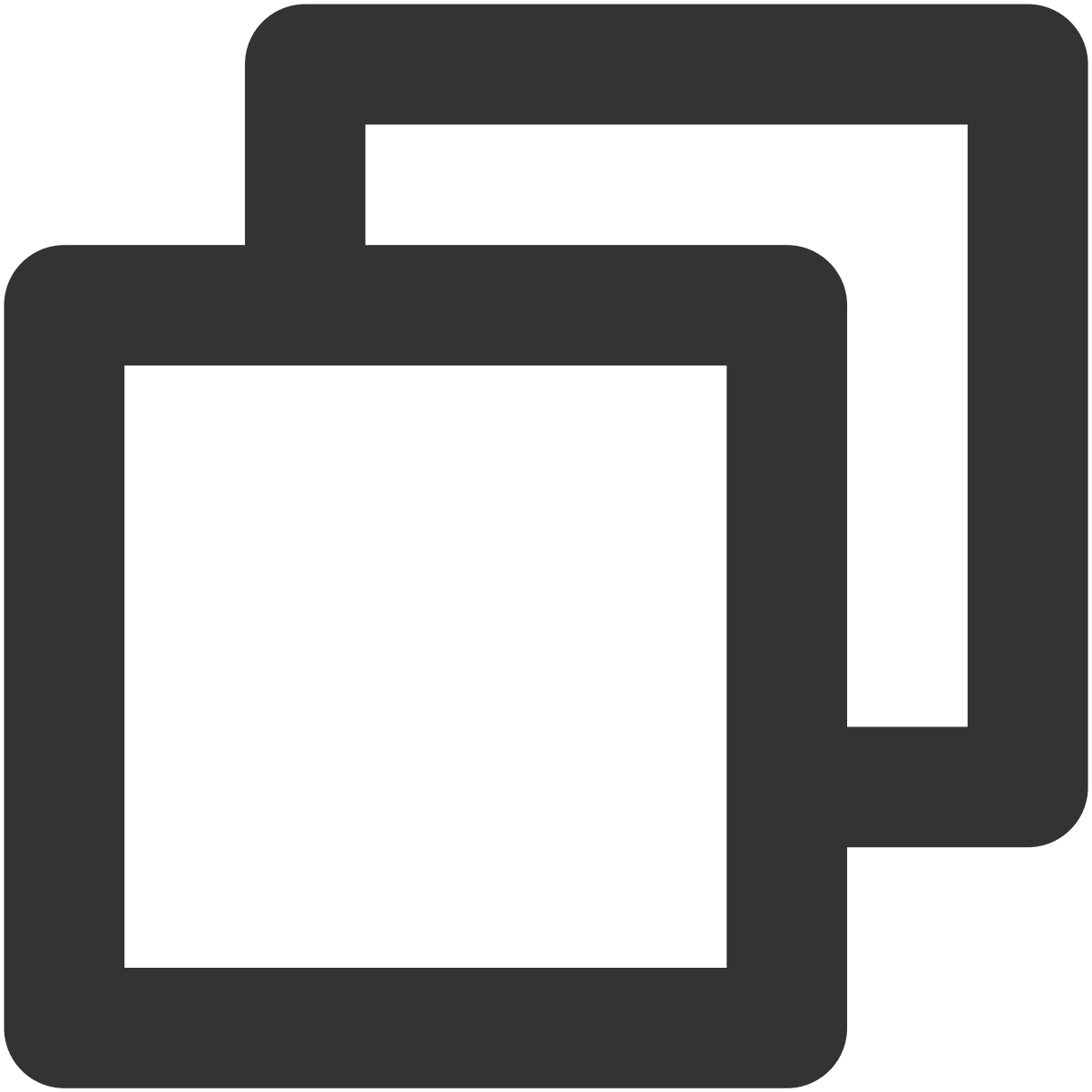
最近更新时间：2023-12-21 11:43:47

功能描述

GET Bucket referer 接口用于读取存储桶 Referer 白名单或者黑名单。

请求

请求示例



```
GET /?referer HTTP 1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

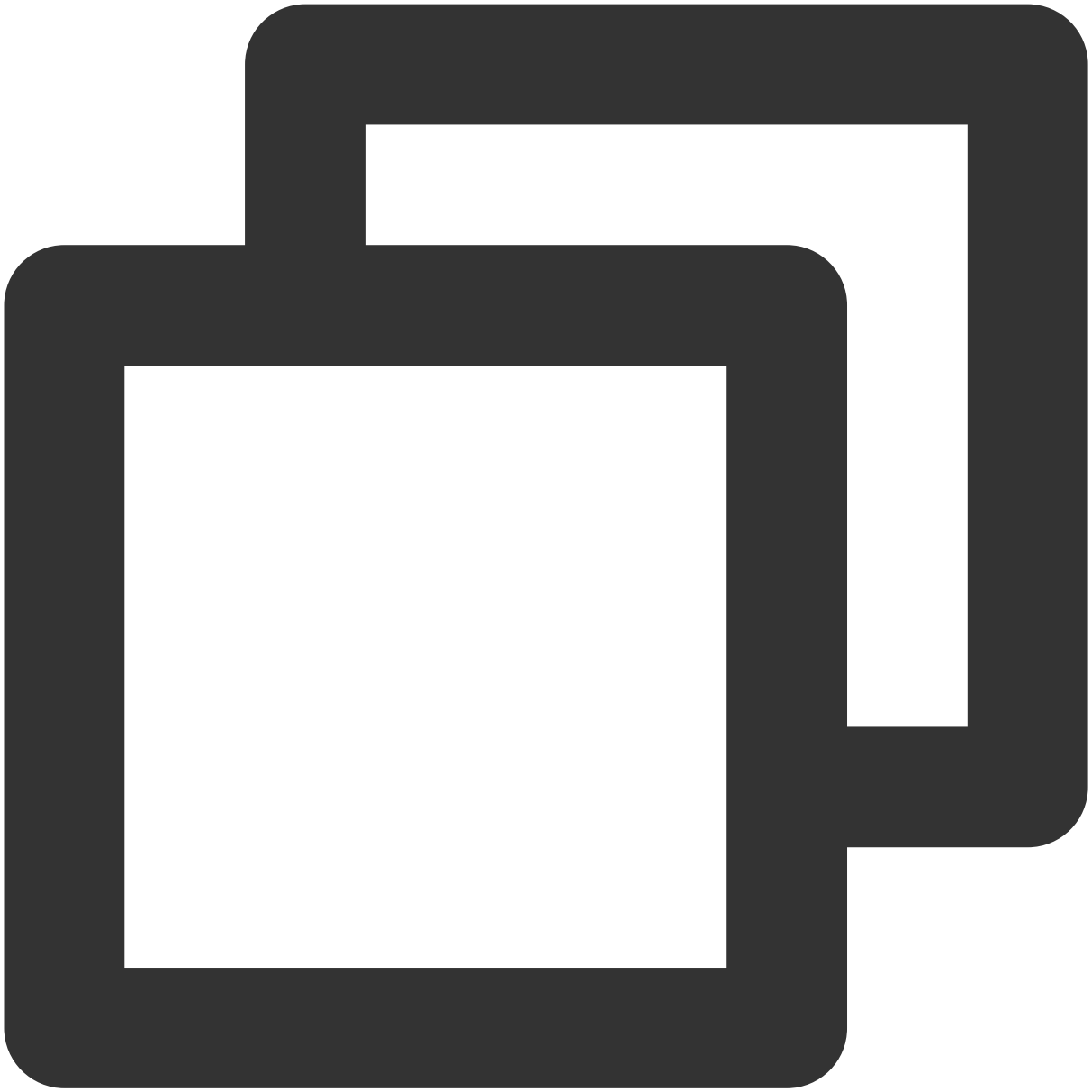
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 application/xml 数据，包含完整节点数据的内容展示如下：



```
<RefererConfiguration>
    <Status>Enabled</Status>
    <RefererType>White-List</RefererType>
    <DomainList>
        <Domain>*.qq.com</Domain>
        <Domain>*.qcloud.com</Domain>
    </DomainList>
    <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
</RefererConfiguration>
```

具体的节点描述如下：

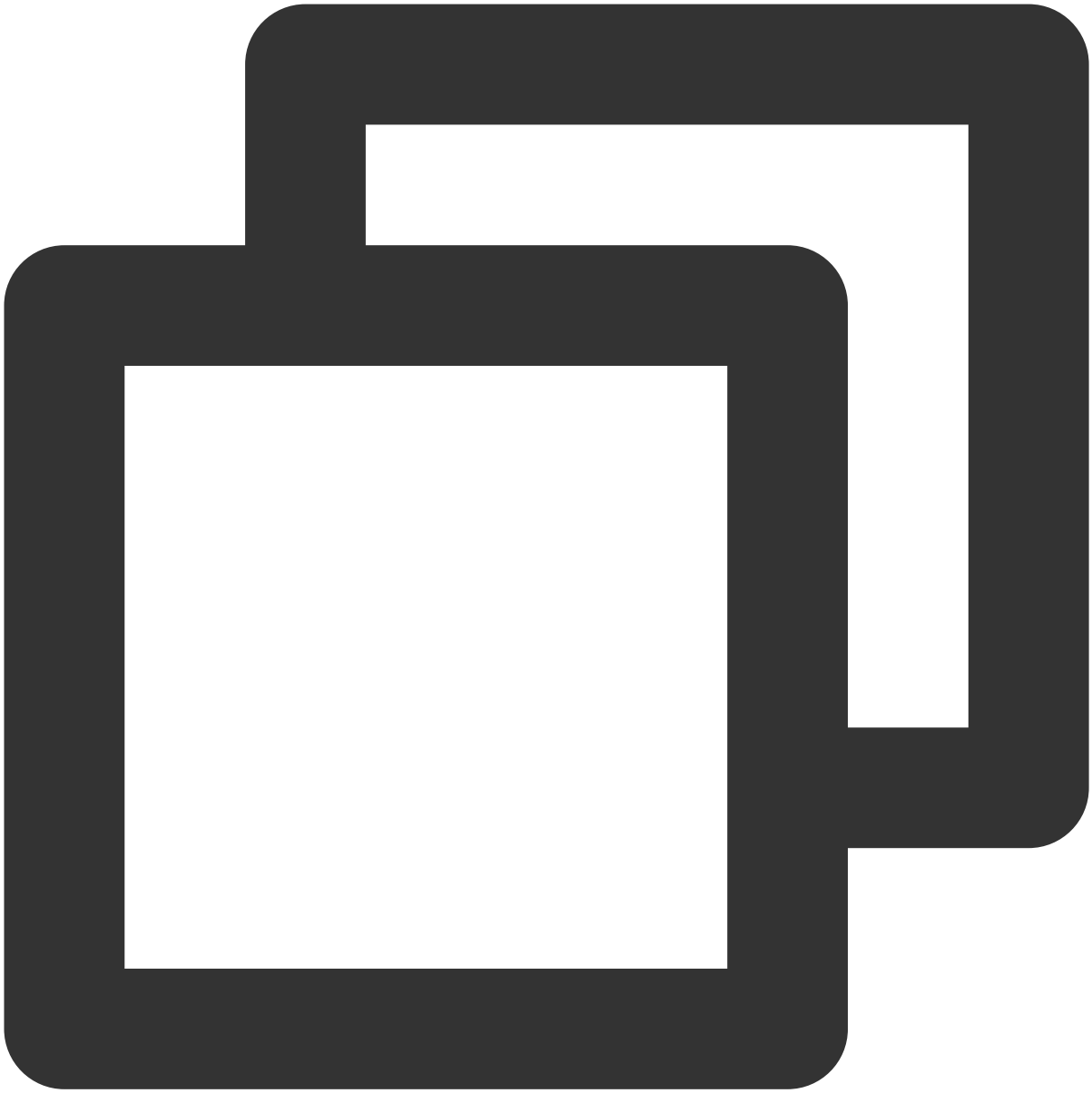
名称	父节点	描述
RefererConfiguration	无	防盗链配置信息
Status	RefererConfiguration	是否开启防盗链，枚举值：Eabled，Disabled
RefererType	RefererConfiguration	防盗链类型，枚举值：Black-List，White-List
DomainList	RefererConfiguration	生效域名列表。 支持多个域名且为前缀匹配， 支持带端口的域名和 IP， 支持通配符 `*` ，做二级域名或多级域名的通配
Domain	DomainList	单条生效域名，例如 `www.qq.com/example` ， `192.168.1.2:8080` ， `*.qq.com`
EmptyReferConfiguration	RefererConfiguration	是否允许空 Referer 访问，枚举值：Allow，Deny，默认值为 Deny

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

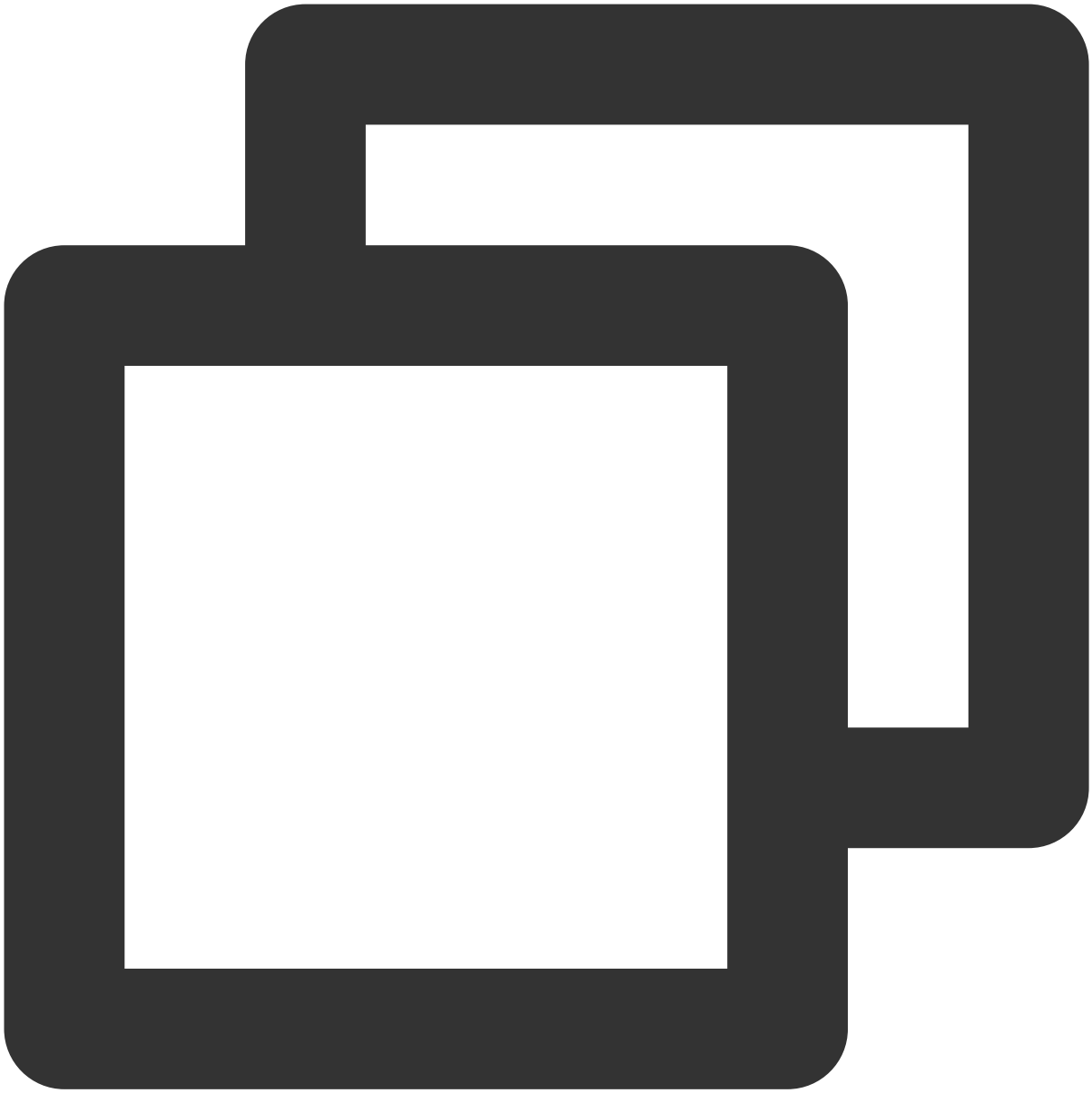
实际案例

请求



```
GET /?referer HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 25 Feb 2017 04:10:22 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 260
Connection: keep-alive
Date: Fri, 25 Feb 2017 04:10:22 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjFjMmJfOWIxZjRlXzZmNDhf****

<RefererConfiguration>
    <Status>Enabled</Status>
    <RefererType>White-List</RefererType>
```

```
    <DomainList>
        <Domain>*.qq.com</Domain>
        <Domain>*.qcloud.com</Domain>
    </DomainList>
    <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
</RefererConfiguration>
```

标签（tagging）

PUT Bucket tagging

最近更新时间：2023-12-21 11:46:06

功能描述

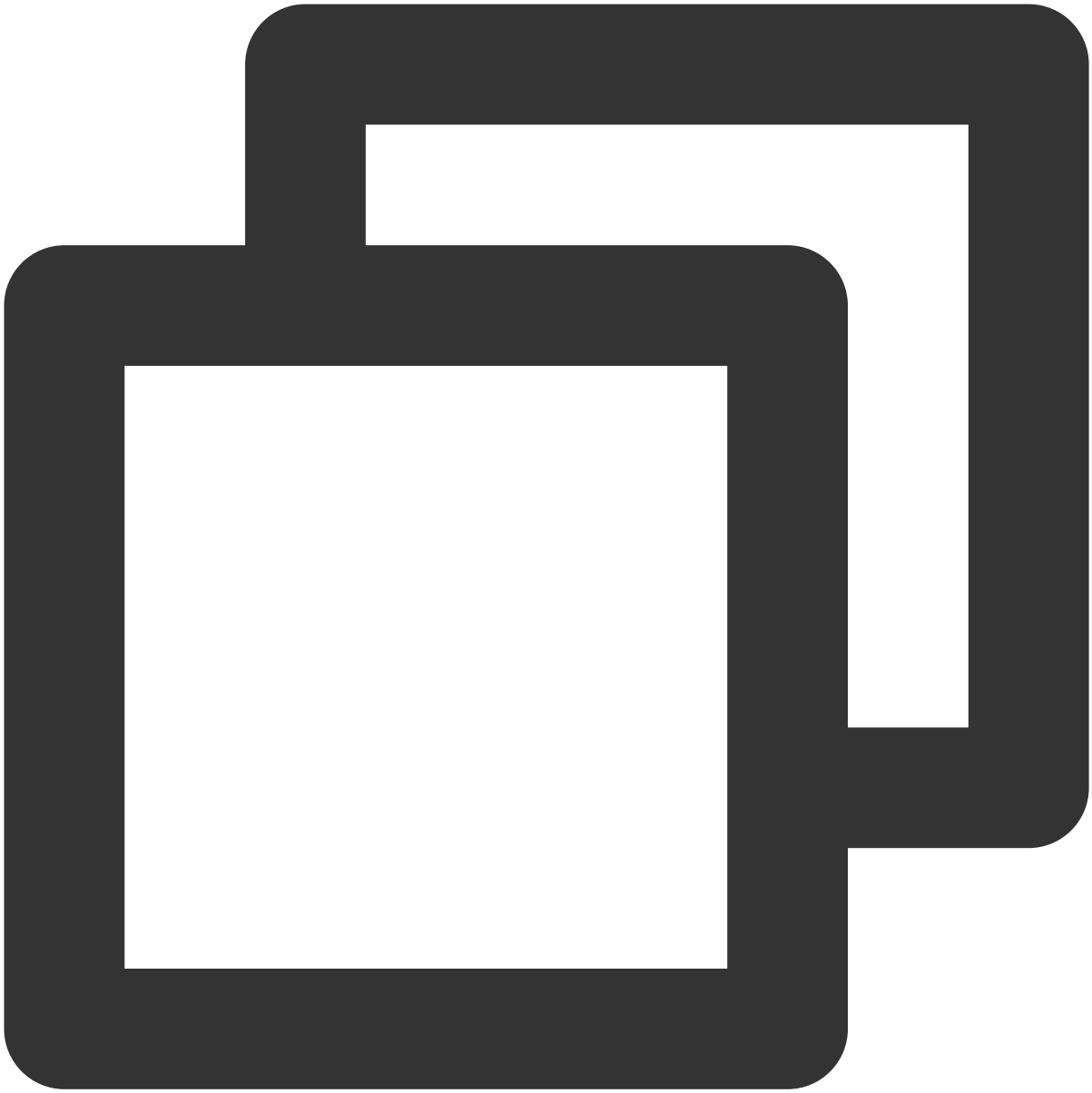
COS 支持为已存在的 Bucket 设置标签（Tag）。PUT Bucket tagging 接口用于为存储桶设置键值对作为存储桶标签，可以协助您管理已有的存储桶资源，并通过标签进行成本管理。

注意

目前存储桶标签功能最多支持一个存储桶下设置50个不同的标签。

请求

请求示例



```
PUT /?tagging HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Content-MD5: MD5
Content-Length: Content Length
Content-Type: application/xml

[Request Body]
```

说明

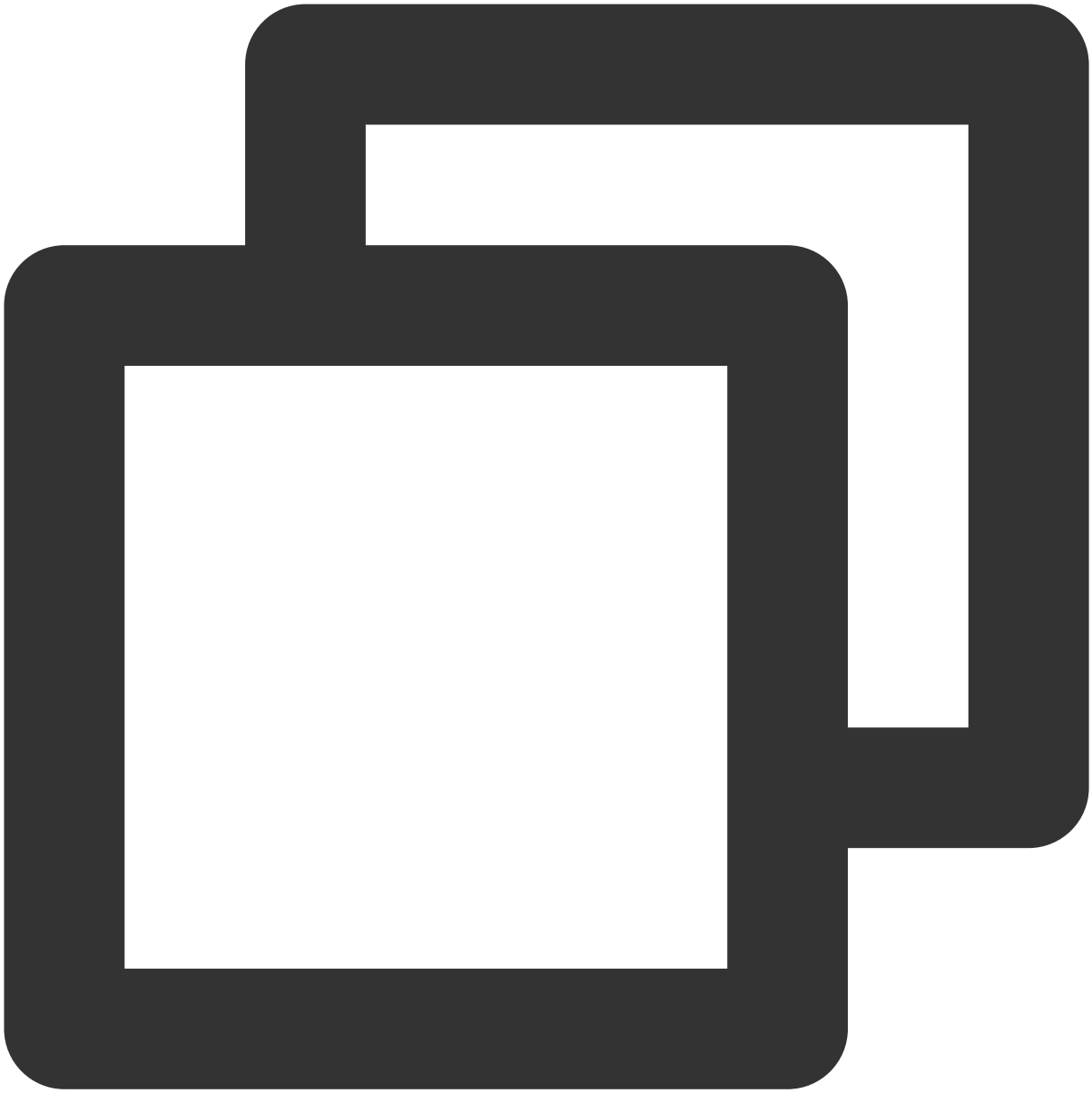
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求需要设置如下标签集合：



```
<?xml version="1.0" encoding="UTF-8" ?>
<Tagging>
    <TagSet>
        <Tag>
            <Key>age</Key>
            <Value>18</Value>
        </Tag>
        <Tag>
            <Key>name</Key>
            <Value>xiaoming</Value>
        </Tag>
```

```
    </TagSet>
</Tagging>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tagging	无	标签集合	Container	是
TagSet	Tagging	标签集合	Container	是
Tag	Tagging.TagSet	标签集合，最多支持50个标签	Containers	是
Key	Tagging.TagSet.Tag	标签的 Key，长度小于128字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String	是
Value	Tagging.TagSet.Tag	标签的 Value，长度小于256字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String	是

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体为空。

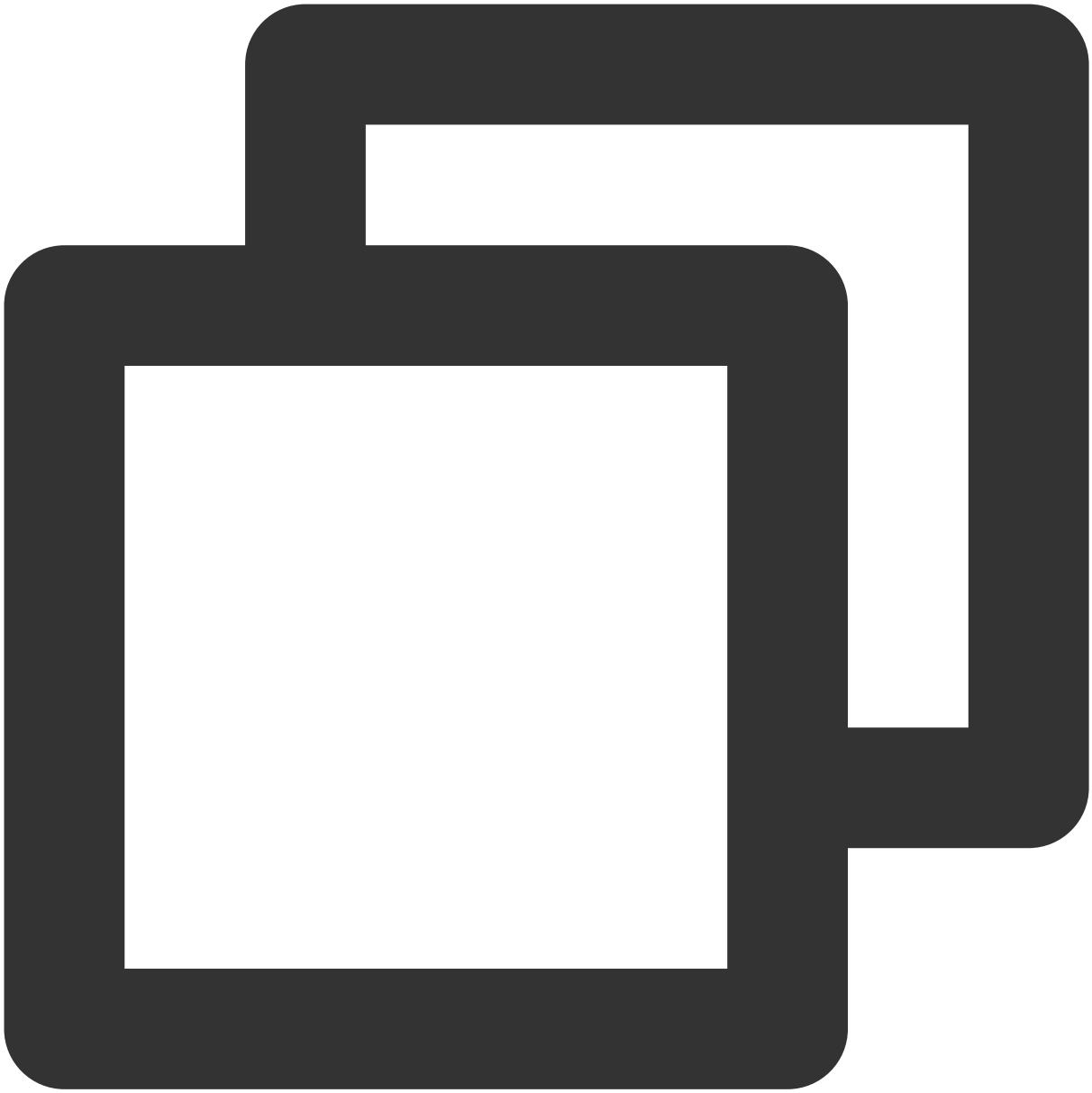
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

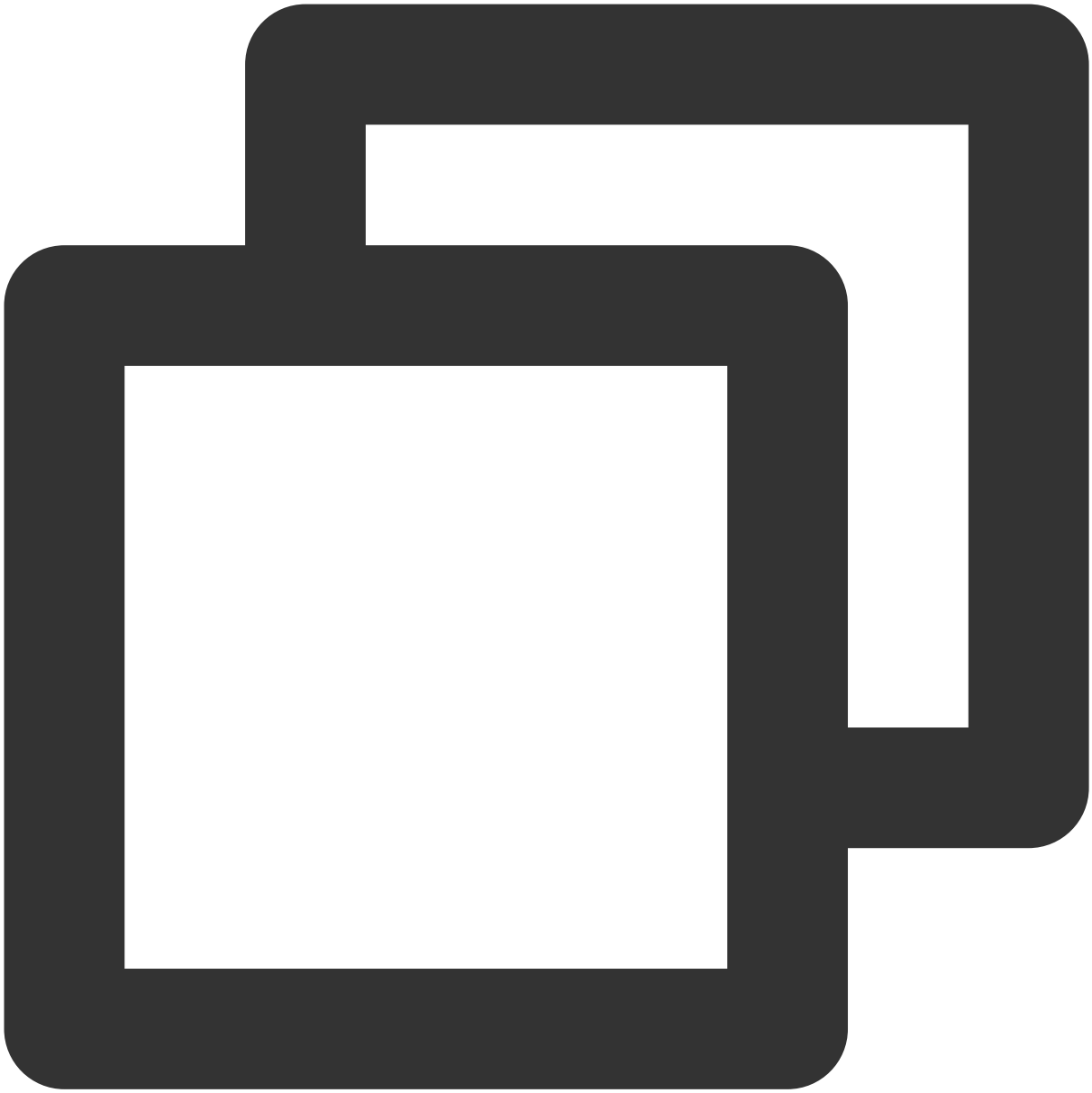
如下请求向存储桶 `examplebucket-1250000000` 中写入了{age:18}和{name:xiaoming}两个标签。COS 配置标签成功并返回204成功请求。



```
PUT /?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDrbAYjEBqqdEconpFi8NPFsOjrnX4****&q-si
Content-MD5: LIbd5t5HLPhuNWYkP6qHcQ==
Content-Length: 127
Content-Type: application/xml
```

```
<Tagging>
    <TagSet>
        <Tag>
            <Key>age</Key>
            <Value>18</Value>
        </Tag>
        <Tag>
            <Key>name</Key>
            <Value>xiaoming</Value>
        </Tag>
    </TagSet>
</Tagging>
```

响应



```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 19 Jan 2018 11:40:22 GMT
Server: tencent-cos
x-cos-request-id: NWE2MWQ5MjZfMTBhYzM1MGFfMTA5ODVfMTVj****
```

GET Bucket tagging

最近更新时间：2023-12-21 11:46:11

功能描述

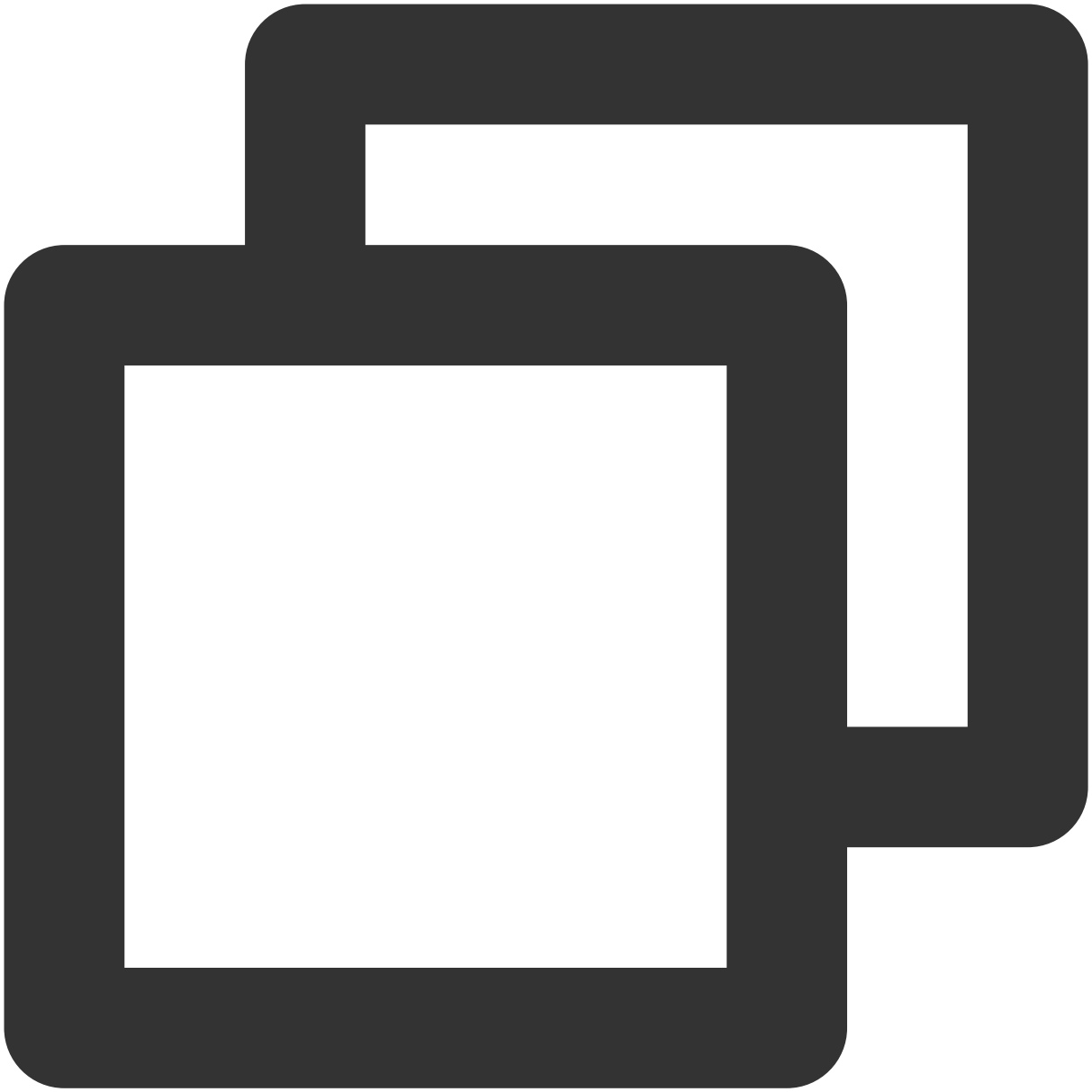
COS 支持为已存在的存储桶查询标签（Tag）。GET Bucket tagging 接口用于查询指定存储桶下已有的存储桶标签。

说明

若您使用子账号调用此项接口，请确保您已经在主账号处获取了 GET Bucket tagging 这个接口的权限。

请求

请求示例



```
GET /?tagging HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

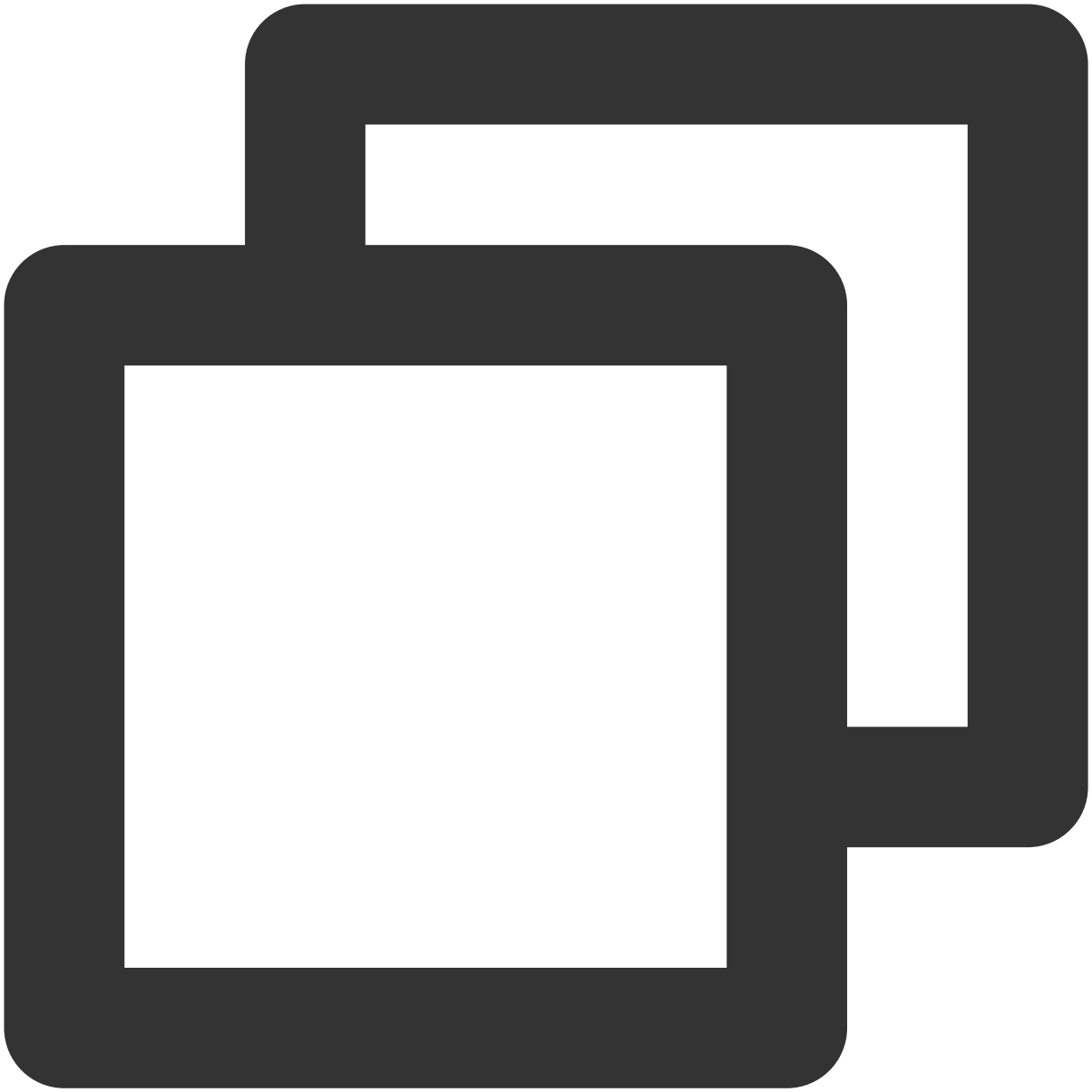
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 application/xml 数据，包含存储桶下已有的标签信息。



```
<Tagging>
    <TagSet>
        <Tag>
            <Key>string</Key>
            <Value>string</Value>
        </Tag>
        <Tag>
            <Key>string</Key>
            <Value>string</Value>
        </Tag>
    </TagSet>
```

```
</Tagging>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
Tagging	无	标签集合	Container
TagSet	Tagging	标签集合	Container
Tag	Tagging.TagSet	标签集合, 最多支持50个标签	Containers
Key	Tagging.TagSet.Tag	标签键, 长度小于128字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String
Value	Tagging.TagSet.Tag	标签值, 长度小于256字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String

错误码

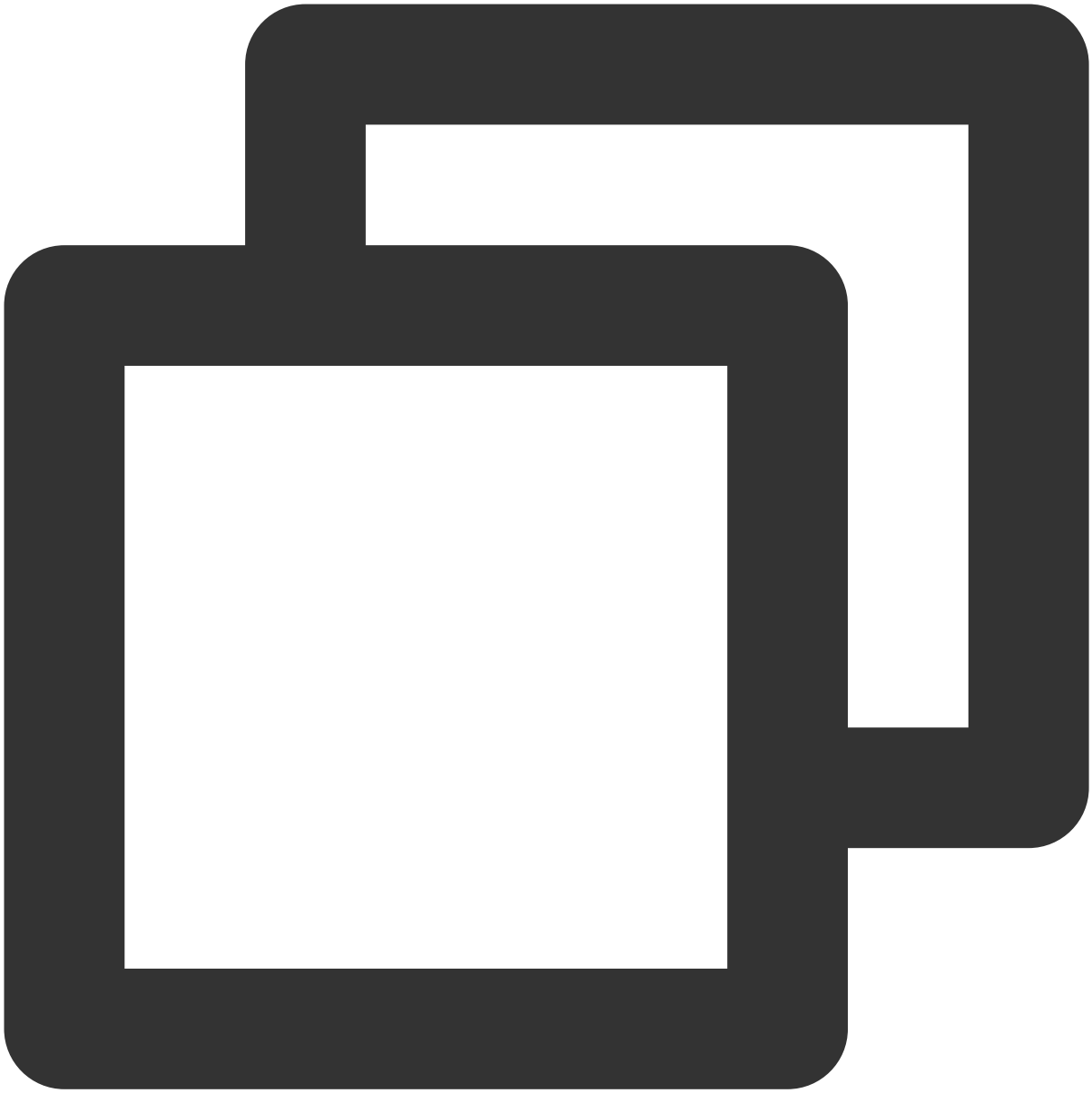
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

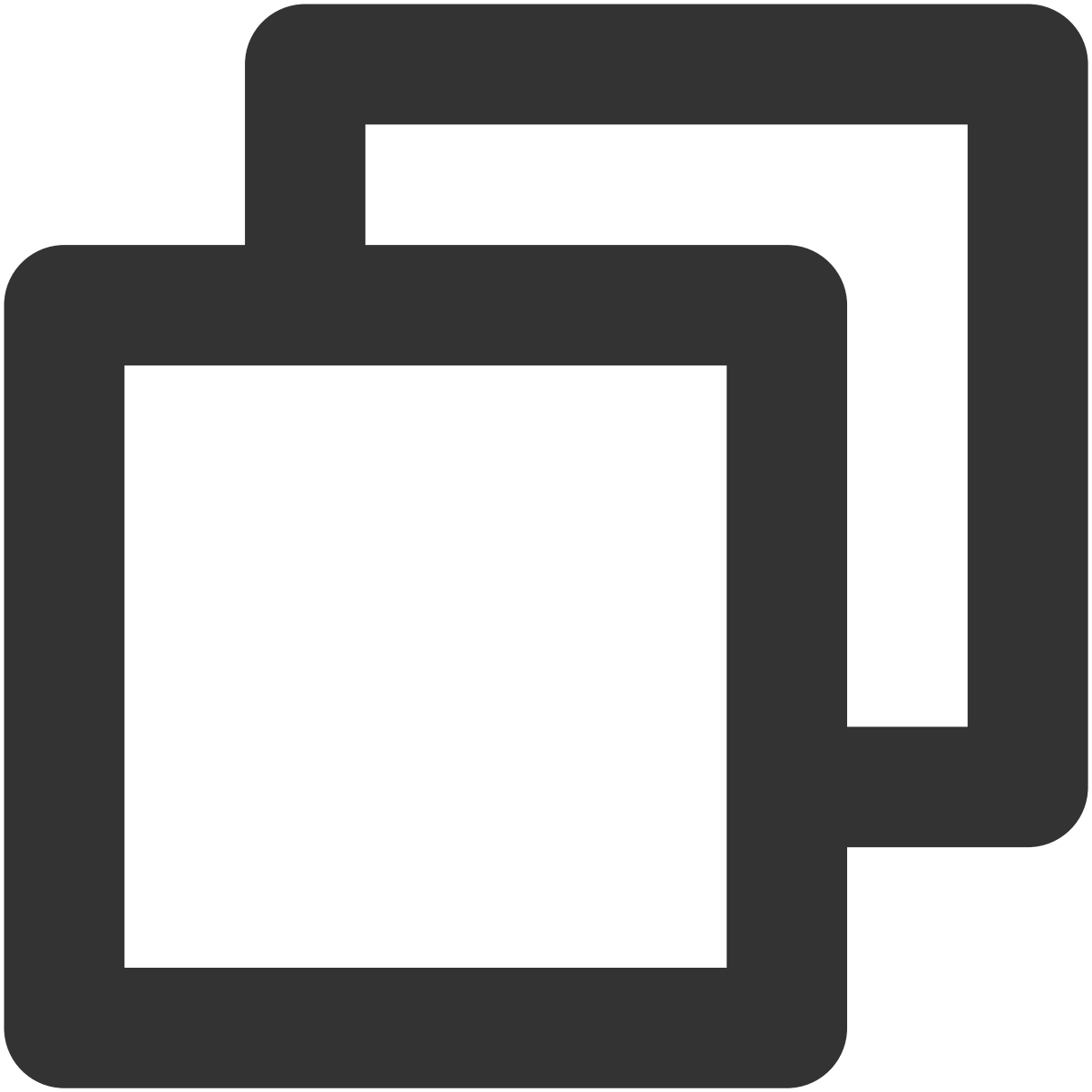
如下请求申请查询存储桶 `examplebucket-1250000000` 下的标签信息，COS 解析该请求后，并返回了该存储桶下已有的 {age:18} 和 {name:xiaoming} 两个标签。

版权所有：腾讯云计算（北京）有限责任公司



```
GET /?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDrbAYjEBqqdEconpFi8NPFsOjrnX4****&q-si
Content-Length: 127
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2018 11:40:22 GMT
Server: tencent-cos
<Tagging>
    <TagSet>
        <Tag>
            <Key>age</Key>
            <Value>18</Value>
        </Tag>
```

```
        <Tag>
            <Key>name</Key>
            <Value>xiaoming</Value>
        </Tag>
    </TagSet>
</Tagging>
```

DELETE Bucket tagging

最近更新时间：2023-12-21 11:46:33

功能描述

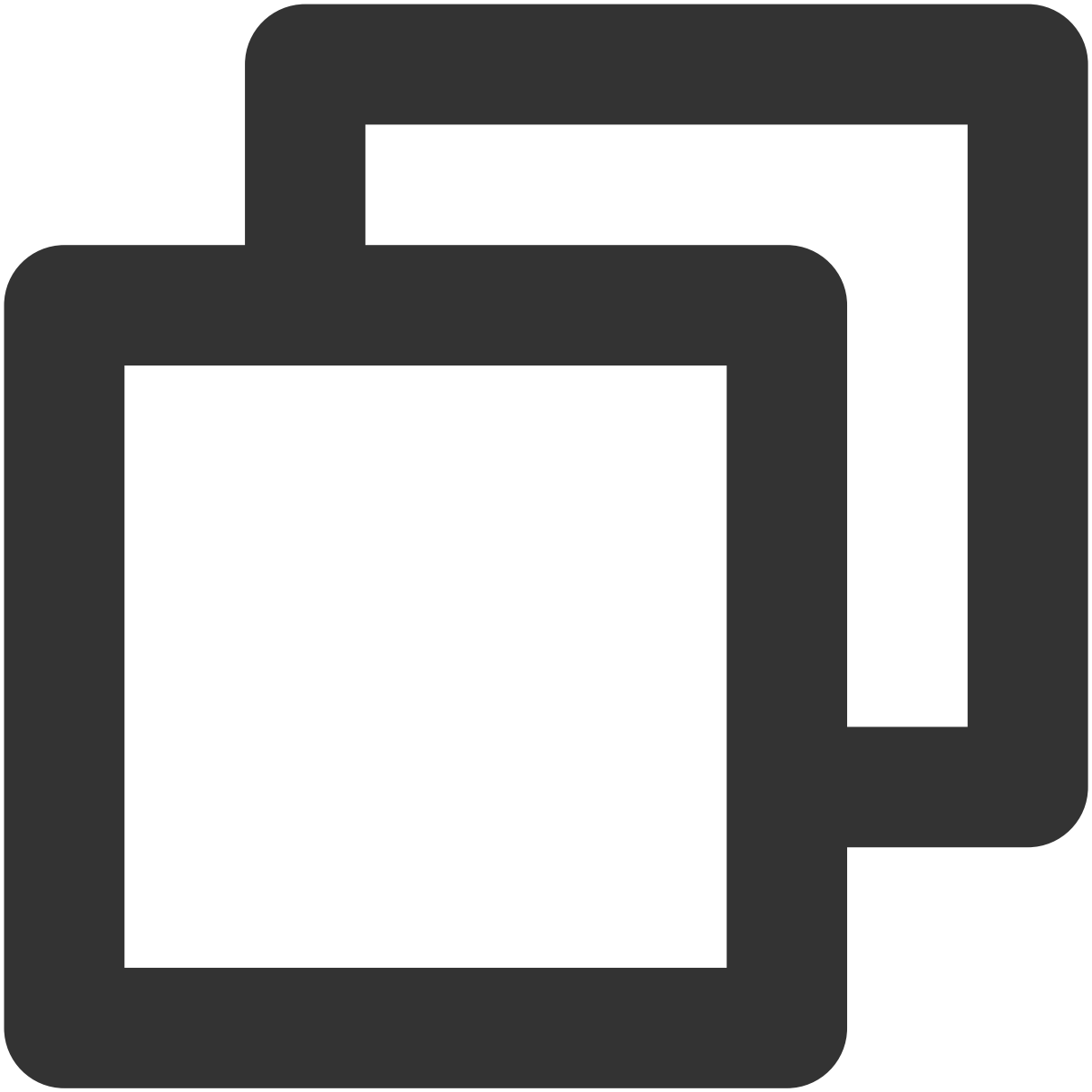
COS 支持为已存在的 Bucket 删除标签（Tag）。DELETE Bucket tagging 接口用于删除指定存储桶下已有的存储桶标签。

说明

如您使用子账号调用此项接口，请确保您已经在主账号处获取了 `DELETE Bucket tagging` 这个接口的权限。

请求

请求示例



```
DELETE /?tagging HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体为空。

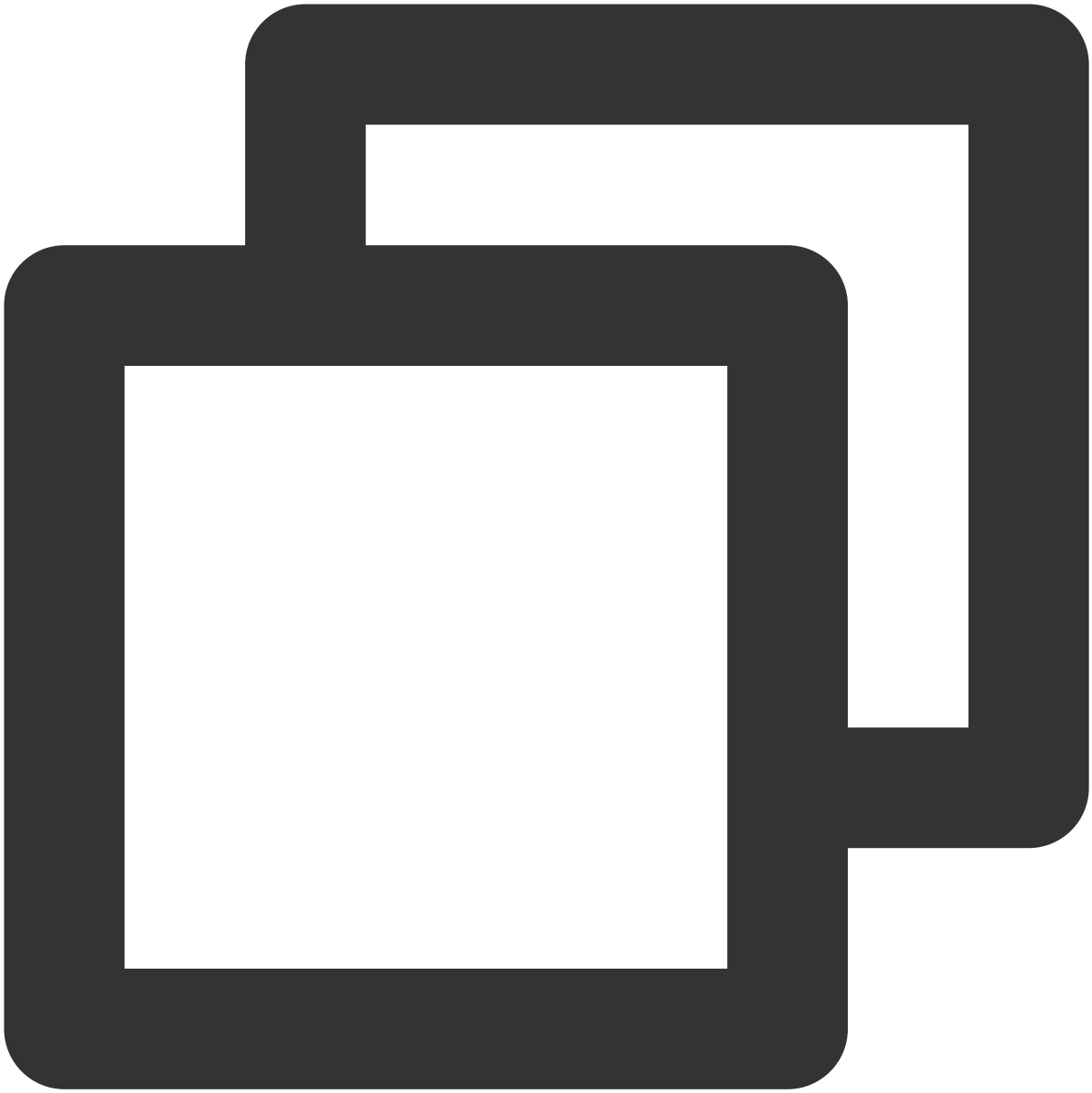
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

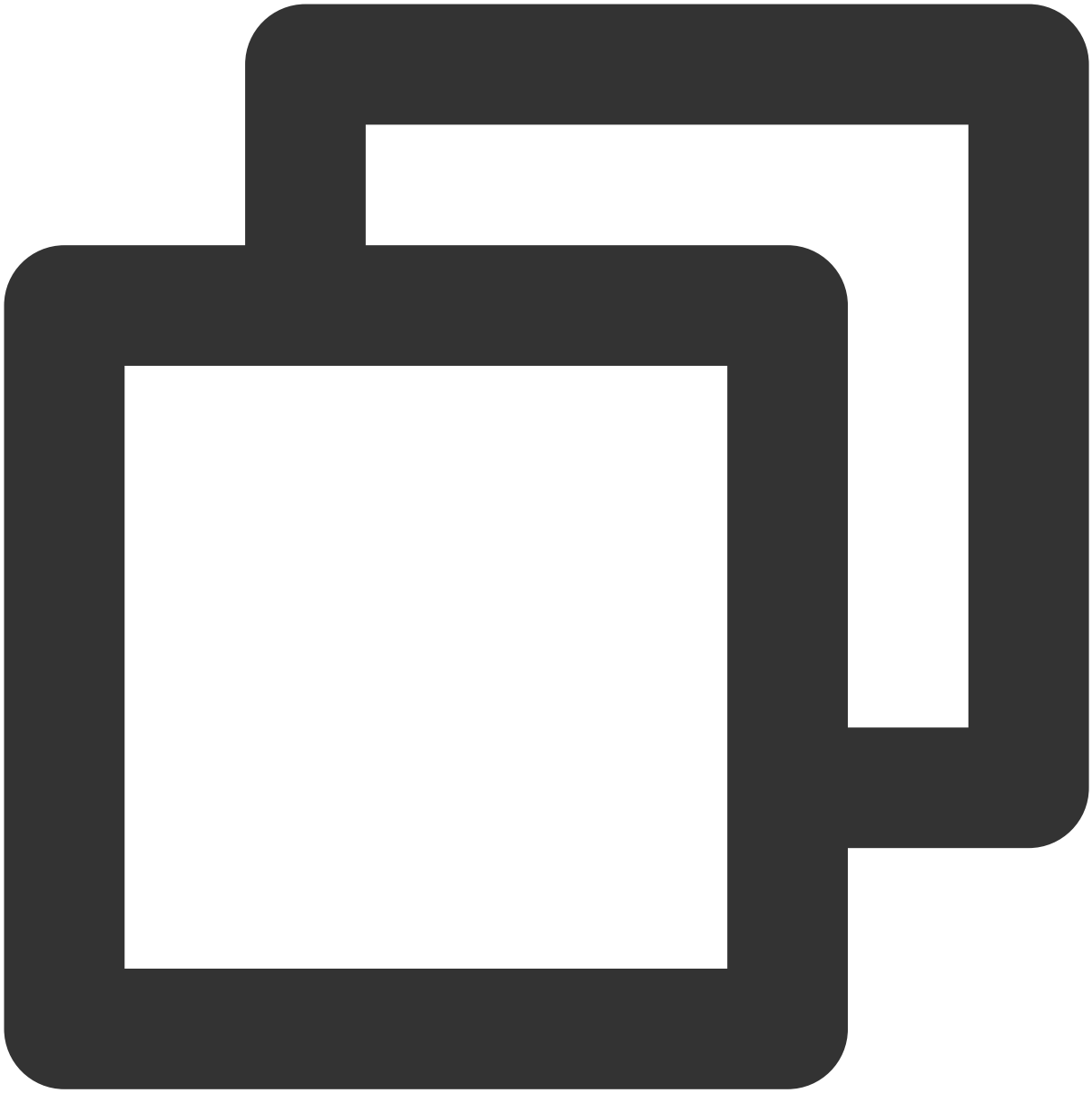
请求

如下请求申请删除存储桶 `examplebucket-1250000000` 下已有的标签信息。COS 解析该请求后删除该存储桶下所有标签。



```
DELETE /?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDrbAYjEBqqdEconpFi8NPFsOjrnX4****&q-si
Content-Md5: LIbd5t5HLPhuNWYkP6qHcQ==
Content-Length: 127
Content-Type: application/xml
```

响应



```
HTTP/1.1 204 No Content
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2018 11:40:22 GMT
```

静态网站（website）

PUT Bucket website

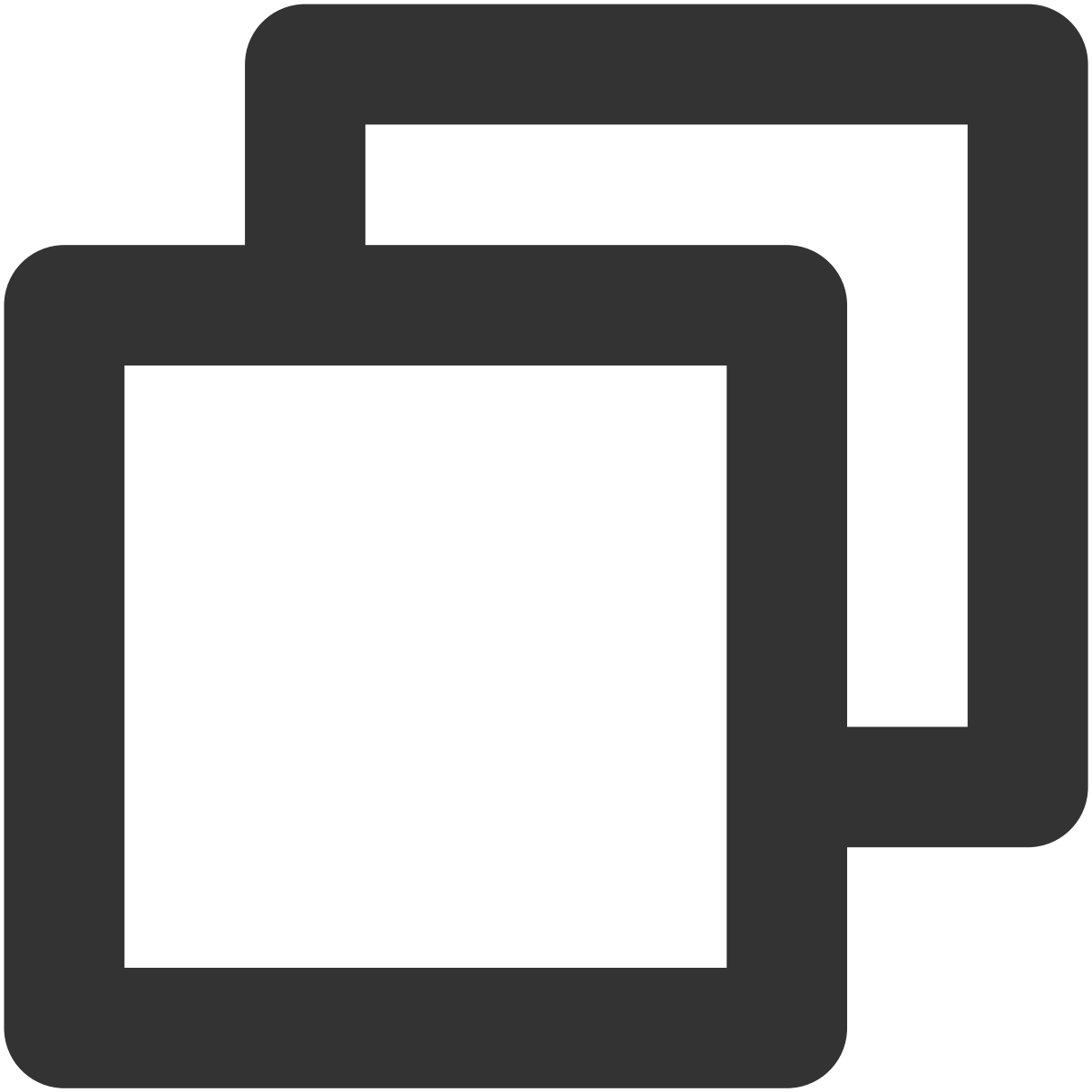
最近更新时间：2023-12-21 11:51:39

功能描述

PUT Bucket website 请求用于为存储桶配置静态网站，您可以通过传入 XML 格式的配置文件进行配置，文件大小限制为64KB。

请求

请求示例



```
PUT /?website HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

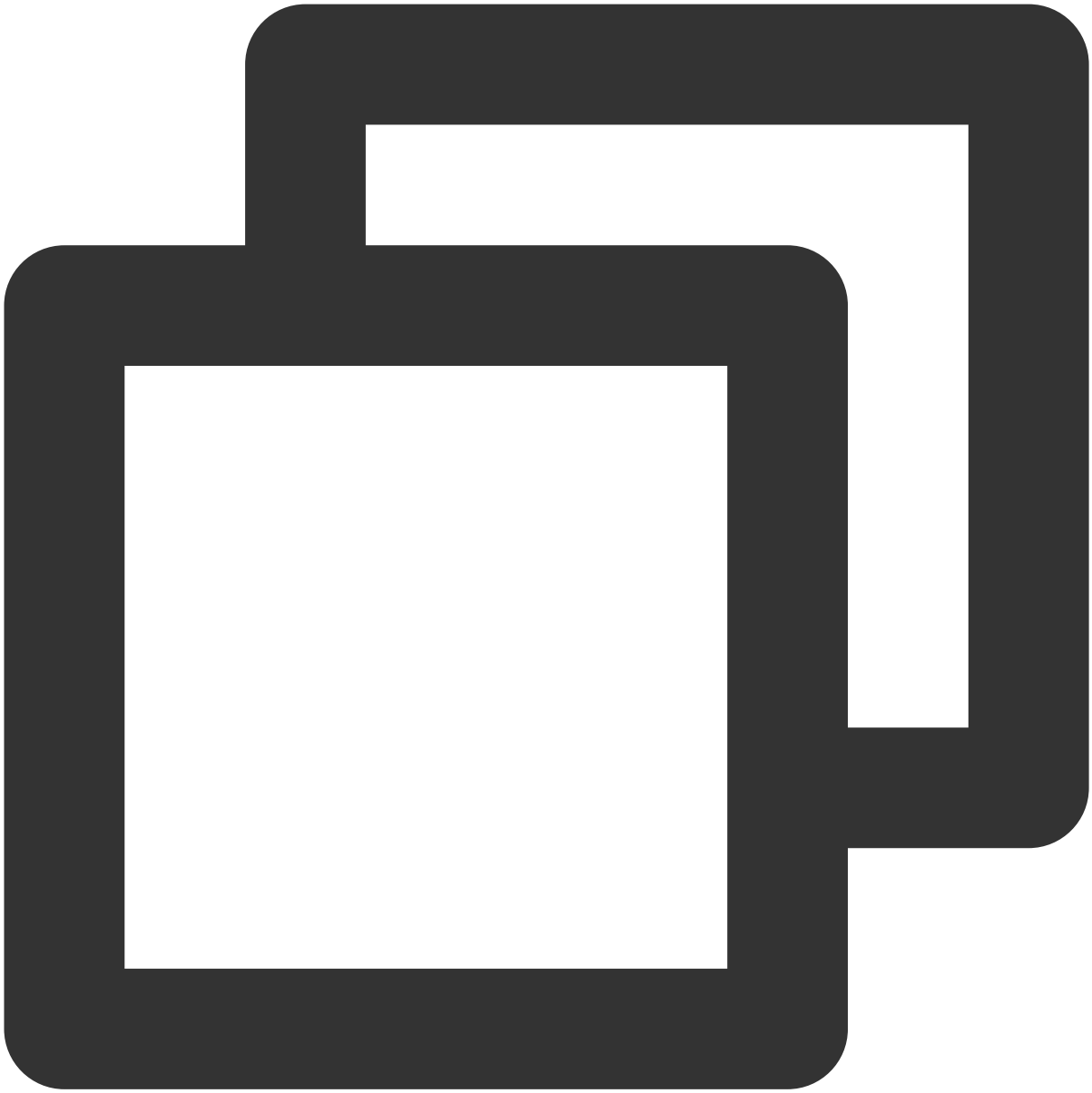
此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

提交 **application/xml** 请求数据，包含完整的存储桶静态网站配置信息。



```
<WebsiteConfiguration>
    <IndexDocument>
        <Suffix>String</Suffix>
    </IndexDocument>
    <RedirectAllRequestsTo>
        <Protocol>String</Protocol>
    </RedirectAllRequestsTo>
    <AutoAddressing>
        <Status>Enabled|Disabled</Status>
    </AutoAddressing>
    <ErrorDocument>
```

```
        <Key>String</Key>
        <OriginalHttpStatus>Enabled|Disabled</OriginalHttpStatus>
    </ErrorDocument>
    <RoutingRules>
        <RoutingRule>
            <Condition>
                <HttpErrorCodeReturnedEquals>Integer</HttpErrorCodeReturnedEquals>
            </Condition>
            <Redirect>
                <Protocol>String</Protocol>
                <ReplaceKeyWith>String</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <KeyPrefixEquals>String</KeyPrefixEquals>
            </Condition>
            <Redirect>
                <Protocol>String</Protocol>
                <ReplaceKeyPrefixWith>String</ReplaceKeyPrefixWith>
            </Redirect>
        </RoutingRule>
    </RoutingRules>
</WebsiteConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
WebsiteConfiguration	无	包含 PUT Bucket website 操作的所有请求信息	Container	否

Container 节点 WebsiteConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
IndexDocument	WebsiteConfiguration	索引文档配置	Container	是
RedirectAllRequestsTo	WebsiteConfiguration	重定向所有请求配置	Container	否
AutoAddressing	WebsiteConfiguration	用于配置是否忽略扩展名	Container	否
ErrorDocument	WebsiteConfiguration	错误文档配置	Container	否
RoutingRules	WebsiteConfiguration	重定向规则配置，最多设置100条 RoutingRule	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 IndexDocument 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Suffix	WebsiteConfiguration.IndexDocument	指定索引文档的对象键后缀。例如指定为 `index.html`，那么当访问到存储桶的根目录时，会自动返回 index.html 的内容，或者当访问到 `article/` 目录时，会自动返回 `article/index.html` 的内容	String	是

Container 节点 RedirectAllRequestsTo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Protocol	WebsiteConfiguration.RedirectAllRequestsTo	指定重定向所有请求的目标协议，只能设置为 https	String	是

Container 节点 AutoAddressing 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Status	WebsiteConfiguration.AutoAddressing	用于配置是否忽略 HTML 拓展名，可选值为 Enabled 或 Disabled，默认为 Disabled	String	否

Container 节点 ErrorDocument 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Key	WebsiteConfiguration.ErrorDocument	指定通用错误文档的对象键，当发生错误且未命中重定向规则中的错误码重定向时，将返回该对象键的内容	String	是
OriginalHttpStatus	WebsiteConfiguration.ErrorDocument	用于配置命中错误文档的	String	否

		HTTP 状态码，可选值为 Enabled 或 Disabled，默认为 Enabled		

Container 节点 RoutingRules 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
RoutingRule	WebsiteConfiguration.RoutingRules	单条重定向规则配置	Container	是

Container 节点 RoutingRules.RoutingRule 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Condition	WebsiteConfiguration.RoutingRules.RoutingRule	重定向规则的条件配置	Container	是
Redirect	WebsiteConfiguration.RoutingRules.RoutingRule	重定向规则的具体重定向目标配置	Container	是

Container 节点 RoutingRules.RoutingRule.Condition 的内容：

节点名称（关键字）	父节点	描述	类型
HttpErrorCodeReturnedEquals	WebsiteConfiguration.RoutingRules.RoutingRule.Condition	指定重定向规则的错误码匹配条件，只支持配置 4XX 返回码，例如 403	Integer

		或 404	
KeyPrefixEquals	WebsiteConfiguration.RoutingRules.RoutingRule.Condition	指定重定向规则的对象键前缀匹配条件	String

Container 节点 RoutingRules.RoutingRule.Redirect 的内容：

节点名称（关键字）	父节点	描述	类型
Protocol	WebsiteConfiguration.RoutingRules.RoutingRule.Redirect	指定重定向规则的目标协议，只能设置为 https	String
ReplaceKeyWith	WebsiteConfiguration.RoutingRules.RoutingRule.Redirect	指定重定向规则的具体重定向目标的对象键，替换方式为替换整个原始请求的对象键	String
ReplaceKeyPrefixWith	WebsiteConfiguration.RoutingRules.RoutingRule.Redirect	指定重定向规则的具体重定向目标的对象键，替换方式为替换原始请求中所匹配到的前缀部分，仅可在 Condition 为 KeyPrefixEquals 时设置	String

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

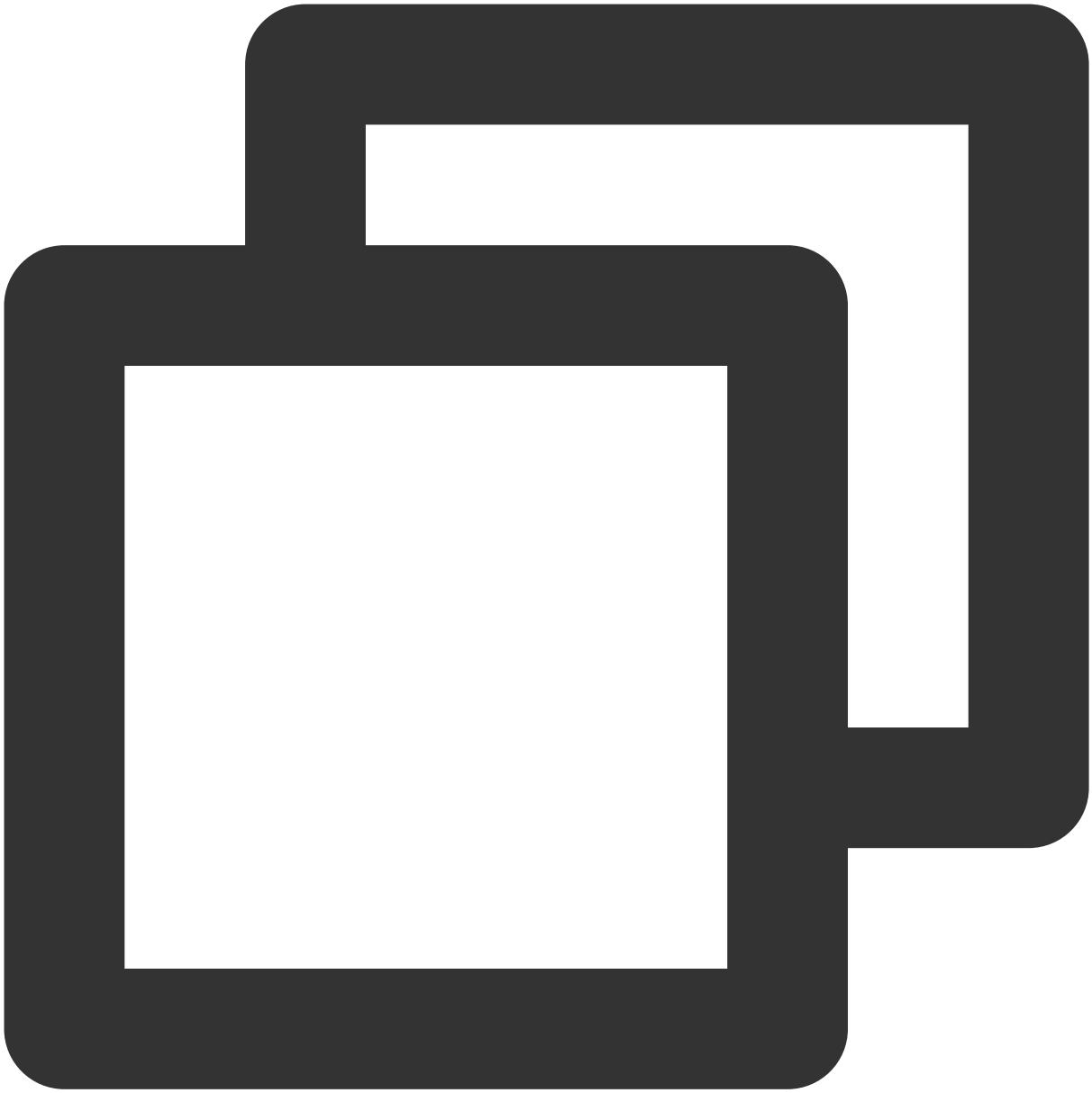
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

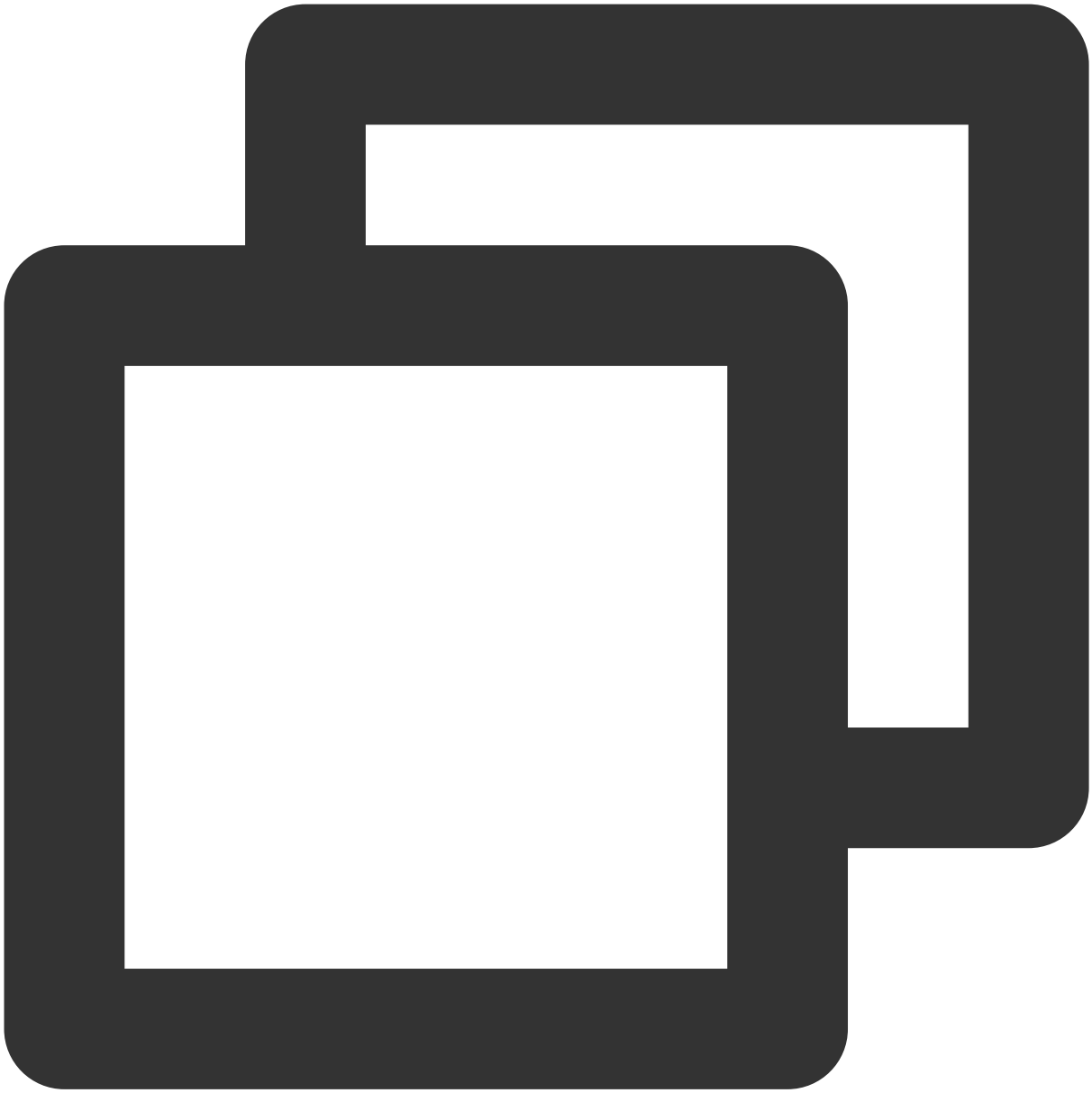
请求



```
PUT /?website HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 20 May 2020 09:33:38 GMT
Content-Type: application/xml
Content-Length: 1209
Content-MD5: VHzj4Uwb++HLyCJp7jUzWg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<WebsiteConfiguration>
    <IndexDocument>
```

```
        <Suffix>index.html</Suffix>
    </IndexDocument>
    <RedirectAllRequestsTo>
        <Protocol>https</Protocol>
    </RedirectAllRequestsTo>
    <ErrorDocument>
        <Key>pages/error.html</Key>
    </ErrorDocument>
    <RoutingRules>
        <RoutingRule>
            <Condition>
                <HttpErrorCodeReturnedEquals>403</HttpErrorCodeReturnedEquals>
            </Condition>
            <Redirect>
                <Protocol>https</Protocol>
                <ReplaceKeyWith>pages/403.html</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
            </Condition>
            <Redirect>
                <ReplaceKeyWith>pages/404.html</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <KeyPrefixEquals>assets/</KeyPrefixEquals>
            </Condition>
            <Redirect>
                <ReplaceKeyWith>index.html</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <KeyPrefixEquals>article/</KeyPrefixEquals>
            </Condition>
            <Redirect>
                <Protocol>https</Protocol>
                <ReplaceKeyPrefixWith>archived/</ReplaceKeyPrefixWith>
            </Redirect>
        </RoutingRule>
    </RoutingRules>
</WebsiteConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Wed, 20 May 2020 09:33:38 GMT
Server: tencent-cos
x-cos-request-id: NWVjNGY5NzJfOThjMjJhMDlfMjg5Ml8yYzNi****
```

GET Bucket website

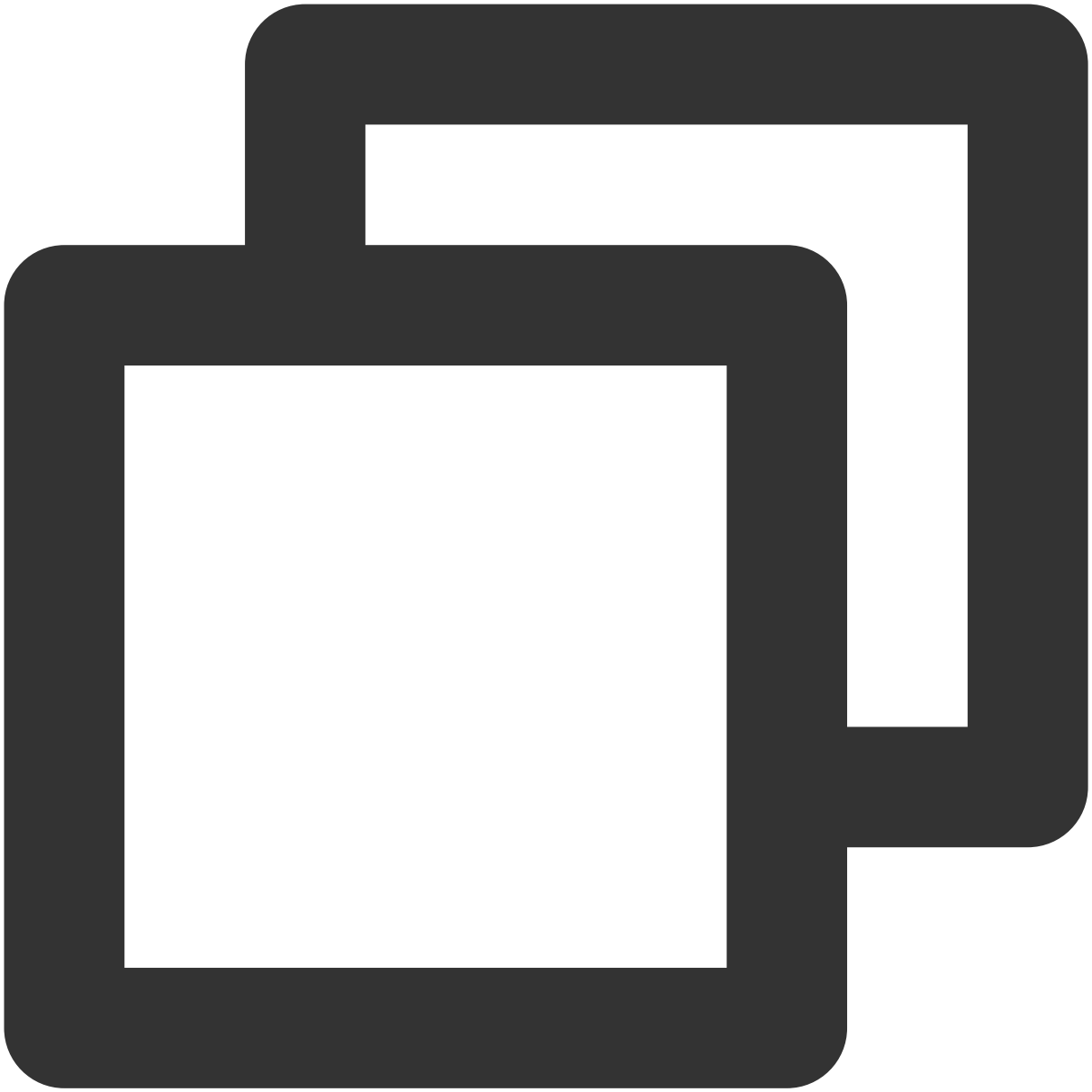
最近更新时间：2023-12-21 11:52:06

功能描述

GET Bucket website 请求用于查询与存储桶关联的静态网站配置信息。

请求

请求示例



```
GET /?website HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

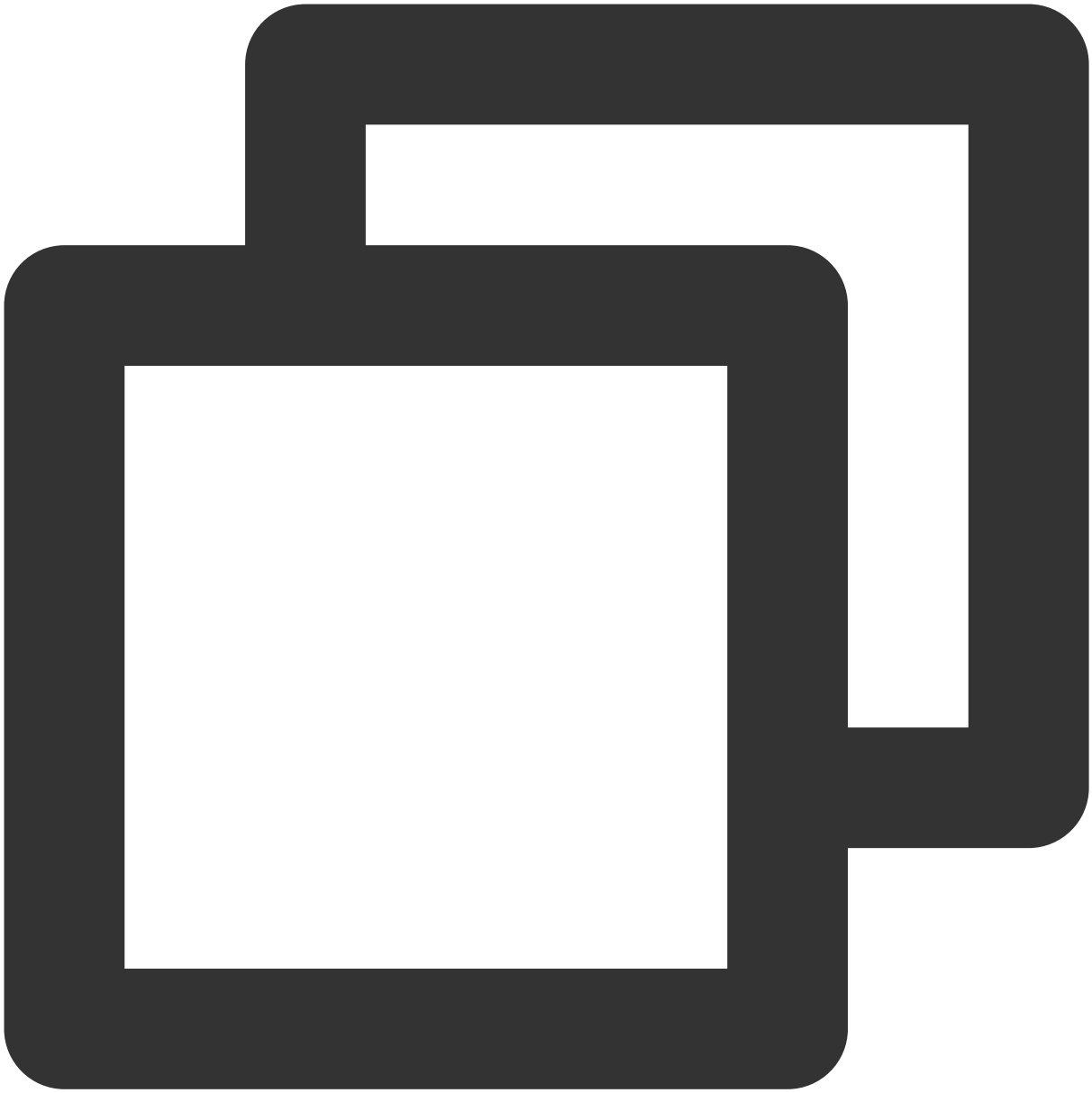
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含完整的存储桶静态网站配置信息。



```
<WebsiteConfiguration>
    <IndexDocument>
        <Suffix>string</Suffix>
    </IndexDocument>
    <RedirectAllRequestsTo>
        <Protocol>string</Protocol>
    </RedirectAllRequestsTo>
    <ErrorDocument>
        <Key>string</Key>
    </ErrorDocument>
    <RoutingRules>
```

```
        <RoutingRule>
            <Condition>
                <HttpErrorCodeReturnedEquals>integer</HttpErrorCodeReturnedEquals>
            </Condition>
            <Redirect>
                <Protocol>string</Protocol>
                <ReplaceKeyWith>string</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <KeyPrefixEquals>string</KeyPrefixEquals>
            </Condition>
            <Redirect>
                <Protocol>string</Protocol>
                <ReplaceKeyPrefixWith>string</ReplaceKeyPrefixWith>
            </Redirect>
        </RoutingRule>
    </RoutingRules>
</WebsiteConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
WebsiteConfiguration	无	保存 GET Bucket website 结果的所有信息	Container

Container 节点 WebsiteConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型
IndexDocument	WebsiteConfiguration	索引文档配置	Container
RedirectAllRequestsTo	WebsiteConfiguration	重定向所有请求配置	Container
ErrorDocument	WebsiteConfiguration	错误文档配置	Container
RoutingRules	WebsiteConfiguration	重定向规则配置	Container

Container 节点 IndexDocument 的内容：

节点名称（关键字）	父节点	描述	类型
Suffix	WebsiteConfiguration.IndexDocument	指定索引文档的对象键后缀。例如指定为 `index.html`，那么当访问到存储	string

版权所有：腾讯云计算（北京）有限责任公司

		桶的根目录时，会自动返回 index.html 的内容，或者当访问到 `article/` 目录时，会自动返回 `article/index.html` 的内容	

Container 节点 RedirectAllRequestsTo 的内容：

节点名称（关键字）	父节点	描述	类型
Protocol	WebsiteConfiguration.RedirectAllRequestsTo	指定重定向所有请求的目标协议	string

Container 节点 ErrorDocument 的内容：

节点名称（关键字）	父节点	描述	类型
Key	WebsiteConfiguration.ErrorDocument	指定通用错误文档的对象键	string

Container 节点 RoutingRules 的内容：

节点名称（关键字）	父节点	描述	类型
RoutingRule	WebsiteConfiguration.RoutingRules	单条重定向规则配置	Container

Container 节点 RoutingRules.RoutingRule 的内容：

节点名称（关键字）	父节点	描述	类型
Condition	WebsiteConfiguration.RoutingRules.RoutingRule	重定向规则的条件配置	Container
Redirect	WebsiteConfiguration.RoutingRules.RoutingRule	重定向规则的具体重定向目标配置	Container

Container 节点 RoutingRules.RoutingRule.Condition 的内容：

节点名称（关键字）	父节点	描述	类型
HttpErrorCodeReturnedEquals	WebsiteConfiguration.RoutingRules.RoutingRule.Condition	指定重定向	integer

版权所有：腾讯云计算（北京）有限责任公司

		规则的错误码匹配条件	
KeyPrefixEquals	WebsiteConfiguration.RoutingRules.RoutingRule.Condition	指定重定向规则的对象键前缀匹配条件	string

Container 节点 RoutingRules.RoutingRule.Redirect 的内容：

节点名称（关键字）	父节点	描述	类型
Protocol	WebsiteConfiguration.RoutingRules.RoutingRule.Redirect	指定重定向规则的目标协议	string
ReplaceKeyWith	WebsiteConfiguration.RoutingRules.RoutingRule.Redirect	指定重定向规则的具体重定向目标的对象键，替换方式为替换整个原始请求的对象键	string
ReplaceKeyPrefixWith	WebsiteConfiguration.RoutingRules.RoutingRule.Redirect	指定重定向规则的具体重定	string

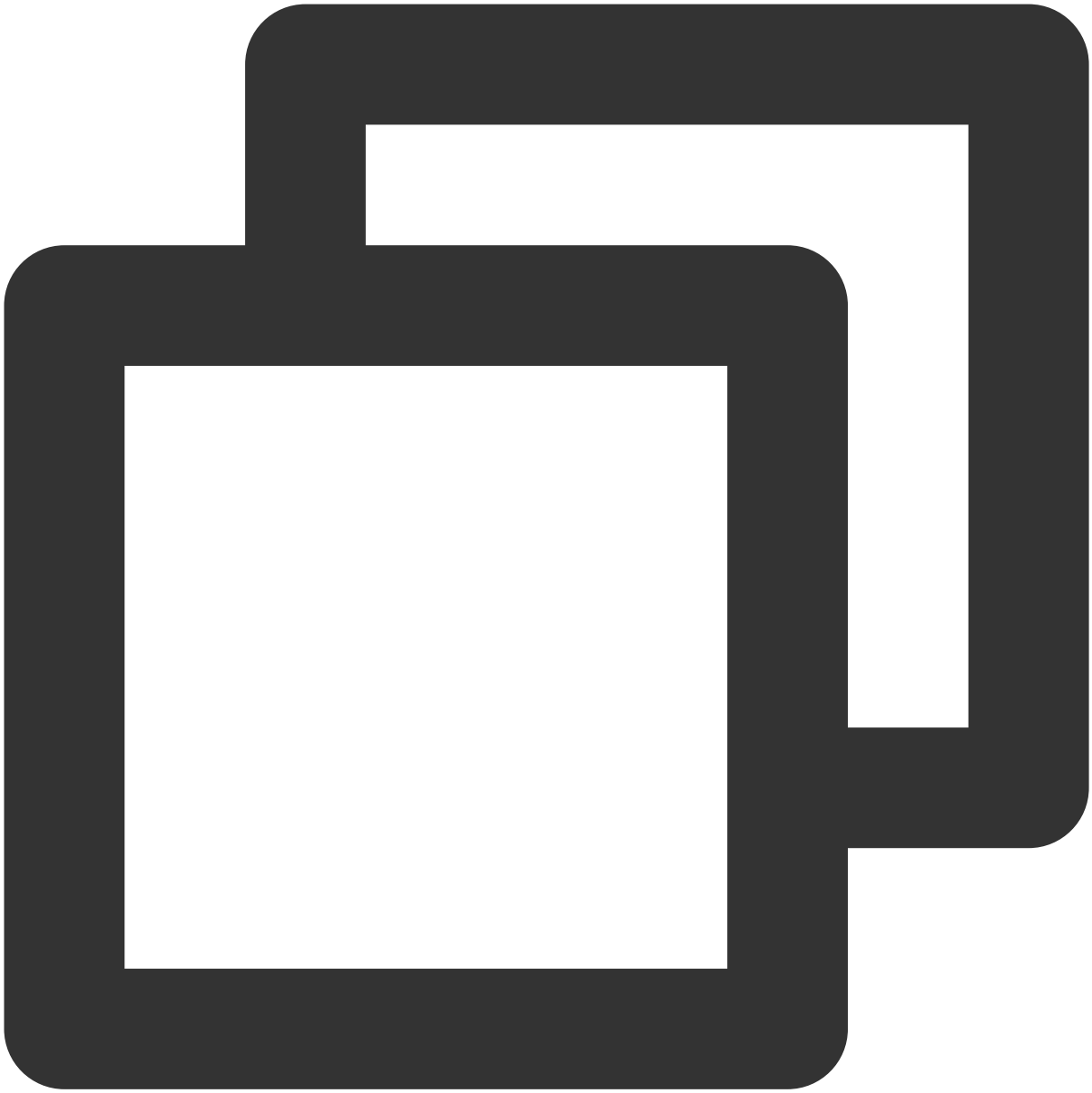
		向目标的对象键，替换方式为替换原始请求中所匹配到的前缀部分	

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

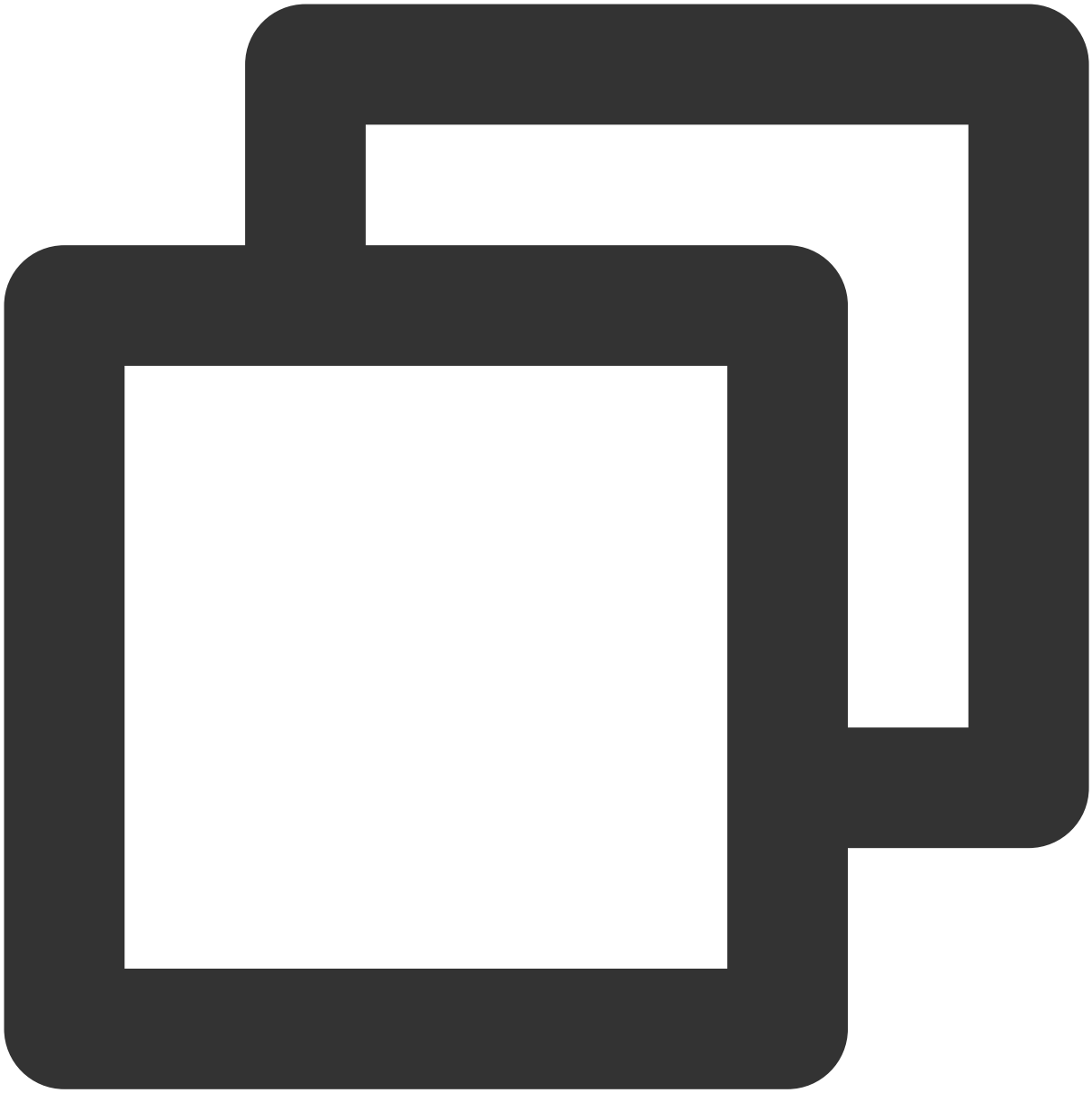
实际案例

请求



```
GET /?website HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 20 May 2020 09:33:49 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1163
Connection: close
Date: Wed, 20 May 2020 09:33:49 GMT
Server: tencent-cos
x-cos-request-id: NWVjNGY5N2RfYTdjMjJhMDlfNjZkY18yYWUx****

<WebsiteConfiguration>
    <IndexDocument>
        <Suffix>index.html</Suffix>
```

```
    </IndexDocument>
    <RedirectAllRequestsTo>
        <Protocol>https</Protocol>
    </RedirectAllRequestsTo>
    <ErrorDocument>
        <Key>pages/error.html</Key>
    </ErrorDocument>
    <RoutingRules>
        <RoutingRule>
            <Condition>
                <HttpErrorCodeReturnedEquals>403</HttpErrorCodeReturnedEquals>
            </Condition>
            <Redirect>
                <Protocol>https</Protocol>
                <ReplaceKeyWith>pages/403.html</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
            </Condition>
            <Redirect>
                <ReplaceKeyWith>pages/404.html</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <KeyPrefixEquals>assets/</KeyPrefixEquals>
            </Condition>
            <Redirect>
                <ReplaceKeyWith>index.html</ReplaceKeyWith>
            </Redirect>
        </RoutingRule>
        <RoutingRule>
            <Condition>
                <KeyPrefixEquals>article/</KeyPrefixEquals>
            </Condition>
            <Redirect>
                <Protocol>https</Protocol>
                <ReplaceKeyPrefixWith>archived/</ReplaceKeyPrefixWith>
            </Redirect>
        </RoutingRule>
    </RoutingRules>
</WebsiteConfiguration>
```

DELETE Bucket website

最近更新时间：2021-08-30 14:39:11

功能描述

DELETE Bucket website 请求用于删除存储桶中的静态网站配置。

推荐使用 API Explorer

点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

请求

请求示例

```
DELETE /?website HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明：

- Host: &dxlt;BucketName-APPID>.cos.&dxlt;Region>.myqcloud.com，其中 &dxlt;BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；&dxlt;Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
- Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

版权所有：腾讯云计算（北京）有限责任公司

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

```
DELETE /?website HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 19 May 2020 07:57:10 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1589875030;1589882230&q-key-time=1589875030;1589882230&q-header-list=date;host&q-url-param-list=website&q-signature=e000543b192f0739b36f420456708fcfb553****
Connection: close
```

响应

```
HTTP/1.1 204 No Content
Connection: close
Date: Tue, 19 May 2020 07:57:10 GMT
Server: tencent-cos
x-cos-request-id: NWVjMzkxNTZfY2ZhZjJhMDlfNWI2OV8yYWFh****
Content-Length: 0
```

智能分层（IntelligentTiering）
PUT Bucket IntelligentTiering

最近更新时间：2023-12-21 12:26:21

功能描述

PUT Bucket IntelligentTiering 接口用于启用存储桶的智能分层存储配置。

说明

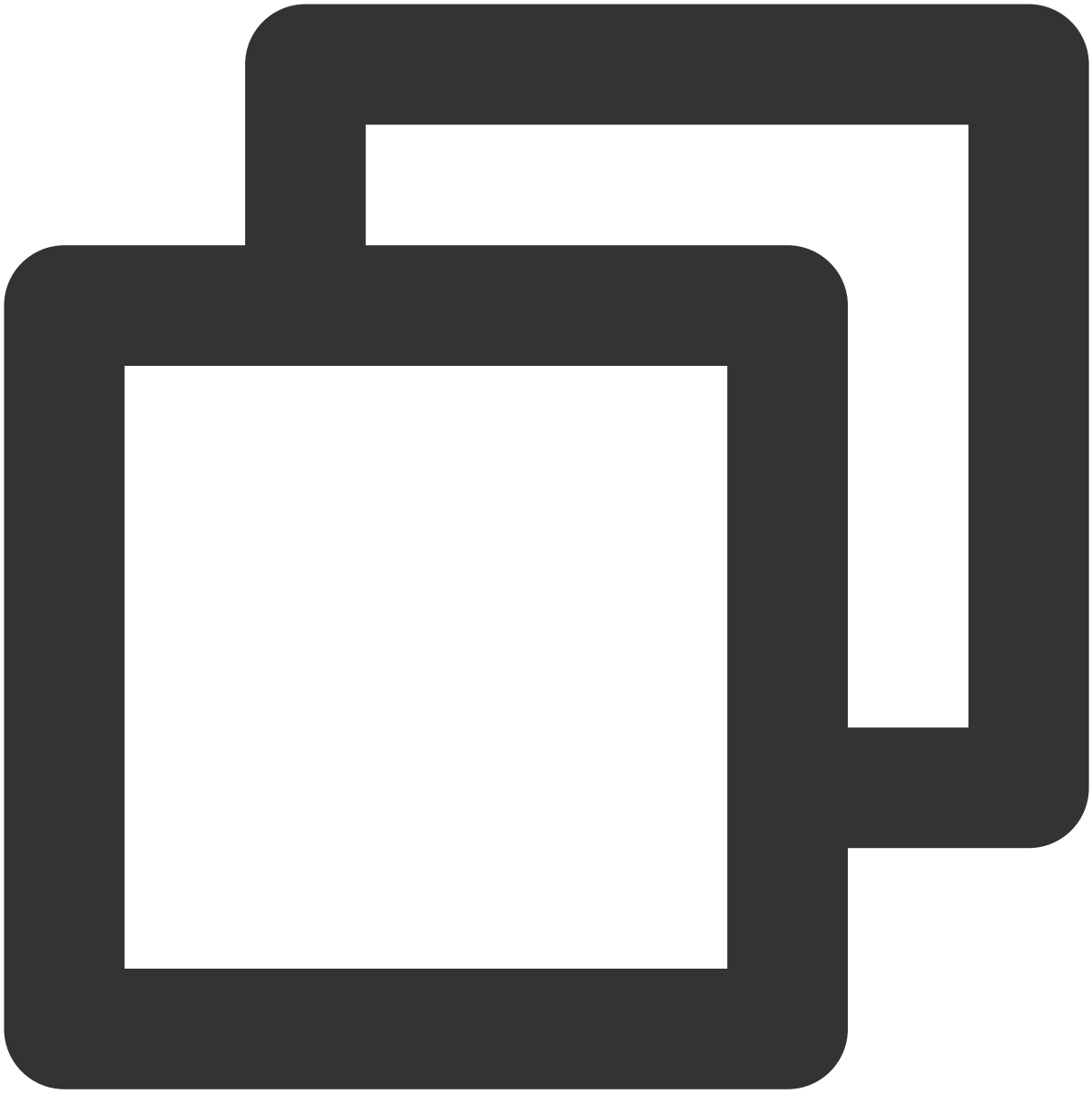
启用智能分层配置后，将无法关闭或修改。

只有主账号或者被授权 PUT Bucket IntelligentTiering 接口权限的子账号可以调用该接口。

您可以通过 查询对象元数据 接口返回的 `x-cos-storage-tier` 获取对象所处的存储层。

请求

请求示例



```
PUT /?intelligenttiering HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Connection: keep-alive
Authorization: authorization string
Content-Type: text/plain
Content-Length: Int
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

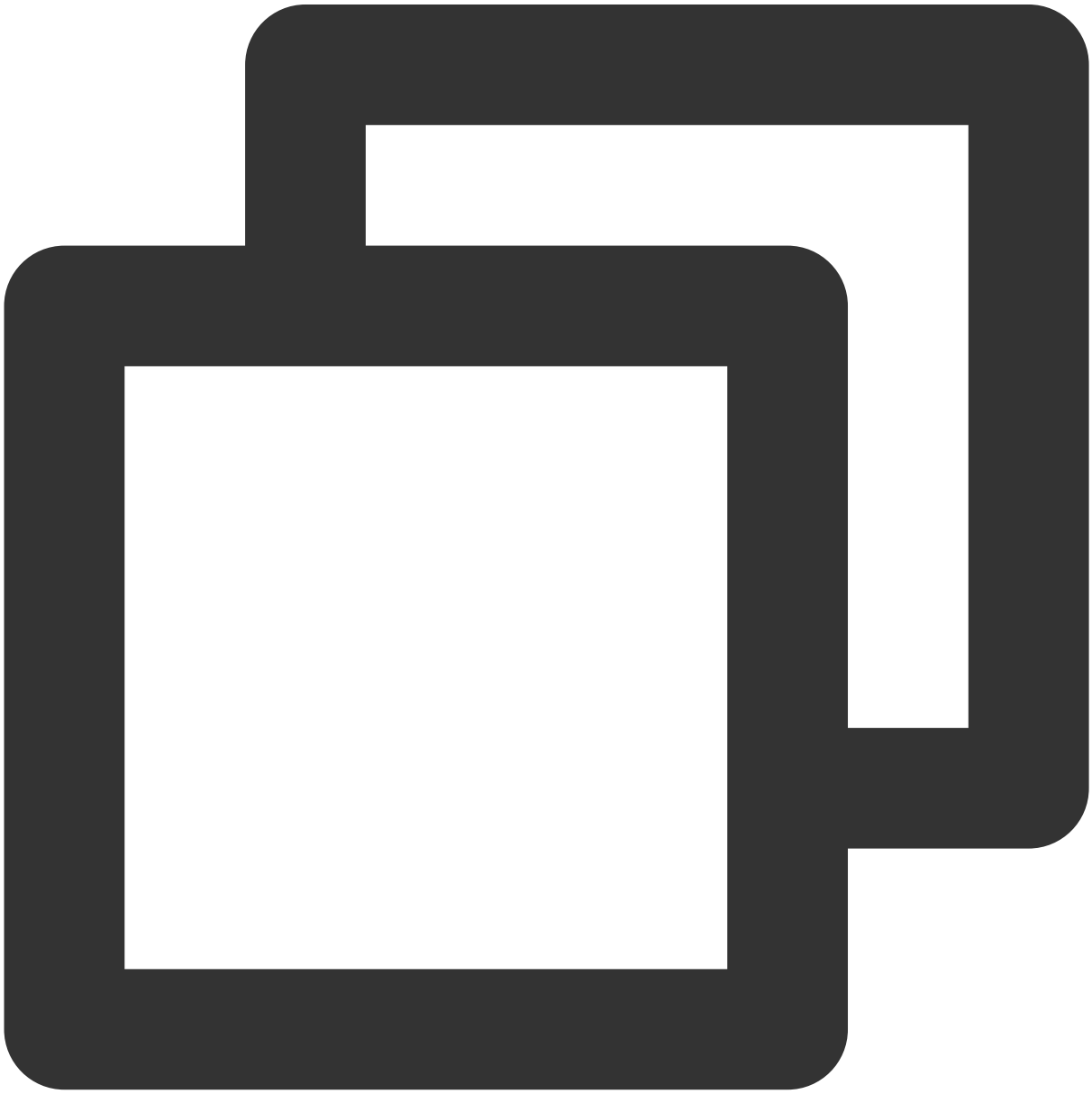
请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体



```
<IntelligentTieringConfiguration xmlns="cos xmlns/">
  <Status>Enabled</Status>
  <Transition>
    <Days>30|60|90</Days>
    <RequestFrequent>1</RequestFrequent>
  </Transition>
</IntelligentTieringConfiguration>
```

具体的数据内容如下：

名称	父节点	描述	类型	是否必选
IntelligentTieringConfiguration	无	智能分层存储配置的具体信息	Container	是
Status	IntelligentTieringConfiguration	说明智能分层存储配置是否开启，枚举值：Enabled	Enum	是
Transition	IntelligentTieringConfiguration	指定智能分层存储配置中有关数据转换的配置信息	Container	是
Days	IntelligentTieringConfiguration.Transition	指定智能分层存储配置中标准层数据转换为低频层数据的天数限制，可选天数为30、60、90。默认值为30	Int	是
RequestFrequent	IntelligentTieringConfiguration.Transition	指定配置中标准层数据转换为低频层数据的访	Int	是

		问次数限制，默认值为1次，和天数搭配使用可以实现转换效果。例如设置该参数为1，访问天数为30，代表连续30天访问次数小于1次的对象会从标准层降到低频层		

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

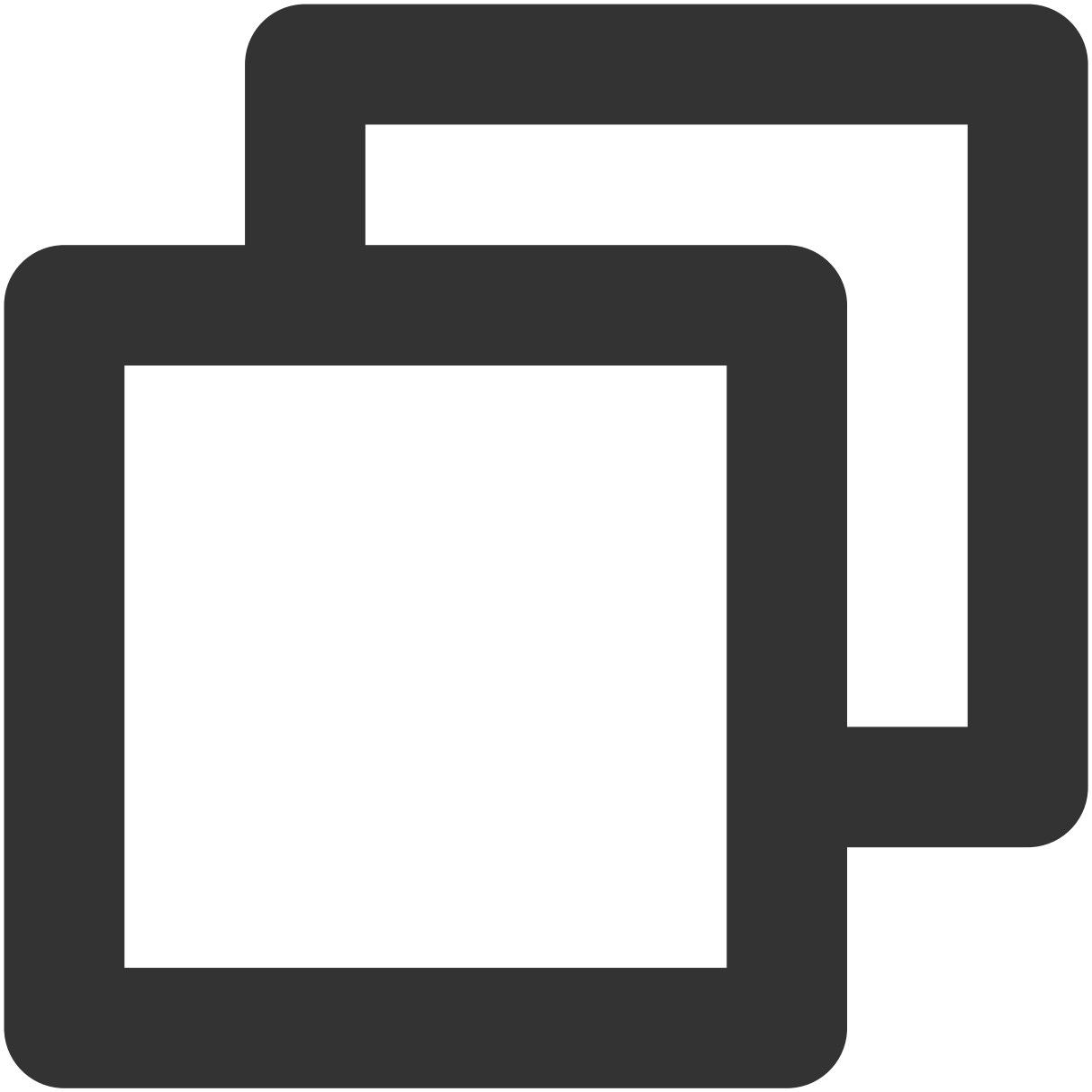
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

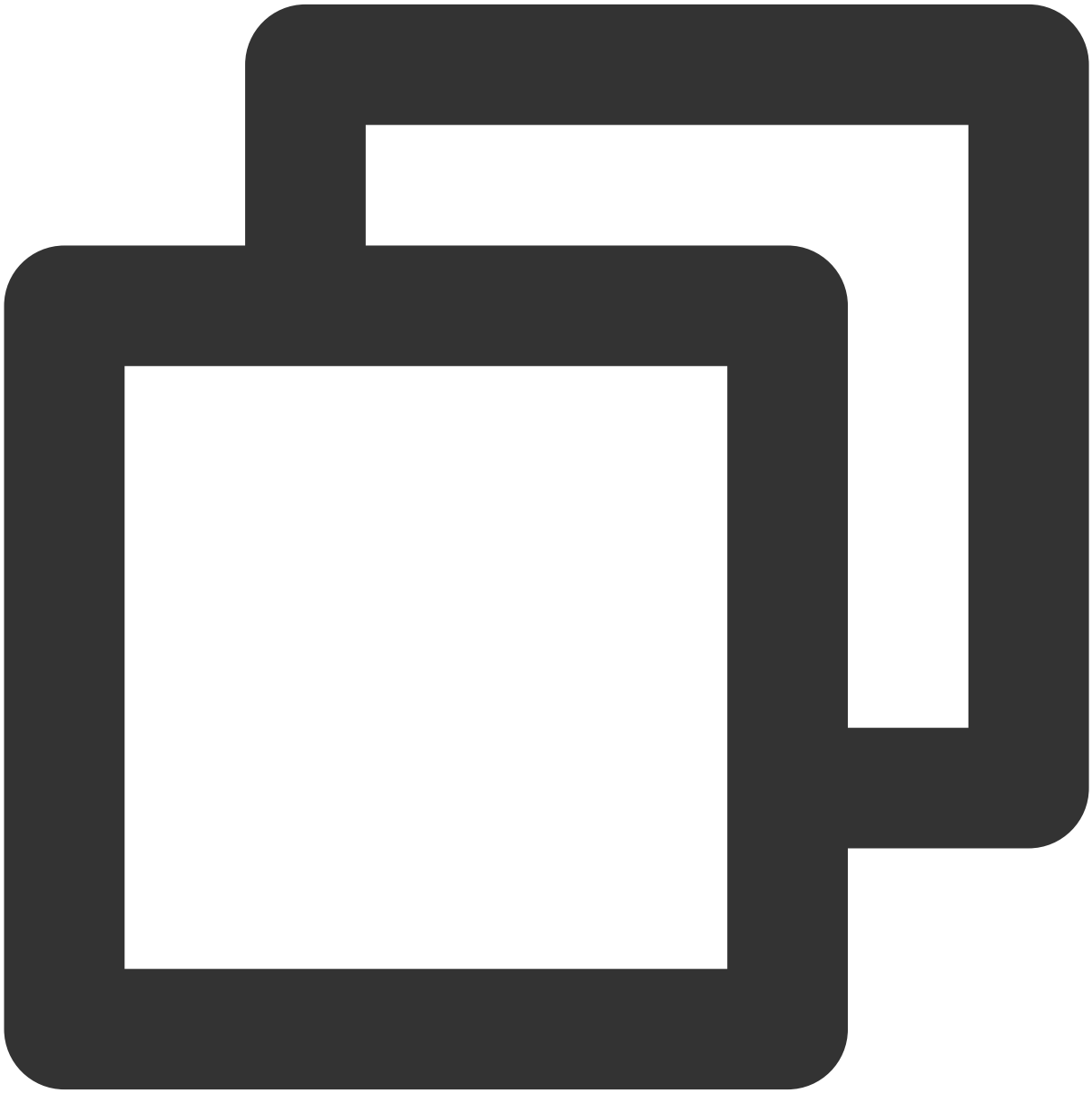
请求



```
PUT /?intelligenttiering HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Content-Type: application/xml
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9Sm****&q-si
Content-Length: 83

<IntelligentTieringConfiguration xmlns="cos xmlns/">
```

```
  <Status>Enabled</Status>
  <Transition>
    <Days>30</Days>
    <RequestFrequent>1</RequestFrequent>
  </Transition>
</IntelligentTieringConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 0
Connection: keep-alive
Date: Sun, 23 Aug 2020 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****
```

GET Bucket IntelligentTiering

最近更新时间：2023-12-21 12:26:16

功能描述

GET Bucket IntelligentTiering 接口用于查询存储桶的智能分层配置信息。

注意

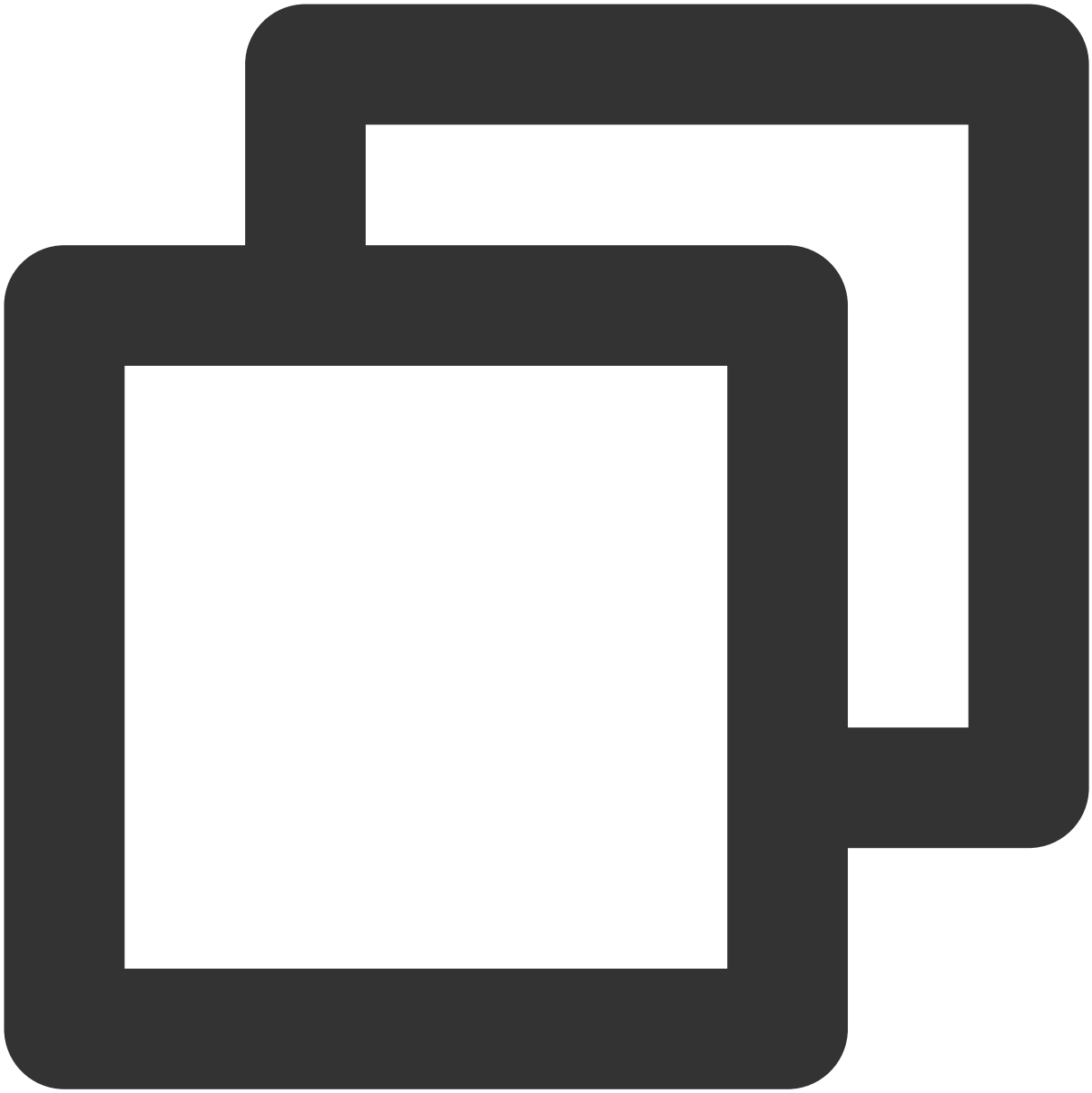
只有主账号或者被授权 GET Bucket IntelligentTiering 接口权限的子账号可以调用该接口。

该配置有未启用和启用两种状态：

如果您从未在存储桶上启用智能分层存储配置，则响应为：

```shell
<IntelligentTieringConfiguration/>
```

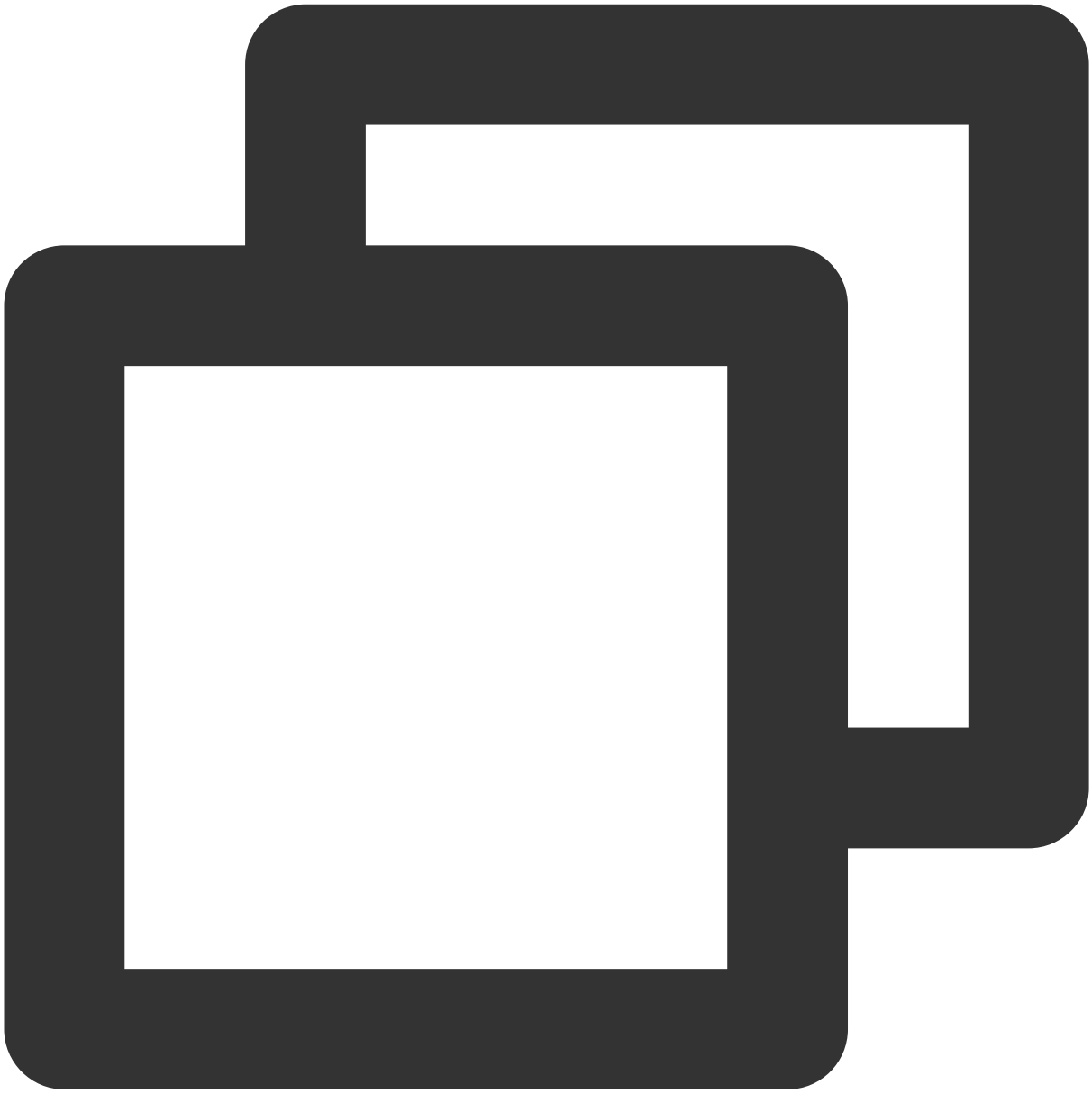
如果您启用了存储桶的智能分层配置，则响应为：



```
<IntelligentTieringConfiguration xmlns="cos xmlns/">
    <Status>Enabled</Status>
    <Transition>
       <Days>30</Days>
       <RequestFrequent>1</RequestFrequent>
    </Transition>
</IntelligentTieringConfiguration>
```

请求

请求示例



```
GET /?intelligenttiering HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

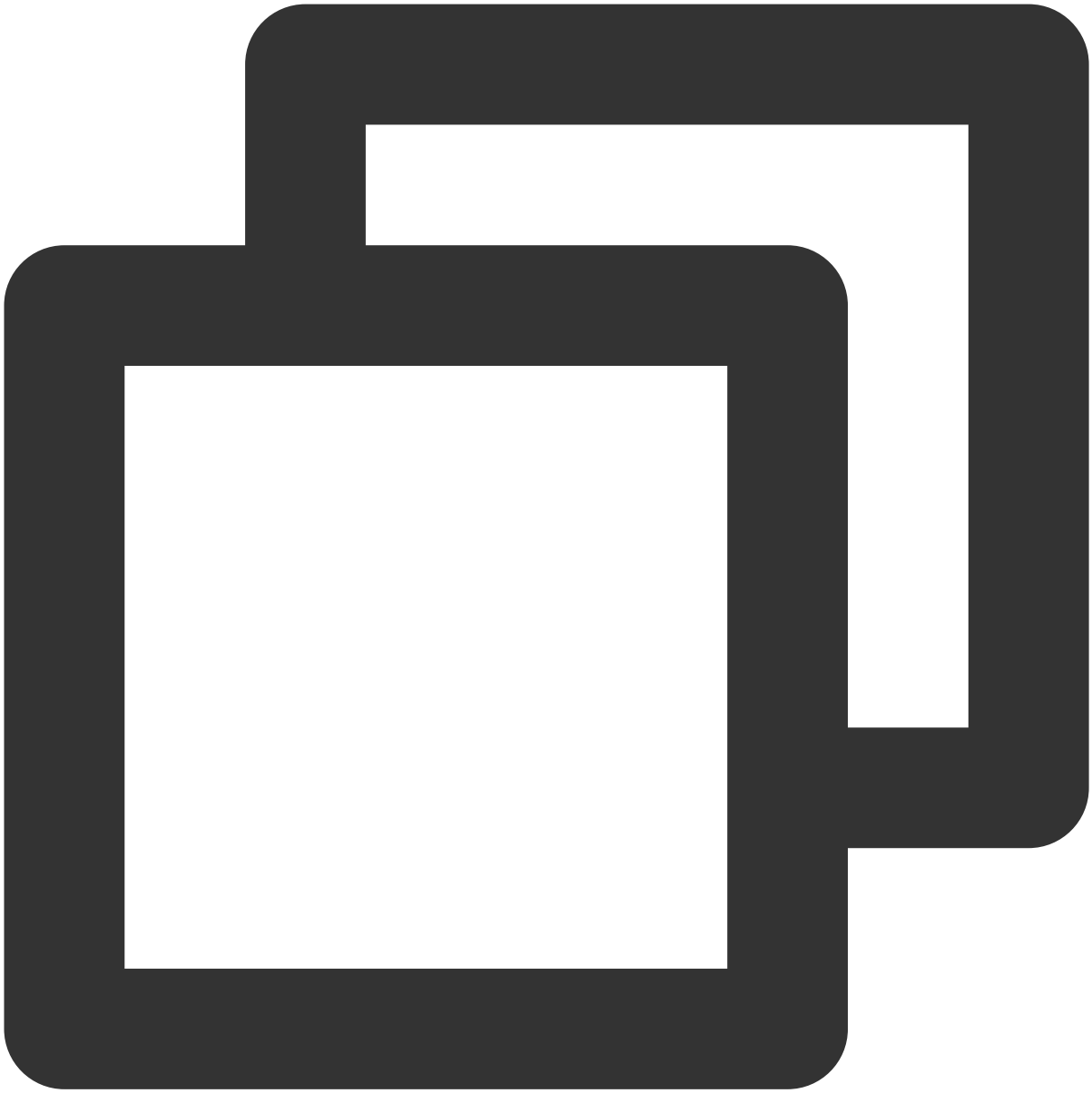
该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<IntelligentTieringConfiguration xmlns="cos xmlns/">
  <Status>Enabled</Status>
  <Transition>
    <Days>30|60|90</Days>
    <RequestFrequent>1</RequestFrequent>
  </Transition>
</IntelligentTieringConfiguration>
```

具体的节点描述如下：

名称	父节点	描述	类型

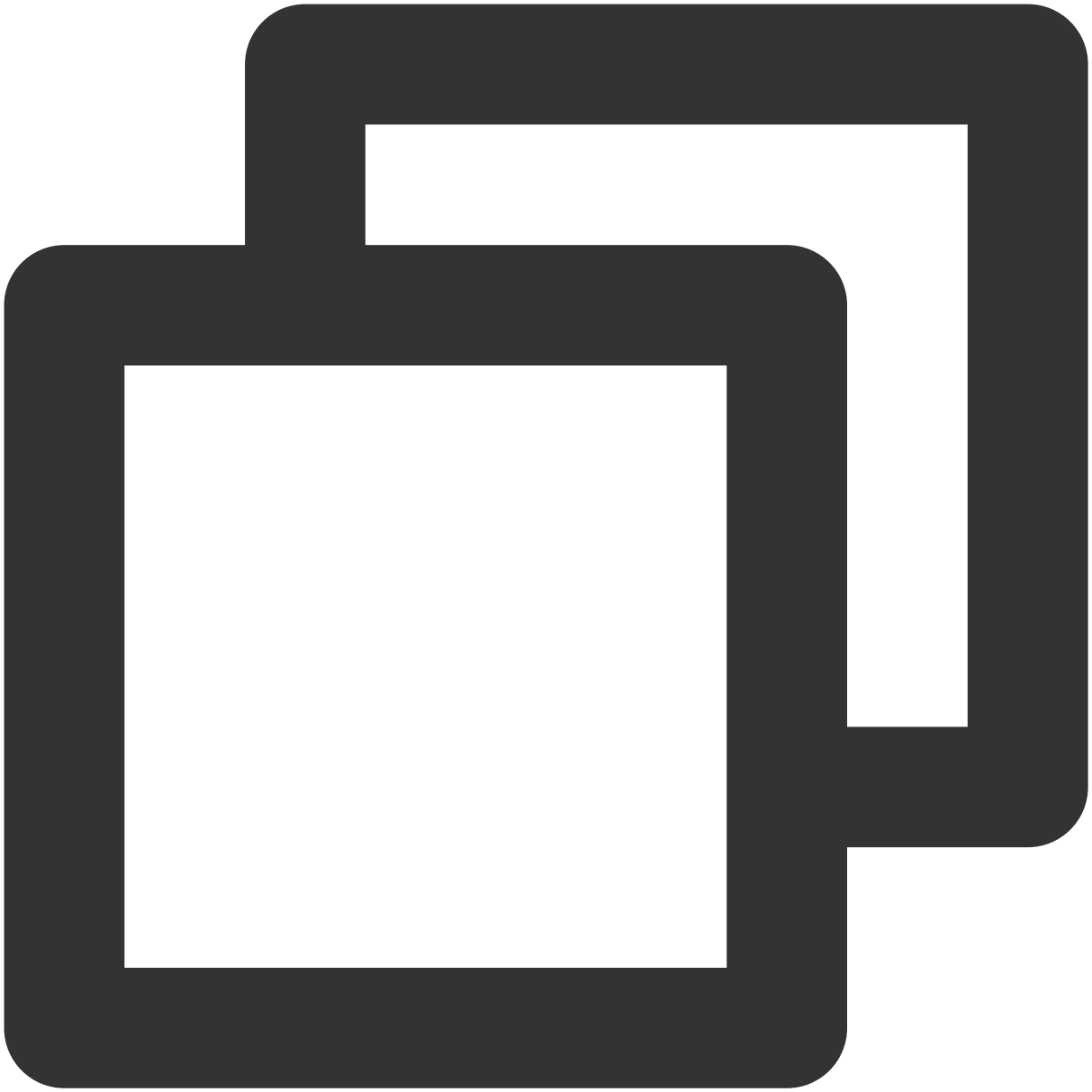
IntelligentTieringConfiguration	无	智能分层存储配置的具体信息	Container
Status	IntelligentTieringConfiguration	说明智能分层存储配置是否开启，枚举值：Enabled	Enum
Transition	IntelligentTieringConfiguration	指定智能分层存储配置中有关数据转换的配置信息	Container
Days	IntelligentTieringConfiguration.Transition	指定智能分层存储配置中标准层数据转换为低频层数据的天数限制，可选值为30、60和90，默认值为30天	Int
RequestFrequent	IntelligentTieringConfiguration.Transition	指定配置中标准层数据转换为低频层数据的访问次数限制，默认值为1次，和天数搭配使用可以实现转换效果。例如设置该参数为1，访问天数为30，代表连续30天访问次数小于1次的对象会从标准层降到低频层	Int

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

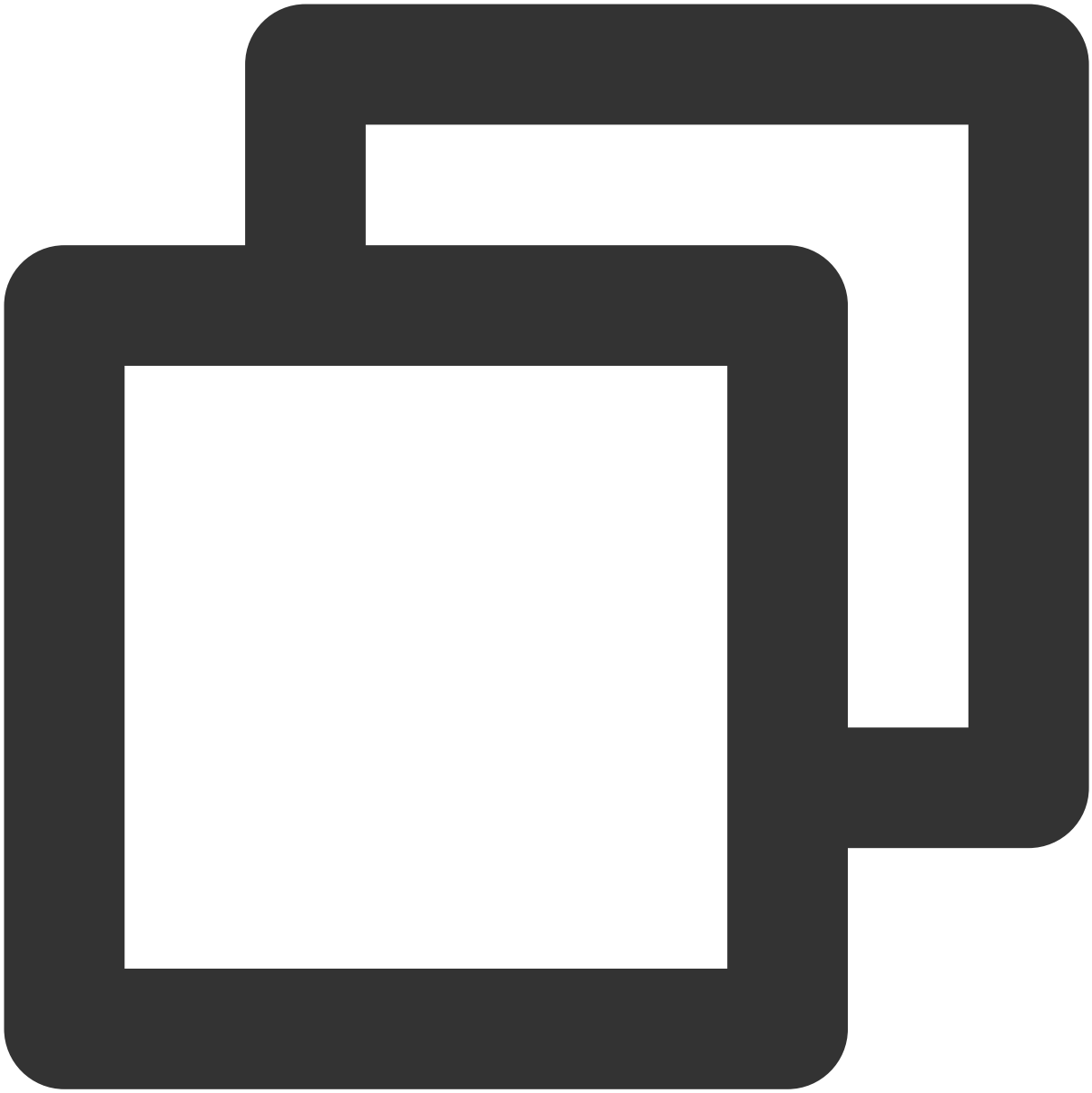
实际案例

请求



```
GET /?intelligenttiering HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9Sm****&q-si
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 120
Connection: keep-alive
Date: Sun, 23 Aug 2020 08:15:16 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5OTRfZDNhZDM1MGFfMjYyMTFfZmU3****

<IntelligentTieringConfiguration xmlns="cos xmlns/">
  <Status>Enabled</Status>
  <Transition>
```

```
    <Days>30</Days>
    <RequestFrequent>1</RequestFrequent>
  </Transition>
</IntelligentTieringConfiguration>
```

清单（inventory）
PUT Bucket inventory

最近更新时间：2023-12-21 12:31:55

功能描述

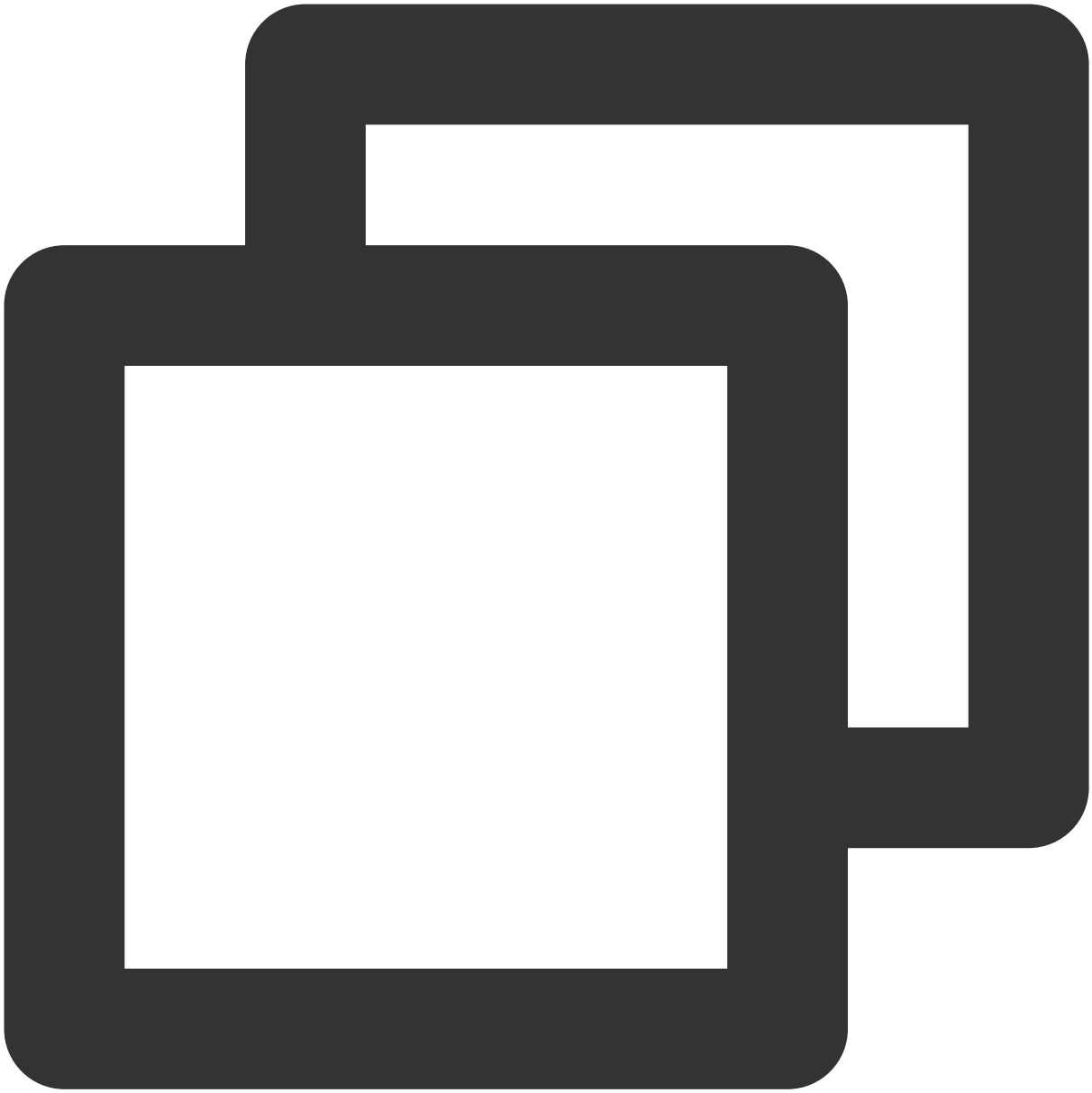
PUT Bucket inventory 用于在存储桶中创建清单任务，您可以对清单任务命名后，使用该请求创建清单任务，详细信息请参见 清单功能概述。

注意

COS 支持在每个存储桶中创建最多1000条清单任务。
您必须在目标存储桶中写入存储桶策略，以供 COS 将清单任务的结果文件写入该存储桶中。
调用该请求时，请确保您有足够的权限对存储桶的清单任务进行操作。存储桶所有者默认拥有该权限，若您无该项权限，请先向存储桶所有者申请该项操作的权限。授权流程请参见 使用方法。
如果您指定了清单投递的前缀，COS 后端会自动在您指定的前缀后边加上 `/` 。如您指定了 `Prefix` 作为前缀，则 COS 后端投递的清单报告路径为 `Prefix/inventory_report` 。

请求

请求示例



```
PUT /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Content-MD5: MD5
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

调用 PUT Bucket inventory 需要使用清单任务名称的参数。该参数格式如下：

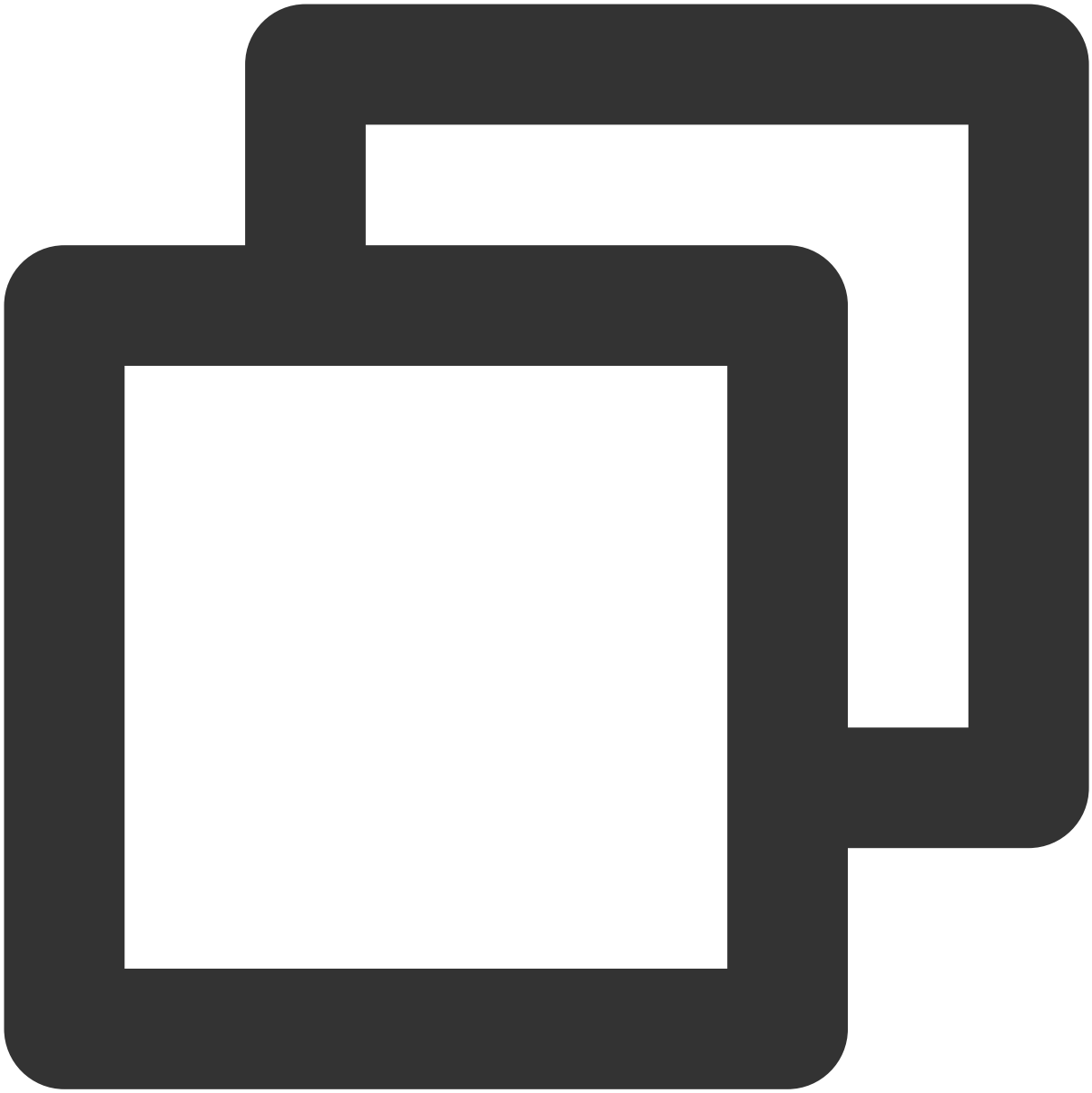
参数	描述	类型	是否必选
id	清单任务的名称。缺省值：None 合法字符：`a-z, A-Z, 0-9, -, _, .`	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

用户在请求体中使用 XML 语言设置清单任务的具体配置信息。配置信息包括清单任务分析的对象，分析的频次，分析的维度，分析结果的格式及存储的位置等信息。



```
<InventoryConfiguration>
    <Id>list1</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>100000000001</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
            <Prefix>list1</Prefix>
            <Encryption>
                <SSE-COS></SSE-COS>
```

```
            </Encryption>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <Prefix>myPrefix</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>ETag</Field>
        <Field>StorageClass</Field>
        <Field>IsMultipartUploaded</Field>
        <Field>ReplicationStatus</Field>
        <Field>Tag</Field>
        <Field>Crc64</Field>
        <Field>x-cos-meta-*</Field>
    </OptionalFields>
</InventoryConfiguration>
```

具体的节点描述如下：

节点名	父节点	描述
InventoryConfiguration	无	包含清单的配置参数
Id	InventoryConfiguration	清单的名称，与请求参数中的 id 对应
IsEnabled	InventoryConfiguration	清单是否启用的标识： 如果设置为 true，清单功能将生效 如果设置为 false，将不生成任何清单
IncludedObjectVersions	InventoryConfiguration	是否在清单中包含对象版本： 如果设置为 All，清单中将会包含所有对象版本，并在清单中 如果设置为 Current，则清单中不包含对象版本信息
Filter	InventoryConfiguration	筛选待分析对象。清单功能将分析符合 Filter 中设置的前缀
And	Filter	筛选待分析对象时，如果同时需要前缀与对象标签条件，需
Prefix	And	需要分析的对象的前缀

Tag	And	筛选待分析对象时，可以用对象标签（支持多个）作为过滤
Period	Filter	需要分析的对象的创建时间范围
StartTime	Period	需要分析的对象创建的起始时间，参数为秒级时间戳，如1
EndTime	Period	需要分析的对象创建的结束时间，参数为秒级时间戳，如1
OptionalFields	InventoryConfiguration	设置清单结果中应包含的分析项目
Field	OptionalFields	清单结果中可选包含的分析项目名称，可选字段包括：`Size`，`LastModifiedDate`，`StorageClas` `cos-meta-*` 注意，如果筛选条件里使用了对象标签，在这里也必须添加 此外，也支持用户填写 `x-cos-meta-*` 形式的自定义头 则为空。
Schedule	InventoryConfiguration	配置清单任务周期
Frequency	Schedule	清单任务周期，可选项为按日或者按周，枚举值：Daily、W
Destination	InventoryConfiguration	描述存放清单结果的信息
COSBucketDestination	Destination	清单结果导出后存放的存储桶信息
Bucket	COSBucketDestination	清单分析结果的存储桶名
AccountId	COSBucketDestination	存储桶的所有者 ID，例如100000000001
Prefix	COSBucketDestination	清单分析结果的前缀
Format	COSBucketDestination	清单分析结果的文件形式，可选项为 CSV 格式
Encryption	COSBucketDestination	为清单结果提供服务端加密的选项
SSE-COS	Encryption	COS 托管密钥的加密方式，无需填充

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求的响应体返回为空。

错误码

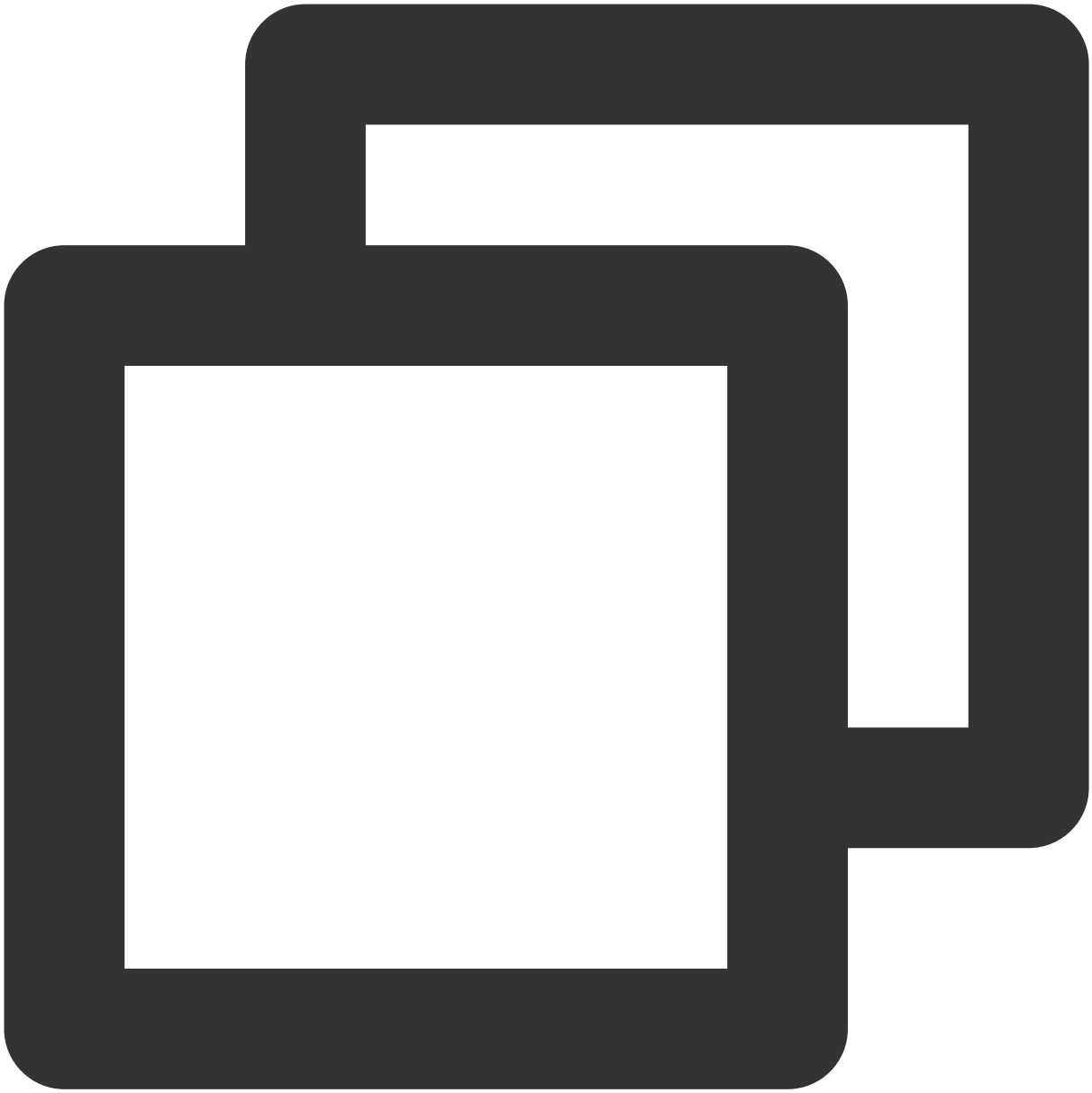
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：添加指定了对象前缀的清单任务

请求

该示例向存储桶 `examplebucket-1250000000` 中添加一条名为 list1 的清单任务。
该清单任务分析存储桶中前缀为 myPrefix 的对象及其所有版本。
分析频次为每天一次。
分析维度包括 Size， LastModifiedDate， StorageClass，ETag，IsMultipartUploaded， ReplicationStatus。
分析结果将以 CSV 格式文件存储在存储桶 examplebucket-1250000000 中，文件添加前缀 list1 且用 SSE-COS 加密。

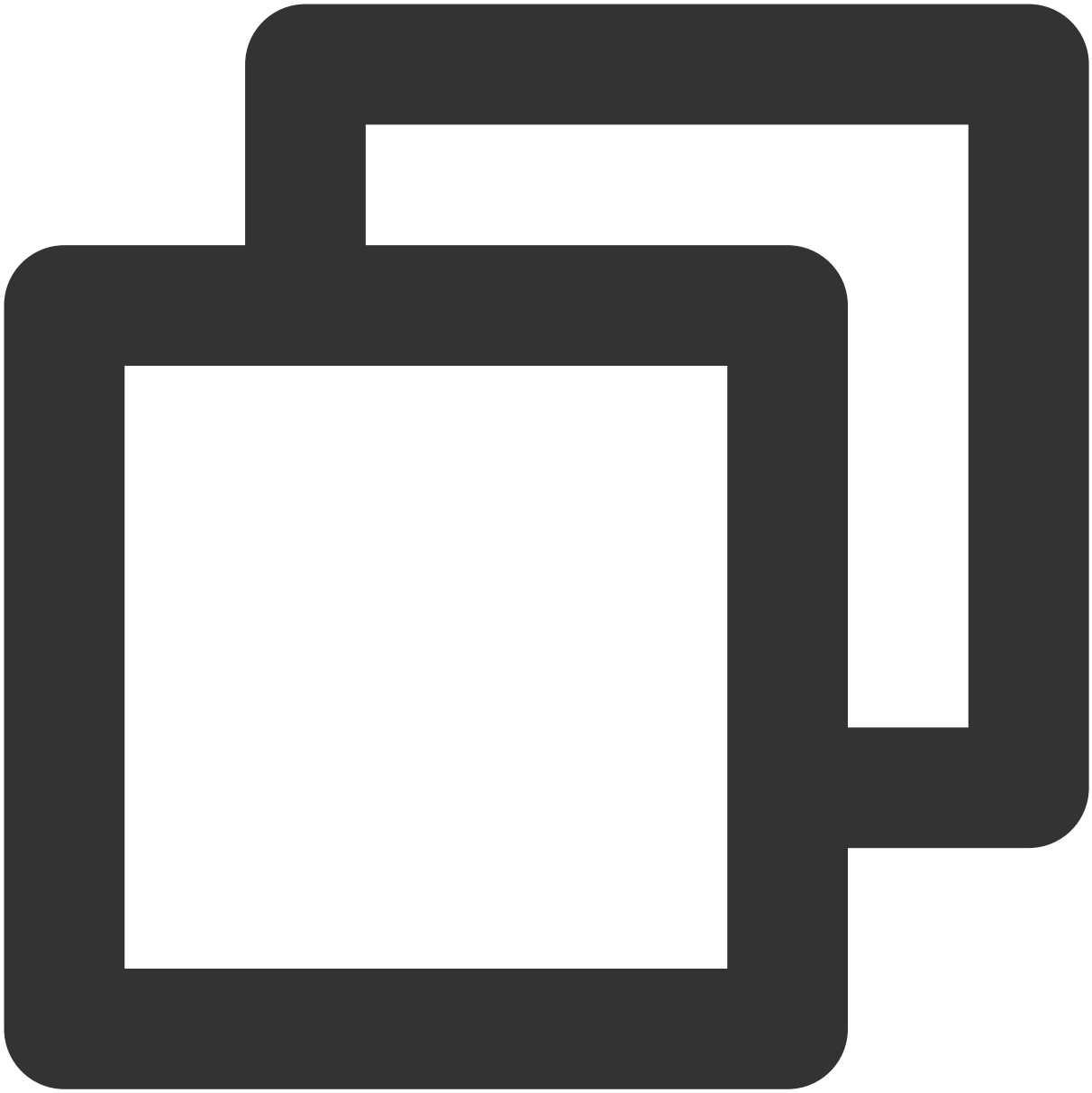


```
PUT /?inventory&id=list1 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Content-MD5: AAq9nzrpsz5LJ4UEe1f6Q==
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 1024

<?xml version = "1.0" encoding = "UTF-8">
<InventoryConfiguration xmlns = "http://....">
    <Id>list1</Id>
    <IsEnabled>true</IsEnabled>
```

```
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>100000000001</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
            <Prefix>list1</Prefix>
            <Encryption>
                <SSE-COS></SSE-COS>
            </Encryption>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <Prefix>myPrefix</Prefix>
      <Period>
        <StartTime>1568688761</StartTime>
        <EndTime>1568688762</EndTime>
      </Period>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>ETag</Field>
        <Field>StorageClass</Field>
        <Field>IsMultipartUploaded</Field>
        <Field>ReplicationStatus</Field>
    </OptionalFields>
</InventoryConfiguration>
```

响应

上述请求后，COS 返回以下响应，表明该清单任务 list1 已经成功设置完毕。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, 28 Aug 2018 02:53:38 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF8****
```

案例二：添加指定了对象前缀和对象标签的清单任务

请求

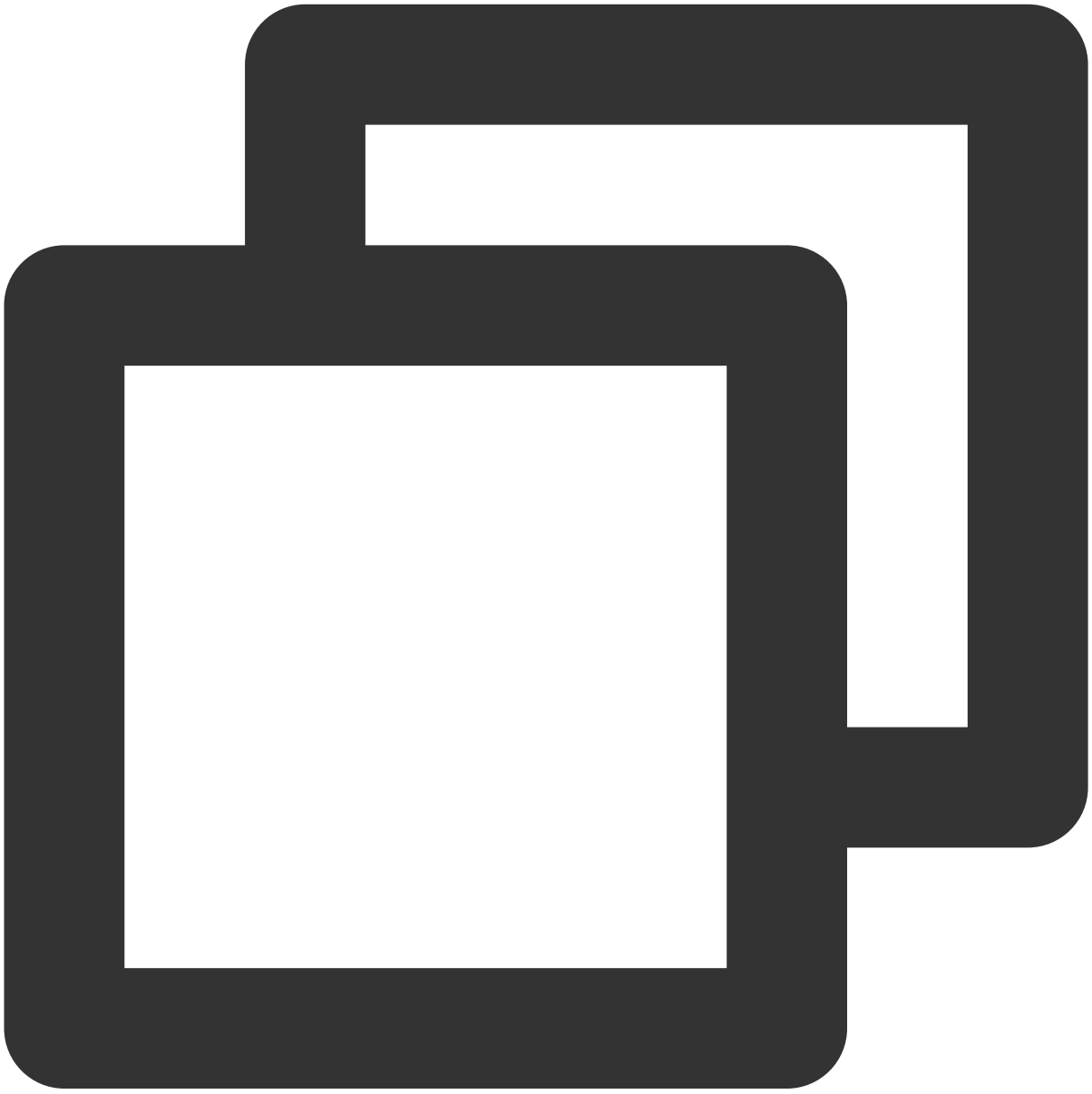
该示例向存储桶 `examplebucket-1250000000` 中添加一条名为 list2 的清单任务。

该清单任务分析存储桶中前缀为 myPrefix ， 且有对象标签含有{age:18}的对象及其所有版本。

分析频次为每天一次。

分析维度包括 Size ， LastModifiedDate， StorageClass，ETag，Tag。

分析结果将以 CSV 格式文件存储在存储桶 inventorybucket-1250000000 中。



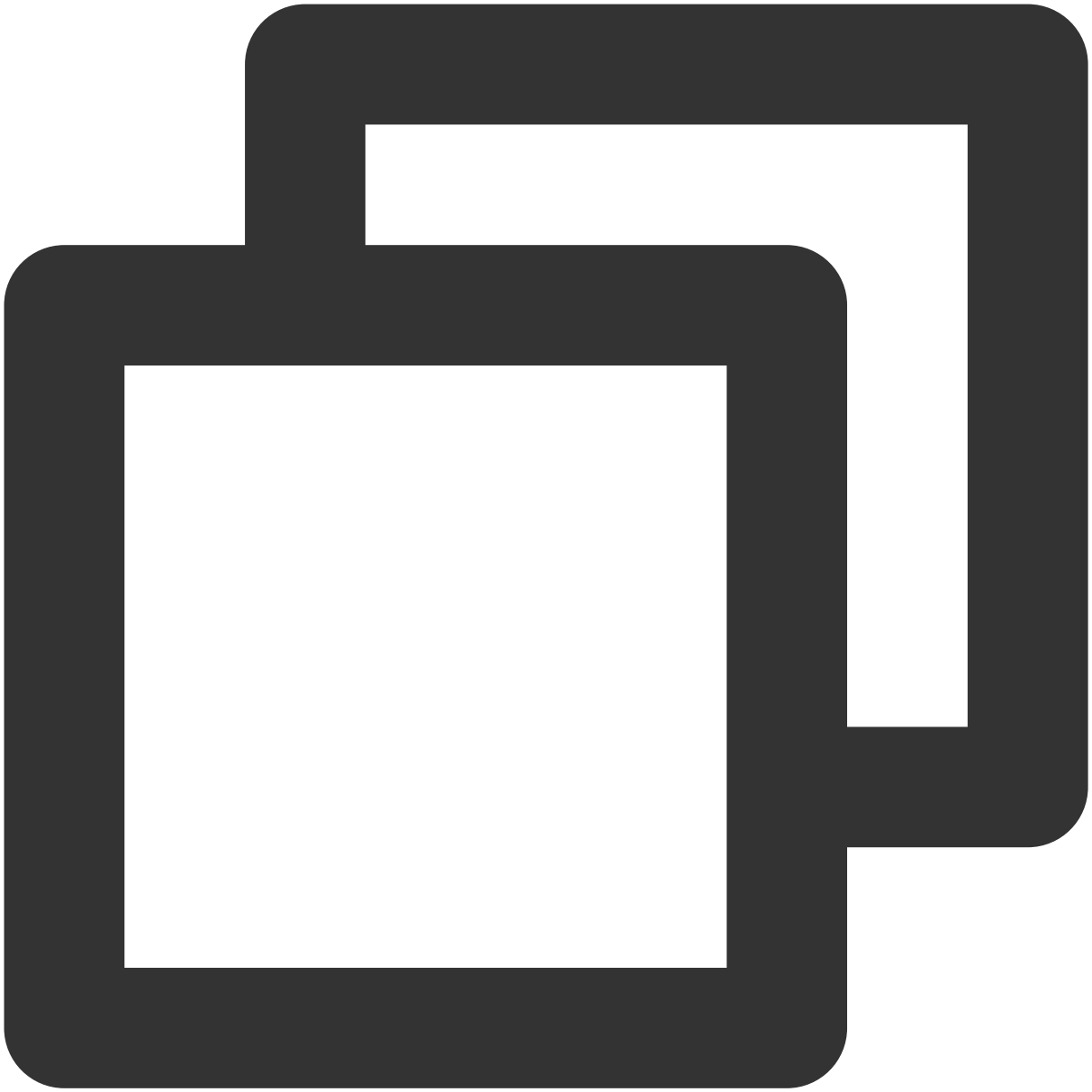
```
PUT /?inventory&id=list2 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Content-MD5: AAq9nzrpsz5LJ4UEe1f6Q==
```

版权所有：腾讯云计算（北京）有限责任公司

```
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 1024

<?xml version = "1.0" encoding = "UTF-8">
<InventoryConfiguration xmlns = "http://....">
    <Id>list2</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>100000000001</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::inventorybucket-1250000000</Bucket>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <And>
            <Prefix>myPrefix</Prefix>
            <Tag>
                <Key>age</Key>
                <Value>18</Value>
            </Tag>
        </And>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>StorageClass</Field>
        <Field>ETag</Field>
        <Field>Tag</Field>
    </OptionalFields>
</InventoryConfiguration>
```

响应

上述请求后，COS 返回以下响应，表明该清单任务 list2 已经成功设置完毕。

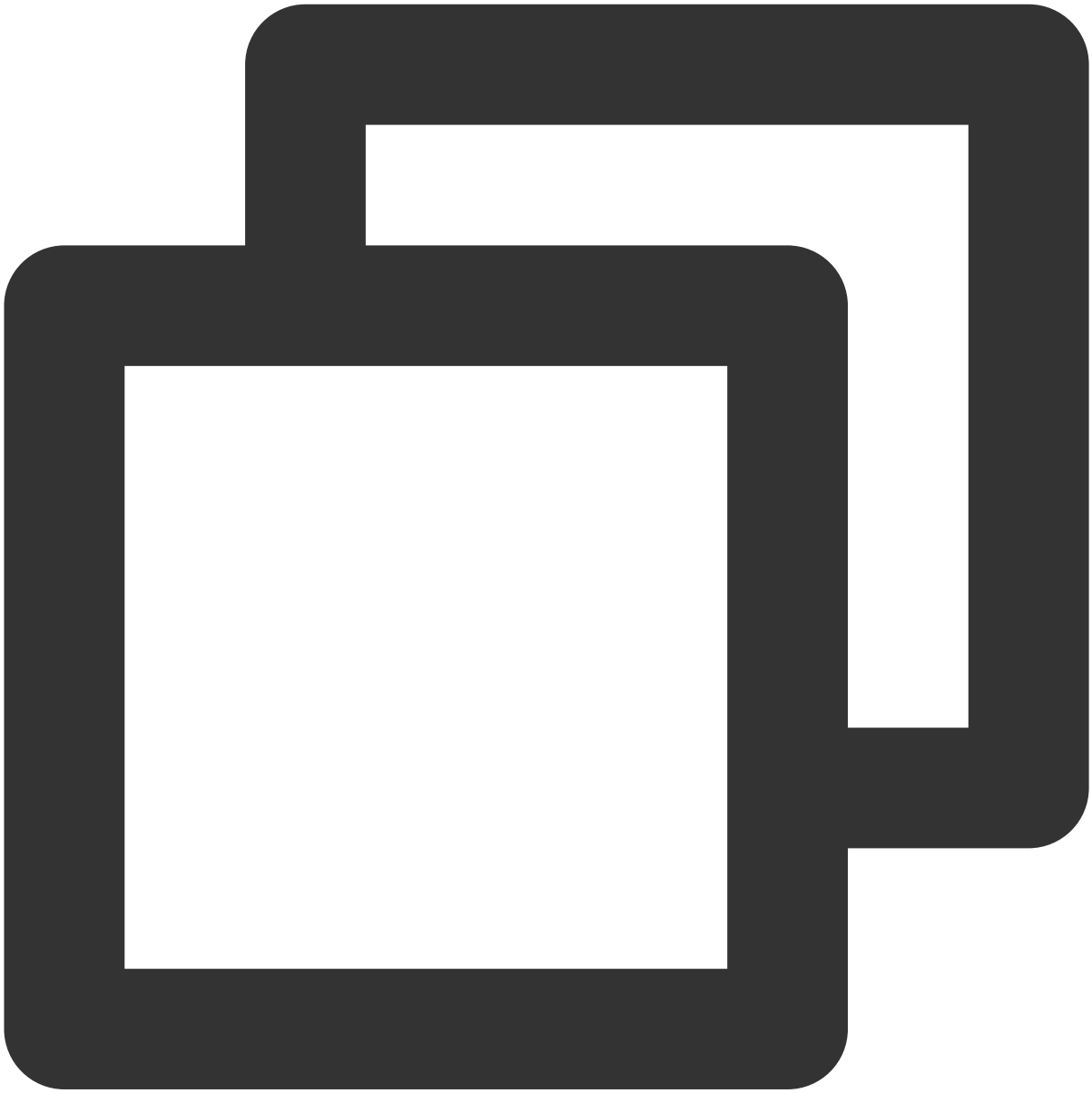


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, 28 Aug 2018 02:53:38 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF8****
```

案例三：清单支持输出对象的自定义头部

请求

该示例向存储桶 `examplebucket-1250000000` 中添加一条名为 list3 的清单任务。
该清单任务分析存储桶中前缀为 myPrefix ， 且有对象标签含有{age:18}的对象及其所有版本。
分析频次为每天一次。
分析维度包括 Size ， LastModifiedDate， StorageClass，ETag，Tag，x-cos-meta-myheader。
分析结果将以 CSV 格式文件存储在存储桶 inventorybucket-1250000000 中。

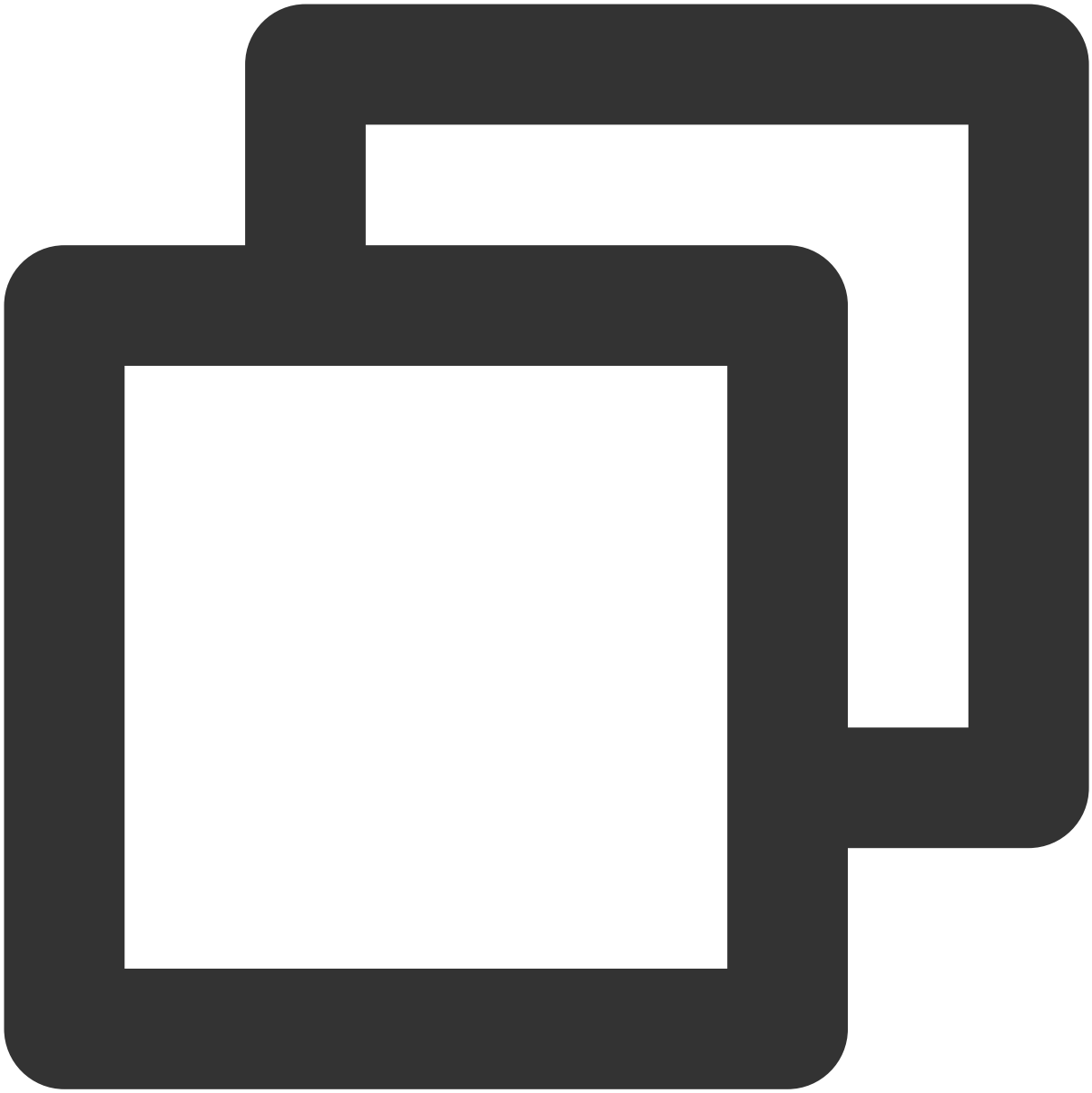


```
PUT /?inventory&id=list2 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Content-MD5: AAq9nzrpsz5LJ4UEe1f6Q==
```

```
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 1024

<?xml version = "1.0" encoding = "UTF-8">
<InventoryConfiguration xmlns = "http://....">
    <Id>list3</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>100000000001</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::inventorybucket-1250000000</Bucket>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <And>
            <Prefix>myPrefix</Prefix>
            <Tag>
                <Key>age</Key>
                <Value>18</Value>
            </Tag>
        </And>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>StorageClass</Field>
        <Field>ETag</Field>
        <Field>Tag</Field>
        <Field>Crc64</Field>
        <Field>x-cos-meta-myheader</Field>
    </OptionalFields>
</InventoryConfiguration>
```

响应

上述请求后，COS 返回以下响应，表明该清单任务 list3 已经成功设置完毕。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, 28 Aug 2018 02:53:38 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF8****
```

POST Bucket inventory

最近更新时间：2023-12-21 12:32:01

功能描述

POST Bucket inventory 用于在一个存储桶中创建一个一次性清单任务。该接口与 PUT Bucket inventory 有所区别，这个清单任务创建后将立即开始执行，并且每个任务只会执行一次，而不会周期性地重复执行。通过使用此接口，您能够以更加灵活的方式获取到存储桶中的对象清单，进而更精细化地管理对象。

注意

您必须在目标存储桶中写入存储桶策略，以供 COS 将清单任务的结果文件写入该存储桶中。

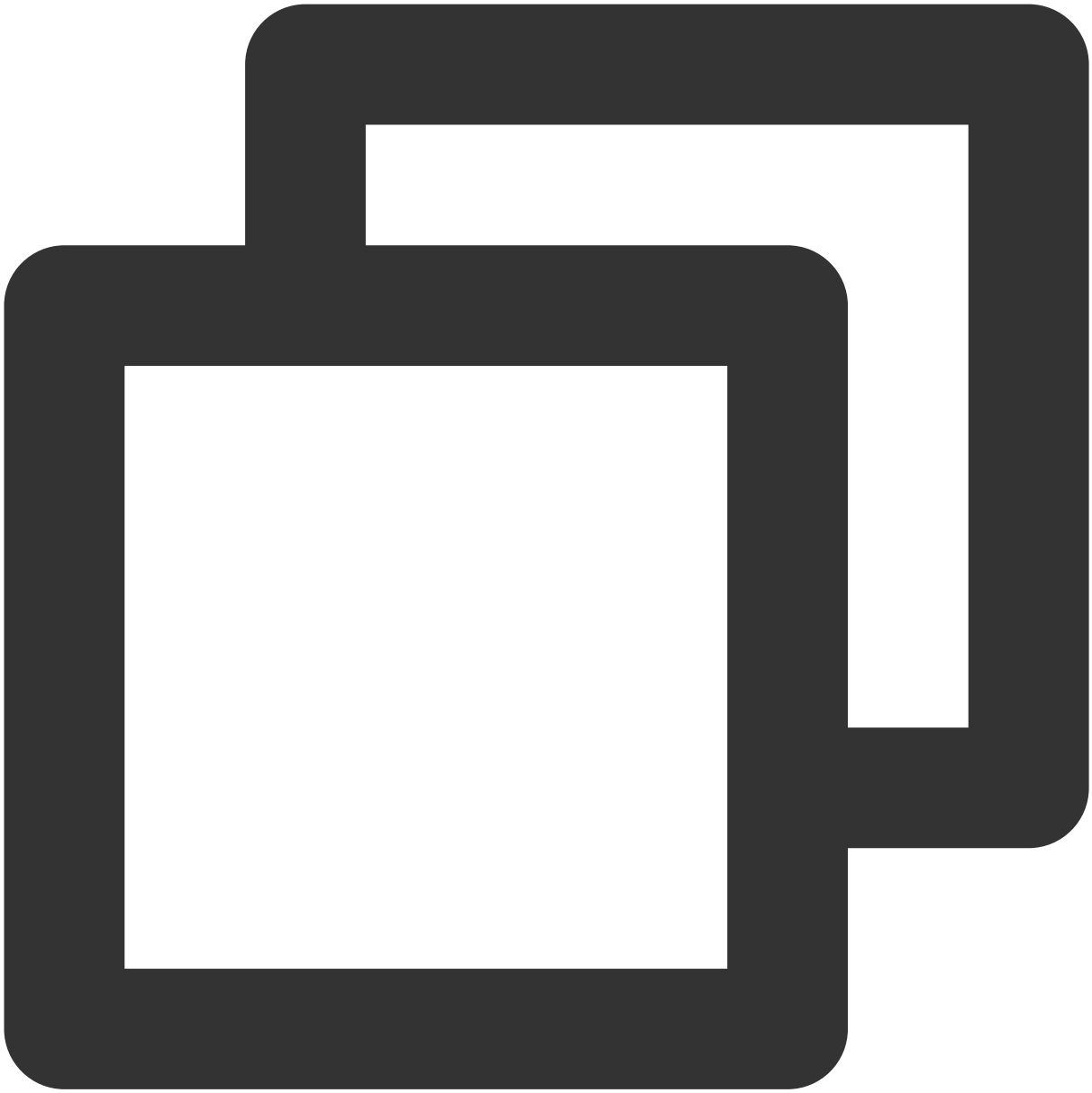
调用该请求时，请确保您有足够的权限对存储桶的清单任务进行操作。存储桶所有者默认拥有该权限，若您无该项权限，请先向存储桶所有者申请该项操作的权限。

如果您指定了清单投递的前缀，COS 后端会自动在您指定的前缀后边加上 `/` 。如您指定了 `Prefix` 作为前缀，则 COS 后端投递的清单报告路径为 `Prefix/inventory_report` 。

在清单任务未完成前，再次重复提交相同 id 的清单任务，或者使用与周期性清单任务相同的 id，服务端会返回重复清单任务的错误。

请求

请求示例



```
POST /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Content-MD5: MD5
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

调用 POST Bucket inventory 需要使用清单任务名称的参数。该参数格式如下：

参数	描述	类型	是否必选
id	清单任务的名称。缺省值：None 合法字符：`a-z，A-Z，0-9，-，_，.` 注意：清单任务 id 建议不要与周期清单任务 id 相同。同一天内不允许提交重复的 id，否则会返回失败。	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

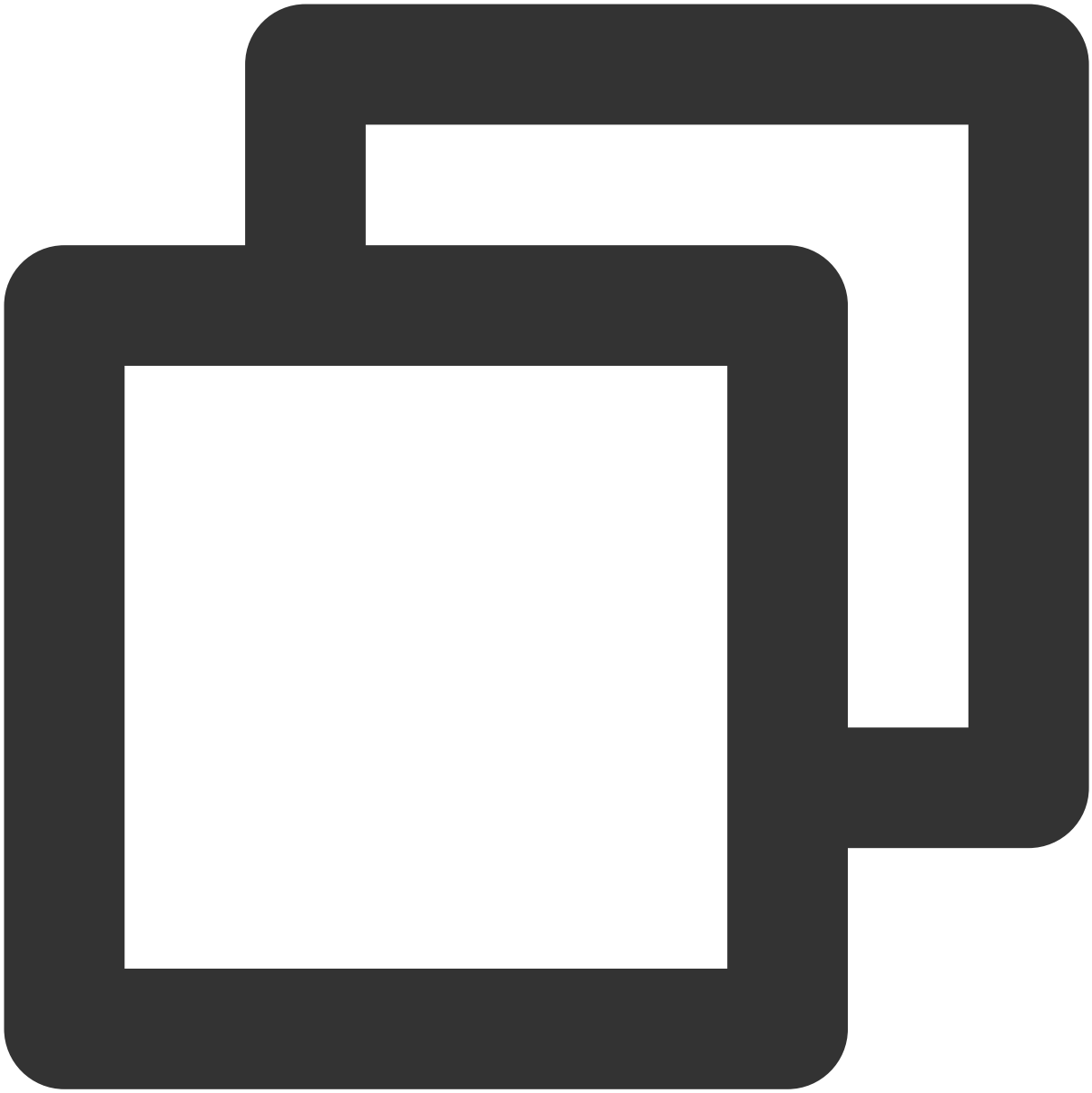
用户在请求体中使用 XML 语言设置清单任务的具体配置信息，示例如下：

注意

区别于传统的周期任务形式的清单规则，一次性清单不允许携带 IsEnabled、Schedule 参数。

其余参数与 PUT Bucket inventory 完全一致。

版权所有：腾讯云计算（北京）有限责任公司



```
<InventoryConfiguration>
    <Id>list1</Id>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>100000000001</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
            <Prefix>list1</Prefix>
            <Encryption>
                <SSE-COS></SSE-COS>
            </Encryption>
```

```
        </COSBucketDestination>
    </Destination>
    <Filter>
        <And>
            <Prefix>myPrefix</Prefix>
            <Tag>
                <Key>string</Key>
                <Value>string</Value>
            </Tag>
        </And>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>ETag</Field>
        <Field>StorageClass</Field>
        <Field>IsMultipartUploaded</Field>
        <Field>ReplicationStatus</Field>
    </OptionalFields>
</InventoryConfiguration>
```

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求的响应体返回为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

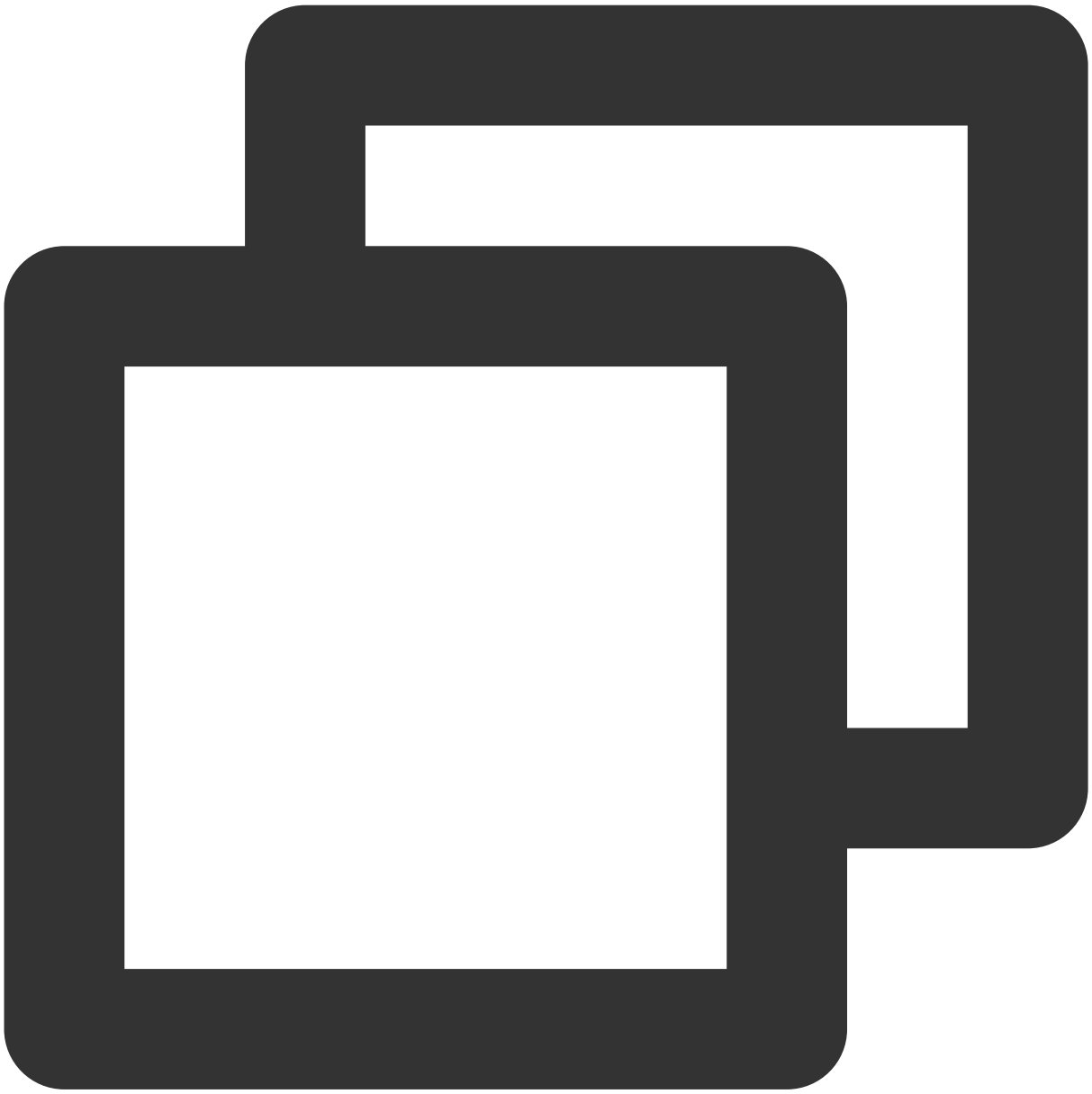
实际案例

案例：使用特定前缀过滤对象生成一次性清单

请求

该示例向存储桶 `examplebucket-1250000000` 中发起一条清单任务。
该清单任务分析存储桶中前缀为 myPrefix，且对象标签含有{age:18}的对象及其所有版本。

分析维度包括 Size，LastModifiedDate，StorageClass，ETag，Tag。
分析结果将以 CSV 格式文件存储在存储桶 inventorybucket-1250000000 中。



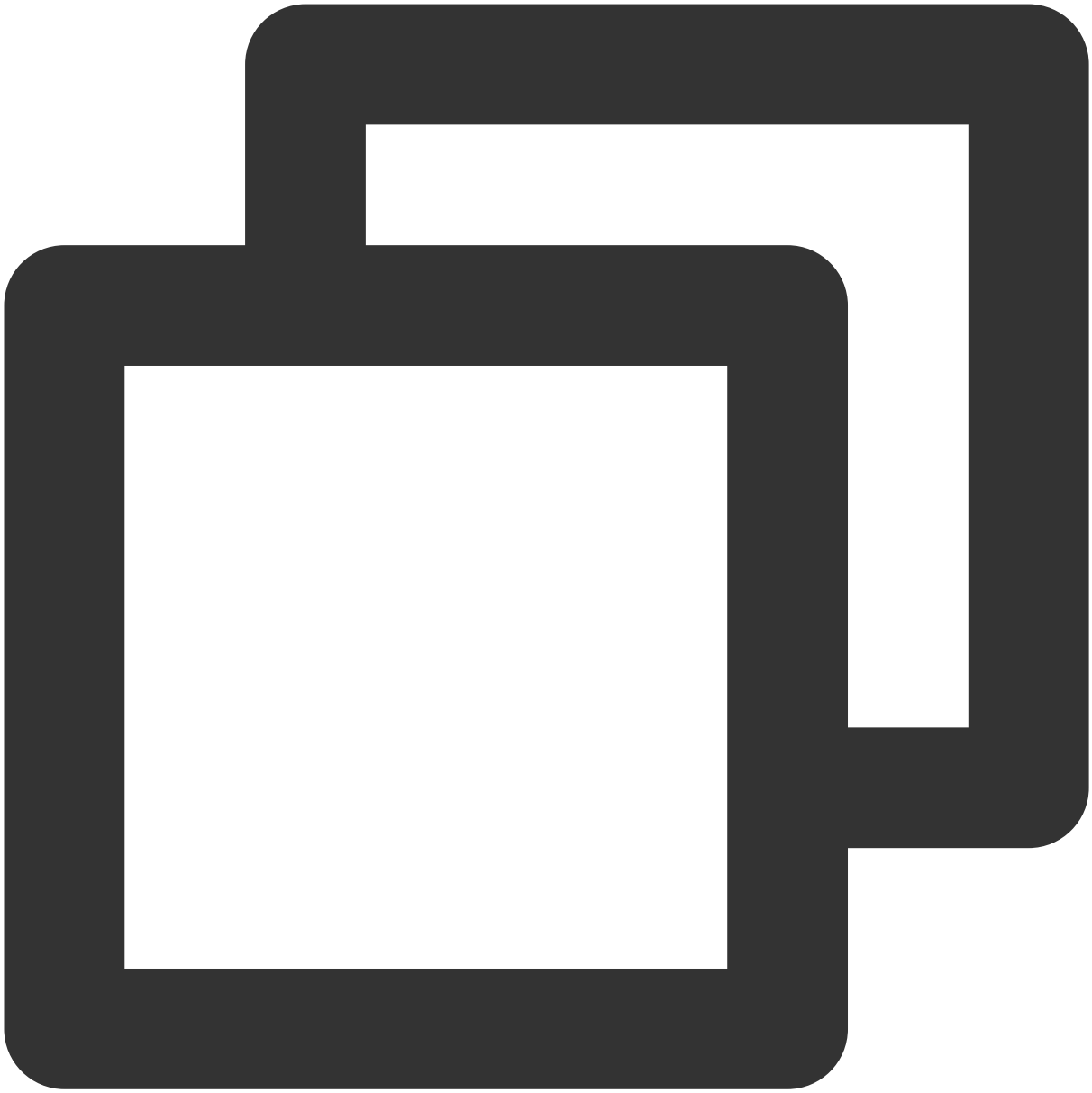
```
POST /?inventory&id=disposable HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Content-MD5: AAq9nzrpsz5LJ4UEe1f6Q==
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 1024

<?xml version = "1.0" encoding = "UTF-8">
```

```
<InventoryConfiguration xmlns = "http://....">
    <Id>disposable</Id>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>100000000001</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::inventorybucket-1250000000</Bucket>
        </COSBucketDestination>
    </Destination>
    <Filter>
        <And>
            <Prefix>myPrefix</Prefix>
            <Tag>
                <Key>age</Key>
                <Value>18</Value>
            </Tag>
        </And>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>StorageClass</Field>
        <Field>ETag</Field>
        <Field>Tag</Field>
    </OptionalFields>
</InventoryConfiguration>
```

响应

上述请求后，COS 返回以下响应，表明该清单任务已经成功提交。

版权所有：腾讯云计算（北京）有限责任公司



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, 28 Aug 2018 02:53:38 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF8****
```

GET Bucket inventory

最近更新时间：2023-12-21 12:32:26

功能描述

GET Bucket inventory 接口用于查询存储桶中用户的清单任务信息。用户在发起该请求时，需要用户提供清单任务的名称，发起该请求时需获得请求签名，表明该请求已获得许可。有关清单的详细特性，请参见 清单功能概述。

注意

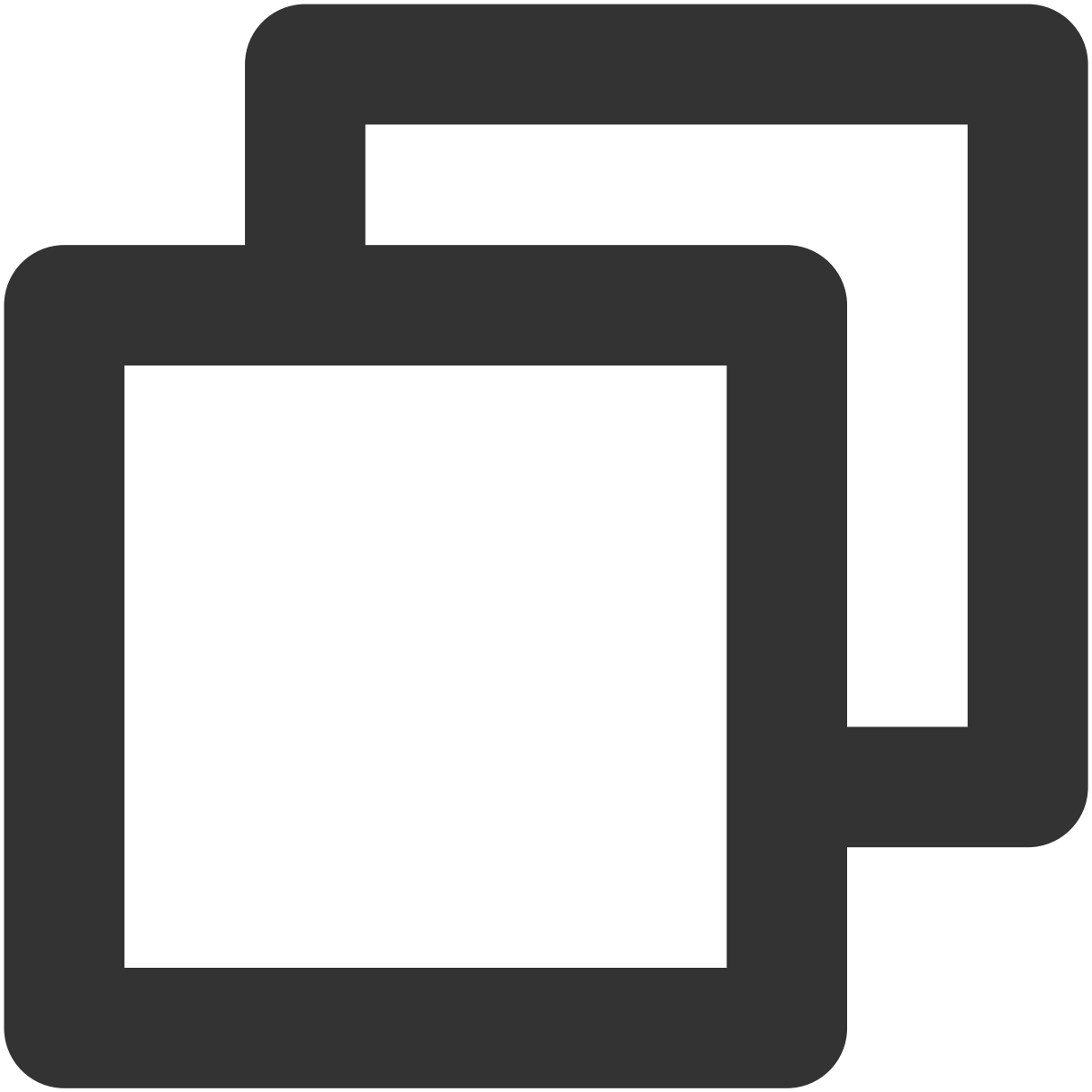
调用该请求时，请确保您有足够的权限对存储桶的清单任务进行操作。

存储桶所有者默认拥有该权限，若您无该项权限，请先向存储桶所有者申请该项操作的权限。

如果您指定了清单投递的前缀，COS 后端会自动在您指定的前缀后边加上 `/` 。如您指定了 `Prefix` 作为前缀，则 COS 后端投递的清单报告路径为 `Prefix/inventory_report` 。

请求

请求示例



```
GET /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

调用 GET Bucket inventory 需要使用清单任务名称的参数。该参数格式如下：

参数	描述	类型	是否必选
id	清单任务的名称。缺省值：None 合法字符：`a-z，A-Z，0-9，-，_，.`	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

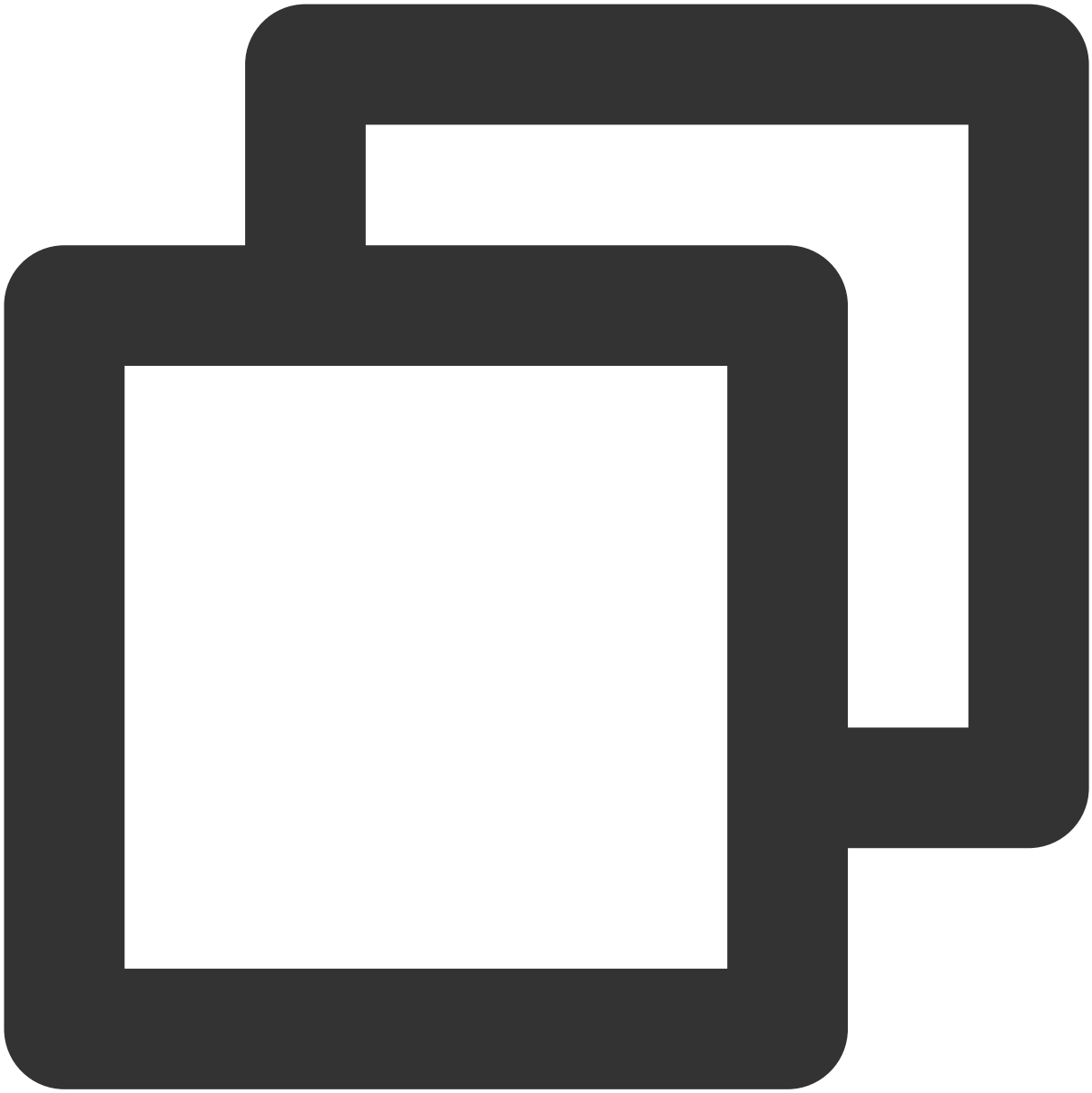
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<InventoryConfiguration>
    <Id>list1</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>1250000000</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
            <Prefix>list1</Prefix>
            <Encryption>
                <SSE-COS></SSE-COS>
```

```
            </Encryption>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <Prefix>myPrefix</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>ETag</Field>
        <Field>StorageClass</Field>
        <Field>IsMultipartUploaded</Field>
        <Field>ReplicationStatus</Field>
        <Field>Tag</Field>
        <Field>Crc64</Field>
        <Field>x-cos-meta-*</Field>
    </OptionalFields>
</InventoryConfiguration>
```

具体的节点描述如下：

节点名	父节点	描述
Inventory Configuration	无	包含清单的配置参数
Id	Inventory Configuration	清单的名称，与请求参数中的 ID 对应
IsEnabled	Inventory Configuration	清单是否启用的标识： 如果设置为 true，清单功能将生效 如果设置为 false，将不生成任何清单
IncludedObject Versions	Inventory Configuration	是否在清单中包含对象版本： 如果设置为 All，清单中将会包含所有对象版本，并在清单中增加 VersionId 如果设置为 Current，则清单中不包含对象版本信息
Filter	Inventory Configuration	筛选待分析对象。清单功能将分析符合 Filter 中设置的前缀的对象
And	Filter	筛选待分析对象时，如果同时需要前缀与对象标签条件，需要用 And 包装
Prefix	And	需要分析的对象的前缀

Tag	And	筛选待分析对象时，可以用对象标签（支持多个）作为过滤条件
Period	Filter	需要分析的对象的创建时间范围
StartTime	Period	需要分析的对象创建的起始时间，参数为秒级时间戳，如1568688761
EndTime	Period	需要分析的对象创建的结束时间，参数为秒级时间戳，如1568688762
OptionalFields	Inventory Configuration	设置清单结果中应包含的分析维度
Field	OptionalFields	清单结果中可选包含的分析项目名称，可选字段包括：`Size`，`LastModifiedDate`，`StorageClass`，`ETag`，`cos-meta-*` 注意，如果筛选条件里使用了对象标签，在这里也必须添加 Tag 此外，也支持用户填写 `x-cos-meta-*` 形式的自定义头部，如 `x-cos-m` 则为空
Schedule	Inventory Configuration	配置清单任务周期
Frequency	Schedule	清单任务周期，可选项为按日或者按周
Destination	Inventory Configuration	描述存放清单结果的信息
COSBucket Destination	Destination	清单结果导出后存放的存储桶信息
Bucket	COSBucket Destination	清单分析结果的存储桶名
AccountId	COSBucket Destination	存储桶的所有者 ID
Prefix	COSBucket Destination	清单分析结果的前缀
Format	COSBucket Destination	清单分析结果的文件形式，可选项为 CSV 格式和 ORC 格式
Encryption	COSBucket Destination	为清单结果提供服务端加密的选项
SSE-COS	Encryption	COS 托管密钥的加密方式

错误码

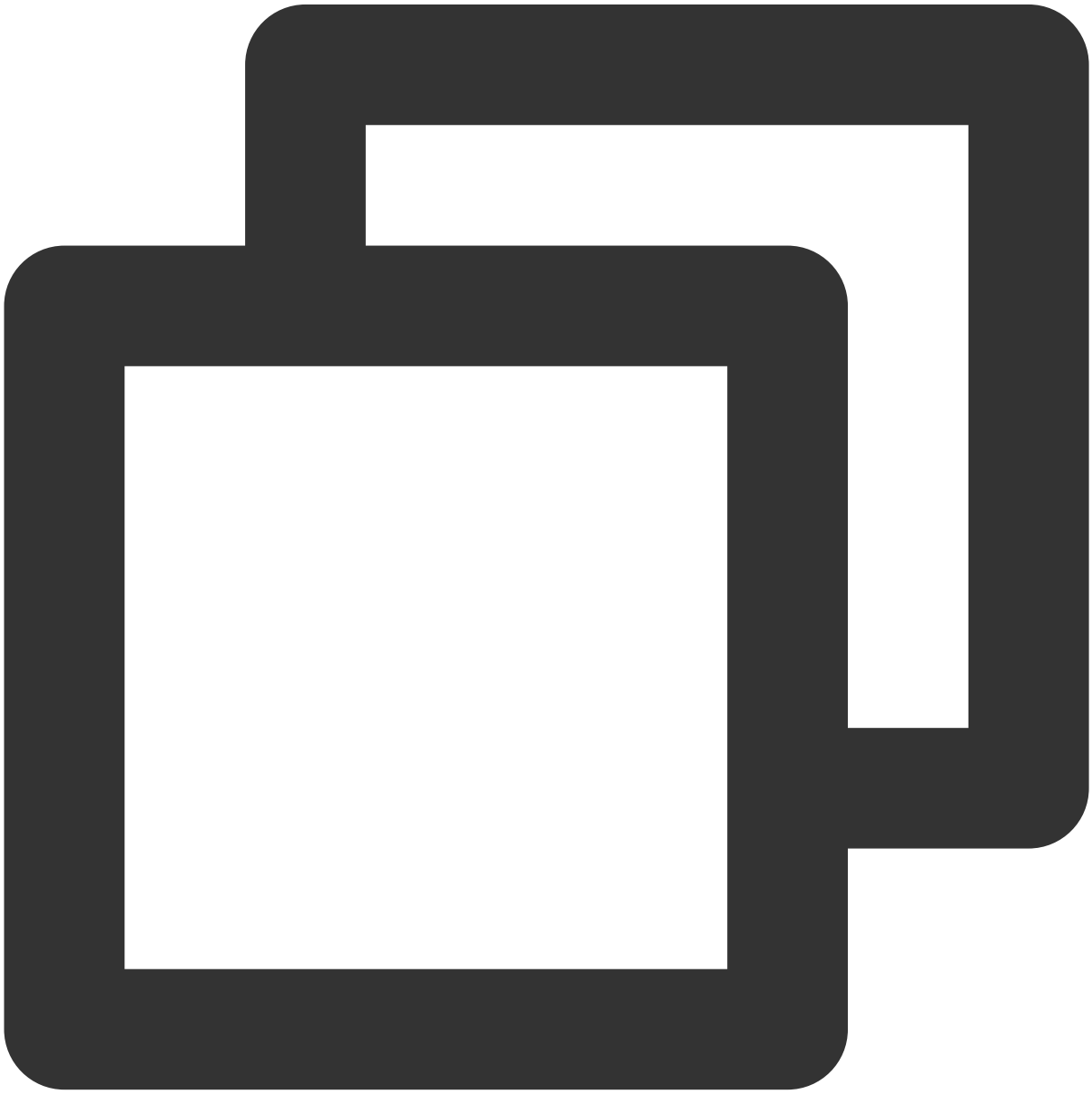
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：获取指定对象前缀的清单任务

请求

下述请求示例展示了从存储桶 `examplebucket-1250000000` 中获取清单任务为 list1 的配置信息。



```
GET /?inventory&id=list1 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
```

响应

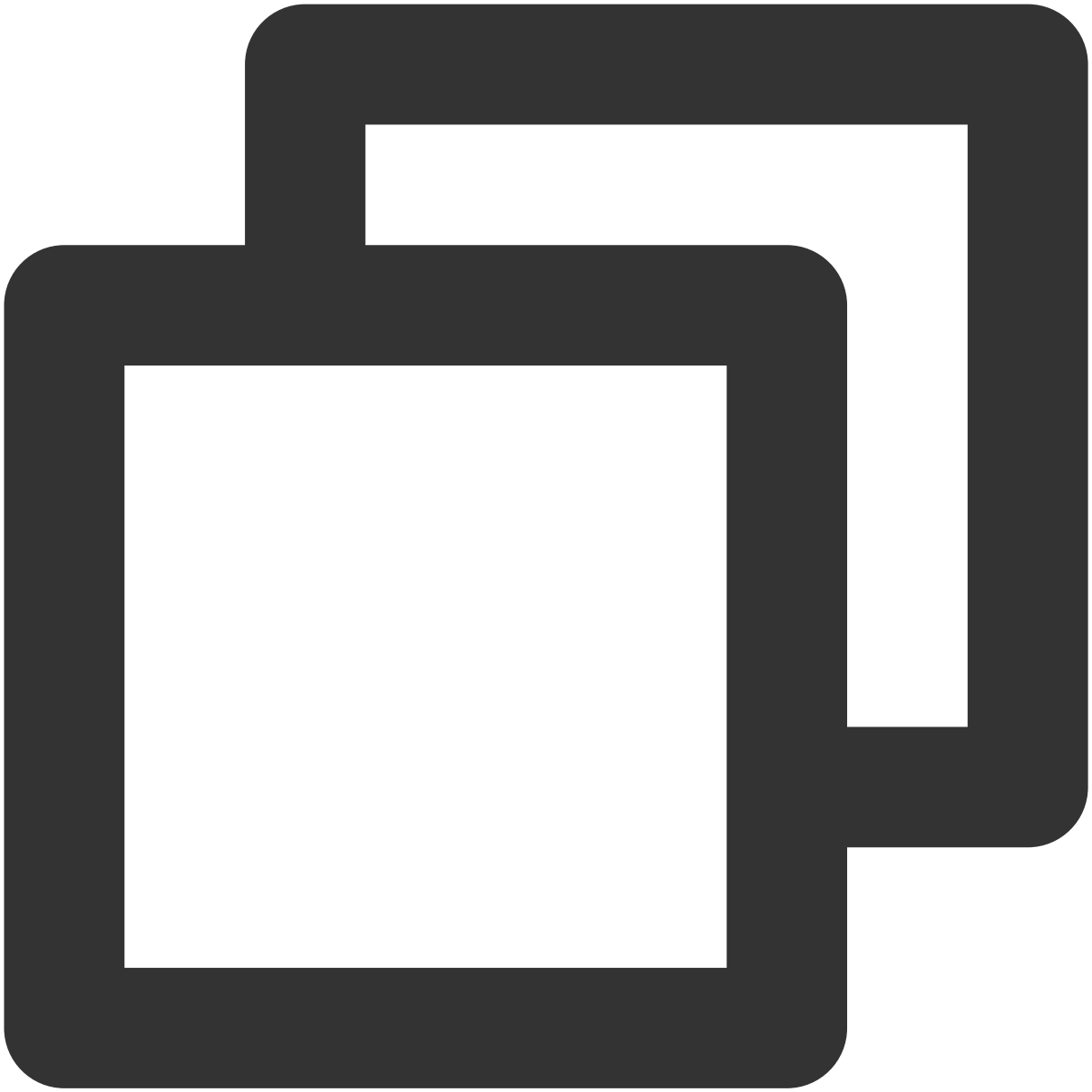
上述请求后，COS 返回以下响应，表明当前该存储桶内的清单任务 list1 处于启用状态。

该清单任务分析存储桶 `examplebucket-1250000000` 中前缀为 myPrefix 的对象及其所有版本。

分析频次为每天一次。

分析维度包括 Size，LastModifiedDate， StorageClass，ETag，IsMultipartUploaded， ReplicationStatus。

分析结果将以 CSV 格式文件存储在存储桶 examplebucket-1250000000 中，文件添加前缀 list1 且用 SSE-COS 加密。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 331
Date: Mon, 28 Aug 2018 02:53:39 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF84Y2M
<?xml version = "1.0" encoding = "UTF-8">
<InventoryConfiguration xmlns = "http://....">
    <Id>list1</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
```

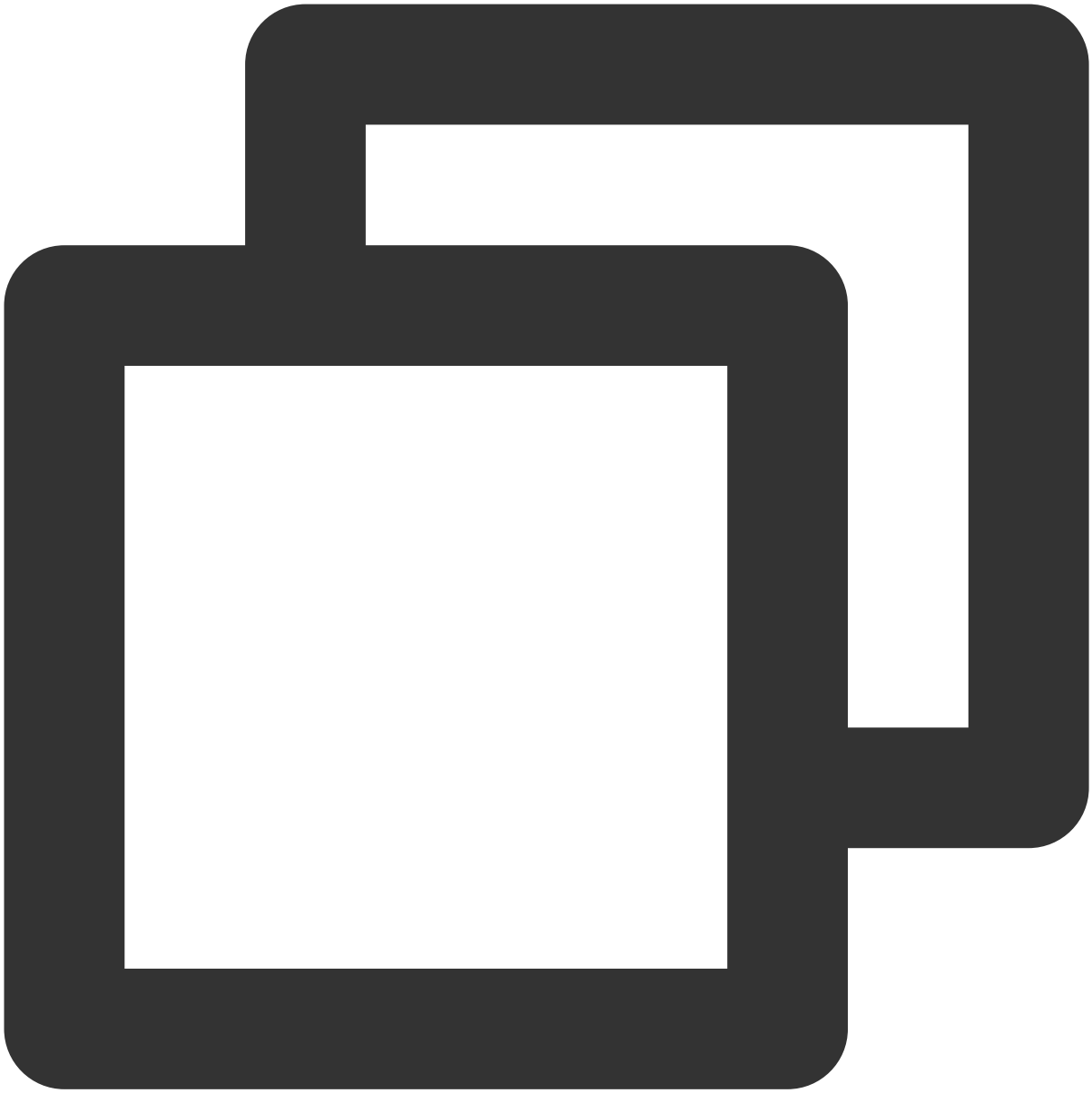
版权所有：腾讯云计算（北京）有限责任公司

```
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>1250000000</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
            <Prefix>list1</Prefix>
            <Encryption>
                <SSE-COS></SSE-COS>
            </Encryption>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <Prefix>myPrefix</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>ETag</Field>
        <Field>StorageClass</Field>
        <Field>IsMultipartUploaded</Field>
        <Field>ReplicationStatus</Field>
    </OptionalFields>
</InventoryConfiguration>
```

案例二：获取指定对象前缀和对象标签的清单任务

请求

下述请求示例展示了从存储桶 `examplebucket-1250000000` 中获取清单任务为 list2 的配置信息。



```
GET /?inventory&id=list2 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
```

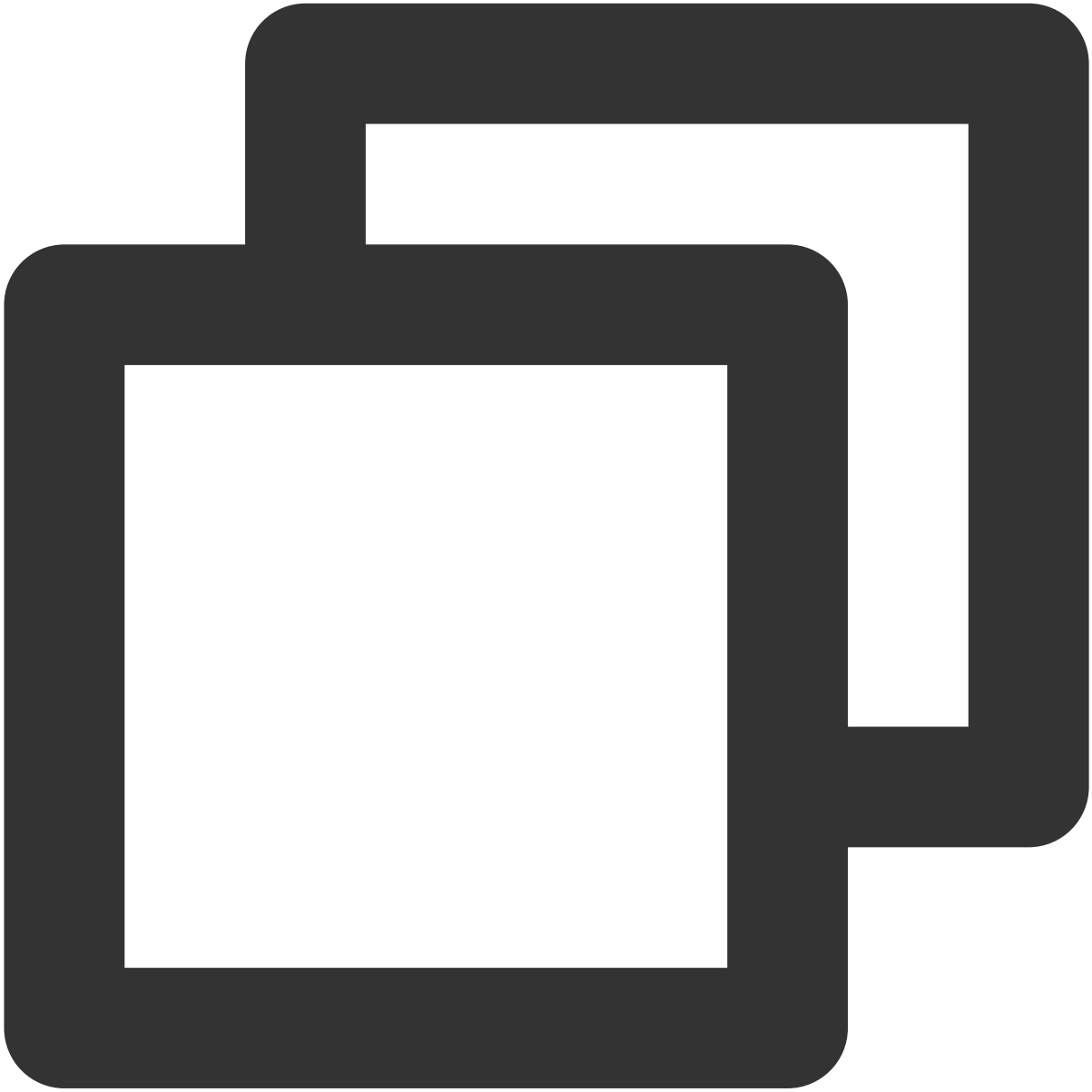
响应

上述请求后，COS 返回以下响应，表明当前该存储桶内的清单任务 list2 处于启用状态。
该清单任务分析存储桶 `examplebucket-1250000000` 中前缀为 myPrefix，且含有对象标签{age:18}的对象及其所有版本。

分析频次为每天一次。

分析维度包括 Size，LastModifiedDate， StorageClass，ETag，Tag。

分析结果将以 CSV 格式文件存储在存储桶 inventorybucket-1250000000 中。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 331
Date: Mon, 28 Aug 2018 02:53:39 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF84Y2M
<?xml version = "1.0" encoding = "UTF-8">
```

版权所有：腾讯云计算（北京）有限责任公司

```
<InventoryConfiguration xmlns = "http://....">
    <Id>list2</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
        <COSBucketDestination>
            <Format>CSV</Format>
            <AccountId>1250000000</AccountId>
            <Bucket>qcs::cos:ap-guangzhou::inventorybucket-1250000000</Bucket>
        </COSBucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <And>
            <Prefix>myPrefix</Prefix>
            <Tag>
                <Key>age</Key>
                <Value>18</Value>
            </Tag>
        </And>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>StorageClass</Field>
        <Field>ETag</Field>
        <Field>Tag</Field>
    </OptionalFields>
</InventoryConfiguration>
```

版权所有：腾讯云计算（北京）有限责任公司

List Bucket Inventory Configurations

最近更新时间：2023-12-21 12:32:57

功能描述

List Bucket Inventory Configurations 用于请求返回一个存储桶中的所有清单任务。每一个存储桶中最多配置1000条清单任务。

该请求支持列表分页，每页一次最多返回100条清单任务。请确认请求中的 IsTruncated 节点的值：

如果 IsTruncated 为 false，则表明已经将存储桶中的所有清单任务全部列出。

如果 IsTruncated 为 true，且 NextContinuationToken 节点中存在参数值，则您可以将 NextContinuationToken 节点的值传递至 continuation-token 节点中，获取下一分页的清单任务信息。

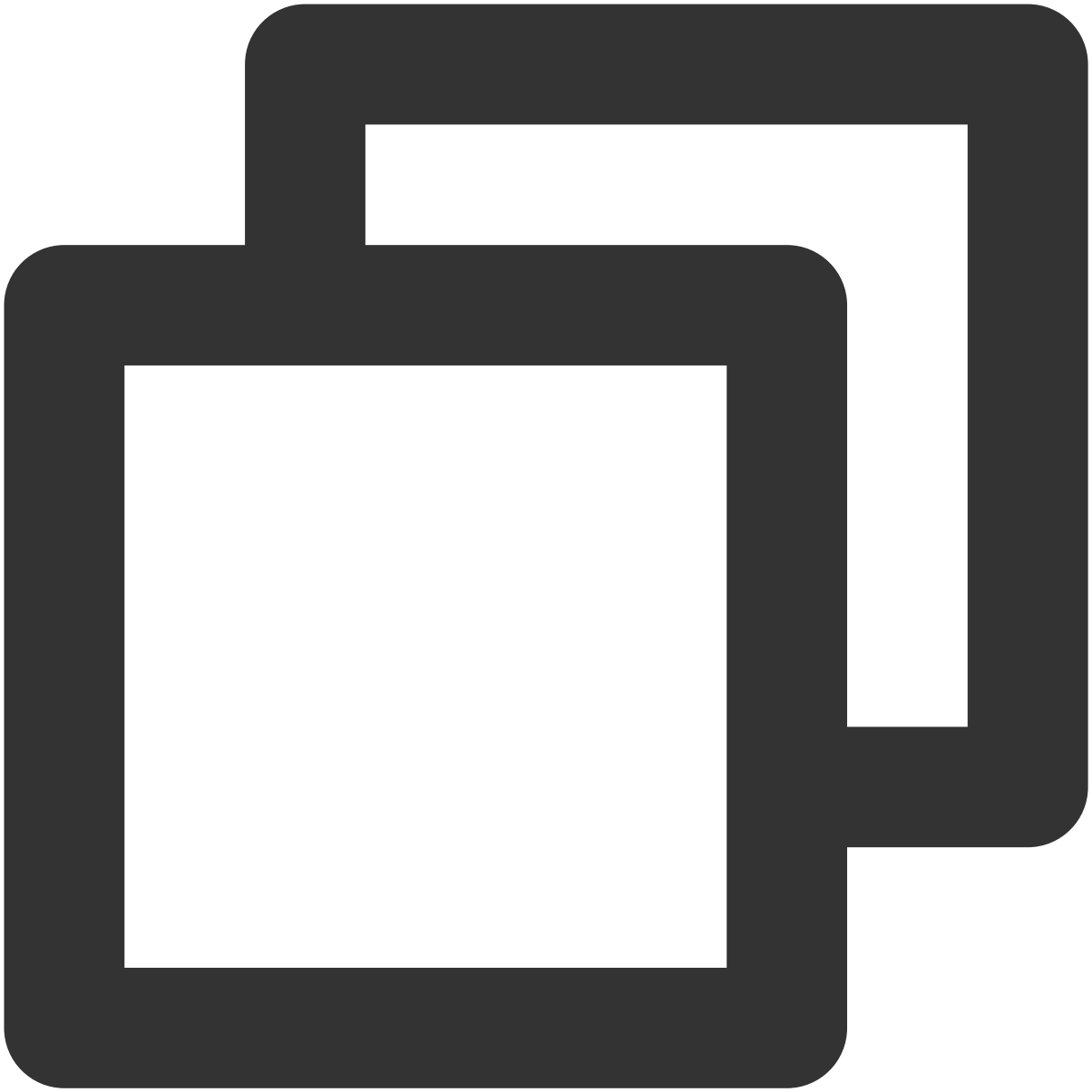
有关清单的详细特性，请参见 清单功能概述。

注意

调用该请求时，请确保您有足够的权限对存储桶的清单任务进行操作。存储桶所有者默认拥有该权限，如您无该项权限，请先向存储桶所有者申请该项操作的权限。

请求

请求示例



```
GET /?inventory HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

请求参数格式如下：

参数	描述	类型	是否必选
continuation-token	当 COS 响应体中 IsTruncated 为 true，且 NextContinuationToken 节点中存在参数值时，您可以将这个参数作为 continuation-token 参数值，以获取下一页的清单任务信息。缺省值：None	String	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

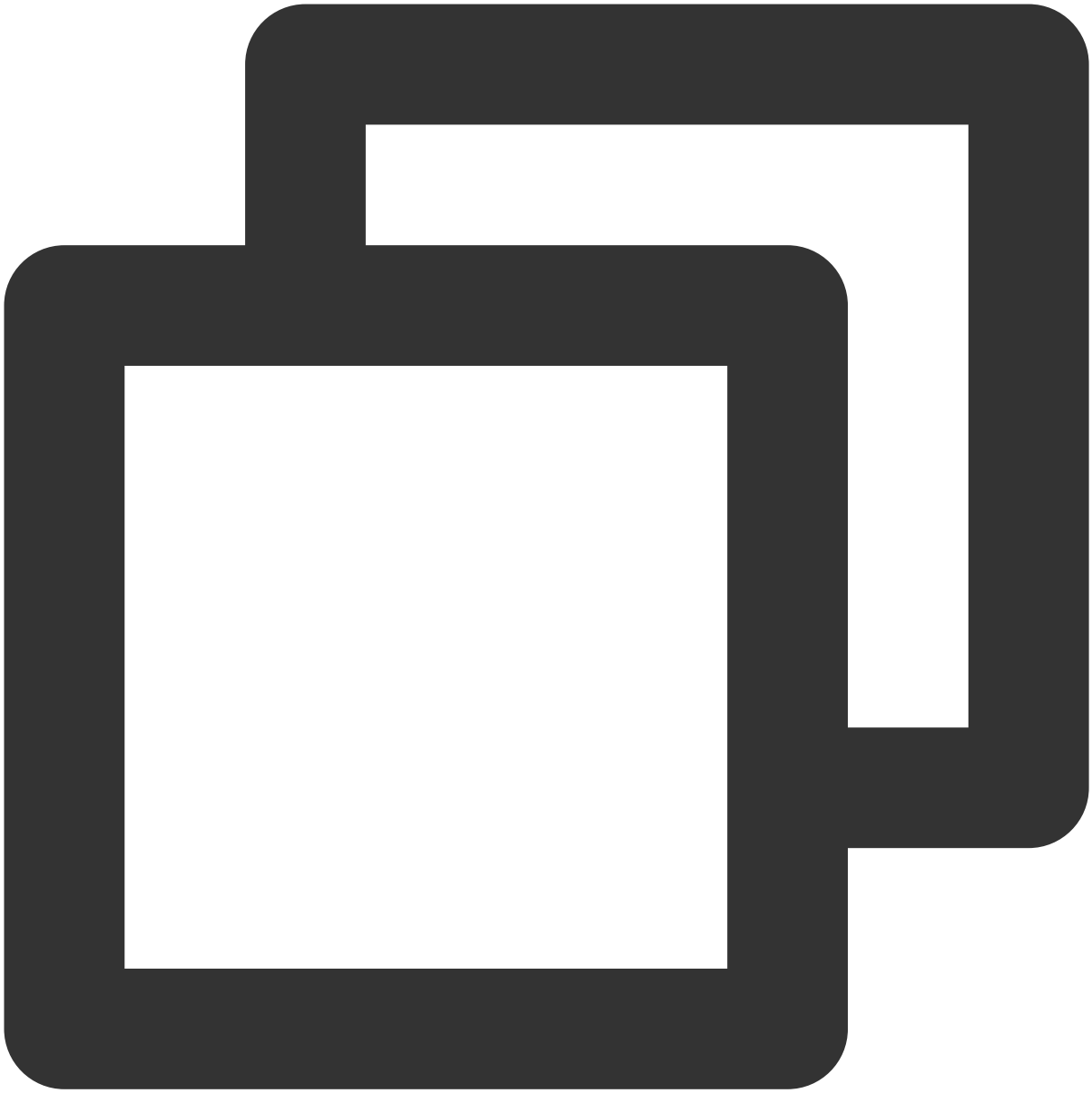
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<ListInventoryConfigurationResult>
    <InventoryConfiguration>
        <Id>list1</Id>
        <IsEnabled>True</IsEnabled>
        <Destination>
            <COSBucketDestination>
                <Format>CSV</Format>
                <AccountId>1250000000</AccountId>
                <Bucket>qcs::cos:ap-beijing::examplebucket-1250000000</Bucket>
                <Prefix>list1</Prefix>
                <SSE-COS></SSE-COS>
```

```
            </COSBucketDestination>
        </Destination>
        <Schedule>
            <Frequency>Daily</Frequency>
        </Schedule>
        <Filter>
            <Prefix>myPrefix</Prefix>
        </Filter>
        <IncludedObjectVersions>All</IncludedObjectVersions>
        <OptionalFields>
            <Field>Size</Field>
            <Field>LastModifiedDate</Field>
            <Field>ETag</Field>
            <Field>StorageClass</Field>
            <Field>IsMultipartUpload</Field>
            <Field>ReplicationStatus</Field>
        </OptionalFields>
    </InventoryConfiguration>
    <InventoryConfiguration>
        <Id>list2</Id>
        <IsEnabled>True</IsEnabled>
        <Destination>
            <COSBucketDestination>
                <Format>CSV</Format>
                <AccountId>1250000000</AccountId>
                <Bucket>qcs::cos:ap-beijing::examplebucket-1250000000</Bucket>
                <Prefix>list2</Prefix>
                <SSE-COS></SSE-COS>
            </COSBucketDestination>
        </Destination>
        <Schedule>
            <Frequency>Weekly</Frequency>
        </Schedule>
        <Filter>
            <Prefix>myPrefix2</Prefix>
        </Filter>
        <IncludedObjectVersions>All</IncludedObjectVersions>
        <OptionalFields>
            <Field>Size</Field>
            <Field>LastModifiedDate</Field>
            <Field>ETag</Field>
            <Field>StorageClass</Field>
        </OptionalFields>
    </InventoryConfiguration>
    <IsTruncated>false</IsTruncated>
    ------If ContinuationToken was provided in the request---
    <ContinuationToken>...</ContinuationToken>
```

```
    <IsTruncated>true</IsTruncated>
    <NextContinuationToken>1ueSDFASDF1Tr/XDAFdadEADadf2J/wm36Hy4vbOwM=</NextContinu
</ListInventoryConfigurationResult>
```

具体的节点描述如下：

节点名	父节点	描述	类型
List InventoryConfigurationResult	无	存储桶中所有清单任务信息的列表	Container
InventoryConfiguration	ListInventoryConfigurationResult	包含清单任务的详细信息，其 XML 结构请参见 GET Bucket inventory	Container
IsTruncated	ListInventoryConfigurationResult	是否已列出所有清单任务信息的标识。如果已经展示完则为 false，否则为 true	Boolean
ContinuationToken	ListInventoryConfigurationResult	当页清单列表的标识，可理解为页数。该标识与请求中的 continuation-token 参数对应	String
NextContinuationToken	ListInventoryConfigurationResult	下一页清单列表的标识。如果该参数中有值，则可将该值作为 continuation-token 参数并发起 GET 请求以获取下一页清单任务信息	String

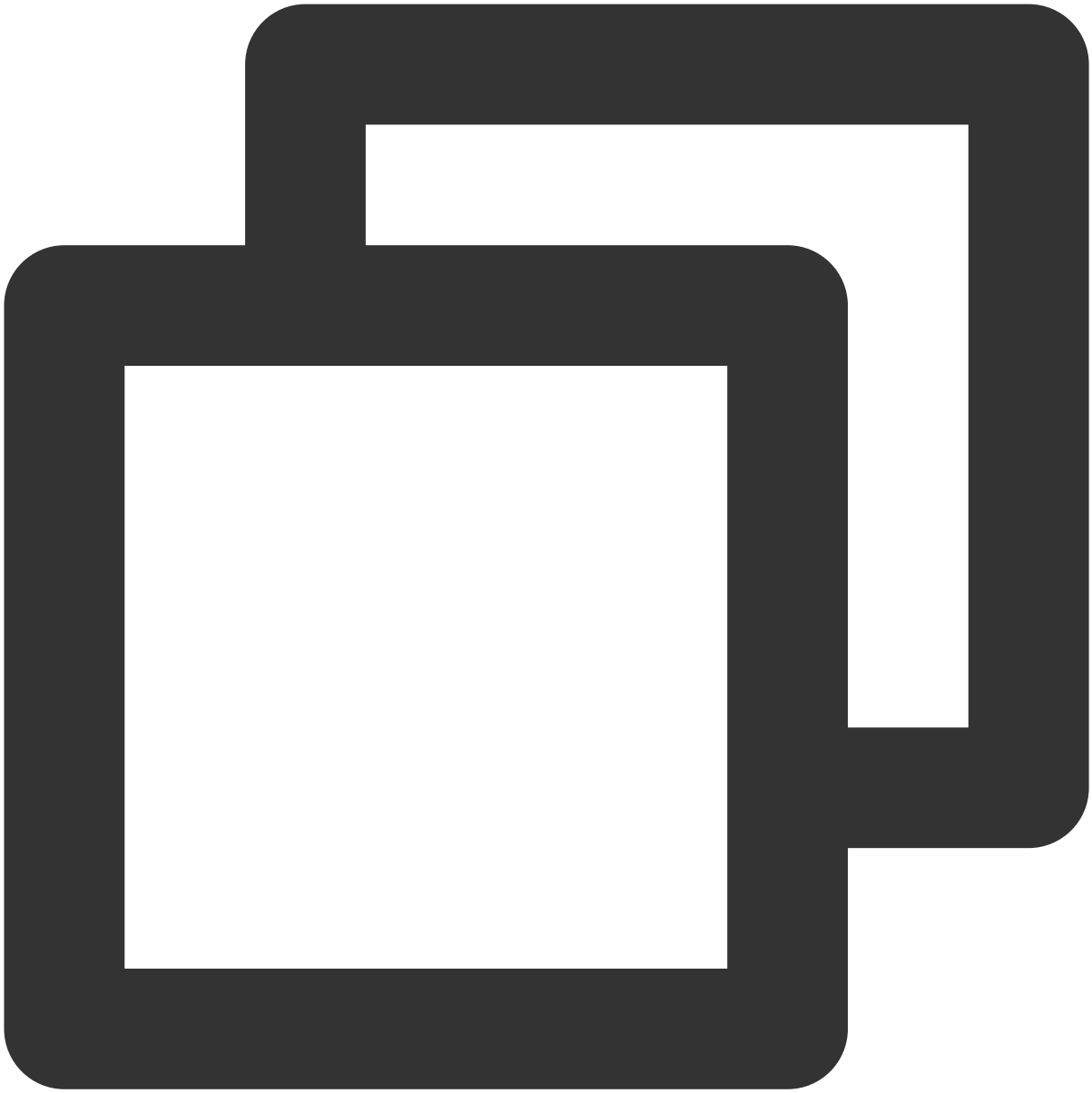
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

下述请求示例展示了从存储桶 examplebucket-1250000000 中获取清单任务为 list1 的配置信息。



```
GET /?inventory HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98JM&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

响应

请求后，COS 返回以下响应，表明当前该存储桶内存在清单任务 list1 和 list2 。

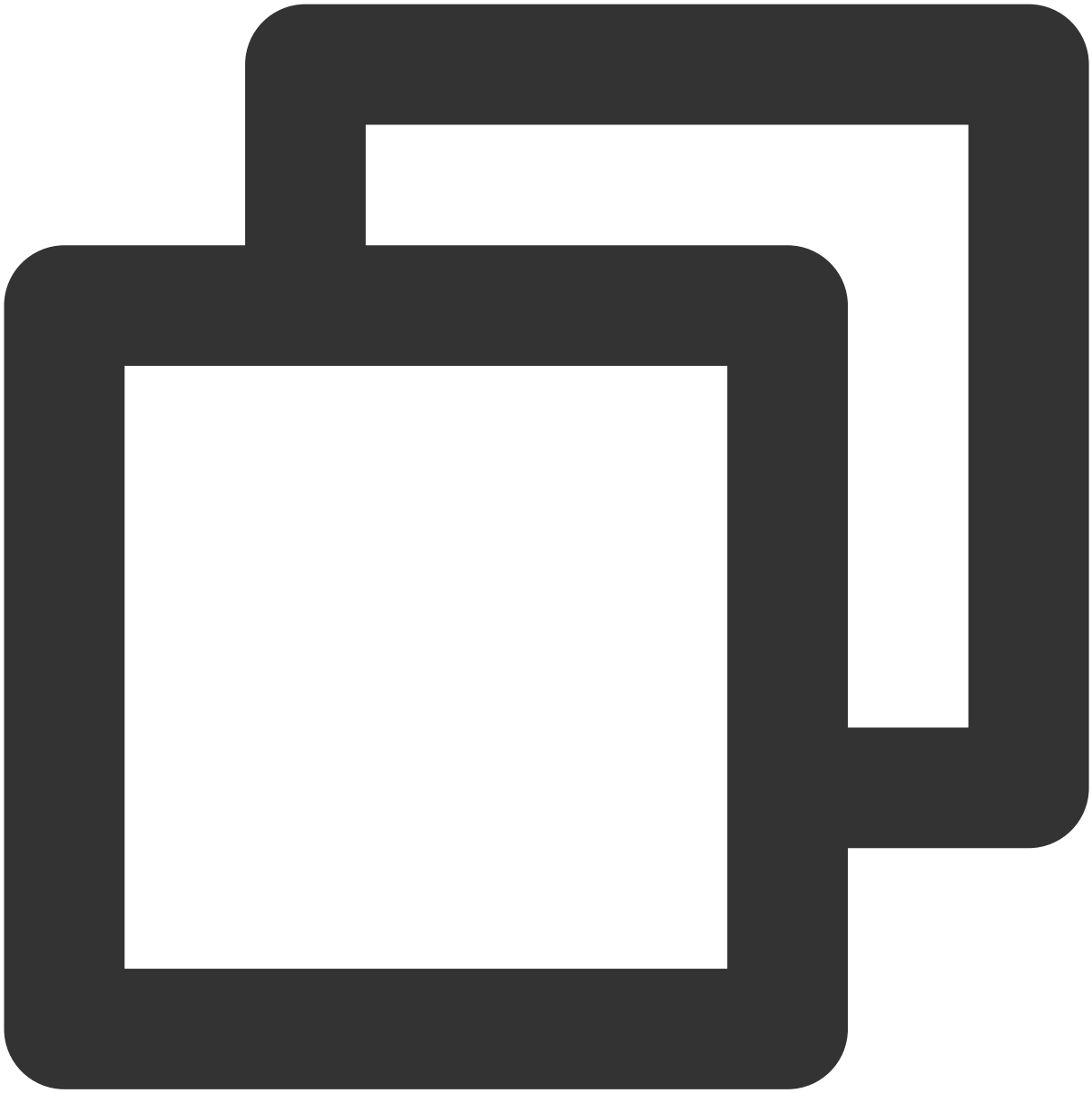
清单任务 list1

分析存储桶 examplebucket-1250000000 中前缀为 myPrefix 的对象及其所有版本。

分析频次为每天一次。
分析维度包括 Size、LastModifiedDate、StorageClass、ETag、IsMultipartUploaded、ReplicationStatus。
分析结果将以 CSV 格式文件存储在存储桶 inventorybucket-1250000000 中，文件添加前缀 list1 且用 SSE-COS 加密。

清单任务 list2

分析存储桶 examplebucket-1250000000 中前缀为 myPrefix2，且含有对象标签{age:18}的对象及其所有版本。
分析频次为每周一次；分析的维度包括 Size， LastModifiedDate， StorageClass， ETag， Tag。
分析结果将以 CSV 格式文件存储在存储桶 inventorybucket-1250000000 中。
假设本页有100条清单任务，当 IsTruncated 为 true 时，COS 将会进一步返回 NextContinuationToken，其中的值可作为 GET 请求中 continuation-token 的参数，以获取下一页信息。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 331
Date: Mon, 28 Aug 2018 02:53:39 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF8****
<?xml version = "1.0" encoding = "UTF-8">
<ListInventoryConfigurationResult xmlns = "http://....">
    <InventoryConfiguration>
        <Id>list1</Id>
        <IsEnabled>True</IsEnabled>
```

```
        <Destination>
            <COSBucketDestination>
                <Format>CSV</Format>
                <AccountId>1250000000</AccountId>
                <Bucket>qcs::cos:ap-beijing::inventorybucket-1250000000</Bucket>
                <Prefix>list1</Prefix>
                <SSE-COS></SSE-COS>
            </COSBucketDestination>
        </Destination>
        <Schedule>
            <Frequency>Daily</Frequency>
        </Schedule>
        <Filter>
            <Prefix>myPrefix</Prefix>
        </Filter>
        <IncludedObjectVersions>All</IncludedObjectVersions>
        <OptionalFields>
            <Field>Size</Field>
            <Field>LastModifiedDate</Field>
            <Field>ETag</Field>
            <Field>StorageClass</Field>
            <Field>IsMultipartUpload</Field>
            <Field>ReplicationStatus</Field>
        </OptionalFields>
    </InventoryConfiguration>
    <InventoryConfiguration>
        <Id>list2</Id>
        <IsEnabled>True</IsEnabled>
        <Destination>
            <COSBucketDestination>
                <Format>CSV</Format>
                <AccountId>1250000000</AccountId>
                <Bucket>qcs::cos:ap-beijing::inventorybucket-1250000000</Bucket>
            </COSBucketDestination>
        </Destination>
        <Schedule>
            <Frequency>Weekly</Frequency>
        </Schedule>
        <Filter>
            <And>
                <Prefix>myPrefix2</Prefix>
                <Tag>
                    <Key>age</Key>
                    <Value>18</Value>
                </Tag>
            </And>
        </Filter>
```

```
        <IncludedObjectVersions>All</IncludedObjectVersions>
        <OptionalFields>
            <Field>Size</Field>
            <Field>LastModifiedDate</Field>
            <Field>StorageClass</Field>
            <Field>ETag</Field>
            <Field>Tag</Field>
        </OptionalFields>
    </InventoryConfiguration>
    <IsTruncated>false</IsTruncated>
    ------If ContinuationToken was provided in the request---
    <ContinuationToken>...</ContinuationToken>
    <IsTruncated>true</IsTruncated>
    <NextContinuationToken>1ueSDFASDF1Tr/XDAFdadEADadf2J/wm36Hy4vbOwM=</NextContinu
</ListInventoryConfigurationResult>
```

DELETE Bucket inventory

最近更新时间：2023-12-21 12:33:20

功能描述

DELETE Bucket inventory 用于删除存储桶中指定的清单任务，用户需提供待删除的清单任务的名称。

有关清单的详细特性，请参见 清单功能概述。

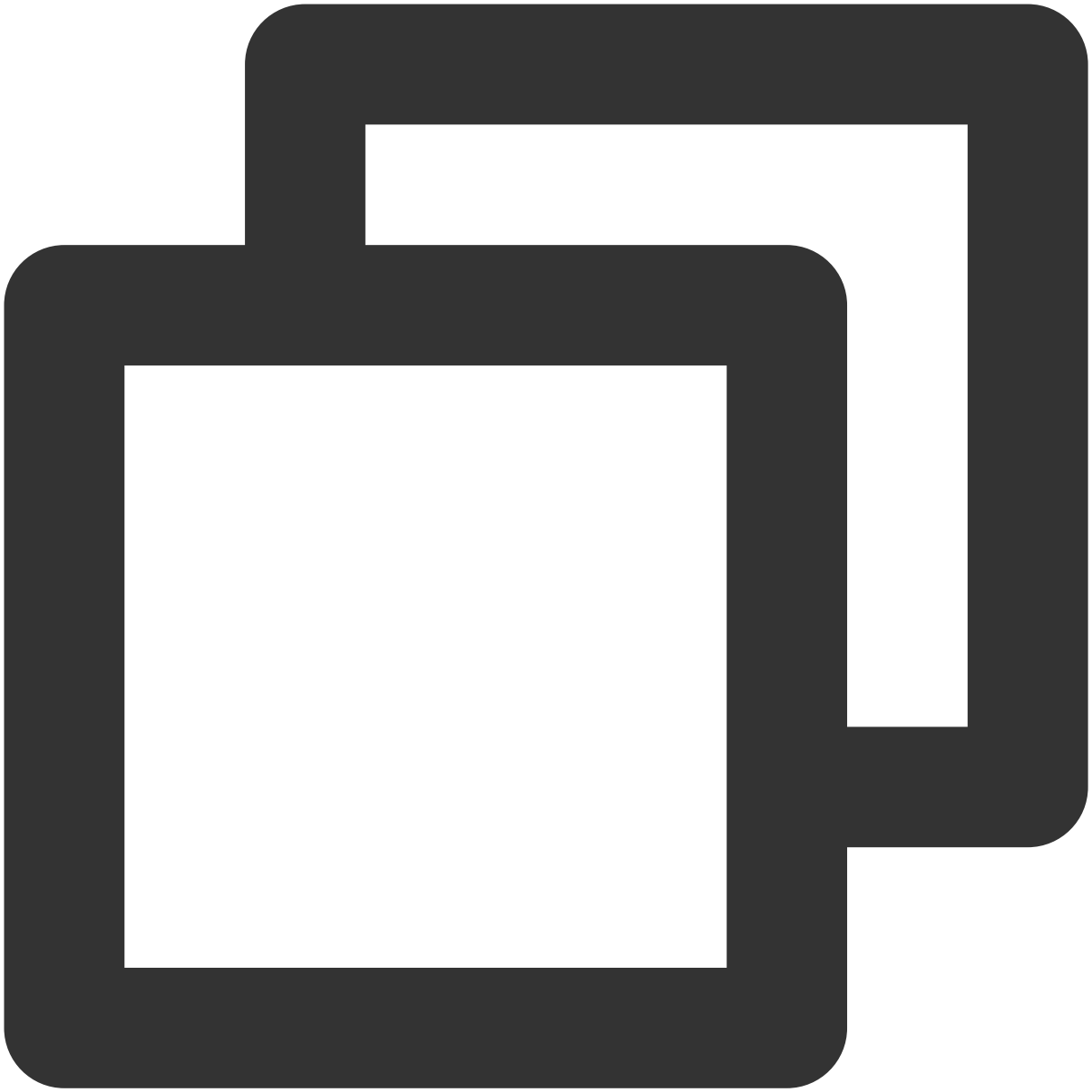
注意

调用该请求时，请确保您有足够的权限对存储桶的清单任务进行操作。

存储桶所有者默认拥有该权限，如您无该项权限，请先向存储桶所有者申请该项操作的权限。

请求

请求示例



```
DELETE /?inventory&id=inventory-configuration-id HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

调用 DELETE Bucket inventory 需要使用清单任务名称的参数。该参数格式如下：

参数	描述	类型	必选
id	清单任务的名称。缺省值：None 合法字符：`a-z，A-Z，0-9，-，_，.`	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为空。

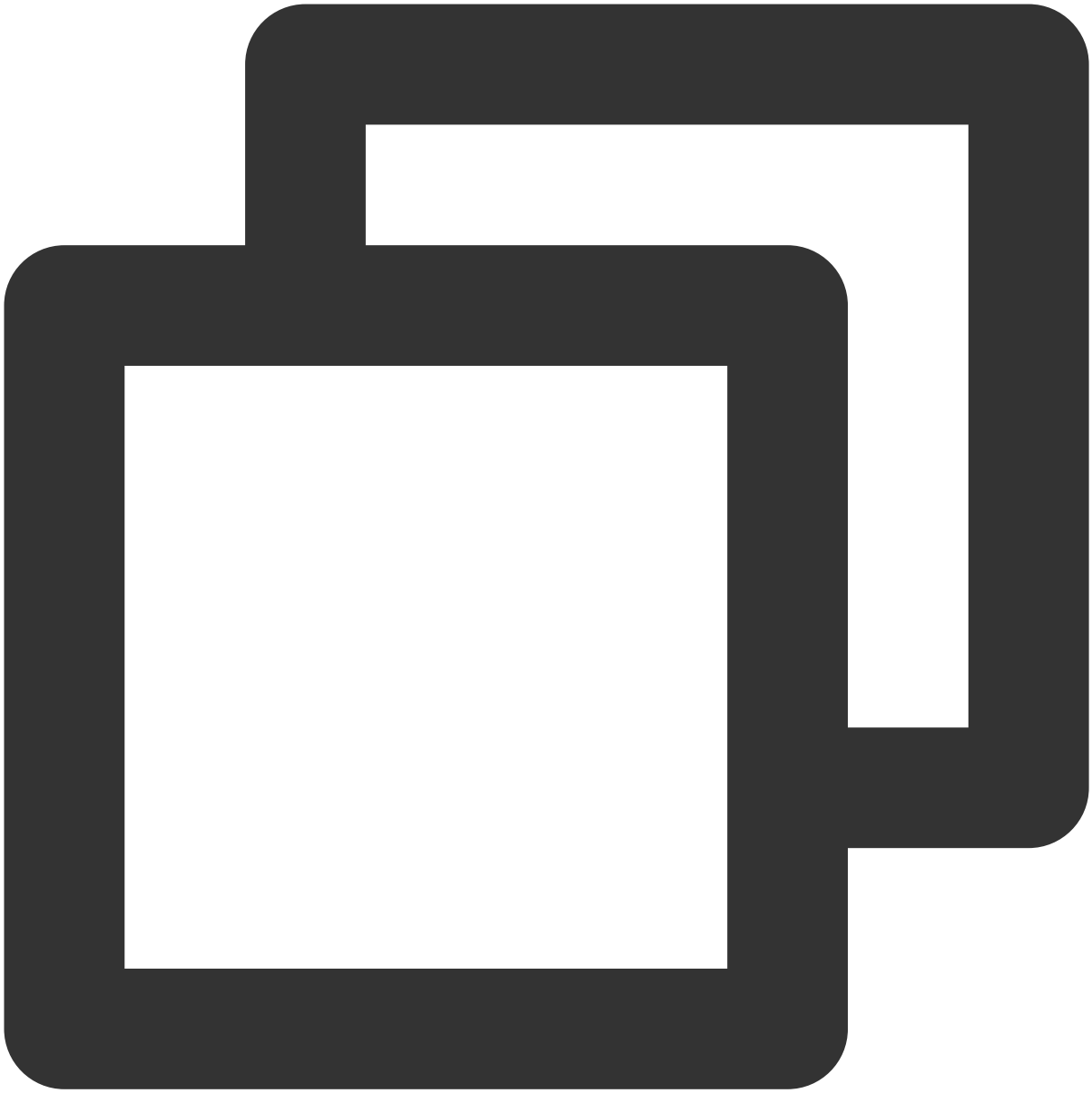
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

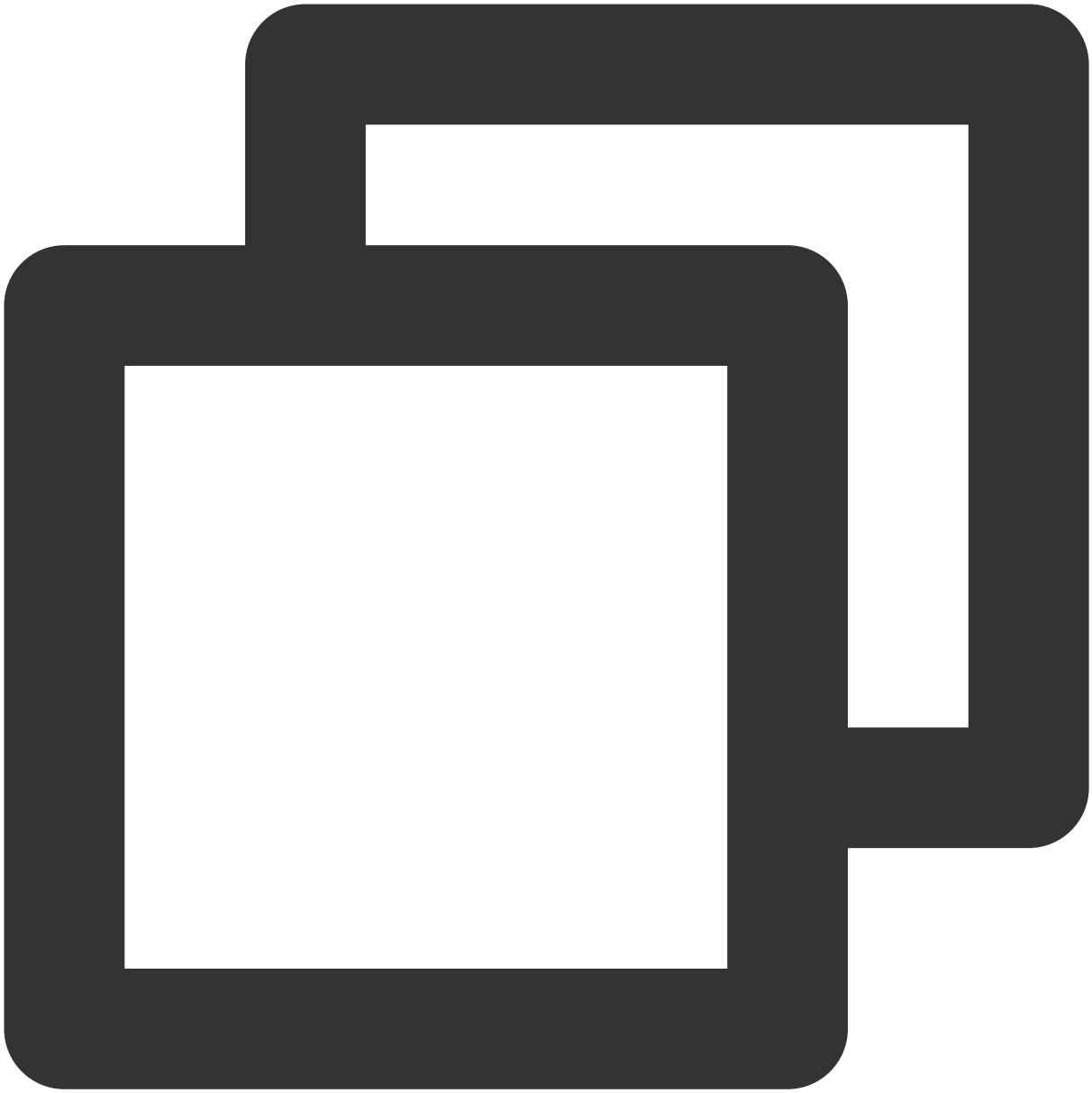
下述请求示例展示了从存储桶 `examplebucket-1250000000` 中删除清单任务 list1。



```
DELETE /?inventory&id=list1 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

响应

上述请求后，COS 返回 204 No Content 的响应表明已成功删除了该存储桶内的清单任务 list1。



```
HTTP/1.1 204 No Content
Server: tencent-cos
Date: Mon, 28 Aug 2018 02:53:40 GMT
x-cos-id-2:0dfafa/DAPDIFdafdsfDdfSFFfdfKKJdafasiuKJK2
x-cos-request-id: NTlhM2I3M2JfMjQ4OGY3MGFfMWE1NF84****
```

版本控制（versioning）
PUT Bucket versioning

最近更新时间：2023-12-21 12:37:28

功能描述

PUT Bucket versioning 接口实现启用或者暂停存储桶的版本控制功能。

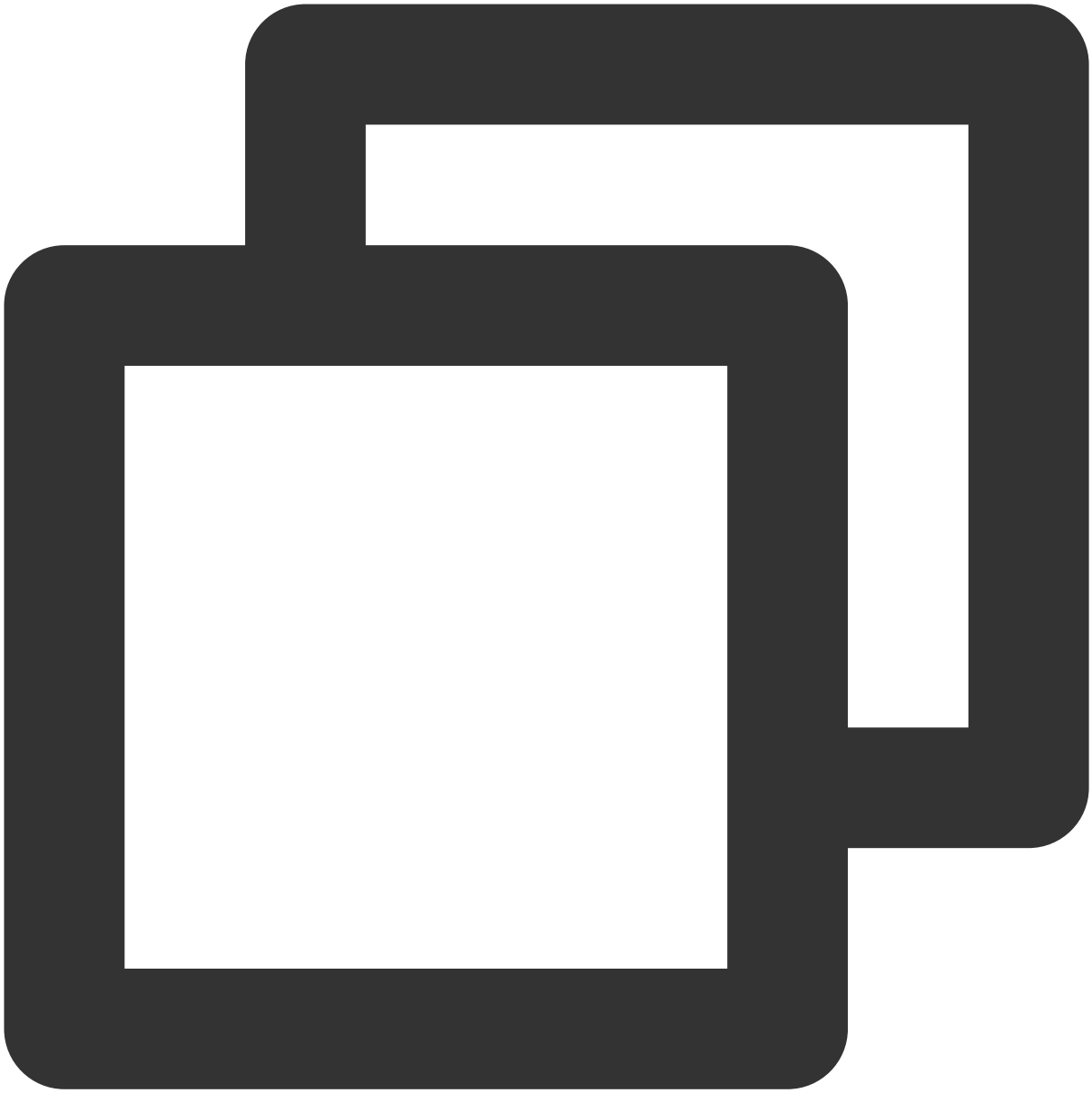
细节分析

1. 如果您从未在存储桶上启用过版本控制，则 GET Bucket versioning 请求不返回版本状态值。
2. 开启版本控制功能后，只能暂停，不能关闭。
3. 设置版本控制状态值为 Enabled 或者 Suspended，表示开启版本控制和暂停版本控制。
4. 设置存储桶的版本控制功能，您需要有存储桶的写权限。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
PUT /?versioning HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

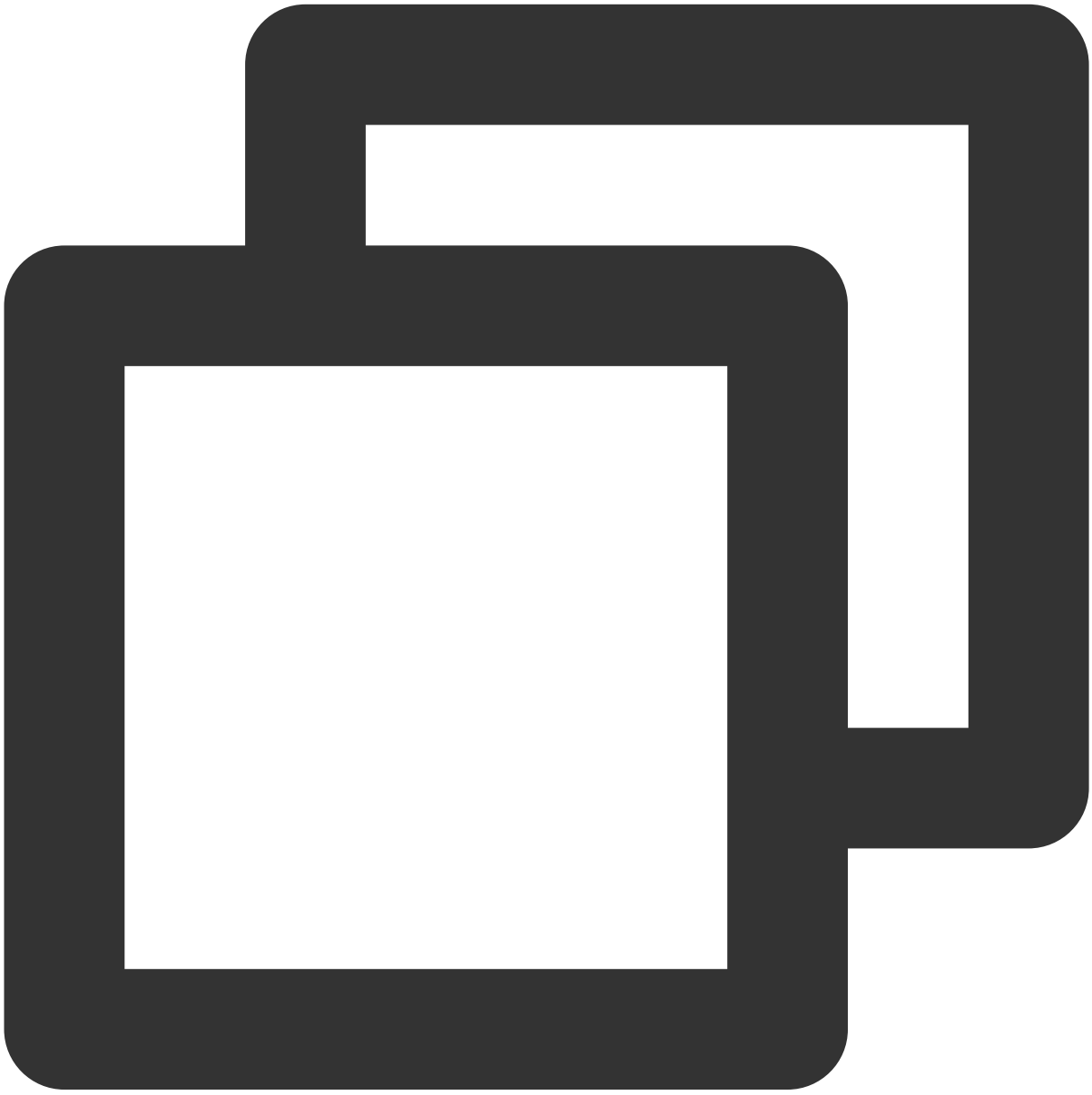
Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体



```
<VersioningConfiguration>
  <Status></Status>
</VersioningConfiguration>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
VersioningConfiguration	无	说明版本控制的具体信息	Container
Status	VersioningConfiguration	说明版本是否开启，枚举值：Suspended、Enabled	Enum

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

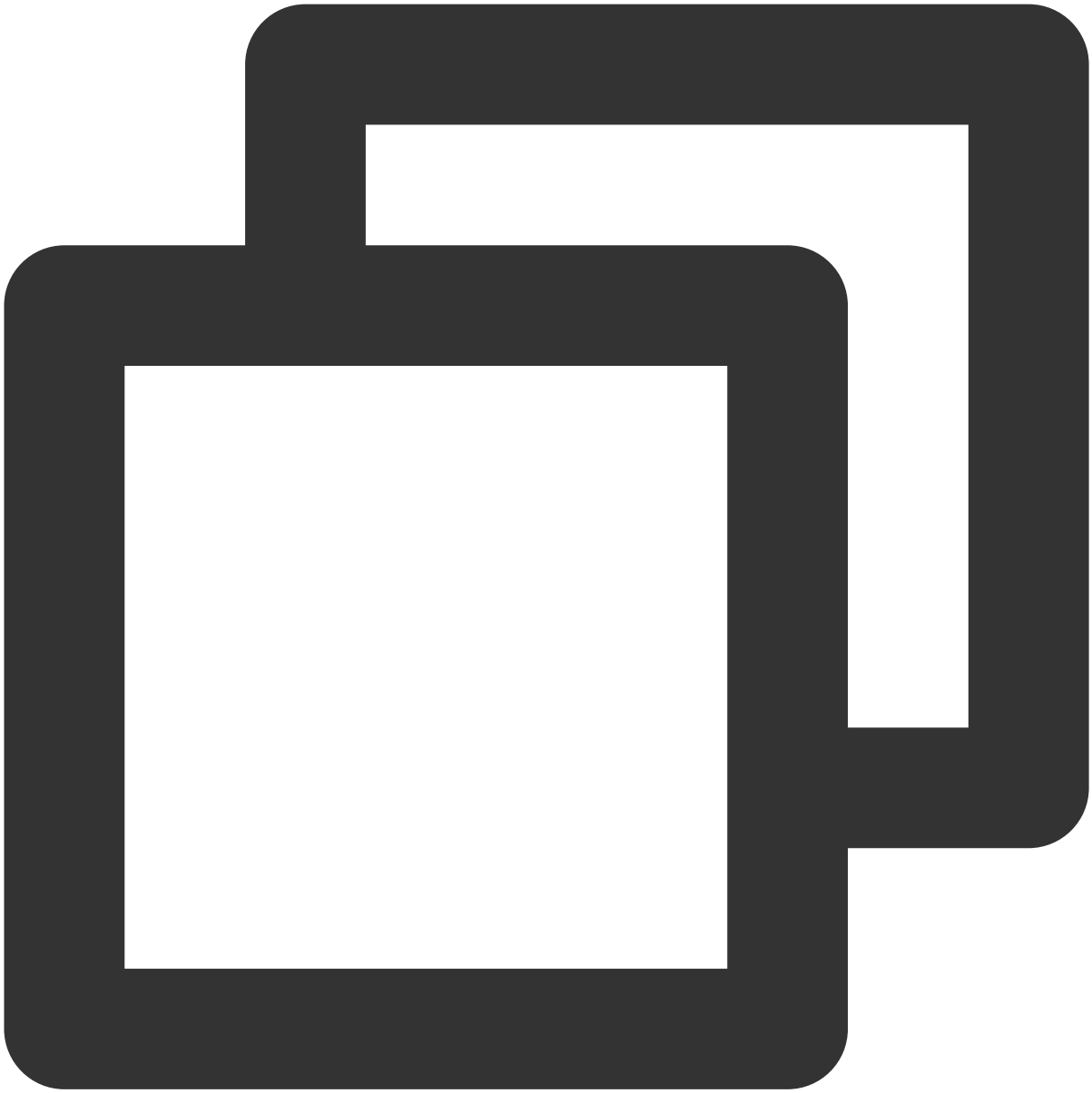
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

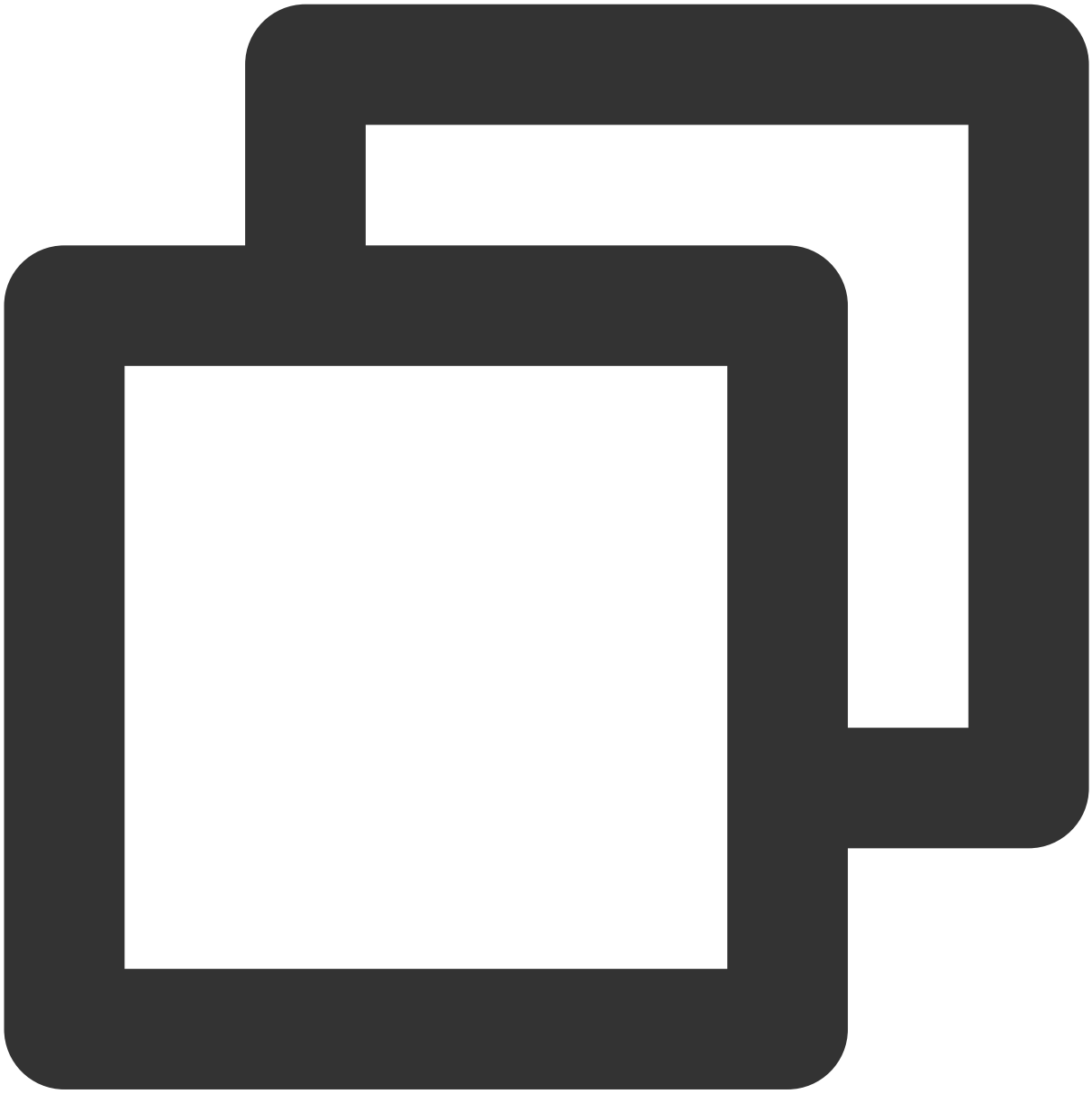
请求



```
PUT /?versioning HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Content-Type: application/xml
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9Sm****&q-si
Content-Length: 83

<VersioningConfiguration>
```

```
    <Status>Enabled</Status>
</VersioningConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Wed, 23 Aug 2017 08:14:53 GMT
Server: tencent-cos
```

```
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****
```

版权所有：腾讯云计算（北京）有限责任公司

GET Bucket versioning

最近更新时间：2023-12-21 12:37:52

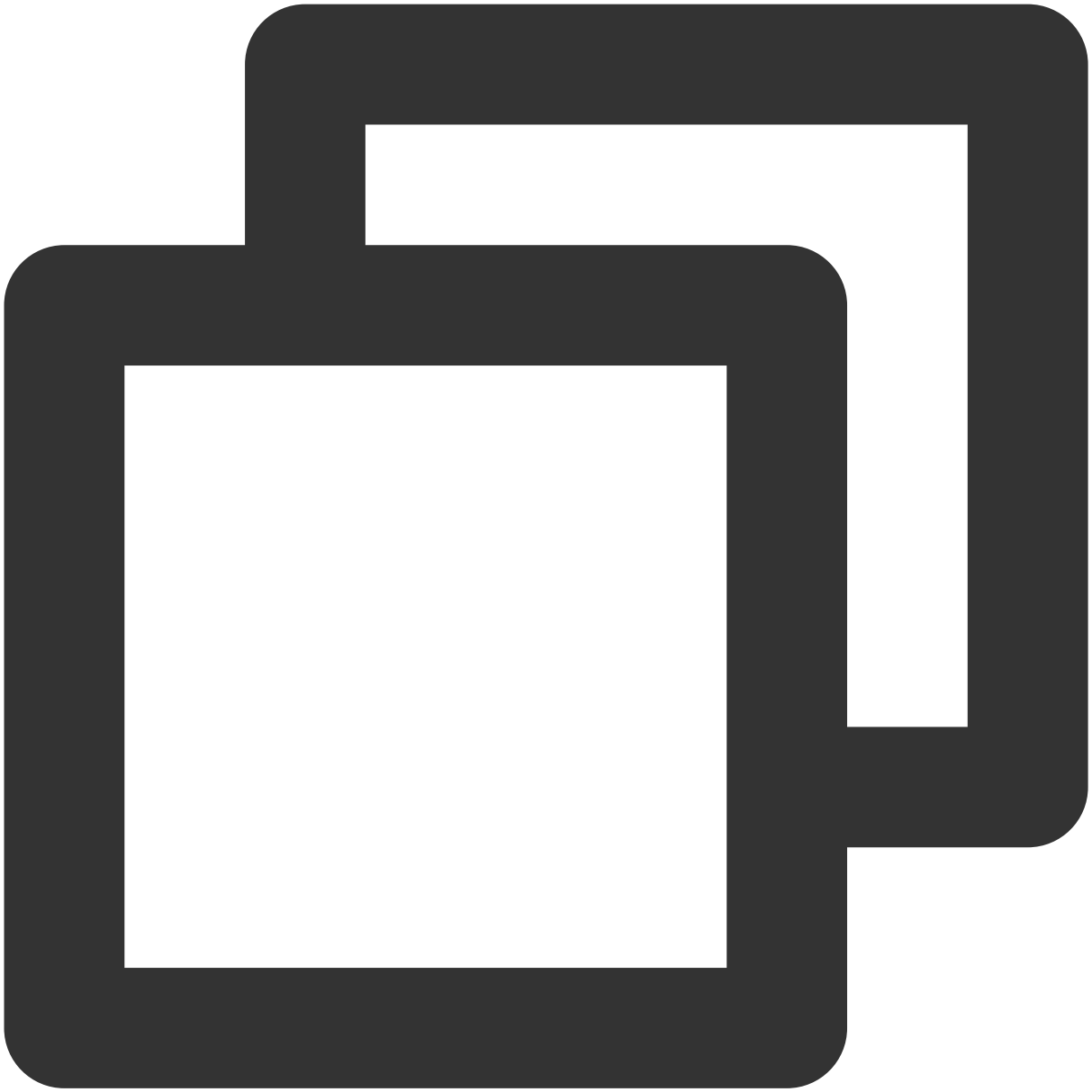
功能描述

GET Bucket versioning 接口用于实现获得存储桶的版本控制信息。

细节分析

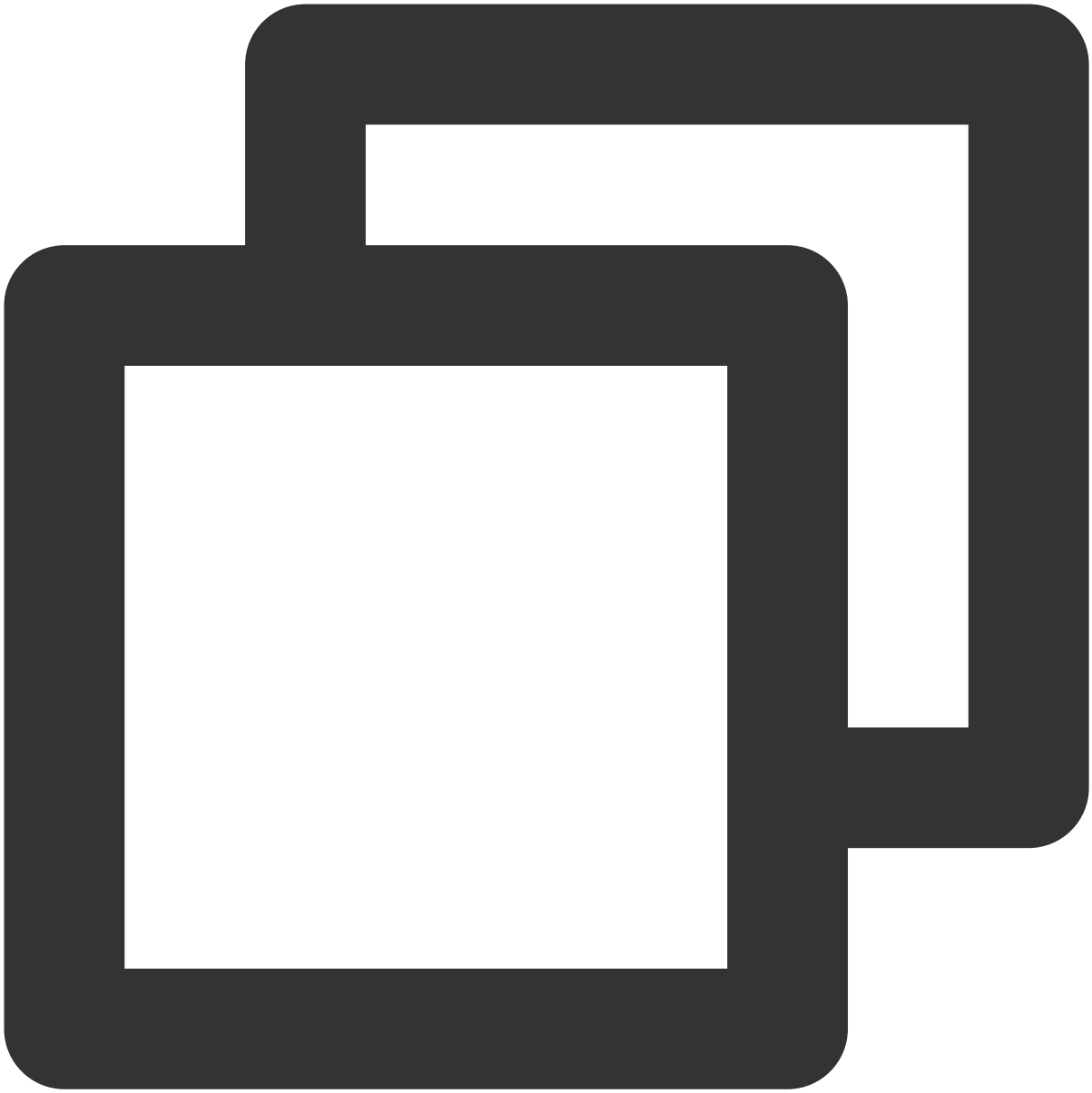
1. 获取存储桶版本控制的状态，需要有该存储桶的读权限。
2. 有三种版本控制状态：未启用版本控制、启用版本控制和暂停版本控制。

如果您从未在存储桶上启用（或暂停）版本控制，则响应为：



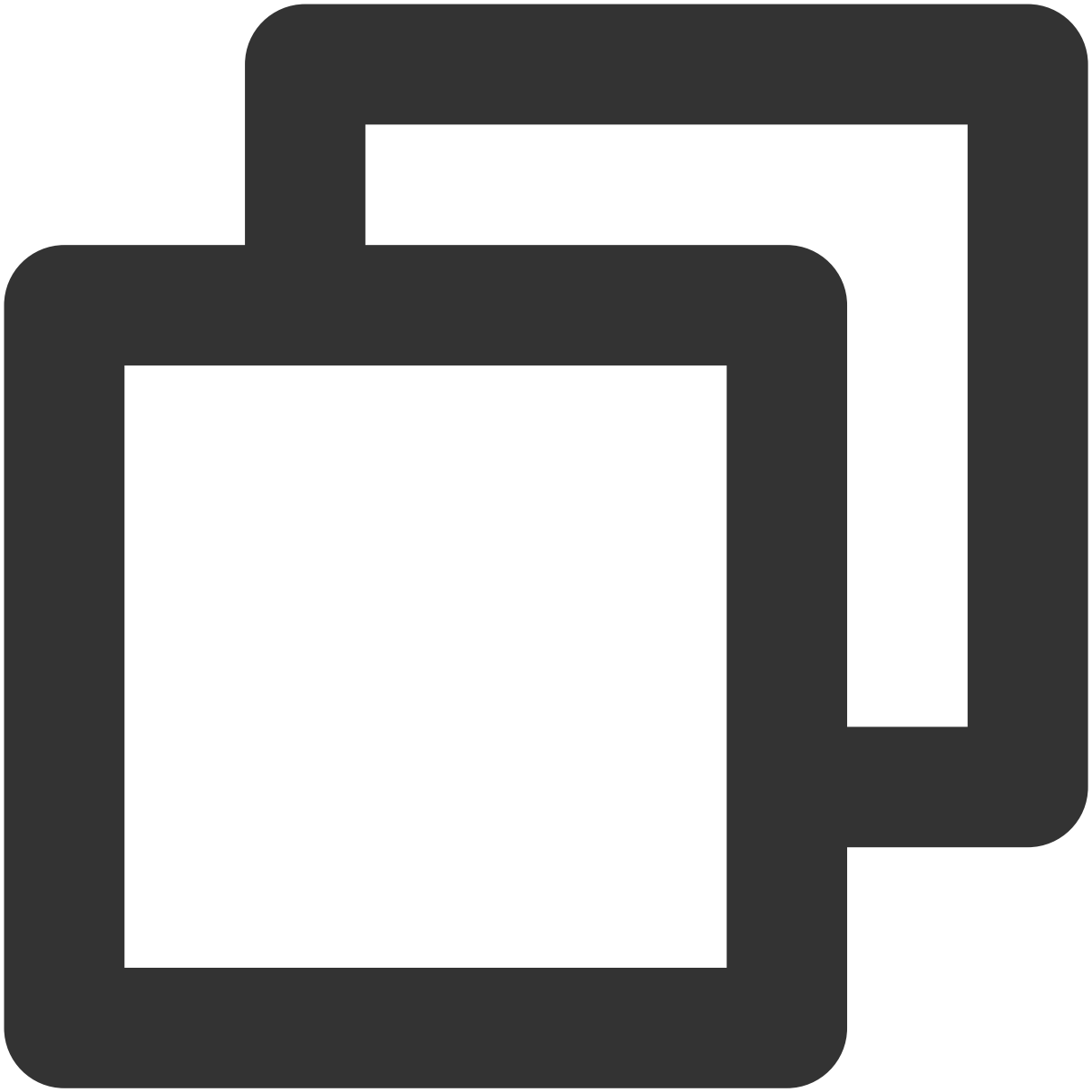
```
<VersioningConfiguration/>
```

如果您启用了存储桶的版本控制功能，则响应为：



```
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

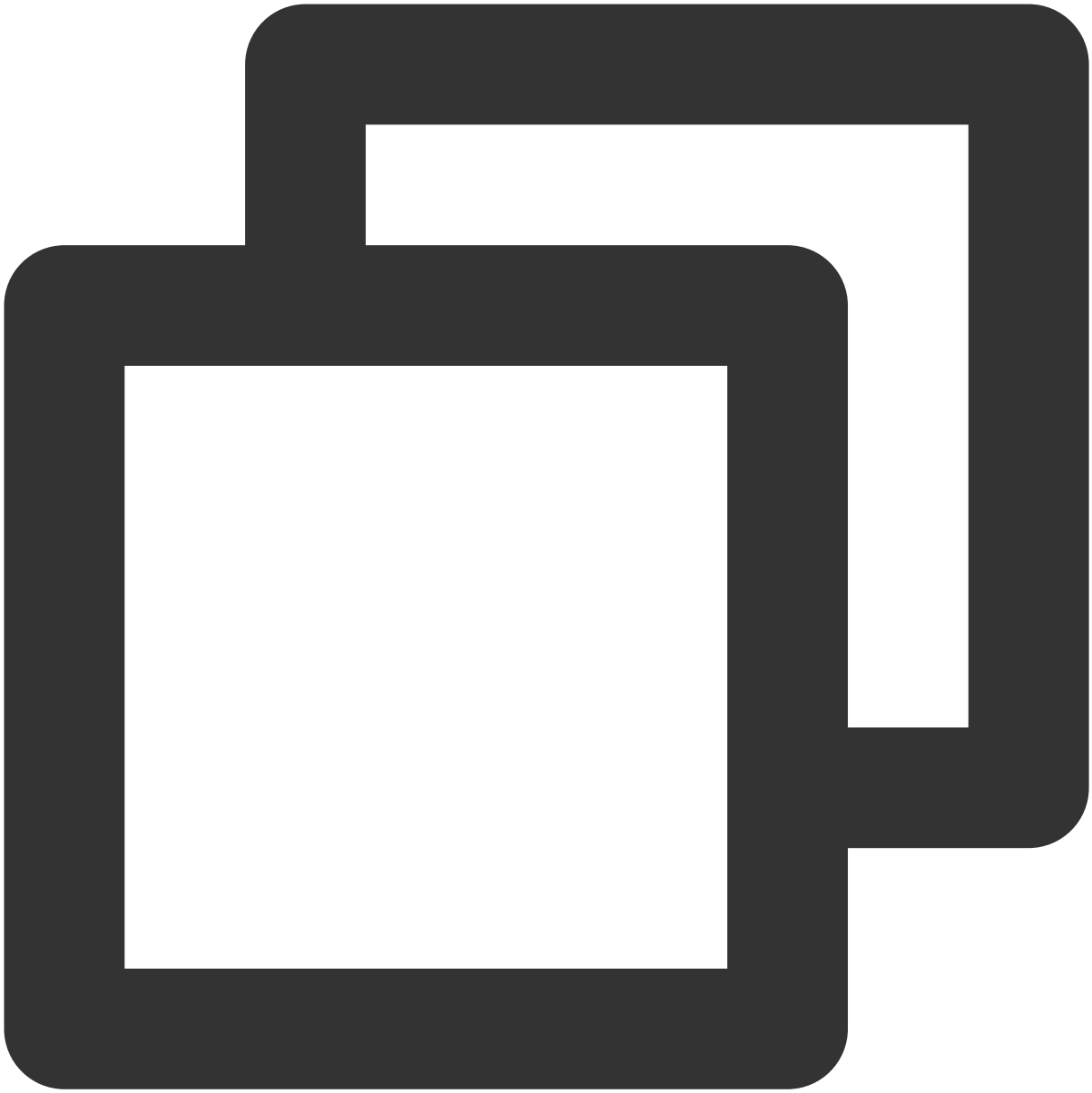
如果您暂停了存储桶的版本控制功能，则响应为：



```
<VersioningConfiguration>
 <Status>Suspended</Status>
</VersioningConfiguration>
```

请求

请求示例



```
GET /?versioning HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

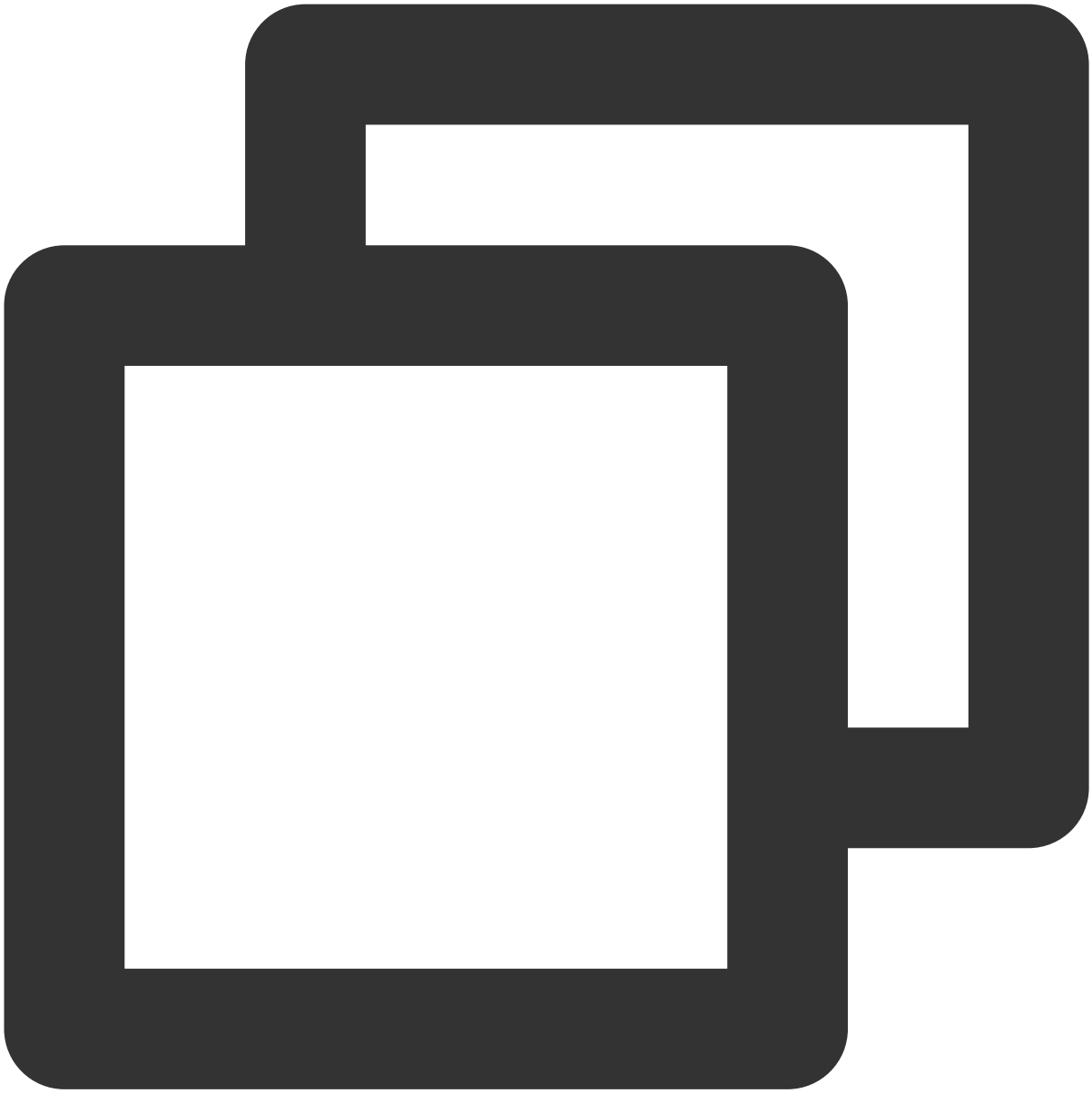
该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<VersioningConfiguration>
    <Status></Status>
</VersioningConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
VersioningConfiguration	无	说明版本控制的具体信息	Container
Status	VersioningConfiguration	说明版本是否开启，枚举值：	Enum

版权所有：腾讯云计算（北京）有限责任公司

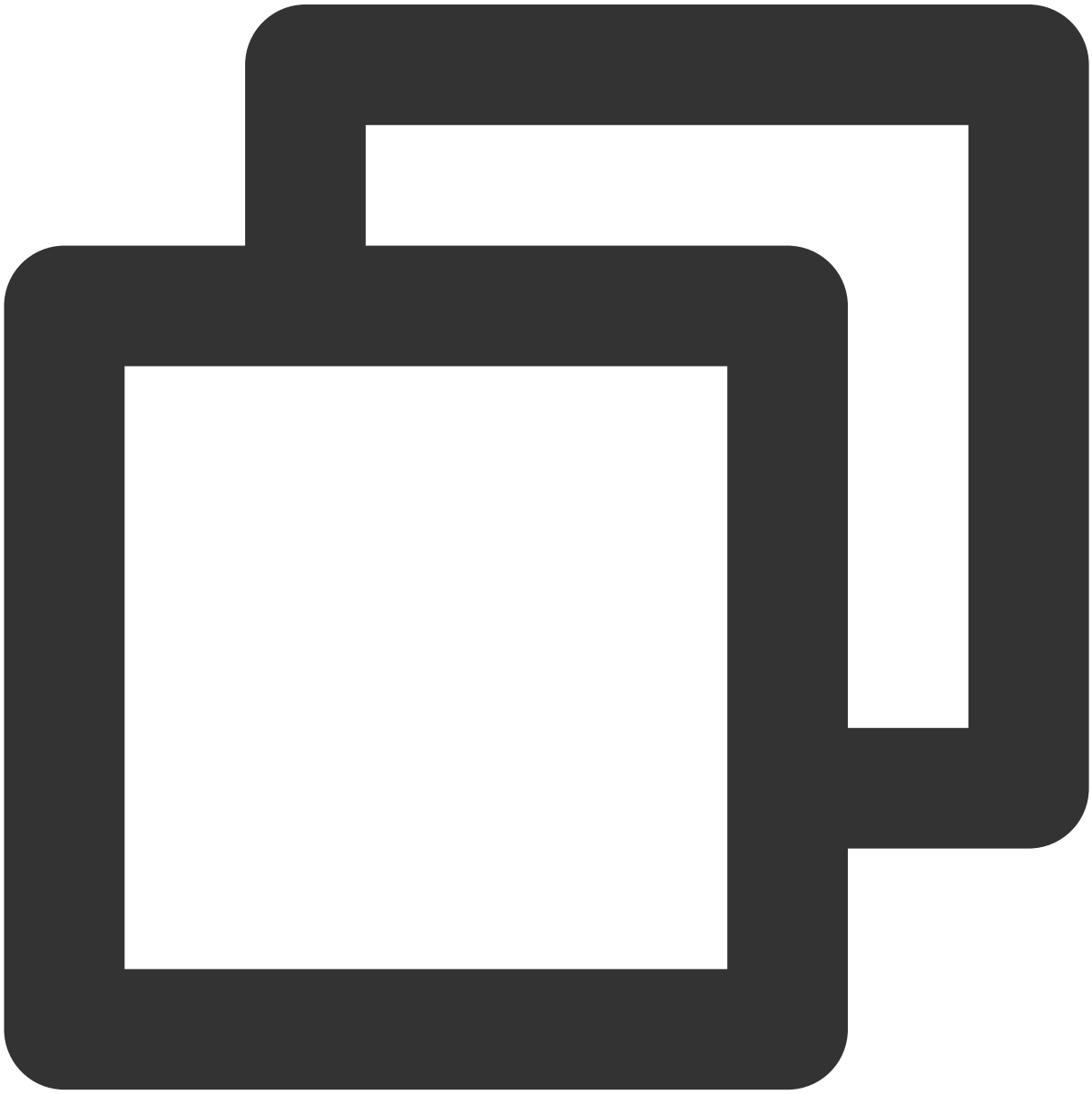
		Suspended、Enabled	

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

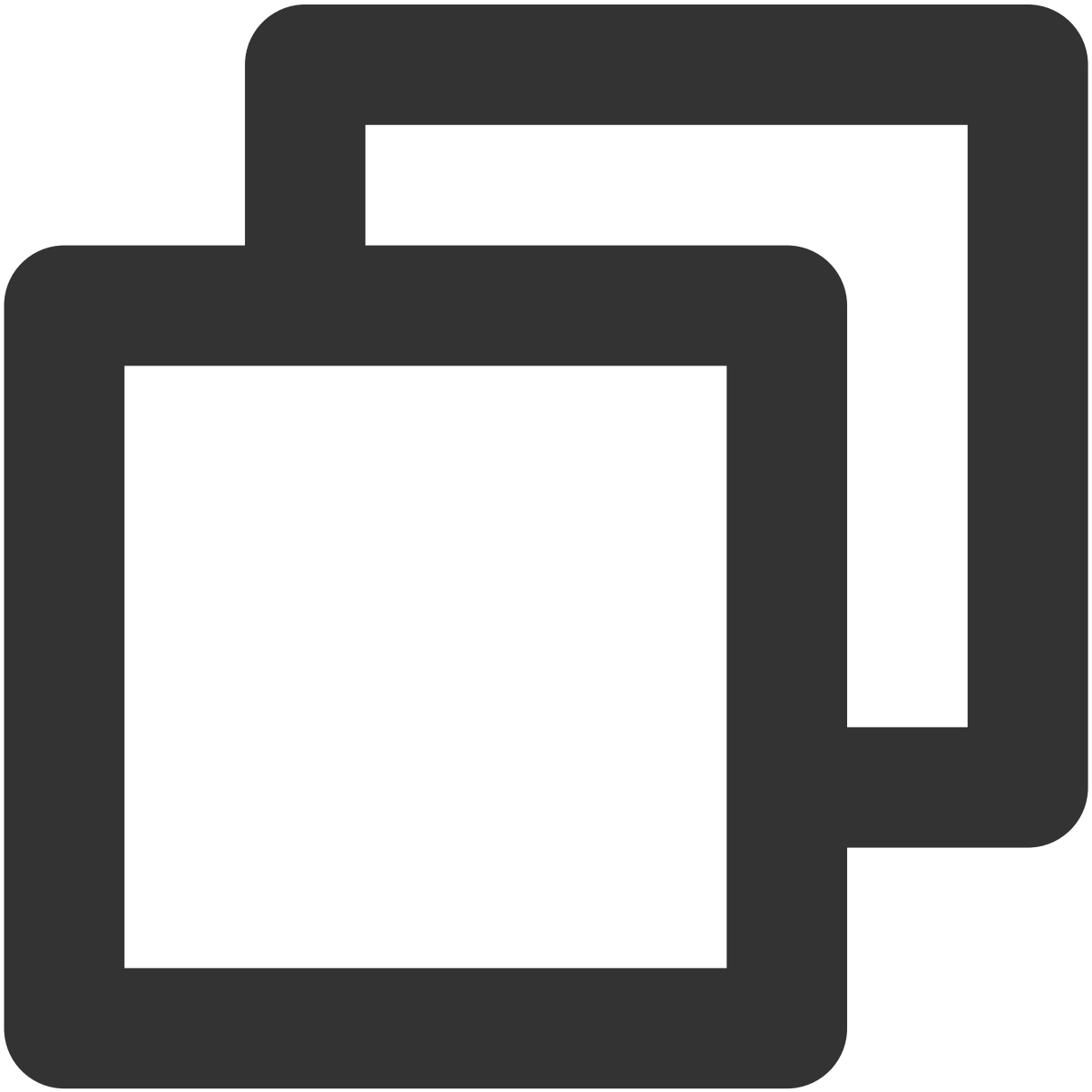
实际案例

请求



```
GET /?versioning HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9Sm****&q-si
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 120
Connection: keep-alive
Date: Wed, 23 Aug 2017 08:15:16 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5OTRfZDNhZDM1MGFfMjYyMTFfZmU3****

<?xml version='1.0' encoding='utf-8' ?>
<VersioningConfiguration>
    <Status>Enabled</Status>
</VersioningConfiguration>
```

存储桶复制（replication）
PUT Bucket replication

最近更新时间：2023-12-21 12:37:34

功能描述

PUT Bucket replication 用于向已启用版本控制的存储桶中配置存储桶复制规则。如果存储桶已经配置了存储桶复制规则，那么该请求会替换现有配置。

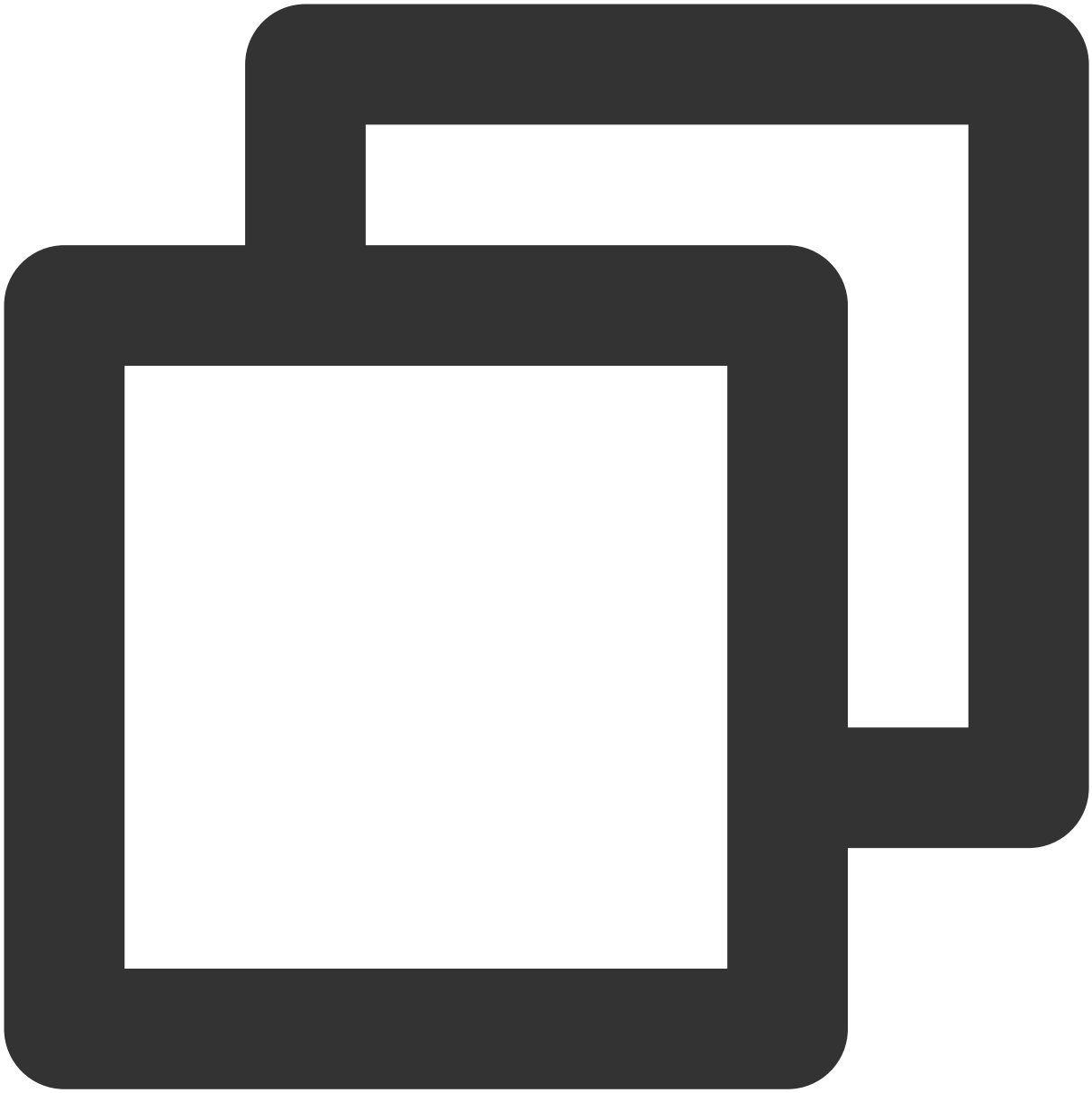
注意

使用该接口时，需确保存储桶已经开启版本控制，开启版本控制的 API 文档请参见 PUT Bucket versioning 接口文档。

开启了多 AZ 配置的存储桶，不支持将多 AZ 存储类型复制为单 AZ 存储类型，例如标准存储（多 AZ）类型不支持复制为标准存储类型。

请求

请求示例



```
PUT /?replication HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-MD5: MD5
Authorization: Auth String
request body
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

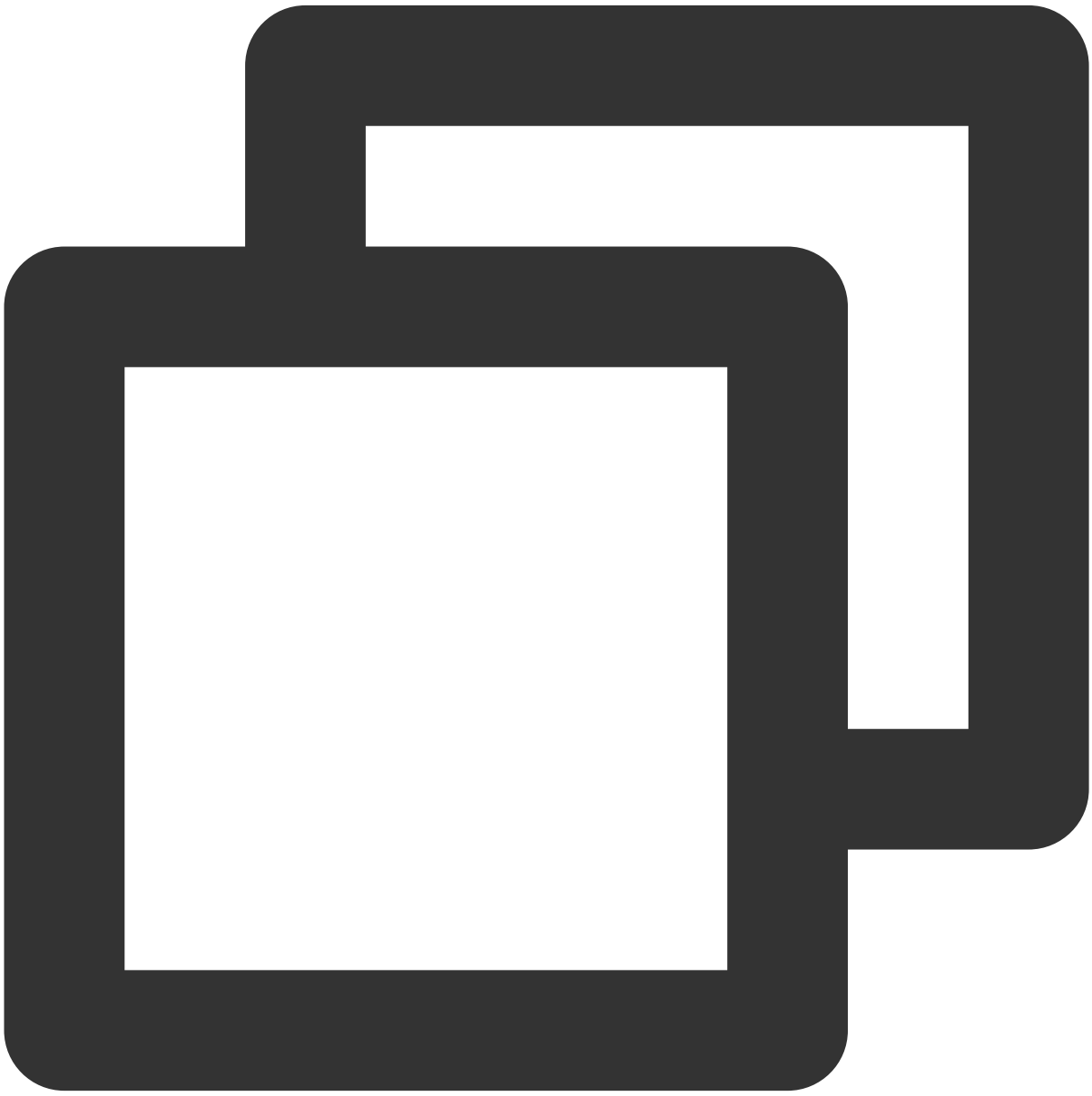
<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

用户在请求体中设置存储桶复制的具体配置信息。配置信息包括存储桶复制规则的启用状态、优先级、复制内容、筛选范围、目标存储桶的存储桶名称和存储地域等信息。



```
<ReplicationConfiguration>
    <Role>qcs::cam::uin/<OwnerUin>:uin/<SubUin></Role>
    <Rule>
        <Status></Status>
        <ID></ID>
        <Prefix></Prefix>
        <Destination>
            <Bucket>qcs::cos:<Region>::<BucketName-APPID></Bucket>
            <StorageClass></StorageClass>
        </Destination>
        <DeleteMarkerReplication>
            <Status></Status>
        </DeleteMarkerReplication>
    </Rule>
</ReplicationConfiguration>
```

具体内容描述如下：

节点名称（关键字）	父节点	描述
ReplicationConfiguration	无	说明所有复制配置信息
Role	ReplicationConfiguration	发起者身份标示：`qcs::cam::uin/<OwnerUin>`
Rule	ReplicationConfiguration	具体配置信息，最多支持1000个
ID	ReplicationConfiguration.Rule	用来标注具体 Rule 的名称
Status	ReplicationConfiguration.Rule	标识 Rule 是否生效，枚举值：Enabled、
Prefix	ReplicationConfiguration.Rule	需要复制的对象前缀
Destination	ReplicationConfiguration.Rule	目标存储桶信息
Bucket	ReplicationConfiguration.Rule.Destination	资源标识符：`qcs::cos:<Region>:APPID>`
DeleteMarkerReplication	ReplicationConfiguration.Rule	是否同步删除标记
Status	ReplicationConfiguration.Rule.DeleteMarkerReplication	是否同步删除标记，支持 `Disabled` 默认值为 `Enabled`，即同步删除标记

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体为空。

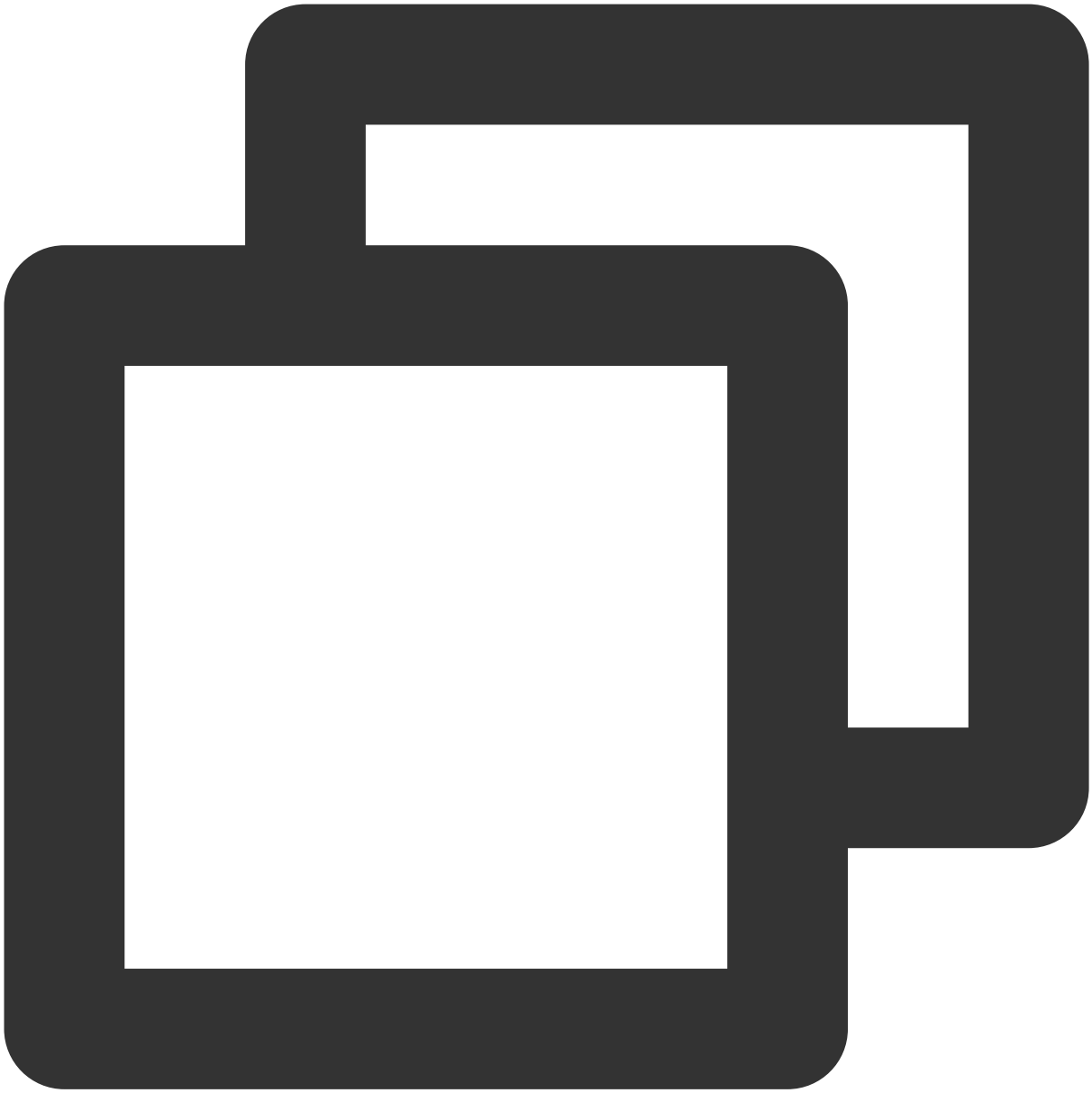
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

以下请求是对存储桶 `originbucket-1250000000` 添加一条存储桶复制配置。该存储桶复制配置中，指定复制对象前缀为 `testPrefix` 的对象，目标存储桶为广州的 `destinationbucket-1250000000` 。

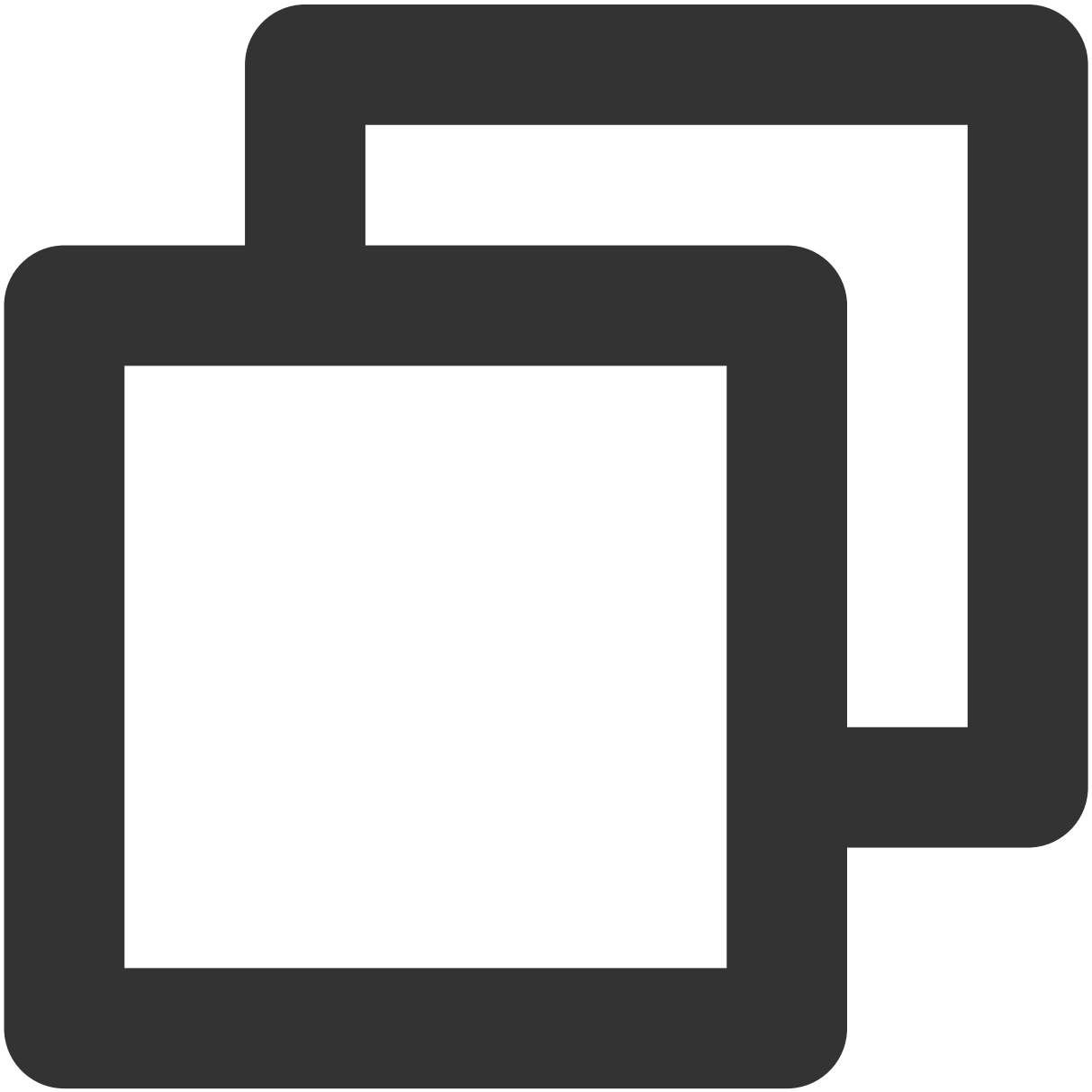


```
PUT /?replication HTTP/1.1
Date: Mon, 28 Aug 2017 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Content-MD5: AAq9nzrpsz5LJ4UEe1f6Q==
Host: originbucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 312

<ReplicationConfiguration>
    <Role>qcs::cam::uin/100000000001:uin/100000000001</Role>
    <Rule>
        <Status>Enabled</Status>
```

```
        <ID>RuleId_01</ID>
        <Prefix>testPrefix</Prefix>
        <Destination>
            <Bucket>qcs::cos:ap-guangzhou::destinationbucket-1250000000</Bucket>
        </Destination>
    </Rule>
</ReplicationConfiguration>
```

响应

上述请求后，COS 返回以下响应，表明当前该跨地域配置已经成功设置完毕。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 14 Apr 2019 07:06:19 GMT
Server: tencent-cos
x-cos-bucket-region: ap-guangzhou
x-cos-request-id: NWQwMzQ3NmJfMjRiMjU4NjRfOTM4NV82ZDU1****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODc0OWRkZjk0ZDM1NmI1M2E2MTR
```

GET Bucket replication

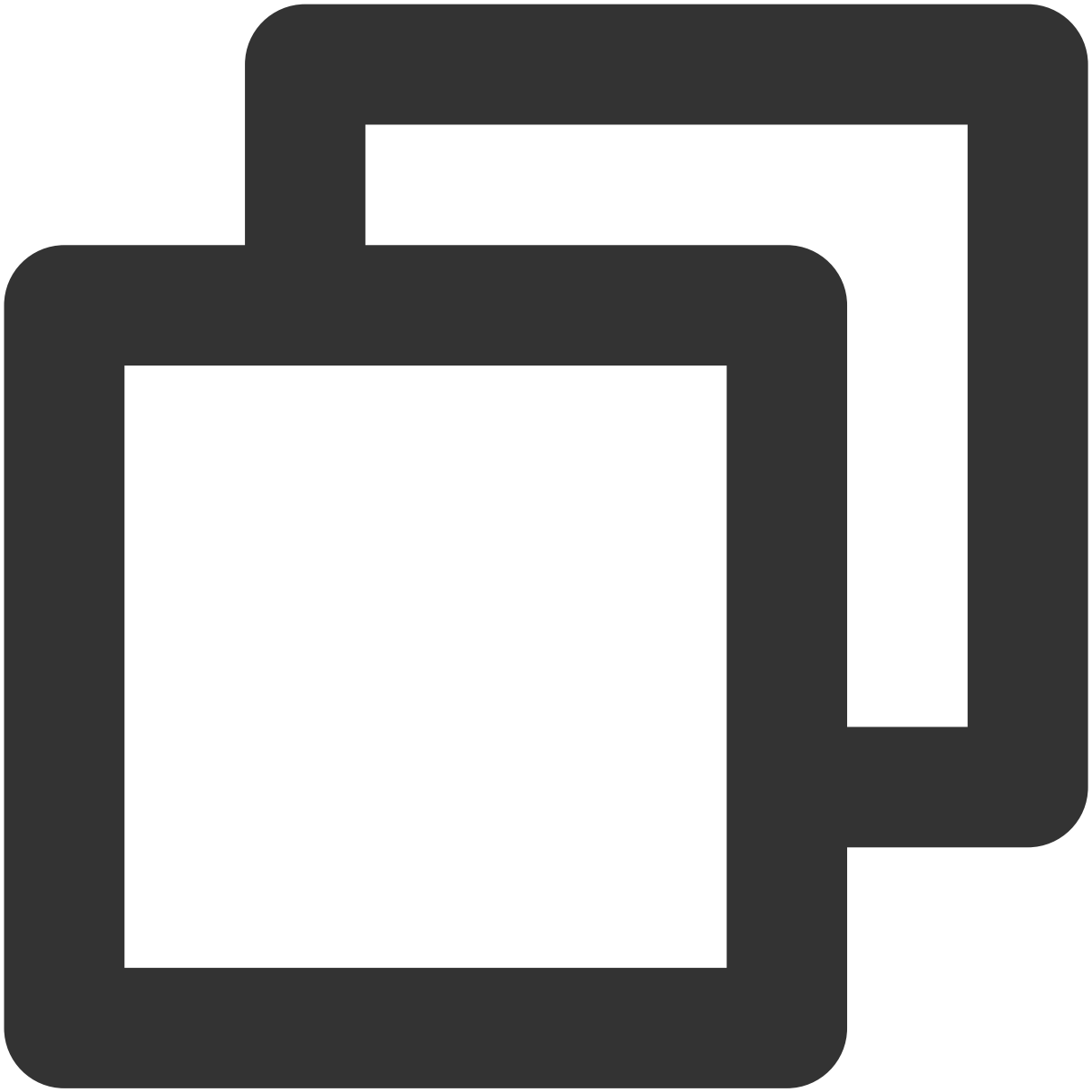
最近更新时间：2023-12-21 12:38:24

功能描述

GET Bucket replication 接口用于查询存储桶中用户存储桶复制配置信息。用户发起该请求时需获得请求签名，表明该请求已获得许可。

请求

请求示例



```
GET /?replication HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

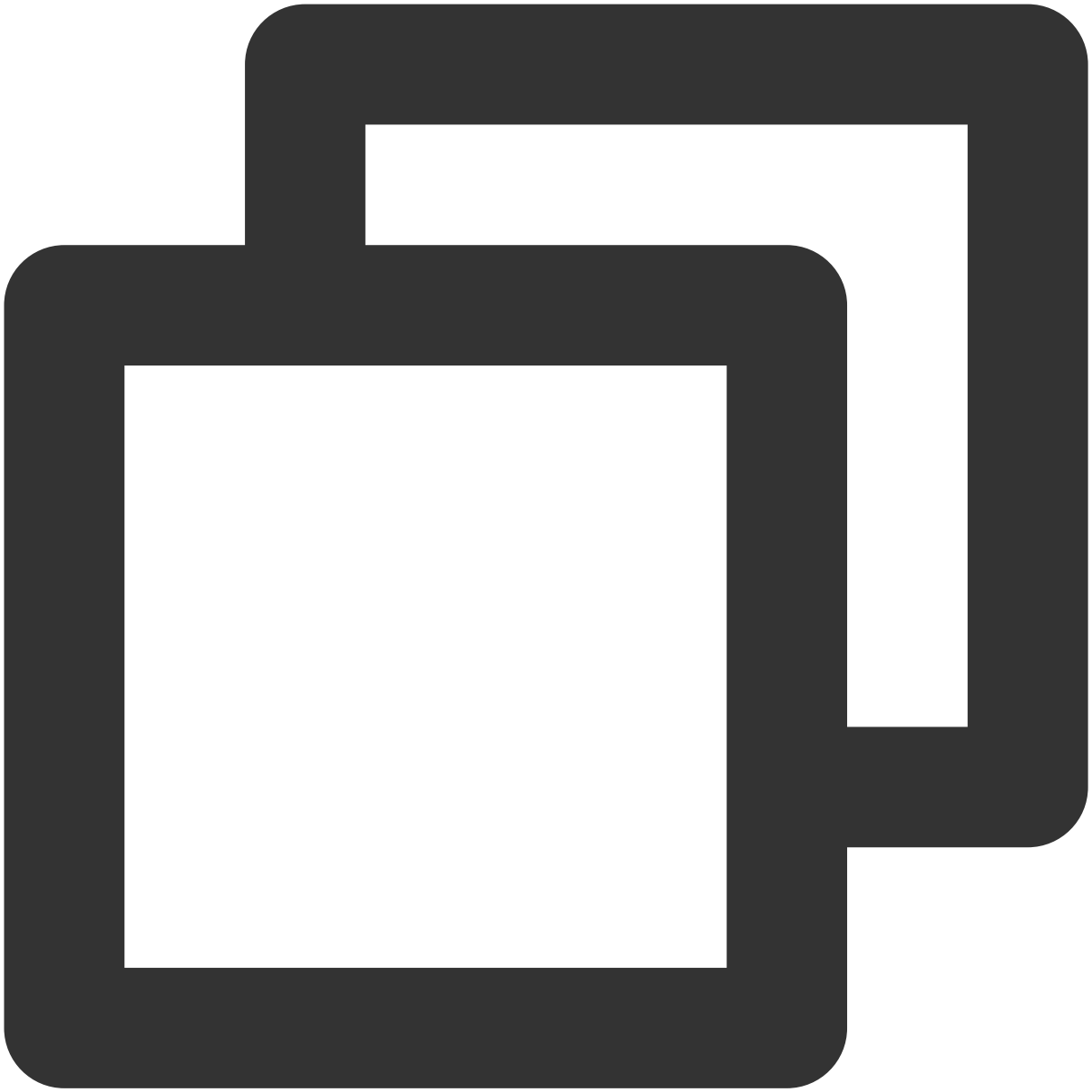
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<ReplicationConfiguration>
    <Role>qcs::cam::uin/<OwnerUin>:uin/<SubUin></Role>
    <Rule>
        <Status></Status>
        <ID></ID>
        <Prefix></Prefix>
        <Destination>
            <Bucket>qcs::cos:<Region>::<BucketName-APPID></Bucket>
            <StorageClass></StorageClass>
        </Destination>
        <DeleteMarkerReplication>
```

```
            <Status></Status>
        </DeleteMarkerReplication>
    </Rule>
</ReplicationConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述
ReplicationConfiguration	无	说明所有复制配置信息
Role	ReplicationConfiguration	发起者身份标示：`qcs::cam::uin/<OwnerU`
Rule	ReplicationConfiguration	具体配置信息，最多支持1000个，所有储桶
ID	ReplicationConfiguration.Rule	用来标注具体 Rule 的名称
Status	ReplicationConfiguration.Rule	标识 Rule 是否生效，枚举值：Enabled,
Prefix	ReplicationConfiguration.Rule	前缀匹配策略，不可重叠，重叠返回错
Destination	ReplicationConfiguration.Rule	目标存储桶信息
Bucket	ReplicationConfiguration.Rule.Destination	资源标识符：`qcs::cos:[region]:APPID]`
StorageClass	ReplicationConfiguration.Rule.Destination	存储类型，枚举值：STANDARD，INTI STANDARD_IA，ARCHIVE，DEEP_A 类型
DeleteMarkerReplication	ReplicationConfiguration.Rule	是否同步删除标记
Status	ReplicationConfiguration.Rule. DeleteMarkerReplication	是否同步删除标记，支持 `Disabled` 为 `Enabled`，即同步删除标记

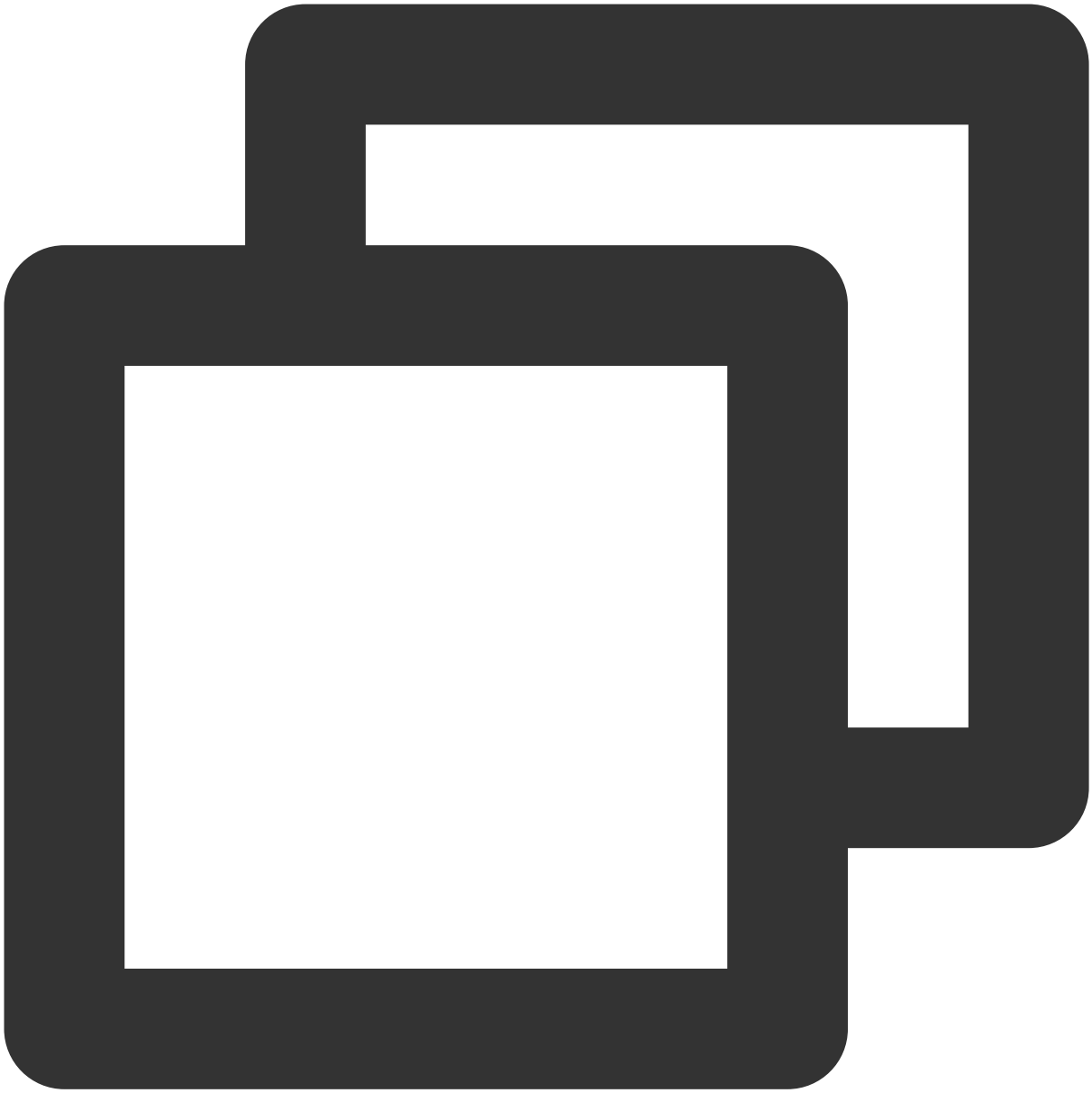
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

下述请求示例展示了从存储桶 `originbucket-1250000000` 中查询配置信息。

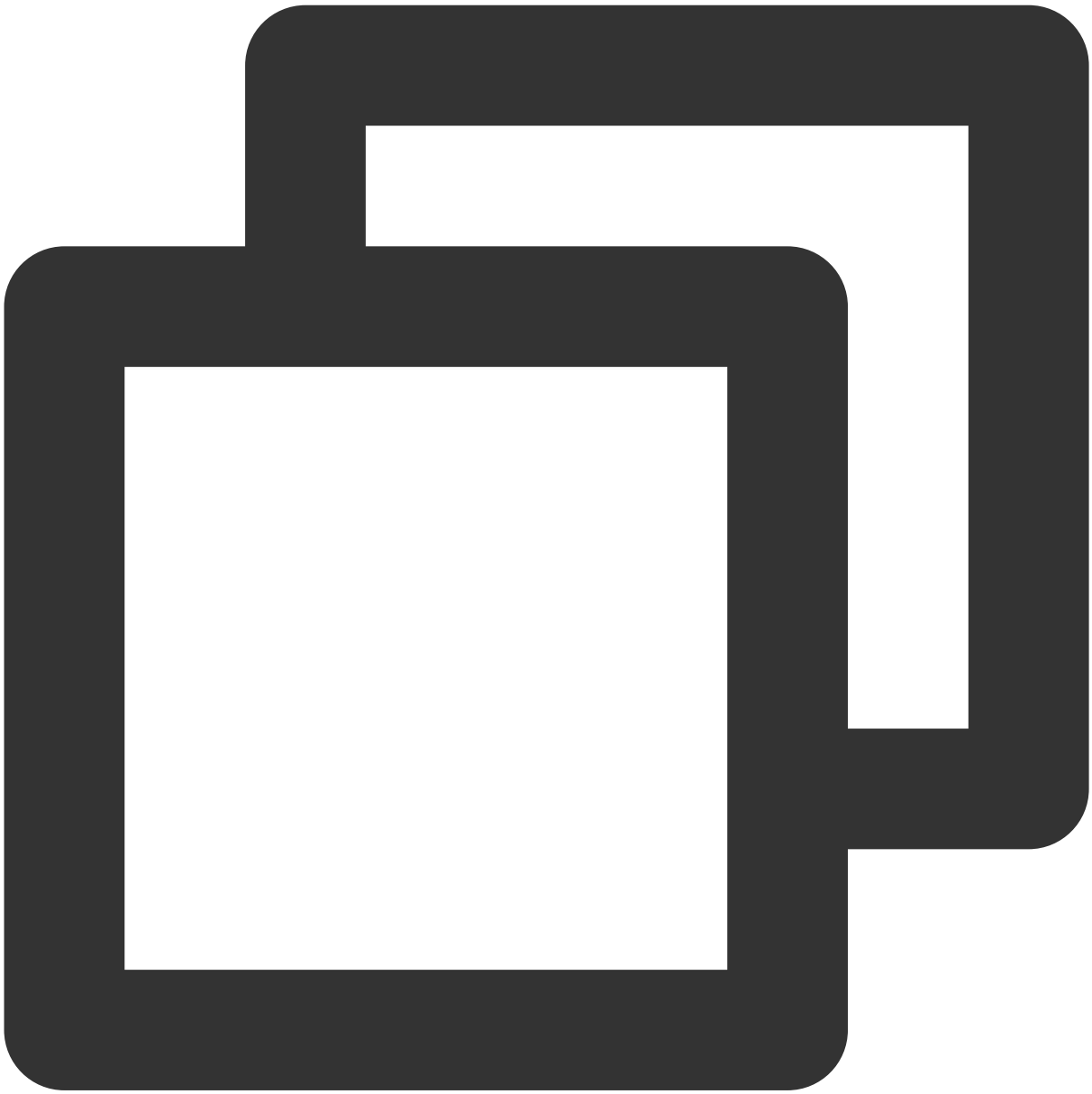


```
GET /?replication HTTP/1.1
Date: Fri, 14 Apr 2019 07:17:19 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98JM&q-si
Host: originbucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 0
```

响应

上述请求后，COS 返回以下响应，表明当前该存储桶内的存储桶复制配置处于启用状态。该规则配置信息中，复制的内容为存储桶 `originbucket-1250000000` 内包含 `testPrefix` 前缀的所有对象。对象副本的存储类型

默认跟随源存储桶内对象的存储类型。



```
Content-Type: application/xml
Content-Length: 309
Connection: keep-alive
Date: Fri, 14 Apr 2019 07:17:19 GMT
Server: tencent-cos
x-cos-replication-rule-creation-time: Fri, 14 Apr 2019 07:06:19 GMT
x-cos-request-id: NWQwMzQ5ZmZfMjBiNDU4NjRfNjAwOV84MzA2****
<ReplicationConfiguration>
    <Role>qcs::cam::uin/100000000001:uin/100000000001</Role>
```

```
    <Rule>
        <Status>Enabled</Status>
        <ID>RuleId_01</ID>
        <Prefix>testPrefix</Prefix>
        <Destination>
            <Bucket>qcs::cos:ap-guangzhou::destinationbucket-1250000000</Bucket>
        </Destination>
    </Rule>
</ReplicationConfiguration>
```

DELETE Bucket replication

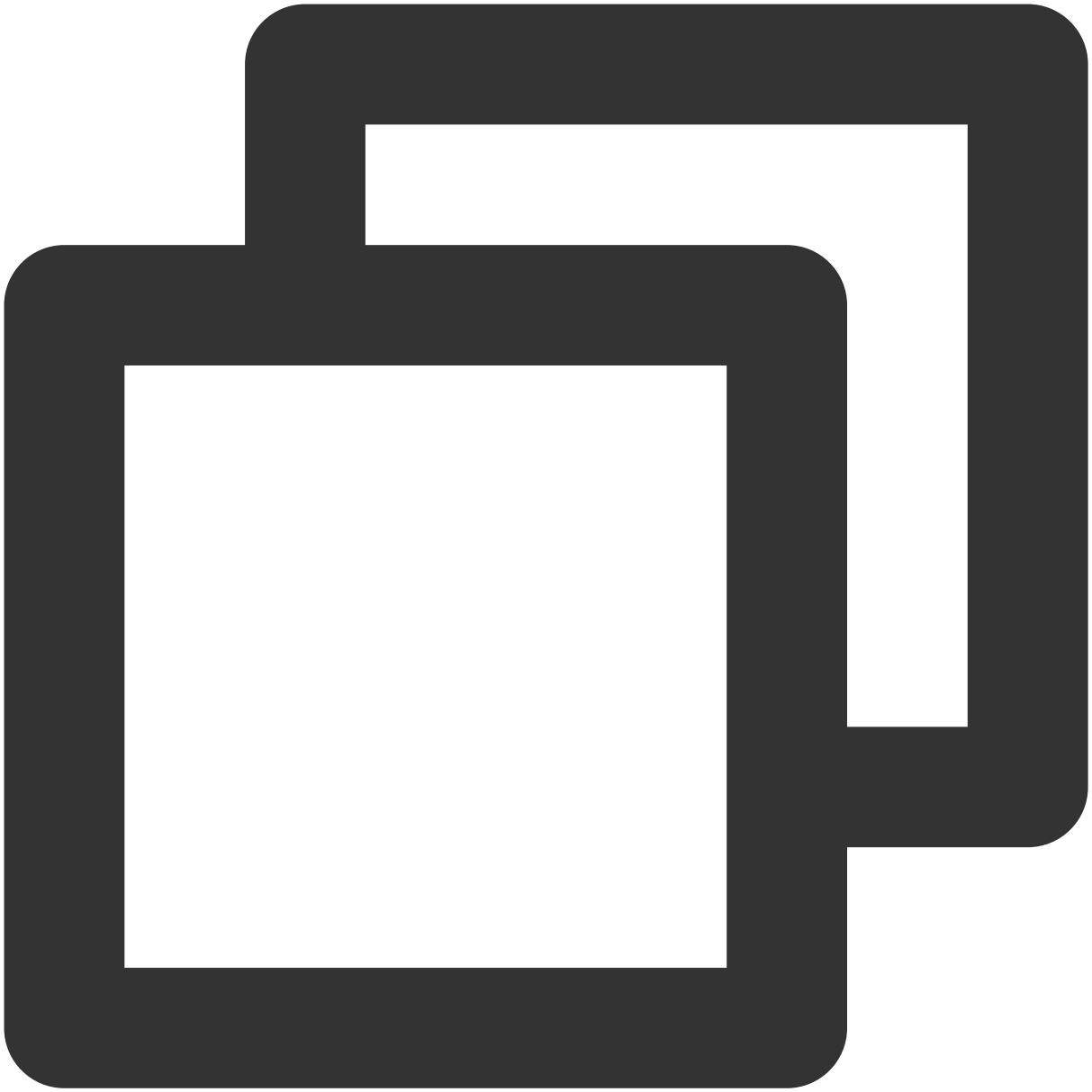
最近更新时间：2023-12-21 12:38:36

功能描述

DELETE Bucket replication 用来删除存储桶中的存储桶复制配置。用户发起该请求时需获得请求签名，表明该请求已获得许可。

请求

请求示例



```
DELETE /?replication HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体为空。

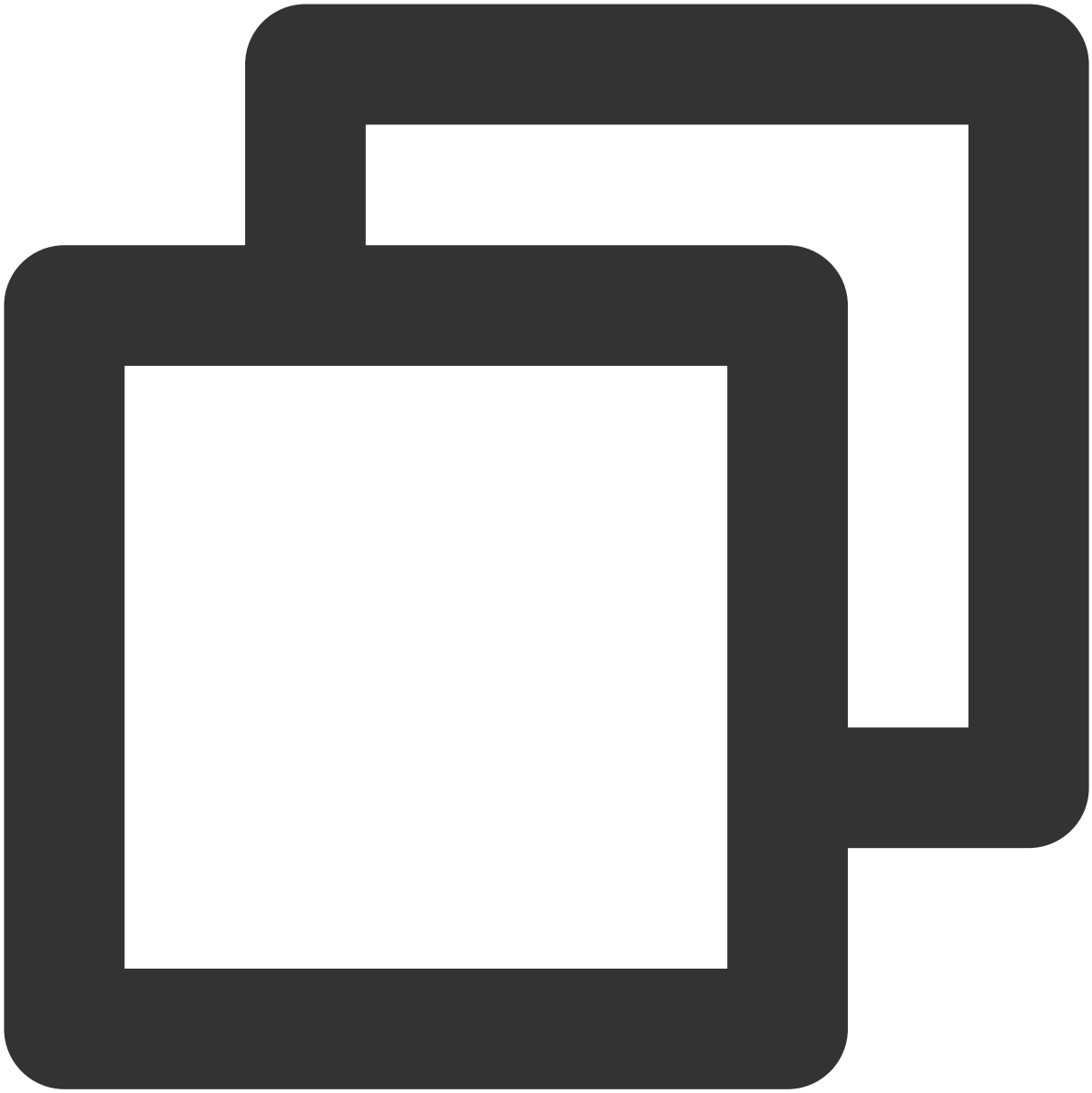
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

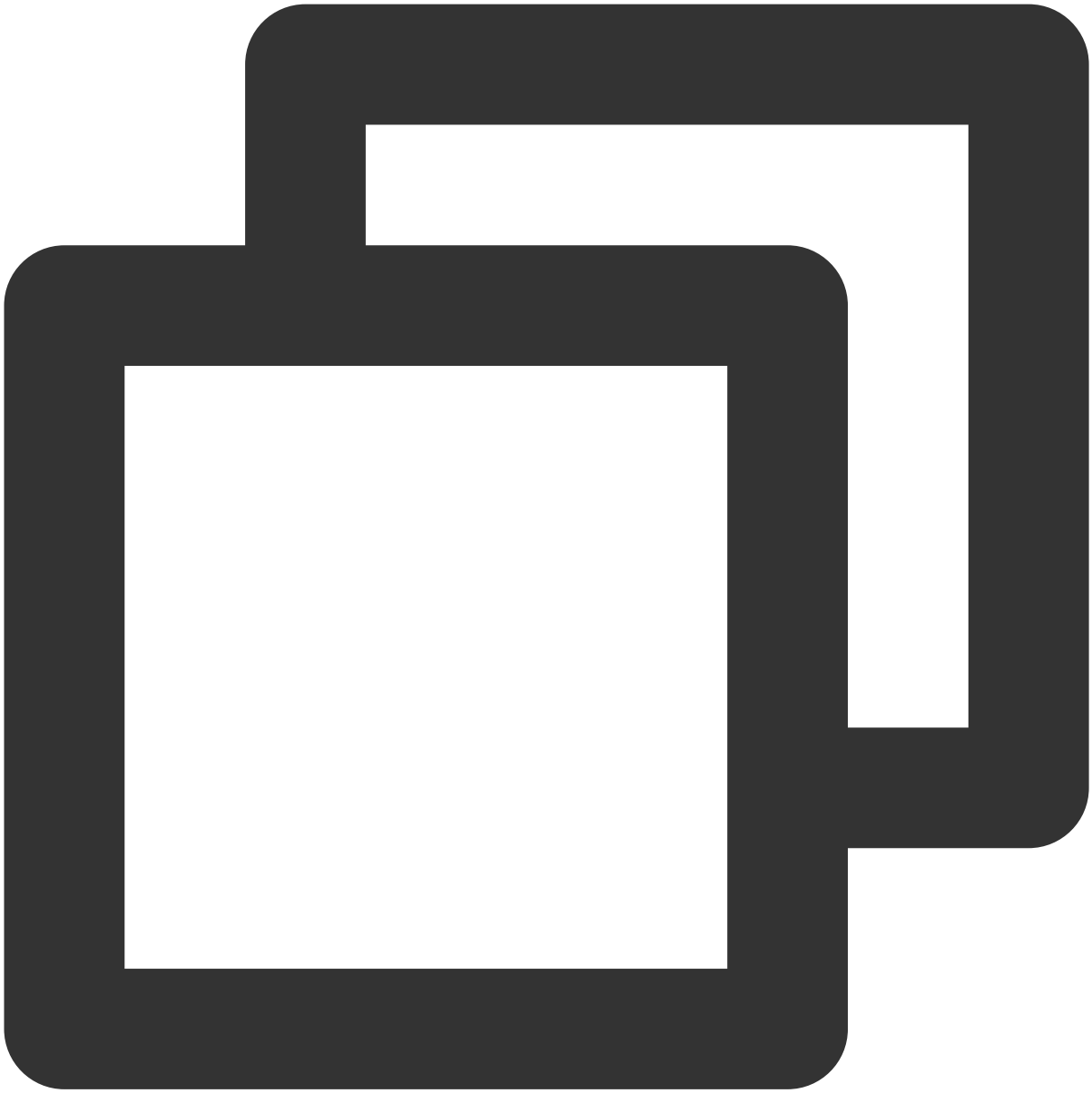
下述请求示例展示了从存储桶 `originbucket-1250000000` 中删除配置信息。



```
DELETE /?replication HTTP/1.1
Date: Fri, 14 Apr 2019 07:47:35 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: originbucket-1250000000.cos.ap-chengdu.myqcloud.com
```

响应

上述请求后，COS 返回 `204 No Content` 的响应表明已成功删除了该存储桶内的存储桶复制配置。删除存储桶复制配置后，COS 将不再复制源存储桶中的对象到目标存储桶中，目标存储桶中已有的对象数据将被保留。



```
Content-Length: 0
Connection: keep-alive
Date: Fri, 14 Apr 2019 07:47:35 GMT
Server: tencent-cos
x-cos-request-id: NWQwMzUxMTdfMjBiNDU4NjRfNWZlZF84Mjdm****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODc0OWRkZjk0ZDM1NmI1M2E2MTR
```

日志管理（logging）
PUT Bucket logging

最近更新时间：2023-12-21 12:40:39

功能描述

PUT Bucket Logging 接口用于为源存储桶开启日志记录，将源存储桶的访问日志保存到指定的目标存储桶中。

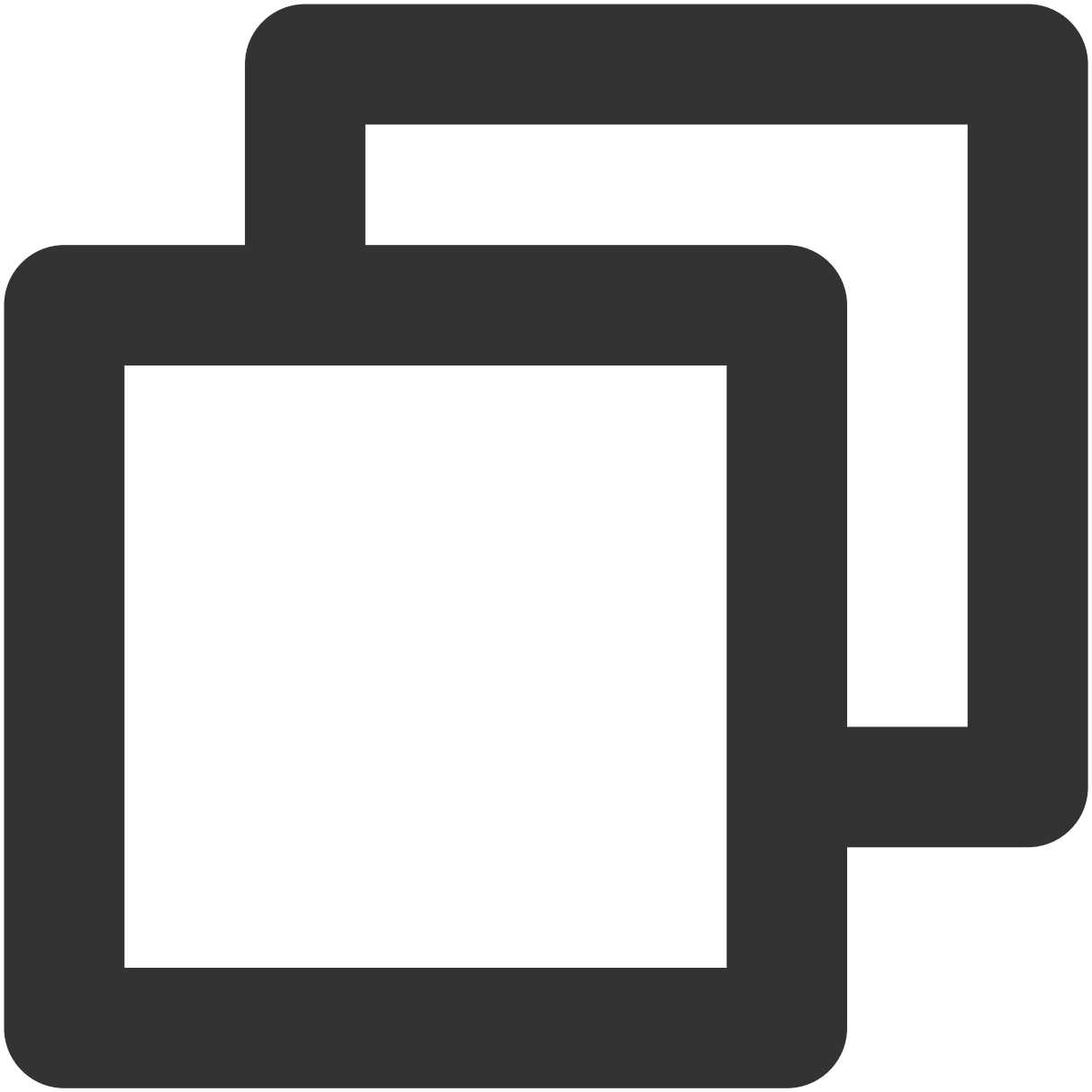
注意

只有源存储桶拥有者才可进行该请求操作。

开启访问日志功能，需要授予日志服务（CLS）产品写入 COS 的权限，授权流程请参见 启用日志管理。

请求

请求示例



```
PUT /?logging HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Content-Length: length
Content-Type: application/xml
Content-MD5: MD5
Authorization: Auth String
```

说明

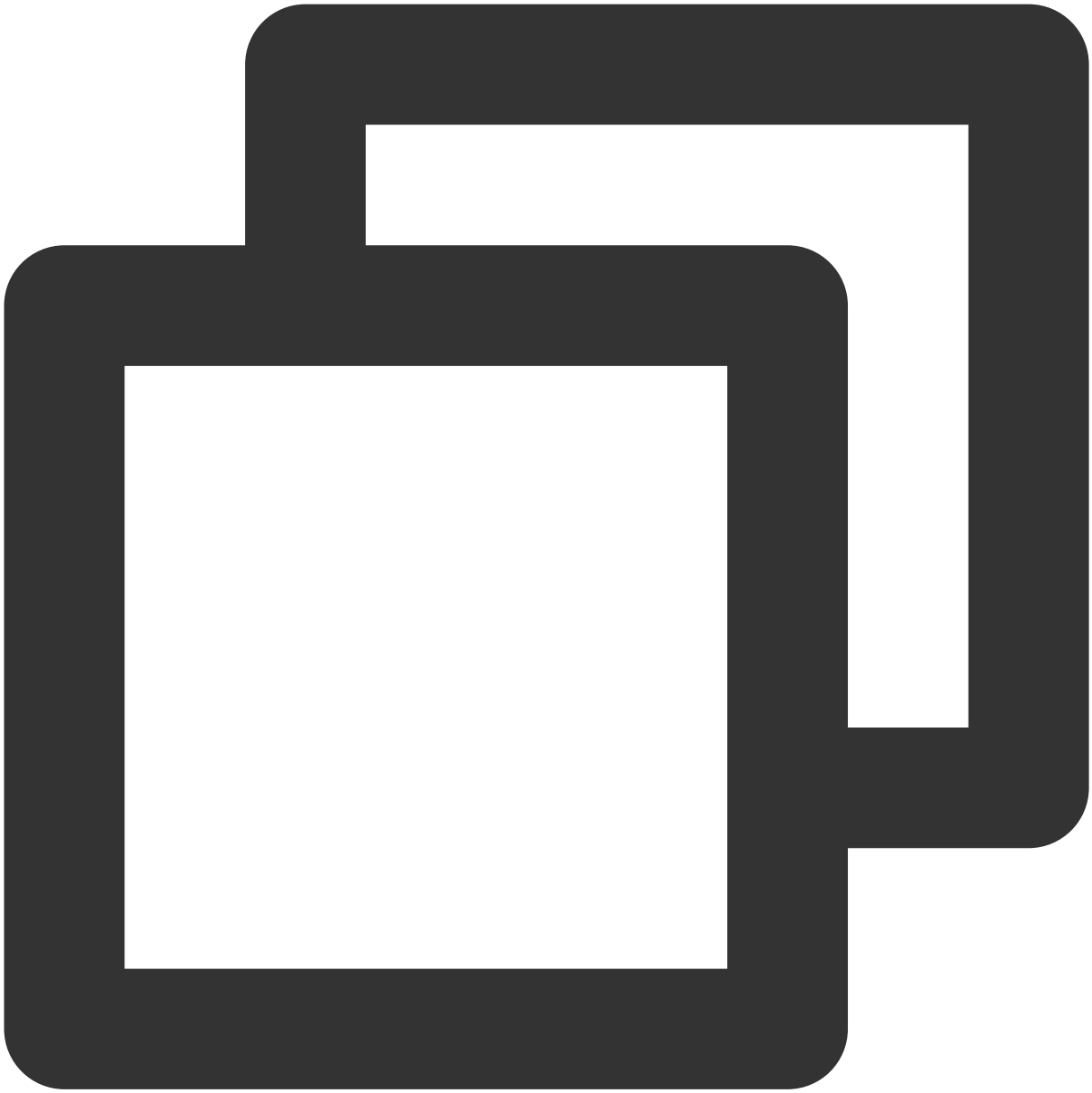
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有请求体。带所有节点的请求体内容示例如下：



```
<BucketLoggingStatus>
  <LoggingEnabled>
    <TargetBucket>examplebucket-1250000000</TargetBucket>
    <TargetPrefix>prefix</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

具体的数据描述如下：

节点名称（关键字）	父节	描述	类型	是否必选

	点			
BucketLoggingStatus	无	说明日志记录配置的状态，如果无子节点信息则意为关闭日志记录	Container	是

Container 节点 BucketLoggingStatus 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
LoggingEnabled	BucketLoggingStatus	存储桶 logging 设置的具体信息，主要是目标存储桶	Container	否

Container 节点 LoggingEnabled 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TargetBucket	LoggingEnabled	存放日志的目标存储桶，可以是同一个存储桶（但不推荐），或同一账户下、同一地域的存储桶	String	否
TargetPrefix	LoggingEnabled	日志存放在目标存储桶的指定路径	String	否

说明

用户指定存放日志的存储桶和路径后，生成的日志文件名格式为：`目标存储桶/路径前缀{YYYY}/{MM}/{DD}/{time}_{random}_{index}.gz`。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

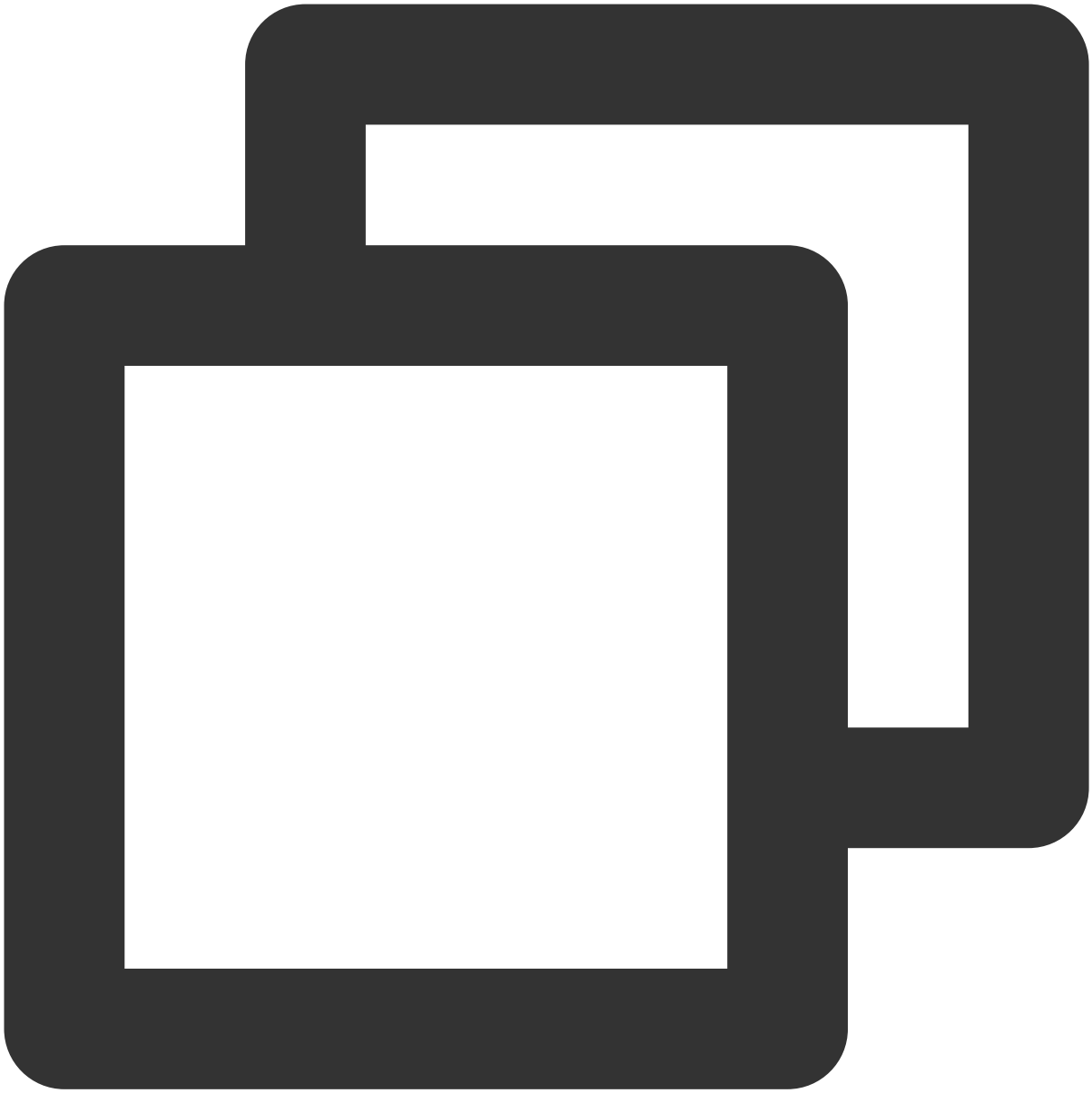
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

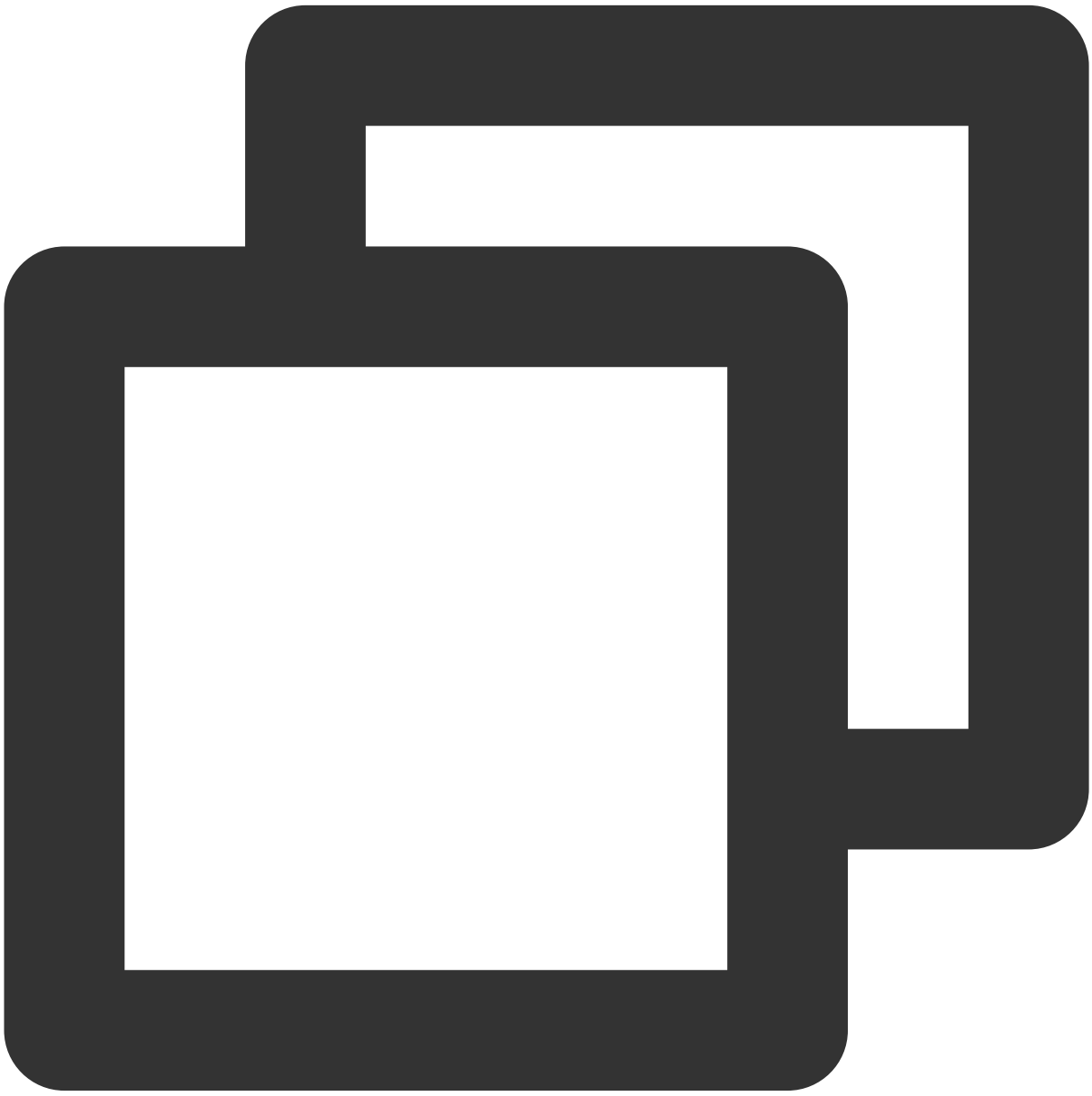
请求



```
PUT /?logging HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2017 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
Content-Type: application/xml
Content-Length: 147

<BucketLoggingStatus>
  <LoggingEnabled>
```

```
    <TargetBucket>examplebucket-1250000000</TargetBucket>
    <TargetPrefix>prefix</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
```

版权所有：腾讯云计算（北京）有限责任公司

```
Date: Fri, 10 Mar 2017 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg4MDdiZWRfOWExZjRlXzQ2OWVf****
```

GET Bucket logging

最近更新时间：2023-12-21 12:41:04

功能描述

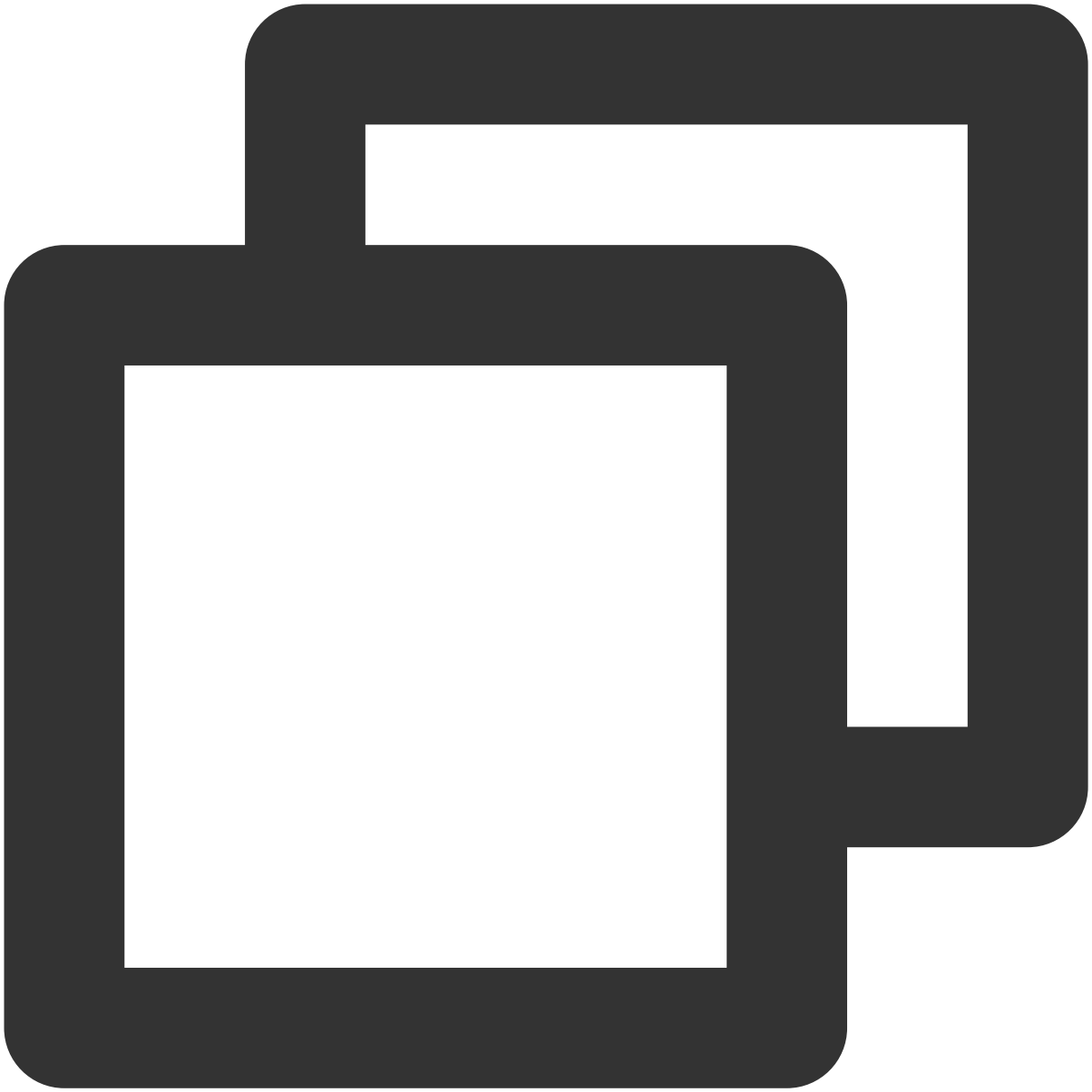
GET Bucket logging 用于获取源存储桶的日志配置信息。

注意

只有源存储桶拥有者才可进行该请求操作。

请求

请求示例



```
GET /?logging HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

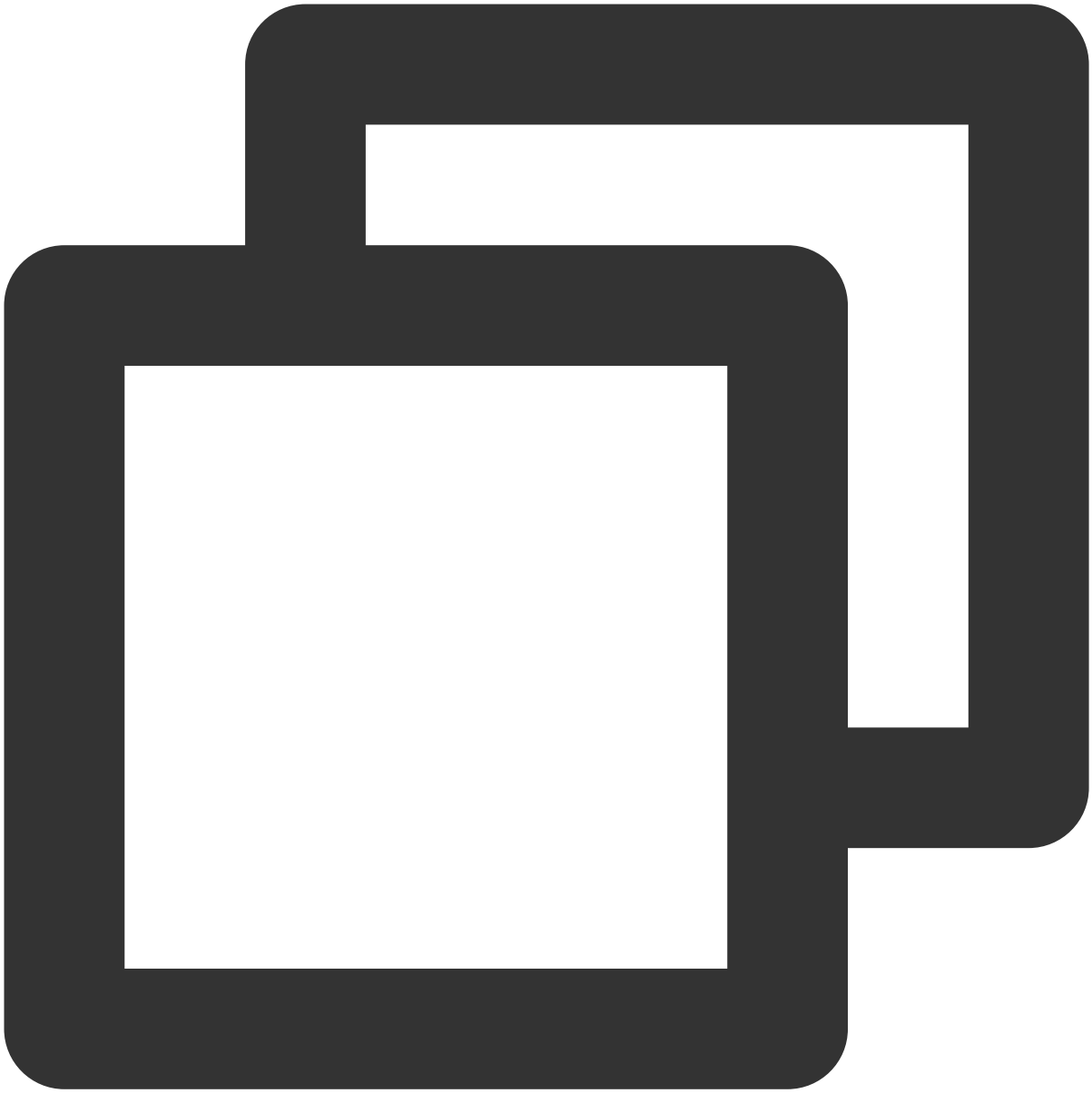
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<BucketLoggingStatus>
  <LoggingEnabled>
    <TargetBucket>examplebucket-1250000000</TargetBucket>
    <TargetPrefix>prefix</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

BucketLoggingStatus	无	存储桶日志状态信息	Container

Container 节点 BucketLoggingStatus 的内容：

节点名称（关键字）	父节点	描述	类型
LoggingEnabled	BucketLoggingStatus	存储桶日志记录配置详细信息	Container

Container 节点 LoggingEnabled 的内容：

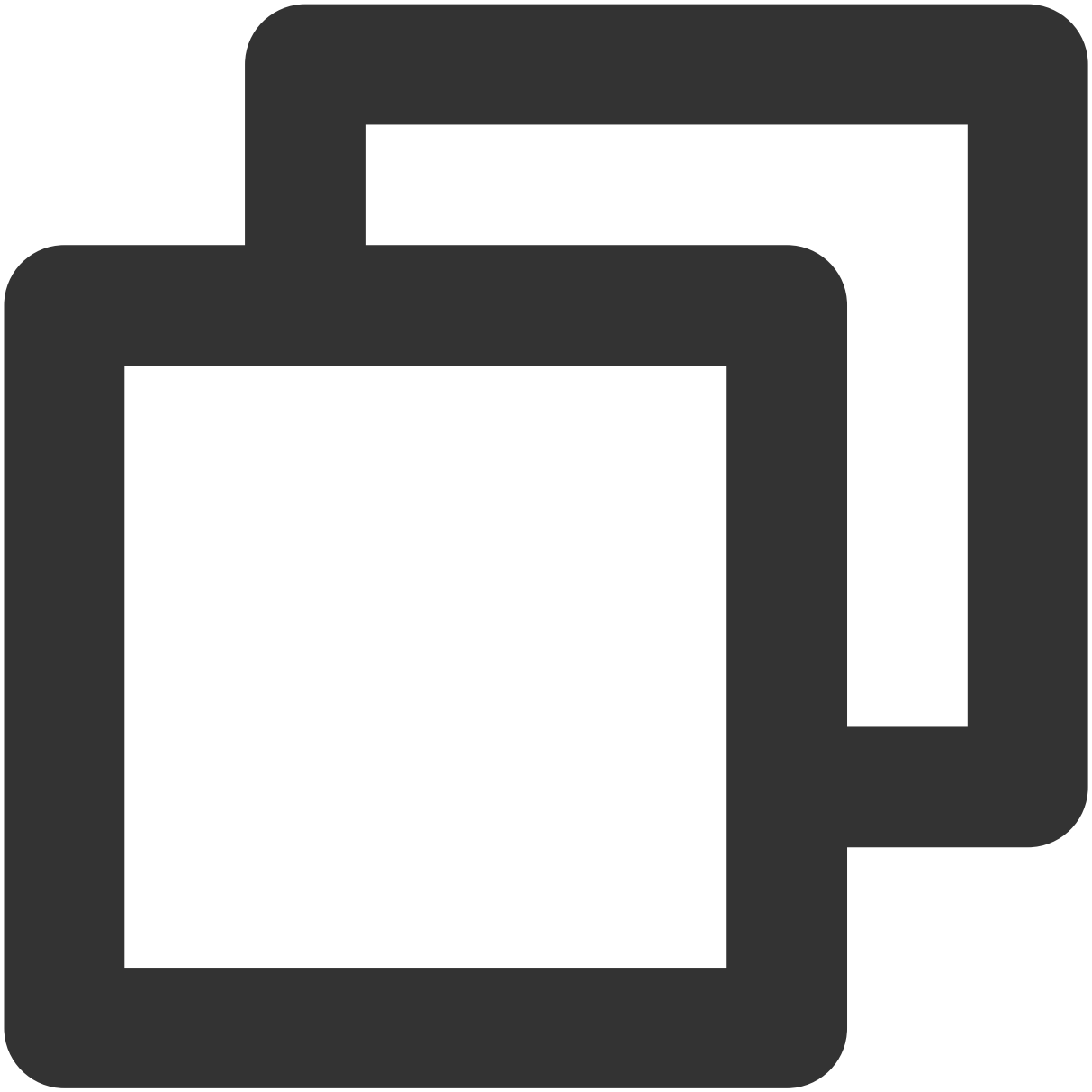
节点名称（关键字）	父节点	描述	类型
TargetBucket	LoggingEnabled	存放日志的目标存储桶，可以是同一个存储桶（但不推荐），或同一账户下、同一地域的存储桶	String
TargetPrefix	LoggingEnabled	日志存放在目标存储桶的指定路径	String

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

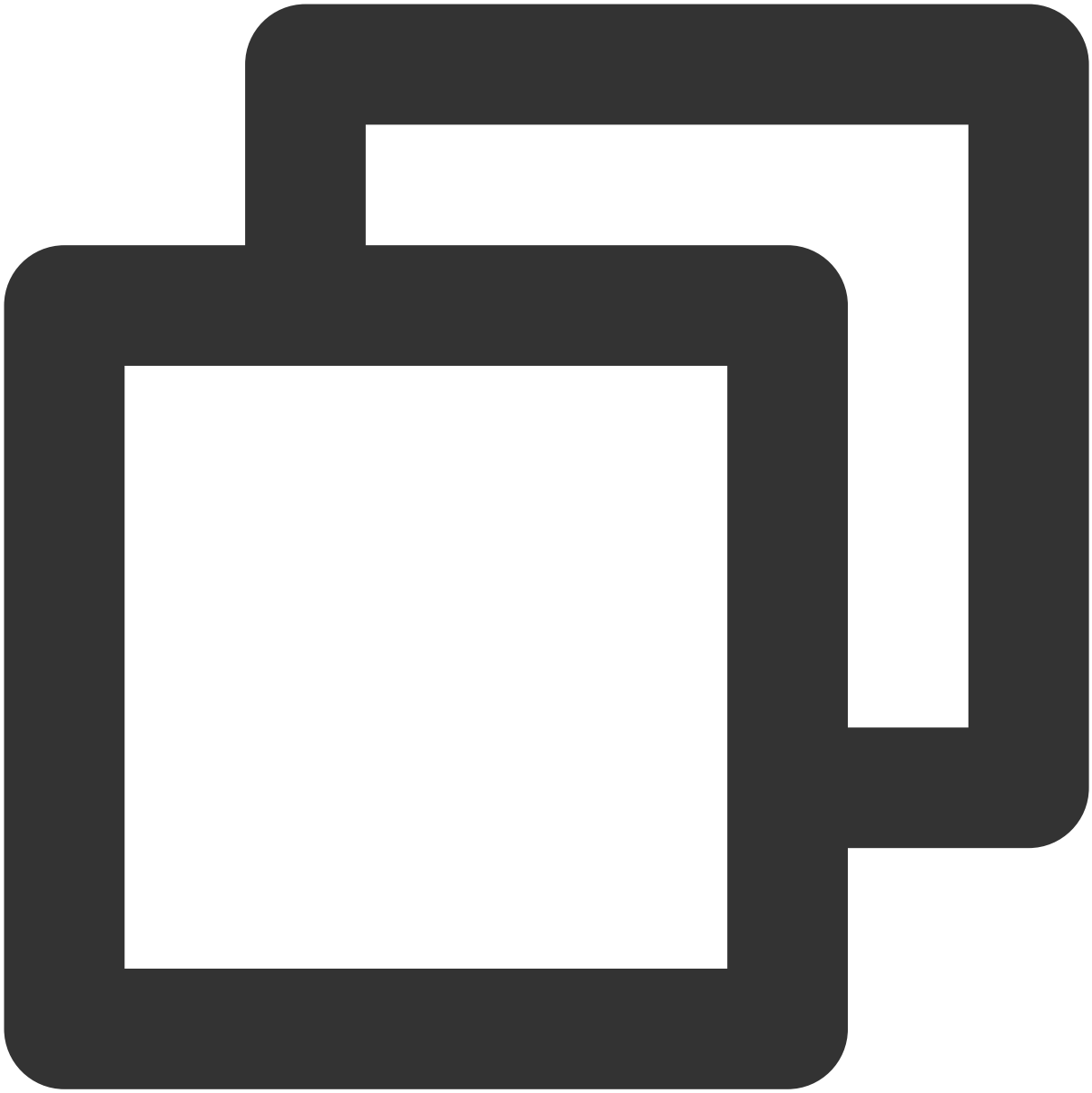
实际案例

请求



```
GET /?logging HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 28 Oct 2017 21:32:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5O****&q-sign-time=1484815944
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 142
Connection: keep-alive
Date: Wed, 28 Oct 2017 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg4MDdlNGZfNDYyMDRlXzM0YWFf****

<BucketLoggingStatus>
  <LoggingEnabled>
    <TargetBucket>examplebucket-1250000000</TargetBucket>
```

```
    <TargetPrefix>prefix</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

全球加速（Accelerate）
PUT Bucket Accelerate

最近更新时间：2023-12-21 12:40:47

功能描述

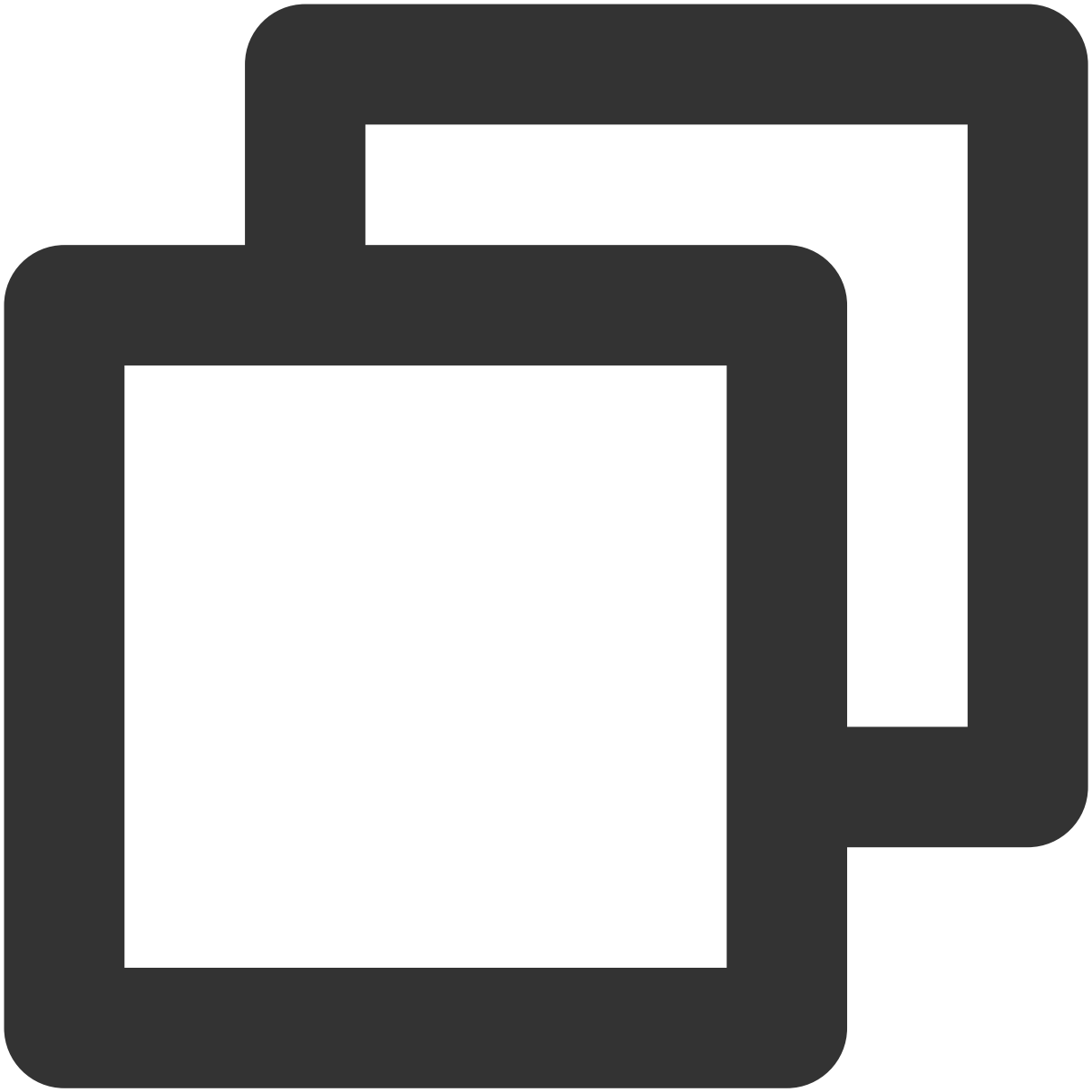
PUT Bucket Accelerate 接口实现启用或者暂停存储桶的全球加速功能。

细节分析

1. 如果您从未在存储桶上启用过全球加速功能，则 GET Bucket Accelerate 请求不返回全球加速功能配置状态。
2. 开启全球加速功能后，只能暂停，不能关闭。
3. 设置全球加速功能状态值为 Enabled 或 Suspended，表示开启或暂停全球加速功能。
4. 如果您是子账号，需要设置存储桶的全球加速功能，您需要有该配置的写入权限。

请求

请求示例



```
PUT /?accelerate HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

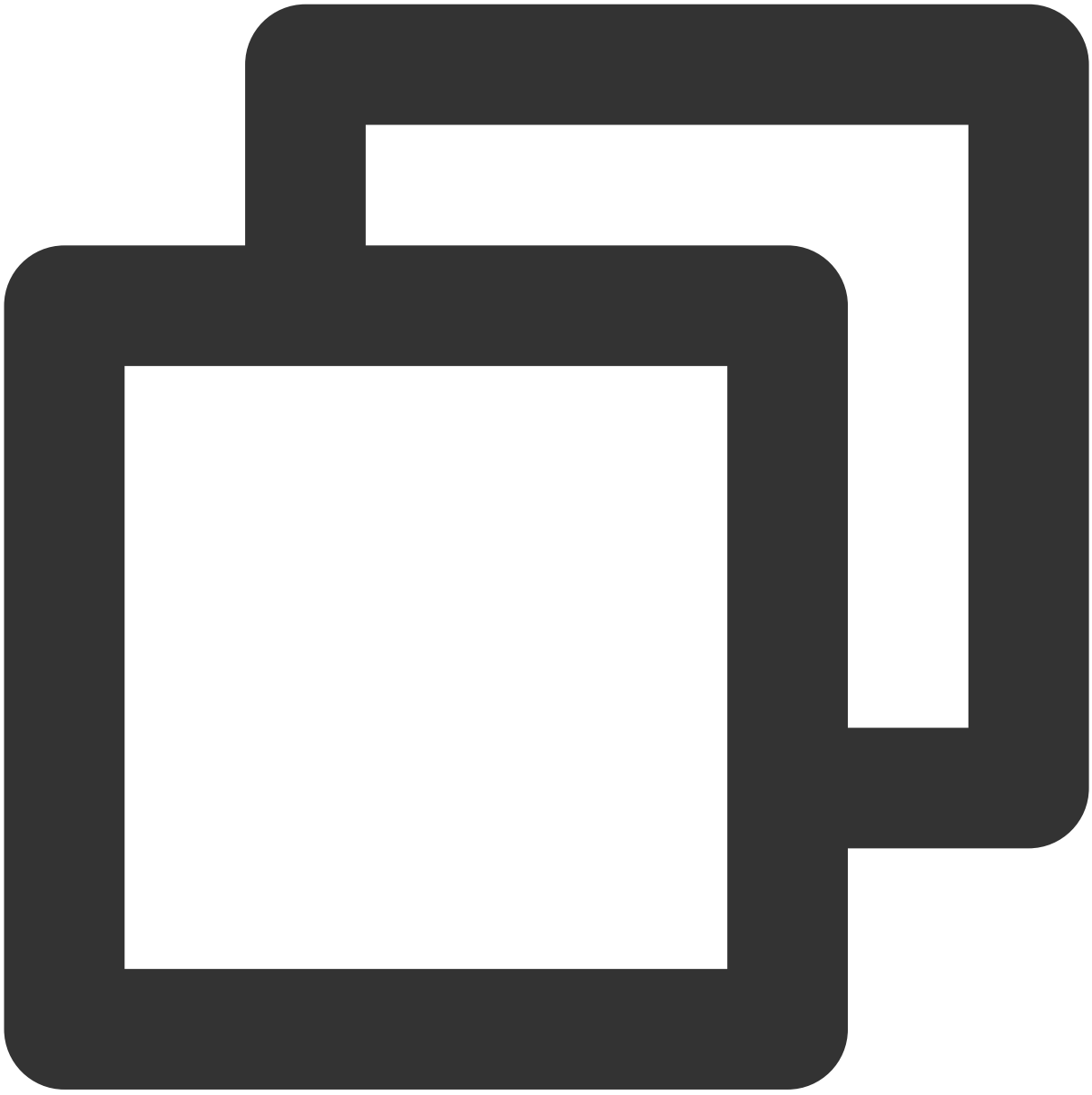
Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体



```
<AccelerateConfiguration xmlns="cos xmlns/">
  <Status>Enabled</Status>
</AccelerateConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
AccelerateConfiguration	无	全球加速的具体信息	Container
Status	AccelerateConfiguration	说明全球加速功能是否开启，枚举值：Suspended、Enabled	Enum

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

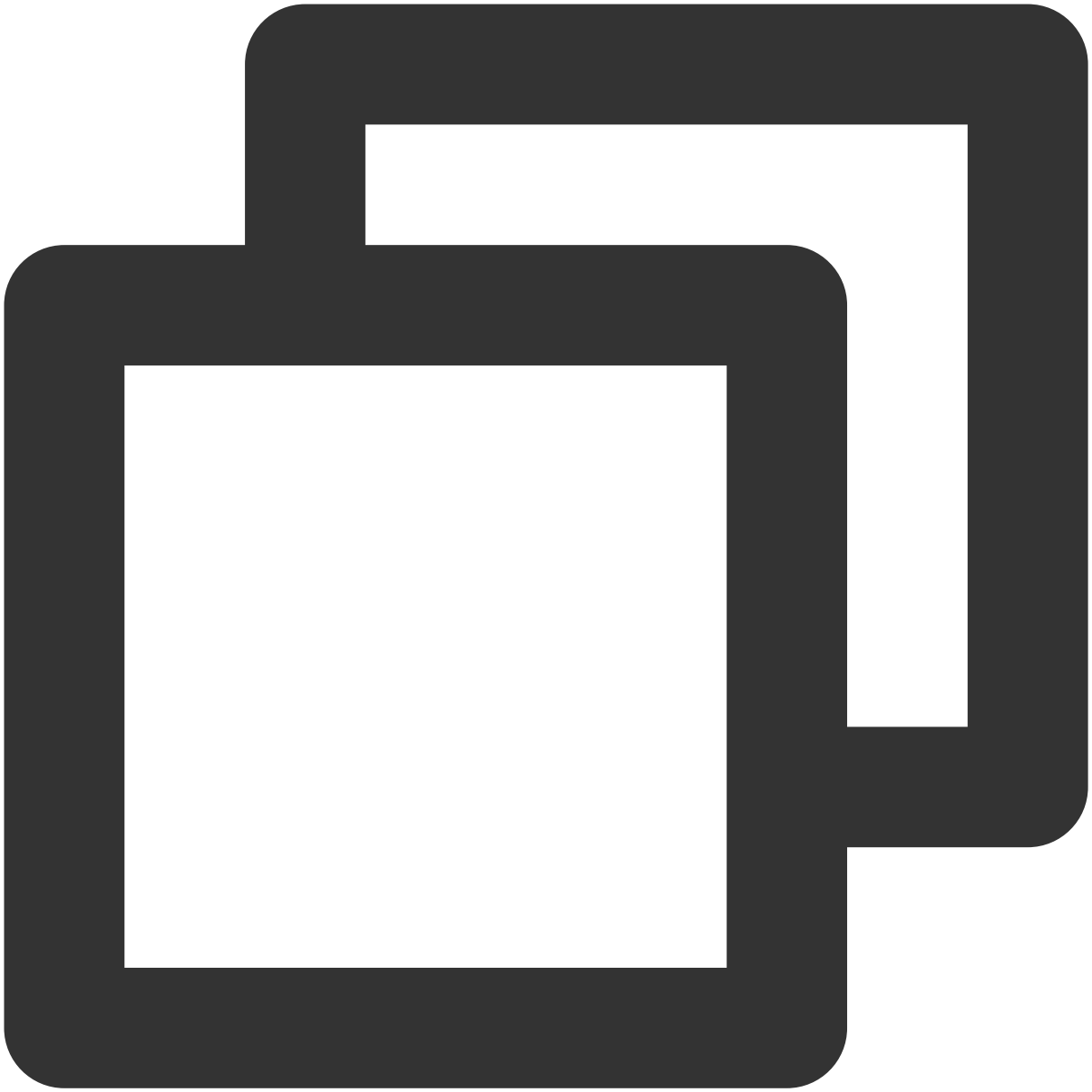
该响应体返回为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

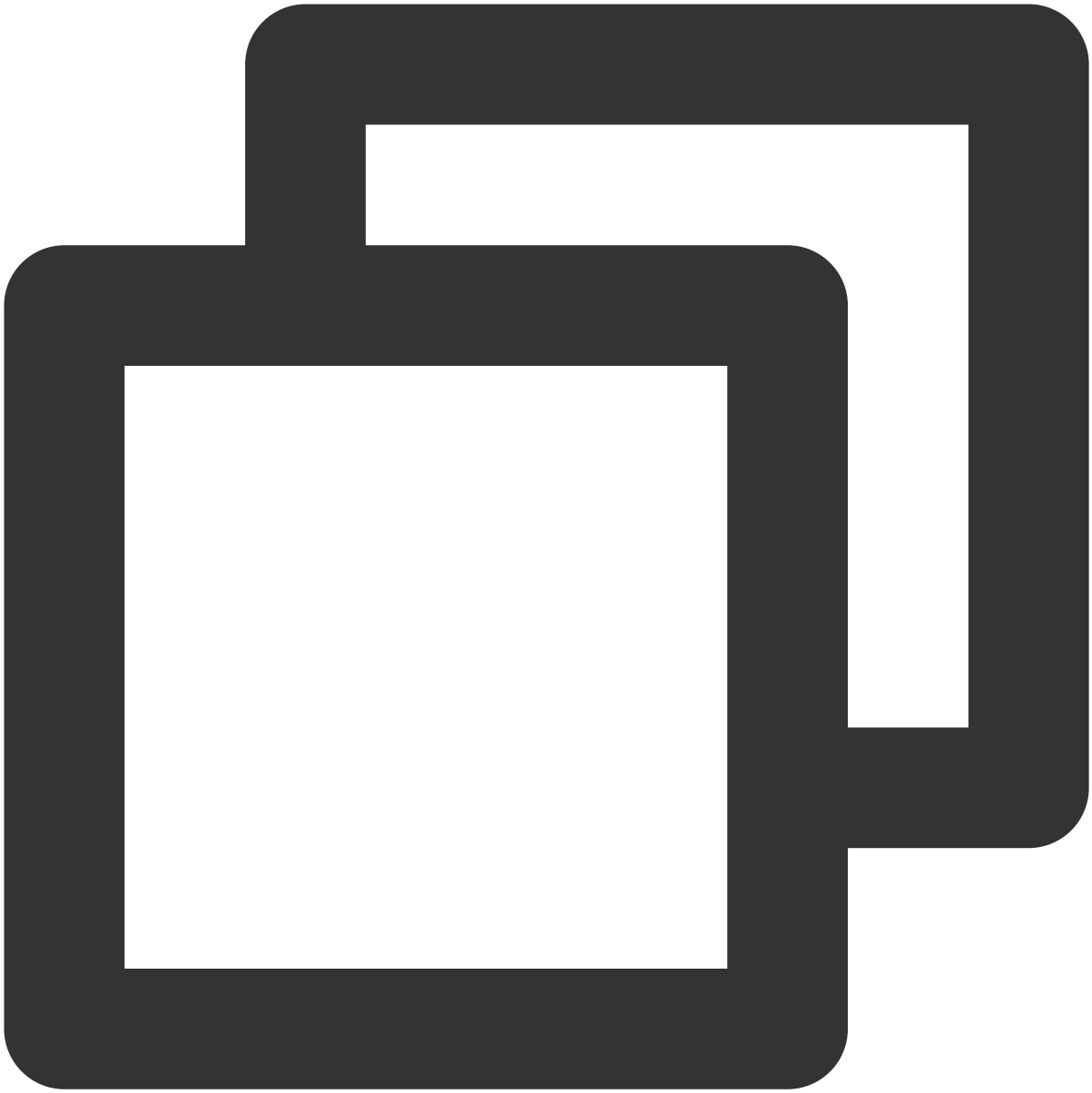
实际案例

请求



```
PUT /?accelerate HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Authorization: authorization string
Content-Type: text/plain
Content-Length: 83

<AccelerateConfiguration>
    <Status>Enabled</Status>
</AccelerateConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****
```

GET Bucket Accelerate

最近更新时间：2023-12-21 12:41:13

功能描述

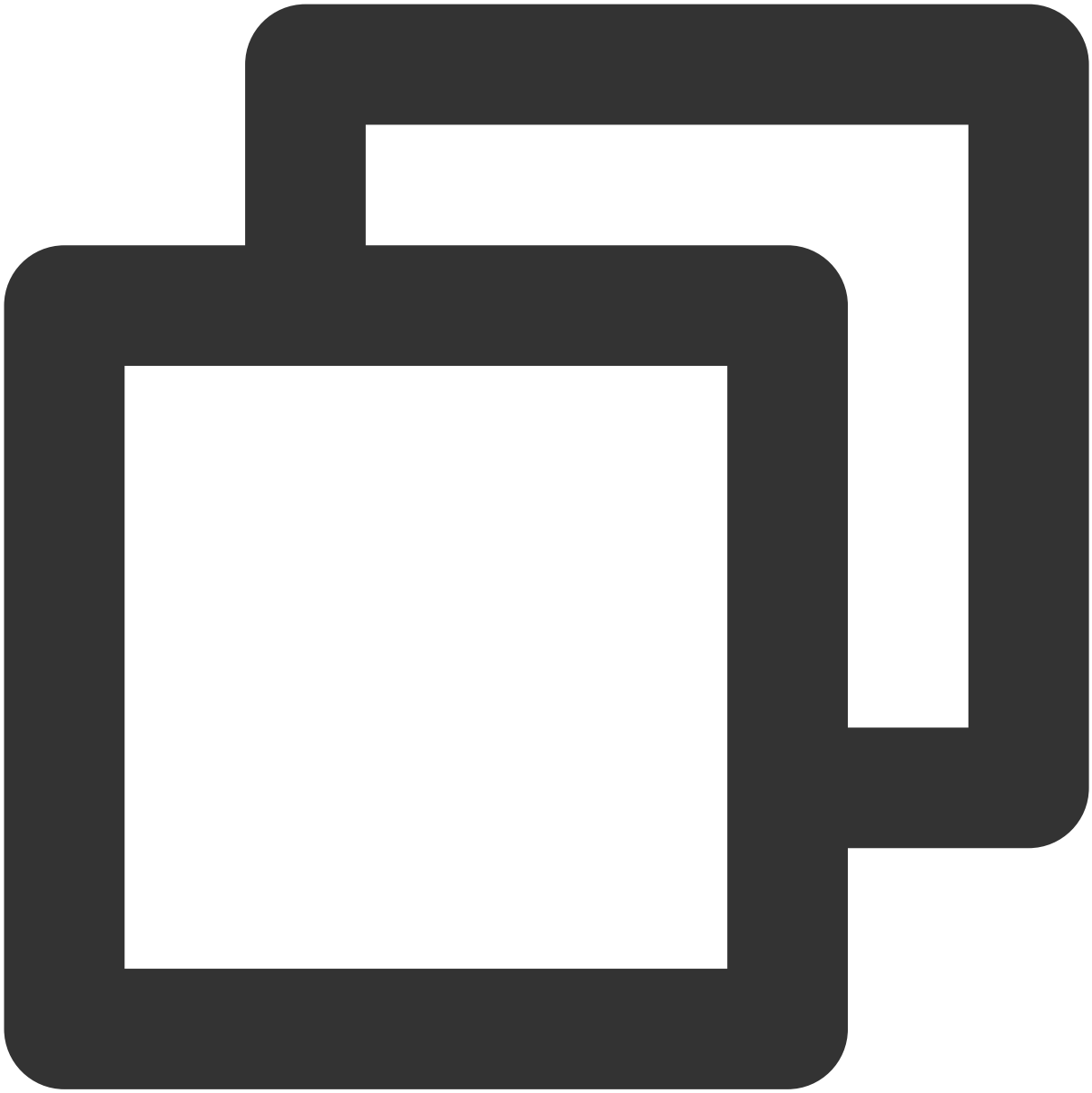
GET Bucket Accelerate 接口实现查询存储桶的全球加速功能配置。

细节分析

1. 如果您从未在存储桶上启用过全球加速功能，则 GET Bucket Accelerate 请求不返回全球加速功能配置状态。
2. 全球加速功能状态值合法返回值为 Enabled 或者 Suspended，表示开启全球加速功能和暂停全球加速功能。
3. 如果您是为子账号，需要查询存储桶的全球加速功能配置信息，您需要有该配置的读取权限。

请求

请求示例



```
GET /?accelerate HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

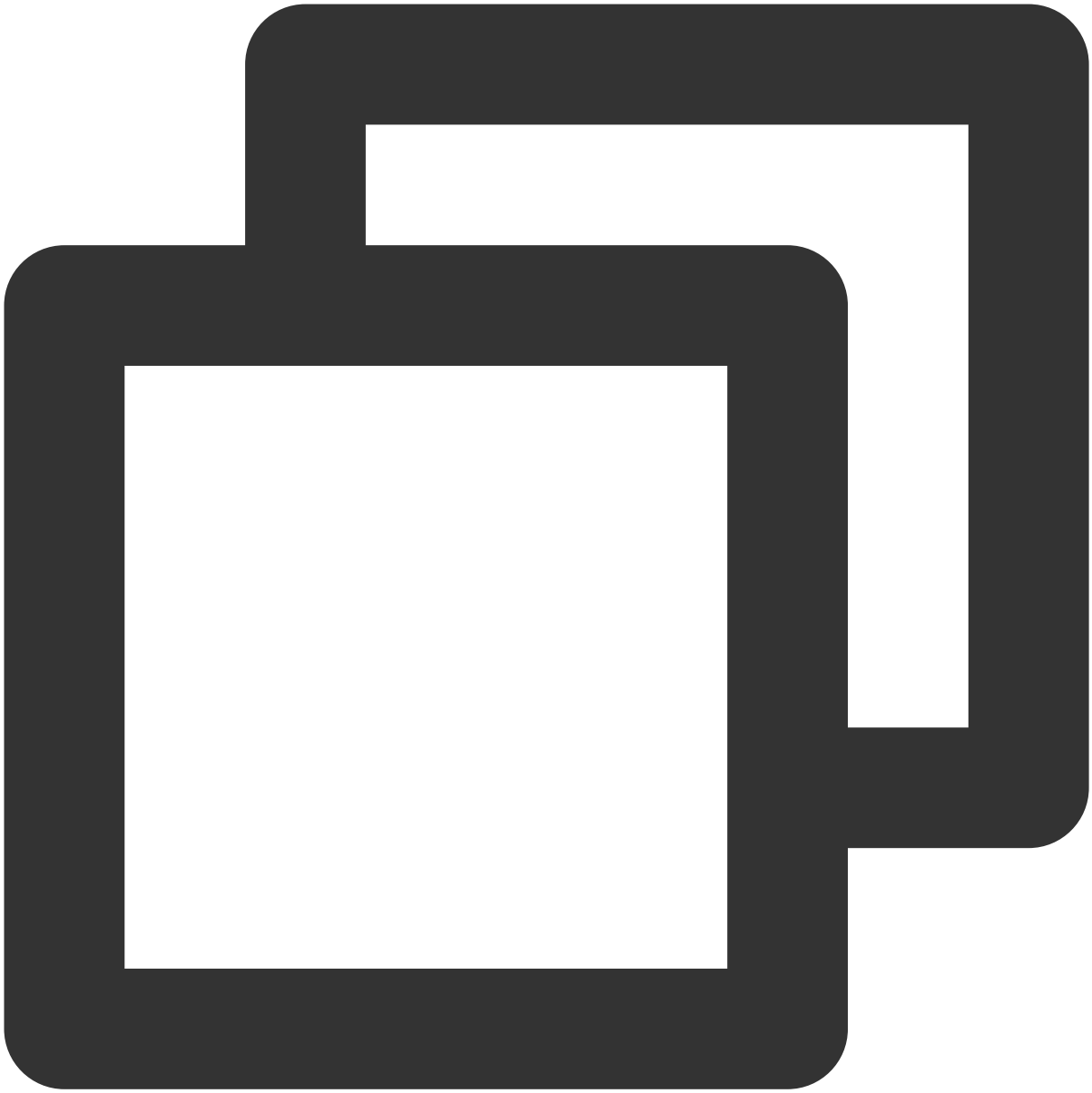
该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<AccelerateConfiguration xmlns="cos xmlns/">
  <Status>Enabled</Status>
  <Type>COS</Type>
</AccelerateConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
AccelerateConfiguration	无	全球加速的具体信息	Container

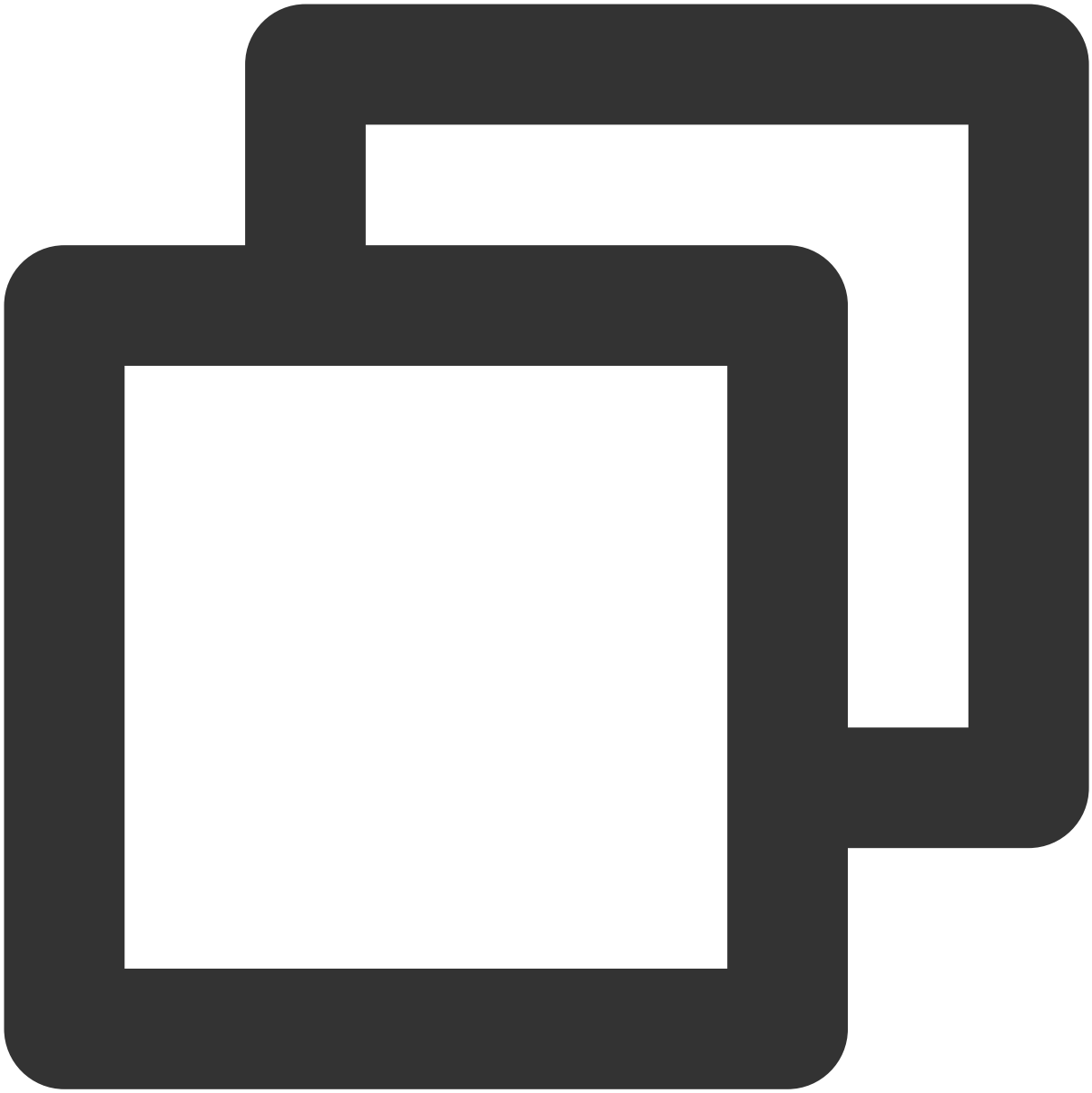
Status	AccelerateConfiguration	说明全球加速功能是否开启，枚举值：Suspended、Enabled	Enum
Type	AccelerateConfiguration	指定全球加速功能的类型，枚举值：COS	Enum

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

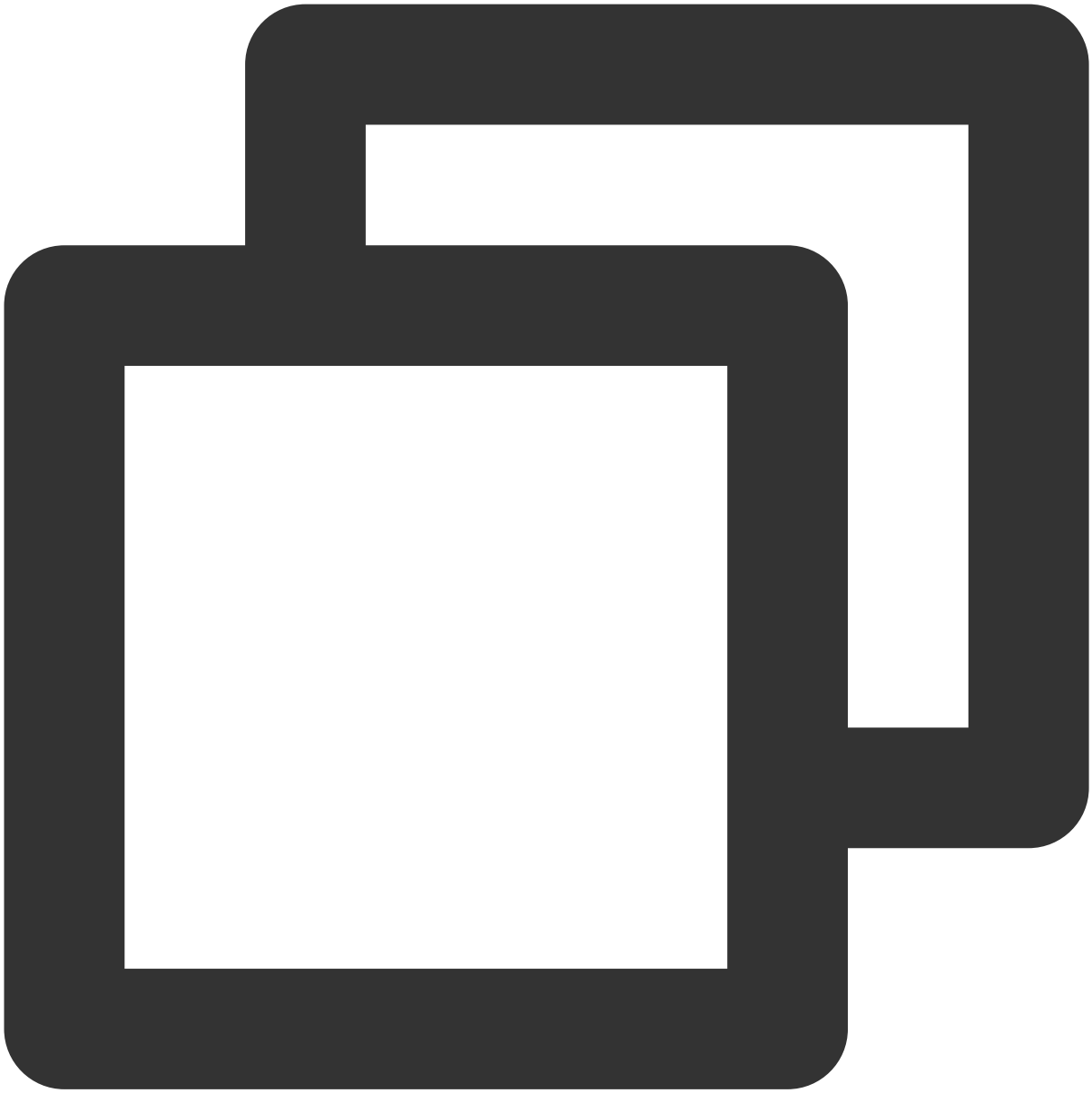
实际案例

请求



```
GET /?accelerate HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Authorization: authorization string
Content-Type: text/plain
```

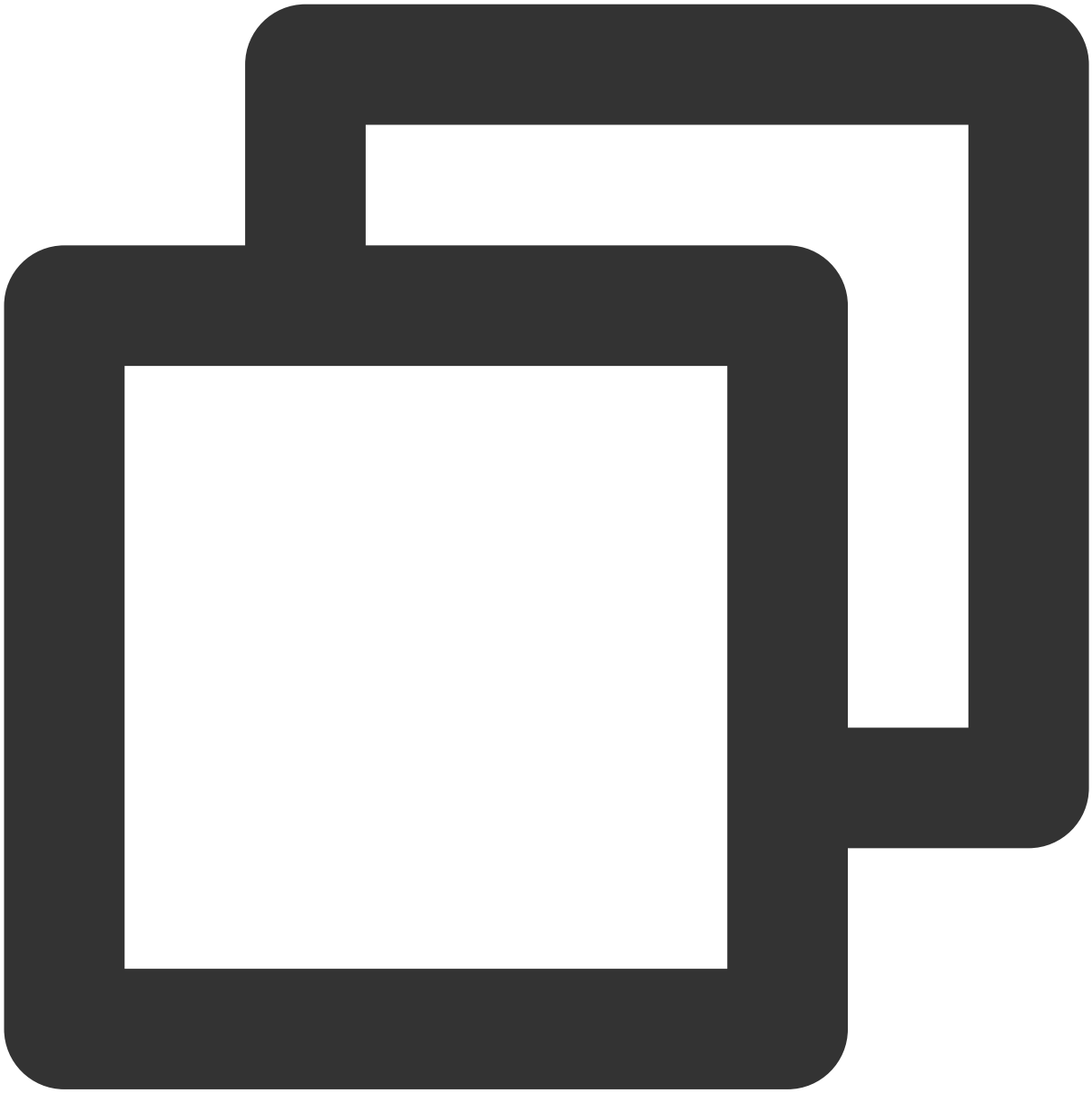
响应1（开启全球加速状态）



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 73
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****

<AccelerateConfiguration>
  <Status>Enabled</Status>
  <Type>COS</Type>
```

```
</AccelerateConfiguration>
```

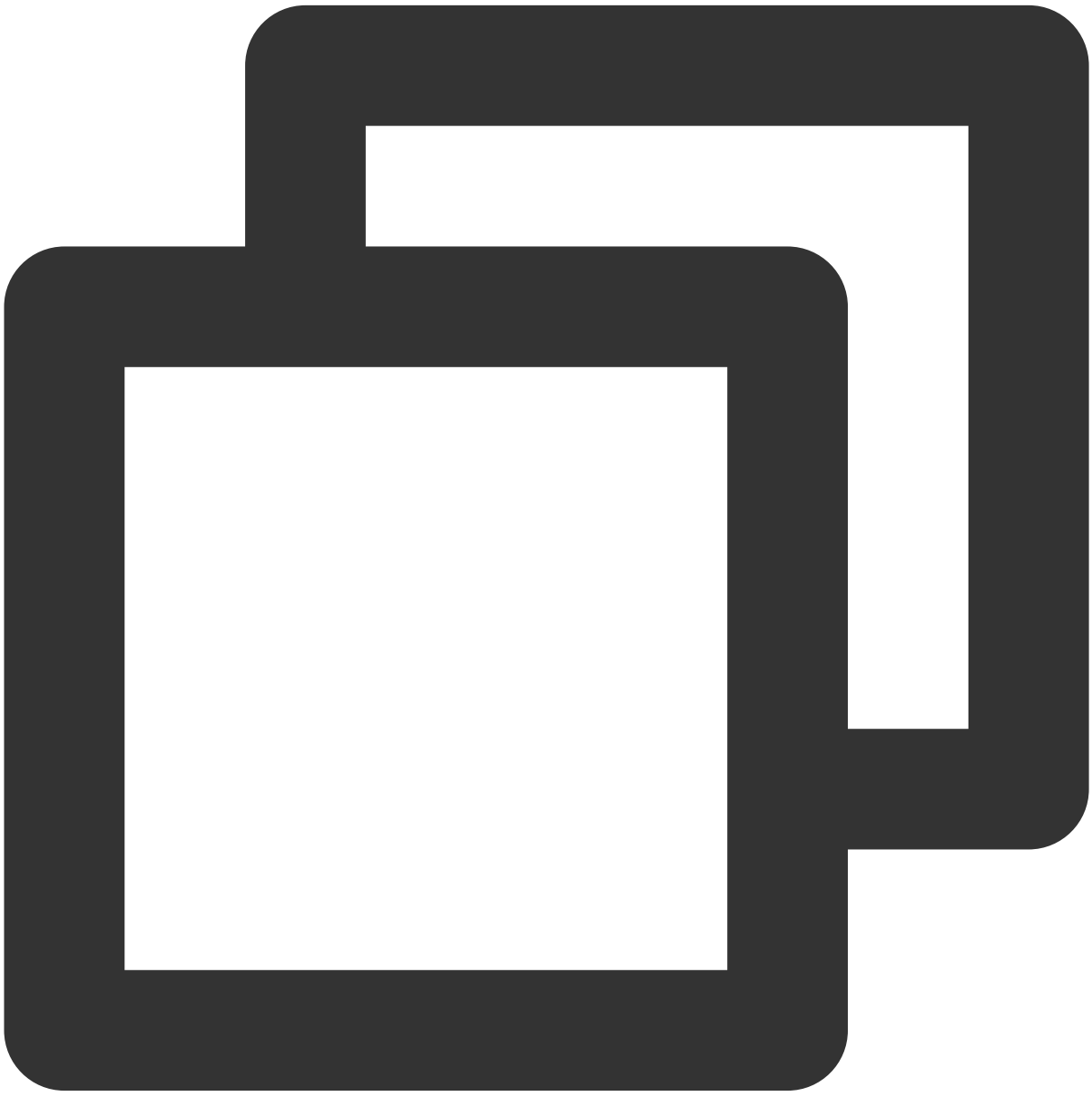
响应2（暂停全球加速状态）



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 73
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****
```

```
<AccelerateConfiguration>
  <Status>Disabled</Status>
  <Type>COS</Type>
</AccelerateConfiguration>
```

响应3（未开启全球加速状态）



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 73
```

```
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****

<AccelerateConfiguration/>
```

存储桶加密（encryption）

PUT Bucket encryption

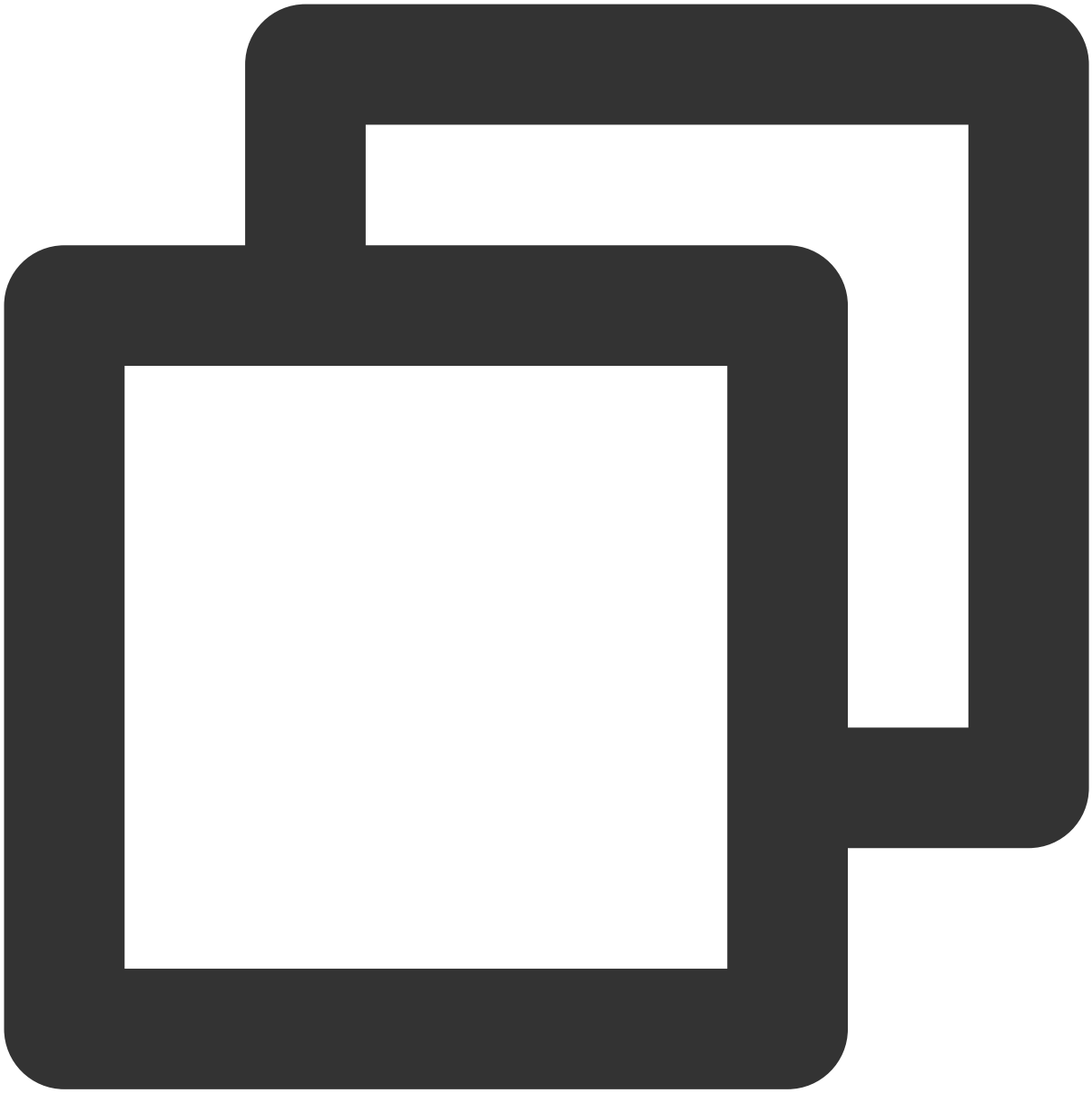
最近更新时间：2023-12-21 12:44:29

功能描述

PUT Bucket encryption 接口用于设置指定存储桶下的默认加密配置。
要执行此接口，必须拥有 PutBucketEncryption 权限。默认情况下，Bucket 的持有者直接拥有权限使用该 API 接口，Bucket 持有者也可以将权限授予其他用户。

请求

请求示例



```
PUT /?encryption HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

此接口无请求参数。

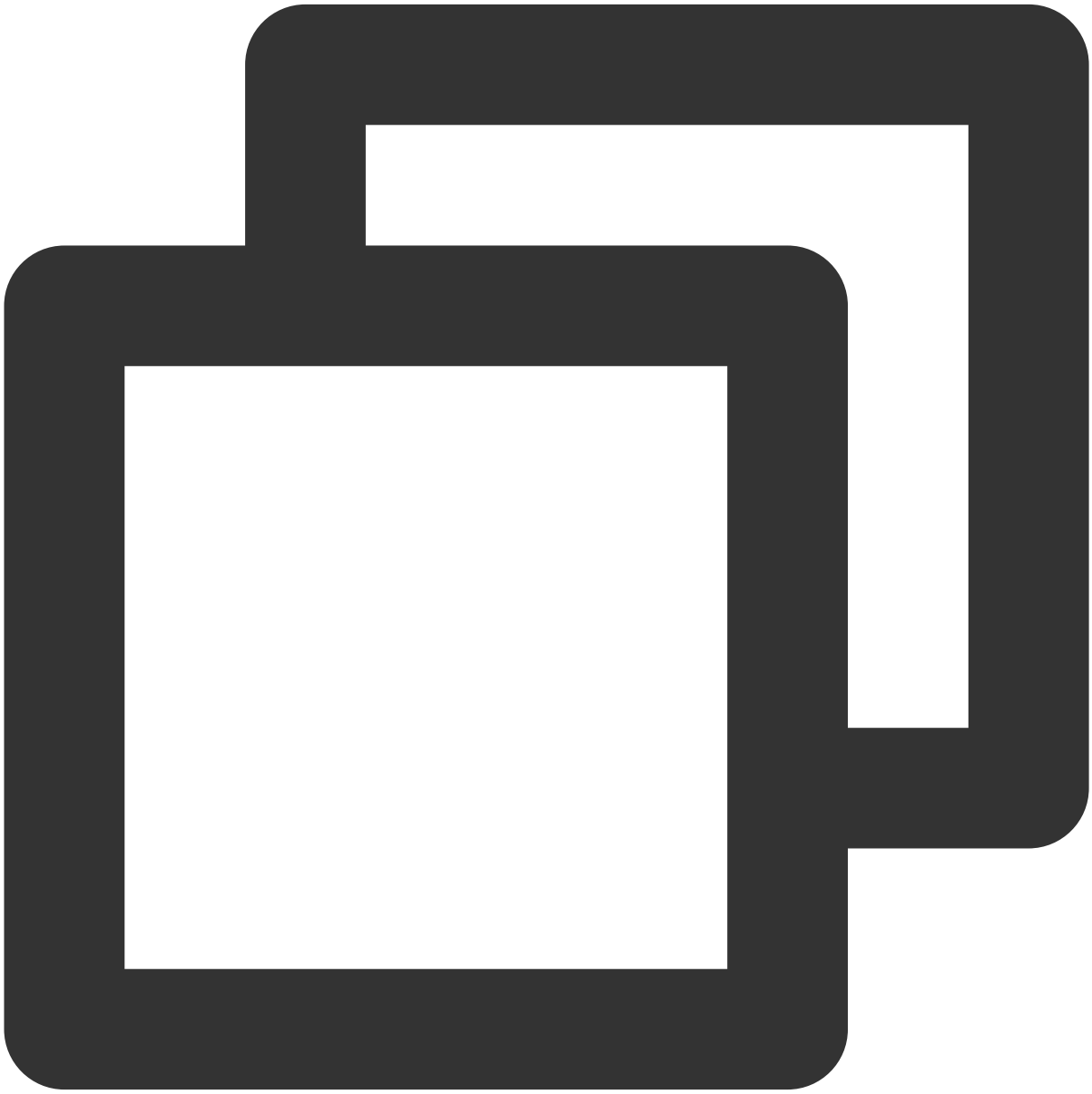
请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

用户在请求体中使用 XML 语言设置存储桶默认加密配置信息。加密配置信息主要为加密项。

以下是用于设置 SSE-COS 的请求体：



```
<ServerSideEncryptionConfiguration>
      <Rule>
```

```
        <ApplyServerSideEncryptionByDefault>
            <SSEAlgorithm>AES256|KMS</SSEAlgorithm>
            <KMSMasterKeyID>String</KMSMasterKeyID>
        </ApplyServerSideEncryptionByDefault>
    </Rule>
</ServerSideEncryptionConfiguration>
```

具体元素如下：

元素名称	父节点
ServerSideEncryptionConfiguration	无
Rule	ServerSideEncryptionConfiguration
ApplyServerSideEncryptionByDefault	ServerSideEncryptionConfiguration.Rule
SSEAlgorithm	ServerSideEncryptionConfiguration.Rule.ApplyServerSideEncryptionByDe
KMSMasterKeyID	ServerSideEncryptionConfiguration.Rule.ApplyServerSideEncryptionByDe

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求的响应体返回为空。

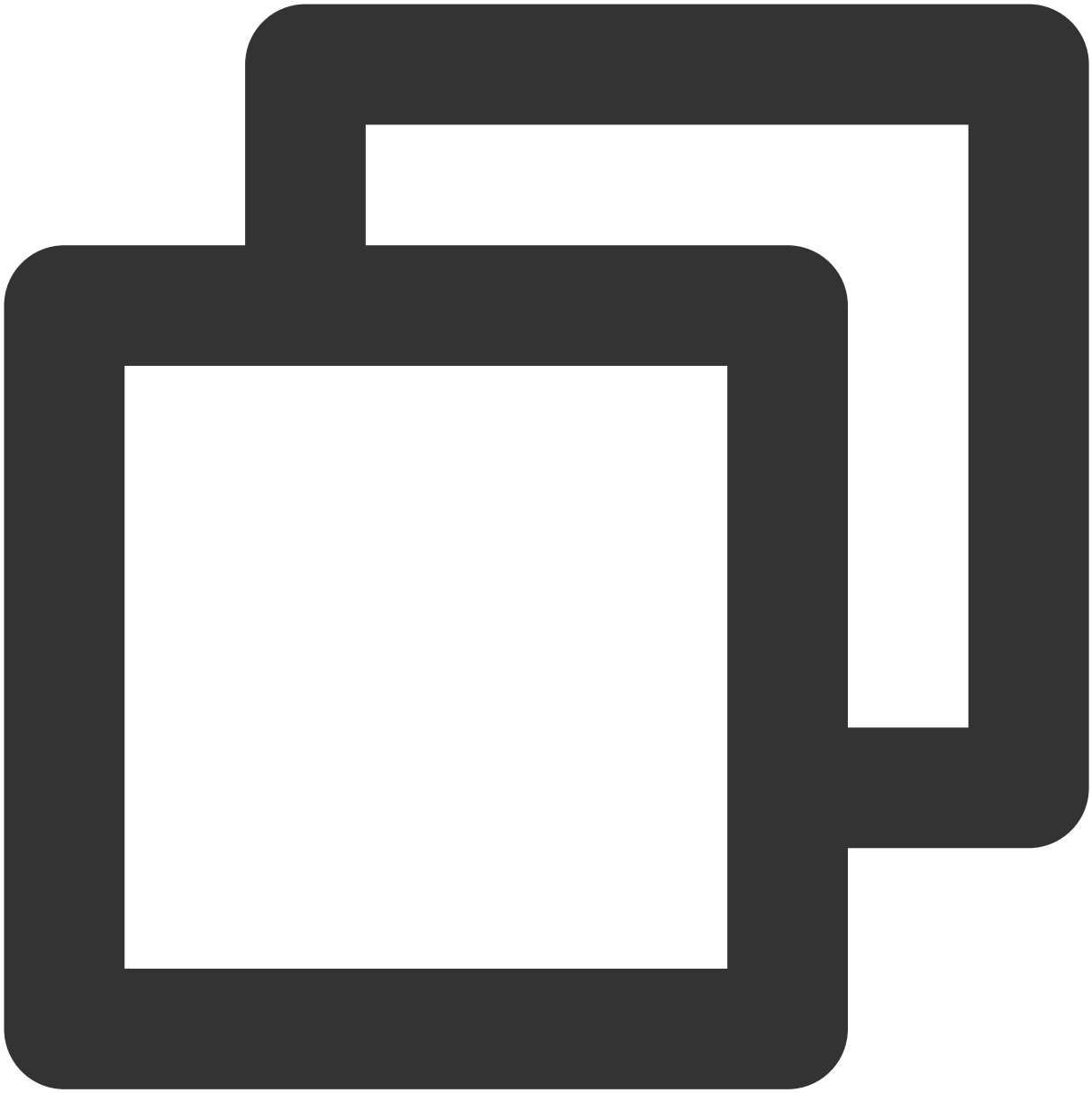
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

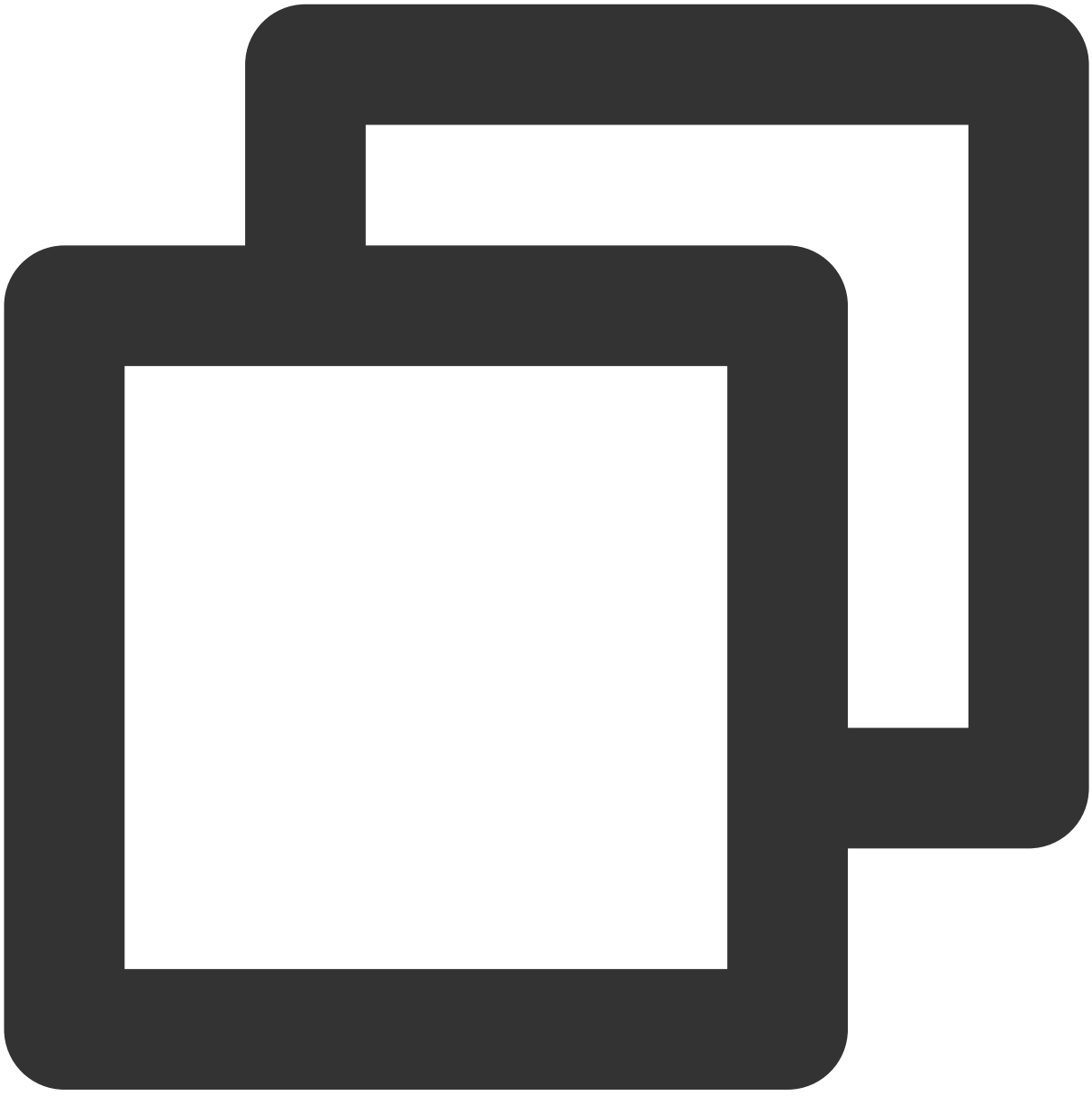
以下示例表示给存储桶 examplebucket-1250000000 设置 SSE-COS 加密。



```
PUT /?encryption HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: signatureValue



<ServerSideEncryptionConfiguration>
      <Rule>
         <ApplyServerSideEncryptionByDefault>
             <SSEAlgorithm>AES256</SSEAlgorithm>
```

```
        </ApplyServerSideEncryptionByDefault>
    </Rule>
</ServerSideEncryptionConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, 17 Jun 2019 08:37:36 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****
```

版权所有：腾讯云计算（北京）有限责任公司

GET Bucket encryption

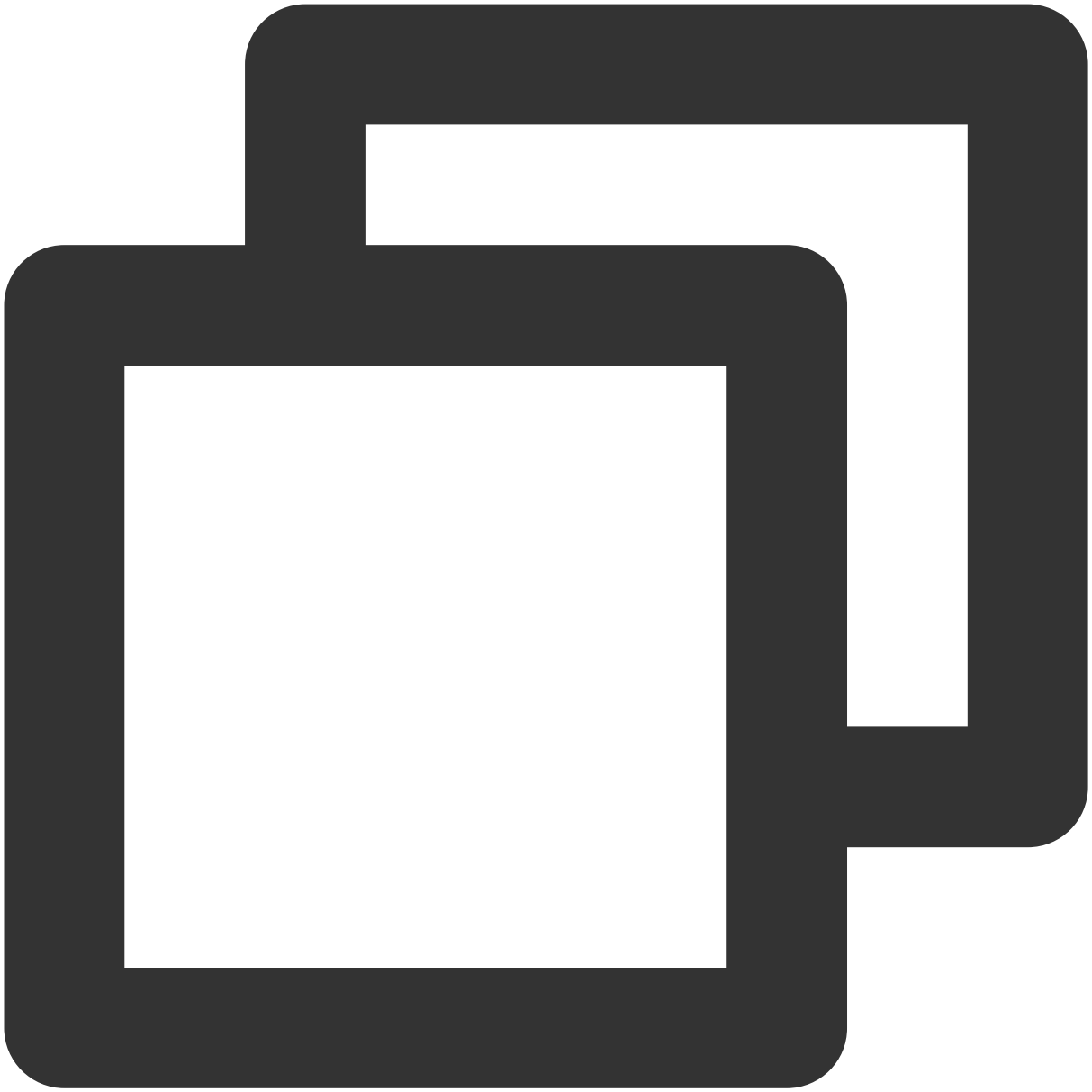
最近更新时间：2023-12-21 12:45:05

功能描述

GET Bucket encryption 接口用于查询指定存储桶下的默认加密配置。
要执行此接口，必须拥有 GetBucketEncryption 权限。默认情况下，Bucket 的持有者直接拥有权限使用该 API 接口，Bucket 持有者也可以将权限授予其他用户。

请求

请求示例



```
GET /?encryption HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

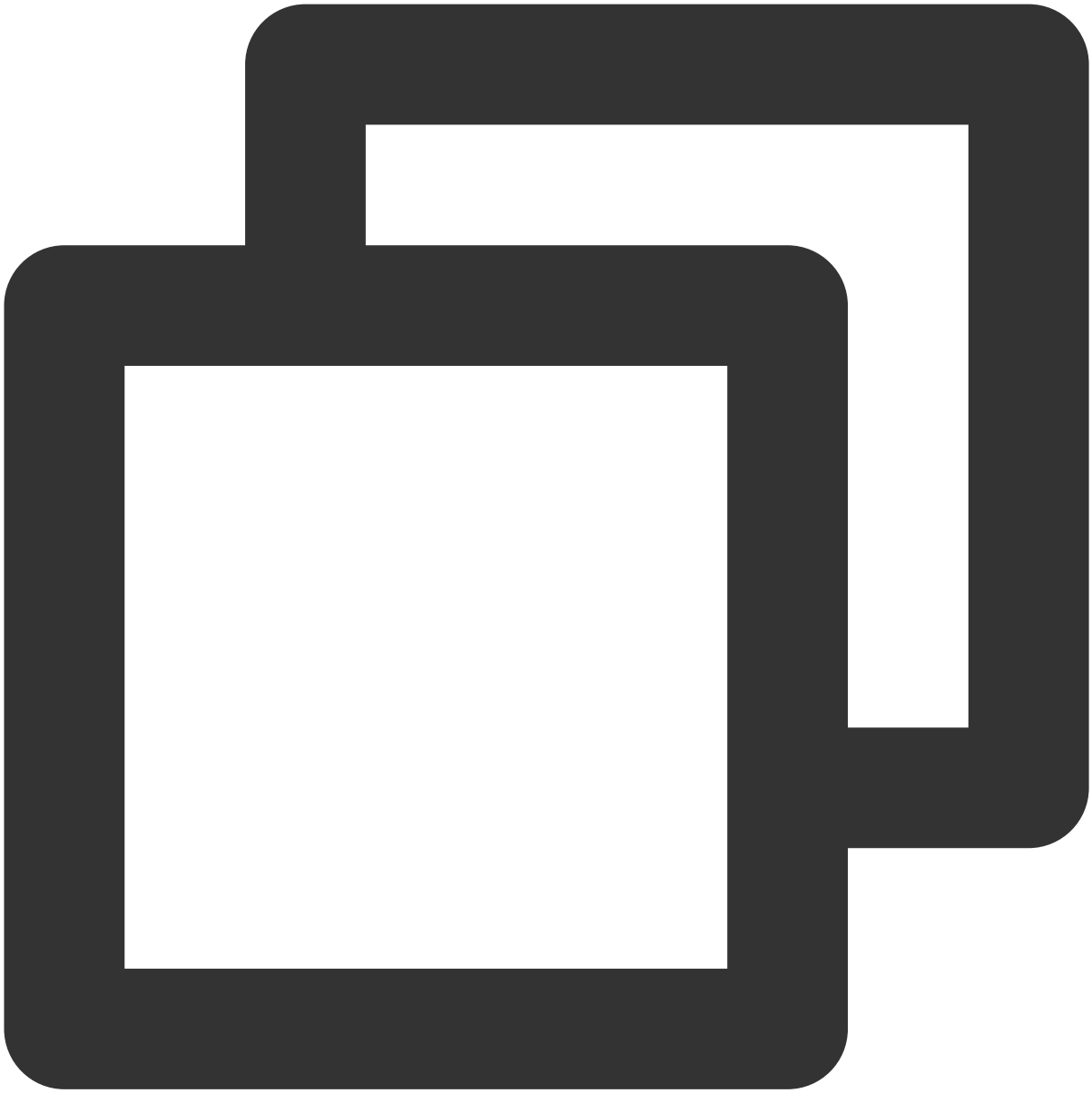
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

以下是返回 SSE-COS 加密的响应体：



```
<ServerSideEncryptionConfiguration>
      <Rule>
            <ApplyServerSideEncryptionByDefault>
                <SSEAlgorithm>AES256|KMS</SSEAlgorithm>
                <KMSMasterKeyID>String</KMSMasterKeyID>
            </ApplyServerSideEncryptionByDefault>
      </Rule>
</ServerSideEncryptionConfiguration>
```

具体元素如下：

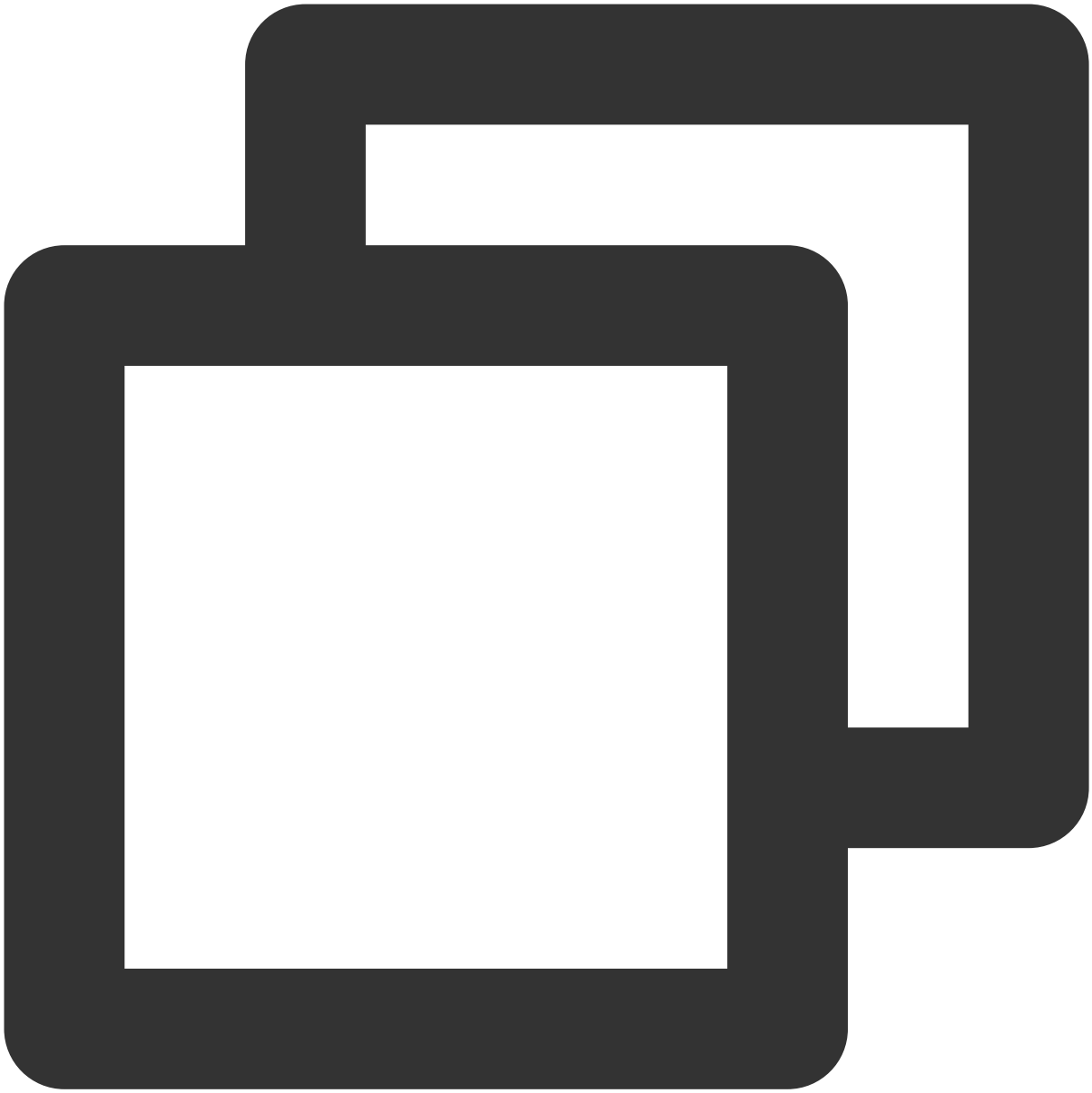
元素名称	父节点
ServerSideEncryptionConfiguration	无
Rule	ServerSideEncryptionConfiguration
ApplyServerSideEncryptionByDefault	ServerSideEncryptionConfiguration.Rule
SSEAlgorithm	ServerSideEncryptionConfiguration.Rule.ApplyServerSideEncryptionByDe
KMSMasterKeyID	ServerSideEncryptionConfiguration.Rule.ApplyServerSideEncryptionByDe

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

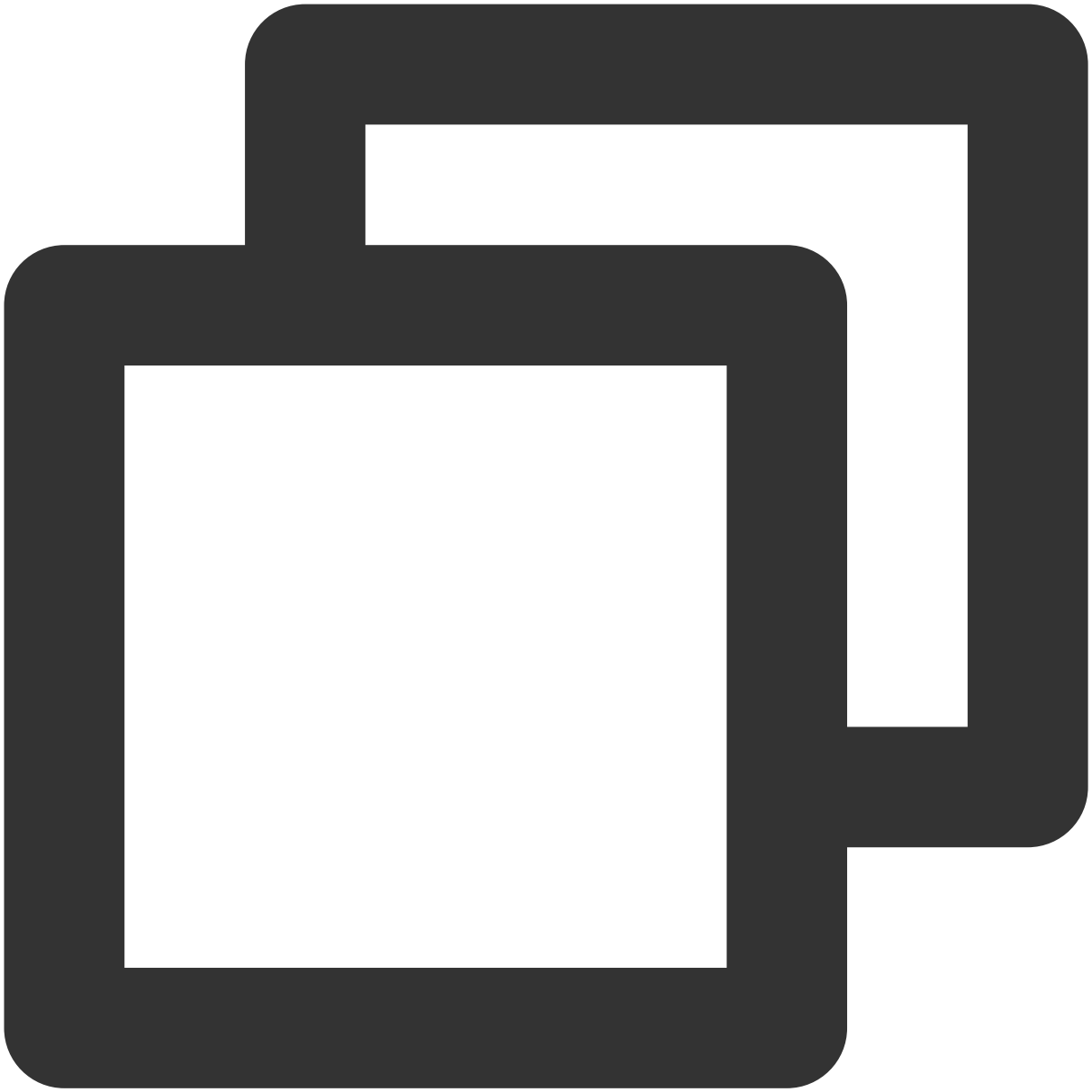
实际案例

请求



```
GET /?encryption HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: signatureValue
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: xxxx
Date: Mon, 17 Jun 2019 08:37:36 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****



<?xml version = "1.0" encoding = "UTF-8">
<ServerSideEncryptionConfiguration>
```

```
        <Rule>
              <ApplyServerSideEncryptionByDefault>
                  <SSEAlgorithm>AES256</SSEAlgorithm>
              </ApplyServerSideEncryptionByDefault>
        </Rule>
</ServerSideEncryptionConfiguration>
```

DELETE Bucket encryption

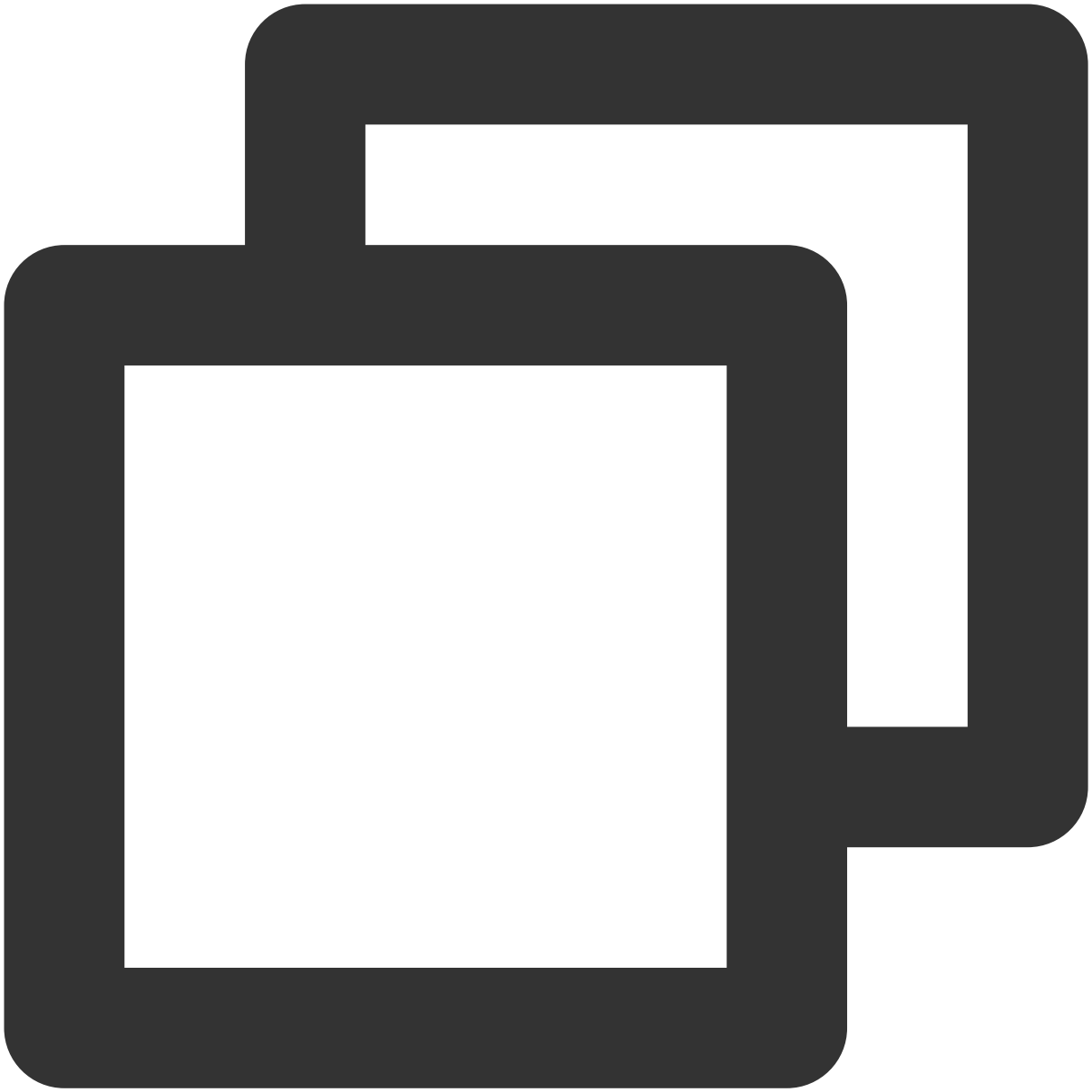
最近更新时间：2023-12-21 12:45:29

功能描述

DELETE Bucket encryption 接口用于删除指定存储桶下的默认加密配置。
要执行此接口，必须拥有 DeleteBucketEncryption 权限。默认情况下，Bucket 的持有者直接拥有权限使用该 API 接口，Bucket 持有者也可以将权限授予其他用户。

请求

请求示例



```
DELETE /?encryption HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求的响应体返回为空。

错误码

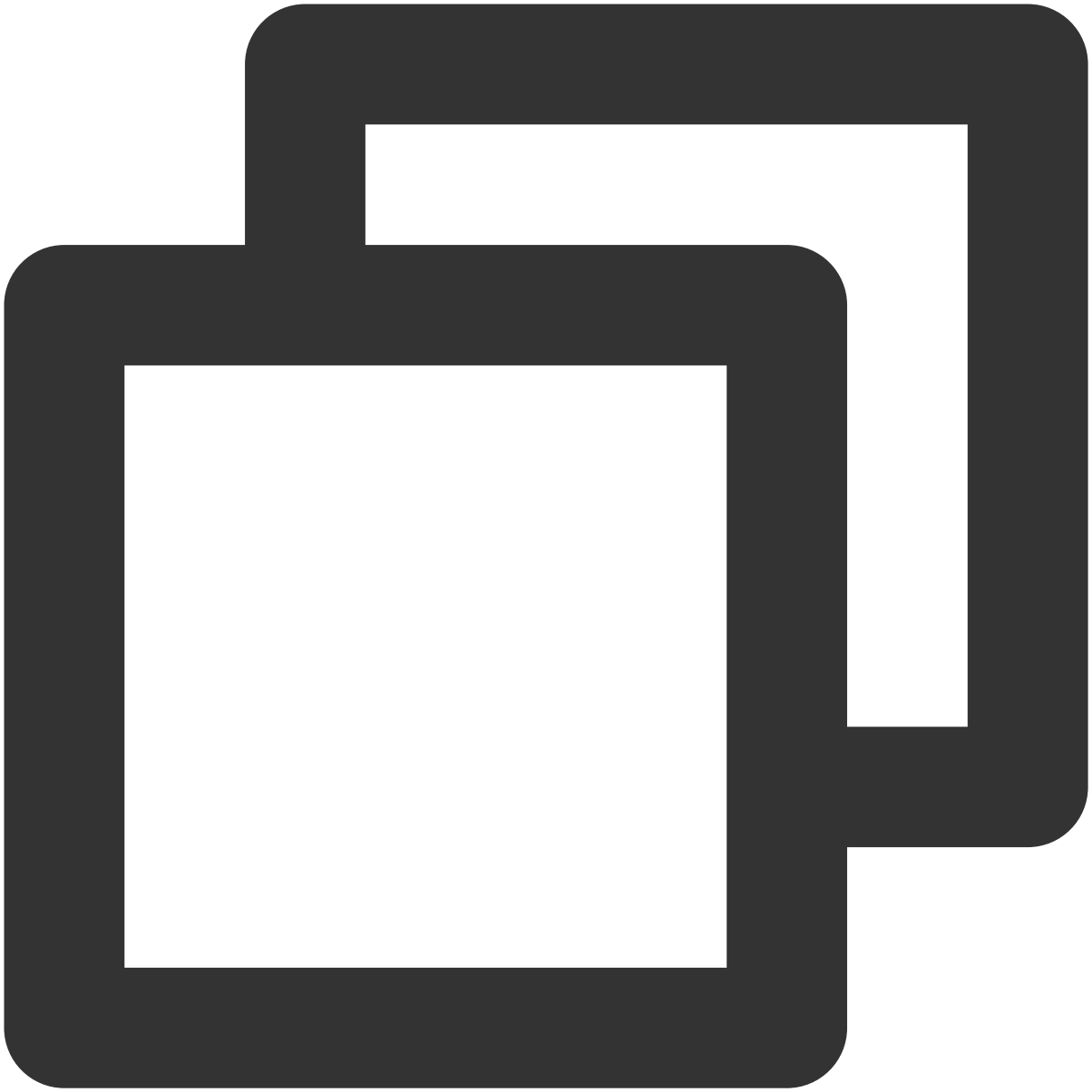
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

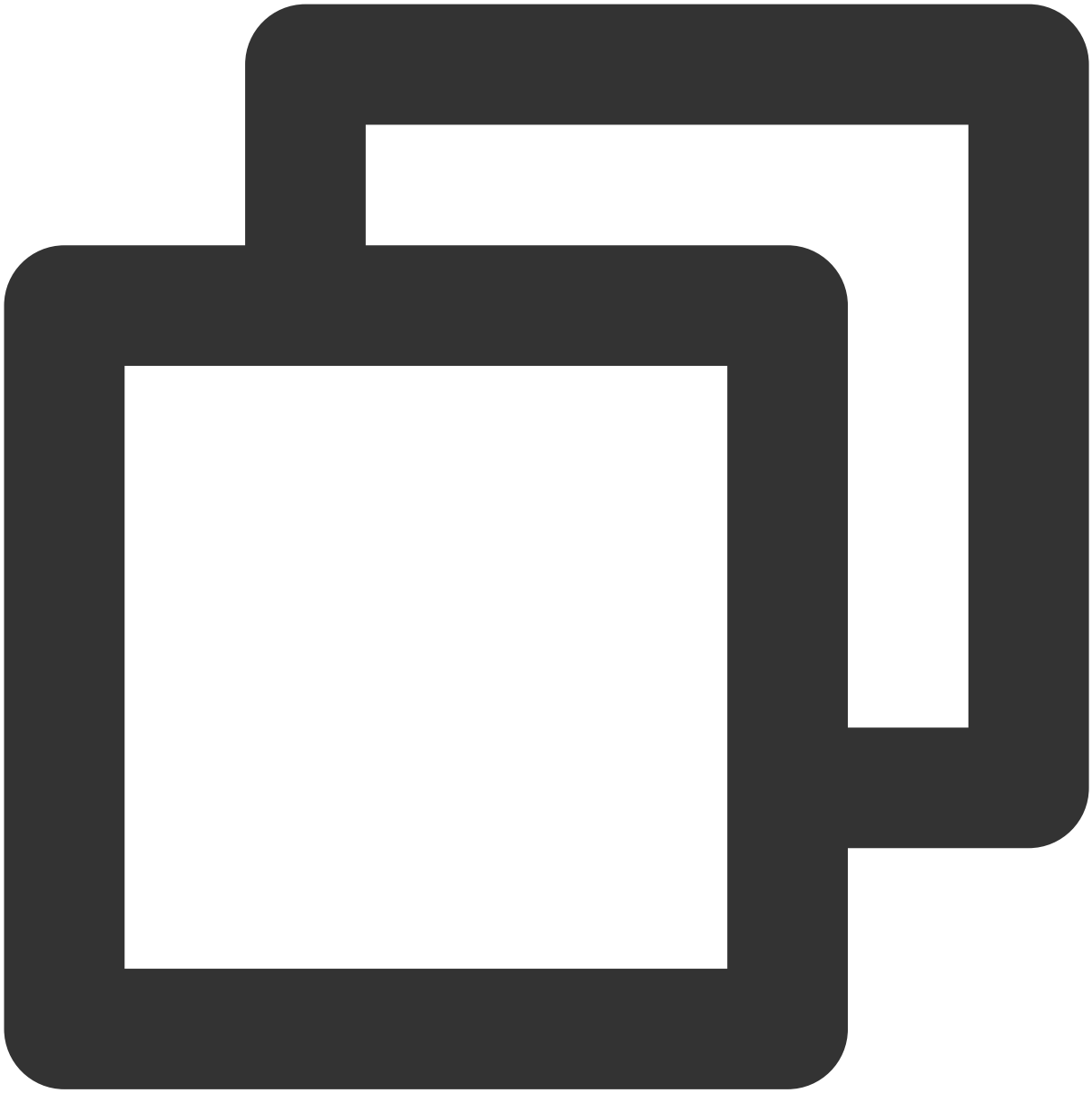
以下示例表示从存储桶 examplebucket-1250000000 中删除默认 SSE-COS 加密配置。

版权所有：腾讯云计算（北京）有限责任公司



```
DELETE /?encryption HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: signatureValue
```

响应



```
HTTP/1.1 204 No Content
Server: tencent-cos
Date: Mon, 17 Jun 2019 08:37:36 GMT
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****
```

自定义域名（domain）

PUT Bucket domain

最近更新时间：2023-12-21 12:44:38

功能描述

PUT Bucket domain 请求用于设置存储桶的自定义域名。

注意

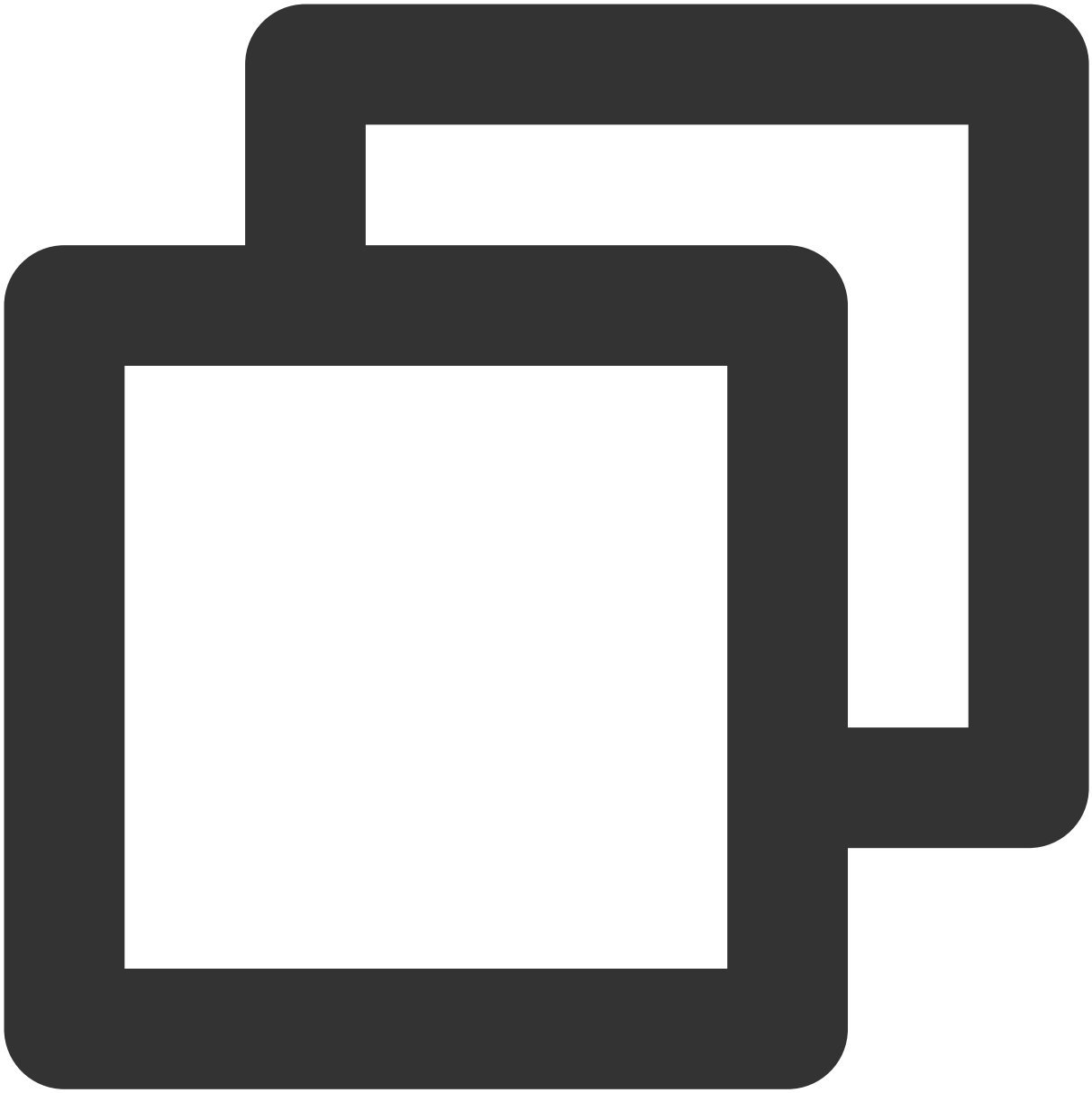
目前最多支持用户添加20条自定义域名，如需添加更多自定义域名，请联系 在线客服。

自定义域名支持默认源站、静态网站源站、全球加速源站三种源站类型，如果需要使用静态网站源站，需要 开启静态网站；如果需要使用全球加速源站，需要 开启全球加速。

主账号默认拥有添加存储桶域名的权限，子账号如需添加存储桶自定义域名，需要主账号在 访问管理控制台 授予 `PutBucketDomain` 接口的权限。

请求

请求示例



```
PUT /?domain HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

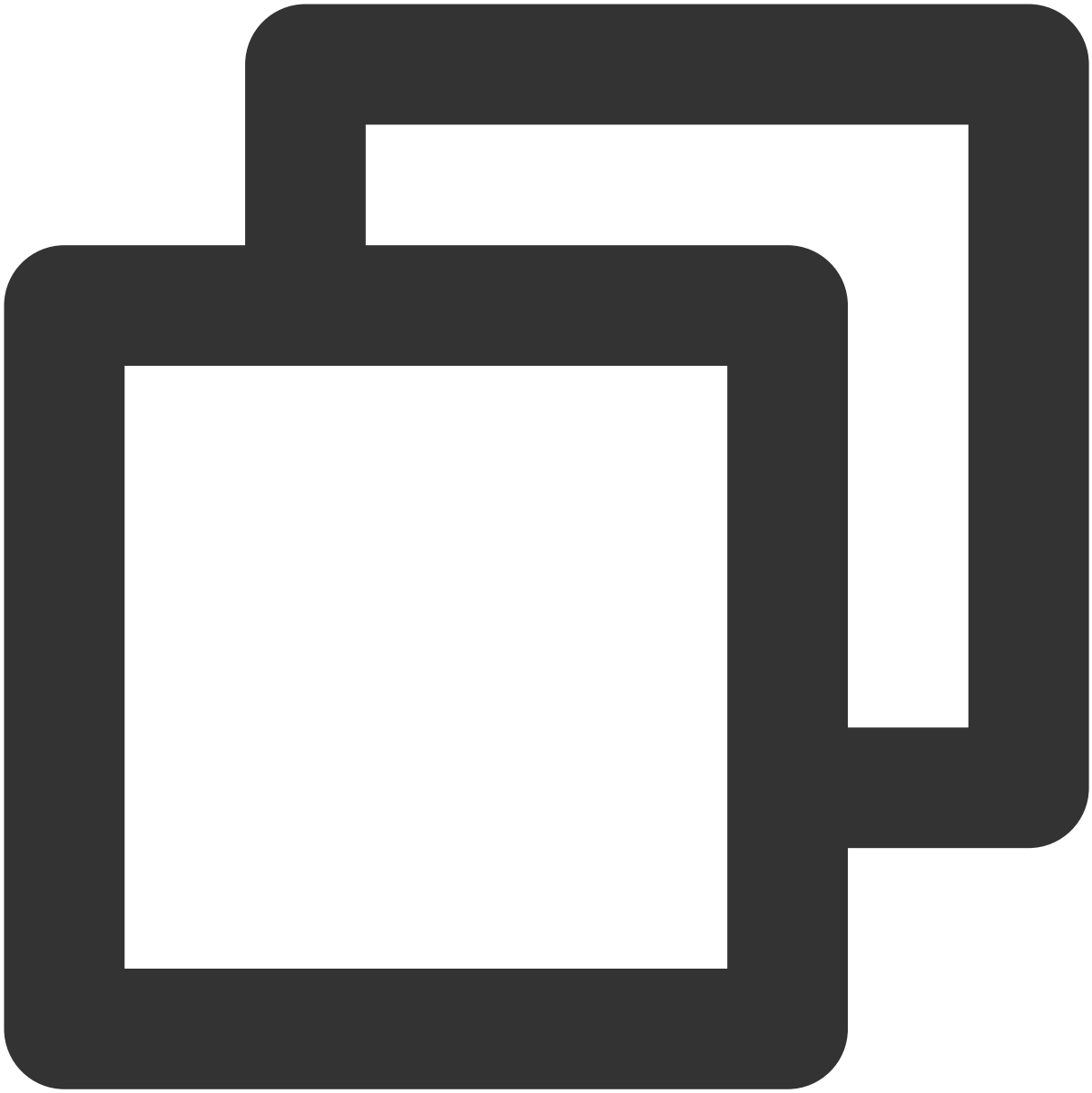
此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

提交 **application/xml** 请求数据，包含完整的存储桶自定义域名信息。



```
<DomainConfiguration>
    <DomainRule>
        <Status>Enum</Status>
        <Name>string</Name>
        <Type>Enum</Type>
    </DomainRule>
    <DomainRule>
        <Status>Enum</Status>
        <Name>string</Name>
        <Type>Enum</Type>
    </DomainRule>
```

```
</DomainConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
DomainConfiguration	无	包含 PUT Bucket domain 操作的所有请求信息	Container	否

Container 节点 DomainConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
DomainRule	DomainConfiguration	域名条目	Container	是

Container 节点 DomainRule 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Status	DomainConfiguration.DomainRule	是否启用。枚举值： ENABLED：启用 DISABLED：禁用	Enum	是
Name	DomainConfiguration.DomainRule	完整域名	string	是
Type	DomainConfiguration.DomainRule	源站类型。枚举值： REST：默认源站 WEBSITE：静态源站源站 ACCELERATE：全球加速源站	Enum	是
ForcedReplacement	DomainConfiguration.DomainRule	如果指定域名已经作为其他存储桶的自定义域名，那么可以指定该元素强制将该域名作为当前存储桶的自定义域名。当前只支持 CNAME，代表您需要先将该域名的 CNAME 指向当前存储桶的源站域名（根据 Type 元素的不同对应为默认源站、静态网站源站或全球加速源站）后才能通过该接口设置自定义域名。	Enum	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

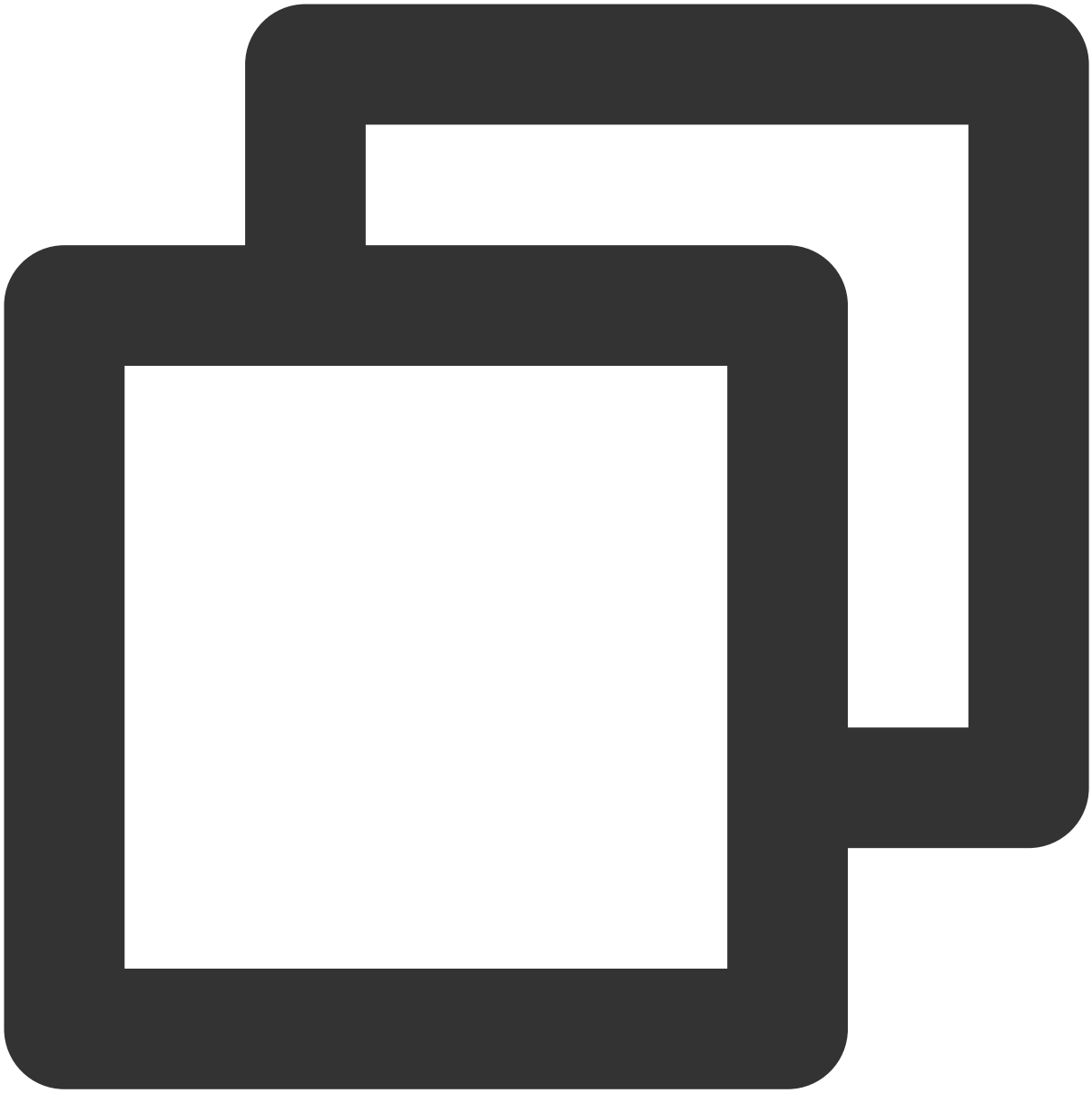
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

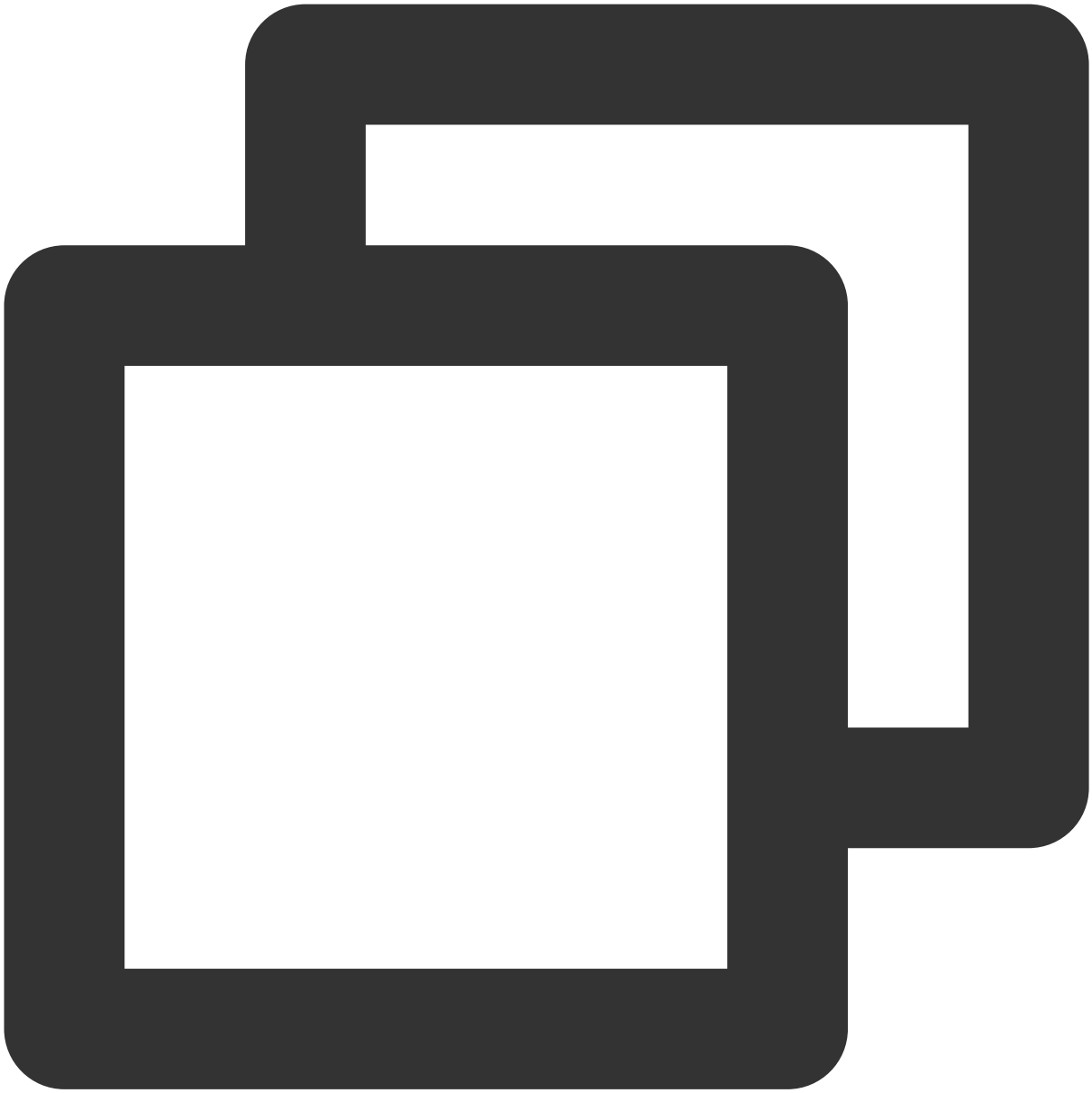
请求



```
PUT /?domain HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Apr 2020 09:16:14 GMT
Content-Type: application/xml
Content-Length: 288
Content-MD5: WHjVNjOz7lW82VThLKf4Lg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<DomainConfiguration>
    <DomainRule>
```

```
        <Status>ENABLED</Status>
        <Name>cos.cloud.tencent.com</Name>
        <Type>REST</Type>
    </DomainRule>
    <DomainRule>
        <Status>ENABLED</Status>
        <Name>www.cos.cloud.tencent.com</Name>
        <Type>WEBSITE</Type>
    </DomainRule>
</DomainConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Wed, 29 Apr 2020 09:16:15 GMT
Server: tencent-cos
x-cos-request-id: NWVhOTQ1ZGVfZTViOTJhMDlfMzA0MjJfMTEwMmJi****
```

GET Bucket domain

最近更新时间：2023-12-21 12:45:14

功能描述

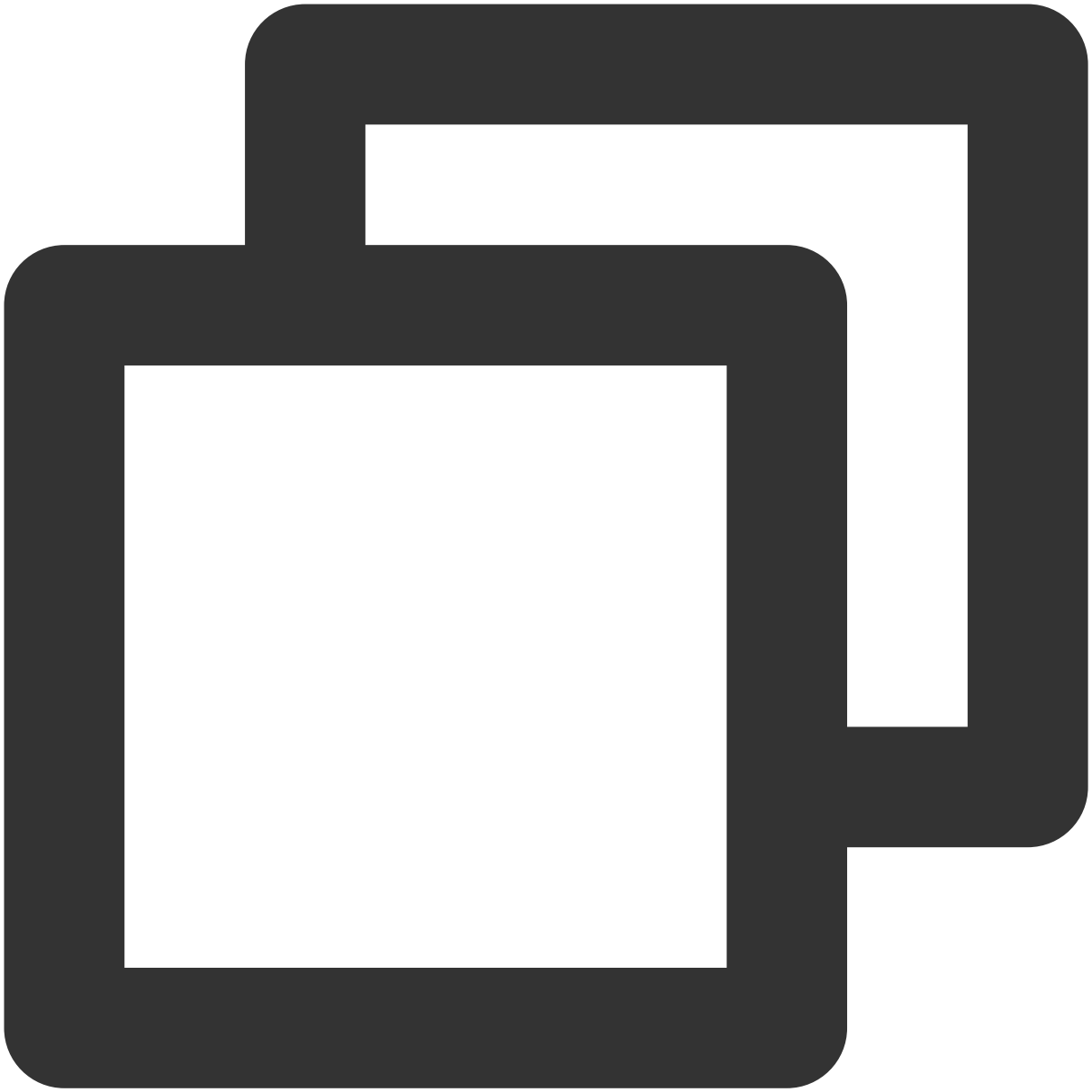
GET Bucket domain 请求用于查询存储桶的自定义域名配置。

注意

主账号默认拥有查询存储桶自定义域名的权限，子账号如需查询存储桶自定义域名，需要通过主账号在 访问管理控制台 授予 `GetBucketDomain` 接口的权限。

请求

请求示例



```
GET /?domain HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

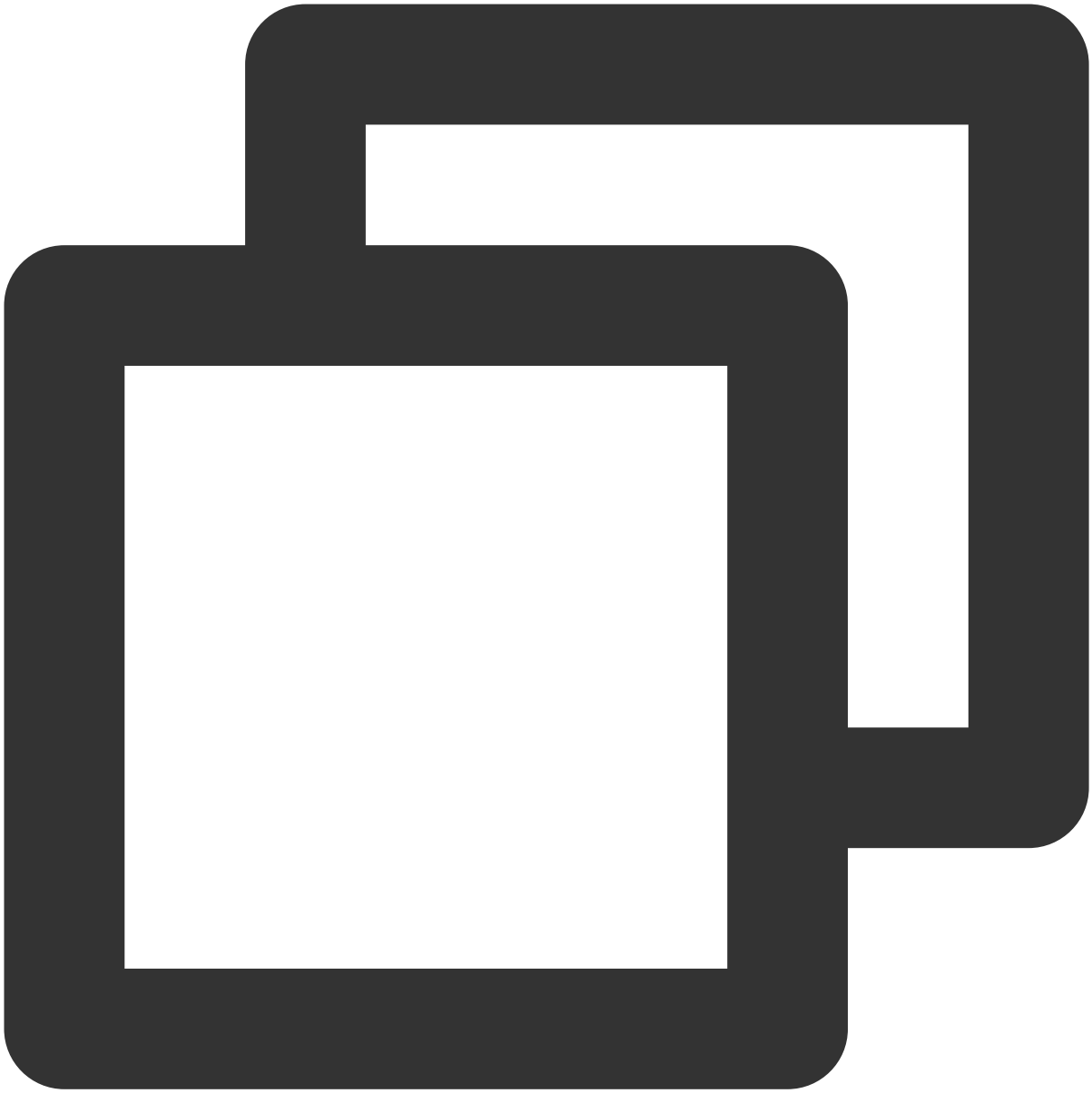
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含完整的存储桶自定义域名信息。



```
<DomainConfiguration>
    <DomainRule>
        <Status>Enum</Status>
        <Name>string</Name>
        <Type>Enum</Type>
    </DomainRule>
    <DomainRule>
        <Status>Enum</Status>
        <Name>string</Name>
        <Type>Enum</Type>
    </DomainRule>
```

```
</DomainConfiguration>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
DomainConfiguration	无	保存 GET Bucket domain 结果的所有信息	Container

Container 节点 DomainConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型
DomainRule	DomainConfiguration	域名条目	Container

Container 节点 DomainRule 的内容：

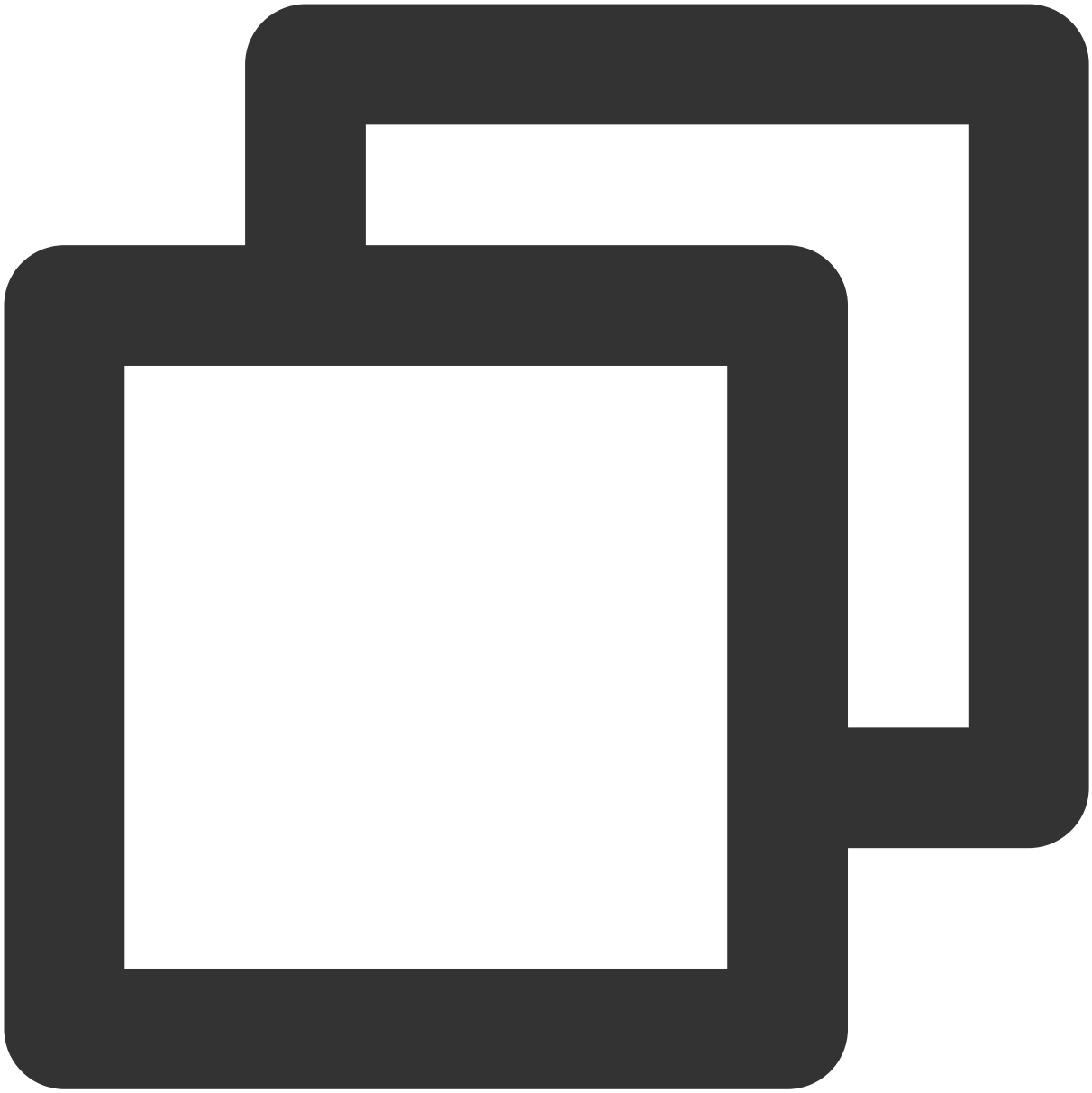
节点名称（关键字）	父节点	描述	类型
Status	DomainConfiguration.DomainRule	是否启用。枚举值： ENABLED：启用 DISABLED：禁用	Enum
Name	DomainConfiguration.DomainRule	完整域名	string
Type	DomainConfiguration.DomainRule	源站类型。枚举值： REST：默认源站 WEBSITE：静态源站源站 ACCELERATE：全球加速源站	Enum

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

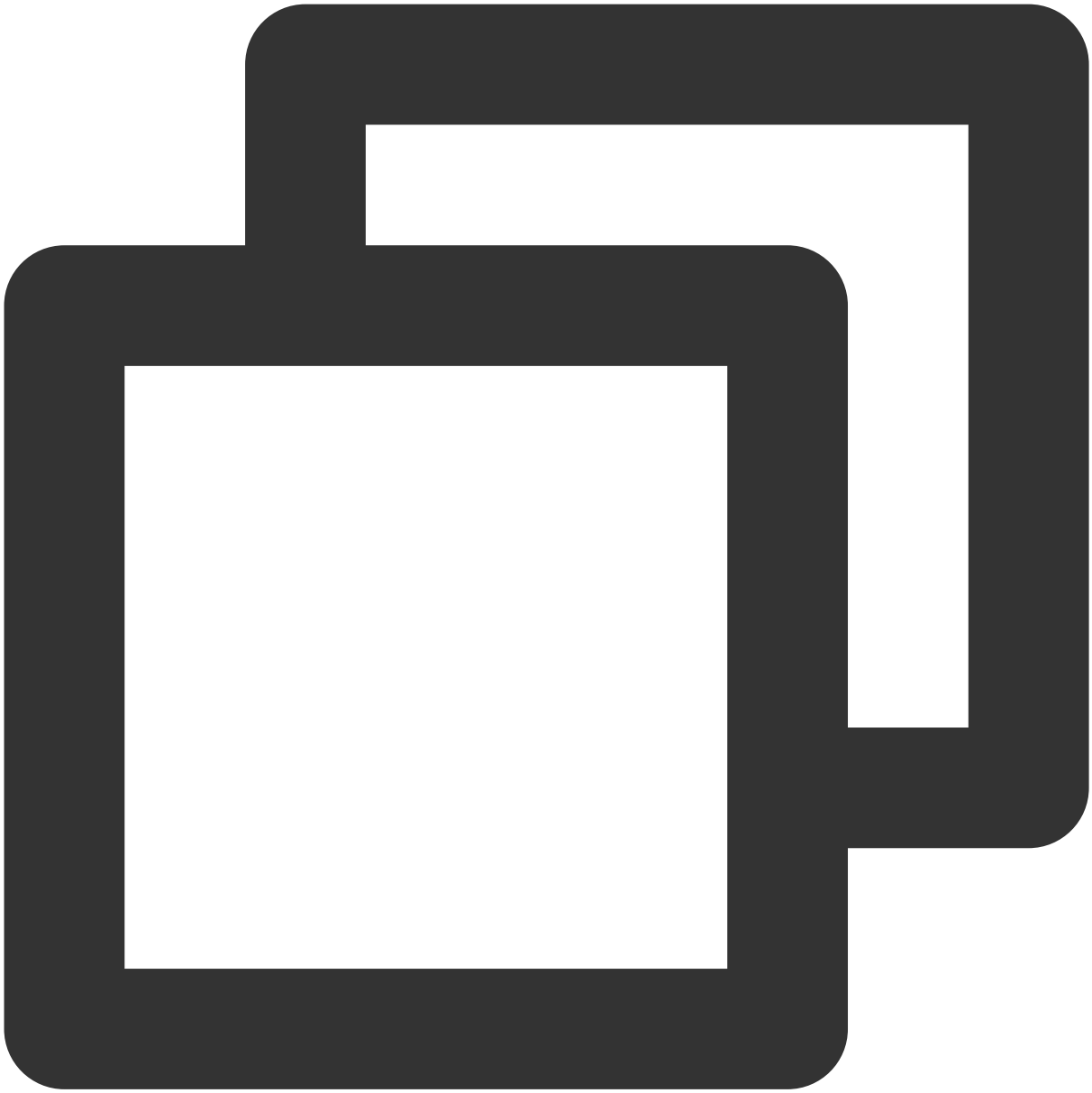
实际案例

请求



```
GET /?domain HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Apr 2020 09:16:26 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 277
Connection: close
Date: Wed, 29 Apr 2020 09:16:26 GMT
Server: tencent-cos
x-cos-domain-txt-verification: tencent-cloud-cos-verification=673029e5ff8e4d2b5723d
x-cos-request-id: NWVhOTQ1ZWFfN2ViMTJhMDlfMjU5Zl8xMzQ1****

<DomainConfiguration>
    <DomainRule>
```

```
        <Status>ENABLED</Status>
        <Name>cos.cloud.tencent.com</Name>
        <Type>REST</Type>
    </DomainRule>
    <DomainRule>
        <Status>ENABLED</Status>
        <Name>www.cos.cloud.tencent.com</Name>
        <Type>WEBSITE</Type>
    </DomainRule>
</DomainConfiguration>
```

DELETE Bucket domain

最近更新时间：2023-12-21 12:45:37

功能描述

DELETE Bucket domain 请求用于删除存储桶的自定义域名配置。

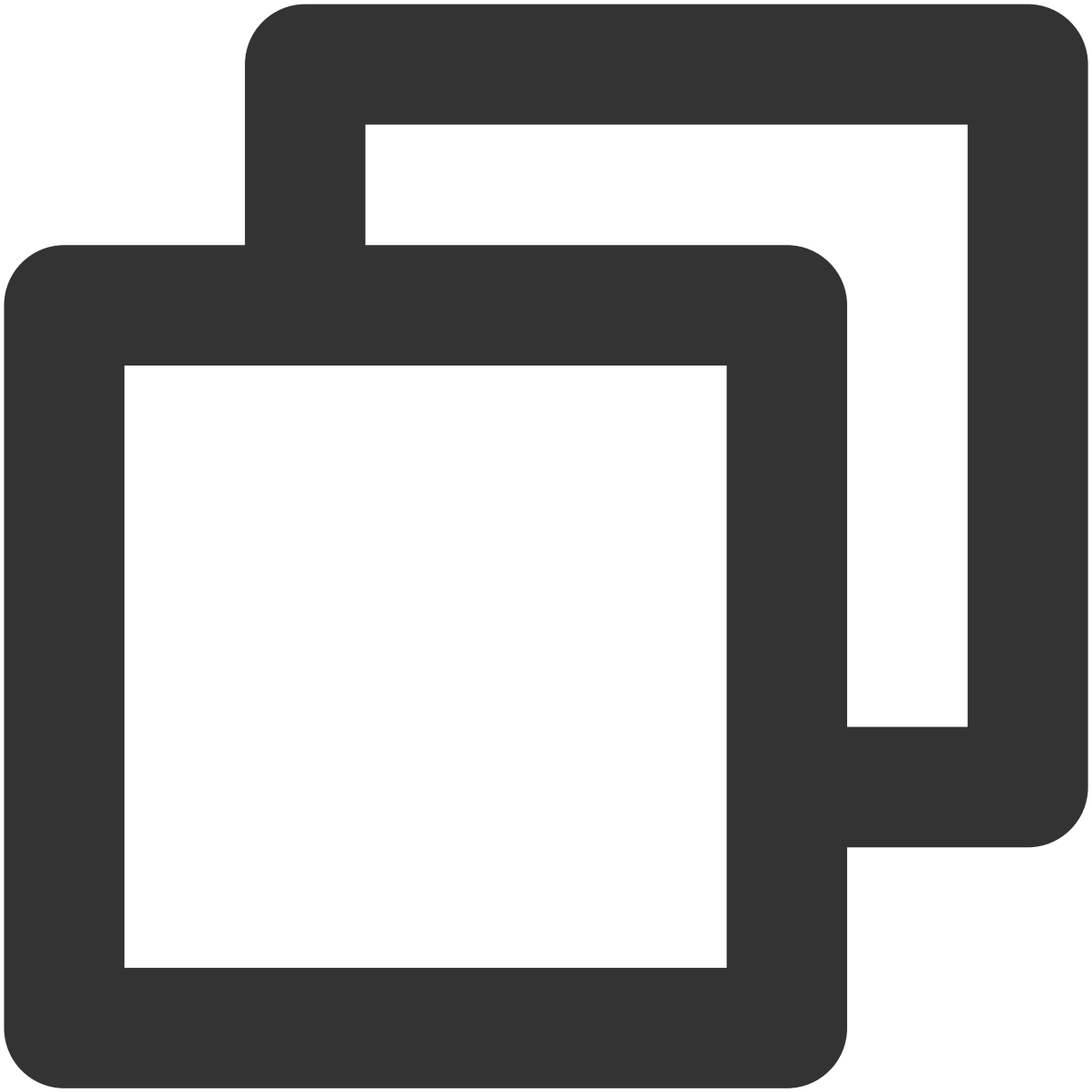
注意

主账号默认拥有删除存储桶域名的权限，子账号删除存储桶自定义域名，需要通过主账号在 访问管理控制台 授予 `DeleteBucketDomain` 接口的权限。

该接口将删除指定存储桶下所有绑定的自定义域名，请谨慎操作；如果需要删除部分自定义域名，可以通过 `GET Bucket Domain` 接口拉取存储桶下绑定的所有自定义域名，然后通过 `PUT Bucket Domain` 接口将不需要删除的自定义域名重新写入即可。

请求

请求示例



```
DELETE /?domain HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

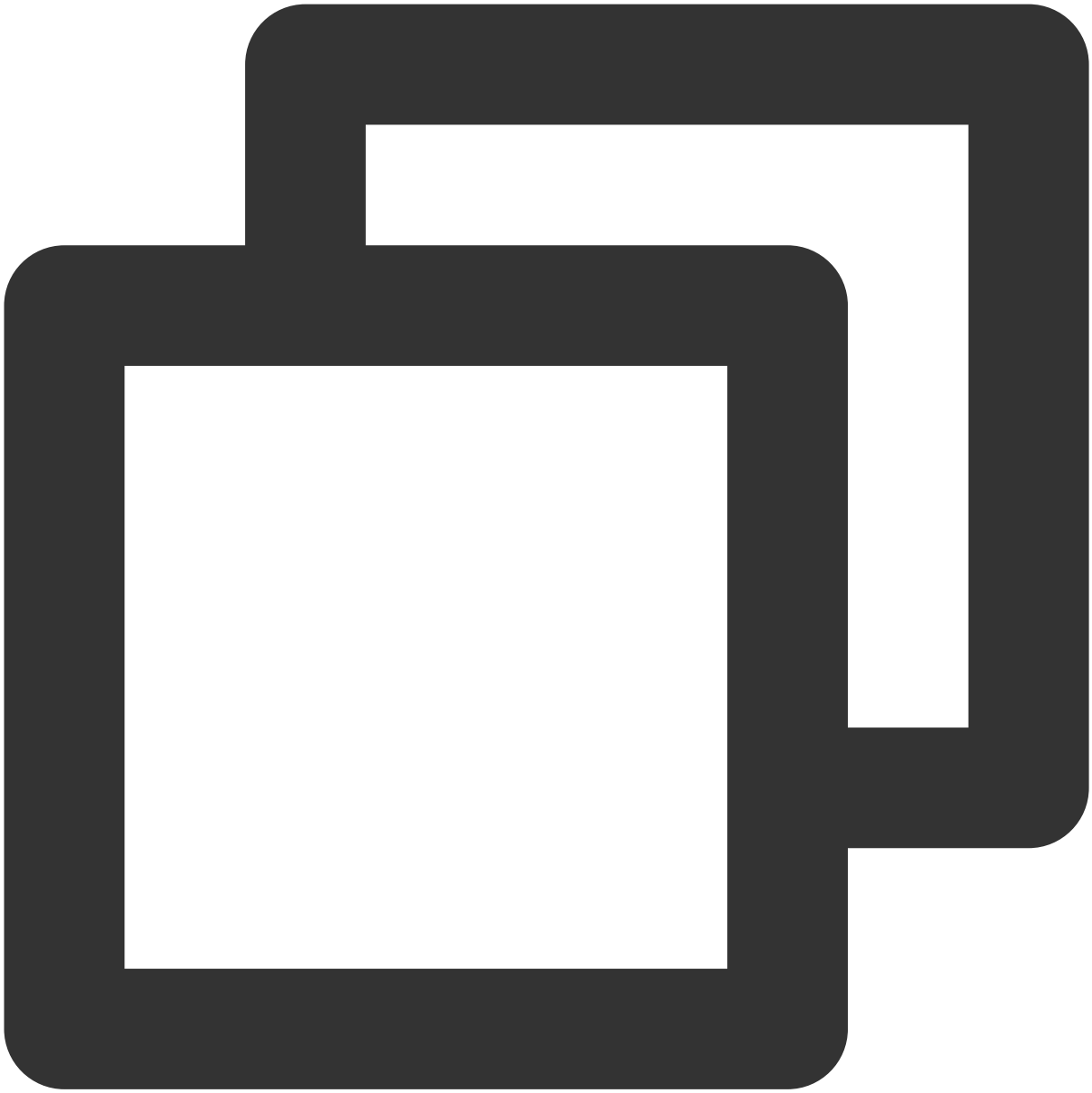
此接口响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

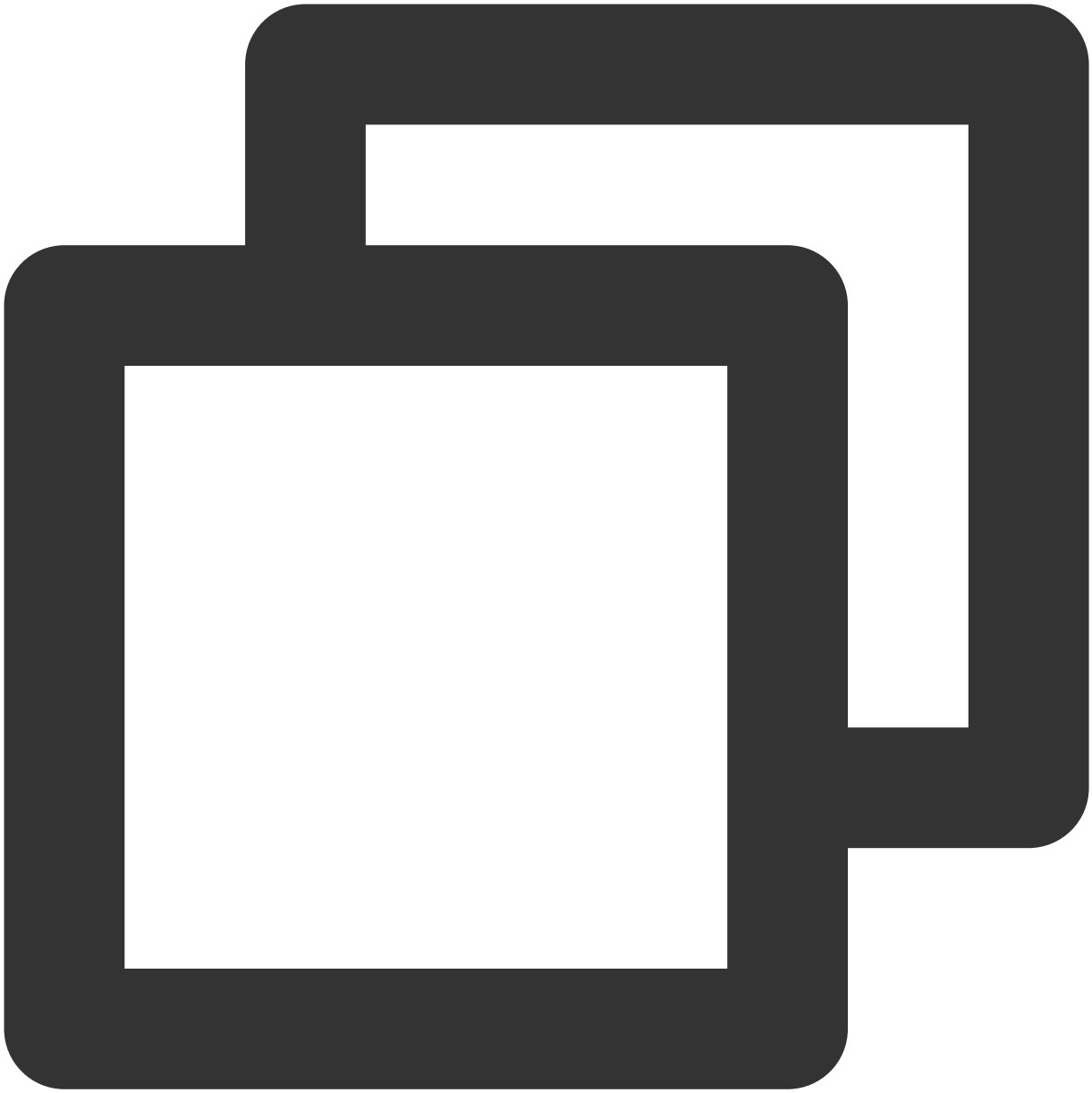
实际案例

请求



```
DELETE /?domain HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Apr 2020 09:10:59 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 204 No Content
Connection: close
Date: Wed, 29 Apr 2020 09:11:00 GMT
Server: tencent-cos
x-cos-request-id: NWVhOTQ0YTNfYWZjOTJhMDlfMWNhYTBfOTEw****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODczNTBmNjMwZmQ0MTZkMjg0Njl
Content-Length: 0
```

回源（Origin）

GET Bucket Origin

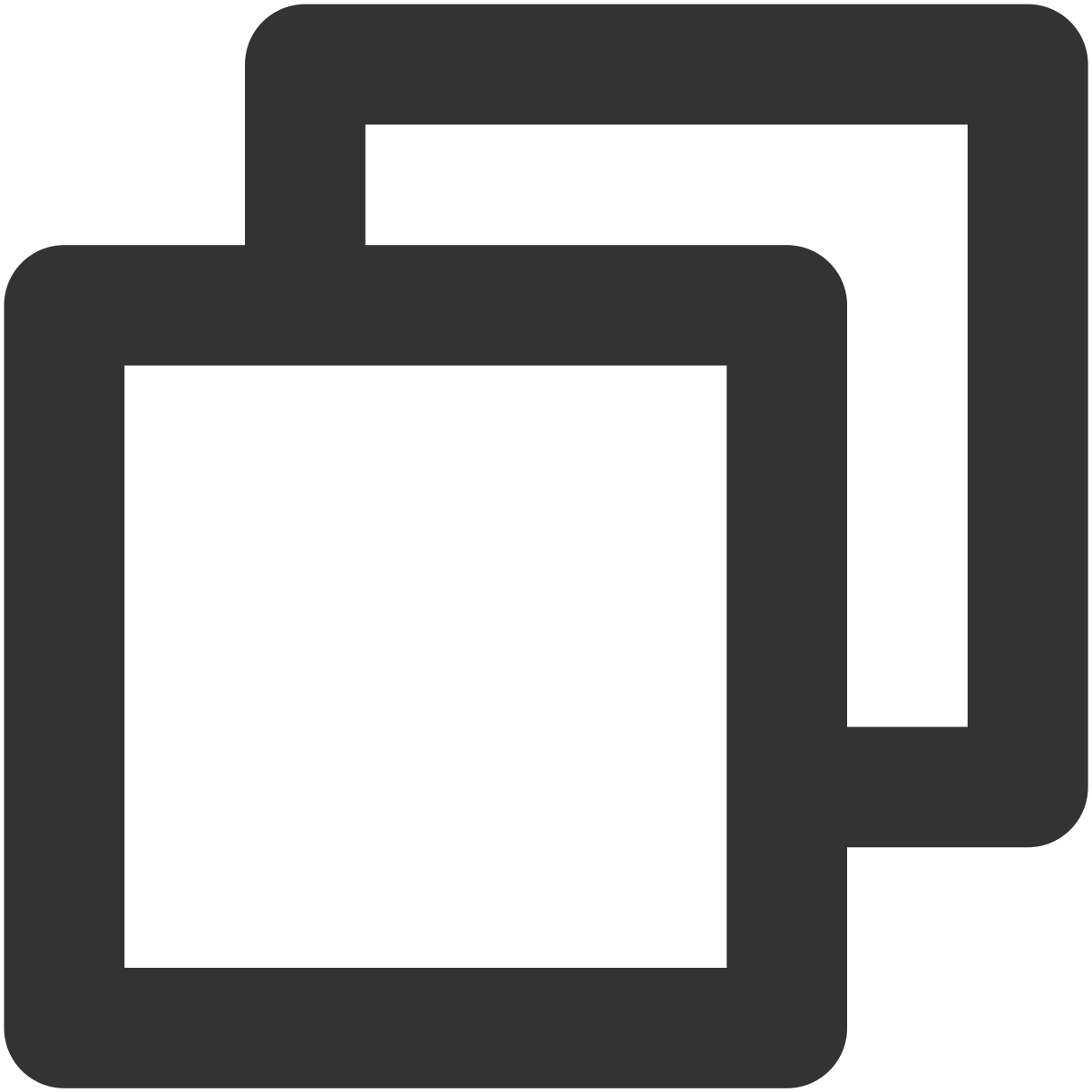
最近更新时间：2023-12-21 12:58:51

功能描述

GET Bucket Origin 接口用于查询存储桶的回源配置。

请求

请求示例



```
GET /?origin HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

该请求需结合请求体一起使用。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

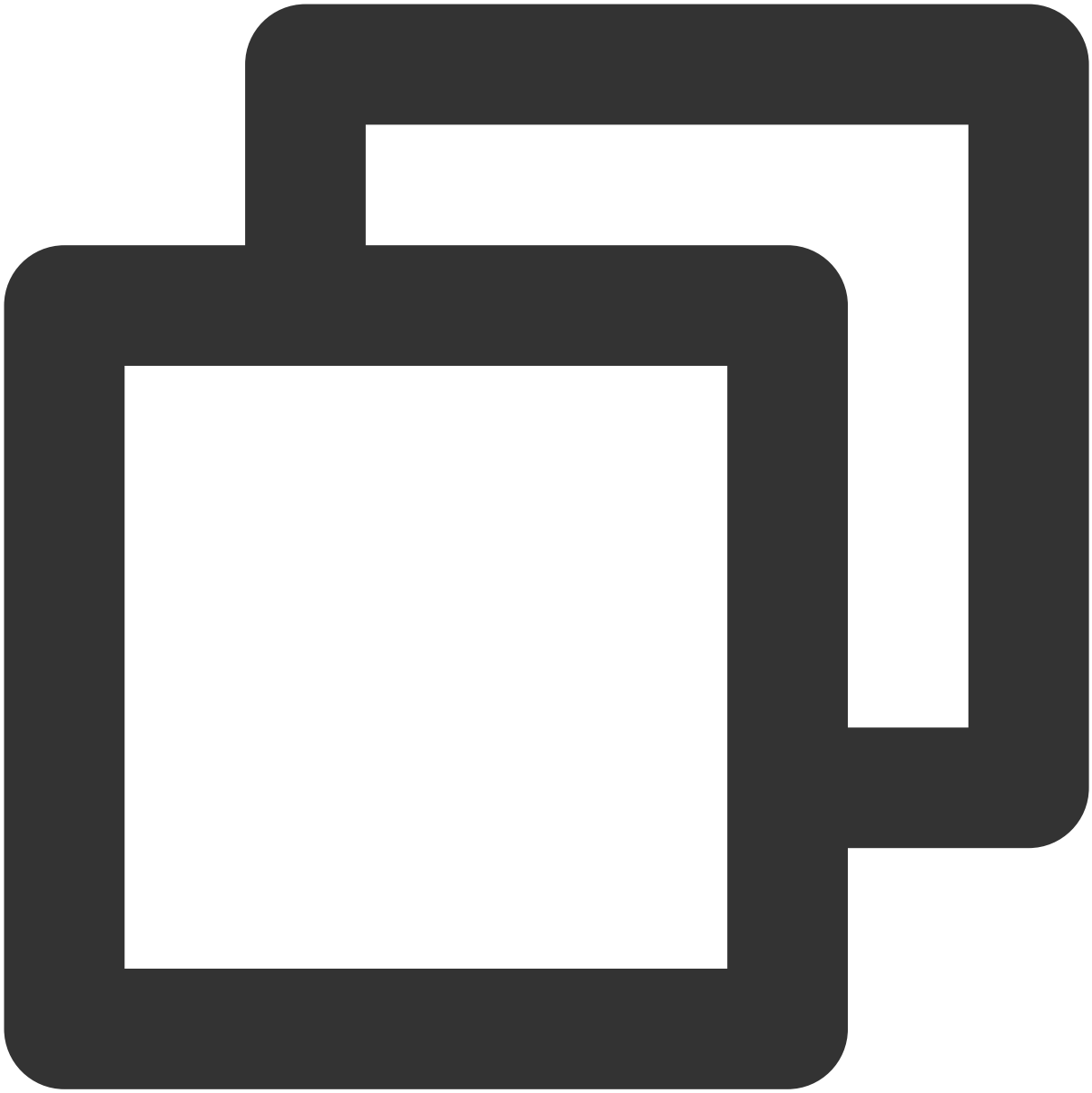
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

响应体为回源规则，结构如下。



```
<OriginConfiguration>
    <OriginRule>
        <RulePriority>Integer</RulePriority>
        <OriginType>Redirect|Proxy|Mirror</OriginType>
        <OriginCondition>
            <!--回源设置的 HTTP 状态码设置根据回源类型决定，如果是 Proxy、Mirror 是404，如果
            <HTTPStatusCode>404</HTTPStatusCode>
            <Prefix></Prefix>
        </OriginCondition>
        <OriginParameter>
            <Protocol>HTTP|HTTPS|FOLLOW</Protocol>
```

```
            <!--回源是否保留源站的错误码-->
            <TransparentErrorCode>true|false</TransparentErrorCode>
            <!--回源是否保留原始查询参数，支持保留、不保留两种-->
            <FollowQueryString>true|false</FollowQueryString>
            <!--回源的请求头部参数-->
            <HttpHeader>
                <!--回源是否传输全部的请求头部，支持传递、不传递两种-->
                <FollowAllHeaders>true|false</FollowAllHeaders>
                <!--回源新增指定头部，最多10个，自定义头部中一对键值算一个-->
                <!--如果有值则新增头部，如果为空则不新增-->
                <NewHttpHeaders>
                    <Header>
                        <Key>x-cos|oss|amz-ContentType|CacheControl|ContentDisposit
                        <Value>string</Value>
                    </Header>
                </NewHttpHeaders>
                <!--回源透传原始请求的指定头部-->
                <FollowHttpHeaders>
                    <Header>
                        <Key>x-cos|oss|amz-ContentType|CacheControl|ContentDisposit
                    </Header>
                </FollowHttpHeaders>
                <!--回源不透传原始请求的指定头部-->
                <ForbidFollowHeaders>
                    <Header>
                        <Key>String</Key>
                    </Header>
                </ForbidFollowHeaders>
            </HttpHeader>
            <!--follow3xx 参数-->
            <FollowRedirection>true|false</FollowRedirection>
            <!--重定向返回码参数只有在回源类型为 Redirect、Proxy 时可选，否则报参数错误-->
            <HttpRedirectCode>301|302|307</HttpRedirectCode>
        </OriginParameter>
    <!--切备错误码，默认5XX 切换备站，添加此选项后支持4XX 切换备站  -->
        <HTTPStandbyCode>
            <StatusCode>404</StatusCode>
            <StatusCode>403</StatusCode>
        </HTTPStandbyCode>
        <OriginInfo>
            <!--Mirror 模式支持设置多个源站，按比例回源，分担单个源站的回源流量，最多支持填入10
            <HostInfo>
                <HostName>bucketname-appid.cos.region.myqcloud.com</HostName>
                <!--源站权重，按比例回源-->
        <Weight>4</Weight>
                <!--备份回源地址，最多支持填入10条备份回源地址，节点命名依次按照1-10编号-->
                <StandbyHostName_1>bucketname2-appid.cos.region.myqcloud.com</Stand
```

```
                <StandbyHostName_2>bucketname3-appid.cos.region.myqcloud.com</Stand
            </HostInfo>
            <HostInfo>
                <HostName>bucketname4-appid.cos.region.myqcloud.com</HostName>
                <!--源站权重，按比例回源-->
                <Weight>2</Weight>
                <!--备份回源地址，最多支持填入10条备份回源地址，节点命名依次按照1-10编号-->
                <StandbyHostName_1>bucketname5-appid.cos.region.myqcloud.com</Stand
                <StandbyHostName_2>bucketname6-appid.cos.region.myqcloud.com</Stand
            </HostInfo>
            <FileInfo>
                <!--回源为指定的文件-->
                <FixedFileConfiguration>
                    <FixedFilePath>String</FixedFilePath>
                </FixedFileConfiguration>
                <!--回源文件的新增文件前缀-->
                <PrefixConfiguration>
                    <Prefix></Prefix>
                </PrefixConfiguration>
                <!--回源文件的新增文件后缀-->
                <SuffixConfiguration>
                    <Suffix></Suffix>
                </SuffixConfiguration>
            </FileInfo>
        </OriginInfo>
    </OriginRule>
</OriginConfiguration>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
OriginConfiguration	无	Origin 回源配置	Container

Container 节点 OriginConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型
OriginRule	OriginConfiguration	支持多条 OriginRule，通过优先级区分规则执行先后	Container

Container 节点 OriginRule 的内容：

节点名称（关键字）	父节点	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

RulePriority	OriginConfiguration.OriginRule	通过优先级区分规则执行先后	Integer
OriginType	OriginConfiguration.OriginRule	回源类型，支持异步回源（Proxy）、同步回源（Mirror）和重定向回源（Redirect）三种模式。 枚举值：`Proxy` 、 `Mirror` 、 `Redirect`	String
OriginCondition	OriginConfiguration.OriginRule	回源配置，配置用户使用的 HTTP 传输协议等信息	Container
OriginParameter	OriginConfiguration.OriginRule	回源地址相关信息	Container
OriginInfo	OriginConfiguration.OriginRule	源站信息。例如源站域名或者源站 IP 等信息	Container

Container 节点 OriginCondition 的内容：

节点名称（关键字）	父节点	描述	类型
HTTPStatusCode	OriginConfiguration.OriginRule.OriginCondition	触发回源的 HTTP 状态码，Proxy 和 Mirror 模式支持填写404，Redirect 模式支持填写4XX 和5XX	String
Prefix	OriginConfiguration.OriginRule.OriginCondition	触发回源的文件前缀，默认为空，任意文件均可触发	String

Container 节点 OriginParameter 的内容：

节点名称（关键字）	父节点	描述	类型
Protocol	OriginConfiguration.OriginRule.OriginParameter	回源使用的协议，枚举值为 HTTP（使用 HTTP 协议），HTTPS（使用 HTTPS 协议）、FOLLOW（跟随用户使用的协议），默认值为 FOLLOW。	String
FollowQueryString	OriginConfiguration.OriginRule.OriginParameter	回源是否需要透传 HTTP 请求串，枚举值： `true` 或	Boolean

		`false`，默认为 `true`	
HttpHeader	OriginConfiguration.OriginRule.OriginParameter	是否需要设置 Http 头部传输配置。	Container
FollowRedirection	OriginConfiguration.OriginRule.OriginParameter	源站 `3XX` 响应策略，枚举值 `true` 或 `false`，选择 true 时跟随源站 `3xx` 重定向请求获取到资源，并将资源保存到 COS 上;选择 `false` 时透传 `3XX` 响应，不获取资源），默认为 `true` 。	Boolean
HttpRedirectCode	OriginConfiguration.OriginRule.OriginParameter	仅支持 `Redirect` 和 `Proxy` 模式，设置重定向返回码参数，枚举值 `301` 或 `302` 或 `307`，默认为 `302` 。	String

Container 节点 HttpHeader 的内容：

节点名称（关键字）	父节点	描述	类型
FollowAllHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	是否传输全部的请求头部，枚举值：`true` 或 `false`，默认为 `false` 。	Boolea
NewHttpHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	设置回源新增指定头部，最多10个。	Contair
FollowHttpHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	设置回源透传原始请求	Contair

版权所有：腾讯云计算（北京）有限责任公司

		的指定头部。	
ForbidFollowHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	设置回源不透传的原始请求的指定头部。	Contair

Container 节点 NewHttpHeaders、FollowHttpHeaders、ForbidFollowHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Header	OriginConfiguration.OriginRule.OriginParameter.HttpHeader.NewHttpHeaders	回源到源站时添加或者指定传递的自定义头部，默认为空。	Container

Container 节点 Header 的内容：

节点名称（关键字）	父节点	描述
Key	OriginConfiguration.OriginRule.OriginParameter.HttpHeader.NewHttpHeader.UserMetaData	用户设置为空。开A 支持填 oss 或填入 C Cache Conte 或 Con

		或 `Htt` 或 `Use`
Value	OriginConfiguration.OriginRule.OriginParameter.HttpHeader.NewHttpHeader.UserMetaData	用户设置 空。

Container 节点 OriginInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HostInfo	OriginConfiguration.OriginRule.OriginInfo	源站信息。Mirror 模式支持设置多个源站，按比例回源，分担单个源站的回源流量，最多支持填入10条回源地址，比例按权重分配；Proxy 和 Redirect 模式仅支持一个源站。	Container
FileInfo	OriginConfiguration.OriginRule.OriginInfo	回源文件信息。	Container

Container 节点 HostInfo 的内容：

节点名称（关键字）	父节点	描述
HostName	OriginConfiguration.OriginRule.OriginInfo.HostInfo	源站域名或者源站 IP。
Weight	OriginConfiguration.OriginRule.OriginInfo.HostInfo	源站权重，Mirror 模式下配置了多
StandbyHostName_N	OriginConfiguration.OriginRule.OriginInfo.HostInfo	备份回源地址，最多支持填入10条 `StandbyHostName_1` 、 `Sta`

Container 节点 FileInfo 的内容：

节点名称（关键字）	父节点	描述	类型
PrefixConfiguration	OriginConfiguration.OriginRule.OriginInfo.FileInfo	回源文件的前缀设置。	Container
SuffixConfiguration	OriginConfiguration.OriginRule.OriginInfo.FileInfo	回源文件的后缀设置。	Container
FixedFileConfiguration	OriginConfiguration.OriginRule.OriginInfo.FileInfo	回源到固定的文件。	Container

Container 节点 FixedFileConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型
FixedFilePath	OriginConfiguration.OriginRule.OriginInfo.FileInfo.FixedFileConfiguration	回源的固定文件路径。	String

Container 节点 PrefixConfiguration 的内容：

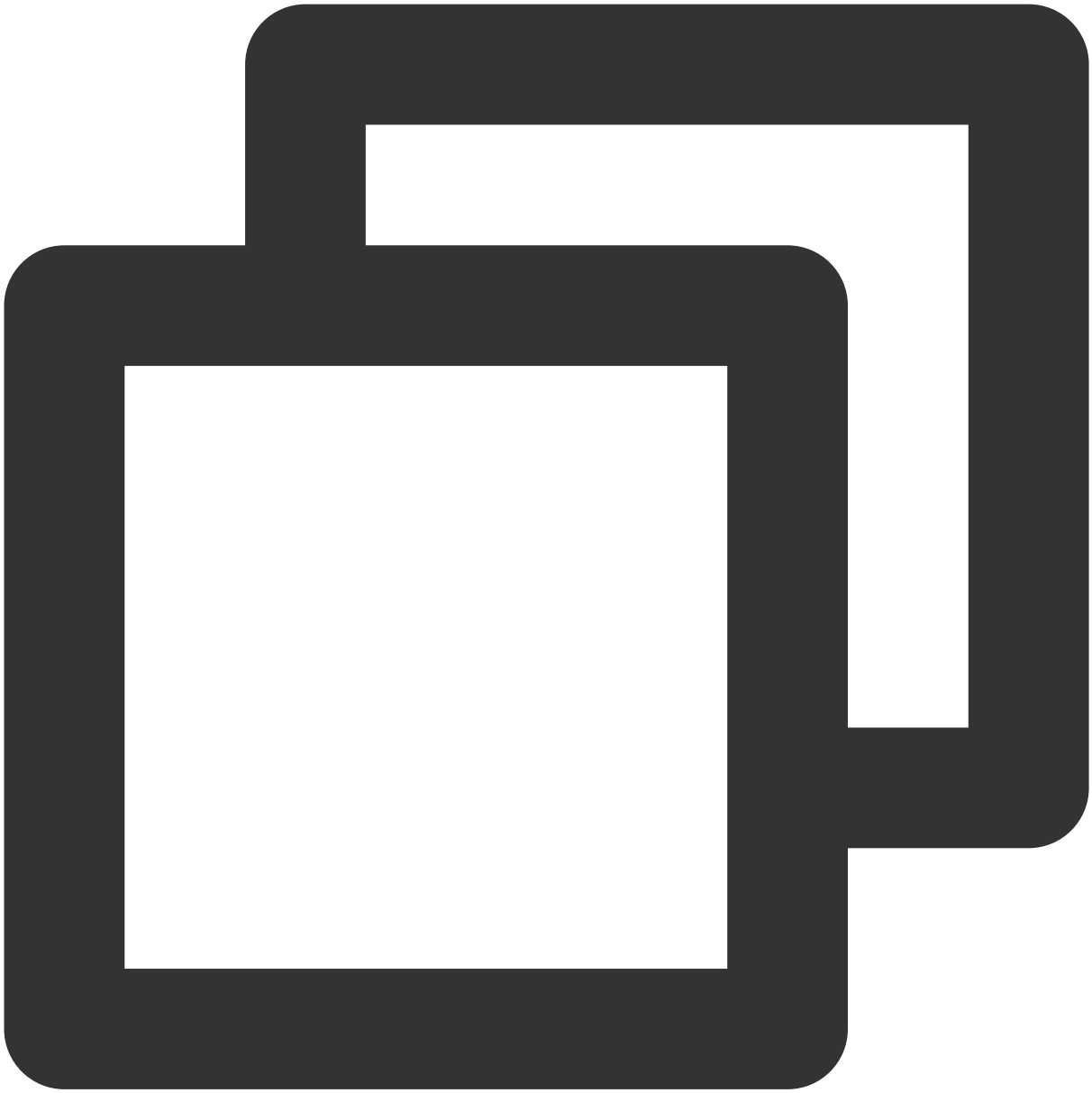
节点名称（关键字）	父节点	描述	类型
Prefix	OriginConfiguration.OriginRule.OriginInfo.FileInfo.PrefixConfiguration	回源文件的新增文件前缀，默认为空。	String

Container 节点 SuffixConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型
Suffix	OriginConfiguration.OriginRule.OriginInfo.FileInfo.SuffixConfiguration	回源文件的新增文件后缀，默认为空。	String

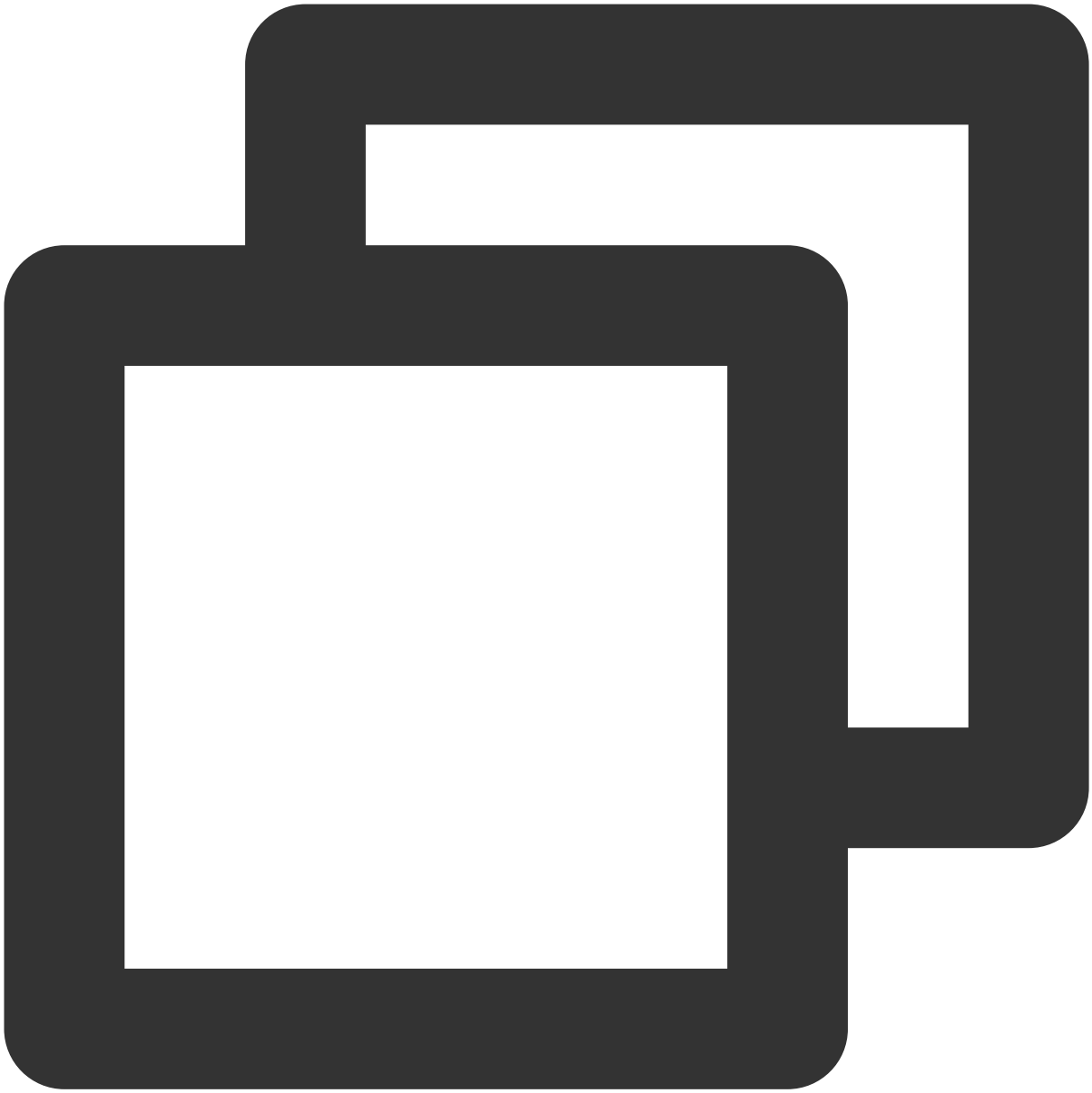
实际案例

请求



```
GET /?origin= HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization: Auth String
Date: Sun, 28 Apr 2019 12:02:24 GMT
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Sun, 28 Apr 2019 12:02:25 GMT
Server: tencent-cos
x-cos-request-id: NWNjNTk2NTFfMmM4OGY3MGFfNTI1****

<?xml version="1.0" encoding="UTF-8"?>
<OriginConfiguration>
    <OriginRule>
        <OriginType>Proxy</OriginType>
```

```
        <OriginCondition>
            <HTTPStatusCode>404</HTTPStatusCode>
            <Prefix></Prefix>
        </OriginCondition>
        <OriginParameter>
            <Protocol>FOLLOW</Protocol>
            <FollowQueryString>true</FollowQueryString>
            <HttpHeader>
                <FollowAllHeaders>true</FollowAllHeaders>
                <NewHttpHeaders>
                    <Header>
                        <Key>x-cos-ContentType</Key>
                        <Value>csv</Value>
                    </Header>
                </NewHttpHeaders>
                <FollowHttpHeaders>
                    <Header>
                        <Key>Content-Type</Key>
                    </Header>
                </FollowHttpHeaders>
            </HttpHeader>
            <FollowRedirection>true</FollowRedirection>
        </OriginParameter>
        <OriginInfo>
            <HostInfo>
                <HostName>examplebucket-1250000000.cos.ap-shanghai.myqcloud.com</Ho
            </HostInfo>
        </OriginInfo>
    </OriginRule>
</OriginConfiguration>
```

PUT Bucket Origin

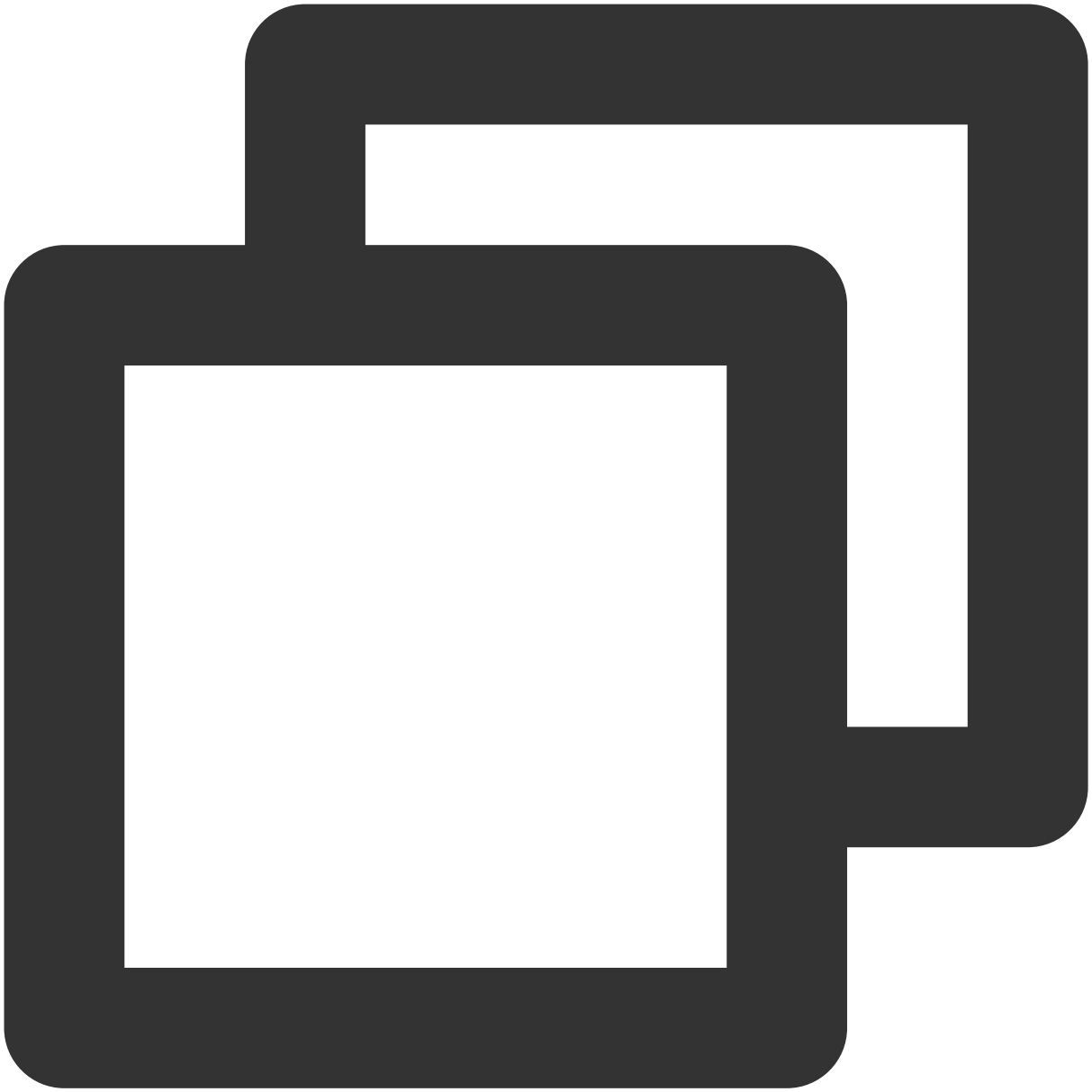
最近更新时间：2023-12-21 12:59:19

功能描述

PUT Bucket Origin 接口用于设置存储桶的回源。

请求

请求示例



```
PUT /?origin HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

该请求需结合请求体一起使用。

请求参数

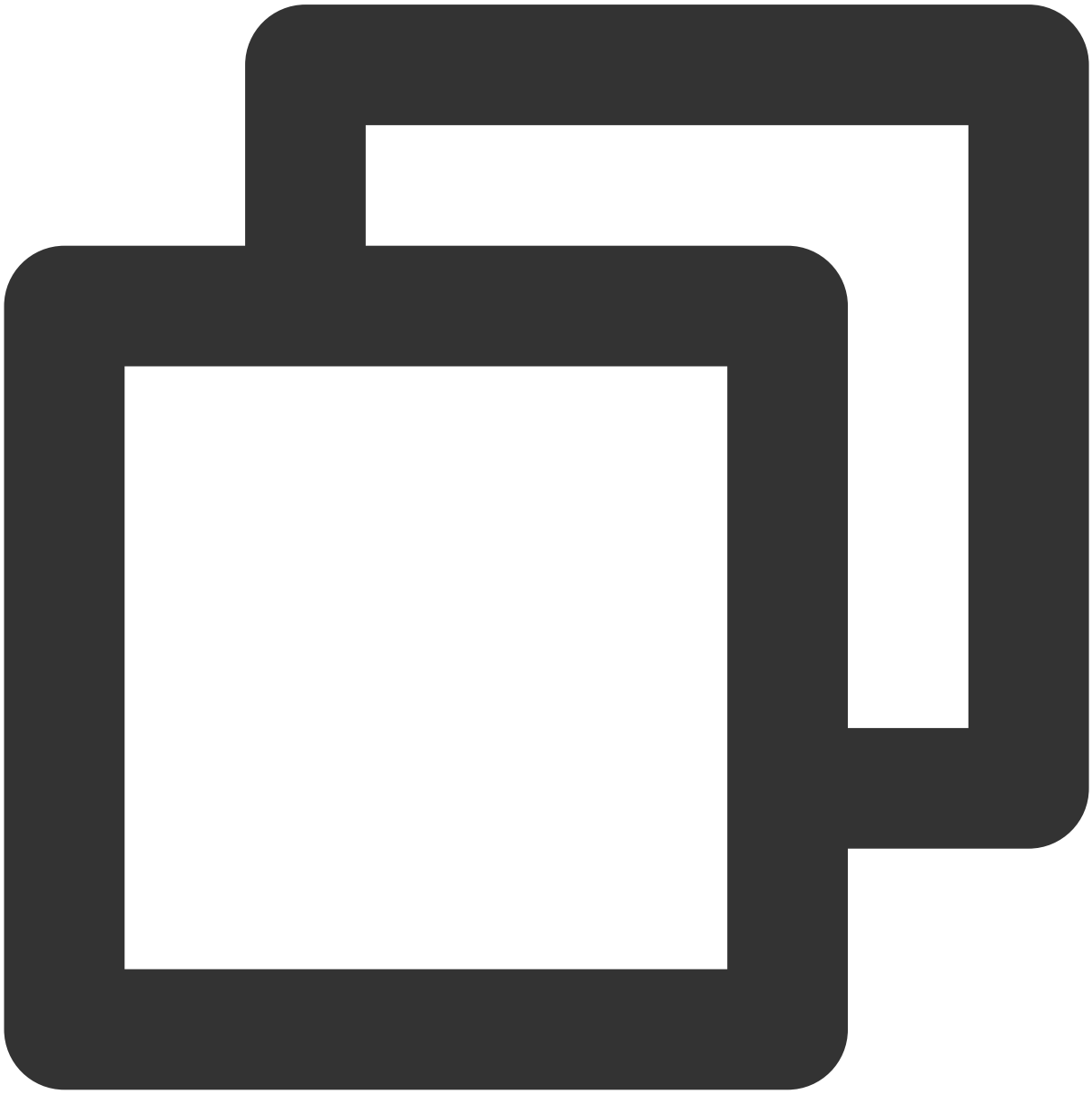
此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

请求的请求体为回源规则。



```
<OriginConfiguration>
    <OriginRule>
        <RulePriority>Integer</RulePriority>
        <OriginType>Redirect|Proxy|Mirror</OriginType>
        <OriginCondition>
            <!--回源设置的 HTTP 状态码设置根据回源类型决定，如果是 Proxy、Mirror 是404，如果
            <HTTPStatusCode>404</HTTPStatusCode>
            <Prefix></Prefix>
        </OriginCondition>
        <OriginParameter>
            <Protocol>HTTP|HTTPS|FOLLOW</Protocol>
            <!--回源是否保留源站的错误码-->
            <TransparentErrorCode>true|false</TransparentErrorCode>
            <!--回源是否保留原始查询参数，支持保留、不保留两种-->
            <FollowQueryString>true|false</FollowQueryString>
            <!--回源的请求头部参数-->
            <HttpHeader>
                <!--回源是否传输全部的请求头部，支持传递、不传递两种-->
                <FollowAllHeaders>true|false</FollowAllHeaders>
                <!--回源新增指定头部，最多10个，自定义头部中一对键值算一个-->
                <!--如果有值则新增头部，如果为空则不新增-->
                <NewHttpHeaders>
                    <Header>
                        <Key>x-cos|oss|amz-ContentType|CacheControl|ContentDisposit
                        <Value>string</Value>
                    </Header>
                </NewHttpHeaders>
                <!--回源透传原始请求的指定头部-->
                <FollowHttpHeaders>
                    <Header>
                        <Key>x-cos|oss|amz-ContentType|CacheControl|ContentDisposit
                    </Header>
                </FollowHttpHeaders>
                <!--回源不透传原始请求的指定头部-->
                <ForbidFollowHeaders>
                    <Header>
                        <Key>String</Key>
                    </Header>
                </ForbidFollowHeaders>
            </HttpHeader>
            <!--follow3xx参数-->
            <FollowRedirection>true|false</FollowRedirection>
            <!--重定向返回码参数只有在回源类型为 Redirect、Proxy 时可选，否则报参数错误-->
            <HttpRedirectCode>301|302|307</HttpRedirectCode>
        </OriginParameter>
    <!--切备错误码，默认5XX切换备站，添加此选项后支持4XX切换备站  -->
```

```
        <HTTPStandbyCode>
            <StatusCode>404</StatusCode>
            <StatusCode>403</StatusCode>
        </HTTPStandbyCode>
        <OriginInfo>
            <!--Mirror 模式支持设置多个源站，按比例回源，分担单个源站的回源流量，最多支持填入10
            <HostInfo>
                <HostName>bucketname-appid.cos.region.myqcloud.com</HostName>
                <!--源站权重，按比例回源-->
        <Weight>4</Weight>
                <!--备份回源地址，最多支持填入10条备份回源地址，节点命名依次按照1-10编号-->
                <StandbyHostName_1>bucketname2-appid.cos.region.myqcloud.com</Stand
                <StandbyHostName_2>bucketname3-appid.cos.region.myqcloud.com</Stand
            </HostInfo>
            <HostInfo>
                <HostName>bucketname4-appid.cos.region.myqcloud.com</HostName>
                <!--源站权重，按比例回源-->
                <Weight>2</Weight>
                <!--备份回源地址，最多支持填入10条备份回源地址，节点命名依次按照1-10编号-->
                <StandbyHostName_1>bucketname5-appid.cos.region.myqcloud.com</Stand
                <StandbyHostName_2>bucketname6-appid.cos.region.myqcloud.com</Stand
            </HostInfo>
            <FileInfo>
                <!--回源为指定的文件-->
                <FixedFileConfiguration>
                    <FixedFilePath>String</FixedFilePath>
                </FixedFileConfiguration>
                <!--回源文件的新增文件前缀-->
                <Prefix></Prefix>
                <!--回源文件的新增文件后缀-->
                <Suffix></Suffix>
            </FileInfo>
        </OriginInfo>
    </OriginRule>
</OriginConfiguration>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型	是否必选
OriginConfiguration	无	Origin 回源配置	Container	是

Container 节点 OriginConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选

OriginRule	OriginConfiguration	支持多条 OriginRule，通过优先级区分规则执行先后	Container	是

Container 节点 OriginRule 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
RulePriority	OriginConfiguration.OriginRule	通过优先级区分规则执行先后	Integer	是
OriginType	OriginConfiguration.OriginRule	回源类型，支持异步回源（Proxy）、同步回源（Mirror）和重定向回源（Redirect）三种模式。枚举值：`Proxy|Mirror|Redirect`	Container	是
OriginCondition	OriginConfiguration.OriginRule	回源配置，配置用户使用的 HTTP 传输协议等信息	Container	是
OriginParameter	OriginConfiguration.OriginRule	回源地址相关信息	Container	是
OriginInfo	OriginConfiguration.OriginRule	源站信息。例如源站域名或者源站 IP 等信息	Container	是

Container 节点 OriginCondition 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
HTTPStatusCode	OriginConfiguration.OriginRule.OriginCondition	触发回源的HTTP状态码，Proxy 和 Mirror 模式支持填写404，Redirect 模式支持填写4XX和5XX	String	是
Prefix	OriginConfiguration.OriginRule.OriginCondition	触发回源的文件前缀，默认为空，任意文件均可触发	String	否

Container 节点 OriginParameter 的内容：

节点名称（关键字）	父节点	描述	类型
Protocol	OriginConfiguration.OriginRule.OriginParameter	回源使用的协议，枚举值为 HTTP（使用 HTTP 协议），HTTPS（使用 HTTPS 协议）、FOLLOW（跟随用户使用的协议），默认值为 FOLLOW。	Strin
FollowQueryString	OriginConfiguration.OriginRule.OriginParameter	回源是否需要透传 HTTP 请求串，枚举值：`true|false`，默认为 `true`	Bool
HttpHeader	OriginConfiguration.OriginRule.OriginParameter	是否需要设置 Http 头部传输配置。	Cont
FollowRedirection	OriginConfiguration.OriginRule.OriginParameter	源站 `3XX` 响应策略，枚举值 `true|false`，选择 true 时跟随源站 `3xx` 重定向请求获取到资源，并将资源保存到 COS 上;选择 `false` 时透传 `3XX` 响应，不获取资源），默认为 `true` 。	Bool
HttpRedirectCode	OriginConfiguration.OriginRule.OriginParameter	仅支持 `Redirect` 和 `Proxy` 模式，设置重定向返回码参数，枚举值 `301|302|307`，默认为 `302` 。	Strin

Container 节点 HttpHeader 的内容：

节点名称（关键字）	父节点	描述
FollowAllHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	是否传输全部的请求头部，枚举值：`true|false`，默认为 `false` 。

NewHttpHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	设置回源新增指定头部，最多10个。
FollowHttpHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	设置回源透传原始请求的指定头部。
ForbidFollowHeaders	OriginConfiguration.OriginRule.OriginParameter.HttpHeader	设置回源不透传的原始请求的指定头部。

Container 节点 NewHttpHeaders、FollowHttpHeaders、ForbidFollowHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Header	OriginConfiguration.OriginRule.OriginParameter.HttpHeader.NewHttpHeader	回源到源站时添加或者指定传递的自定义头部，默认为空。	Container

Container 节点 Header 的内容：

节点名称（关键字）	父节点	描述
Key	OriginConfiguration.OriginRule.OriginParameter.HttpHeader.NewHttpHeader.UserMetaData	用户设置 Conten
Value	OriginConfiguration.OriginRule.OriginParameter.HttpHeader.NewHttpHeader.UserMetaData	用户设置

Container 节点 OriginInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
HostInfo	OriginConfiguration.OriginRule.OriginInfo	源站信息。Mirror 模式支持设置多个源站，按比例回源，分担单个源站的回源流量，最多支持填入10条回源地址，比例按权重分配；Proxy 和 Redirect 模式仅支持一个源站。	Container	是
FileInfo	OriginConfiguration.OriginRule.OriginInfo	回源文件信息。	Container	是

Container 节点 HostInfo 的内容：

节点名称（关键字）	父节点	描述
HostName	OriginConfiguration.OriginRule.OriginInfo.HostInfo	源站域名或者源站 IP。
Weight	OriginConfiguration.OriginRule.OriginInfo.HostInfo	源站权重，Mirror 模式下配置了多
StandbyHostName_N	OriginConfiguration.OriginRule.OriginInfo.HostInfo	备份回源地址，最多支持填入10条 如 `StandbyHostName_1` 、 `S`

Container 节点 FileInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Prefix	OriginConfiguration.OriginRule.OriginInfo.FileInfo	回源文件的新增文件前缀，默认为空。	String	否
Suffix	OriginConfiguration.OriginRule.OriginInfo.FileInfo	回源文件的新增文件后缀，默认为空。	String	否
FixedFileConfiguration	OriginConfiguration.OriginRule.OriginInfo.FileInfo	回源到固定	String	否

版权所有：腾讯云计算（北京）有限责任公司

		的文件。		

Container 节点 FixedFileConfiguration 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
FixedFilePath	OriginConfiguration.OriginRule.OriginInfo.FileInfo.FixedFileConfiguration	回源的固定文件路径。	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

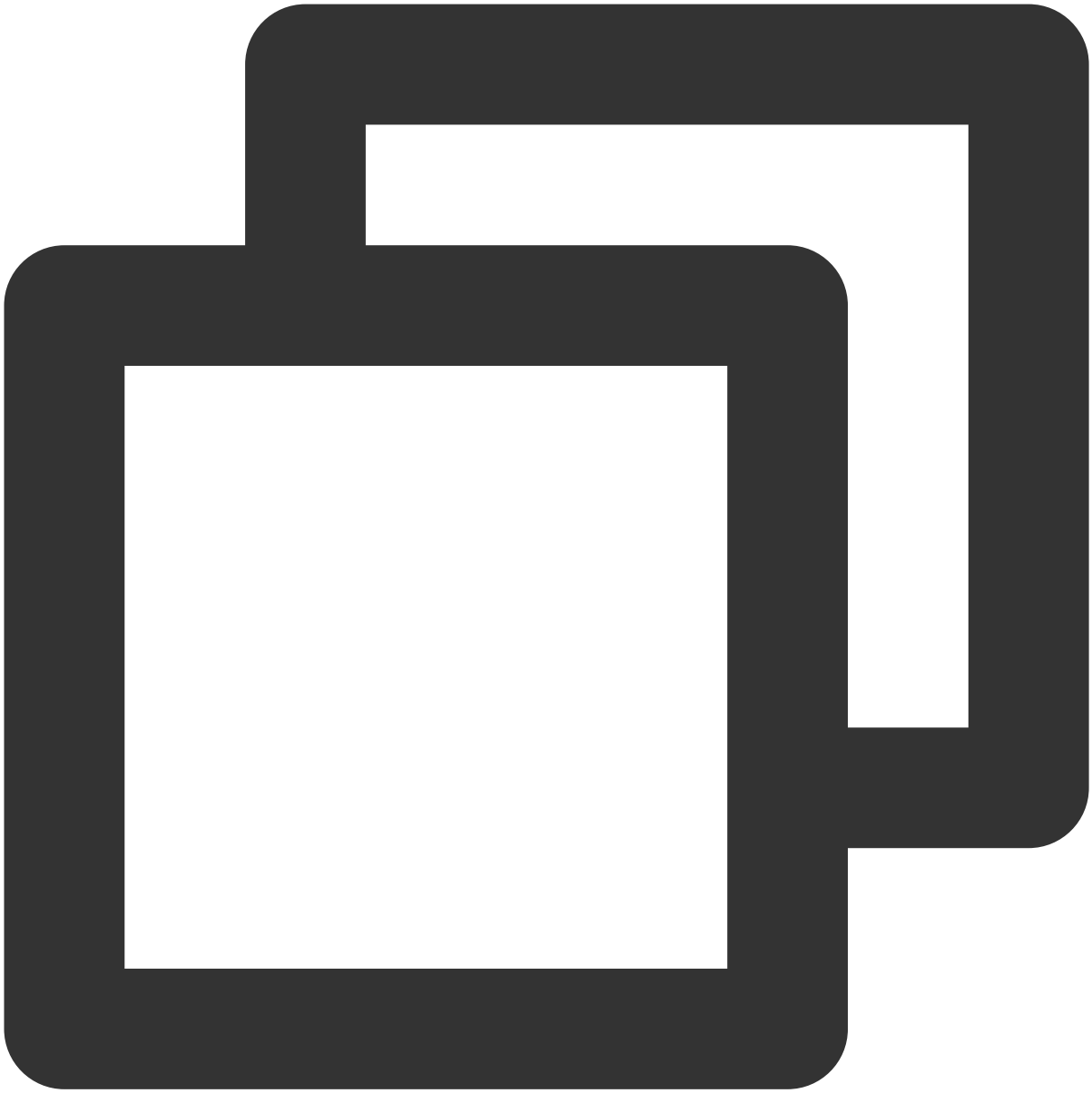
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：普通 Mirror 模式

请求

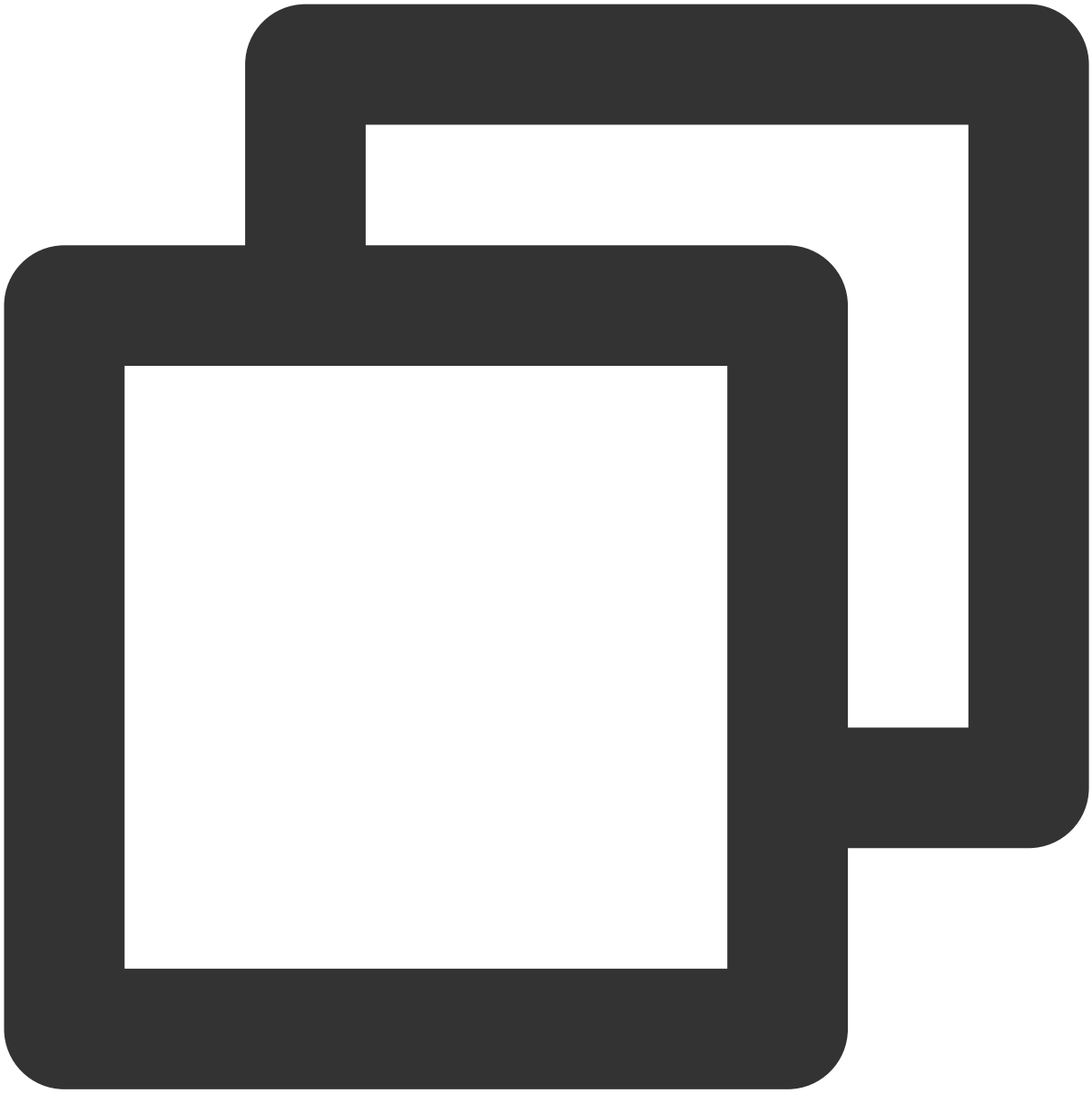


```
PUT /?origin= HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
Content-Length: 347
Date: Sun, 28 Apr 2019 12:02:24 GMT

<?xml version="1.0" encoding="UTF-8"?>
<OriginConfiguration>
  <OriginRule>
    <RulePriority>1</RulePriority>
    <OriginType>Mirror</OriginType>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <OriginCondition>
      <HTTPStatusCode>404</HTTPStatusCode>
      <Prefix></Prefix>
    </OriginCondition>
    <OriginParameter>
      <Protocol>HTTP</Protocol>
      <FollowQueryString>true</FollowQueryString>
      <HttpHeader>
        <FollowAllHeaders>false</FollowAllHeaders>
        <NewHttpHeaders>
          <Header>
            <Key>x-cos</Key>
            <Value>exampleHeader</Value>
          </Header>
        </NewHttpHeaders>
        <FollowHttpHeaders>
          <Header>
            <Key>exampleHeaderKey</Key>
          </Header>
        </FollowHttpHeaders>
      </HttpHeader>
      <FollowRedirection>true</FollowRedirection>
      <HttpRedirectCode>302</HttpRedirectCode>
    </OriginParameter>
    <OriginInfo>
      <HostInfo>
        <HostName>examplebucket-1250000000.cos.ap-shanghai.myqcloud.com</HostName>
      </HostInfo>
    </OriginInfo>
  </OriginRule>
</OriginConfiguration>
```

响应

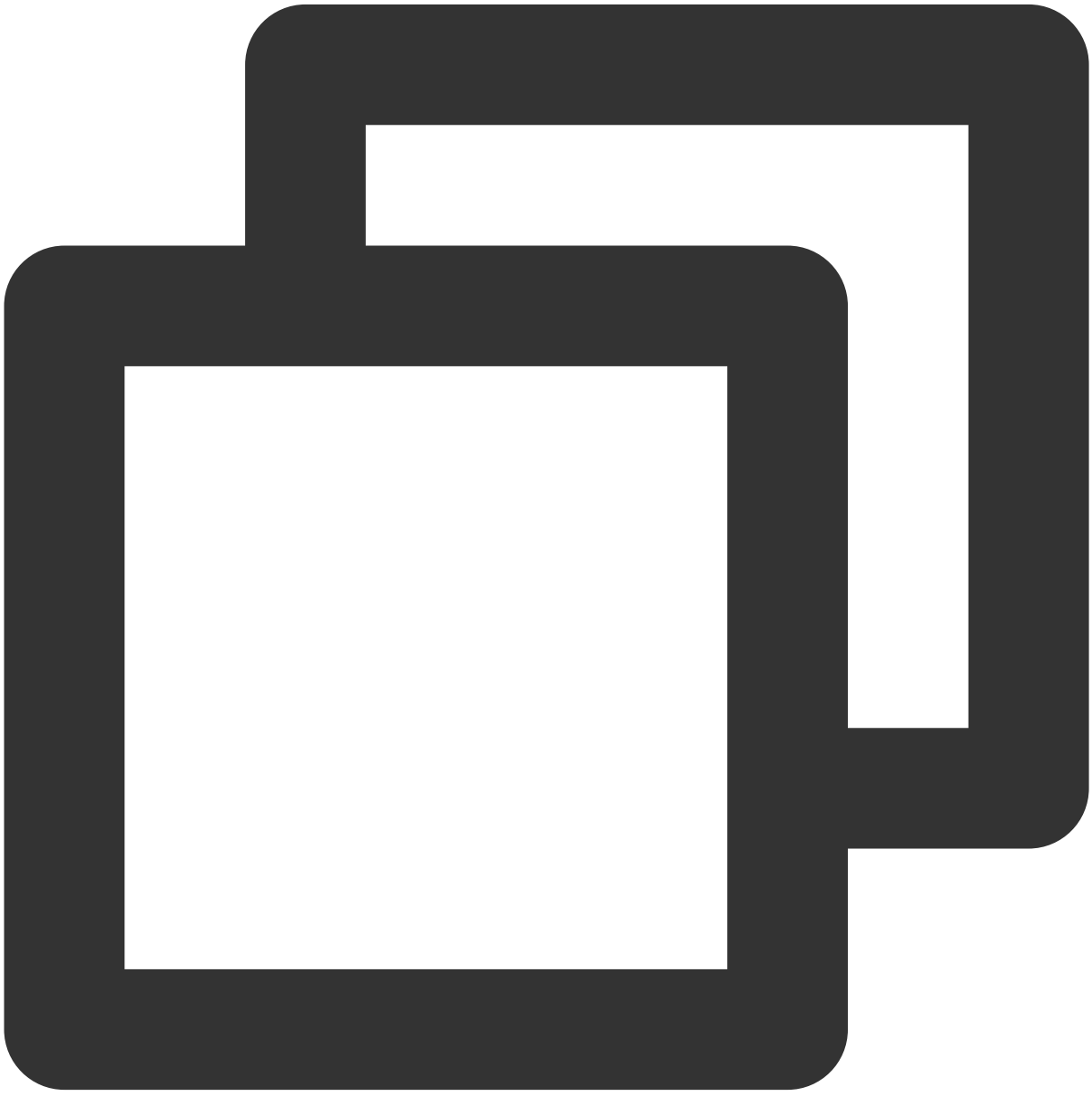


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Sun, 28 Apr 2019 12:02:45 GMT
Server: tencent-cos
x-cos-request-id: NWNjNTk2NTFfMmM4OGY3MGFfNadfadsfY2****
```

案例二：带权重的 Mirror 模式

在如下示例中，回源会透传除 x-cos-example-header 之外原始请求的所有头部；规则指定了两个回源源站，源站1 `bucketname1-appid.cos.region.myqcloud.com` 和源站2 `bucketname2-appid.cos.region.myqcloud.com` ，权重分别为8和2，80%的回源请求会访问源站1，20%的回源请求会访问源站2。

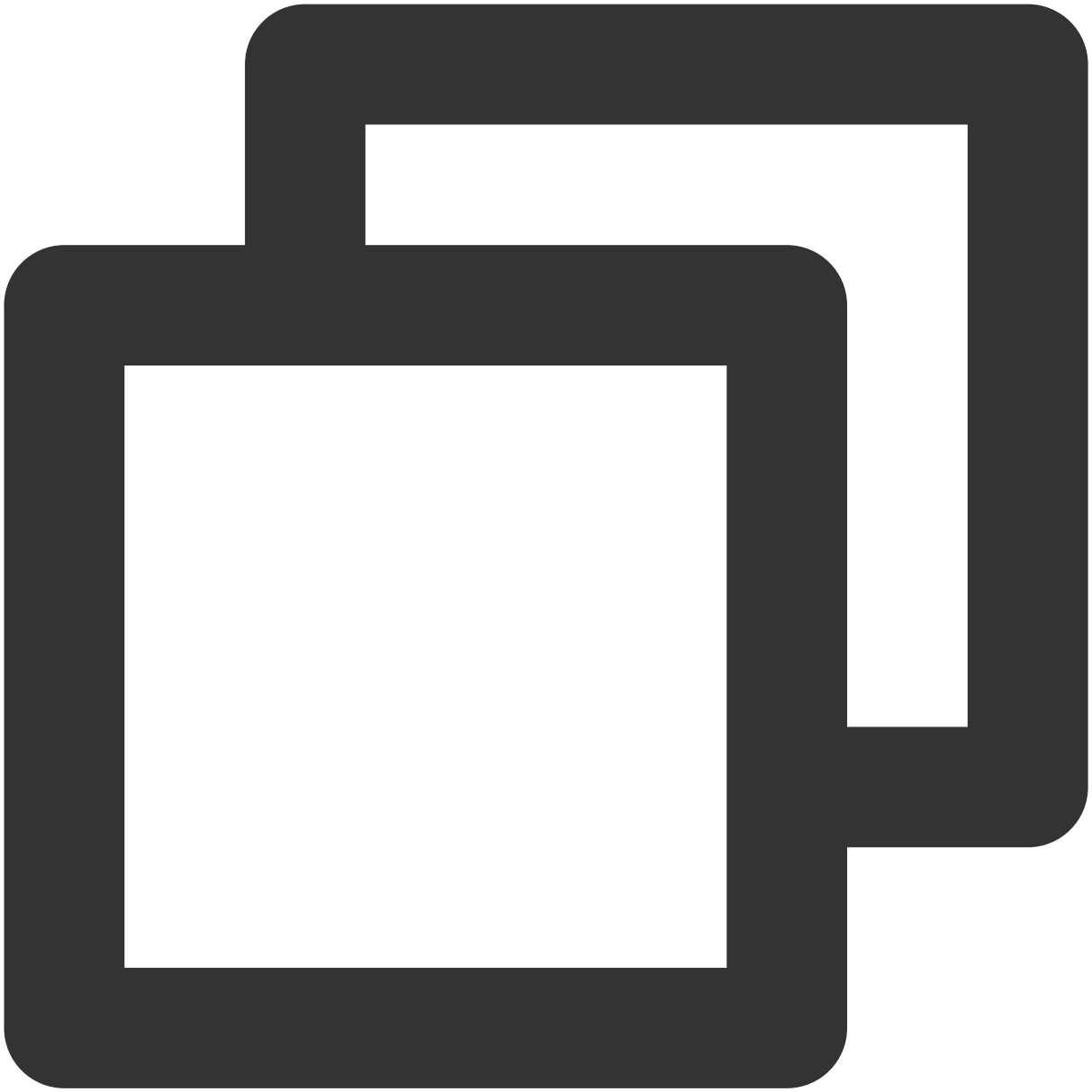
请求



```
PUT /?origin= HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization:Authstring
```

```
Content-Length:Content-Length
DATE:Sun, 28 Apr 2019 12:02:24 GMT

<?xml version="1.0" encoding="UTF-8"?>
<OriginConfiguration>
    <OriginRule>
        <RulePriority>1</RulePriority>
        <OriginType>Mirror</OriginType>
        <OriginCondition>
            <HTTPStatusCode>404</HTTPStatusCode>
            <Prefix></Prefix>
        </OriginCondition>
        <OriginParameter>
            <Protocol>HTTPS</Protocol>
            <FollowQueryString>true</FollowQueryString>
            <HttpHeader>
                <FollowAllHeaders>true</FollowAllHeaders>
                <ForbidFollowHeaders>
                    <Header>
                        <Key>x-cos-example-header</Key>
                    </Header>
                </ForbidFollowHeaders>
            </HttpHeader>
        </OriginParameter>
        <OriginInfo>
            <HostInfo>
                <HostName>bucketname1-appid.cos.region.myqcloud.com</HostName>
                <Weight>8</Weight>
            </HostInfo>
            <HostInfo>
                <HostName>bucketname2-appid.cos.region.myqcloud.com</HostName>
                <Weight>2</Weight>
            </HostInfo>
        </OriginInfo>
    </OriginRule>
</OriginConfiguration>
```

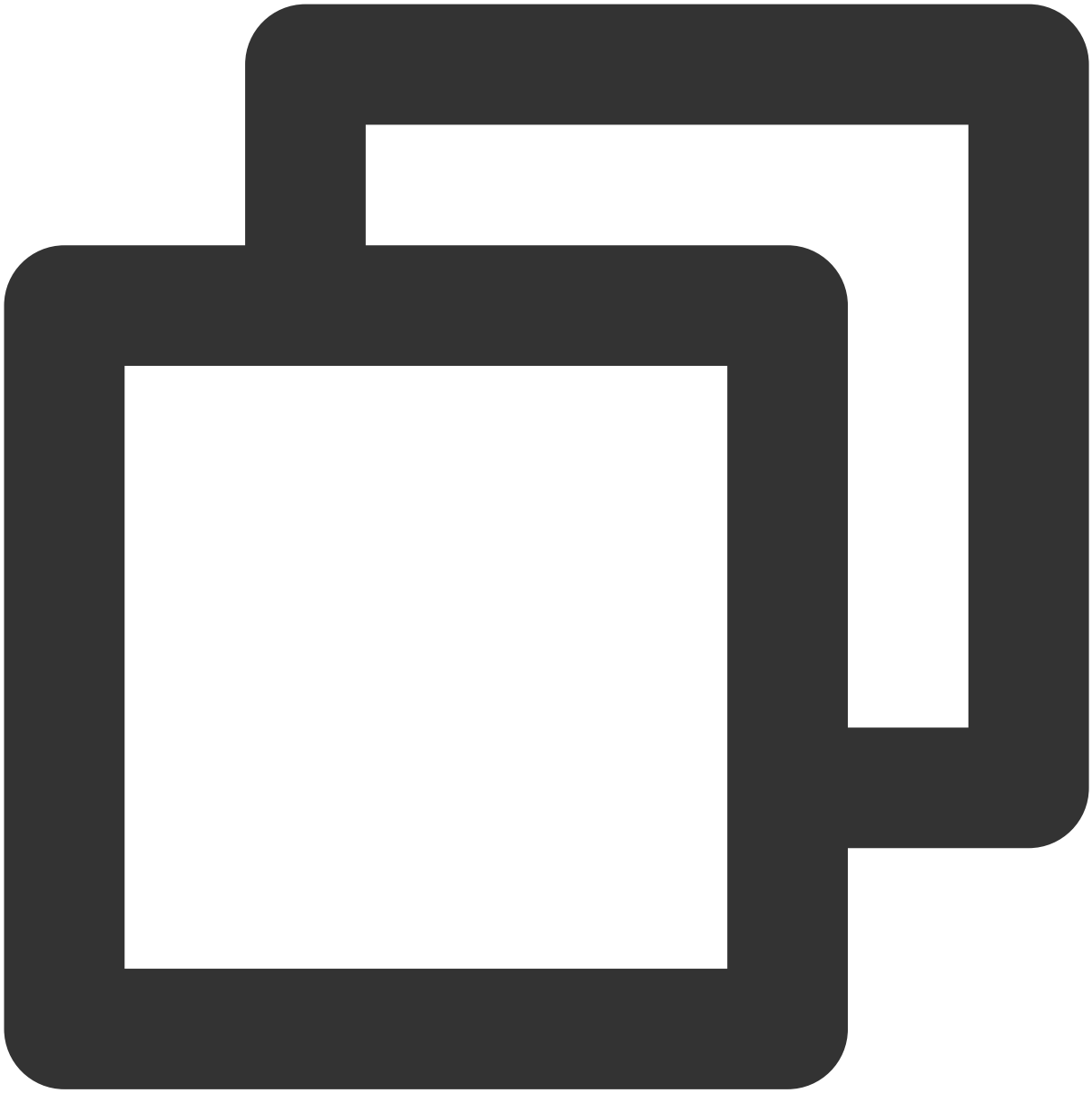
响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Sun, 28 Apr 2019 12:02:25 GMT
Server: tencent-cos
x-cos-request-id: NWNjNTk2NTFfMmM4OGY3MGFfNTI1****
```

案例三：Proxy 模式

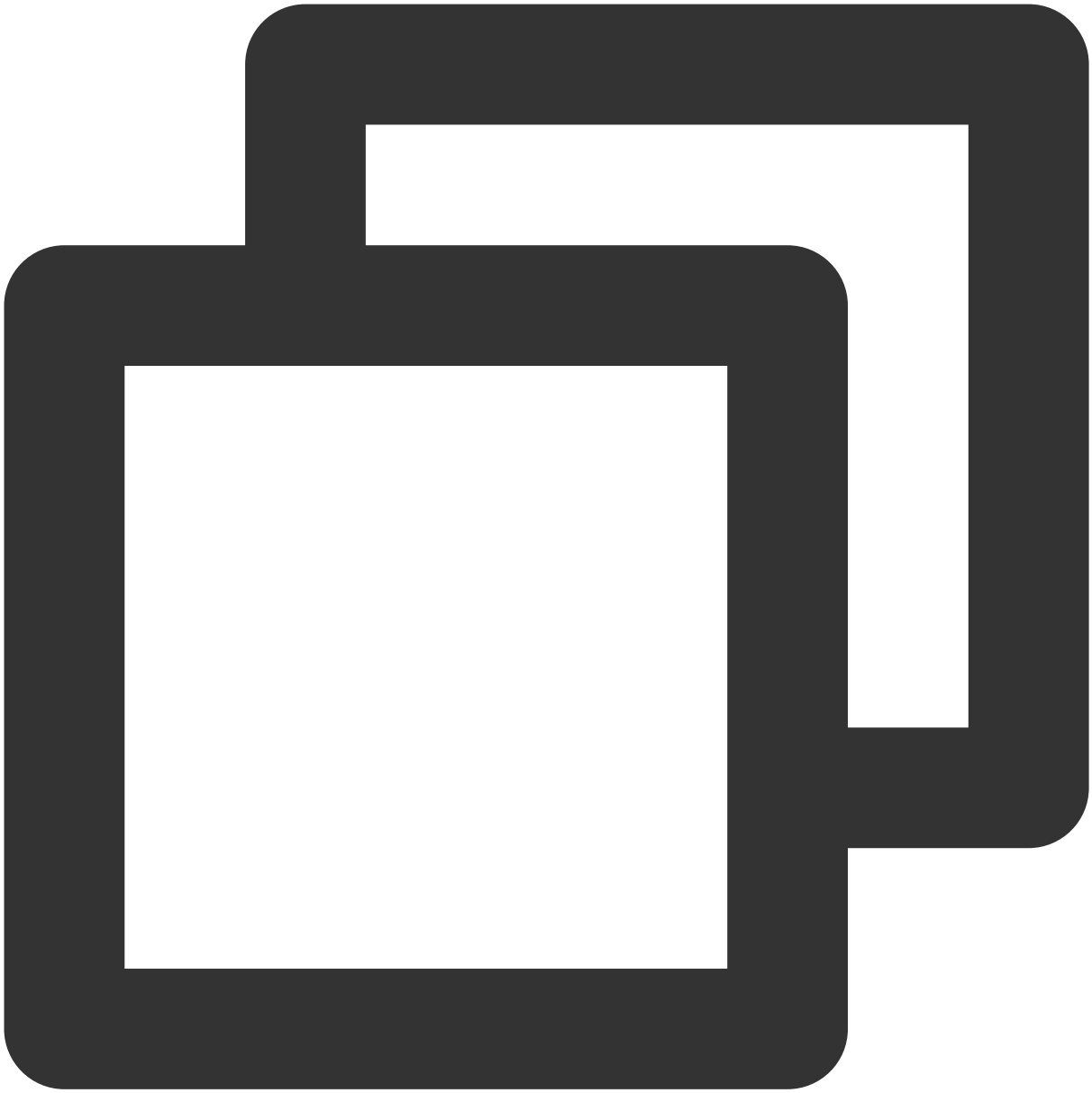
请求



```
PUT /?origin= HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization:Authstring
Content-Length: 347
DATE:Sun, 28 Apr 2019 12:02:24 GMT

<?xml version="1.0" encoding="UTF-8"?>
<OriginConfiguration>
  <OriginRule>
    <RulePriority>1</RulePriority>
    <OriginType>Proxy</OriginType>
```

```
    <OriginCondition>
        <HTTPStatusCode>404</HTTPStatusCode>
        <Prefix></Prefix>
    </OriginCondition>

    <OriginParameter>
        <Protocol>FOLLOW</Protocol>
        <FollowQueryString>true</FollowQueryString>
        <HttpHeader>
            <FollowAllHeaders>true</FollowAllHeaders>
            <NewHttpHeaders>
                <Header>
                    <Key>x-cos-ContentType</Key>
                    <Value>csv</Value>
                </Header>
            </NewHttpHeader>
            <FollowHttpHeaders>
                <Header>
                    <Key>Content-Type</Key>
                </Header>
            </FollowHttpHeaders>
        </HttpHeader>
        <FollowRedirection>true</FollowRedirection>
    </OriginParameter>

    <OriginInfo>
        <HostInfo>
            <HostName>examplebucket-1250000000.cos.ap-shanghai.myqcloud.com</HostNa
        </HostInfo>
    </OriginInfo>
  </OriginRule>
</OriginConfiguration>
```

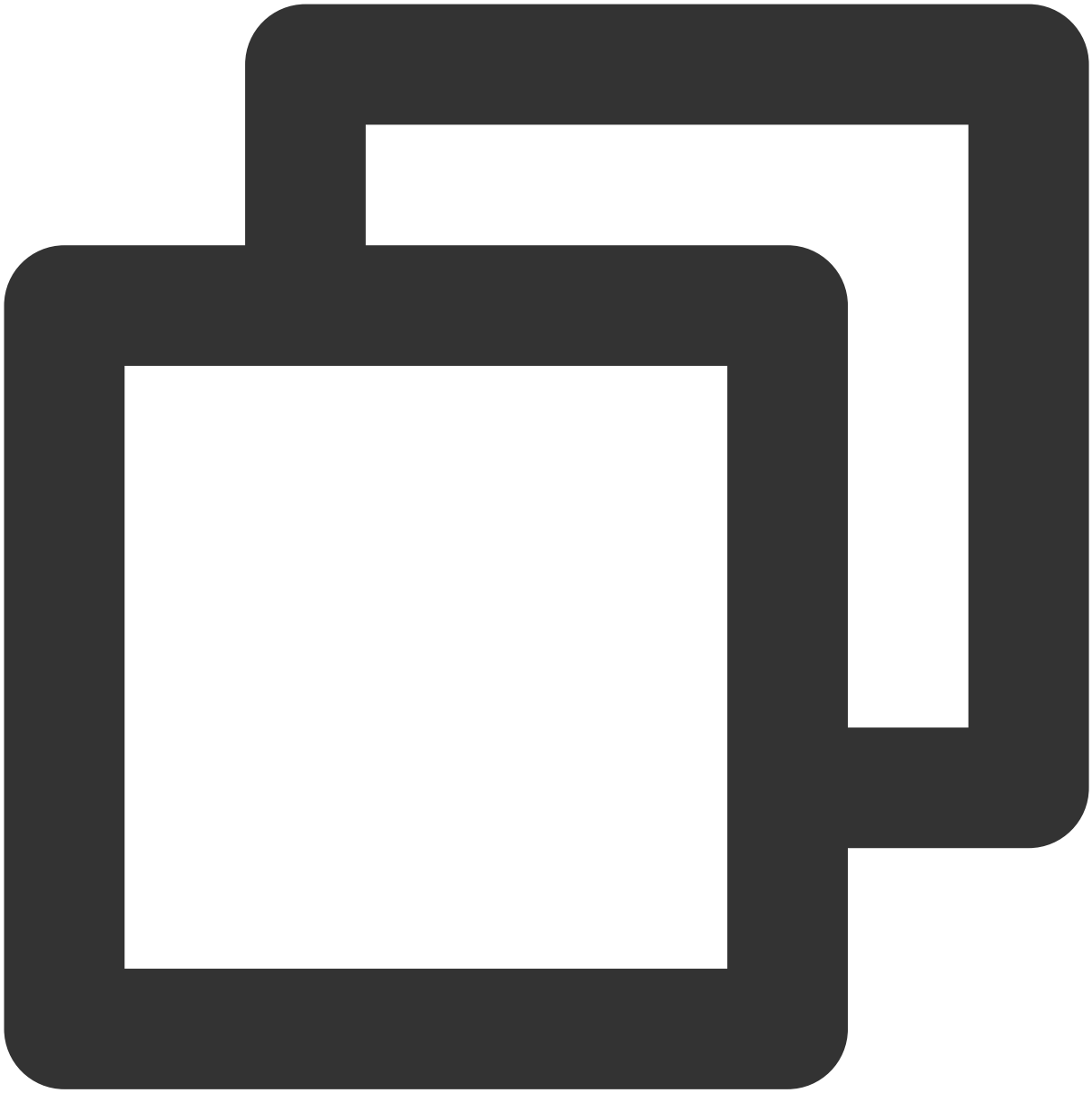
响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Sun, 28 Apr 2019 12:02:45 GMT
Server: tencent-cos
x-cos-request-id: NWNjNTk2NTFfMmM4OGY3MGFfNadfadsfY****
```

案例四：Redirect 模式

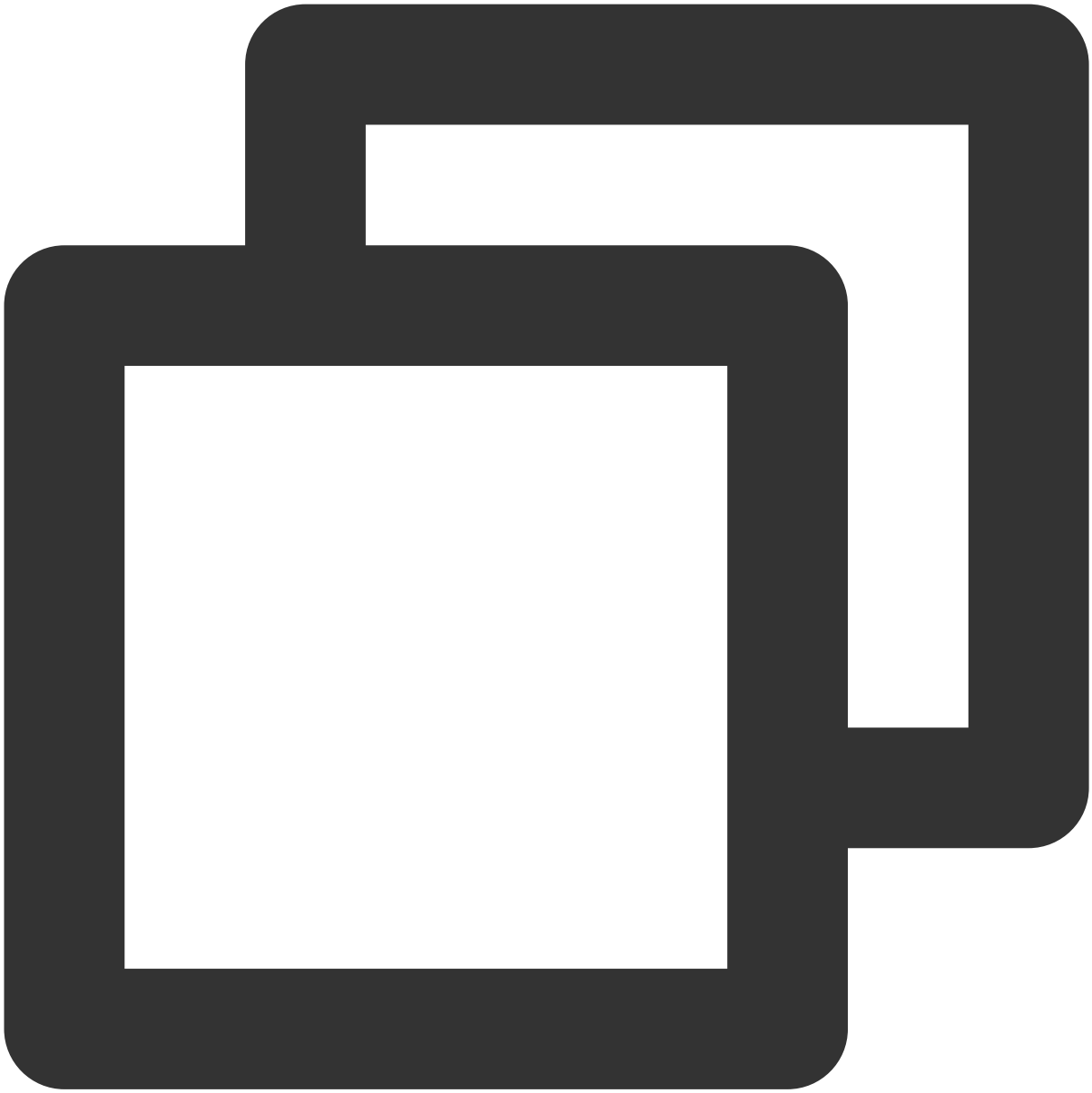
请求



```
PUT /?origin= HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization:Authstring
Content-Length: 347
DATE:Sun, 28 Apr 2019 12:02:24 GMT

<?xml version="1.0" encoding="UTF-8"?>
<OriginConfiguration>
  <OriginRule>
    <RulePriority>1</RulePriority>
    <OriginType>Redirect</OriginType>
```

```
    <OriginCondition>
        <HTTPStatusCode>403</HTTPStatusCode>
        <Prefix></Prefix>
    </OriginCondition>
    <OriginParameter>
        <HttpRedirectCode>301</HttpRedirectCode>
    </OriginParameter>
    <OriginInfo>
        <HostInfo>
        <HostName>examplebucket-1250000000.cos.ap-shanghai.myqcloud.com</HostName>
        </HostInfo>
    </OriginInfo>
  </OriginRule>
</OriginConfiguration>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Sun, 28 Apr 2019 12:02:25 GMT
Server: tencent-cos
x-cos-request-id: NWNjNTk2NTFfMmM4OGY3MGFfNTI1****
```

DELETE Bucket Origin

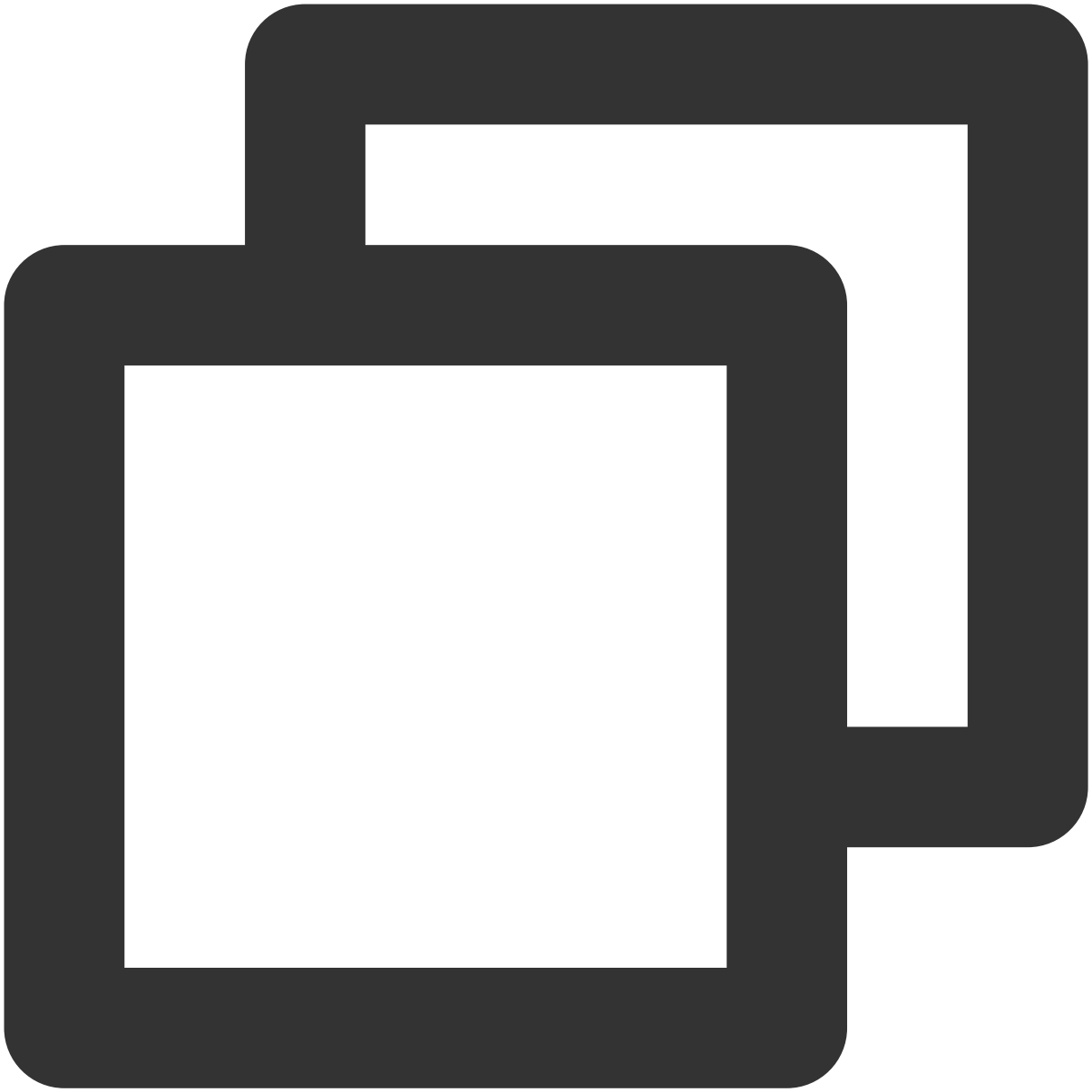
最近更新时间：2023-12-21 12:59:51

功能描述

DELETE Bucket Origin 接口用于删除存储桶的回源配置。

请求

请求示例



```
DELETE /?origin HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

该请求需结合请求体一起使用。

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

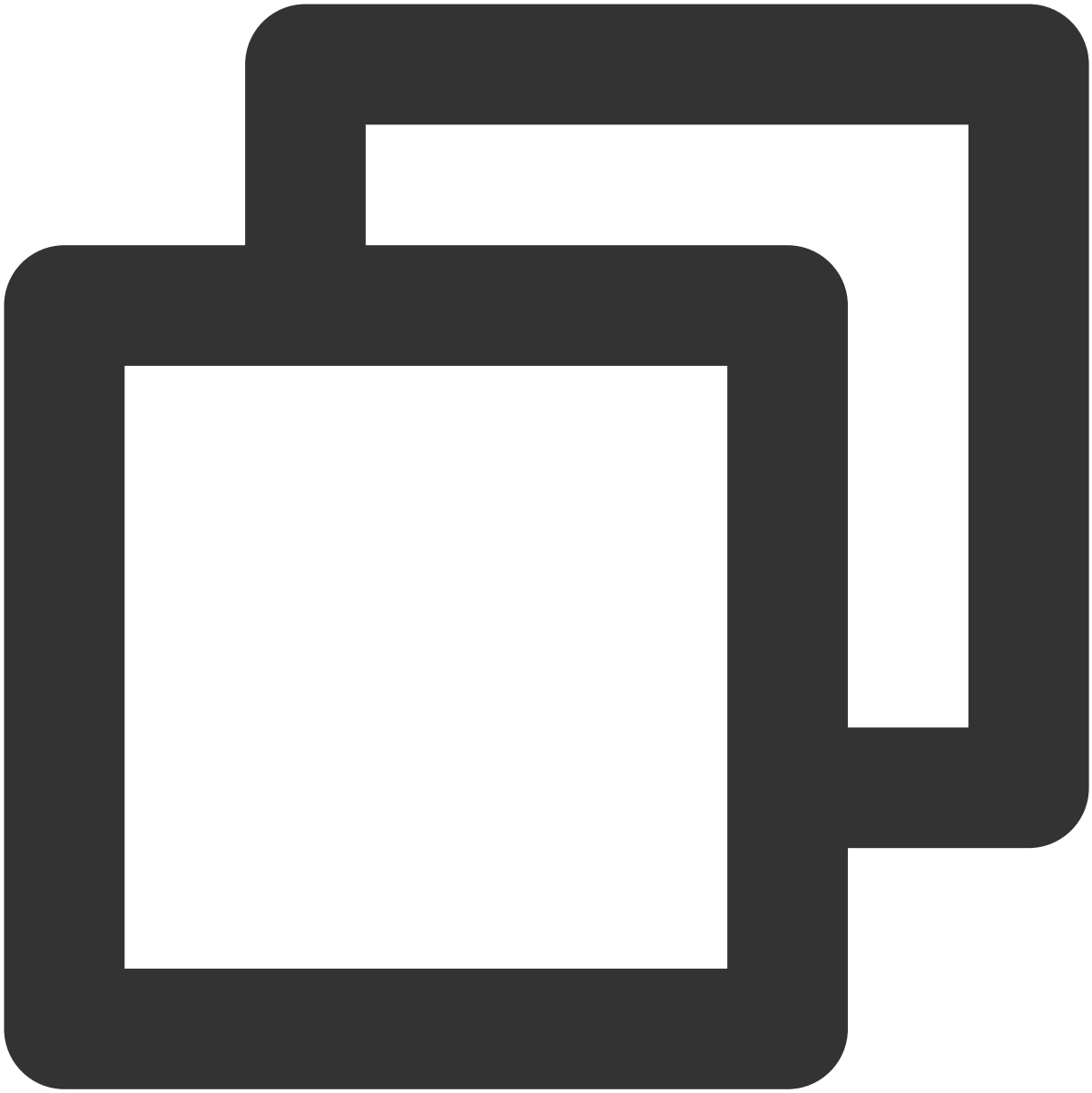
此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

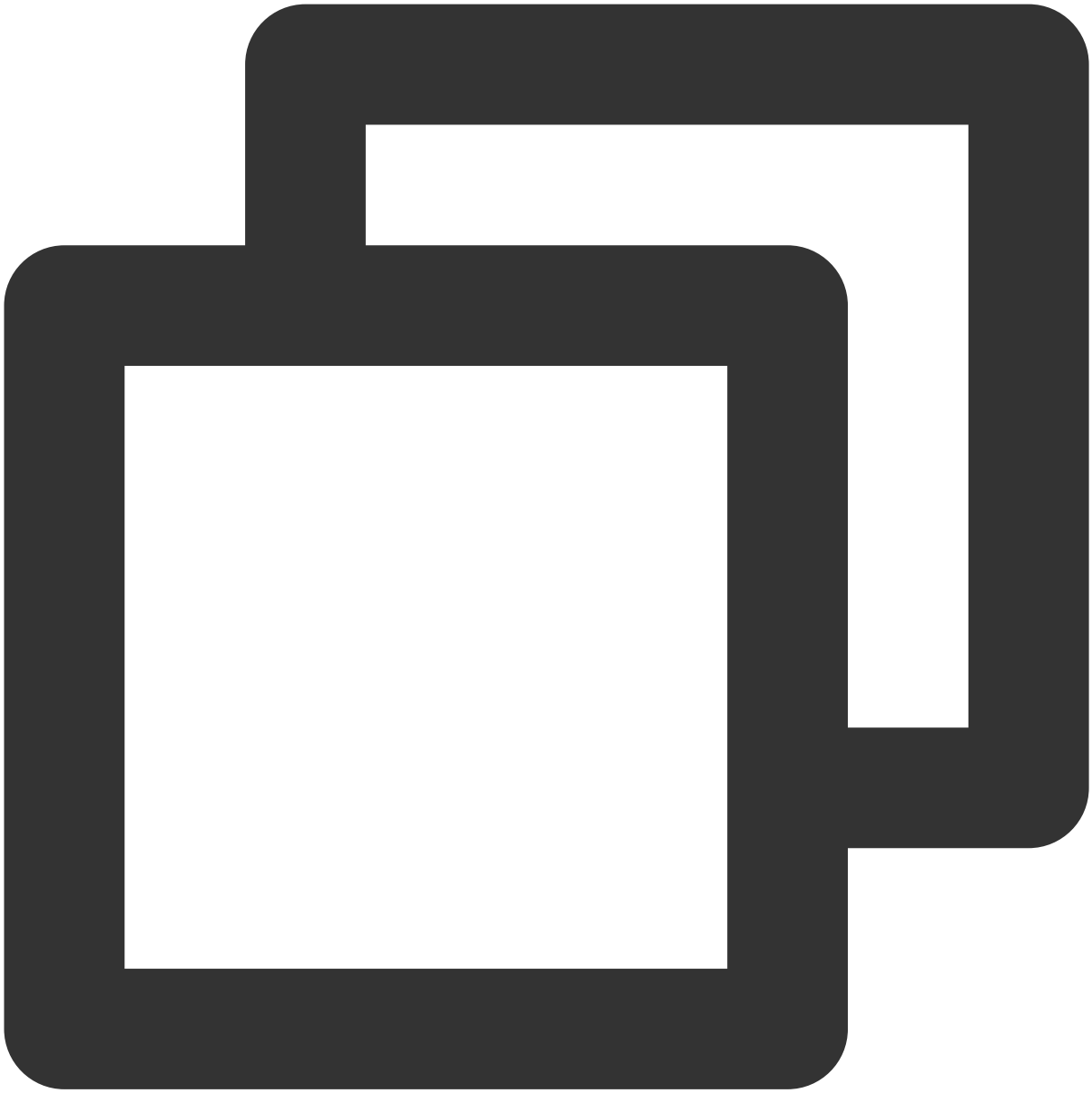
实际案例

请求



```
DELETE /?origin= HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization: Auth String
Date: Sun, 28 Apr 2019 12:02:24 GMT
```

响应



```
HTTP/1.1 204 No Content
Server: tencent-cos
Date: Mon, 28 Aug 2018 02:53:40 GMT
x-cos-request-id: NWNjNTk2NTFfMmM4OGY3MGFfNTI1****
```

Object 接口
基本操作
PUT Object

最近更新时间：2024-02-27 10:28:02

功能描述

PUT Object 接口请求可以将本地的对象（Object）上传至指定存储桶中。该 API 的请求者需要对存储桶有写入权限。

说明

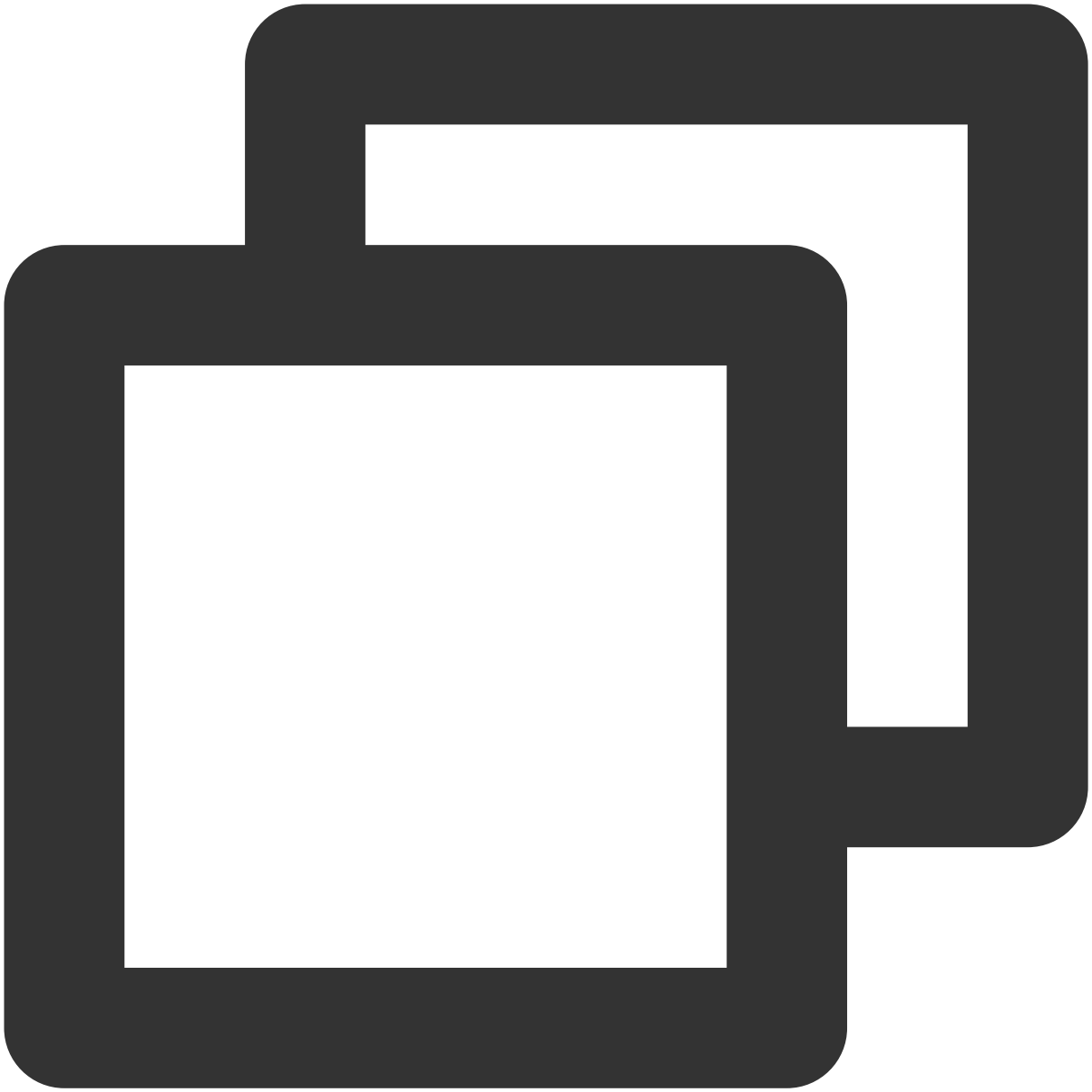
PUT Object 接口最大支持上传5GB文件。如需上传大于5GB的文件，请使用 分块上传 的 API 接口。
如果请求头的 Content-Length 值小于实际请求体（body）中传输的数据长度，COS 仍将成功创建文件，但对象大小只等于 Content-Length 中定义的大小，其他数据将被丢弃。
如果试图添加已存在的同名对象且没有启用版本控制，则新上传的对象将覆盖原来的对象，成功时返回200 OK。
对象存储中本身没有文件夹和目录的概念，如需将对象上传到指定的文件夹或路径，可通过 `/` 实现，例如将 picture.png 上传到 doc 文件夹，则对象键设置为 doc/picture.png；如需创建文件夹 doc，则对象键设置为 doc/。更多说明请参见 文件夹和目录。

版本控制

如果对存储桶启用版本控制，对象存储将自动为要添加的对象生成唯一的版本 ID。对象存储使用 x-cos-version-id 响应头部在响应中返回此标识。
如果暂停存储桶的版本控制，则对象存储始终将 null 用作存储在存储桶中的对象的版本 ID，且不返回 x-cos-version-id 响应头部。

请求

请求示例



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: Content Type
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String


[Object Content]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
Cache-Control	RFC 2616 中定义的缓存指令，将作为对象元数据保存	string	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为对象元数据保存	string	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为对象元数据保存	string	否
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型（MIME），此头部用于描述待上传对象的内容类型，将作为对象元数据保存。 例如 `text/html` 或 `image/jpeg`	string	是
Expires	RFC 2616 中定义的缓存失效时间，将作为对象元数据保存	string	否
Transfer-Encoding	如果希望在上传时分块传输，则指定 Transfer-Encoding: chunked 请求头部，此时请求体遵循 RFC 2616 中定义的传输编码格式，且不能指定 Content-Length 请求头部	string	否
x-cos-meta-*	包括用户自定义元数据头部后缀和用户自定义元数据信息，将作为对象元数据保存，大小限制为2KB **注意：**用户自定义元数据信息支持下划线（_），但用户自定义元数据头部后缀不支持下划线，仅支持减号（-）	string	否
x-cos-storage-class	对象存储类型。枚举值请参见 存储类型概述 文档，例如 MAZ_STANDARD，MAZ_STANDARD_IA，INTELLIGENT_TIERING，MAZ_INTELLIGENT_TIERING，STANDARD_IA，ARCHIVE，DEEP_ARCHIVE。默认值：STANDARD	enum	否
x-cos-traffic-	针对本次上传进行流量控制的限速值，必须为数字，单位默认为	integer	否

limit	bit/s。限速值设置范围为819200 - 838860800，即800Kb/s - 800Mb/s，如果超出该范围将返回400错误		
x-cos-tagging	对象的标签集合，最多可设置10个标签（例如，Key1=Value1&Key2=Value2）。 标签集合中的 Key 和 Value 必须先进行 URL 编码。	string	否

访问控制列表（ACL）相关头部

在上传对象时可以通过指定下列请求头部来设置对象的访问权限：

名称	描述	类型	是否必选
x-cos-acl	定义对象的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中对象的预设 ACL 部分，例如 default，private，public-read 等，默认为 default **注意：**如果您不需要进行对象 ACL 控制，请设置为 default 或者此项不进行设置，默认继承存储桶权限	enum	否
x-cos-grant-read	赋予被授权者读取对象的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-full-control	赋予被授权者操作对象的所有权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否

服务端加密（SSE）相关头部

在上传对象时可以使用服务端加密，请参见 服务端加密专用头部。

对象锁定相关头部

在上传对象时可以使用对象锁定，请参见 对象锁定专用头部。

请求体

此接口请求的请求体为对象（文件）内容。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

版本控制相关头部

在启用版本控制的存储桶中上传对象，将返回下列响应头部：

名称	描述	类型
x-cos-version-id	对象的版本 ID	string

服务端加密（SSE）相关头部

如果在上传对象时使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

此接口响应体为空。

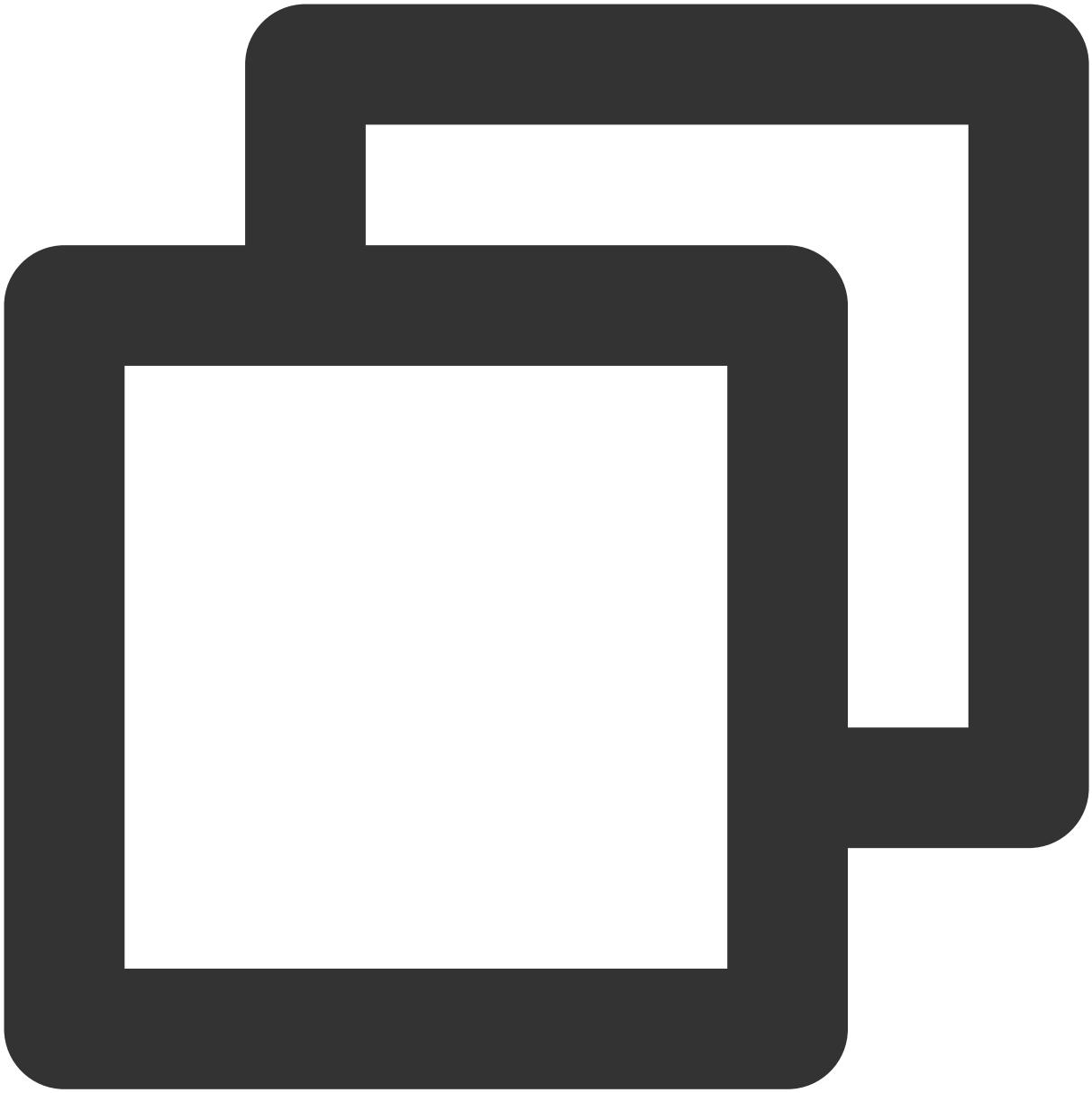
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

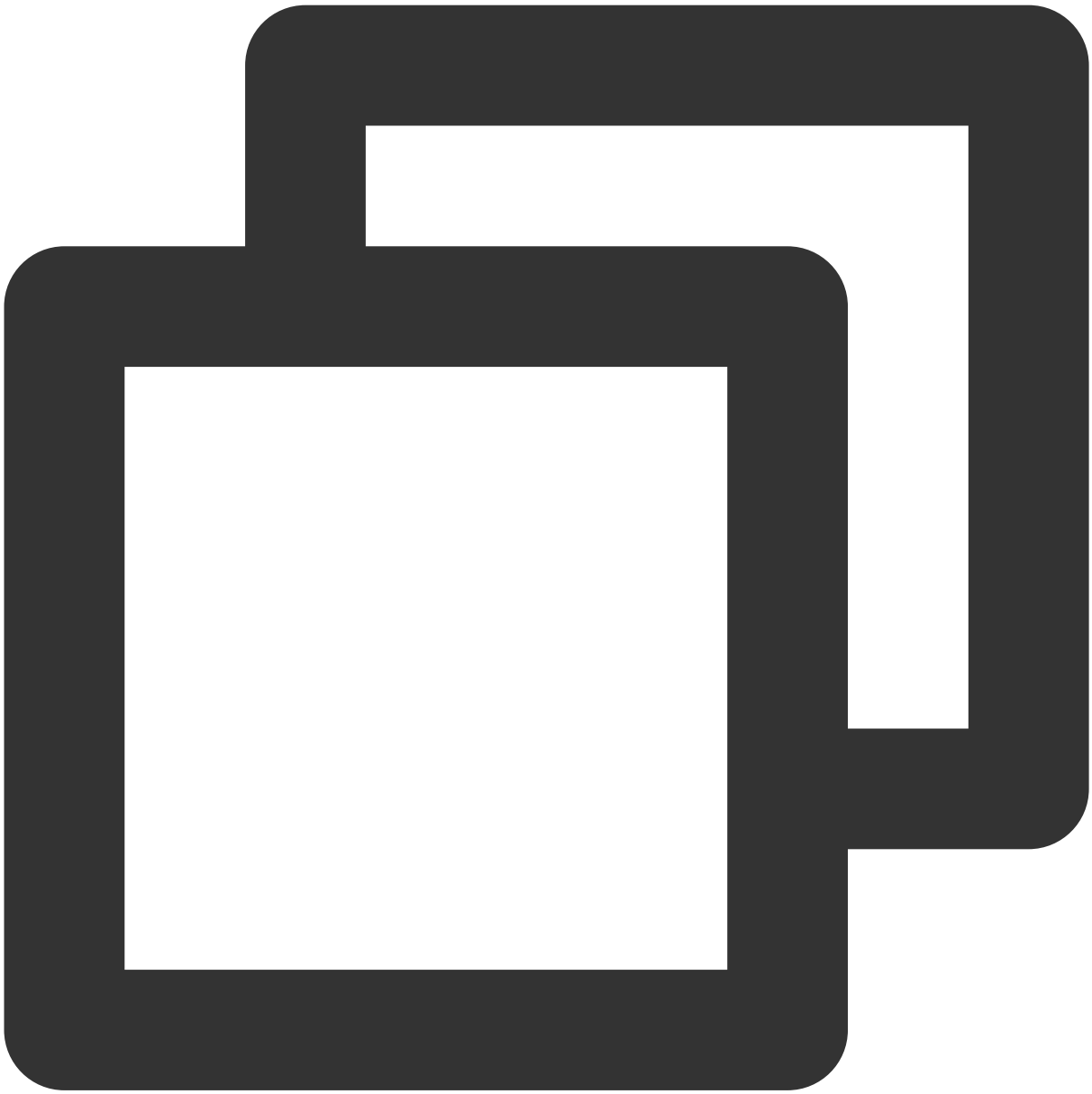
案例一：简单案例（未启用版本控制）

请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:35:05 GMT
Content-Type: image/jpeg
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

[Object Content]
```

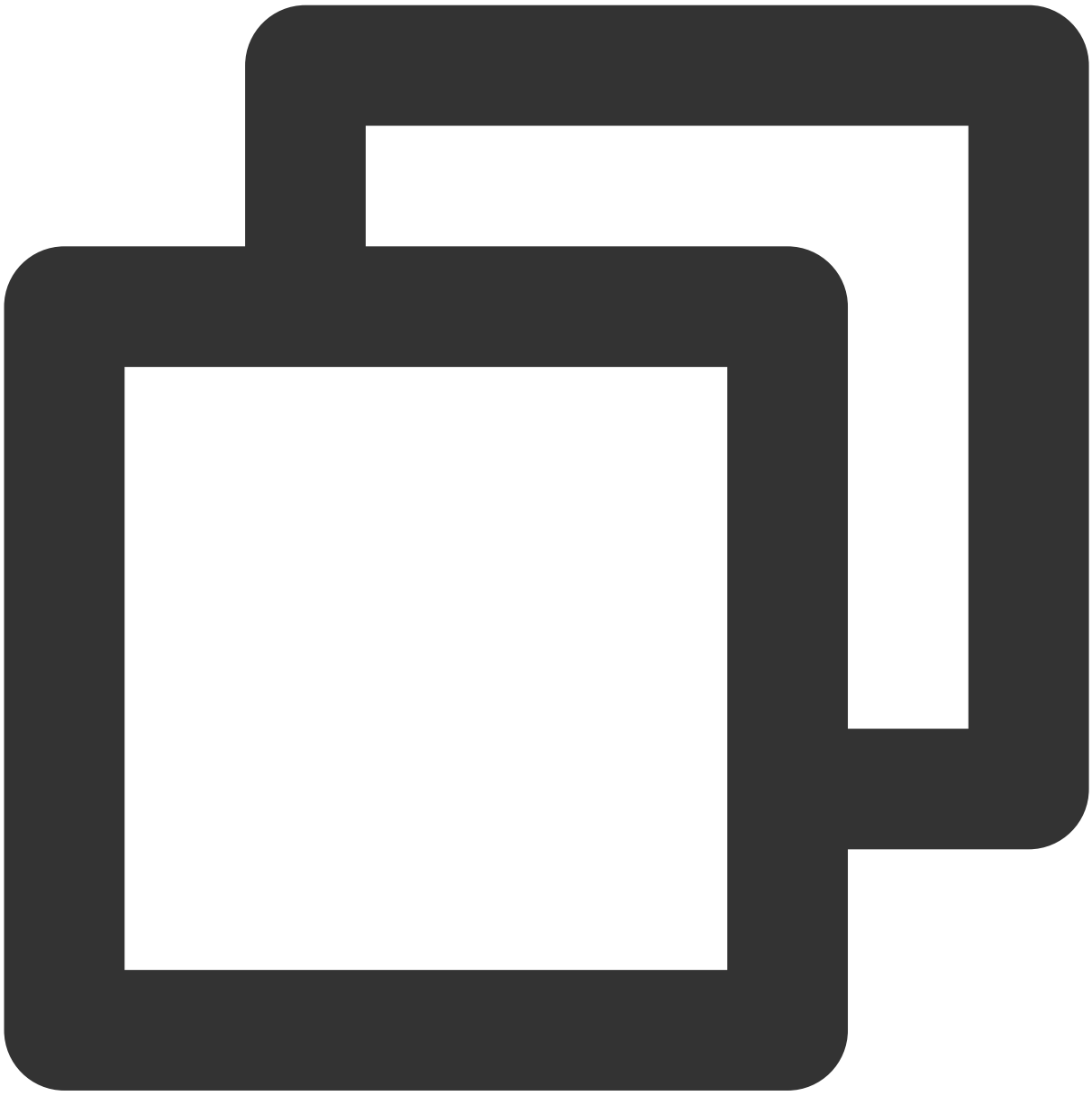
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:35:05 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNkYzlfNjRiODJhMDlfMzFmYzhfMTFm****
```

案例二：使用请求头部指定元数据和 ACL

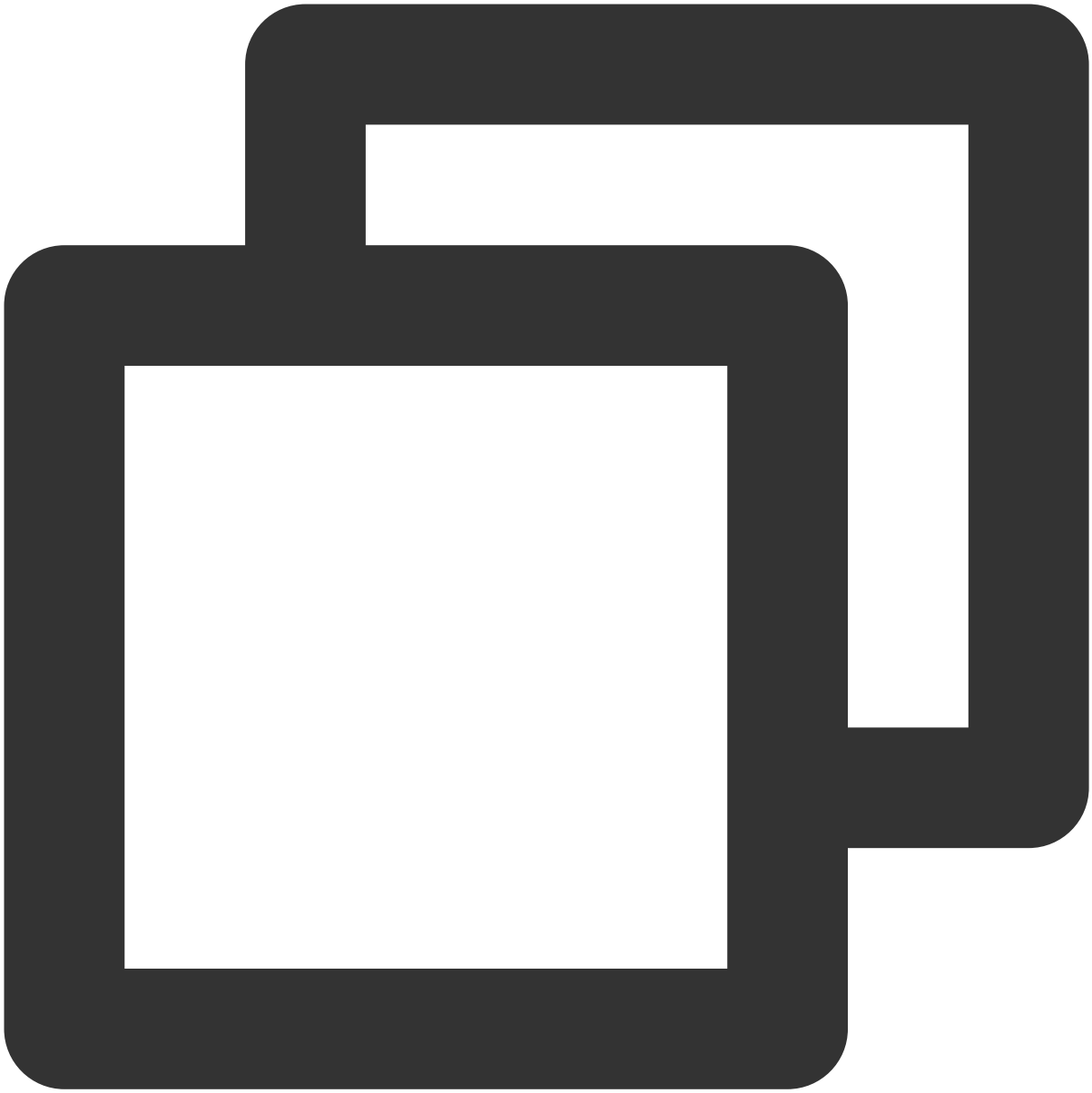
请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:35:28 GMT
Content-Type: image/jpeg
Cache-Control: max-age=86400
Content-Disposition: attachment; filename=example.jpg
```

```
x-cos-meta-example-field: example-value
x-cos-acl: public-read
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close



[Object Content]
```

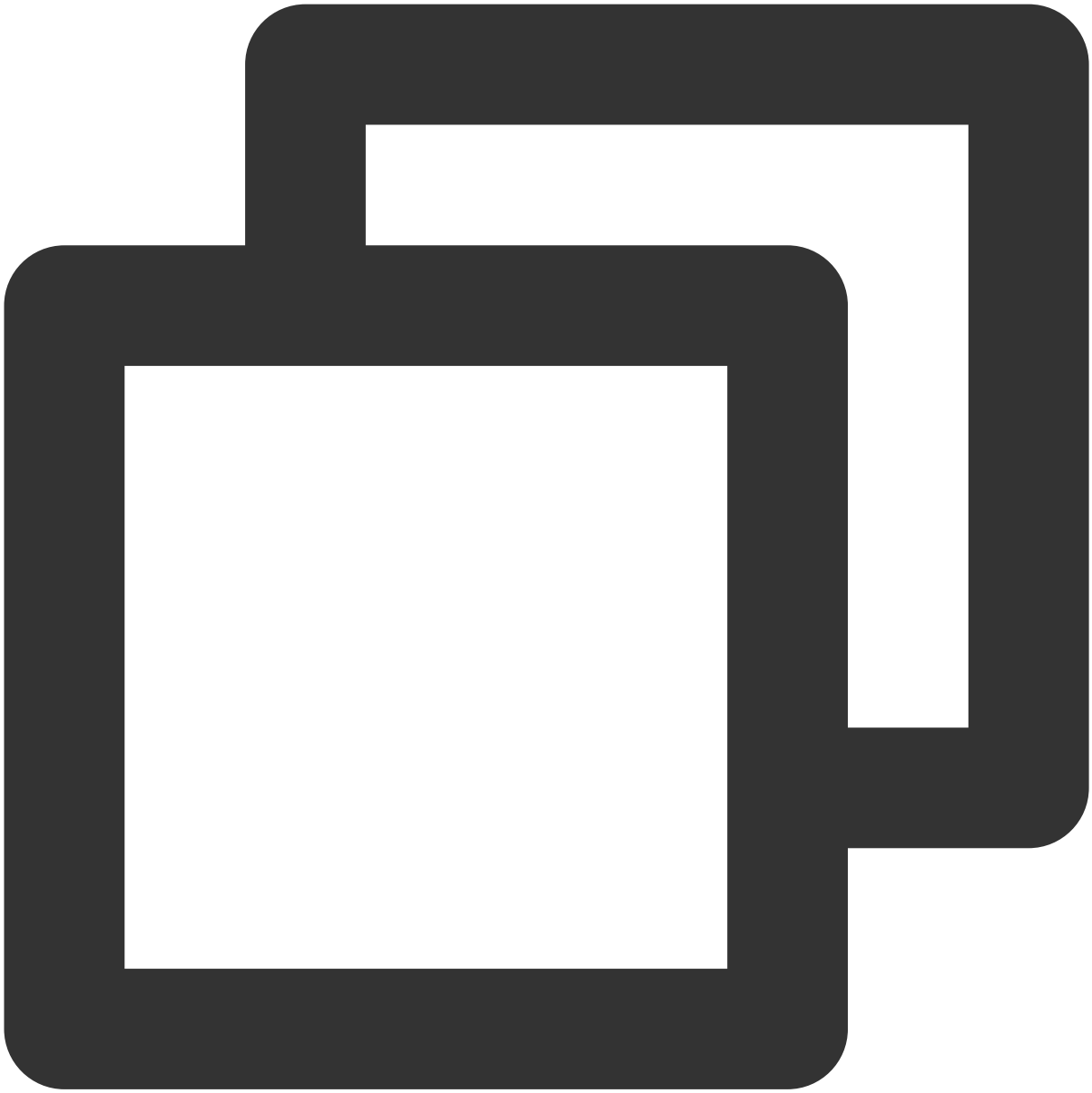
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:35:28 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNkZTBfZjhjMDBiMDlfNzdmN18xMGFi****
```

案例三：使用服务端加密 SSE-COS

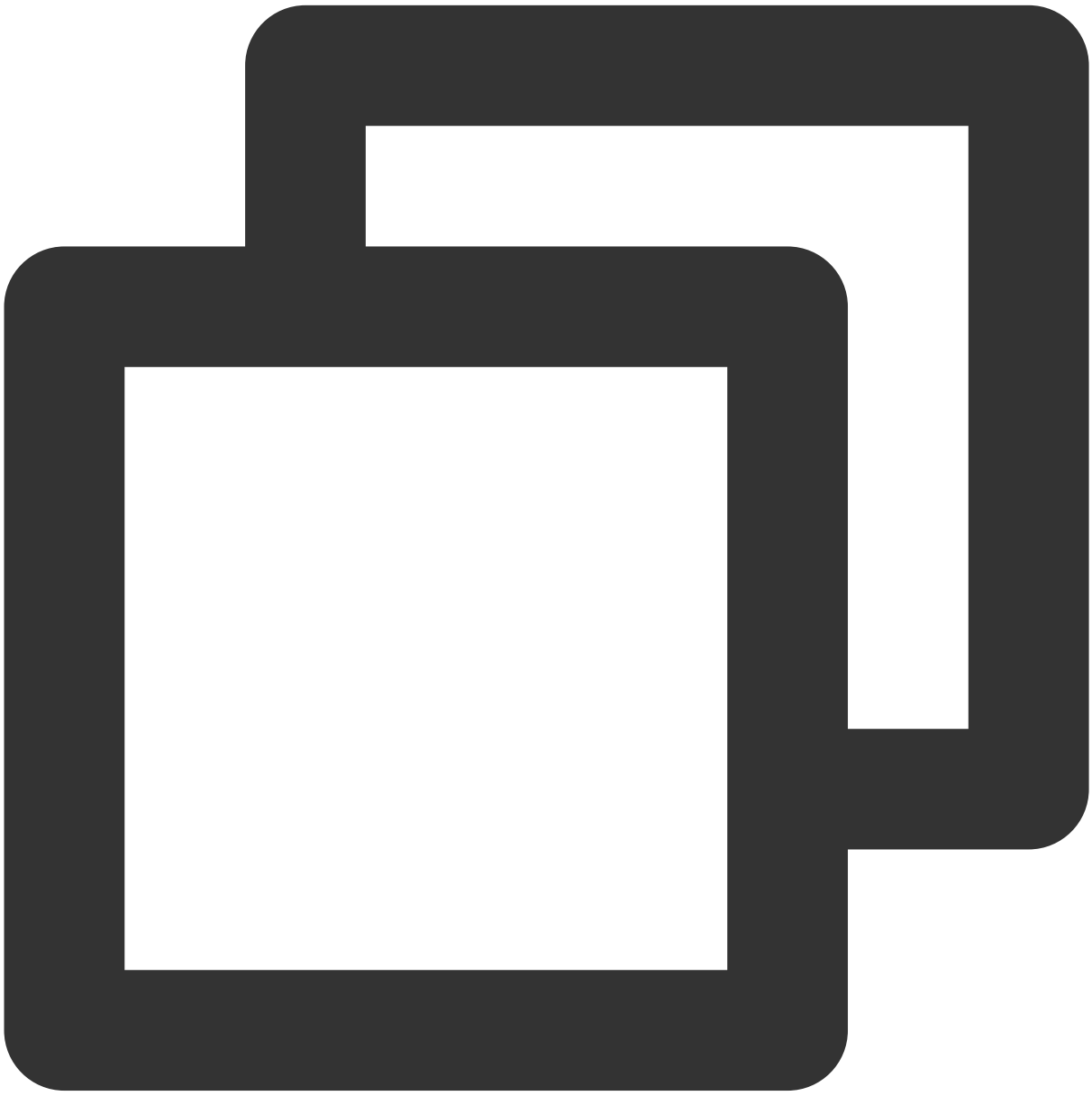
请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:35:49 GMT
Content-Type: image/jpeg
x-cos-server-side-encryption: AES256
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

```
[Object Content]
```

响应

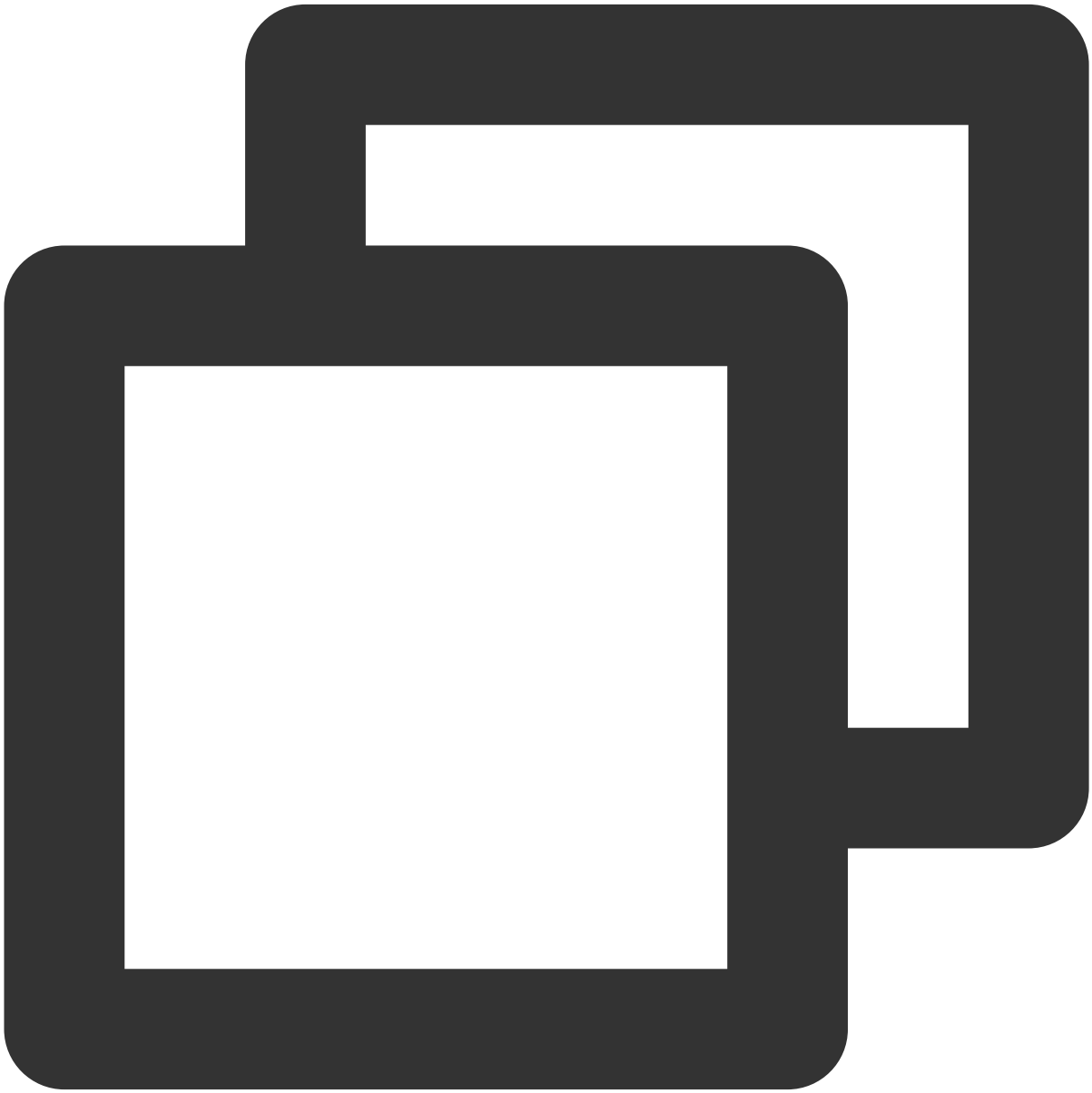


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:35:49 GMT
```

```
ETag: "ee8de918d05640145b18f70f4c3aa602"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749656679157681890
x-cos-request-id: NWU5MDNkZjVfYzVjNzJhMDlfMjVhNzNfMWMy****
x-cos-server-side-encryption: AES256
```

案例四：使用服务端加密 SSE-KMS

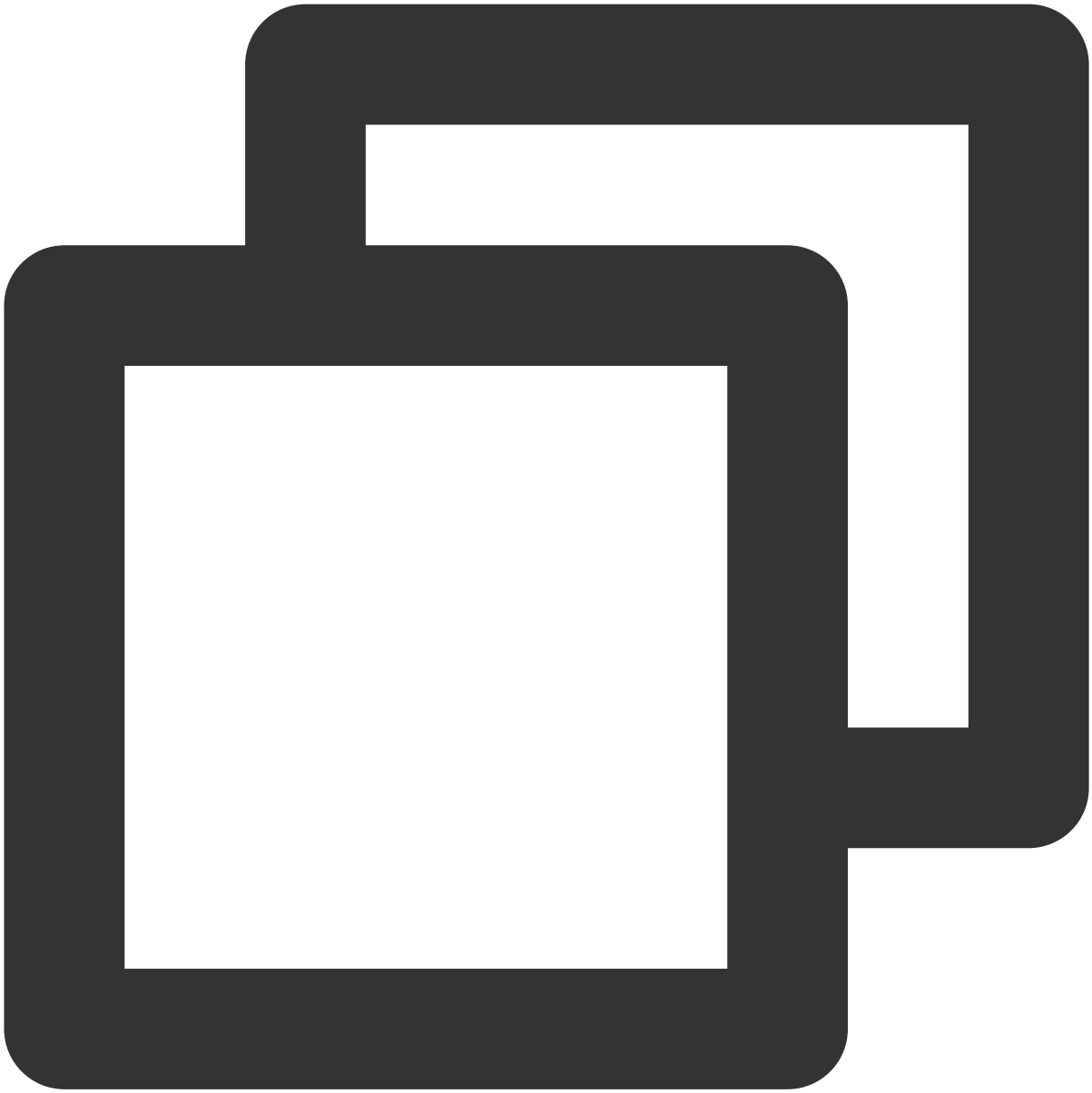
请求



```
PUT /exampleobject HTTP/1.1
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:00 GMT
Content-Type: image/jpeg
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
x-cos-server-side-encryption-context: eyJhdXRob3IiOiJmeXNudGlhbiIsImNvbXBhbnkiOiJUZ
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close



[Object Content]
```

响应

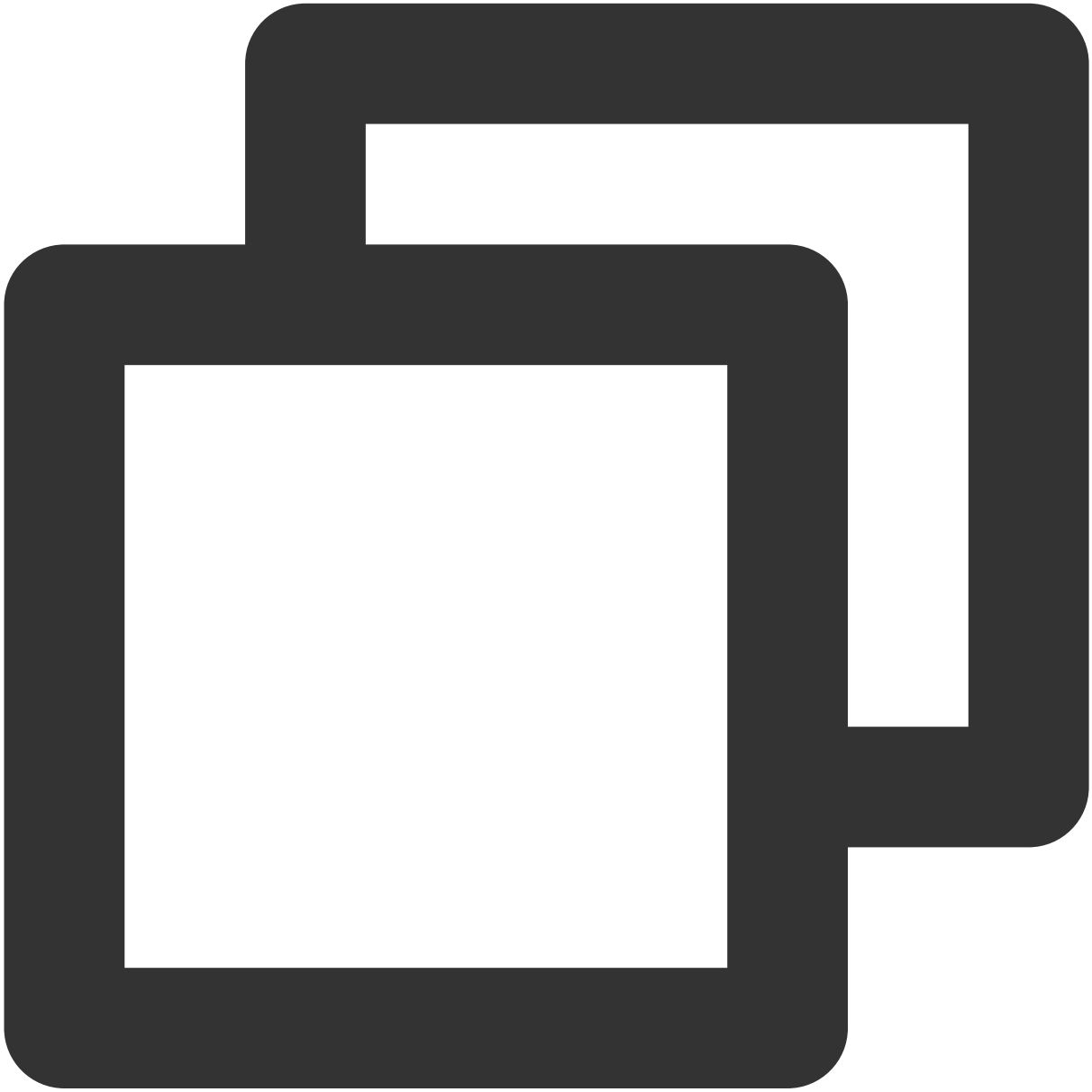


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:36:01 GMT
ETag: "840af7c921f4b3230049af8663145bd0"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNlMDFfOThjMjJhMDlfMjhhM18xNTlm****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
```

版权所有：腾讯云计算（北京）有限责任公司

案例五：使用服务端加密 SSE-C

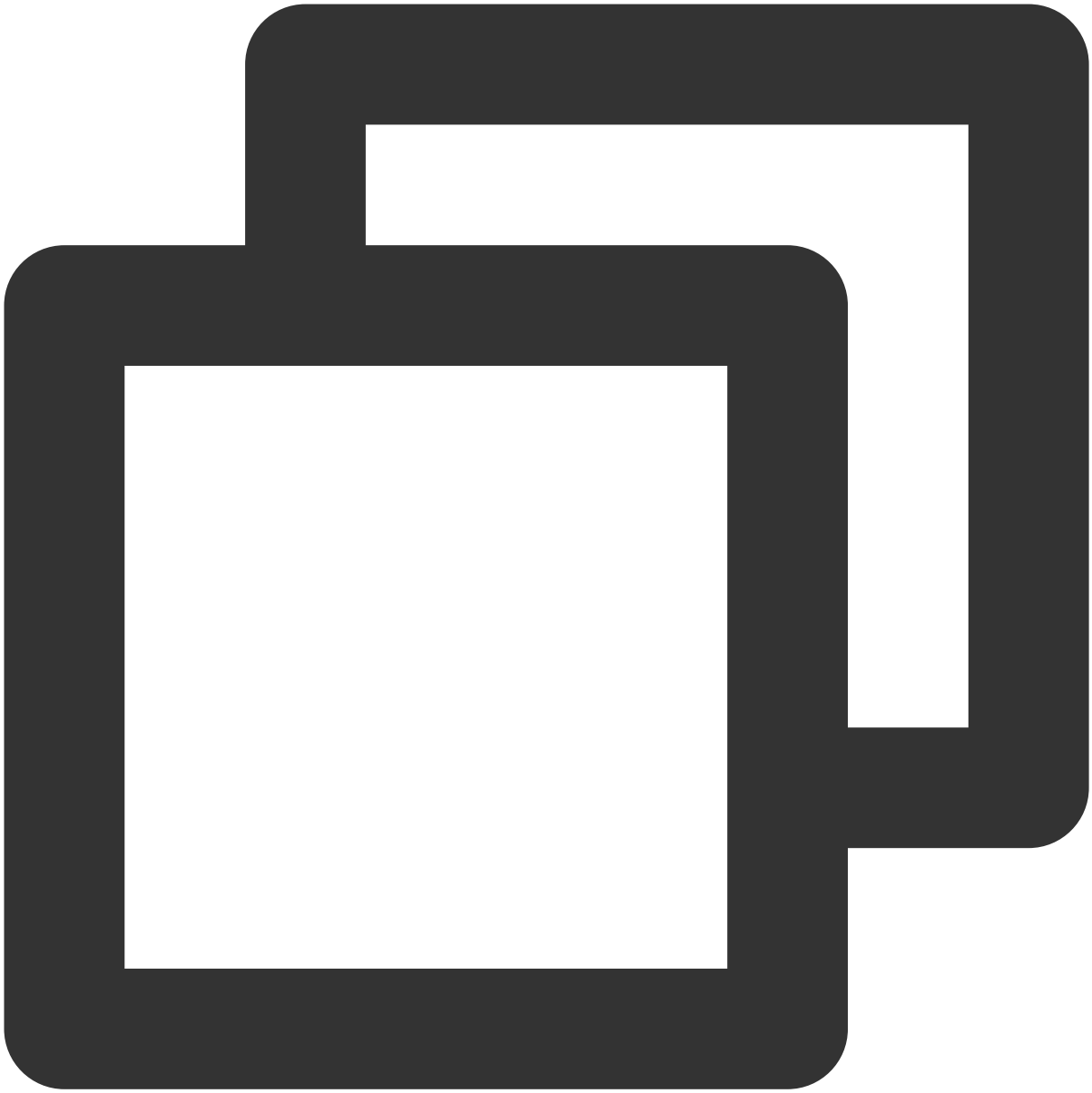
请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:12 GMT
Content-Type: image/jpeg
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNE
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
```

```
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close



[Object Content]
```

响应

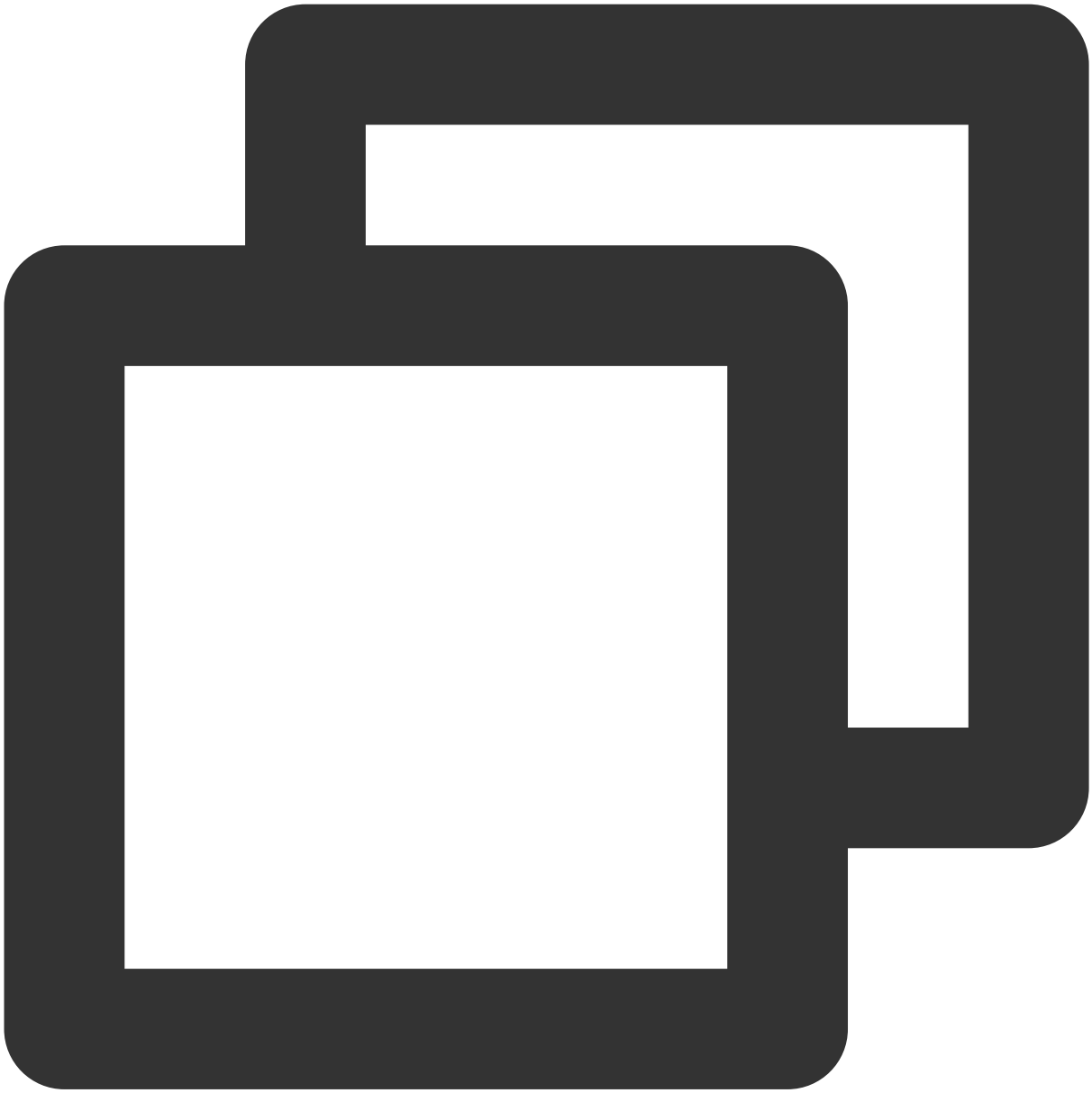


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:36:13 GMT
ETag: "582d9105f71525f3c161984bc005efb5"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNlMGNfZTFjODJhMDlfMzVlMDFfZTk1****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
```

版权所有：腾讯云计算（北京）有限责任公司

案例六：启用版本控制

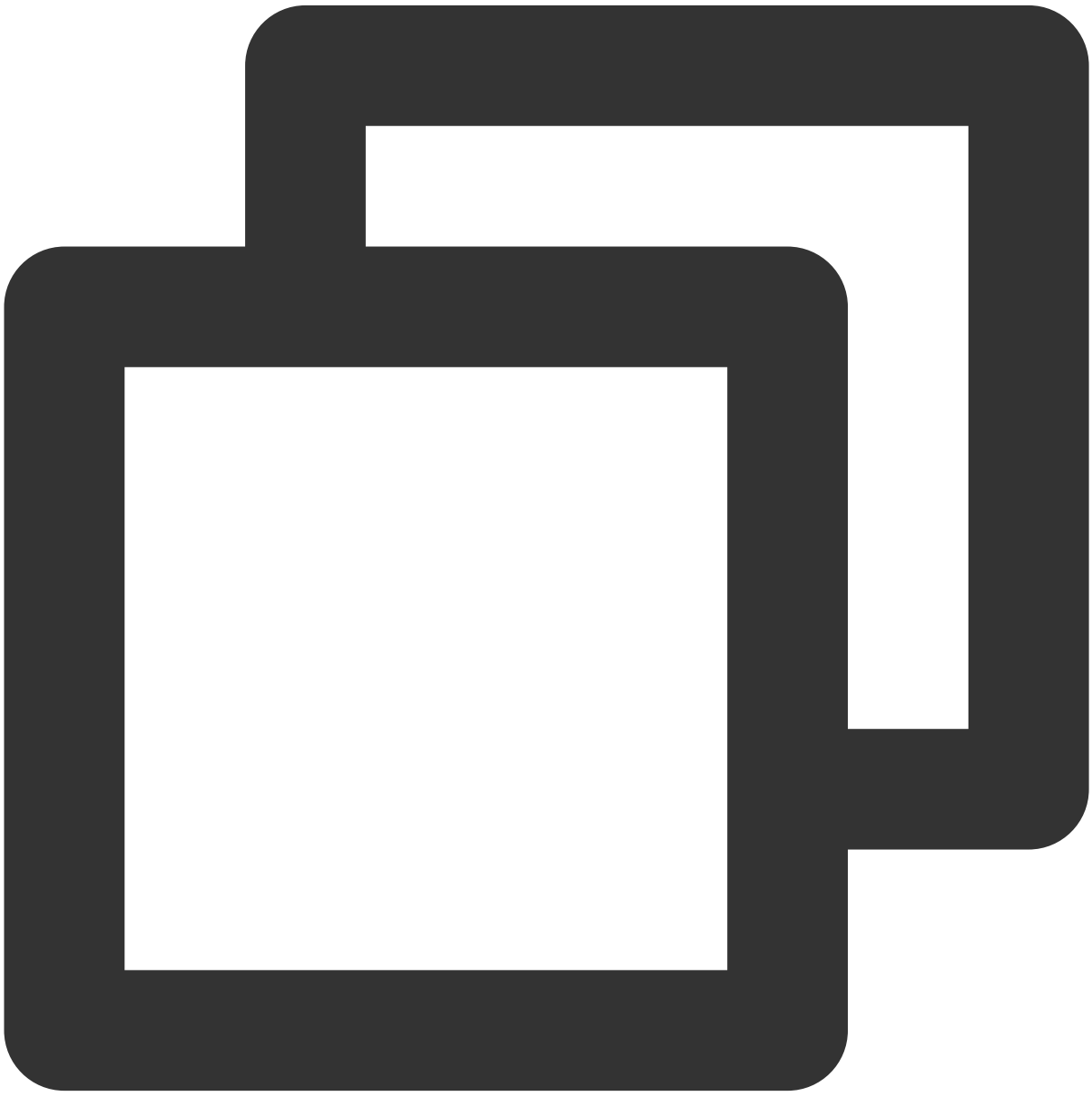
请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:34 GMT
Content-Type: image/jpeg
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
```

```
Connection: close



[Object Content]
```

响应

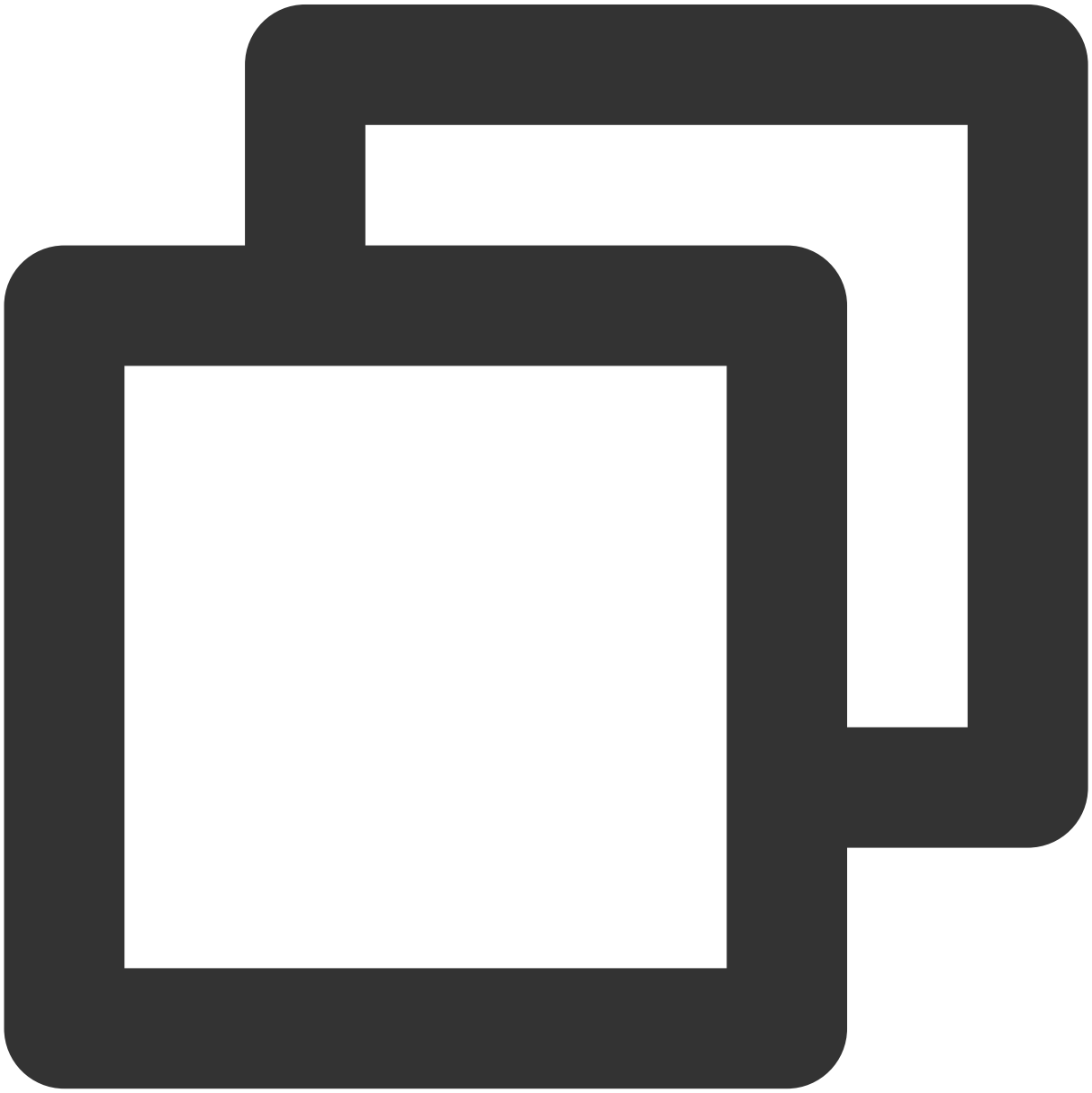


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
```

```
Date: Fri, 10 Apr 2020 09:36:35 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNlMjNfMThiODJhMDlfNGQ1OF8xMWY4****
x-cos-version-id: MTg0NDUxNTc1NjIzMTQ1MDAwODg
```

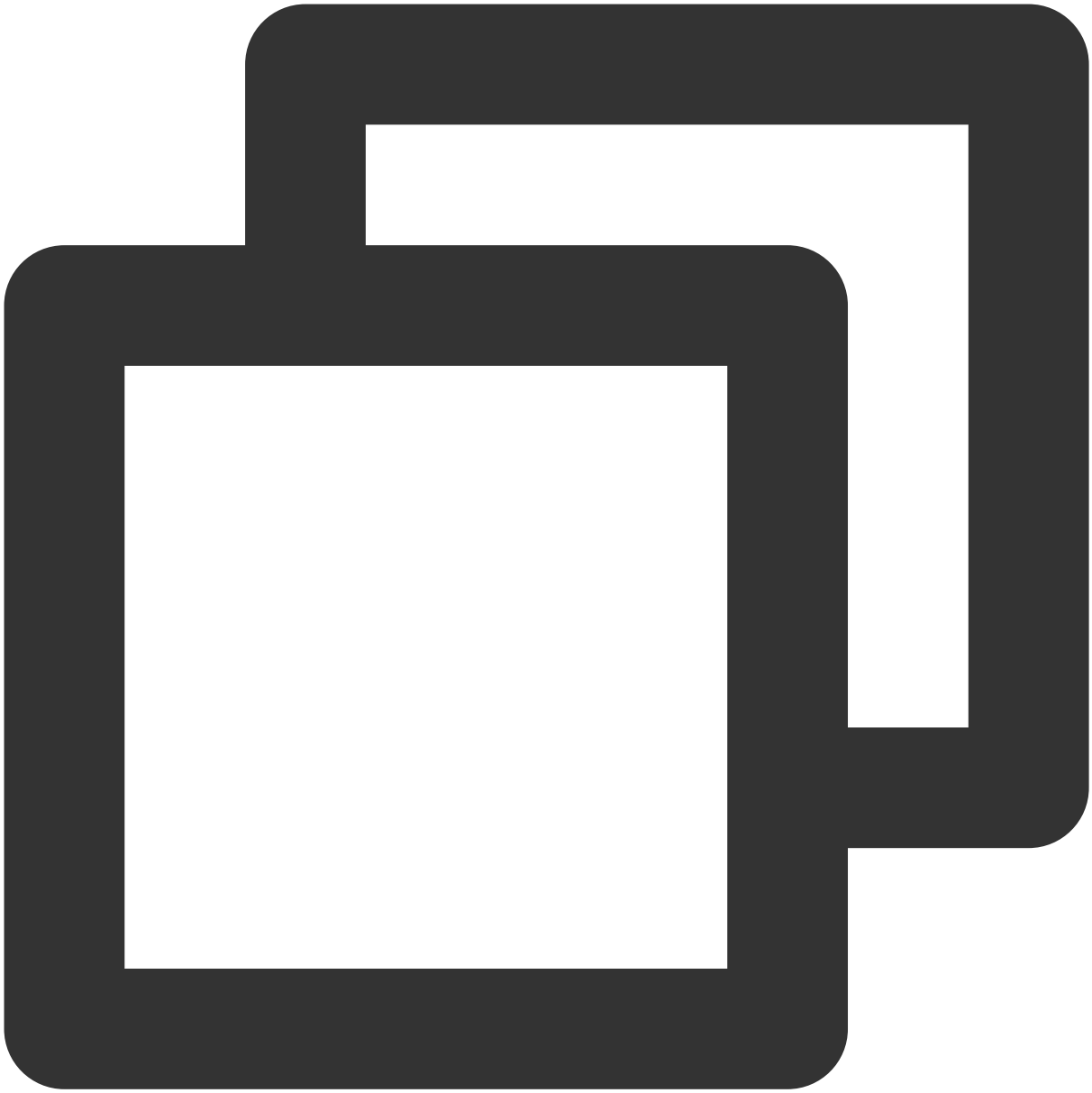
案例七：暂停版本控制

请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:37:07 GMT
Content-Type: image/jpeg
Content-Length: 16
Content-MD5: 7o3pGNBWQBRbGPcPTDqmAg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close



[Object Content]
```

响应

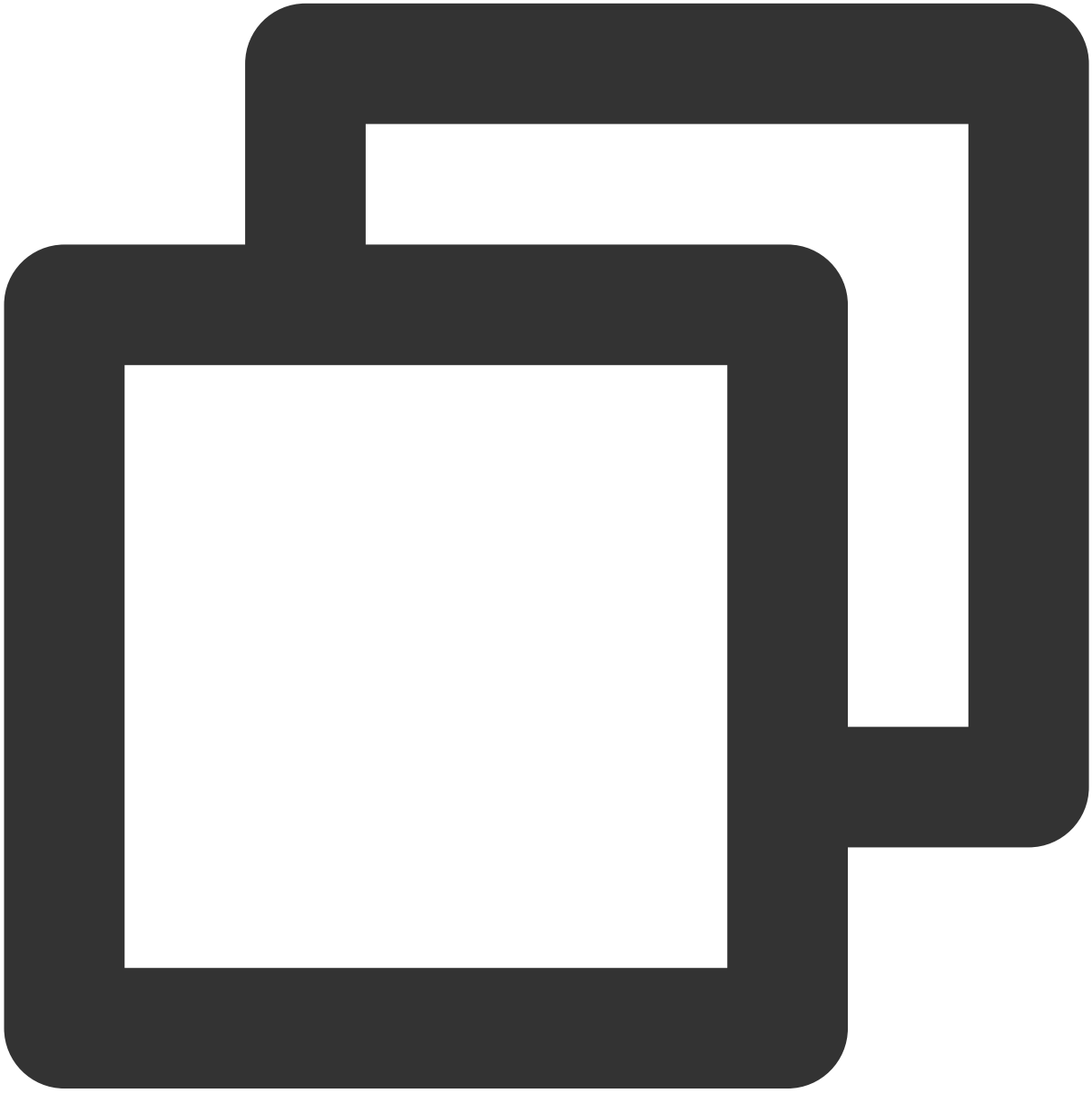


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:37:07 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNlNDNfZTZjNzJhMDlfMmYwMDlfMTVi****
```

案例八：使用 chunked 传输编码分块传输

本案例请求中使用 Transfer-Encoding: chunked 编码，案例描述的是 HTTP 请求中的原始数据，在使用过程中根据不同语言和库将有不同的调用方法，请开发者查阅语言和库的相关文档。

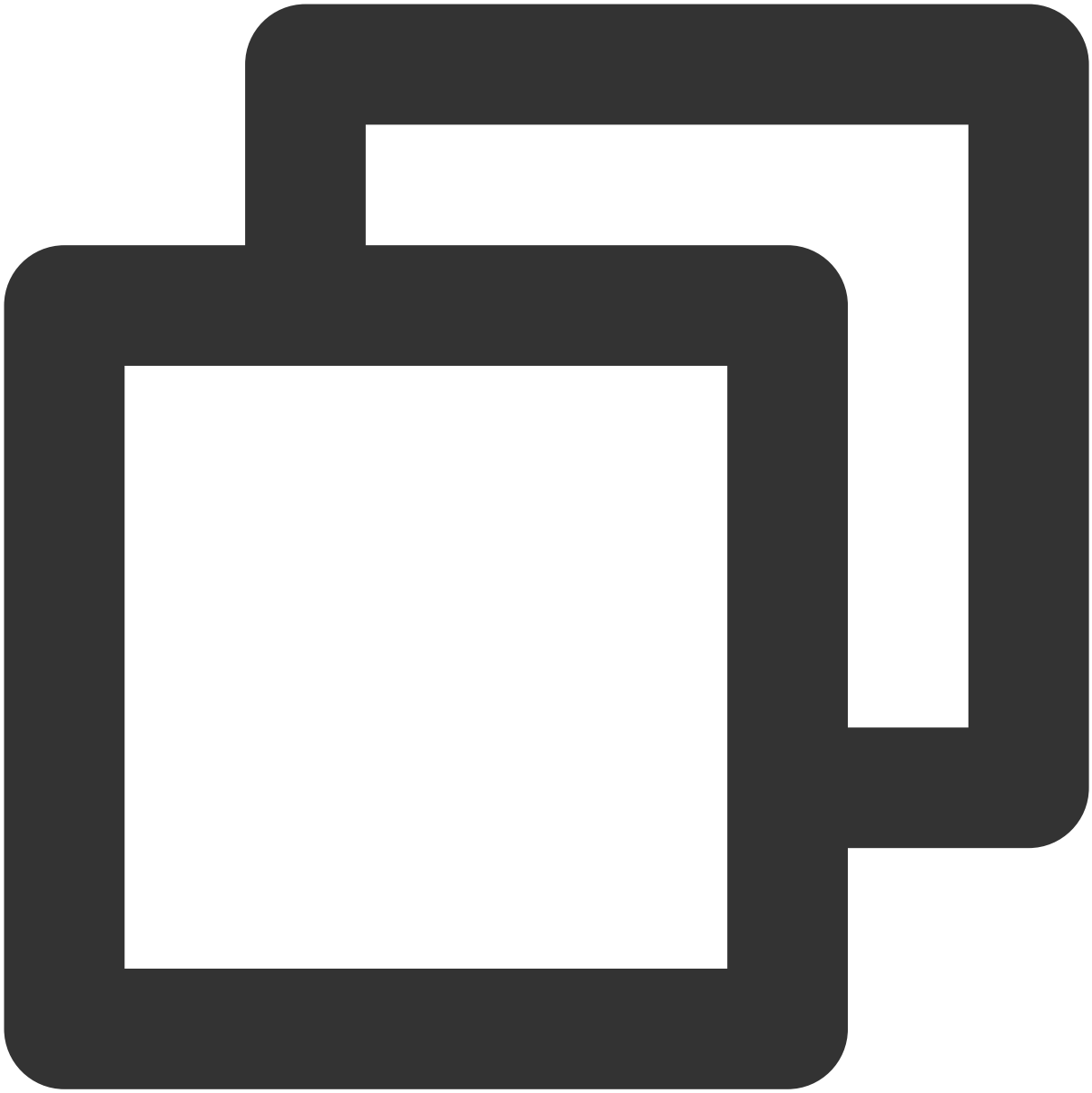
请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 08 Aug 2019 09:15:29 GMT
Content-Type: text/plain
Transfer-Encoding: chunked
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
```

```
Connection: close



11
[Chunked Content]
b
[2nd chunk]
b
[3rd chunk]
b
[4th chunk]
0
```

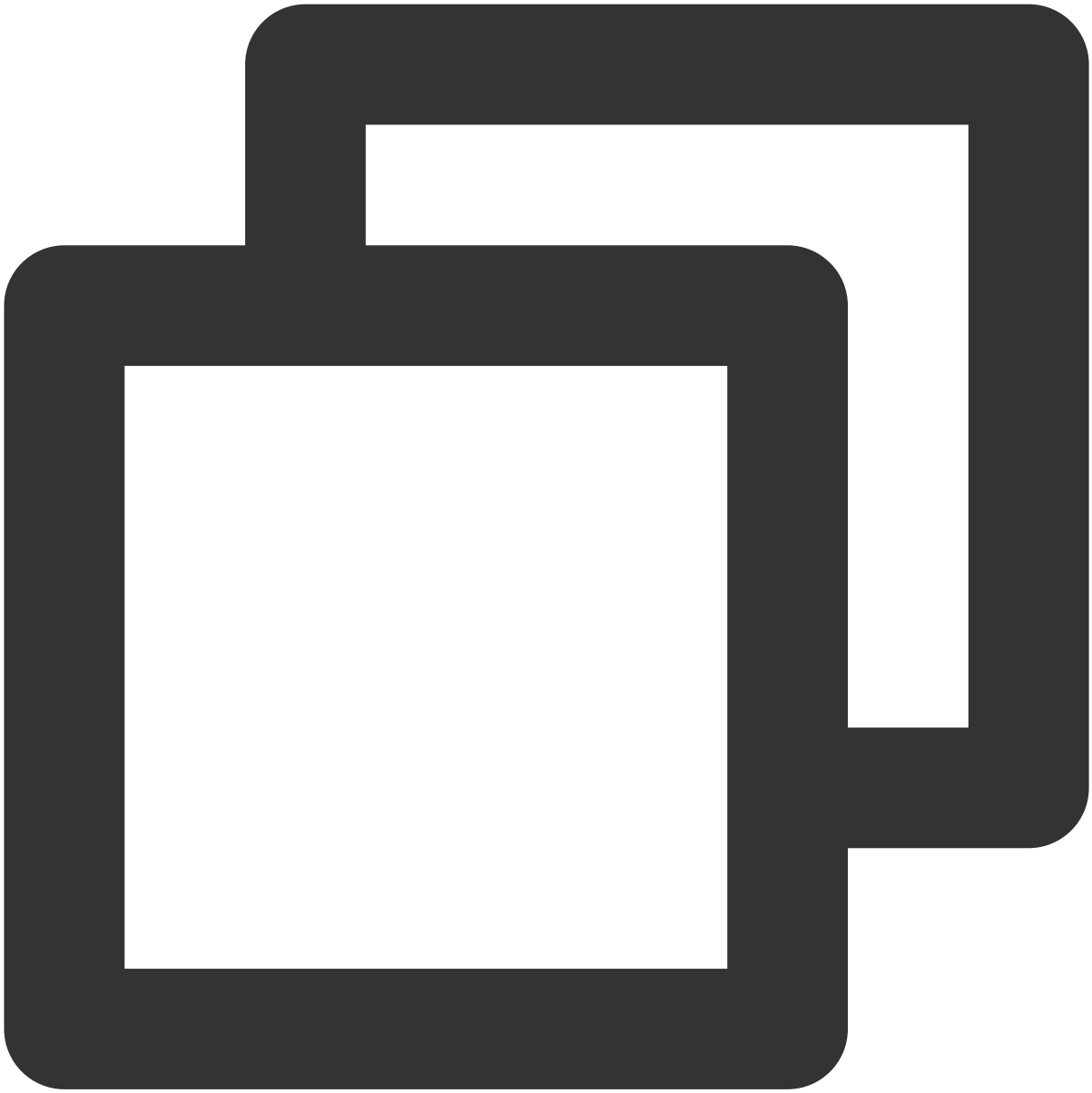
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Thu, 08 Aug 2019 09:15:29 GMT
ETag: "aa488bb80185a6be87f4a7b936a80752"
Server: tencent-cos
x-cos-hash-crc64ecma: 7188322482464764960
x-cos-request-id: NWQ0YmU4MzFfNzFiNDBiMDlfMWJhYTlfMTY2Njll****
```

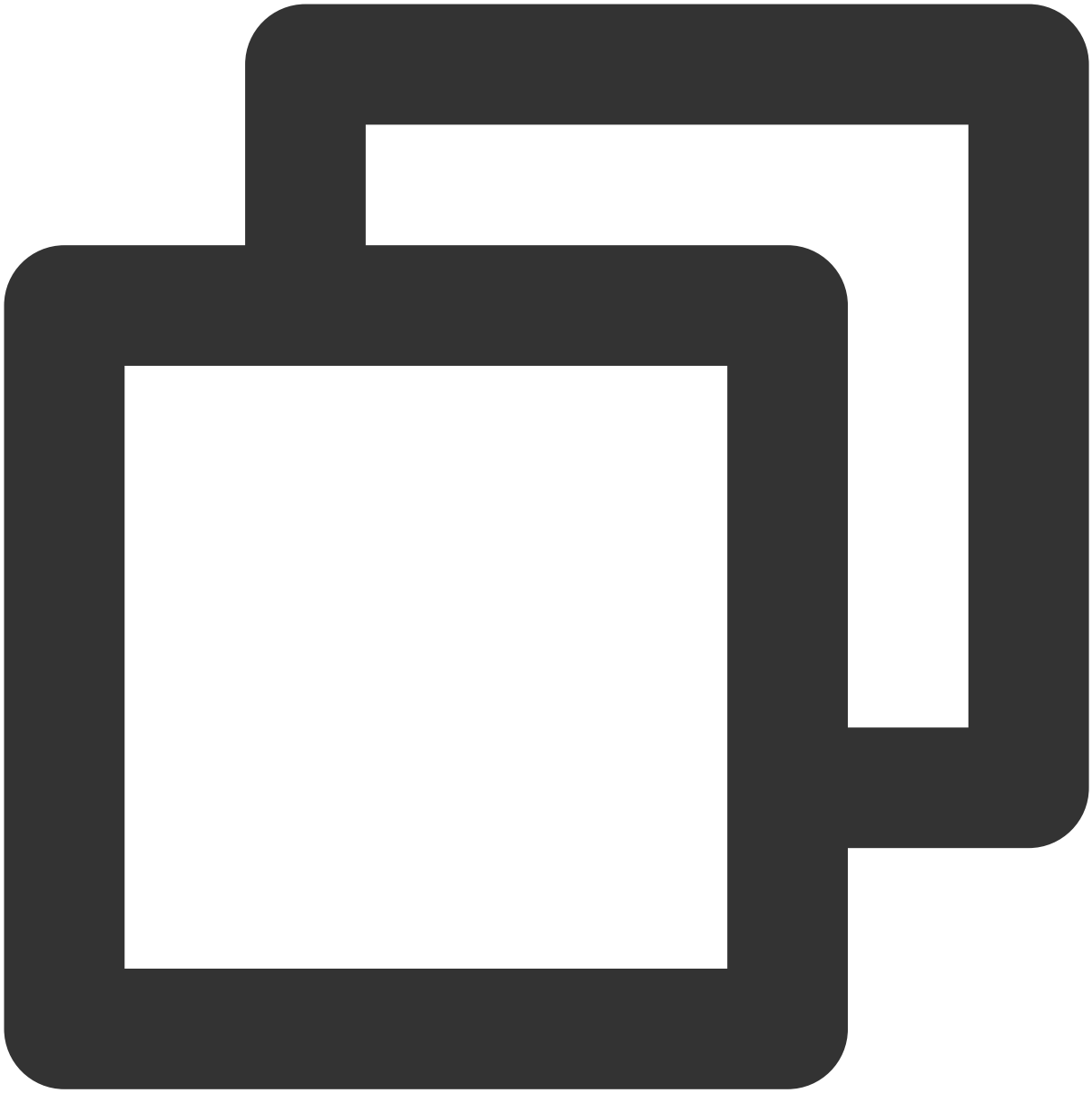
案例九：设置对象锁定

请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
x-cos-object-lock-retain-until-date: 2022-12-30T10:30:09.000Z
x-cos-object-lock-mode: COMPLIANCE

[Object Content]
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 10 Apr 2020 09:35:05 GMT
ETag: "ee8de918d05640145b18*****"
Server: tencent-cos
x-cos-hash-crc64ecma: 167495656791****
x-cos-request-id: NWU5MDNkYzlfNjRiODJhMDlfMzFmYz********
```

PUT Object - Copy

最近更新时间：2023-12-21 14:31:19

功能描述

PUT Object - Copy 接口请求创建一个已存在 COS 的对象的副本，即将一个对象从源路径（对象键）复制到目标路径（对象键）。建议对象大小为1M到5G，超过5G的对象请使用分块上传 Upload Part - Copy。
在复制过程中，可以指定元数据的处理方式。默认情况下元数据将被复制到目标对象中，用户也可以选择不复制源对象的元数据信息，而在此接口请求中指定新的元数据信息。但是，除非在此接口请求中显式指定存储类型、访问控制列表（ACL）和服务端加密（SSE），否则无论何种处理方式，目标对象均保持为标准存储，继承（default）目标存储桶的 ACL 且不会使用 SSE。
用户可以通过此接口实现对象移动、重命名、修改对象元数据和创建副本。
该 API 的请求者需要对被复制对象有读取权限，或者被复制对象向所有人开放了读取权限（公有读），且需要对目标存储桶有写入权限。

注意

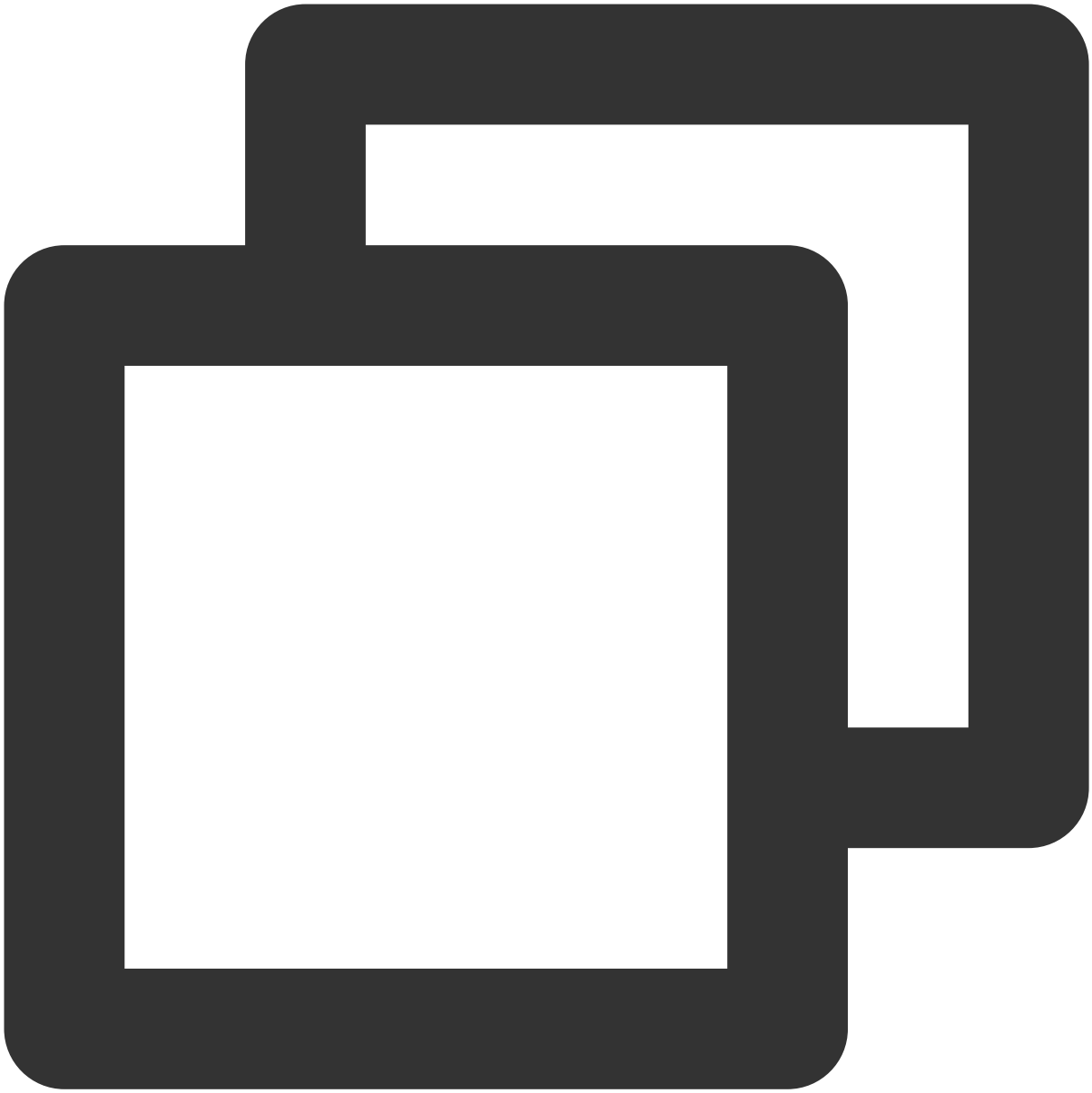
当 COS 收到复制请求或 COS 正在复制对象时可能会返回错误。如果在复制操作开始之前发生错误，则会收到标准的错误返回。如果在复制操作执行期间发生错误，则依然会返回 HTTP 200 OK，并将错误作为响应体返回。这意味着 HTTP 200 OK 响应既可以包含成功也可以包含错误，在使用此接口时应当进一步根据响应体的内容来判断复制请求的成功与失败并正确的处理结果。
标准存储（多 AZ）类型目前仅支持复制为标准存储（多 AZ）类型，不支持复制为标准存储、低频和归档存储类型。
低频存储（多 AZ）类型目前仅支持复制为低频存储（多 AZ）类型，不支持复制为标准存储、低频和归档存储类型。
调用 PUT Object - Copy 等同于在源存储桶请求一次 GET Object，在目标存储桶请求一次 PUT Object。请注意该操作产生的计量计费，例如读写请求、存储容量、跨地域复制流量、工作流产生的数据处理和审核次数等费用。

版本控制

如果源对象所在存储桶启用了版本控制，则默认复制源对象的最新版本，可以在请求头部 x-cos-copy-source 中指定 versionId 参数来复制指定版本。
如果对目标存储桶启用版本控制，对象存储将自动为目标对象生成唯一的版本 ID。

请求

请求示例



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
x-cos-copy-source: <SourceBucketName-SourceAPPID>.cos.<SourceRegion>.myqcloud.com/<
Content-Length: 0
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
x-cos-copy-source	源对象的 URL，其中对象键需经过 URLEncode，可以通过 versionId 参数指定源对象的版本，例如：`sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8%85%BE%E8%AE%AF%E4%BA%91.jpg` 或 `sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8%85%BE%E8%AE%AF%E4%BA%91.jpg?versionId=MTg0NDUxNzYzMDc0NDMzNDExOTc`	string	是
x-cos-metadata-directive	是否复制源对象的元数据信息，枚举值：Copy，Replaced，默认为 Copy。 如果标记为 Copy，则复制源对象的元数据信息 如果标记为 Replaced，则按本次请求的请求头中的元数据信息作为目标对象的元数据信息 当目标对象和源对象为同一对象时，即用户试图修改元数据时，则标记必须为 Replaced	Enum	否
x-cos-copy-source-If-Modified-Since	当对象在指定时间后被修改，则执行复制操作，否则返回 HTTP 状态码为412（Precondition Failed）	string	否
x-cos-copy-source-If-Unmodified-Since	当对象在指定时间后未被修改，则执行复制操作，否则返回 HTTP 状态码为412（Precondition Failed）	string	否
x-cos-copy-source-If-Match	当对象的 ETag 与指定的值一致，则执行复制操作，否则返回 HTTP 状态码为412（Precondition Failed）	string	否
x-cos-copy-source-If-None-Match	当对象的 ETag 与指定的值不一致，则执行复制操作，否则返回 HTTP 状态码为412（Precondition Failed）	string	否
x-cos-storage-	目标对象的存储类型。枚举值请参见 存储类型 文档，例如	Enum	否

class	INTELLIGENT_TIERING，MAZ_INTELLIGENT_TIERING，STANDARD_IA，ARCHIVE，DEEP_ARCHIVE。默认值：STANDARD		
x-cos-tagging	对象的标签集合，最多可设置10个标签（例如，Key1=Value1&Key2=Value2）。 标签集合中的 Key 和 Value 必须先进行 URL 编码	string	否
x-cos-tagging-directive	是否复制源对象的标签信息，枚举值：Copy，Replaced，默认为 Copy 如果标记为 Copy，则复制源对象的标签信息 如果标记为 Replaced，则按本次请求的请求头中的标签信息作为目标对象的标签信息 当目标对象和源对象为同一对象时，即用户试图修改对象标签时，则标记必须为 Replaced	Enum	否

目标对象元数据相关头部

在复制对象时可以通过指定下列请求头部来设置目标对象的元数据信息，此时请求头部 x-cos-metadata-directive 需指定为 Replaced，否则目标对象将使用源对象的元数据信息且不能指定下列任何头部。

名称	描述	类型	是否必选
Cache-Control	RFC 2616 中定义的缓存指令，将作为目标对象元数据保存	string	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为目标对象元数据保存	string	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为目标对象元数据保存	string	否
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型（MIME），此头部用于描述目标对象的内容类型，将作为目标对象元数据保存 例如 `text/html` 或 `image/jpeg`	string	是
Expires	RFC 2616 中定义的缓存失效时间，将作为目标对象元数据保存	string	否
x-cos-meta-*	包括用户自定义元数据头部后缀和用户自定义元数据信息，将作为目标对象元数据保存，大小限制为2KB 例如 x-cos-meta-via: homepage，自定义元数据头部为 x-cos-meta-via，后缀为 via，自定义元数据信息为 homepage **注意：用户自定义元数据信息支持下划线（_），但用户自定义元数据头部后缀不支持下划线，仅支持减号（-）**	string	否

目标对象访问控制列表（ACL）相关头部

在复制对象时可以通过指定下列请求头部来设置目标对象的访问权限：

名称	描述	类型	是否必选
x-cos-acl	定义目标对象的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中对象的预设 ACL 部分，例如 default，private，public-read 等，默认为 default **注意：如果您不需要进行对象 ACL 控制，请设置为 default 或者此项不进行设置，默认继承存储桶权限**	Enum	否
x-cos-grant-read	赋予被授权者读取目标对象的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取目标对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入目标对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-full-control	赋予被授权者操作目标对象的所有权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否

源对象服务端加密（SSE）相关头部

如果源对象使用了服务端加密且加密方式为 SSE-C 时，则需要指定下列请求头部来解密源对象：

名称	描述	类型	是否必选
x-cos-copy-source-server-side-encryption-customer-algorithm	服务端加密算法，目前仅支持 AES256	string	源对象使用 SSE-C 时，此头部是必选项
x-cos-copy-source-server-side-encryption-customer-key	服务端加密密钥的 Base64 编码，例如 `MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=`	string	源对象使用 SSE-C 时，此头部是必选项
x-cos-copy-source-server-	服务端加密密钥的 MD5 哈希值，使用 Base64 编码，例如 `U5L61r7jcwdNvT7frmUG8g==`	string	源对象使用 SSE-C 时，

side-encryption-customer-key-MD5			此头部是必选项

目标对象服务端加密（SSE）相关头部

在复制对象时可以使用服务端加密，请参见 服务端加密专用头部。

请求体

此接口无请求体。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

版本控制相关头部

当指定了源对象的版本 ID 时，将返回下列响应头部：

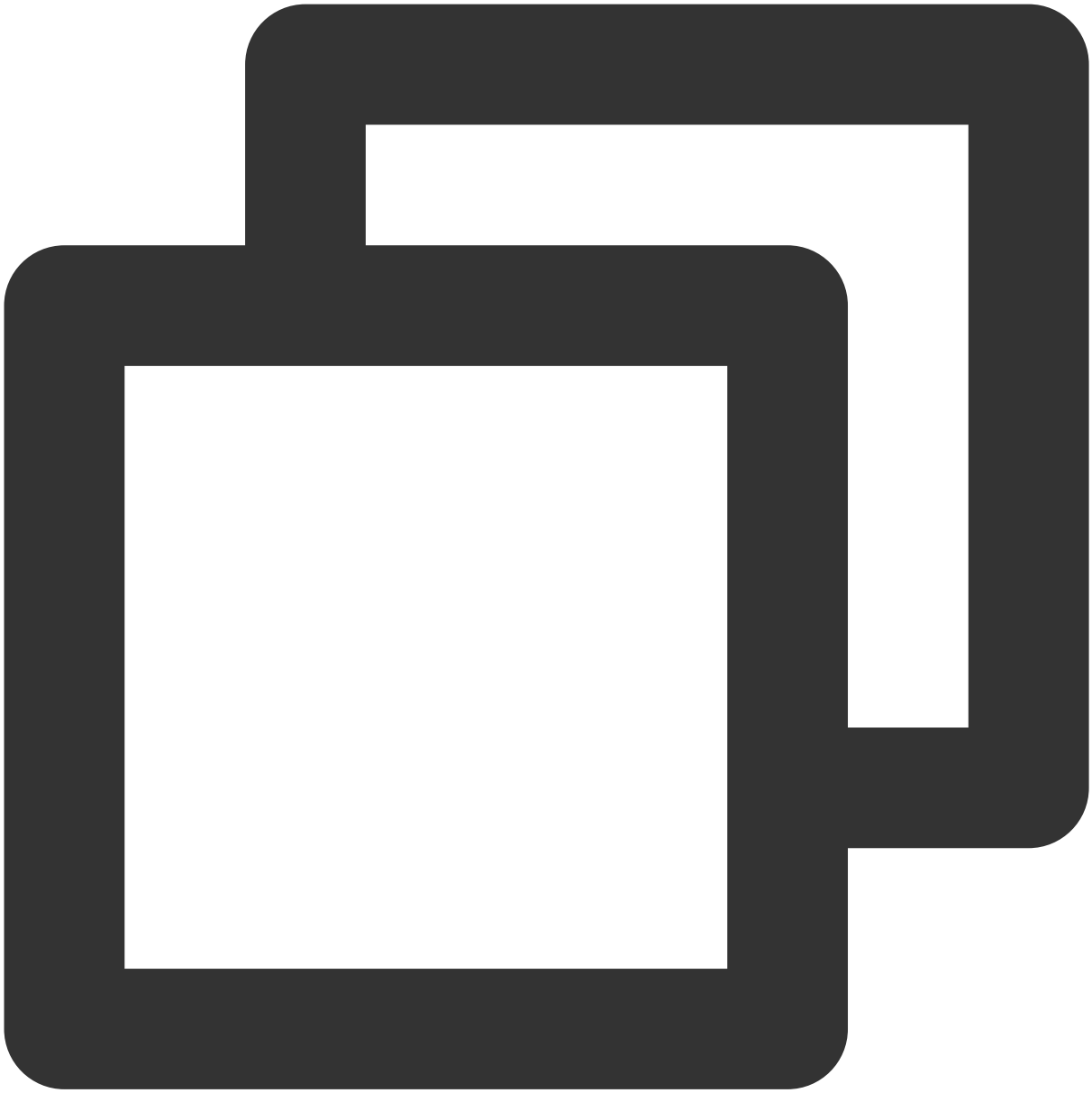
名称	描述	类型
x-cos-copy-source-version-id	源对象的版本 ID	string

服务端加密（SSE）相关头部

如果在复制对象时使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

查询成功，返回 **application/xml** 数据，包含对象复制结果信息。



```
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>string</ETag>
            <CRC64>number</CRC64>
            <LastModified>date</LastModified>
            <VersionId>string</VersionId>
</CopyObjectResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型

CopyObjectResult	无	保存 PUT Object - Copy 结果的所有信息	Container

Container 节点 CopyObjectResult 的内容：

节点名称（关键字）	父节点	描述	类型
ETag	CopyObjectResult	对象的实体标签（Entity Tag），是对象被创建时标识对象内容的信息标签，可用于检查对象的内容是否发生变化例如 `8e0b617ca298a564c3331da28dcb50df`，此头部并不一定返回对象的 MD5 值，而是根据对象上传和加密方式而有所不同	string
CRC64	CopyObjectResult	对象的 CRC64 值，详情请参见 CRC64 校验 文档	number
LastModified	CopyObjectResult	对象最后修改时间，为 ISO8601 格式，例如 `2019-05-24T10:56:40Z`	date
VersionId	CopyObjectResult	对象的版本 ID，仅当目标存储桶启用了版本控制时才返回该元素	string

错误码

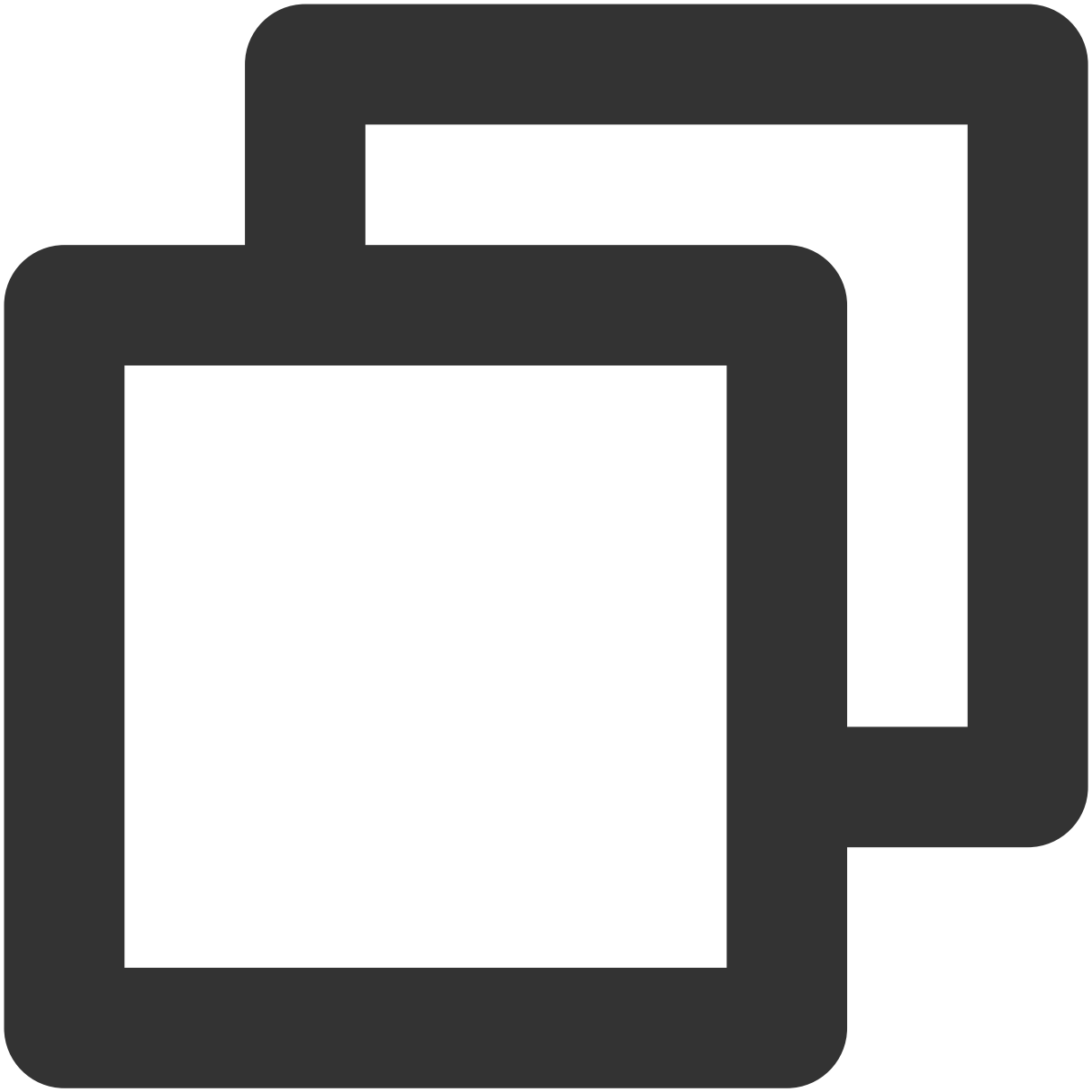
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

此接口响应默认为 Transfer-Encoding: chunked 编码方式，为了方便阅读，本文档实际案例均采用无 Transfer-Encoding 的方式展示，在使用过程中，不同语言和库可以自动处理这种编码形式，请开发者注意识别和处理，更多信息请查阅语言和库的相关文档。

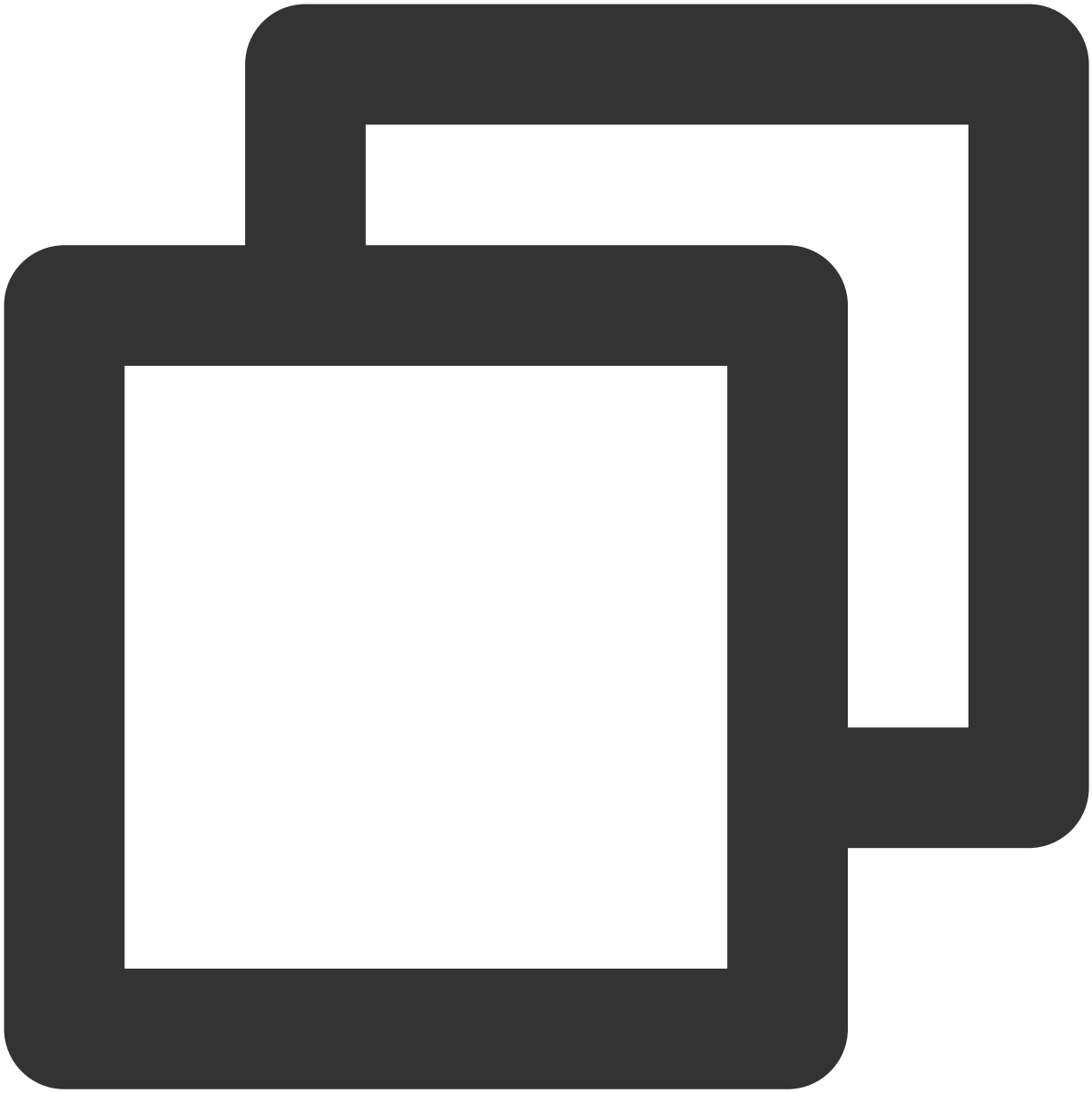
案例一：简单案例

请求



```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:20:30 GMT
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

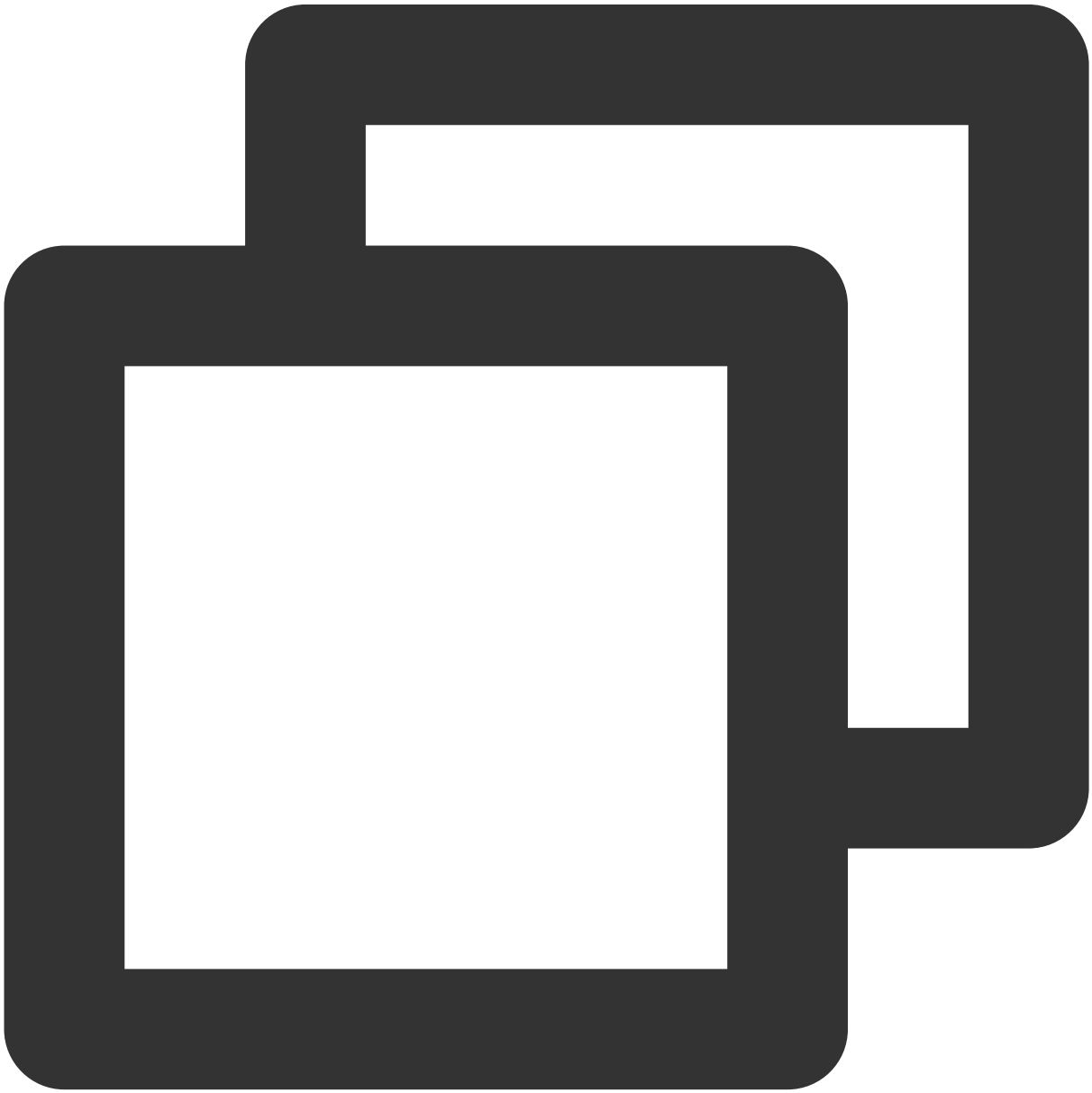


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:20:30 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI4ZWVfNzljMDBiMDlfMWM3MjlfMWQ1****



<?xml version="1.0" encoding="UTF-8"?>
```

```
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:20:30Z</LastModified>
</CopyObjectResult>
```

案例二：复制时替换元数据

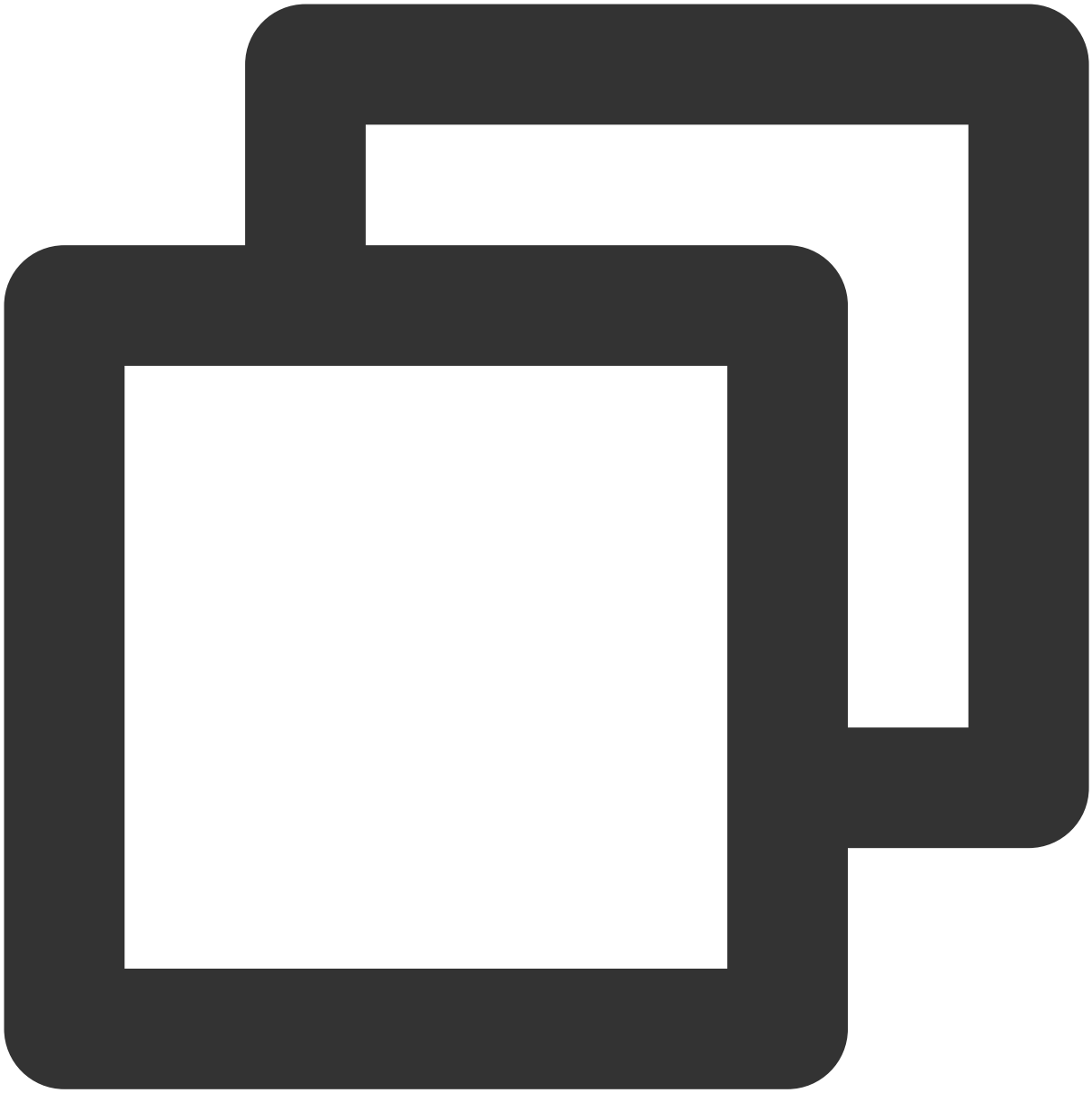
请求



```
PUT /exampleobject HTTP/1.1
```

```
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:20:41 GMT
x-cos-metadata-directive: Replaced
Content-Type: application/octet-stream
Content-Disposition: attachment; filename=example.jpg
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

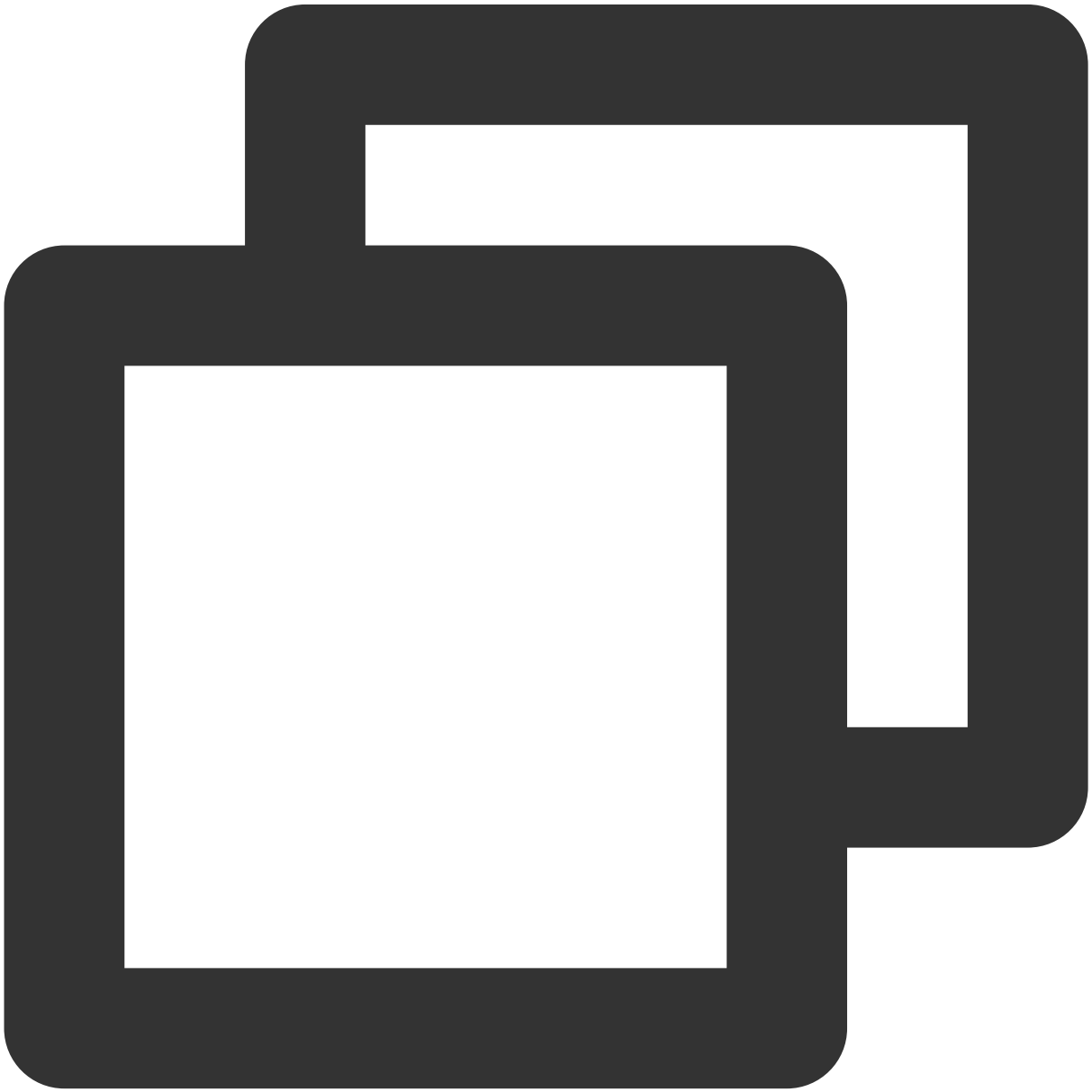


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:20:41 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI4ZjlfYTZjMDBiMDlfN2Y1YV8xYjI4****



<?xml version="1.0" encoding="UTF-8"?>
```

```
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:20:41Z</LastModified>
</CopyObjectResult>
```

案例三：修改对象元数据

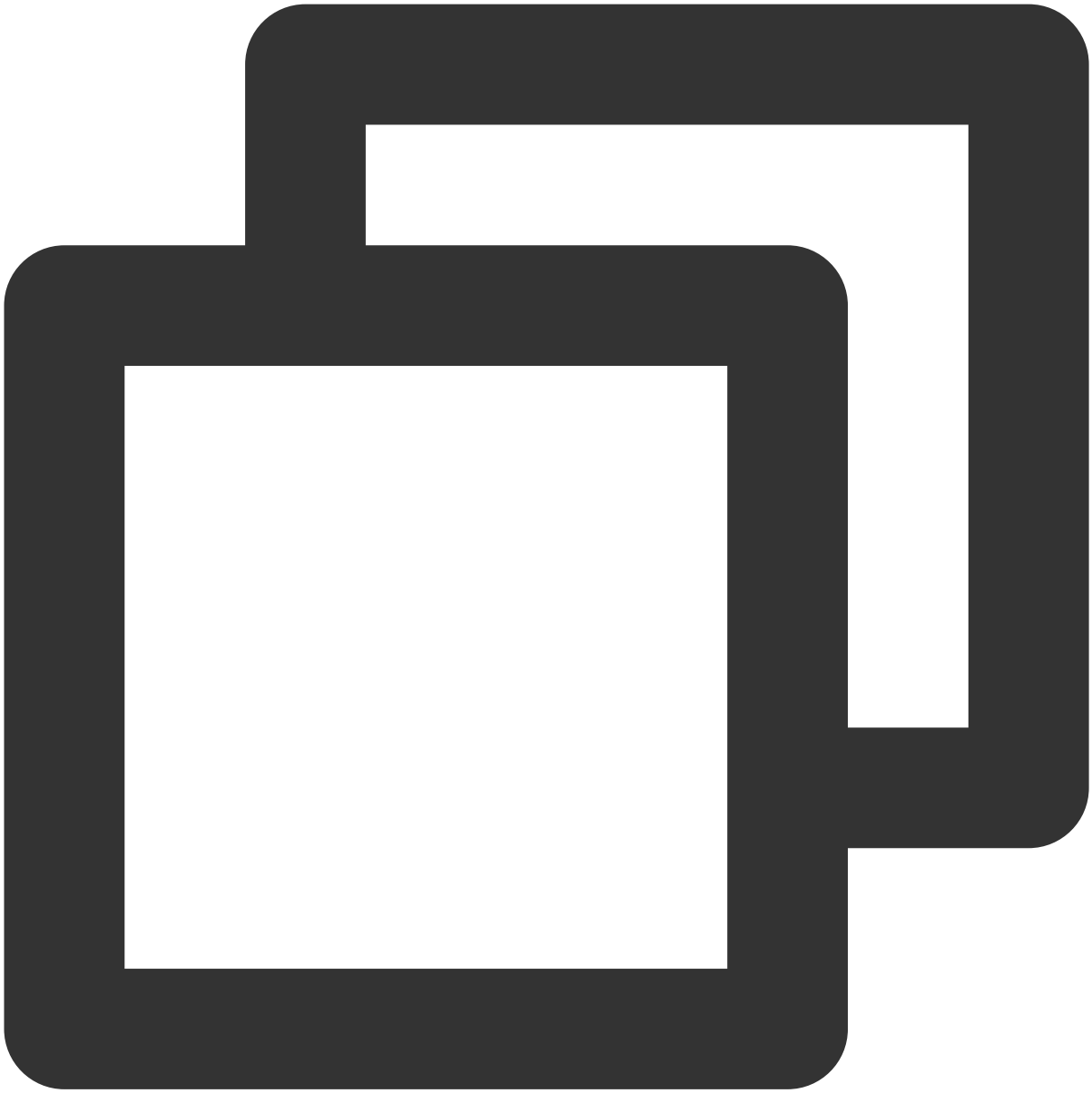
请求



```
PUT /exampleobject HTTP/1.1
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:20:52 GMT
x-cos-metadata-directive: Replaced
Cache-Control: max-age=86400
Content-Type: image/jpeg
x-cos-copy-source: examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobje
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



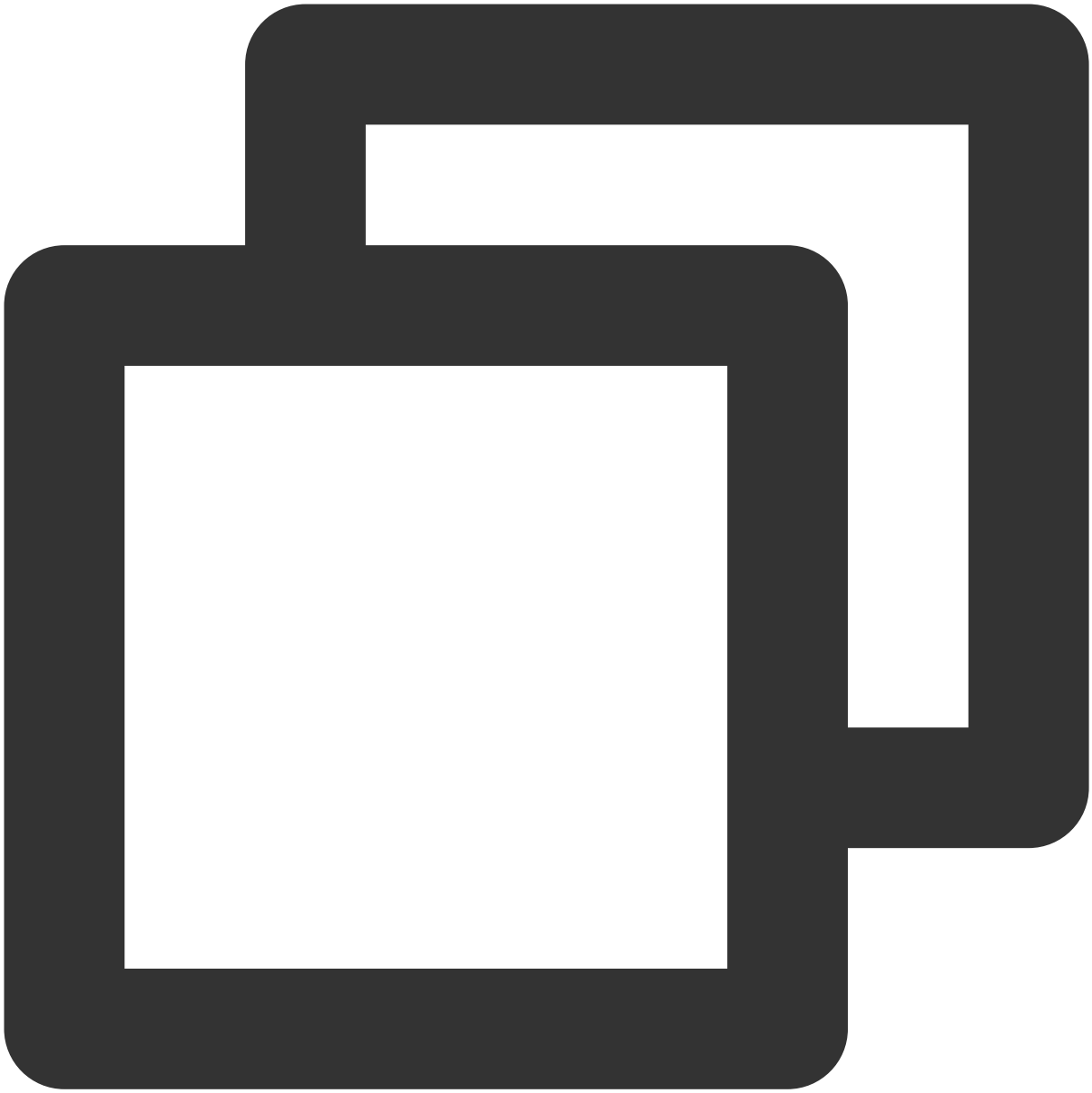
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:20:52 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI5MDRfNmRjMDJhMDlfZGNmYl8yMDVh****



<?xml version="1.0" encoding="UTF-8"?>
```

```
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:20:52Z</LastModified>
</CopyObjectResult>
```

案例四：修改对象存储类型

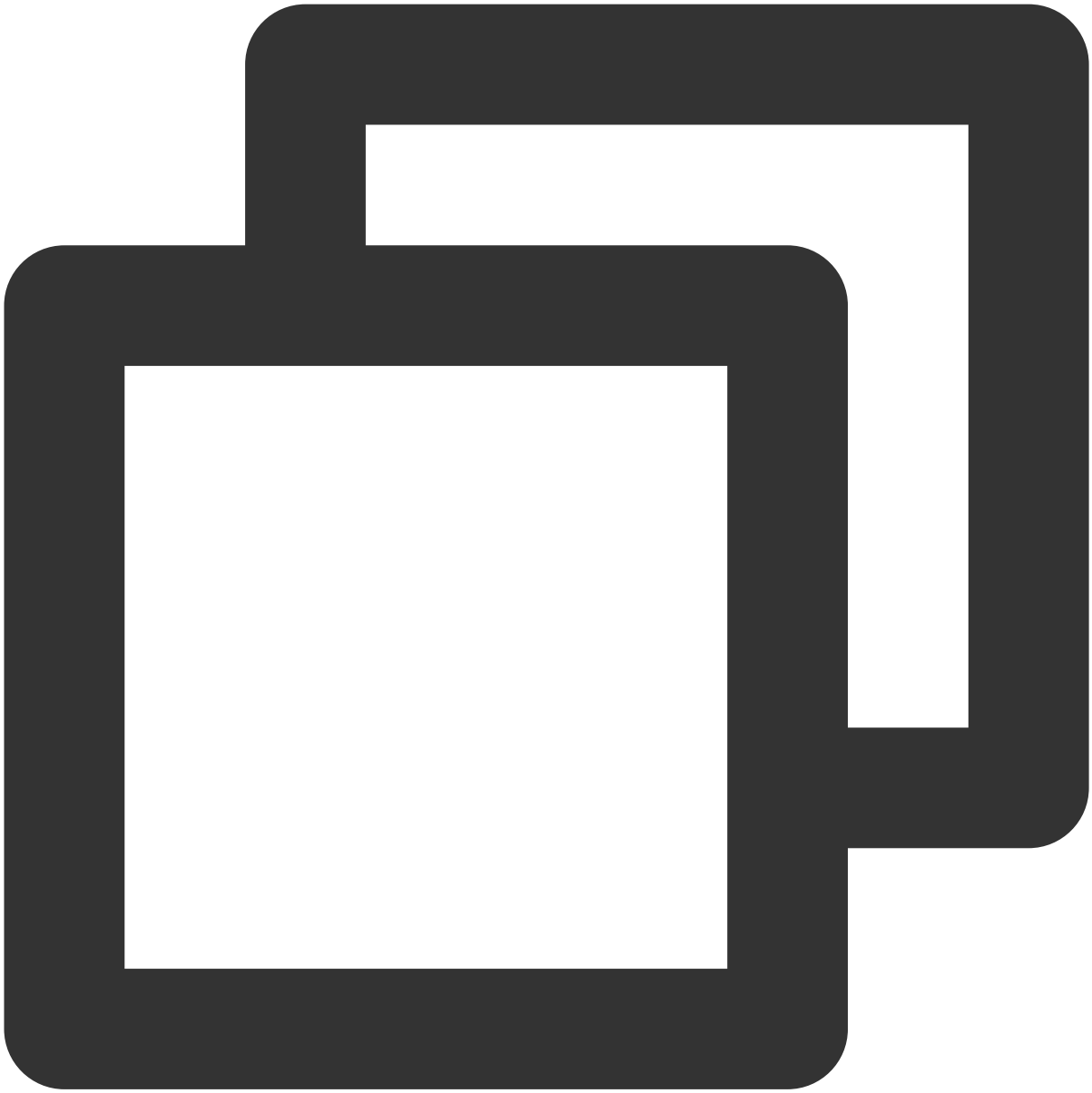
本案例演示将对象从标准存储转换为归档存储，该使用方法也适合标准存储与低频存储之间的互相转换，如果希望将归档存储或深度归档存储的对象转换为其他存储类型，需要首先使用 POST Object restore 将归档存储或深度归档存储的对象回热，才能使用该接口请求转换存储类型。

请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:21:02 GMT
x-cos-metadata-directive: Replaced
x-cos-storage-class: ARCHIVE
x-cos-copy-source: examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobje
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

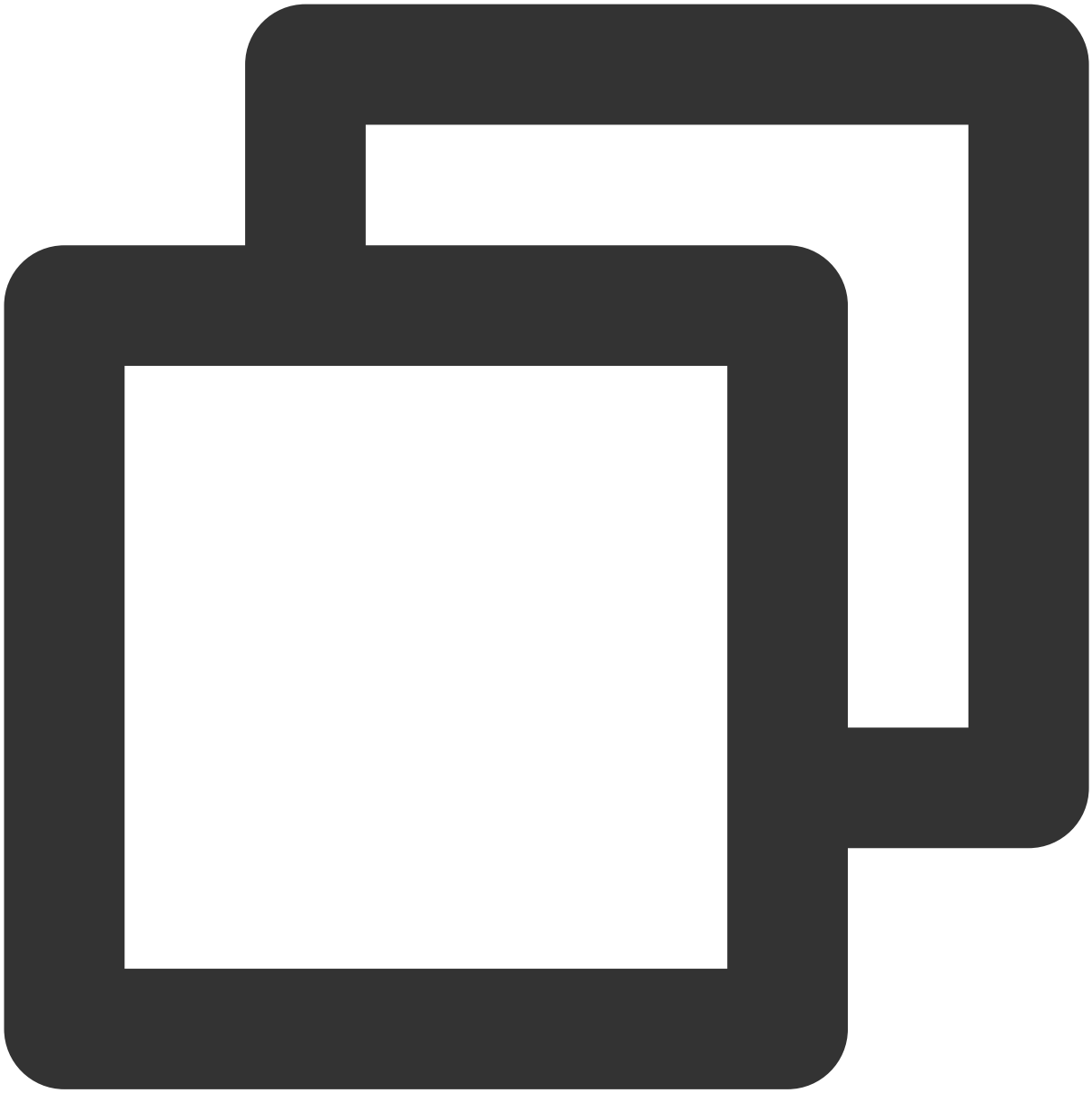


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:21:02 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI5MGVfN2RiNDBiMDlfMTk1MjhfMWZm****
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:21:55Z</LastModified>
</CopyObjectResult>
```

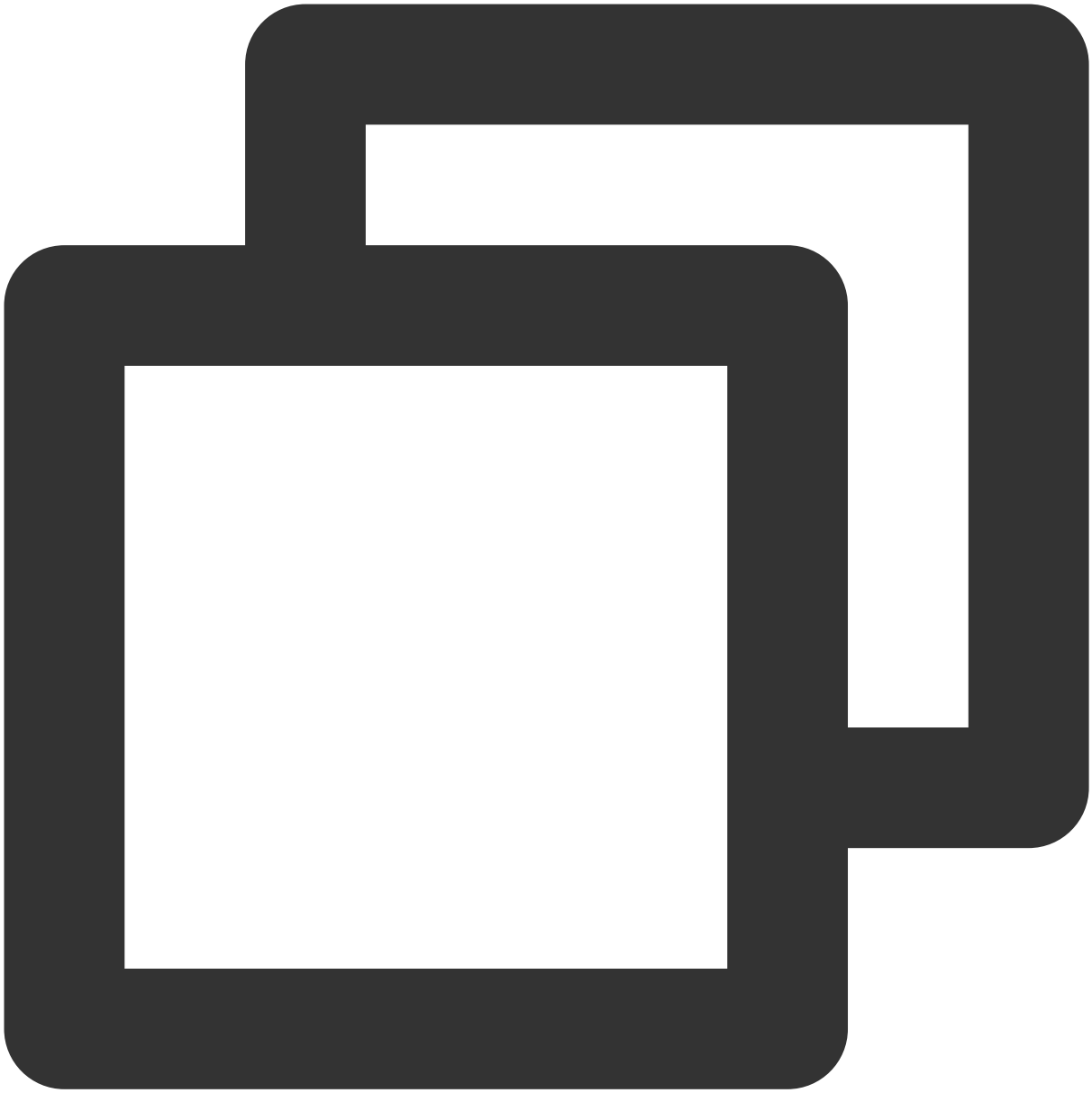
案例五：将未加密的对象复制为使用 SSE-COS 加密的目标对象

请求



```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:21:12 GMT
x-cos-server-side-encryption: AES256
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

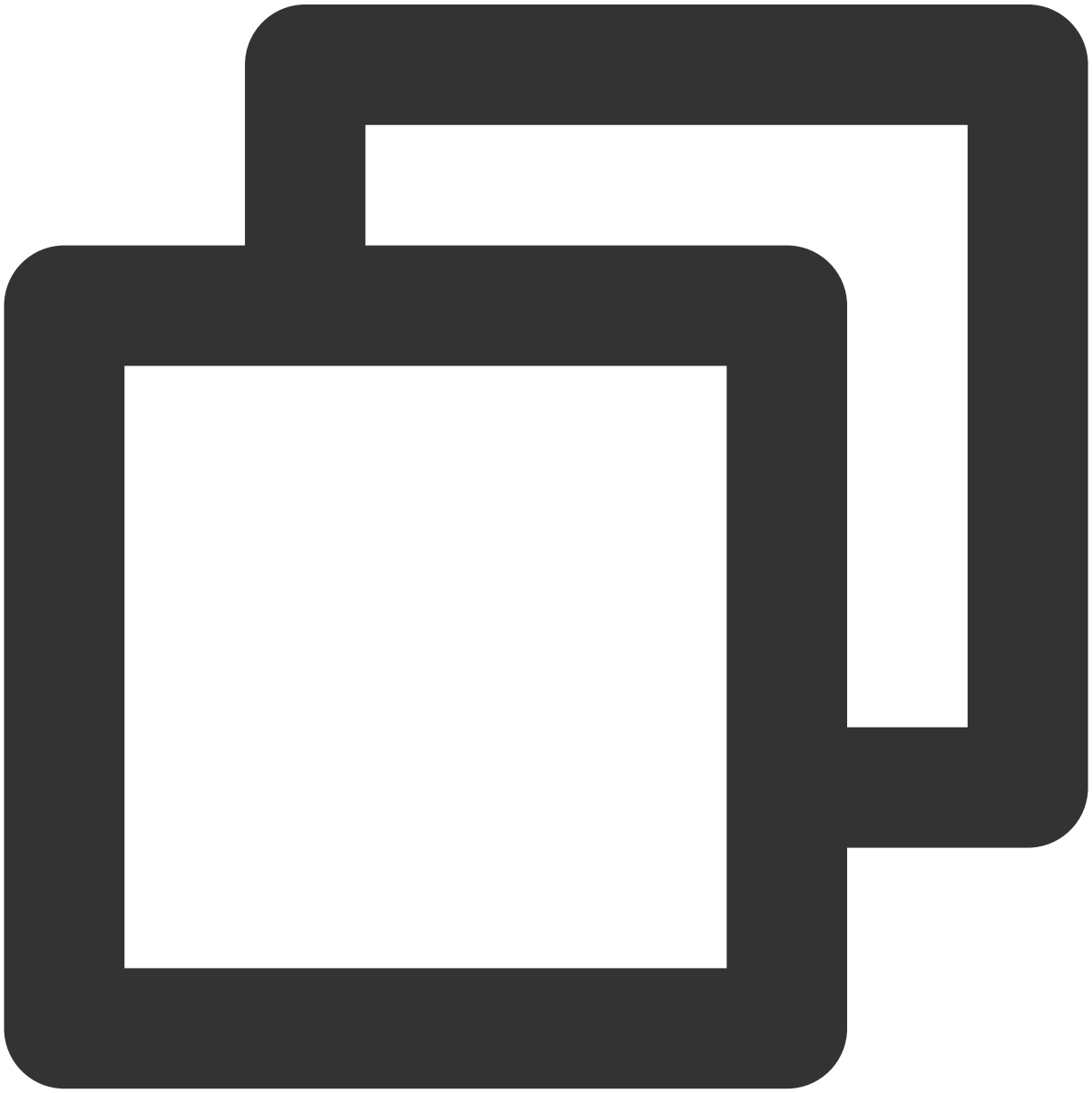


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:21:13 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI5MTlfYmIwMmEwOV9hMmUxXzFkMDQ2****
x-cos-server-side-encryption: AES256
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:21:13Z</LastModified>
</CopyObjectResult>
```

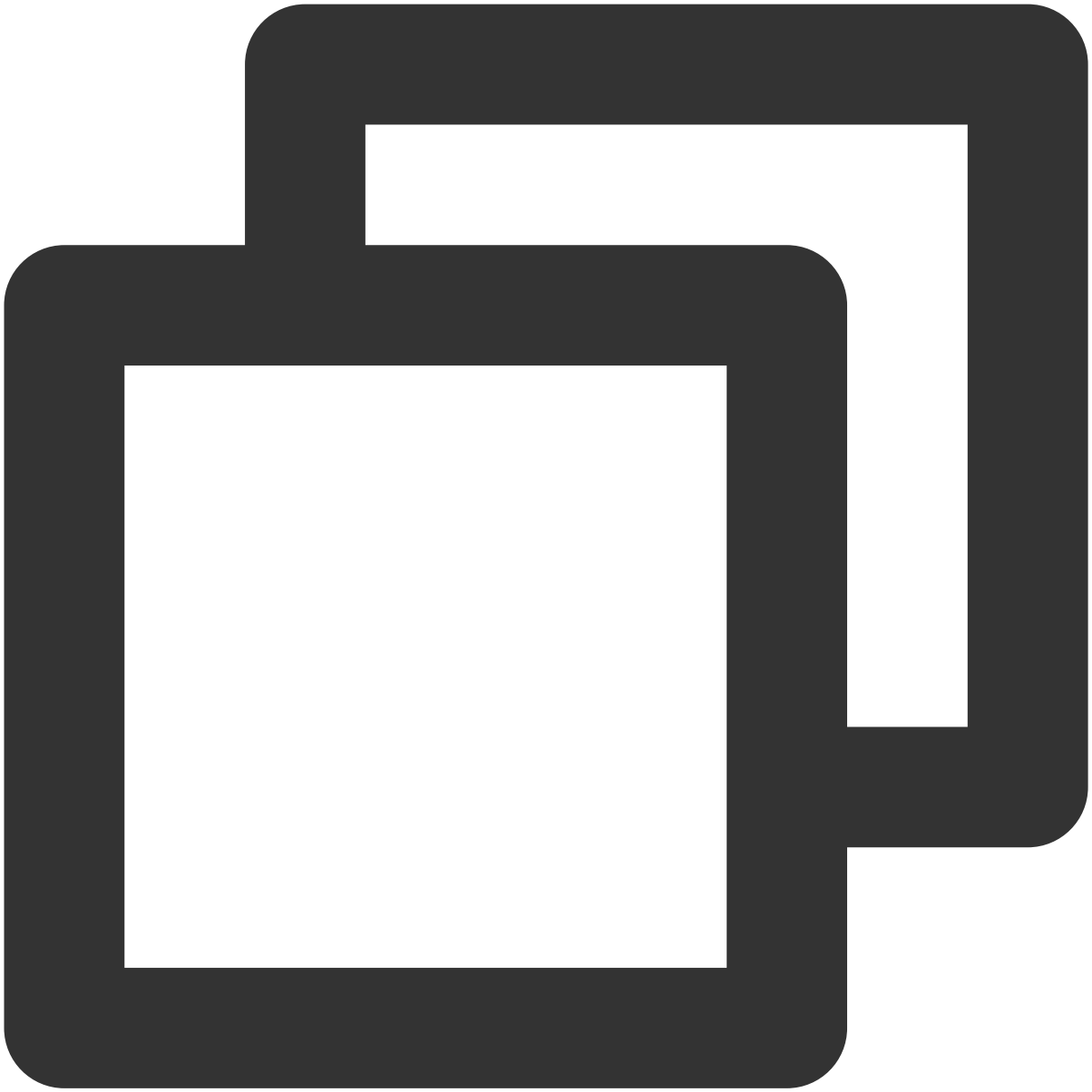
案例六：将未加密的对象复制为使用 SSE-KMS 加密的目标对象

请求



```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:21:23 GMT
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
x-cos-server-side-encryption-context: eyJhdXRob3IiOiJmeXNudGlhbiIsImNvbXBhbnkiOiJUZ
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



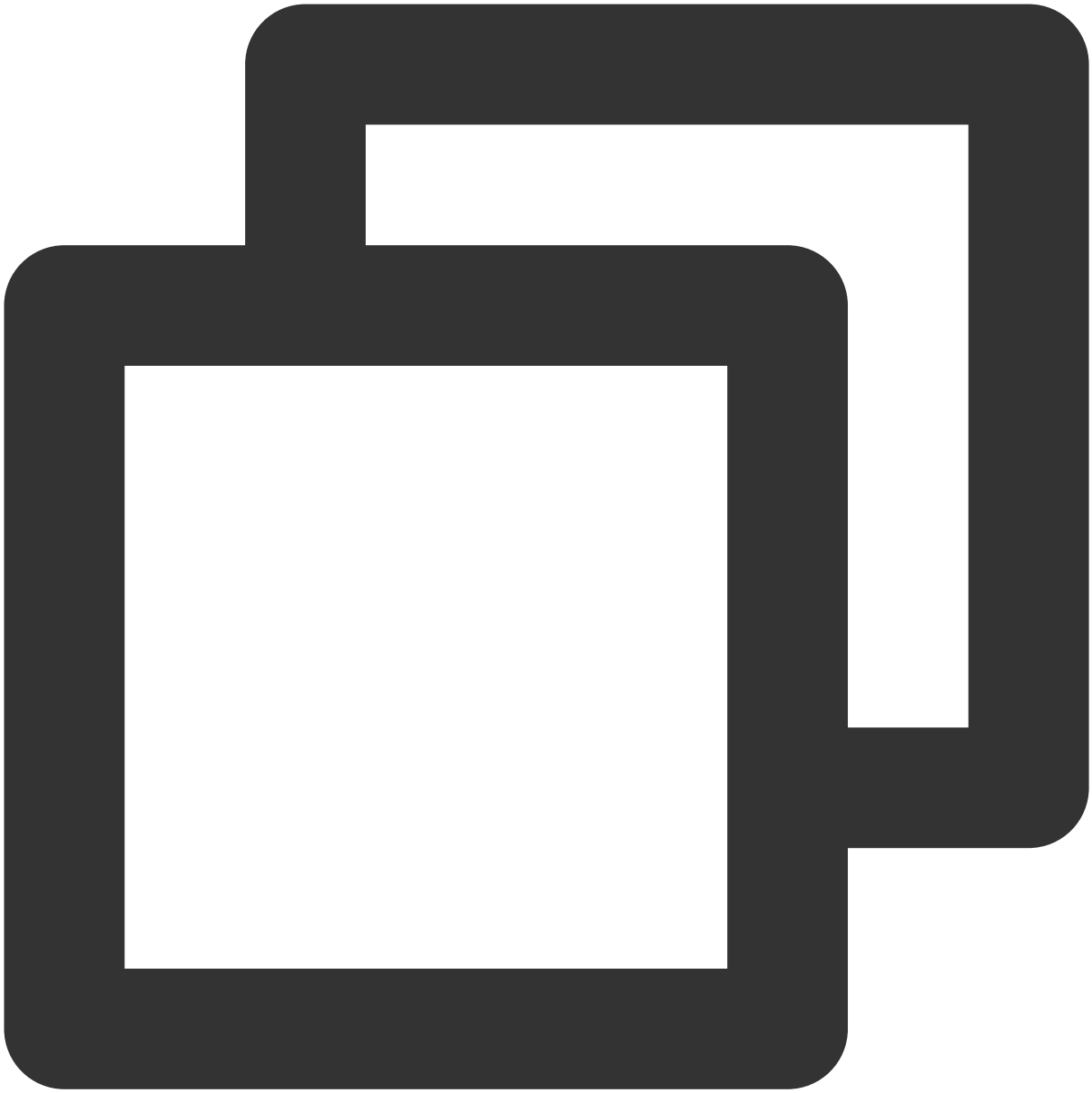
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:21:23 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI5MjNfMTliOTJhMDlfMjRiYThfMTdk****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
```

版权所有：腾讯云计算（北京）有限责任公司

```
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"f69901ec9755a5defc29057e9ec69126"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:22:16Z</LastModified>
</CopyObjectResult>
```

案例七：复制 SSE-C 加密的对象并更换密钥

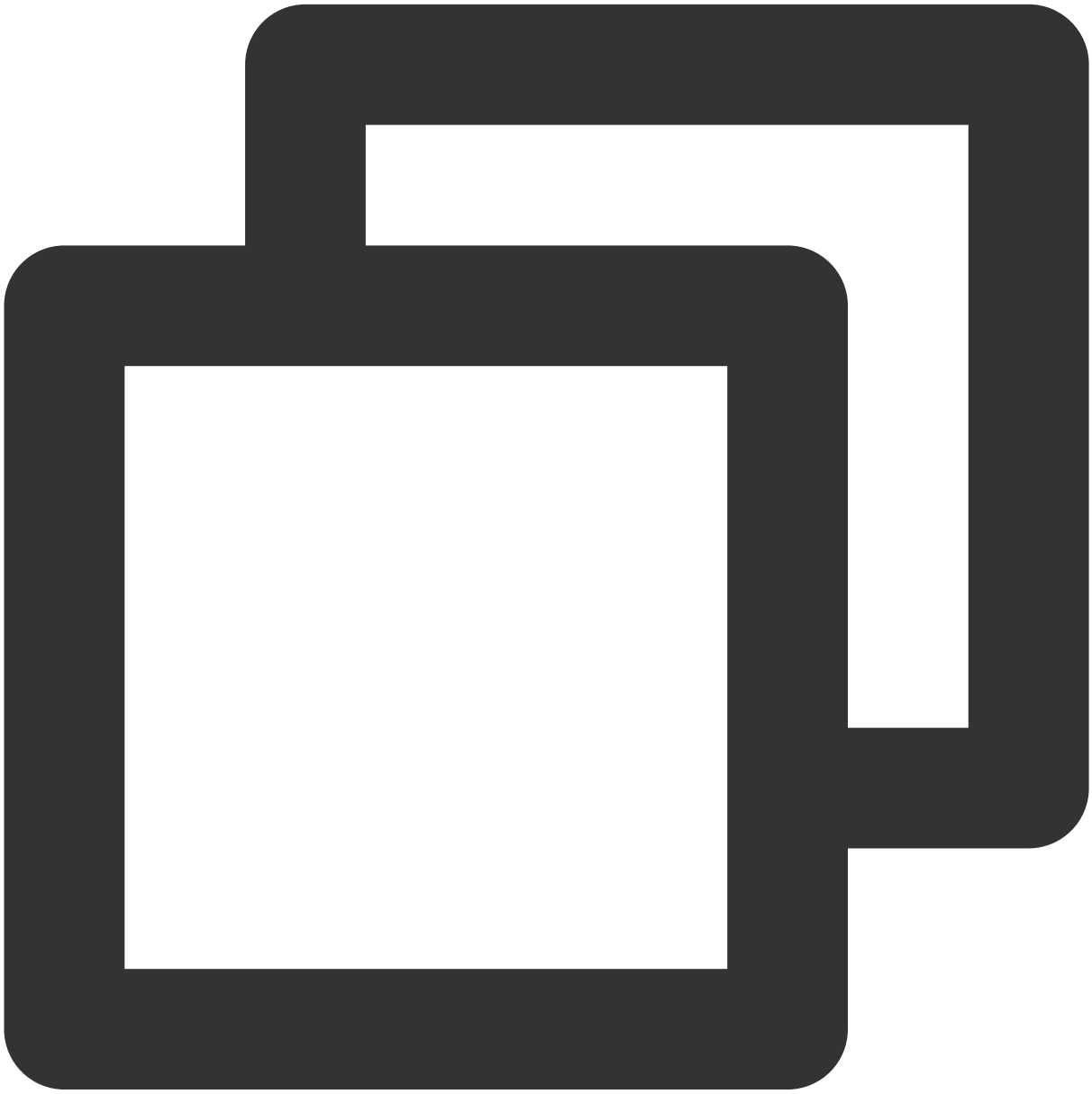
请求



```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:21:44 GMT
x-cos-copy-source-server-side-encryption-customer-algorithm: AES256
x-cos-copy-source-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0
x-cos-copy-source-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OWFiY2RlZjAxMjM0NTY3ODlhYmNk
x-cos-server-side-encryption-customer-key-MD5: hRasmdxgYDKV3nvbahU1MA==
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example-%E8
Content-Length: 0
```

```
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

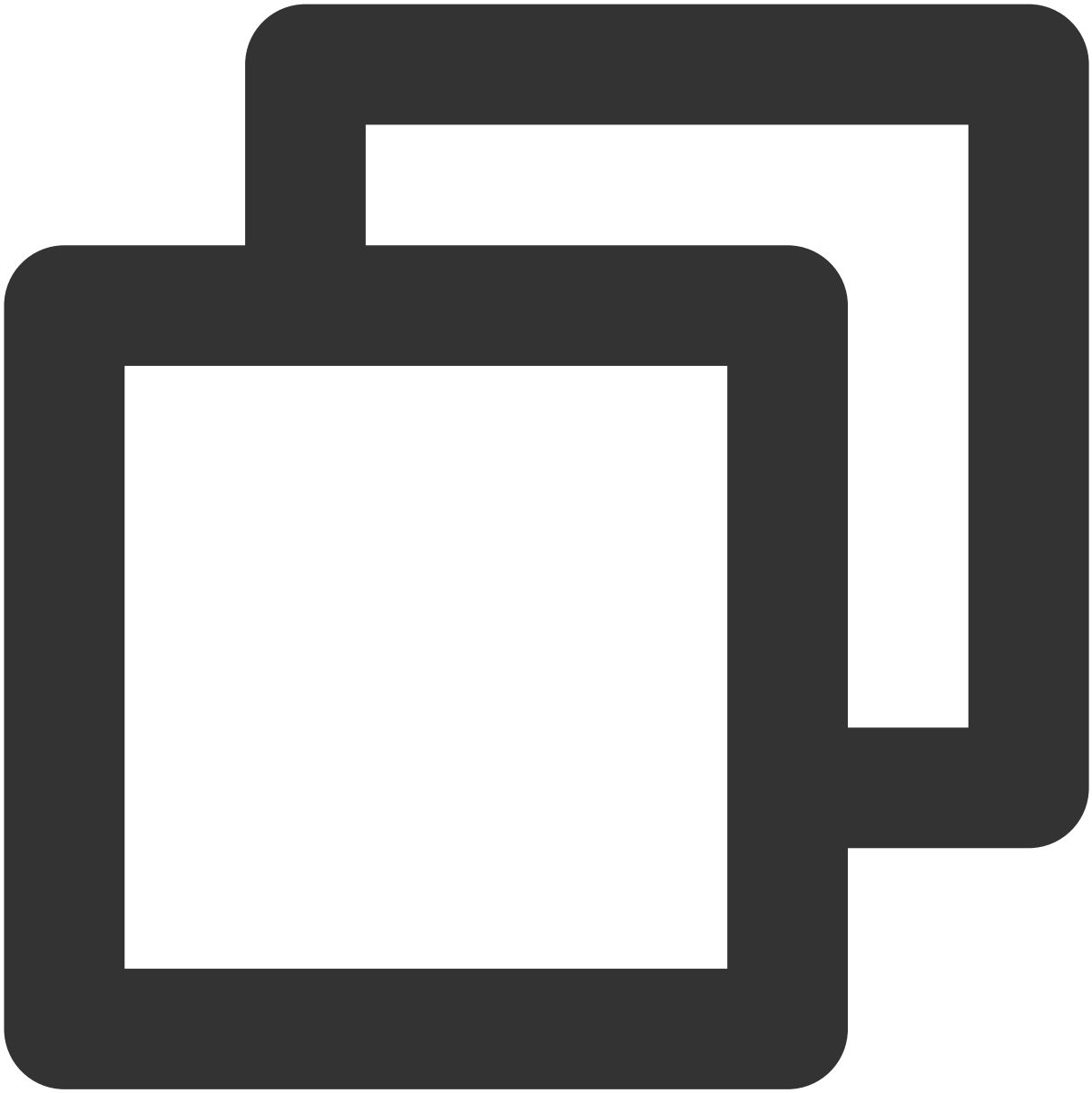


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:21:44 GMT
Server: tencent-cos
```

```
x-cos-request-id: NWU5MGI5MzhfZmFjODJhMDlfMTdlYzZfYmU1****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: hRasmdxgYDKV3nvbahU1MA==



<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"bf314b89d34119395d5610982d6581b1"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:22:31Z</LastModified>
</CopyObjectResult>
```

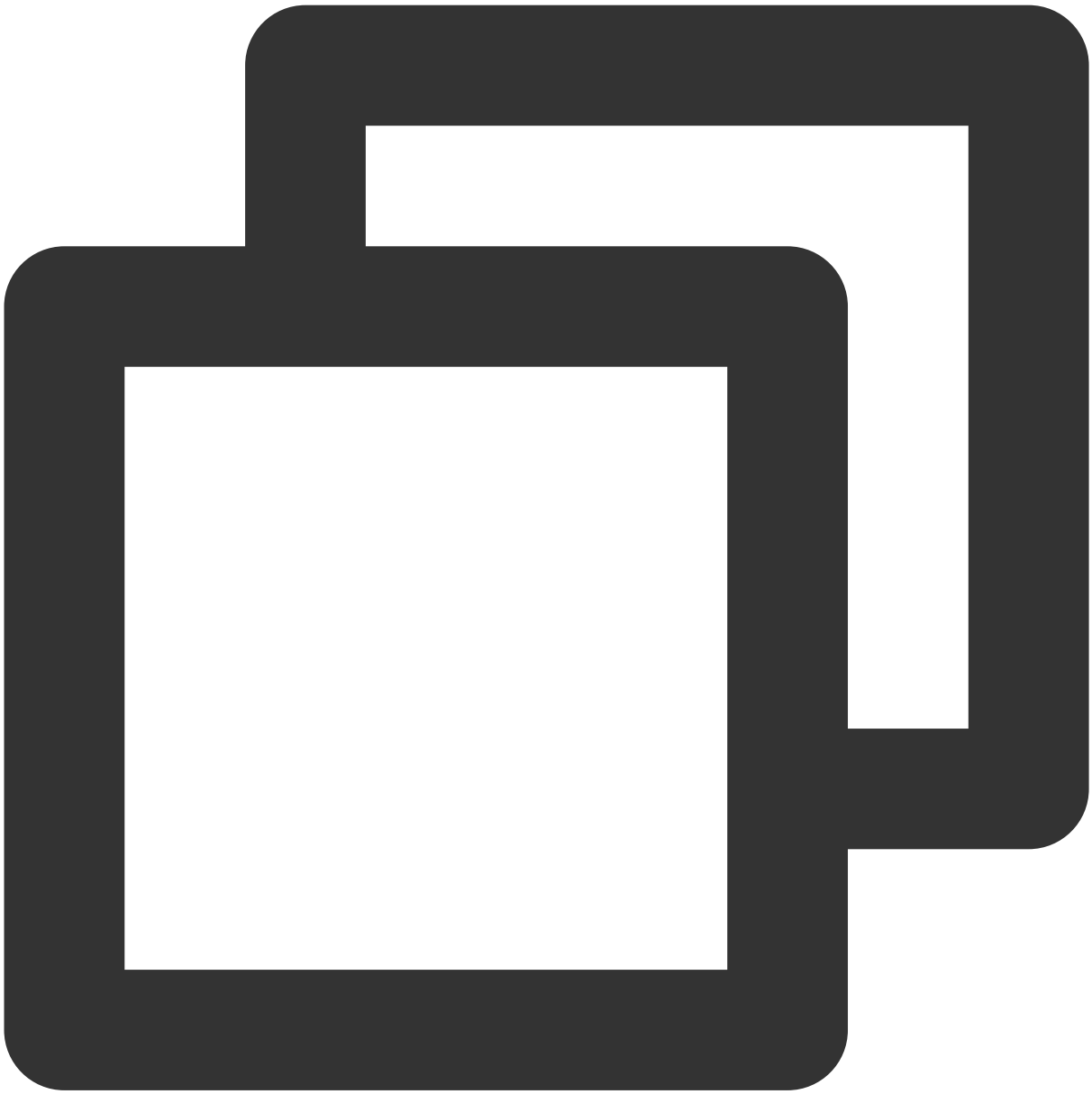
案例八：将 SSE-C 加密的对象修改为不加密

请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:22:05 GMT
x-cos-metadata-directive: Replaced
x-cos-copy-source-server-side-encryption-customer-algorithm: AES256
x-cos-copy-source-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0
x-cos-copy-source-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
x-cos-copy-source: examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobje
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

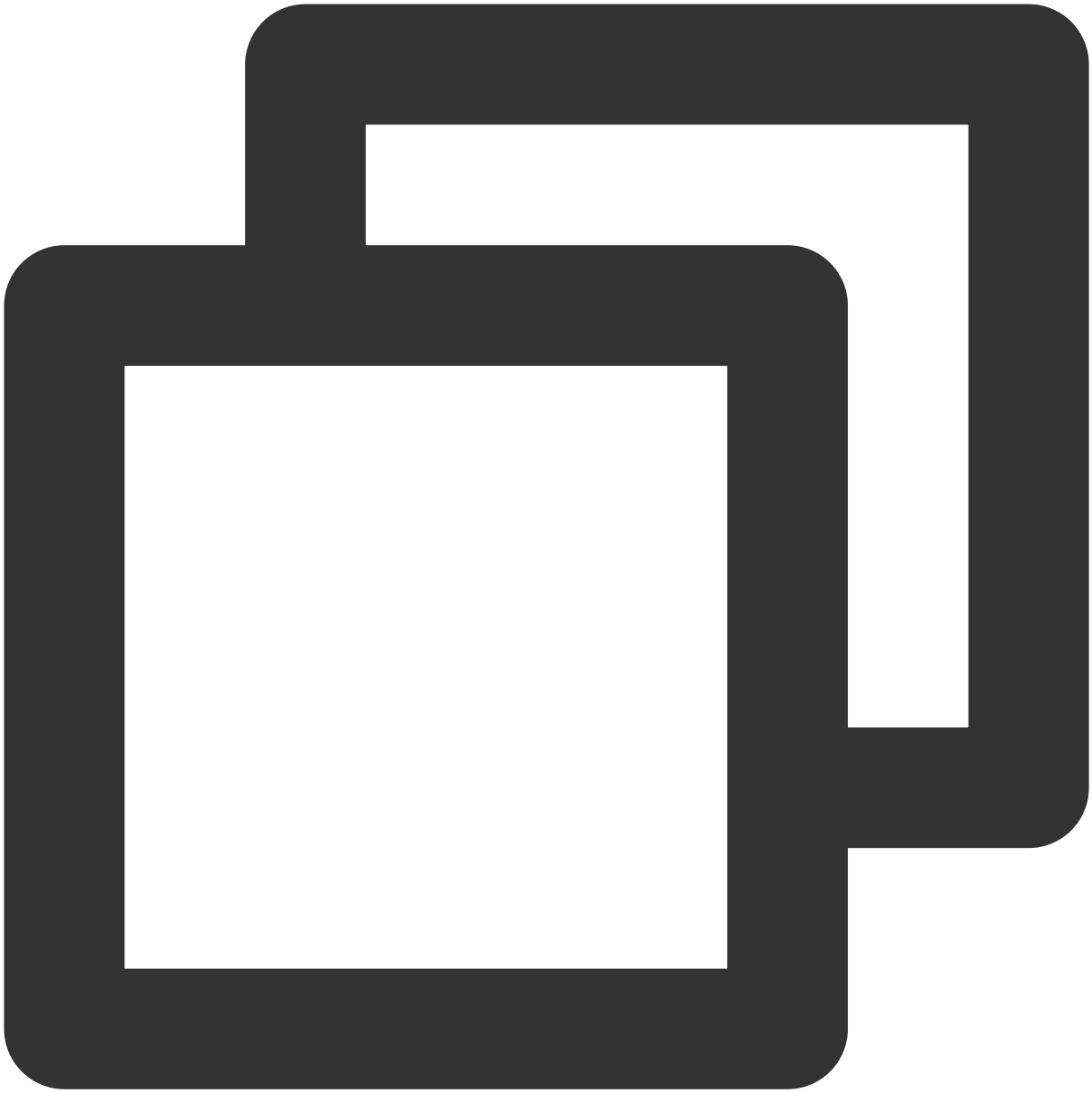


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Fri, 10 Apr 2020 18:22:05 GMT
Server: tencent-cos
x-cos-request-id: NWU5MGI5NGRfOWFjOTJhMDlfMjg2NDdfMTA0****
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-10T18:22:58Z</LastModified>
</CopyObjectResult>
```

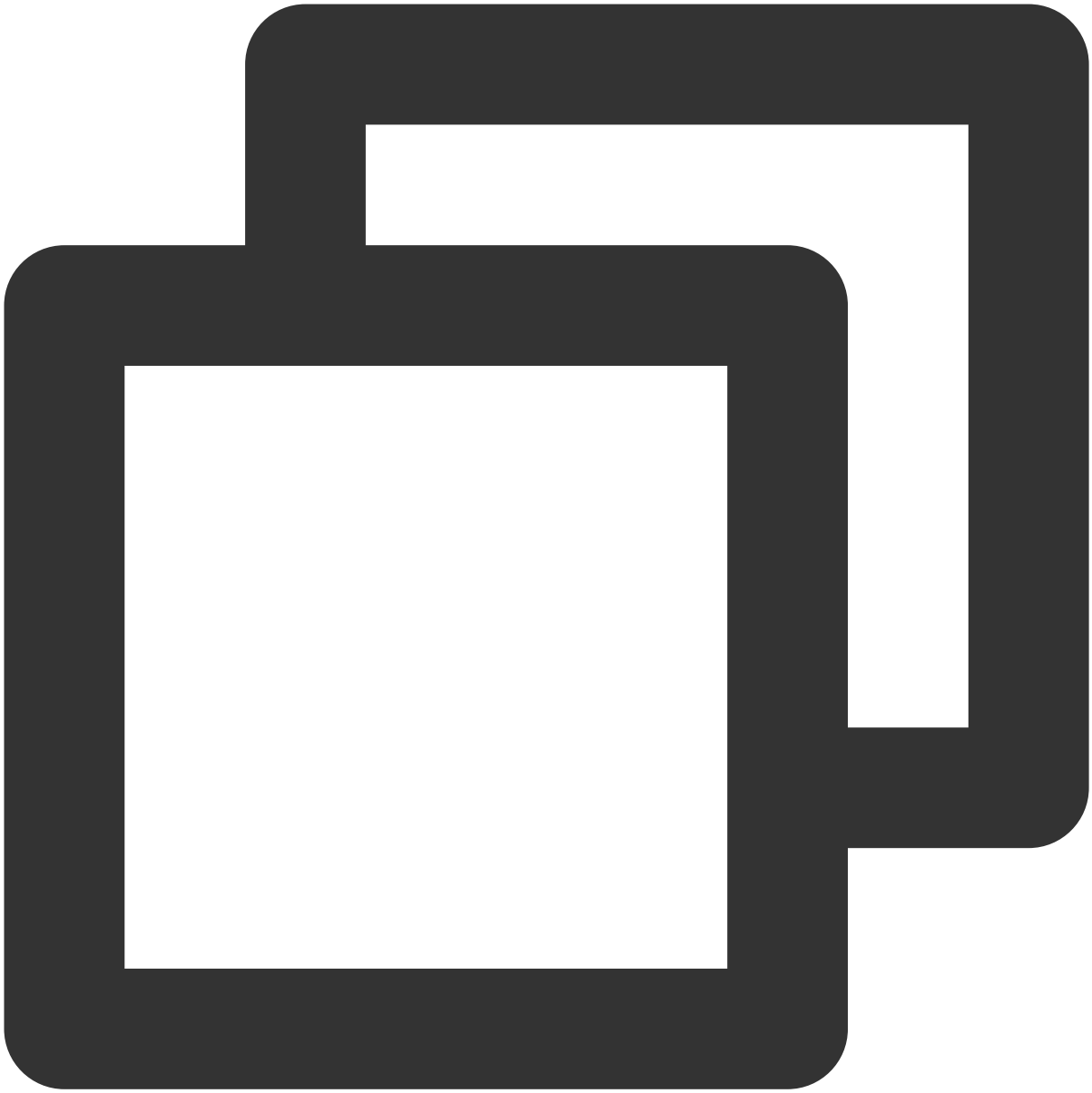
案例九：指定源对象的版本

请求



```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 11 Apr 2020 17:51:35 GMT
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example.jpg
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

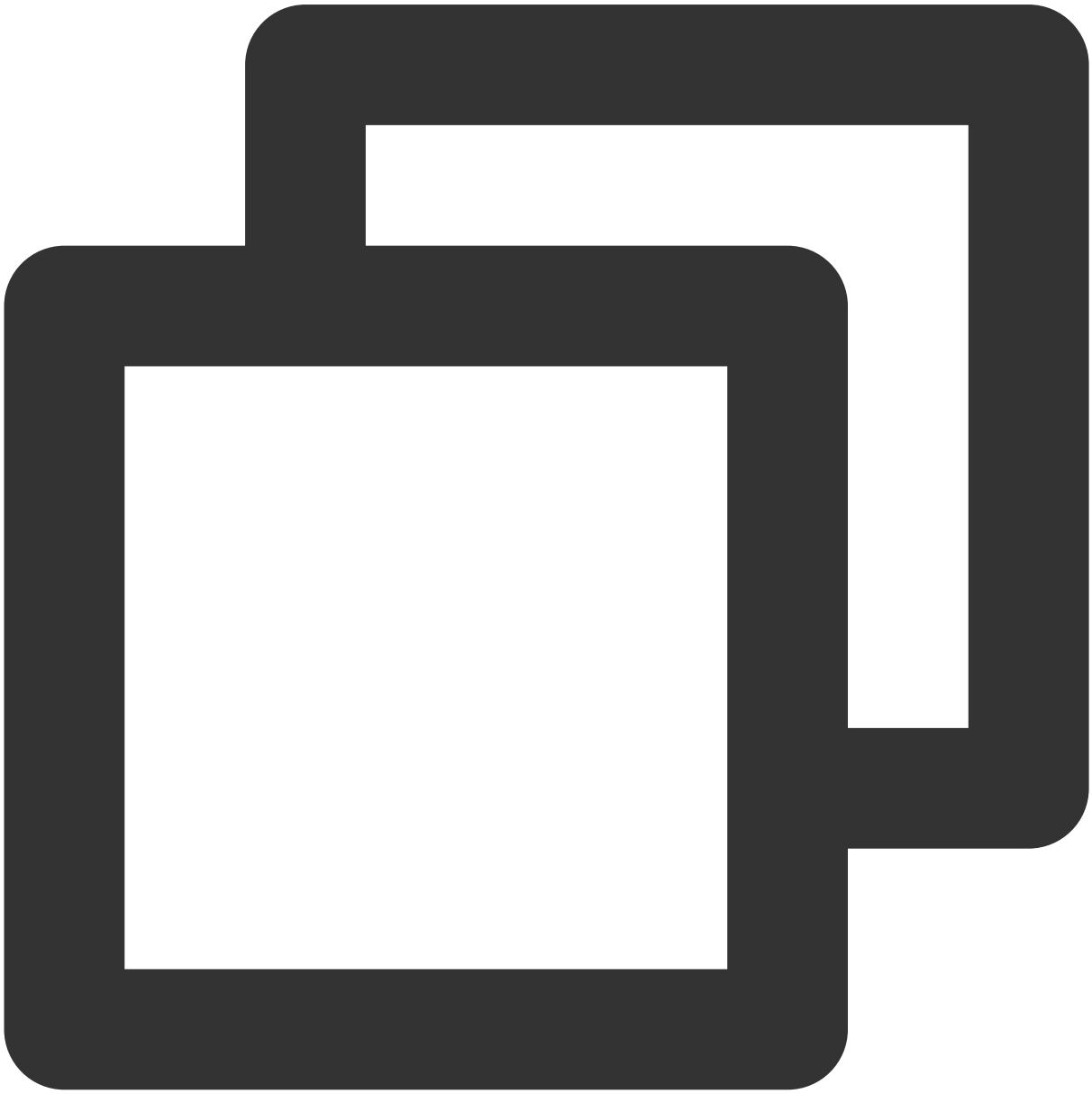


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 219
Connection: close
Date: Sat, 11 Apr 2020 17:51:35 GMT
Server: tencent-cos
x-cos-copy-source-version-id: MTg0NDUxNTc0NDYyMjQ2MzUzMjQ
x-cos-request-id: NWU5MjAzYTdfMWZjMDJhMDlfNTE4N18zNGU2****
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"ee8de918d05640145b18f70f4c3aa602"</ETag>
            <CRC64>16749565679157681890</CRC64>
            <LastModified>2020-04-11T17:51:35Z</LastModified>
</CopyObjectResult>
```

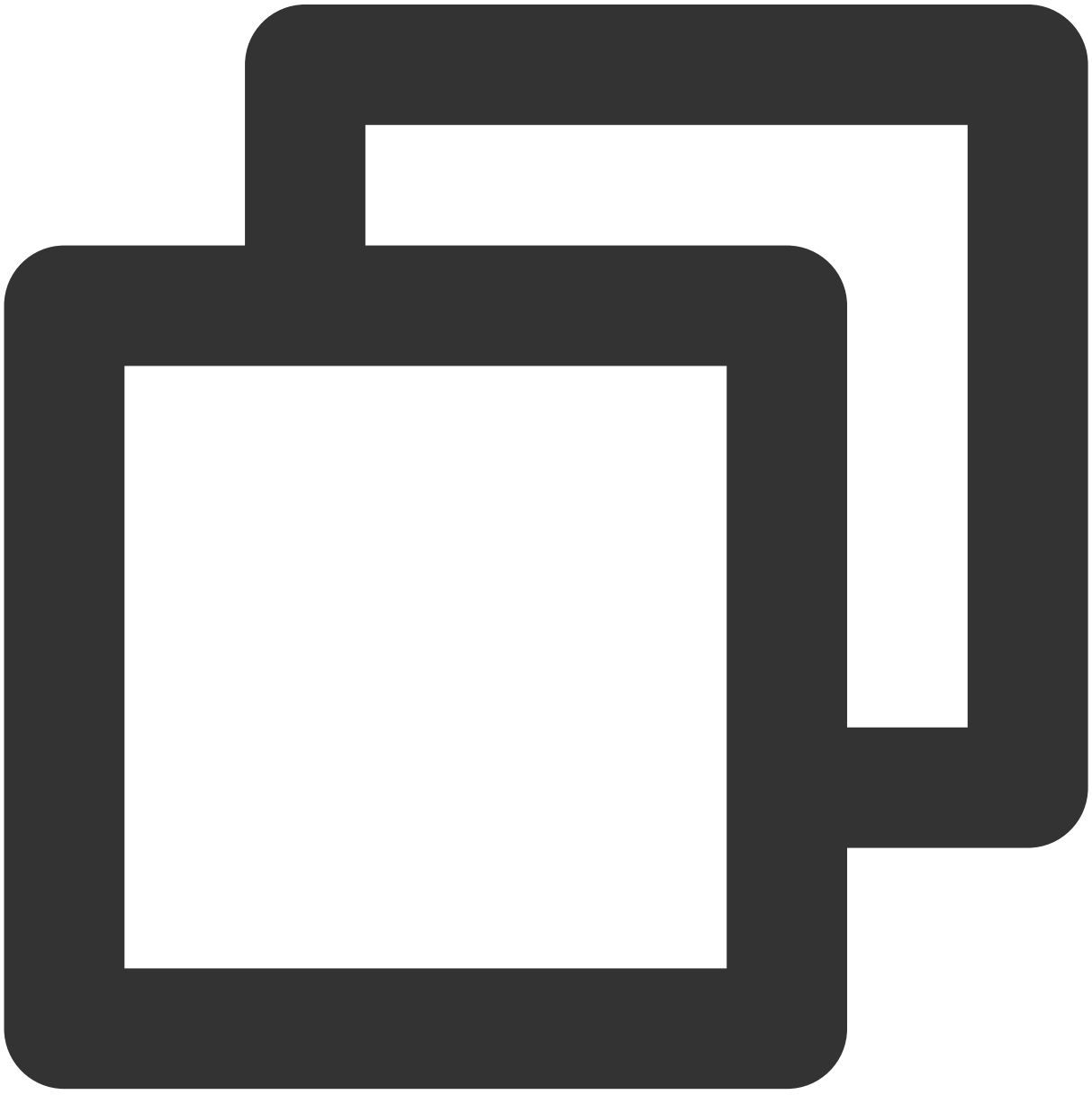
案例十：复制对象到启用版本控制的存储桶

请求



```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 11 Apr 2020 17:51:56 GMT
x-cos-copy-source: sourcebucket-1250000001.cos.ap-shanghai.myqcloud.com/example.jpg
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 272
Connection: close
Date: Sat, 11 Apr 2020 17:51:56 GMT
Server: tencent-cos
x-cos-request-id: NWU5MjAzYmNfNjRiMDJhMDlfOTE3N18yYWI4****



<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
            <ETag>"22e024392de860289f0baa7d6cf8a549"</ETag>
            <CRC64>11596229263574363878</CRC64>
            <LastModified>2020-04-11T17:51:56Z</LastModified>
            <VersionId>MTg0NDUxNTc0NDYxOTI4MzU0MDI</VersionId>
</CopyObjectResult>
```

POST Object

最近更新时间：2023-12-21 14:32:04

功能描述

POST Object 接口请求可以将本地不超过5GB的对象（Object）以网页表单（HTML Form）的形式上传至指定存储桶中。该 API 的请求者需要对存储桶有写入权限。

注意

POST Object 接口不使用 COS 对象存储统一的请求签名，而是拥有自己的签名要求，请参见本文档的 签名保护 及相关字段的描述。

如果试图添加已存在的同名对象且没有启用版本控制，则新上传的对象将覆盖原来的对象，成功时按照指定的返回方式正常返回。

上传图片时，请参见 PUT Object 接口，可参考代码示例进行数据填充。

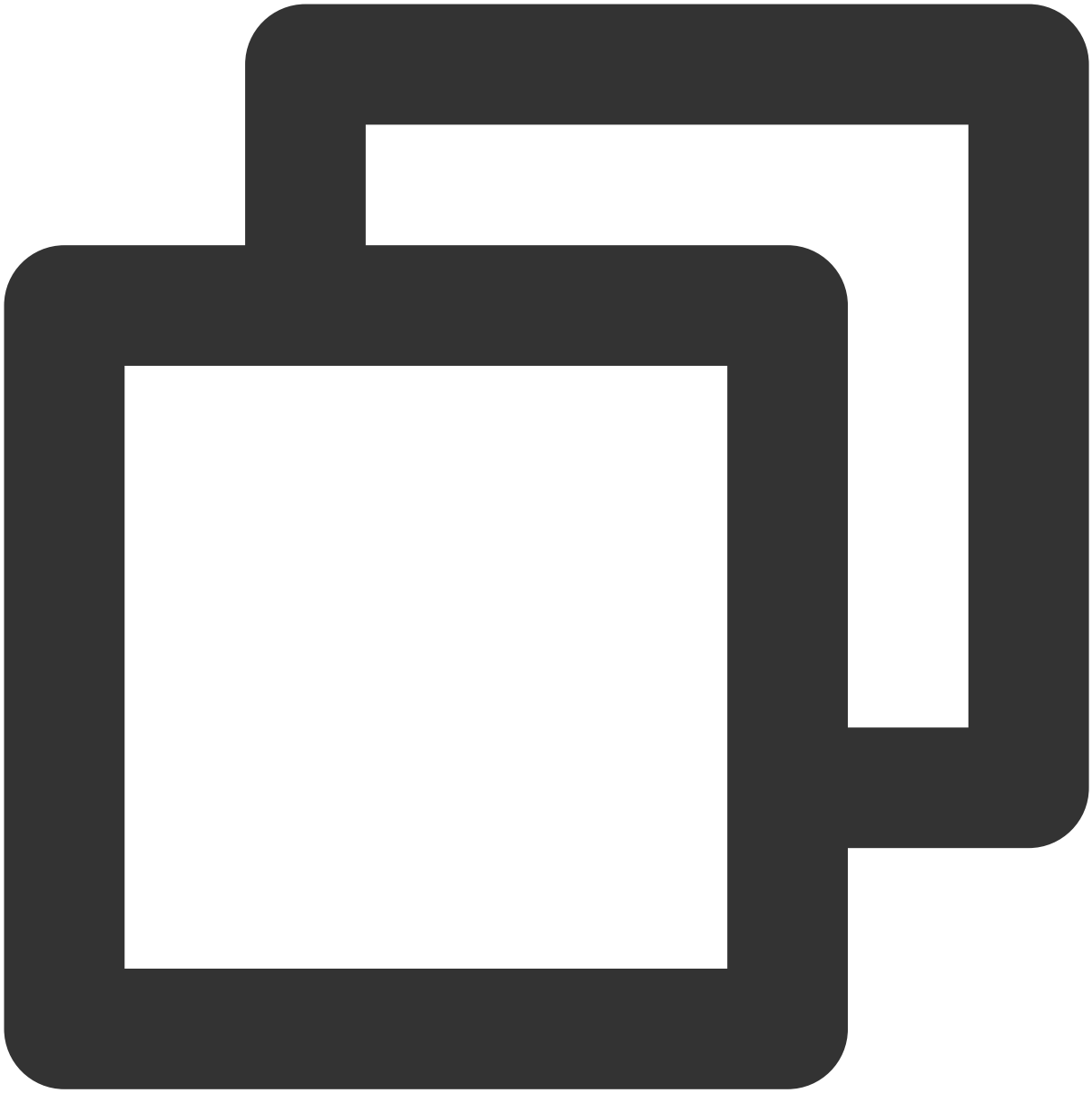
版本控制

如果对存储桶启用版本控制，对象存储将自动为要添加的对象生成唯一的版本 ID。对象存储使用 x-cos-version-id 响应头部在响应中返回此标识。

如果暂停存储桶的版本控制，则对象存储始终将 null 用作存储在存储桶中的对象的版本 ID，且不返回 x-cos-version-id 响应头部。

请求

请求示例



```
POST / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: multipart/form-data; boundary=Multipart Boundary
Content-Length: Content Length



[Multipart Form Data]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

请求表单

此接口请求体通过 multipart/form-data 编码，在 HTML 网页中通过 <form> 元素发送请求时，需将 <form> 元素的 enctype 属性设置为 multipart/form-data，随后使用 HTML 表单元素（例如 <input>、<select> 等）添加所需表单字段。

表单字段

名称	描述	类型	是否必选
key	对象键，可在对象键中指定 `${filename}` 通配符，此时将使用实际上传的文件的文件名替换对象键中的通配符，相关示例请参见本文档的 案例七	string	是
Cache-Control	RFC 2616 中定义的缓存指令，将作为对象元数据保存	string	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为对象元数据保存	string	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为对象元数据保存	string	否
Content-Type	RFC 2616 中定义的 HTTP 内容类型（MIME），将作为对象元数据保存 **注意：**通过网页表单上传文件时，浏览器会自动把指定文件的 MIME 类型携带在请求中，但对象存储 COS 并不会使用浏览器携带的 MIME 类型，您需要显式指定 Content-Type 表单字段作为对象的内容类型	string	否
Expires	RFC 2616 中定义的缓存失效时间，将作为对象元数据保存	string	否
success_action_redirect	上传成功时重定向的目标 URL 地址，如果设置，那么在上传成功时将返回 HTTP 状态码为303（Redirect）及 Location 响应头部，Location 响应头部的值为该字段指定的 URL 地址，并附加 bucket、key 和 etag 参数，相关示例请参见本文档的 案例八	string	否
success_action_status	上传成功时返回的 HTTP 状态码，可选200、201或204，默认为204。如果指定了 success_action_redirect 字段，则此字段会被忽略。相关示例请参见本文档的 案例九	number	否
x-cos-meta-*	包括用户自定义元数据头部后缀和用户自定义元数据信息，将作为对象元数据保存，大小限制为2KB **注意：**用户自定义元数据信息支持下划线（_），但用户自定义元数据头部后缀不支持下划线，仅支持减号（-）	string	否

版权所有：腾讯云计算（北京）有限责任公司

x-cos-storage-class	对象存储类型。枚举值请参见 存储类型 文档，例如 MAZ_STANDARD、MAZ_STANDARD_IA、INTELLIGENT_TIERING、MAZ_INTELLIGENT_TIERING、STANDARD_IA、ARCHIVE、DEEP_ARCHIVE。默认值：STANDARD	Enum	否
x-cos-traffic-limit	针对本次上传进行流量控制的限速值，必须为数字，单位默认为 bit/s。限速值设置范围为819200 - 838860800，即100KB/s - 100MB/s，如果超出该范围将返回400错误	integer	否
Content-MD5	经过 Base64 编码的文件内容 MD5 哈希值，用于完整性检查，验证文件内容在传输过程中是否发生变化	string	否
file	文件的信息和内容，通过网页表单上传时，浏览器将自动设置该字段的值为正确的格式 **注意：**file 字段必须放在整个表单的最后面。	file	是

访问控制列表（ACL）相关表单字段

在上传对象时可以通过指定下列表单字段来设置对象的访问权限：

名称	描述	类型	是否必选
acl	定义对象的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中对象的预设 ACL 部分，例如 default，private，public-read 等，默认为 default **注意：**如果您不需要进行对象 ACL 控制，请设置为 default 或者此项不进行设置，默认继承存储桶权限	Enum	否
x-cos-grant-read	赋予被授权者读取对象的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-full-control	赋予被授权者操作对象的所有权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否

服务端加密（SSE）相关表单字段

在上传对象时可以通过指定下列表单字段来使用服务端加密：

名称	描述	类型	是否必选
x-cos-server-side-encryption	服务端加密算法，支持 AES256、cos/kms	string	使用 SSE-COS 或 SSE-KMS 时，此字段为必选项
x-cos-server-side-encryption-customer-algorithm	服务端加密算法，支持 AES256	string	使用 SSE-C 时，此字段为必选项
x-cos-server-side-encryption-cos-kms-key-id	当 x-cos-server-side-encryption 值为 cos/kms 时，用于指定 kms 的用户主密钥 CMK，如不指定则使用 COS 默认创建的 CMK，更多详细信息可参见 SSE-KMS 加密	string	否
x-cos-server-side-encryption-context	当 x-cos-server-side-encryption 值为 cos/kms 时，用于指定加密上下文，值为 JSON 格式加密上下文键值对的 Base64 编码 例如 `eyJhIjoiYXNkZmEiLCJiIjoiMTIzMzIxIn0=`	string	否
x-cos-server-side-encryption-customer-key	服务端加密密钥的 Base64 编码 例 如 `MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=`	string	使用 SSE-C 时，此字段为必选项
x-cos-server-side-encryption-customer-key-MD5	服务端加密密钥的 MD5 哈希值，使用 Base64 编码 例如 `U5L61r7jcwdNvT7frmUG8g==`	string	使用 SSE-C 时，此字段为必选项

签名保护

POST Object 接口要求在请求中携带签名相关字段，COS 服务器端收到消息后，进行身份验证，验证成功则可接受并执行请求，否则将会返回错误信息并丢弃此请求。

签名流程如下：

1. 准备工作

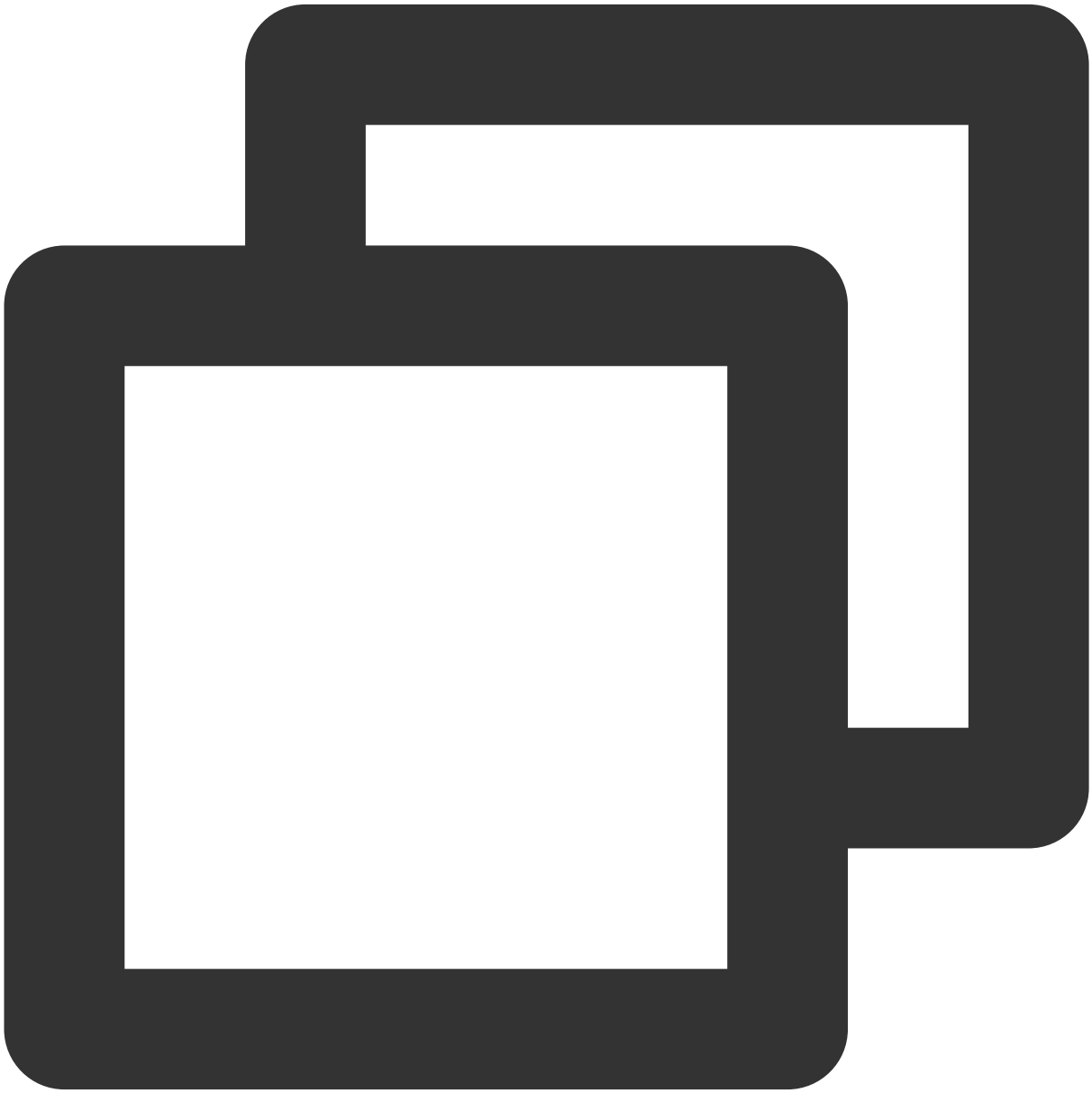
在访问管理控制台的 API 密钥管理 页面中获取 SecretId 和 SecretKey。

2. 生成 KeyTime

a. 获取当前时间对应的 Unix 时间戳 StartTimestamp，Unix 时间戳是从 UTC（协调世界时，或 GMT 格林威治时间）1970年1月1日0时0分0秒起至现在的总秒数。
b. 根据上述时间戳和期望的签名有效时长算出签名过期时间对应的 Unix 时间戳 EndTimestamp。
c. 拼接签名有效时间，格式为 `StartTimestamp;EndTimestamp` ，即为 KeyTime。

3. 构造“策略”（Policy）

策略为一个 JSON 文本，一个典型的策略如下：



```
{
    "expiration": "2019-08-30T09:38:12.414Z",
```

```
    "conditions": [
        { "acl": "default" },
        { "bucket": "examplebucket-1250000000" },
        [ "starts-with", "$key", "folder/subfolder/" ],
        [ "starts-with", "$Content-Type", "image/" ],
        [ "starts-with", "$success_action_redirect", "https://my.website/" ],
        [ "eq", "$x-cos-server-side-encryption", "AES256" ],
        { "q-sign-algorithm": "sha1" },
        { "q-ak": "AKIDQjz3ltompVjBni5LitkWHFlFpwkn9U5q" },
        { "q-sign-time": "1567150692;1567157892" }
    ]
}
```

其中：

expiration：该策略的过期时间，ISO8601 格式字符串

conditions：该策略的具体条件限定数组，限定条件的具体规则如下表。

类型	描述
完全匹配	使用 `{ "key": "value" }` 或 `[ "eq", "$key", "value" ]` 方式表达，其中 key 为被限定的表单字段，value 为被限定的值
前缀匹配	使用 `[ "starts-with", "$key", "value" ]` 方式表达，其中 key 为被限定的表单字段，value 为被限定的前缀，可为空
范围匹配	仅适用于 `[ "content-length-range", minNum, maxNum ]`，用于限定文件的长度必须在 minNum 和 maxNum 范围内

支持被限定的表单字段如下：

字段名称	描述	匹配方式	是否必选
acl	对象的访问控制列表（ACL）属性	完全、前缀	否
bucket	上传的存储桶	完全	否
key	对象键。如果在上传时对象键使用 `${filename}` 通配符，那么对象键将在验证策略前被处理为最终的对象键，此时在策略中应该使用前缀匹配，而不应该出现 `${filename}` 通配符	完全、前缀	否
content-length-range	文件长度范围	范围	否
Cache-Control, Content-Type, Content-	RFC 2616 中定义的相关头部，将在下载对象时作为响应头部返回	完全、前缀	否

版权所有：腾讯云计算（北京）有限责任公司

Disposition, Content-Encoding, Expires			
success_action_redirect	上传成功时重定向的目标 URL 地址	完全、前缀	否
success_action_status	上传成功时返回的 HTTP 状态码	完全	否
x-cos-meta-*	用户自定义的元数据头部字段	完全、前缀	否
x-cos-*	本文档中提到的其他 COS 相关表单字段，例如 ACL 和 SSE 相关字段	完全	否
q-sign-algorithm	签名哈希算法，固定为 sha1	完全	是
q-ak	上文所述的 SecretId	完全	是
q-sign-time	上文所生成的 KeyTime	完全	是

注意

“策略”（Policy）中限定的除 bucket 以外的字段，都必须出现在表单字段中。例如限定了 `{ "acl": "default" }`，那么表单中必须出现 acl 且值为 default。
基于安全考虑，强烈建议您对所有可以限定的表单字段进行限定。

4. 生成 SignKey

使用 HMAC-SHA1 以 SecretKey 为密钥，以 KeyTime 为消息，计算消息摘要（哈希值，16进制小写形式），即为 SignKey，例如：`39acc8c9f34ba5b19bce4e965b370cd3f62d2fba`。

5. 生成 StringToSign

使用 SHA1 对上文中构造的策略（Policy）文本计算消息摘要（哈希值，16进制小写形式），即为 StringToSign，例如：`d5d903b8360468bc81c1311f134989bc8c8b5b89`。

6. 生成 Signature

使用 HMAC-SHA1 以 SignKey 为密钥（字符串形式，非原始二进制），以 StringToSign 为消息（字符串形式，非原始二进制），计算消息摘要（哈希值，16进制小写形式），即为 Signature，例如：`7758dc9a832e9d301dca704cacbf9d9f8172fdef`。

7. 将签名附加到表单

将上述策略和签名相关信息，以下表中描述的方式附加到表单中：

名称	描述	类型	是否必选

x-cos-security-token	使用临时安全凭证时需要传入的安全令牌字段，详情请参见 临时安全凭证 相关说明	string	否，当使用临时密钥时，此表单项为必选项
policy	经过 Base64 编码的“策略”（Policy）内容	string	是
q-sign-algorithm	签名哈希算法，固定为 sha1	string	是
q-ak	上文所述的 SecretId	string	是
q-key-time	上文所生成的 KeyTime	string	是
q-signature	上文所生成的 Signature	string	是

注意

签名表单字段需要在 file 表单字段之前。

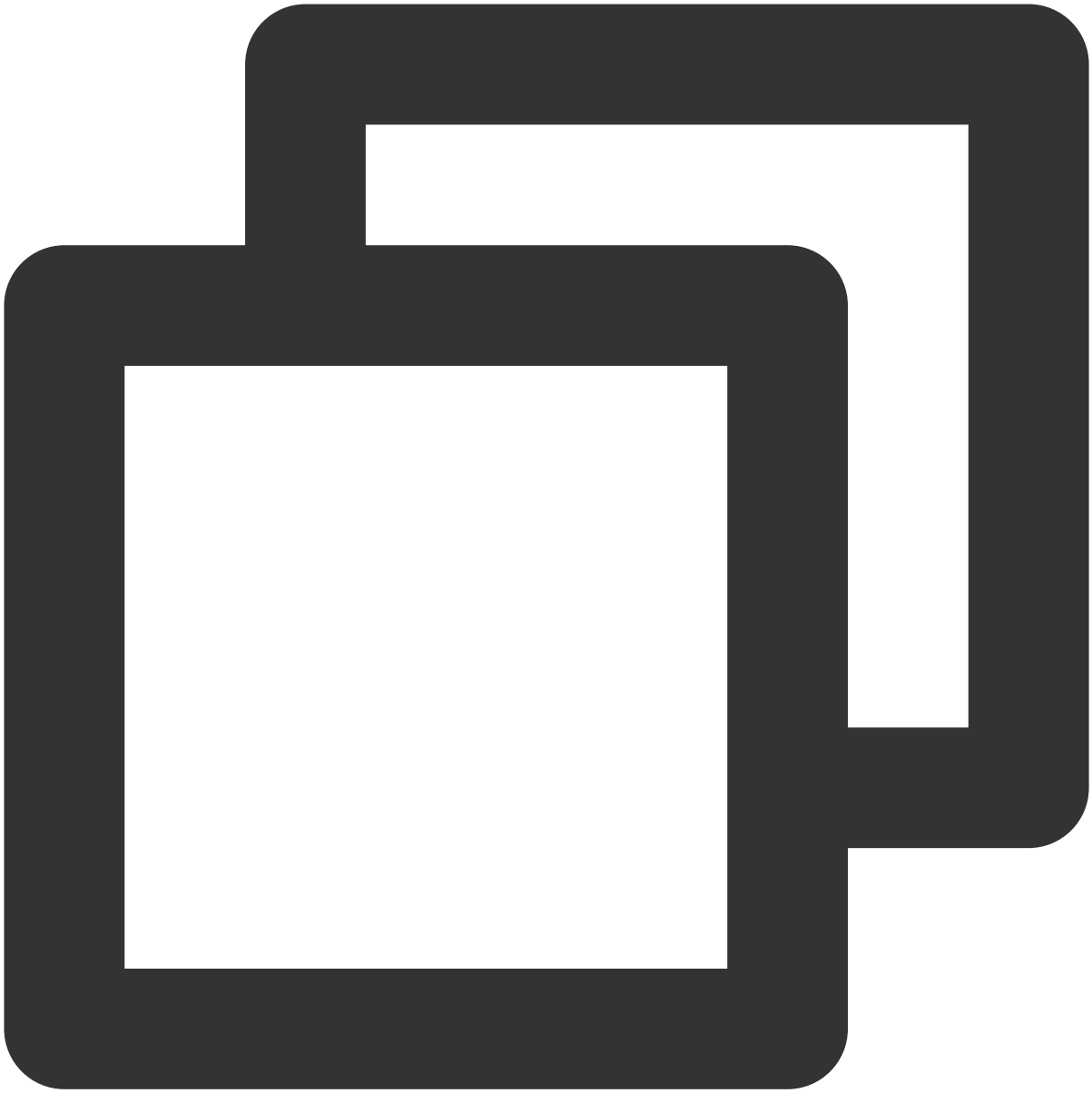
签名保护实际案例

准备工作

登录访问管理控制台的 API 密钥管理 页面获取其 APPID、SecretId 和 SecretKey，举例如下：

APPID	SecretId	SecretKey
1250000000	AKIDQjz3ltompVjBni5LitkWHFlFpwkn9U5q	BQYIM75p8x0iWVFSIgqEKwFprpRSVHlz

构造策略



```
{
    "expiration": "2019-08-30T09:38:12.414Z",
    "conditions": [
        { "acl": "default" },
        { "bucket": "examplebucket-1250000000" },
        [ "starts-with", "$key", "folder/subfolder/" ],
        [ "starts-with", "$Content-Type", "image/" ],
        [ "starts-with", "$success_action_redirect", "https://my.website/" ],
        [ "eq", "$x-cos-server-side-encryption", "AES256" ],
        { "q-sign-algorithm": "sha1" },
        { "q-ak": "AKIDQjz3ltompVjBni5LitkWHFlFpwkn9U5q" },
```

```
        { "q-sign-time": "1567150692;1567157892" }
    ]
}
```

中间变量

KeyTime = `1567150692;1567157892`

SignKey = `39acc8c9f34ba5b19bce4e965b370cd3f62d2fba`

StringToSign = `d5d903b8360468bc81c1311f134989bc8c8b5b89`

Signature = `7758dc9a832e9d301dca704cacbf9d9f8172fdef`

签名表单字段

policy =

`ewogICAgImV4cGlyYXRpb24iOiAiMjAxOS0wOC0zMFQwOTozODoxMi40MTRaIiwKICAgICJjb25kaXRpb25zIjogWwogICAgICAgIHsgImFjbCI6ICJkZWZhdWx0IiB9LAogICAgICAgIHsgImJ1Y2tldCI6ICJleGFtcGxlYnVja2V0LTEyNTAwMDAwMDAiIH0sCiAgICAgICAgWyAic3RhcnRzLXdpdGgiLCAiJGtleSIsICJmb2xkZXIvc3ViZm9sZGVyLyIgXSwKICAgICAgICBbICJzdGFydHMtd2l0aCIsICIkQ29udGVudC1UeXBlIiwgImltYWdlLyIgXSwKICAgICAgICBbICJzdGFydHMtd2l0aCIsICIkc3VjY2Vzc19hY3Rpb25fcmVkaXJlY3QiLCAiaHR0cHM6Ly9teS53ZWJzaXRlLyIgXSwKICAgICAgICBbICJlcSIsICIkeC1jb3Mtc2VydmVyLXNpZGUtZW5jcnlwdGlvbiIsICJBRVMyNTYiIF0sCiAgICAgICAgeyAicS1zaWduLWFsZ29yaXRobSI6ICJzaGExIiB9LAogICAgICAgIHsgInEtYWsiOiAiQUtJRFFqejNsdG9tcFZqQm5pNUxpdGtXSEZsRnB3a245VTVxIiB9LAogICAgICAgIHsgInEtc2lnbi10aW1lIjogIjE1NjcxNTA2OTI7MTU2NzE1Nzg5MiIgfQogICAgXQp9`

q-sign-algorithm = `sha1`

q-ak = `AKIDQjz3ltompVjBni5LitkWHFlFpwkn9U5q`

q-key-time = `1567150692;1567157892`

q-signature = `7758dc9a832e9d301dca704cacbf9d9f8172fdef`

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

名称	描述	类型
Location	当使用 success_action_redirect 表单字段时，此响应头部的值为 success_action_redirect 指定的 URL 地址，并附加 bucket、key 和 etag 参数，相关示例请参见本文档的 案例八 当未使用 success_action_redirect 表单字段时，此响应头部的值为完整的对象访问 URL 地址，相关示例请参见本文档的 案例一	string

版本控制相关头部

在启用版本控制的存储桶中上传对象，将返回下列响应头部：

名称	描述	类型
x-cos-version-id	对象的版本 ID	string

服务端加密（SSE）相关头部

如果在上传对象时使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

此接口响应体为空。

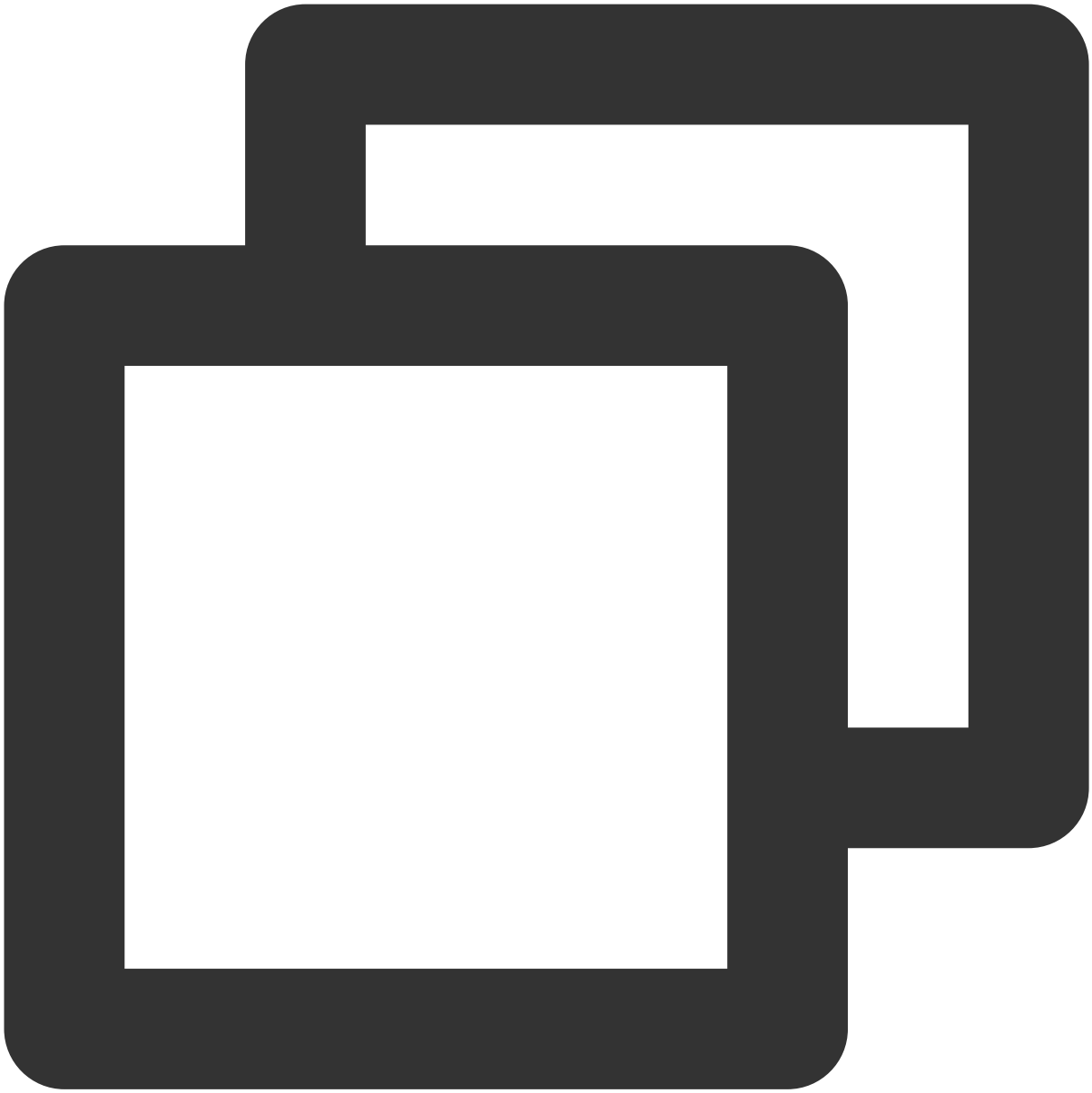
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：简单案例（未启用版本控制）

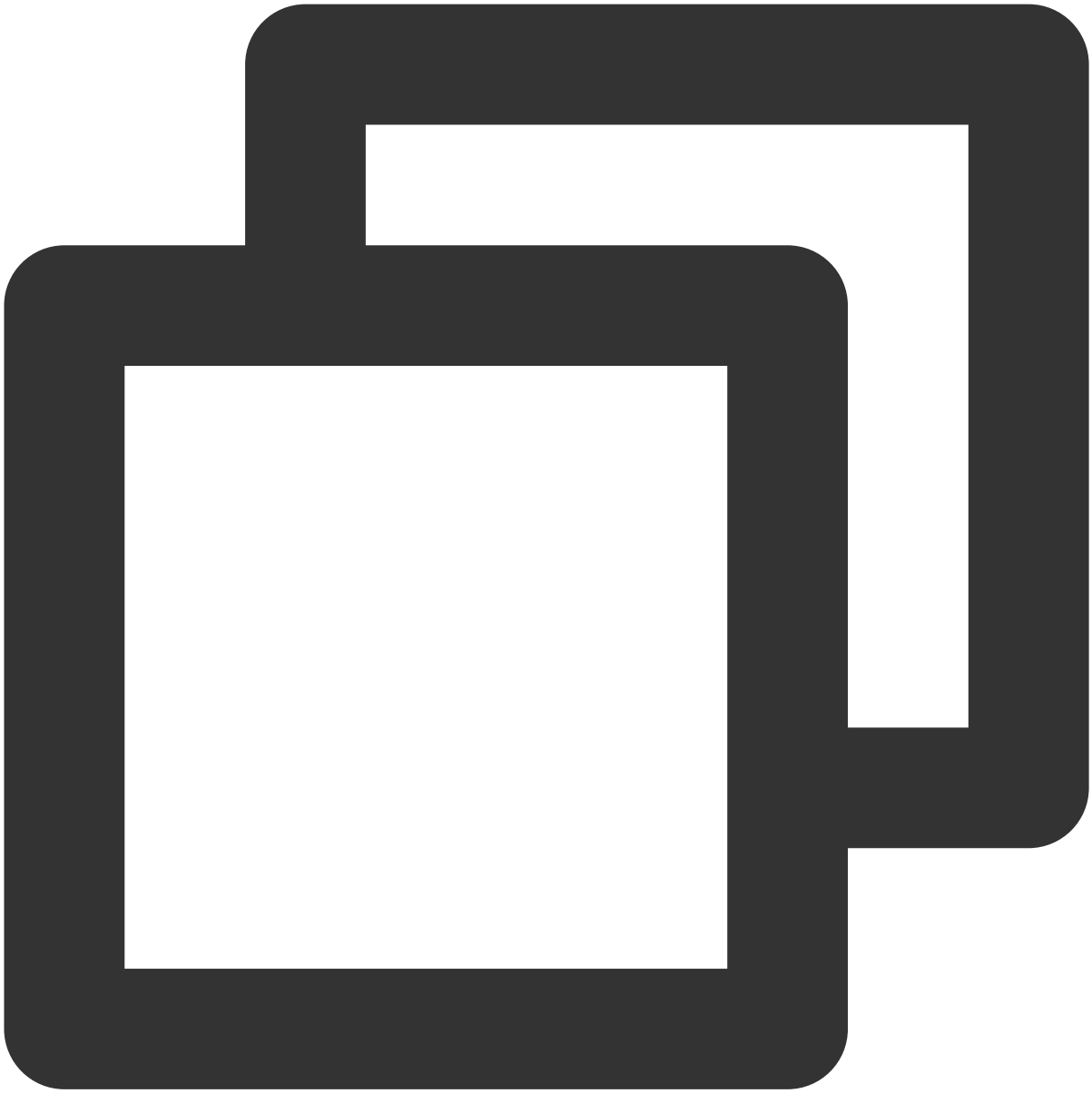
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 07:39:34 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Length: 1119
Connection: close

------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="key"

exampleobject
```

```
------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJxLXNpZ24tYWxnb3JpdGhtIjoic2hhMSJ9LHsicS1hayI6IkFLSUQ4QTBmQlZ
------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="q-key-time"

1567064374;1567071574
------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="q-signature"

74ba120129a13d8f0e19479fbdc01bca3bca****
------WebKitFormBoundaryZBPbaoYE2gqeB21N
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryZBPbaoYE2gqeB21N--
```

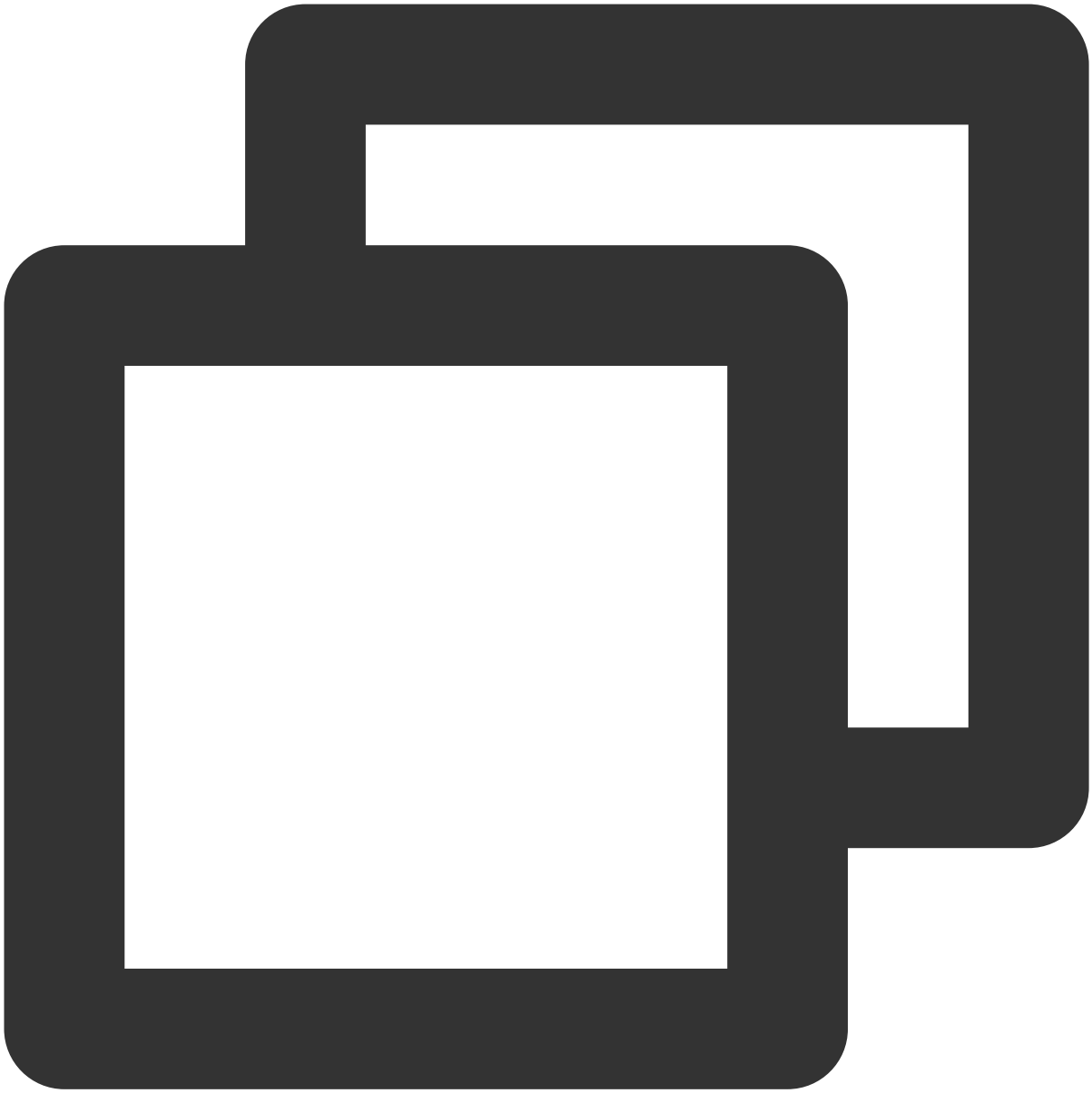
响应



```
HTTP/1.1 204
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 07:39:34 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2NzgxMzZfMmViMDJhMDlfY2NjOF84NGQz****
```

案例二：使用表单字段指定元数据和 ACL

请求

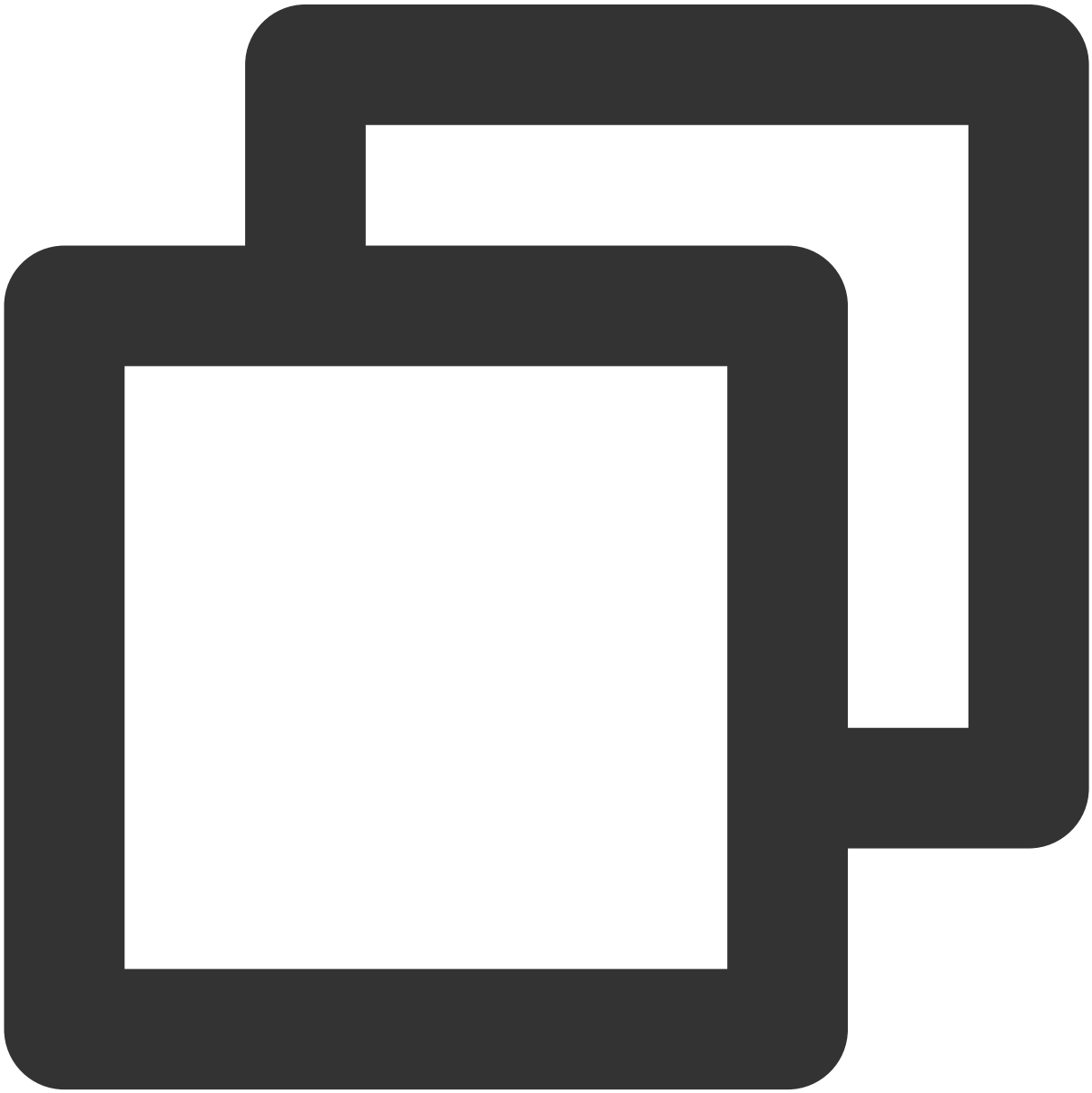


```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 07:39:34 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Length: 2146
Connection: close

------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="key"
```

```
exampleobject
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="acl"

public-read
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="Cache-Control"

max-age=86400
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="Content-Disposition"

attachment; filename=example.jpg
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="Content-Type"

image/jpeg
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="x-cos-meta-example-field"

example-value
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="Content-MD5"

7o3pGNBWQBRbGPcPTDqmAg==
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJhY2wiOiJwdWJsaWMtcmVhZCJ9LHsiYnVja2V0IjoiZXhhbXBsZWJ1Y2tldC0
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="q-key-time"

1567064374;1567071574
------WebKitFormBoundary9JtEhEGHSdx8Patg
Content-Disposition: form-data; name="q-signature"

228a89b5f7b8fce7fdfa4a3b36cfb5a5eafb****
------WebKitFormBoundary9JtEhEGHSdx8Patg
```

```
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundary9JtEhEGHSdx8Patg--
```

响应

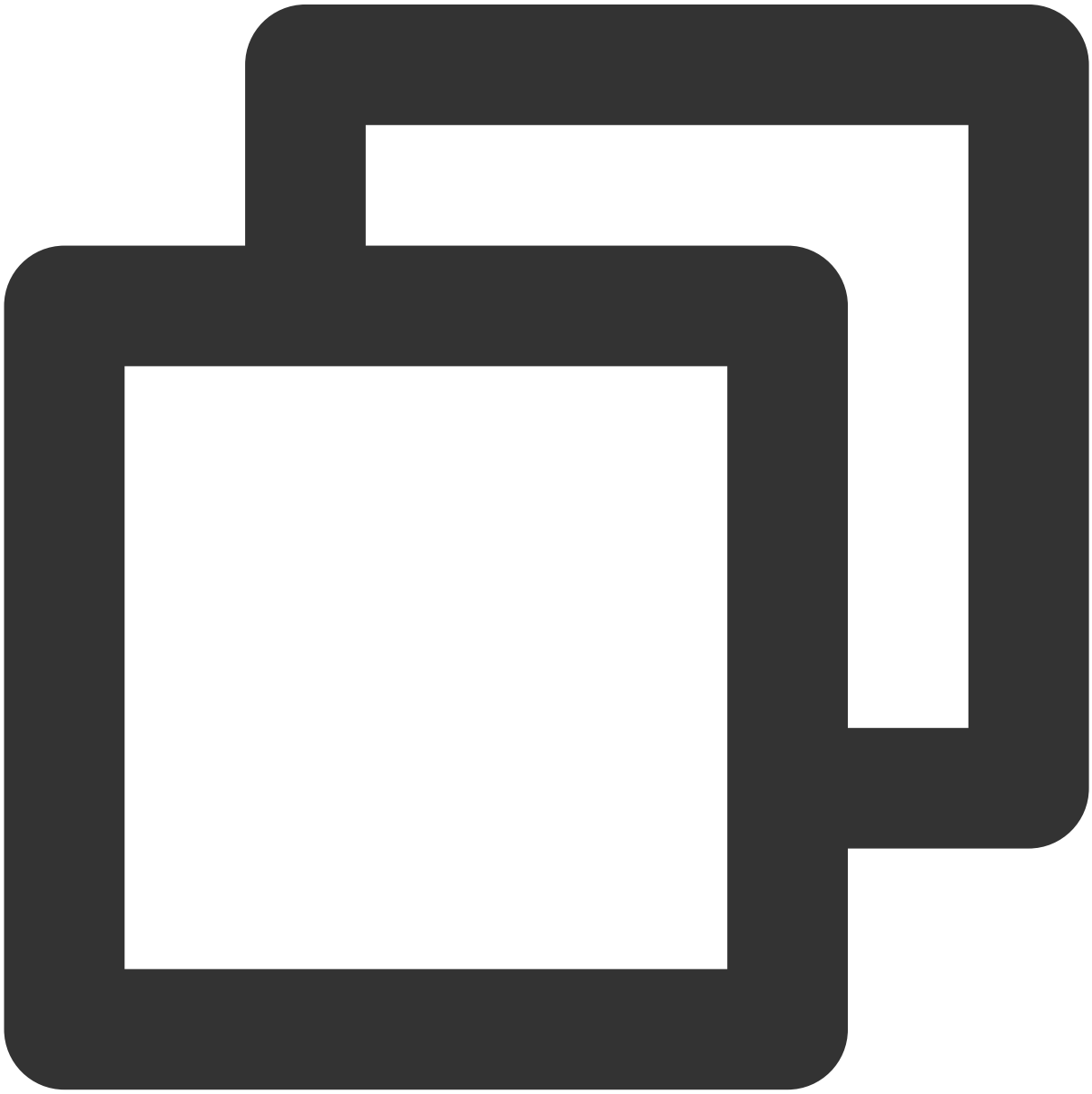


```
HTTP/1.1 204
Content-Length: 0
```

```
Connection: close
Date: Thu, 29 Aug 2019 07:39:35 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2NzgxMzdfM2NhZjJhMDlfMTQzYV84Nzhh****
```

案例三：使用服务端加密 SSE-COS

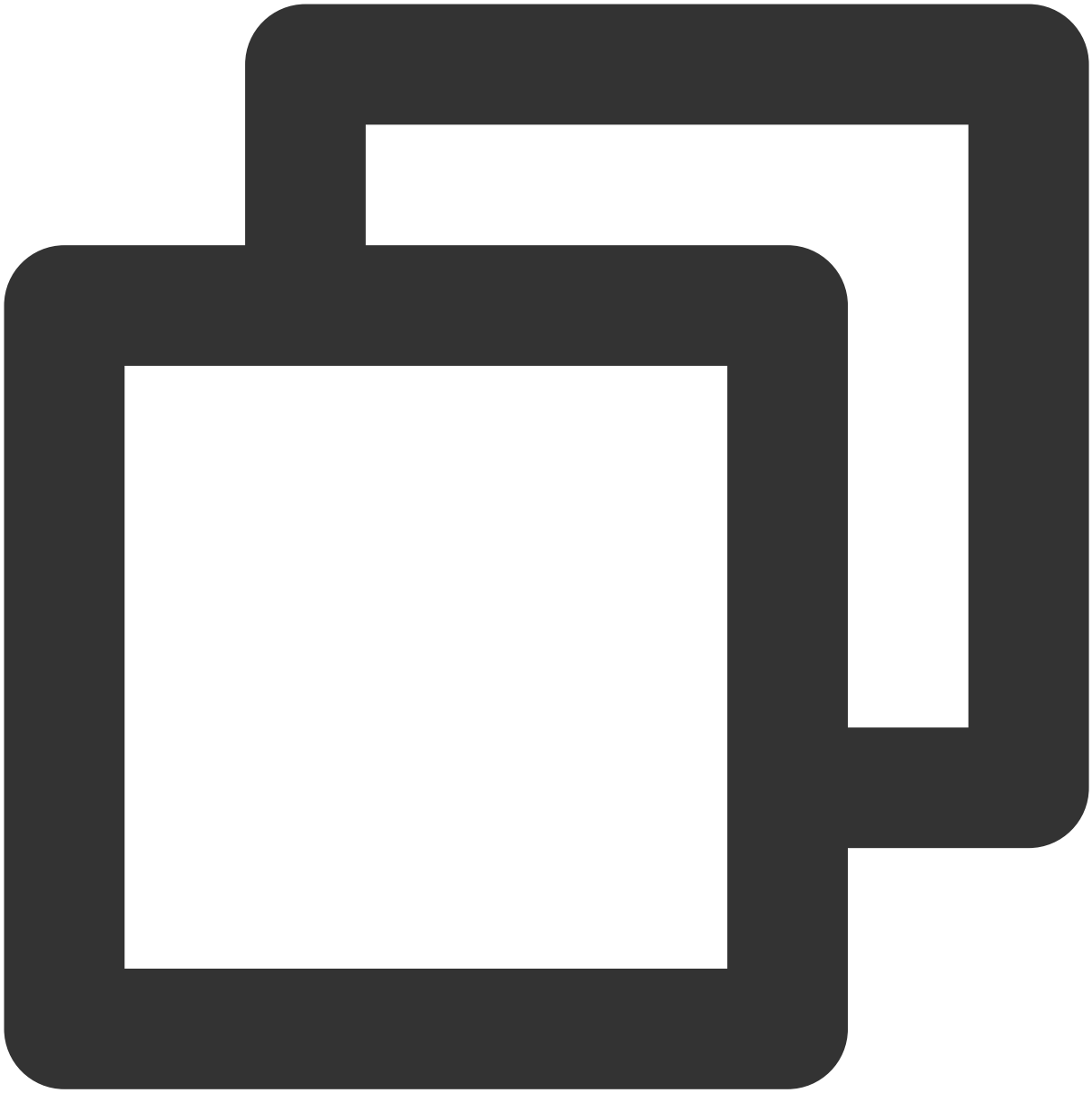
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 07:39:35 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Length: 1296
Connection: close

------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="key"

exampleobject
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="x-cos-server-side-encryption"

AES256
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJ4LWNvcy1zZXJ2ZXItc2lkZS1lbmNyeXB0aW9uIjoiQUVTMjU2In0seyJxLXN
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="q-key-time"

1567064375;1567071575
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="q-signature"

65f3f8864bb1b271e1235d1ec7d1cb508ffa****
------WebKitFormBoundaryBVaHvBJQJnQrAxKY
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryBVaHvBJQJnQrAxKY--
```

响应

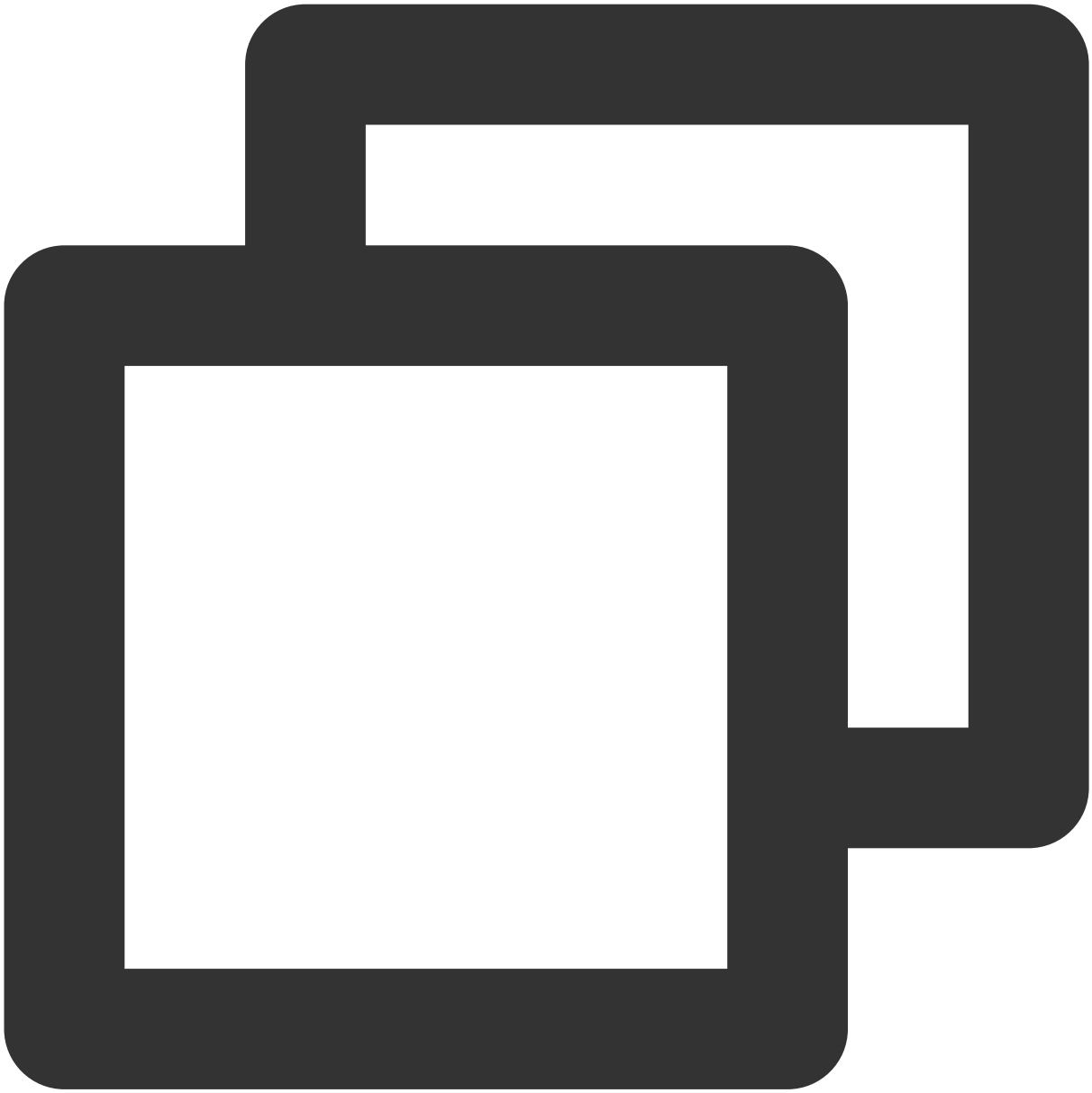
版权所有：腾讯云计算（北京）有限责任公司



```
HTTP/1.1 204
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 07:39:35 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2NzgxMzdfMTljMDJhMDlfNTg4ZF84Njgx****
x-cos-server-side-encryption: AES256
```

案例四：使用服务端加密 SSE-C

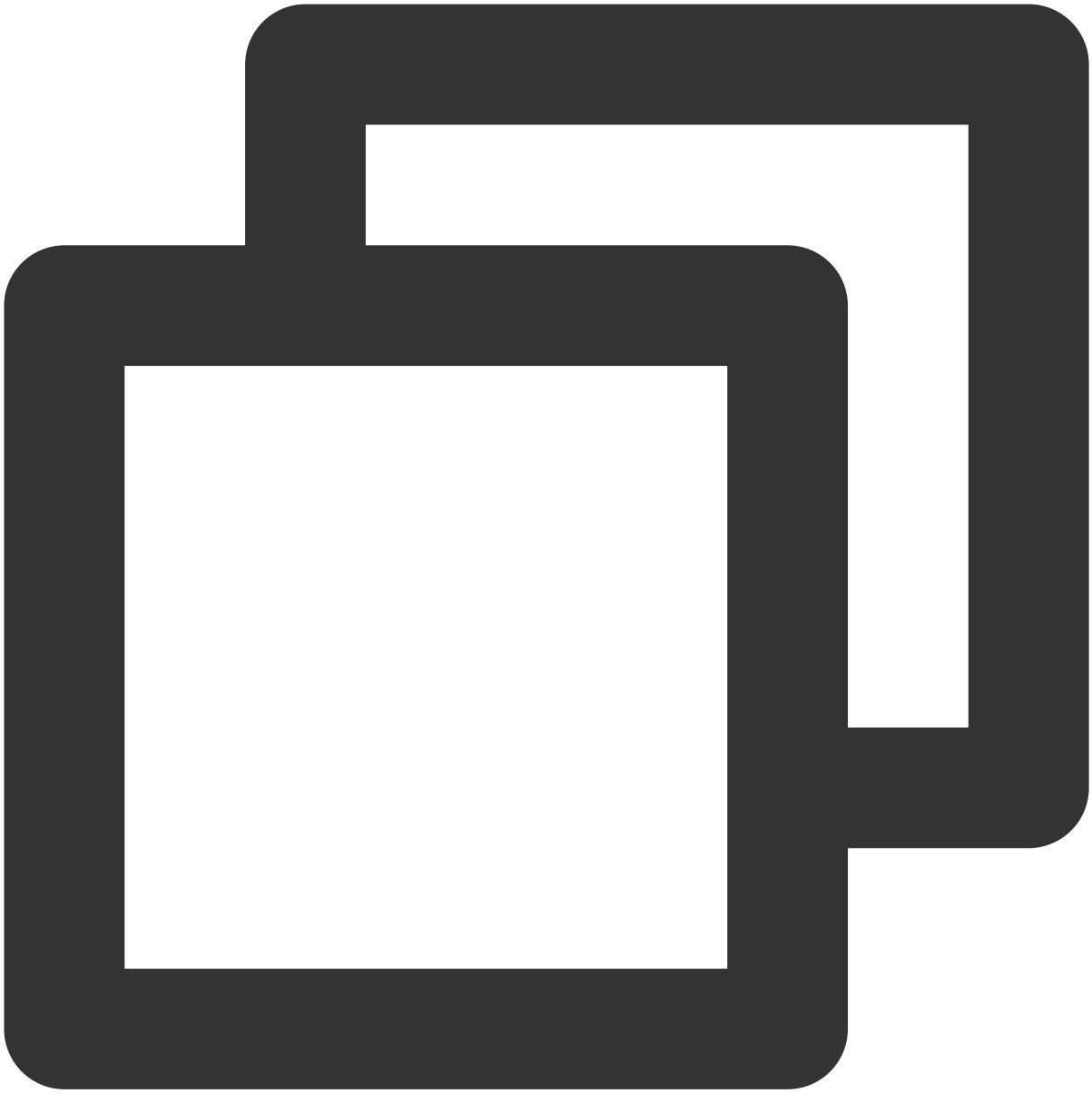
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 07:39:36 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Length: 1667
Connection: close
```

```
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="key"

exampleobject
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="x-cos-server-side-encryption-customer-algorit

AES256
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="x-cos-server-side-encryption-customer-key"

MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="x-cos-server-side-encryption-customer-key-MD5

U5L61r7jcwdNvT7frmUG8g==
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJ4LWNvcy1zZXJ2ZXItc2lkZS1lbmNyeXB0aW9uLWN1c3RvbWVyLWFsZ29yaXR
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="q-key-time"

1567064376;1567071576
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="q-signature"

0273a4b4ede39d0e5162758e145ea0c3e9ef****
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryYa6H7Gd4xuhlyfJb--
```

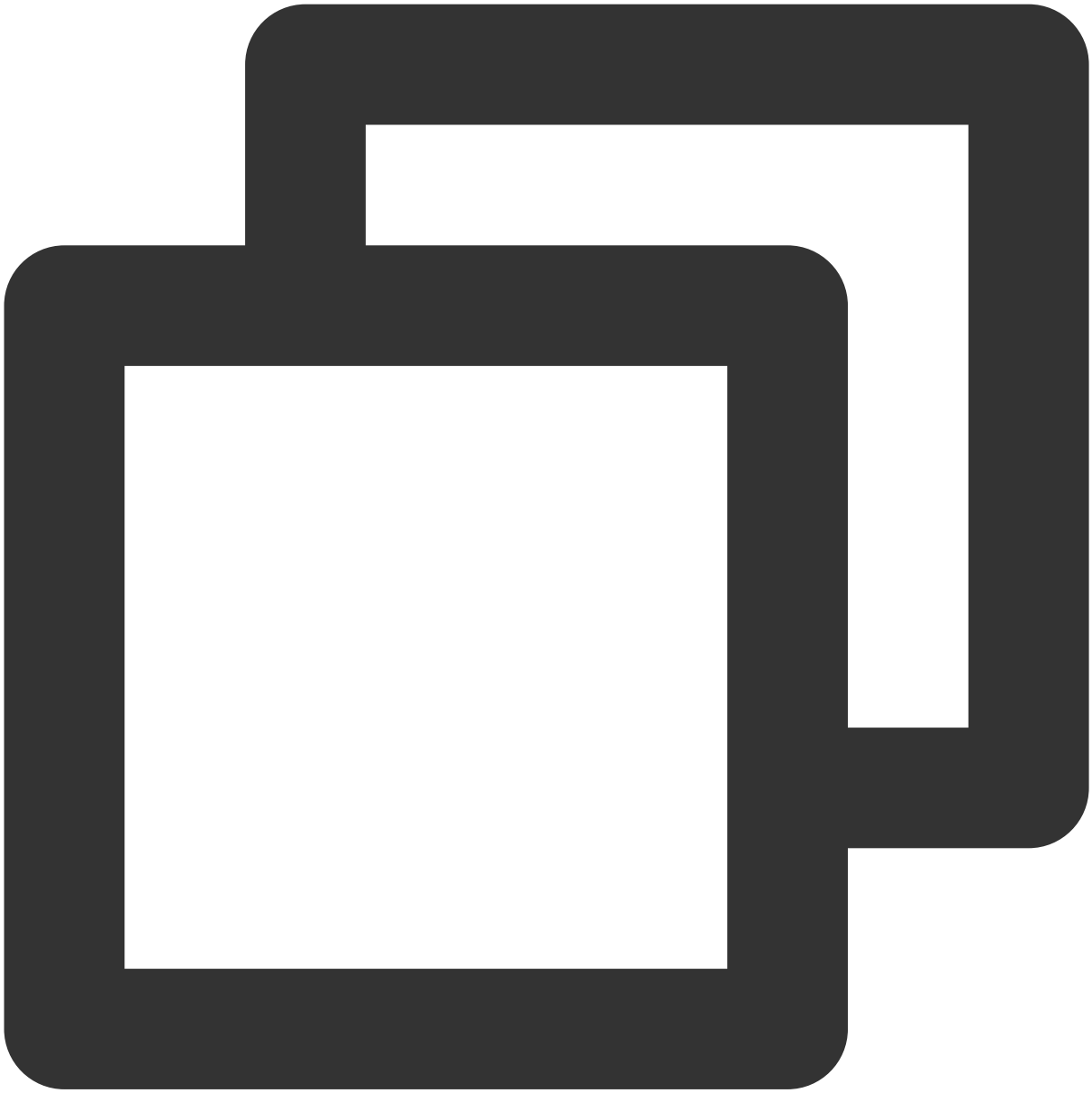
响应



```
HTTP/1.1 204
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 07:39:36 GMT
ETag: "582d9105f71525f3c161984bc005efb5"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2NzgxMzhfMzdiMDJhMDlfNDA4YV84MzQx****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
```

案例五：启用版本控制

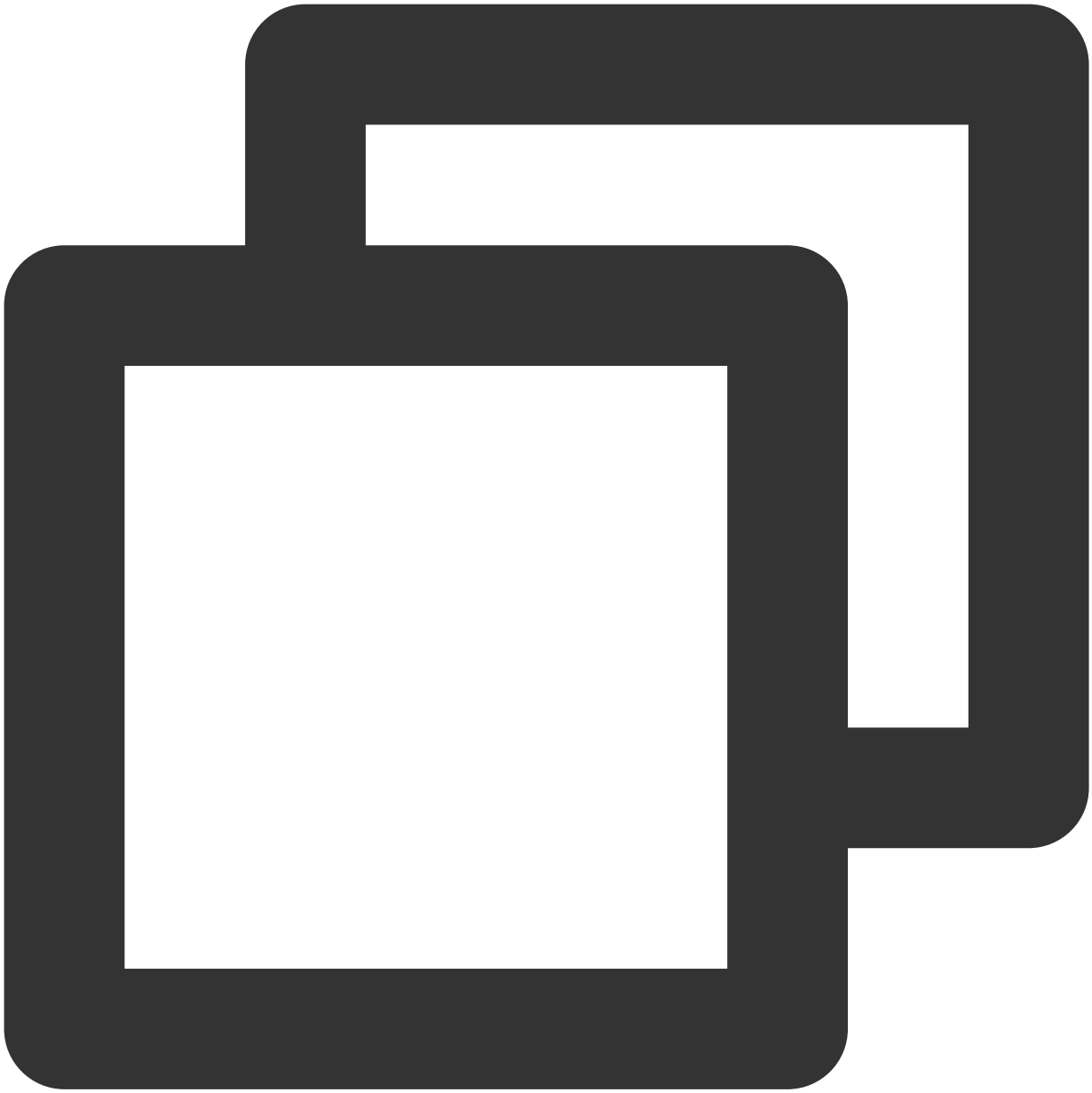
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 07:40:07 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Length: 1119
Connection: close
```

```
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="key"

exampleobject
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJxLXNpZ24tYWxnb3JpdGhtIjoic2hhMSJ9LHsicS1hayI6IkFLSUQ4QTBmQlZ
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="q-key-time"

1567064407;1567071607
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="q-signature"

699ad0ce7780eb559b75e88f77e95743d829****
------WebKitFormBoundaryJspR3QIUhGJLALwf
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryJspR3QIUhGJLALwf--
```

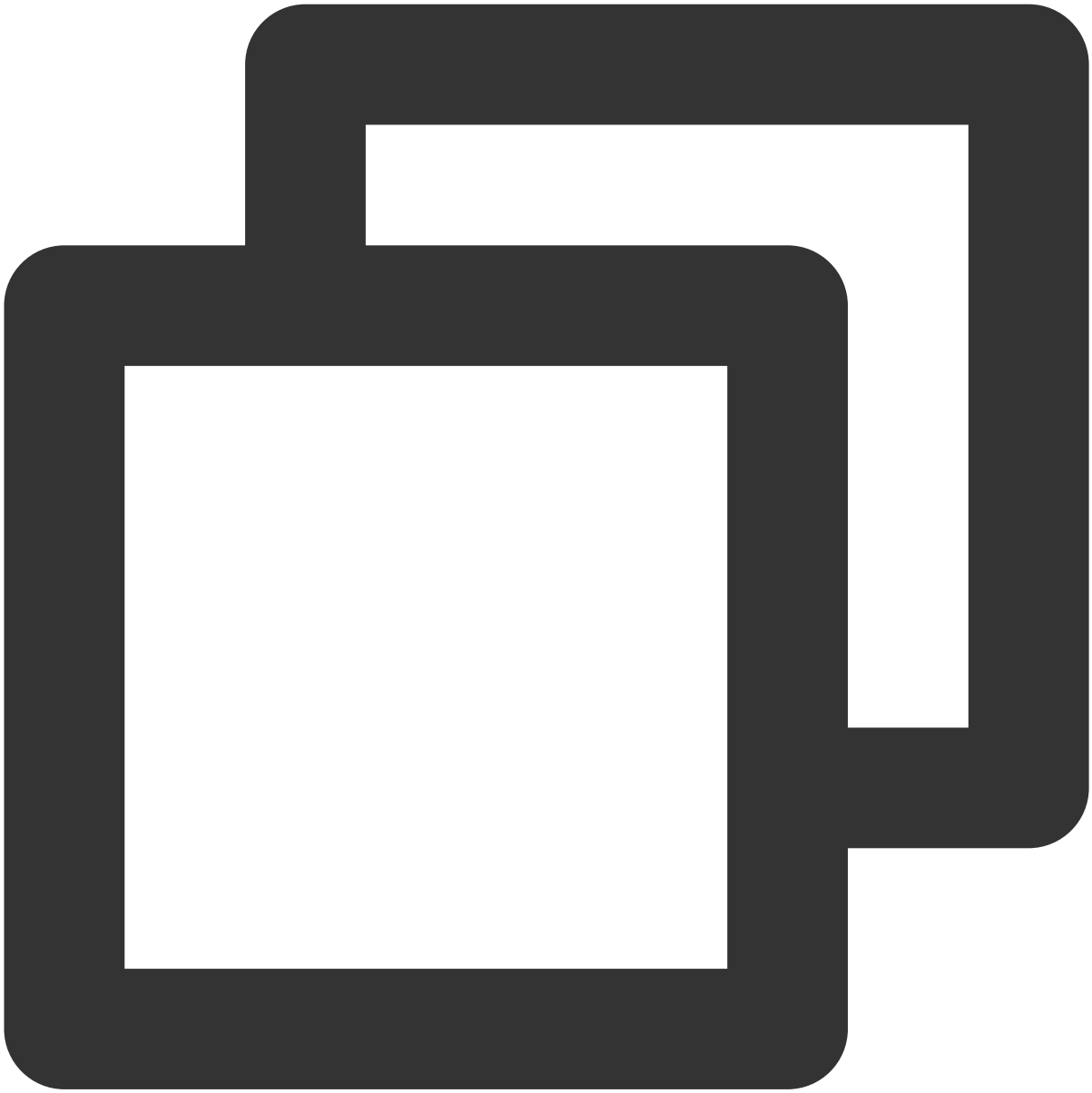
响应



```
HTTP/1.1 204
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 07:40:07 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2NzgxNTdfNzFiNDBiMDlfMmE3ZmJfODQ1****
x-cos-version-id: MTg0NDUxNzcwMDkzMDE3NDQ0MDU
```

案例六：暂停版本控制

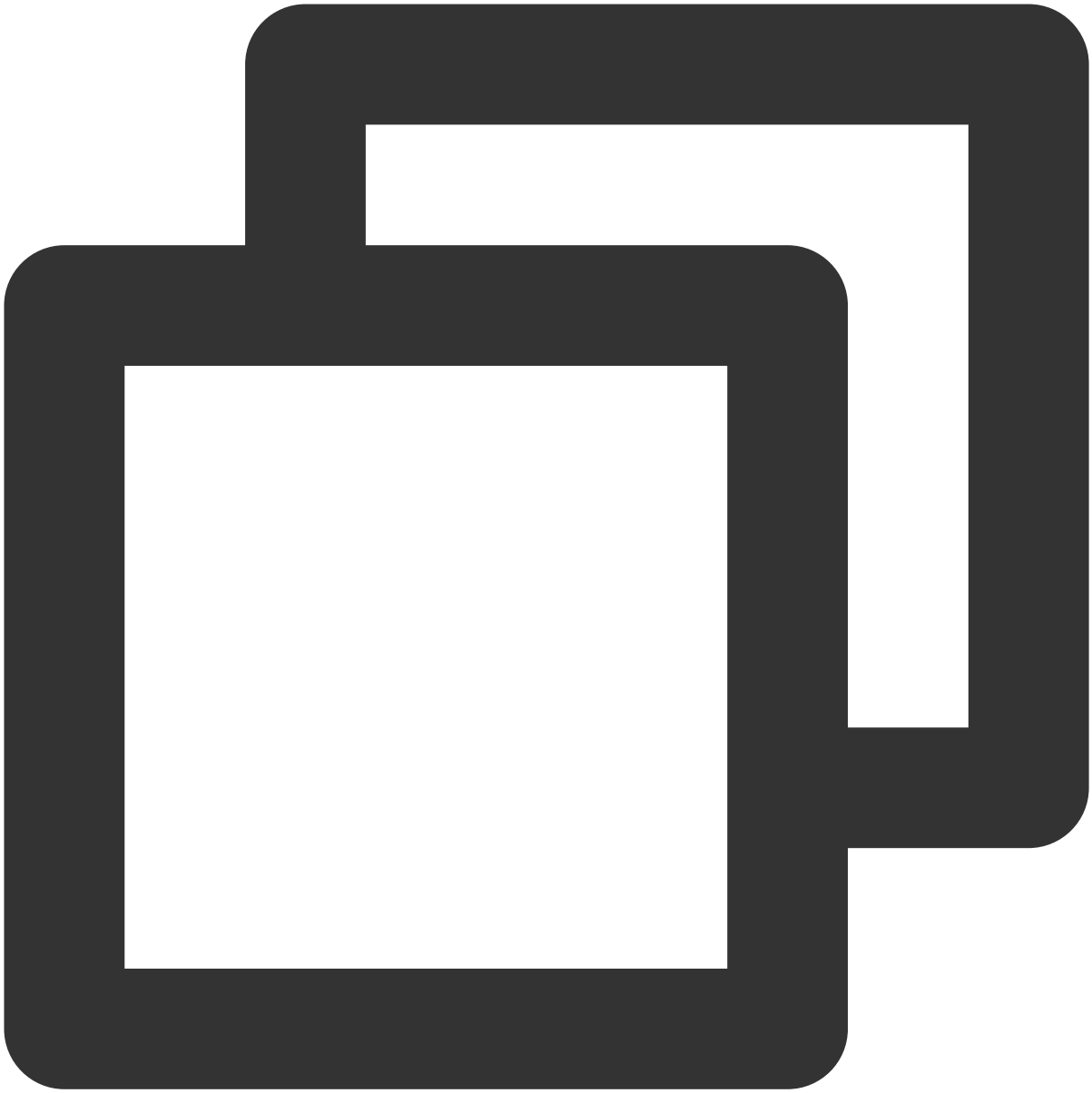
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 07:40:38 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Length: 1119
Connection: close
```

```
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="key"

exampleobject
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJxLXNpZ24tYWxnb3JpdGhtIjoic2hhMSJ9LHsicS1hayI6IkFLSUQ4QTBmQlZ
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="q-key-time"

1567064438;1567071638
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="q-signature"

bb04222322bfb17f4d1f43833bbbac0a03aa****
------WebKitFormBoundaryX8hd2lxTMzIBk5Li
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryX8hd2lxTMzIBk5Li--
```

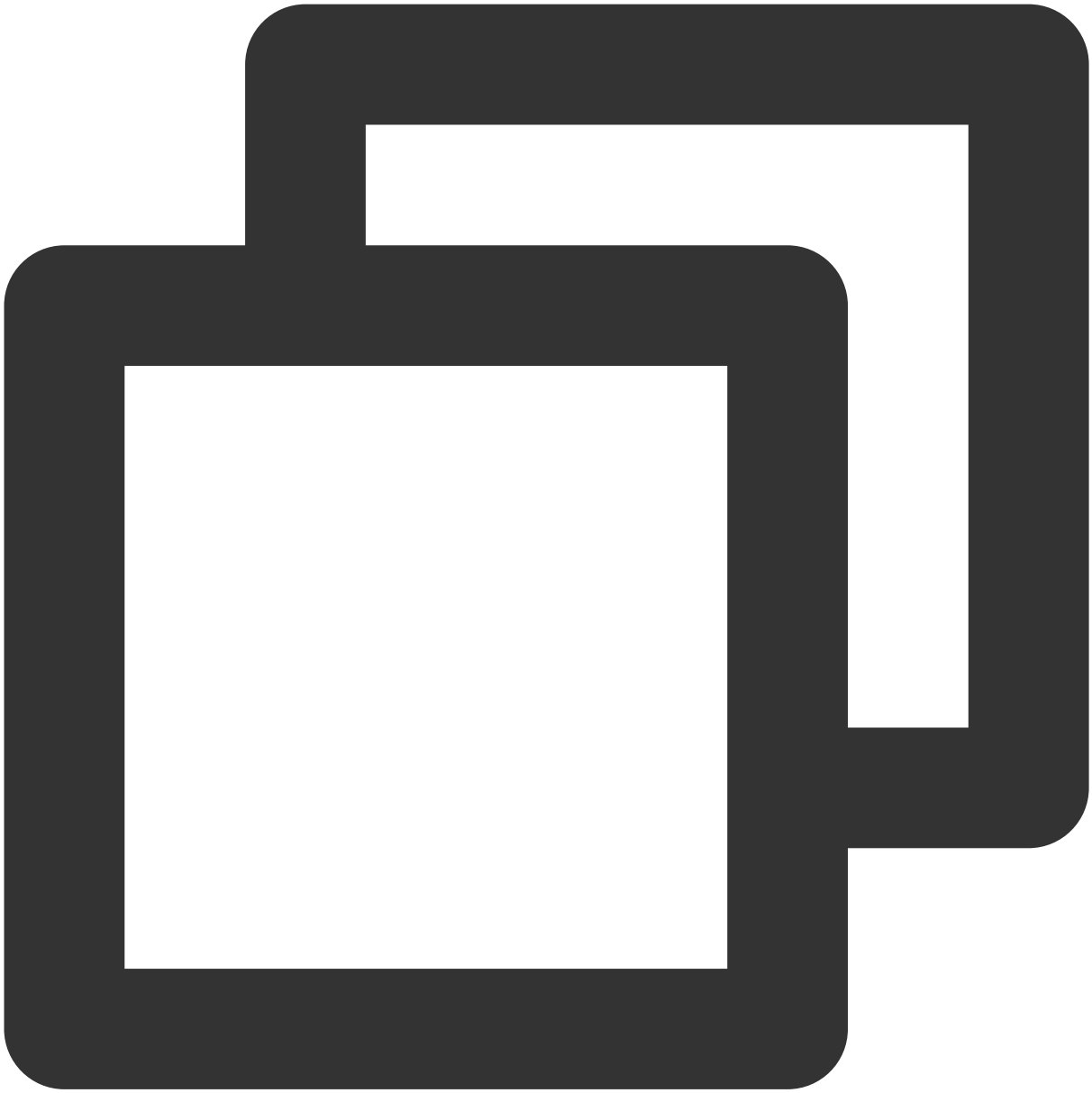
响应



```
HTTP/1.1 204
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 07:40:38 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2NzgxNzZfMjFjOTBiMDlfMWY3YTFfNjY2****
```

案例七：对象键（表单字段 key）使用 `${filename}` 通配符

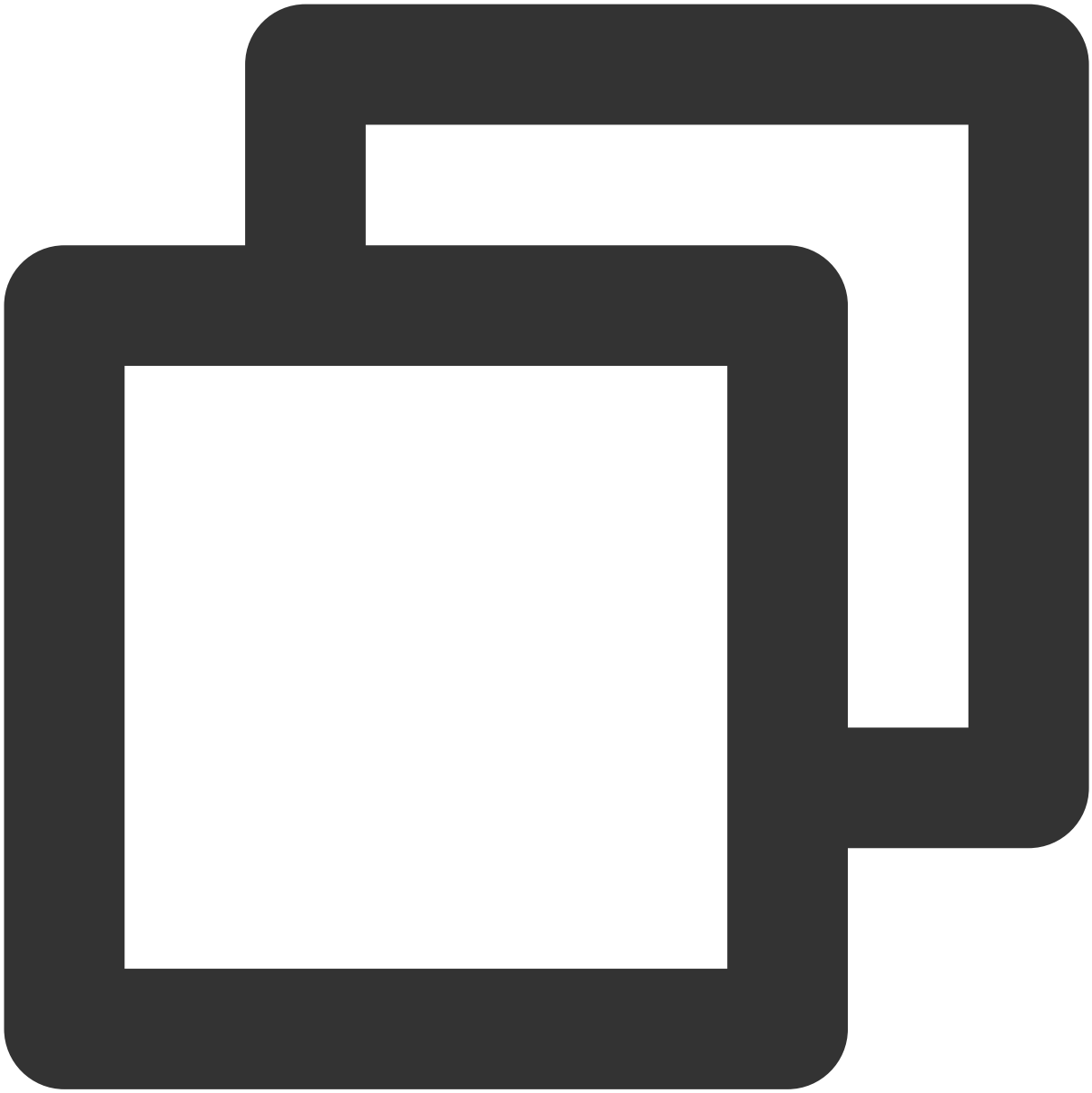
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 12:35:07 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Length: 1188
Connection: close

------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="key"
```

```
folder/subfolder/${filename}
------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbWyJzdGFydHMtd2l0aCIsIiRrZXkiLCJmb2xkZXIvc3ViZm9sZGVyLyJdLHsicS1
------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="q-key-time"

1567082107;1567089307
------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="q-signature"

3cc37f8c81e36f57506efa02d0a3b6c9d551****
------WebKitFormBoundaryHrAMWZO4BNyT0rca
Content-Disposition: form-data; name="file"; filename="photo.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryHrAMWZO4BNyT0rca--
```

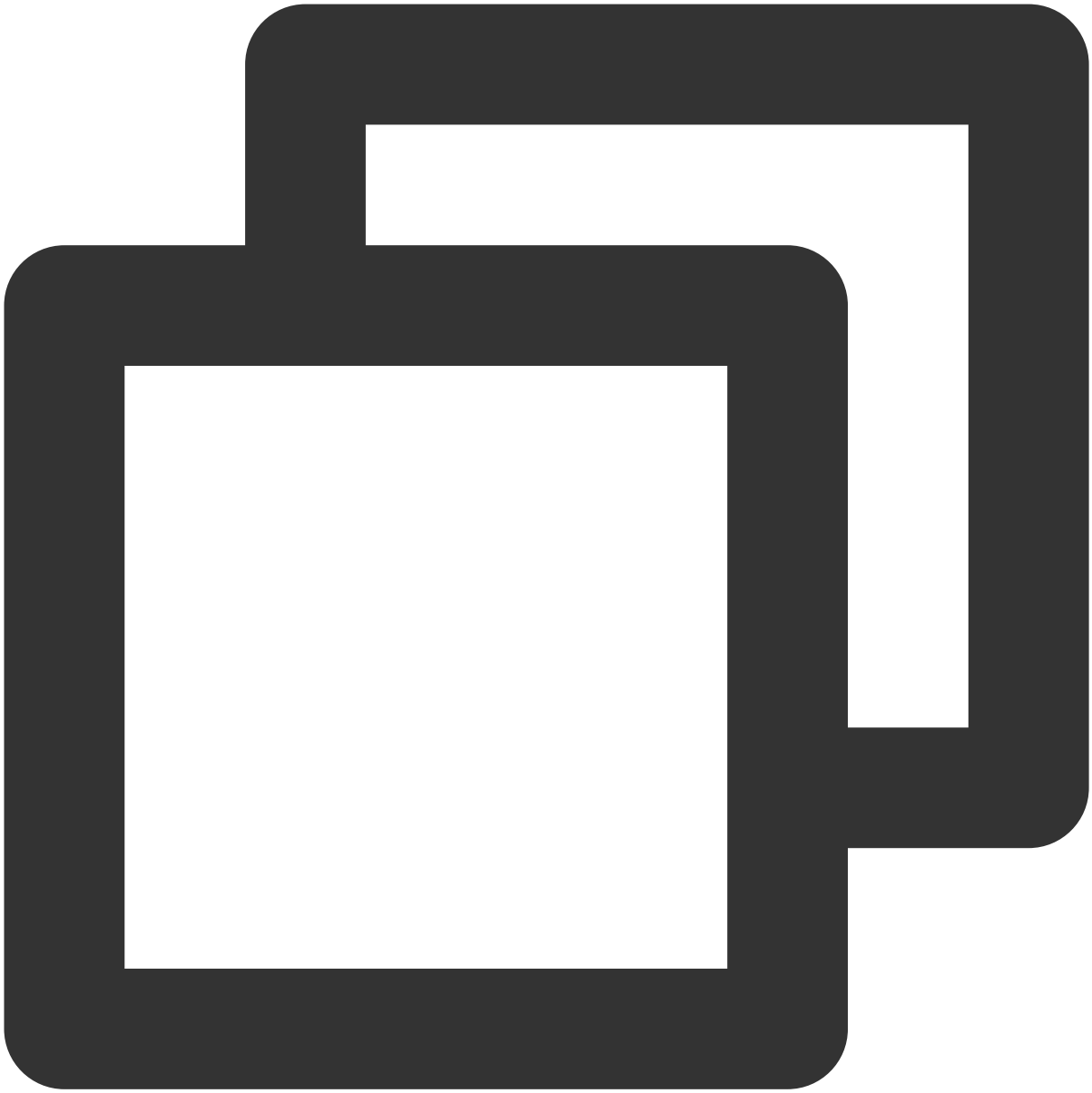
响应



```
HTTP/1.1 204
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 12:35:08 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/folder/subfol
Server: tencent-cos
x-cos-request-id: NWQ2N2M2N2NfNWZhZjJhMDlfNmUzMV84OTg4****
```

案例八：指定 success_action_redirect 表单字段

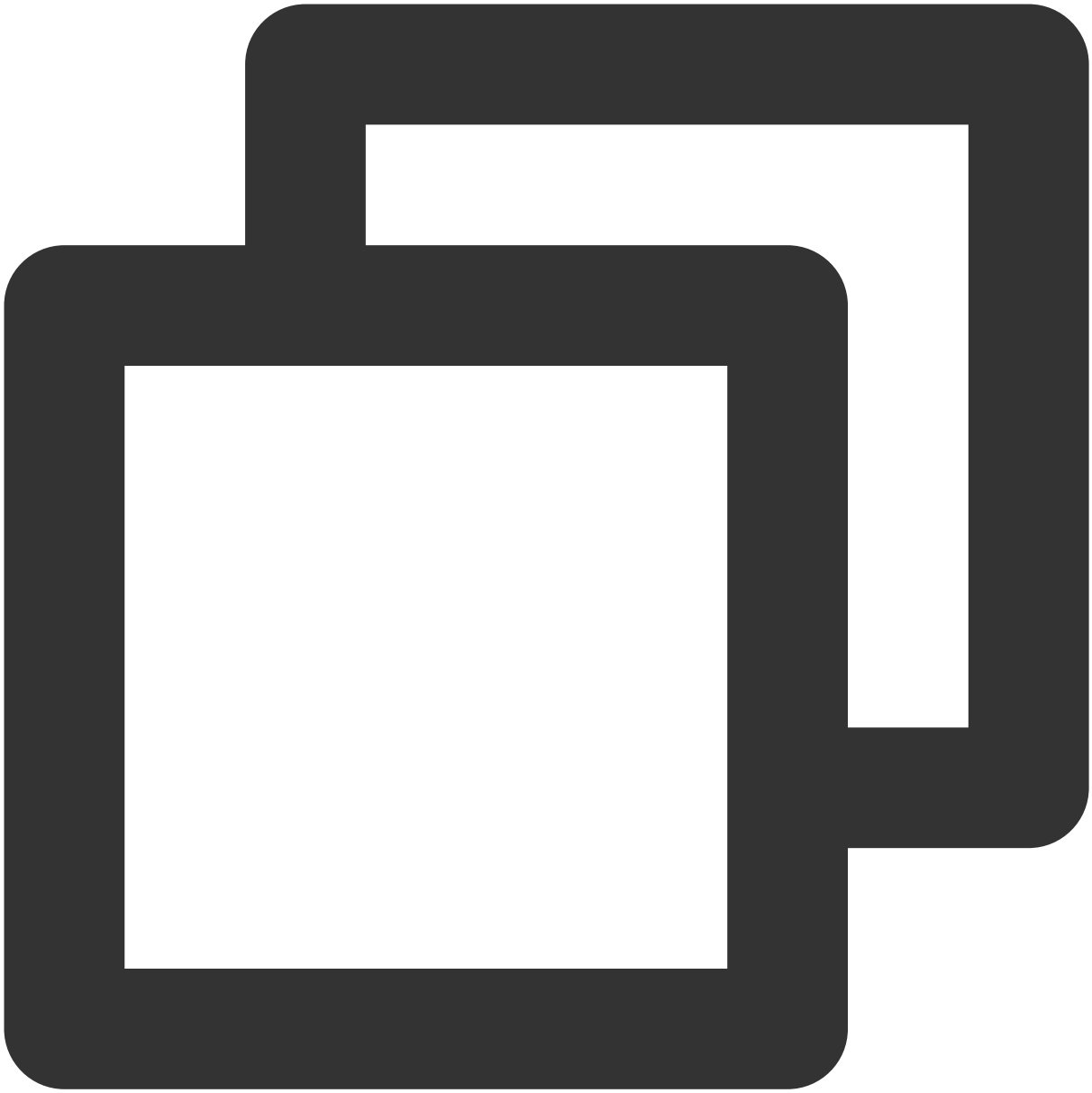
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 08:02:29 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Length: 1351
Connection: close

------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="key"
```

```
exampleobject
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="success_action_redirect"

https://my.website/upload_success.html
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbWyJzdGFydHMtd2l0aCIsIiRzdWNjZXNzX2FjdGlvbl9yZWRpcmVjdCIsImh0dHB
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="q-key-time"

1567065749;1567072949
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="q-signature"

c4a8ae7411687bc3d6ed2ac9b249e87a50b5****
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryJ0bRH1MwgMq5eu6H--
```

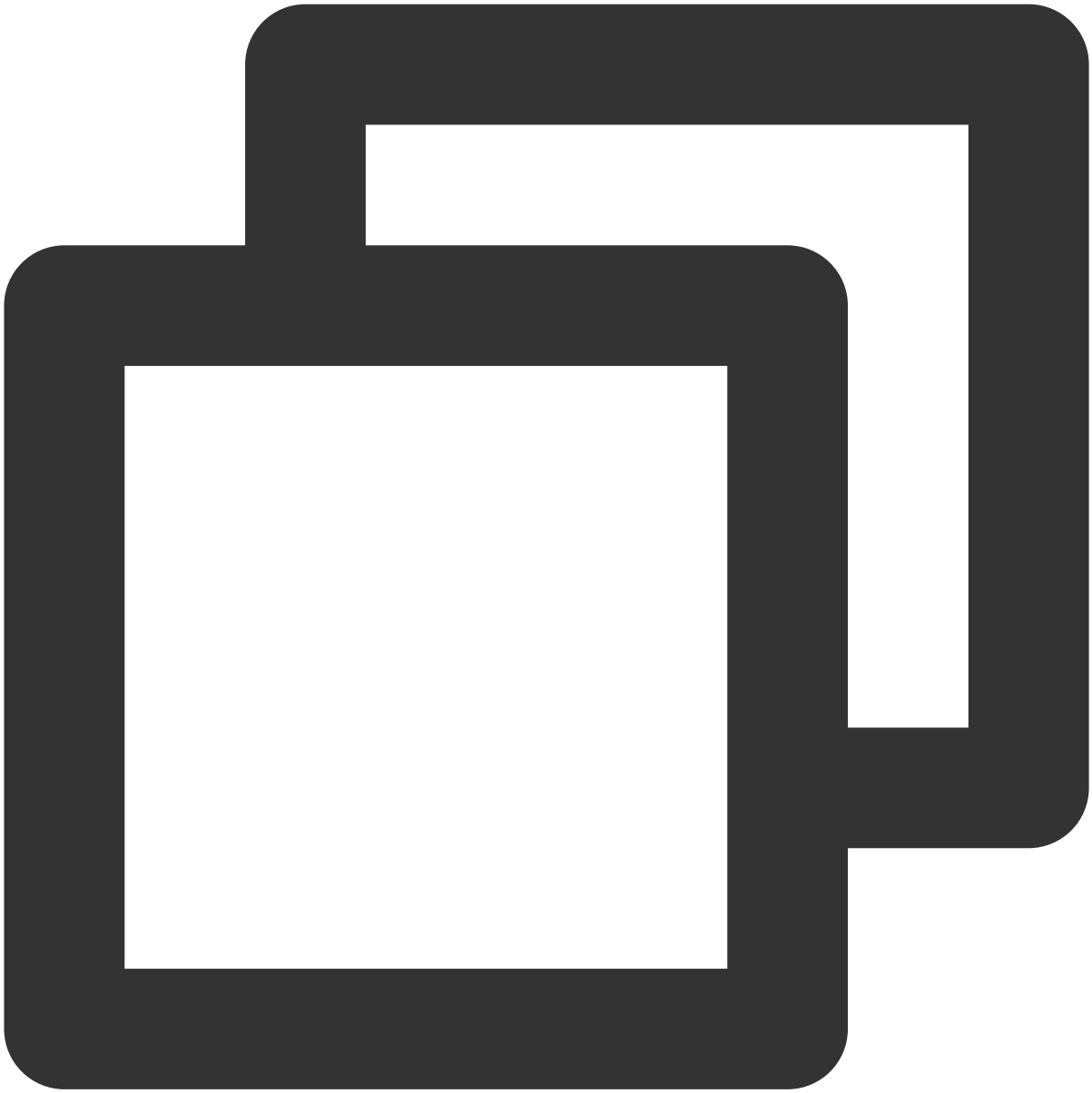
响应



```
HTTP/1.1 303 Redirect
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 08:02:29 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: https://my.website/upload_success.html?bucket=examplebucket-1250000000&ke
Server: tencent-cos
x-cos-request-id: NWQ2Nzg2OTVfMTRiYjI0MDlfZGFkOV85MDA4****
```

案例九：指定 success_action_status 表单字段

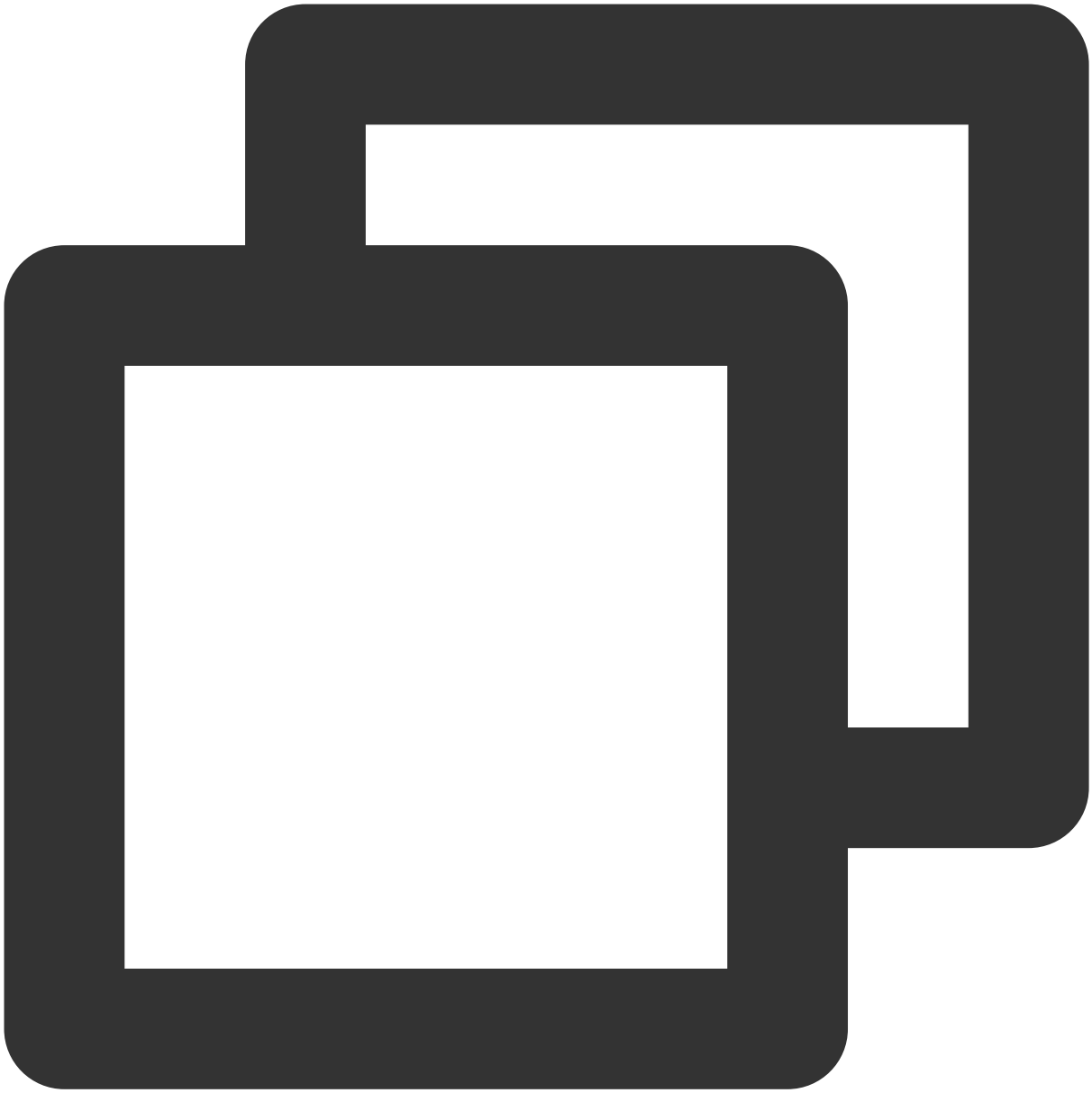
请求



```
POST / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 29 Aug 2019 08:04:29 GMT
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Length: 1270
Connection: close

------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="key"
```

```
exampleobject
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="success_action_status"

200
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="policy"

eyJjb25kaXRpb25zIjpbeyJzdWNjZXNzX2FjdGlvbl9zdGF0dXMiOiIyMDAifSx7InEtc2lnbi1hbGdvcml
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="q-sign-algorithm"

sha1
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="q-ak"

AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="q-key-time"

1567065869;1567073069
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="q-signature"

e46285af04d4fb68e0624fdd0a525b6a07ab****
------WebKitFormBoundaryST9Mz8AGzCDphgJF
Content-Disposition: form-data; name="file"; filename="example.jpg"
Content-Type: image/jpeg

[Object Content]
------WebKitFormBoundaryST9Mz8AGzCDphgJF--
```

响应



```
HTTP/1.1 200
Content-Length: 0
Connection: close
Date: Thu, 29 Aug 2019 08:04:29 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Location: http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject
Server: tencent-cos
x-cos-request-id: NWQ2Nzg3MGRfZjhjODBiMDlfOGM3N184Nzdl****
```

GET Object

最近更新时间：2023-12-21 14:31:55

功能描述

GET Object 接口请求可以将 COS 存储桶中的对象（Object）下载至本地。该 API 的请求者需要对目标对象有读取权限，或者目标对象向所有人开放了读取权限（公有读）。

注意

如果使用了 `response-*` 请求参数，那么该请求操作不支持匿名请求，必须携带签名。

当通过 COS 控制台 设置回源，但未开启**同步回源**时，需注意，COS 从用户配置的源站拉取数据时，发起 GET Object 请求将返回 302 并重定向到设置的回源地址（如果该回源地址是不受信任的，强烈建议在使用 SDK 或自行调用 API 时，不要直接跟随 302，而应该由业务后端验证回源地址的合法性后再去请求回源地址，否则可能产生 SSRF 等安全风险，例如回源到一个内网地址）。

版本控制

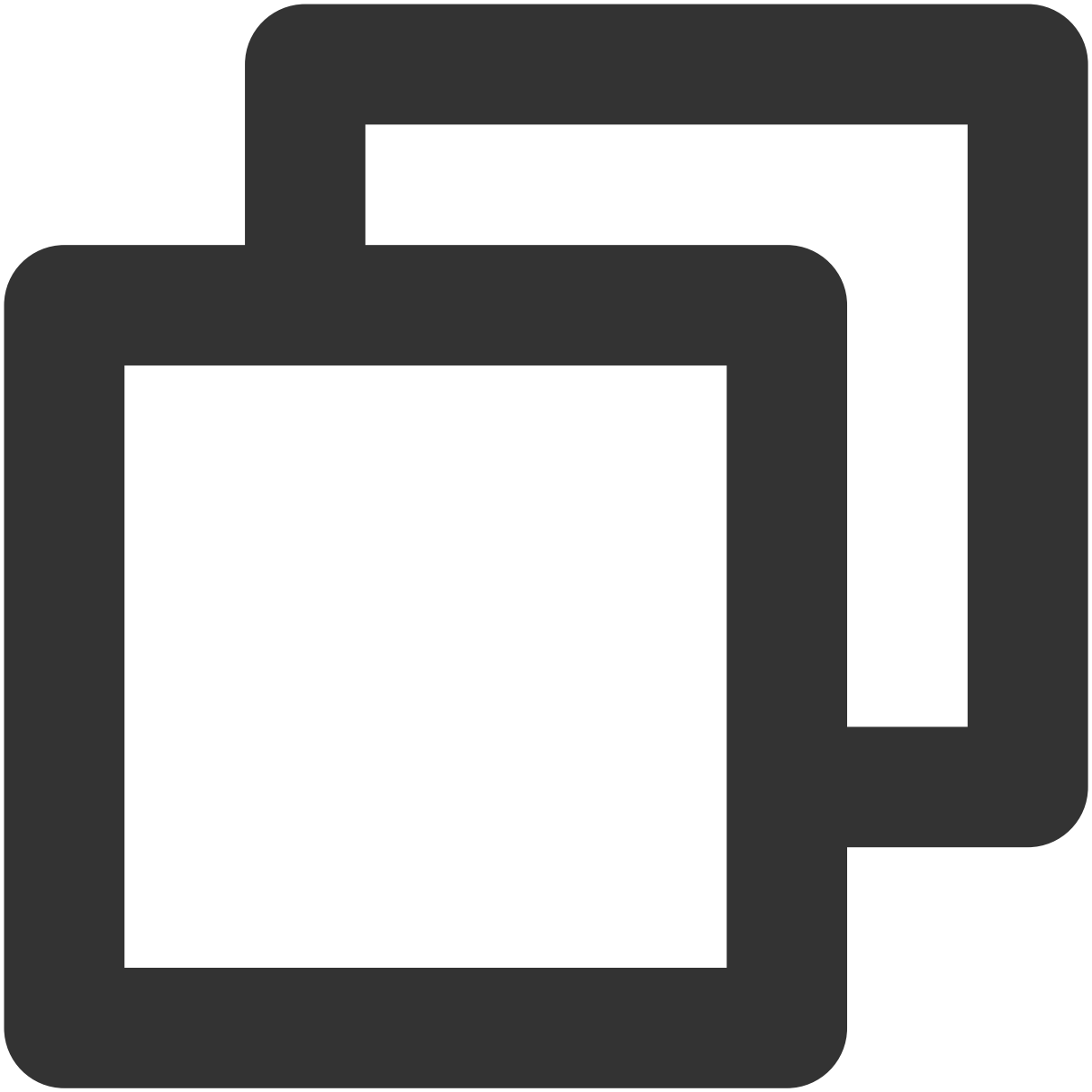
当启用版本控制时，该 GET 操作可以使用 versionId 请求参数指定要返回的版本 ID，此时将返回对象的指定版本。若指定版本为删除标记，则返回 HTTP 响应码404（Not Found），否则将返回指定对象的最新版本。

归档类型

如果该 GET 请求操作的对象为**归档存储和深度归档存储类型**，且没有使用 POST Object restore 进行恢复（或恢复后的副本已被过期删除），那么该请求将返回 HTTP 响应码403（Forbidden），同时在响应体中包含错误信息，其中错误码（Code）为 InvalidObjectState，表示对象的当前状态无法被 GET 请求操作，需要先经过恢复。

请求

请求示例



```
GET /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

名称	描述	类型	是否必选
response-cache-control	设置响应中的 Cache-Control 头部的值	string	否
response-content-disposition	设置响应中的 Content-Disposition 头部的值	string	否
response-content-encoding	设置响应中的 Content-Encoding 头部的值	string	否
response-content-language	设置响应中的 Content-Language 头部的值	string	否
response-content-type	设置响应中的 Content-Type 头部的值	string	否
response-expires	设置响应中的 Expires 头部的值	string	否
versionId	当启用版本控制时，指定要下载的版本 ID，如不指定则下载对象的最新版本	string	否

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
Range	RFC 2616 中定义的字节范围，范围值必须使用 bytes=first-last 格式且仅支持单一范围，不支持多重范围。first 和 last 都是基于0开始的偏移量。 例如 bytes=0-9，表示下载对象的开头10个字节的数据；bytes=5-9，表示下载对象的第6到第10个字节。此时返回 HTTP 状态码206（Partial Content）及 Content-Range 响应头部。 如果 first 超过对象的大小，则返回 HTTP 状态码416（Requested Range Not Satisfiable）错误。如果不指定，则表示下载整个对象	string	否
If-Modified-Since	当对象在指定时间后被修改，则返回对象，否则返回 HTTP 状态码为304（Not Modified）	string	否
If-Unmodified-Since	当对象在指定时间后未被修改，则返回对象，否则返回 HTTP 状态码为412（Precondition Failed）	string	否
If-Match	当对象的 ETag 与指定的值一致，则返回对象，否则返回 HTTP 状态码为412（Precondition Failed）	string	否

版权所有：腾讯云计算（北京）有限责任公司

If-None-Match	当对象的 ETag 与指定的值不一致，则返回对象，否则返回 HTTP 状态码为304（Not Modified）	string	否
x-cos-traffic-limit	针对本次下载进行流量控制的限速值，必须为数字，单位默认为 bit/s。限速值设置范围为819200 - 838860800，即100KB/s - 100MB/s，如果超出该范围将返回400错误	integer	否

服务端加密相关头部

如果指定的对象使用了服务端加密且加密方式为 SSE-C 时，则需要指定服务端加密的相关头部来解密对象，请参见服务端加密专用头部。

请求体

此接口无请求体。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

名称	描述	类型
Cache-Control	RFC 2616 中定义的缓存指令，仅当对象元数据包含此项或通过请求参数指定了此项时才会返回该头部	string
Content-Disposition	RFC 2616 中定义的文件名称，仅当对象元数据包含此项或通过请求参数指定了此项时才会返回该头部	string
Content-Encoding	RFC 2616 中定义的编码格式，仅当对象元数据包含此项或通过请求参数指定了此项时才会返回该头部	string
Content-Range	RFC 2616 中定义的返回内容的字节范围，仅当请求中指定了 Range 请求头部时才会返回该头部	string
Expires	RFC 2616 中定义的缓存失效时间，仅当对象元数据包含此项或通过请求参数指定了此项时才会返回该头部	string
x-cos-meta-*	包括用户自定义元数据头部后缀和用户自定义元数据信息	string
x-cos-storage-class	对象存储类型，枚举值请参见 存储类型 文档，例如 MAZ_STANDARD、MAZ_STANDARD_IA、INTELLIGENT_TIERING、MAZ_INTELLIGENT_TIERING、STANDARD_IA、ARCHIVE、DEEP_ARCHIVE。仅当对象不是标准存储（STANDARD）时才会返回该头部	enum
x-cos-storage-	当对象的存储类型为智能分层存储时，该头部表示对象所处的存储层，有效	enum

tier	值：FREQUENT、INFREQUENT。	

版本控制相关头部

启用版本控制的存储桶内的对象将返回下列响应头部：

名称	描述	类型
x-cos-version-id	对象的版本 ID	string

服务端加密相关头部

如果指定的对象使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

此接口请求的响应体为对象（文件）内容。

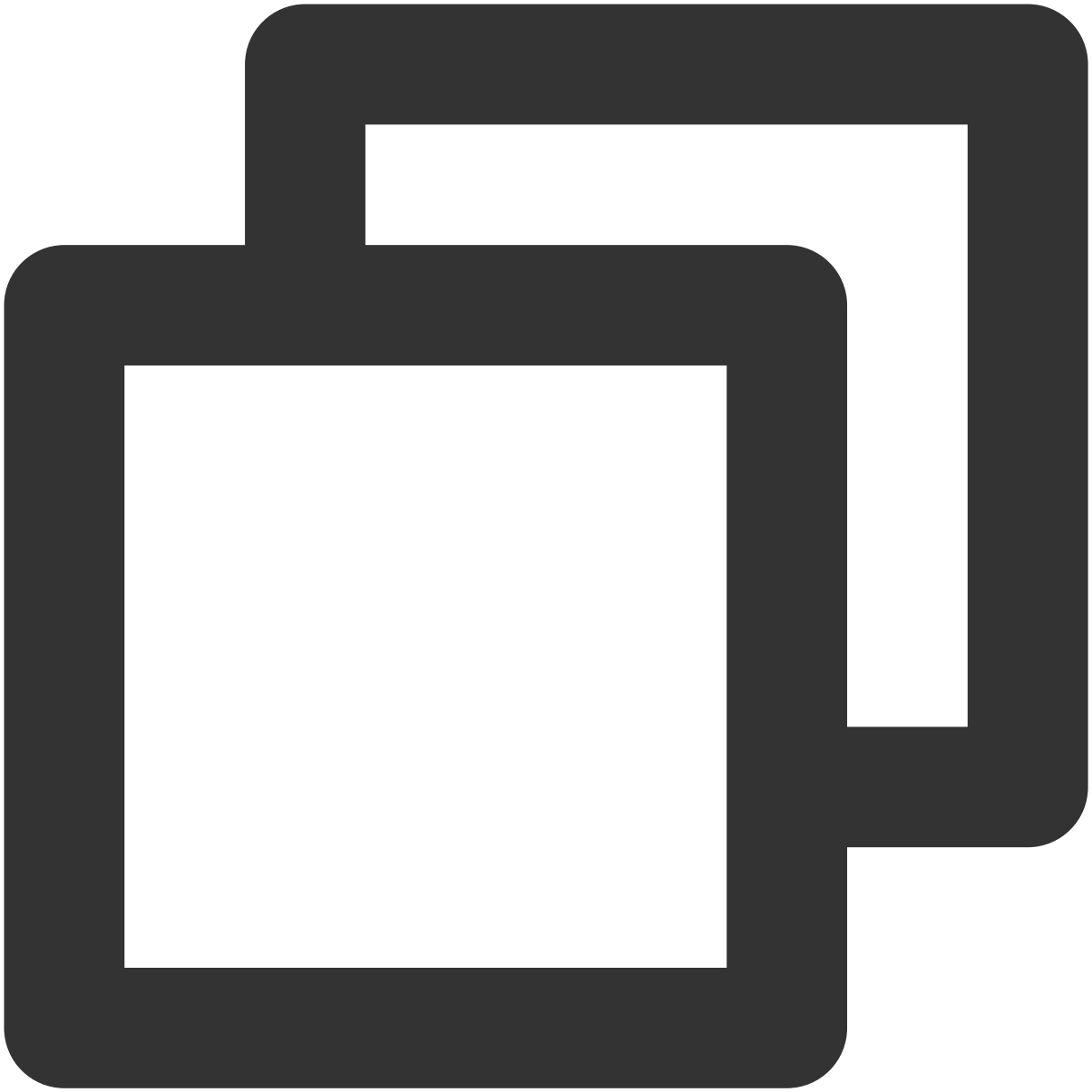
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

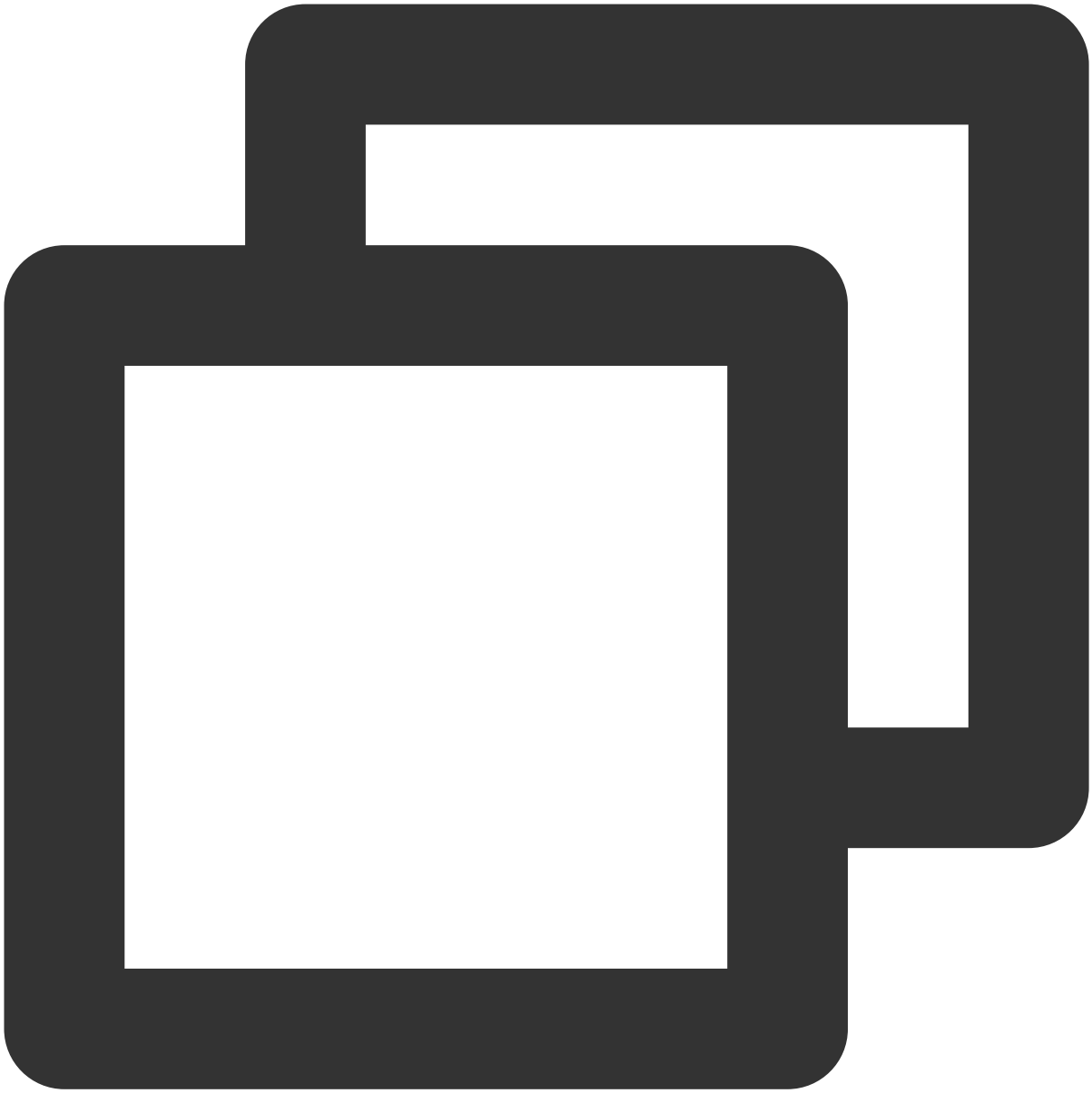
案例一：简单案例（未启用版本控制）

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:35:16 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

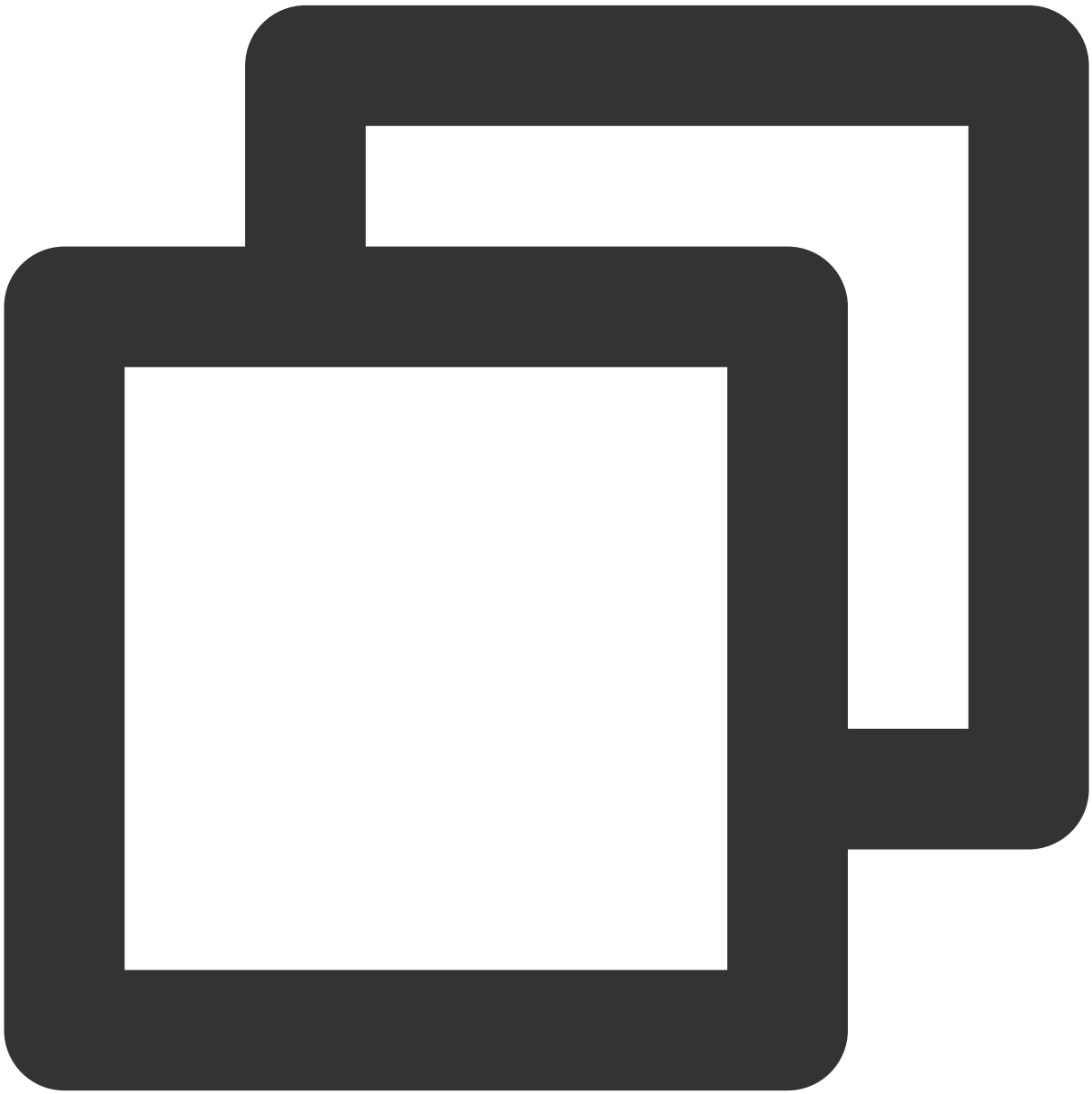


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Accept-Ranges: bytes
Date: Fri, 10 Apr 2020 09:35:16 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 09:35:05 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749656679157681890
x-cos-request-id: NWU5MDNkZDRfZDgyNzVkNjRfN2Q5M18xOWVi****
```

```
[Object Content]
```

案例二：通过请求参数指定响应头部

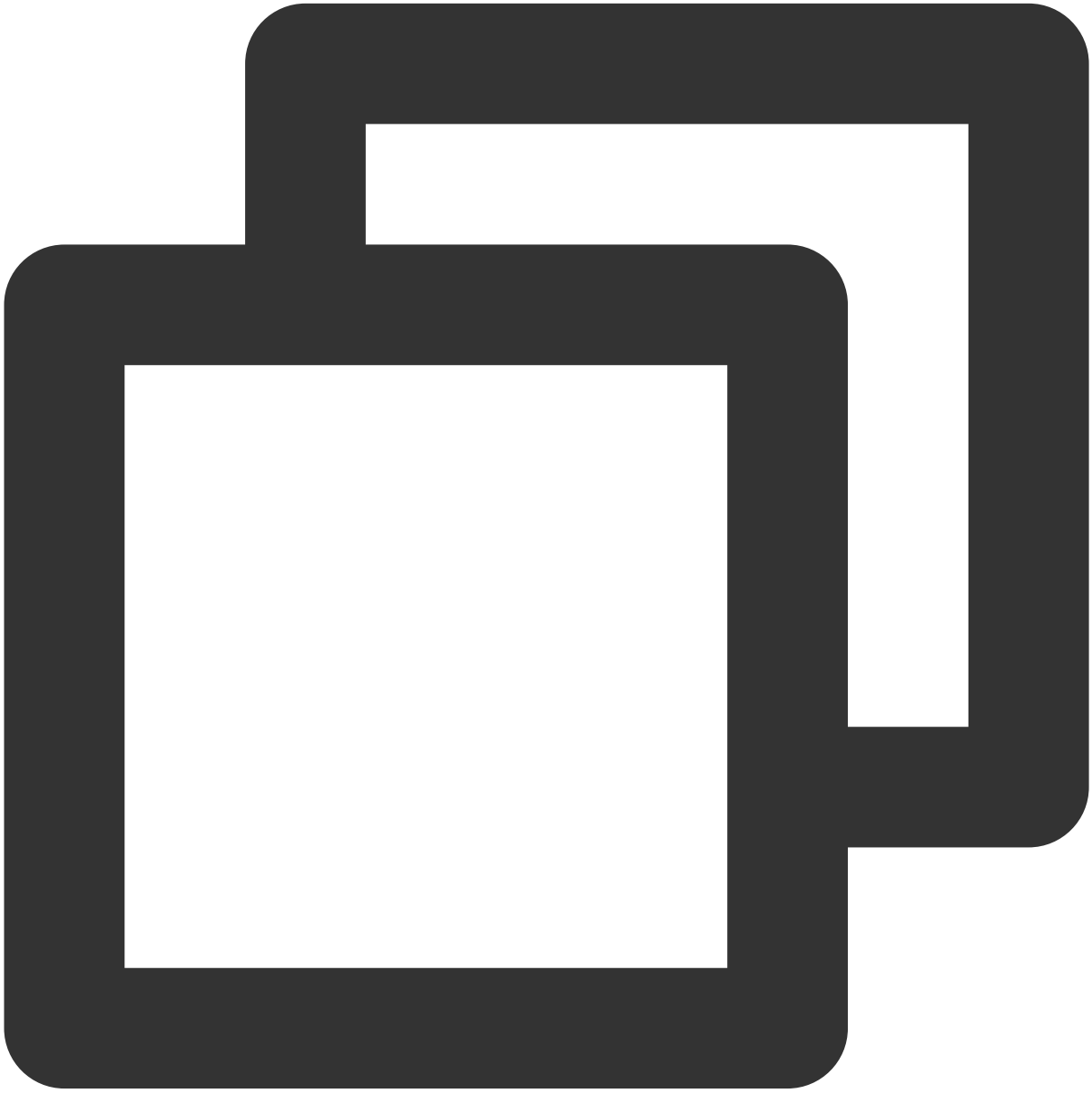
请求



```
GET /exampleobject?response-content-type=application%2Foctet-stream&response-cache-
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Date: Fri, 10 Apr 2020 09:35:17 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

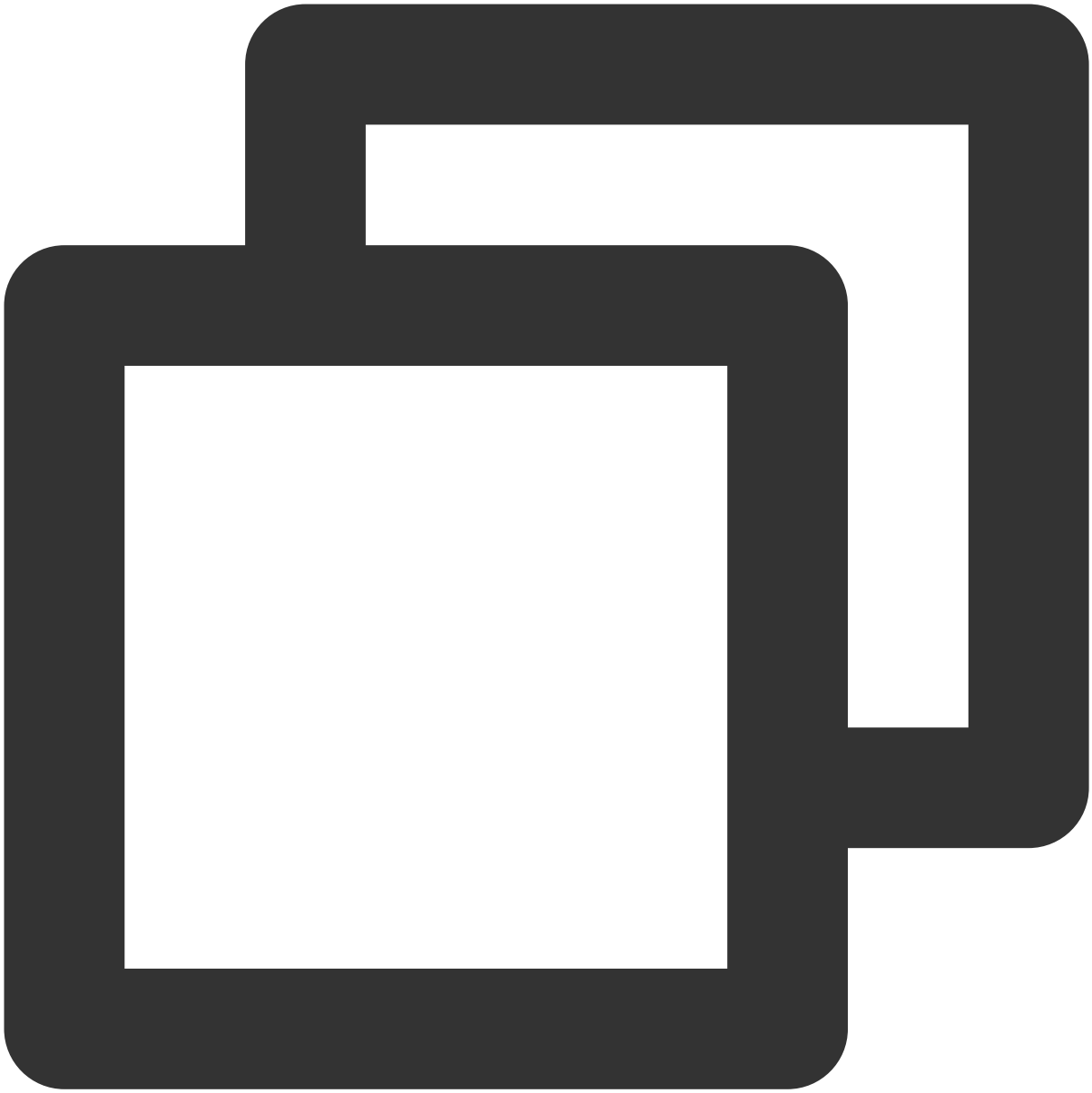


```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
Content-Length: 16
Connection: close
Accept-Ranges: bytes
```

```
Cache-Control: max-age=86400
Content-Disposition: attachment; filename=example.jpg
Date: Fri, 10 Apr 2020 09:35:17 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 09:35:05 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNkZDVfNjZjODJhMDlfMTY2MDdfMThm****



[Object Content]
```

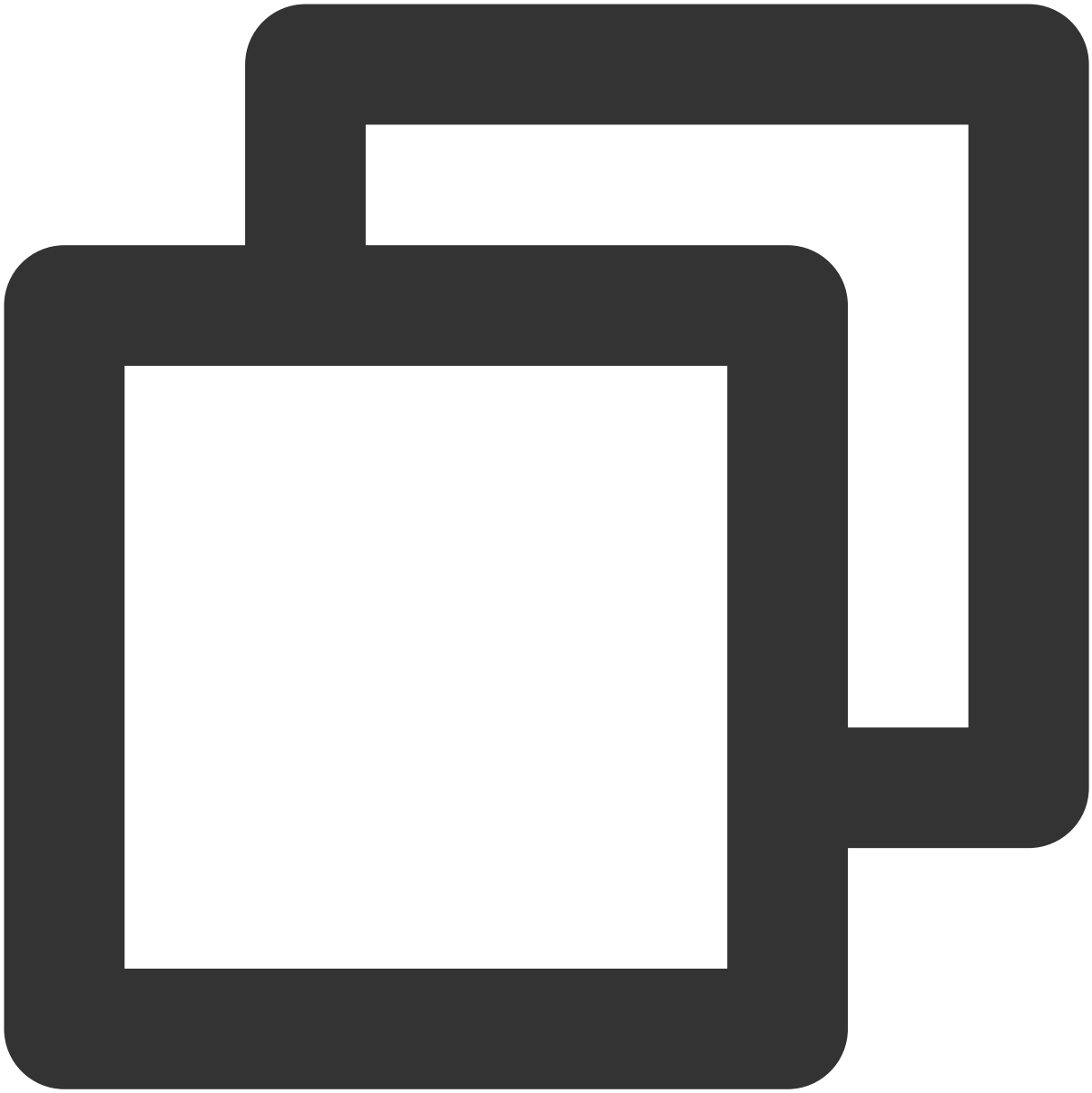
案例三：通过请求头指定查询条件并返回 HTTP 状态码为304（Not Modified）

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Jul 2020 06:51:49 GMT
If-None-Match: "ee8de918d05640145b18f70f4c3aa602"
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

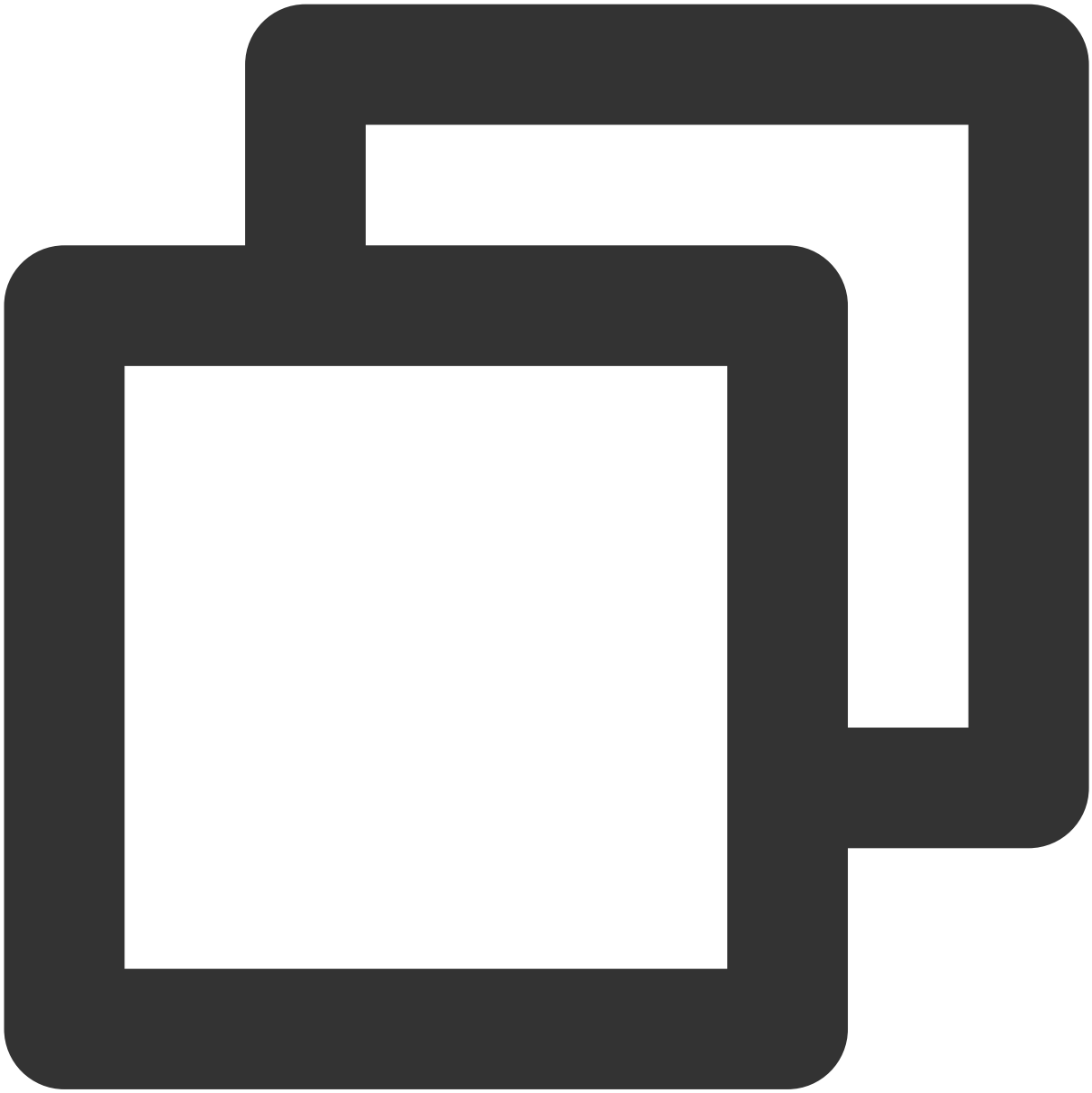


```
HTTP/1.1 304 Not Modified
Content-Type: image/jpeg
Content-Length: 0
Connection: close
Date: Wed, 29 Jul 2020 06:51:49 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWYyMTFjODVfOGZiNzJhMDlfNDcxZjZfZDY2****
```

版权所有：腾讯云计算（北京）有限责任公司

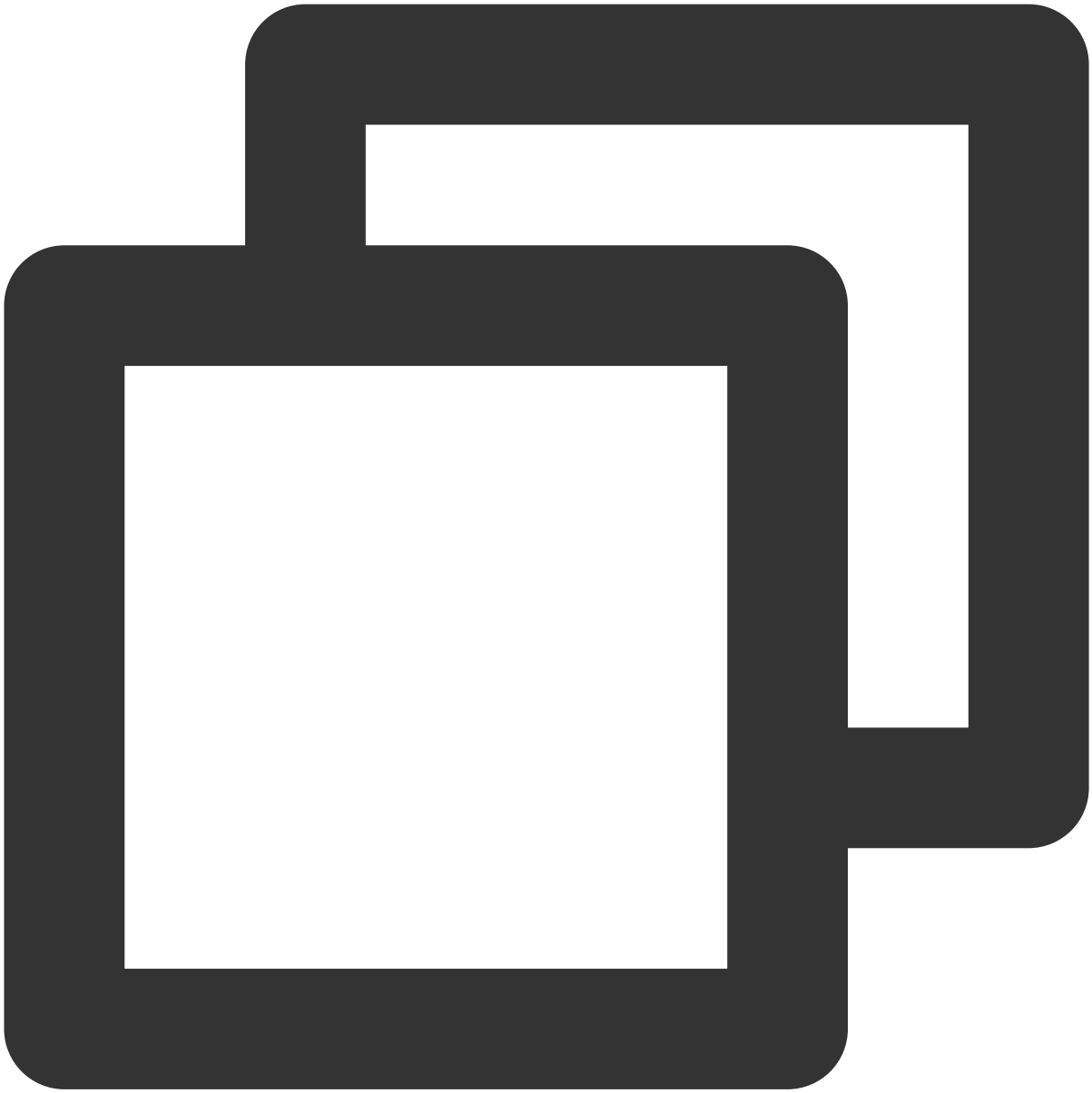
案例四：通过请求头指定查询条件并返回 HTTP 状态码为412（Precondition Failed）

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Jul 2020 06:51:50 GMT
If-Match: "aa488bb80185a6be87f4a7b936a80752"
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

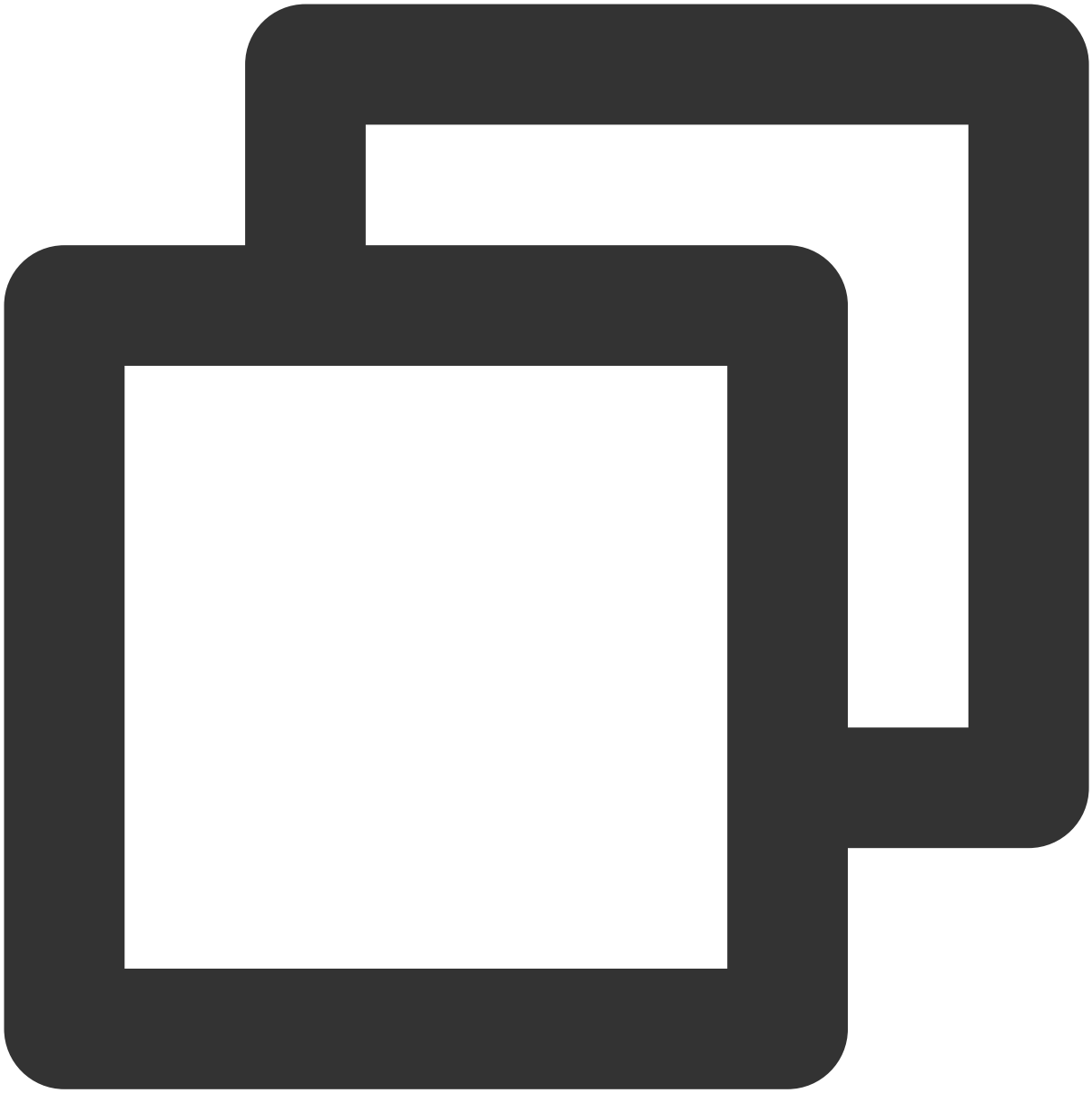


```
HTTP/1.1 412 Precondition Failed
Content-Type: application/xml
Content-Length: 480
Connection: close
Date: Wed, 29 Jul 2020 06:51:50 GMT
Server: tencent-cos
x-cos-request-id: NWYyMTFjODZfOGRjOTJhMDlfMmIyMWVfOTJl****
```

```
<?xml version='1.0' encoding='utf-8' ?>
<Error>
            <Code>PreconditionFailed</Code>
            <Message>Precondition not match.</Message>
            <Resource>examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleo
            <RequestId>NWYyMTFjODZfOGRjOTJhMDlfMmIyMWVfOTJl****</RequestId>
            <TraceId>OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODc0OWRkZjk0ZDM1NmI1M2
</Error>
```

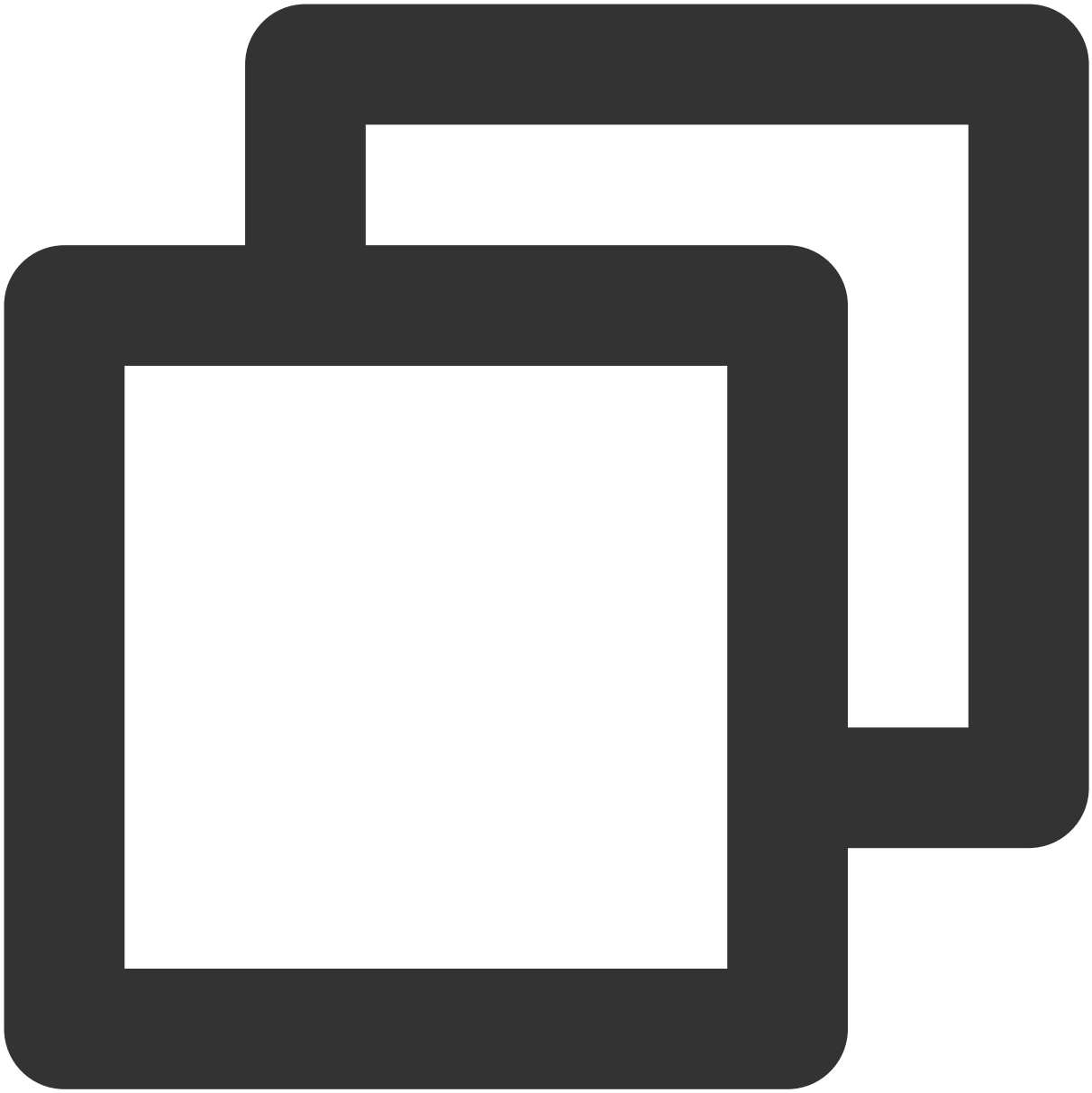
案例五：使用服务端加密 SSE-COS

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

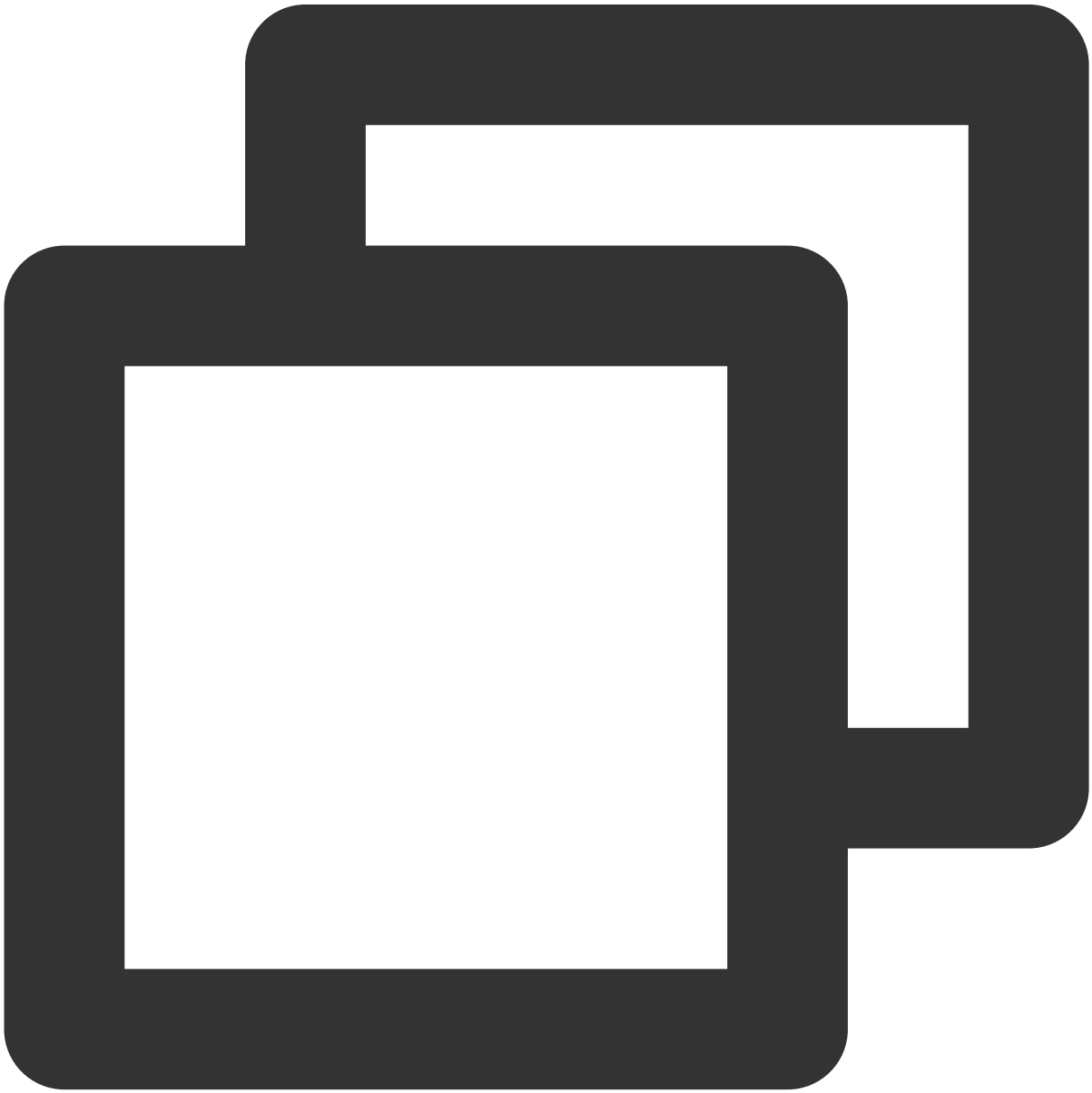


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Accept-Ranges: bytes
Date: Fri, 10 Apr 2020 09:36:00 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 09:35:49 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749656679157681890
x-cos-request-id: NWU5MDNlMDBfMzdiMDJhMDlfYTgyNl8xNjA2****
```

```
x-cos-server-side-encryption: AES256



[Object Content]
```

案例六：使用服务端加密 SSE-KMS

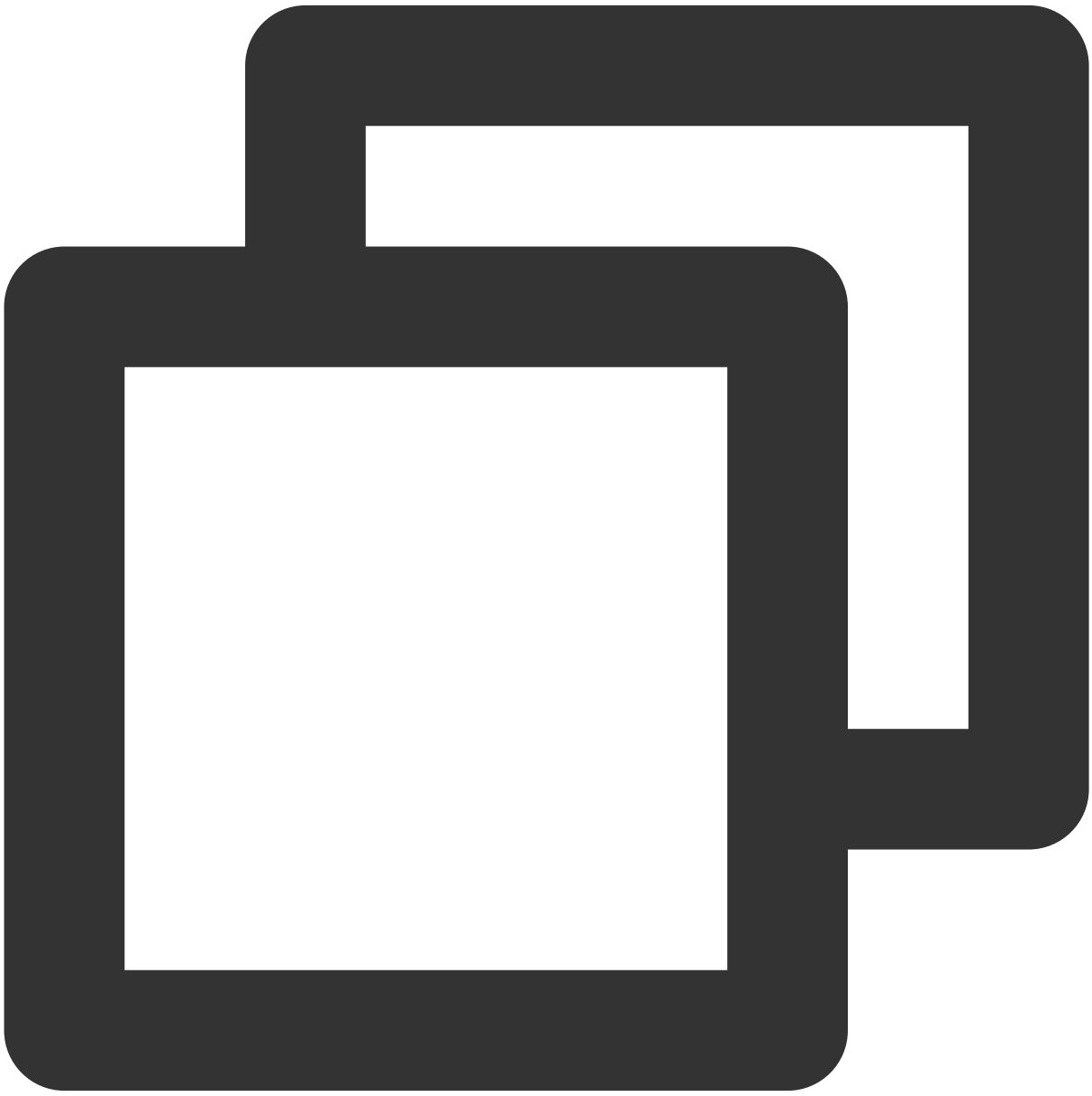
请求



```
GET /exampleobject HTTP/1.1
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:11 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

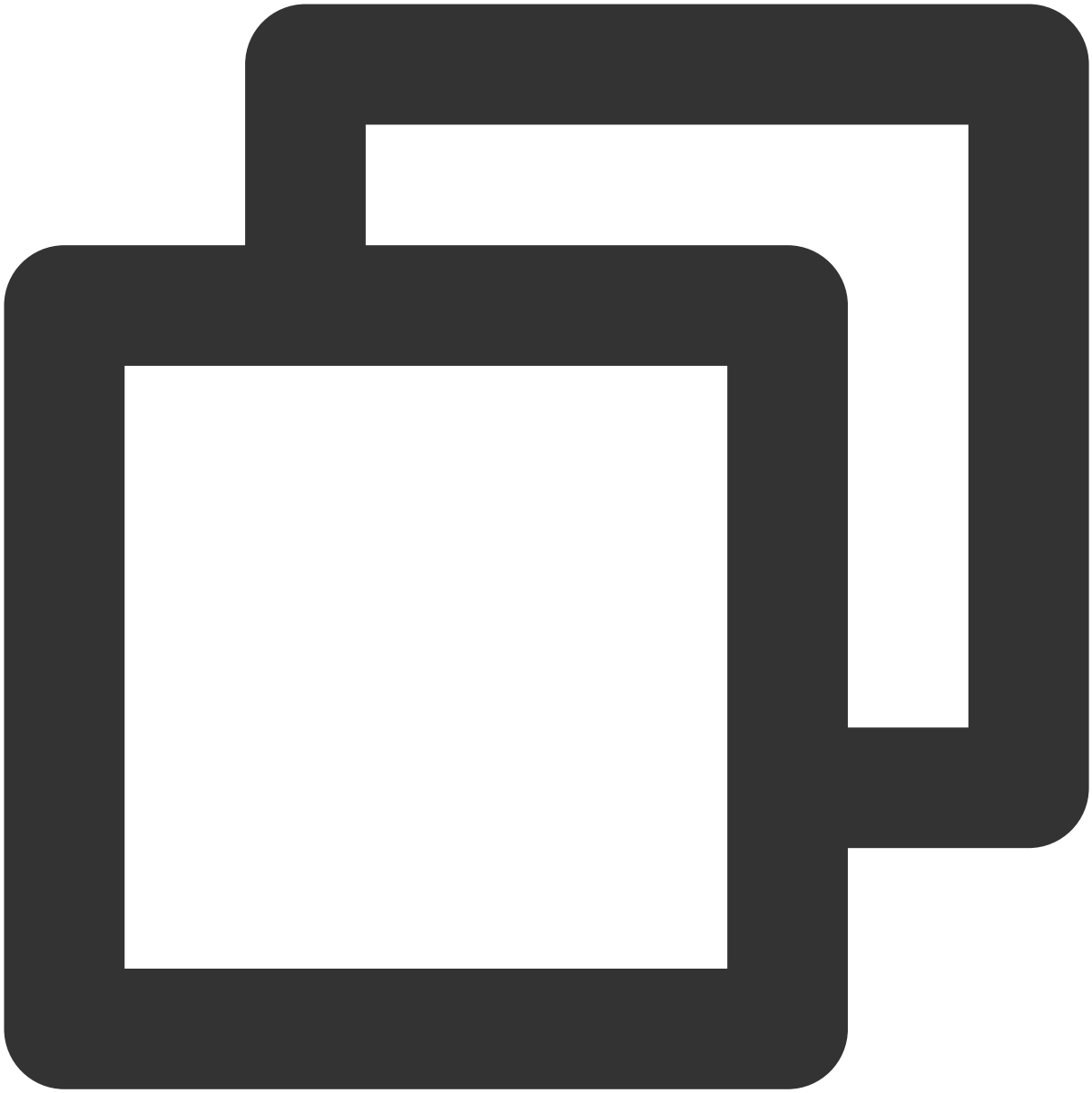


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
```

```
Accept-Ranges: bytes
Date: Fri, 10 Apr 2020 09:36:11 GMT
ETag: "840af7c921f4b3230049af8663145bd0"
Last-Modified: Fri, 10 Apr 2020 09:36:01 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749656679157681890
x-cos-request-id: NWU5MDNlMGJfZGEyNzVkNjRfZDgxY18xYTBj****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****



[Object Content]
```

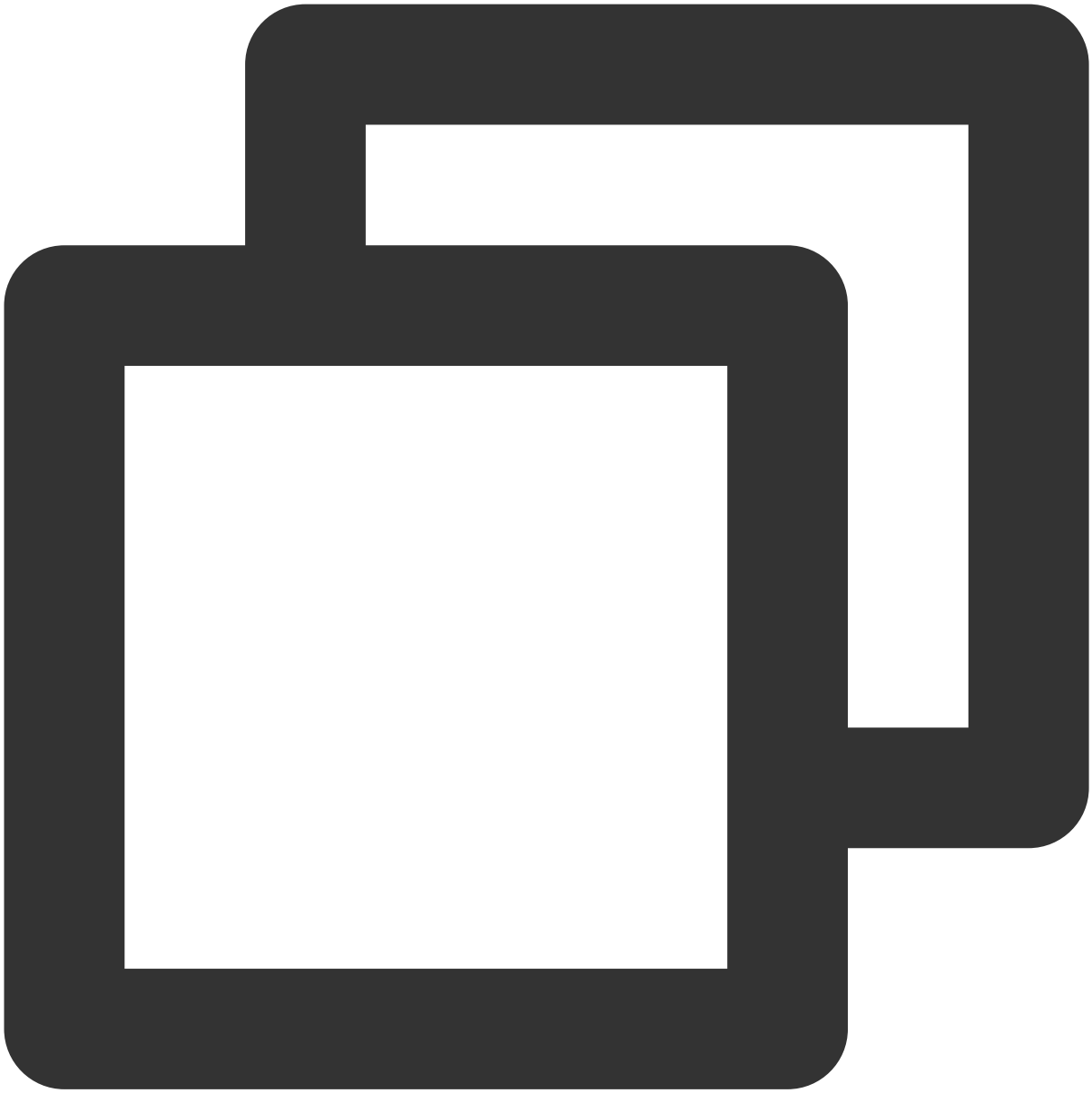
案例七：使用服务端加密 SSE-C

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:23 GMT
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNE
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

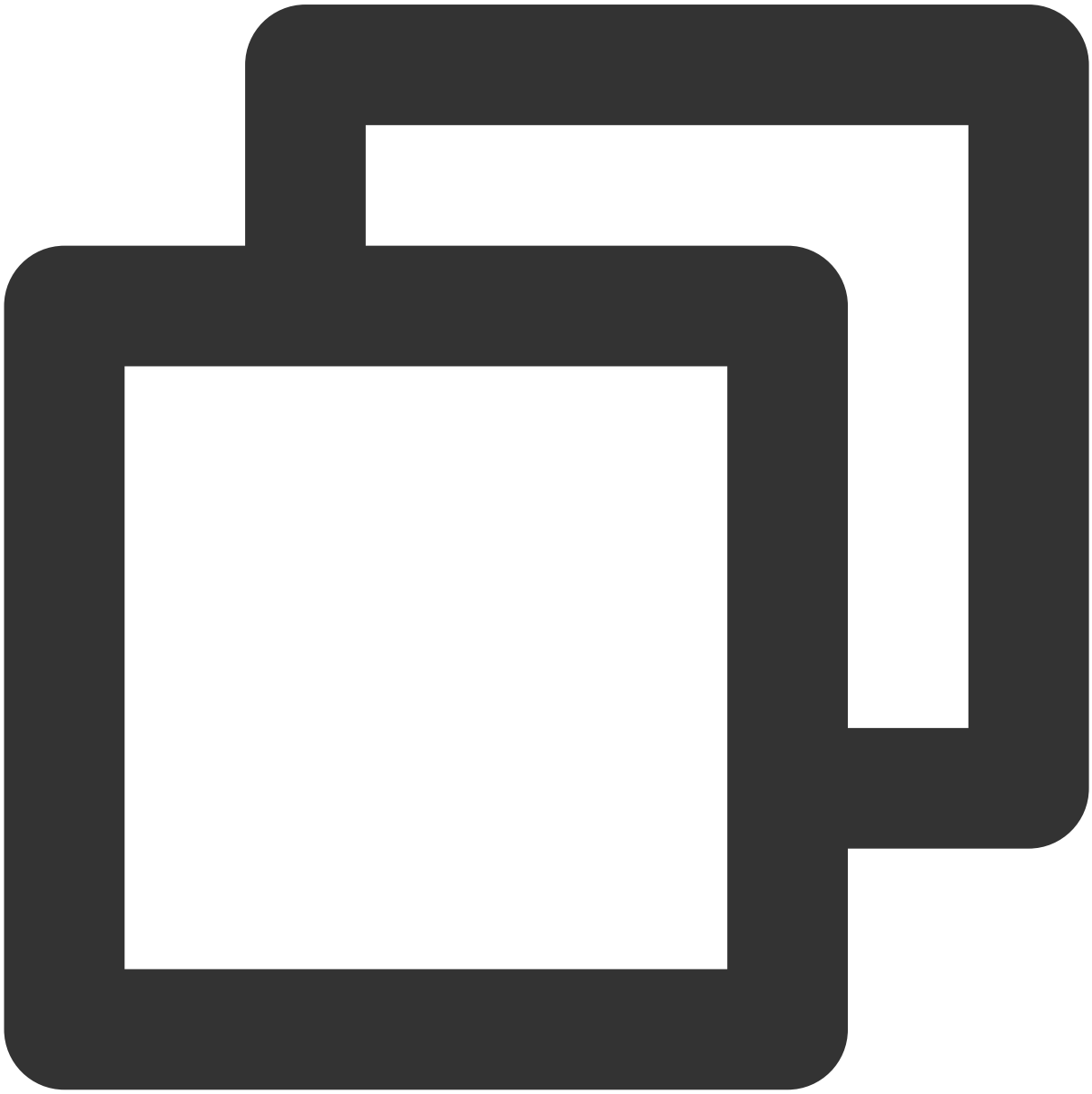


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Accept-Ranges: bytes
Date: Fri, 10 Apr 2020 09:36:23 GMT
ETag: "582d9105f71525f3c161984bc005efb5"
Last-Modified: Fri, 10 Apr 2020 09:36:12 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MDNlMTdfNzBiODJhMDlfZTVmMV8xNDAy****
```

```
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==



[Object Content]
```

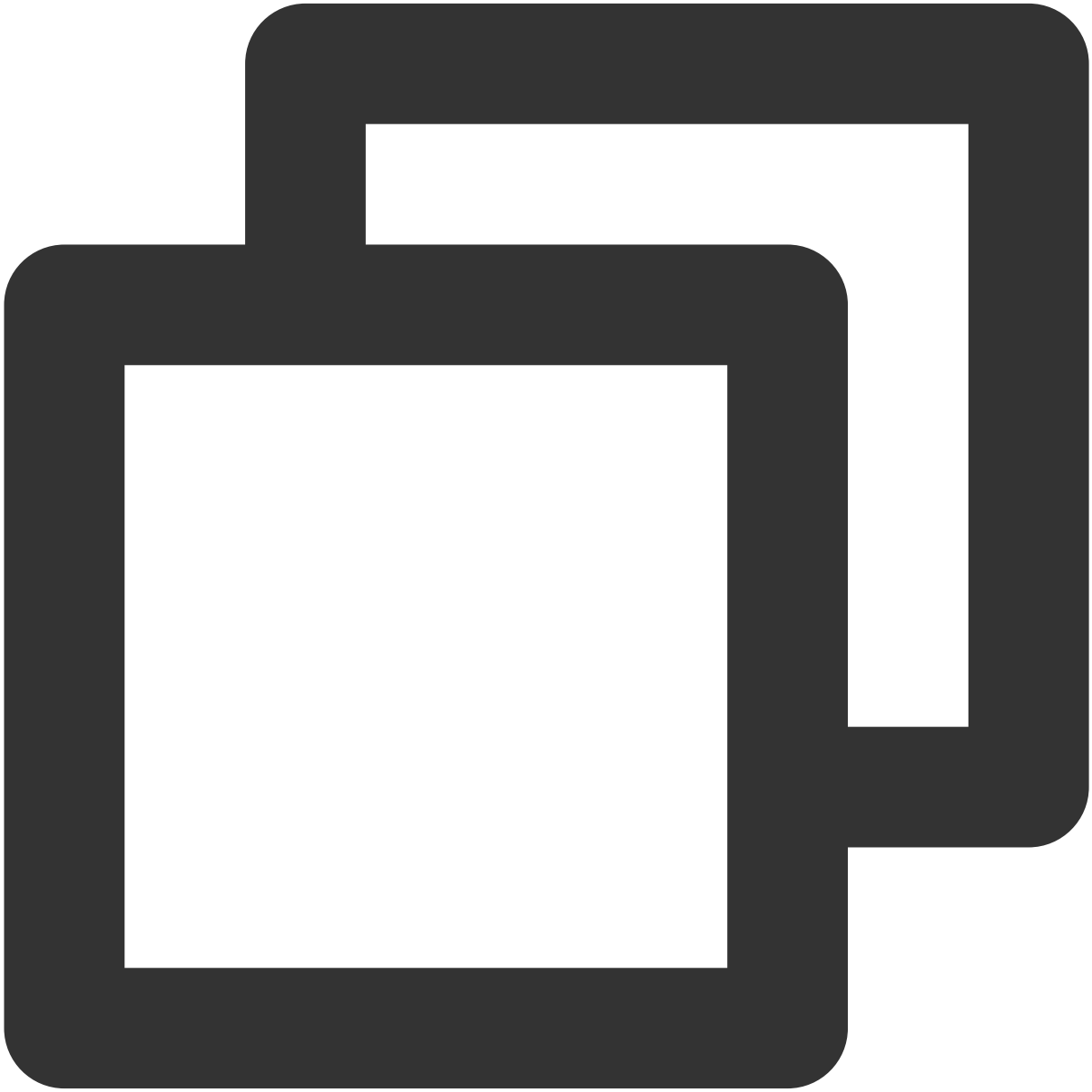
案例八：下载对象最新版本（启用版本控制）

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 12:30:02 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

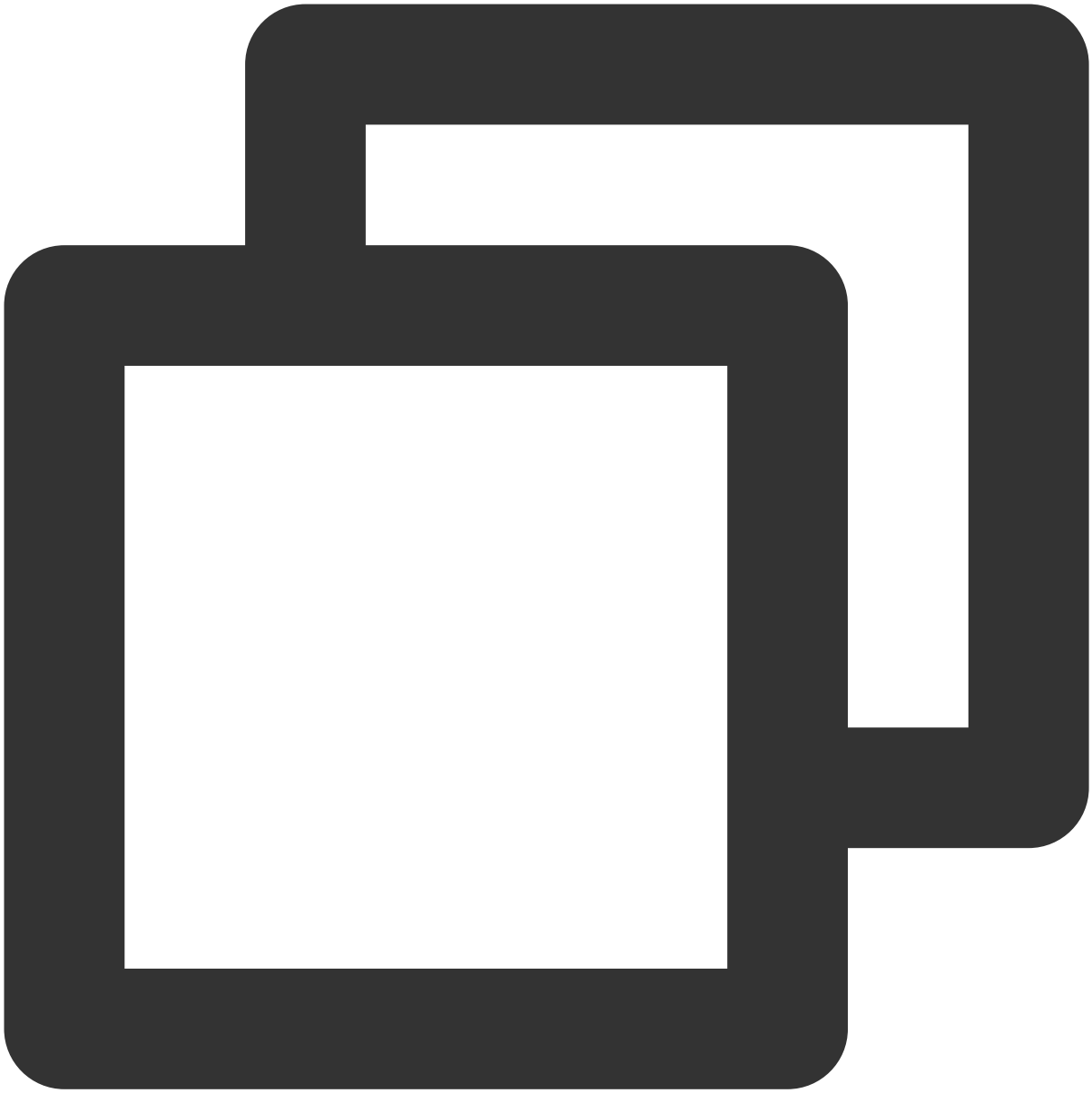


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
```

```
Content-Length: 26
Connection: close
Accept-Ranges: bytes
Date: Fri, 10 Apr 2020 12:30:02 GMT
ETag: "22e024392de860289f0baa7d6cf8a549"
Last-Modified: Fri, 10 Apr 2020 12:29:52 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 11596229263574363878
x-cos-request-id: NWU5MDY2Y2FfMzFiYjBiMDlfMjE2NzVfMTgz****
x-cos-version-id: MTg0NDUxNTc1NTE5MTc1NjM4MDA


[Object Content Version 2]
```

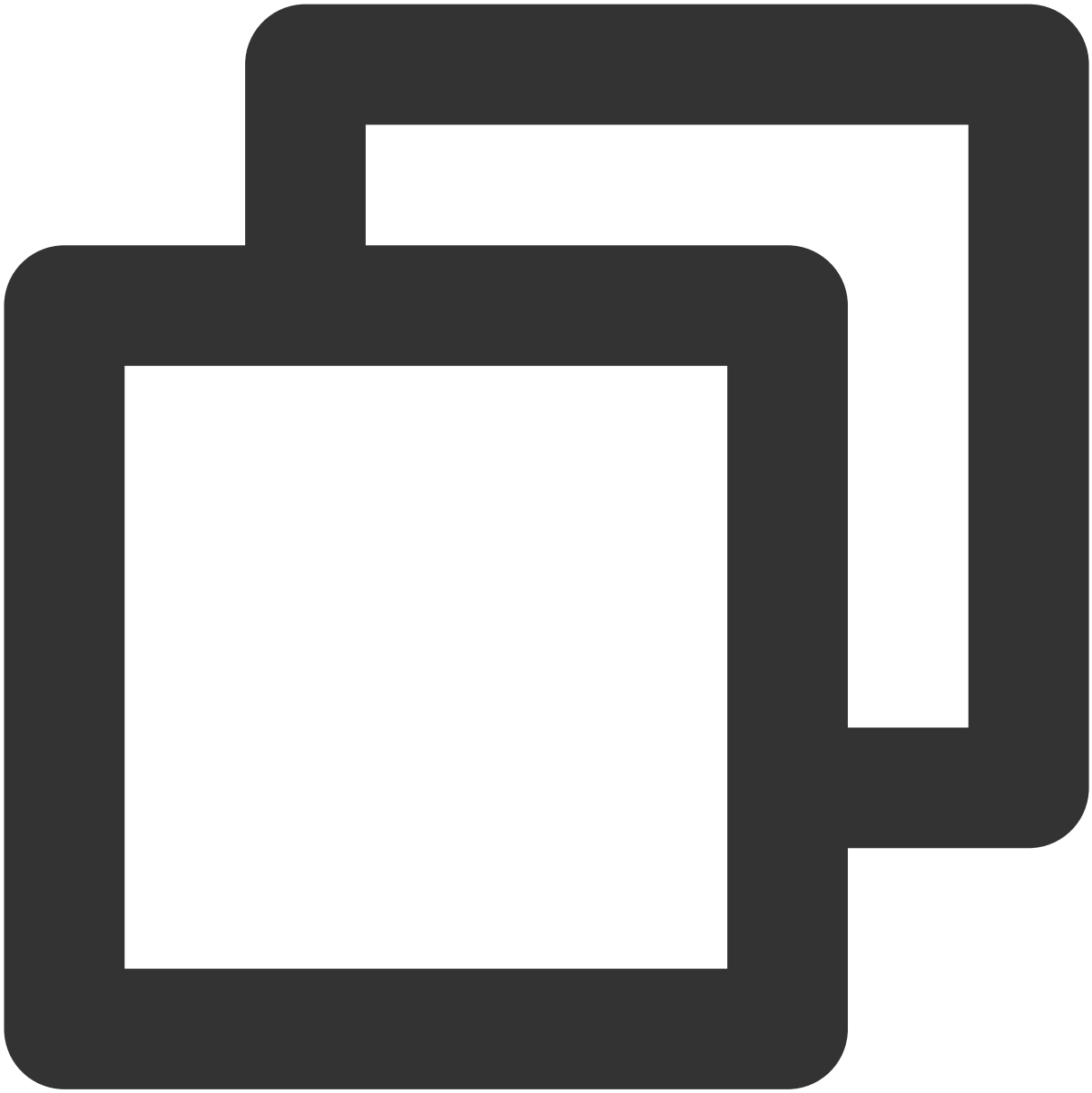
案例九：下载对象指定版本（启用版本控制）

请求



```
GET /exampleobject?versionId=MTg0NDUxNTc1NjIzMTQ1MDAwODg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:36:45 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

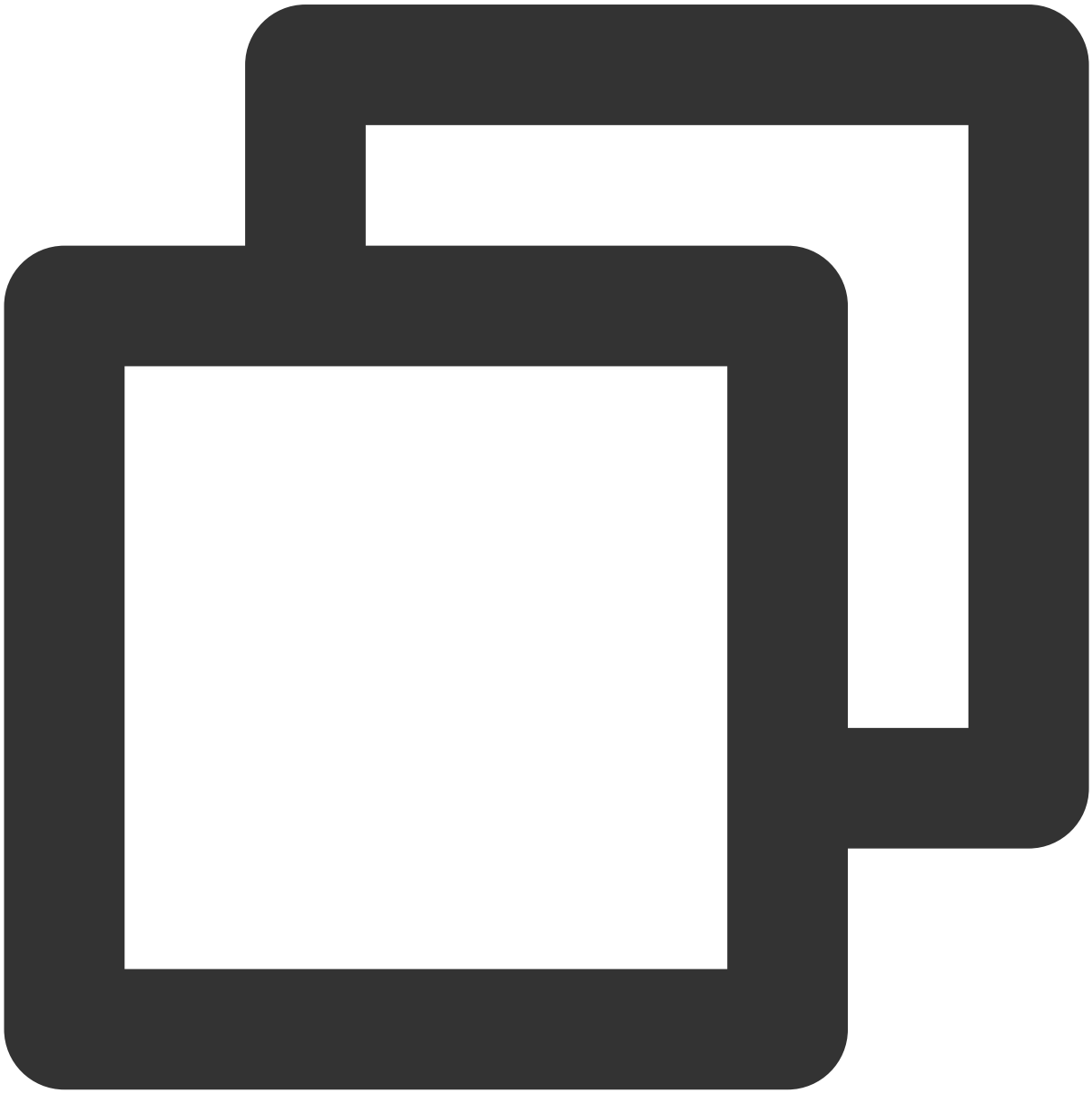


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Accept-Ranges: bytes
Date: Fri, 10 Apr 2020 09:36:45 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 09:36:35 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749656679157681890
x-cos-request-id: NWU5MDNlMmRfNzBiODJhMDlfZTYwZl8xM2Fh****
```

```
x-cos-version-id: MTg0NDUxNTc1NjIzMTQ1MDAwODg


[Object Content]
```

案例十：指定 Range 请求头部下载部分内容

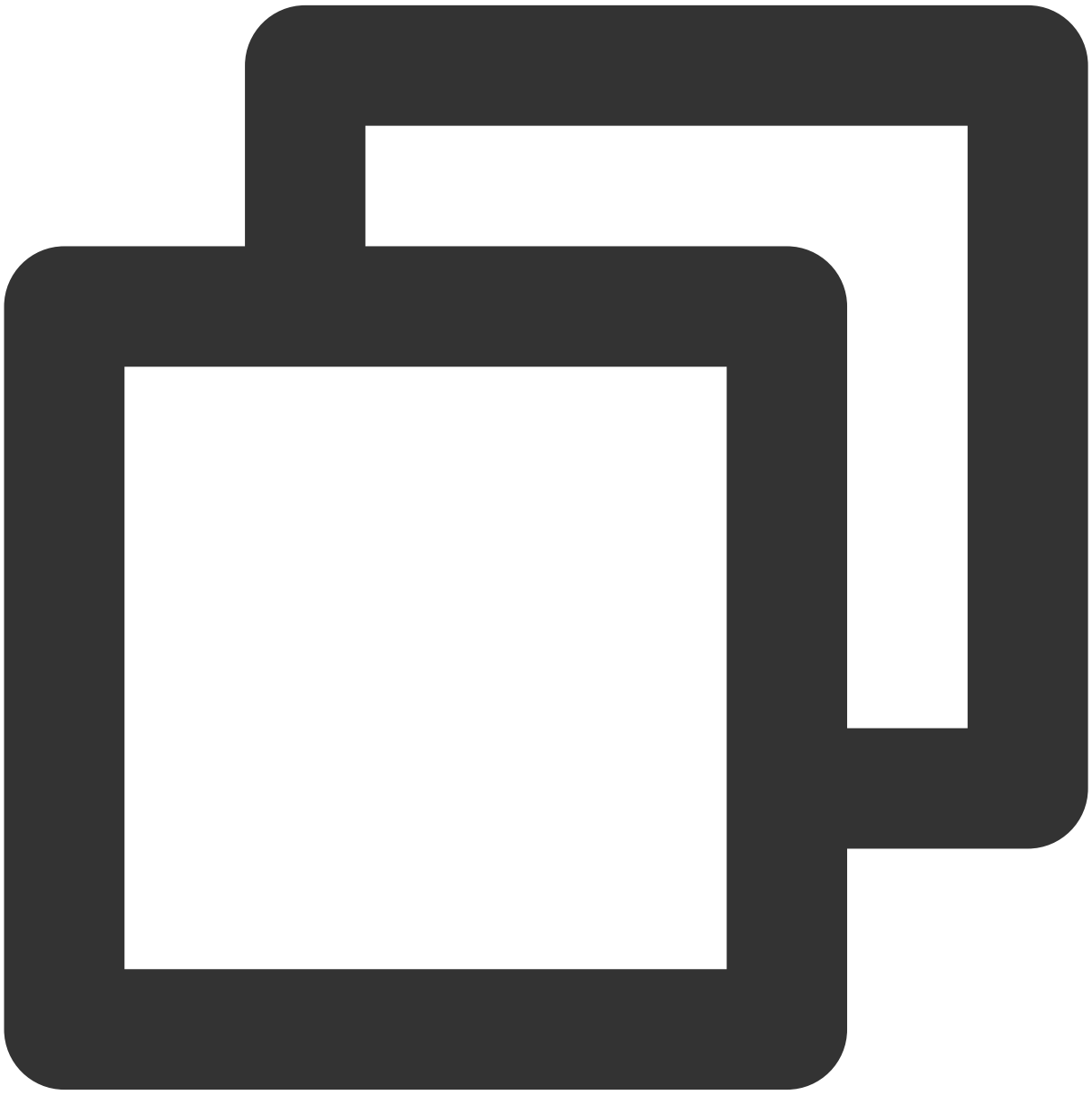
请求



```
GET /exampleobject HTTP/1.1
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 12:32:37 GMT
Range: bytes=8-14
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

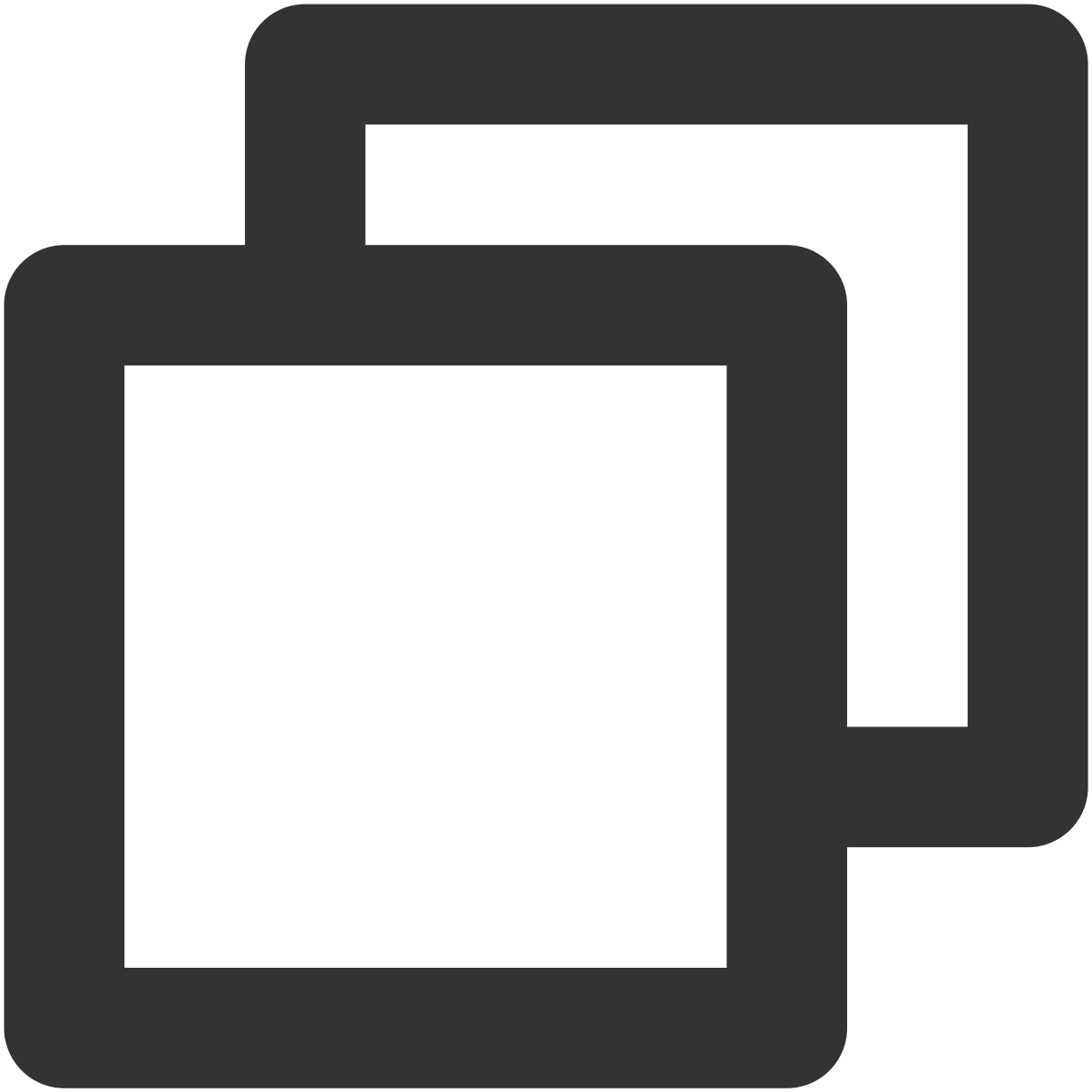


```
HTTP/1.1 206 Partial Content
Content-Type: image/jpeg
Content-Length: 7
```

```
Connection: close
Accept-Ranges: bytes
Content-Range: bytes 8-14/16
Date: Fri, 10 Apr 2020 12:32:37 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 12:32:25 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749656679157681890
x-cos-request-id: NWU5MDY3NjVfY2VjODJhMDlfOWVlZl8xNmMy****



Content
```

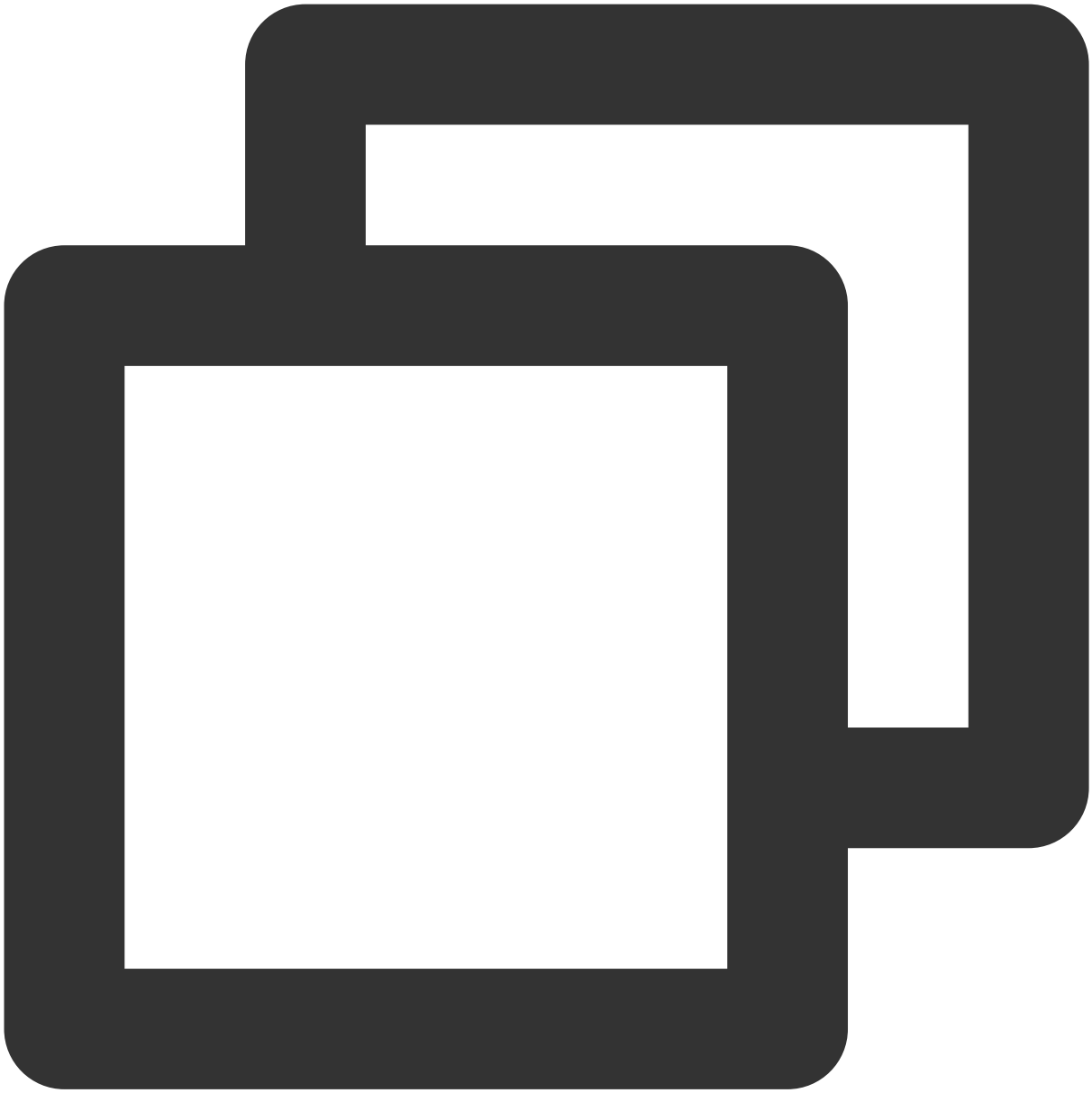
案例十一：下载未经恢复的归档（ARCHIVE）存储类型的对象

请求



```
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 26 Dec 2019 11:57:24 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: 513
Connection: close
Date: Thu, 26 Dec 2019 11:57:24 GMT
Server: tencent-cos
x-cos-request-id: NWUwNGEwMjRfZDcyNzVkNjRfNjZlM183Zjcx****
x-cos-storage-class: ARCHIVE
```

```
<?xml version='1.0' encoding='utf-8' ?>
<Error>
            <Code>InvalidObjectState</Code>
            <Message>The operation is not valid for the object storage class.</Mess
            <Resource>examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleo
            <RequestId>NWUwNGEwMjRfZDcyNzVkNjRfNjZlM183Zjcx****</RequestId>
            <TraceId>OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODc0OWRkZjk0ZDM1NmI1M2
</Error>
```

HEAD Object

最近更新时间：2023-12-21 14:32:32

功能描述

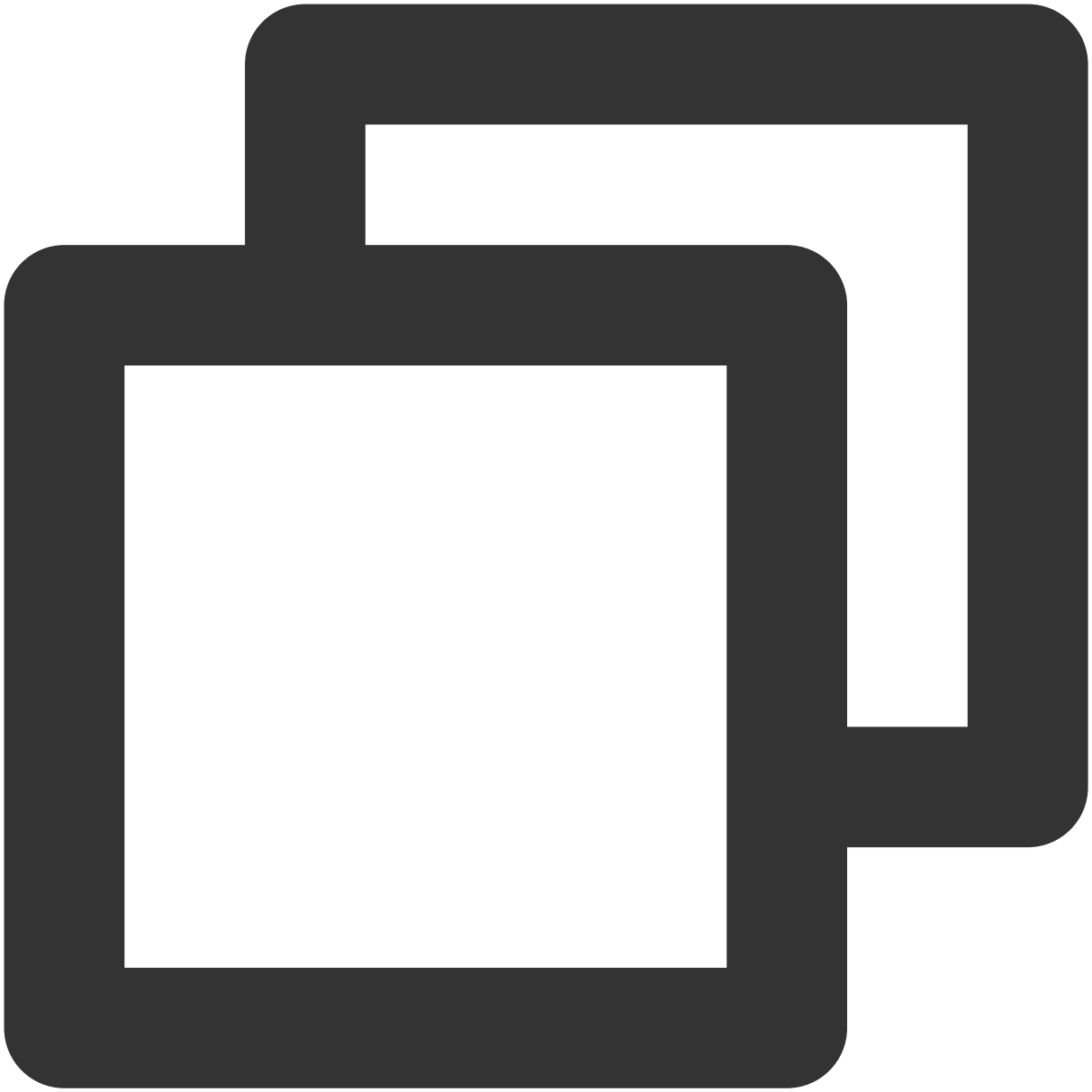
HEAD Object 接口请求可以判断指定对象是否存在和有权限，并在指定对象可访问时获取其元数据。该 API 的请求者需要对目标对象有读取权限，或者目标对象向所有人开放了读取权限（公有读）。

版本控制

当启用版本控制时，该 HEAD 操作可以使用 versionId 请求参数指定要返回的版本 ID，此时将返回指定版本的元数据，如指定版本为删除标记则返回 HTTP 响应码404（Not Found），否则将返回最新版本的元数据。

请求

请求示例



```
HEAD /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

名称	描述	类型	是否必选
versionId	当启用版本控制时，指定要查询的版本 ID，如不指定则查询对象的最新版本	string	否

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
If-Modified-Since	当对象在指定时间后被修改，则返回 HTTP 状态码为200（OK），否则返回 HTTP 状态码为304（Not Modified）	string	否
If-Unmodified-Since	当对象在指定时间后未被修改，则返回 HTTP 状态码为200（OK），否则返回 HTTP 状态码为412（Precondition Failed）	string	否
If-Match	当对象的 ETag 与指定的值一致，则返回 HTTP 状态码为200（OK），否则返回 HTTP 状态码为412（Precondition Failed）	string	否
If-None-Match	当对象的 ETag 与指定的值不一致，则返回 HTTP 状态码为200（OK），否则返回 HTTP 状态码为304（Not Modified）	string	否

服务端加密相关头部

如果指定的对象使用了服务端加密且加密方式为 SSE-C 时，则需要指定服务端加密的相关头部来解密对象，请参见 服务端加密专用头部。

请求体

此接口无请求体。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

名称	描述	类型
Cache-Control	RFC 2616 中定义的缓存指令，仅当对象元数据包含此项时才会返回该头部	string
Content-	RFC 2616 中定义的文件名称，仅当对象元数据包含此项时才会返回该头部	string

版权所有：腾讯云计算（北京）有限责任公司

Disposition		
Content-Encoding	RFC 2616 中定义的编码格式，仅当对象元数据包含此项时才会返回该头部	string
Expires	RFC 2616 中定义的缓存失效时间，仅当对象元数据包含此项时才会返回该头部	string
x-cos-meta-*	包括用户自定义元数据头部后缀和用户自定义元数据信息	string
x-cos-storage-class	对象存储类型，枚举值请参见 存储类型 文档，例如 MAZ_STANDARD，MAZ_STANDARD_IA，INTELLIGENT_TIERING，MAZ_INTELLIGENT_TIERING，STANDARD_IA，ARCHIVE，DEEP_ARCHIVE。**仅当对象不是标准存储（STANDARD）时才会返回该头部**	enum
x-cos-storage-tier	当对象的存储类型为智能分层存储时，该头部表示对象所处的存储层，有效值：FREQUENT、INFREQUENT。	enum

归档类型对象相关头部

当对象为归档类型且使用 POST Object restore 请求进行恢复操作时，该 HEAD 请求将返回下列响应头部：

名称	描述	类型
x-cos-restore	指示当前恢复操作的状态： 当恢复操作正在进行中时，该响应头部的值为 `ongoing-request="true"` 当对象已经恢复时，该响应头部包含 COS 即将删除临时副本的时间，例如 `ongoing-request="false", expiry-date="Tue, 19 Nov 2019 16:00:00 GMT"`	string
x-cos-restore-status	当恢复操作正在进行中时返回该响应头部，指示当前的恢复模式和恢复操作请求时间，例如 `tier="bulk"; request-date="Mon, 18 Nov 2019 09:34:50 GMT"`，有关恢复模式请参见 POST Object restore	string

版本控制相关头部

启用版本控制的存储桶内的对象将返回下列响应头部：

名称	描述	类型
x-cos-version-id	对象的版本 ID	string

服务端加密相关头部

如果指定的对象使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

此接口响应体为空。

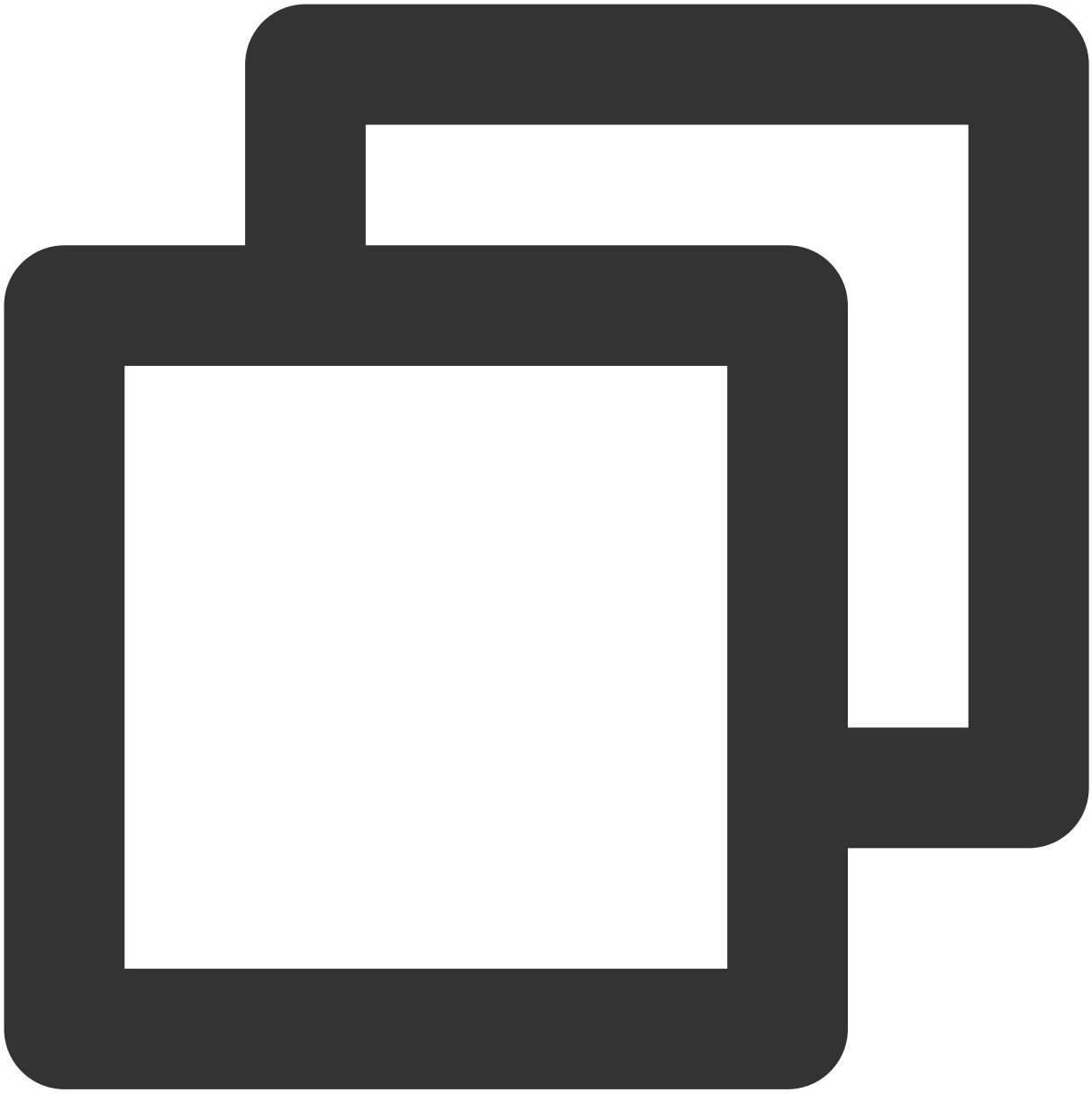
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

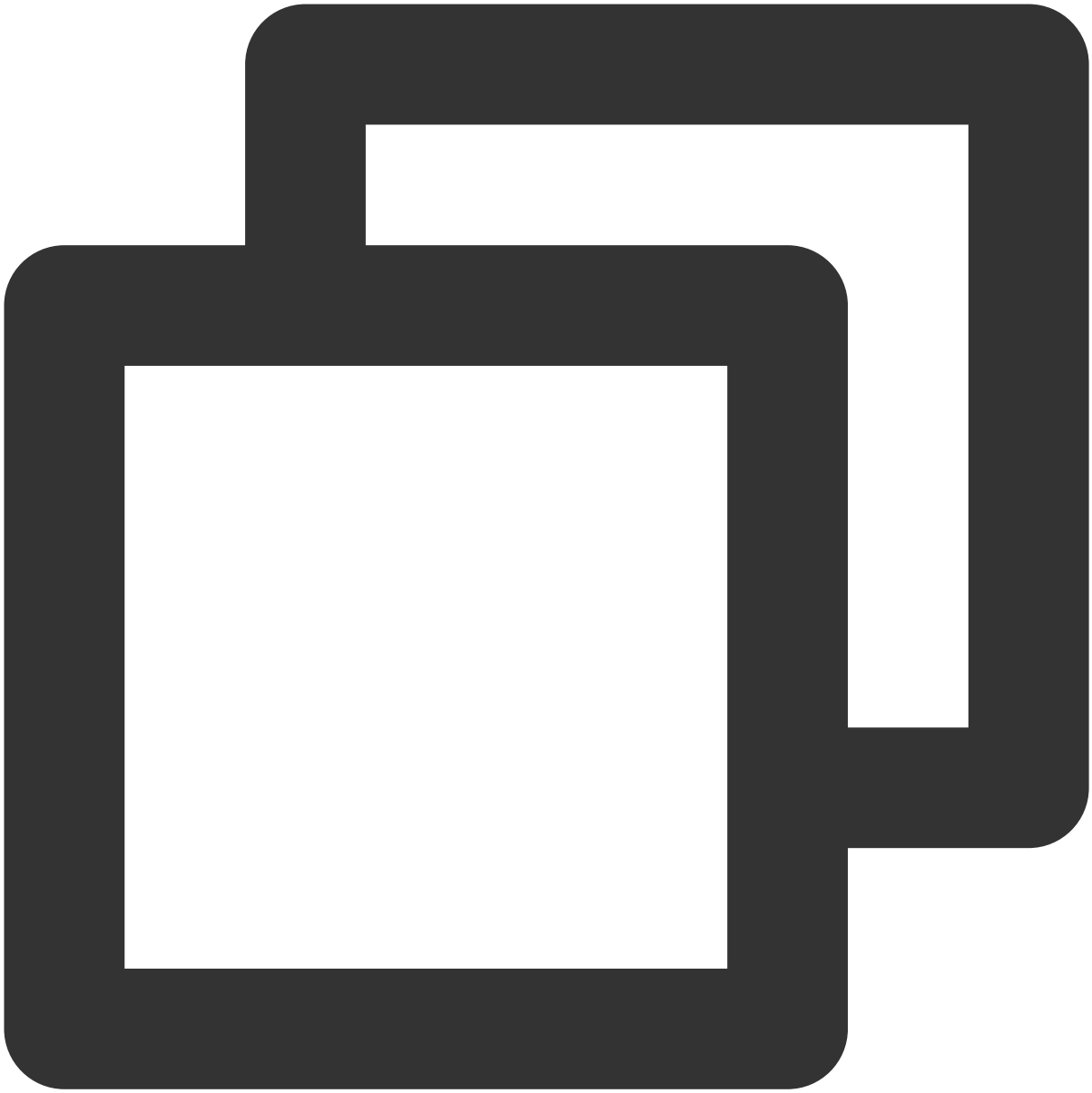
案例一：简单案例（未启用版本控制）

请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:17:36 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

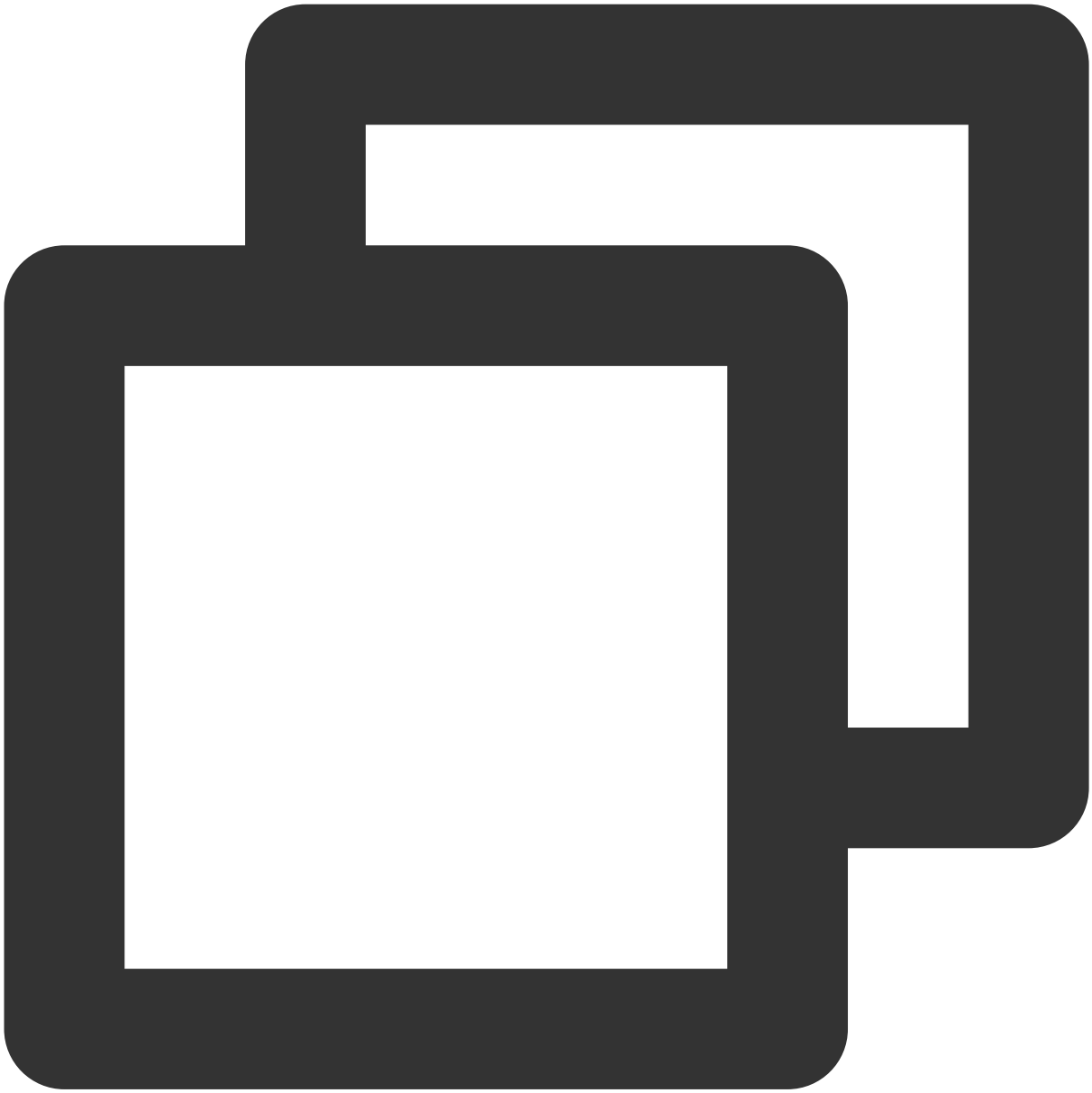
响应



```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Date: Fri, 10 Apr 2020 18:17:36 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 18:17:25 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MGI4NDBfNjFjODJhMDlfMzY2NjVfMjNi****
```

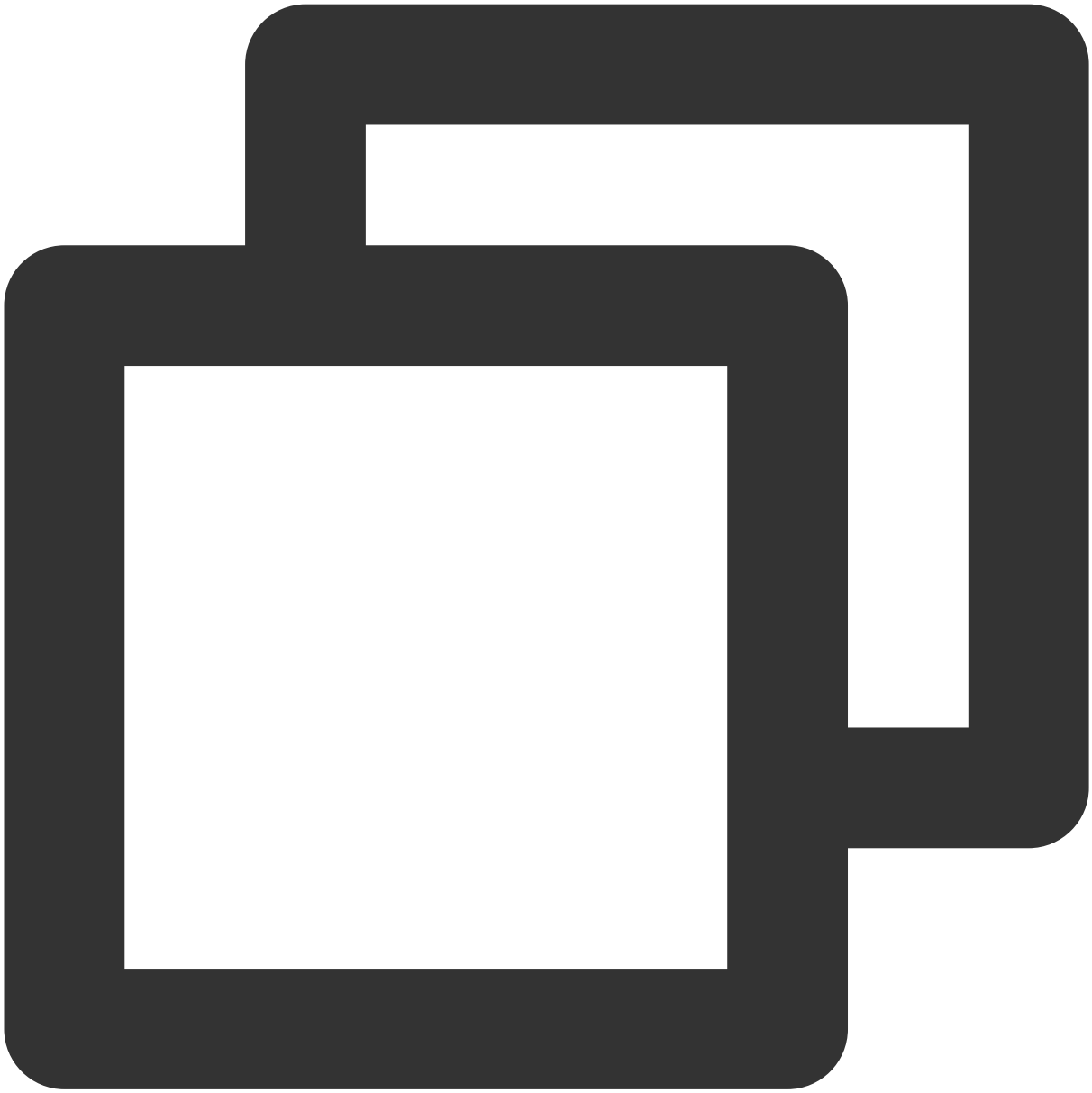
案例二：通过请求头指定查询条件并返回 HTTP 状态码为304（Not Modified）

请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Jul 2020 06:51:49 GMT
If-None-Match: "ee8de918d05640145b18f70f4c3aa602"
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

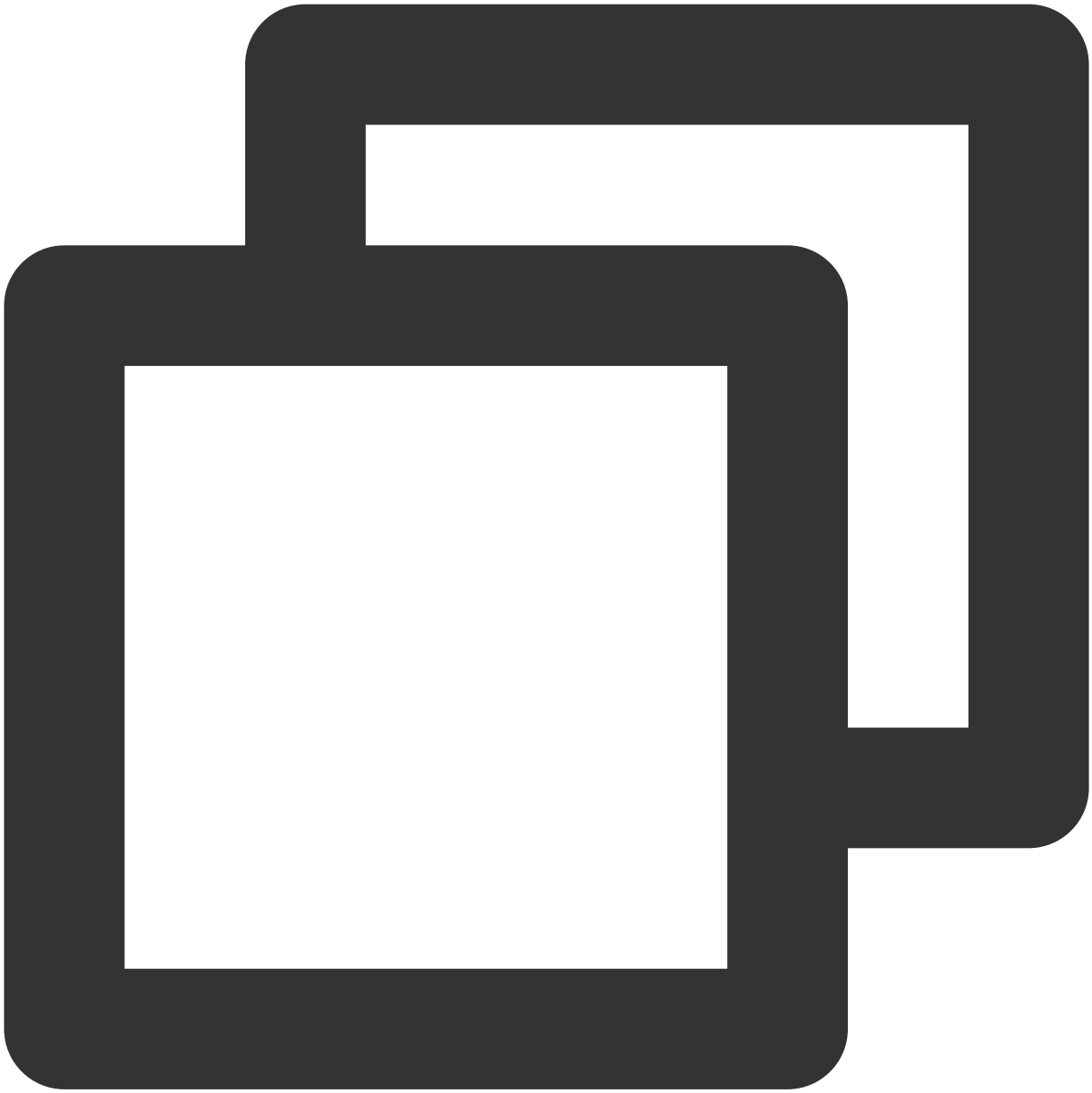
响应



```
HTTP/1.1 304 Not Modified
Content-Type: application/xml
Content-Length: 0
Connection: close
Date: Wed, 29 Jul 2020 06:51:49 GMT
Server: tencent-cos
x-cos-request-id: NWYyMTFjODVfZDNjODJhMDlfMWU1MWVfOTUy****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODczNTBmNjMwZmQ0MTZkMjg0Njl
```

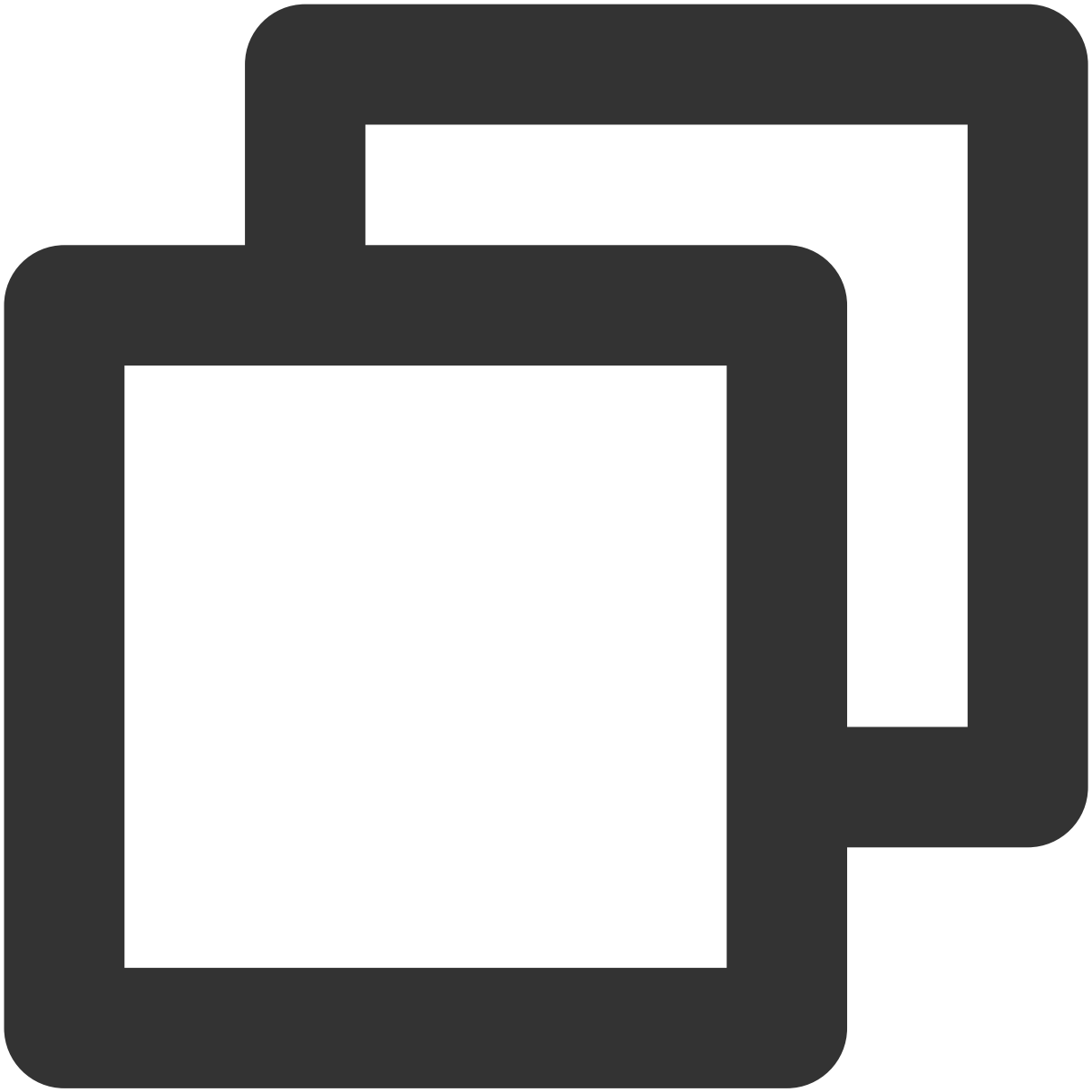
案例三：通过请求头指定查询条件并返回 HTTP 状态码为412（Precondition Failed）

请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 29 Jul 2020 06:51:50 GMT
If-Match: "aa488bb80185a6be87f4a7b936a80752"
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

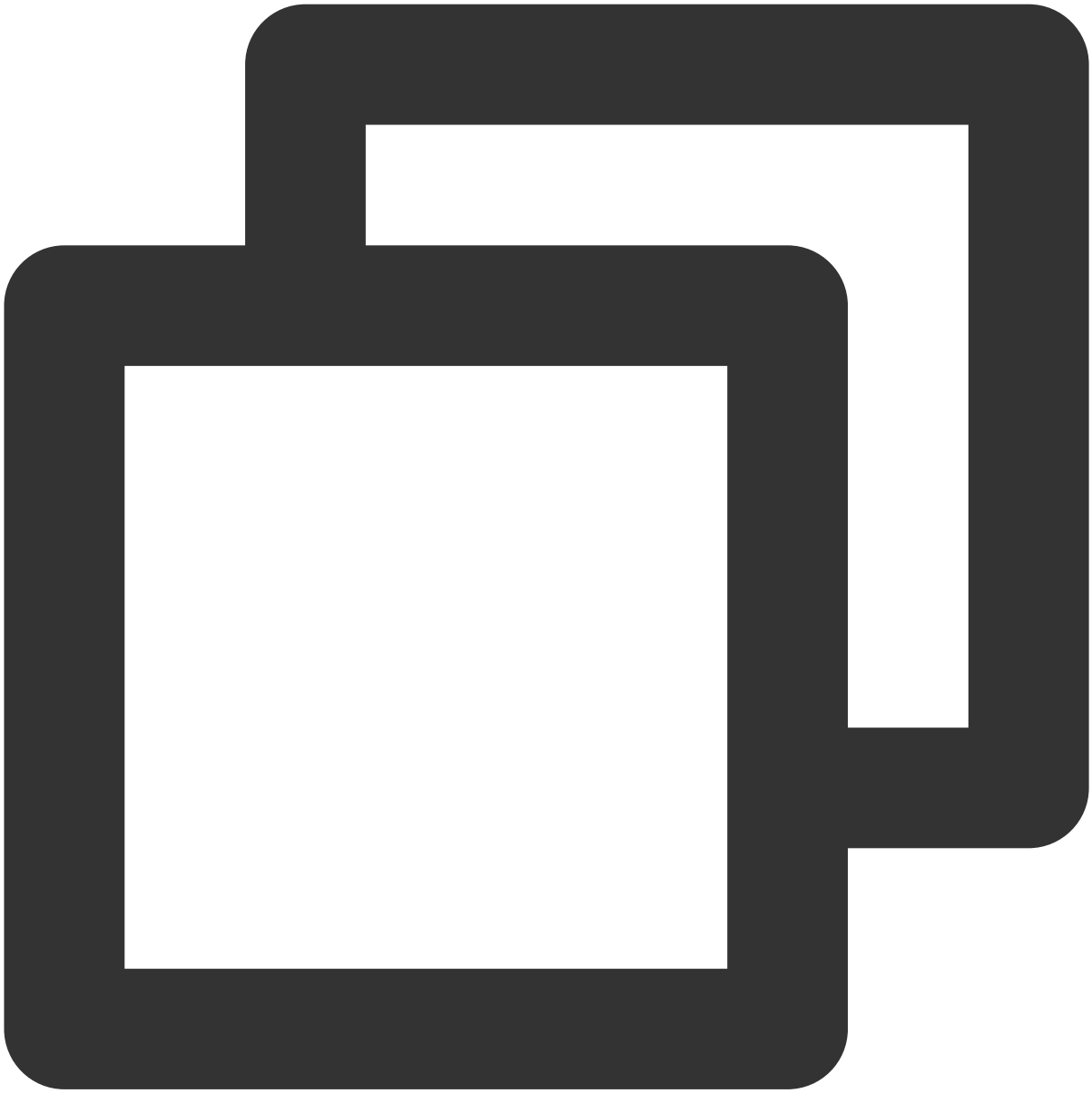
响应



```
HTTP/1.1 412 Precondition Failed
Content-Type: application/xml
Content-Length: 0
Connection: close
Date: Wed, 29 Jul 2020 06:51:50 GMT
Server: tencent-cos
x-cos-request-id: NWYyMTFjODZfMzBjMDJhMDlfMmU3ZF9kYTE4****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODczNTBmNjMwZmQ0MTZkMjg0Njl
```

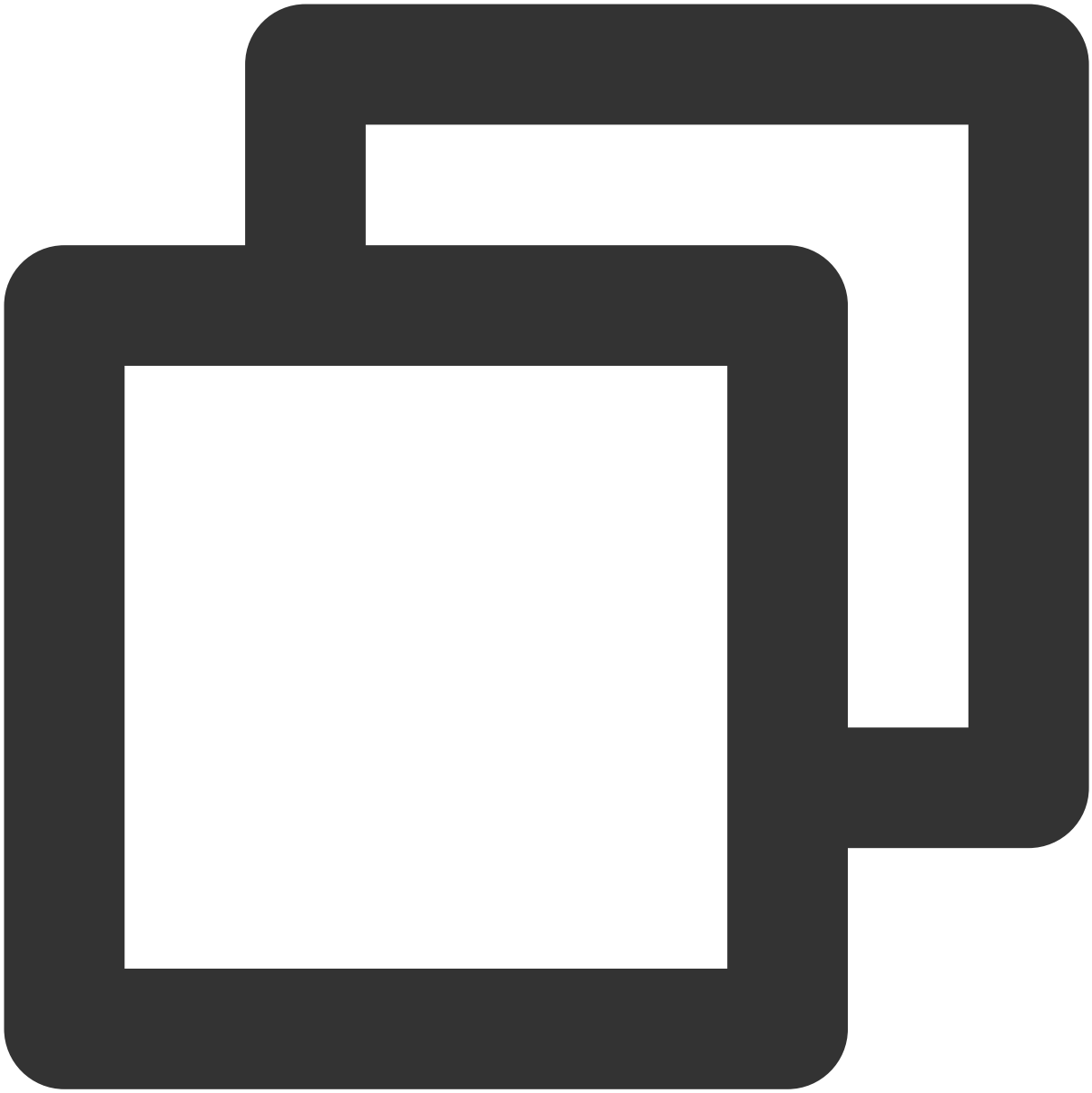
案例四：使用服务端加密 SSE-COS

请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:18:19 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

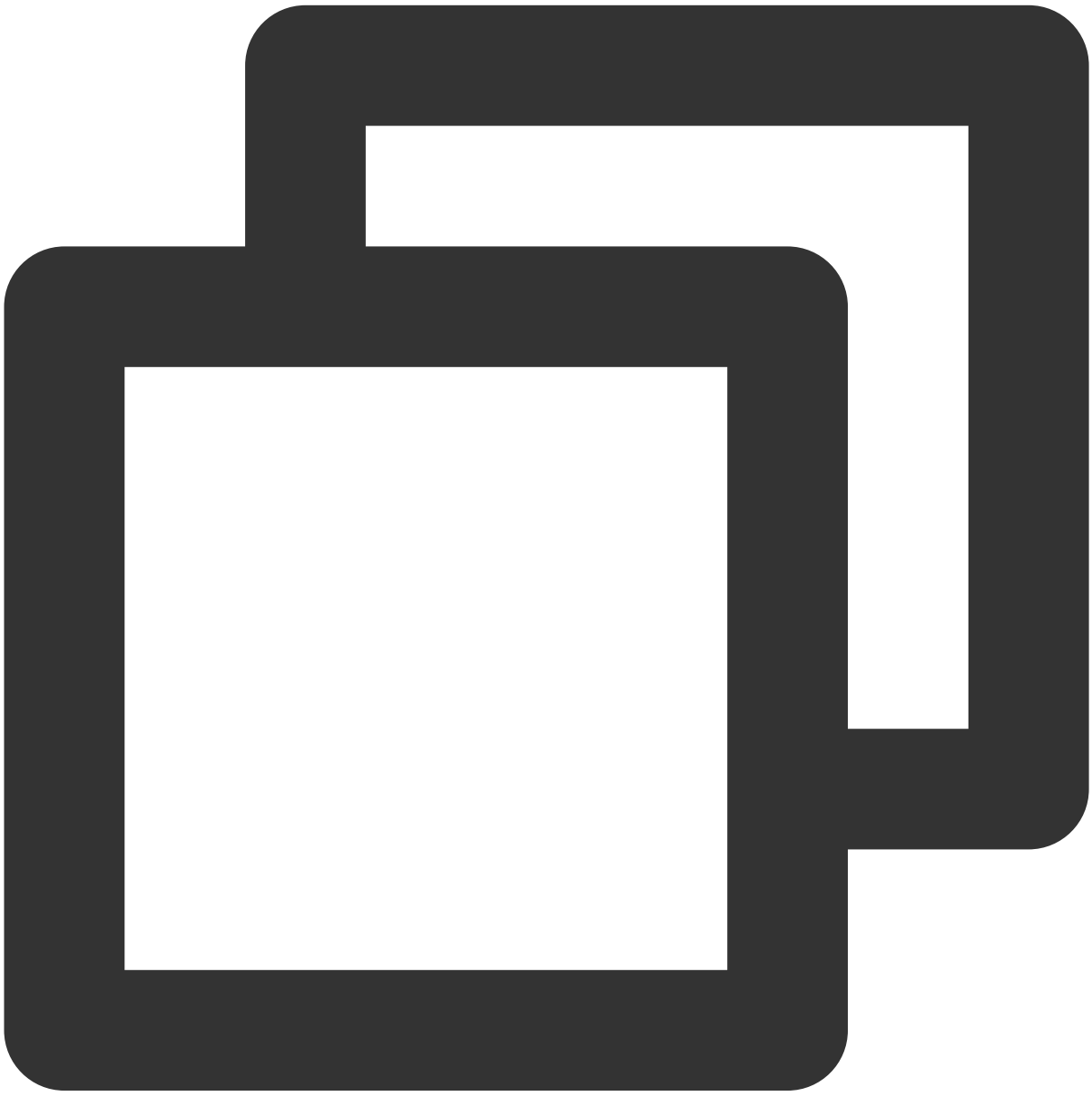


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Date: Fri, 10 Apr 2020 18:18:19 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 18:18:08 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MGI4NmJfNmRjMDJhMDlfZGQwNl8xZmE4****
x-cos-server-side-encryption: AES256
```

版权所有：腾讯云计算（北京）有限责任公司

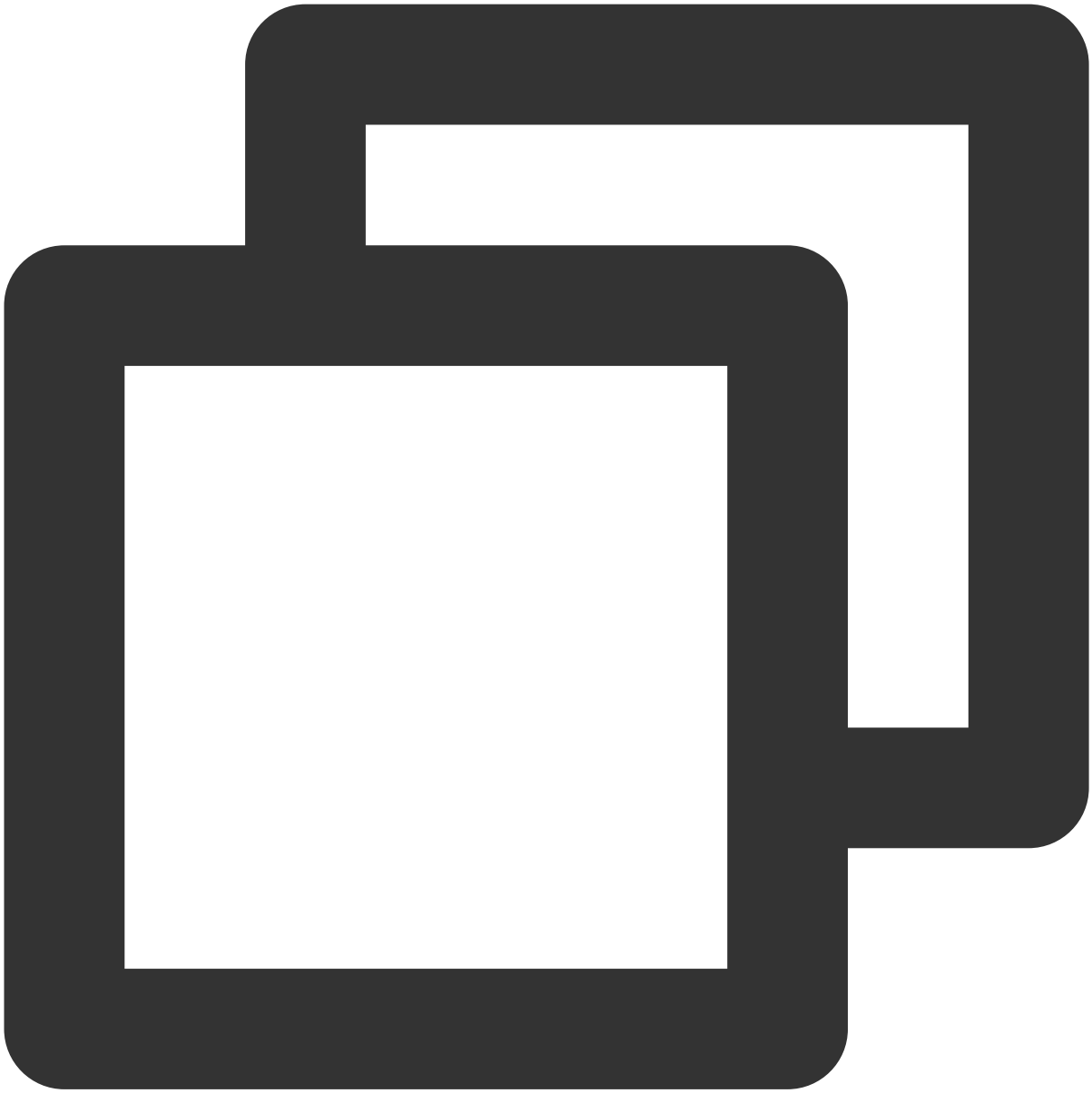
案例五：使用服务端加密 SSE-KMS

请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:18:30 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

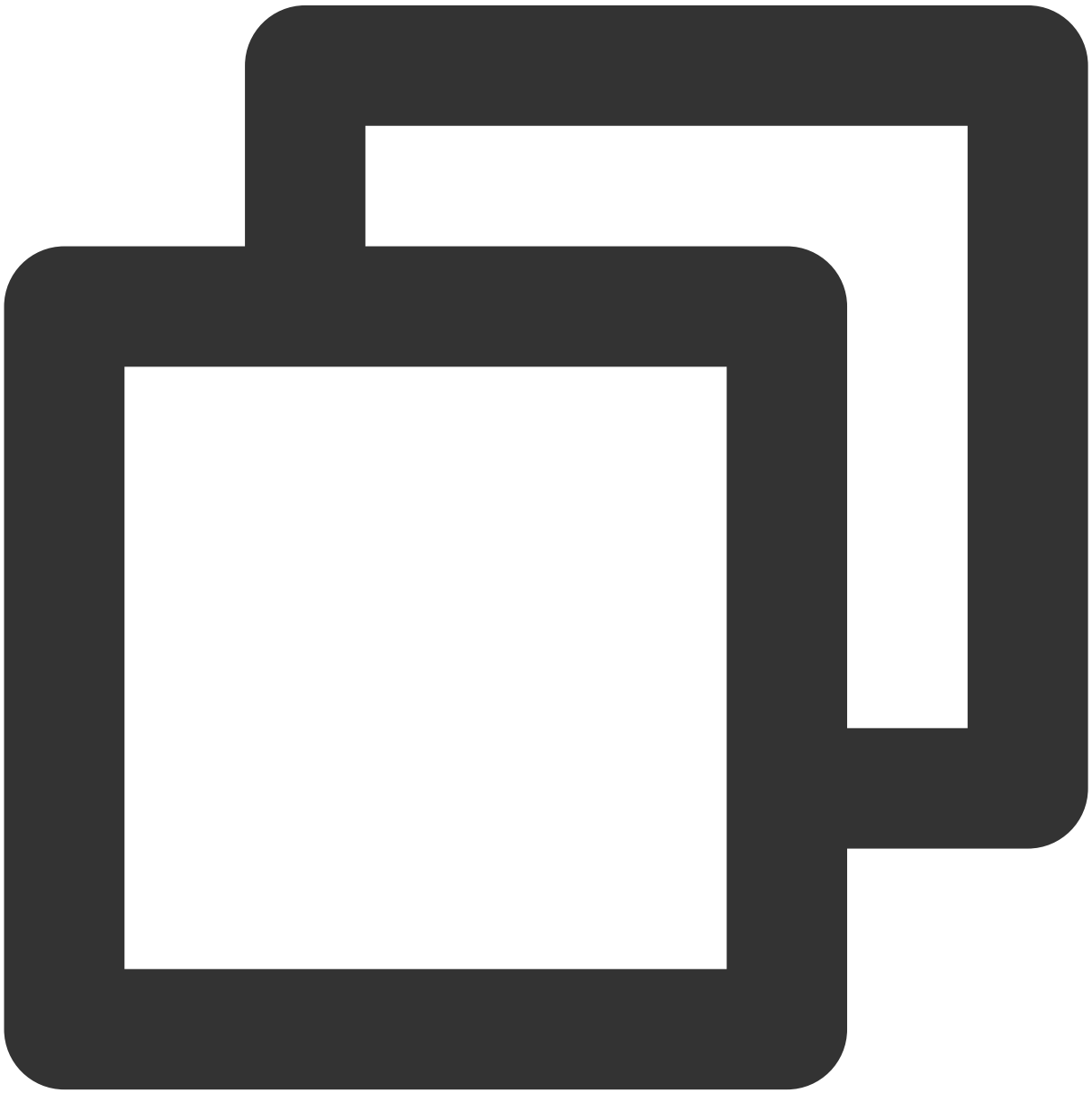


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Date: Fri, 10 Apr 2020 18:18:30 GMT
ETag: "00ca268468481b847fc2d8a9bd84578d"
Last-Modified: Fri, 10 Apr 2020 18:18:19 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
```

```
x-cos-request-id: NWU5MGI4NzZfN2FjODJhMDlfMjU3N18xYTEz****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
```

案例六：使用服务端加密 SSE-C

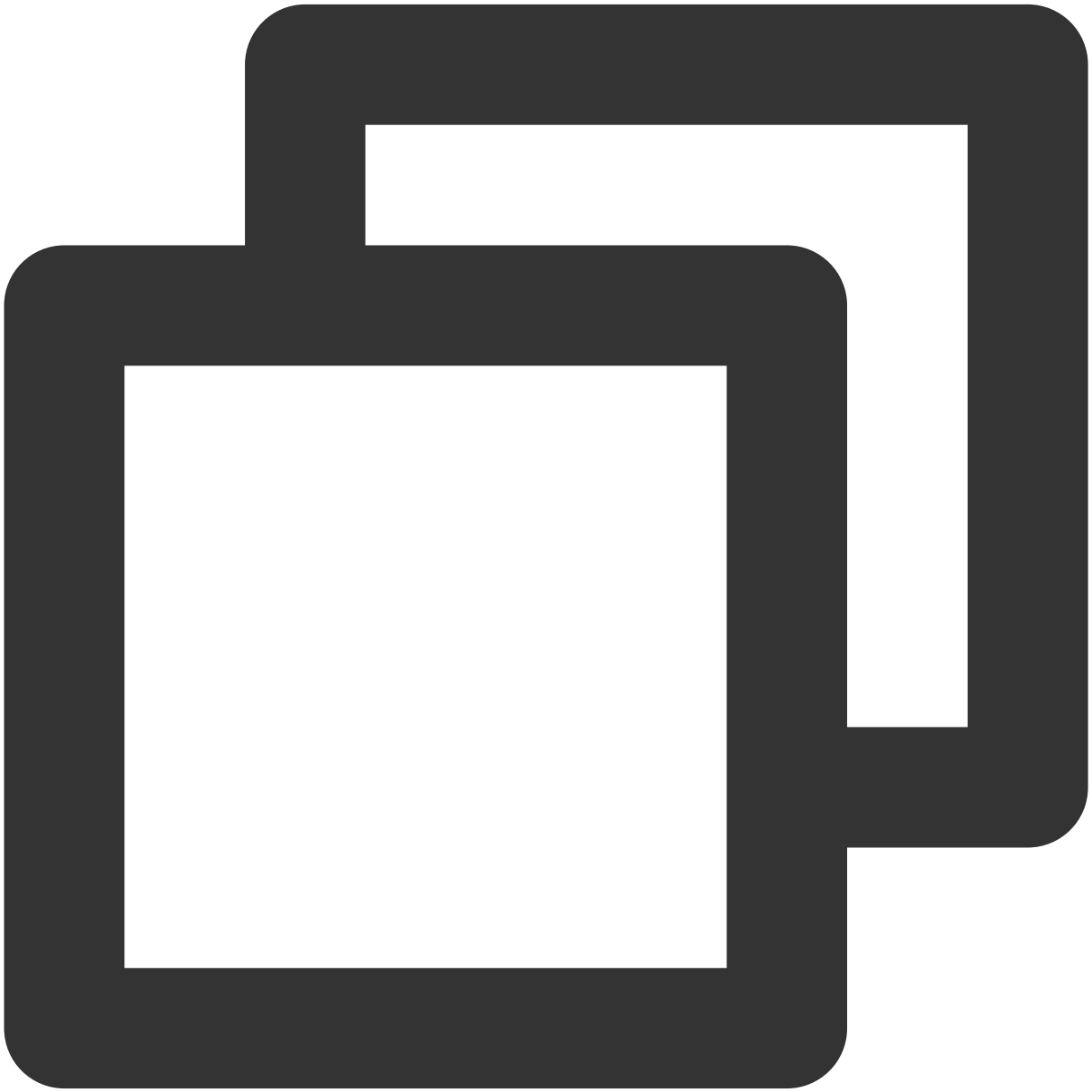
请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Date: Fri, 10 Apr 2020 18:18:41 GMT
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNE
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG****
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

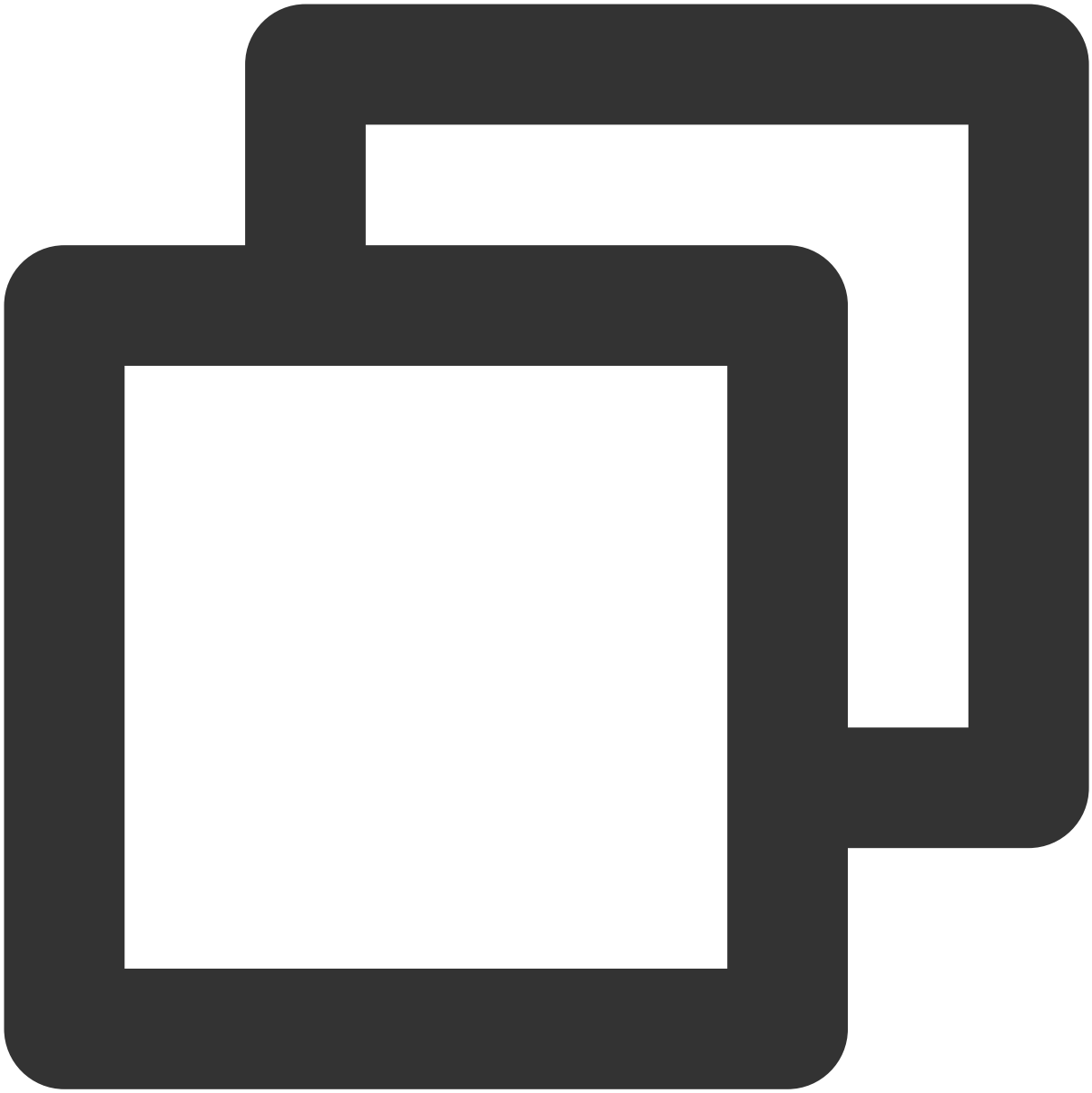


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
```

```
Content-Length: 16
Connection: close
Date: Fri, 10 Apr 2020 18:18:41 GMT
ETag: "582d9105f71525f3c161984bc005efb5"
Last-Modified: Fri, 10 Apr 2020 18:18:31 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MGI4ODFfN2NiODJhMDlfMmFmYTVfMWRh****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG****
```

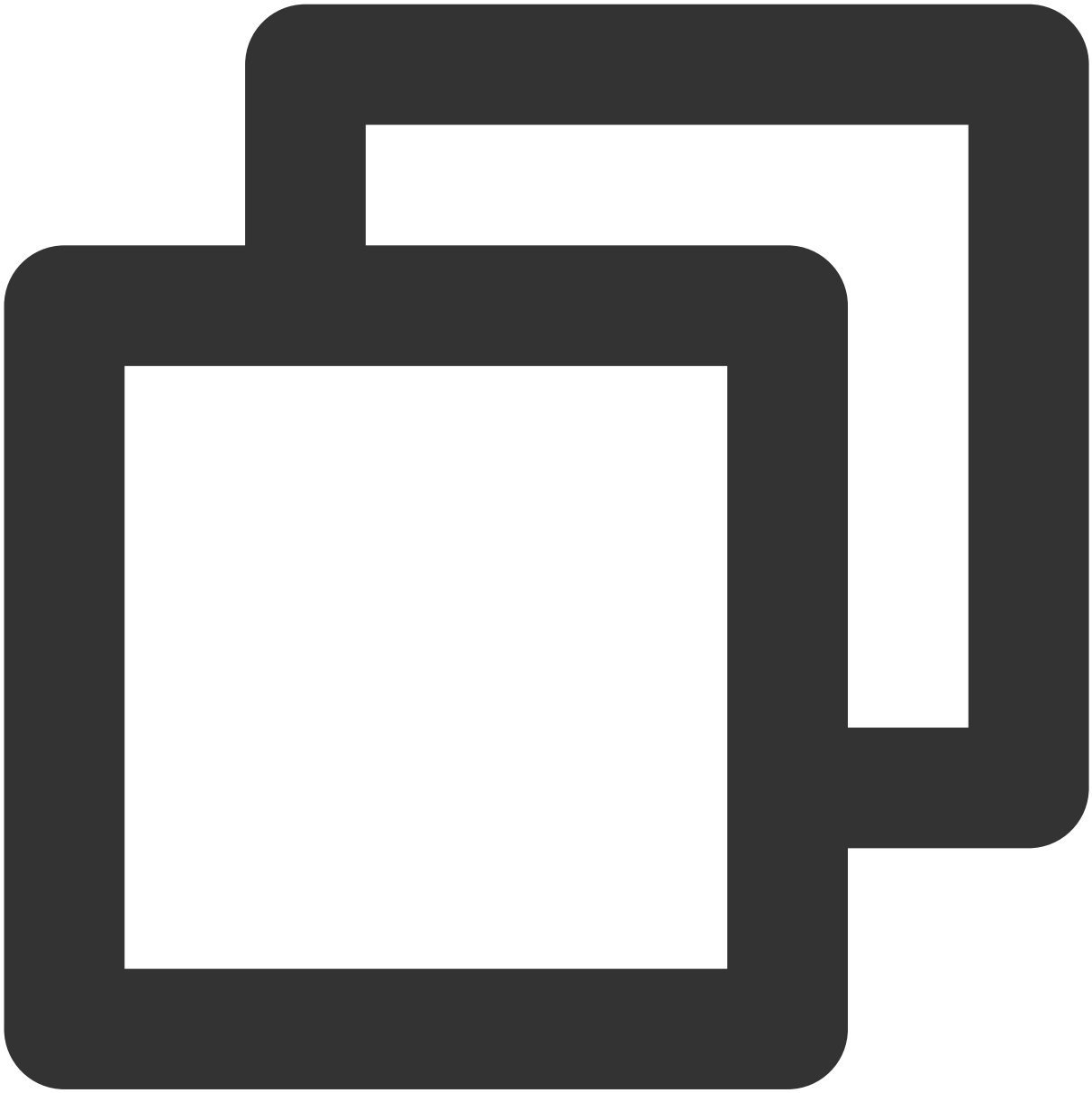
案例七：请求对象最新版本（启用版本控制）

请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:19:15 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

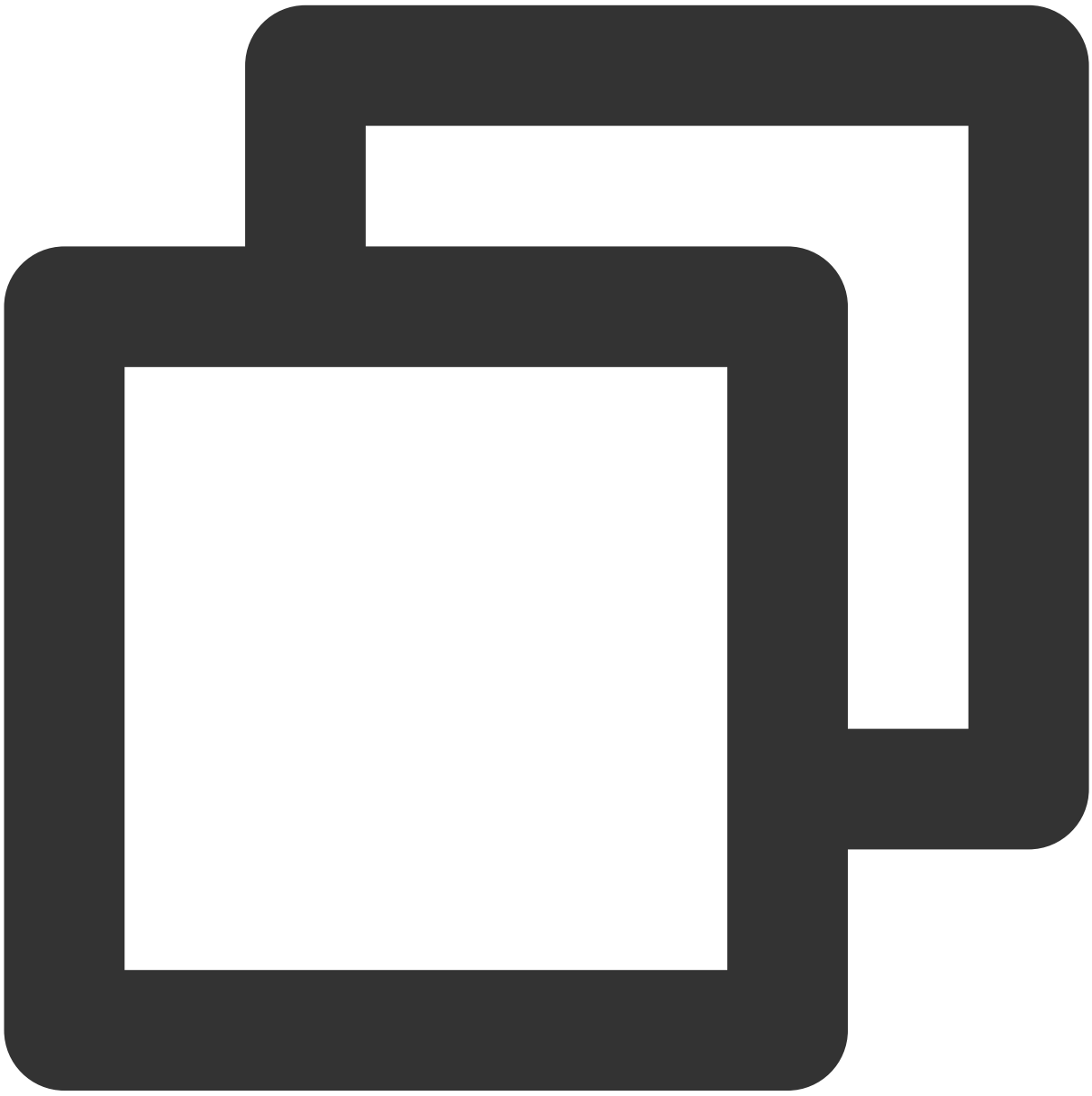
响应



```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 26
Connection: close
Date: Fri, 10 Apr 2020 18:19:15 GMT
ETag: "22e024392de860289f0baa7d6cf8a549"
Last-Modified: Fri, 10 Apr 2020 18:19:04 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 11596229263574363878
x-cos-request-id: NWU5MGI4YTNfZDZjODJhMDlfYmU0MV8xM2Y5****
x-cos-version-id: MTg0NDUxNTc1MzA5NjQ2ODI5MTg
```

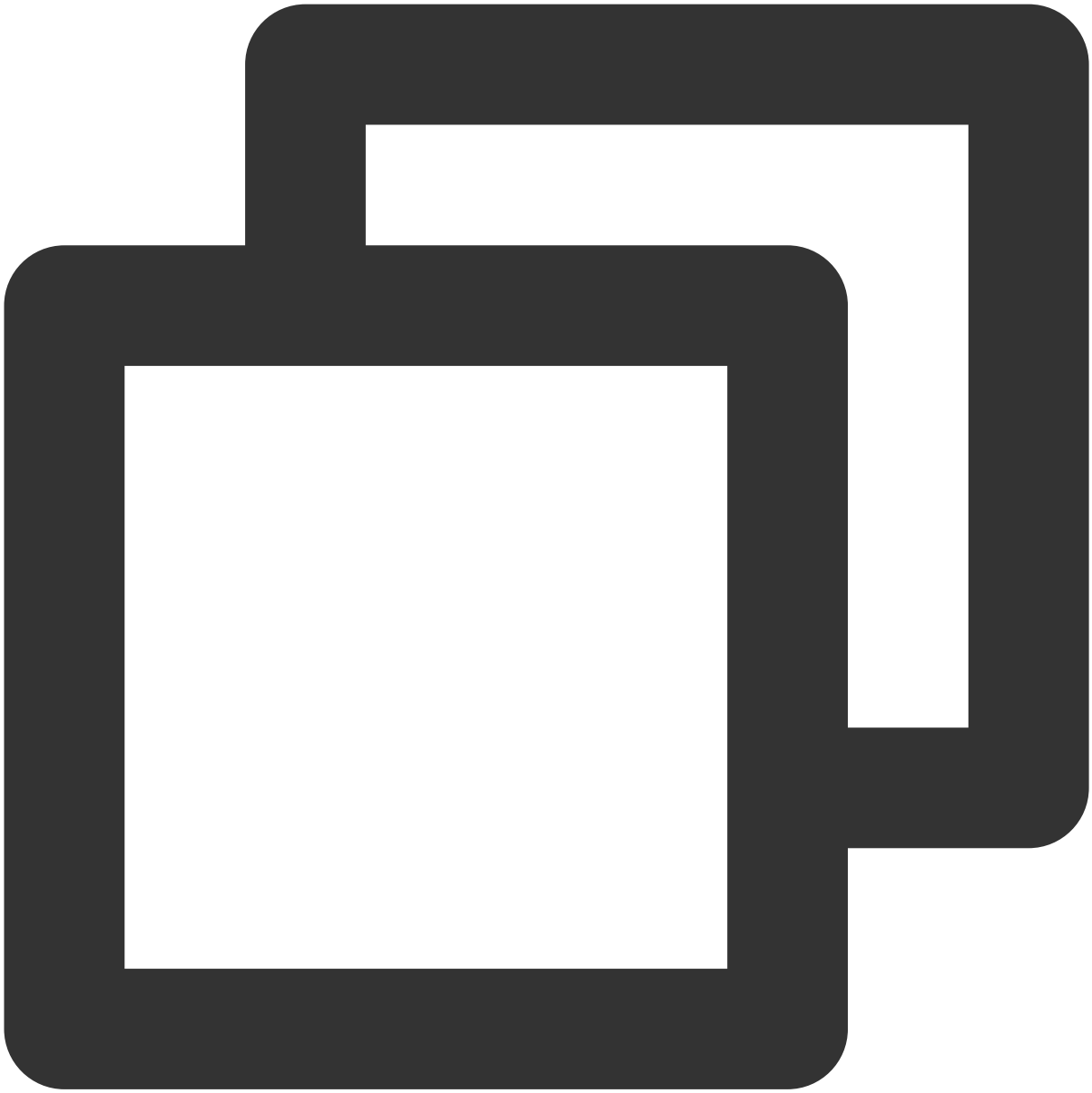
案例八：请求对象指定版本（启用版本控制）

请求



```
HEAD /exampleobject?versionId=MTg0NDUxNTc1MzA5NzU4ODg1Mjg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 18:19:04 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

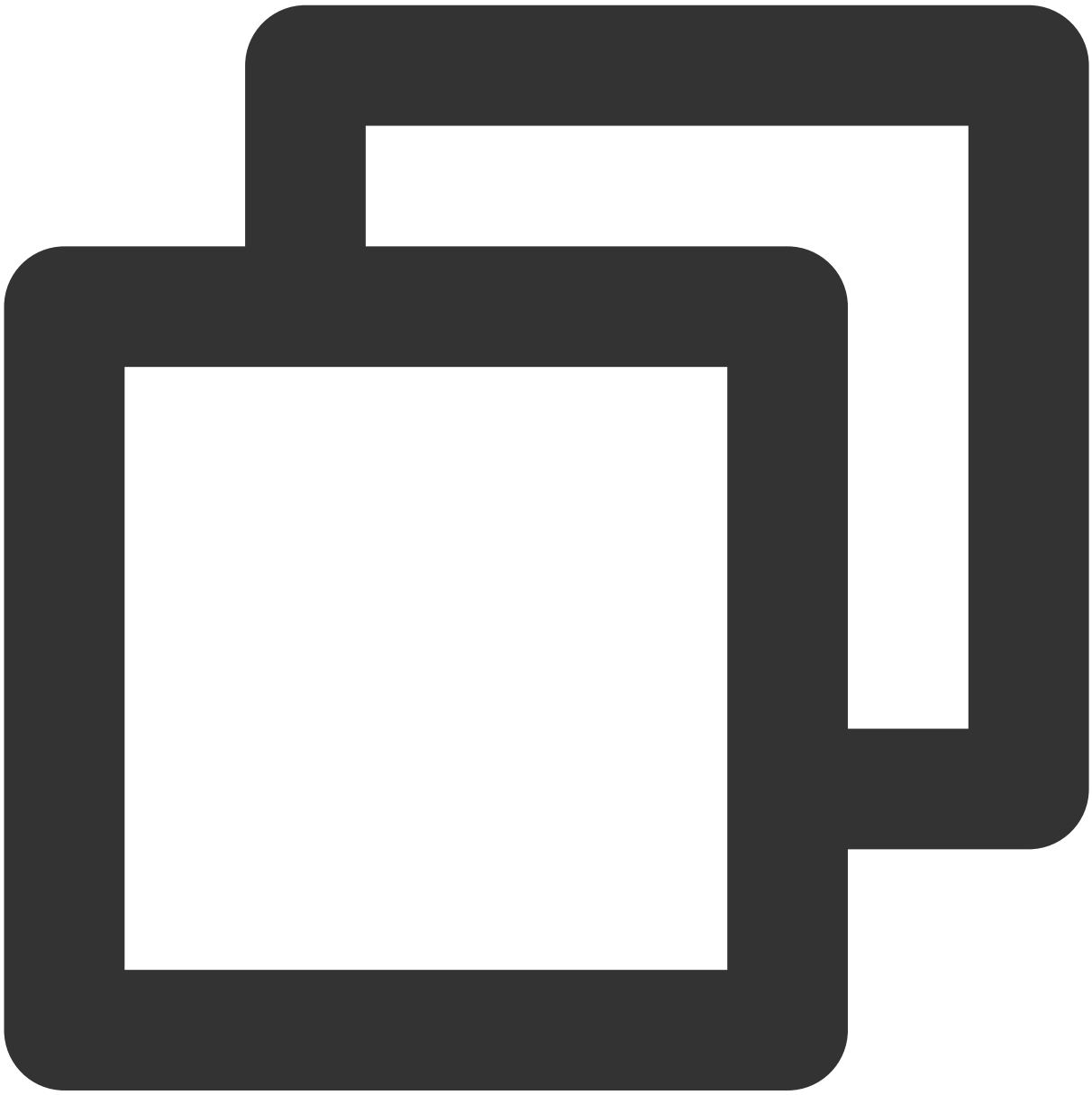


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
Date: Fri, 10 Apr 2020 18:19:04 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 18:18:53 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
```

```
x-cos-request-id: NWU5MGI4OThfN2RiNDBiMDlfMTk1MTBfMWZj****
x-cos-version-id: MTg0NDUxNTc1MzA5NzU4ODg1Mjg
```

案例九：归档存储正在恢复中

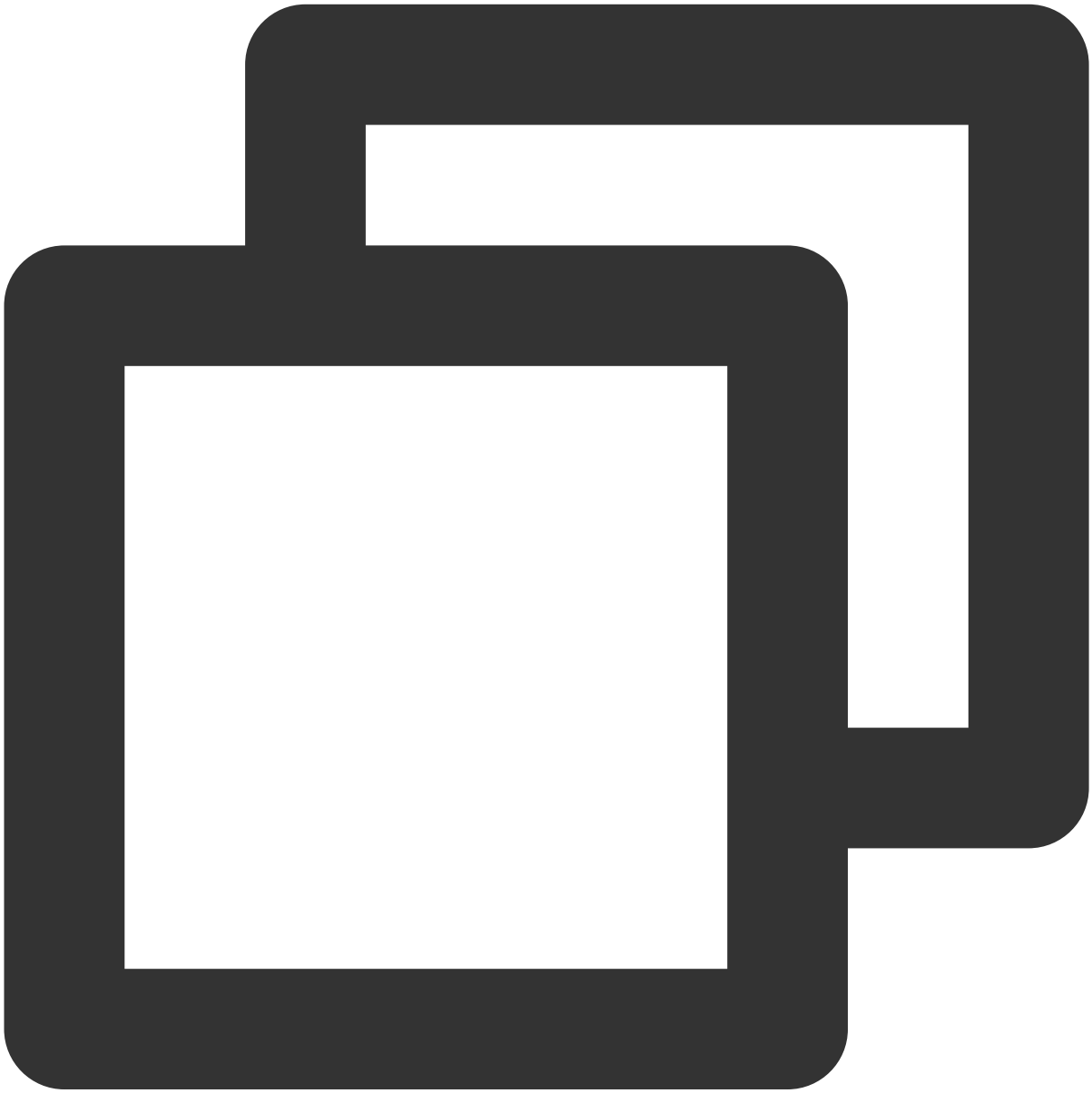
请求



```
HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 27 Dec 2019 08:19:35 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
```

```
Connection: close
```

响应

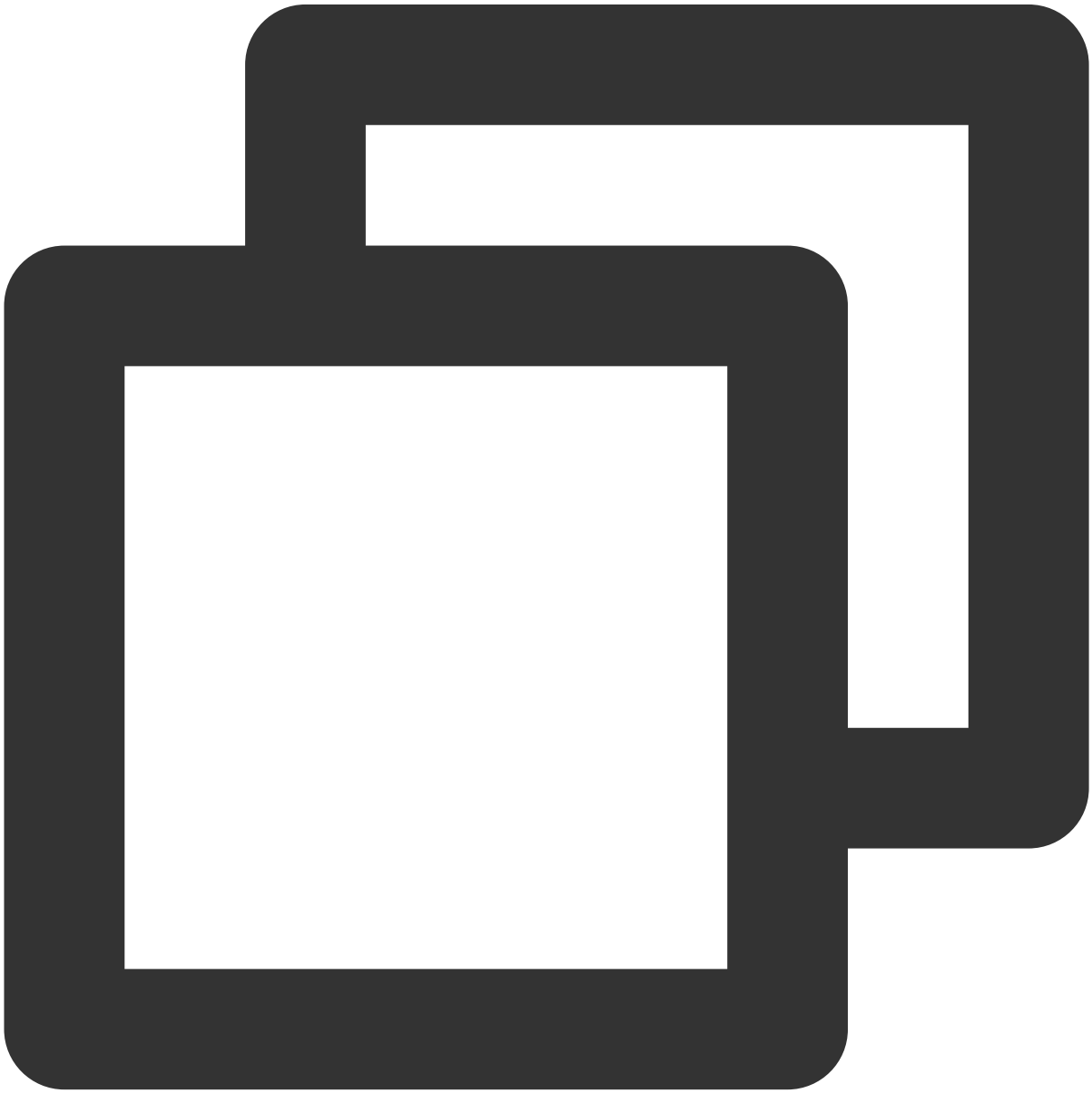


```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Date: Fri, 27 Dec 2019 08:19:35 GMT
ETag: "b62e10bcab55a88240bd9c436cffdcf9"
Last-Modified: Fri, 27 Dec 2019 08:19:23 GMT
```

```
Server: tencent-cos
x-cos-request-id: NWUwNWJlOTdfN2VjODJhMDlfOGI1N18yYjYz****
x-cos-restore: ongoing-request="true"
x-cos-restore-status: tier="expedited"; request-date="Fri, 27 Dec 2019 08:19:29 GMT
x-cos-storage-class: ARCHIVE
```

案例十：归档存储已恢复

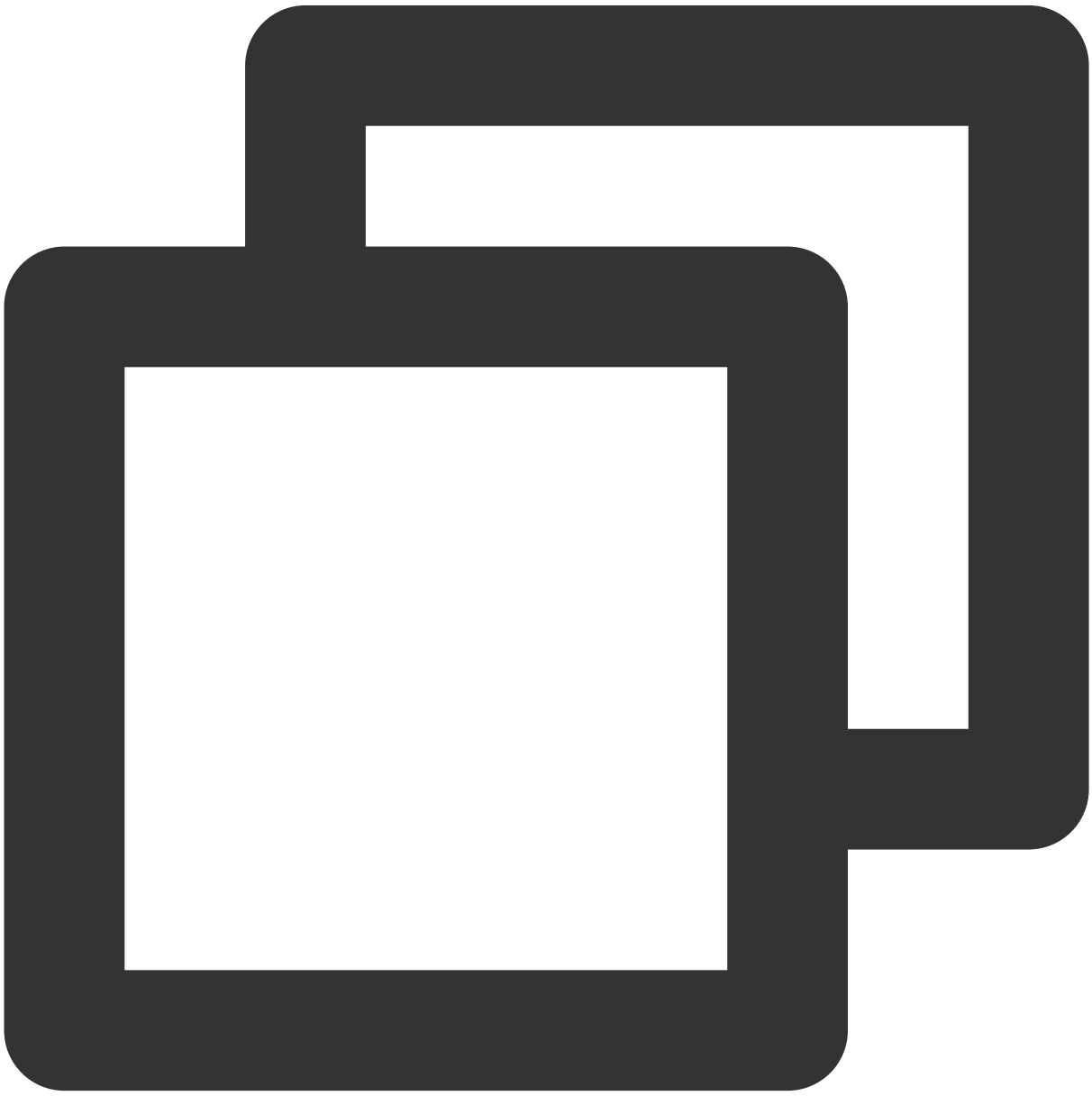
请求



```
HEAD /exampleobject HTTP/1.1
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 02 Jan 2020 18:09:51 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 16
Connection: close
```

```
Date: Fri, 10 Apr 2020 18:17:36 GMT
ETag: "ee8de918d05640145b18f70f4c3aa602"
Last-Modified: Fri, 10 Apr 2020 18:17:25 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 16749565679157681890
x-cos-request-id: NWU5MWRiZjFfMmViMDJhMDlfNjIwOF8zNTU0****
x-cos-restore: ongoing-request="false", expiry-date="Mon, 13 Apr 2020 16:00:00 GMT"
x-cos-storage-class: ARCHIVE
```

DELETE Object

最近更新时间：2023-12-21 14:32:38

功能描述

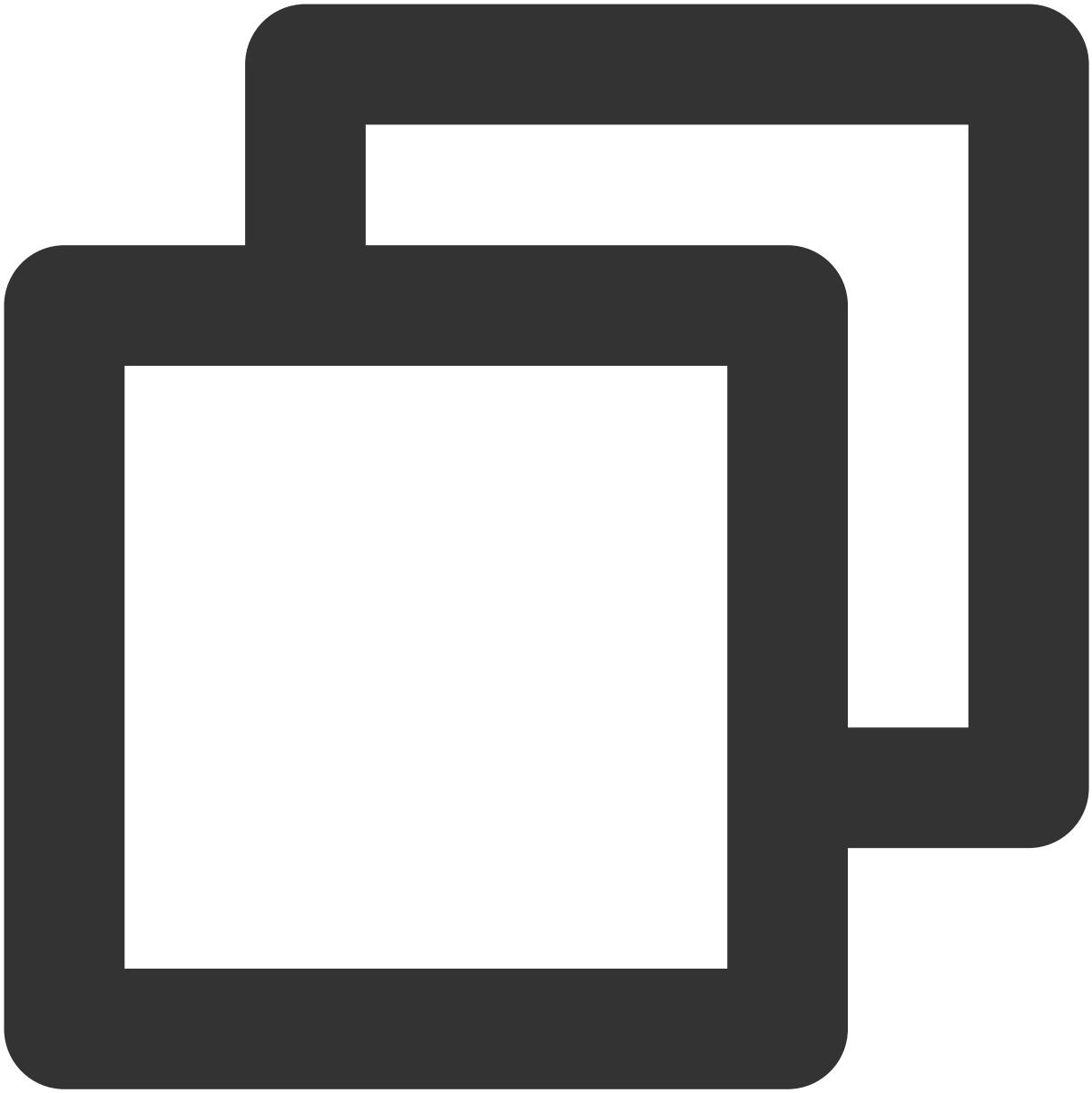
DELETE Object 接口请求可以删除一个指定的对象（Object）。该 API 的请求者需要对存储桶有写入权限。

版本控制

如需删除对象的指定版本（包括删除标记，下同），请使用 versionId 请求参数指定对应的版本 ID （包括删除标记的版本 ID，下同），此时响应将返回 x-cos-version-id 响应头部，代表该请求操作删除的版本 ID。
如未指定 versionId 请求参数：
当版本控制为启用时，该 DELETE 操作将创建一个删除标记作为指定对象的最新版本，此时响应将返回 x-cos-version-id 响应头部，代表该请求操作创建的删除标记的版本 ID。
当版本控制为暂停时，该 DELETE 操作将创建一个版本 ID 为 null 的删除标记作为指定对象的最新版本，同时删除任何已存在的版本 ID 为 null 的其他版本（如有）。
当该 DELETE 操作创建或删除了删除标记，那么将返回 x-cos-delete-marker: true 响应头部，代表该 DELETE 操作创建或删除了指定对象的删除标记。
有关版本控制的启用或暂停状态说明，请参见 版本控制概述。

请求

请求示例



```
DELETE /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

名称	描述	类型	是否必选
versionId	指定要删除的版本 ID	string	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

版本控制相关头部

删除启用版本控制的存储桶内的对象或对象的指定版本将返回下列响应头部：

名称	描述	类型
x-cos-version-id	对象的版本 ID 或删除标记的版本 ID	string
x-cos-delete-marker	当使用 versionId 请求参数指定删除标记的版本 ID 时，将返回此响应头部且值为 true，代表删除的版本 ID 对应的是一个删除标记 当未使用 versionId 请求参数，且指定对象所在的存储桶启用了版本控制时，将返回此响应头部且值为 true，代表该删除请求创建了一个删除标记作为对象的最新版本	boolean

响应体

此响应体为空。

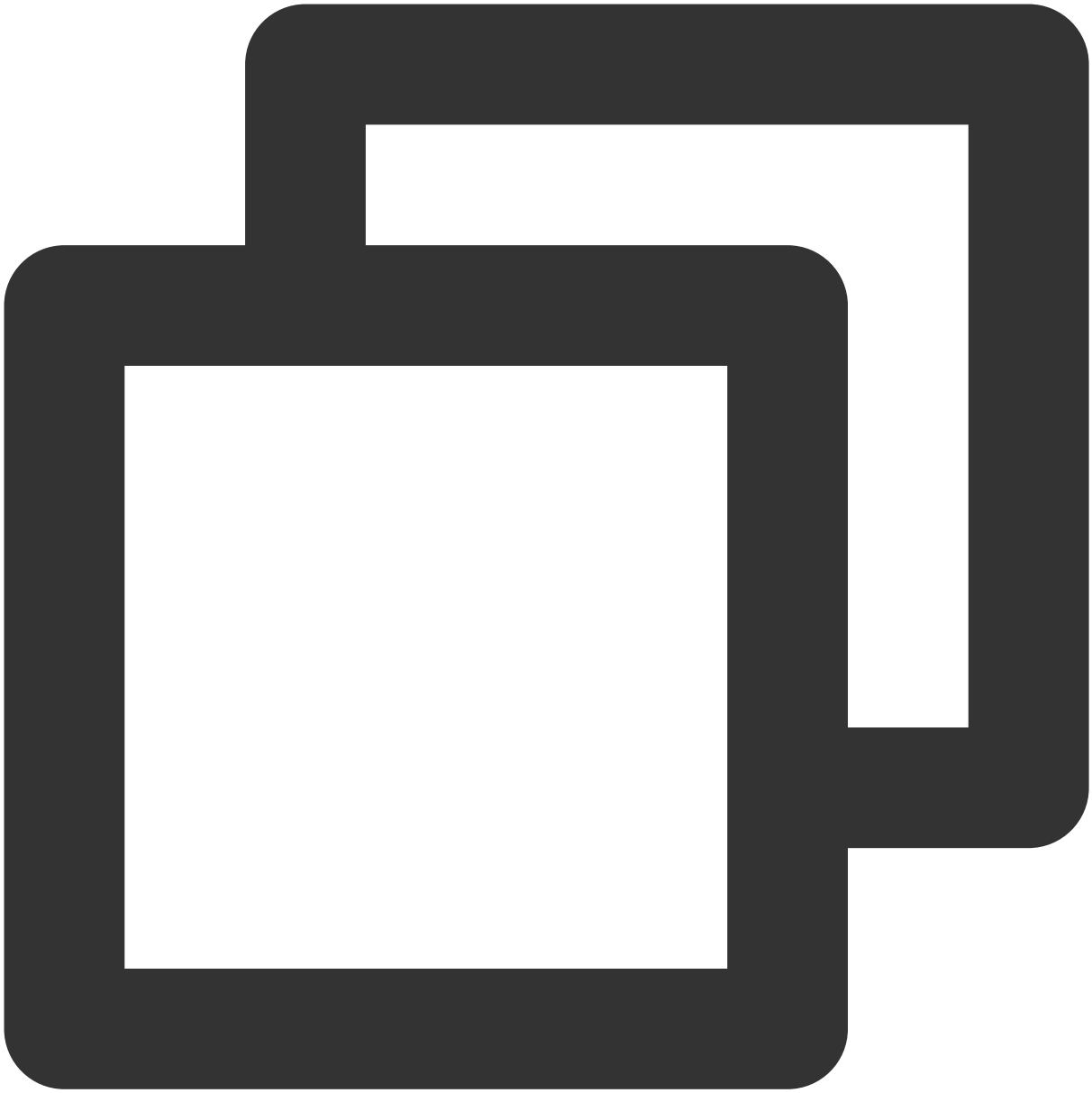
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

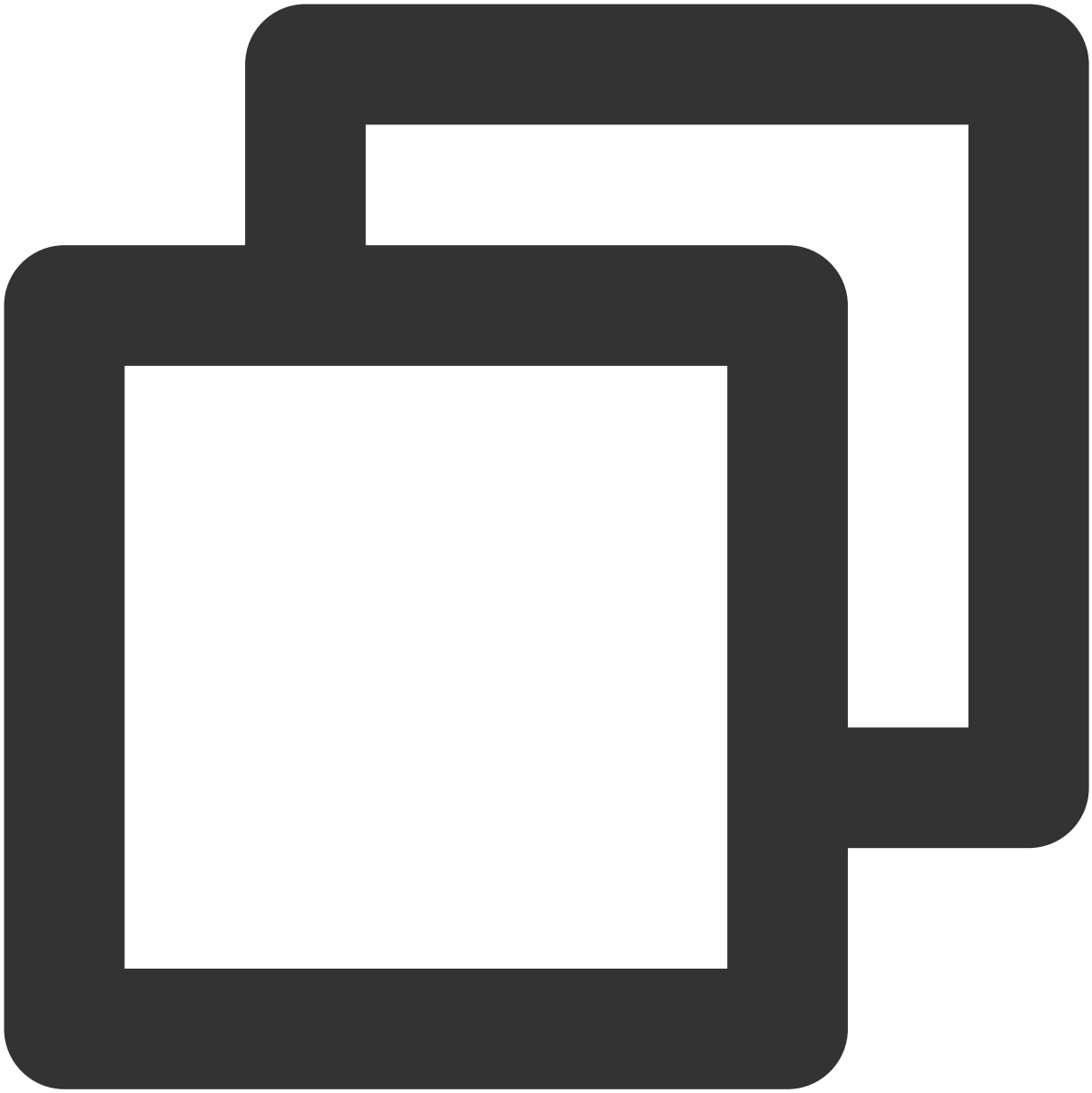
案例一：未启用版本控制

请求



```
DELETE /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 14 Aug 2019 11:59:40 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

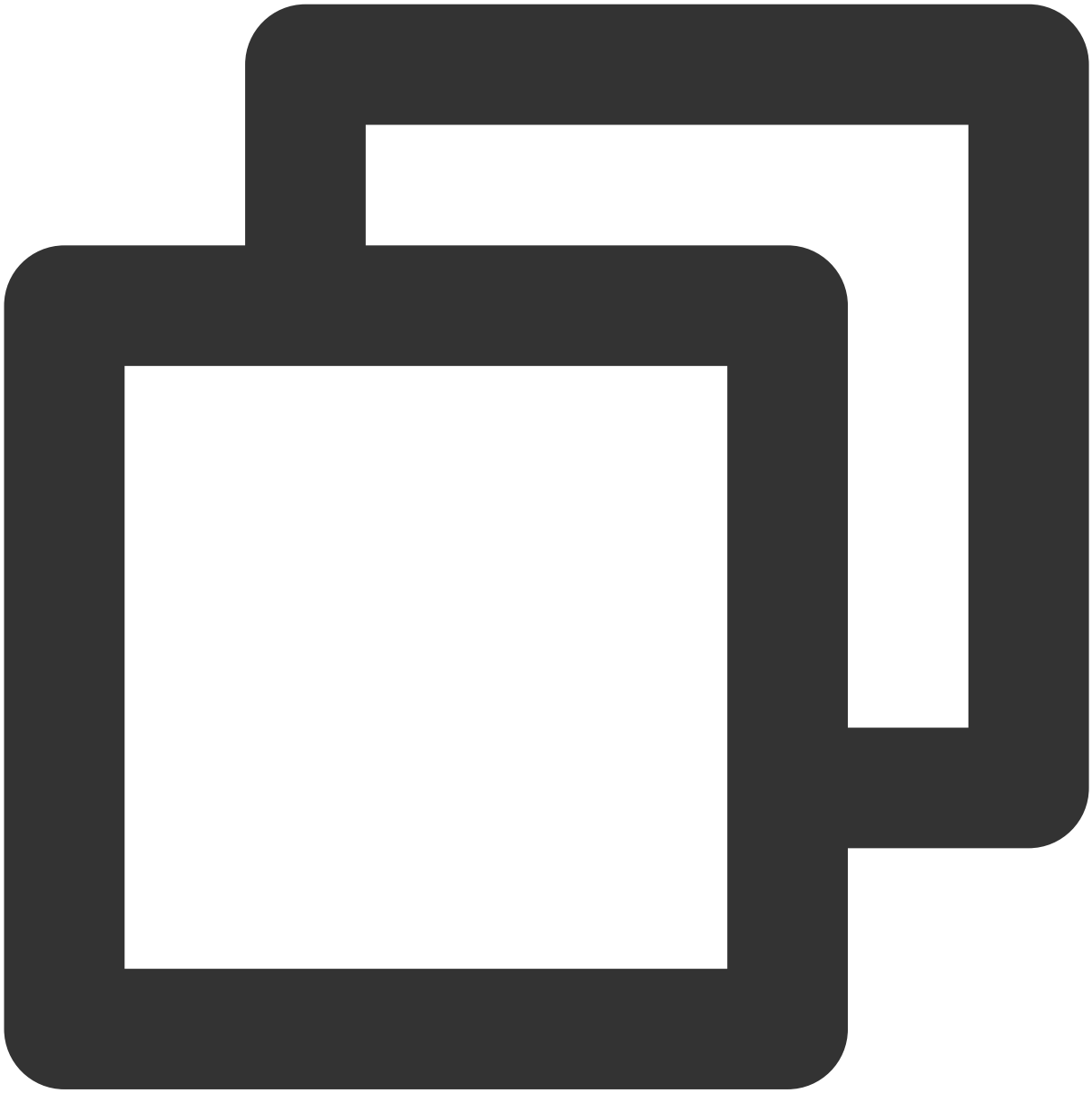
响应



```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
Date: Wed, 14 Aug 2019 11:59:40 GMT
Server: tencent-cos
x-cos-request-id: NWQ1M2Y3YWNfMzdiMDJhMDlfODA1Yl8xZThj****
```

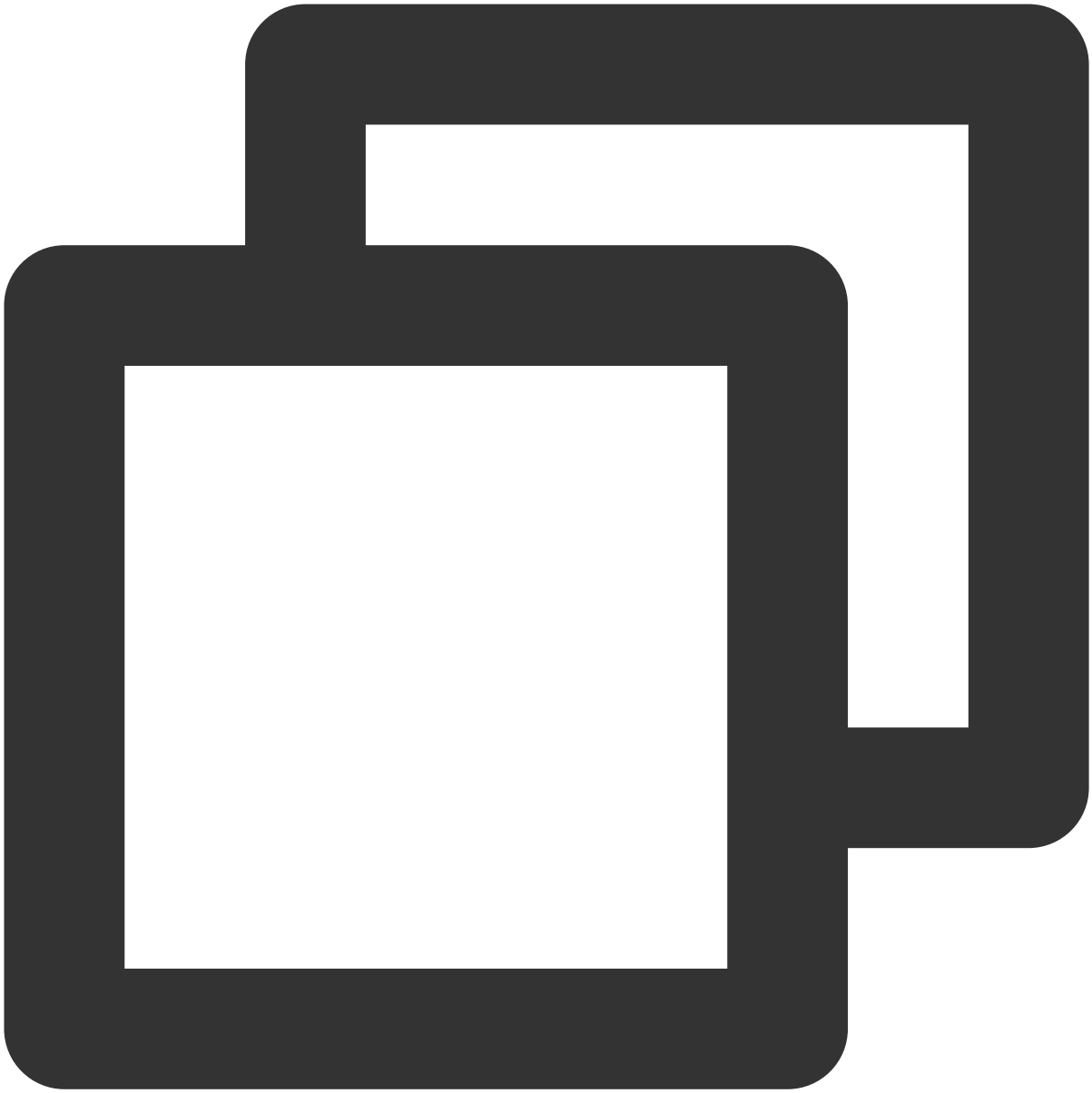
案例二：启用版本控制（创建删除标记）

请求



```
DELETE /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 14 Aug 2019 12:00:21 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

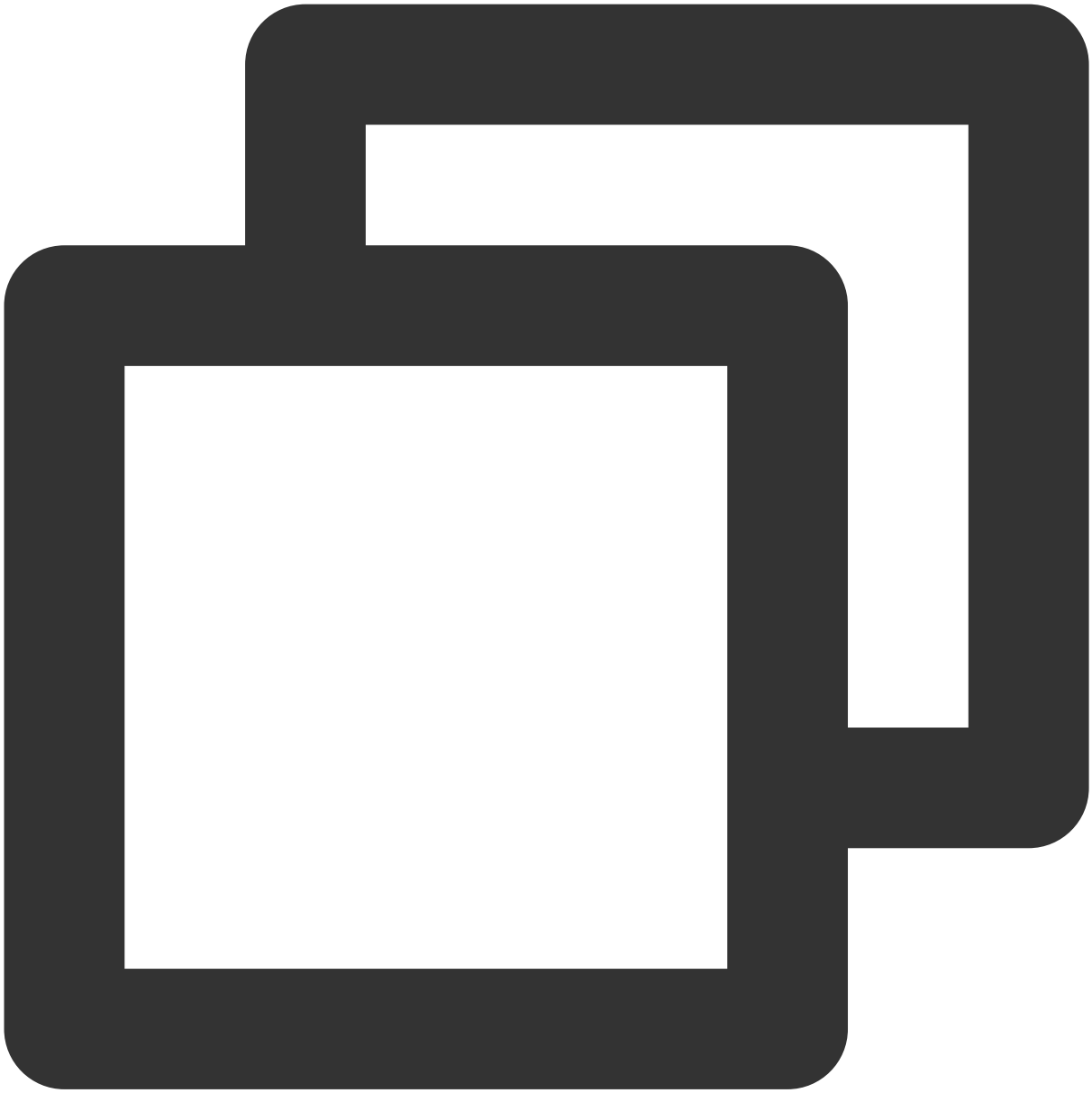
响应



```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
Date: Wed, 14 Aug 2019 12:00:21 GMT
Server: tencent-cos
x-cos-delete-marker: true
x-cos-request-id: NWQ1M2Y3ZDVfN2RiNDBiMDlfMmMwNmVfMTc4****
x-cos-version-id: MTg0NDUxNzgyODk2ODc1NjY0NzQ
```

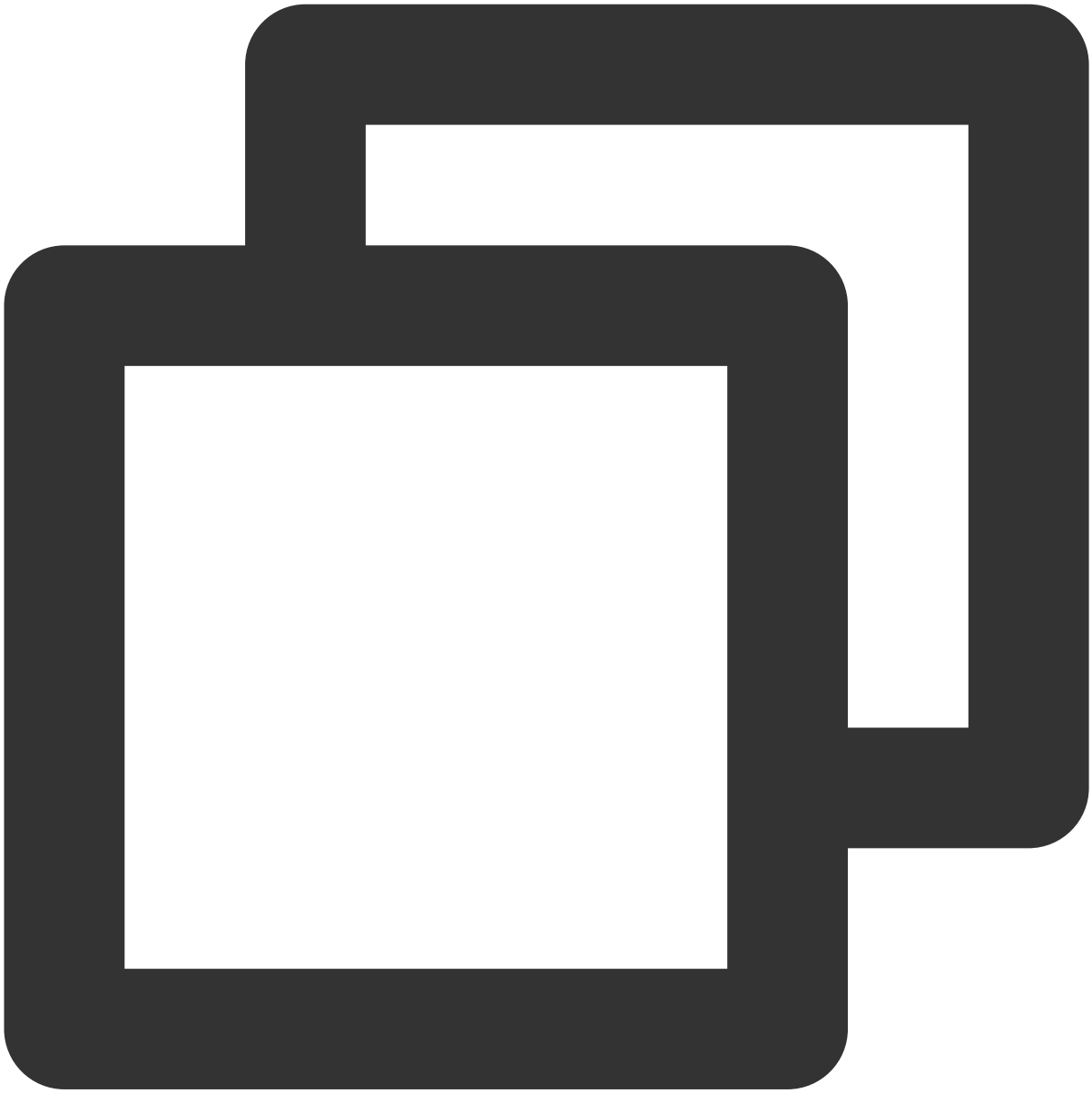
案例三：永久删除指定版本

请求



```
DELETE /exampleobject?versionId=MTg0NDUxNzgyODk3MDgyMzI4NDY HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 14 Aug 2019 12:00:32 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

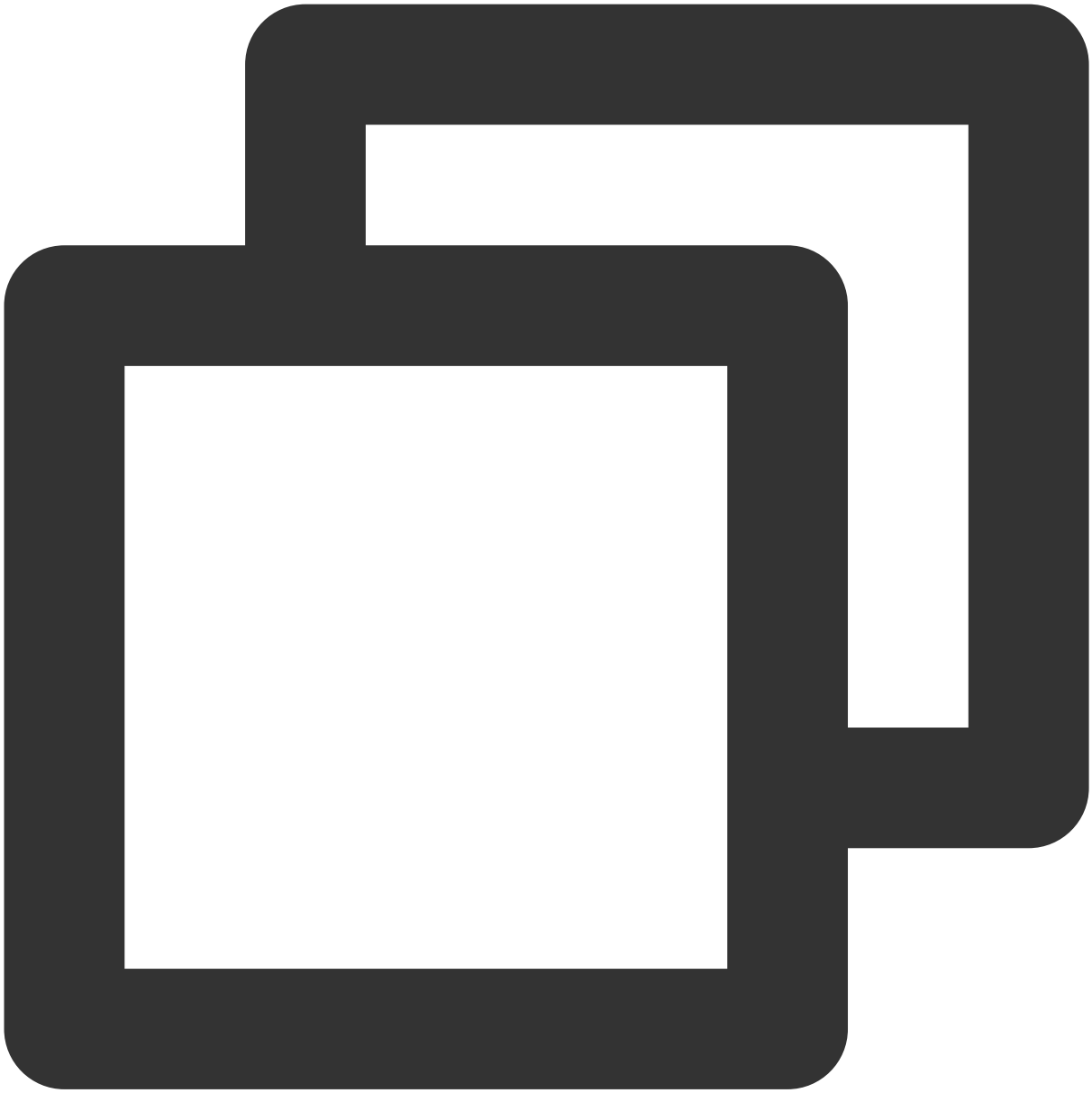
响应



```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
Date: Wed, 14 Aug 2019 12:00:32 GMT
Server: tencent-cos
x-cos-request-id: NWQ1M2Y3ZTBfODhjMjJhMDlfMWNkOF8xZDZi****
x-cos-version-id: MTg0NDUxNzgyODk3MDgyMzI4NDY
```

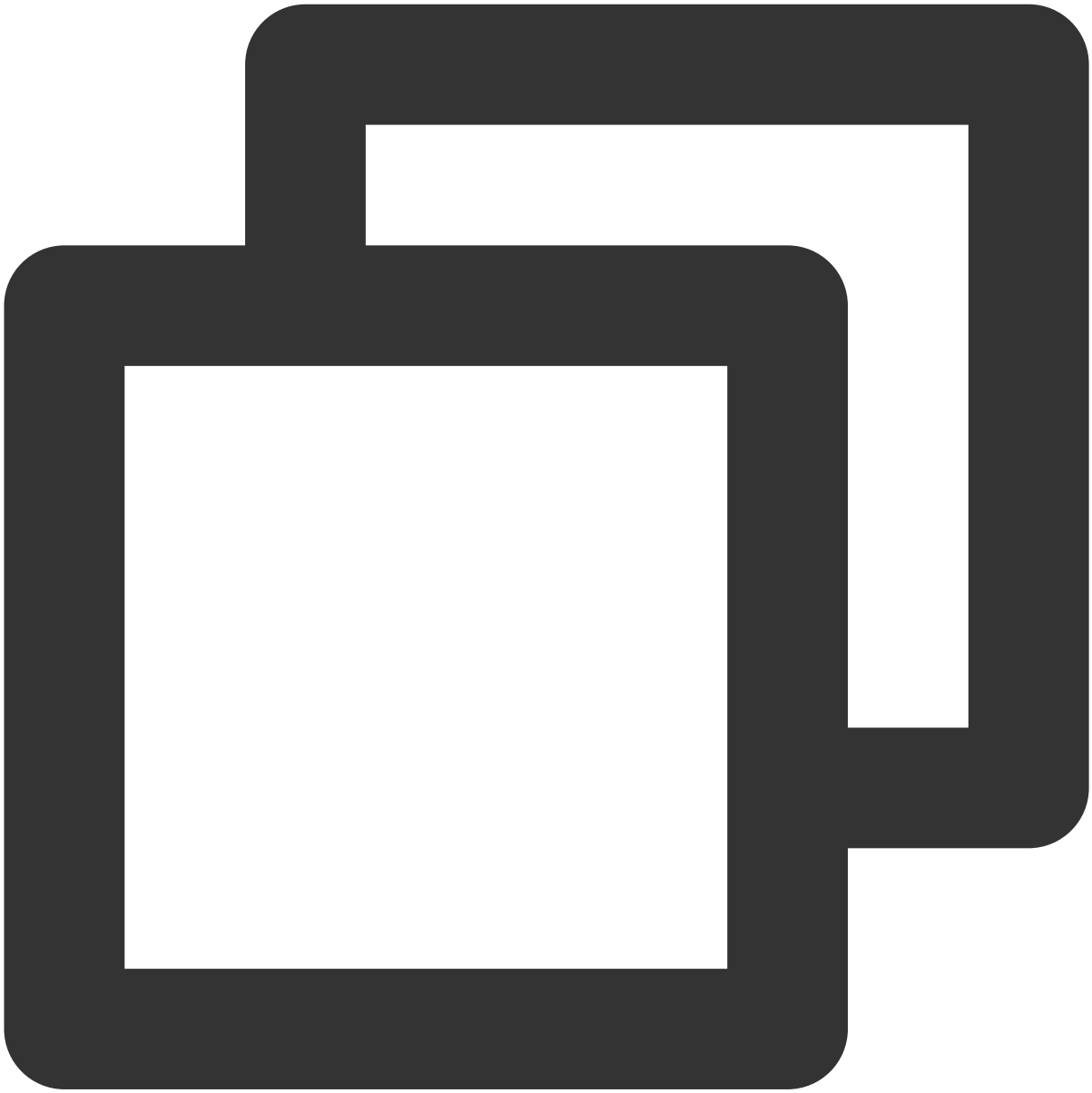
案例四：永久删除指定删除标记

请求



```
DELETE /exampleobject?versionId=MTg0NDUxNzgyODk2ODc1NjY0NzQ HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 14 Aug 2019 12:00:42 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
Date: Wed, 14 Aug 2019 12:00:42 GMT
Server: tencent-cos
x-cos-delete-marker: true
x-cos-request-id: NWQ1M2Y3ZWFfNzljMDBiMDlfMjkyMDJfMWRjNjVm****
x-cos-version-id: MTg0NDUxNzgyODk2ODc1NjY0NzQ
```

DELETE Multiple Objects

最近更新时间：2023-12-21 14:32:59

功能描述

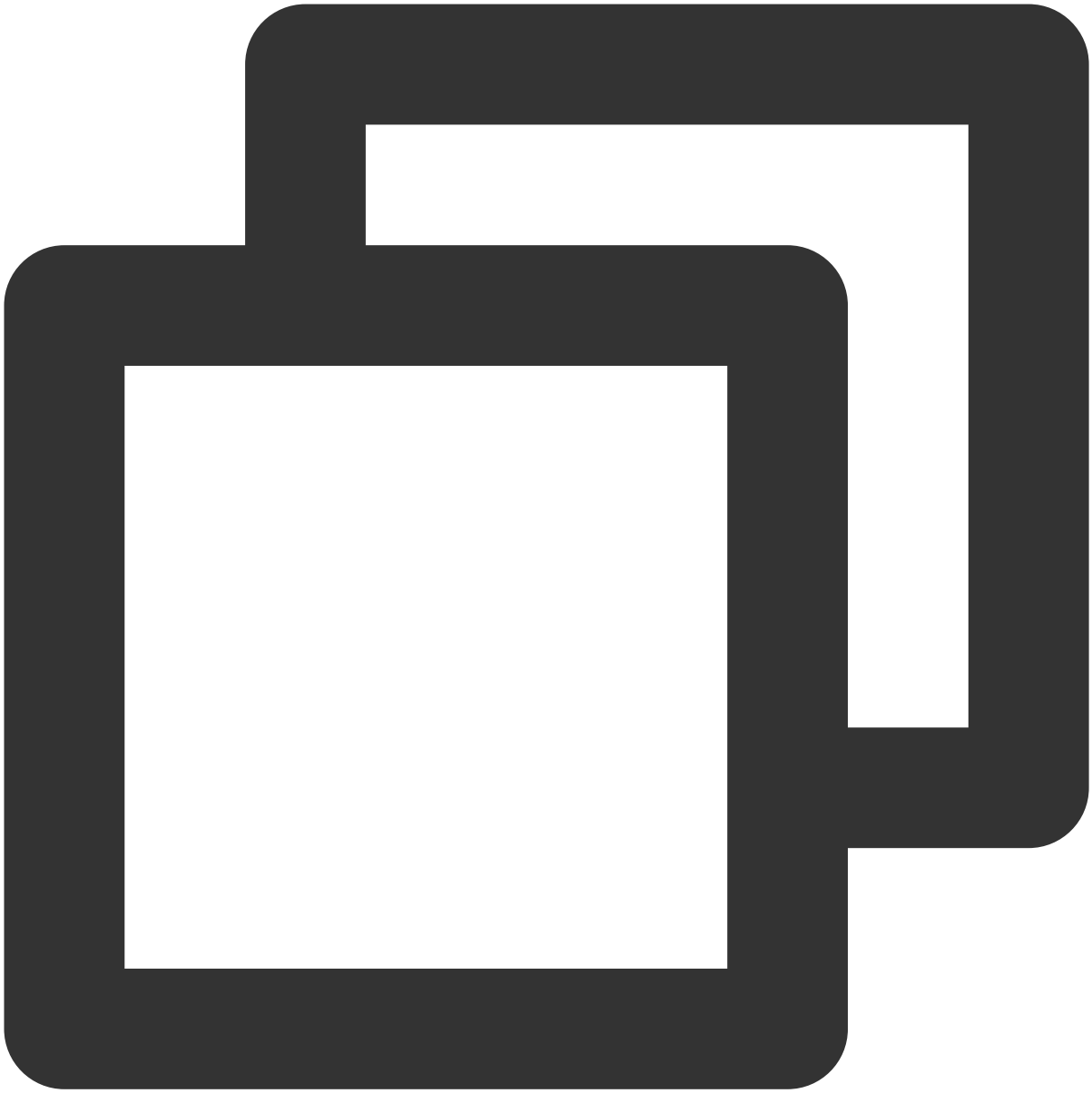
DELETE Multiple Objects 接口请求可以批量删除指定存储桶中的多个对象（Object），单次请求支持最多删除1000个对象。对于响应结果，COS 提供 Quiet 模式和 Verbose 模式：
Quiet 模式在响应中仅包含删除失败的对象信息和错误信息。
Verbose 模式在响应中包含每个对象的删除结果信息。
该 API 的请求者需要对存储桶有写入权限。

版本控制

当启用版本控制时，该请求操作可以为每一个要删除的对象指定版本 ID，此时将永久删除对象的指定版本或指定删除标记，否则将创建一个删除标记作为指定对象的最新版本。
当针对某个对象的删除操作创建或删除了删除标记，那么该对象的删除结果将同时返回 <DeleteMarker>true</DeleteMarker> 和 <DeleteMarkerVersionId> 元素，代表该请求操作创建或删除了指定对象的删除标记。
当针对某个对象的删除操作永久删除了特定的版本 ID（包括删除标记的版本 ID），那么该对象的删除结果将返回 <VersionId> 元素，代表该请求操作删除的版本 ID。

请求

请求示例



```
POST /?delete HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

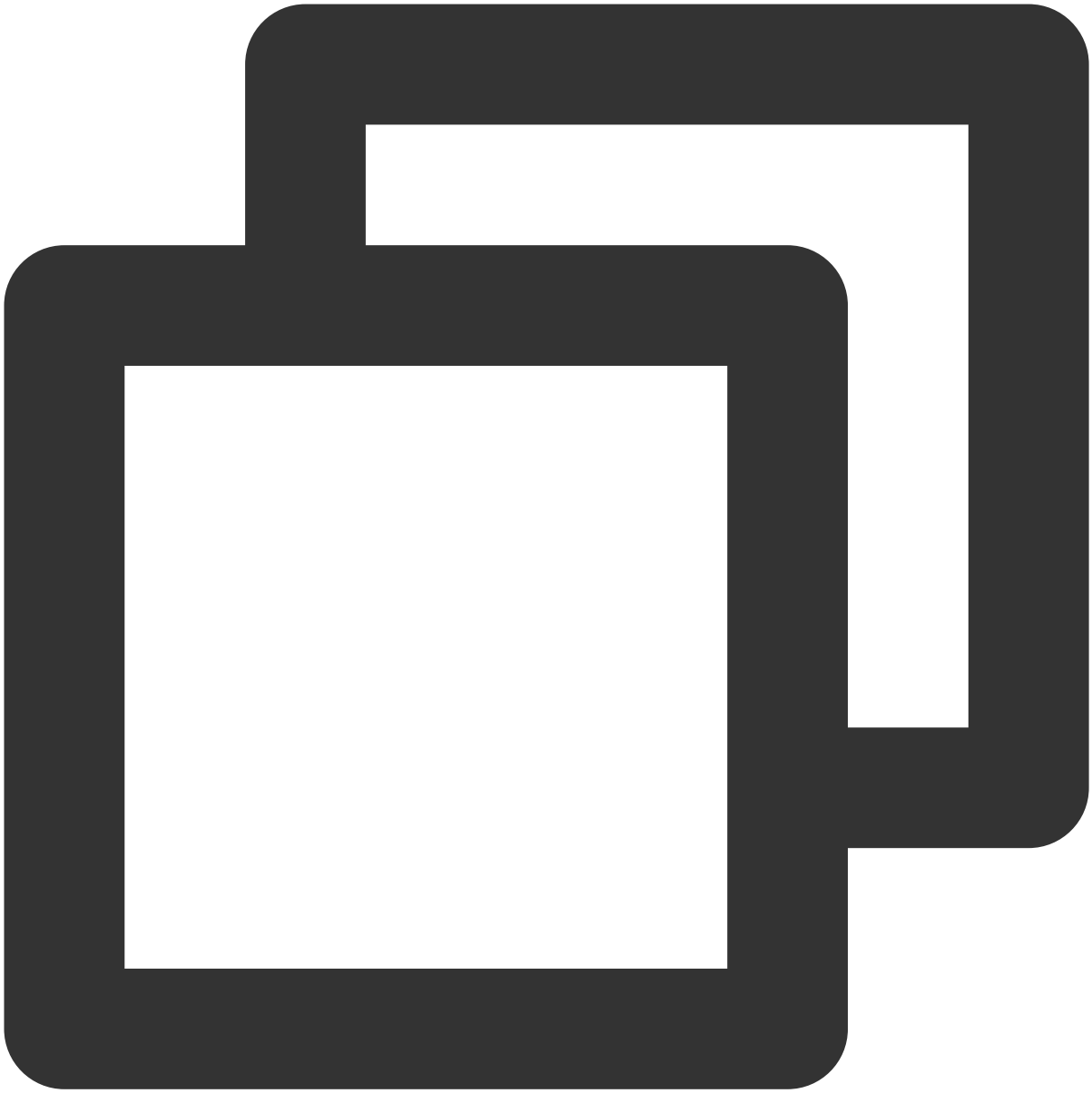
此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

提交 **application/xml** 请求数据，包含要删除的对象信息。



```
<Delete>
    <Quiet>boolean</Quiet>
    <Object>
        <Key>string</Key>
    </Object>
    <Object>
        <Key>string</Key>
        <VersionId>string</VersionId>
    </Object>
</Delete>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Delete	无	包含 DELETE Multiple Objects 操作的所有请求信息	Container	是

Container 节点 Delete 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Quiet	Delete	布尔值，默认为 false true 为使用 Quiet 模式，在响应中仅包含删除失败的对象信息和错误信息 false 为使用 Verbose 模式，在响应中包含每个对象的删除结果	boolean	是
Object	Delete	单个要删除的目标对象的信息	Container	是

Container 节点 Object 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Key	Delete.Object	要删除的目标对象的对象键	string	是
VersionId	Delete.Object	当启用版本控制并且要删除对象的指定版本时需指定该元素，值为要删除的版本 ID。若未开启版本控制或开启版本控制但需要插入删除标记，则无需指定该元素	string	是

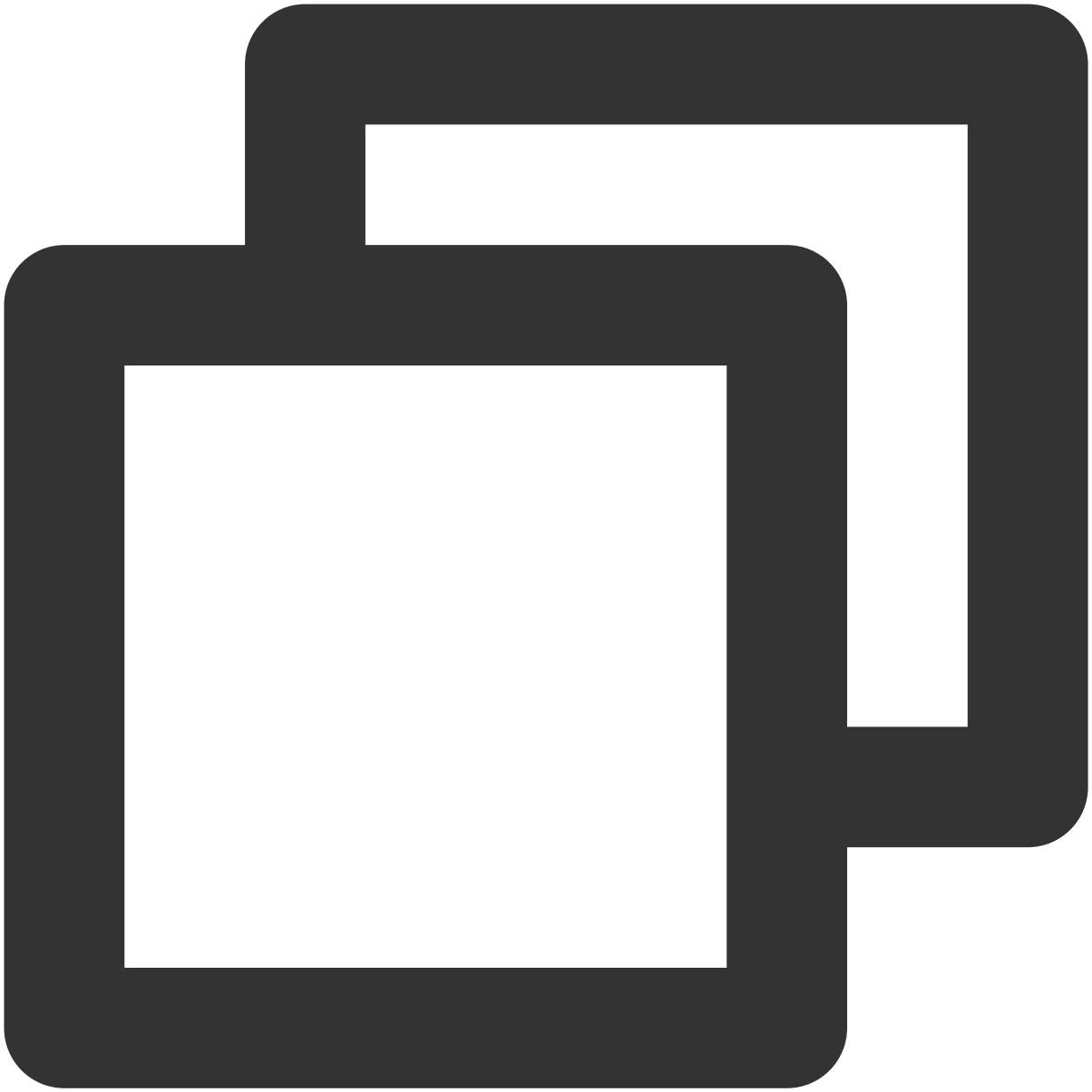
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

请求成功，返回 **application/xml** 数据，包含删除结果信息。



```
<DeleteResult>
    <Deleted>
        <Key>string</Key>
        <DeleteMarker>boolean</DeleteMarker>
        <DeleteMarkerVersionId>string</DeleteMarkerVersionId>
    </Deleted>
    <Deleted>
        <Key>string</Key>
        <VersionId>string</VersionId>
    </Deleted>
    <Deleted>
```

```
        <Key>string</Key>
        <DeleteMarker>boolean</DeleteMarker>
        <DeleteMarkerVersionId>string</DeleteMarkerVersionId>
        <VersionId>string</VersionId>
    </Deleted>
    <Error>
        <Key>string</Key>
        <VersionId>string</VersionId>
        <Code>string</Code>
        <Message>string</Message>
    </Error>
</DeleteResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
DeleteResult	无	保存 DELETE Multiple Objects 结果的所有信息	Container

Container 节点 DeleteResult 的内容：

节点名称（关键字）	父节点	描述	类型
Deleted	DeleteResult	单个删除成功的对象条目，仅当使用 Verbose 模式才会返回该元素	Container
Error	DeleteResult	单个删除失败的对象条目	Container

Container 节点 Deleted 的内容：

节点名称（关键字）	父节点	描述	类型
Key	DeleteResult.Deleted	删除成功的对象的对象键	string
DeleteMarker	DeleteResult.Deleted	仅当对该对象的删除创建了一个删除标记，或删除的是该对象的一个删除标记时才返回该元素，布尔值，固定为 true	boolean
DeleteMarkerVersionId	DeleteResult.Deleted	仅当对该对象的删除创建了一个删除标记，或删除的是该对象的一个删除标记时才返回该元素，值为创建或删除的删除标记的版本 ID	string
VersionId	DeleteResult.Deleted	删除成功的版本 ID，仅当请求中指定了要删除对象的版本 ID 时才返回该元素	string

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 Error 的内容：

节点名称（关键字）	父节点	描述	类型
Key	DeleteResult.Error	删除失败的对象的对象键	string
VersionId	DeleteResult.Error	删除失败的版本 ID，仅当请求中指定了要删除对象的版本 ID 时才返回该元素	string
Code	DeleteResult.Error	删除失败的错误码，用来定位唯一的错误条件和确定错误场景	string
Message	DeleteResult.Error	删除失败的具体错误信息	string

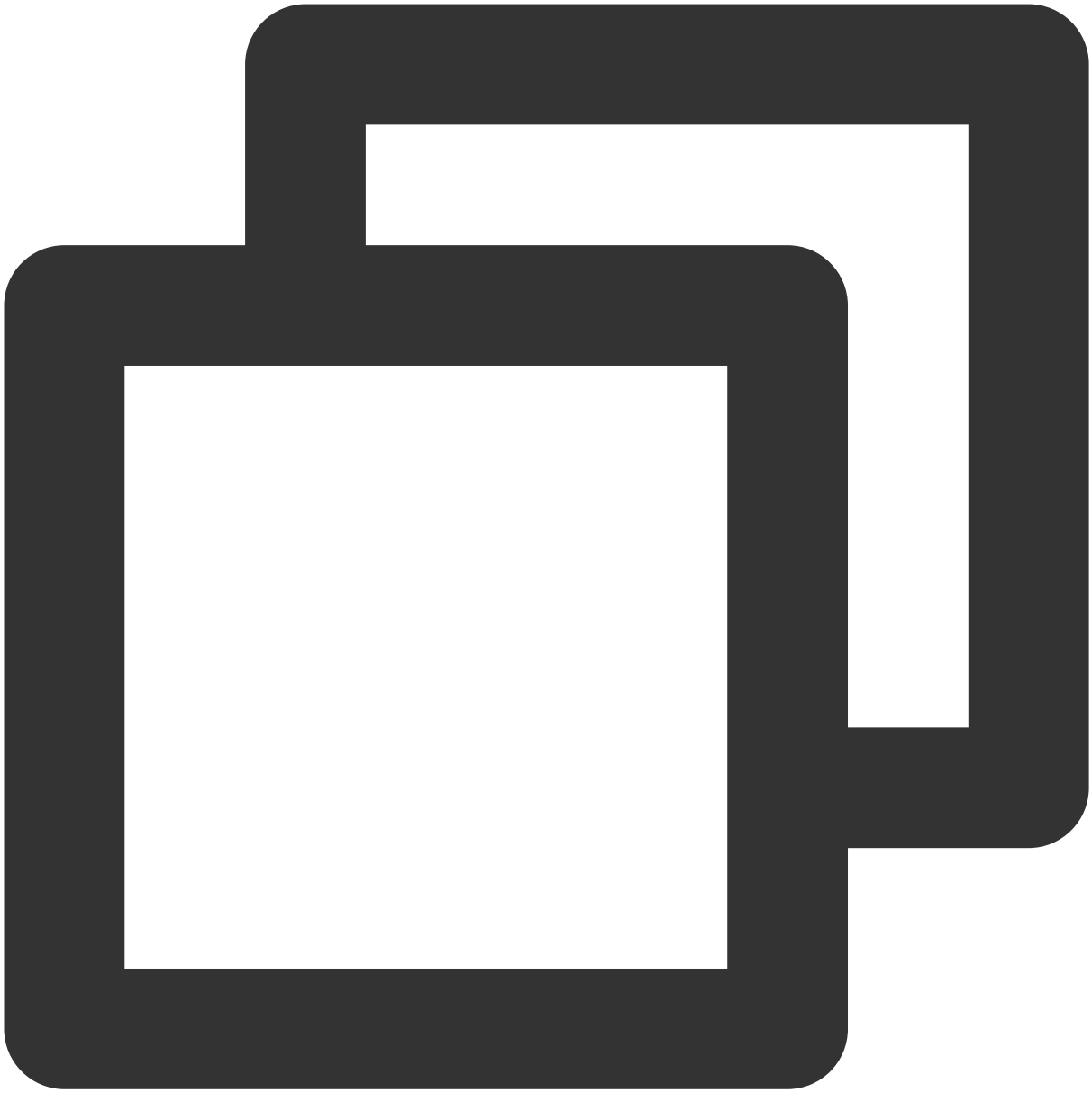
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：简单案例

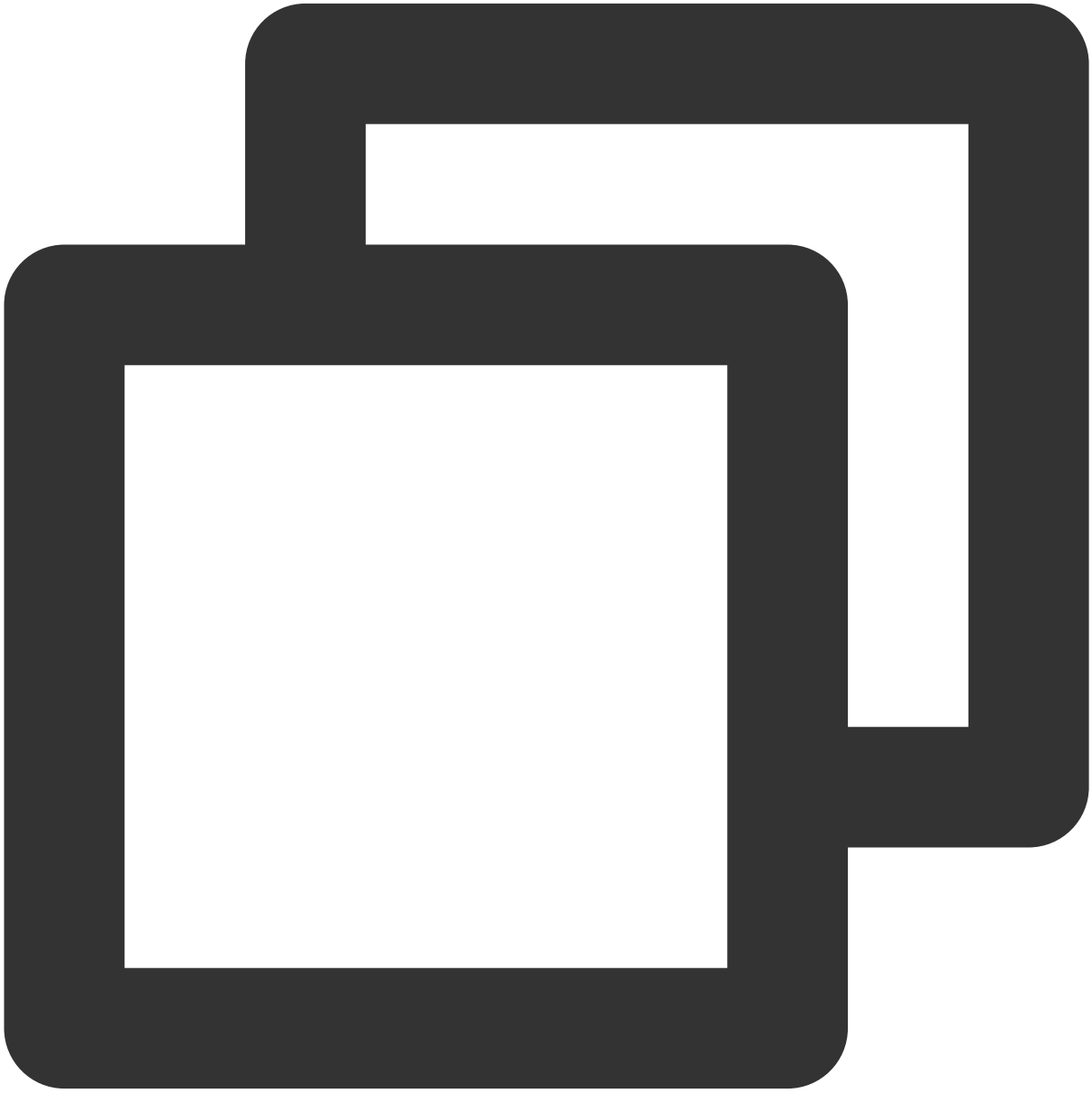
请求



```
POST /?delete HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 20 Aug 2019 11:59:35 GMT
Content-Type: application/xml
Content-Length: 158
Content-MD5: zUd/xgzNGDrqJMJUOWV2AQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<Delete>
    <Quiet>false</Quiet>
```

```
    <Object>
        <Key>example-object-1.jpg</Key>
    </Object>
    <Object>
        <Key>example-object-2.jpg</Key>
    </Object>
</Delete>
```

响应

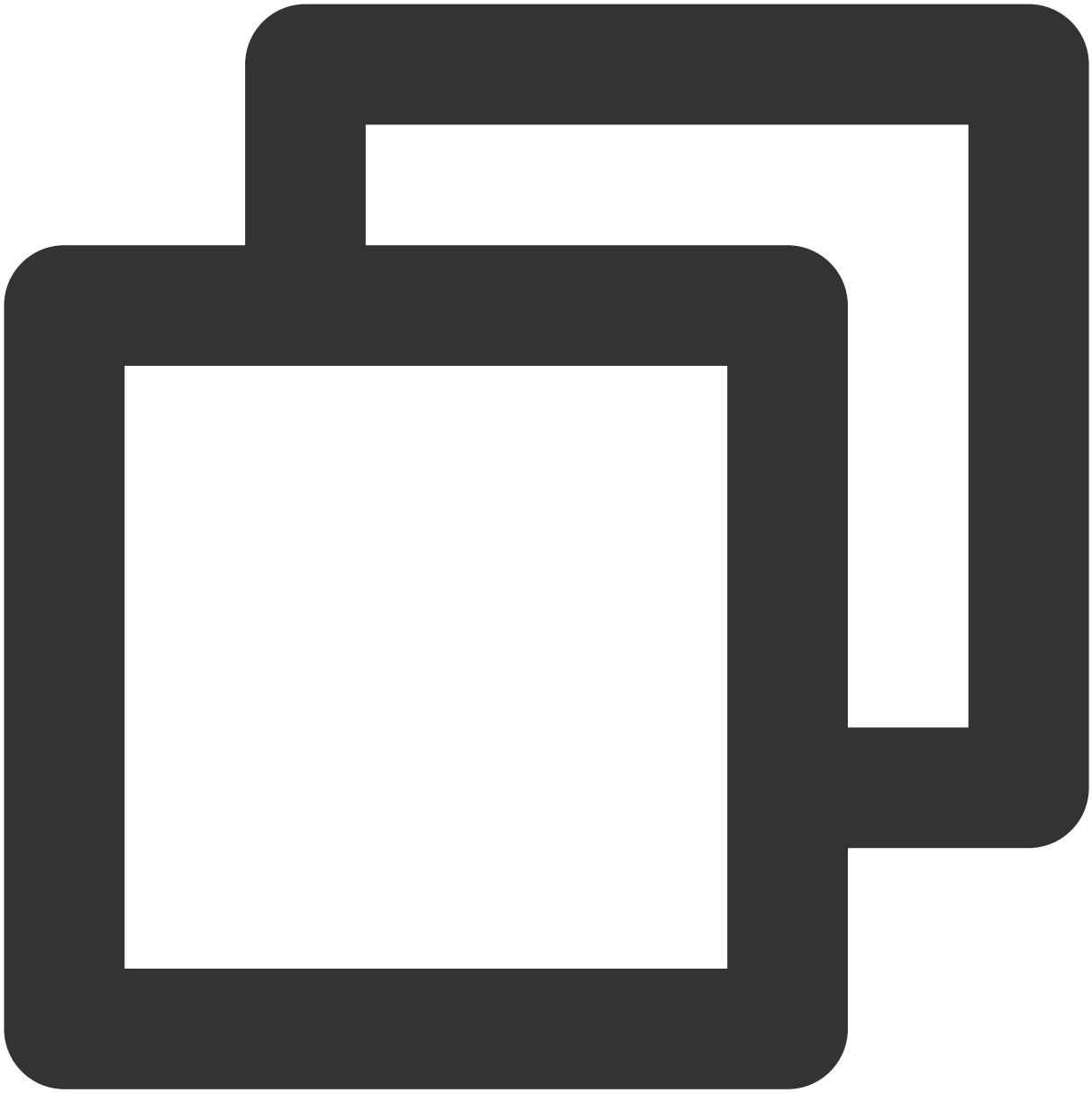


```
HTTP/1.1 200 OK
```

```
Content-Type: application/xml
Content-Length: 144
Connection: close
Date: Tue, 20 Aug 2019 11:59:35 GMT
Server: tencent-cos
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****

<DeleteResult>
    <Deleted>
        <Key>example-object-1.jpg</Key>
    </Deleted>
    <Deleted>
        <Key>example-object-2.jpg</Key>
    </Deleted>
</DeleteResult>
```

案例二：简单案例（Quiet 模式）

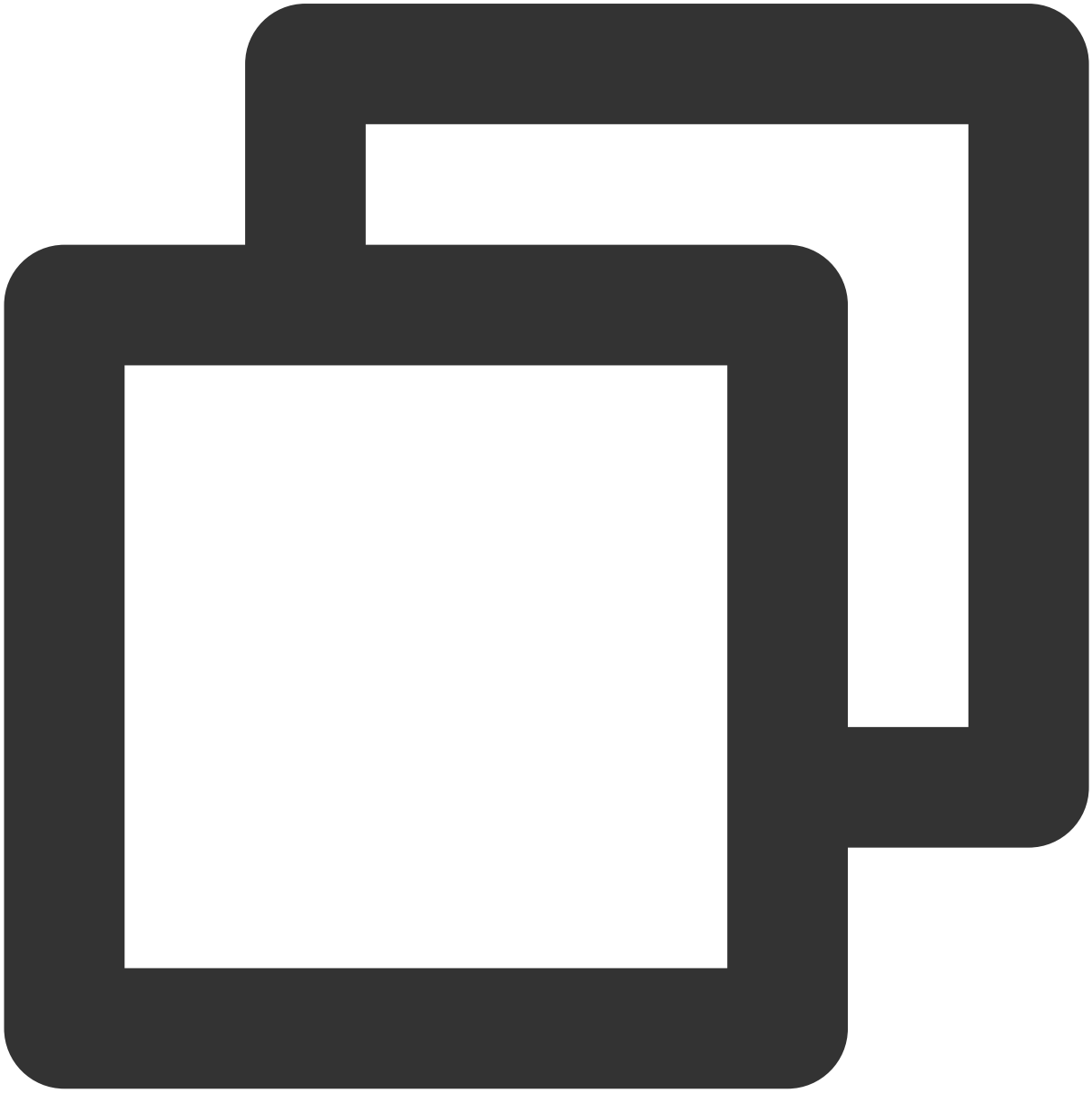
请求



```
POST /?delete HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 20 Aug 2019 12:12:26 GMT
Content-Type: application/xml
Content-Length: 157
Content-MD5: +iI9kJvM2k/y5y3nHcn8BQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<Delete>
    <Quiet>true</Quiet>
```

```
    <Object>
        <Key>example-object-1.jpg</Key>
    </Object>
    <Object>
        <Key>example-object-2.jpg</Key>
    </Object>
</Delete>
```

响应

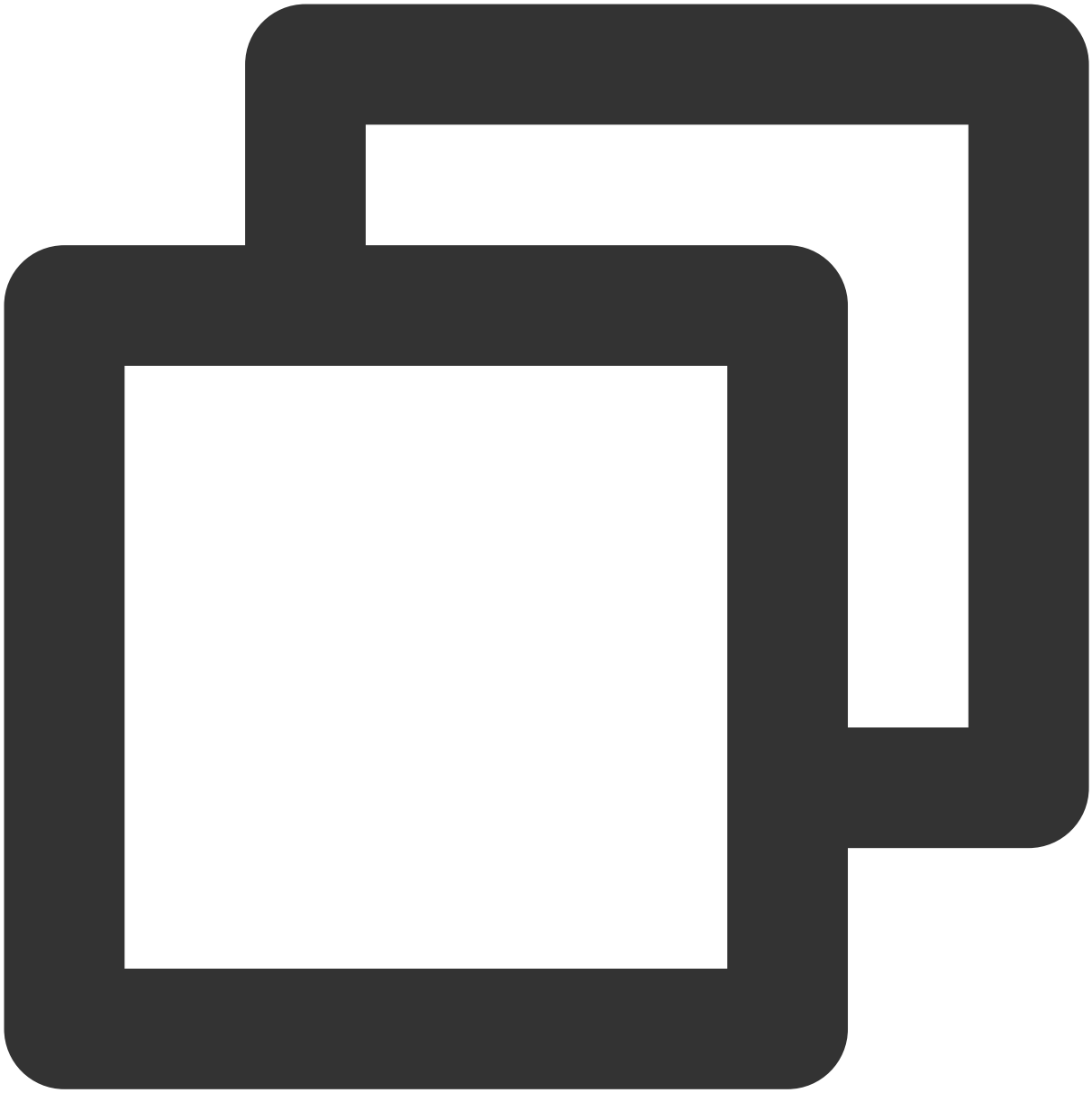


```
HTTP/1.1 200 OK
```

```
Content-Type: application/xml
Content-Length: 15
Connection: close
Date: Tue, 20 Aug 2019 12:12:27 GMT
Server: tencent-cos
x-cos-request-id: NWQ1YmUzYWFfMTljMDJhMDlfNTg3ZV8zNDI0****

<DeleteResult/>
```

案例三：启用版本控制（创建删除标记）

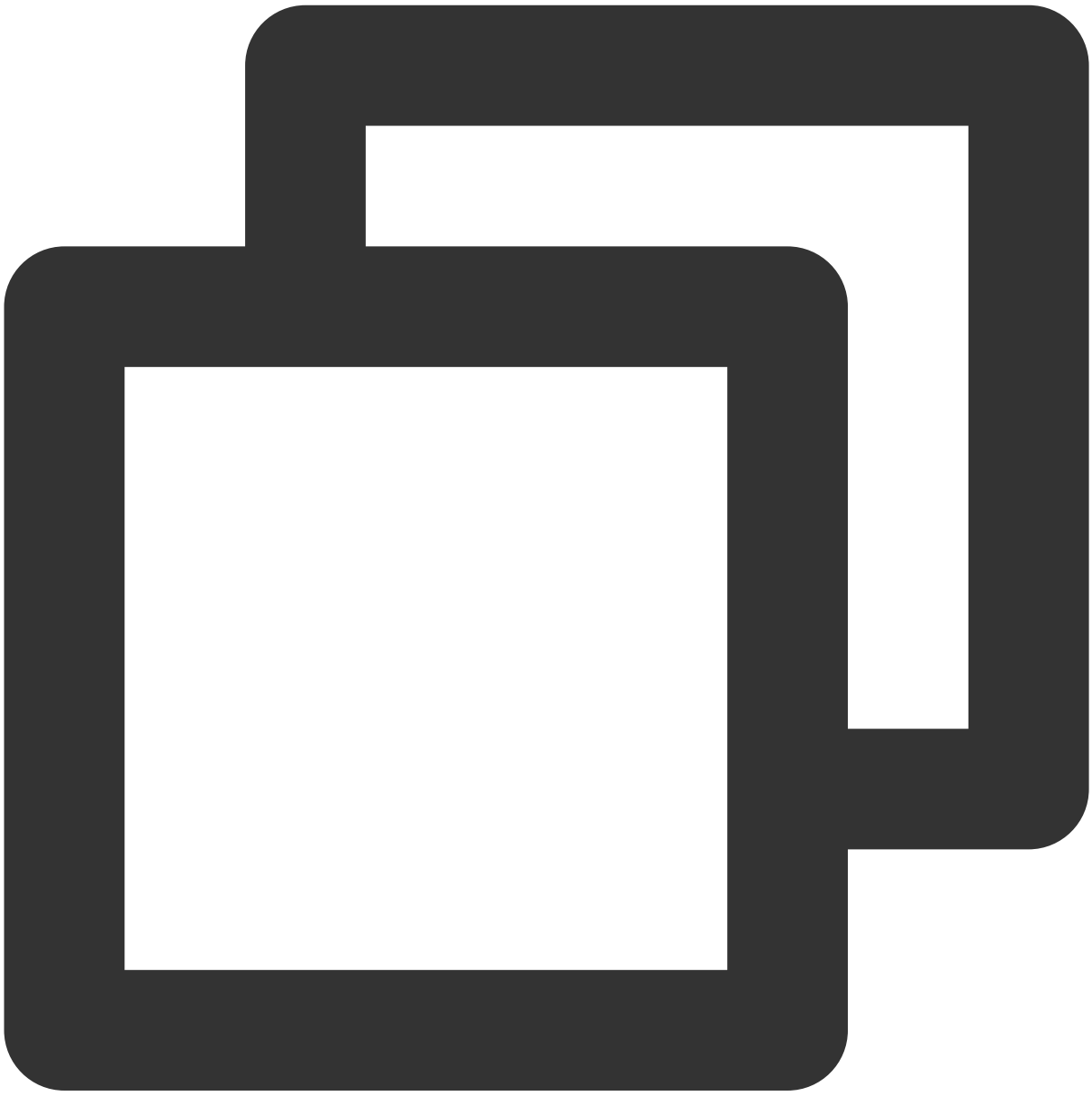
请求



```
POST /?delete HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 21 Aug 2019 12:04:03 GMT
Content-Type: application/xml
Content-Length: 100
Content-MD5: MowFtlG7iwK7Wmk79IVXFA==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<Delete>
    <Quiet>false</Quiet>
```

```
    <Object>
        <Key>example-object-1.jpg</Key>
    </Object>
</Delete>
```

响应

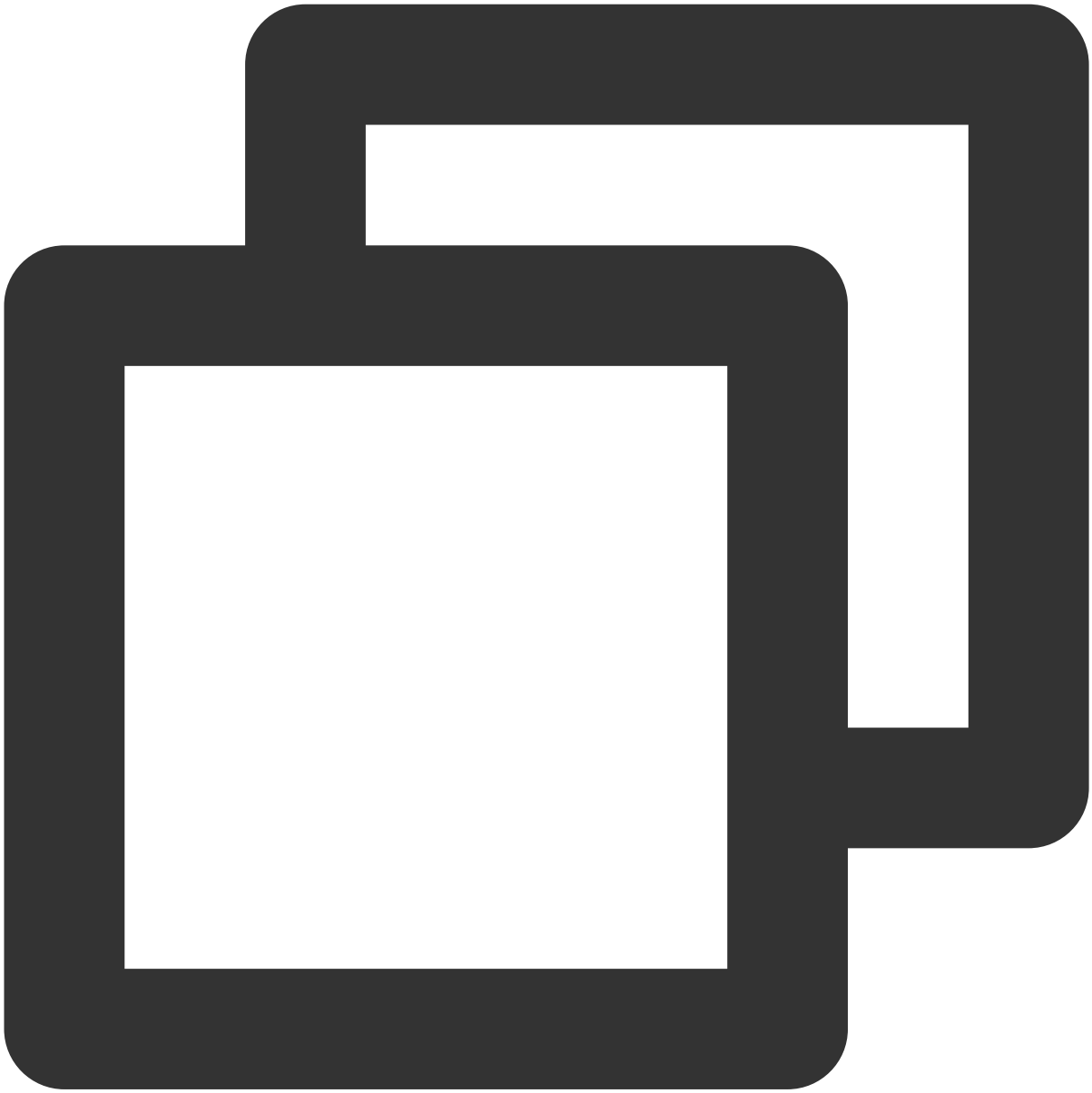


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 200
Connection: close
```

```
Date: Wed, 21 Aug 2019 12:04:03 GMT
Server: tencent-cos
x-cos-request-id: NWQ1ZDMzMzNfNDhiNDBiMDlfMmIzNzZfMTBh****

<DeleteResult>
    <Deleted>
        <Key>example-object-1.jpg</Key>
        <DeleteMarker>true</DeleteMarker>
        <DeleteMarkerVersionId>MTg0NDUxNzc2ODQ2NjU3ODM4NTc</DeleteMarkerVersionId>
    </Deleted>
</DeleteResult>
```

案例四：删除指定版本

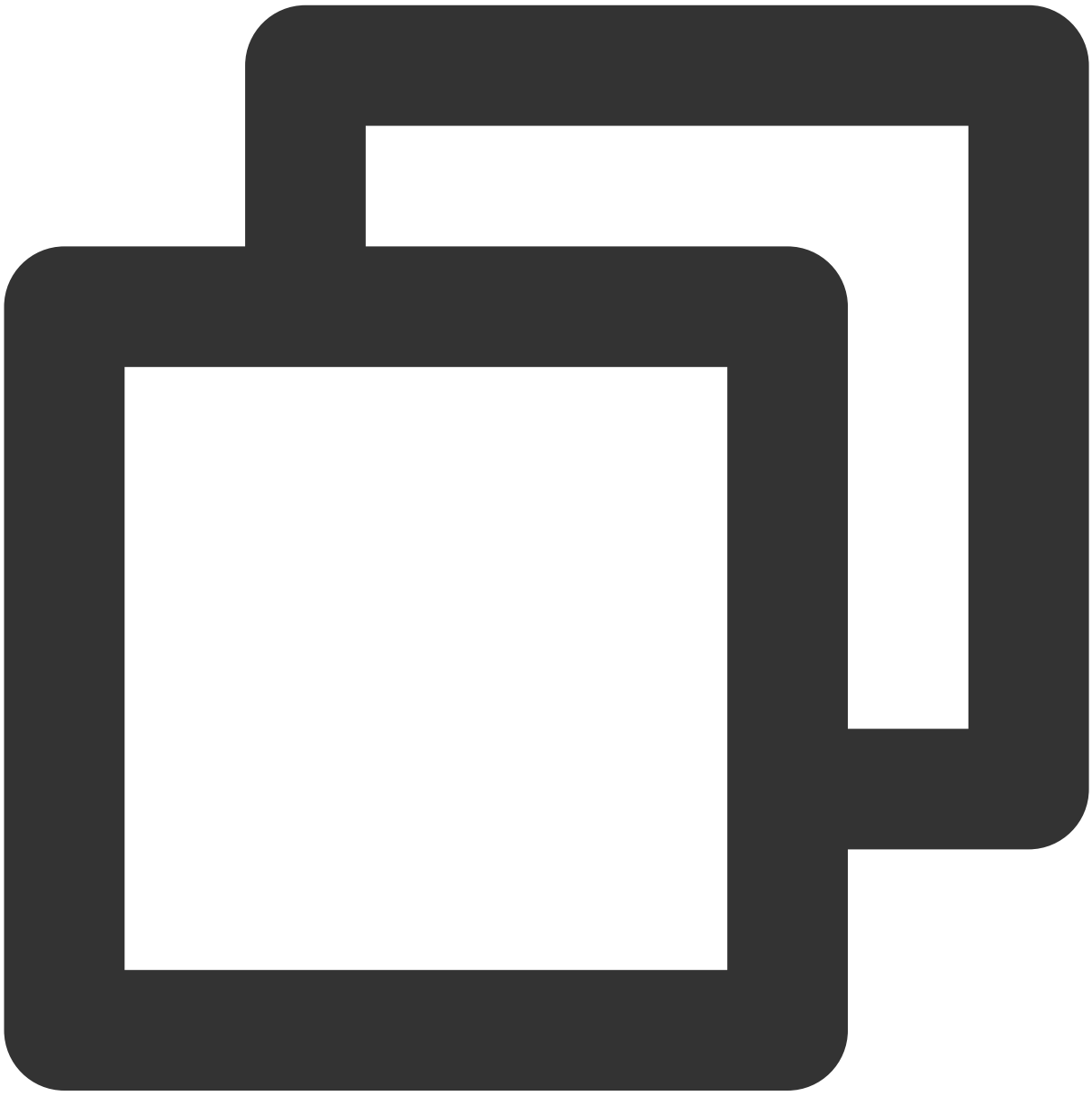
请求



```
POST /?delete HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 21 Aug 2019 11:24:43 GMT
Content-Type: application/xml
Content-Length: 154
Content-MD5: EwFydeQSMzaHWi0qMTOGWw==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<Delete>
    <Quiet>false</Quiet>
```

```
    <Object>
        <Key>example-object-2.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODcwMjYyNjIwMTM</VersionId>
    </Object>
</Delete>
```

响应

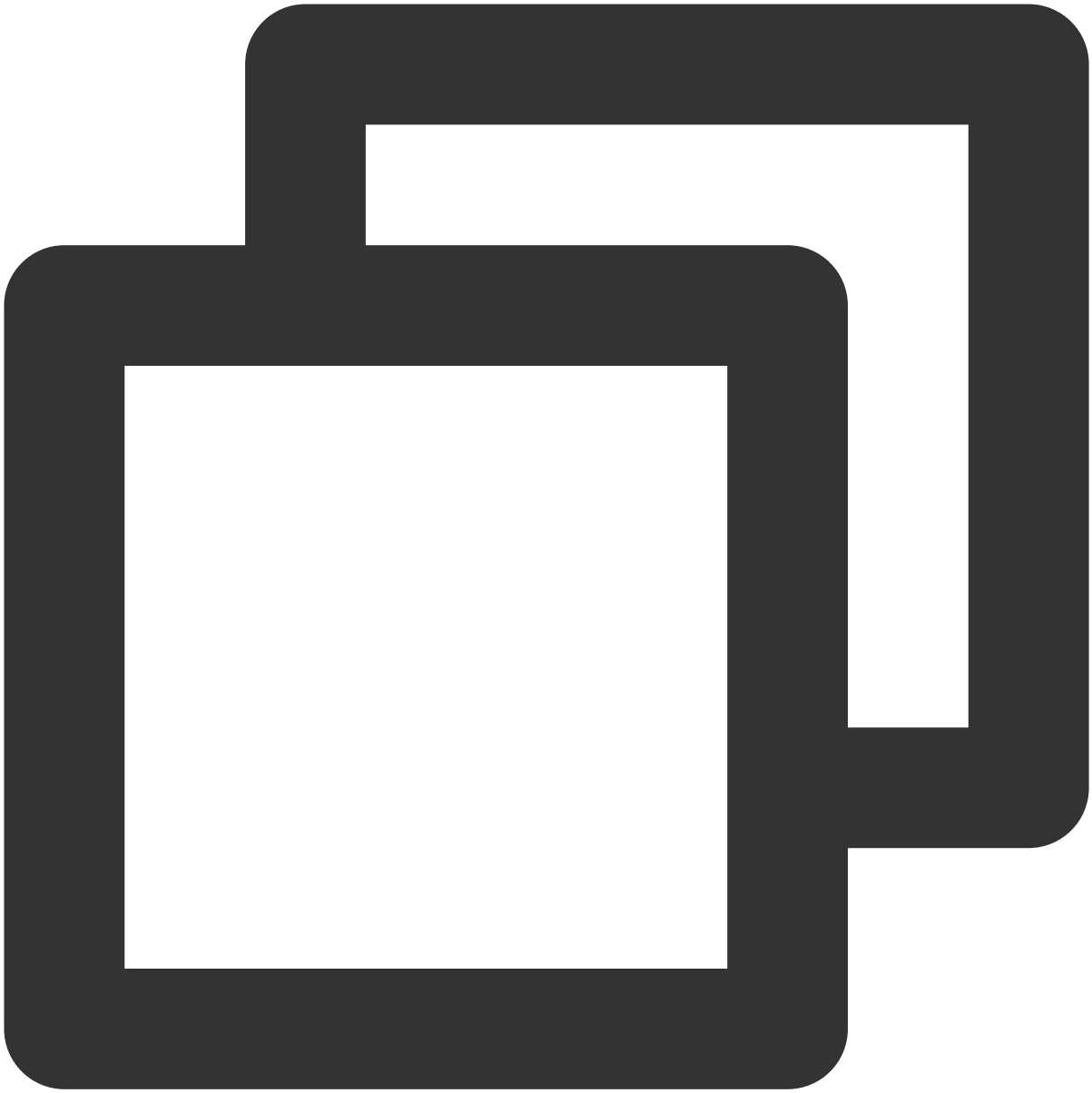


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 140
```

```
Connection: close
Date: Wed, 21 Aug 2019 11:24:44 GMT
Server: tencent-cos
x-cos-request-id: NWQ1ZDI5ZmJfNDhiNDBiMDlfMmIzODNfMTA0****

<DeleteResult>
    <Deleted>
        <Key>example-object-2.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODcwMjYyNjIwMTM</VersionId>
    </Deleted>
</DeleteResult>
```

案例五：删除指定删除标记

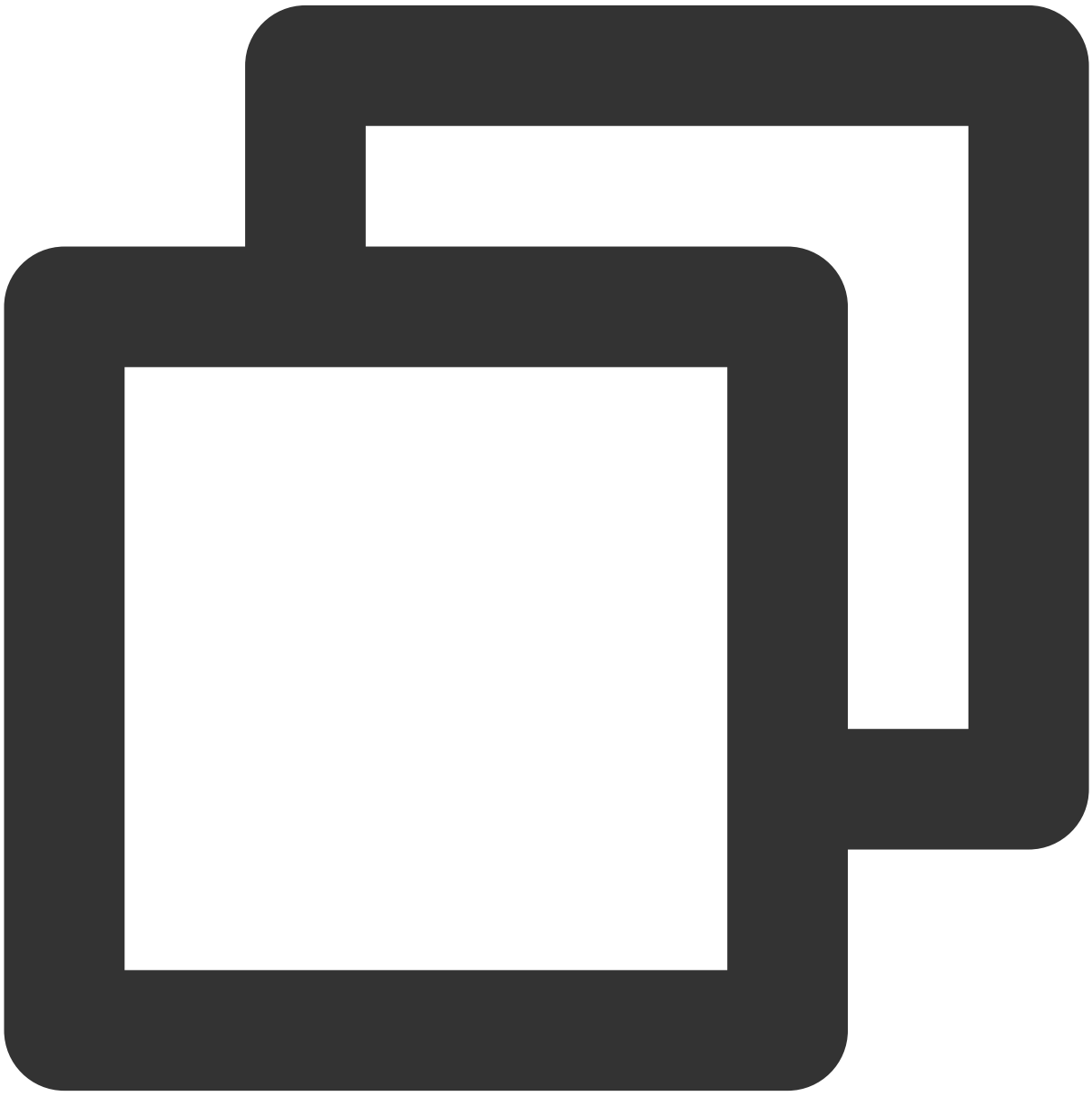
请求



```
POST /?delete HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 21 Aug 2019 12:04:04 GMT
Content-Type: application/xml
Content-Length: 154
Content-MD5: EKphCPpHcKiVqJtMqE+DmA==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<Delete>
    <Quiet>false</Quiet>
```

```
    <Object>
        <Key>example-object-1.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODQ2NjU3ODM4NTc</VersionId>
    </Object>
</Delete>
```

响应

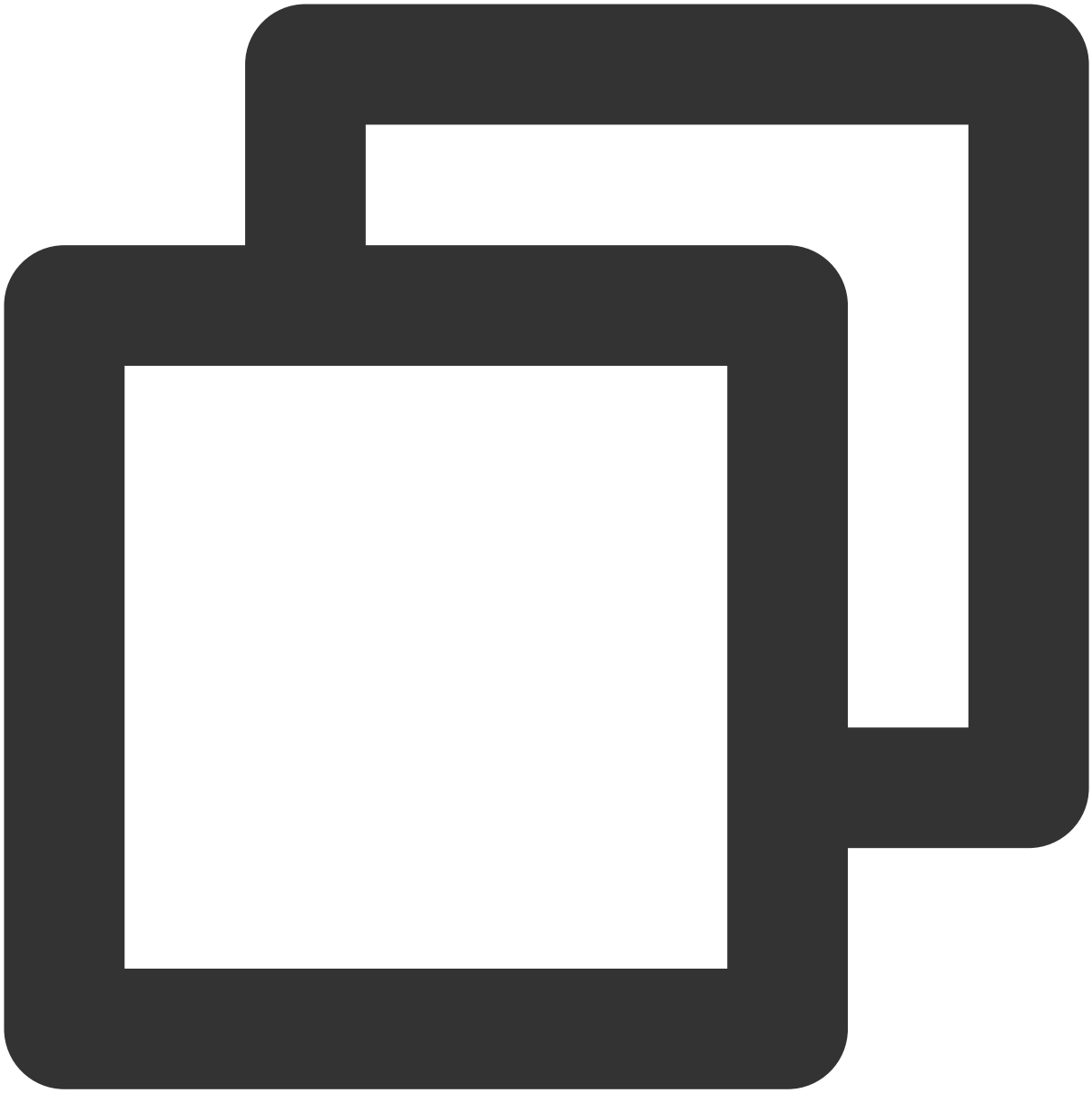


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 253
```

```
Connection: close
Date: Wed, 21 Aug 2019 12:04:04 GMT
Server: tencent-cos
x-cos-request-id: NWQ1ZDMzMzRfYmIwMmEwOV83YTQzXzEyM2Ri****

<DeleteResult>
    <Deleted>
        <Key>example-object-1.jpg</Key>
        <DeleteMarker>true</DeleteMarker>
        <DeleteMarkerVersionId>MTg0NDUxNzc2ODQ2NjU3ODM4NTc</DeleteMarkerVersionId>
        <VersionId>MTg0NDUxNzc2ODQ2NjU3ODM4NTc</VersionId>
    </Deleted>
</DeleteResult>
```

案例六：部分删除失败

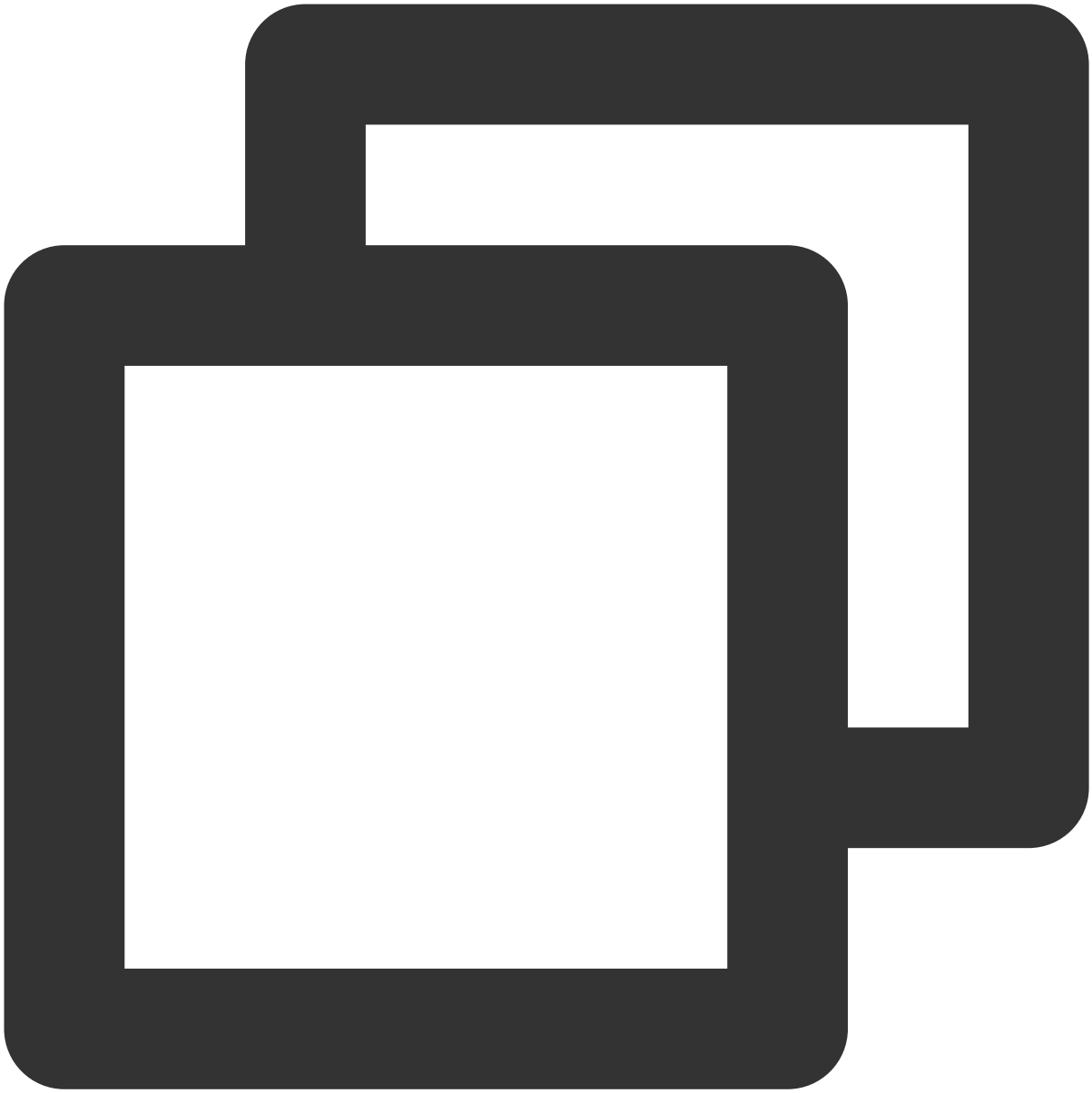
请求



```
POST /?delete HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 21 Aug 2019 12:04:05 GMT
Content-Type: application/xml
Content-Length: 436
Content-MD5: ZAbgvje31aO+0j7pkEkYvQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<Delete>
    <Quiet>false</Quiet>
```

```
    <Object>
        <Key>example-object-1.jpg</Key>
    </Object>
    <Object>
        <Key>example-object-2.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODQ2NjQ1MjM5MTk</VersionId>
    </Object>
    <Object>
        <Key>example-object-3.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODQ2NjQwMTIwMDI</VersionId>
    </Object>
    <Object>
        <Key>example-object-4.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODQ2NjQ0NjI0MDQ</VersionId>
    </Object>
</Delete>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 703
Connection: close
Date: Wed, 21 Aug 2019 12:04:06 GMT
Server: tencent-cos
x-cos-request-id: NWQ1ZDMzMzVfOTNjMjJhMDlfMzhiM18xMWY3****

<DeleteResult>
    <Deleted>
        <Key>example-object-1.jpg</Key>
```

```
        <DeleteMarker>true</DeleteMarker>
        <DeleteMarkerVersionId>MTg0NDUxNzc2ODQ2NjM1NTI2NDY</DeleteMarkerVersionId>
    </Deleted>
    <Deleted>
        <Key>example-object-2.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODQ2NjQ1MjM5MTk</VersionId>
    </Deleted>
    <Deleted>
        <Key>example-object-3.jpg</Key>
        <DeleteMarker>true</DeleteMarker>
        <DeleteMarkerVersionId>MTg0NDUxNzc2ODQ2NjQwMTIwMDI</DeleteMarkerVersionId>
        <VersionId>MTg0NDUxNzc2ODQ2NjQwMTIwMDI</VersionId>
    </Deleted>
    <Error>
        <Key>example-object-4.jpg</Key>
        <VersionId>MTg0NDUxNzc2ODQ2NjQ0NjI0MDQ</VersionId>
        <Code>PathConflict</Code>
        <Message>Path conflict.</Message>
    </Error>
</DeleteResult>
```

OPTIONS Object

最近更新时间：2023-12-21 14:33:05

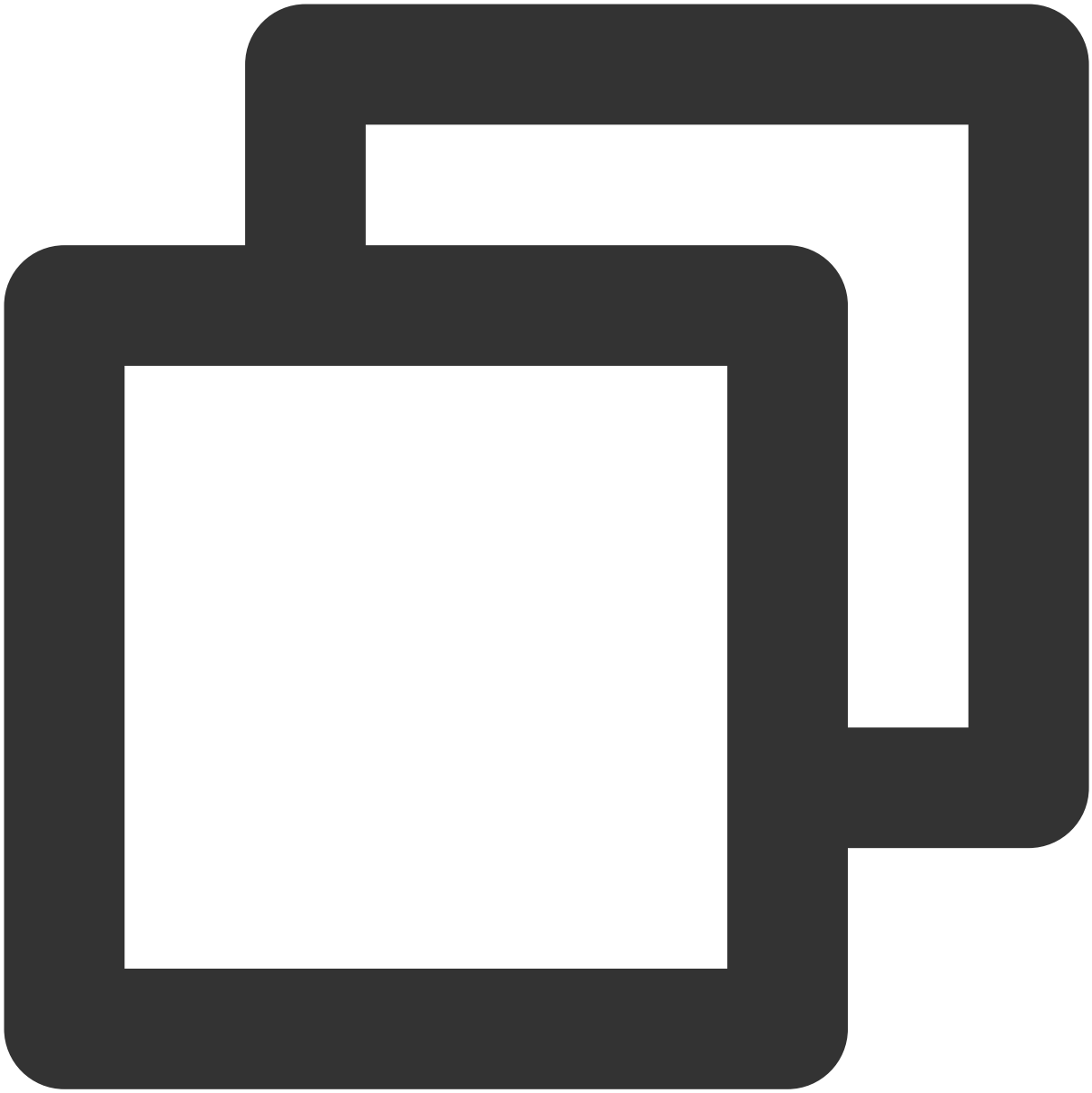
功能描述

OPTIONS Object 用于跨域资源共享（CORS）的预检（Preflight）请求。当浏览器发起 CORS 请求时，浏览器会判断是否有必要发起预检请求，如有必要则浏览器会在发起 CORS 请求前自动发出预检请求，所以在正常情况下，前端开发者不需要自己去发起这样的请求。
如果指定存储桶存在 CORS 配置且预检条件符合存储桶的 CORS 配置，则 COS 正常返回，允许浏览器继续 CORS 请求。
如果指定存储桶不存在 CORS 配置或预检条件不符合存储桶的 CORS 配置，则 COS 返回 HTTP 403 Forbidden，此时浏览器将停止 CORS 请求并向前端抛出异常。
有关存储桶的 CORS 配置，可参见 PUT Bucket cors 文档。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
OPTIONS /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Origin: Origin
Access-Control-Request-Method: RequestMethod
```

注意

虽然上述示例指定了对象键（ObjectKey），但在实际使用过程中可针对存储桶域名下的任意资源（包含根目录）发起预检请求，例如 `OPTIONS / HTTP/1.1` 或者 `OPTIONS /?lifecycle HTTP/1.1` 等。

预检请求由浏览器自动发出，因此预检请求无需也无法携带 Authorization 请求签名。

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

请求参数

此接口的请求参数由浏览器自动根据跨域访问的目标资源决定，开发者无需关注。

请求头

此接口的请求头由浏览器自动根据跨域访问的具体行为决定，开发者无需关注。

名称	描述	类型	是否必选
Origin	发起 CORS 请求的域名。	string	是
Access-Control-Request-Method	发起 CORS 请求所用的方法（Method）	string	是
Access-Control-Request-Headers	发起 CORS 请求时使用的 HTTP 请求头部，不区分英文大小写，可使用英文逗号(,)分隔多个头部	string	否

请求体

此接口无请求体。

响应

响应头

此接口的响应头由浏览器自动识别处理并控制是否允许 CORS 请求，开发者无需关注。

名称	描述	类型
Access-Control-Allow-Origin	允许发起 CORS 的域名，可能的值有以下两种： `*` ：代表允许所有域名 请求头 Origin 中指定的域名：代表允许指定域名	string
Access-Control-Allow-Methods	允许发起 CORS 请求所使用的方法（Method），可使用英文逗号(,)分隔多个方法	string
Access-Control-Expose-Headers	允许浏览器获取的 CORS 请求中的 HTTP 响应头部，不区分英文大小写，可使用英文逗号(,)分隔多个头部	string
Access-Control-Max-Age	CORS 配置的有效时间，单位为秒，在有效时间内，浏览器无须为同一请求再次发起预检请求	integer

版权所有：腾讯云计算（北京）有限责任公司

响应体

此接口响应体为空。

错误码

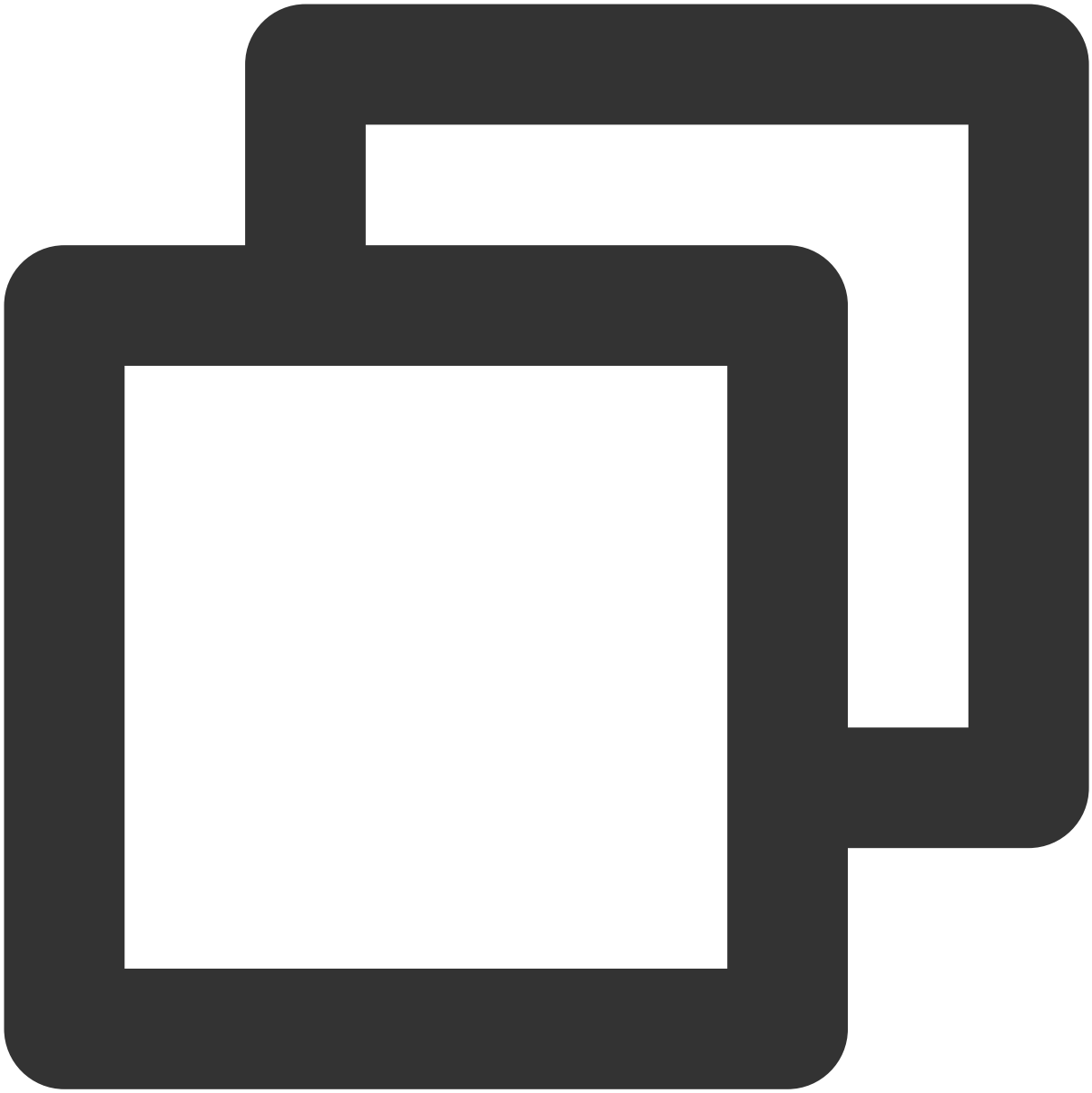
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

以下案例均由浏览器根据发起的 CORS 请求自动发起的预检请求，开发者无需关注。

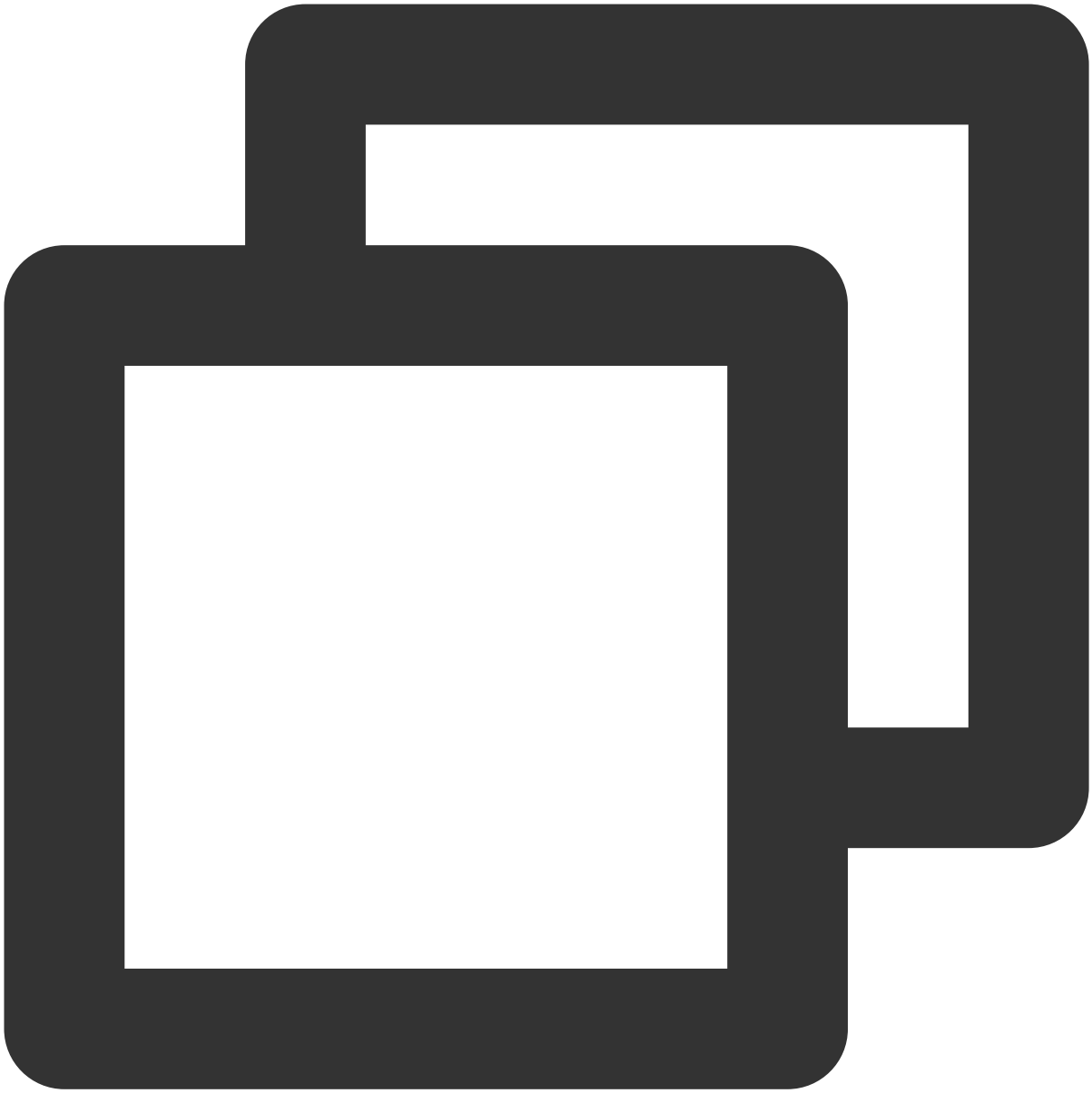
案例一：针对 PUT Object 发起预检请求

请求



```
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 14:49:22 GMT
Origin: https://example.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: content-md5,content-type,x-cos-meta-author
Connection: close
```

响应



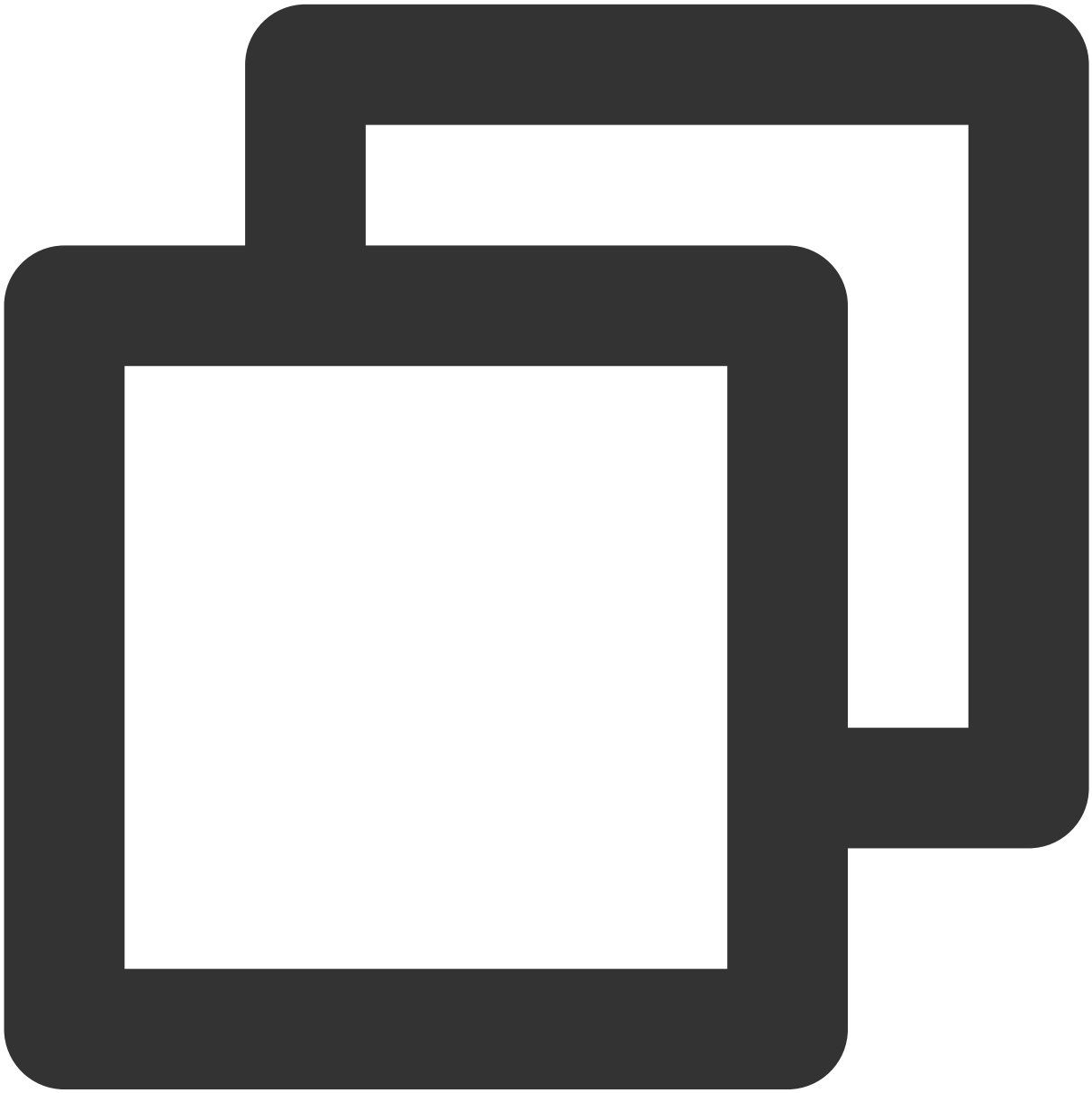
```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: content-md5,content-type,x-cos-meta-author
Access-Control-Allow-Methods: PUT,GET,POST,DELETE,HEAD
Access-Control-Allow-Origin: https://example.com
Access-Control-Expose-Headers: Content-Length,ETag,x-cos-meta-author
Access-Control-Max-Age: 600
Date: Thu, 09 Jul 2020 14:49:22 GMT
Server: tencent-cos
```

版权所有：腾讯云计算（北京）有限责任公司

```
x-cos-request-id: NWYwNzJlNzJfODRjOTJhMDlfMjU0MWNfMTNmZDM5****
```

案例二：针对 GET Object 时携带 Range 请求头部发起预检请求

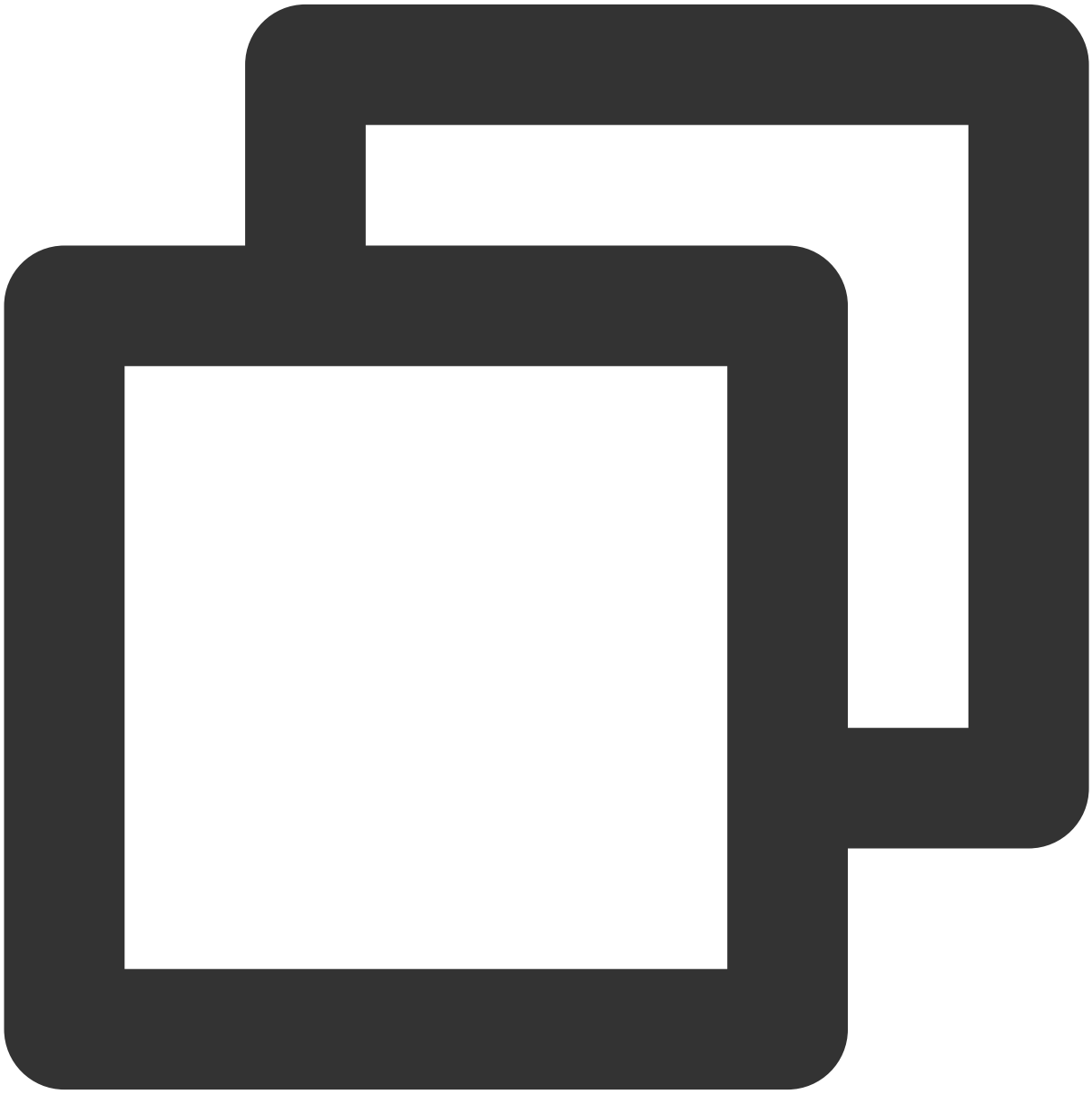
请求



```
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 14:49:22 GMT
Origin: https://example.com
Access-Control-Request-Method: GET
```

```
Access-Control-Request-Headers: range
Connection: close
```

响应

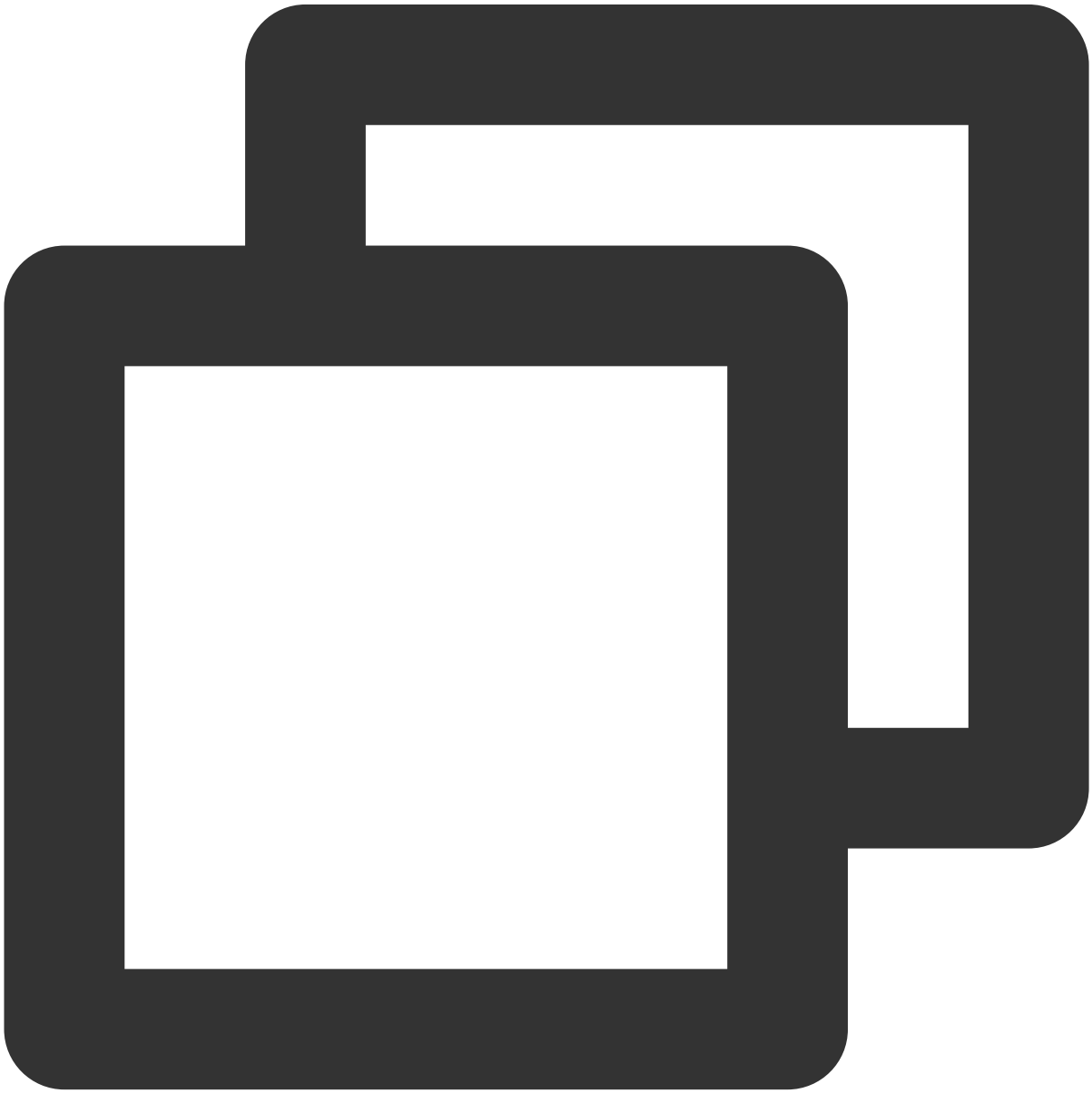


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Access-Control-Allow-Headers: range,x-cos-server-side-encryption-customer-algorithm
Access-Control-Allow-Methods: GET,HEAD
Access-Control-Allow-Origin: *
```

```
Access-Control-Expose-Headers: Content-Length,ETag,x-cos-meta-author
Access-Control-Max-Age: 600
Date: Thu, 09 Jul 2020 14:49:22 GMT
Server: tencent-cos
x-cos-request-id: NWYwNzJlNzJfZDUyNzVkNjRfYTA2Ml8yNGEz****
```

案例三：针对 PUT Bucket lifecycle 发起预检请求

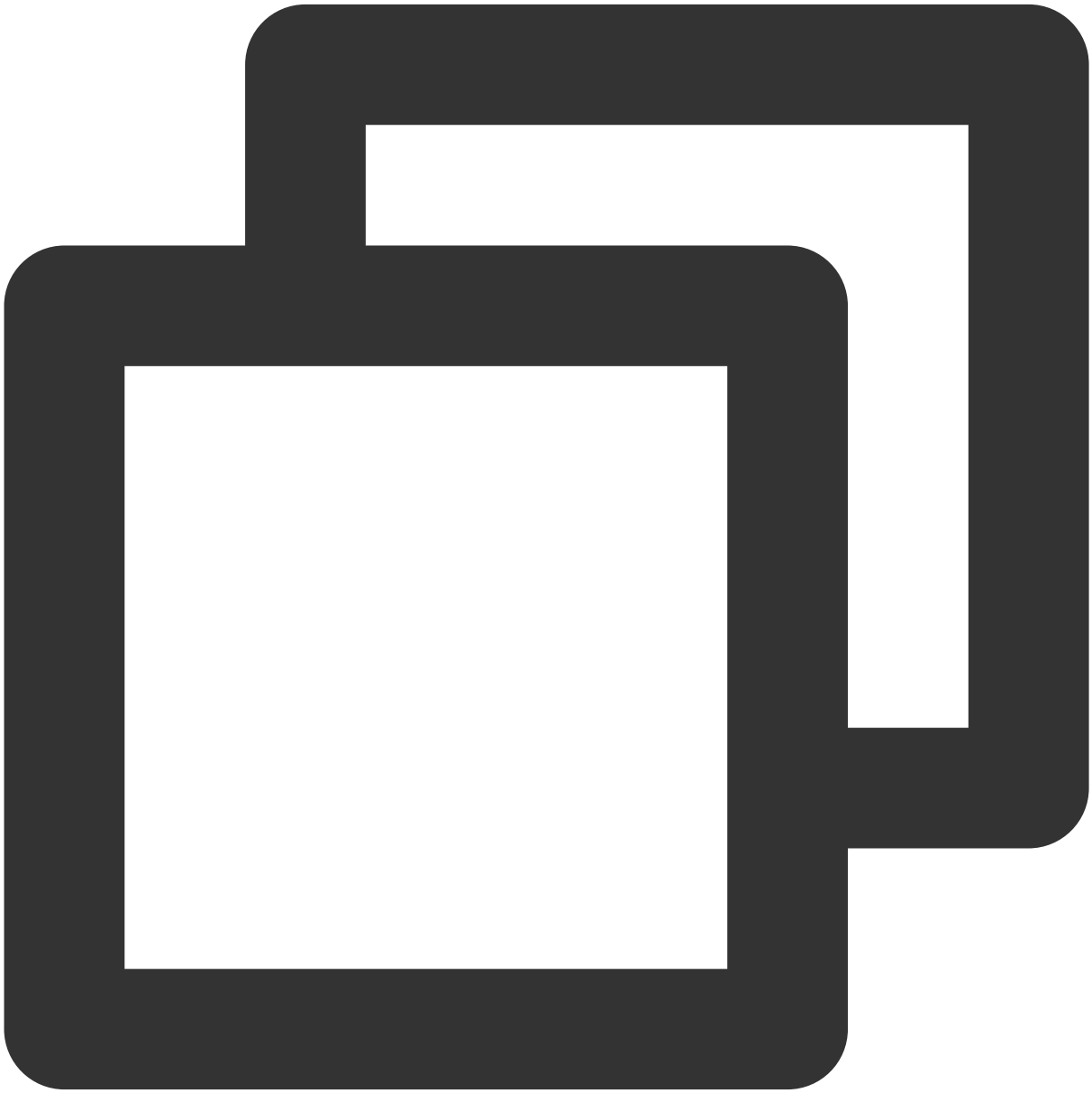
请求



```
OPTIONS /?lifecycle HTTP/1.1
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 14:29:40 GMT
Origin: https://bar.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: content-md5,content-type
Connection: close
```

响应

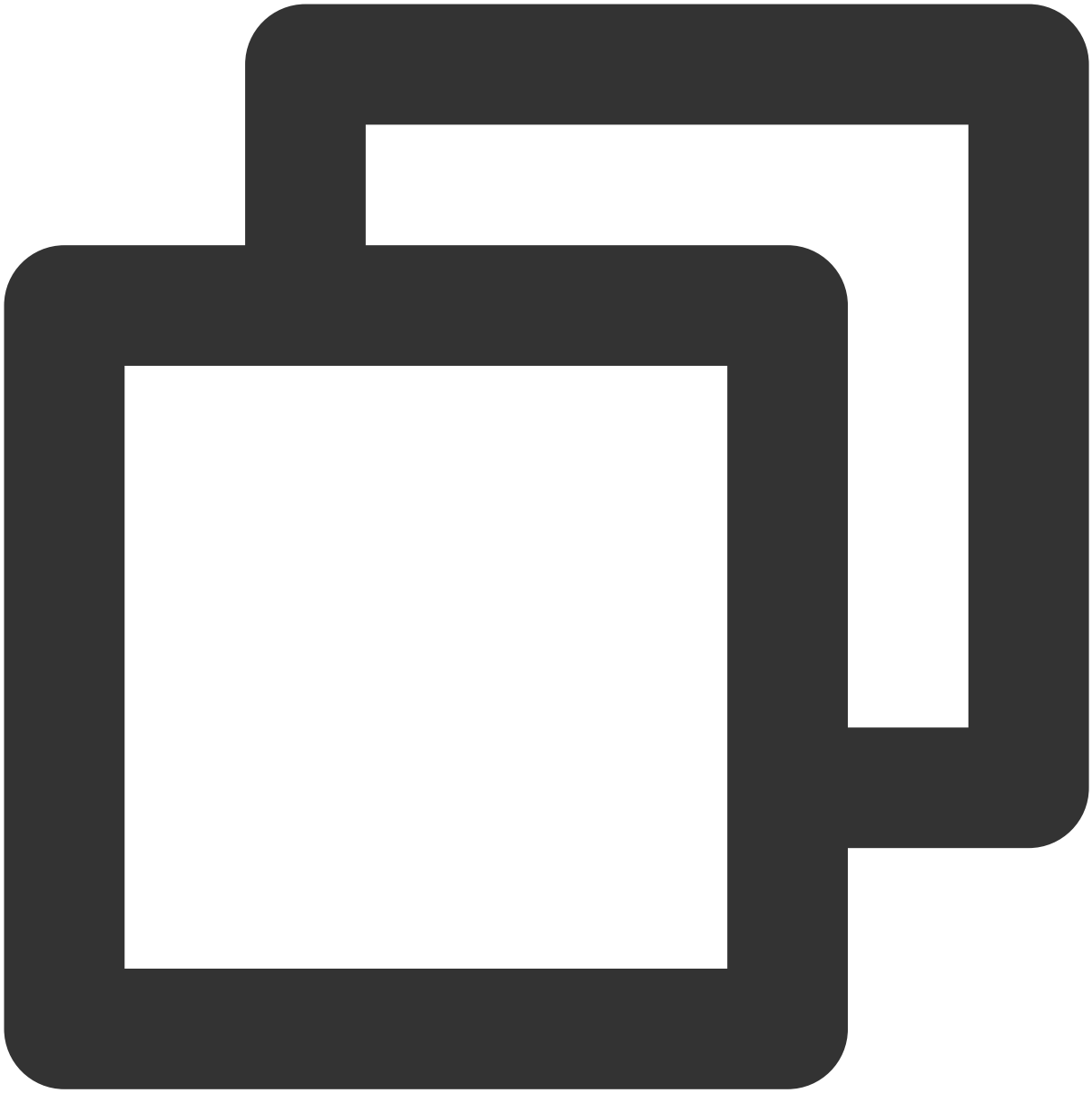


```
HTTP/1.1 200 OK
Content-Length: 0
```

```
Connection: close
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: content-md5,content-type
Access-Control-Allow-Methods: PUT,GET,POST,DELETE,HEAD
Access-Control-Allow-Origin: https://bar.com
Access-Control-Expose-Headers: Content-Length,ETag,x-cos-meta-author
Access-Control-Max-Age: 600
Date: Thu, 09 Jul 2020 14:29:40 GMT
Server: tencent-cos
x-cos-request-id: NWYwNzI5ZDRfNjFiMDJhMDlfYzk2NF8xYmZl****
```

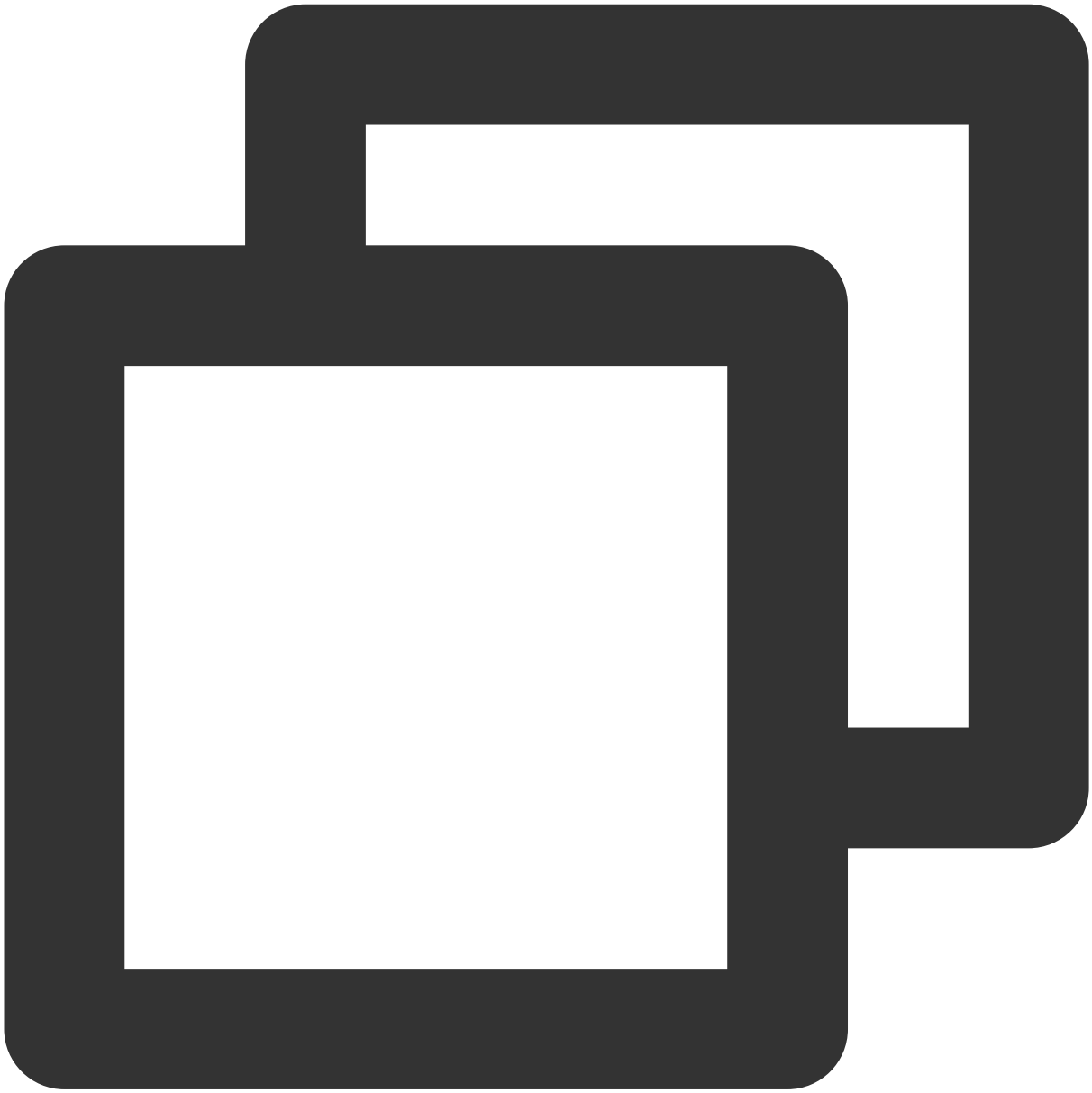
案例四：指定存储桶不存在 CORS 配置或预检条件不符合存储桶的 CORS 配置

请求



```
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 09 Jul 2020 11:45:26 GMT
Origin: https://example.com
Access-Control-Request-Method: PUT
Connection: close
```

响应



```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: 687
Connection: close
Date: Thu, 09 Jul 2020 11:45:26 GMT
Server: tencent-cos
x-cos-request-id: NWYwNzAzNTZfNzNjODJhMDlfMzRiM2ZfMThjMjk4****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODc0OWRkZjk0ZDM1NmI1M2E2MTR

<?xml version='1.0' encoding='utf-8' ?>
<Error>
```

```
    <Code>AccessForbidden</Code>
    <Message>CORSResponse: This CORS request is not allowed. This is usually becaus
    <Resource>examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject</R
    <RequestId>NWYwNzAzNTZfNzNjODJhMDlfMzRiM2ZfMThjMjk4****</RequestId>
    <TraceId>OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODc0OWRkZjk0ZDM1NmI1M2E2MTRlY2
</Error>
```

POST Object restore

最近更新时间：2023-12-21 14:33:28

功能描述

POST Object restore 接口请求可以对一个归档存储或深度归档存储类型的对象进行恢复（解冻）以便读取该对象内容，恢复出的可读取对象是临时的，您可以设置需要保持可读以及随后删除该临时副本的时间。您可以用 Days 参数来指定临时对象的过期时间，若超出该时间且期间您没有发起任何复制、延长等操作，该临时对象将被系统自动删除。临时对象仅为归档存储类型对象的副本，原始归档存储对象在此期间将始终存在。有关归档存储的进一步说明，请参见 存储类型 - 归档存储 文档。

说明

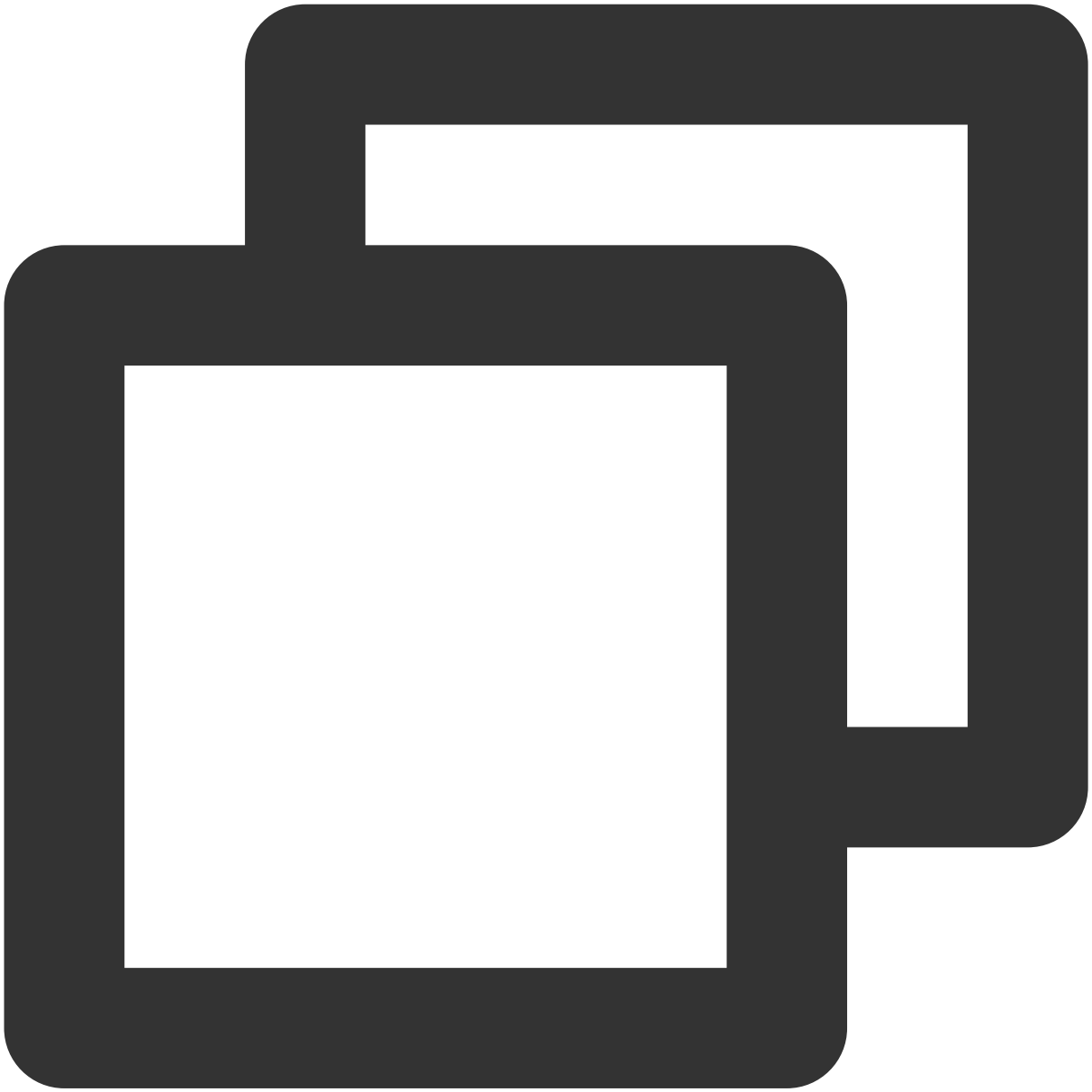
POST Object restore 接口请求存在 QPS 限制，限制为100次/秒。

版本控制

当启用版本控制时，该请求操作可以使用 versionId 请求参数指定要恢复的版本 ID，此时将恢复对象的指定版本，否则将恢复指定对象的最新版本。

请求

请求示例



```
POST /<ObjectKey>?restore HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String



[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

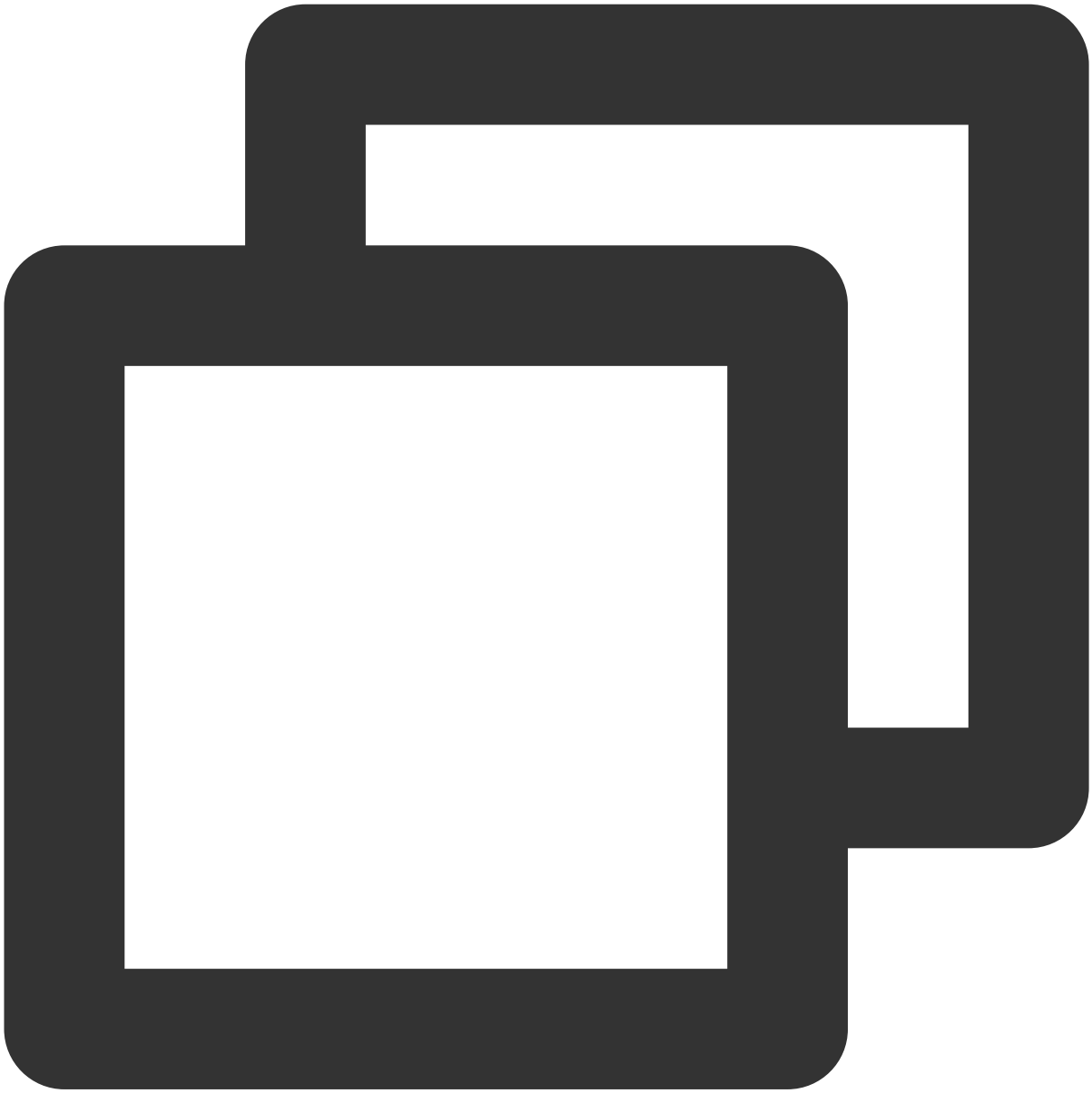
名称	描述	类型	是否必选
versionId	当启用版本控制时，指定要恢复的版本 ID，如不指定则恢复对象的最新版本	string	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

提交 **application/xml** 请求数据，包含恢复操作的具体参数。



```
<RestoreRequest>
   <Days>number</Days>
   <CASJobParameters>
       <Tier>Enum</Tier>
   </CASJobParameters>
</RestoreRequest>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选

RestoreRequest	无	包含 POST Object restore 操作的所有请求信息	Container	是

Container 节点 RestoreRequest 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Days	RestoreRequest	指定恢复出的临时副本的有效时长，单位为“天”	number	是
CASJobParameters	RestoreRequest	恢复工作参数	Container	是

Container 节点 CASJobParameters 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Tier	RestoreRequest.CASJobParameters	恢复时，Tier 可以指定为支持的恢复模式。 对于恢复归档存储类型数据，有三种恢复模式，分别为： Expedited：快速取回模式，恢复任务在1 - 5分钟内可完成。 Standard：标准取回模式，恢复任务在3 - 5小时内完成 Bulk：批量取回模式，恢复任务在5 - 12小时内完成。 对于恢复深度归档存储类型数据，有两种恢复模式，分别为： Standard：标准取回模式，恢复时间为12 - 24小时。 Bulk：批量取回模式，恢复时间为24 - 48小时。	Enum	是

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

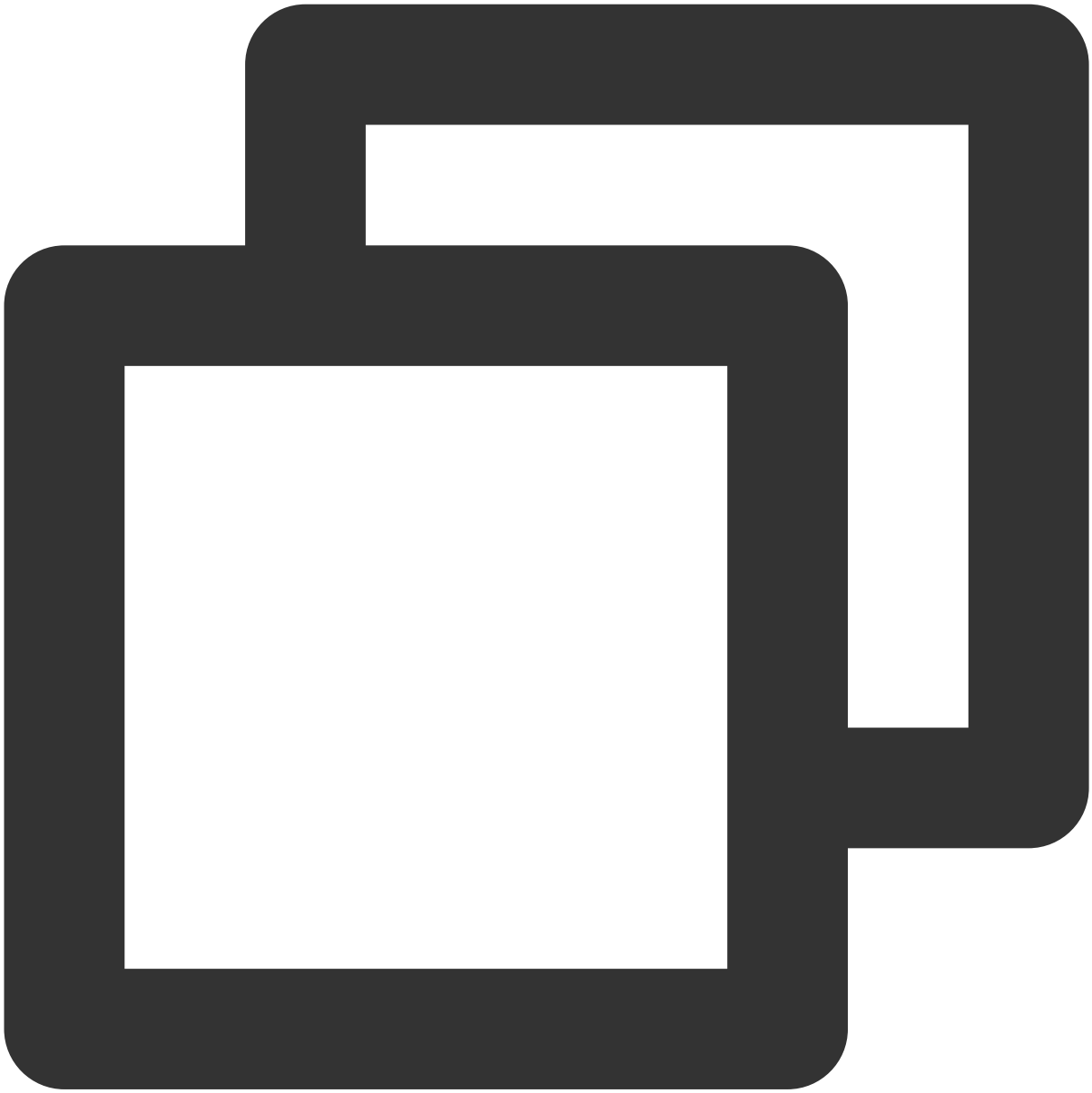
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：简单案例（未启用版本控制）

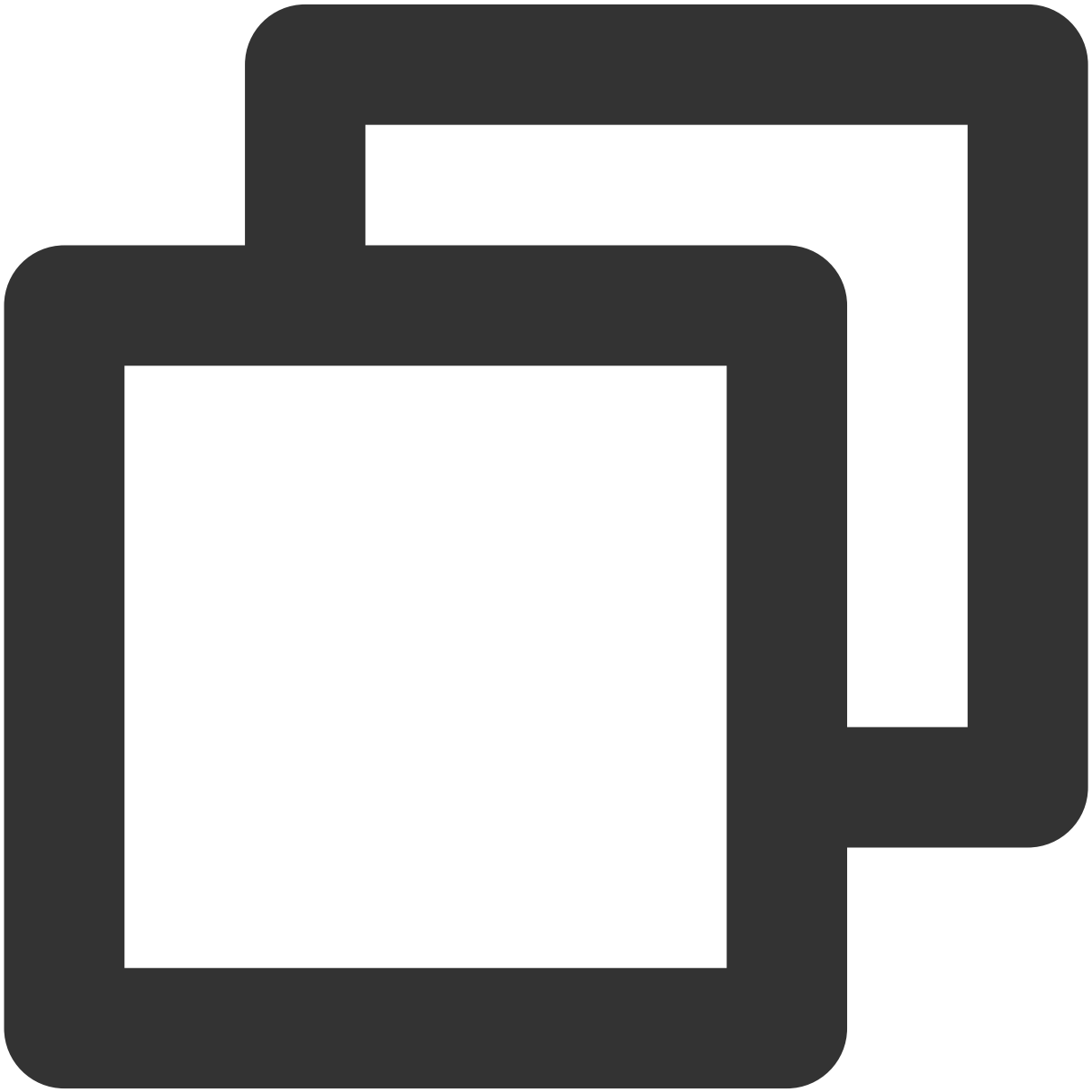
请求



```
POST /exampleobject?restore HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 27 Dec 2019 08:19:29 GMT
Content-Type: application/xml
Content-Length: 121
Content-MD5: Nr7RAnRMgrplFvD8bt5+0w==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

```
<RestoreRequest>
            <Days>1</Days>
            <CASJobParameters>
                <Tier>Expedited</Tier>
            </CASJobParameters>
</RestoreRequest>
```

响应

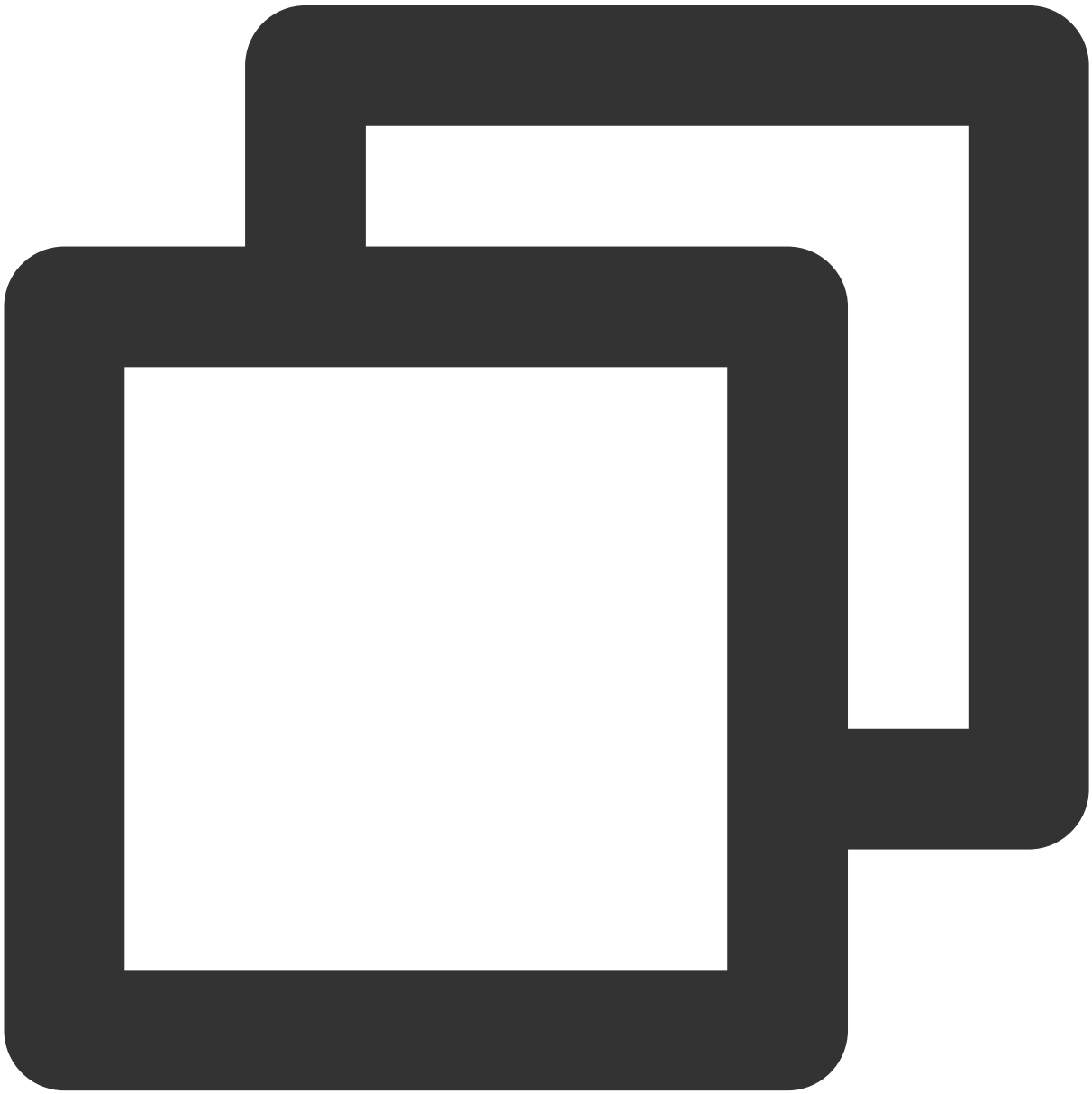


```
HTTP/1.1 202 Accepted
Content-Length: 0
```

```
Connection: close
Date: Fri, 27 Dec 2019 08:19:29 GMT
Server: tencent-cos
x-cos-request-id: NWUwNWJlOTFfMjljOTBiMDlfMTQ2MmNfNzAw****
```

案例二：恢复对象的指定版本（启用版本控制）

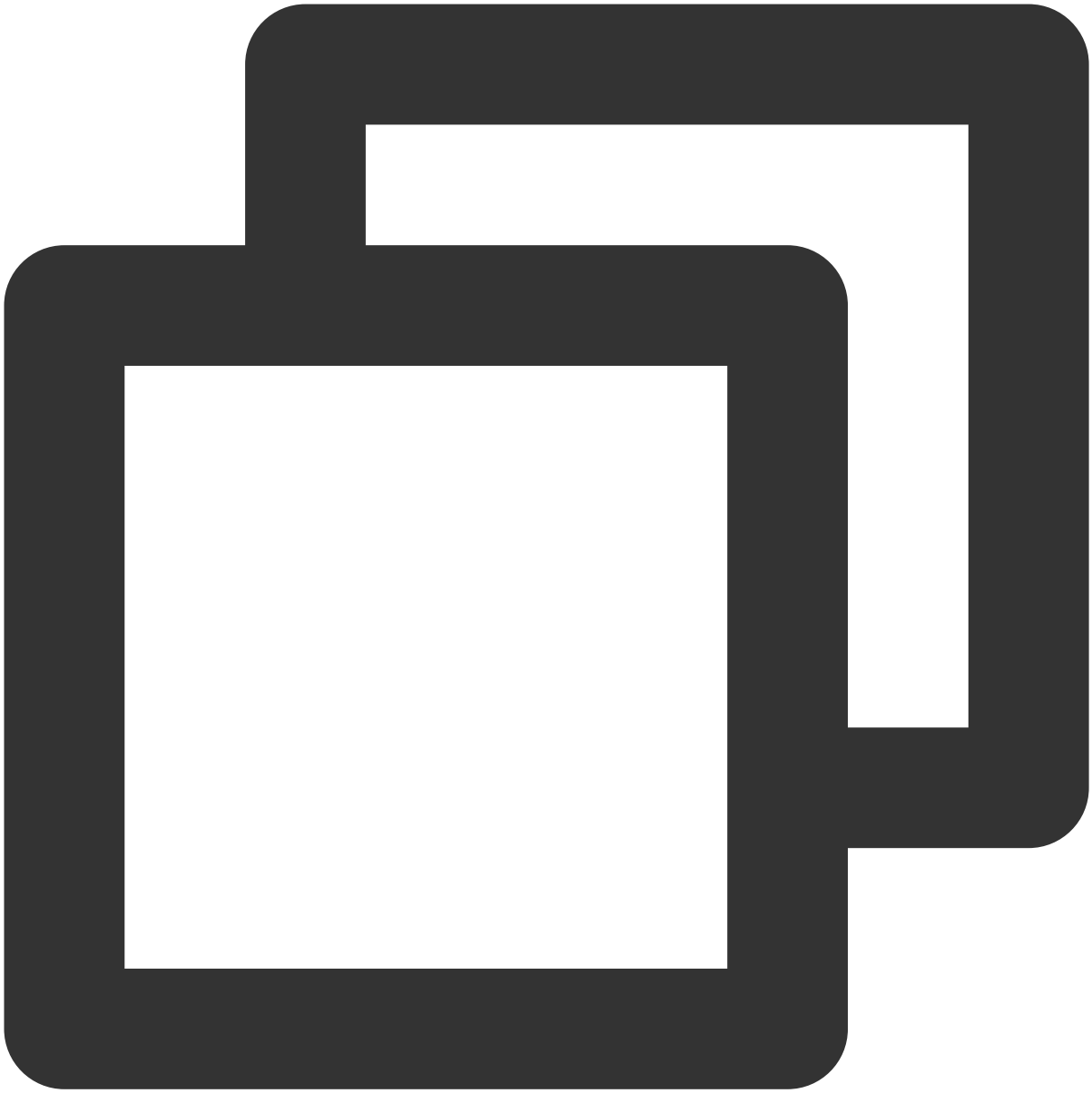
请求



```
POST /exampleobject?restore&versionId=MTg0NDUxNjQ1NjM4OTkzNzY3NDk HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Date: Mon, 20 Jan 2020 08:43:40 GMT
Content-Type: application/xml
Content-Length: 121
Content-MD5: Nr7RAnRMgrplFvD8bt5+0w==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close



<RestoreRequest>
            <Days>1</Days>
            <CASJobParameters>
                <Tier>Expedited</Tier>
            </CASJobParameters>
</RestoreRequest>
```

响应



```
HTTP/1.1 202 Accepted
Content-Length: 0
Connection: close
Date: Mon, 20 Jan 2020 08:43:41 GMT
Server: tencent-cos
x-cos-request-id: NWUyNTY4M2NfZTNjODJhMDlfMWZkM2VfNWZm****
```

APPEND Object

最近更新时间：2023-12-21 14:33:35

功能描述

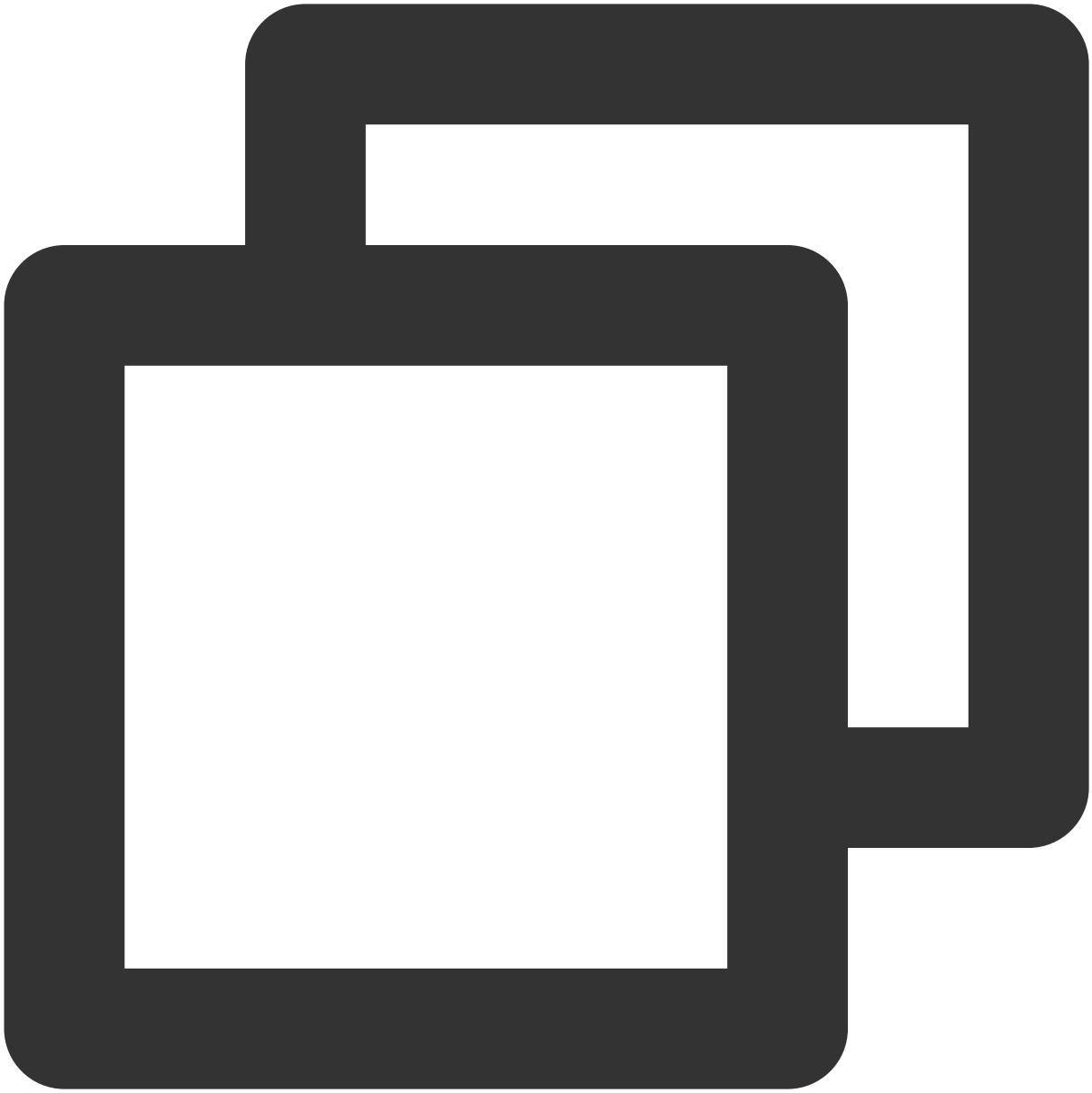
APPEND Object 接口请求可以将一个对象以分块追加的方式上传至指定存储桶中。对象首次使用 APPEND Object 接口上传时，该对象的属性自动为 appendable ，使用其他接口上传时则属性自动为 normal （如果该对象已存在则属性会被覆盖为 normal），可以使用 GET Object 或 HEAD Object 接口获取 x-cos-object-type 响应头来判断对象属性。对象属性为 appendable 时才能使用本接口追加上传。
追加上传的对象，每个分块大小默认最大为5GB，无最小限制，同时通过追加方式产生的对象大小不得超过5GB。
如果 Position 的值和当前对象的长度不致，COS 将返回409错误。如果追加一个 normal 属性的文件，COS 将返回 409 ObjectNotAppendable。

注意

Appendable 的对象不可以被复制，不参与版本管理，不参与生命周期管理。
使用 APPEND 接口进行追加上传时，COS 不会校验请求携带的存储类型，仅会以当前对象的存储类型为准。
APPEND 接口不支持智能分层存储类型。
开启了多 AZ 特性的存储桶，不支持使用 APPEND Object 接口。

请求

请求示例



```
POST /ObjectName?append&position=*position* HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Content-Length: size
Content-Type: ContentType
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

公共头部

该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

非公共头部

必选头部

该请求操作的实现使用如下必选头部：

名称	描述	类型	是否必选
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）	String	是

推荐头部

该请求操作的实现使用如下推荐请求头部信息：

节点名称（关键字）	描述	类型	是否必选
Cache-Control	RFC 2616 中定义的缓存策略，将作为 Object 元数据返回	String	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为 Object 元数据返回	String	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为 Object 元数据返回	String	否
Content-MD5	RFC 1864中定义的请求体内容的16字节二进制 MD5 哈希值的 Base64 编码形式，用于完整性检查，验证请求体在传输过程中是否发生变化，最终的取值长度应为24个字符，请注意在编写代码时使用正确的方法和参数，例如 `ZzD3iDJdrMAAb00lgLLeig==`	String	否
Content-Type	RFC 2616 中定义的内容类型（MIME），将作为 Object 元数据返回	String	否
Expect	当使用 Expect: 100-continue 时，在收到服务端确认后，才会发送请求内容	String	否
Expires	RFC 2616 中定义的过期时间，将作为 Object 元数据返回	String	否
x-cos-meta- *	允许用户自定义的头部信息，将作为 Object 元数据返回，大小限制 2K	String	否

版权所有：腾讯云计算（北京）有限责任公司

权限相关头部

该请求操作的实现可以用 POST 请求中的 x-cos-acl 头来设置文件访问权限。目前 Object 访问权限有三种：public-read-write，public-read 和 private。如果不设置，默认为 private 权限。也可以单独明确赋予用户读、写或读写权限。内容如下：

说明

了解更多 ACL 请求可详细请参见 PUT Bucket acl 文档。

名称	描述	类型	是否必选
x-cos-acl	定义 Object 的 ACL 属性，有效值为 private，public-read-write，public-read 默认值为 private	String	否
x-cos-grant-read	赋予被授权者读的权限，格式：`x-cos-grant-read: id=" ",id=" "` 当需要给子账号授权时，`id="qcs::cam::uin/<OwnerUin>:uin/<SubUin>"` 当需要给主账号授权时，`id="qcs::cam::uin/<OwnerUin>:uin/<OwnerUin>"`	String	否
x-cos-grant-write	赋予被授权者写的权限，格式：`x-cos-grant-write: id=" ",id=" "` 当需要给子账号授权时，`id="qcs::cam::uin/<OwnerUin>:uin/<SubUin>"` 当需要给主账号授权时，`id="qcs::cam::uin/<OwnerUin>:uin/<OwnerUin>"`	String	否
x-cos-grant-full-control	赋予被授权者读写权限，格式：`x-cos-grant-full-control: id=" ",id=" "` 当需要给子账号授权时，`id="qcs::cam::uin/<OwnerUin>:uin/<SubUin>"` 当需要给主账号授权时，`id="qcs::cam::uin/<OwnerUin>:uin/<OwnerUin>"`	String	否

请求参数

具体内容如下：

参数名称	描述	类型	是否必选
position	追加操作的起始点，单位为字节。首次追加则设置 Position=0，后续追加则设置 Position 为当前 Object 的 content-length	Int	是

请求体

该请求的请求体为空。

响应

响应头

公共响应头

该响应使用公共响应头，了解公共请求头详情请参见 公共响应头部 文档。

特有响应头

该请求操作的响应头具体数据为：

节点名称（关键字）	描述	类型
x-cos-next-append-position	下一次追加操作的起始点，单位：字节	String
ETag	文件的唯一标识	String

响应体

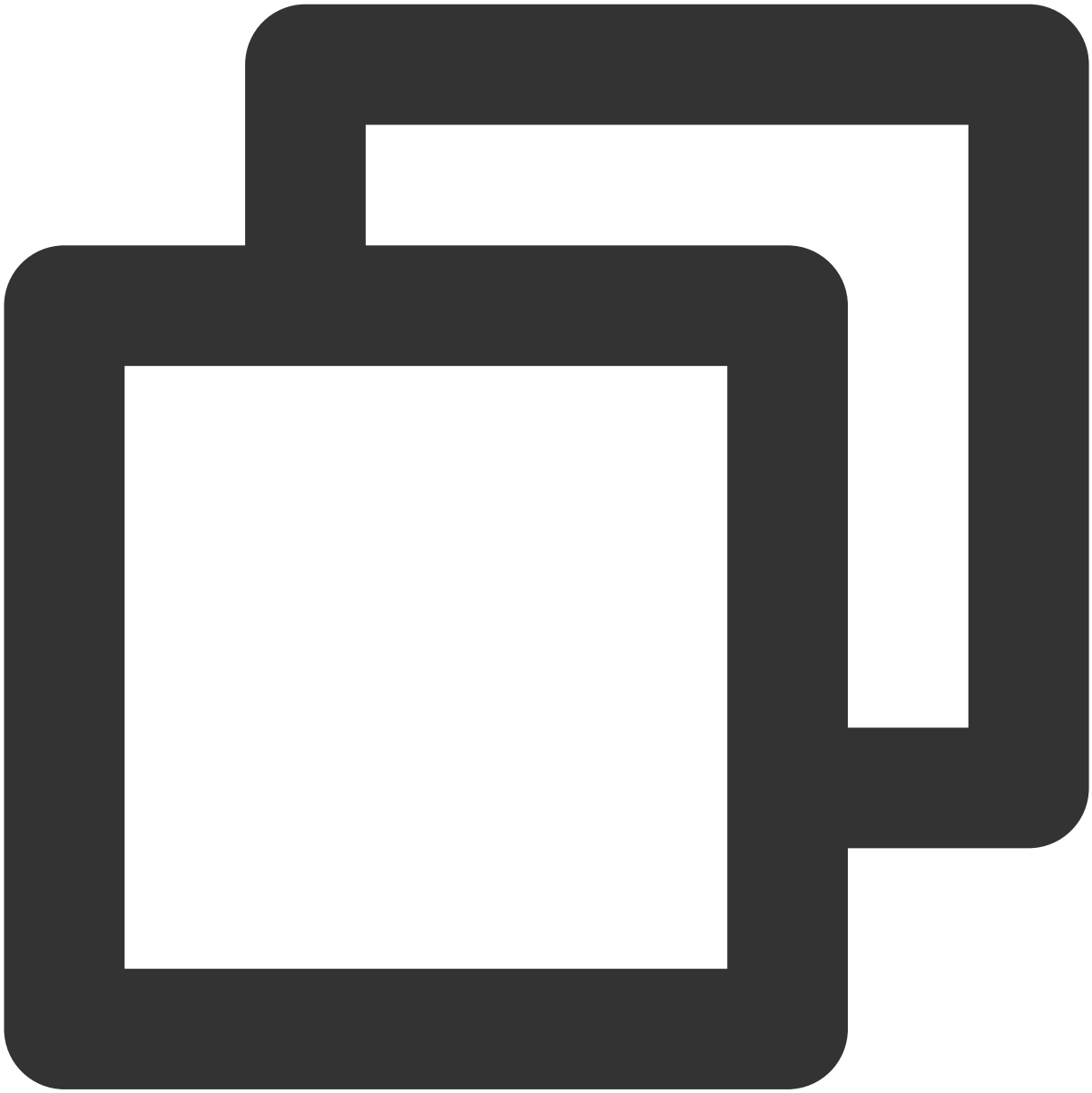
该响应体返回为空。

错误分析

如果对一个非 appendable 的文件进行 APPEND 操作，那么会返回409 Confilct，错误信息：The operation is not valid for the current state of the object。
如果请求中未携带 position 参数，会返回400 Bad Request，错误信息：InvalidArgument。
如果请求中缺失 Content-Length 头部，会返回 411 Length Required，错误信息：You must provide the Content-Length HTTP header。
获取更多关于 COS 的错误码的信息，或者产品所有的错误列表，请参见 错误码 文档。

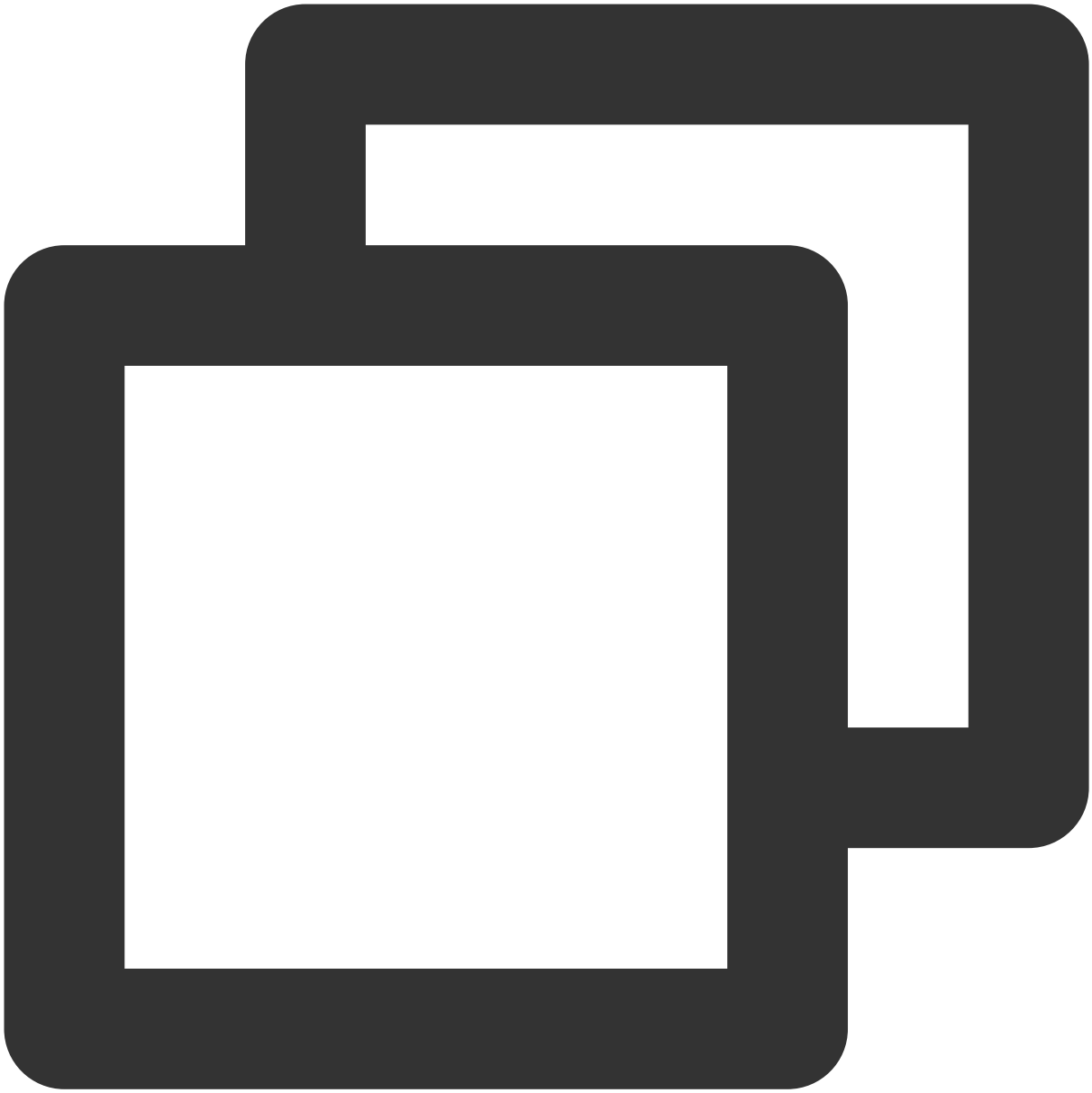
实际案例

请求



```
POST /coss3/app?append&position=0 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 16 Jan 2016 21:32:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3M****&q-si
Content-Length: 4096

[Object]
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Tue, 16 Jan 2016 21:32:00 GMT
ETag: 1ce5b469b7d6600ecc2fd112e570917b
Server: tencent-cos
x-cos-content-sha1: 1ceaf73df40e531df3bfb26b4fb7cd95fb7bff1d
x-cos-next-append-position: 4096
x-cos-request-id: NTg3NzNhZGZfMmM4OGY3X2I2Zl8x****
```

版权所有：腾讯云计算（北京）有限责任公司

版权所有：腾讯云计算（北京）有限责任公司

SELECT Object Content

最近更新时间：2023-12-21 14:34:01

概述

COS Select 接口可以使用结构化查询语句（Structured Query Language，SQL）从指定对象（CSV、JSON 或者 Parquet 格式）中检索内容。在检索过程中，您需要指定对象内容的分隔符，并使用合适的 SQL 函数进行检索，COS Select 将返回相匹配的检索结果，您可以指定检索结果的保存格式。
如您需要了解 COS Select 的更多介绍，请参见 COS Select 概述。有关 COS Select 的 SQL 表达式的介绍，您可以在开发者指南中参见 Select 命令 进一步了解。

注意

Select Object Content 接口当前仅支持 virtual-hosted 方式访问，不支持 path-style 方式访问。

权限限制

使用 COS Select，您必须具有 `cos:GetObject` 的授权。
如果您是主账号，则默认拥有该权限。
如果您是子账号，请联系您的主账号获取该操作的权限。有关权限设置，请参见 授权子账号访问 COS 文档。

对象数据格式

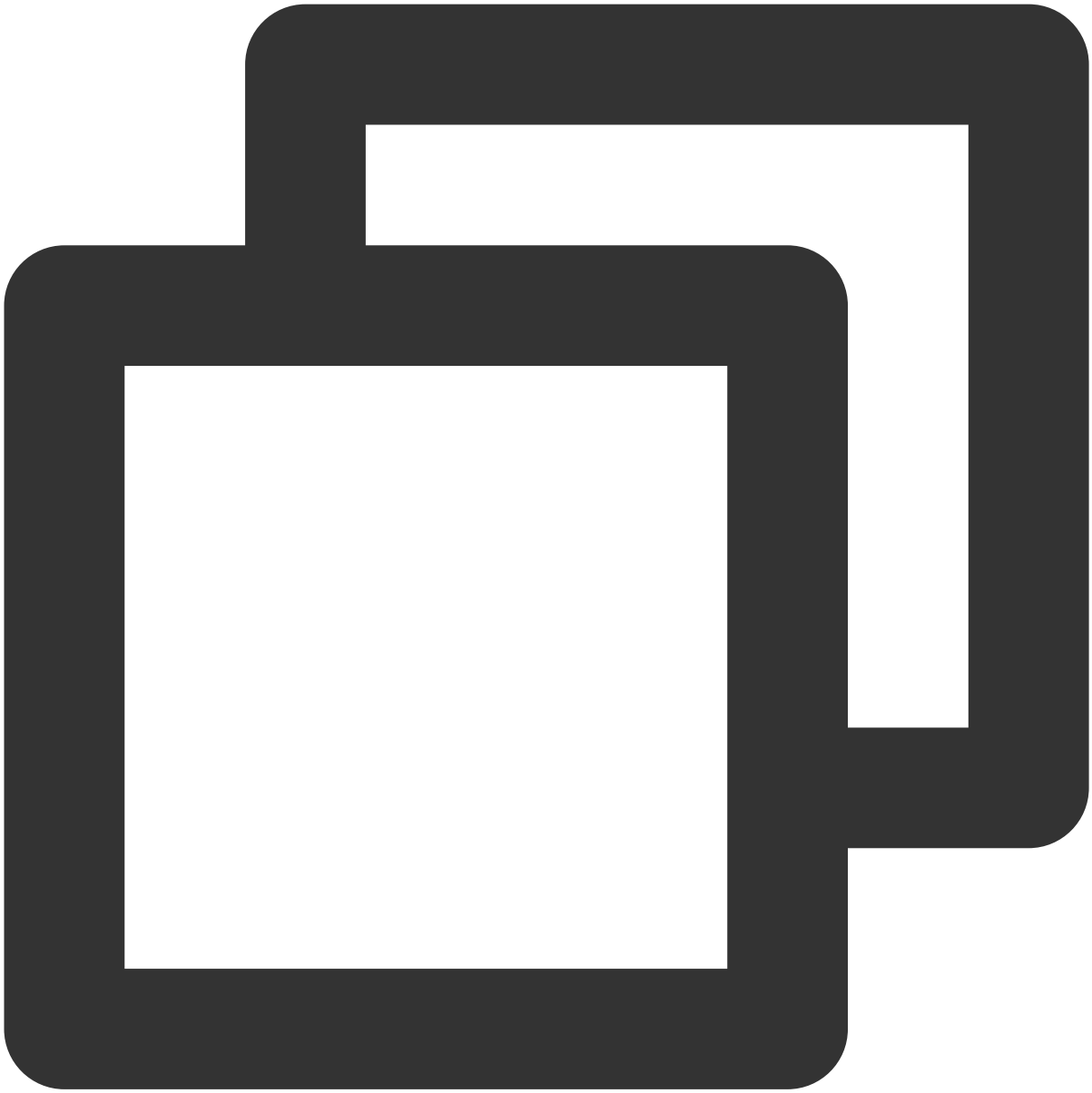
COS Select 支持检索以下格式的对象数据：
CSV 格式：对象以 CSV 格式存储，并以固定的分隔符划分。
JSON 格式：对象以 JSON 格式存储，可以是 JSON 文件或者 JSON 列表。
Parquet 格式：对象以 Parquet 格式存储，可以包含嵌套结构。

注意

COS Select 的对象必须以 UTF-8 格式编码。
COS Select 支持检索 GZIP 或者 BZIP2 压缩的 CSV、JSON 对象；支持检索经过 GZIP 或 Snappy 列压缩的 Parquet 对象。
COS Select 支持检索 SSE-COS 加密的对象。

请求

请求示例



```
POST /<ObjectKey>?select&select-type=2 HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: date
Authorization: Auth String

Request body
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

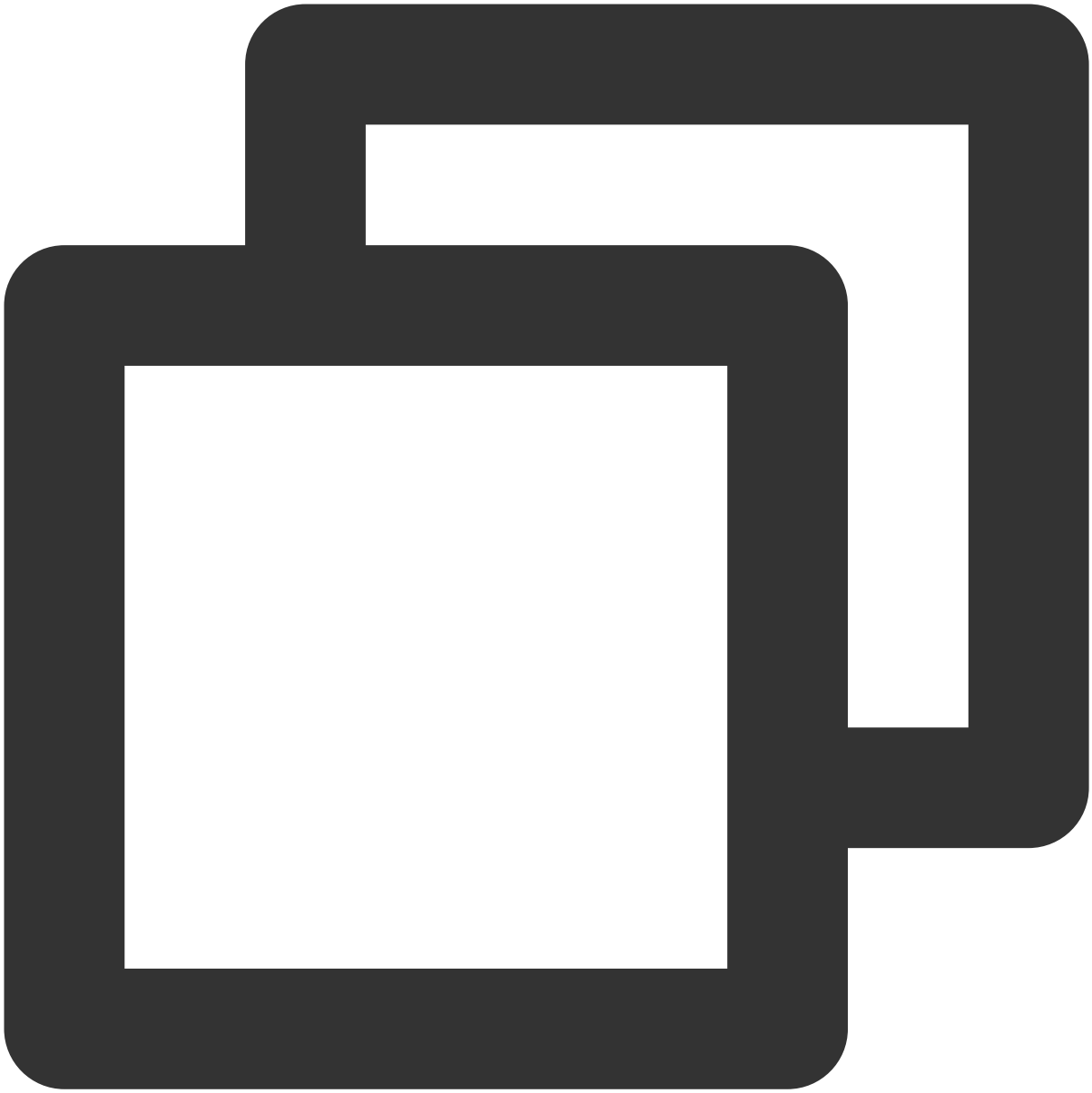
<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String （详情请参见 请求签名 文档）。
请求参数中 select 和 select-type=2 参数均为必填参数，其中 select 代表发起 select 请求，select-type=2 代表这一接口的版本信息。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

以下请求展示了用户发起一个 COS Select 请求，检索 CSV 格式对象的所有内容，并将结果保存为 CSV 格式对象。

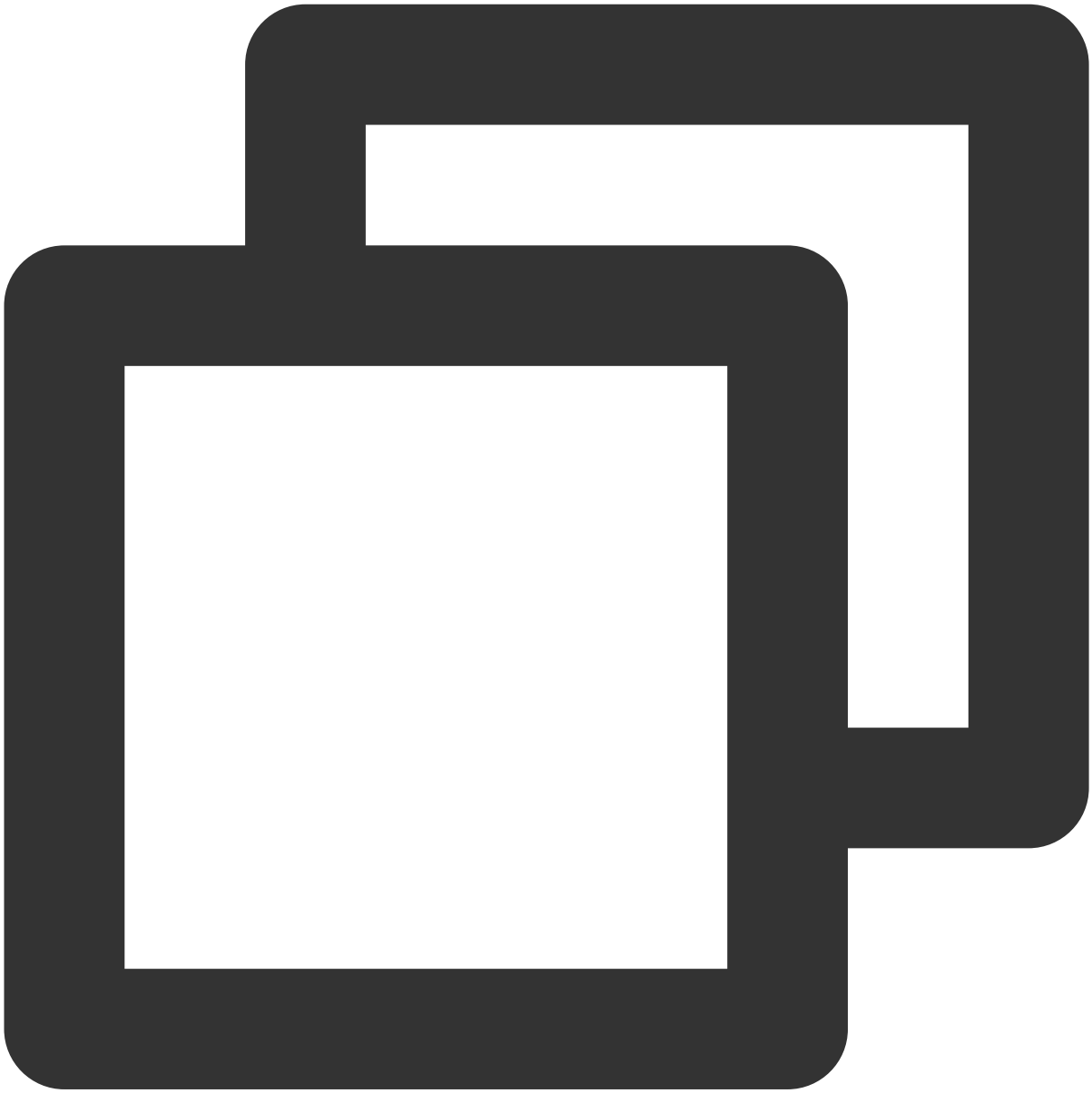


```
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from COSObject</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>GZIP</CompressionType>
        <CSV>
            <FileHeaderInfo>IGNORE</FileHeaderInfo>
            <RecordDelimiter>\\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
            <Comments>#</Comments>
            <AllowQuotedRecordDelimiter>FALSE</AllowQuotedRecordDelimiter>
        </CSV>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
    <RequestProgress>
        <Enabled>FALSE</Enabled>
    </RequestProgress>
</SelectRequest>
```

以下请求展示了用户发起一个 COS Select 请求，检索 JSON 格式对象的所有内容，并将结果保存为 JSON 格式对象。

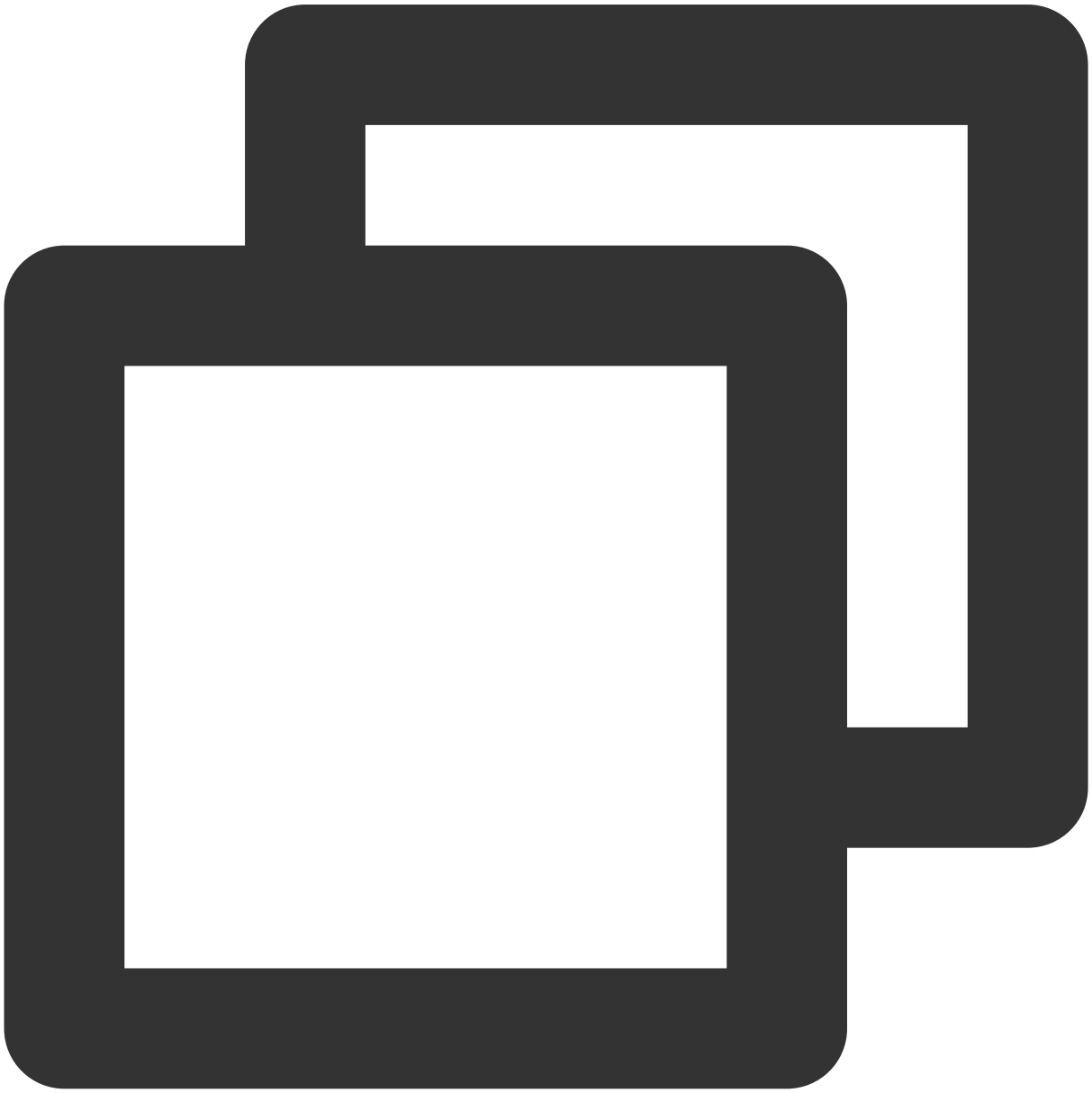


```
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from COSObject</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>GZIP</CompressionType>
        <JSON>
            <Type>DOCUMENT</Type>
        </JSON>
    </InputSerialization>
    <OutputSerialization>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <JSON>
                <RecordDelimiter>\\n</RecordDelimiter>
            </JSON>
        </OutputSerialization>
        <RequestProgress>
            <Enabled>FALSE</Enabled>
        </RequestProgress>
    </SelectRequest>
```

以下请求展示了用户发起一个 COS Select 请求，检索 Parquet 格式对象的所有内容，并将结果保存为 JSON 格式对象。



```
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from COSObject</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>GZIP</CompressionType>
        <Parquet>
        </Parquet>
    </InputSerialization>
    <OutputSerialization>
        <JSON>
            <RecordDelimiter>\\n</RecordDelimiter>
        </JSON>
    </OutputSerialization>
    <RequestProgress>
        <Enabled>FALSE</Enabled>
    </RequestProgress>
</SelectRequest>
```

说明

InputSerialization 元素描述了待检索的对象格式，为必填参数，该参数可以指定为 CSV 、JSON 或 Parquet 格式。

OutputSerialization 元素描述了检索结果的保存格式，该参数可以仅可指定为 CSV 或者 JSON 格式。

待检索的对象格式无需和检索结果的保存格式互相匹配，您可以检索一个 JSON 格式的对象，并将检索结果保存为 CSV 格式。

下表展示了请求体中的各项元素组成：

名称	父节点	描述	类型	是否必选
Expression	SelectRequest	SQL 表达式，代表您需要发起的检索操作。例如 `SELECT s._1 FROM COSObject s` 。这个表达式可以从 CSV 格式的对象中检索第一列内容。有关 SQL 表达式的详细介绍，请参见 Select 命令	String	是
ExpressionType	SelectRequest	表达式类型，该项为扩展项，目前只支持 SQL 表达式，仅支持 SQL 参数	String	是
InputSerialization	SelectRequest	描述待检索对象的格式	Container	是
OutputSerialization	SelectRequest	描述检索结果的输出格式	Container	是
RequestProgress	SelectRequest	是否需要返回查询进度 QueryProgress 信息，如果选中 COS Select 将周期性返回查询进度	Container	否

版权所有：腾讯云计算（北京）有限责任公司

InputSerialization container element

名称	父节点	描述	类型	是否必选
CompressionType	InputSerialization	描述待检索对象的压缩格式：如果对象未被压缩过，则该项为 NONE。如果对象被压缩过，COS Select 目前支持的两种压缩格式为 GZIP 和 BZIP2，可选项为 NONE、GZIP、BZIP2，默认值为 NONE	String	否
CSV/JSON/PARQUET	InputSerialization	描述在相应的对象格式下所需的文件参数。例如 CSV 格式需要指定分隔符	Container	是

CSV container element (InputSerialization 子元素)

名称	父节点	描述	类型	是否必选
RecordDelimiter	CSV	将 CSV 对象中记录分隔为不同行的字符，默认您通过 `\\n` 进行分隔。您可以指定任意8进制字符，如逗号、分号、Tab 等。该参数最多支持2个字节，即您可以输入 `\\r\\n` 这类格式的分隔符。默认值为 `\\n` 。	String	否
FieldDelimiter	CSV	指定分隔 CSV 对象中每一行的字符，默认您通过,进行分隔。您可以指定任意8进制字符，该参数最多支持1个字节。默认值为 `,` 。	String	否
QuoteCharacter	CSV	如果您待检索的 CSV 对象中存在包含分隔符的字符串，您可以使用 QuoteCharacter 进行转义，避免该字符串被切割成几个部分。如 CSV 对象中存在 `"a, b"` 这个字符串，双引号"可以避免这一字符串被分隔成 `a` 和 `b` 两个字符。默认值为 `"` 。	String	否
QuoteEscapeCharacter	CSV	如果您待检索的字符串中已经存在 `"` ，那您需要使用 `"` 进行转义以保证字符串可以正常转义。如您的字符串 `""" a , b """` 将会被解析为 `" a , b "` 。默认值为"。	String	否
AllowQuotedRecordDelimiter	CSV	指定待检索对象中是否存在与分隔符相同且	Boolean	否

		需要用"转义的字符。设定为 TRUE 时，COS Select 将会在检索进行转义，这会导致检索性能下降；设定为 FALSE 时，则不会做转义处理。默认值为 FALSE。		
FileHeaderInfo	CSV	待检索对象中是否存在列表头。该参数为存在 NONE、USE、IGNORE 三个选项。NONE 代表对象中没有列表头，USE 代表对象中存在列表头并且您可以使用表头进行检索（例如 `SELECT "name" FROM COSObject`），IGNORE 代表对象中存在列表头且您不打算使用表头进行检索（但您仍然可以通过列索引进行检索，如 `SELECT s._1 FROM COSObject s`）。合法值为 NONE、USE、IGNORE。	Enum	否
Comments	CSV	指定某行记录为注释行，该字符会被添加到该行记录的首字符。如果某一行记录被指定为注释，则 COS Select 将不对此行做任何分析。默认值为 `#`。	String	否

JSON container element (InputSerialization 子元素)

名称	父节点	描述	类型	是否必选
Type	JSON	JSON 文件的类型： DOCUMENT 表示 JSON 文件仅包含一个独立的 JSON 对象，且该对象可以被切割成多行 LINES 表示 JSON 对象中的每一行包含了一个独立的 JSON 对象 合法值为 DOCUMENT 、LINES	Enum	是

OutputSerialization container element

名称	父节点	描述	类型	是否必选
CSV /JSON	OutputSerialization	指定检索结果的输出格式，可选项为 CSV 或者 JSON	Container	是，必须是 CSV 或者 JSON 中的一个

CSV container element (OutputSerialization 子元素)

名称	父节点	描述	类型	是否必选
QuoteFields	CSV	指定输出结果为文件时，是否需要使用 `"` 进行转	String	是

		义。可选项包括 ALWAYS、ASNEEDED、ALWAYS 代表对所有本次输出的检索文件应用 `"`，ASNEEDED 代表仅在需要时使用。合法值为 ALWAYS、ASNEEDED，默认值为 ASNEEDED。		
RecordDelimiter	CSV	将输出结果中的记录分隔为不同行的字符，默认通过 `\\n` 进行分隔。您可以指定任意8进制字符，如逗号、分号、Tab 等。该参数最多支持2个字节，即您可以输入 `\\r\\n` 这类格式的分隔符。默认值为 `\\n` 。	String	否
FieldDelimiter	CSV	将输出结果中的每一行进行分列的字符，默认通过 `,` 进行分隔。您可以指定任意8进制字符，该参数最多支持1个字节。默认值为 `,` 。	String	否
QuoteCharacter	CSV	如果输出结果中存在包含分隔符的字符串，可以使用 QuoteCharacter 进行转义，保证该字符串不会在后续分析中被切割。如输出结果中存在 `a,b` 这个字符串，双引号 `"` 可以避免这一字符串被分隔成 `a` 和 `b` 两个字符，COS Select 将会将其转为 `"a, b"` 写入文件。默认值为 `"` 。	String	否
QuoteEscapeCharacter	CSV	如果您即将输出的字符串中已经存在 `"`，那您需要使用 `"` 进行转义以保证该字符串可以正常转义。如您的字符串 `" a , b"` 将会被在写入文件时被转换为 `""" a , b """` 。默认值为 `"` 。	String	否

JSON container element (OutputSerialization 子元素)

名称	父节点	描述	类型	是否必选
RecordDelimiter	JSON	将输出结果中的记录分隔为不同行的字符，默认通过 `\\n` 进行分隔。您可以指定任意8进制字符，如逗号、分号、Tab 等。该参数最多支持2个字节，即您可以输入 `\\r\\n` 这类格式的分隔符。默认值为 `\\n`	String	否

RequestProgress container element

名称	父节点	描述	类型	是否必选
Enabled	RequestProgress	指定是否需要 COS Select 定期返回查询进度。默	Boolean	否

		认值为 FALSE		

响应

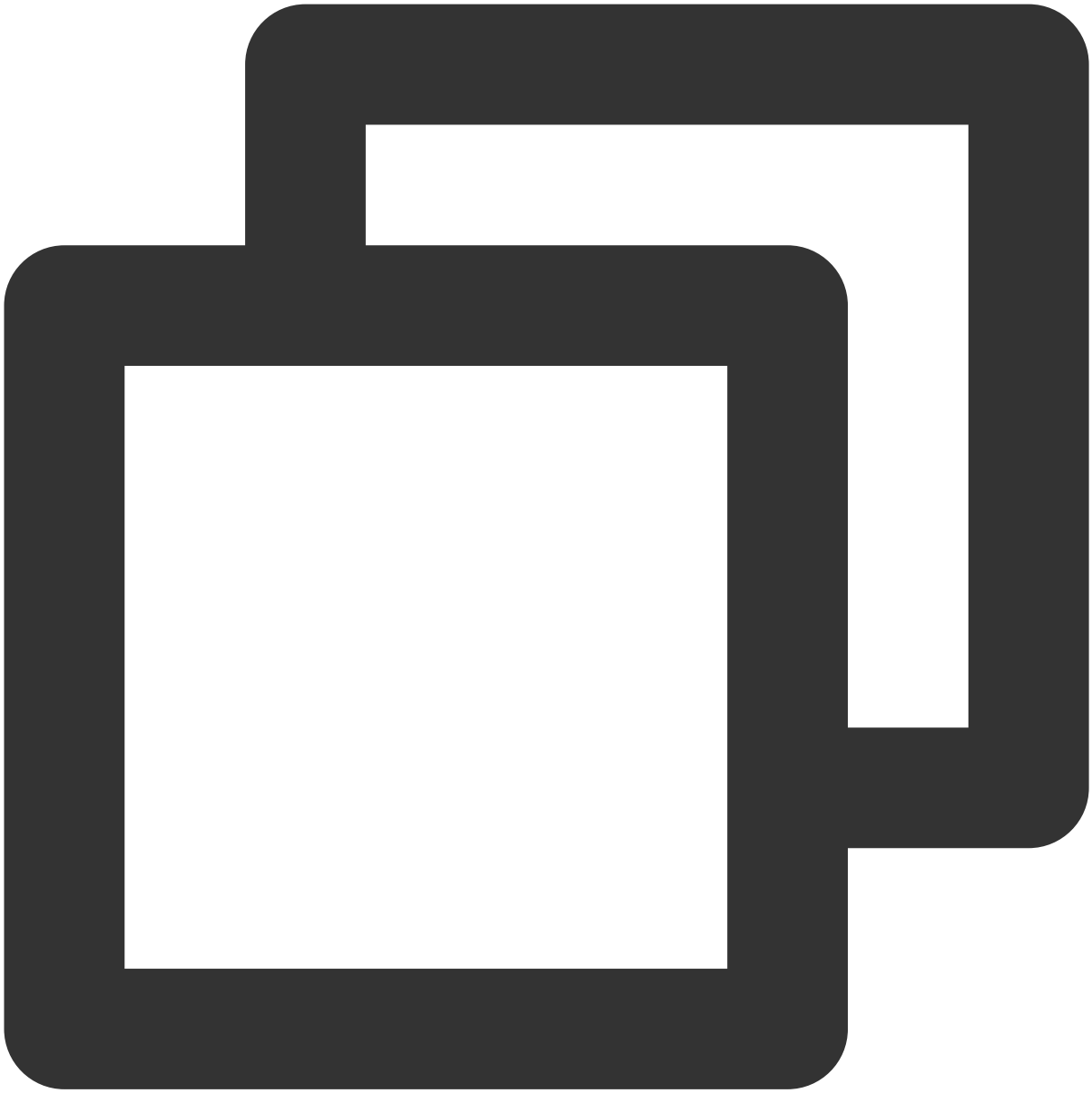
执行成功的检索操作将返回 `200 OK` 状态码。

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

由于响应体的大小无法预知，COS 将用户请求响应体以序列化形式展示，即将响应体切分成多个分块返回，如下展示了返回响应体的概览：



```
<Message 1>
<Message 2>
<Message 3>
......
<Message n>
```

预响应（prelude）和响应结果（data）

COS 将检索结果切成多个分块，每个分块即一个 Message。每一个 Message 由预响应（prelude）和响应结果（data）组成。

预响应包含两个部分：

所在分块 Message 的总长度。

所有头部的总长度。

响应结果包含两个部分：

响应报头（header）。

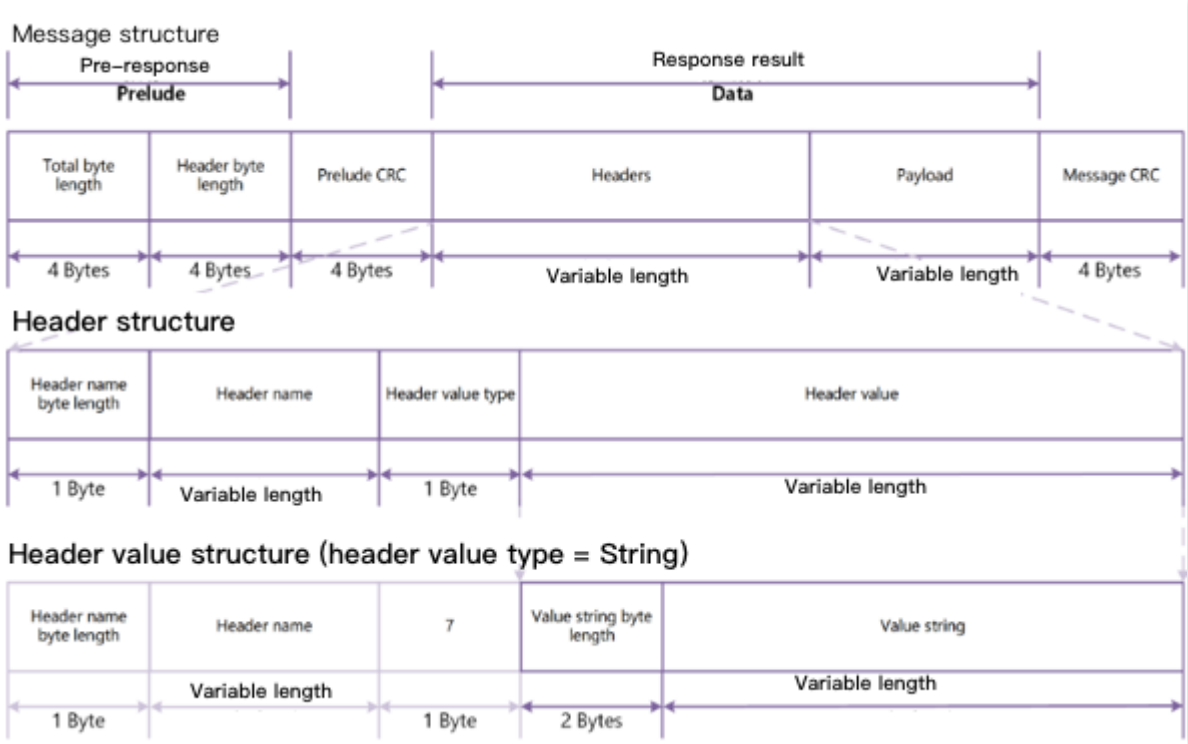
响应正文（payload）。

预响应和响应结果均以4字节的经过大端编码（big-endian）的 CRC 校验码结尾。COS Select 使用 CRC32计算 CRC 校验码，有关 CRC32的详细信息，请参见 RFC 文档。除了响应结果之外，COS Select 总共额外花费了16字节用于传输预响应和校验码信息。

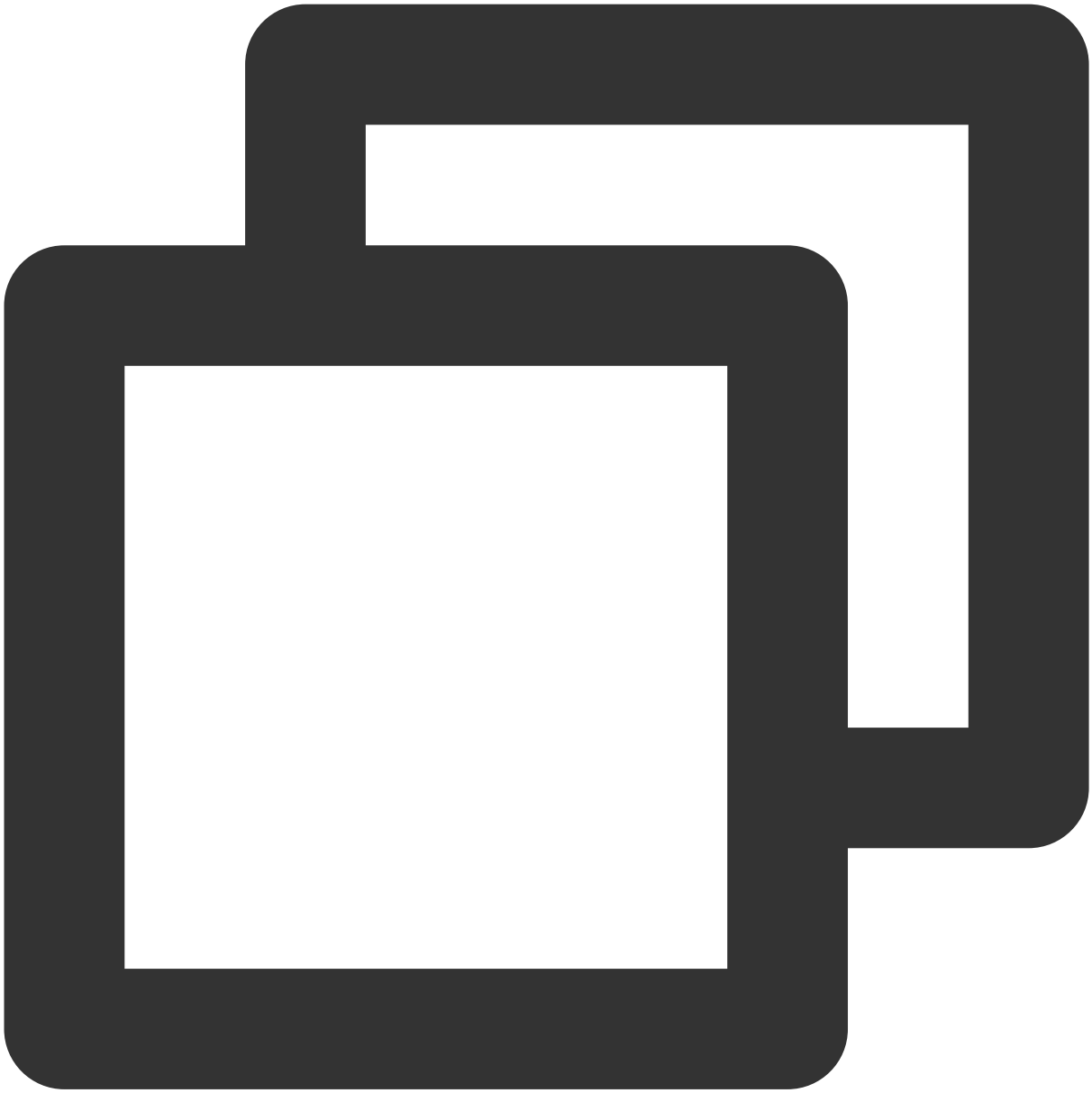
注意

所有分块 Message 中的整数值均以网络字节序，即大端编码，传输。

下图展示了分块 Message 以及检索结果中的响应报头 header 是如何构成的。每一个分块 Message 中可能包含多个 header 。

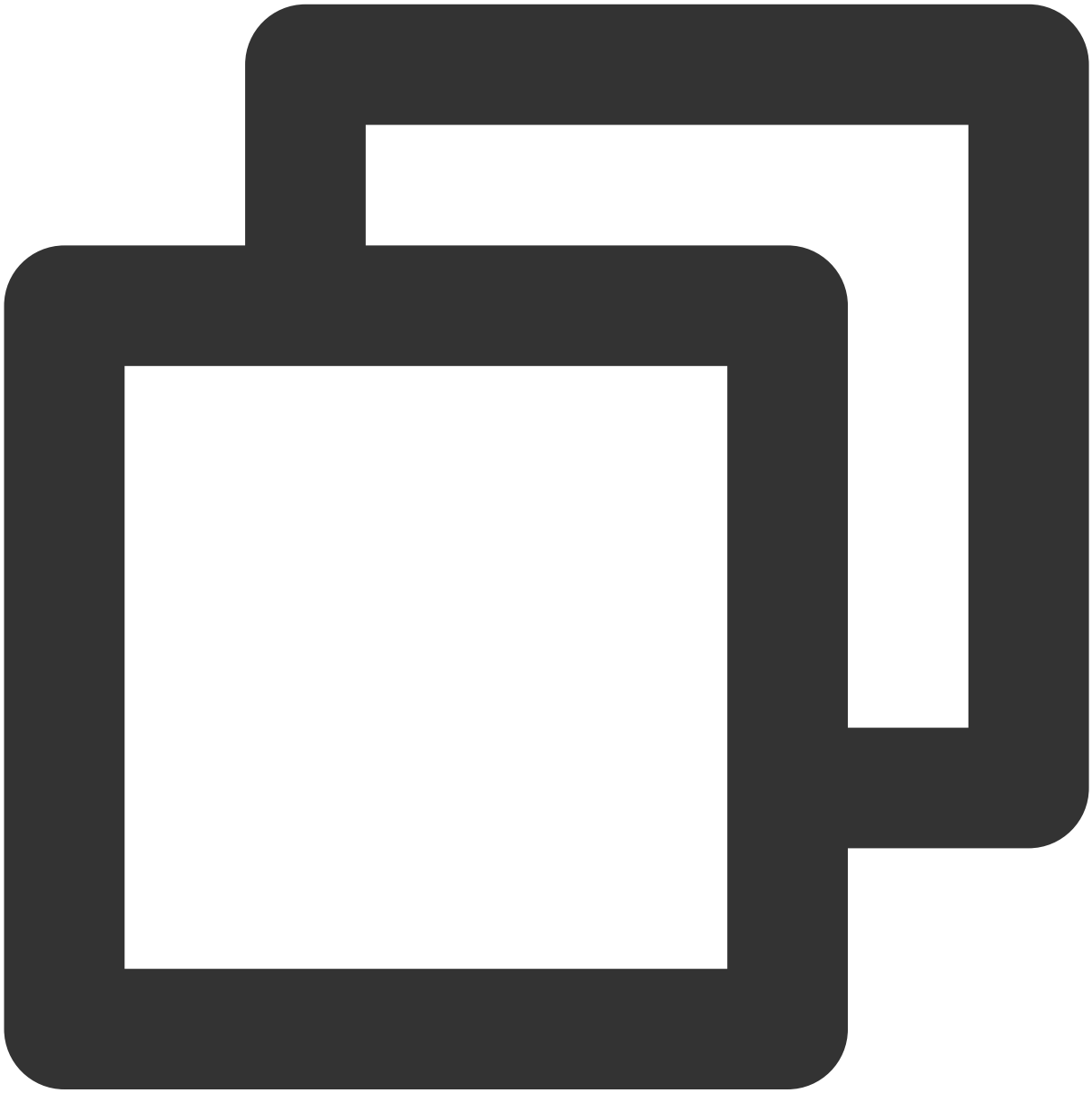


如上图所示，每一个分块 Message 均由预响应 prelude，预响应校验码 prelude CRC（由两个记录字节数的信息组成），报头信息 header ，响应正文 Payload 和正文校验码 Message CRC 构成。从上图可以看到，整个响应体的长度计算方式如下：



```
响应体总长度 = 预响应长度 + 预响应校验码长度 + 响应正文长度 + 响应报头长度 + 响应正文校验码长度
```

由于校验码（prelude CRC 和 Message CRC）和预响应 prelude 总长度固定为16字节，因此响应体总长还可以通过如下方式快速计算：



```
响应体总长度 =  响应正文长度 + 响应报头长度 + 16
```

以下详细介绍响应体各部分组成：

组成	描述
预响应 prelude	分别记录了分块 Message 的总长度和所有报头的总长度，每个记录4字节，总长8字节： 1. total byte-length：所在分块 Message 的总长度，使用大端编码，包含该记录本身容量共4字节。

	2. headers byte-length：所有报头的总长度，使用大端编码，不包含该记录所占空间共4字节。
预响应校验码 prelude CRC	预响应的 CRC 校验码，使用大端编码，总共4字节。预响应校验码可以帮助程序快速识别预响应信息是否正确，减少缓冲时的阻塞。
报头信息 header	分块 Message 记录的检索结果的元数据信息，诸如数据类型，正文格式。根据数据类型的差异，本部分的字节长度也有所差异。响应报头以 kv 键值对形式存储，使用 UTF-8编码。响应报头中所记录的元数据信息可以以任意顺序展示，但每一项元数据仅记录一次。根据数据类型的差异，以下响应报头均有可能在 COS Select 返回的结果中出现：、 1. MessageType Header：该报头代表响应类型。Key 值为":message-type"，合法的 Value 值为"error"或者"event"，"error"代表本条记录为报错信息，"event"代表本条记录为具体的事件。 2. `EventType Header` ：该报头记录事件类型。Key 值为":event-type"，合法的 Value 值为"Records"，"Cont"，"Progress"，"Stats"或"End"。"Records"代表事件为返回检索记录，"Cont"代表事件为保持 TCP 连接，"Progress"代表事件为定期返回的检索结果，"Stats"代表事件为本次查询的统计信息，"End"代表本次查询结束。 3. `ErrorCode Header` ：该报头记录报错类型。Key 值为":error-code"，合法的 Value 值为 特殊错误码 中的错误码信息。 4. `ErrorMessage Header` ：该报头记录错误码信息。Key 值为":error-message"，合法的 Value 值为服务端返回的错误码信息，可用于定位错误。
响应正文 Payload	记录检索结果，或者与请求相关的正式信息。
正文校验码 Message CRC	使用大端编码的 CRC 校验码，总长4字节。

同一个分块 Message 中可能记录了多个 header，每一个响应报头 header 由以下几部分组成：

组成	描述
Header Name Byte-Length	记录 Header Name 的字节长度信息
Header Name	报头类型，合法值包括 ":message-type"， ":event-type"， ":error-code"和":error-message" ":message-type"代表该报头记录了响应类型 ":event-type"代表了该报头记录事件类型 ":error-code"代表该报头记录报错类型 ":error-message"代表该报头记录错误码信息
Header Value Type	Header Value 的类型，对于 COS Select 而言这个值固定为7，代表类型为 String
Value String Byte-Length	Header Value 的字节长度信息，固定2字节

Header Value String	响应报头的正文，即响应正文的元数据信息，Header Value 的字节长度与响应类型相关

COS Select 的响应类型主要可以分为以下几种：

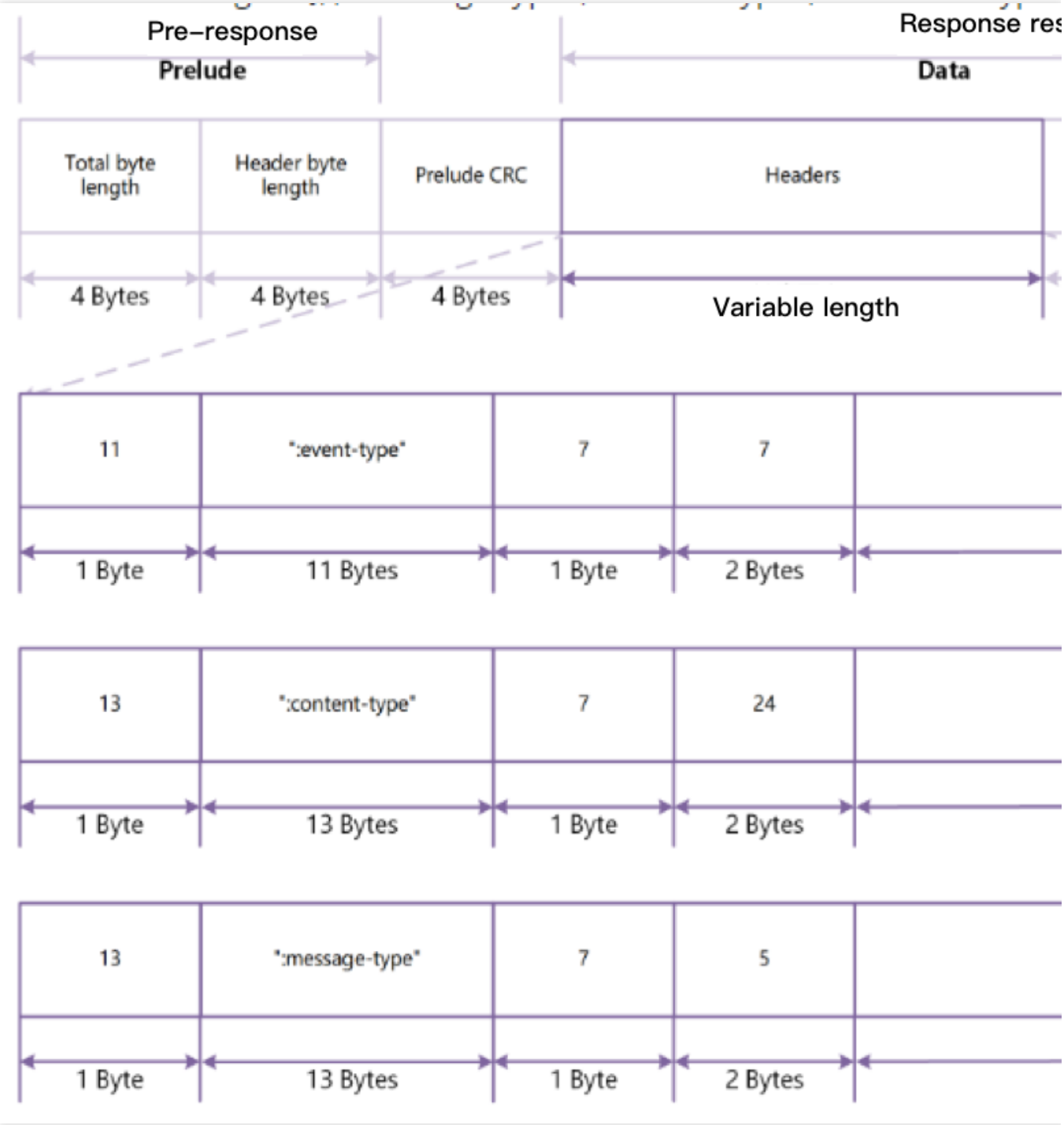
响应类型	描述
Records message	检索信息，可以包含单条记录，部分记录或者多条记录，取决于检索结果的多少。一个响应体中可能包含多个 Records message
Continuation message	连接信息，COS Select 周期性地发送这些信息以保持 TCP 连接，这些信息随机出现在响应体中。客户端最好能够自动识别这类信息，并对其做过滤处理以免弄脏检索结果
Progress message	进度信息，COS Select 周期性地返回这些信息以反馈当前查询进度
Stats message	统计信息，COS Select 在查询结束后返回本次查询的相关统计信息
End message	结束信息，代表本次查询已经结束，没有后续响应数据。只有在接受到该类型的信息时才能认为查询结束
RequestLevelError message	报错信息，COS Select 在查询出现错误时将会返回这一信息，包含请求的错误原因。如果 COS Select 返回了这一信息，则将不会再返回 End message 信息

下面将进一步介绍这些响应类型的详情。

Records message

报头格式

Records message 包括":message-type"， ":event-type"， ":content-type"3种类型的报头。如下所示：



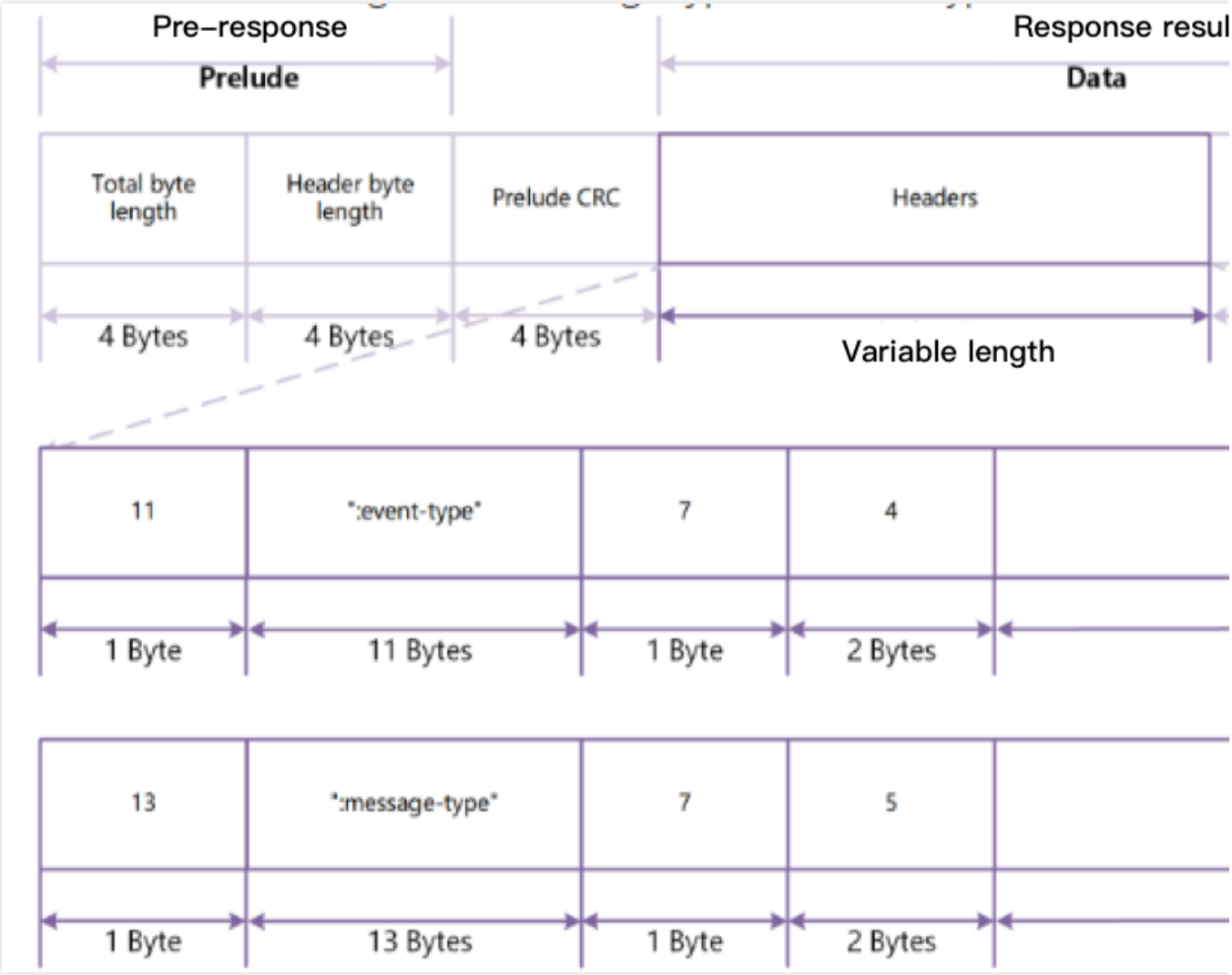
正文格式

Records message 正文可能包含单条记录，部分记录或者多条记录，取决于检索结果的多少。

Continuation Message

报头格式

Continuation Message 包括":message-type"， ":event-type"2种类型的报头，如下所示：



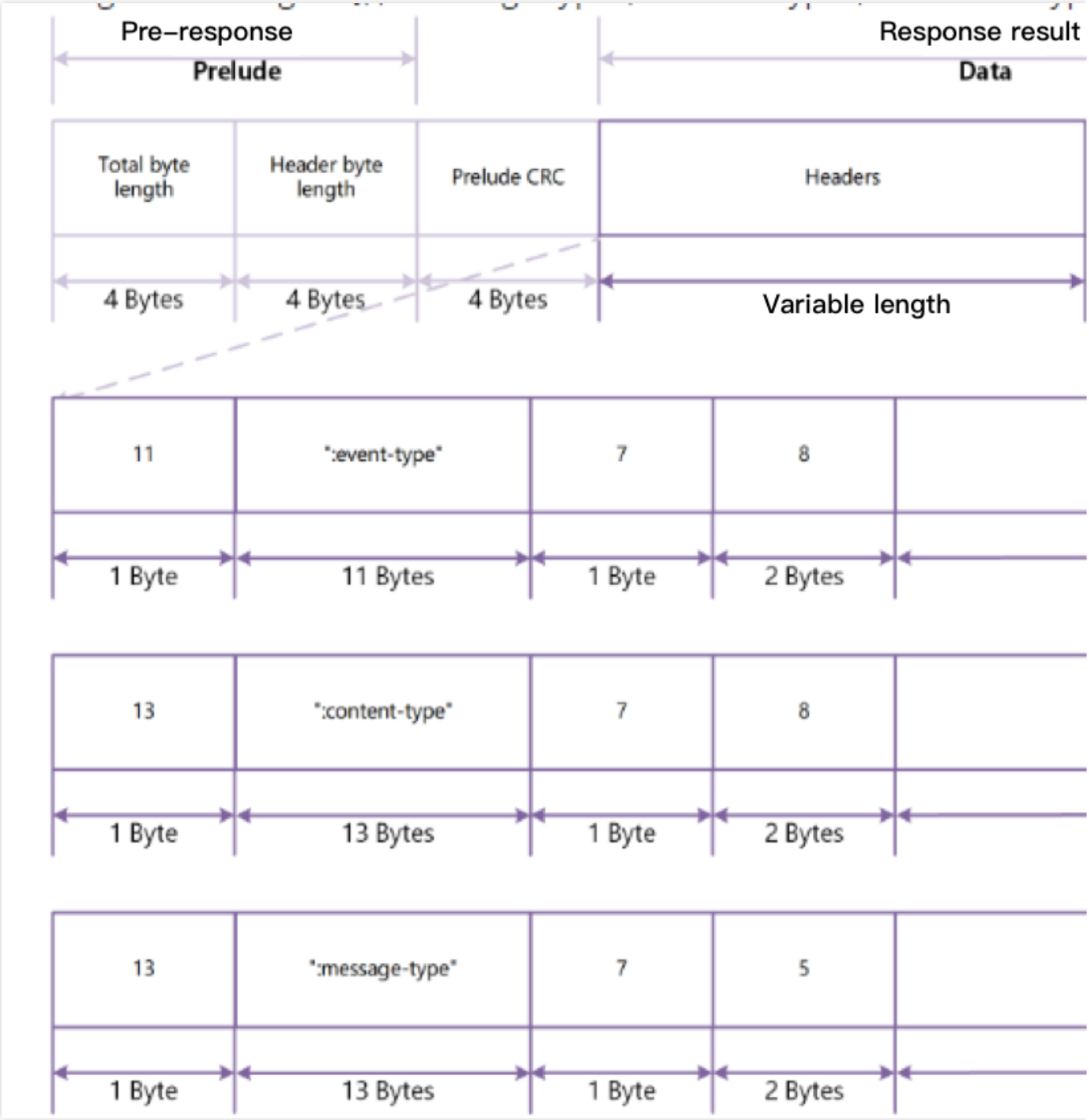
正文格式

Continuation Message 中不包含正文内容。

Progress Message

报头格式

Progress Message 包括":message-type"， ":event-type"， ":content-type"3种类型的报头，如下所示：



正文格式

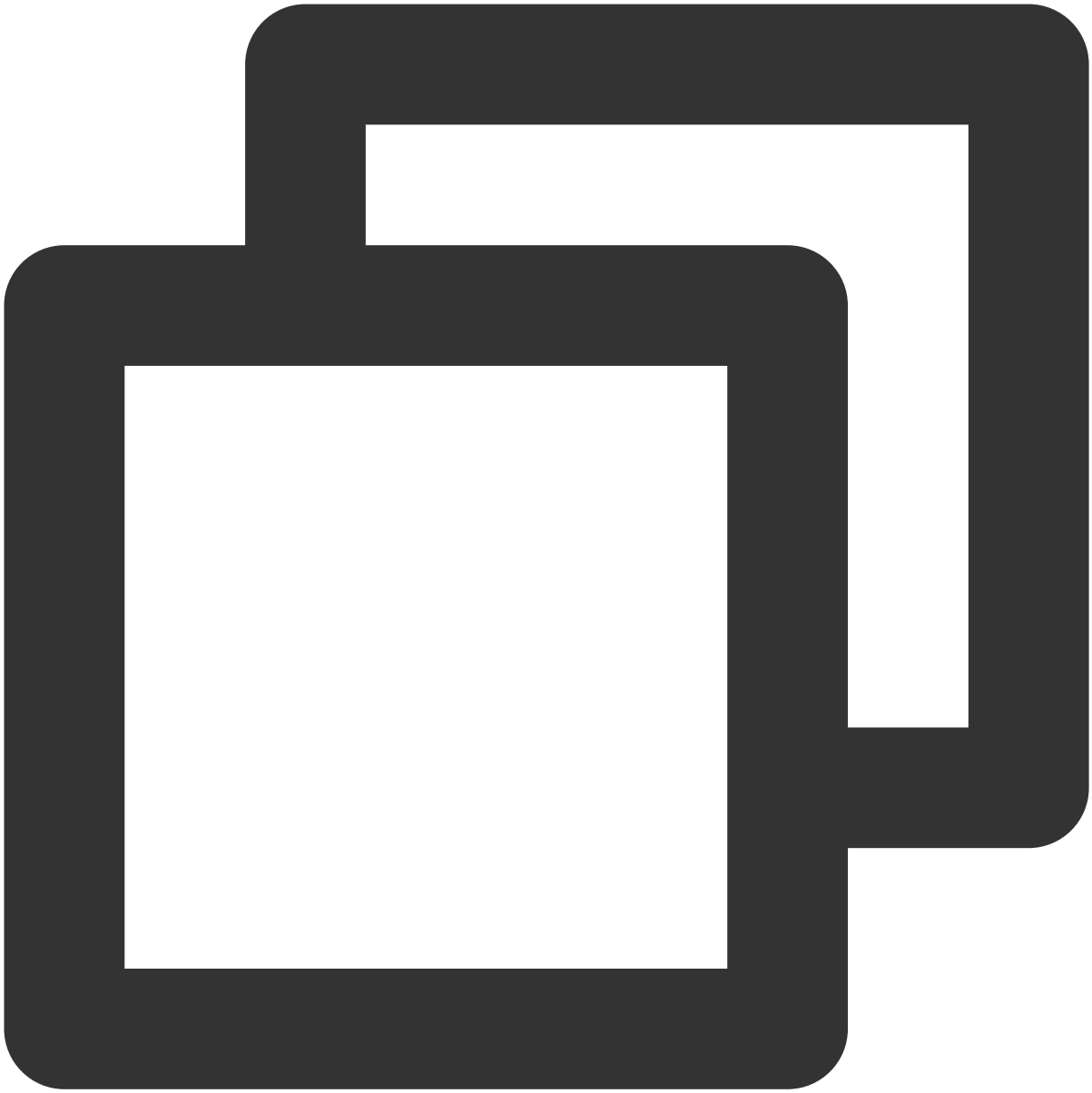
Progress Message 的正文是一个包含了当前查询进度的 XML 文本，主要包含以下信息：

BytesScanned：如果文件是压缩文件，该数值代表文件解压前的字节大小。如果文件不是压缩文件，该数值即文件的字节大小。

BytesProcessed：如果文件是压缩文件，该数值代表文件解压后的字节大小。如果文件不是压缩文件，该数值即文件的字节大小。

BytesReturned：COS Select 目前返回的检索结果字节大小。

示例如下：

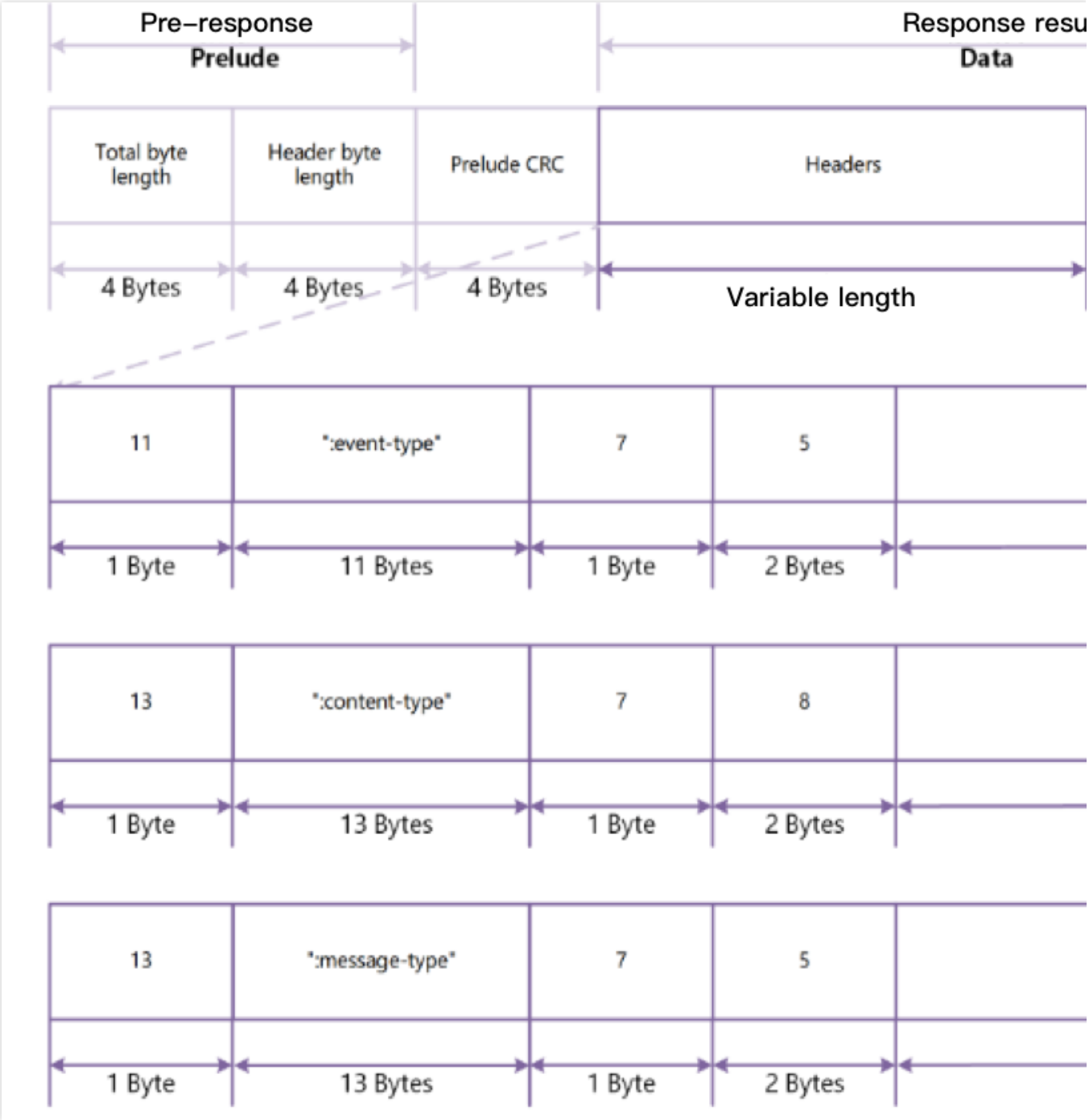


```
<?xml version="1.0" encoding="UTF-8"?>
<Progress>
     <BytesScanned>512</BytesScanned>
     <BytesProcessed>1024</BytesProcessed>
     <BytesReturned>1024</BytesReturned>
</Progress>
```

Stats Message

报头格式

Stats Message 包括":message-type"， ":event-type"， ":content-type"3种类型的报头，如下所示：



正文格式

Stats message 的正文是一个包含了本次查询统计的 XML 文本，主要包含以下信息：

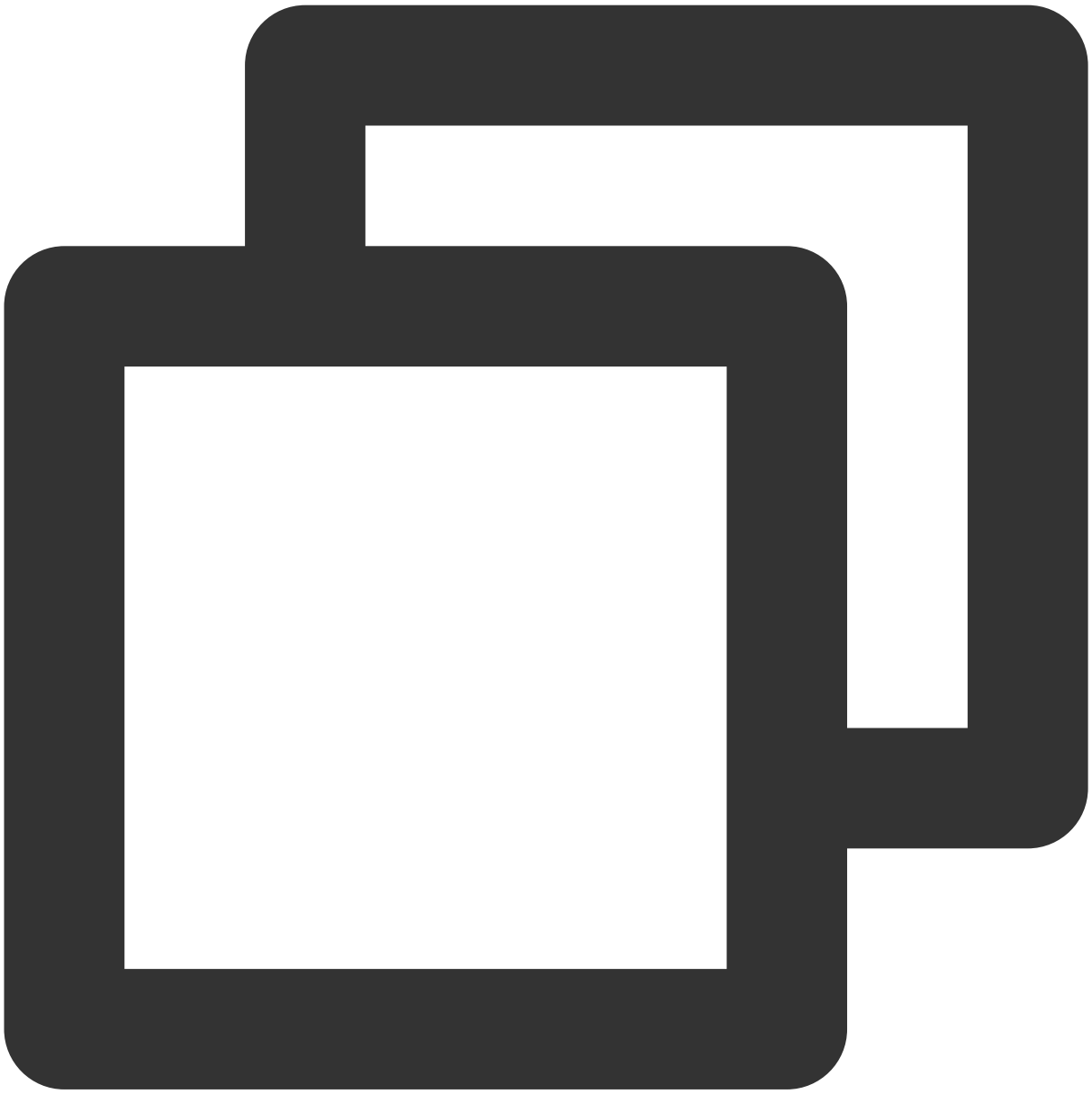
BytesScanned：如果文件是压缩文件，该数值代表文件解压前的字节大小；如果文件不是压缩文件，该数值即文件的字节大小。

BytesProcessed：如果文件是压缩文件，该数值代表文件解压后的字节大小；如果文件不是压缩文件，该数值即文件的字节大小。

版权所有：腾讯云计算（北京）有限责任公司

BytesReturned：COS Select 在本次查询中返回的检索结果字节大小。

示例如下：

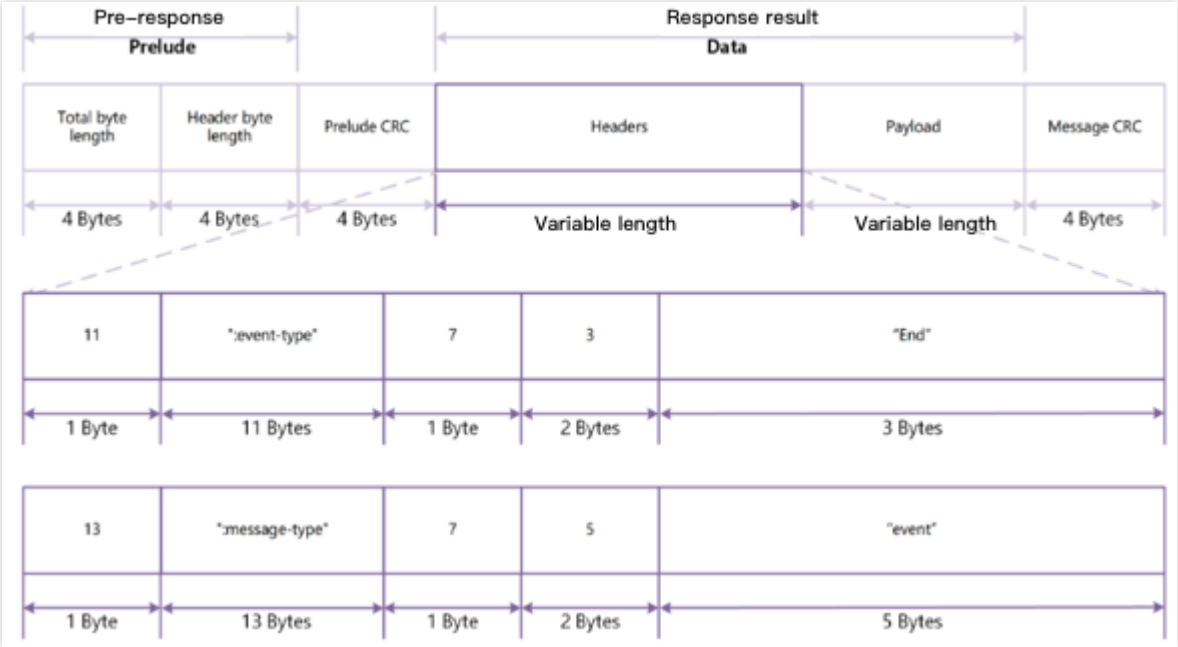


```
<?xml version="1.0" encoding="UTF-8"?>
<Stats>
     <BytesScanned>512</BytesScanned>
     <BytesProcessed>1024</BytesProcessed>
     <BytesReturned>1024</BytesReturned>
</Stats>
```

End Message

报头格式

End messages 包括":message-type"，":event-type"2种类型的报头，如下所示：



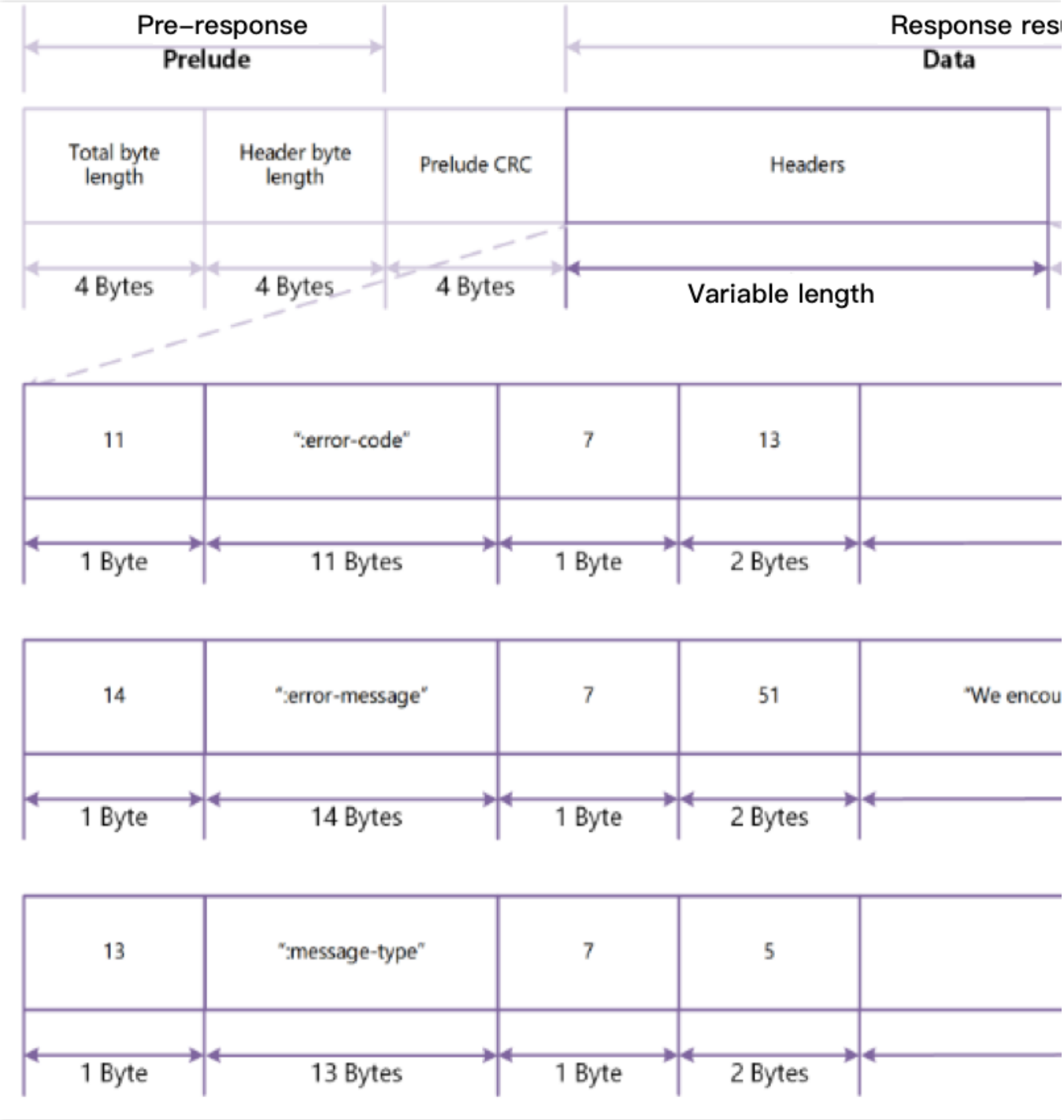
正文格式

End messages 中不包含正文内容。

Request Level Error Message

报头格式

Request Level Error Message 包括“:error-code”，“:error-message”，“:message-type”3种类型的报头，如下所示：



如果您需要了解 Request Level Error Message 中记录的错误码详情，可以查看 特殊错误码。

正文格式

Request Level Error Message 信息中不包含正文内容。

特殊错误码

该请求常见的错误信息请参见 错误码 文档，特殊错误码信息如下所示：

错误码	错误信息	含义	HTTP 状态码

InvalidXML	The XML is invalid	XML 格式不合法	400 Bad Request
MissingRequiredParameter	The SelectRequest entity is missing a required parameter	检索请求缺少必填参数项	400 Bad Request
MissingExpectedExpression	The SQL expression is missing	缺少 SQL 表达式	400 Bad Request
MissingInputSerialization	The input serialization is missing	未指定输入 CSV 对象的数据序列化格式	400 Bad Request
InvalidCompressionFormat	The file is not in a supported compression format. Only GZIP and BZIP2 are supported	不合法的文件压缩格式，仅支持 GZIP 和 BZIP2 两种格式	400 Bad Request
MissingInputFormat	The input format is missing	缺少输入格式	400 Bad Request
InvalidFileHeaderInfo	The input FileHeaderInfo is invalid. Only NONE, USE, and IGNORE are supported	输入的文件表头信息不合法。仅支持 NONE, USE 和 IGNORE	400 Bad Request
InvalidRequestParameter	The input RecordDelimiter of CSV is invalid	输入的 CSV 文件换行符不合法	400 Bad Request
InvalidRequestParameter	The input FieldDelimiter of CSV is invalid	输入的 CSV 文件列分隔符不合法	400 Bad Request
InvalidRequestParameter	The input QuoteCharacter of CSV is invalid	输入的 CSV 文件引用符不合法	400 Bad Request
InvalidRequestParameter	The input AllowQuoteRecordDelimiter of csv is invalid. Only TRUE and FALSE are supported	在输入 CSV 文件中启用转义符的配置不合法，仅支持 TRUE 和 FALSE	400 Bad Request
InvalidJsonType	The JsonType is invalid. Only DOCUMENT and LINES are supported	不合法的 JSON 类型，仅支持 DOCUMENT 和 LINES	400 Bad Request
MissingOutputSerialization	The output serialization is missing.	未指定输出 CSV 对象的数据序列格式	400 Bad Request
MissingOutputFormat	The output format is missing	缺少输出格式	400 Bad Request
InvalidQuoteFields	The QuoteFields is invalid. Only	不合法的转义规则，仅	400 Bad

	ALWAYS and ASNEEDED are supported	支持 ALWAYS 和 ASNEEDED	Request
InvalidRequestParameter	The output RecordDelimiter of CSV is invalid	输出的 CSV 文件换行符不合法	400 Bad Request
InvalidRequestParameter	The output FieldDelimiter of CSV is invalid	输出的 CSV 文件列分隔符不合法	400 Bad Request
InvalidRequestParameter	The output QuoteCharacter of CSV is invalid	输出的 CSV 文件转义符不合法	400 Bad Request
InvalidRequestParameter	The output QuoteEscapeCharacter of CSV is invalid	输出的 CSV 的双引号转义符不合法	400 Bad Request
InvalidRequestParameter	The output RecordDelimiter of JSON is invalid	输出的 JSON 文件换行符不合法	400 Bad Request
SQLParsingError	Encountered an error parsing the SQL expression	解析 SQL 表达式出现问题	400 Bad Request
SQLParsingError	Other expressions are not allowed in the SELECT list when '*' is used without dot notation.	SELECT list 不允许在未使用点符的时候使用 `'*'`	400 Bad Request
SQLParsingError	The SQL expression contains an empty SELECT	SQL 表达式中包含了空的 SELECT 子句	400 Bad Request
SQLParsingError	GROUP is not supported in the SQL expression	SQL 表达式中不支持 GROUP 子句	400 Bad Request
SQLParsingError	UNION is not supported in the SQL expression	SQL 表达式中不支持 UNION 子句	400 Bad Request
SQLParsingError	FROM is missing in the SQL expression	SQL 表达式中缺少 FROM 子句	400 Bad Request
SQLParsingError	ORDER is not supported in the SQL expression	SQL 表达式中不支持 ORDER 子句	400 Bad Request
SQLParsingError	The column index is invalid in the SQL expression	SQL 表达式中指定的列索引不合法	400 Bad Request
SQLParsingError	The table alias is invalid in WHERE	WHERE 子句中的表别名不合法	400 Bad Request
Bzip2DecompressError	Encountered an error	解压 BZIP2 格式的文件	400 Bad

版权所有：腾讯云计算（北京）有限责任公司

	decompressing the bzip2 file	时出现问题	Request
Bzip2DecompressError	BZIP2 is not applicable to the queried object	BZIP2 格式不适用于解压待查询对象	400 Bad Request
GzipDecompressError	Encountered an error decompressing the gzip file	解压 GZIP 格式的文件时出现问题	400 Bad Request
GzipDecompressError	GZIP is not applicable to the queried object	GZIP 格式不适用于解压待查询对象	400 Bad Request
Busy	The service is busy. Please retry later	后端服务阻塞，请稍后重试	400 Bad Request
Overload	The service is overload. Please retry later	后端服务过载，请稍后重试	400 Bad Request
AmbiguousFieldName	Field name matches to multiple fields in the file	指定的表头名称存在多个相同的值	400 Bad Request
ComparisonFailed	Attempt to compare failed	匹配失败，请重试	400 Bad Request
CastFailed	Attempt to convert from one data type to another using CAST failed in the SQL expression.	在 SQL 表达式中通过 CAST 函数转换数据类型时出现错误	400 Bad Request
OverMaxRecordSize	The length of a record in the input or result is greater than maxCharsPerRecord of 1 MB	输入或输出的文件中，单行记录大小超过1MB限制	400 Bad Request
LastRecordParseFail	Please check the last record in the input	请检查输入文件的最后一行记录	400 Bad Request
CSVParsingError	Encountered an error parsing the CSV file	解析 CSV 格式文件的时候出现问题	400 Bad Request
JSONParsingError	Encountered an error parsing the JSON file	解析 JSON 格式文件的时候出现问题	400 Bad Request
ErrorWritingRow	Encountered an error parsing the SELECT result. Please try again	无法格式化您的查询结果，请检查文件并重试	400 Bad Request
InvalidRequestParameter	The input Comment of CSV is invalid	不合法的 CSV 文件注释符	400 Bad Request
InvalidTextEncoding	UTF-8 encoding is required. Please check the file and try	检索文件和检索结果仅支持UTF-8编码	400 Bad Request

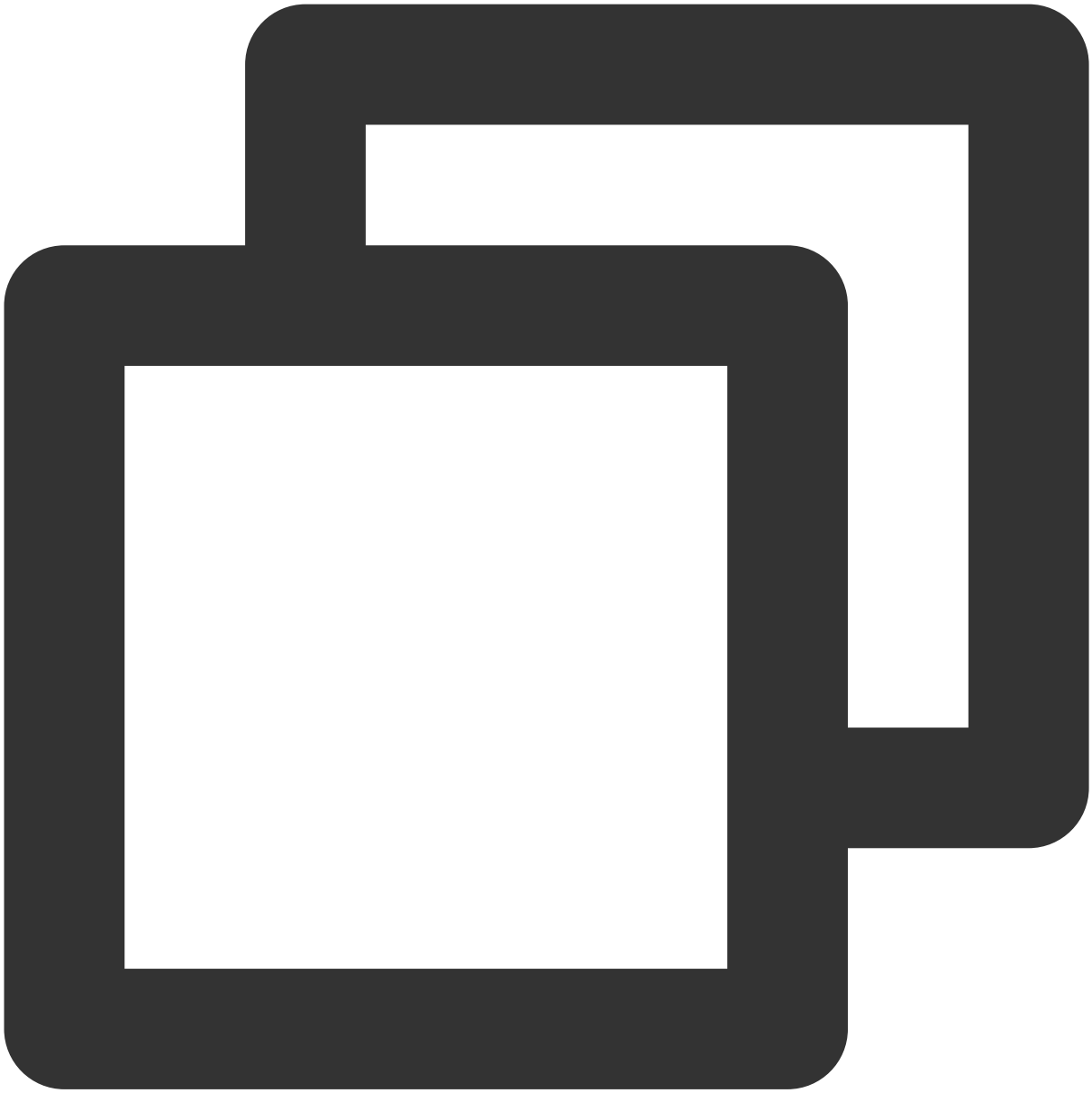
版权所有：腾讯云计算（北京）有限责任公司

	again.		
NoSuchKey	The specified key does not exist	指定的对象键不存在	404 Not Found
AccessDenied	Access Denied	签名或者权限不正确，拒绝访问	403 Forbidden
MethodNotAllowed	The specified method is not allowed against this resource	当前资源不支持该 HTTP 方法	405 Method Not Allowed
InternalError	We encountered an internal error. Please try again	服务端内部错误	500 Internal Server

示例

示例 1: 从 CSV 格式的对象中检索内容

以下示例展示了调用该接口从 CSV 格式的对象中检索全部内容，并将检索结果输出为 CSV 格式的过程。待检索的对象名为 `exampleobject.csv` ，该对象存储于北京地域（ap-beijing）的存储桶 examplebucket-1250000000 中。

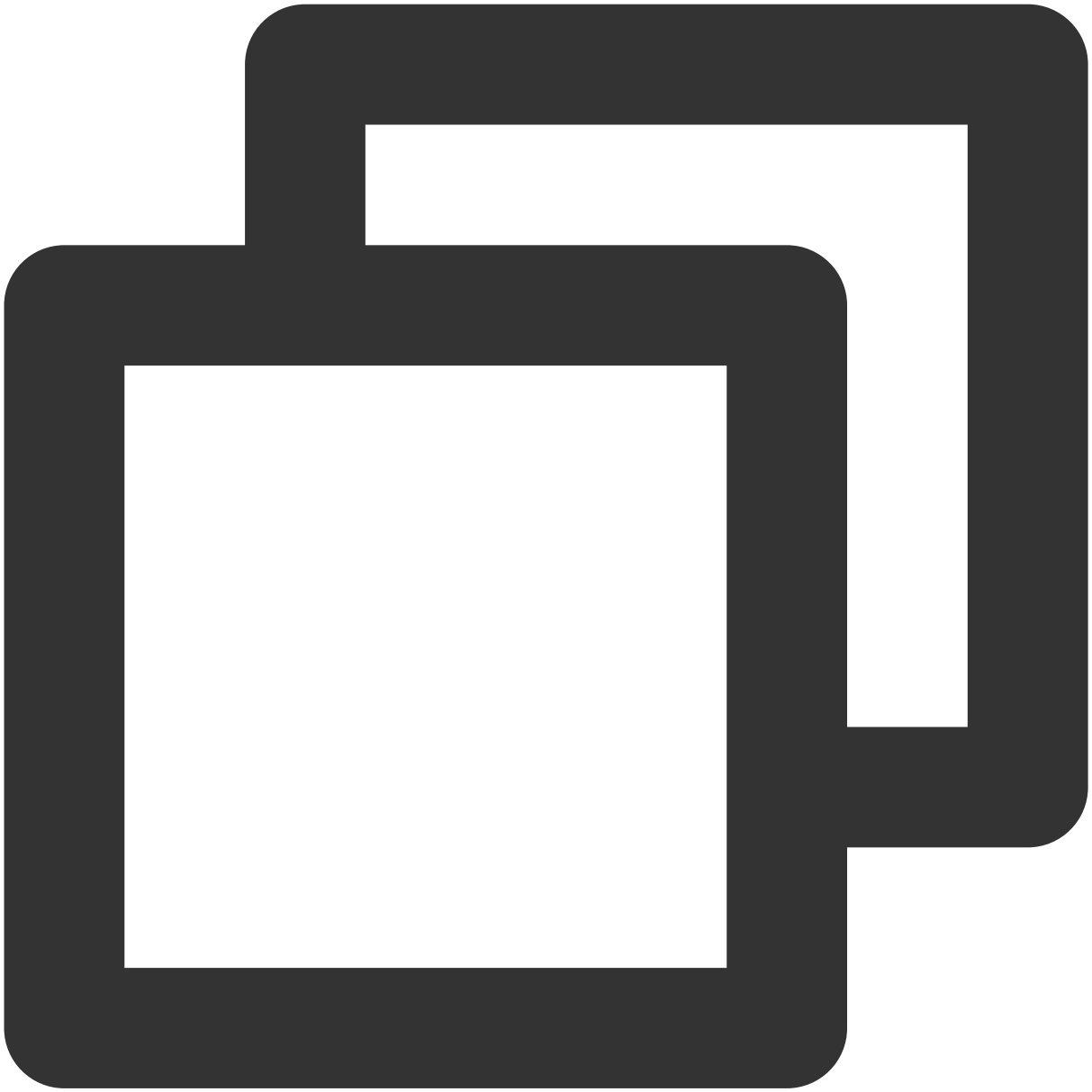


```
POST /exampleobject.csv?select&select-type=2 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 12 Jan 2019 11:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from COSObject</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
```

```
        <CompressionType>None</CompressionType>
        <CSV>
            <FileHeaderInfo>IGNORE</FileHeaderInfo>
            <RecordDelimiter>\\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
            <Comments>#</Comments>
        </CSV>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
</SelectRequest>
```

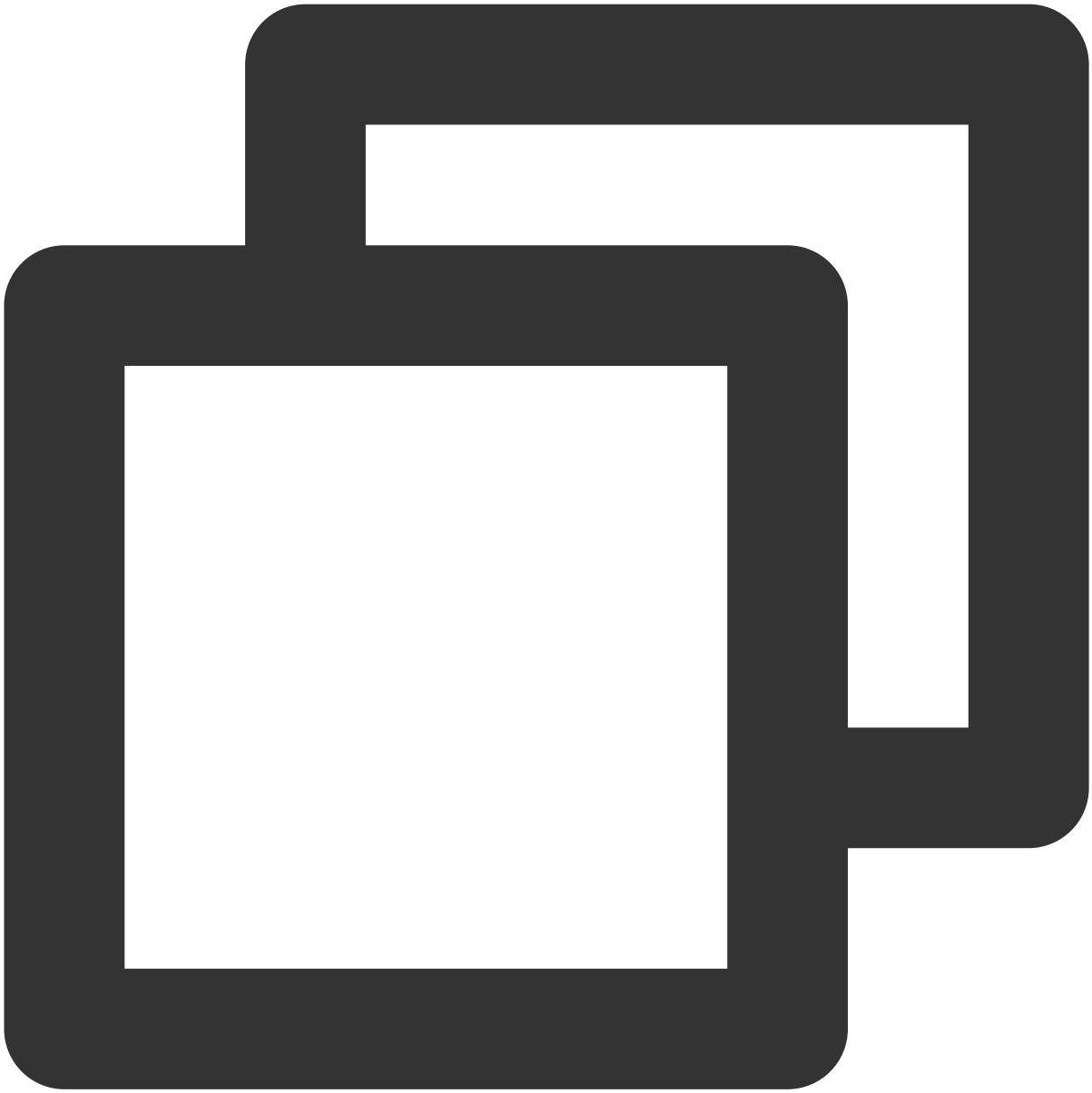
如果您需要执行不同的检索指令，可以在 Expression 元素中修改 SQL 指令，有关指令的详细介绍，请参见 Select 命令，以下为部分常见检索场景的简介。

假设您使用列索引筛选对象中的内容，您可以使用 `s._n` 筛选第 `n` 列的数据， `n` 最小为1。如下指令将从对象中筛选第3列数值大于100的记录，并返回这些记录的第1和第2列：



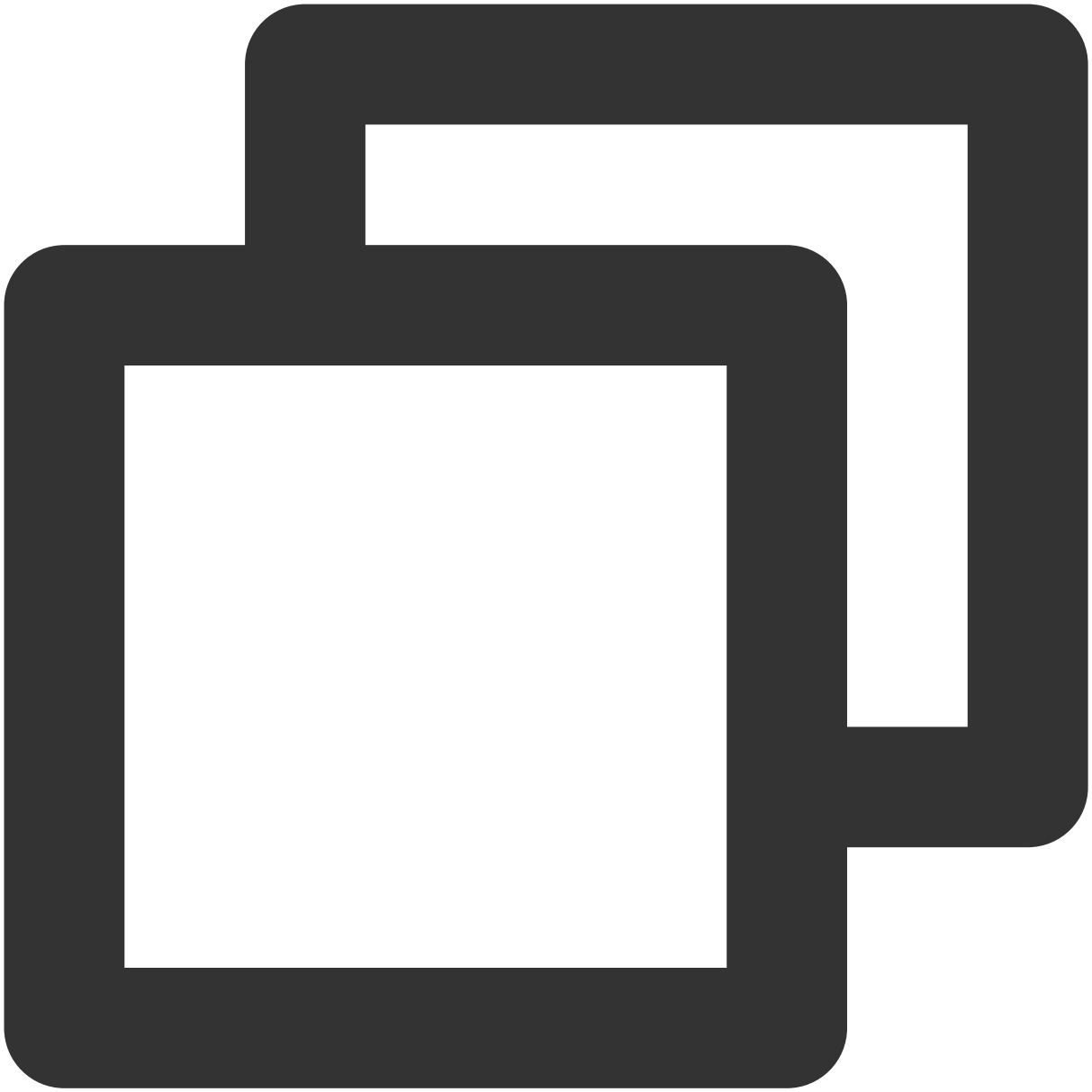
```
SELECT s._1, s._2 FROM COSObject s WHERE s._3 > 100
```

如果您的 CSV 对象中具有列表头，且您打算使用列表头的名称筛选对象中的内容（将 `FileHeaderInfo` 设置为 `Use` ），您可以使用 `s.name` 进行索引，如下指令将从对象中筛选表头名为 `Id` 和 `FirstName` 的对象：



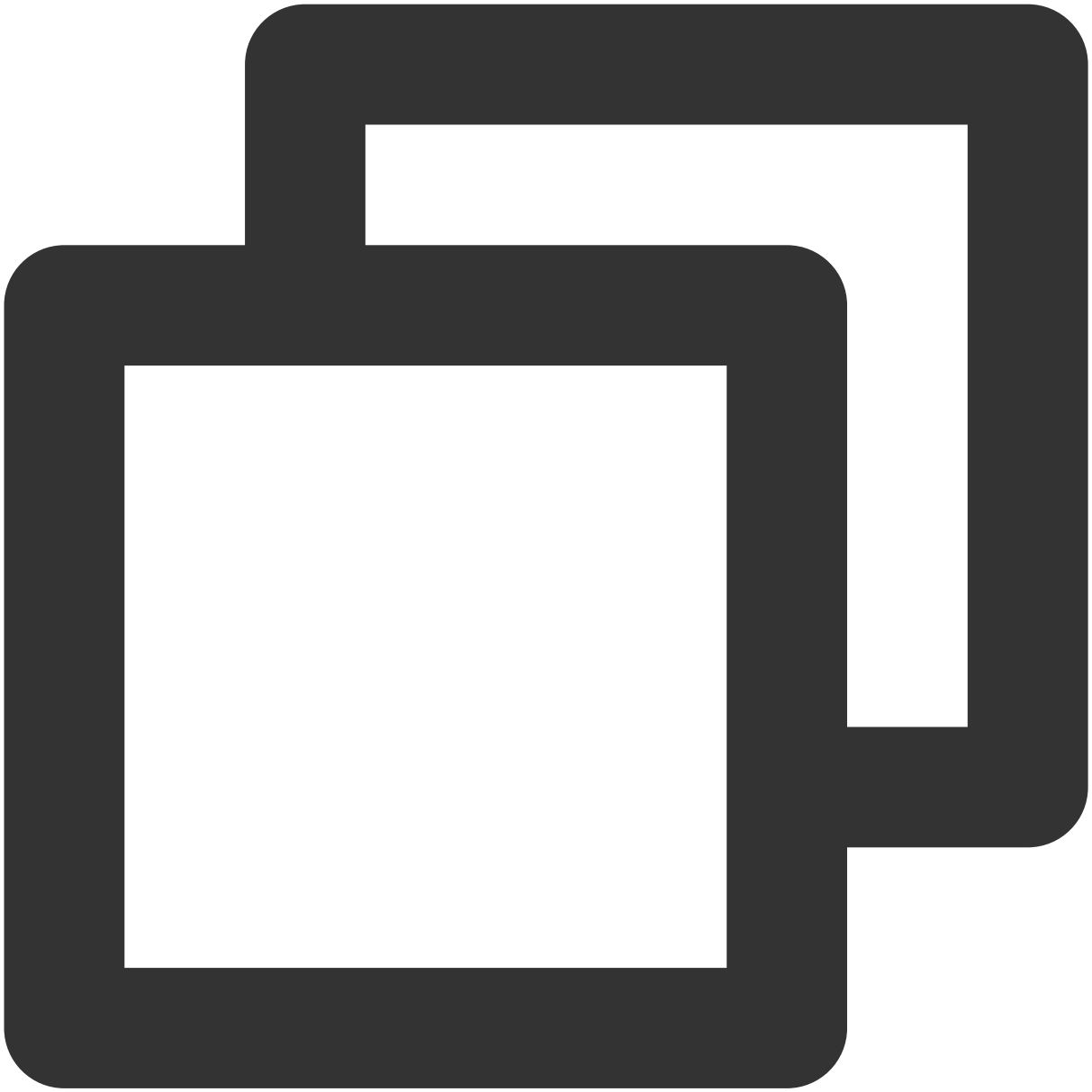
```
SELECT s.Id, s.FirstName FROM COSObject s
```

您也可以在 SQL 表达式中指定函数，如下指令将统计出第一列中小于1的记录数：



```
SELECT count(*) FROM COSObject s WHERE s._1 < 1
```

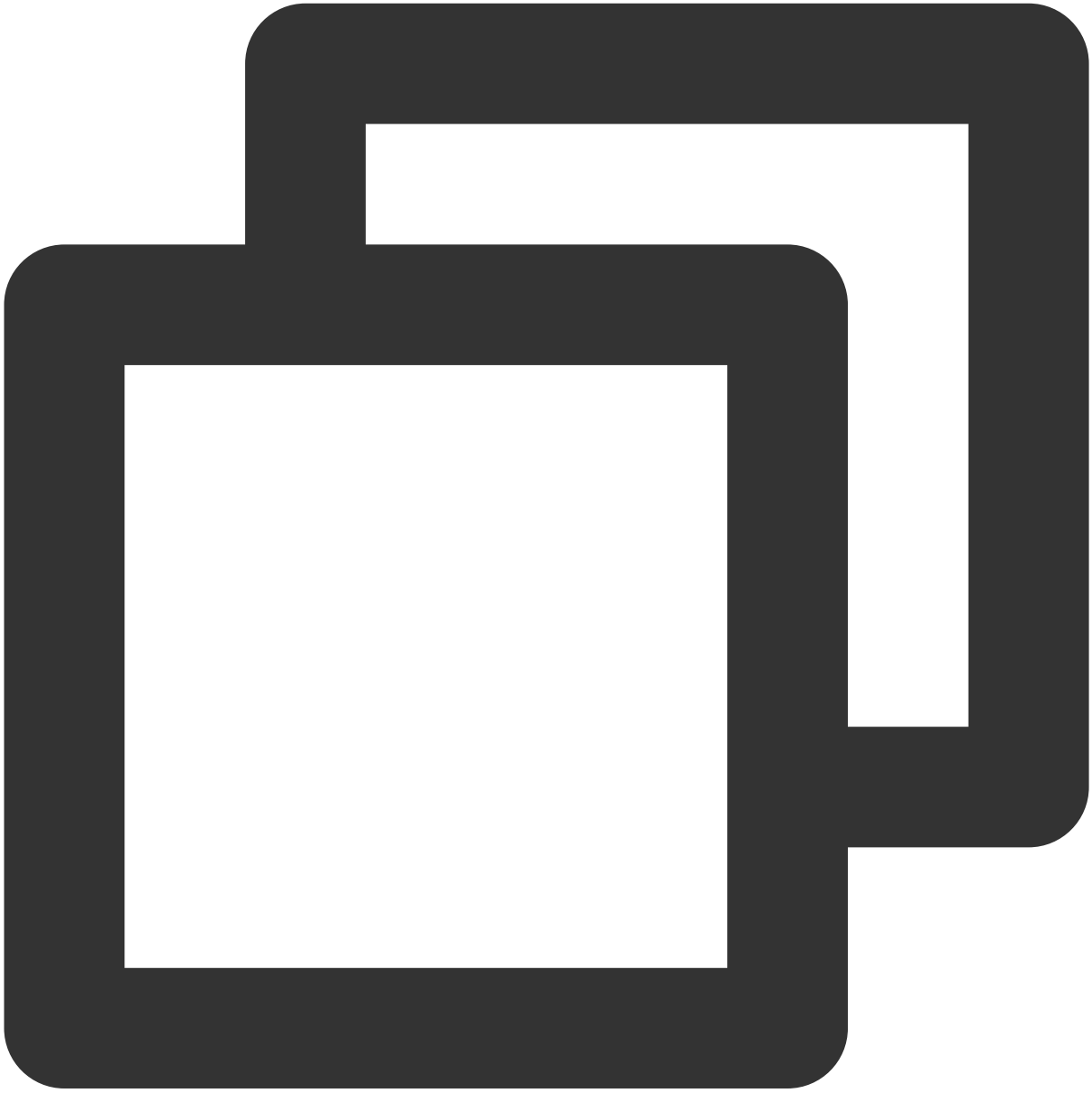
如下为响应的例子：



```
HTTP/1.1 200 OK
x-cos-id-2: cos_id_demo
x-cos-request-id: cos_request_id_demo
Date: Tue, 12 Jan 2019 11:50:29 GMT

A series of messages
```

示例2: 从 JSON 格式的对象中检索内容

以下示例展示了调用该接口从 JSON 格式的对象中检索全部内容，并将检索结果输出为 CSV 格式的过程。待检索的对象名为 `exampleobject.json` ，该对象存储于北京地域（ap-beijing）的存储桶 examplebucket-1250000000 中。

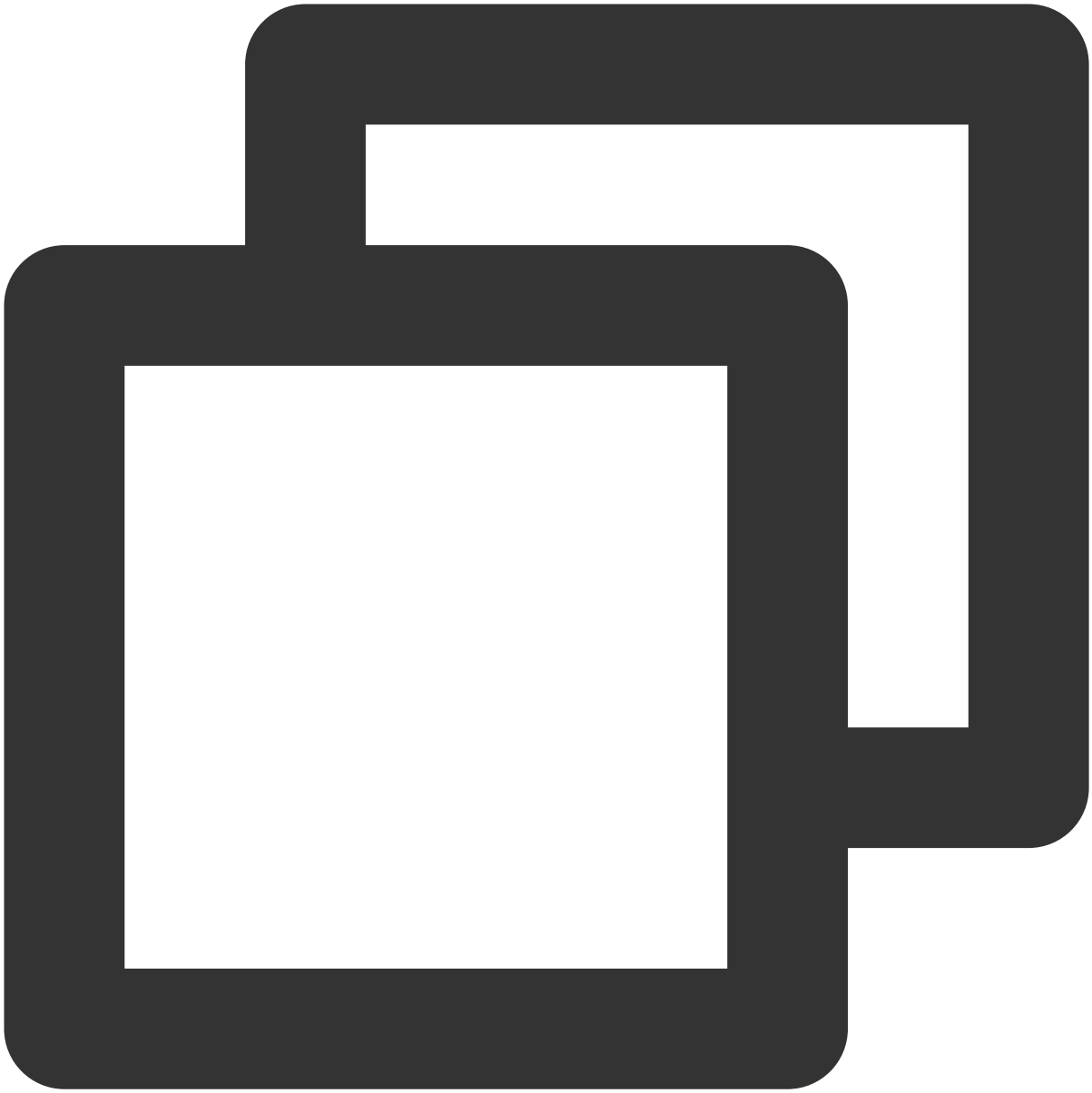


```
POST /exampleobject.json?select&select-type=2 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 12 Jan 2019 11:52:29 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
```

```
<SelectRequest>
    <Expression>Select * from COSObject</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>NONE</CompressionType>
        <JSON>
            <Type>DOCUMENT</Type>
        </JSON>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
</SelectRequest>
```

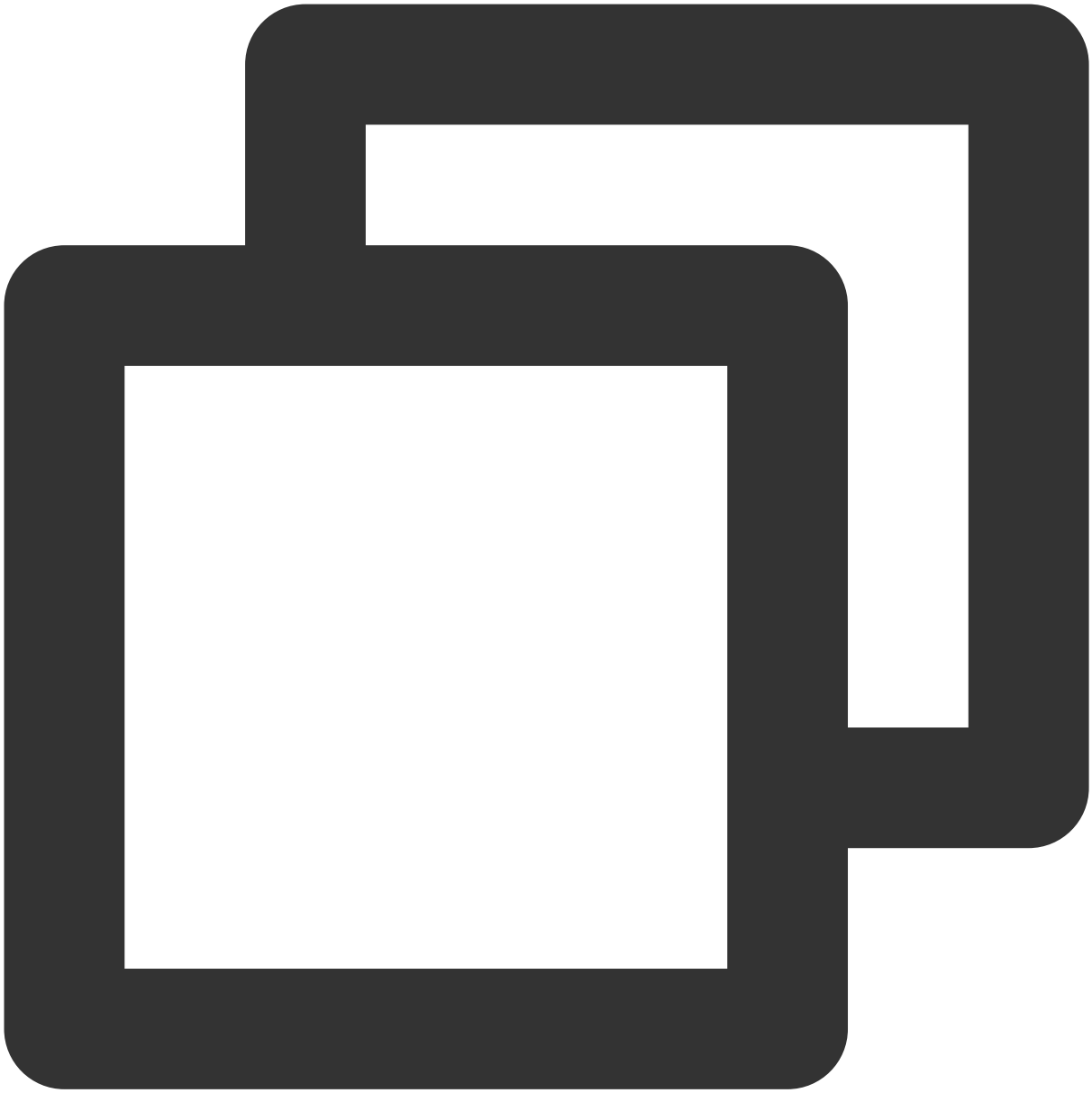
同样的，您也可以对 JSON 对象执行不同的检索指令，可以在 `Expression` 元素中修改 SQL 指令，有关指令的详细介绍，请参见 Select 命令，以下为部分常见检索场景的简介。

您可以通过 JSON 属性名称检索相应的数据，如下指令将从对象中筛选 `city` 数值为 Seattle 的记录，并返回这些记录的 `country` 和 `city` 信息：



```
SELECT s.country, s.city from COSObject s where s.city = 'Seattle'
```

您也可以在 SQL 表达式中指定函数，如下指令将统计出 JSON 对象中的记录总数：



```
SELECT count(*) FROM COSObject s
```

注意事项

与 GET Object 接口不同， SELECT Object Content 不支持以下功能：
返回对象的某一片段：您不能通过 Range 这类参数指定返回对象的某一部分。

对于归档存储（ARCHIVE）和深度归档存储（DEEP_ARCHIVE）类型的对象，COS Select 无法直接进行检索，您需要取回数据后再进行操作。

版权所有：腾讯云计算（北京）有限责任公司

访问控制（acl）

PUT Object acl

最近更新时间：2023-12-21 14:42:49

功能描述

PUT Object acl 接口用来写入对象的访问控制列表（ACL），您可以通过请求头 `x-cos-acl` 和 `x-cos-grant-*` 传入 ACL 信息，或者通过请求体以 XML 格式传入 ACL 信息。

注意

通过请求头设置 ACL 和通过请求体设置 ACL，两种方式只能选择其中一种。

PUT Object acl 是一个覆盖操作，传入新的 ACL 将覆盖原有 ACL。

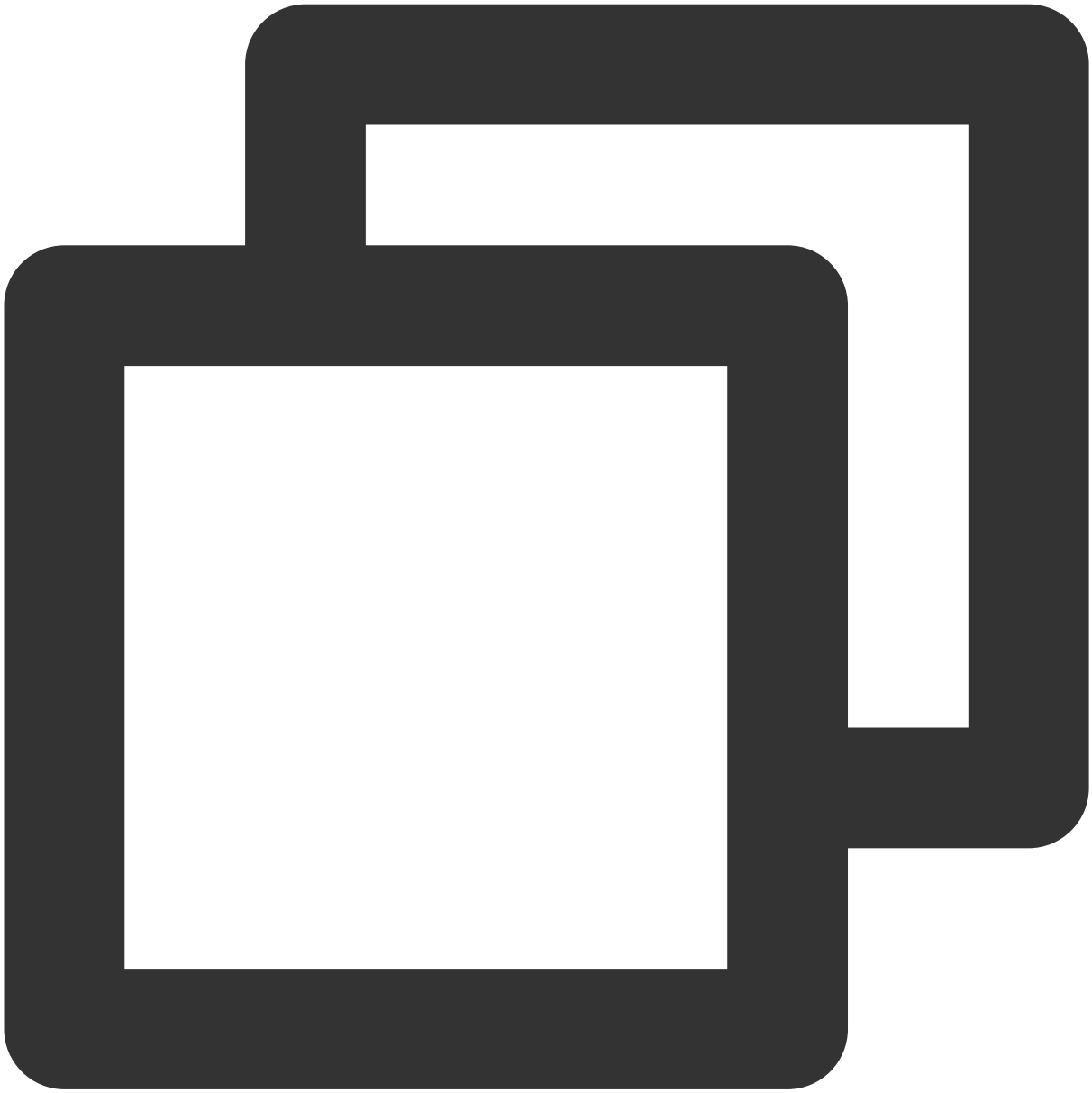
仅可对腾讯云 CAM 主账号或匿名用户授予权限，如需授予子用户或用户组权限请使用 PUT Bucket policy 接口。有关 ACL 的详细说明，请参见 ACL 概述。

该 API 的请求者需要对指定对象有写入 ACL 权限。

请求

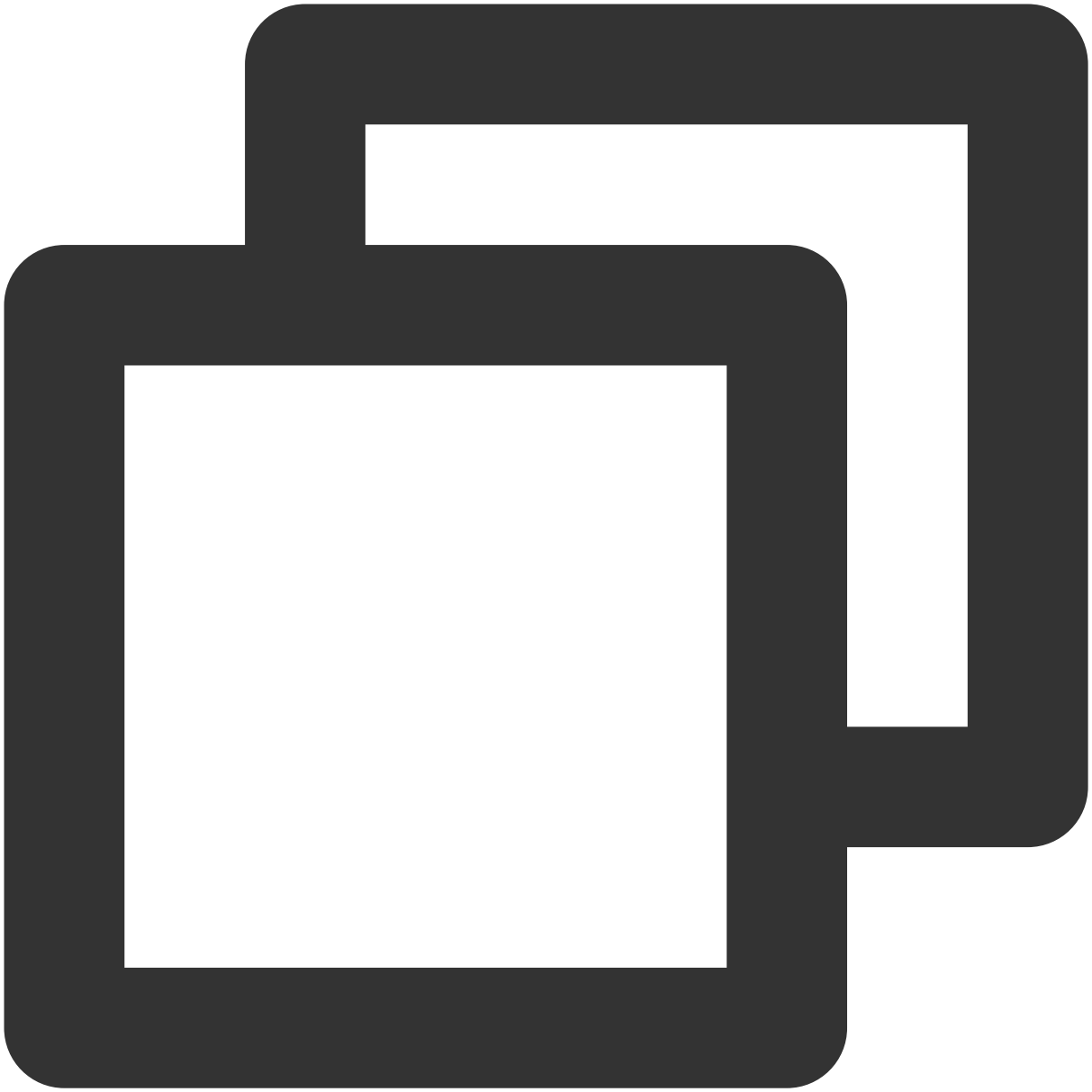
请求示例

示例一



```
PUT /<ObjectKey>?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: 0
Authorization: Auth String
```

示例二



```
PUT /<ObjectKey>?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String



[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

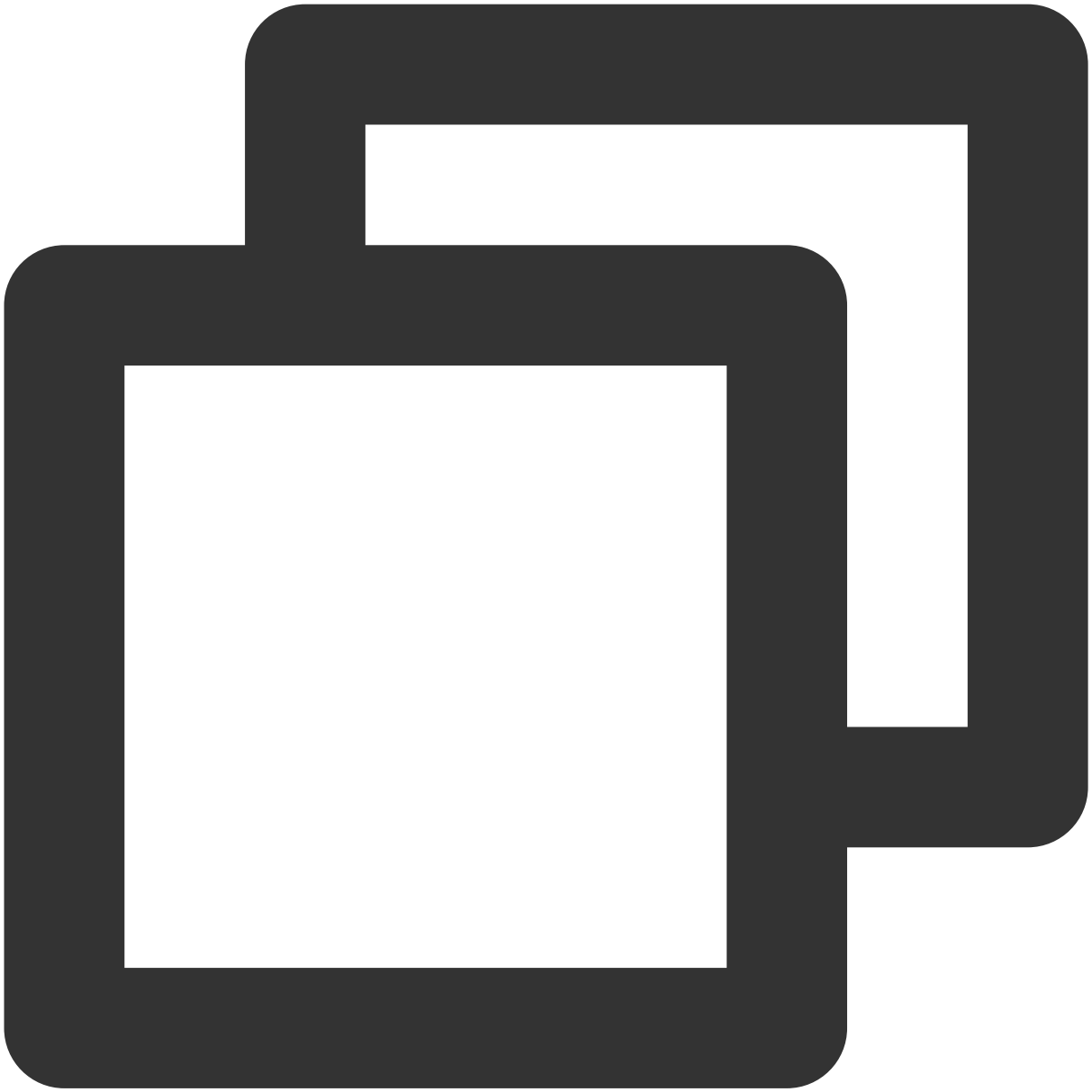
请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
x-cos-acl	定义对象的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中对象的预设 ACL 部分，例如 default，private，public-read 等，默认为 default **注意：**如果您不需要进行对象 ACL 控制，请设置为 default 或者此项不进行设置，默认继承存储桶权限	Enum	否
x-cos-grant-read	赋予被授权者读取对象的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-full-control	赋予被授权者操作对象的所有权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否

请求体

提交 **application/xml** 请求数据，包含对象所有者和完整的授权信息。



```
<AccessControlPolicy>
            <Owner>
                <ID>string</ID>
            </Owner>
            <AccessControlList>
                <Grant>
                    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        <URI>string</URI>
                    </Grantee>
                    <Permission>Enum</Permission>
                </Grant>
```

```
                <Grant>
                    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        <ID>string</ID>
                    </Grantee>
                    <Permission>Enum</Permission>
                </Grant>
            </AccessControlList>
</AccessControlPolicy>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
AccessControlPolicy	无	包含 PUT Object acl 操作的所有请求信息	Container	是

Container 节点 AccessControlPolicy 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Owner	AccessControlPolicy	对象持有者信息	Container	是
AccessControlList	AccessControlPolicy	被授权者信息与权限信息	Container	是

Container 节点 AccessControlPolicy.Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	AccessControlPolicy.Owner	对象持有者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:uin/[OwnerUin]` 例如 `qcs::cam::uin/100000000001:uin/100000000001`	strin

Container 节点 AccessControlPolicy.AccessControlList 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Grant	AccessControlPolicy.AccessControlList	单个授权信息，一个 AccessControlList 最多只能拥有100条 Grant	Container	是

Container 节点 AccessControlPolicy.AccessControlList.Grant 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Grantee	AccessControlPolicy.AccessControlList.Grant	被授权者信息，`xsi:type` 可指定为 Group 或 CanonicalUser，当指定为 Group 时子节点包括且仅允许包括 URI，当指定为 CanonicalUser 时子节点包括且仅允许包括 ID	Container	是
Permission	AccessControlPolicy.AccessControlList.Grant	授予的权限信息，枚举值请参见 ACL 概述 文档中对象的操作部分，例如 READ，FULL_CONTROL 等	Enum	是

Container 节点 AccessControlPolicy.AccessControlList.Grant.Grantee 的内容：

节点名称（关键字）	父节点	描述
URI	AccessControlPolicy.AccessControlList.Grant.Grantee	预设用户组，请参见 ACL 概述 文档中预设用户 如 `http://cam.qcloud.com/groups/gl`
ID	AccessControlPolicy.AccessControlList.Grant.Grantee	被授权者的完整 ID，格式为 `qcs::cam::uin` 如 `qcs::cam::uin/100000000001:uin/1`

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口响应体为空。

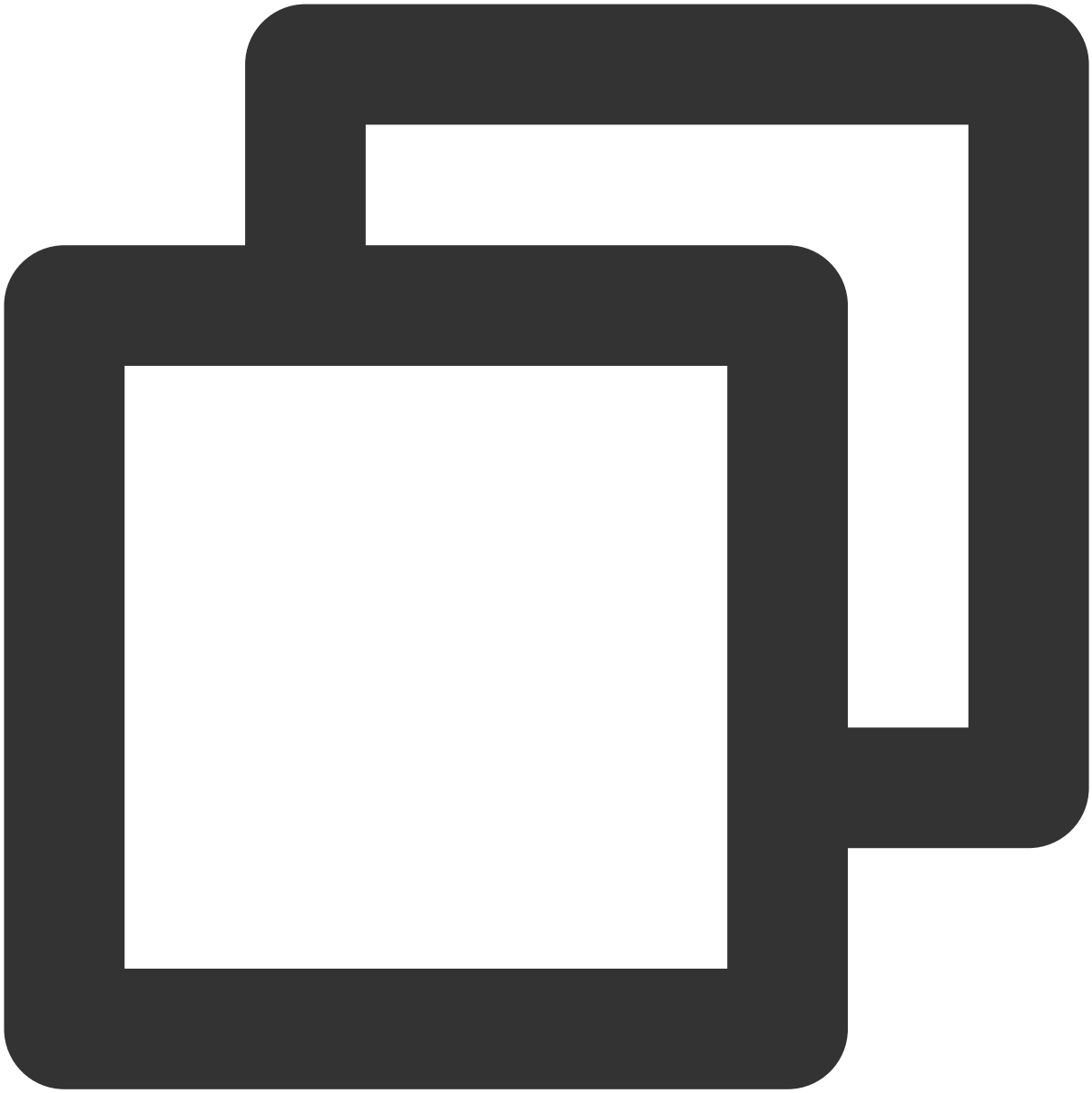
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

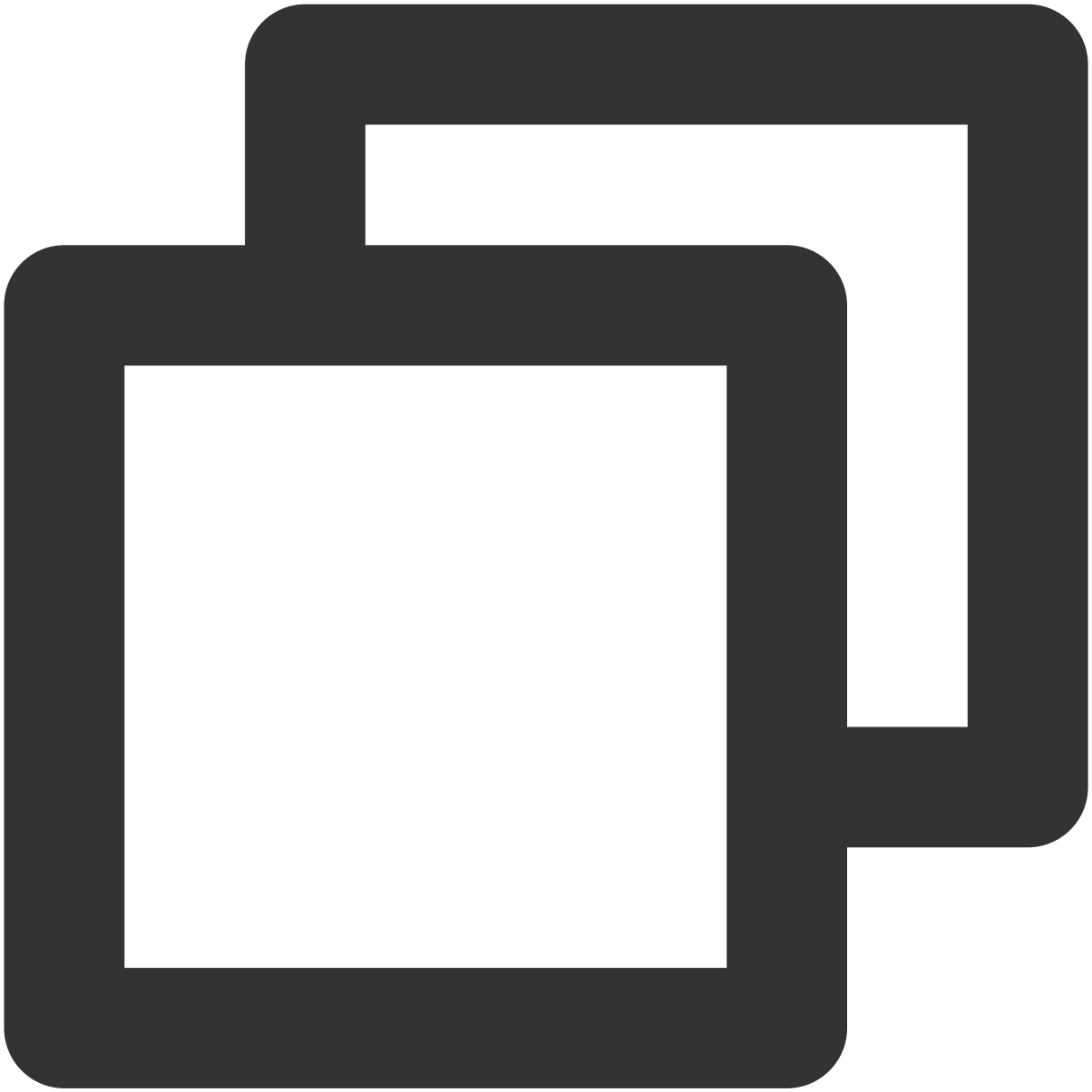
案例一：通过请求头设置 ACL

请求



```
PUT /exampleobject?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 09 Sep 2019 13:11:09 GMT
x-cos-acl: public-read
x-cos-grant-read-acp: id="100000000002"
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

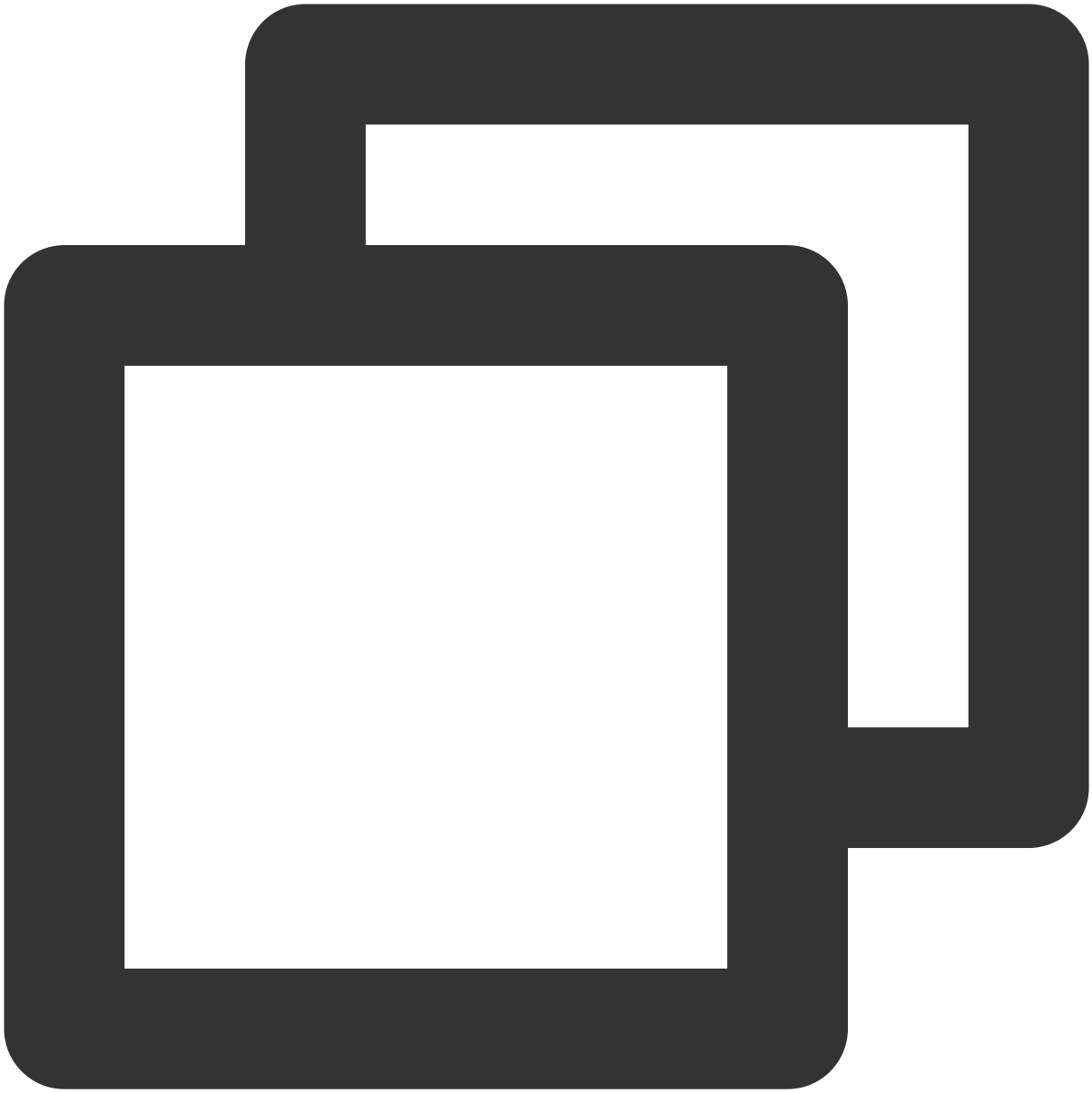
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Mon, 09 Sep 2019 13:11:10 GMT
Server: tencent-cos
x-cos-request-id: NWQ3NjRmNmRfZjZjMjBiMDlfMmE5MWJfMTI3OWZh****
```

案例二：通过请求体设置 ACL

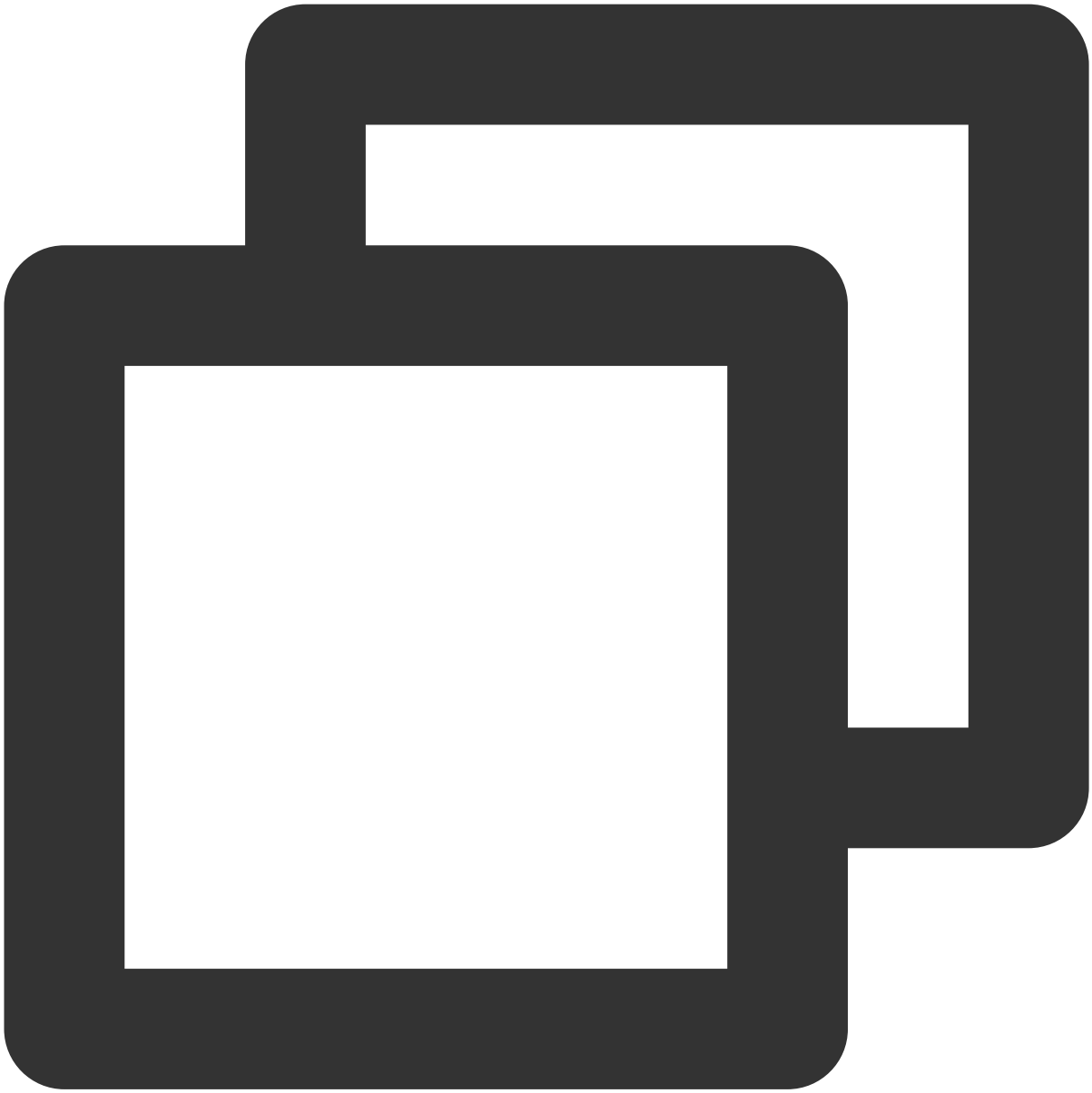
请求



```
PUT /exampleobject?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 10 Sep 2019 06:32:02 GMT
Content-Type: application/xml
Content-Length: 594
Content-MD5: zUPEBc1TeGrqTqEfPV7rxg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

```
<AccessControlPolicy>
            <Owner>
                <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
            </Owner>
            <AccessControlList>
                <Grant>
                    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
                    </Grantee>
                    <Permission>READ</Permission>
                </Grant>
                <Grant>
                    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
                    </Grantee>
                    <Permission>READ_ACP</Permission>
                </Grant>
            </AccessControlList>
</AccessControlPolicy>
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Tue, 10 Sep 2019 06:32:02 GMT
Server: tencent-cos
x-cos-request-id: NWQ3NzQzNjJfZmVhODBiMDlfMjc5MGVfMTM4OTky****
```

GET Object acl

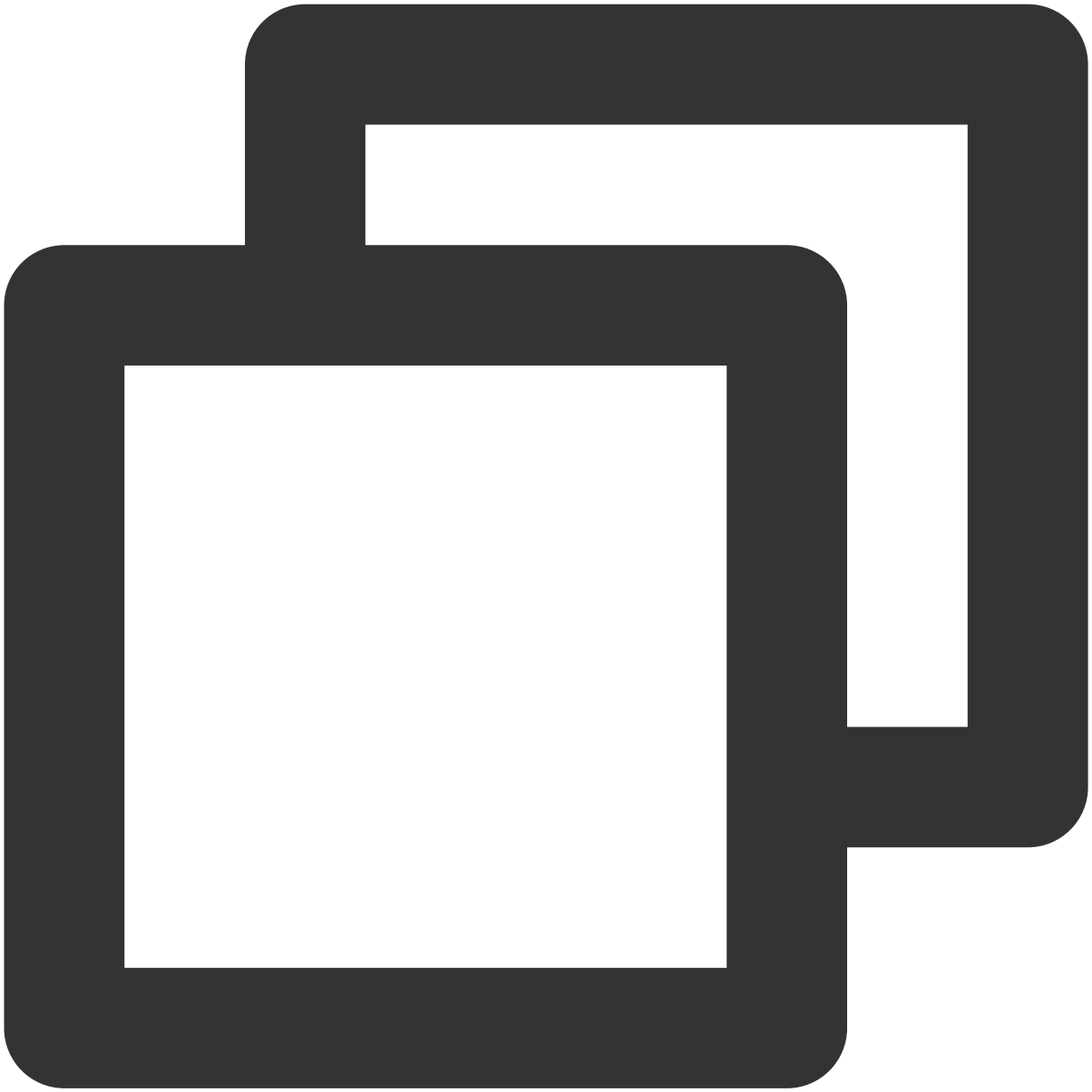
最近更新时间：2023-12-21 14:42:41

功能描述

GET Object acl 接口用来获取对象的访问控制列表（ACL）。该 API 的请求者需要对指定对象有读取 ACL 权限。

请求

请求示例



```
GET /<ObjectKey>?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

此接口无请求参数。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

此接口无请求体。

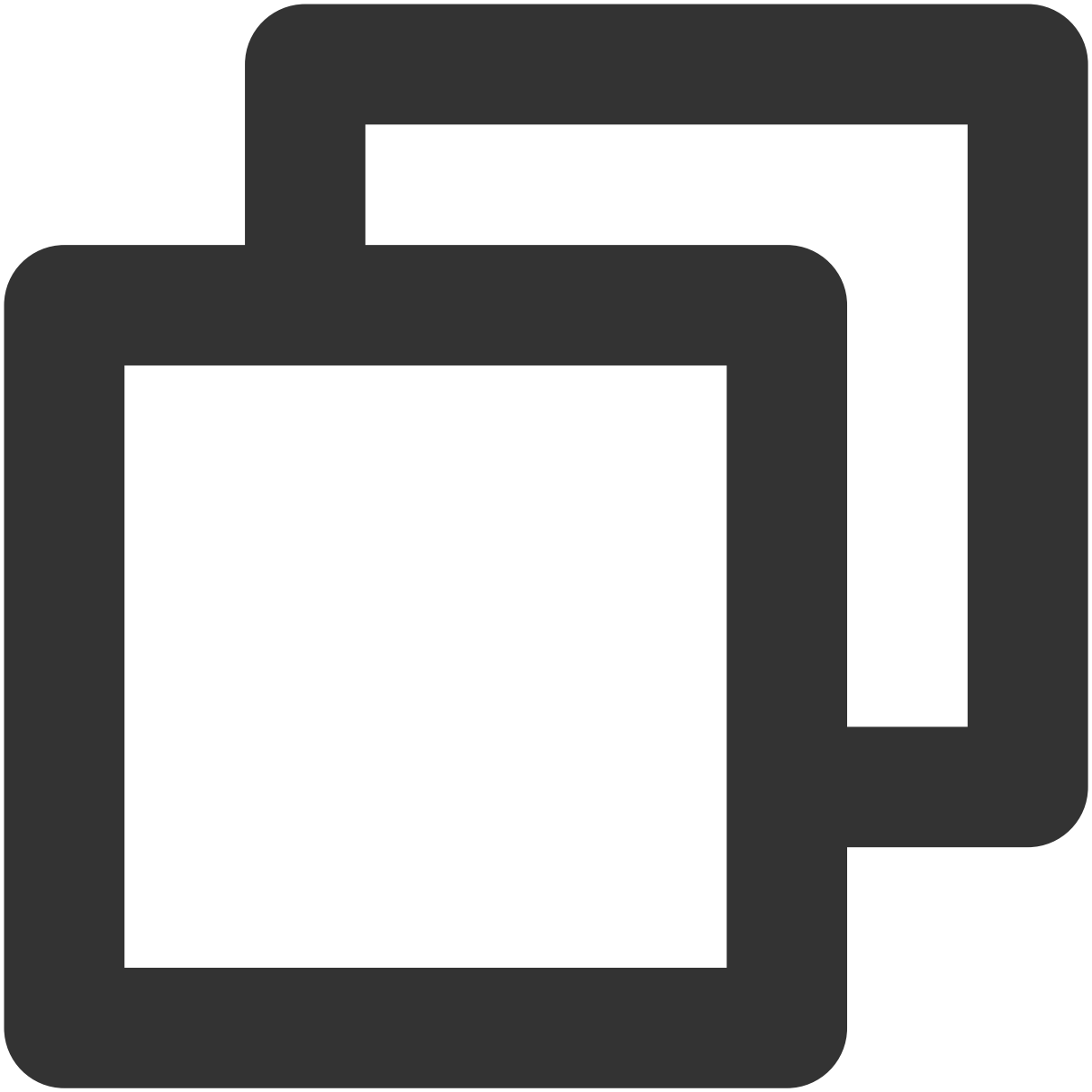
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含对象所有者和完整的授权信息。



```
<AccessControlPolicy>
    <Owner>
        <ID>string</ID>
        <DisplayName>string</DisplayName>
    </Owner>
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <URI>string</URI>
            </Grantee>
            <Permission>Enum</Permission>
```

```
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>string</ID>
                <DisplayName>string</DisplayName>
            </Grantee>
            <Permission>Enum</Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
AccessControlPolicy	无	保存 GET Object acl 结果的所有信息	Container

Container 节点 AccessControlPolicy 的内容：

节点名称（关键字）	父节点	描述	类型
Owner	AccessControlPolicy	对象持有者信息	Container
AccessControlList	AccessControlPolicy	被授权者信息与权限信息	Container

Container 节点 AccessControlPolicy.Owner 的内容：

节点名称（关键字）	父节点	描述
ID	AccessControlPolicy.Owner	对象持有者的完整 ID，格式为 `qcs::cam::uin/[OwnerUin]:uin/[OwnerUin]` 例如 `qcs::cam::uin/100000000001:uin/100000000001`
DisplayName	AccessControlPolicy.Owner	对象持有者的名字

Container 节点 AccessControlPolicy.AccessControlList 的内容：

节点名称（关键字）	父节点	描述	类型
Grant	AccessControlPolicy.AccessControlList	单个授权信息	Container

Container 节点 AccessControlPolicy.AccessControlList.Grant 的内容：

节点名称（关键	父节点	描述	类型

字）			
Grantee	AccessControlPolicy.AccessControlList.Grant	被授权者信息，`xsi:type` 为 Group 或 CanonicalUser，当为 Group 时子节点包括且仅包括 URI，当指定为 CanonicalUser 时子节点包括且仅包括 ID 和 DisplayName	Container
Permission	AccessControlPolicy.AccessControlList.Grant	授予的权限信息，枚举值请参见 ACL 概述 文档中对象的操作部分，例如 READ，FULL_CONTROL 等	Enum

Container 节点 AccessControlPolicy.AccessControlList.Grant.Grantee 的内容：

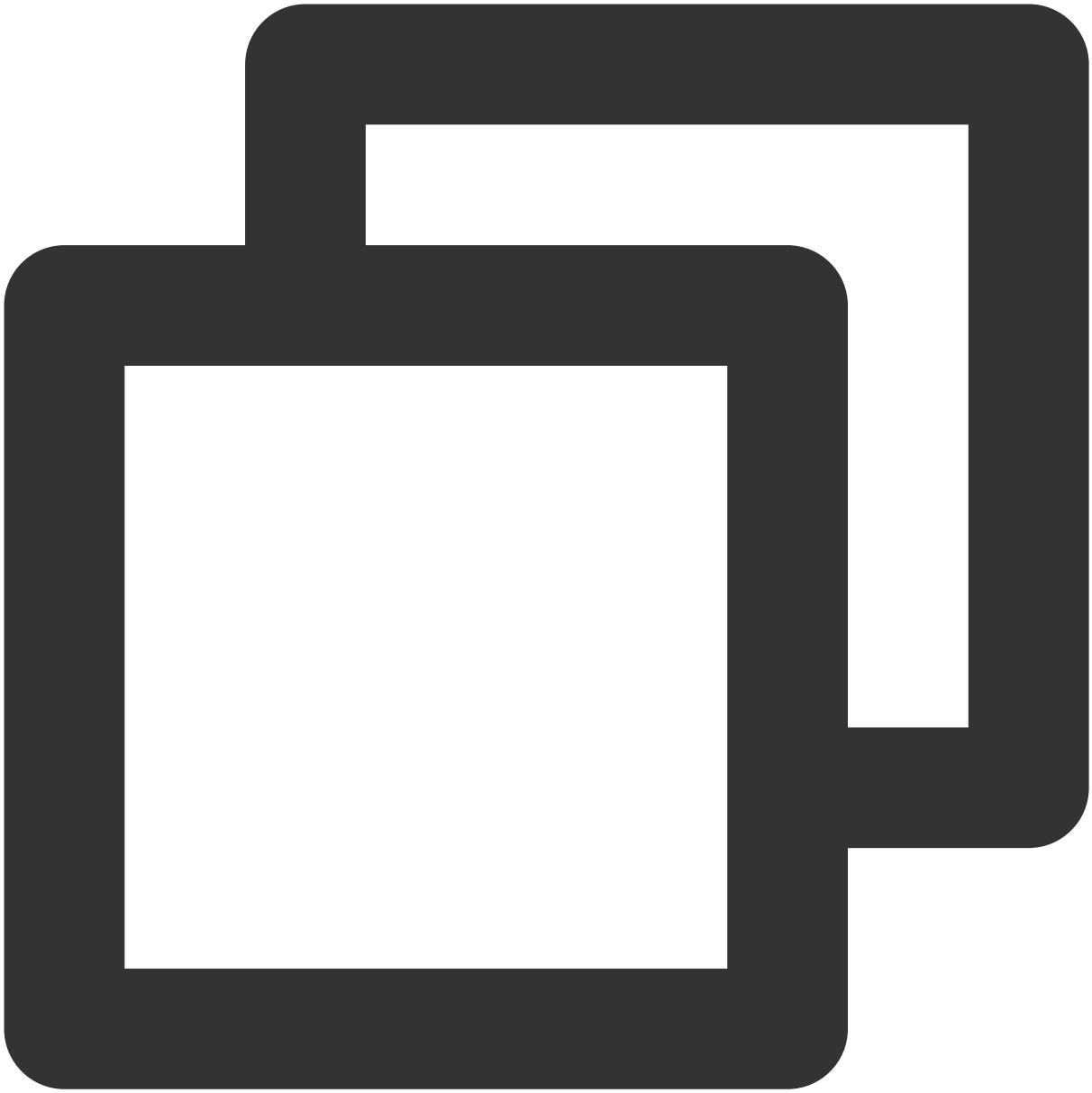
节点名称（关键字）	父节点	描述
URI	AccessControlPolicy.AccessControlList.Grant.Grantee	预设用户组，请参见 ACL 概述 文档中例 如 `http://cam.qcloud.com/gro`
ID	AccessControlPolicy.AccessControlList.Grant.Grantee	被授权者的完整 ID，格式为 `qcs::c` 例如 `qcs::cam::uin/100000000`
DisplayName	AccessControlPolicy.AccessControlList.Grant.Grantee	被授权者的名字

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

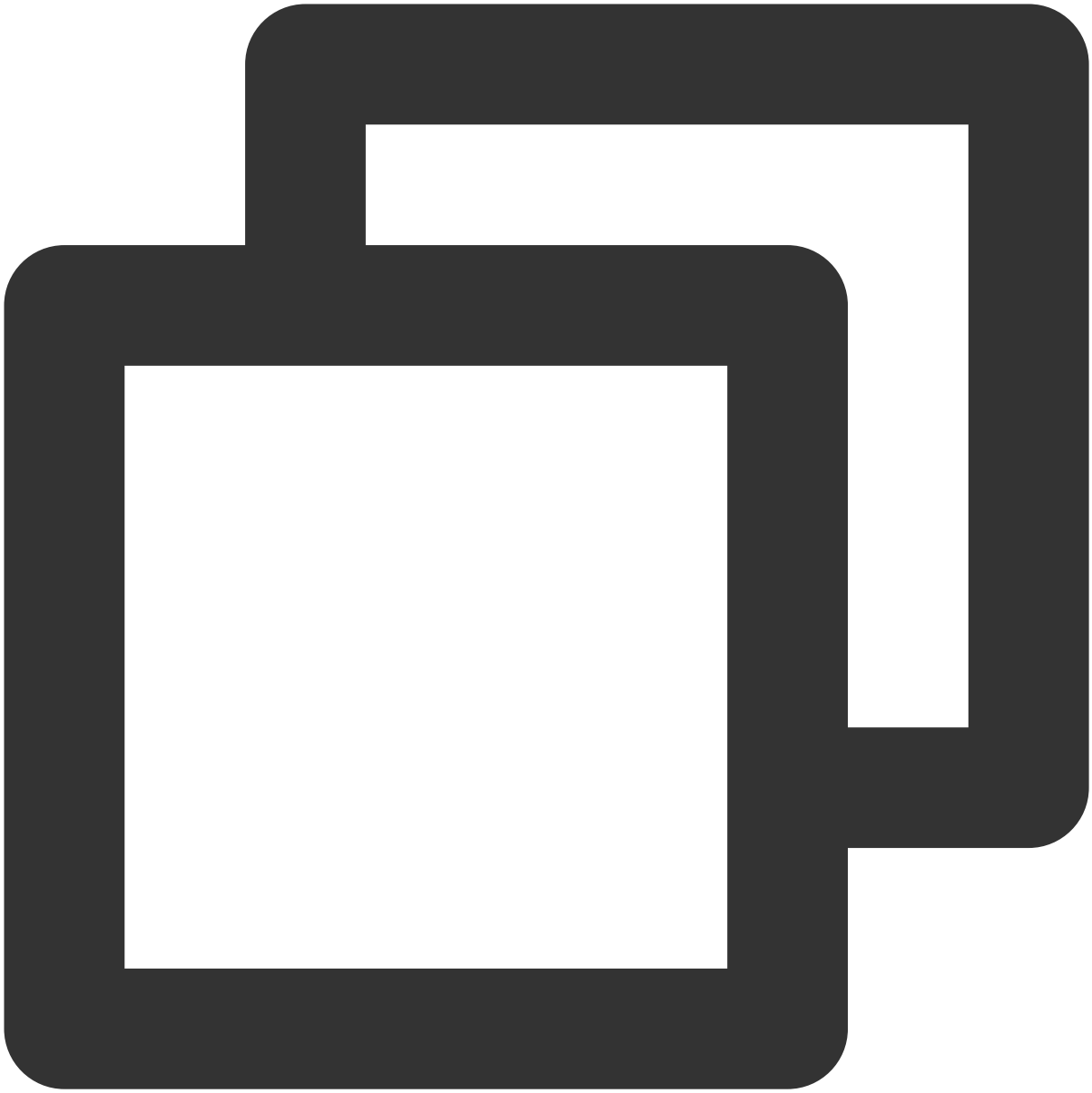
实际案例

请求



```
GET /exampleobject?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 10 Sep 2019 08:29:26 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 742
Connection: close
Date: Tue, 10 Sep 2019 08:29:26 GMT
Server: tencent-cos
x-cos-request-id: NWQ3NzVlZTZfYmIwMmEwOV83YTQ5XzEzNTcx****

<AccessControlPolicy>
    <Owner>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
```

```
        <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
    </Owner>
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
            </Grantee>
            <Permission>READ</Permission>
        </Grant>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
                <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
                <DisplayName>qcs::cam::uin/100000000002:uin/100000000002</DisplayNa
            </Grantee>
            <Permission>READ_ACP</Permission>
        </Grant>
    </AccessControlList>
</AccessControlPolicy>
```

分块上传
Initiate Multipart Upload

最近更新时间：2023-12-21 15:05:35

功能描述

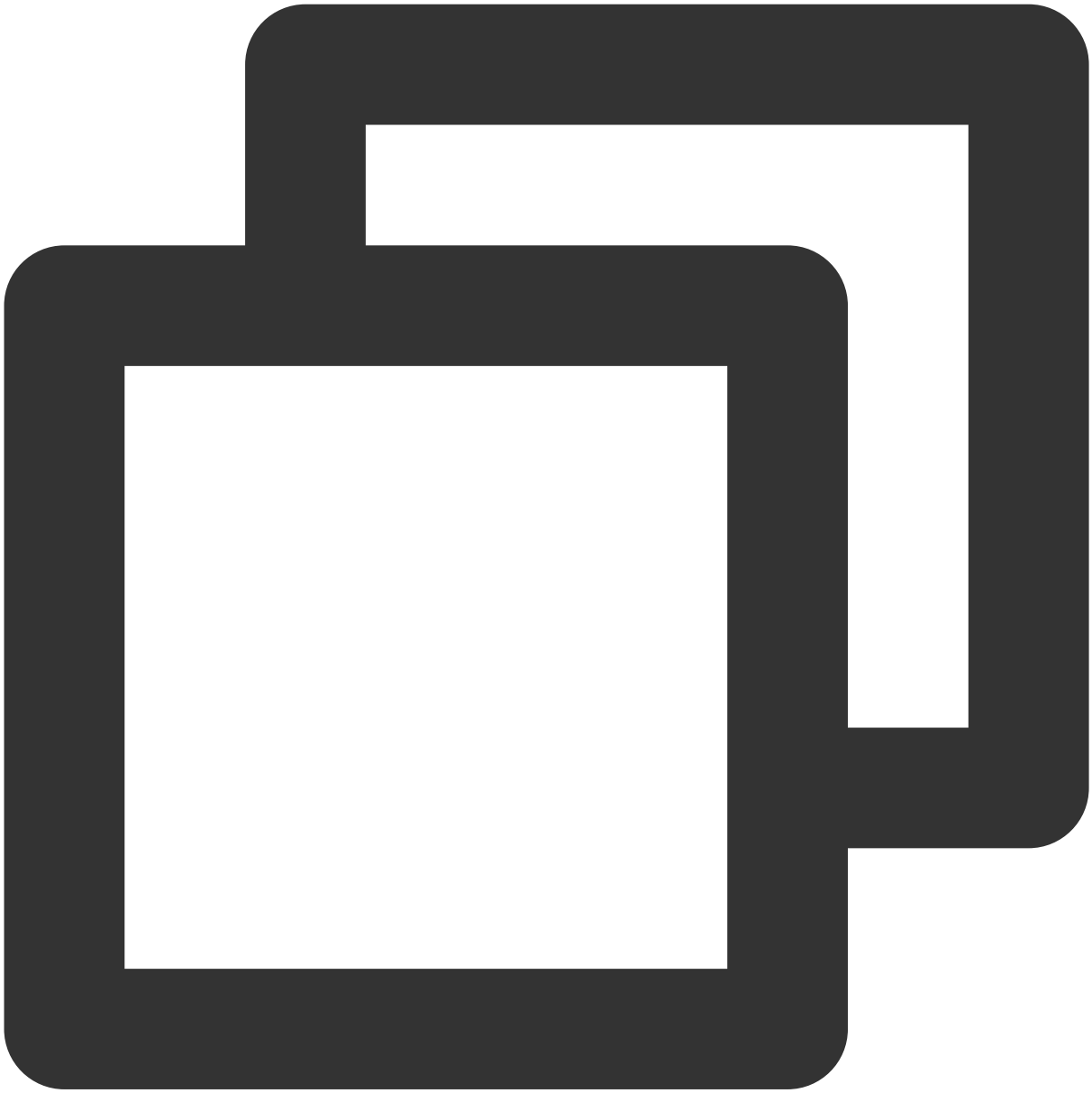
Initiate Multipart Upload 接口请求实现初始化分块上传，成功执行此请求后将返回 UploadId，用于后续的 Upload Part 请求。

说明

只有主账号或者被授权 Initiate Multipart Upload 接口权限的子账号可以调用该接口。

请求

请求示例



```
POST /<ObjectKey>?uploads HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: Content Type
Content-Length: 0
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；

<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

此接口无请求参数。

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

名称	描述	类型	是否必选
Cache-Control	RFC 2616 中定义的缓存指令，将作为对象元数据保存	string	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为对象元数据保存	string	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为对象元数据保存	string	否
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型（MIME），此头部用于描述待上传对象的内容类型，将作为对象元数据保存。 例如 `text/html` 或 `image/jpeg`	string	是
Expires	RFC 2616 中定义的缓存失效时间，将作为对象元数据保存	string	否
x-cos-meta-*	包括用户自定义元数据头部后缀和用户自定义元数据信息，将作为对象元数据保存，大小限制为2KB **注意：**用户自定义元数据信息支持下划线（_），但用户自定义元数据头部后缀不支持下划线，仅支持减号（-）	string	否
x-cos-storage-class	对象存储类型。枚举值请参见 存储类型 文档，例如 MAZ_STANDARD，MAZ_STANDARD_IA，INTELLIGENT_TIERING，MAZ_INTELLIGENT_TIERING，STANDARD_IA，ARCHIVE，DEEP_ARCHIVE。默认值：STANDARD	Enum	否
x-cos-tagging	对象的标签集合，最多可设置10个标签（例如，Key1=Value1&Key2=Value2）。 标签集合中的 Key 和 Value 必须先进行 URL 编码。	string	否

访问控制列表（ACL）相关头部

在上传对象时可以通过指定下列请求头部来设置对象的访问权限：

名称	描述	类型	是否必选

x-cos-acl	定义对象的访问控制列表（ACL）属性。枚举值请参见 ACL 概述 文档中对象的预设 ACL 部分，例如 default，private，public-read 等，默认为 default。 **注意：**如果您不需要进行对象 ACL 控制，请设置为 default 或者此项不进行设置，默认继承存储桶权限	Enum	否
x-cos-grant-read	赋予被授权者读取对象的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-read-acp	赋予被授权者读取对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-write-acp	赋予被授权者写入对象的访问控制列表（ACL）的权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否
x-cos-grant-full-control	赋予被授权者操作对象的所有权限，格式为 id="[OwnerUin]"，例如 id="100000000001"，可使用半角逗号（,）分隔多组被授权者，例如 `id="100000000001",id="100000000002"`	string	否

服务端加密（SSE）相关头部

在上传对象时可以使用服务端加密，请参见 服务端加密专用头部。

请求体

此接口无请求体。

响应

响应头

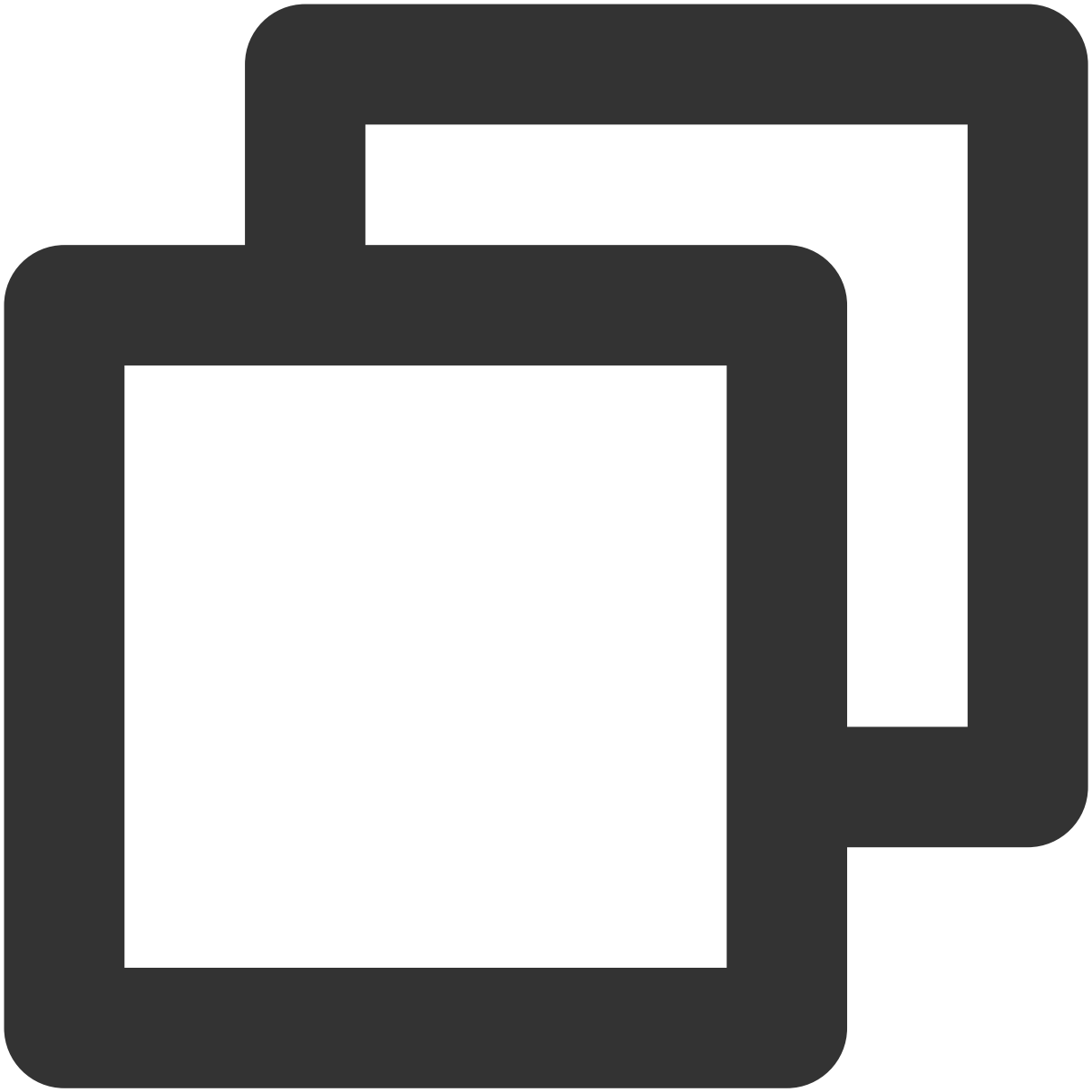
此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

服务端加密（SSE）相关头部

如果在上传对象时使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

请求成功，返回 **application/xml** 数据，包含分块上传初始化信息。



```
<InitiateMultipartUploadResult>
       <Bucket>string</Bucket>
       <Key>string</Key>
       <UploadId>string</UploadId>
</InitiateMultipartUploadResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
InitiateMultipartUploadResult	无	保存 Initiate Multipart Upload 结果的所有信息	Container

Container 节点 InitiateMultipartUploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	InitiateMultipartUploadResult	目标存储桶	string
Key	InitiateMultipartUploadResult	目标对象键	string
UploadId	InitiateMultipartUploadResult	在后续 Upload Part 中使用的 UploadId	string

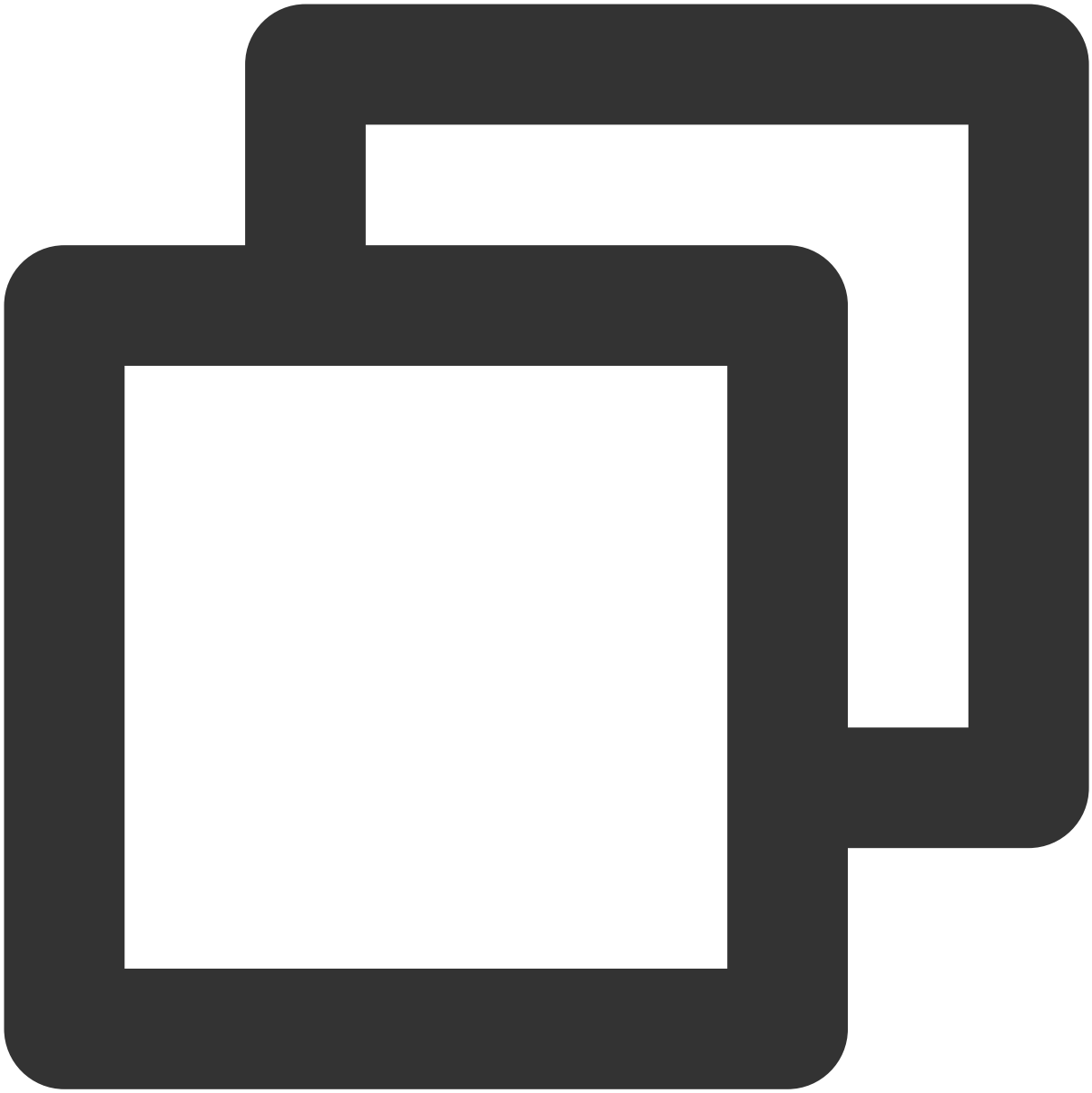
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

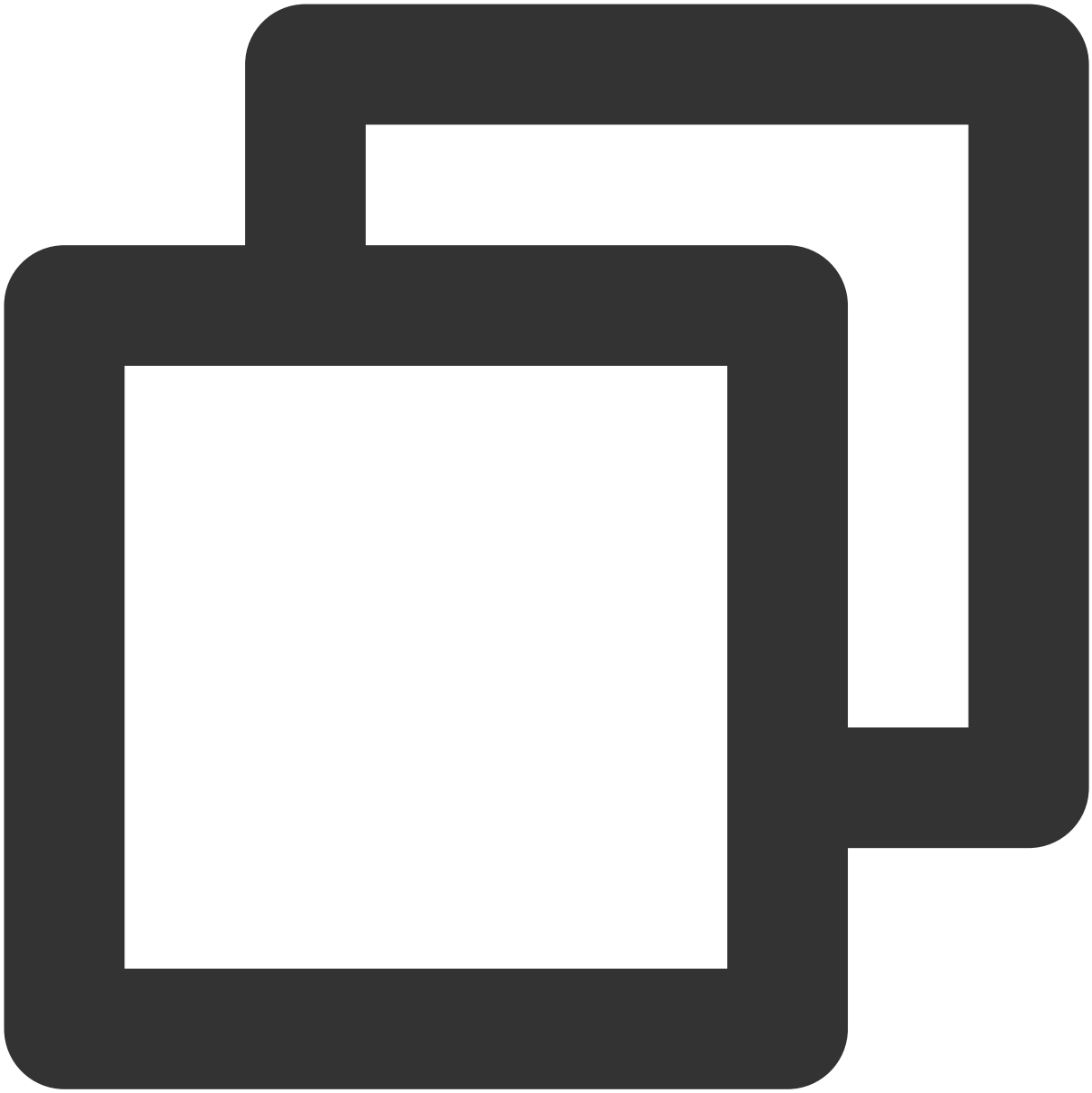
案例一：简单案例

请求



```
POST /exampleobject?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 25 Mar 2020 10:07:01 GMT
Content-Type: video/mp4
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

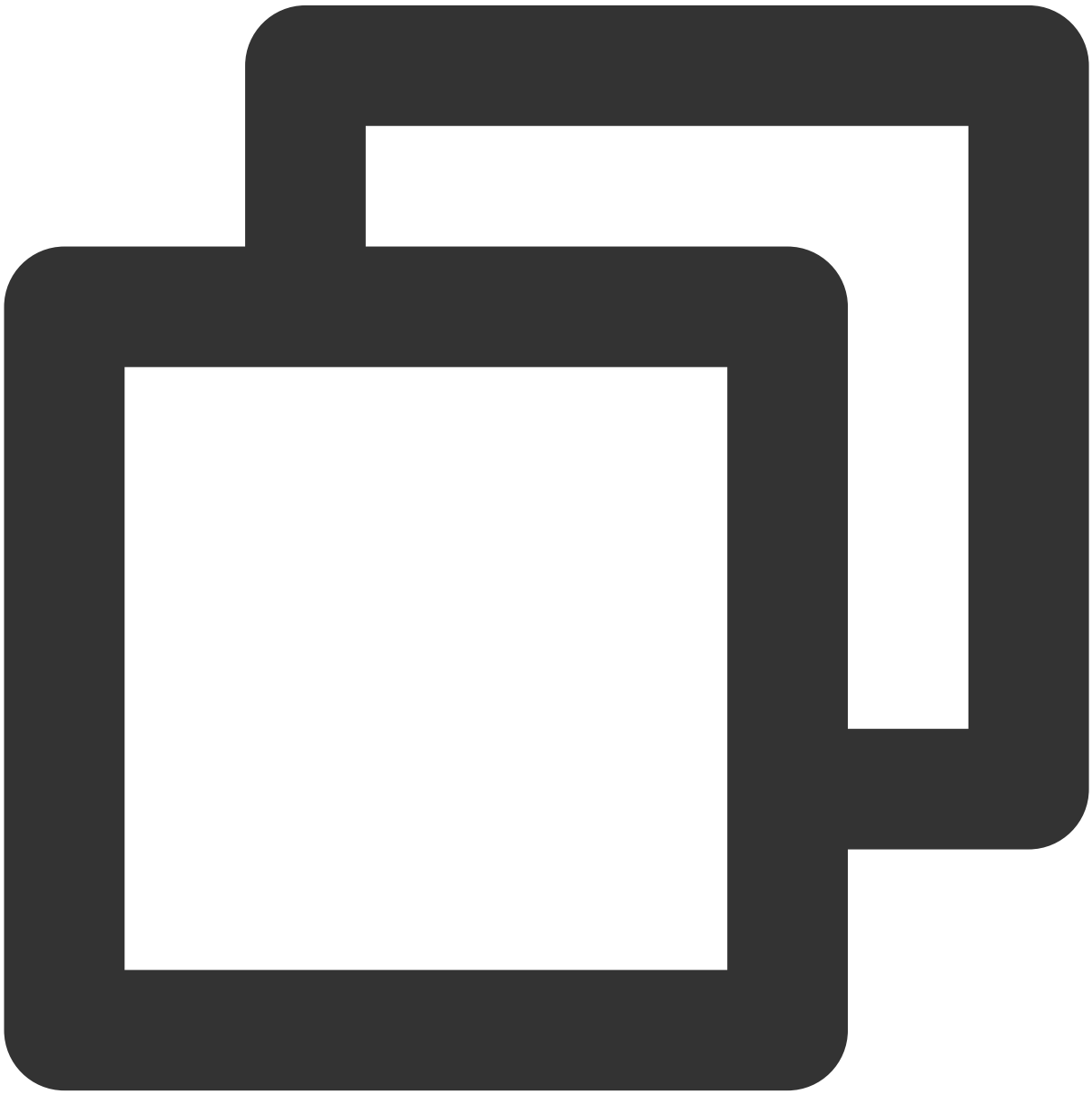


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: close
Date: Wed, 25 Mar 2020 10:07:01 GMT
Server: tencent-cos
x-cos-request-id: NWU3YjJkNDVfNDliNTJhMDlfYzZhMl8yOTVj****


<InitiateMultipartUploadResult>
```

```
            <Bucket>examplebucket-1250000000</Bucket>
            <Key>exampleobject</Key>
            <UploadId>1585130821cbb7df1d11846c073ad648e8f33b087cec2381df437acdc833c
</InitiateMultipartUploadResult>
```

案例二：使用请求头部指定元数据和 ACL

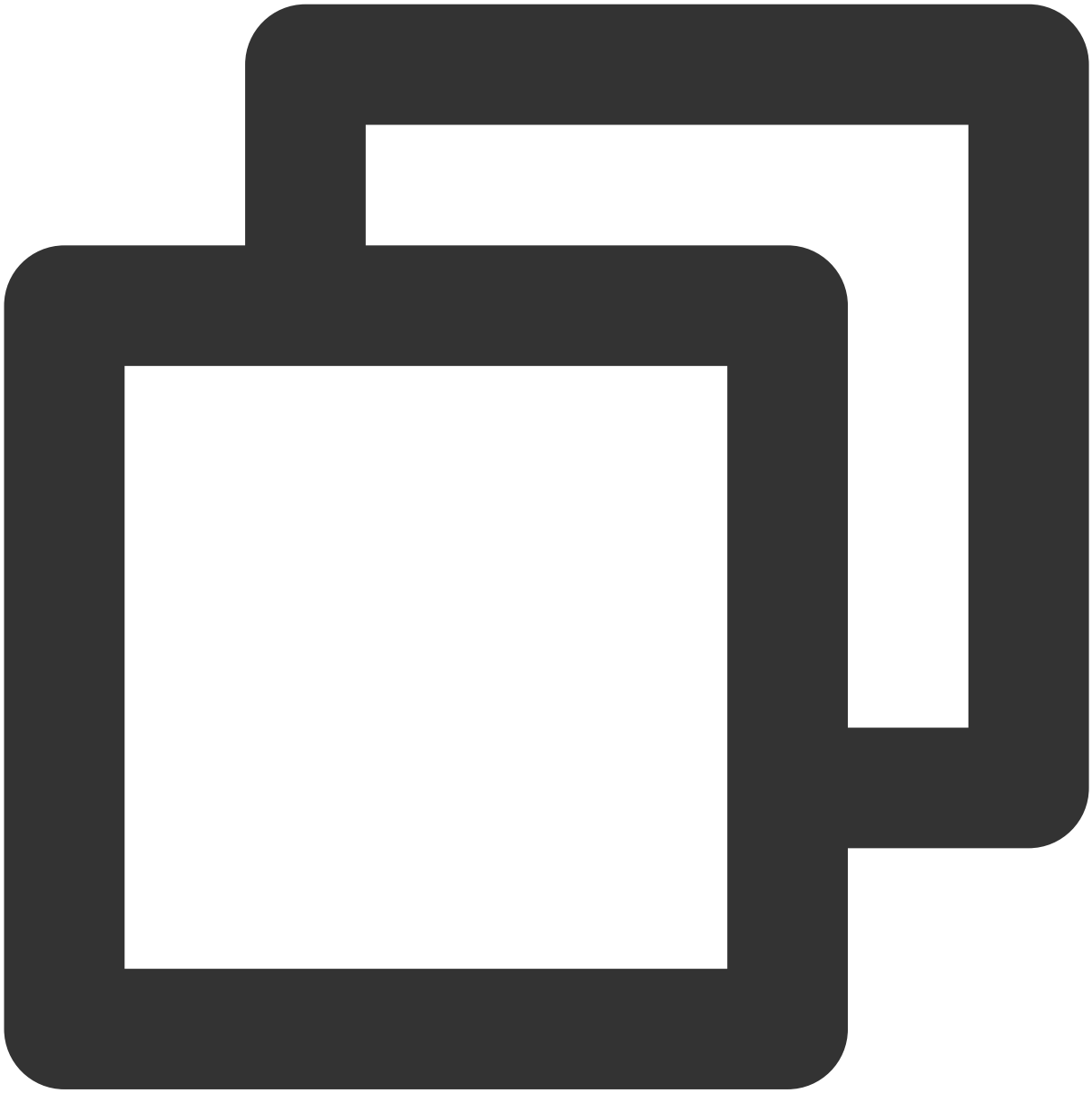
请求



```
POST /exampleobject?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Date: Thu, 28 May 2020 08:35:34 GMT
Content-Type: video/mp4
Cache-Control: max-age=86400
Content-Disposition: attachment; filename=example.jpg
x-cos-meta-example-field: example-value
x-cos-acl: public-read
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

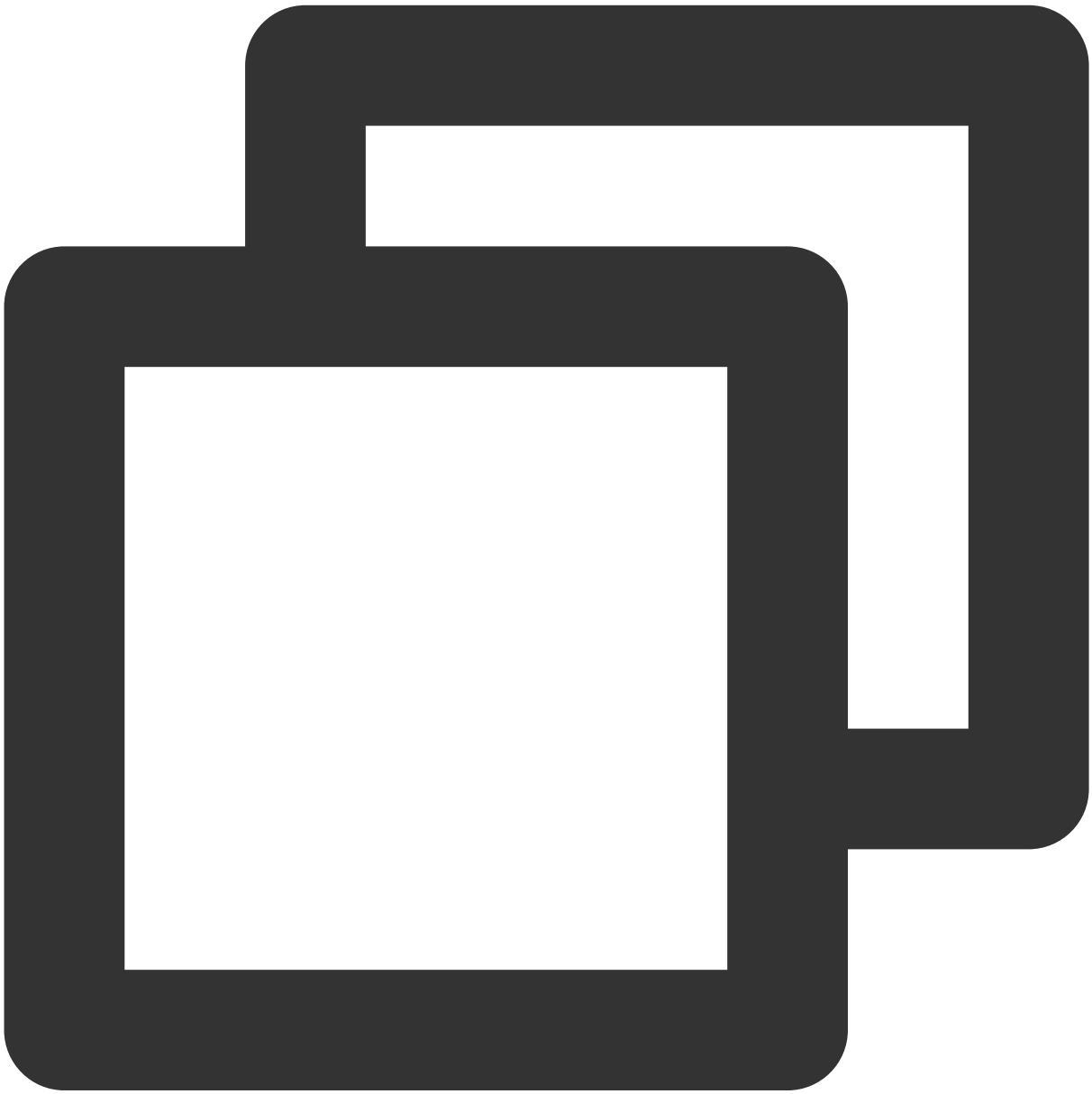


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: close
Date: Thu, 28 May 2020 08:35:34 GMT
Server: tencent-cos
x-cos-request-id: NWVjZjc3ZDZfOThjMjJhMDlfMjg5N18zNWYy****



<InitiateMultipartUploadResult>
```

```
            <Bucket>examplebucket-1250000000</Bucket>
            <Key>exampleobject</Key>
            <UploadId>1590654934dfb1343b4323b711afc22569c18af51596d4f2e40faf392fe1b
</InitiateMultipartUploadResult>
```

案例三：使用服务端加密 SSE-COS

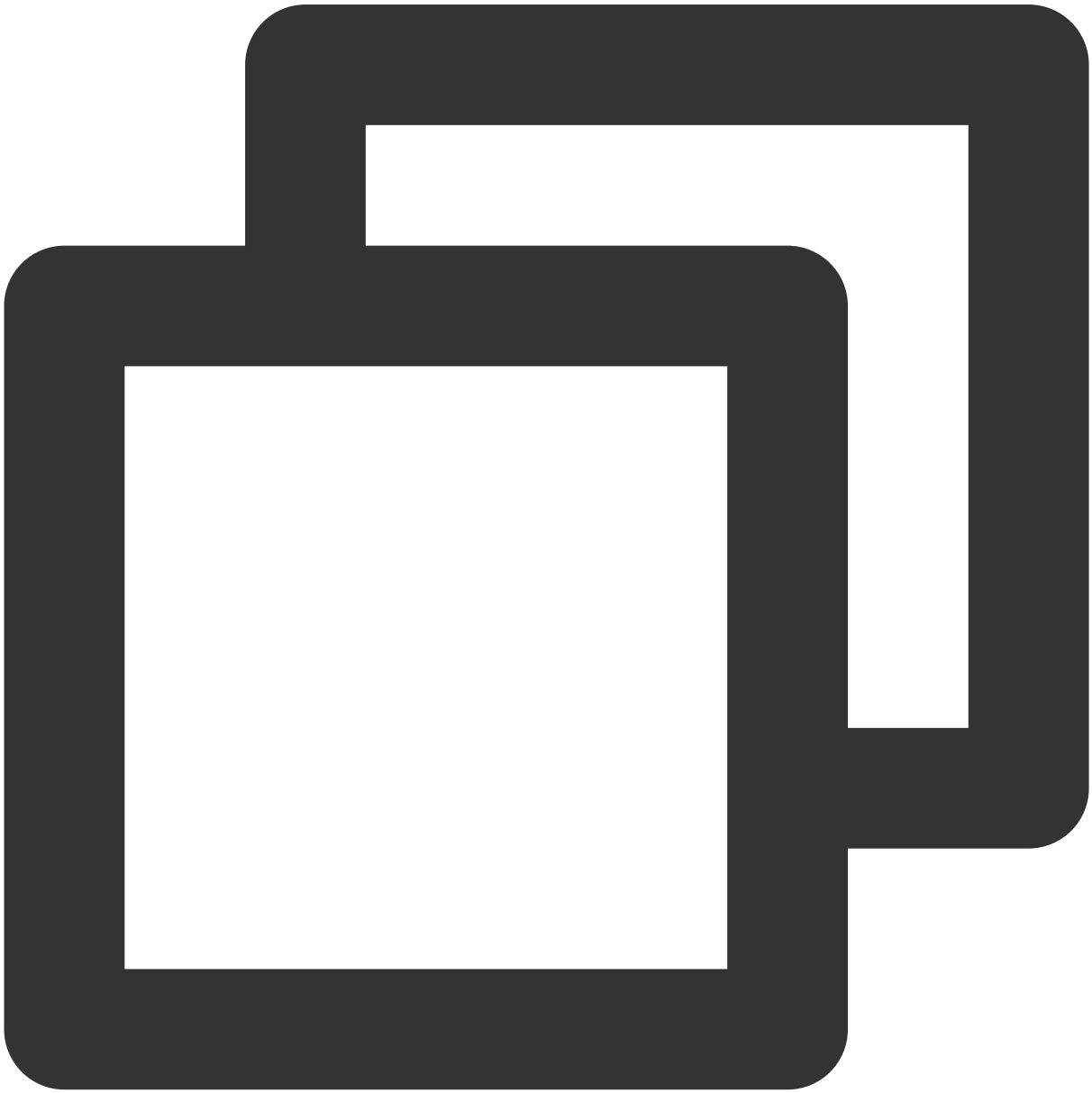
请求



```
POST /exampleobject?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Date: Thu, 28 May 2020 08:43:58 GMT
Content-Type: video/mp4
x-cos-server-side-encryption: AES256
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

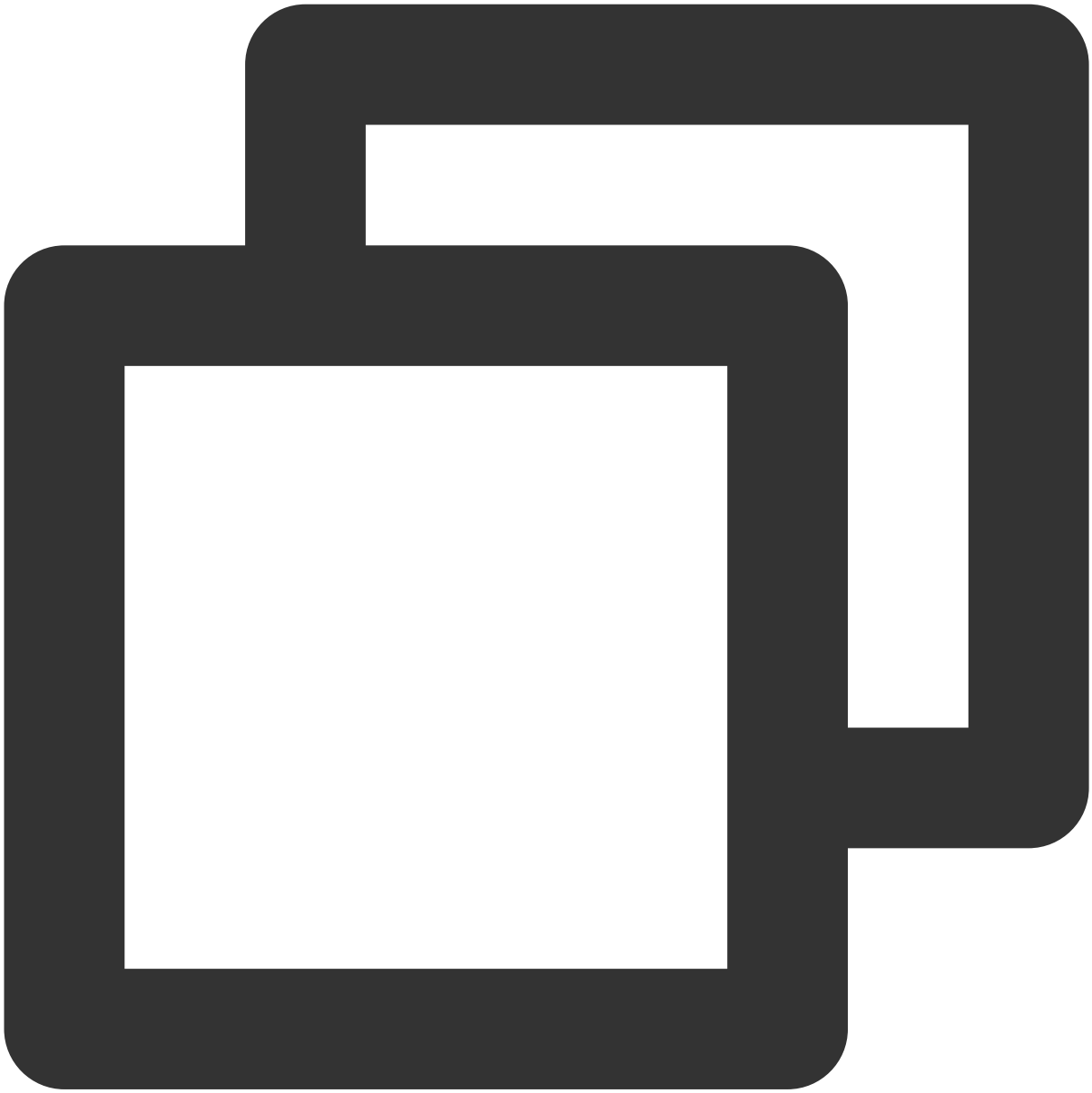


```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 230
Connection: close
Date: Thu, 28 May 2020 08:43:58 GMT
Server: tencent-cos
x-cos-request-id: NWVjZjc5Y2VfZjhjODBiMDlfMjIyOGFfMzYxYWVm****
x-cos-server-side-encryption: AES256



<InitiateMultipartUploadResult>
            <Bucket>examplebucket-1250000000</Bucket>
            <Key>exampleobject</Key>
            <UploadId>15906554384f160dd0a272ebb6fbcdb0ffbb61adb2b46fa6b9f2ffabcfb29
</InitiateMultipartUploadResult>
```

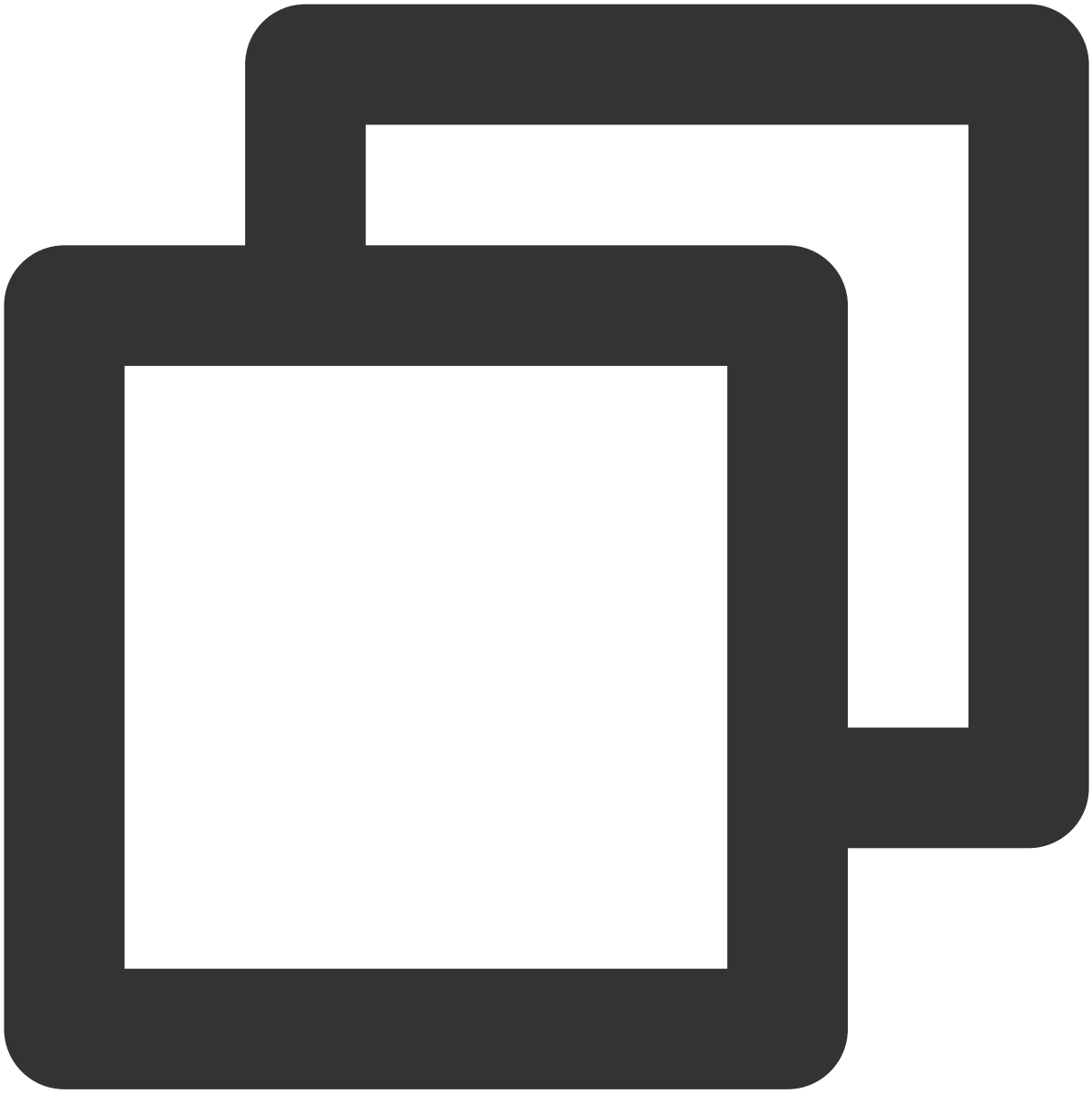
案例四：使用服务端加密 SSE-KMS

请求



```
POST /exampleobject?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 28 May 2020 08:44:19 GMT
Content-Type: video/mp4
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
x-cos-server-side-encryption-context: eyJhdXRob3IiOiJmeXNudGlhbiIsImNvbXBhbnkiOiJUZ
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应

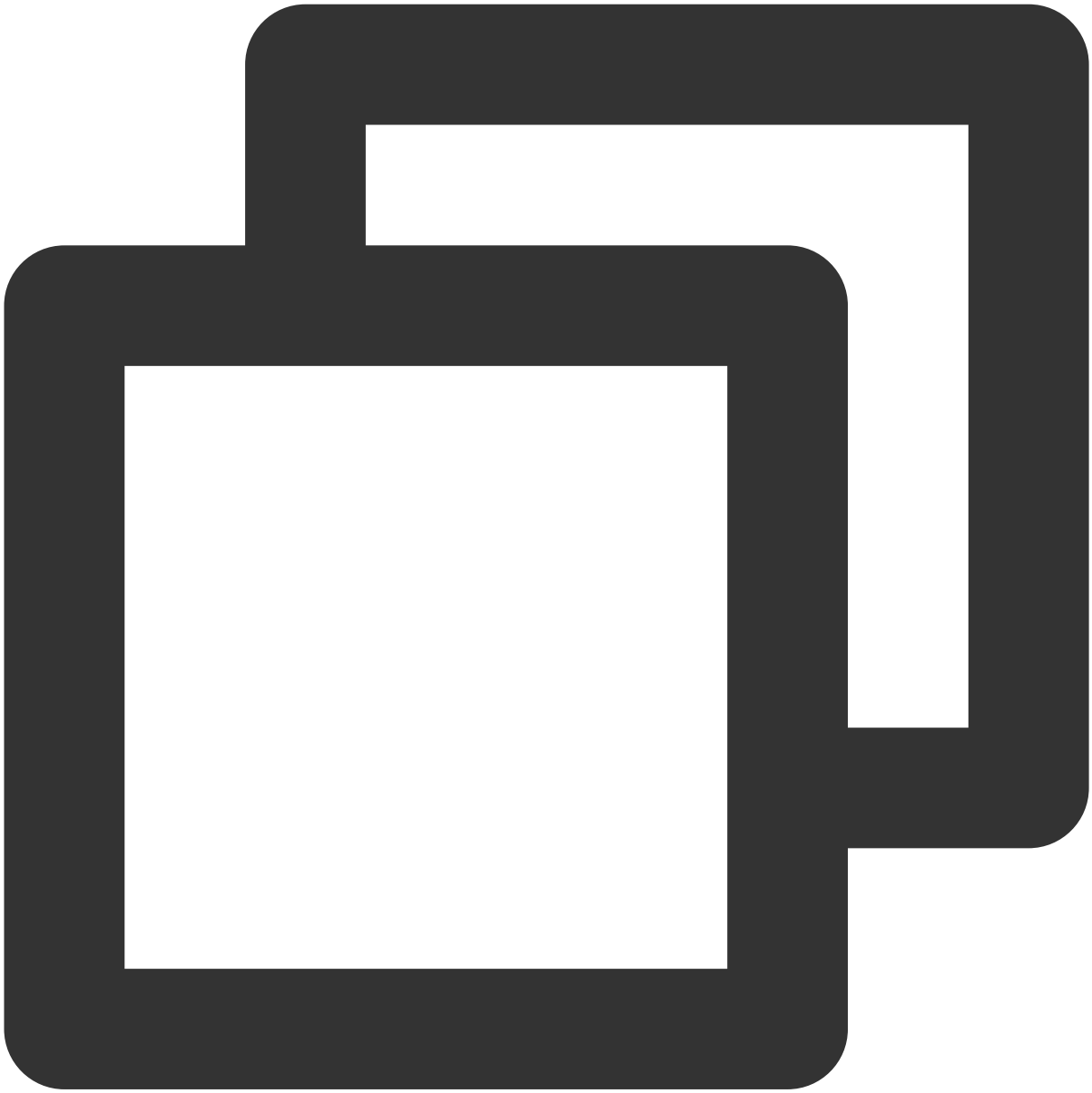


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: close
Date: Thu, 28 May 2020 08:44:20 GMT
Server: tencent-cos
x-cos-request-id: NWVjZjc5ZTNfMmZiOTJhMDlfMzJlNDJfMjkzNGJi****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
```

```
<InitiateMultipartUploadResult>
            <Bucket>examplebucket-1250000000</Bucket>
            <Key>exampleobject</Key>
            <UploadId>1590655460799012170 2e8e4b706eb0f12b8568a3f3b0b76b884e4df676ed
</InitiateMultipartUploadResult>
```

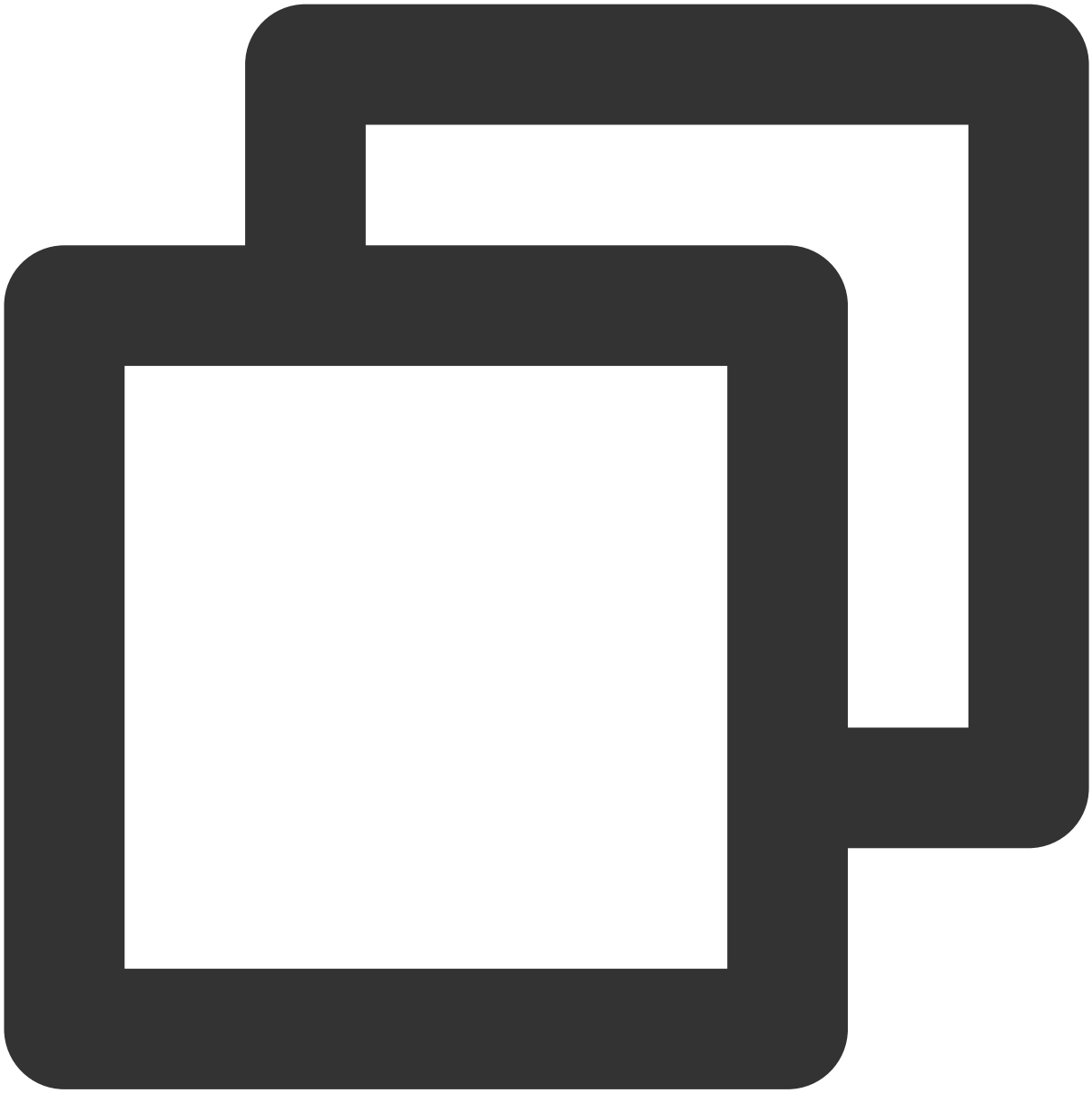
案例五：使用服务端加密 SSE-C

请求



```
POST /exampleobject?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 28 May 2020 08:44:41 GMT
Content-Type: video/mp4
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNE
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: close
Date: Thu, 28 May 2020 08:44:41 GMT
Server: tencent-cos
x-cos-request-id: NWVjZjc5ZjlfOGJjOTJhMDlfNzJmYV8xOTcy****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
```

版权所有：腾讯云计算（北京）有限责任公司

```
<InitiateMultipartUploadResult>
            <Bucket>examplebucket-1250000000</Bucket>
            <Key>exampleobject</Key>
            <UploadId>15906554815fb0c8bda2edae20d895ad7452e949bf51541b31ca14a029fb6
</InitiateMultipartUploadResult>
```

Upload Part

最近更新时间：2023-12-21 15:05:40

功能描述

Upload Part 接口请求实现将对象按照分块的方式上传到 COS。最多支持10000分块，每个分块大小为1MB - 5GB，最后一个分块可以小于1MB。

说明

分块上传步骤说明：

1.1 分块上传首先需要进行初始化，使用 Initiate Multipart Upload 接口实现，初始化后将得到一个 UploadId，唯一标识本次上传。

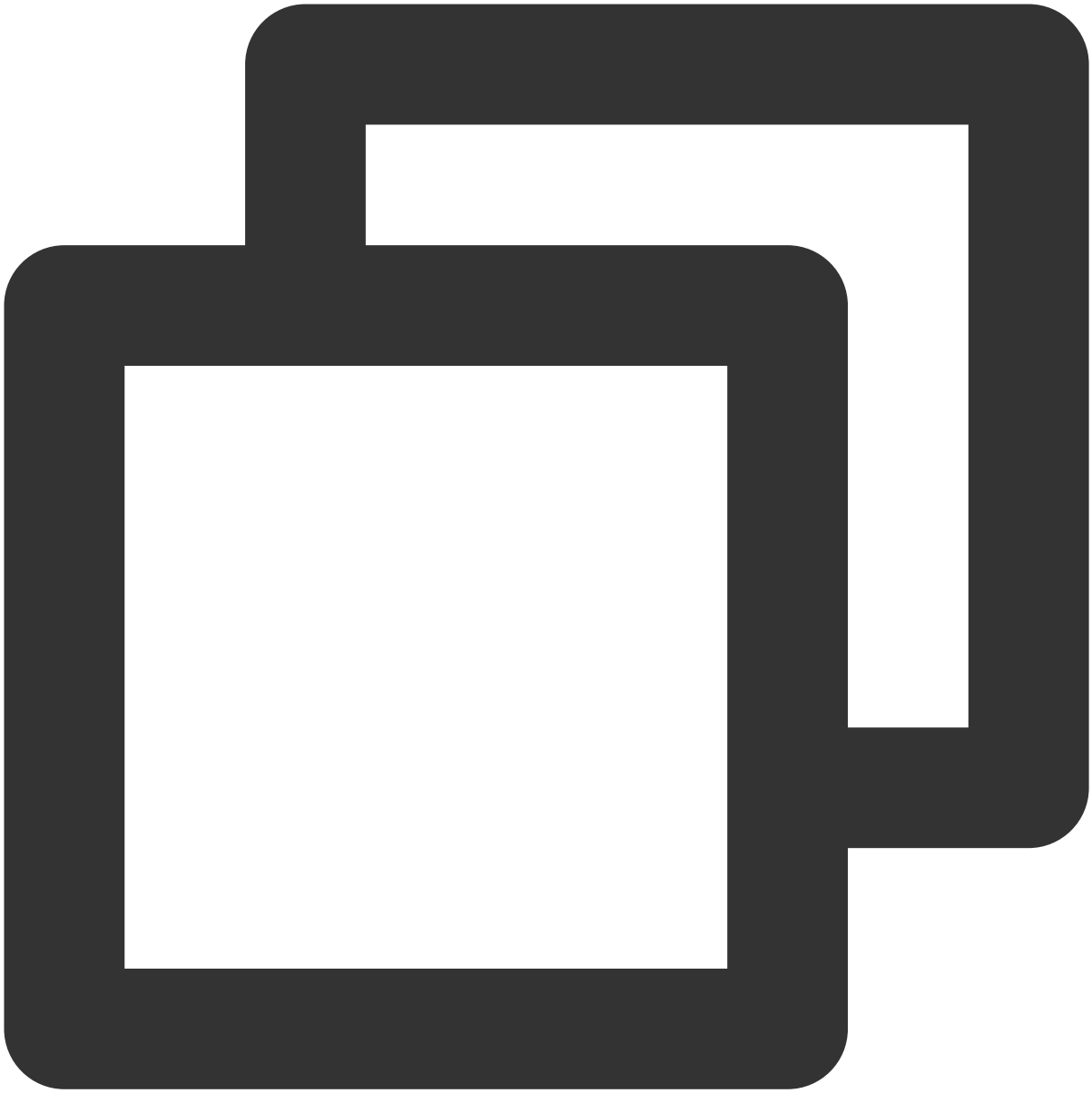
1.2 在每次请求 Upload Part 时，需要携带 partNumber 和 uploadId，partNumber 为块的编号，支持乱序上传。

1.3 当传入 uploadId 和 partNumber 都相同的时候，后发起上传请求的块将覆盖之前传入的块。当 uploadId 不存在时将返回404错误，NoSuchUpload。

只有主账号或者被授权 Upload Part 接口权限的子账号可以调用该接口。

请求

请求示例



```
PUT /<ObjectKey>?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: Content Type
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Object Part]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

名称	描述	类型	是否必选
partNumber	标识本次分块上传的编号，范围在1 - 10000	integer	是
uploadId	标识本次分块上传的 ID，使用 Initiate Multipart Upload 接口初始化分块上传时得到的 UploadId	string	是

请求头

此接口除使用公共请求头部外，还支持以下请求头部，了解公共请求头部详情请参见 公共请求头部 文档。

服务端加密（SSE）相关头部

如果当前分块上传在初始化时使用了 SSE-C，那么在上传分块时需指定与初始化分块上传时所指定的加密算法和密钥信息，否则不能指定下列头部。

名称	描述	类型	是否必选
x-cos-server-side-encryption-customer-algorithm	服务端加密算法，目前仅支持 AES256	string	是
x-cos-server-side-encryption-customer-key	服务端加密密钥的 Base64 编码 例如 `MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=`	string	是
x-cos-server-side-encryption-customer-key-MD5	服务端加密密钥的 MD5 哈希值，使用 Base64 编码 例如 `U5L61r7jcwdNvT7frmUG8g==`	string	是
x-cos-traffic-limit	针对本分块上传进行流量控制的限速值，必须为数字，单位默认为 bit/s。限速值设置范围为819200 - 838860800，即100KB/s - 100MB/s，如果超出该范围将返回400错误	integer	否

请求体

此接口请求的请求体为对象（文件）分块内容。

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

服务端加密（SSE）相关头部

如果当前分块上传在初始化时使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

此接口响应体为空。

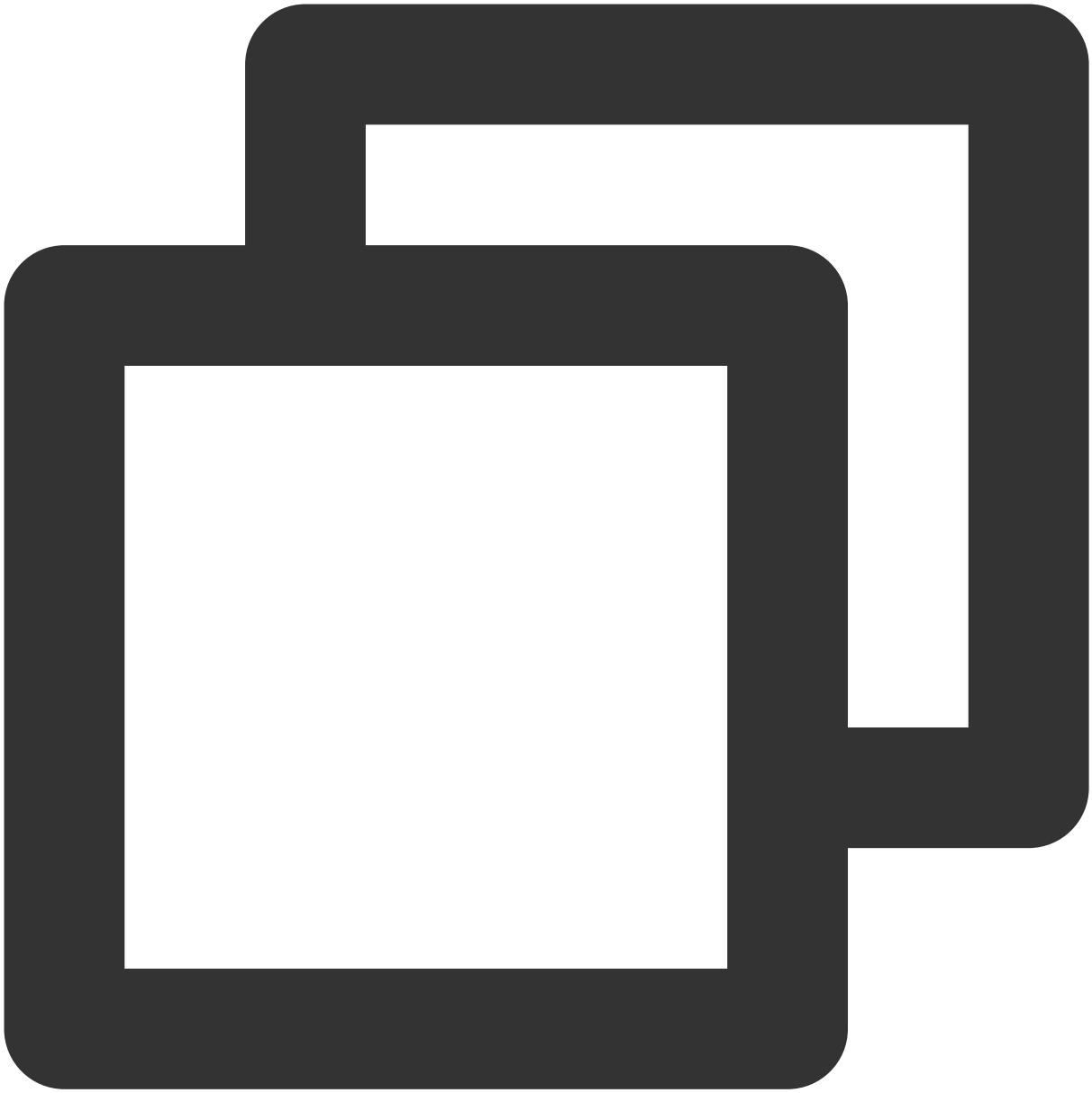
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

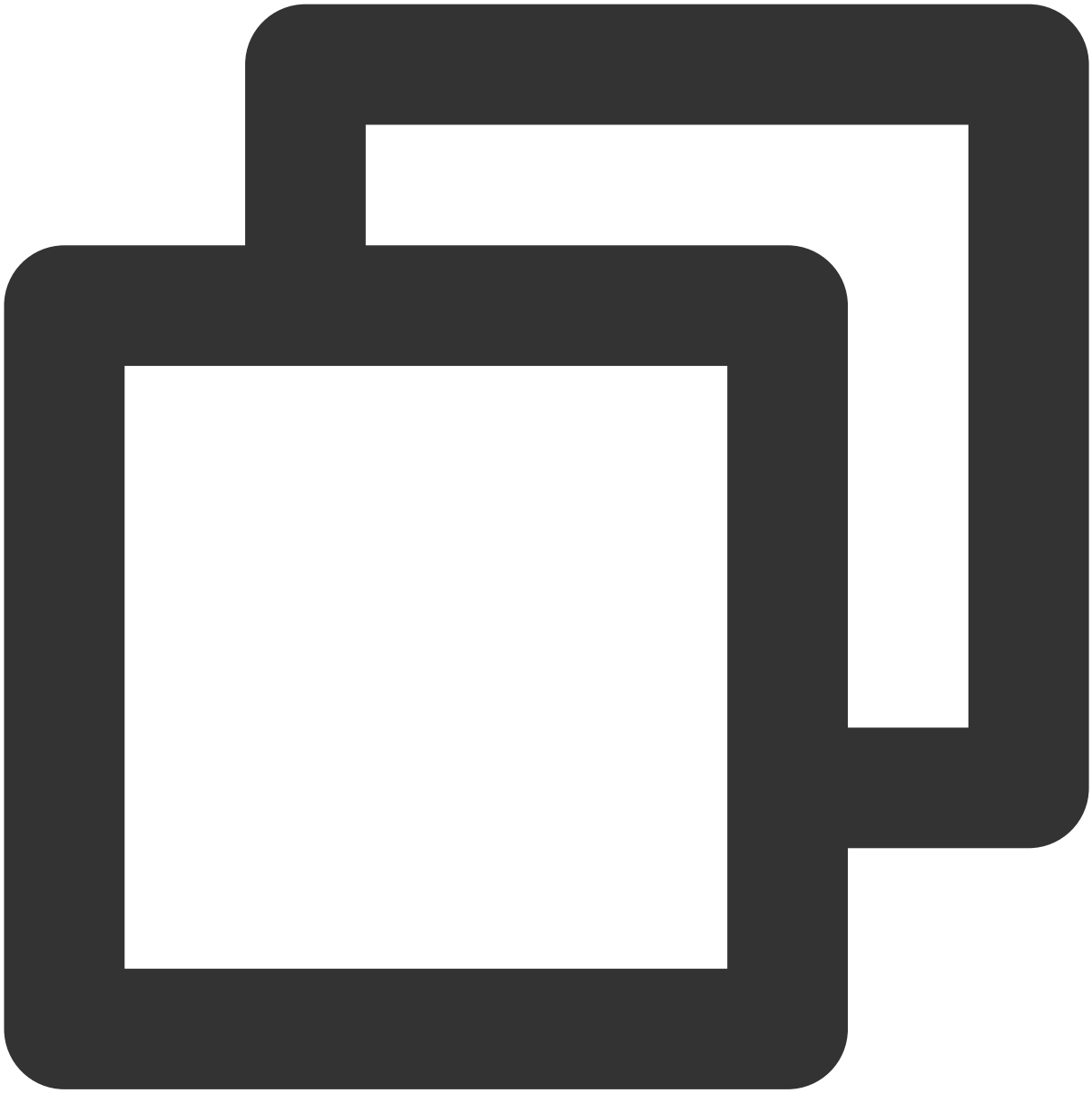
案例一：简单案例

请求



```
PUT /exampleobject?partNumber=1&uploadId=1585130821cbb7df1d11846c073ad648e8f33b087c
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 25 Mar 2020 10:07:14 GMT
Content-Type: application/octet-stream
Content-Length: 1048576
Content-MD5: OScKloo1fSQgfpkRFiUH6w==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

[Object Part]
```

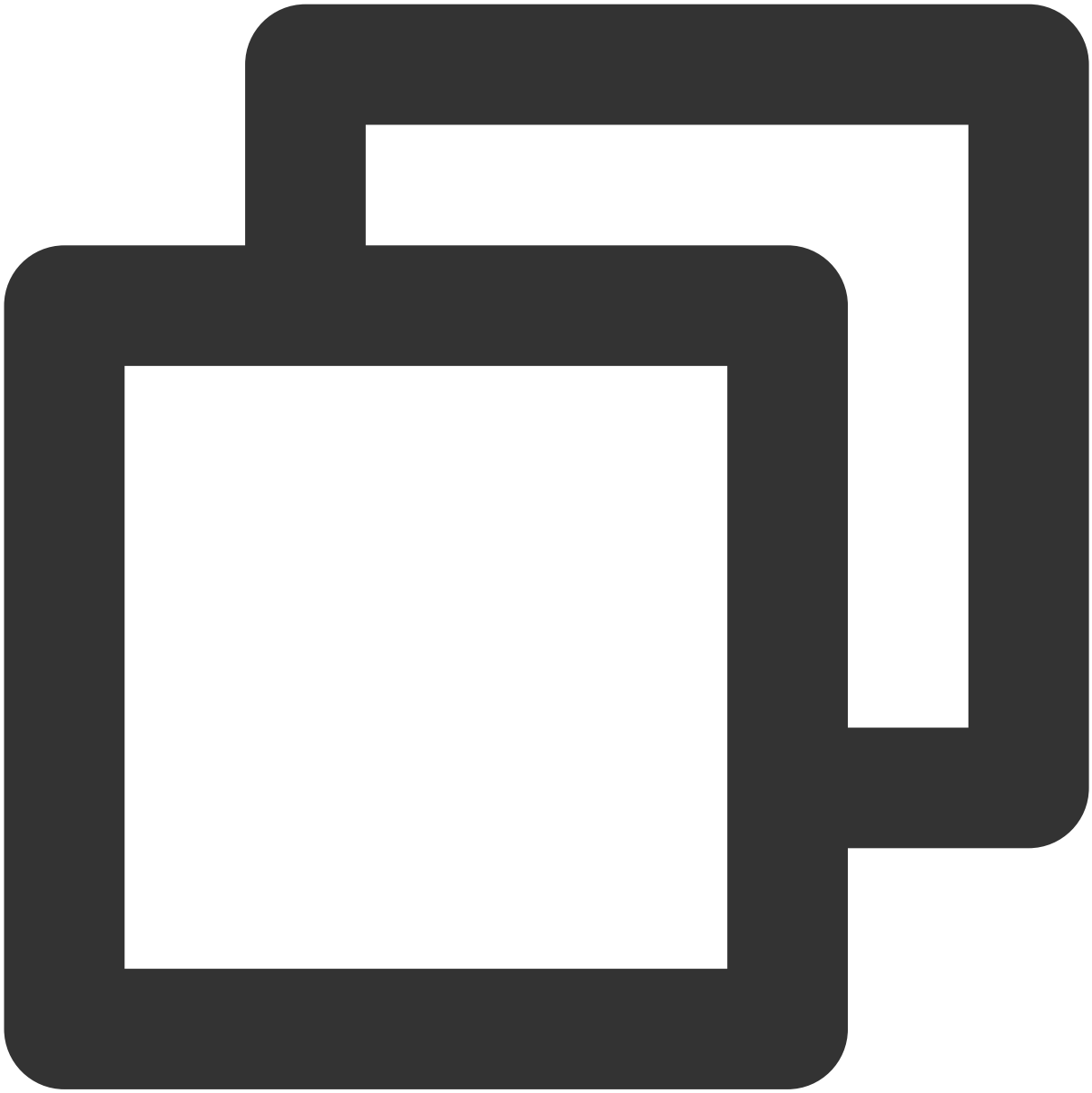
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Wed, 25 Mar 2020 10:07:14 GMT
ETag: "39270a968a357d24207e9911162507eb"
Server: tencent-cos
x-cos-hash-crc64ecma: 9973912126177188060
x-cos-request-id: NWU3YjJkNTJfZDBjODJhMDlfMjU4NTZfMjc5MzBh****
```

案例二：使用服务端加密 SSE-COS

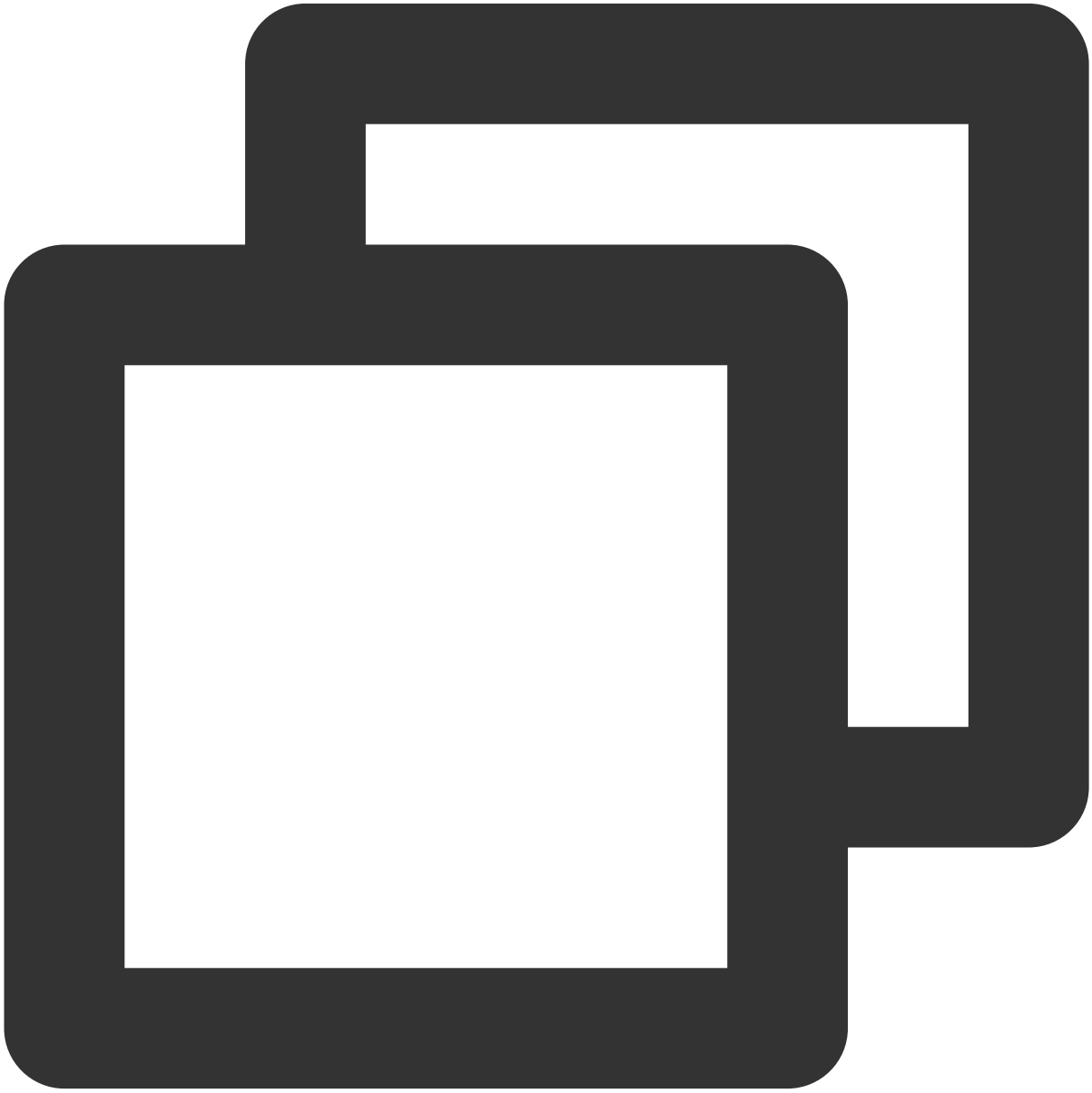
请求



```
PUT /exampleobject?partNumber=1&uploadId=159086254025135529 5a5c736513d70d42b92bbde4
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:15:50 GMT
Content-Type: application/octet-stream
Content-Length: 13
Content-MD5: EI9SjrY7Zec08nrjMfX/qg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
```

```
Connection: close

[Object Part]
```

响应

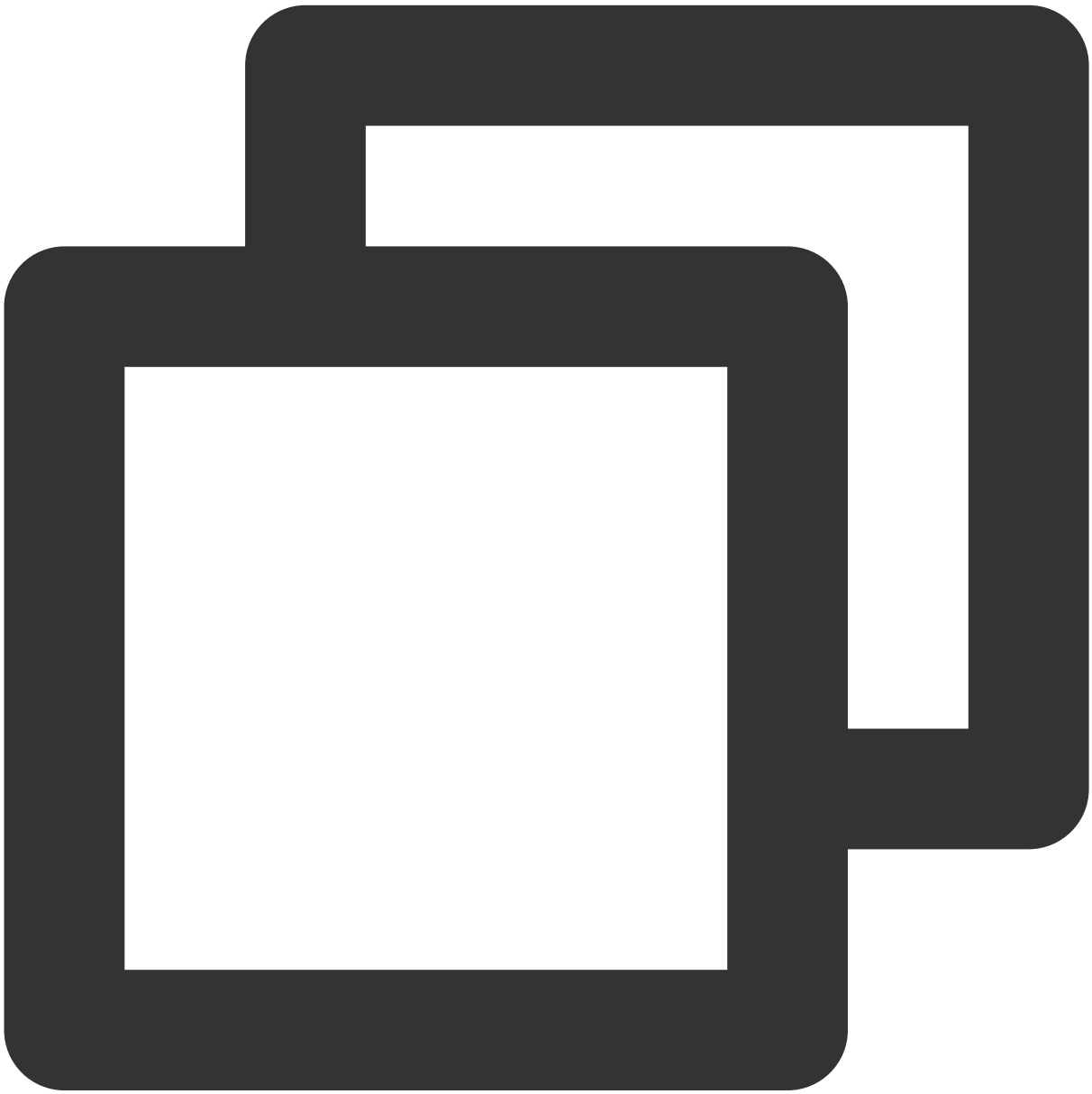


```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Sat, 30 May 2020 18:15:51 GMT
ETag: "108f528eb63b65e734f27ae331f5ffaa"
```

```
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEyZDZfYjNjMjJhMDlfMmJlM18zOWI2****
x-cos-server-side-encryption: AES256
```

案例三：使用服务端加密 SSE-KMS

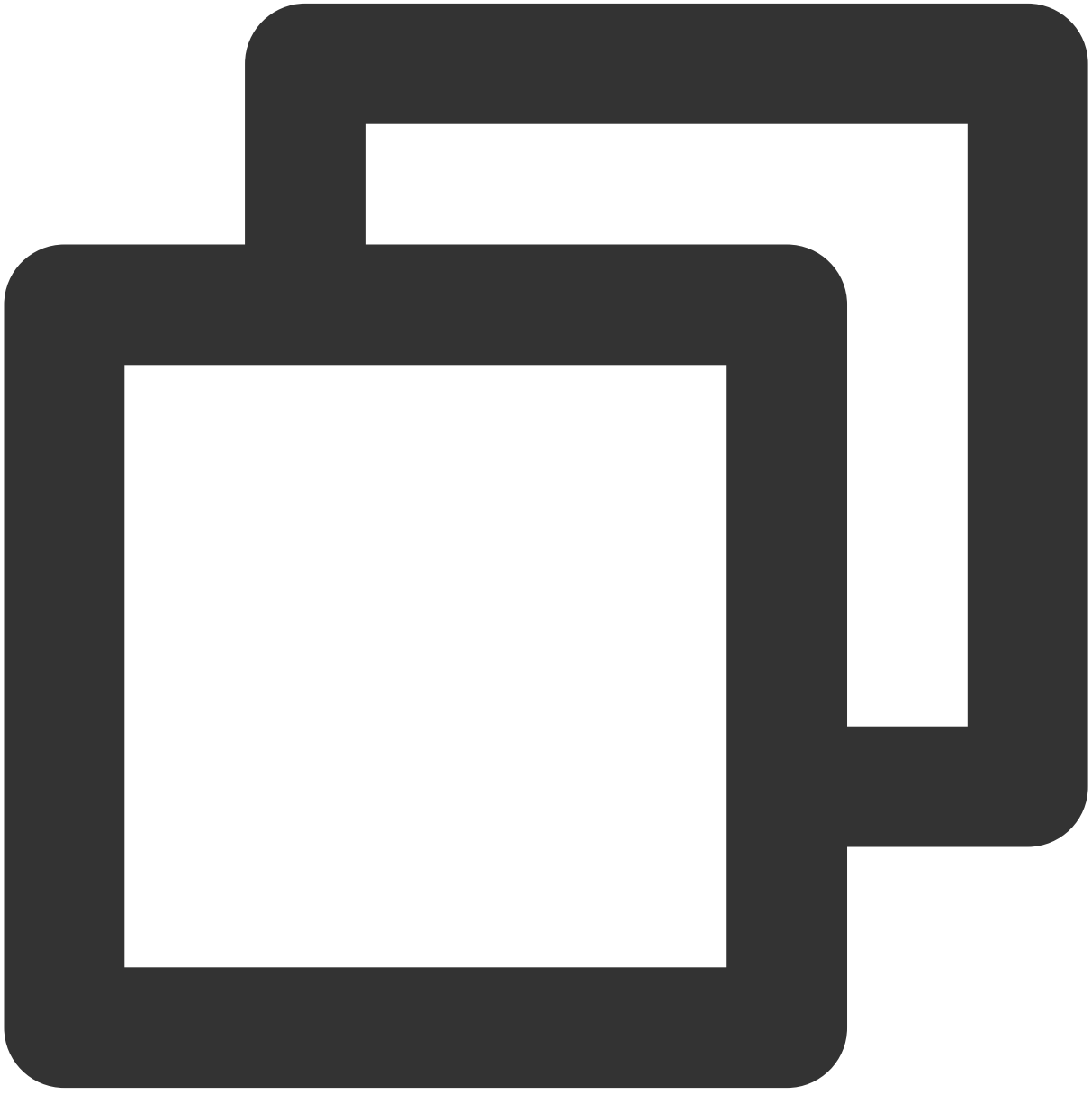
请求



```
PUT /exampleobject?partNumber=1&uploadId=15908625793957d71176fa878282d266a95b79dc2a
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Date: Sat, 30 May 2020 18:16:29 GMT
Content-Type: application/octet-stream
Content-Length: 13
Content-MD5: EI9SjrY7Zec08nrjMfX/qg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

[Object Part]
```

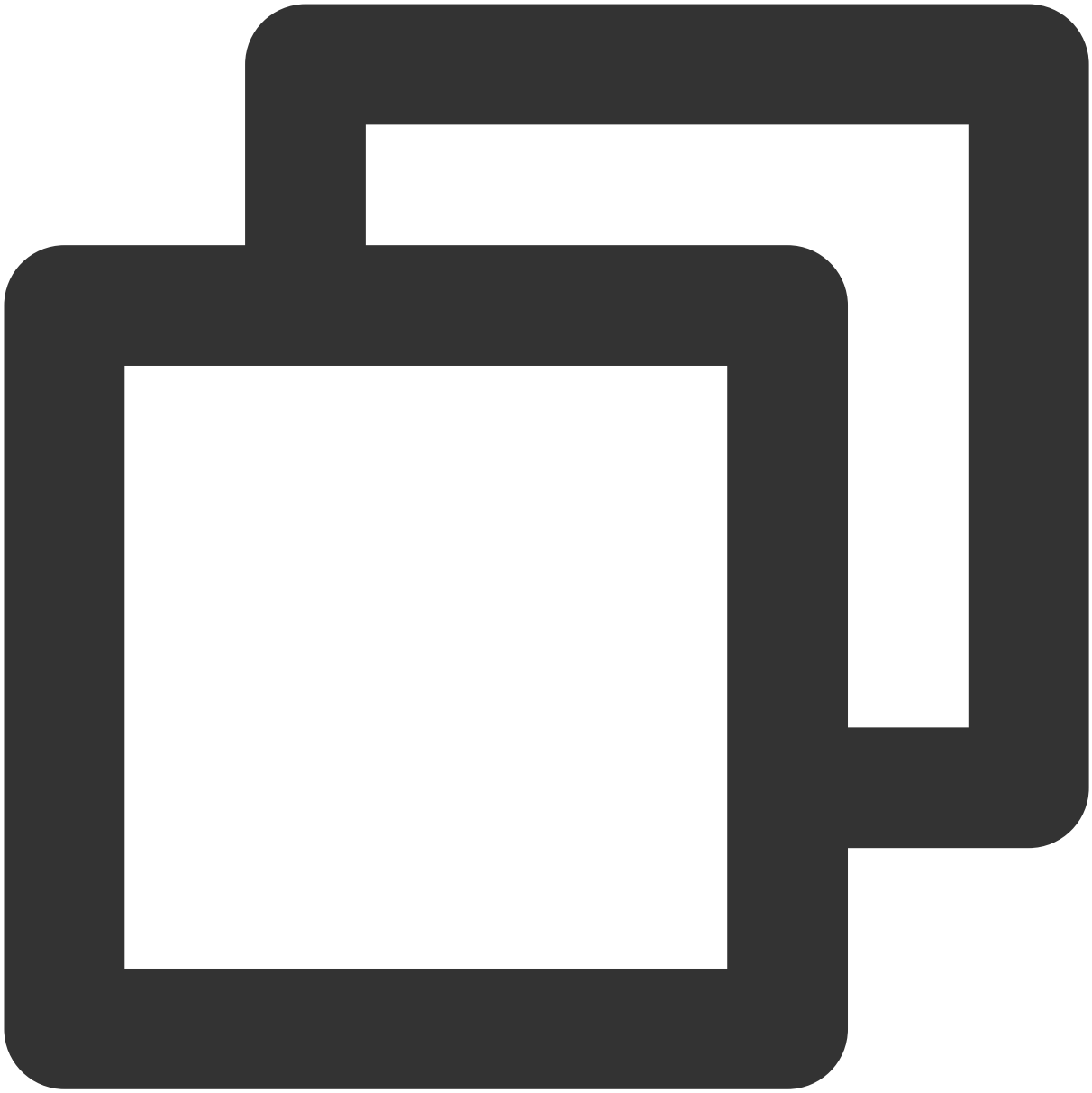
响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Sat, 30 May 2020 18:16:30 GMT
ETag: "1d3e8ae9da423b440341b09f1616f074"
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEyZmRfYTRiOTJhMDlfMTE0ZGRfMmE3OTk4****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
```

案例四：使用服务端加密 SSE-C

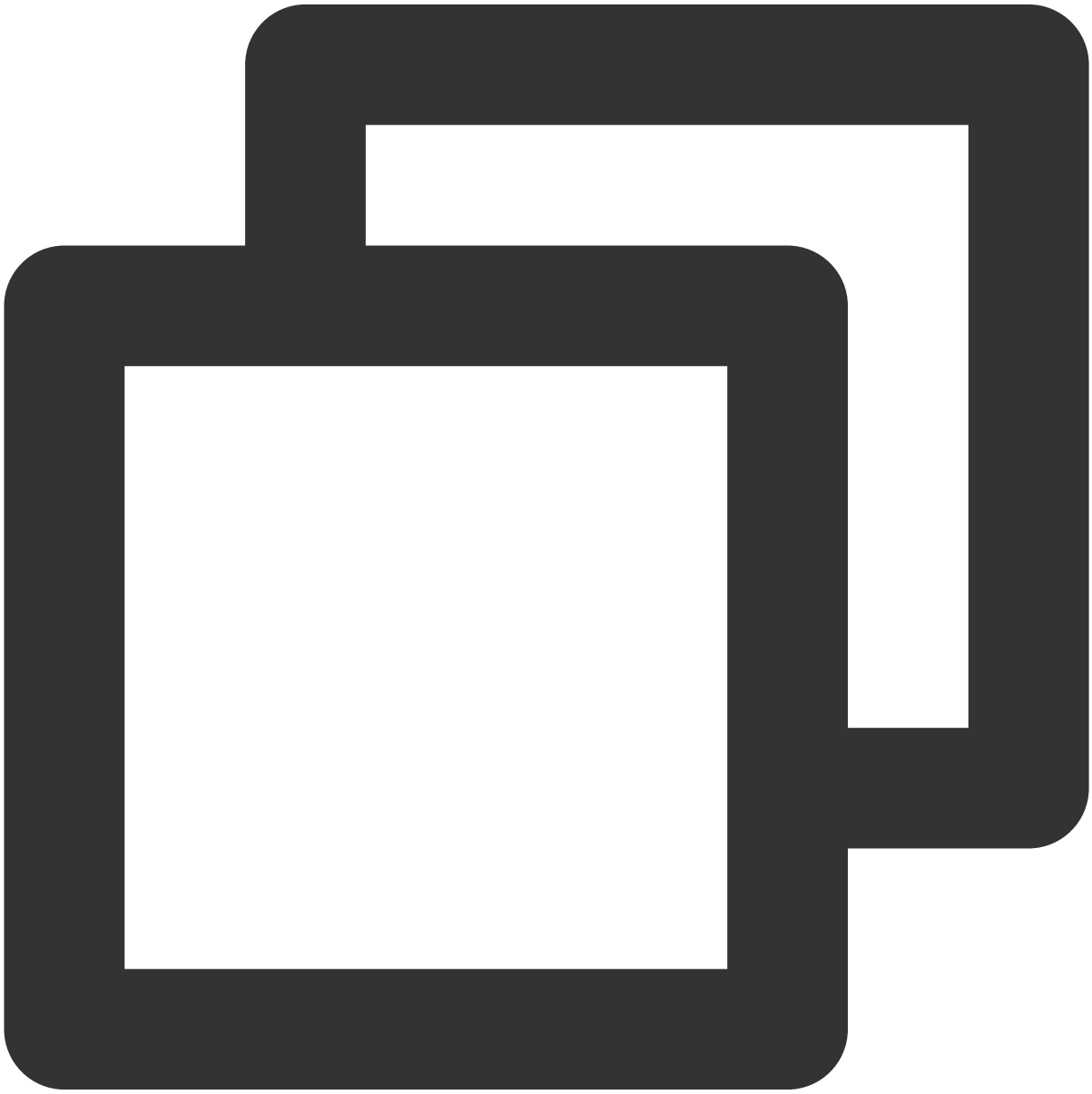
请求



```
PUT /exampleobject?partNumber=1&uploadId=1590862610acd643927bad05cd3947bf98b56b04b0
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:17:01 GMT
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNE
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
Content-Type: application/octet-stream
Content-Length: 13
Content-MD5: EI9SjrY7Zec08nrjMfX/qg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

```
[Object Part]
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Sat, 30 May 2020 18:17:01 GMT
ETag: "ff14981a35a58eb97bca838b055c573b"
Server: tencent-cos
```

```
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEzMWRfYjRjOTJhMDlfYWRhXzFhZmYw****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==
```

Upload Part - Copy

最近更新时间：2023-12-21 15:06:17

功能描述

Upload Part - Copy 请求用于实现将一个对象的分块内容从源路径复制到目标路径。通过指定 x-cos-copy-source 来指定源对象，x-cos-copy-source-range 指定字节范围（允许分块的大小为1MB - 5GB）。

说明

以下两种情况拥有调用此 API 的权限：

拥有主账号权限。

拥有源对象 GetObjecet 接口权限，且在目标路径下有 InitiateMultipartUpload、ListMultipartUploads、ListParts、PutObject、CompleteMultipartUpload、AbortMultipartUpload 接口权限。

注意

使用 Upload Part - Copy 接口前，需先使用 Initiate Multipart Upload 接口初始化分块上传任务并指定目标路径。

如果目标对象和源对象不属于同一个地域，且目标对象分块会超过5GB，那么需要使用分块上传或者分块拷贝的接口来复制对象。

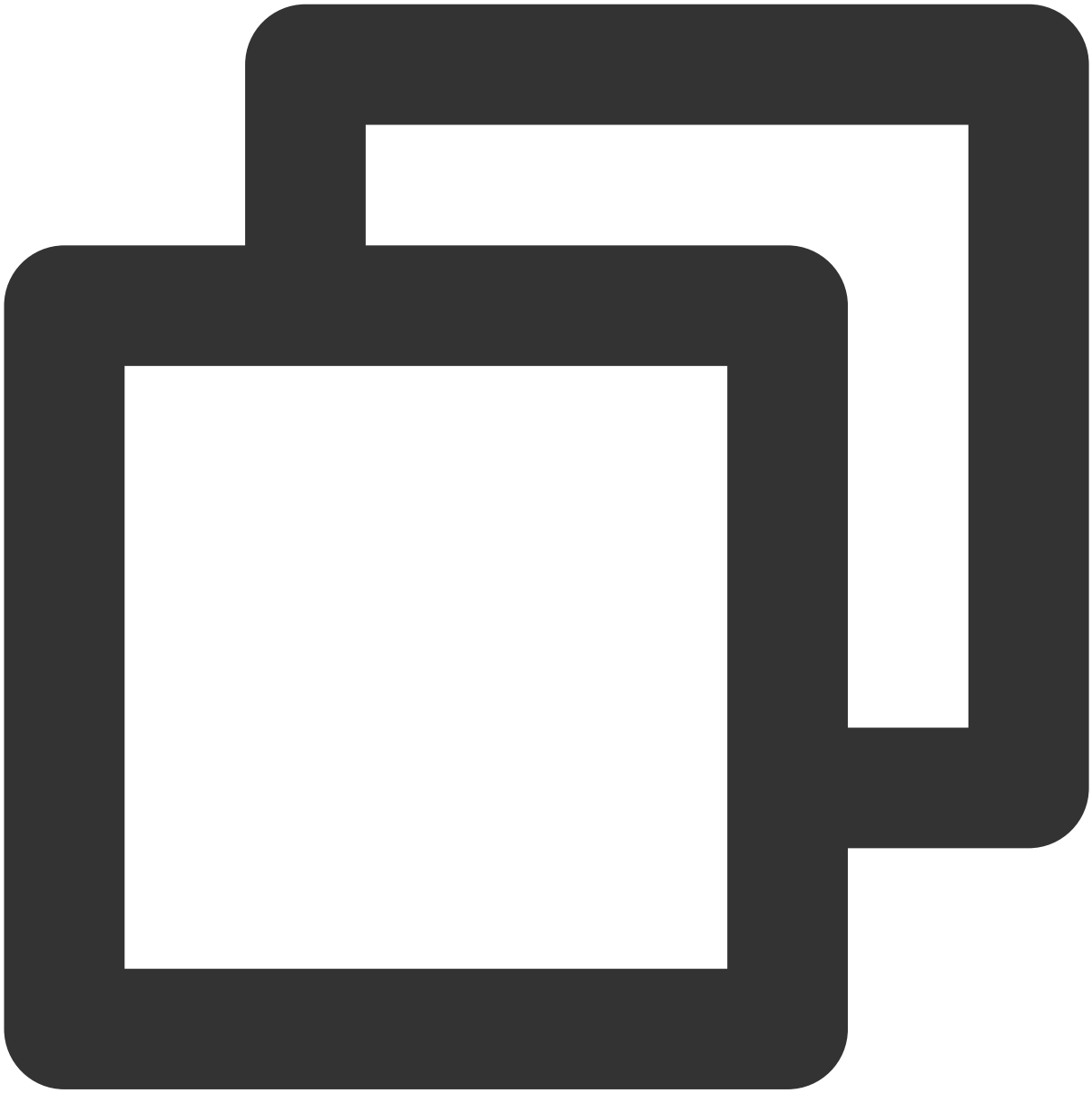
使用上传分块对象，必须先初始化分块上传。在初始化分块上传的响应中，会返回一个唯一的描述符（upload ID），您需要在分块上传请求中携带此 ID。

版本控制

当存储桶启用了版本控制，x-cos-copy-source 标识被复制的对象的当前版本。如果当前版本是删除标记，并且 x-cos-copy-source 不指定版本，则对象存储会认为该对象已删除并返回404错误。如果您在 x-cos-copy-source 中指定 versionId 且 versionId 是删除标记，则对象存储会返回 HTTP 400错误，因为删除标记不允许作为 x-cos-copy-source 的版本。

请求

请求示例



```
PUT /<ObjectKey>?partNumber=PartNumber&uploadId=UploadId  HTTP/1.1
Host: <Bucketname-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
x-cos-copy-source: <Bucketname>-<APPID>.cos.<Region>.myqcloud.com/filepath
x-cos-copy-source-range: bytes=first-last
x-cos-copy-source-if-match: etag
x-cos-copy-source-if-none-match : etag
x-cos-copy-source-if-unmodified-since: time_stamp
x-cos-copy-source-if-modified-since: time_stamp
```

版权所有：腾讯云计算（北京）有限责任公司

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

公共头部

该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

非公共头部

必选头部

该请求操作的实现使用如下必选头部：

名称	描述	类型	是否必选
x-cos-copy-source	源对象 URL 路径，可以通过 versionid 子资源指定历史版本	String	是

推荐头部

该请求操作的实现使用如下推荐请求头部信息：

名称	描述	类型	是否必选
x-cos-copy-source-range	源对象的字节范围，范围值必须使用 bytes=first-last 格式，first 和 last 都是基于 0 开始的偏移量。 例如 bytes=0-9 表示您希望拷贝源对象的开头10个字节的数据，如果不指定，则表示拷贝整个对象	String	否
x-cos-copy-source-If-Modified-Since	当 Object 在指定时间后被修改，则执行操作，否则返回412，可与 x-cos-copy-source-If-None-Match 一起使用，与其他条件联合使用返回冲突	String	否
x-cos-copy-source-If-Unmodified-Since	当 Object 在指定时间后未被修改，则执行操作，否则返回412，可与 x-cos-copy-source-If-Match 一起使用，与其他条件联合使用返回冲突	String	否
x-cos-copy-source-If-Match	当 Object 的 Etag 和给定一致时，则执行操作，否则返回412，可与 x-cos-copy-source-If-Unmodified-Since 一起使用，与其他条件联合使用返回冲突	String	否
x-cos-copy-source-If-None-	当 Object 的 Etag 和给定不一致时，则执行操作，否则返回412，可与 x-cos-copy-source-If-Modified-Since 一起使用，与其他条件	String	否

版权所有：腾讯云计算（北京）有限责任公司

Match	联合使用返回冲突		

请求参数

名称	描述	类型	是否必选
partNumber	分块拷贝的块号	String	是
uploadId	使用上传分块文件，必须先初始化分块上传。在初始化分块上传的响应中，会返回一个唯一的描述符（upload ID），您需要在分块上传请求中携带此 ID	String	是

请求体

该请求的请求体为空。

响应

响应头

公共响应头

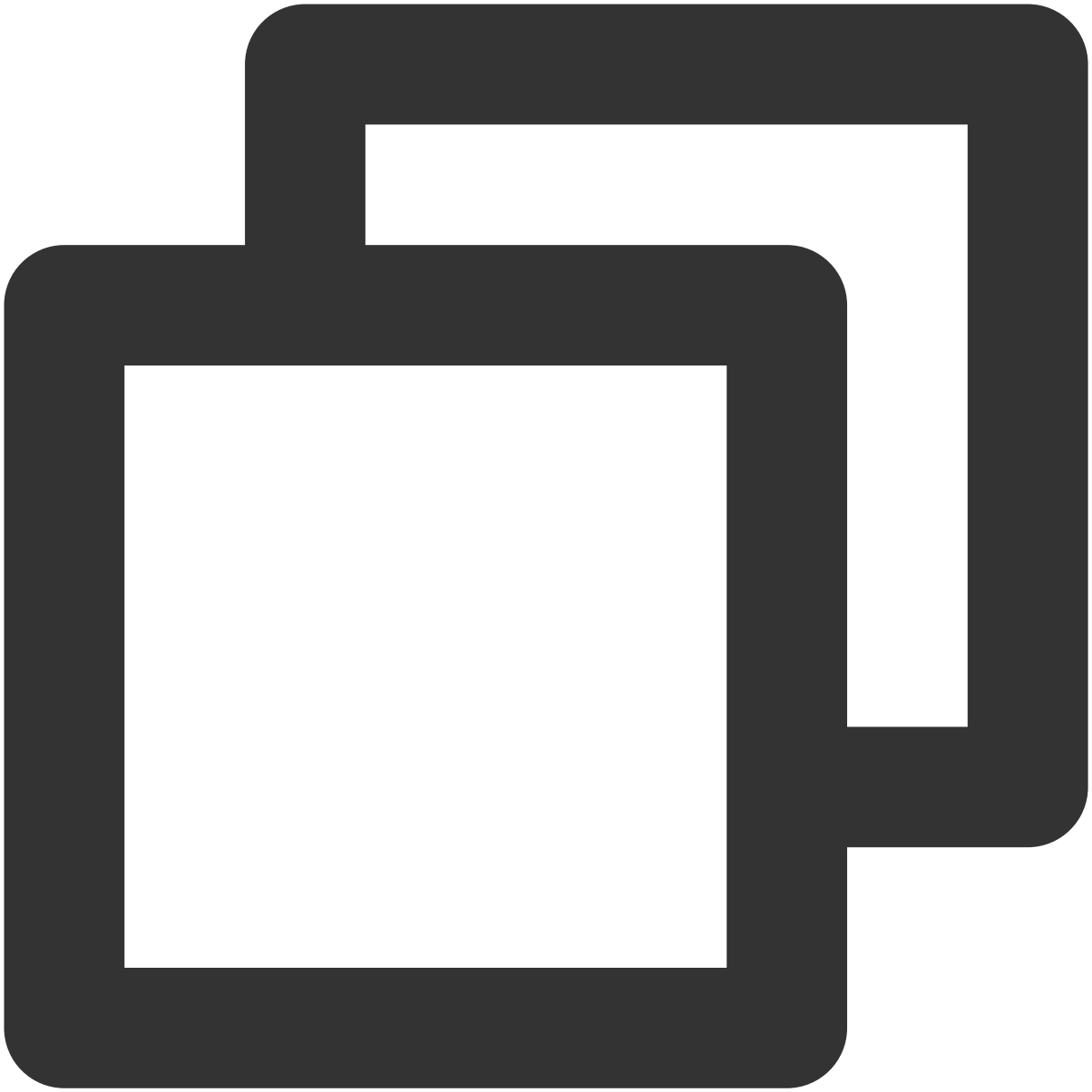
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

特有响应头

名称	描述	类型
x-cos-copy-source-version-id	如果已在源存储桶上启用版本控制，则复制源对象的版本	String
x-cos-server-side-encryption	如果通过 COS 管理的服务端加密来存储对象，响应将包含此头部和所使用的加密算法的值，AES256	String

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<?xml version="1.0" encoding="UTF-8" ?>
<CopyPartResult>
   <ETag>"ba82b57cfdfda8bd17ad4e5879ebb4fe"</ETag>
   <LastModified>2017-09-04T04:45:45</LastModified>
</CopyPartResult>
```

具体的节点描述如下：

名称	描述	类型
CopyPartResult	返回复制结果信息	String

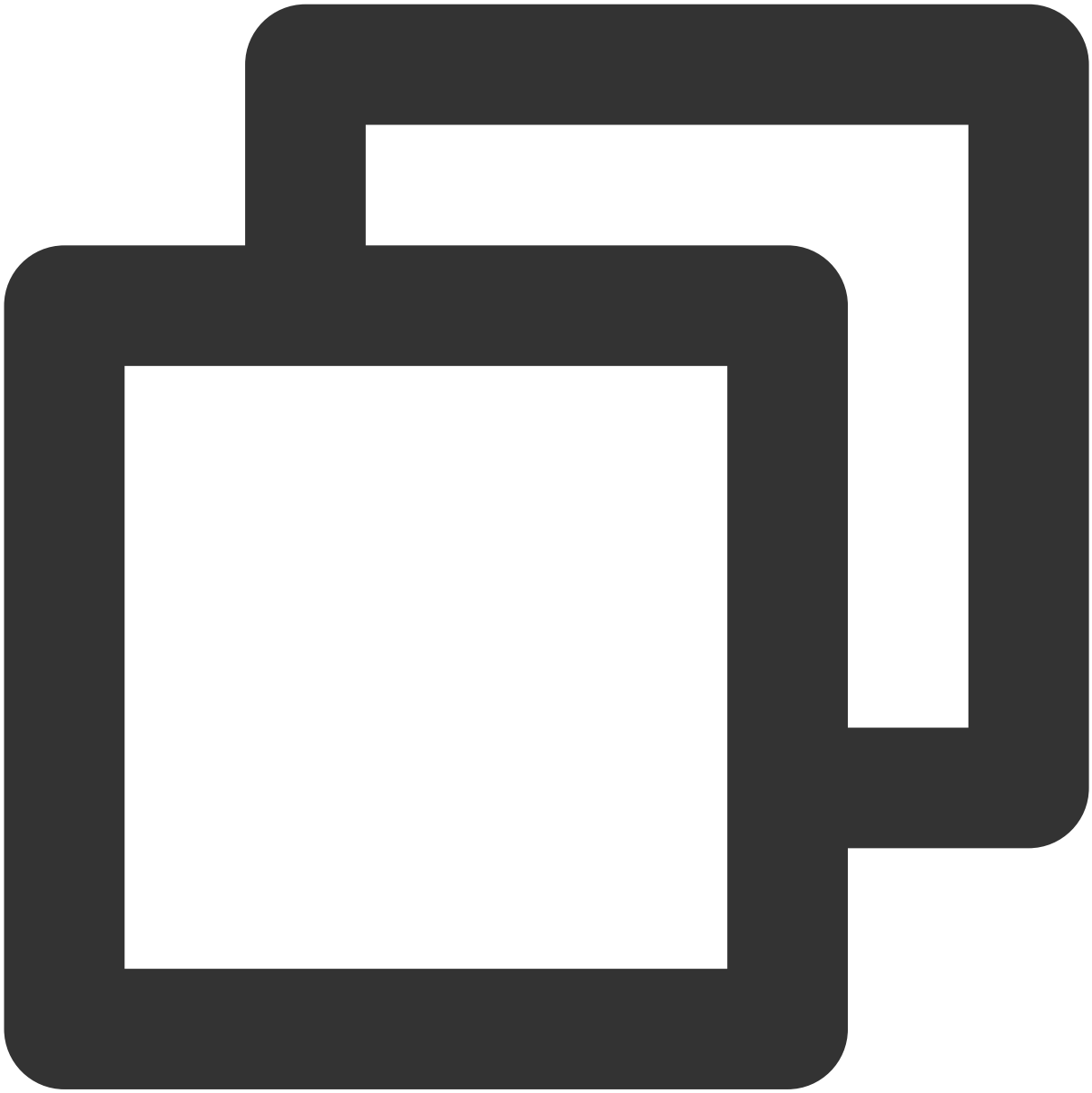
ETag	返回对象的 MD5 算法校验值，ETag 的值可以用于检查 Object 的内容是否发生变化	String
LastModified	返回对象最后修改时间，GMT 格式	String

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

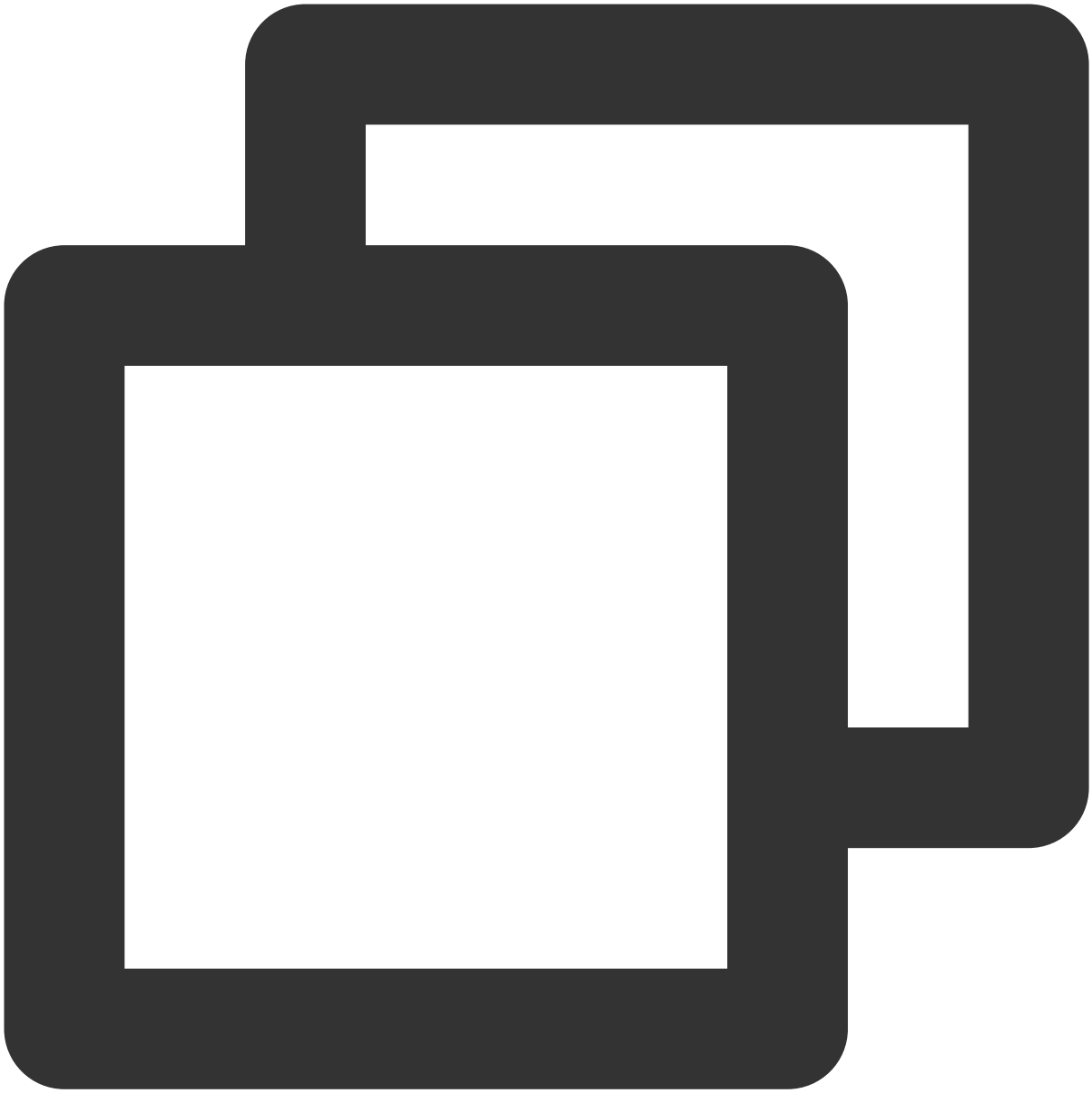
实际案例

请求



```
PUT /exampleobject?partNumber=1&uploadId=1505706248ca8373f8a5cd52cb129f4bcf85e11dc8
User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.13.1.0 zlib/
Accept: */*
x-cos-copy-source:examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/exampleobje
x-cos-copy-source-range: bytes=10-100
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3M****&q-sig
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 133
Connection: keep-alive
Date: Mon, 04 Sep 2017 04:45:45 GMT
Server: tencent-cos
x-cos-request-id: NTlkYjFjYWJfMjQ4OGY3MGFfNGIz****

<CopyPartResult>
   <ETag>"ba82b57cfdfda8bd17ad4e5879ebb4fe"</ETag>
   <LastModified>2017-09-04T04:45:45</LastModified>
```

```
</CopyPartResult>
```

版权所有：腾讯云计算（北京）有限责任公司

Complete Multipart Upload

最近更新时间：2023-12-21 15:06:08

功能描述

Complete Multipart Upload 接口请求用来实现完成整个分块上传。当使用 Upload Part 上传所有分块完成后，必须调用该 API 来完成整个文件的分块上传。在使用该 API 时，您必须在请求体中给出每一个块的 PartNumber 和 ETag，用来校验块的准确性。
由于分块上传完成后需要合并，而合并需要数分钟时间，因而当合并分块开始时，COS 会立即返回200的状态码并指定 Transfer-Encoding: chunked 响应头部，在合并的过程中，COS 会周期性的使用 chunked 方式返回空格信息来保持连接活跃，直到合并完成，COS 会在最后一个 chunk 中返回合并完成后整个对象的信息。
当上传的分块小于1MB的时候，在调用该 API 时，会返回400 EntityTooSmall。
当上传块编号不连续的时候，在调用该 API 时，会返回400 InvalidPart。
当请求 Body 中的块信息没有按序号从小到大排列的时候，在调用该 API 时，会返回400 InvalidPartOrder。
当 UploadId 不存在的时候，在调用该 API 时，会返回404 NoSuchUpload。

说明

只有主账号或者被授权 Complete Multipart Upload 接口权限的子账号可以调用该接口。

注意

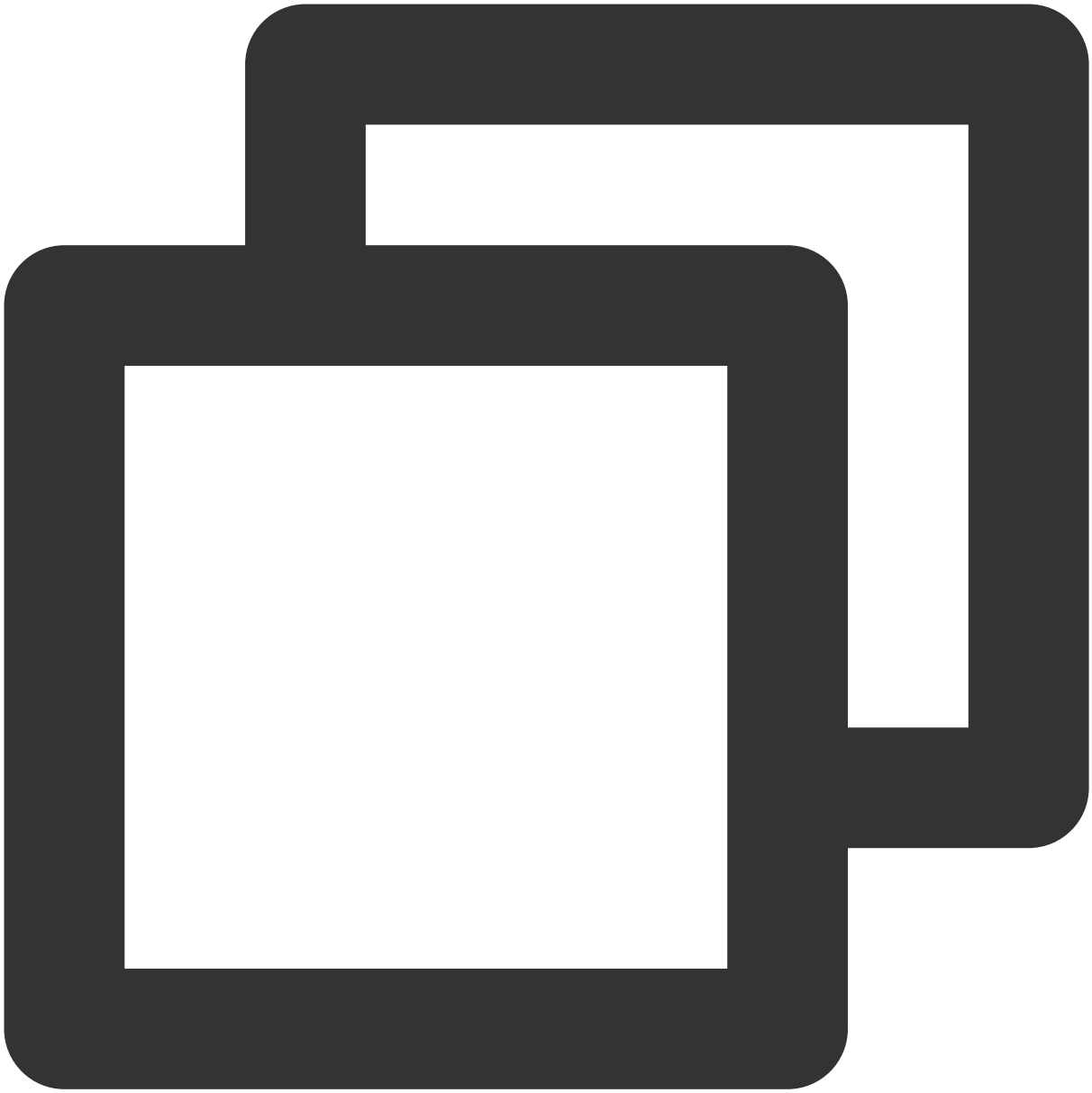
建议您及时完成分块上传或者舍弃分块上传，因为已上传但是未终止的块会占用存储空间进而产生存储费用。

版本控制

如果对存储桶启用版本控制，对象存储将自动为要添加的对象生成唯一的版本 ID。对象存储使用 x-cos-version-id 响应头部在响应中返回此标识。
如果暂停存储桶的版本控制，则对象存储始终将 null 用作存储在存储桶中的对象的版本 ID，并返回 x-cos-version-id: null 响应头部。

请求

请求示例



```
POST /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。
Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

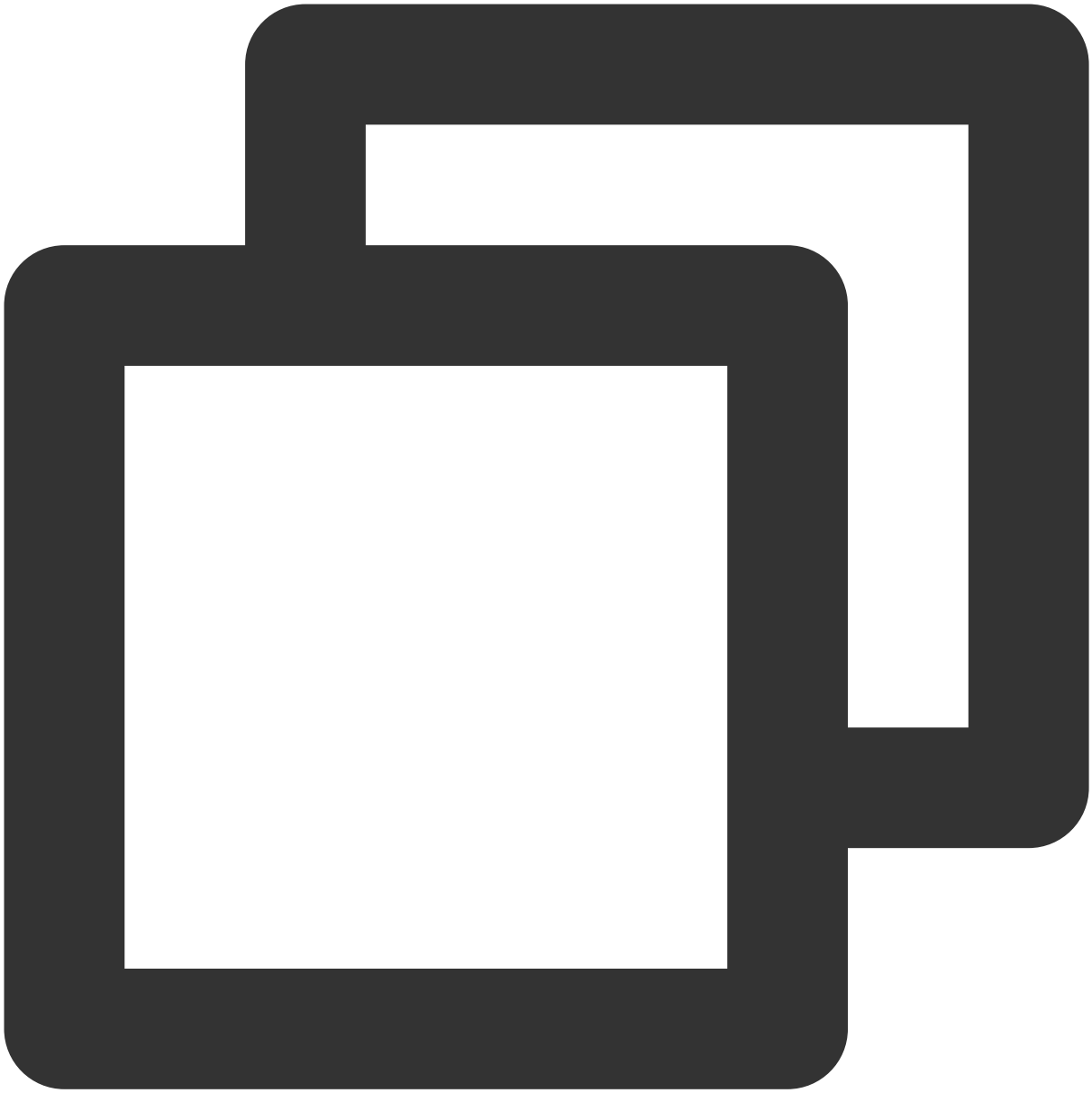
名称	描述	类型	是否必选
uploadId	标识本次分块上传的 ID，使用 Initiate Multipart Upload 接口初始化分块上传时得到的 UploadId。	string	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

提交 **application/xml** 请求数据，包含所有分块信息。



```
<CompleteMultipartUpload>
    <Part>
        <PartNumber>integer</PartNumber>
        <ETag>string</ETag>
    </Part>
    <Part>
        <PartNumber>integer</PartNumber>
        <ETag>string</ETag>
    </Part>
</CompleteMultipartUpload>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
CompleteMultipartUpload	无	包含 Complete Multipart Upload 操作的所有请求信息	Container	否

Container 节点 CompleteMultipartUpload 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Part	CompleteMultipartUpload	用来说明本次分块上传中每个块的信息	Container	是

Container 节点 Part 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
PartNumber	CompleteMultipartUpload.Part	块编号	integer	是
ETag	CompleteMultipartUpload.Part	使用 Upload Part 请求上传分块成功后返回的 ETag 响应头部的值	string	是

响应

响应头

此接口除返回公共响应头部外，还返回以下响应头部，了解公共响应头部详情请参见 公共响应头部 文档。

版本控制相关头部

在启用版本控制的存储桶中上传对象，将返回下列响应头部：

名称	描述	类型
x-cos-version-id	对象的版本 ID	string

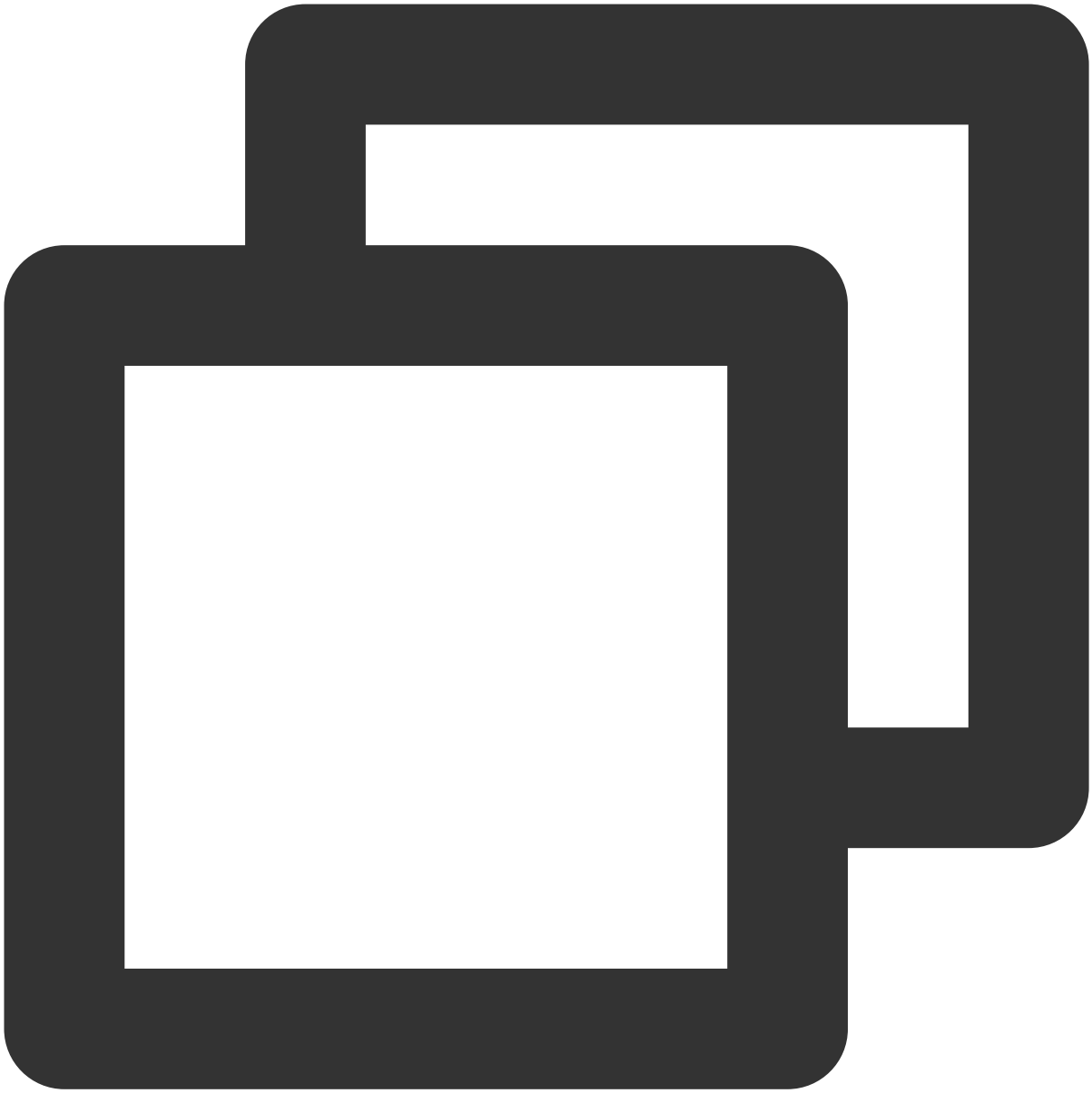
服务端加密（SSE）相关头部

如果在上传对象时使用了服务端加密，则此接口将返回服务端加密专用头部，请参见 服务端加密专用头部。

响应体

请求成功，返回 **application/xml** 数据，包含合并完成后的整个对象信息。

版权所有：腾讯云计算（北京）有限责任公司



```
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
    <Location>string</Location>
    <Bucket>string</Bucket>
    <Key>string</Key>
    <ETag>string</ETag>
</CompleteMultipartUploadResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型

CompleteMultipartUploadResult	无	保存 Complete Multipart Upload 结果的所有信息	Container

Container 节点 CompleteMultipartUploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
Location	CompleteMultipartUploadResult	分块上传完成的对象位置，格式为 `http://<BucketName-APPID>.cos.<Region>.myqcloud.com/<ObjectKey>`，例如 `http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleobject`	string
Bucket	CompleteMultipartUploadResult	分块上传的目标存储桶，格式为 `<BucketName-APPID>`，例如 `examplebucket-1250000000`	string
Key	CompleteMultipartUploadResult	对象键	string
ETag	CompleteMultipartUploadResult	分块合并后的对象 ETag 值	string

错误码

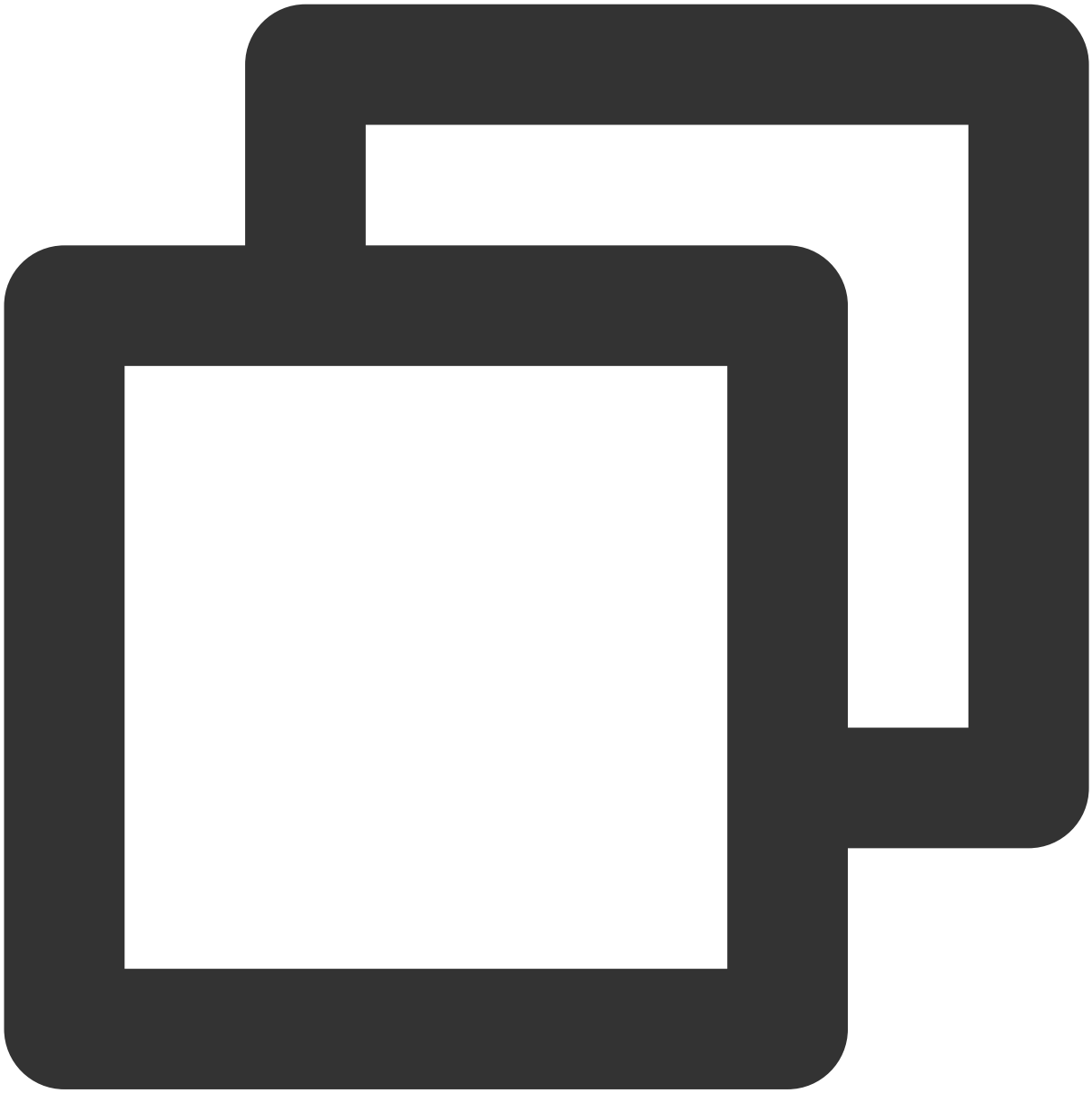
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

此接口响应默认为 Transfer-Encoding: chunked 编码方式，为了方便阅读，本文档实际案例均采用无 Transfer-Encoding 的方式展示，在使用过程中，不同语言和库可以自动处理这种编码形式，请开发者注意识别和处理，更多信息请查阅语言和库的相关文档。

案例一：简单案例（未启用版本控制）

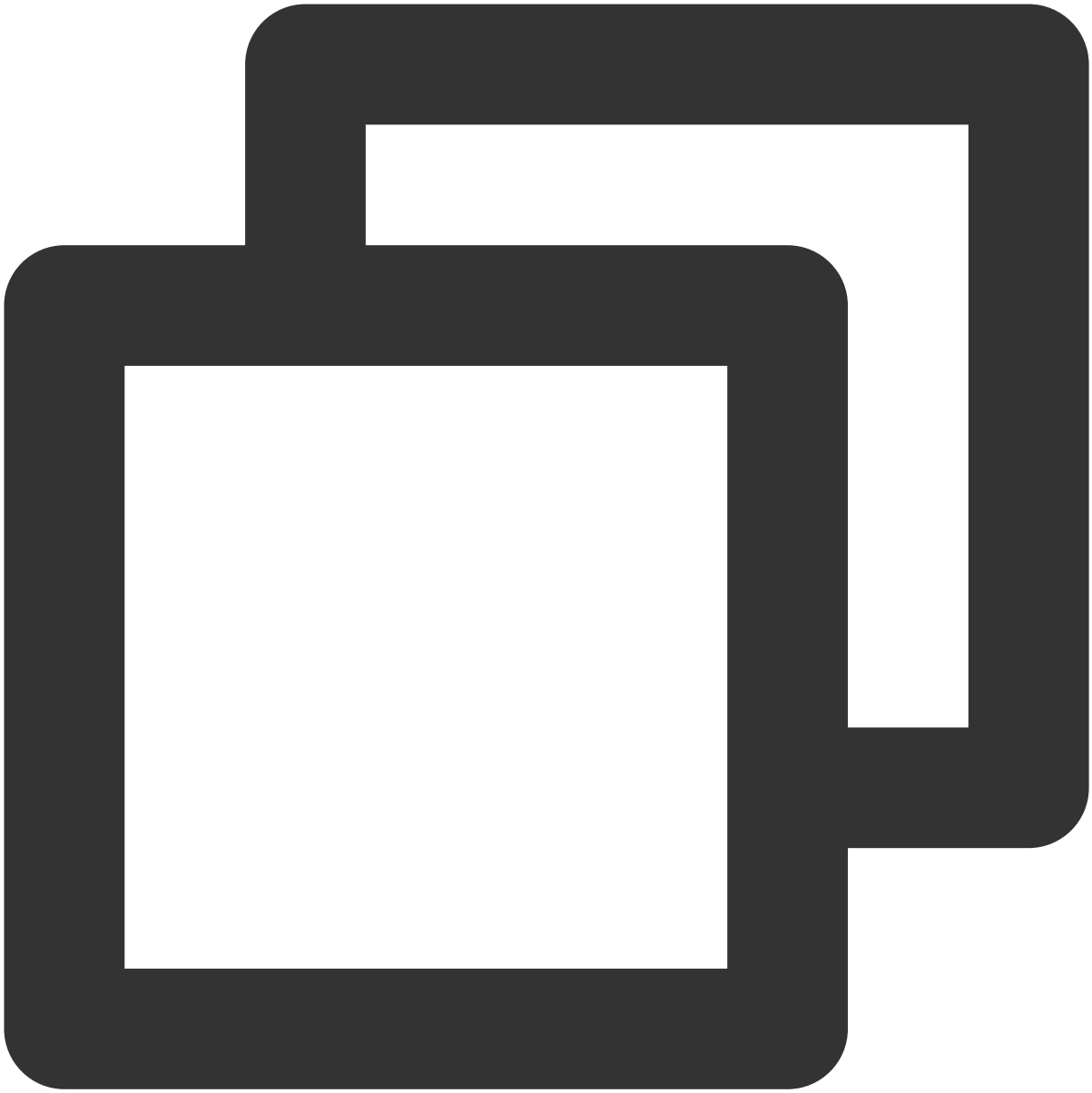
请求



```
POST /exampleobject?uploadId=1585130821cbb7df1d11846c073ad648e8f33b087cec2381df437a
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 25 Mar 2020 10:07:26 GMT
Content-Type: application/xml
Content-Length: 353
Content-MD5: Me/0Gvtc2x4VPIOhoIRllw==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CompleteMultipartUpload>
    <Part>
```

```
        <PartNumber>1</PartNumber>
        <ETag>"39270a968a357d24207e9911162507eb"</ETag>
    </Part>
    <Part>
        <PartNumber>2</PartNumber>
        <ETag>"d899fbd1e06109ea2e4550f5751c88d6"</ETag>
    </Part>
    <Part>
        <PartNumber>3</PartNumber>
        <ETag>"762890d6c9a871b7bd136037cb2260cd"</ETag>
    </Part>
</CompleteMultipartUpload>
```

响应

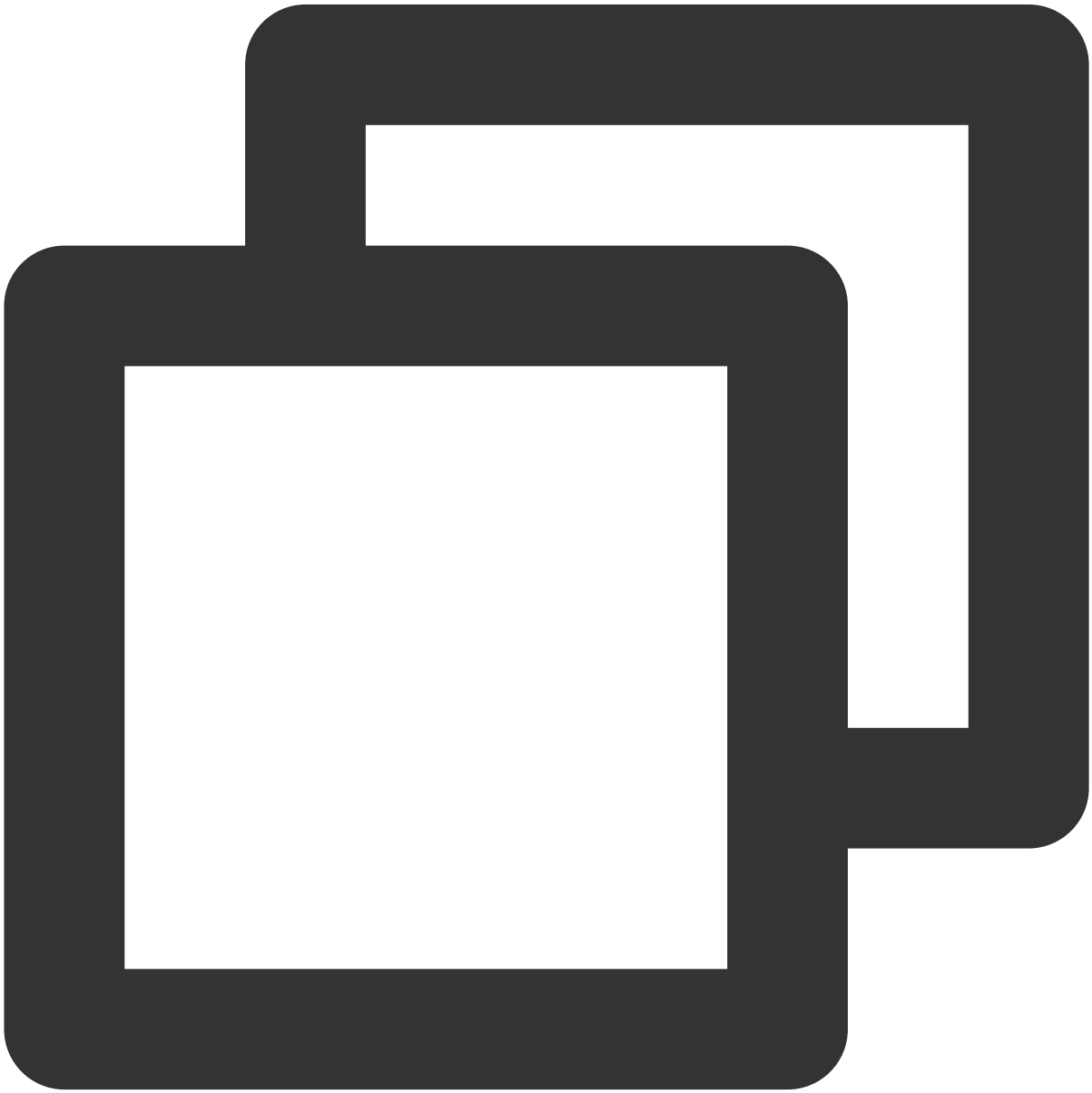


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 378
Connection: close
Date: Wed, 25 Mar 2020 10:07:26 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 2290339086971918696
x-cos-request-id: NWU3YjJkNWVfZDFjODJhMDlfMTk2ODJfMmEyNTA0****

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
```

```
    <Location>http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleob
    <Bucket>examplebucket-1250000000</Bucket>
    <Key>exampleobject</Key>
    <ETag>&amp;quot;aa259a62513358f69e98e72e59856d88-3&amp;quot;</ETag>
</CompleteMultipartUploadResult>
```

案例二：使用服务端加密 SSE-COS

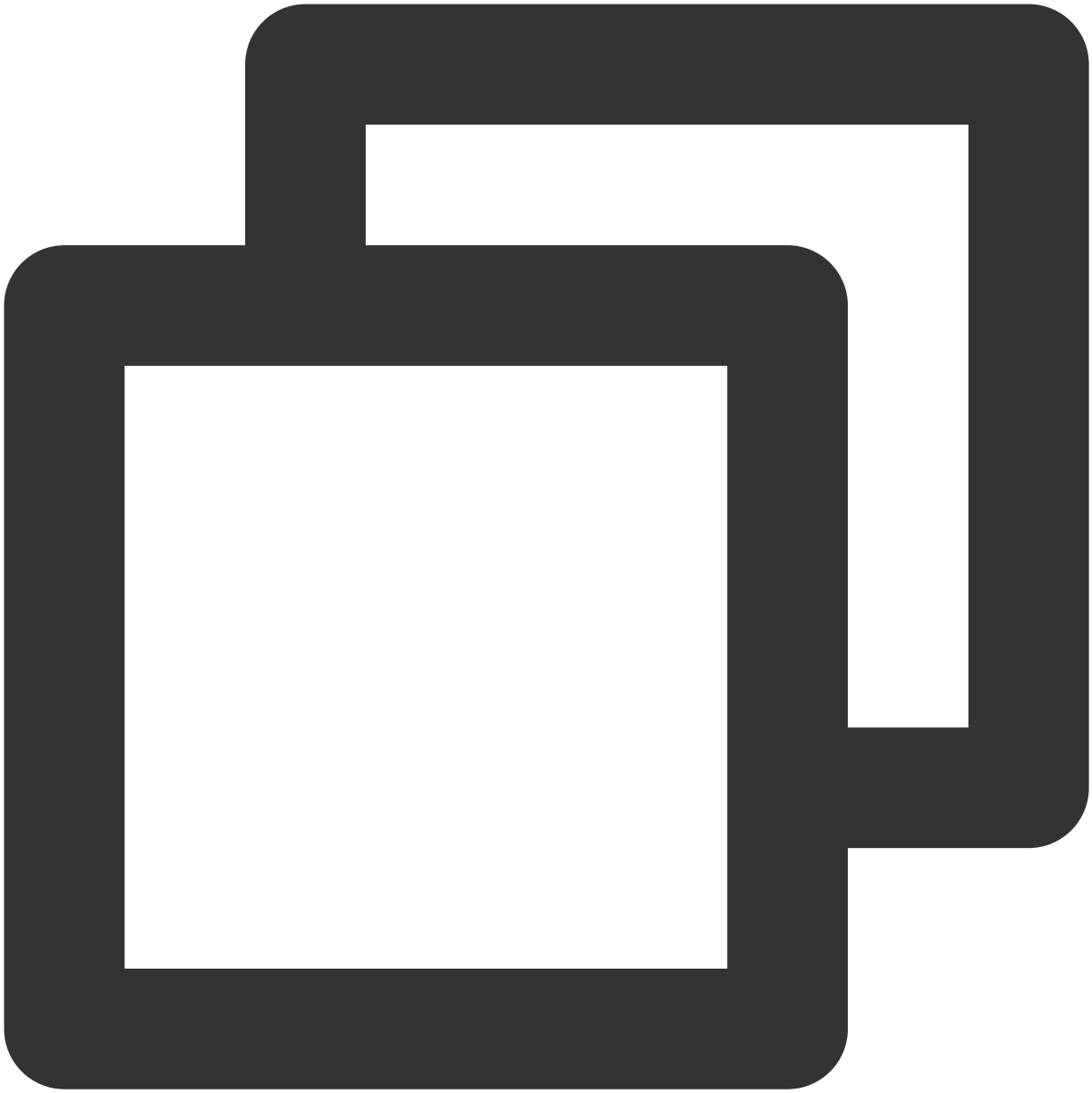
请求



```
POST /exampleobject?uploadId=159086254025135529 5a5c736513d70d42b92bbde4f0fafb5e0ecb
```

```
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:16:08 GMT
Content-Type: application/xml
Content-Length: 153
Content-MD5: wmpyfdr9s/A+VCBwDdC1bA==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CompleteMultipartUpload>
    <Part>
        <PartNumber>1</PartNumber>
        <ETag>"108f528eb63b65e734f27ae331f5ffaa"</ETag>
    </Part>
</CompleteMultipartUpload>
```

响应

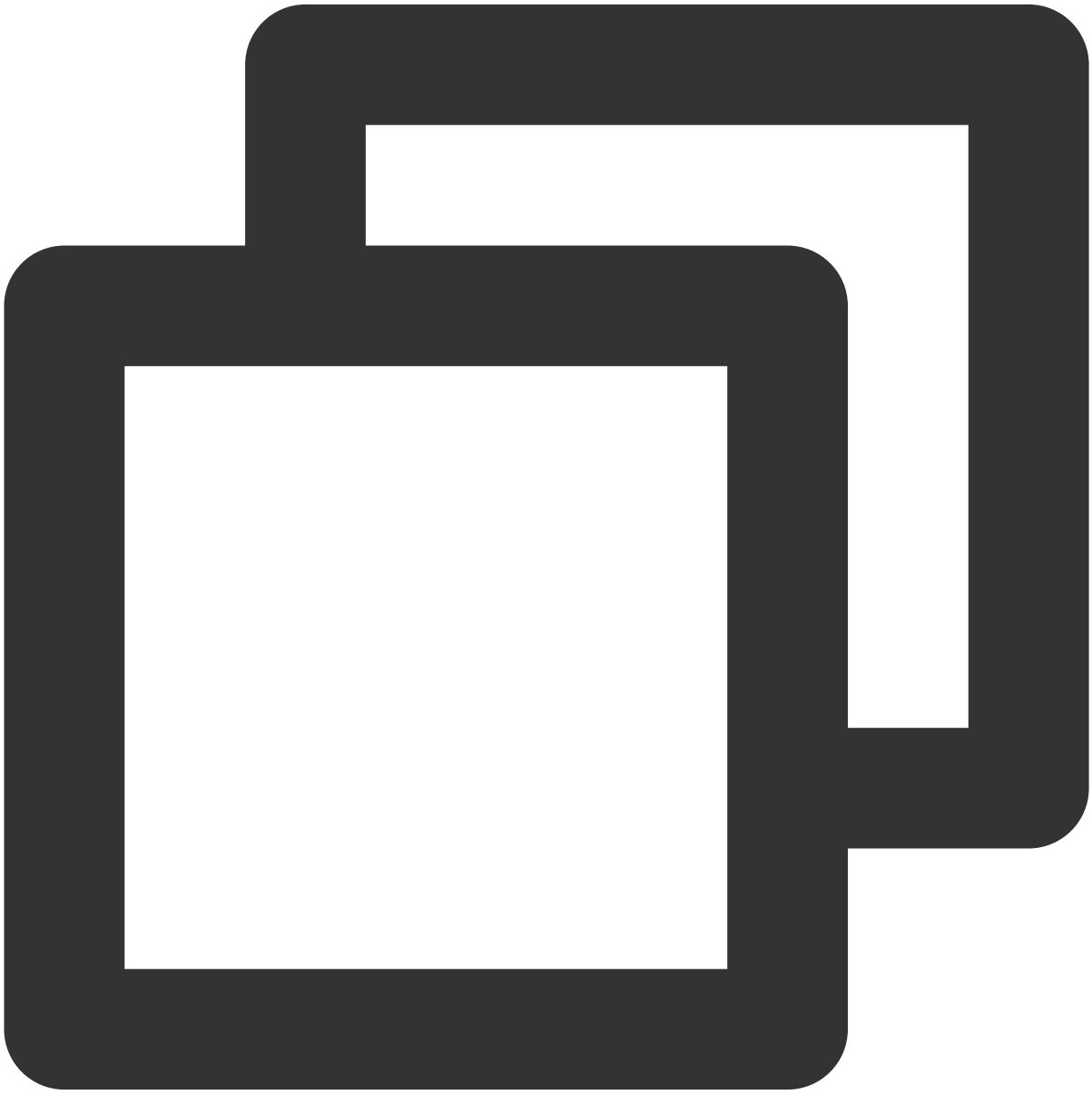


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 378
Connection: close
Date: Sat, 30 May 2020 18:16:08 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEyZThfYTJjMjJhMDlfMmQzOV8zYTM5****
x-cos-server-side-encryption: AES256

<?xml version="1.0" encoding="UTF-8"?>
```

```
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
    <Location>http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleob
    <Bucket>examplebucket-1250000000</Bucket>
    <Key>exampleobject</Key>
    <ETag>&amp;quot;915fca1c3b2737c262458b3a1a43c683-1&amp;quot;</ETag>
</CompleteMultipartUploadResult>
```

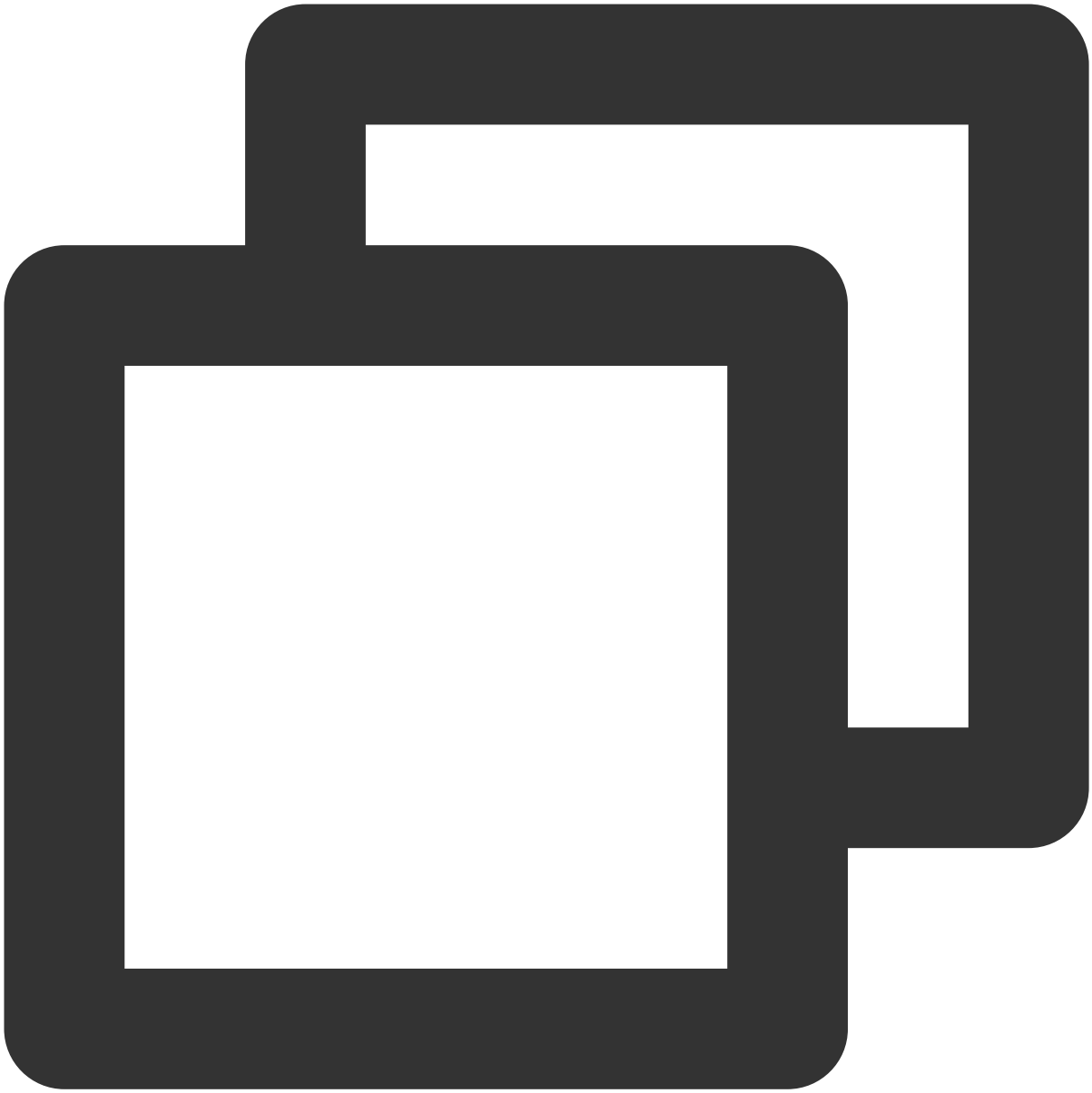
案例三：使用服务端加密 SSE-KMS

请求



```
POST /exampleobject?uploadId=15908625793957d71176fa878282d266a95b79dc2aec159b4a73d1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:16:40 GMT
Content-Type: application/xml
Content-Length: 153
Content-MD5: jUFgTSZThmPT3qseea0mRQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CompleteMultipartUpload>
    <Part>
        <PartNumber>1</PartNumber>
        <ETag>"1d3e8ae9da423b440341b09f1616f074"</ETag>
    </Part>
</CompleteMultipartUpload>
```

响应

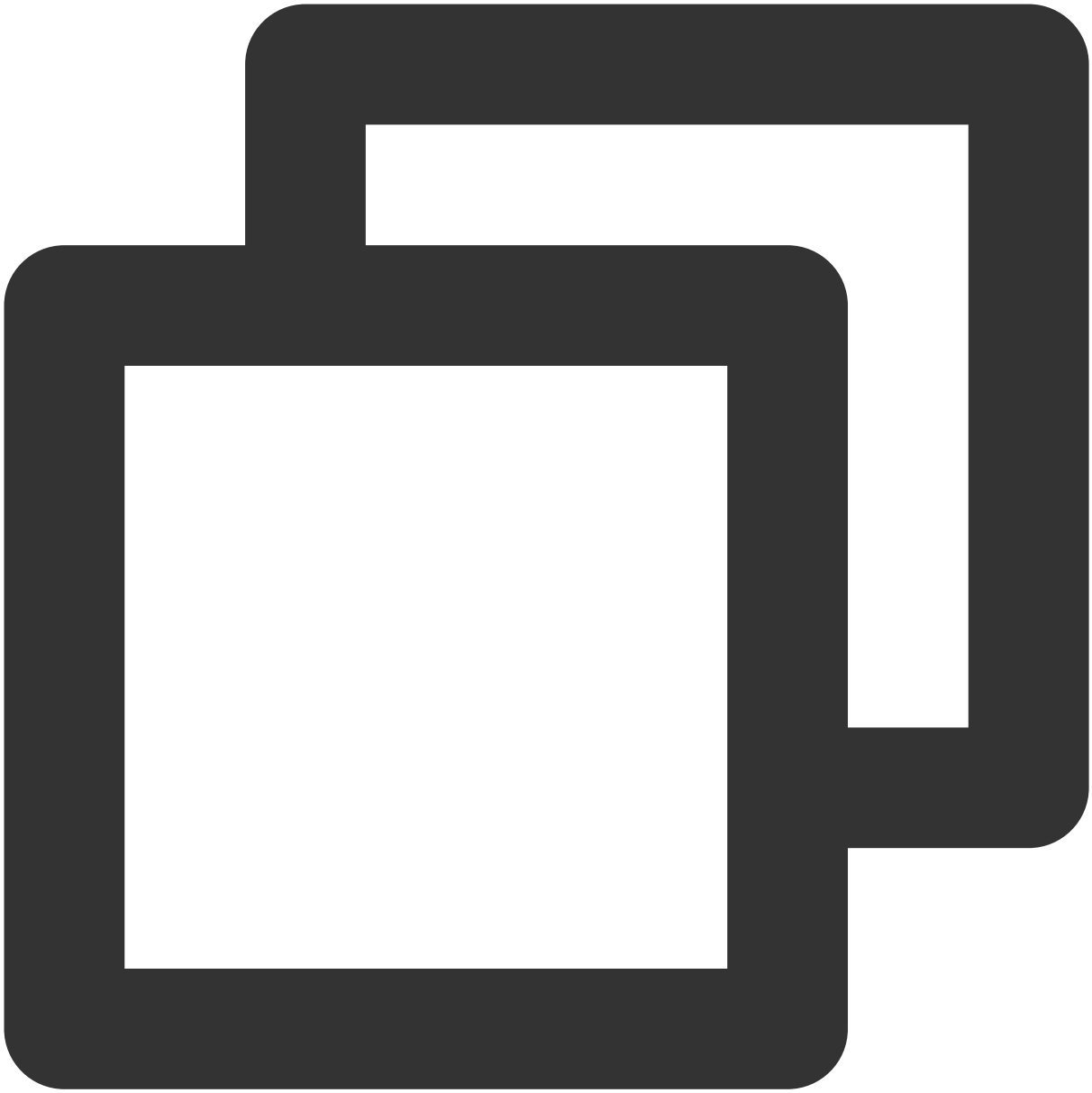


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 378
Connection: close
Date: Sat, 30 May 2020 18:16:40 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEzMDhfYzhjODJhMDlfMjNhYTBfMjdmNjNl****
x-cos-server-side-encryption: cos/kms
x-cos-server-side-encryption-cos-kms-key-id: 48ba38aa-26c5-11ea-855c-52540085****
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
    <Location>http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleob
    <Bucket>examplebucket-1250000000</Bucket>
    <Key>exampleobject</Key>
    <ETag>&amp;quot;8093dc3e18f7070444e6ca21789eb8d4-1&amp;quot;</ETag>
</CompleteMultipartUploadResult>
```

案例四：使用服务端加密 SSE-C

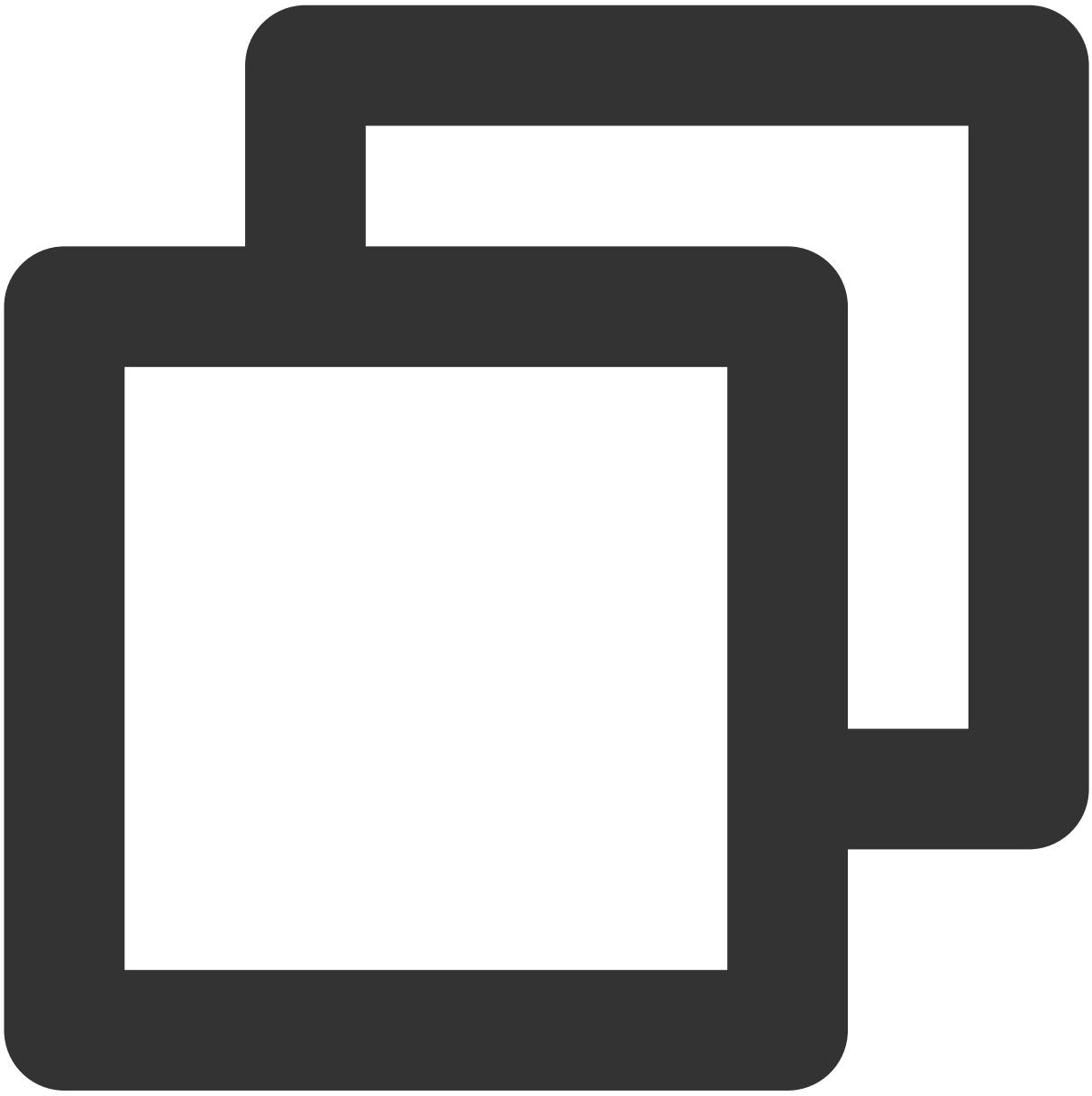
请求



```
POST /exampleobject?uploadId=1590862610acd643927bad05cd3947bf98b56b04b0b0b2a45a4996
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:17:11 GMT
Content-Type: application/xml
Content-Length: 153
Content-MD5: MnXmb3yvrbwMfBlUwUE1Hg==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CompleteMultipartUpload>
    <Part>
```

```
        <PartNumber>1</PartNumber>
        <ETag>"ff14981a35a58eb97bca838b055c573b"</ETag>
    </Part>
</CompleteMultipartUpload>
```

响应

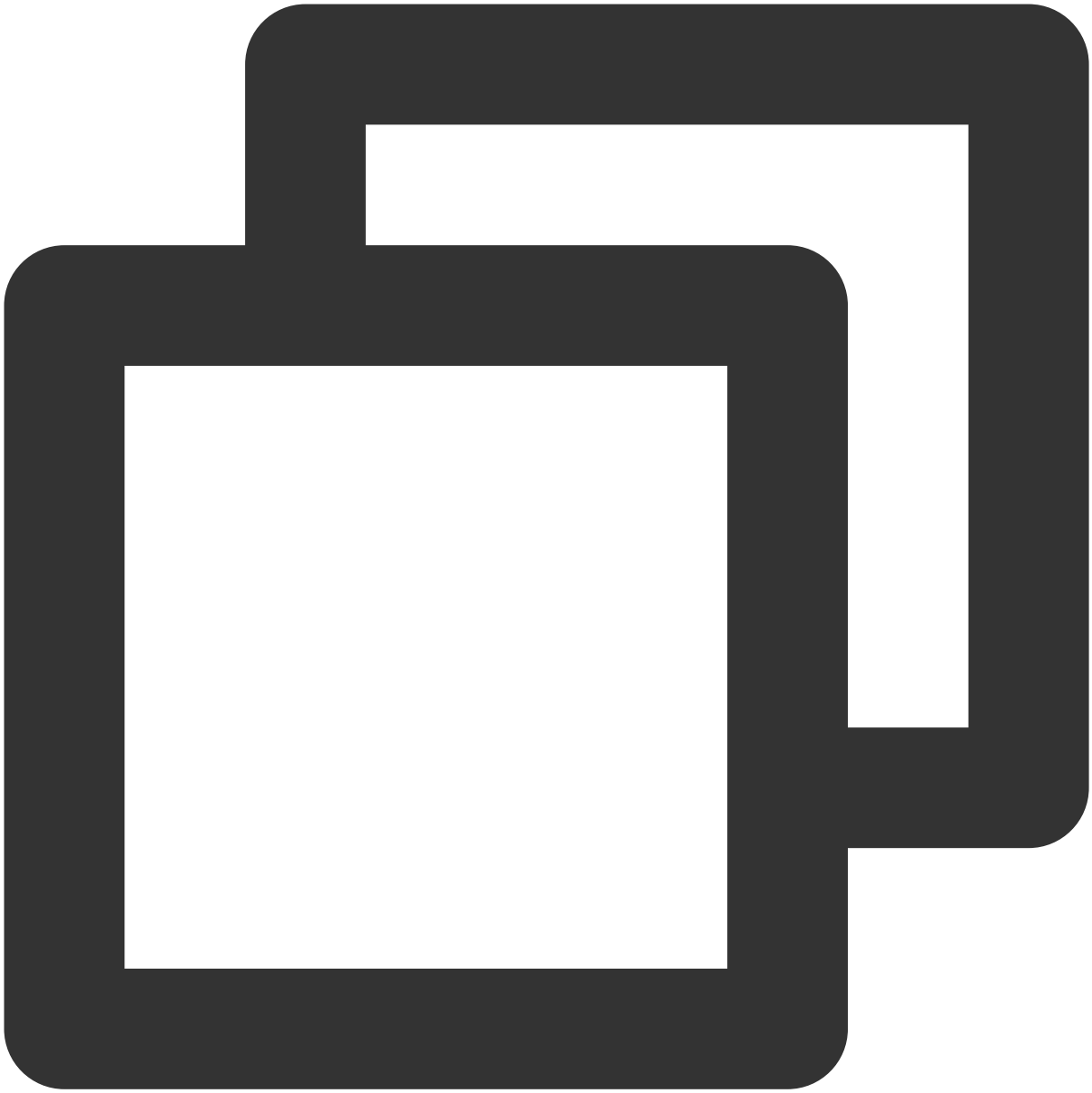


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 378
Connection: close
```

```
Date: Sat, 30 May 2020 18:17:11 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEzMjdfODljOTJhMDlfMzQ2ZDNfMWMwZWYy****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7frmUG8g==

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
    <Location>http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleob
    <Bucket>examplebucket-1250000000</Bucket>
    <Key>exampleobject</Key>
    <ETag>&amp;quot;55973f71e8e892273053617e6b83d1c7-1&amp;quot;</ETag>
</CompleteMultipartUploadResult>
```

案例五：启用版本控制

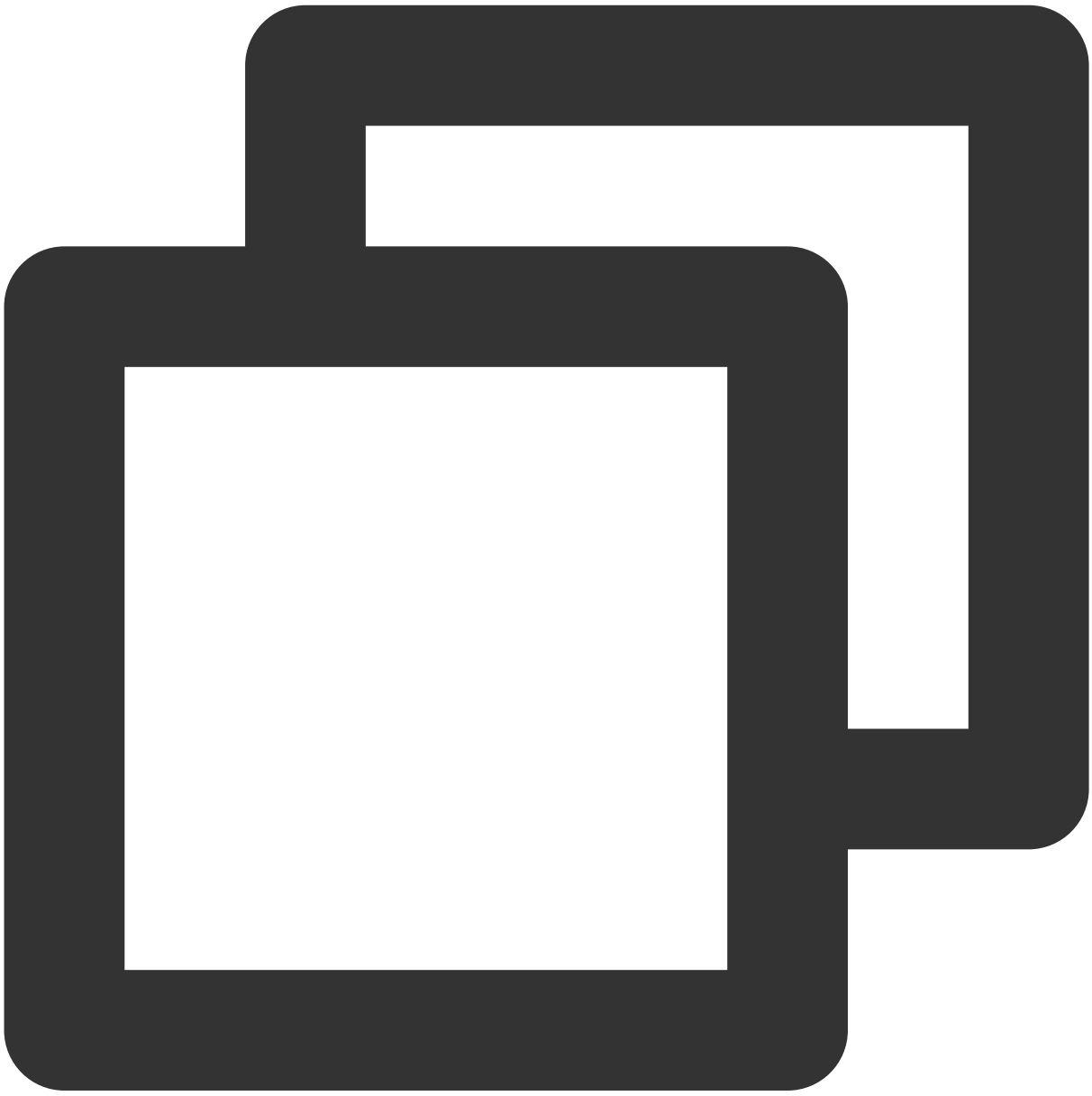
请求



```
POST /exampleobject?uploadId=15908626631e1995018f81a5e563837d6d7f1b51d7c97dff099892
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:18:04 GMT
Content-Type: application/xml
Content-Length: 153
Content-MD5: wmpyfdr9s/A+VCBwDdC1bA==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CompleteMultipartUpload>
    <Part>
```

```
        <PartNumber>1</PartNumber>
        <ETag>"108f528eb63b65e734f27ae331f5ffaa"</ETag>
    </Part>
</CompleteMultipartUpload>
```

响应

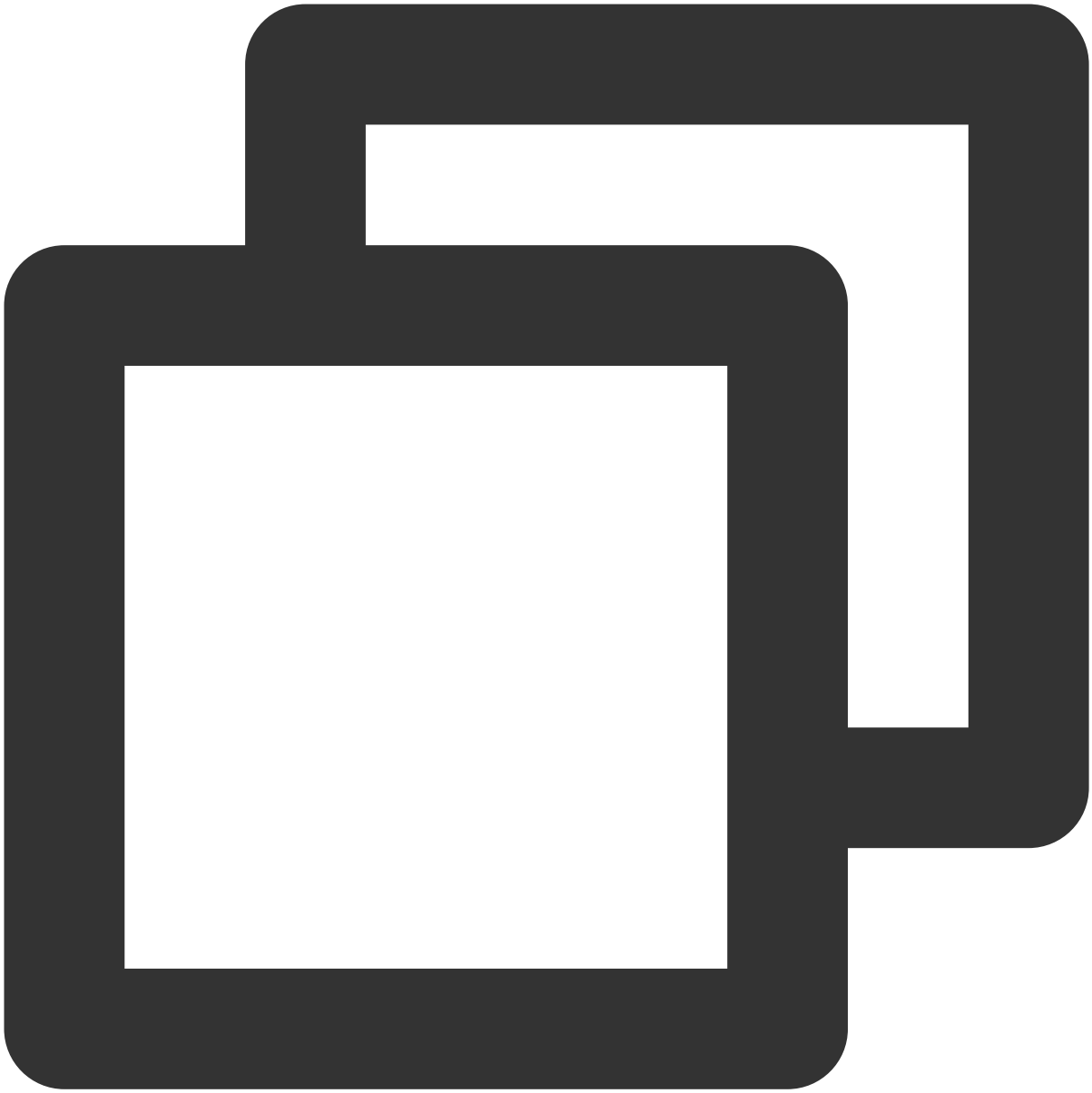


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 378
Connection: close
```

```
Date: Sat, 30 May 2020 18:18:04 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEzNWNfOGNjOTJhMDlfMWQ4NjJfMWM0NGUw****
x-cos-version-id: MTg0NDUxNTMyMTEwNDU1NDc3OTc

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
    <Location>http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleob
    <Bucket>examplebucket-1250000000</Bucket>
    <Key>exampleobject</Key>
    <ETag>&amp;quot;42318fc0ec58952b0d9ab4d7a006f595-1&amp;quot;</ETag>
</CompleteMultipartUploadResult>
```

案例六：暂停版本控制

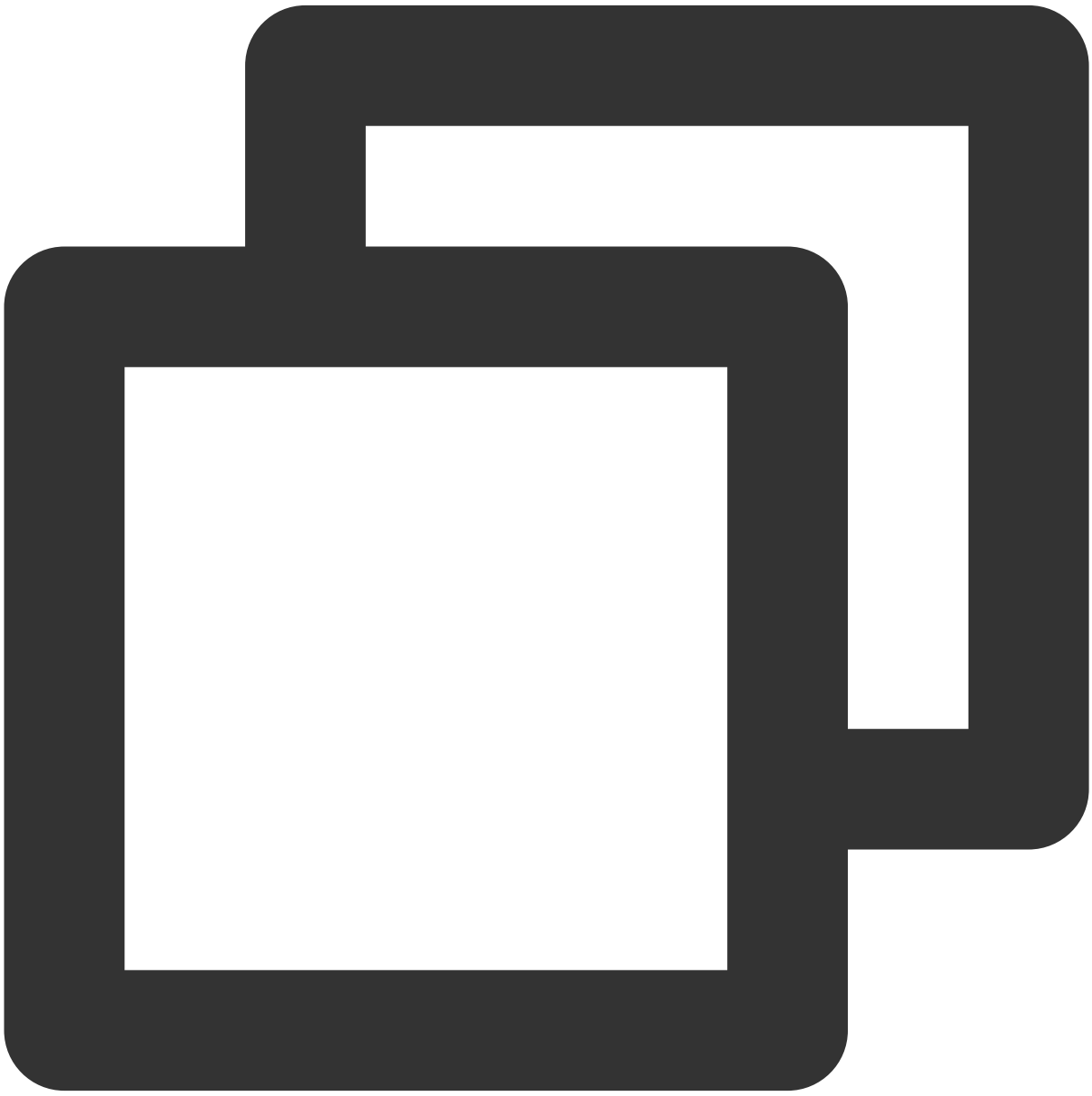
请求



```
POST /exampleobject?uploadId=1590862705b4349d597c0db1281e1c666bc431eb468b3a64076be7
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sat, 30 May 2020 18:18:45 GMT
Content-Type: application/xml
Content-Length: 153
Content-MD5: wmpyfdr9s/A+VCBwDdC1bA==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CompleteMultipartUpload>
    <Part>
```

```
        <PartNumber>1</PartNumber>
        <ETag>"108f528eb63b65e734f27ae331f5ffaa"</ETag>
    </Part>
</CompleteMultipartUpload>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 378
Connection: close
```

```
Date: Sat, 30 May 2020 18:18:45 GMT
Server: tencent-cos
x-cos-hash-crc64ecma: 4510578591875220731
x-cos-request-id: NWVkMmEzODVfZjZjMjBiMDlfNGYzZV8zODc4****
x-cos-version-id: null

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://www.qcloud.com/document/product/436/77
    <Location>http://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/exampleob
    <Bucket>examplebucket-1250000000</Bucket>
    <Key>exampleobject</Key>
    <ETag>&amp;quot;491509b1fdf8e13d1f51d323c4a6d0e8-1&amp;quot;</ETag>
</CompleteMultipartUploadResult>
```

Abort Multipart Upload

最近更新时间：2023-12-21 15:06:48

功能描述

Abort Multipart Upload 用来实现舍弃一个分块上传并删除已上传的块。当您调用 Abort Multipart Upload 时，如果有正在使用这个 Upload Parts 上传块的请求，则 Upload Parts 会返回失败。当该 UploadId 不存在时，会返回404 NoSuchUpload。

说明

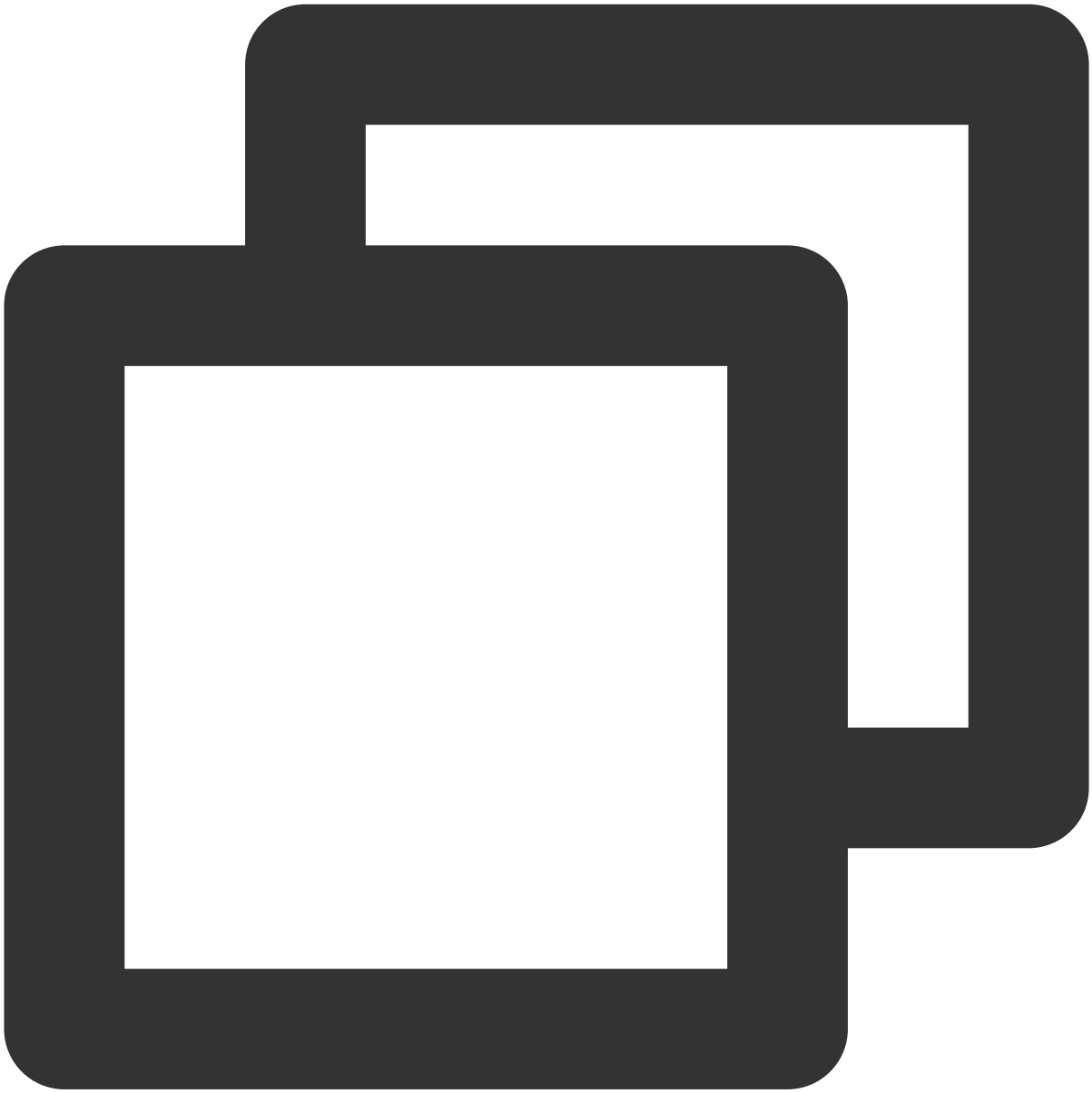
只有主账号或者被授权 Abort Multipart Upload 接口权限的子账号可以调用该接口。

注意

建议您及时完成分块上传或者舍弃分块上传，因为已上传但是未终止的块会占用存储空间进而产生存储费用。

请求

请求示例



```
DELETE /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

具体内容如下：

参数名称	描述	类型	是否必选
uploadId	标识本次分块上传的 ID，使用 Initiate Multipart Upload 接口初始化分块上传时得到的 UploadId	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

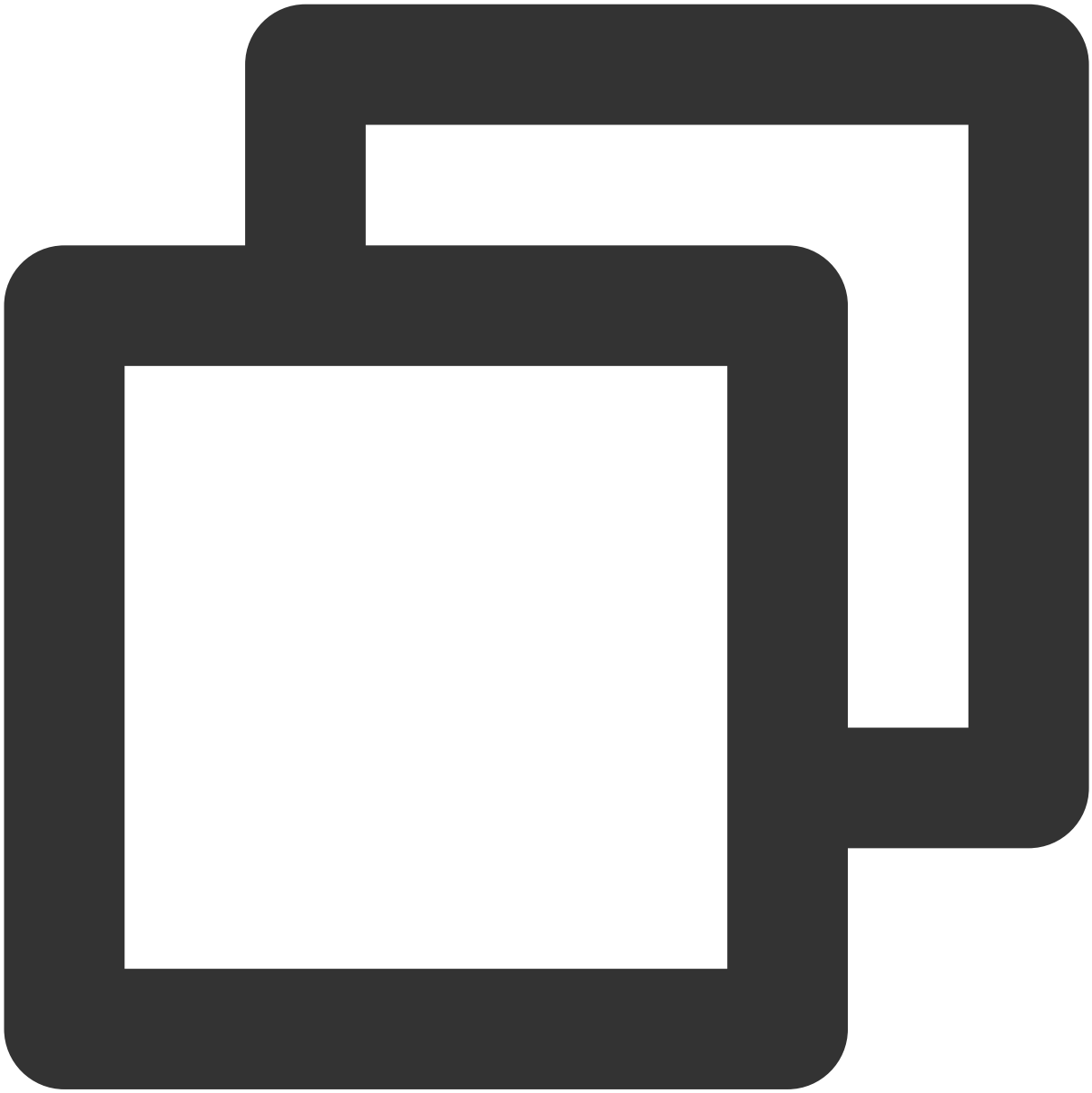
该响应体为空。

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

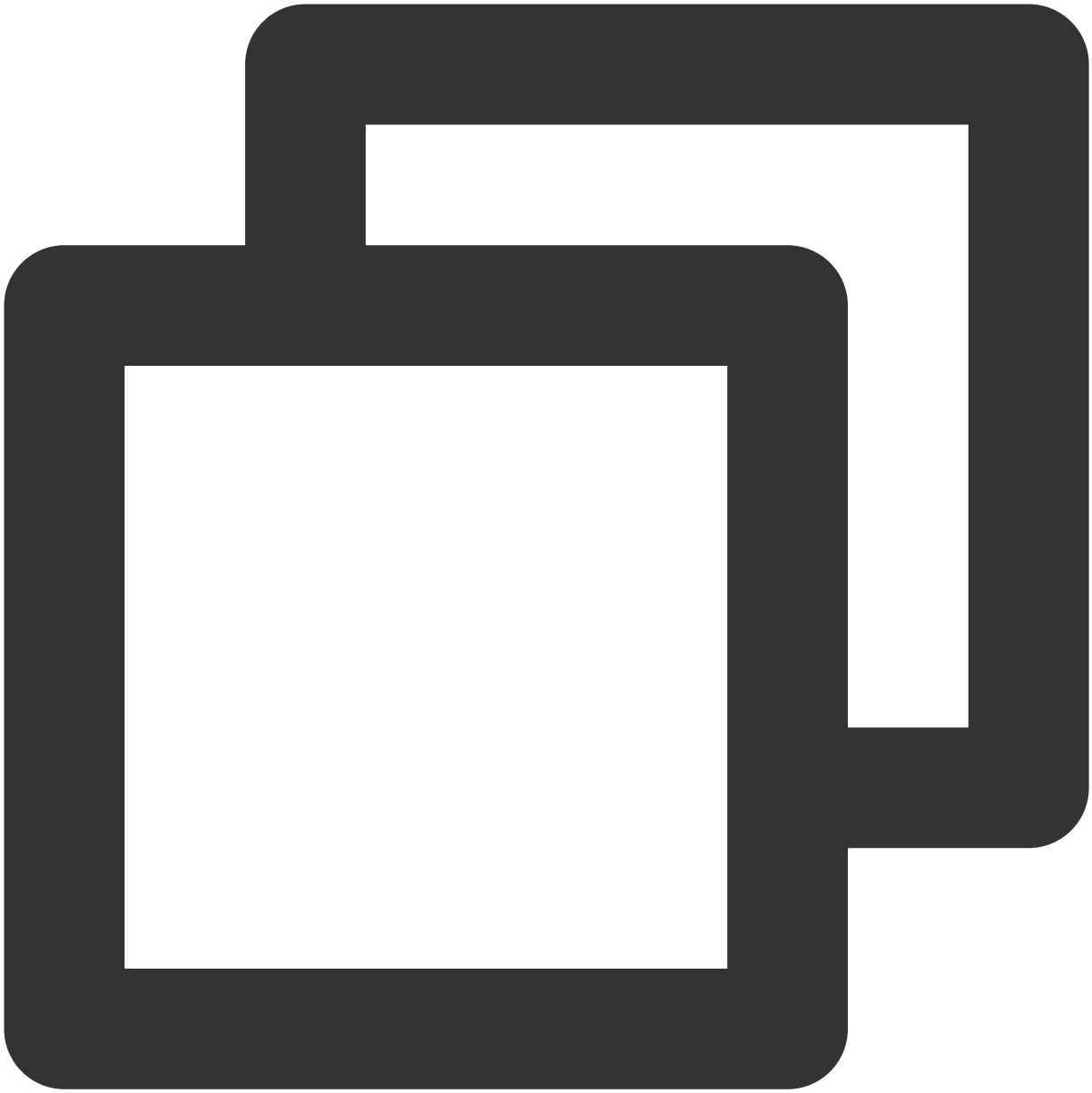
实际案例

请求



```
DELETE /exampleobject?uploadId=1484727270323ddb949d528c629235314a9ead80f0ba5d993a3d
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 26 Oct 2013 21:22:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
```

响应



```
HTTP/1.1 204 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Tue, 26 Oct 2013 21:22:00 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjI5MzlfOTgxZjRlXzZhYjNf****
```

List Multipart Uploads

最近更新时间：2023-12-21 15:06:42

功能描述

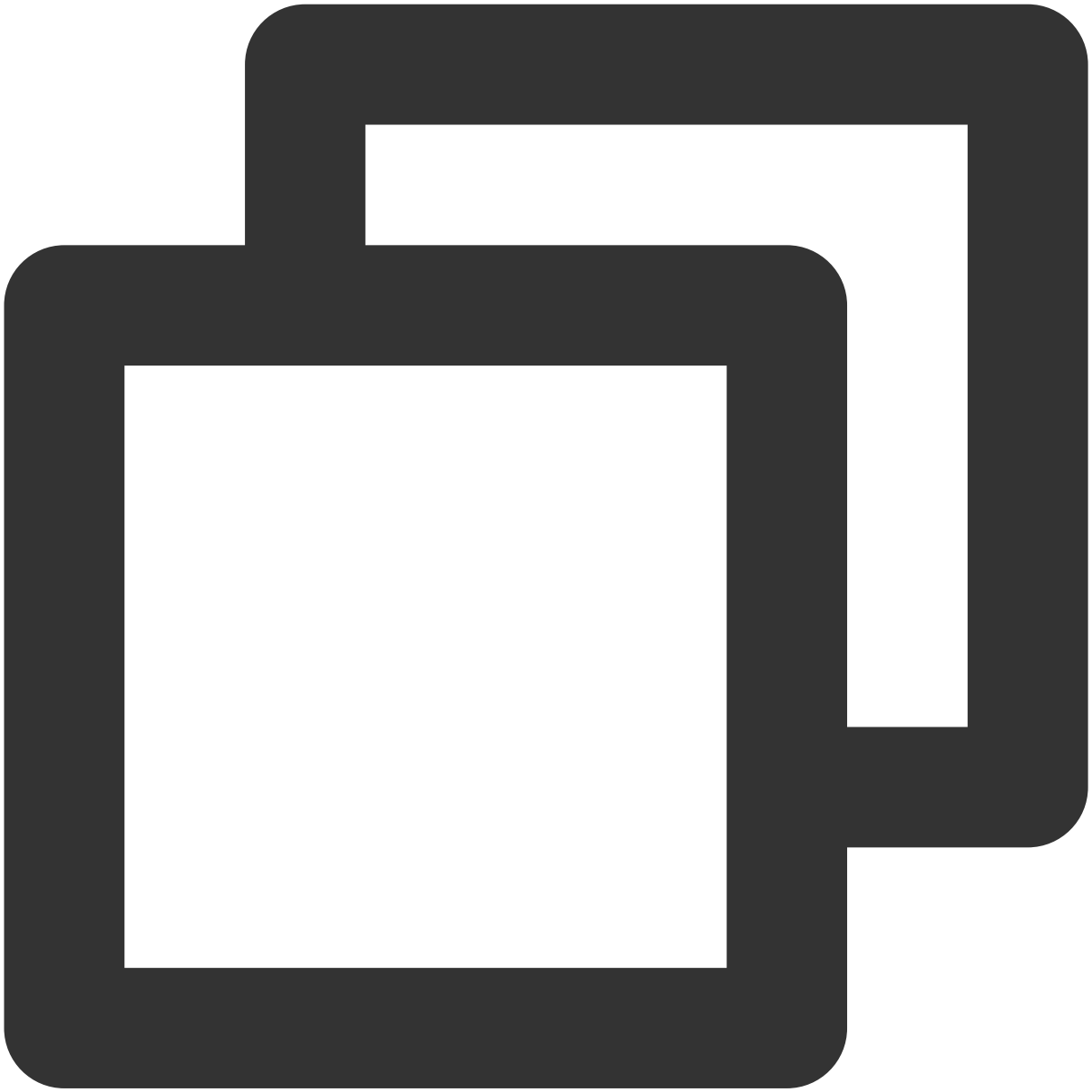
List Multipart Uploads 用于查询正在进行中的分块上传任务。单次请求操作最多列出1000个正在进行中的分块上传。

注意

该请求需要有存储桶的读权限。同时需要主账号或者被授权 List Multipart Upload 接口权限的子账号可以调用该接口。

请求

请求示例



```
GET /?uploads HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

具体内容如下：

名称	描述	类型	是否必选
delimiter	定界符为一个符号，对 Object 名字包含指定前缀且第一次出现 delimiter 字符之间的 Object 作为一组元素：common prefix。若无 prefix，则从路径起点开始。	String	否
encoding-type	规定返回值的编码格式，合法值：url。	String	否
prefix	限定返回的 Object key 必须以 Prefix 作为前缀。 注意使用 prefix 查询时，返回的 key 中仍会包含 Prefix。	String	否
max-uploads	设置最大返回的 multipart 数量，合法取值从1到1000，默认1000。	int	否
key-marker	与 upload-id-marker 一起使用： 当 upload-id-marker 未被指定时，ObjectName 字母顺序大于 key-marker 的条目将被列出。 当 upload-id-marker 被指定时，ObjectName 字母顺序大于 key-marker 的条目被列出，ObjectName 字母顺序等于 key-marker 同时 UploadId 大于 upload-id-marker 的条目将被列出。	String	否
upload-id-marker	与 key-marker 一起使用： 当 key-marker 未被指定时，upload-id-marker 将被忽略。 当 key-marker 被指定时，ObjectName字母顺序大于 key-marker 的条目被列出，ObjectName 字母顺序等于 key-marker 同时 UploadId 大于 upload-id-marker 的条目将被列出。	String	否

请求体

该请求的请求体为空。

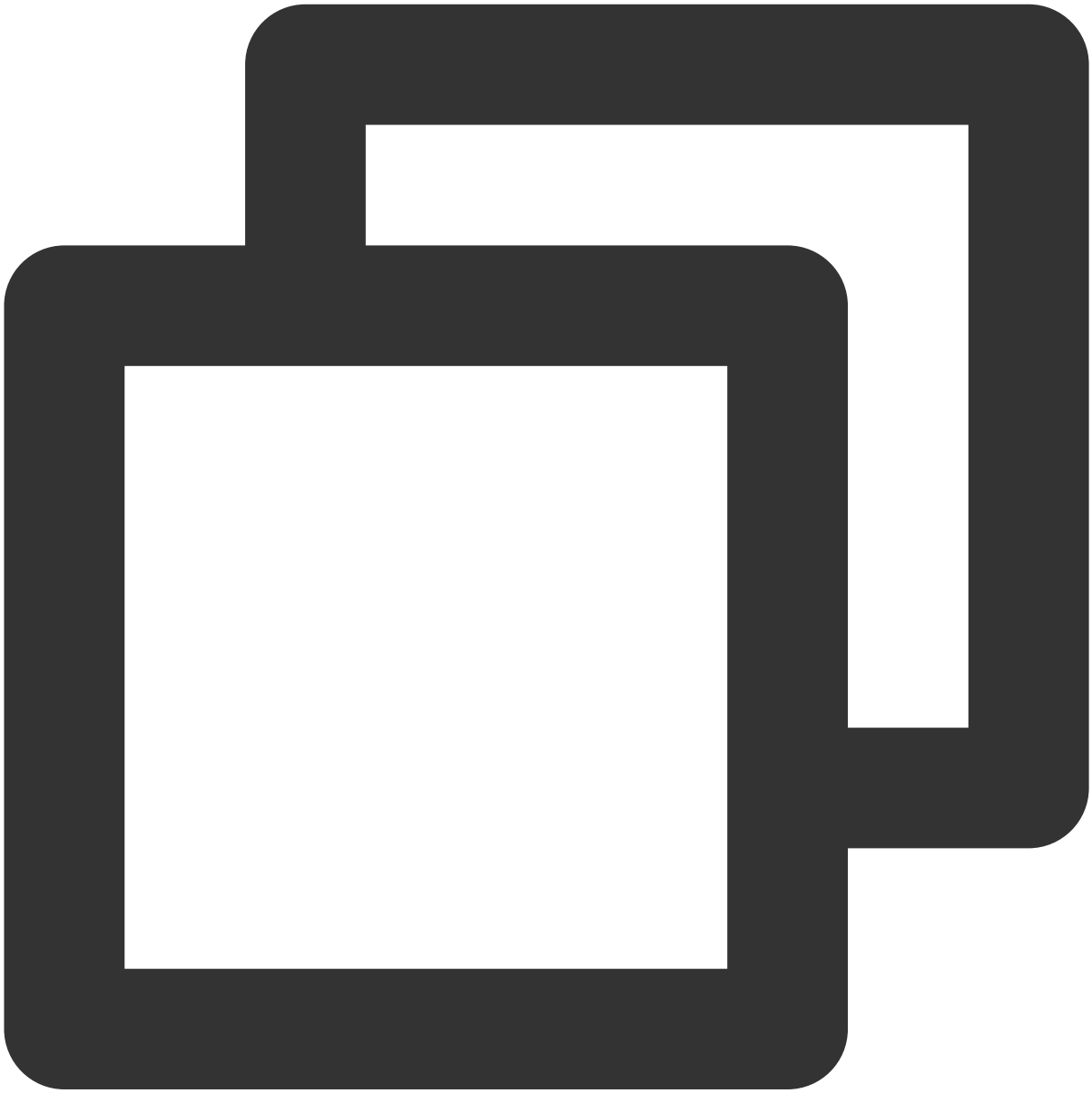
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<ListMultipartUploadsResult>
  <Bucket></Bucket>
  <Encoding-Type></Encoding-Type>
  <KeyMarker></KeyMarker>
  <UploadIdMarker></UploadIdMarker>
  <NextKeyMarker></NextKeyMarker>
  <NextUploadIdMarker></NextUploadIdMarker>
  <MaxUploads></MaxUploads>
```

```
  <IsTruncated></IsTruncated>
  <Prefix></Prefix>
  <Delimiter></Delimiter>
  <Upload>
    <Key></Key>
    <UploadId></UploadId>
    <StorageClass></StorageClass>
    <Initiator>
      <ID></ID>
    <DisplayName></DisplayName>
    </Initiator>
    <Owner>
      <ID></ID>
    <DisplayName></DisplayName>
    </Owner>
    <Initiated></Initiated>
  </Upload>
  <CommonPrefixes>
    <Prefix></Prefix>
  </CommonPrefixes>
</ListMultipartUploadsResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
ListMultipartUploadsResult	无	用来表述所有分块上传的信息。	Container

Container 节点 ListMultipartUploadsResult 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	ListMultipartUploadsResult	分块上传的目标 Bucket，由用户自定义字符串和系统生成 APPID 数字串由中划线连接而成，例如 `examplebucket-1250000000` 。	String
Encoding-Type	ListMultipartUploadsResult	规定返回值的编码格式，合法值：url。	String
KeyMarker	ListMultipartUploadsResult	列出条目从该 key 值开始。	String
UploadIdMarker	ListMultipartUploadsResult	列出条目从该 UploadId 值开始。	String
NextKeyMarker	ListMultipartUploadsResult	假如返回条目被截断，则返回的 NextKeyMarker 就是下一个条目的起点。	String
NextUploadIdMarker	ListMultipartUploadsResult	假如返回条目被截断，则返回的	String

		UploadId 就是下一个条目的起点。	
MaxUploads	ListMultipartUploadsResult	设置最大返回的 multipart 数量，合法取值从0 - 1000。	int
IsTruncated	ListMultipartUploadsResult	返回条目是否被截断，布尔值：TRUE，FALSE。	Boolean
Prefix	ListMultipartUploadsResult	限定返回的 Objectkey 必须以 Prefix 作为前缀，注意使用 prefix 查询时，返回的 key 中仍会包含 Prefix。	String
Delimiter	ListMultipartUploadsResult	定界符为一个符号，对 object 名字包含指定前缀且第一次出现 delimiter 字符之间的 object 作为一组元素：common prefix。若无 prefix，则从路径起点开始。	String
Upload	ListMultipartUploadsResult	每个 Upload 的信息。	Container
CommonPrefixes	ListMultipartUploadsResult	将 prefix 到 delimiter 之间的相同路径归为一类，定义为 Common Prefix。	Container

Container 节点 Upload 的内容：

节点名称（关键字）	父节点	描述	类型
Key	ListMultipartUploadsResult.Upload	Object 的名称。	String
UploadId	ListMultipartUploadsResult.Upload	标示本次分块上传的 ID。	String
StorageClass	ListMultipartUploadsResult.Upload	用来表示分块的存储级别，枚举值：STANDARD，STANDARD_IA，ARCHIVE。	String
Initiator	ListMultipartUploadsResult.Upload	用来表示本次上传发起者的信息。	Container
Owner	ListMultipartUploadsResult.Upload	用来表示这些分块所有者的信息。	Container
Initiated	ListMultipartUploadsResult.Upload	分块上传的起始时间。	Date

Container 节点 Initiator 的内容：

节点名称（关键字）	父节点	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

ID	ListMultipartUploadsResult.Upload.Initiator	用户唯一的 CAM 身份 ID。	String
DisplayName	ListMultipartUploadsResult.Upload.Initiator	用户身份 ID 的简称（UIN）。	String

Container 节点 Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	ListMultipartUploadsResult.Upload.Owner	用户唯一的 CAM 身份 ID。	String
DisplayName	ListMultipartUploadsResult.Upload.Owner	用户身份 ID 的简称（UIN）。	String

Container 节点 CommonPrefixes 的内容：

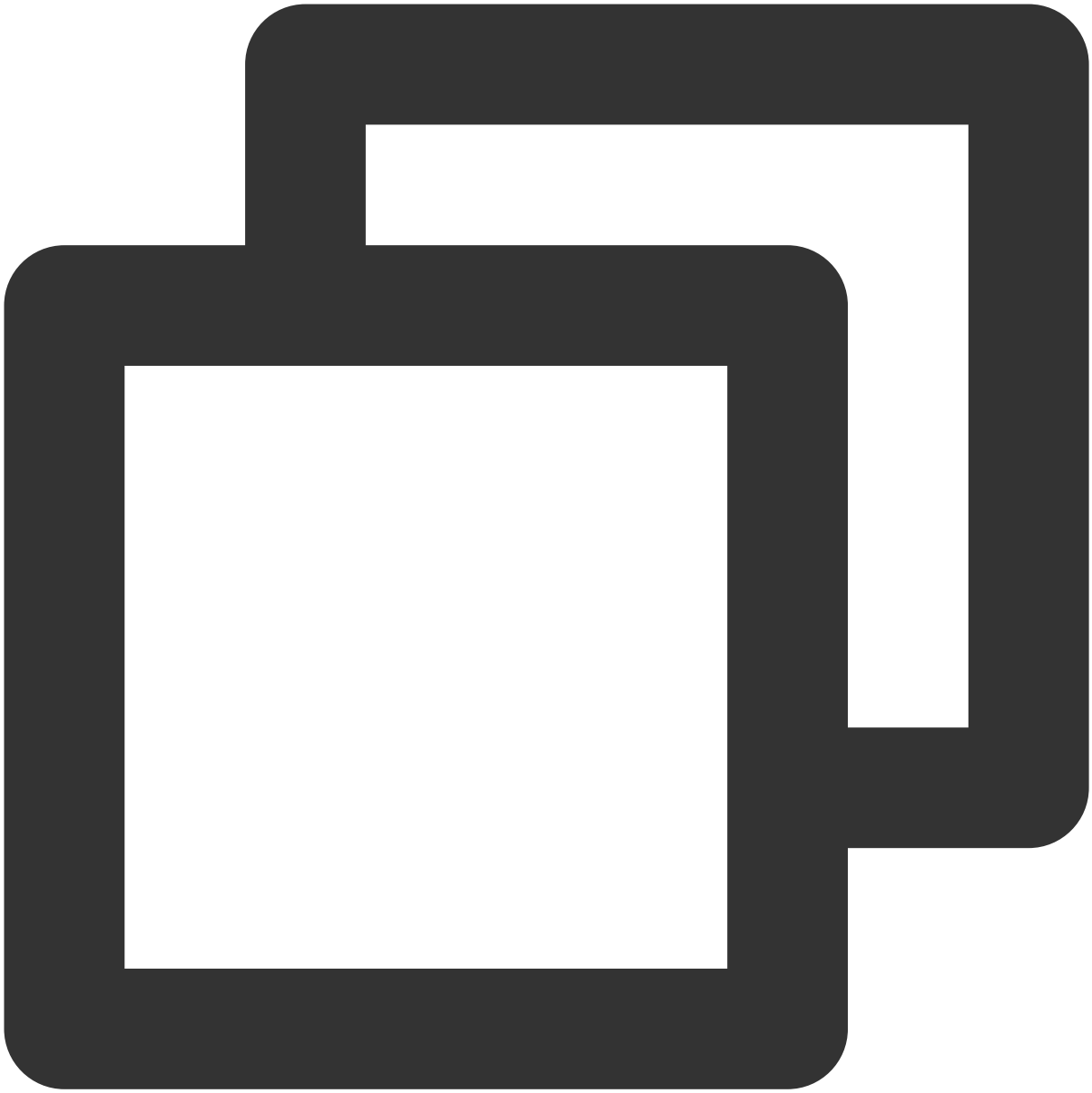
节点名称（关键字）	父节点	描述	类型
Prefix	ListMultipartUploadsResult.CommonPrefixes	显示具体的 CommonPrefixes。	String

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

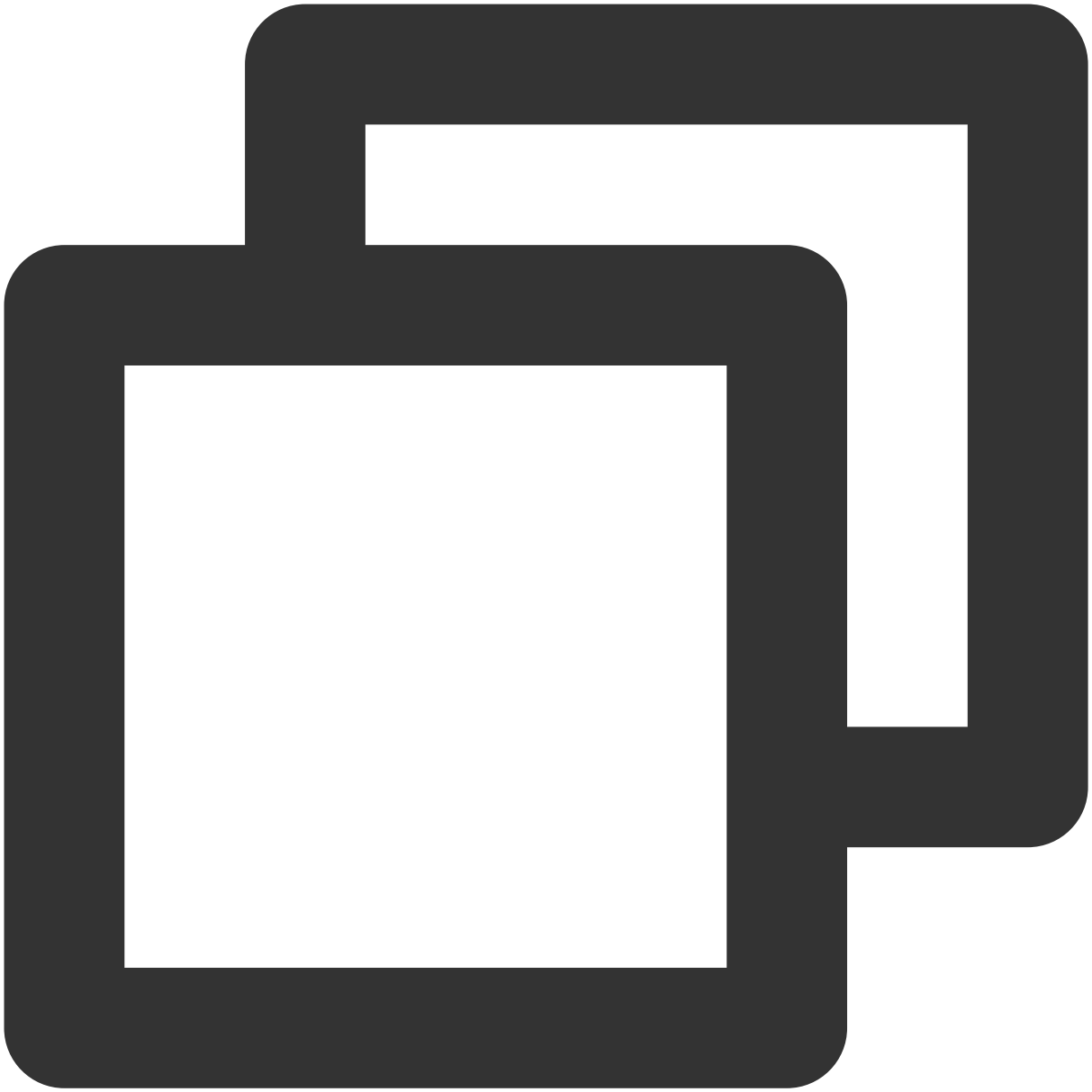
实际案例

请求



```
GET /?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2015 21:32:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1203
Date: Wed, 18 Jan 2015 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjI0ZGRfNDQyMDRlXzNhZmRf****

<ListMultipartUploadsResult>
    <Bucket>examplebucket-1250000000</Bucket>
    <Encoding-Type/>
    <KeyMarker/>
```

```
    <UploadIdMarker/>
    <MaxUploads>1000</MaxUploads>
    <Prefix/>
    <Delimiter>/</Delimiter>
    <IsTruncated>false</IsTruncated>
    <Upload>
        <Key>Object</Key>
        <UploadId>1484726657932bcb5b17f7a98a8cad9fc36a340ff204c79bd2f51e7dddf0b6d1d
        <Initiator>
            <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
        </Initiator>
        <Owner>
            <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
        </Owner>
        <StorageClass>Standard</StorageClass>
        <Initiated>Wed Jan 18 16:04:17 2017</Initiated>
    </Upload>
    <Upload>
        <Key>Object</Key>
        <UploadId>1484727158f2b8034e5407d18cbf28e84f754b791ecab607d25a2e52de9fee641
        <Initiator>
            <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
        </Initiator>
        <Owner>
            <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
        </Owner>
        <StorageClass>Standard</StorageClass>
        <Initiated>Wed Jan 18 16:12:38 2017</Initiated>
    </Upload>
    <Upload>
        <Key>exampleobject</Key>
        <UploadId>1484727270323ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cce
        <Initiator>
            <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
        </Initiator>
        <Owner>
            <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
        </Owner>
        <StorageClass>Standard</StorageClass>
        <Initiated>Wed Jan 18 16:14:30 2017</Initiated>
    </Upload>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</ListMultipartUploadsResult>
```

版权所有：腾讯云计算（北京）有限责任公司

List Parts

最近更新时间：2023-12-21 15:07:04

功能描述

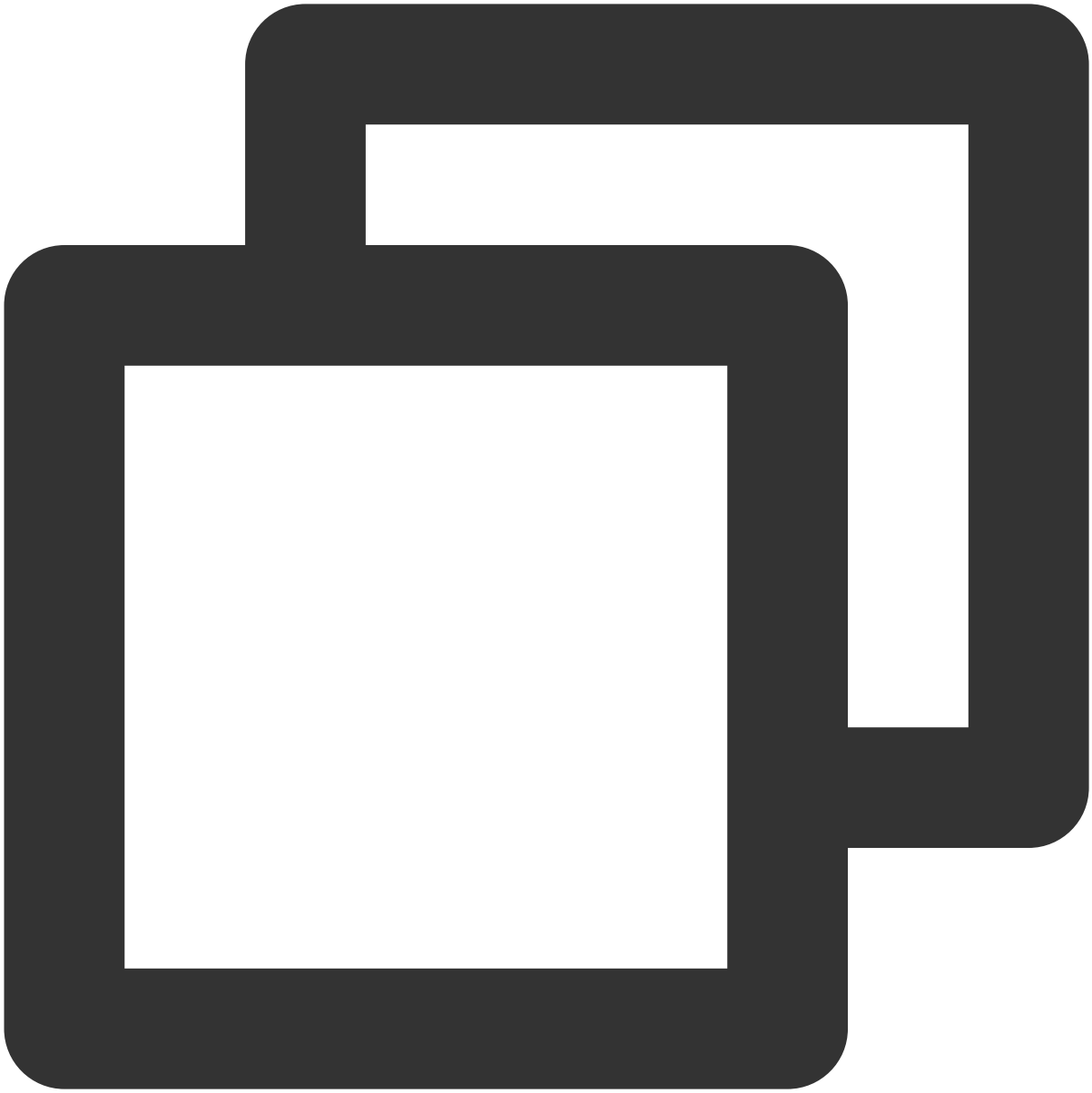
List Parts 用来查询特定分块上传中的已上传的块，即罗列出指定 UploadId 所属的所有已上传成功的分块。

说明

只有主账号或者被授权 List Part 接口权限的子账号可以调用该接口。

请求

请求示例



```
GET /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参见 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参见 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

名称	描述	类型	是否必选
UploadId	标识本次分块上传的 ID，使用 Initiate Multipart Upload 接口初始化分块上传时得到的 UploadId	string	是
encoding-type	规定返回值的编码方式	string	否
max-parts	单次返回最大的条目数量，最大值为1000，默认值为1000	int	否
part-number-marker	默认以 UTF-8 二进制顺序列出条目，所有列出条目从 marker 开始	string	否

请求体

该请求的请求体为空。

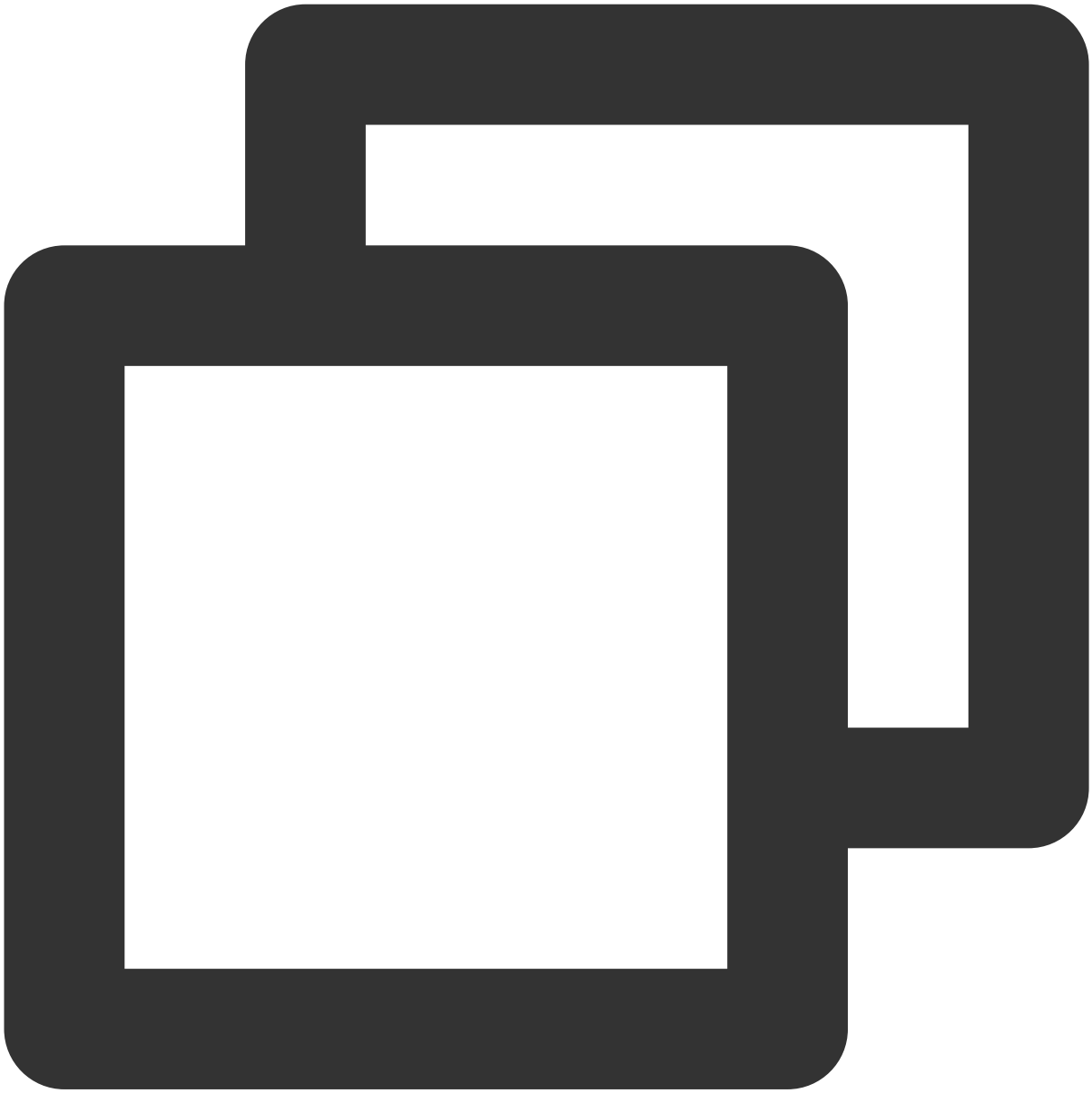
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

查询成功，返回 **application/xml** 数据，包含已完成的分片信息。



```
<?xml version="1.0" encoding="UTF-8" ?>
<ListPartsResult>
    <Bucket>examplebucket-1250000000</Bucket>
    <Encoding-type/>
    <Key>exampleobject</Key>
    <UploadId>14846420620b1f381e5d7b057692e131dd8d72dfa28f2633cfbbe4d0a9e8bd0719933
    <Initiator>
        <ID>1250000000</ID>
        <DisplayName>1250000000</DisplayName>
    </Initiator>
    <Owner>
```

```
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>100000000001</DisplayName>
    </Owner>
    <PartNumberMarker>0</PartNumberMarker>
    <Part>
        <PartNumber>1</PartNumber>
        <LastModified>Tue Jan 17 16:43:37 2017</LastModified>
        <ETag>"a1f8e5e4d63ac6970a0062a6277e191fe09a1382"</ETag>
        <Size>5242880</Size>
    </Part>
    <NextPartNumberMarker>1</NextPartNumberMarker>
    <StorageClass>STANDARD</StorageClass>
    <MaxParts>1</MaxParts>
    <IsTruncated>true</IsTruncated>
</ListPartsResult>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型
ListPartsResult	无	保存 List Parts 请求结果的所有信息	Container

Container 节点 ListPartsResult 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	ListPartsResult	分块上传的目标 Bucket，存储桶的名字，由用户自定义字符串和系统生成 APPID 数字串由中划线连接而成，如：examplebucket-1250000000	string
Encoding-Type	ListPartsResult	编码格式	string
Key	ListPartsResult	Object 的名称	string
UploadId	ListPartsResult	标识本次分块上传的 ID	string
Initiator	ListPartsResult	用来表示这些分块所有者的信息	Container
Owner	ListPartsResult	用来表示这些分块所有者的信息	Container
StorageClass	ListPartsResult	用来表示这些分块的存储类型，枚举值：STANDARD，STANDARD_IA，ARCHIVE，DEEP_ARCHIVE 等，更多存储类型请参见 存储类型概述	string
PartNumberMarker	ListPartsResult	默认以 UTF-8 二进制顺序列出条目，所有列出条目从 marker 开始	string

NextPartNumberMarker	ListPartsResult	假如返回条目被截断，则返回 NextMarker 就是下一个条目的起点	string
MaxParts	ListPartsResult	单次返回最大的条目数量	int
IsTruncated	ListPartsResult	响应请求条目是否被截断，布尔值：true，false	boolean
Part	ListPartsResult	元数据信息	Container

Container 节点 Initiator 的内容：

节点名称（关键字）	父节点	描述	类型
ID	ListPartsResult.Initiator	创建者的一个唯一标识	string
DisplayName	ListPartsResult.Initiator	创建者的用户名描述	string

Container 节点 Owner 的内容：

节点名称（关键字）	父节点	描述	类型
ID	ListPartsResult.Owner	创建者的一个唯一标识	string
DisplayName	ListPartsResult.Owner	创建者的用户名描述	string

Container 节点 Part 的内容：

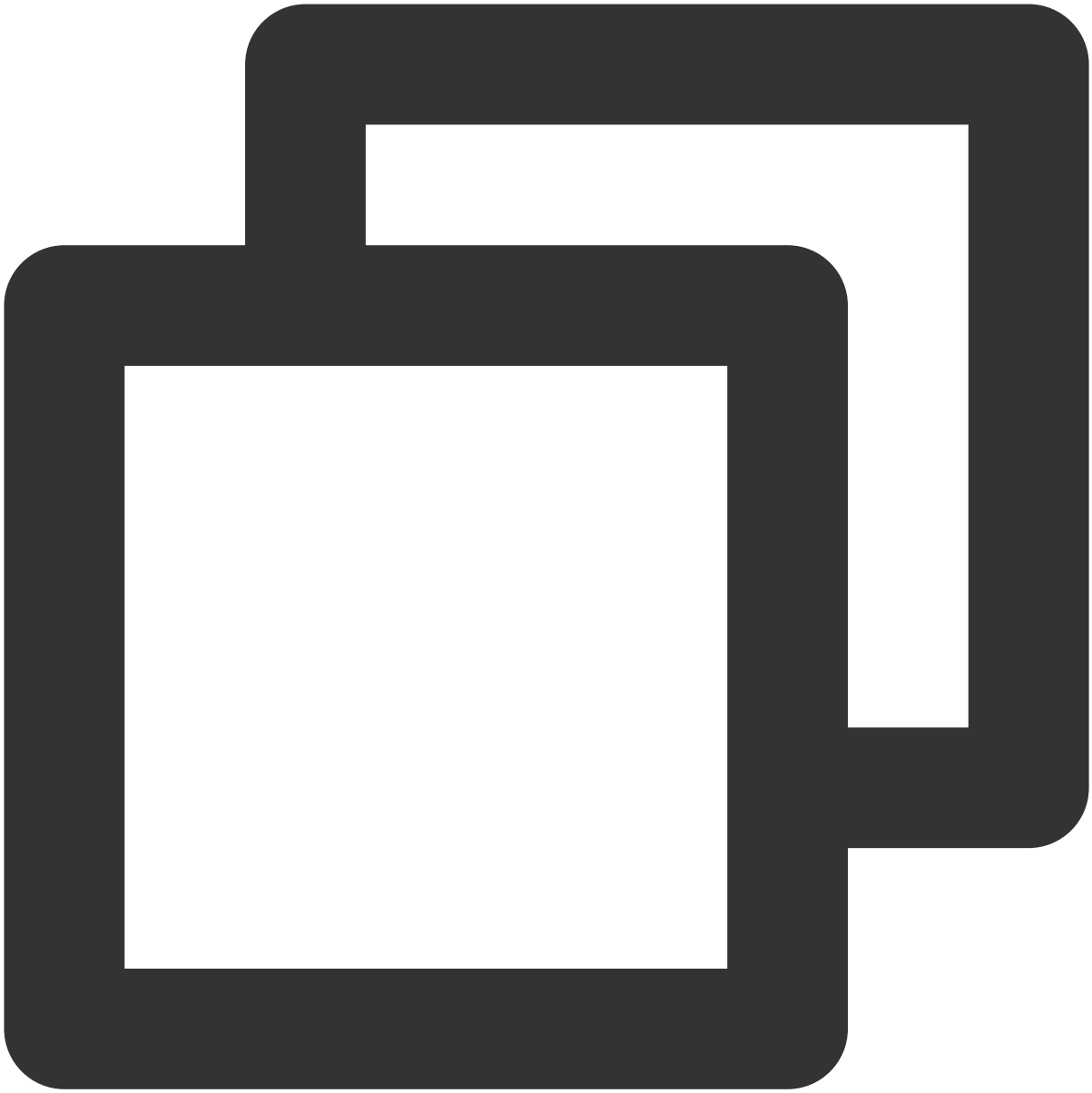
节点名称（关键字）	父节点	描述	类型
PartNumber	ListPartsResult.Part	块的编号	string
LastModified	ListPartsResult.Part	说明块最后被修改时间	string
ETag	ListPartsResult.Part	块的 MD-5 算法校验值	string
Size	ListPartsResult.Part	说明块大小，单位是 Byte	string

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

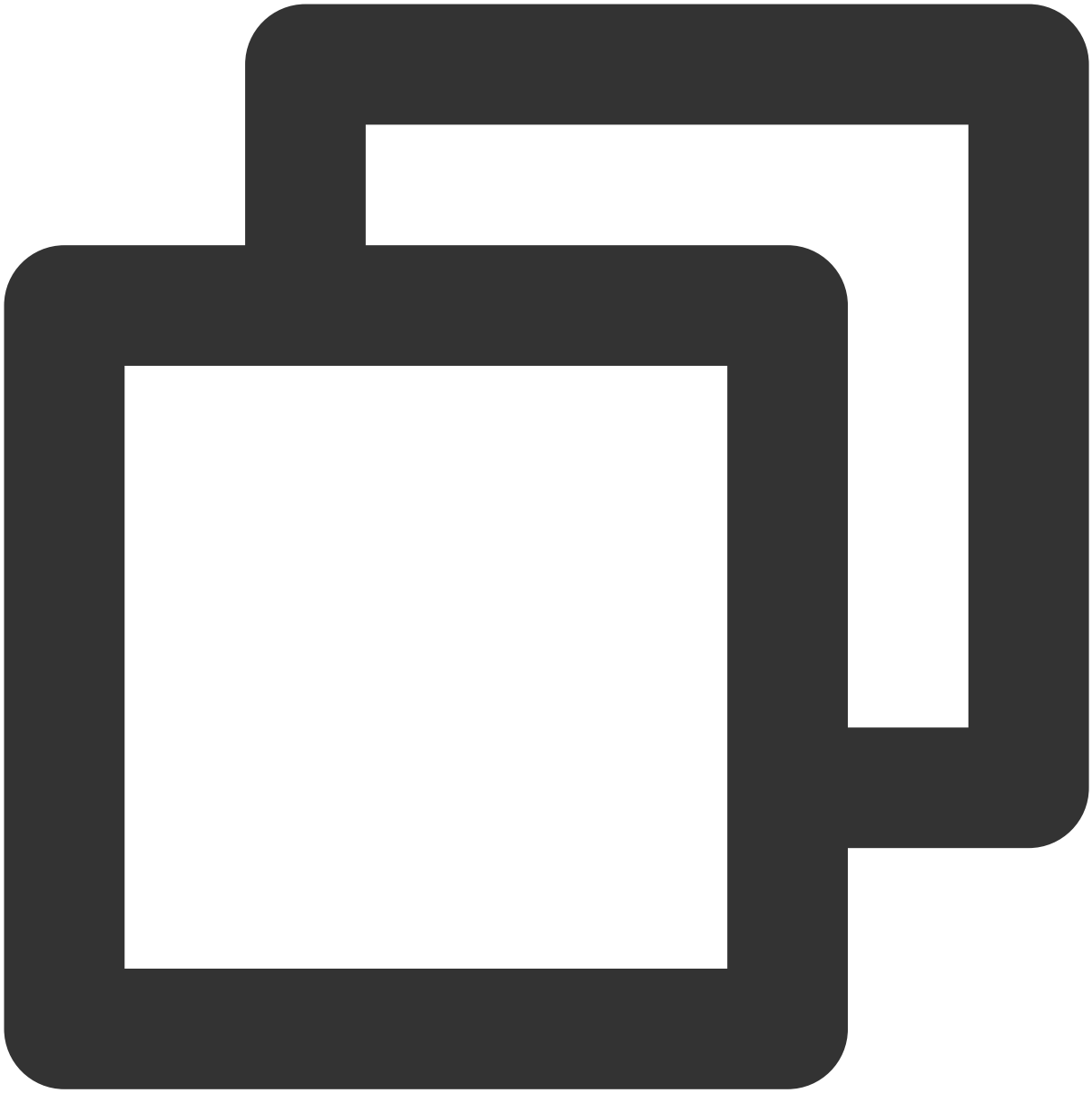
实际案例

请求



```
GET /exampleobject?uploadId=1585130821cbb7df1d11846c073ad648e8f33b087cec2381df437ac
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 25 Mar 2020 10:07:25 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1119
Connection: close
Date: Wed, 25 Mar 2020 10:07:25 GMT
Server: tencent-cos
x-cos-request-id: NWU3YjJkNWRfMjNhZjJhMDlfNWY5Ml8zMmUy****



<ListPartsResult>
```

```
            <Bucket>examplebucket-1250000000</Bucket>
            <EncodingType/>
            <Key>exampleobject</Key>
            <UploadId>1585130821cbb7df1d11846c073ad648e8f33b087cec2381df437acdc833c
            <Owner>
                <ID>1250000000</ID>
                <DisplayName>1250000000</DisplayName>
            </Owner>
            <PartNumberMarker>0</PartNumberMarker>
            <Initiator>
                <ID>qcs::cam::uin/100000000001:uin/100000000011</ID>
                <DisplayName>100000000011</DisplayName>
            </Initiator>
            <Part>
                <PartNumber>1</PartNumber>
                <LastModified>2020-03-25T10:07:14.000Z</LastModified>
                <ETag>&amp;quot;39270a968a357d24207e9911162507eb&amp;quot;</ETag>
                <Size>1048576</Size>
            </Part>
            <Part>
                <PartNumber>2</PartNumber>
                <LastModified>2020-03-25T10:07:13.000Z</LastModified>
                <ETag>&amp;quot;d899fbd1e06109ea2e4550f5751c88d6&amp;quot;</ETag>
                <Size>1048576</Size>
            </Part>
            <Part>
                <PartNumber>3</PartNumber>
                <LastModified>2020-03-25T10:07:13.000Z</LastModified>
                <ETag>&amp;quot;762890d6c9a871b7bd136037cb2260cd&amp;quot;</ETag>
                <Size>1048576</Size>
            </Part>
            <StorageClass>Standard</StorageClass>
            <MaxParts>1000</MaxParts>
            <IsTruncated>false</IsTruncated>
</ListPartsResult>
```

对象标签

PUT Object tagging

最近更新时间：2023-12-21 15:08:53

功能描述

COS 支持为已存在的对象设置标签。PUT Object tagging 接口通过为对象添加键值对作为对象标签，可以协助您分组管理已有的对象资源，详情请参见 对象标签概述。
如您使用子账号调用此项接口，请确保您已经在主账号处获取了 `PUT Object tagging` 这个接口的权限。

注意

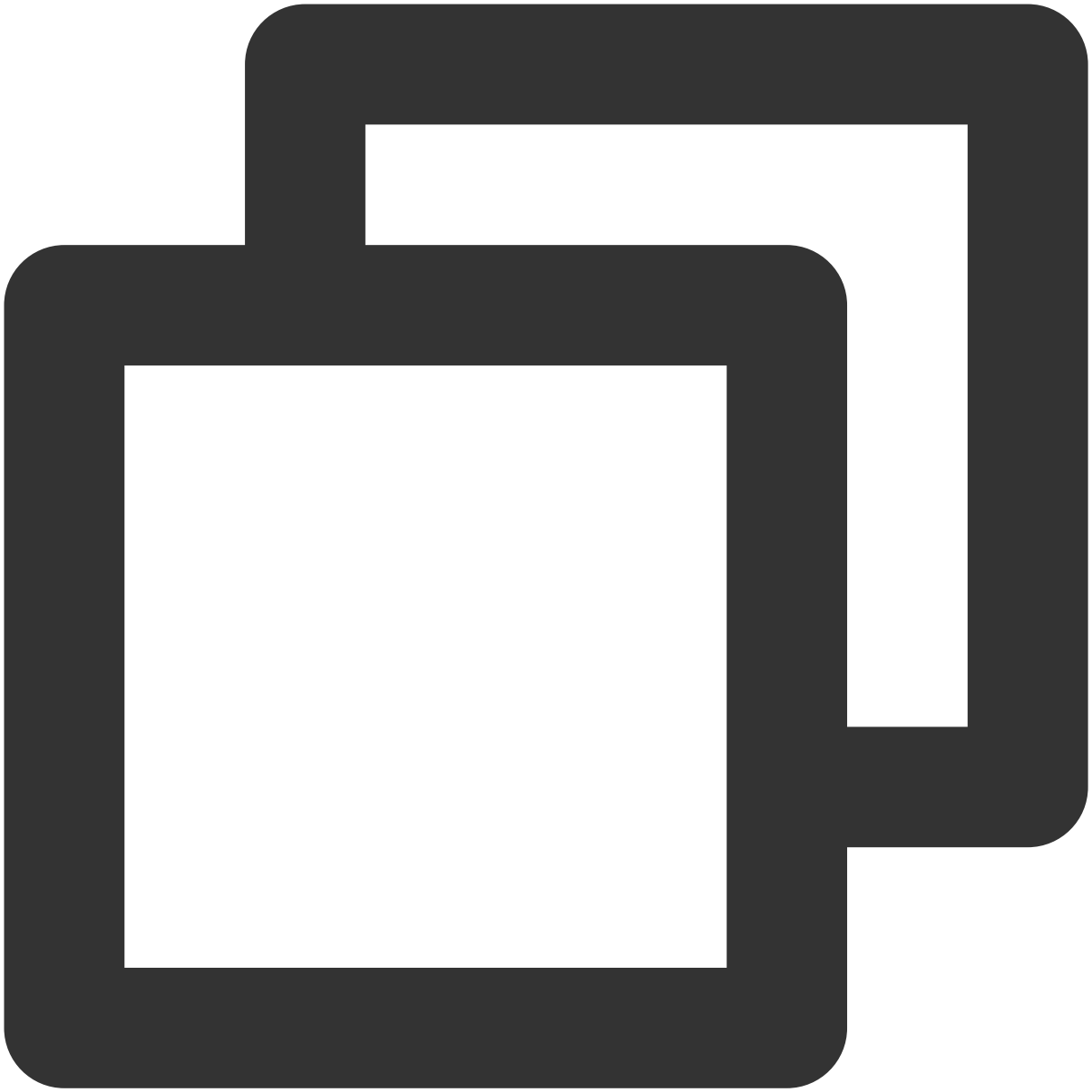
目前对象标签功能最多支持一个对象下设置10个不同的标签，如果超出设置上限，COS 将覆盖已有的对象标签为新的对象标签。

版本控制

如果您的存储桶开启了版本控制，并且需要对指定版本的对象添加标签，可以在发起请求时携带 VersionId 参数，对象标签将添加到指定的对象版本中。

请求

请求示例



```
PUT /<ObjectKey>?tagging&VersionId=VersionId HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

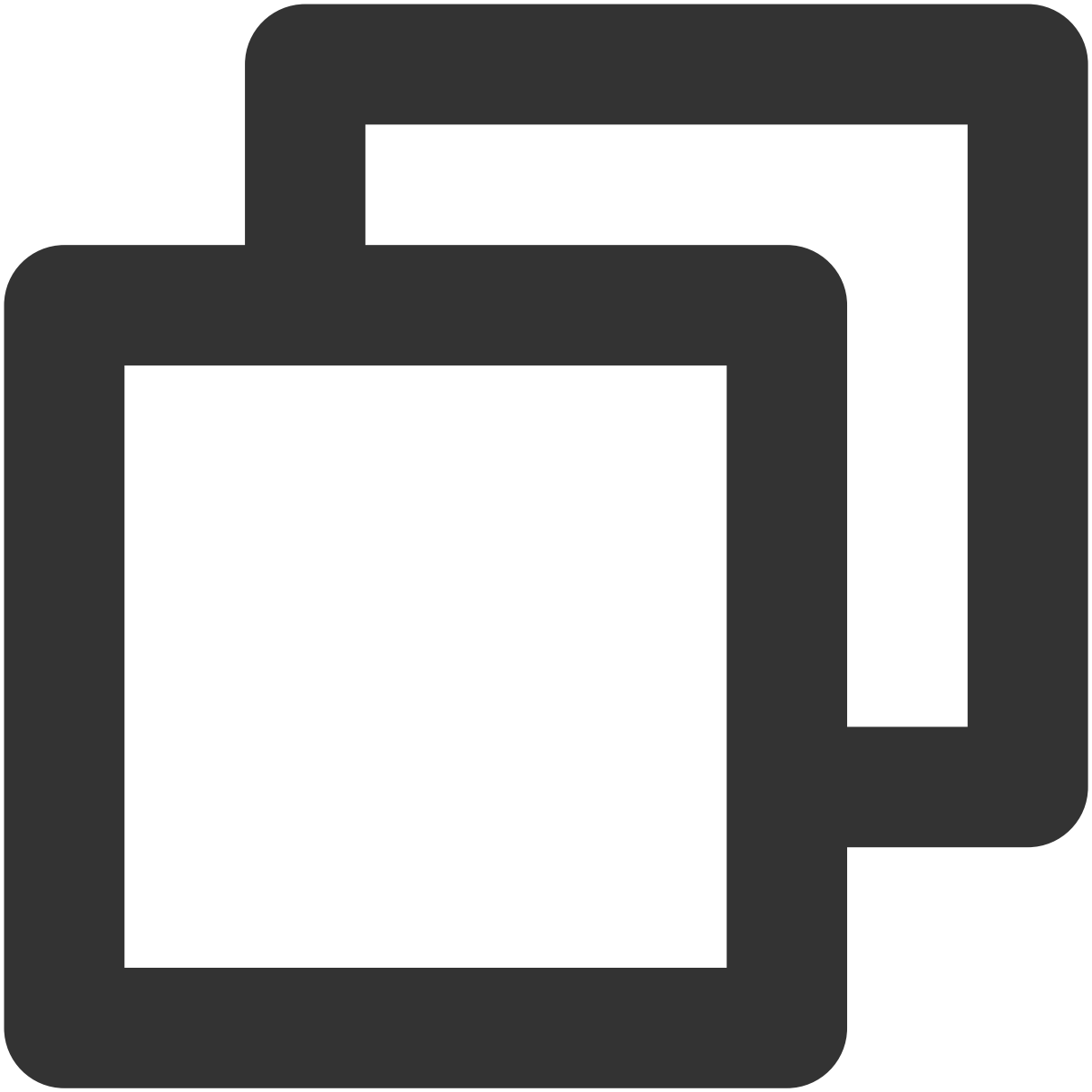
名称	描述	类型	是否必选
versionId	当启用版本控制时，指定要操作的对象版本 ID，如不指定则将添加标签到最新版本的对象	string	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求需要设置如下标签集合：



```
<?xml version="1.0" encoding="UTF-8" ?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </TagSet>
</Tagging>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tagging	无	标签集合	Container	是
TagSet	Tagging	标签集合	Container	是
Tag	Tagging.TagSet	标签集合，最多支持10个标签	Containers	是
Key	Tagging.TagSet.Tag	标签键，长度小于128字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String	是
Value	Tagging.TagSet.Tag	标签值，长度小于256字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String	是

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求响应体为空。

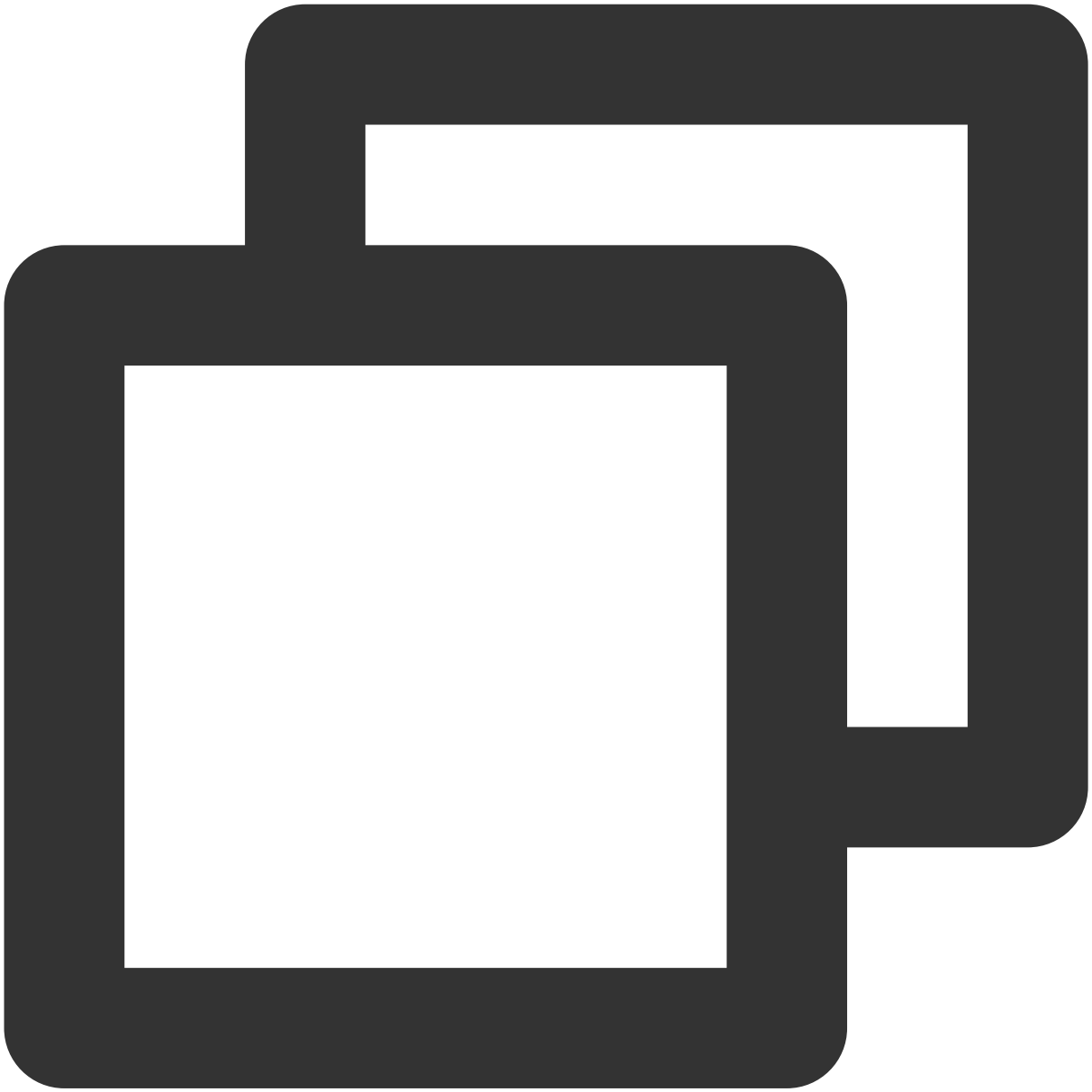
错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

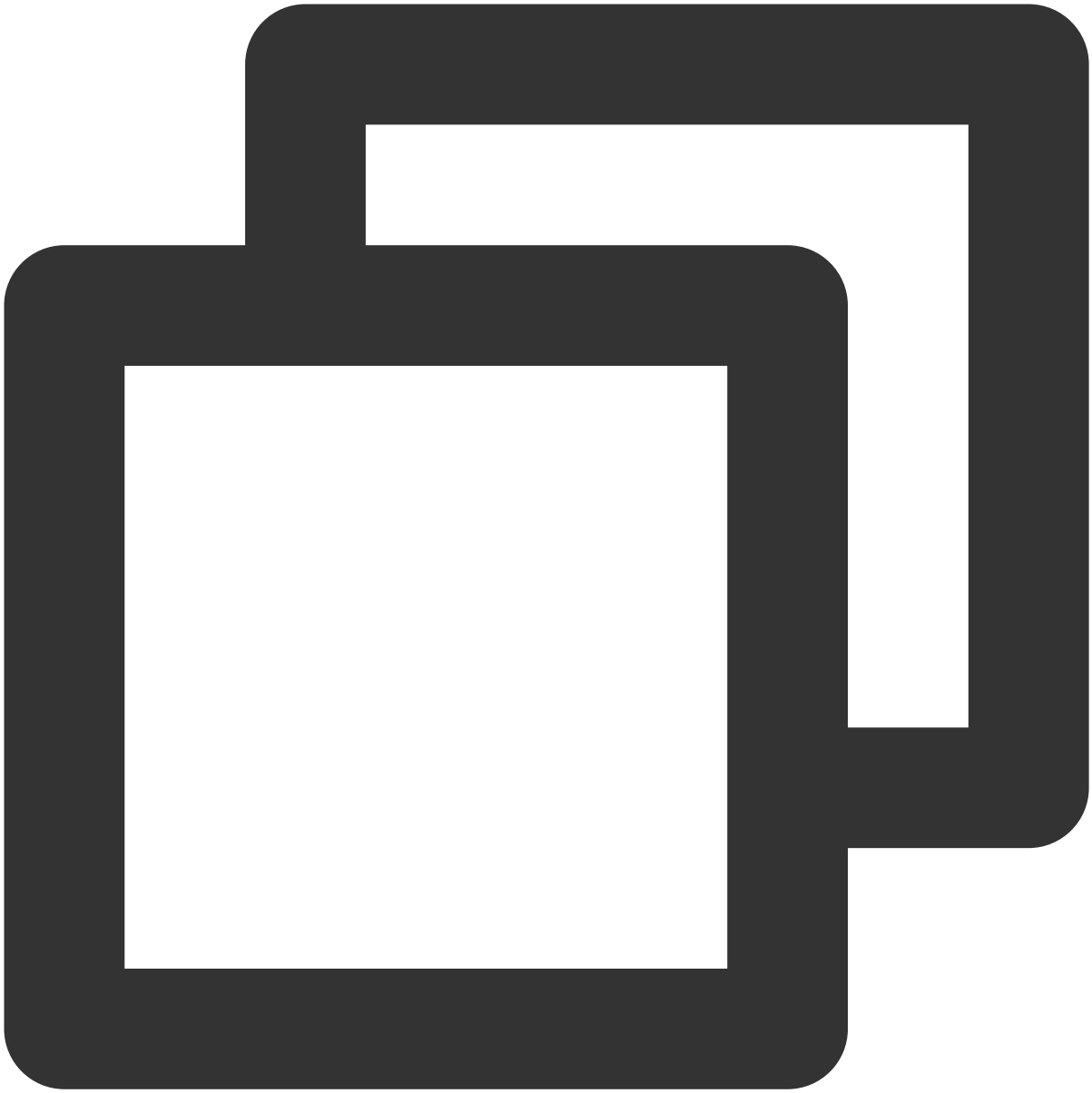
如下请求向存储桶 `examplebucket-1250000000` 中的对象 `exampleobject.txt` 写入了 `{age:18}` 和 `{name:xiaoming}` 两个标签。COS 配置标签成功并返回200成功请求。



```
PUT /exampleobject.txt?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Authorization: Auth String
Content-Length: 127
Content-MD5:MD5 String
Content-Type: application/xml

<Tagging>
    <TagSet>
```

```
        <Tag>
            <Key>age</Key>
            <Value>18</Value>
        </Tag>
        <Tag>
            <Key>name</Key>
            <Value>xiaoming</Value>
        </Tag>
    </TagSet>
</Tagging>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 19 Jan 2020 11:40:22 GMT
Server: tencent-cos
x-cos-request-id: NWE2MWQ5MjZfMTBhYzM1MGFfMTA5ODVfMTVj****
```

GET Object tagging

最近更新时间：2023-12-21 15:08:58

功能描述

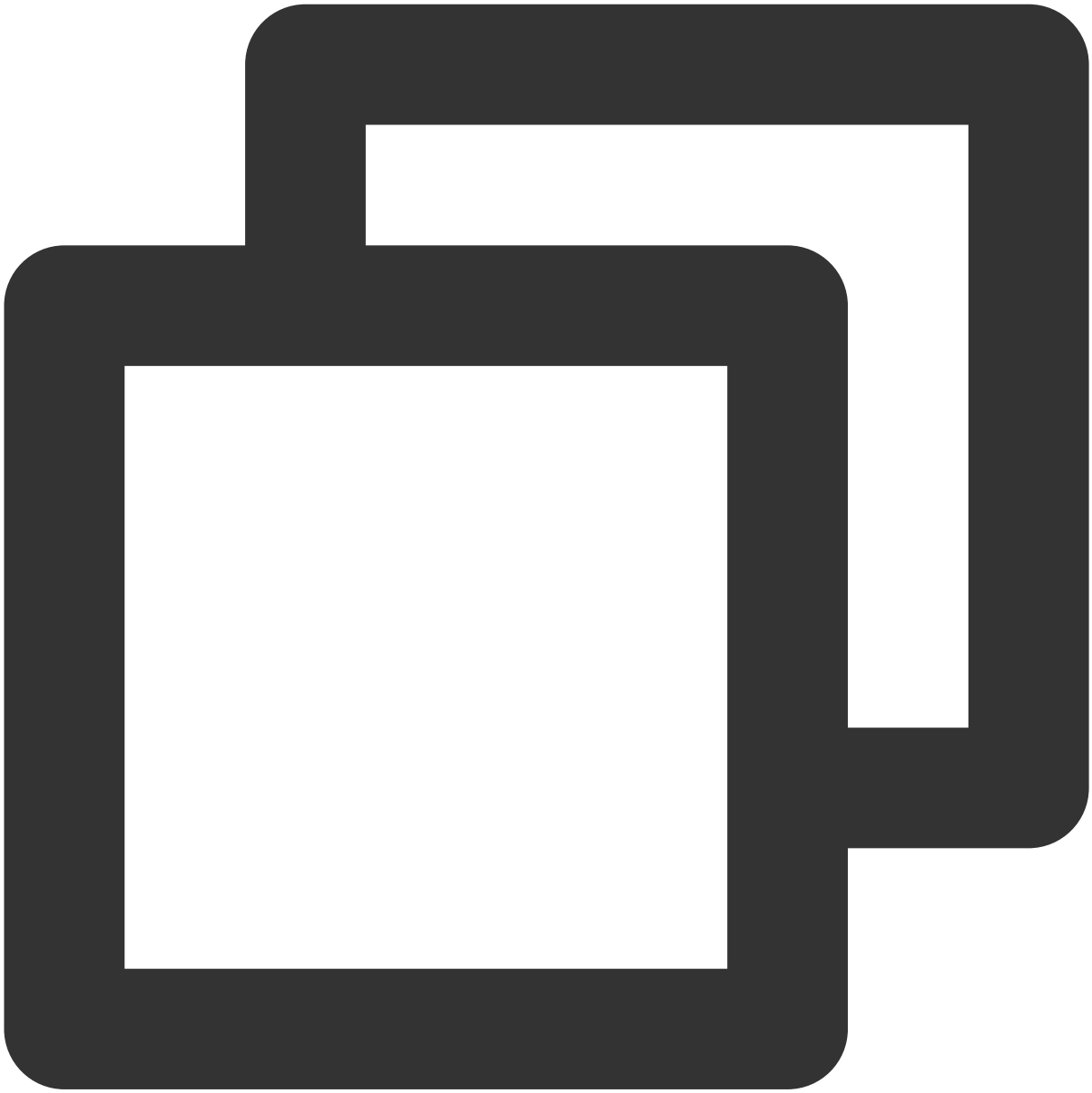
GET Object tagging 接口用于查询指定对象下已有的对象标签。
如您使用子账号调用此项接口，请确保您已经在主账号处获取了 `GET Object tagging` 这个接口的权限。

版本控制

如果您的存储桶开启了版本控制，并且需要查询指定版本的对象的标签，可以在发起请求时携带 VersionId 参数，此时将查询指定版本对象的标签信息。

请求

请求示例



```
GET /<ObjectKey>?tagging&VersionId=VersionId HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

名称	描述	类型	是否必选
versionId	当启用版本控制时，指定要操作的对象版本 ID，如不指定则查询最新版本对象的对象标签	string	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求返回的响应体节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tagging	无	标签集合	Container	是
TagSet	Tagging	标签集合	Container	是
Tag	Tagging.TagSet	标签集合，最多支持 10 个标签	Containers	是
Key	Tagging.TagSet.Tag	标签键，长度不超过 128 字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String	是
Value	Tagging.TagSet.Tag	标签值，长度不超过 256 字节，支持英文字母、数字、空格、加号、减号、下划线、等号、点号、冒号、斜线	String	是

错误码

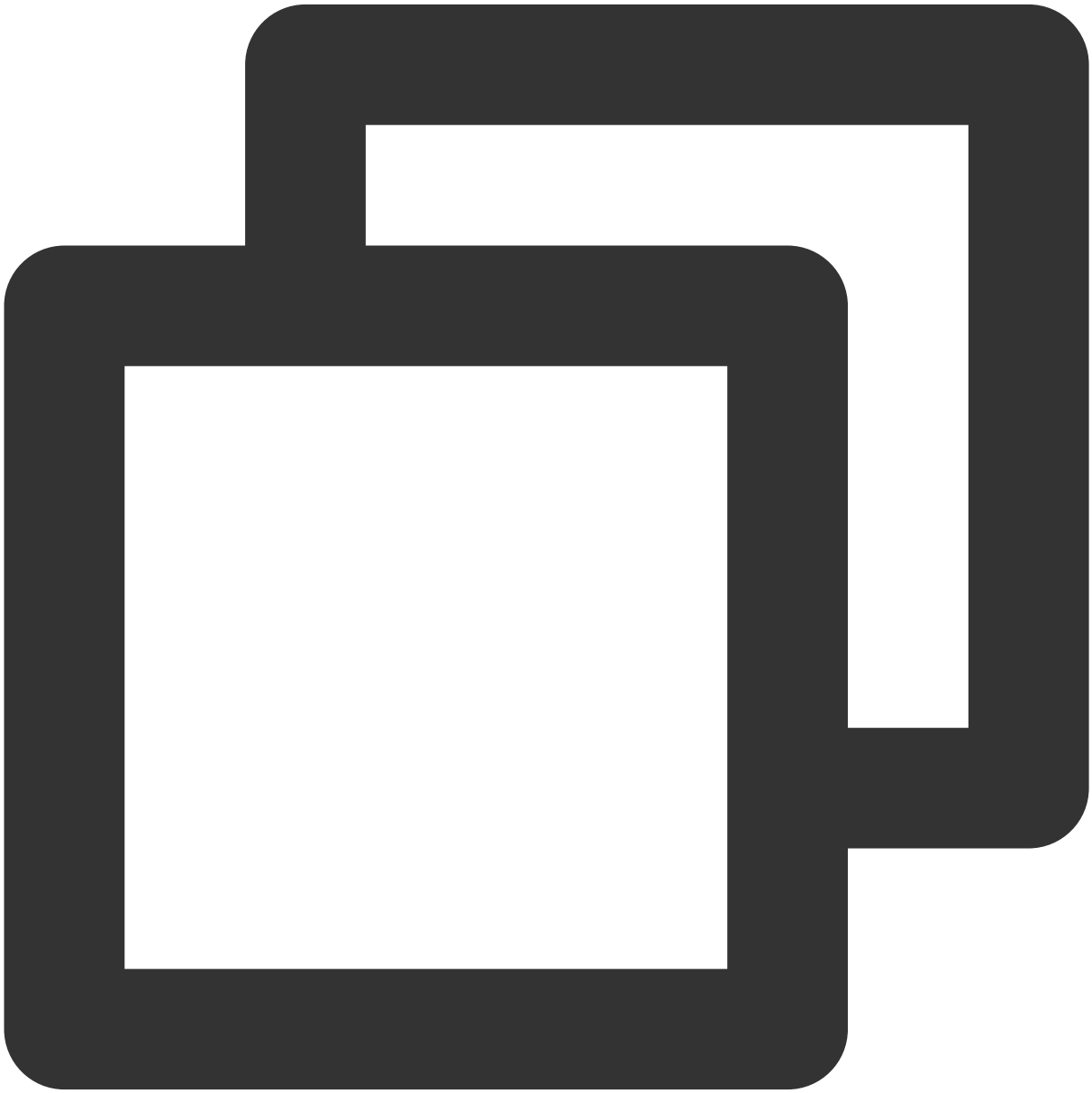
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

版权所有：腾讯云计算（北京）有限责任公司

实际案例

请求

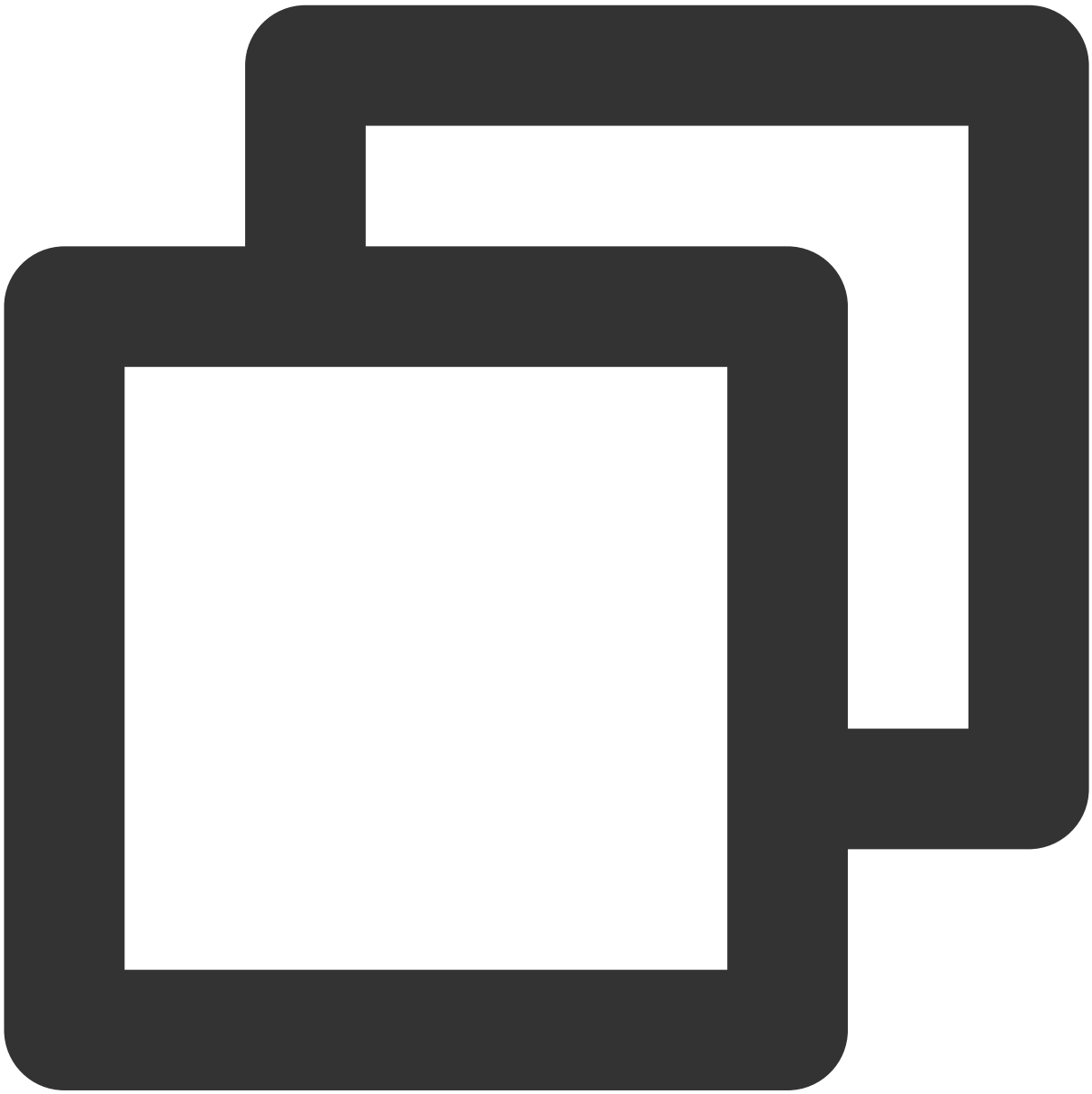
如下请求申请查询存储桶 `examplebucket-1250000000` 中的对象 `exampleobject.txt` 下的标签信息，COS 解析该请求后并返回该存储桶下已有的 `{age:18}` 和 `{name:xiaoming}` 两个标签。



```
GET /exampleobject.txt?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

```
Authorization: Auth String
Content-Md5: MD5 String
Content-Length: 127
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2020 11:40:22 GMT
```

```
Server: tencent-cos
<Tagging>
    <TagSet>
        <Tag>
            <Key>age</Key>
            <Value>18</Value>
        </Tag>
        <Tag>
            <Key>name</Key>
            <Value>xiaoming</Value>
        </Tag>
    </TagSet>
</Tagging>
```

DELETE Object tagging

最近更新时间：2023-12-21 15:09:12

功能描述

DELETE Object tagging 接口用于删除指定对象下已有的对象标签。

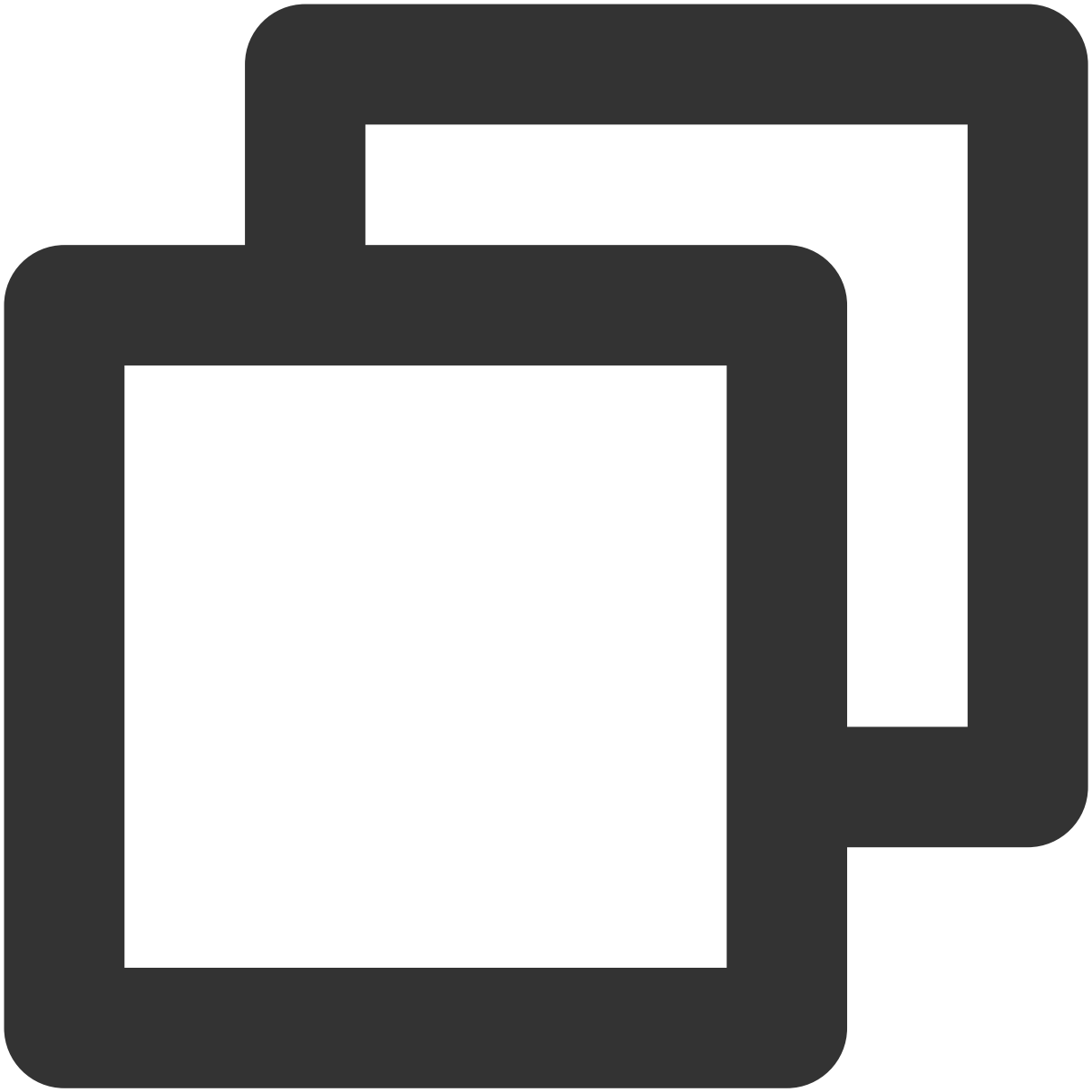
如您使用子账号调用此项接口，请确保您已经在主账号处获取了 `DELETE Object tagging` 这个接口的权限。

版本控制

如果您的存储桶开启了版本控制，并且需要删除指定版本对象的标签，可以在发起请求时携带 VersionId 参数，此时将删除指定版本 ID 对象的对象标签集。

请求

请求示例



```
DELETE /<ObjecKey>?tagging&VersionId=VersionId HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String
```

说明

Host: <BucketName-APPID>.cos.<Region>.myqcloud.com，其中 <BucketName-APPID> 为带 APPID 后缀的存储桶名字，例如 examplebucket-1250000000，可参阅 存储桶概览 > 基本信息 和 存储桶概述 > 存储桶命名规范 文档；<Region> 为 COS 的可用地域，可参阅 地域和访问域名 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

版权所有：腾讯云计算（北京）有限责任公司

请求参数

名称	描述	类型	是否必选
versionId	当启用版本控制时，指定要操作的对象版本 ID，如不指定则删除最新版本对象的标签	string	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该请求无特殊响应体信息。

错误码

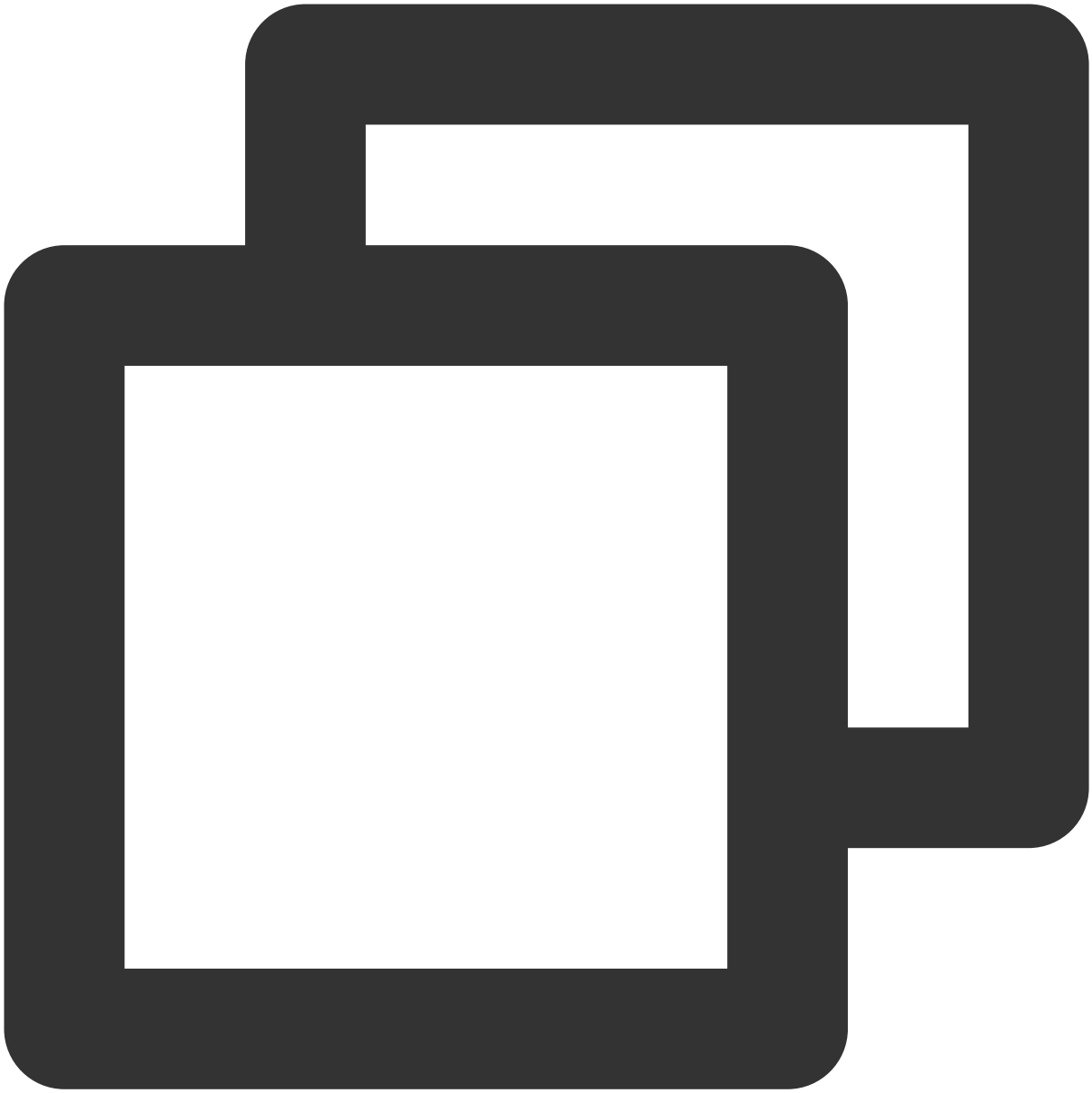
此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

案例一：删除对象标签（未开启版本控制）

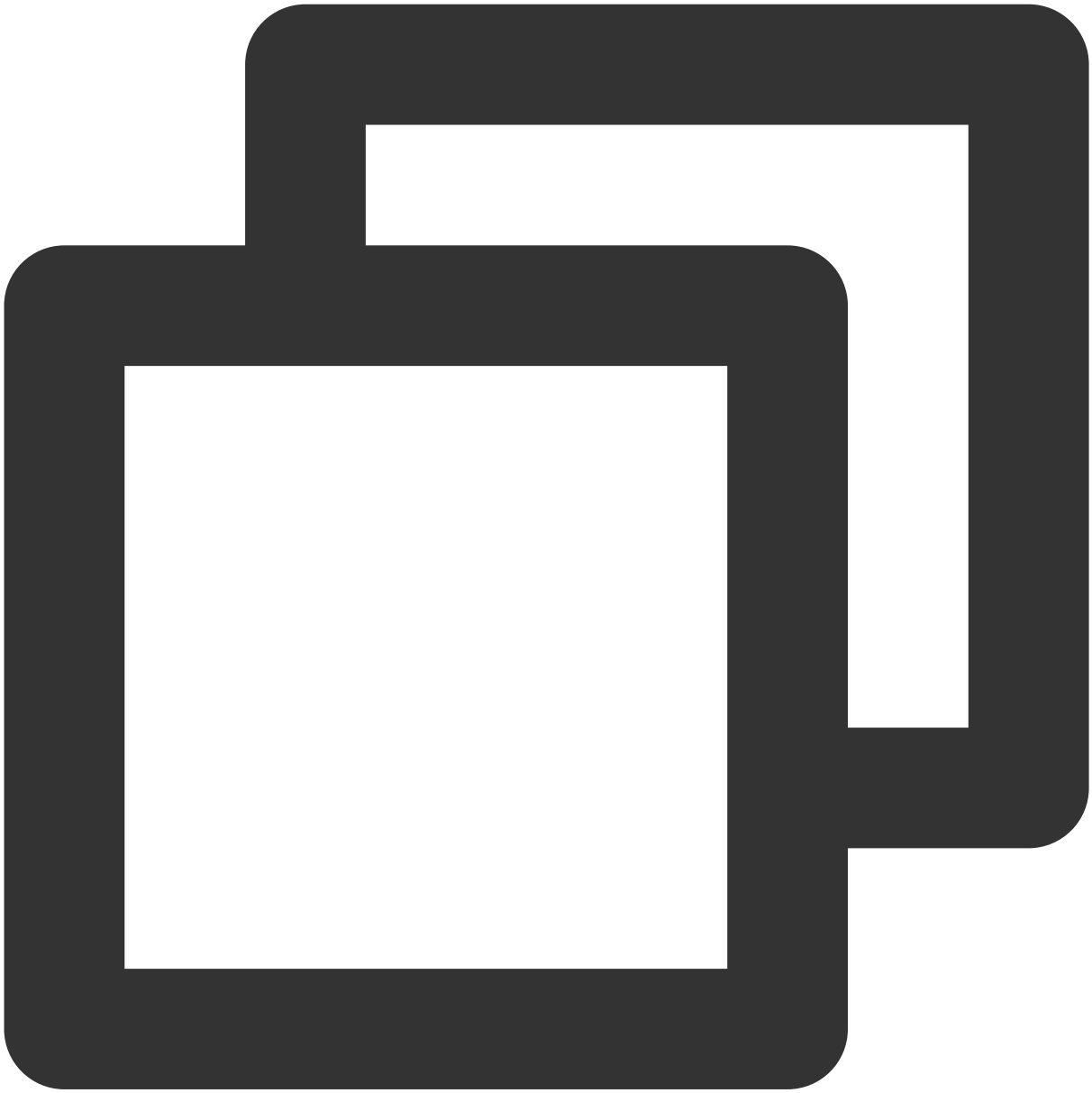
请求

如下请求申请删除存储桶 `examplebucket-1250000000` 中的对象 `exampleobject.txt` 下已有的标签信息。COS 解析该请求后删除该对象中的所有标签。



```
DELETE /exampleobject.txt?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Authorization: Auth String
Content-Length: 0
Content-Type: application/xml
```

响应

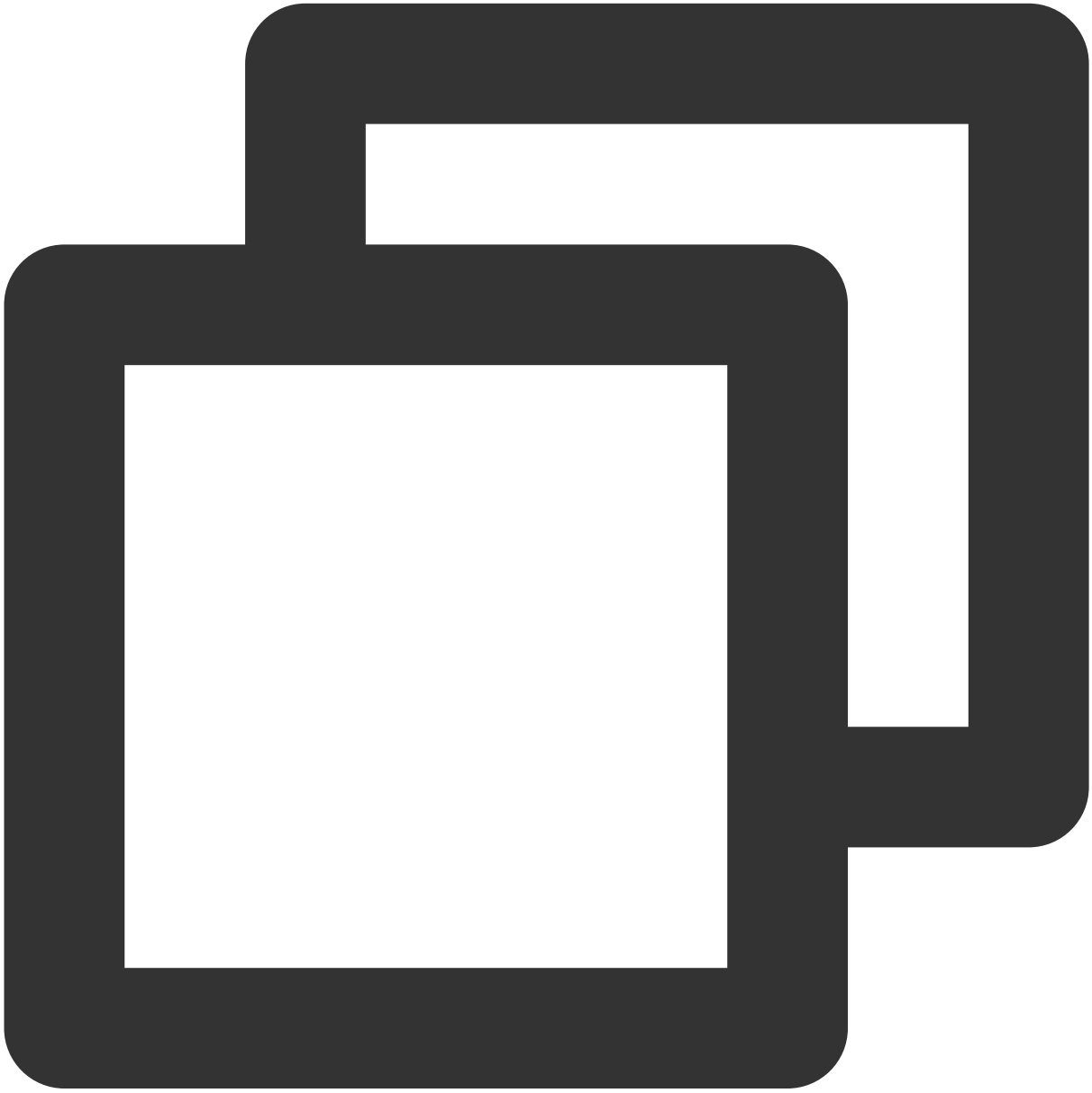


```
HTTP/1.1 204 No Content
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2020 11:40:22 GMT
```

案例二：删除对象标签（开启版本控制）

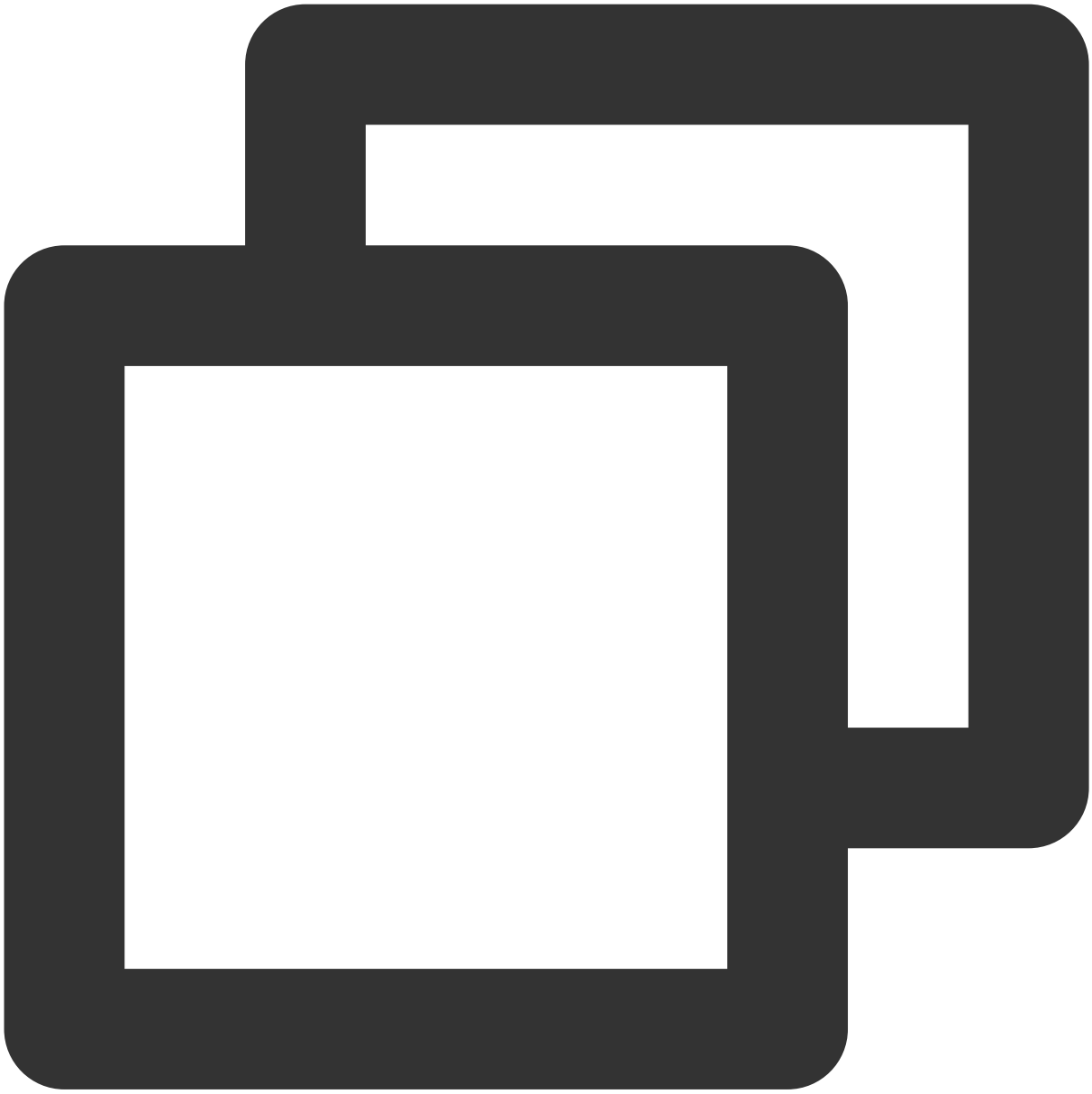
请求

如下请求申请删除存储桶 `examplebucket-1250000000` 中的对象 `exampleobject.txt` 下已有的标签信息，删除时指定对象版本号为 `MTg0NDUxNTgwODg2MTEzMTQyOTI` 。COS解析该请求后删除该版本的对象中的所有标签。



```
DELETE /exampleobject.txt?tagging&VersionId=MTg0NDUxNTgwODg2MTEzMTQyOTI  HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Authorization: Auth String
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 204 No Content
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2020 11:45:22 GMT
```

批量处理接口

CreateJob

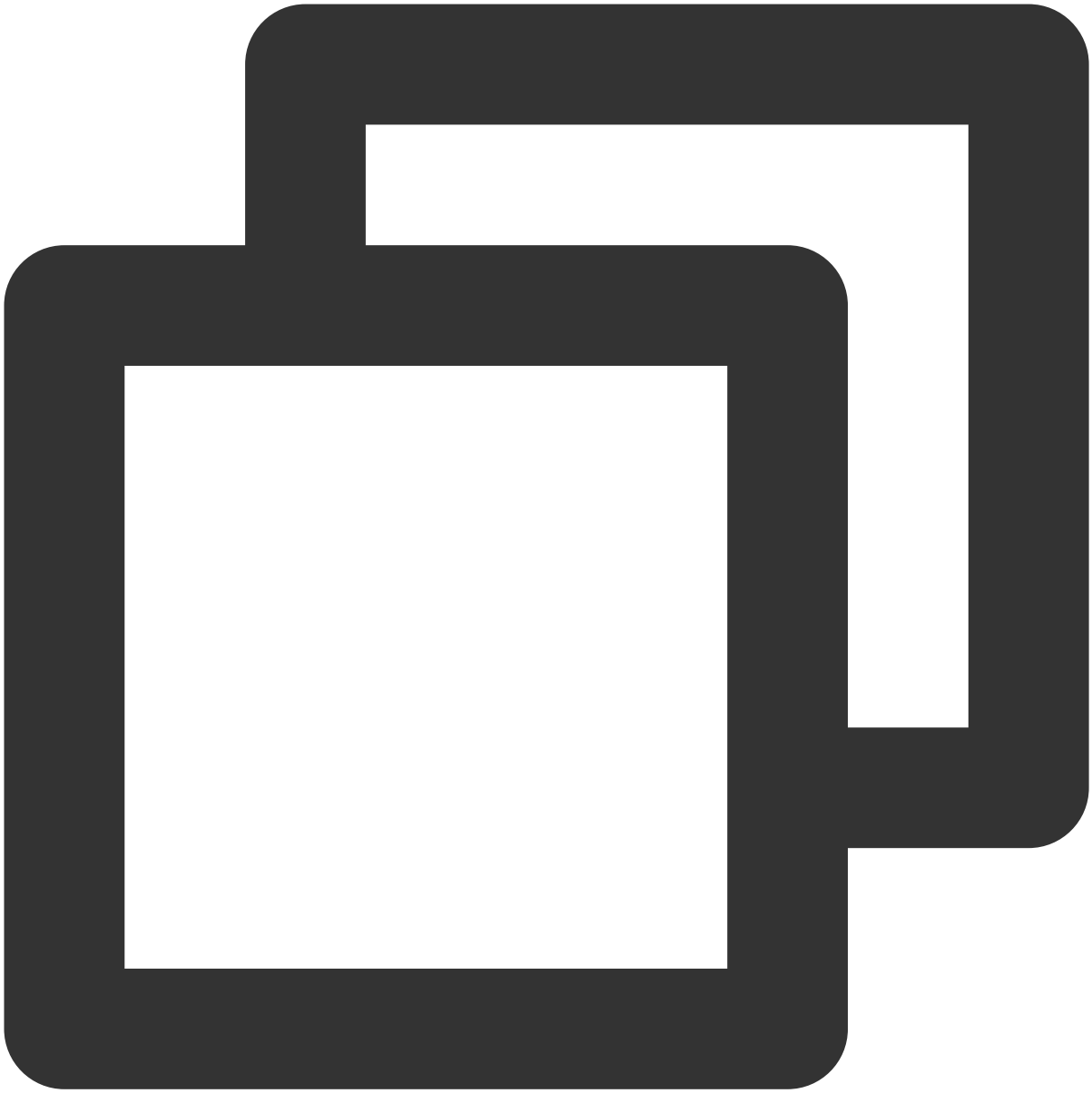
最近更新时间：2023-12-21 17:43:15

功能描述

CreateJob 用于在存储桶中创建批量处理任务。详细信息可参见 批量处理概述。
调用该请求时，请确保您有足够的权限对存储桶的对象进行操作。存储桶所有者默认拥有该权限，若您无该项权限，请先向存储桶所有者申请该项操作的权限。

请求

请求示例



```
POST /jobs HTTP/1.1
Host: <UIN>.cos-control.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Authorization: Auth String
x-cos-appid: <APPID>

<?xml version="1.0" encoding="UTF-8"?>
<CreateJobRequest>
...
```

```
</CreateJobRequest>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

调用 CreateJob 所需的参数。该参数格式如下：

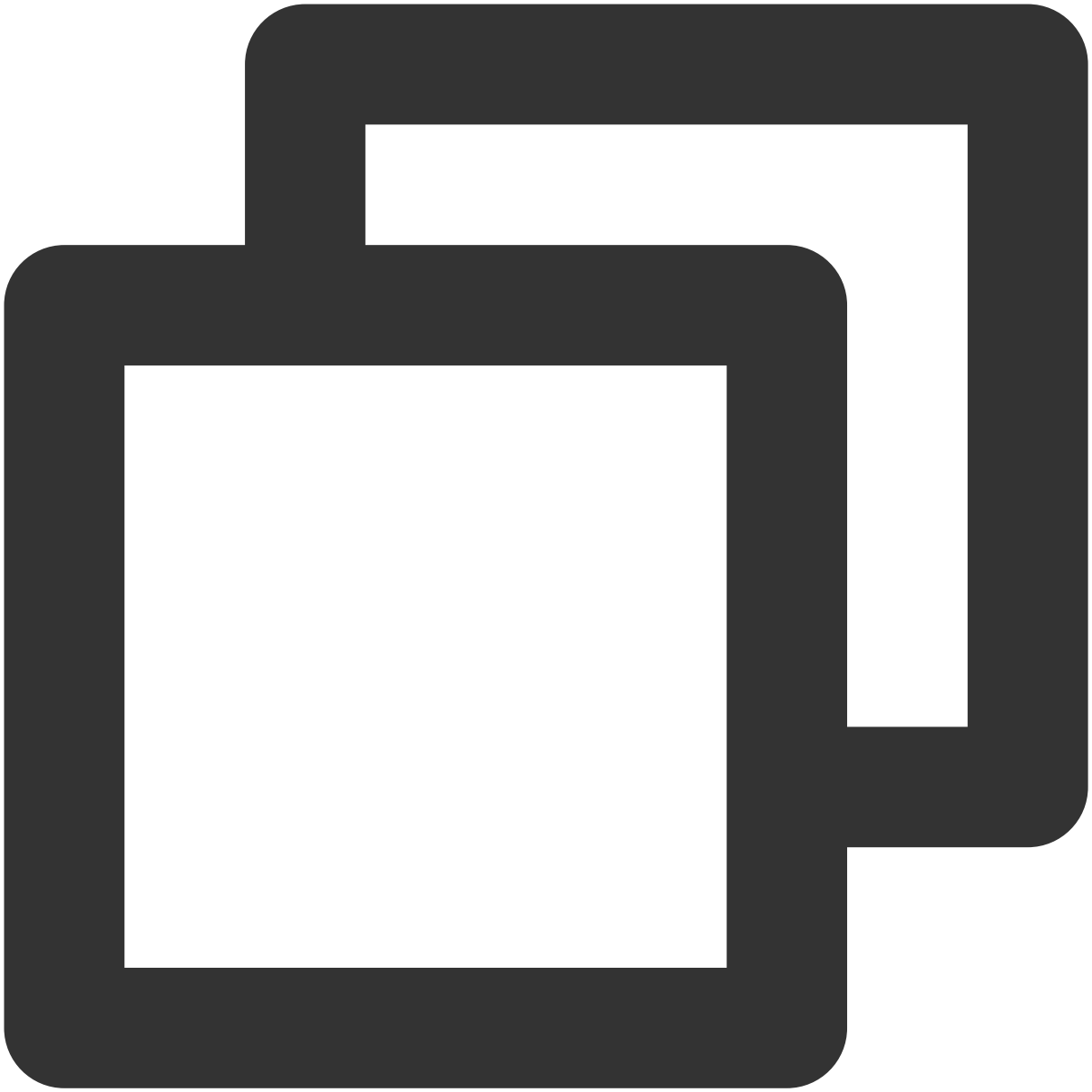
参数	描述	是否必选
x-cos-appid	用户的 APPID，长度为1 - 64字节。	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

用户在请求体中使用 XML 语言设置清单任务的具体配置信息。配置信息包括清单任务分析的对象，分析的频次，分析的维度，分析结果的格式及存储的位置等信息。



```
<CreateJobRequest>
    <ClientRequestToken>string</ClientRequestToken>
    <ConfirmationRequired>boolean</ConfirmationRequired>
    <Description>string</Description>
    <Manifest>
        <Location>
            <ETag>string</ETag>
            <ObjectArn>string</ObjectArn>
            <ObjectVersionId>string</ObjectVersionId>
        </Location>
        <Spec>
```

```
            <Fields>
                <member>string</member>
                <member>string</member>
            </Fields>
            <Format>string</Format>
        </Spec>
    </Manifest>
    <Operation>
        <COSInitiateRestoreObject>
            <ExpirationInDays> integer </ExpirationInDays>
            <JobTier> string </JobTier>
        </COSInitiateRestoreObject>
        <COSPutObjectCopy>
            <AccessControlDirective>string</AccessControlDirective>
            <AccessControlGrants>
                <COSGrant>
                    <Grantee>
                        <Identifier>string</Identifier>
                        <TypeIdentifier>string</TypeIdentifier>
                    </Grantee>
                    <Permission>string</Permission>
                </COSGrant>
                <COSGrant>
                    <Grantee>
                        <Identifier>string</Identifier>
                        <TypeIdentifier>string</TypeIdentifier>
                    </Grantee>
                    <Permission>string</Permission>
                </COSGrant>
            </AccessControlGrants>
            <PrefixReplace>boolean</PrefixReplace>
            <ResourcesPrefix>string</ResourcesPrefix>
            <TargetKeyPrefix>string</TargetKeyPrefix>
            <CannedAccessControlList>string</CannedAccessControlList>
            <MetadataDirective>string</MetadataDirective>
            <ModifiedSinceConstraint>timestamp</ModifiedSinceConstraint>
            <UnModifiedSinceConstraint>timestamp</UnModifiedSinceConstraint>
            <MetadataDirective>string</MetadataDirective>
            <NewObjectMetadata>
                <CacheControl>string</CacheControl>
                <ContentDisposition>string</ContentDisposition>
                <ContentEncoding>string</ContentEncoding>
                <ContentType>string</ContentType>
                <HttpExpiresDate>timestamp</HttpExpiresDate>
                <SSEAlgorithm>string</SSEAlgorithm>
                <UserMetadata>
                    <member>
```

```
                        <Key>string</Key>
                        <Value>string</Value>
                    </member>
                    <member>
                        <Key>string</Key>
                        <Value>string</Value>
                    </member>
                </UserMetadata>
            </NewObjectMetadata>
            <TaggingDirective>string</TaggingDirective>
            <NewObjectTagging>
                <COSTag>
                    <Key>string</Key>
                    <Value>string</Value>
                </COSTag>
            </NewObjectTagging>
            <StorageClass>string</StorageClass>
            <TargetResource>string</TargetResource>
        </COSPutObjectCopy>
    </Operation>
    <Priority>integer</Priority>
    <Report>
        <Bucket>string</Bucket>
        <Enabled>boolean</Enabled>
        <Format>string</Format>
        <Prefix>string</Prefix>
        <ReportScope>string</ReportScope>
    </Report>
    <RoleArn>string</RoleArn>
</CreateJobRequest>
```

请求体内相关元素的具体描述如下。其他元素请参见 批量处理功能公共元素。

节点名	父节点	描述	类型	是否必选
ClientRequestToken	CreateJobRequest	每个请求唯一的 token，用于避免前端重复发起同一批处理任务。长度为1 - 64字节，建议使用 uuid。	String	是
ConfirmationRequired	CreateJobRequest	是否在执行任务前进行确认。默认值为 false。	Boolean	否
Description	CreateJobRequest	任务描述。若您在创建任务时配置了此信息，则会返回该项内容。描述长度范围为0 - 256字节。	String	否
Manifest	CreateJobRequest	待处理的对象清单。您需要将待处理	Manifest	是

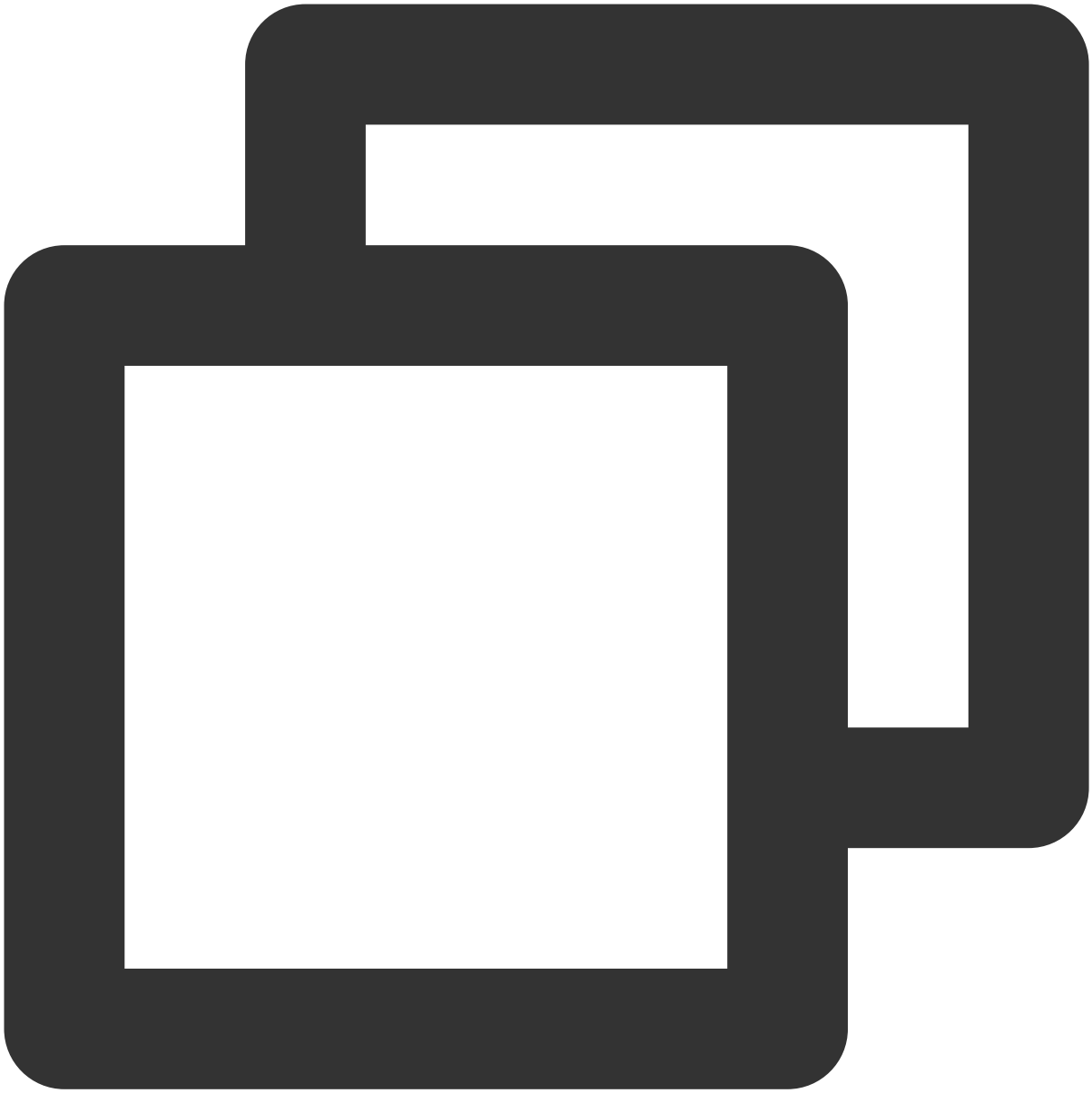
		的对象记录在此对象清单内。您可以使用 清单功能 生成文件清单。	Object	
Operation	CreateJobRequest	选择对清单文件中的对象将要执行的操作。目前 COS 支持 批量复制对象、批量恢复归档 等操作，您可以据此处理存储桶内的存量数据。	Operation Object	是
Priority	CreateJobRequest	任务优先级。越高的数值代表此项任务的优先级越高。优先级数值范围为0 - 2147483647。	Integer	是
Report	CreateJobRequest	任务完成报告。您可配置此参数以在任务完成时输出报告，方便审计任务执行状况。	Report Object	是
RoleArn	CreateJobRequest	COS 资源标识符，此处用于标识您创建的角色。您需要此资源标识符以验证您的身份。	String	是

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<CreateJobResult>
   <JobId>string</JobId>
</CreateJobResult>
```

具体内容描述如下：

节点名	父节点	描述	类型
JobId	CreateJobResult	任务 ID。当您成功创建一项任务后，COS 自动返回的 ID，长度1 - 64字节。	String

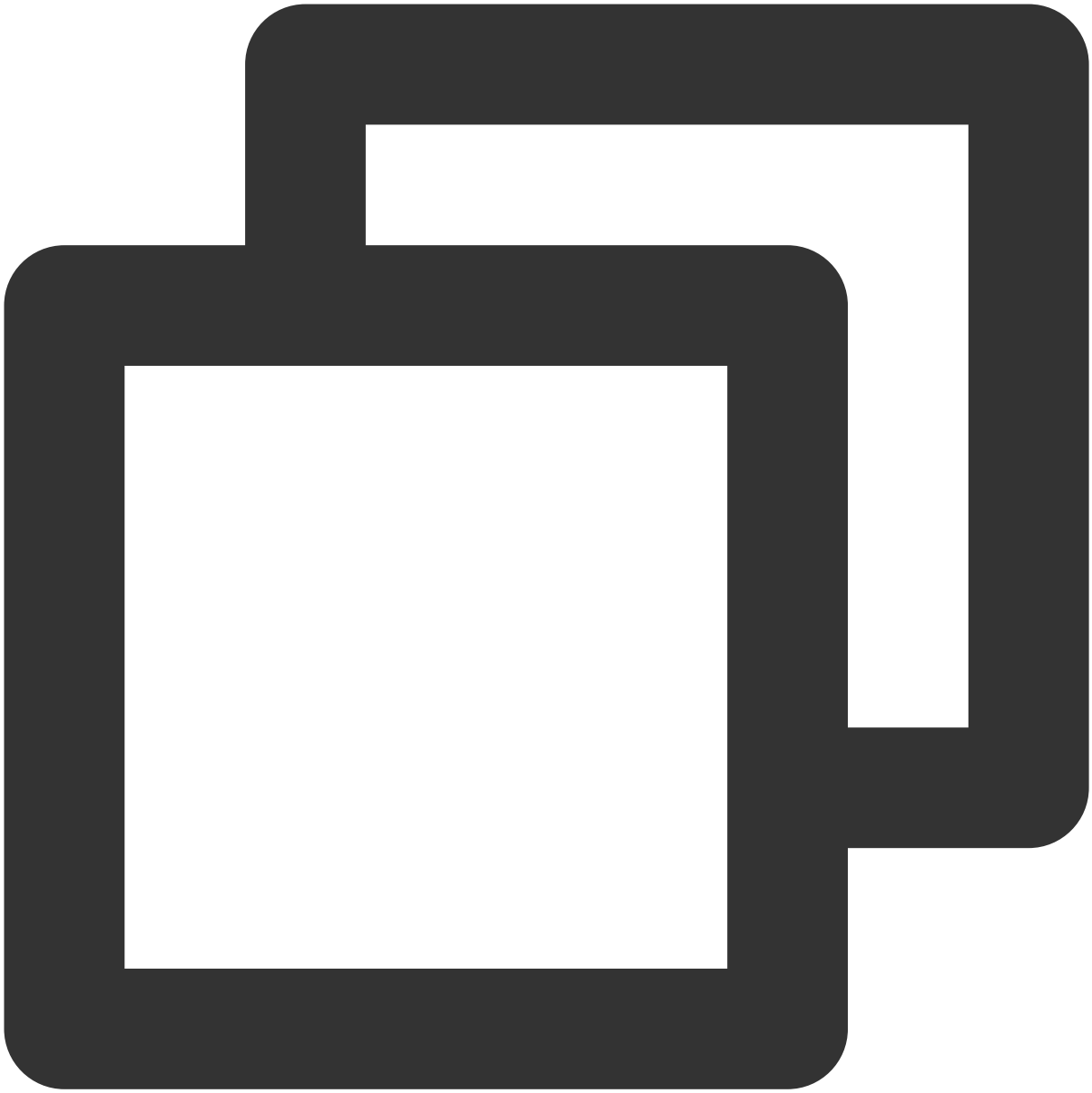
错误分析

该请求操作可能会出现如下错误信息。其他错误请参见 批量处理功能错误响应。

错误代码	描述	状态码	API
InvalidRequest	重复的请求	400	CreateJob
InvalidRequest	优先级的有效范围是0 - 2147483647的整数	400	CreateJob
MalformedXML	请求体的 XML Manifest 字段不符合 XML 语法	400	CreateJob
MalformedXML	请求体的 XML Operation 字段不符合 XML 语法	400	CreateJob
MalformedXML	请求体的 XML Report 字段不符合 XML 语法	400	CreateJob
ServiceUnavailable	服务暂不可用，无法建立新的任务	500	CreateJob
TooManyJobs	任务已达上限，服务器不可用	500	CreateJob

实际案例

请求

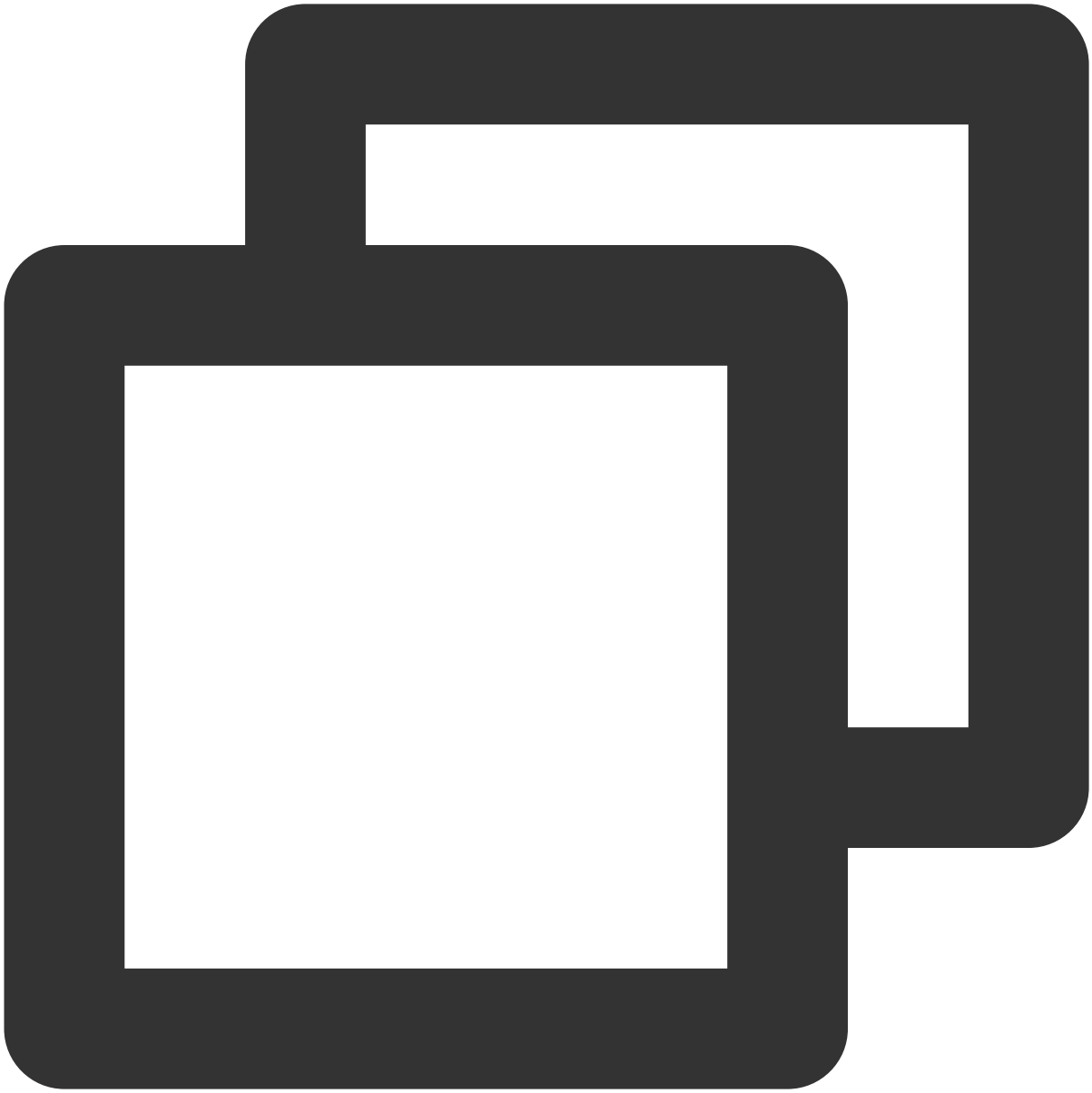


```
POST /jobs HTTP/1.1
Host: 100000000001.cos-control.ap-chengdu.myqcloud.com
Date: Thu, 19 Dec 2019 18:00:29 GMT
x-cos-appid: 1250000000
Content-Type: application/xml
Content-Length: 1056
Content-MD5: hHcgq5mu8s0YP4WTGiQ+uA==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Connection: close

<CreateJobRequest>
```

```
    <ClientRequestToken>184ce261-18af-5e3d-3e30-253723cfd937</ClientRequestToken>
    <ConfirmationRequired>false</ConfirmationRequired>
    <Description>example-job</Description>
    <Manifest>
        <Location>
            <ETag>"15150651828fa9cdcb8356b6d1c7638b"</ETag>
            <ObjectArn>qcs::cos:ap-chengdu::sourcebucket-1250000000/manifests/batch
        </Location>
        <Spec>
            <Fields>
                <member>Bucket</member>
                <member>Key</member>
            </Fields>
            <Format>COSBatchOperations_CSV_V1</Format>
        </Spec>
    </Manifest>
    <Operation>
        <COSPutObjectCopy>
            <TargetResource>qcs::cos:ap-chengdu:uid/1250000000:destinationbucket-12
        </COSPutObjectCopy>
    </Operation>
    <Priority>10</Priority>
    <Report>
        <Bucket>qcs::cos:ap-chengdu::sourcebucket-1250000000</Bucket>
        <Enabled>true</Enabled>
        <Format>Report_CSV_V1</Format>
        <Prefix>job-result</Prefix>
        <ReportScope>AllTasks</ReportScope>
    </Report>
    <RoleArn>qcs::cam::uin/100000000001:roleName/COS_Batch_QcsRole</RoleArn>
</CreateJobRequest>
```

响应

上述请求后，COS 返回以下响应，表明该清单任务已经成功设置完毕。



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 89
Connection: close
Date: Thu, 19 Dec 2019 18:00:30 GMT
Server: tencent-cos
x-cos-request-id: NWRmYmJhYmRfMjViMjU4NjRfNmIzYV8xMDE2****

<CreateJobResult>
    <JobId>53dc6228-c50b-46f7-8ad7-65e7159f1aae</JobId>
</CreateJobResult>
```

DescribeJob

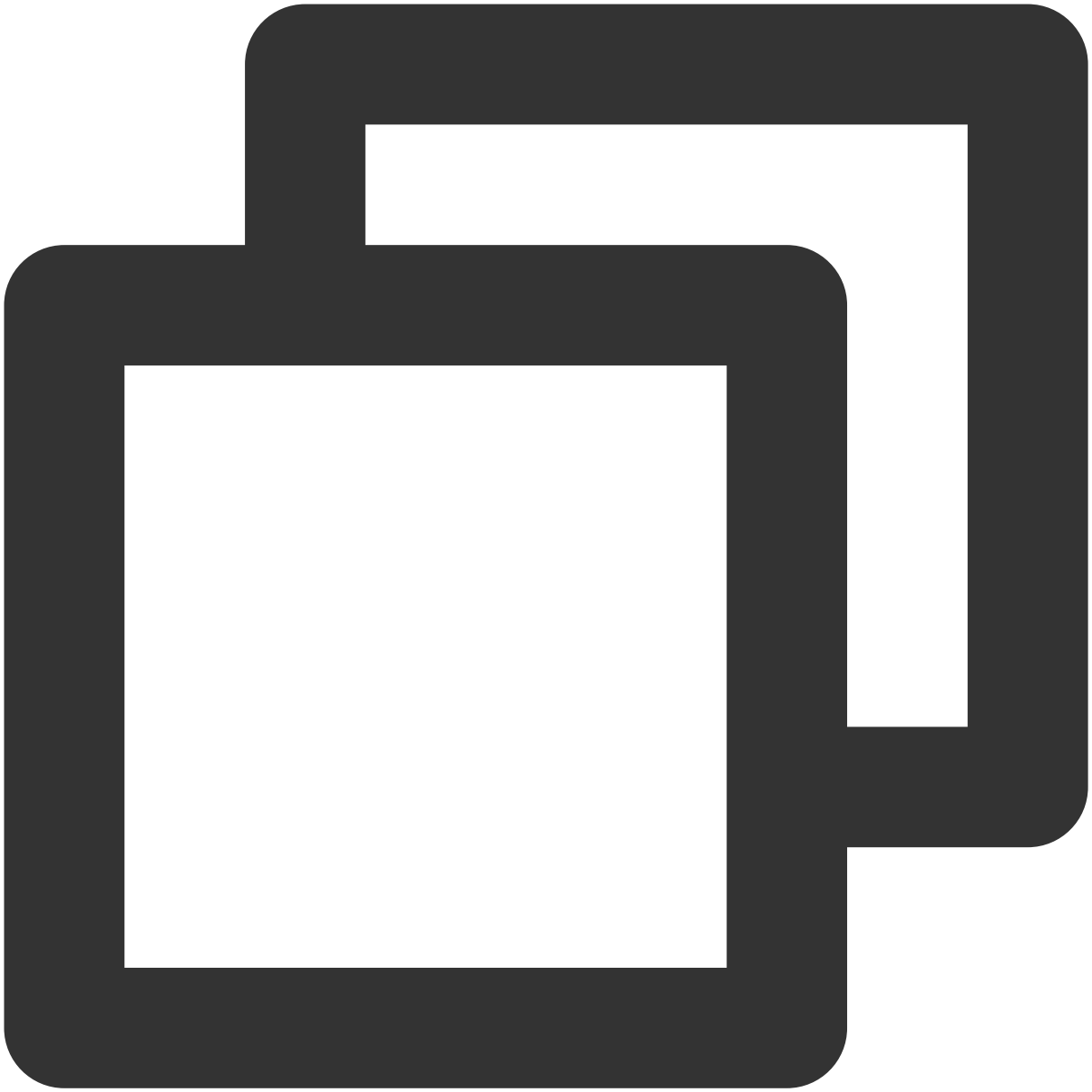
最近更新时间：2023-12-21 17:43:22

功能描述

DescribeJob 用于获取您配置的批量处理任务的参数和任务执行状态。有关批量处理任务的详细信息，可参见 批量处理概述。

请求

请求示例



```
GET /jobs/<JobId> HTTP/1.1
x-cos-appid: <appid>
```

请求参数

调用 DescribeJob 所需的参数。该参数格式如下：

参数	描述	是否必选
JobId	任务 ID。	是
x-cos-appid	用户的 APPID，长度为1 - 64字节。	是

请求头

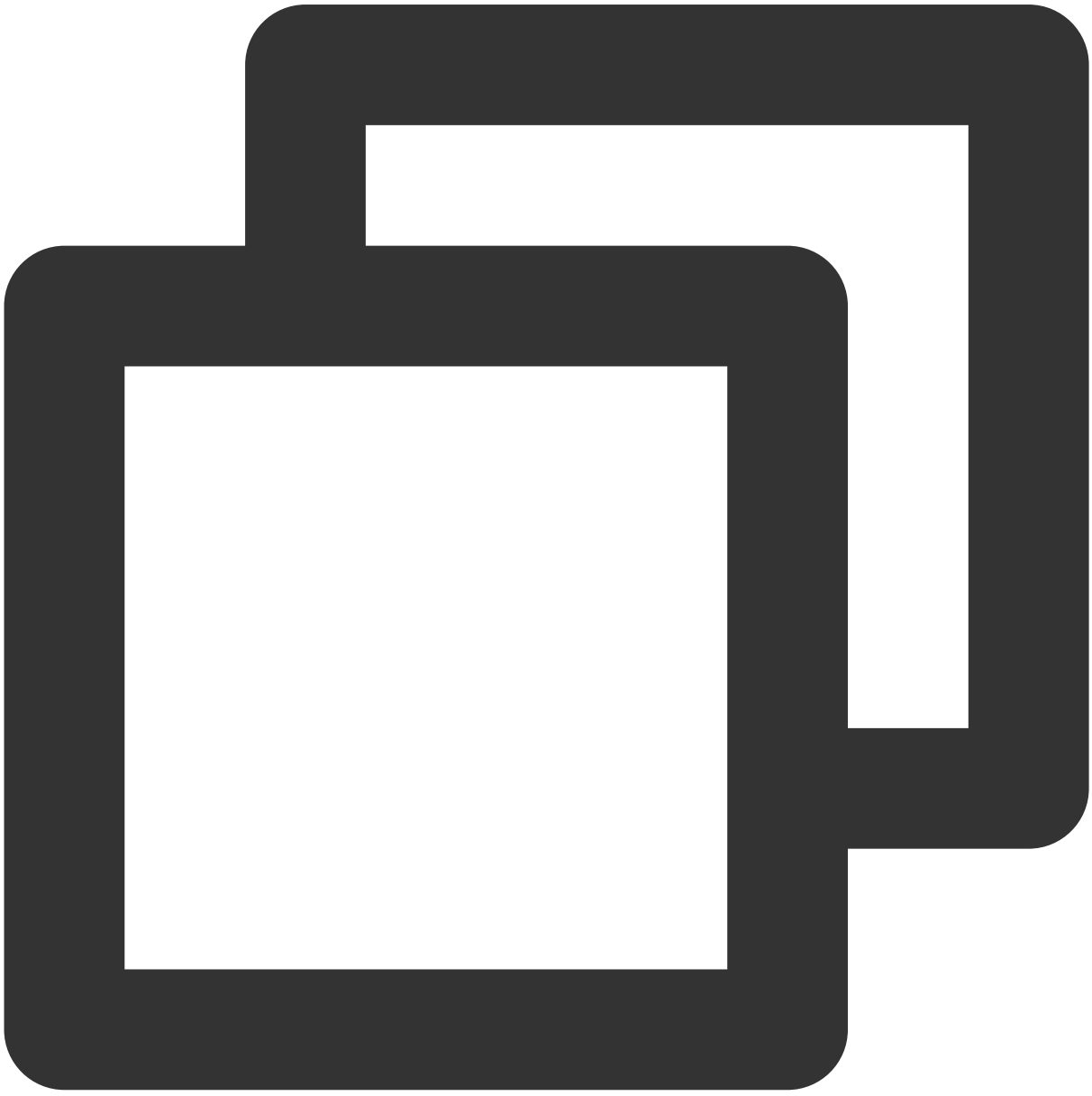
此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应示例

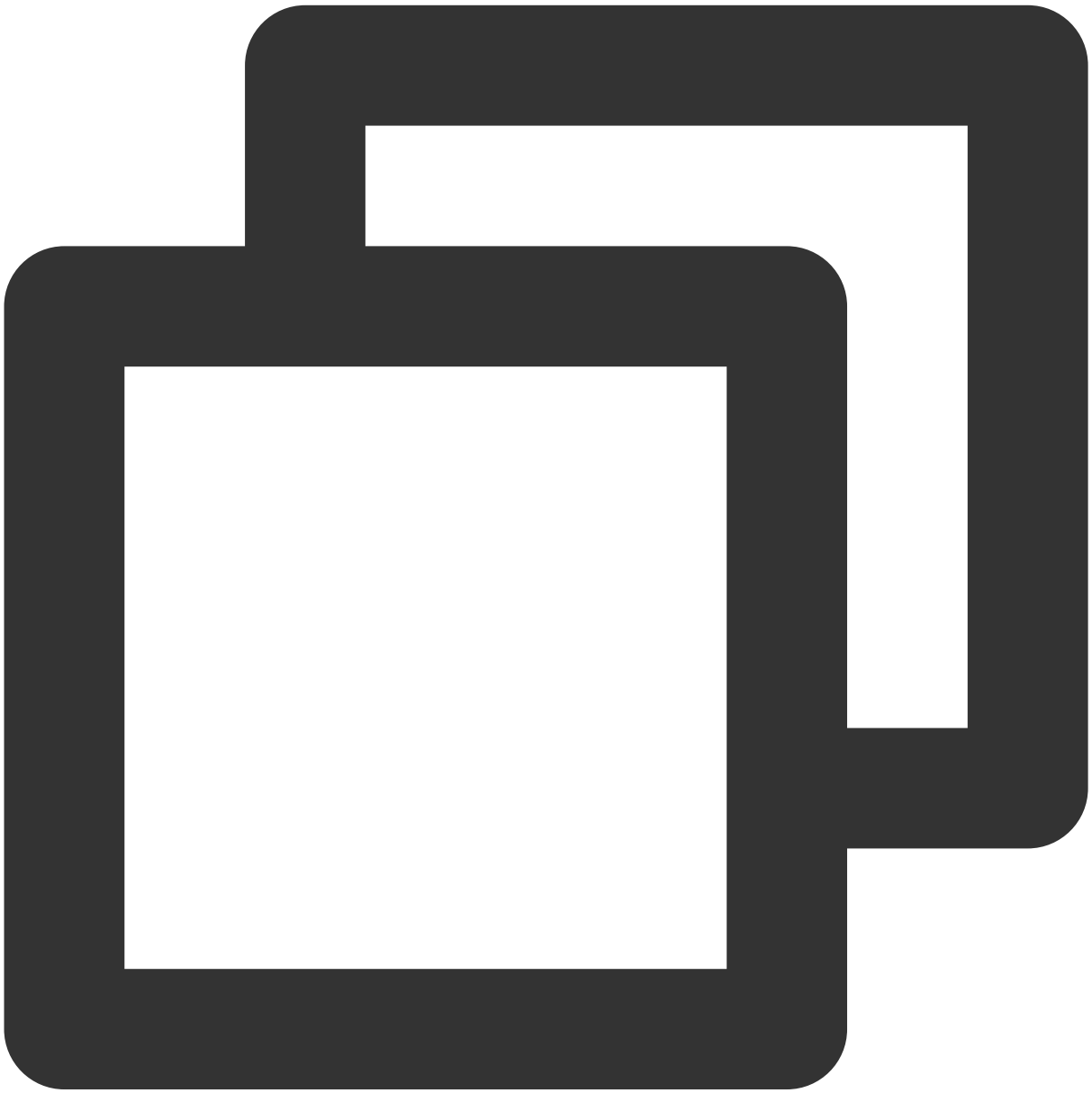


```
HTTP/1.1 200
<DescribeJobResult>
...
</DescribeJobResult>
```

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<DescribeJobResult>
```

```
    <Job>
        <ConfirmationRequired>boolean</ConfirmationRequired>
        <CreationTime>timestamp</CreationTime>
        <Description>string</Description>
        <FailureReasons>
            <JobFailure>
                <FailureCode>string</FailureCode>
                <FailureReason>string</FailureReason>
            </JobFailure>
        </FailureReasons>
        <JobId>string</JobId>
        <Manifest>
            <Location>
                <ETag>string</ETag>
                <ObjectArn>string</ObjectArn>
            </Location>
            <Spec>
                <Format>string</Format>
            </Spec>
        </Manifest>
        <Operation>
            <COSPutObjectCopy>
                <CannedAccessControlList>string</CannedAccessControlList>
                <MetadataDirective>string</MetadataDirective>
                <NewObjectMetadata>
                    <SSEAlgorithm>string</SSEAlgorithm>
                    <UserMetadata>
                        <member>
                            <Key>string</Key>
                            <Value>string</Value>
                        </member>
                        <member>
                            <Key>string</Key>
                            <Value>string</Value>
                        </member>
                    </UserMetadata>
                </NewObjectMetadata>
                <StorageClass>string</StorageClass>
                <TargetResource>string</TargetResource>
            </COSPutObjectCopy>
        </Operation>
        <Priority>integer</Priority>
        <ProgressSummary>
            <NumberOfTasksFailed>integer</NumberOfTasksFailed>
            <NumberOfTasksSucceeded>integer</NumberOfTasksSucceeded>
            <TotalNumberOfTasks>integer</TotalNumberOfTasks>
        </ProgressSummary>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Report>
            <Bucket>string</Bucket>
            <Enabled>boolean</Enabled>
            <Format>string</Format>
            <Prefix>string</Prefix>
            <ReportScope>string</ReportScope>
        </Report>
        <RoleArn>string</RoleArn>
        <Status>string</Status>
        <StatusUpdateReason>string</StatusUpdateReason>
        <SuspendedCause>string</SuspendedCause>
        <SuspendedDate>timestamp</SuspendedDate>
        <TerminationDate>timestamp</TerminationDate>
    </Job>
</DescribeJobResult>
```

具体内容描述如下：

DescribeJobResult

包含待描述批量处理任务的参数和状态信息。

节点名	父节点	描述	类型
Job	DescribeJobResult	您指定查询的批量处理任务的参数和状态信息。	Job Object

Job

节点名	父节点	描述	类型
ClientRequestToken	Job	每个请求唯一的 token，用于避免前端重复发起同一批处理任务。长度为1 - 64字节，建议使用 UUID。	String
CreationTime	Job	任务创建时间。	Timestamp
Description	Job	任务描述。若您在创建任务时配置了此信息，则会返回该项内容。长度范围为1 - 256字节。	String
FailureReasons	Job	如果任务失败，描述失败的原因。	FailureReasons Object
JobId	Job	创建任务成功后，生成的任务 ID。长度1 - 64字节。	String
Manifest	Job	待处理的对象清单。您需要将需要处理的对象记录在此对象清单内。	Manifest Object
Operation	Job	您需要对清单内的对象批量执行的操作。	Operation Object
Priority	Job	任务优先级。越高的数值代表此项任务的优先级越	Integer

		高。优先级数值范围为0 - 2147483647	
ProgressSummary	Job	任务执行状况概述。描述您此项任务中所执行的操作总数，成功的操作数量以及失败的操作数量。	ProgressSummary Object
Report	Job	指定清单报告的相关配置。	Report Object
RoleArn	Job	您为该任务分配的角色的标识符。长度1 - 1024字节。	String
Status	Job	任务当前状态。合法参数值包括 Active、Cancelled、Cancelling、Complete、Completing、Failed、Failing、New、Paused、Pausing、Preparing、Ready、Suspended	String
StatusUpdateReason	Job	状态更新原因。长度为0 - 256字节。	String
SuspendedCause	Job	任务中断的原因。任务仅在您使用控制台创建任务时会进入中断状态，等待您的确认后再进入执行状态。长度为0 - 1024字节。	String
SuspendedDate	Job	任务中断的时间。当任务进入中断状态时，会记录任务中断的时间。	Timestamp
TerminationDate	Job	任务终止的时间。	Timestamp

FailureReasons

节点名	父节点	描述	类型
JobFailure	FailureReasons	任务失败代码和原因。	JobFailure Object

FailureCode

节点名	父节点	描述	类型
FailureCode	JobFailure	任务失败代码。长度0 - 64字节。	String
FailureReason	JobFailure	任务失败原因。长度0 - 256字节。	String

其他元素请参见 批量处理功能公共元素。

错误分析

该请求可能会发生的一些常见的特殊错误如下，其他错误请参见 批量处理功能错误响应。

错误代码	描述	状态码	API
NoSuchJob	指定任务不存在	404	DescribeJob

版权所有：腾讯云计算（北京）有限责任公司

ListJobs

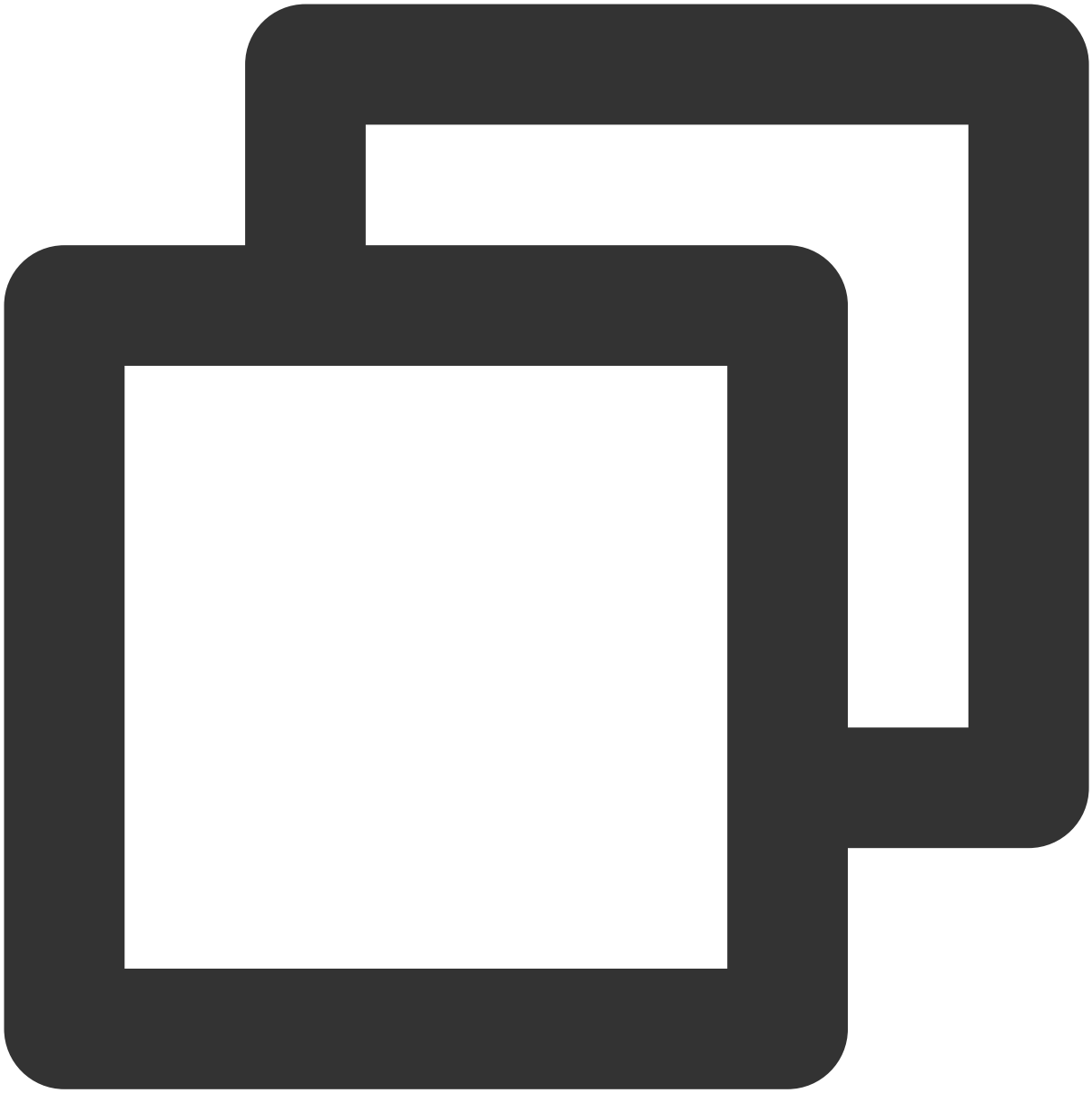
最近更新时间：2023-12-21 17:43:49

功能描述

ListJobs 用于列出您的批量处理任务。有关批量处理任务的详细介绍，可参见 批量处理概述。

请求

请求示例



```
GET /jobs?jobStatuses=<JobStatuses>&maxResults=<MaxResults>&nextToken=<NextToken> H
x-cos-appid: <appid>
```

请求参数

调用 ListJobs 所需要使用的参数。该参数格式如下：

参数	描述	是否必选
jobStatuses	您所需查询的任务状态信息。如您未指定任务状态，则 COS 将返回所有执行过的任务状态，包括正在执行的任务。若您指定任务状态，则 COS 将返回指定状态的任	否

	务。可选的任务状态包括：Active、Cancelled、Cancelling、Complete、Completing、Failed、Failing、New、Paused、Pausing、Preparing、Ready、Suspended。	
maxResults	COS 返回的任务数量最大值。如您配置了此项参数，每次返回的任务数最多不会超过这个最大值。您可以配合 nextToken 参数进行分页返回。该参数范围为1 - 1000，默认为1000。	否
nextToken	翻页符。每次 list 操作会返回本次任务列表的最后一个 JobId 作为 nextToken，在下一次 list 传入该值，即可接续上一次 list 的内容进行 list，用于分页。该参数的长度限制在1 - 64字节。	否
x-cos-appid	用户的 APPID，长度为1 - 64字节。	是

请求头

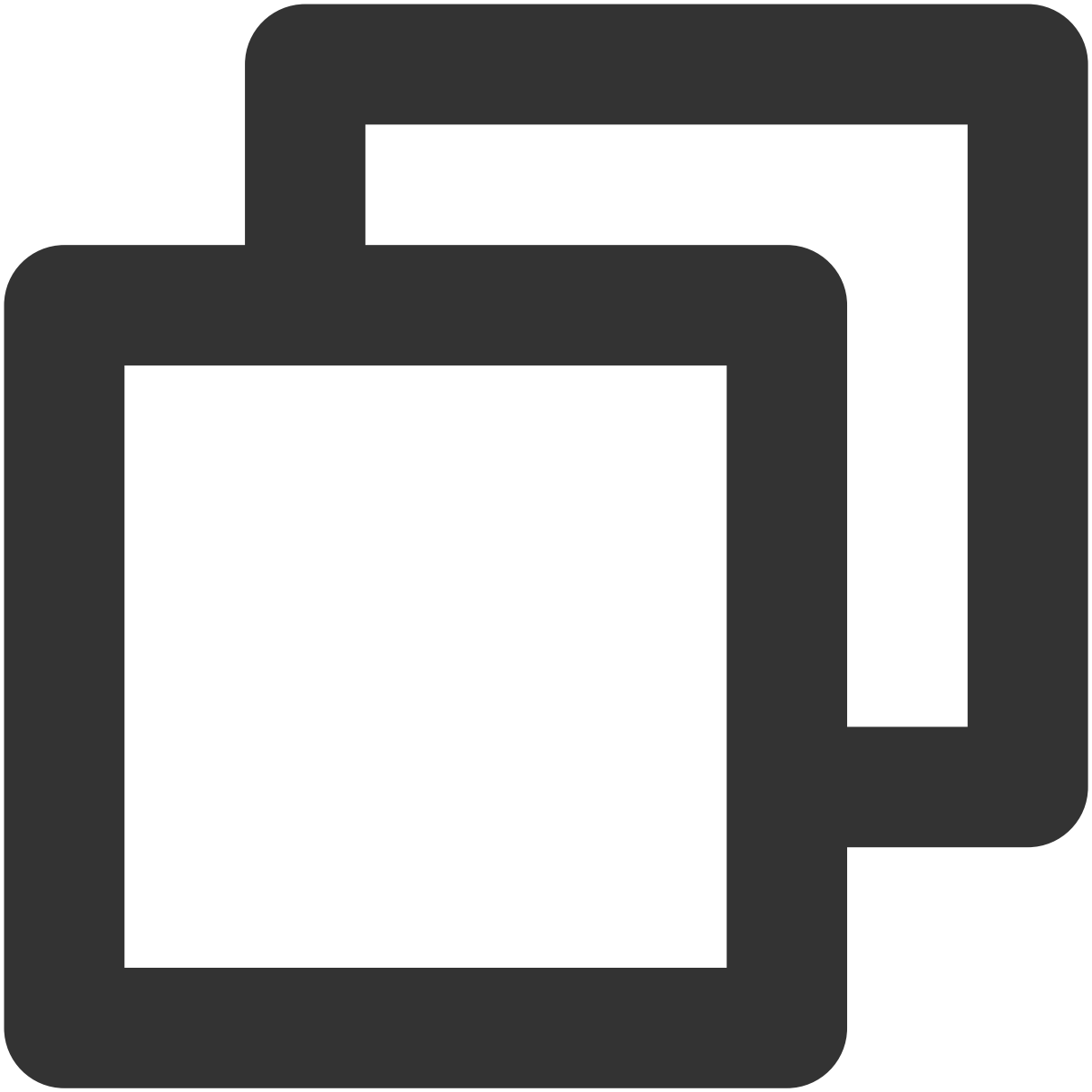
此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应示例

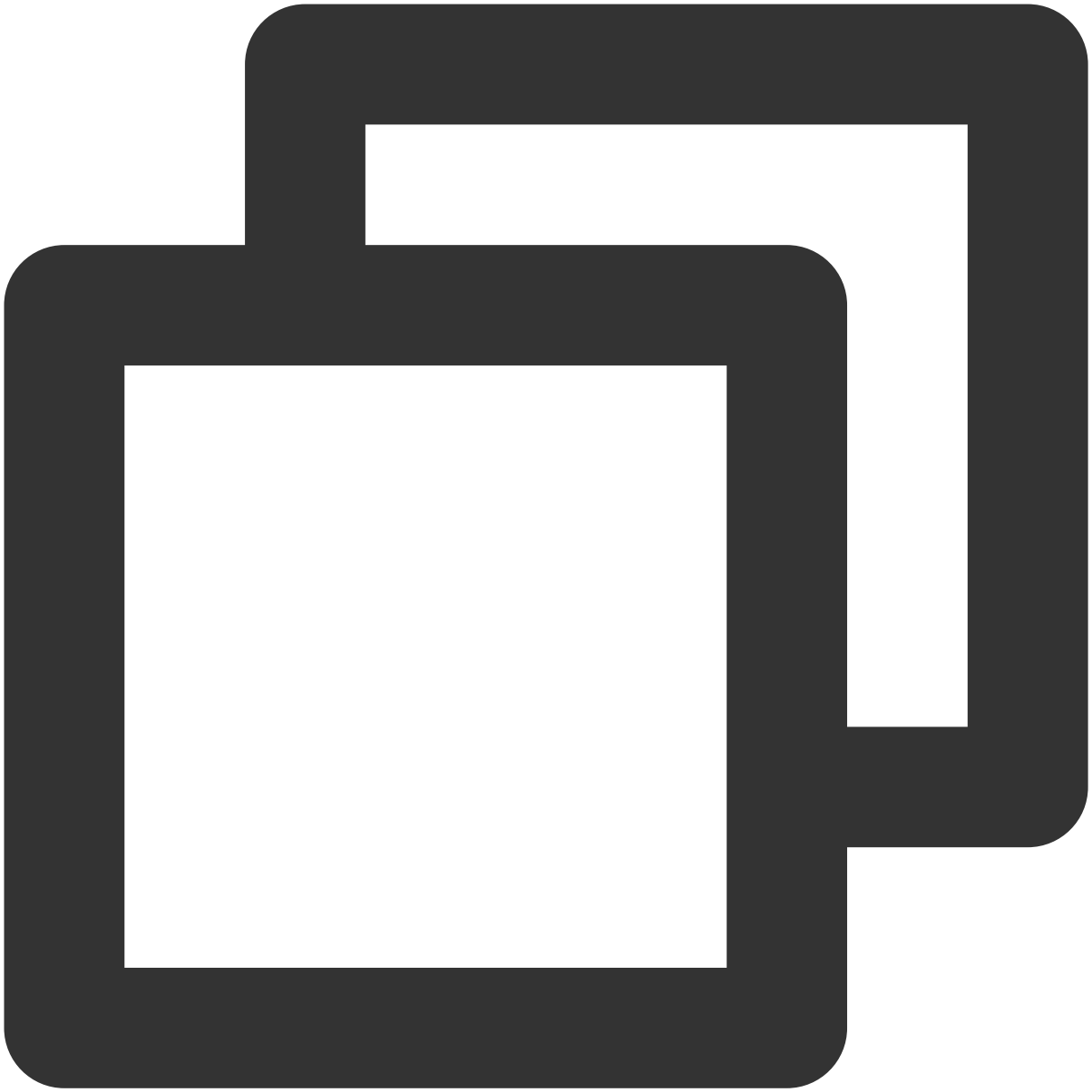


```
HTTP/1.1 200
<ListJobsResult>
...
</ListJobsResult>
```

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<ListJobsResult>
    <Jobs>
        <member>
            <CreationTime>timestamp</CreationTime>
            <Description>string</Description>
            <JobId>string</JobId>
            <Operation>string</Operation>
            <Priority>integer</Priority>
            <ProgressSummary>
                <NumberOfTasksFailed>integer</NumberOfTasksFailed>
                <NumberOfTasksSucceeded>integer</NumberOfTasksSucceeded>
```

```
                <TotalNumberOfTasks>integer</TotalNumberOfTasks>
            </ProgressSummary>
            <Status>string</Status>
            <TerminationDate>timestamp</TerminationDate>
        </member>
    </Jobs>
    <NextToken></NextToken>
</ListJobsResult>
```

具体内容描述如下：

ListJobsResult

包含COS返回的批量处理任务列表信息。

节点名	父节点	描述	类型
Jobs	ListJobsResult	包含 COS 返回的多个批量处理任务信息。	Jobs Object
NextToken	ListJobsResult	每次ListJobs操作会返回本次任务列表的最后一个 JobId 作为 nextToken，在下一次 ListJobs 传入该值，返回结果即可接续上一次 ListJobs 的内容，用于分页。该参数的长度限制在1 - 64字节。	String

Jobs

节点名	父节点	描述	类型
member	Jobs	包含 COS 返回的某个批量处理任务信息。	member Object

member

节点名	父节点	描述	类型
CreationTime	member	任务创建时间。	Timestamp
Description	member	任务描述。长度限制为0 - 256字节。	String
JobId	member	任务 ID。长度限制为1 - 64字节。	String
Operation	member	批量处理任务中对对象执行的操作。例如 COSPutObjectCopy	String
Priority	member	任务优先级。值越大的任务将会被优先执行。优先级大小限制为0 - 2147483647。	Integer
ProgressSummary	member	任务执行状况概述。描述您此项任务中所执行的操作总数，成功的操作数量以及失败的操作数量。	ProgressSummary Object

版权所有：腾讯云计算（北京）有限责任公司

Status	member	任务执行状态。合法参数值包括 Active、Cancelled、Cancelling、Complete、Completing、Failed、Failing、New、Paused、Pausing、Preparing、Ready、Suspended。	String
TerminationDate	member	批量处理任务的结束时间。	Timestamp

其他元素请参见 批量处理功能公共元素。

错误分析

该请求可能会发生的一些常见的特殊错误如下：

错误代码	描述	状态码	API
InvalidArgument	Parameter jobStatuses is invalid	400	ListJobs
InvalidArgument	maxResults 参数必须为整数	400	ListJobs
InvalidArgument	nextToken 参数无效	400	ListJobs

其他错误请参考 批量处理功能错误响应。

版权所有：腾讯云计算（北京）有限责任公司

UpdateJobPriority

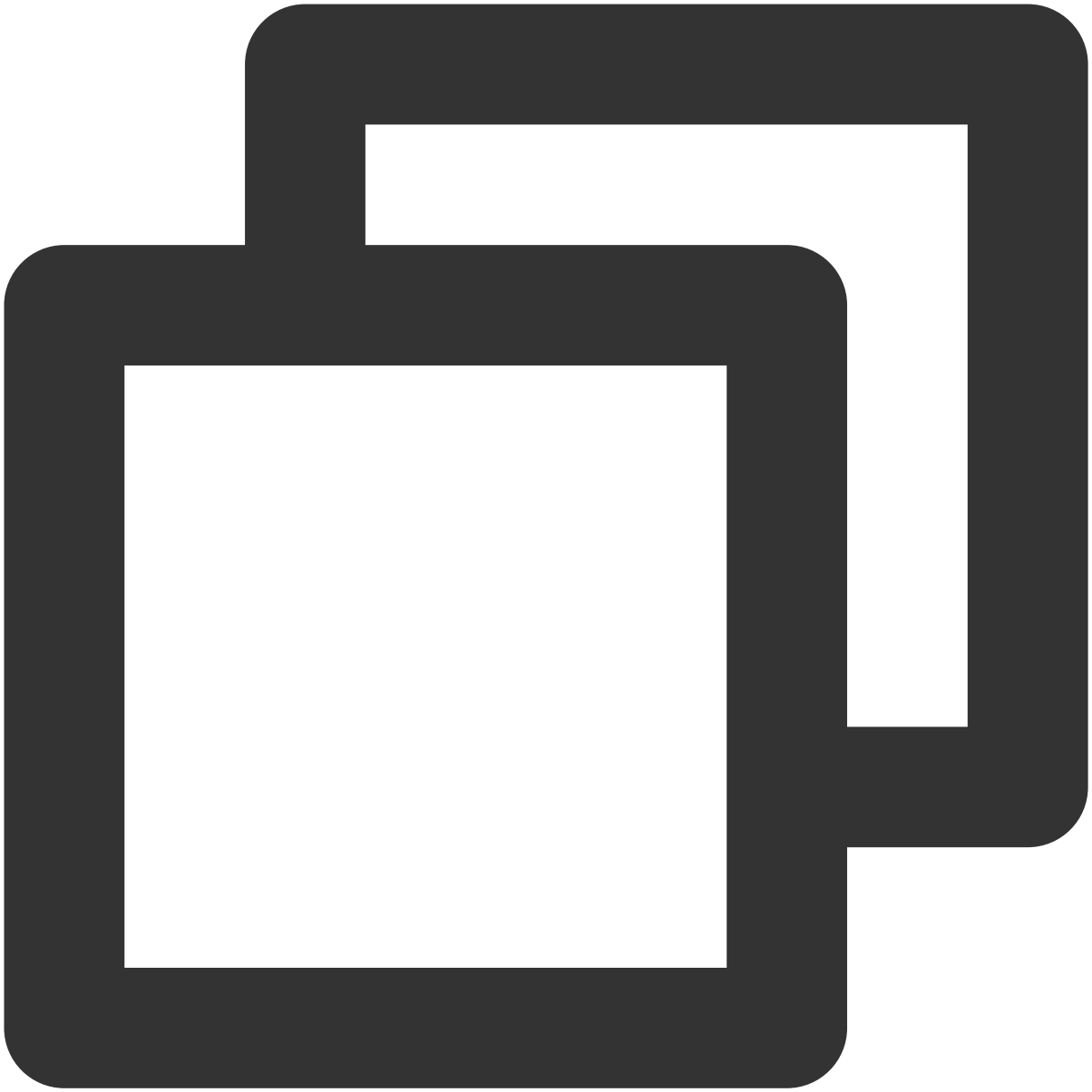
最近更新时间：2023-12-21 17:43:45

功能描述

UpdateJobPriority 用于更新任务优先级。您可以使用这一接口以更新一项任务的优先级，数值越大，优先级越高，高优先级的 Job 会被优先执行。有关批量处理任务的详细介绍，可参见 批量处理概述。

请求

请求示例



```
POST /jobs/<JobId>/priority?priority=<Priority> HTTP/1.1
x-cos-appid: <appid>
```

请求参数

调用 UpdateJobPriority 接口所需的参数。该参数格式如下：

参数	描述	是否必选
JobId	您想要更新的批量处理任务的 ID。长度限制为1 - 64字节。	是
priority	更新后的任务优先级。此项参数大小限制为0 - 2147483647。	是

x-cos-appid	用户的 APPID，长度为1 - 64字节。	是

请求头

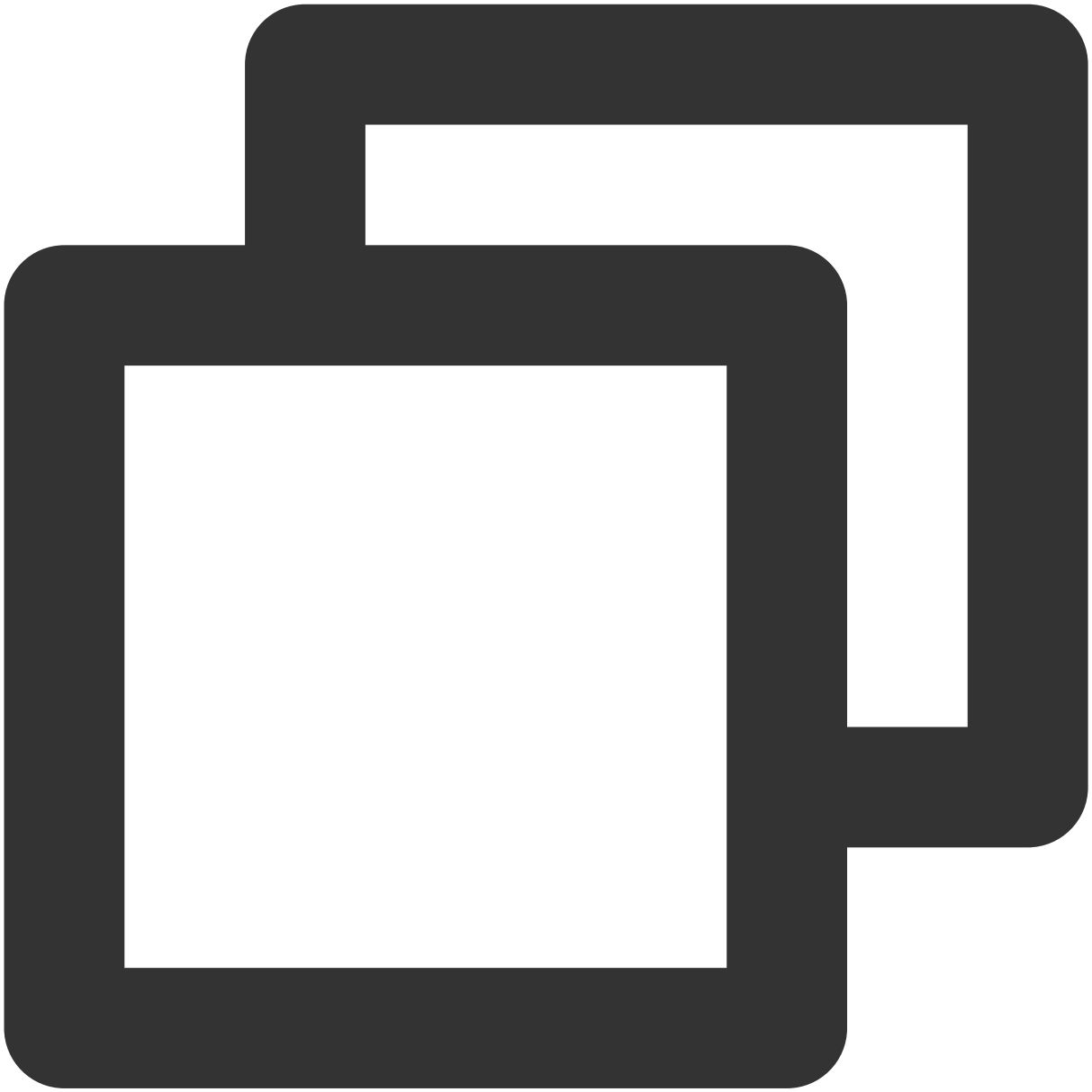
此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应示例

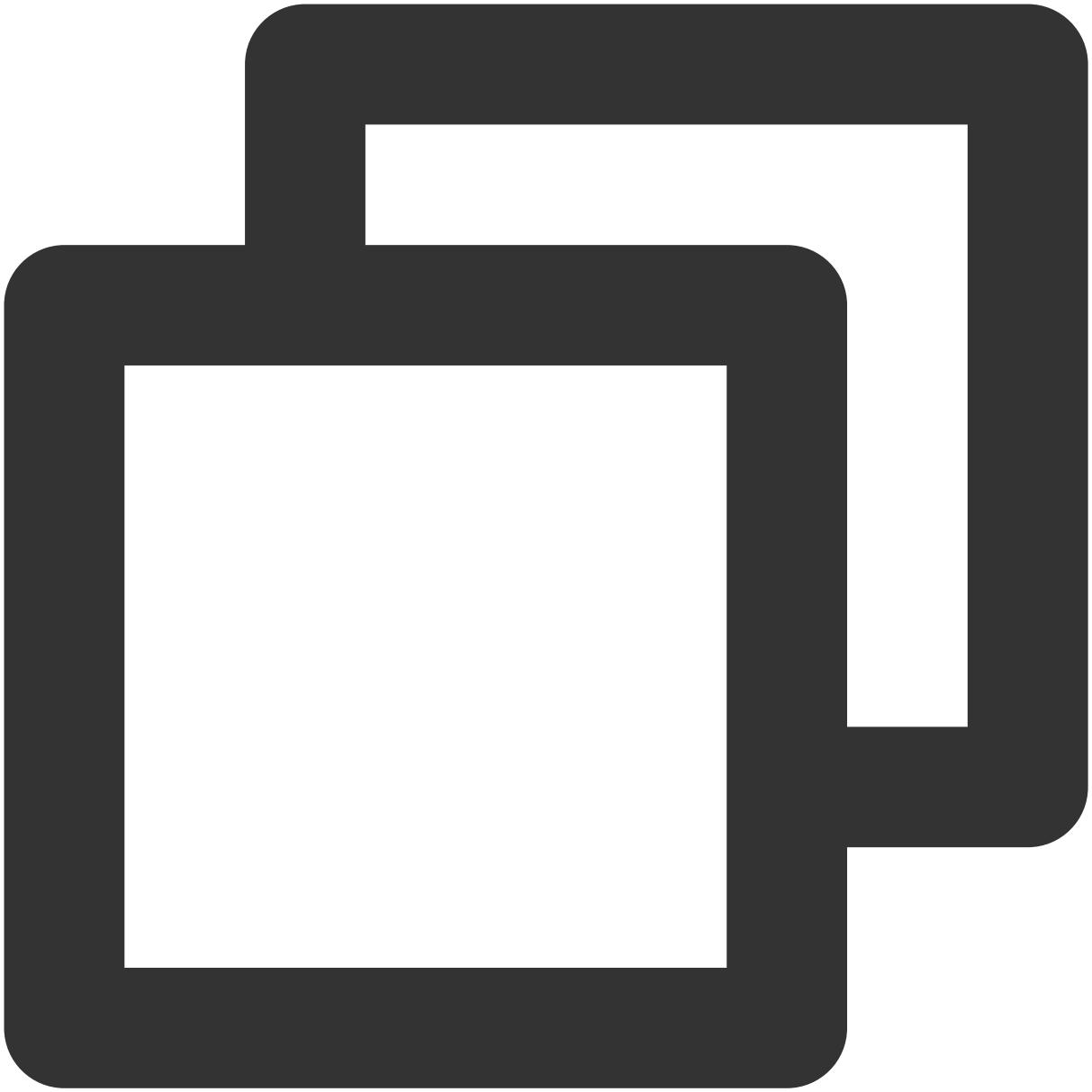


```
HTTP/1.1 200
<UpdateJobPriorityResult>
    <JobId>string</JobId>
    <Priority>integer</Priority>
</UpdateJobPriorityResult>
```

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<UpdateJobPriorityResult>
    <JobId>string</JobId>
    <Priority>integer</Priority>
</UpdateJobPriorityResult>
```

具体内容描述如下：

UpdateJobStatusResult

节点名	父节点	描述	类型
JobId	UpdateJobPriorityResult	任务 ID。您所更新的任务的 ID，长度限制为1 - 64字节。	String

Priority	UpdateJobPriorityResult	任务的当前优先级。大小限制为0 - 2147483647。	Integer

错误分析

该请求可能会发生的一些常见的特殊错误如下：

错误代码	描述	状态码	API
InvalidRequest	未提供任务优先级	400	UpdateJobPriority
NoSuchJob	指定任务不存在或已完成	404	UpdateJobStatus，UpdateJobPriority

其他错误请参见 批量处理功能错误响应。

UpdateJobStatus

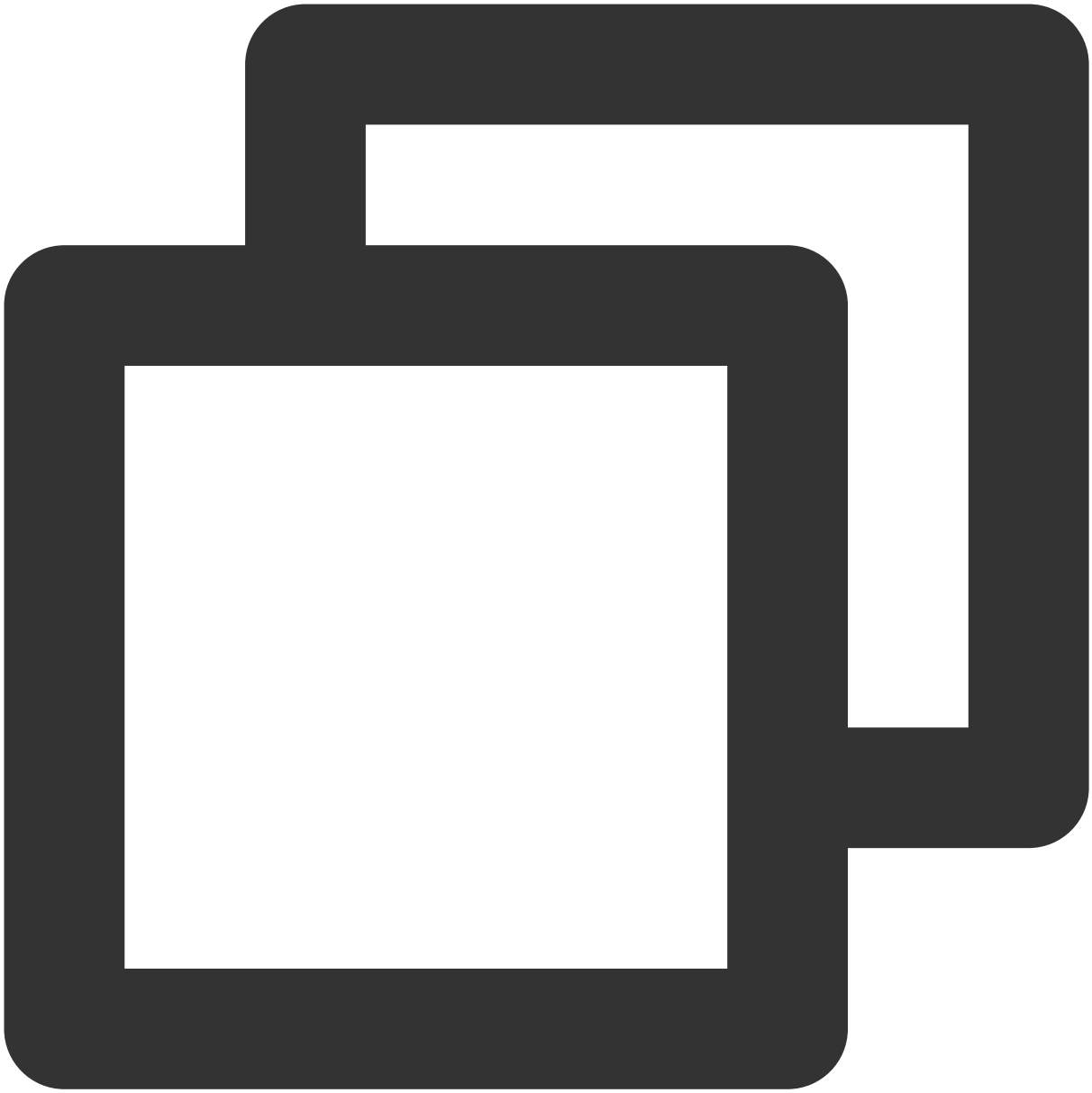
最近更新时间：2023-12-21 17:44:13

功能描述

UpdateJobStatus 用于更新任务状态。您可以使用这一接口以启动一项任务或者取消一项正在进行的任务。有关批量处理任务的详细介绍，可参见 批量处理概述。

请求

请求示例



```
POST /jobs/<JobId>/status?requestedJobStatus=<RequestedJobStatus>&statusUpdateReaso
x-cos-appid: <appid>
```

请求参数

调用 UpdateJobStatus 需要使用清单任务名称的参数。该参数格式如下：

参数	描述	是否必选
JobId	您想要更新的批量处理任务的 ID。	是

requestedJobStatus	您期望的批量处理任务的状态。当您将任务转移至 `Ready` 状态时，COS 将认为您已确认此项任务，并将执行此项任务。当您将任务转移至 `Cancelled` 状态时，COS 将取消此项任务。可选参数包括：Ready、Cancelled。	是
statusUpdateReason	更新任务状态的原因。此项参数长度限制为0 - 256字节。	否
x-cos-appid	用户的 APPID，长度为1 - 64字节。	是

请求头

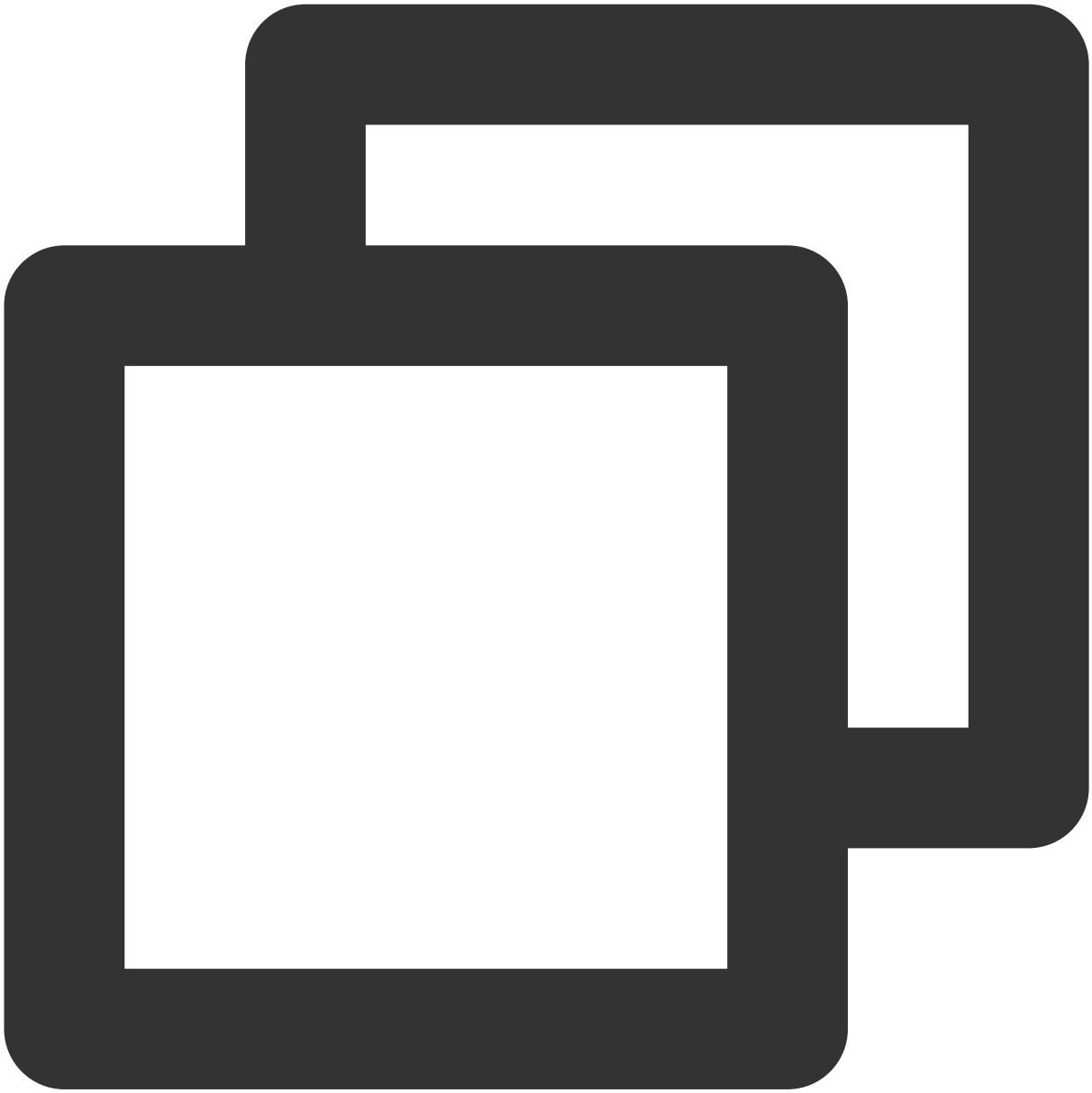
此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应示例

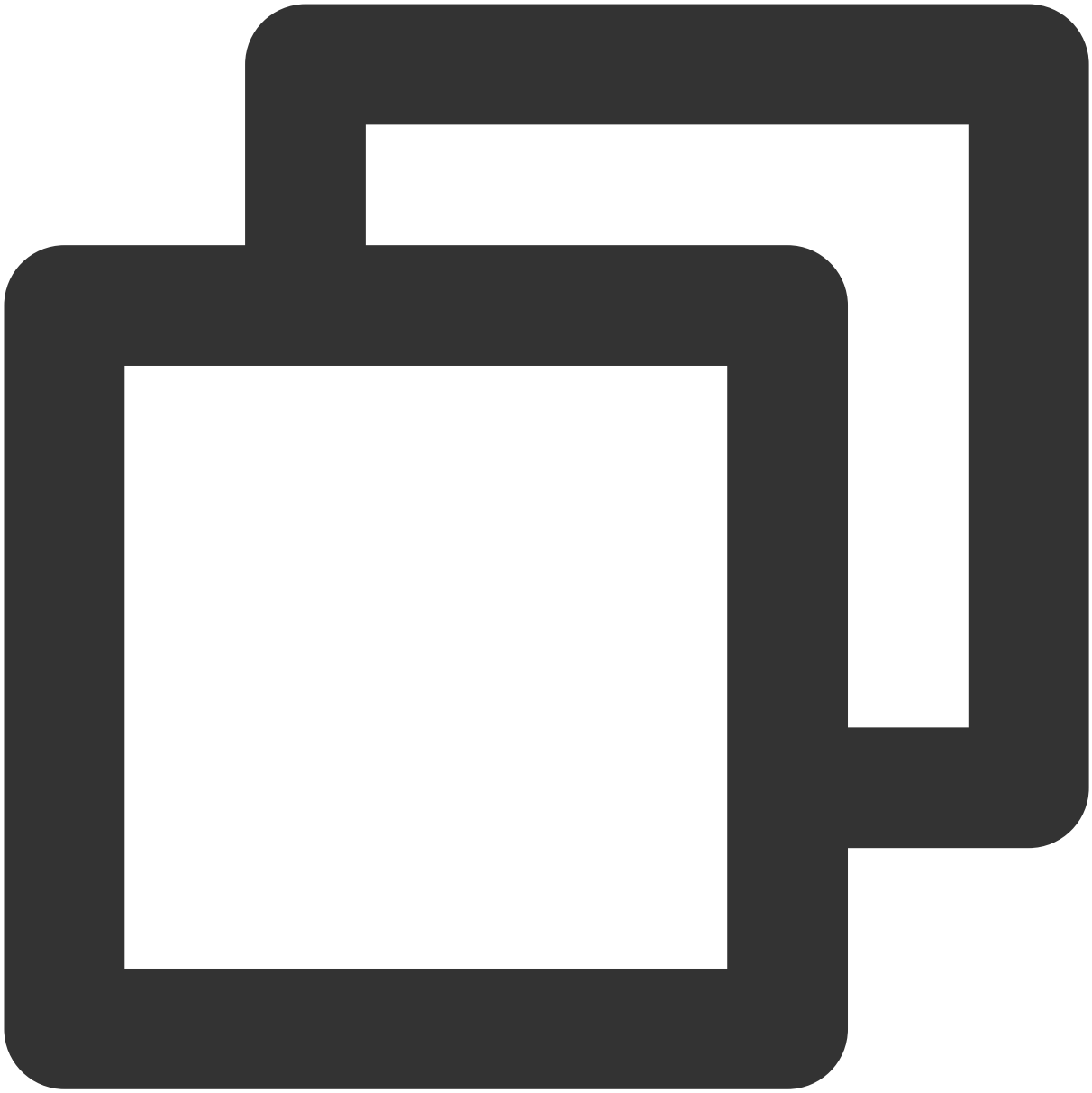


```
HTTP/1.1 200
<UpdateJobStatusResult>
    <JobId>string</JobId>
    <Status>string</Status>
    <StatusUpdateReason>string</StatusUpdateReason>
</UpdateJobStatusResult>
```

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<UpdateJobStatusResult>
   <JobId>string</JobId>
   <Status>string</Status>
   <StatusUpdateReason>string</StatusUpdateReason>
</UpdateJobStatusResult>
```

具体内容描述如下。

UpdateJobStatusResult

节点名	父节点	描述	类型

JobId	UpdateJobStatusResult	任务 ID。您所更新的任务的 ID，长度限制为5 - 36字节。	String
Status	UpdateJobStatusResult	任务的当前状态。合法参数项包括：Active、Cancelled、Cancelling、 Complete、Completing、Failed、Failing、New、Paused、Pausing、Preparing、Ready、Suspended。	String
StatusUpdateReason	UpdateJobStatusResult	任务更新原因。长度限制为0 - 256字节。	String

错误分析

该请求可能会发生的一些常见的特殊错误如下：

错误代码	描述	状态码	API
InvalidArgument	requestedJobStatus 参数必须为 Cancelled 或 Ready	400	UpdateJobStatus
InvalidRequest	指定任务已完成	400	UpdateJobStatus
InvalidRequest	任务状态变更错误	400	UpdateJobStatus
NoSuchJob	指定任务不存在或已完成	404	UpdateJobStatus, UpdateJobPriority

其他错误请参见 批量处理功能错误响应。

版权所有：腾讯云计算（北京）有限责任公司

公共元素

最近更新时间：2023-12-21 17:44:20

本文提供以下批量处理功能的公共元素。

Manifest

节点名	父节点	描述	类型	是否必选
Location	Manifest	对象清单的位置信息。	Location Object	是
Spec	Manifest	描述对象清单的格式信息。如果为 CSV 文件，此元素将描述清单中包含的字段。	Spec Object	是

Location

节点名	父节点	描述	类型	是否必选
ETag	Location	指定对象清单的 etag。长度1 - 1024字节。	String	是
ObjectArn	Location	指定对象清单的唯一资源标识符，长度为1 - 1024字节。	String	是
ObjectVersionId	Location	指定对象清单的版本 ID，长度为1 - 1024字节。	String	否

Spec

节点名	父节点	描述	类型	是否必选
Fields	Spec	描述清单中包含的字段，当 Format 为 COSBatchOperations_CSV_V1 时，需要使用此元素指定 CSV 文件字段。合法字段为：Ignore、Bucket、Key、VersionId	Array of Strings	否
Format	Spec	指定对象清单的格式信息。合法字段为：COSBatchOperations_CSV_V1、COSInventoryReport_CSV_V1	String	是

版权所有：腾讯云计算（北京）有限责任公司

Operation

注意

单个任务中 Operation 只能包含下列多种操作中的一个。

节点名	父节点	描述	类型	是否必选
COSPutObjectCopy	Operation	指定对清单内的对象批量复制操作的具体参数。	COSPutObjectCopyObject	否
COSInitiateRestoreObject	Operation	指定对清单内的归档存储类型对象批量恢复操作的具体参数。	COSInitiateRestoreObject	否

COSPutObjectCopy

节点名	父节点	描述
AccessControlDirective	COSPutObjectCopy	该元素指定 ACL 的复制方式，有效值：Copy、Replaced、Add。 Copy：继承源对象ACL； Replaced ：替换源 ACL； Add：在源 ACL 的基础上添加新的 ACL。
AccessControlGrants	COSPutObjectCopy	控制对象的具体访问权限。
CannedAccessControlList	COSPutObjectCopy	定义对象的 ACL 属性。有效值：private、public-read
PrefixReplace	COSPutObjectCopy	指定是否需要替换源对象的前缀，值为 true 时，表示替换对象前缀，需搭配 `<ResourcesPrefix>` 和 `<TargetKeyPrefix>` 使用。默认值：false。
ResourcesPrefix	COSPutObjectCopy	当 `<PrefixReplace>` 值为 true 时，这个字段才有效。指定需要替换的源对象前缀，替换目录请以/结尾。可以为空，最长1024字节。
TargetKeyPrefix	COSPutObjectCopy	当 `<PrefixReplace>` 值为 true 时，这个字段才有效。该值表示替换后的前缀，替换目录请以/结尾。可以为空，最长1024字节。

版权所有：腾讯云计算（北京）有限责任公司

		例：源对象为 picture.jpg，设置 `ResourcesPrefix` 值为 pic，`TargetKeyPrefix` 值为 abc，则执行结果为 picture.jpg 变为 abcture.jpg。 注意： `ResourcesPrefix` 为空、`TargetKeyPrefix` 有赋值，可实现增加前缀操作； `ResourcesPrefix` 和 `TargetKeyPrefix` 同时有赋值，可实现替换前缀操作； 若 `ResourcesPrefix` 有赋值、`TargetKeyPrefix` 为空，可实现删除前缀操作。
ModifiedSinceConstraint	COSPutObjectCopy	当对象在指定时间后被修改，则执行操作，否则返回412。
UnModifiedSinceConstraint	COSPutObjectCopy	当对象在指定时间后未被修改，则执行操作，否则返回412。
MetadataDirective	COSPutObjectCopy	该元素指定是从源对象复制对象元数据还是替换为 `<NewObjectMetadata>` 元素中的元数据，有效值为：Copy、Replaced、Add。Copy：继承源对象元数据；Replaced ：替换源元数据；Add：在源元数据的基础上添加新的元数据。
NewObjectMetadata	COSPutObjectCopy	配置对象的元数据。
TaggingDirective	COSPutObjectCopy	该元素指定是从源对象复制对象标签还是替换为 `<NewObjectTagging>` 元素中的标签，有效值为：Copy、Replaced、Add。Copy：继承源对象标签；Replaced ：替换源标签；Add：在源标签的基础上添加新的标签。
NewObjectTagging	COSPutObjectCopy	配置对象的标签，当 `<TaggingDirective>` 值为 Replace 或 Add 时必须指定。
StorageClass	COSPutObjectCopy	设置对象的存储级别，枚举值：STANDARD，STANDARD_IA。默认值：STANDARD
TargetResource	COSPutObjectCopy	设置 Copy 的目标存储桶。请使用 qcs 指定，例如 `qcs::cos:ap-chengdu:uid/1250000000:examplebucket-1250000000`

COSInitiateRestoreObject

节点名	父节点	描述	类型	是否必选

ExpirationInDays	COSInitiateRestoreObject	设置副本在多少天后自动过期删除，设置范围为1 - 365的整数。	Integer	是
JobTier	COSInitiateRestoreObject	归档恢复模式选择，可选值：Bulk、Standard。	String	是

AccessControlGrants

节点名	父节点	描述	类型	是否必选
COSGrant	AccessControlGrants	配置一项权限控制。	COSGrant Object	否

COSGrant

节点名	父节点	描述	类型	是否必选
Grantee	COSGrant	指定将权限授予给哪个用户。	Grantee Object	是
Permission	COSGrant	指定要授予的某项权限。枚举值：READ，WRITE，FULL_CONTROL 。	String	是

Grantee

节点名	父节点	描述	类型	是否必选
DisplayName	Grantee	用户名称。	String	否
Identifier	Grantee	qcs 格式的用户 ID（UIN）。例如：`qcs::cam::uin/100000000001:uin/100000000001`	String	是
TypeIdentifier	Grantee	指定 Identifier 的类型，目前仅支持用户 ID。枚举值：ID 。	String	是

NewObjectMetadata

节点名	父节点	描述	类型	是否必选

CacheControl	NewObjectMetadata	RFC 2616 中定义的缓存指令，将作为对象元数据保存。	String	否
ContentDisposition	NewObjectMetadata	RFC 2616 中定义的文件名称，将作为对象元数据保存。	String	否
ContentEncoding	NewObjectMetadata	RFC 2616 中定义的编码格式，将作为对象元数据保存。	String	否
ContentType	NewObjectMetadata	RFC 2616 中定义的内容类型，将作为对象元数据保存。	String	否
HttpExpiresDate	NewObjectMetadata	RFC 2616 中定义的缓存失效时间，将作为对象元数据保存。	String	否
SSEAlgorithm	NewObjectMetadata	服务端加密算法，目前仅支持 AES256。	String	否
UserMetadata	NewObjectMetadata	包括用户自定义元数据。	Array of Key and Value	否

Report

节点名	父节点	描述	类型	是否必选
Bucket	Report	任务完成报告的投递存储桶。	String	是
Enabled	Report	是否输出任务完成报告。	Boolean	是
Format	Report	任务完成报告格式信息。合法值： Report_CSV_V1	String	是
Prefix	Report	任务完成报告的前缀信息。长度0 - 256字节	String	否
ReportScope	Report	任务完成报告所需记录的任务信息，以确定记录所有操作执行信息还是失败操作的信息。合法值：AllTasks、FailedTasksOnly	String	是

ProgressSummary

节点名	父节点	描述	类型
NumberOfTasksFailed	ProgressSummary	当前失败的操作数。	Integer

版权所有：腾讯云计算（北京）有限责任公司

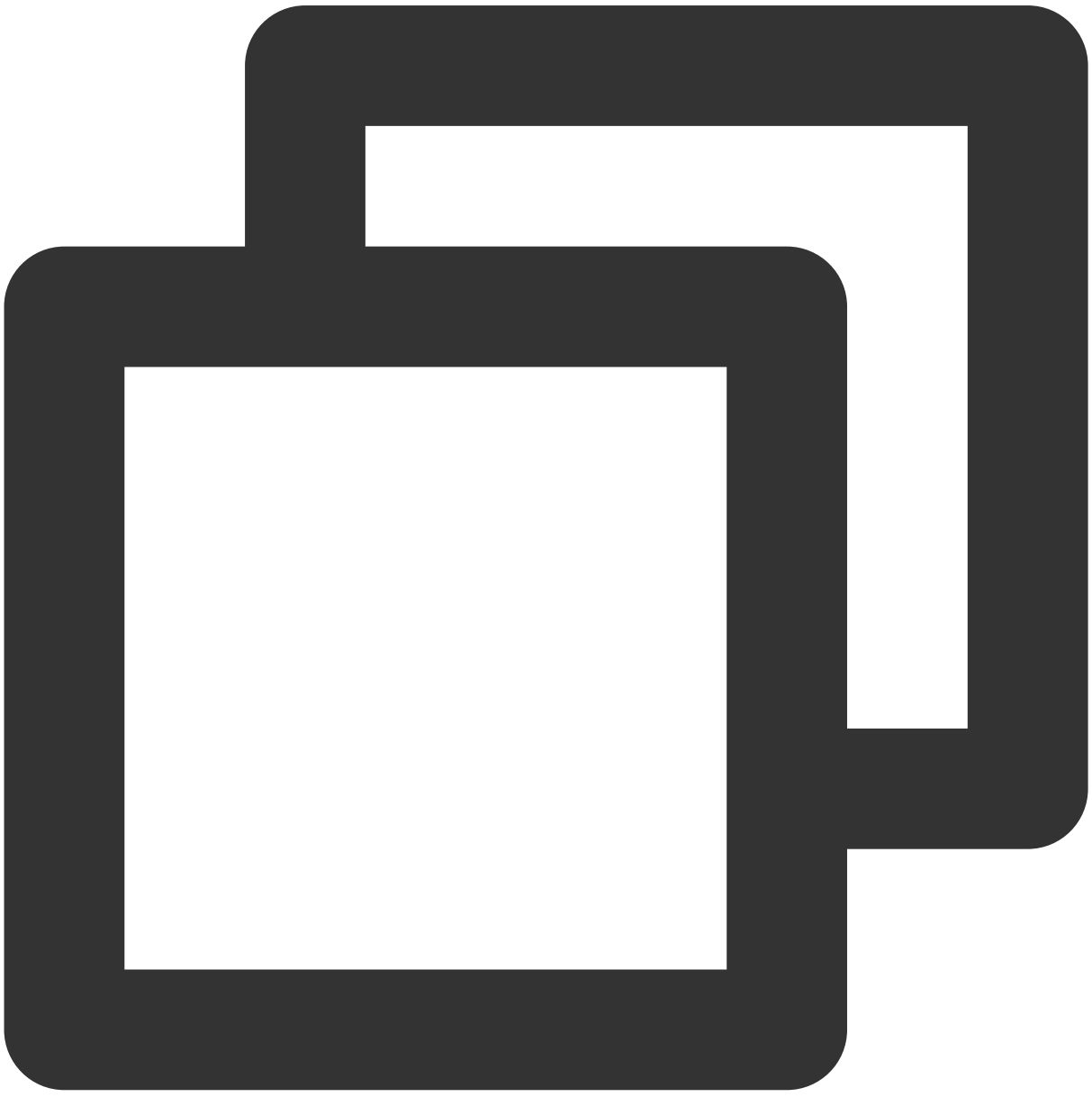
NumberOfTasksSucceeded	ProgressSummary	当前成功的操作数。	Integer
TotalNumberOfTasks	ProgressSummary	总操作数。	Integer

错误响应

最近更新时间：2023-12-21 17:44:49

本文提供以下批量处理功能的错误响应。

失败响应格式



```
HTTP/1.1 400
<Error>
    <Code>string</Code>
    <Message>string</Message>
    <Resource>string</Resource>
    <RequestId>string</RequestId>
    <TraceId>string</TraceId>
</Error>
```

常见错误代码

错误代码	描述	状态码	API
InvalidArgument	参数为空	400	CreateJob
InvalidArgument	无效参数	400	CreateJob
InvalidArgument	x-cos-appid 不能为空	400	Any
InvalidArgument	优先级有效范围为0 - 2147483647的整数	400	CreateJob
InvalidArgument	requestedJobStatus 参数状态必须为 Cancelled 或 Ready	400	UpdateJobStatus
InvalidArgument	jobStatuses 参数无效	400	ListJobs
InvalidArgument	maxResults 参数必须为正整数	400	ListJobs
InvalidArgument	nextToken 参数无效	400	ListJobs
InvalidRequest	请求不合法	400	Any
InvalidRequest	请求体为空	400	Any
InvalidRequest	未提供任务 ID	400	Any
InvalidRequest	未提供任务优先级	400	UpdateJobPriority
InvalidRequest	ClientRequestToken 参数重复	400	CreateJob
InvalidRequest	指定作业已完成	400	UpdateJobStatus
InvalidRequest	任务状态变更错误	400	UpdateJobStatus

版权所有：腾讯云计算（北京）有限责任公司

InternalError	格式化响应 xml 失败	500	Any
MalformedXML	请求格式不合法	400	Any
MalformedXML	提供的 Manifest 格式不正确	400	CreateJob
MalformedXML	提供的 Operation 格式不正确	400	CreateJob
MalformedXML	提供的 Report 格式不正确	400	CreateJob
NoSuchJob	指定作业不存在	404	DescribeJob
NoSuchJob	指定作业已完成	404	UpdateJobStatus, UpdateJobPriority
ServiceUnavailable	持久化数据错误	500	Any
ServiceUnavailable	加载持久性数据出错	500	Any
ServiceUnavailable	无法创建新任务	500	CreateJob
TooManyJobs	任务已达上限，服务器不可用	500	CreateJob

版权所有：腾讯云计算（北京）有限责任公司

数据处理接口
图片处理
基础图片处理
缩放

最近更新时间：2023-12-21 17:53:21

功能概述

对象存储通过数据万象 **imageMogr2** 接口提供图片缩放功能。处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。

该功能支持以下处理方式：

下载时处理

上传时处理

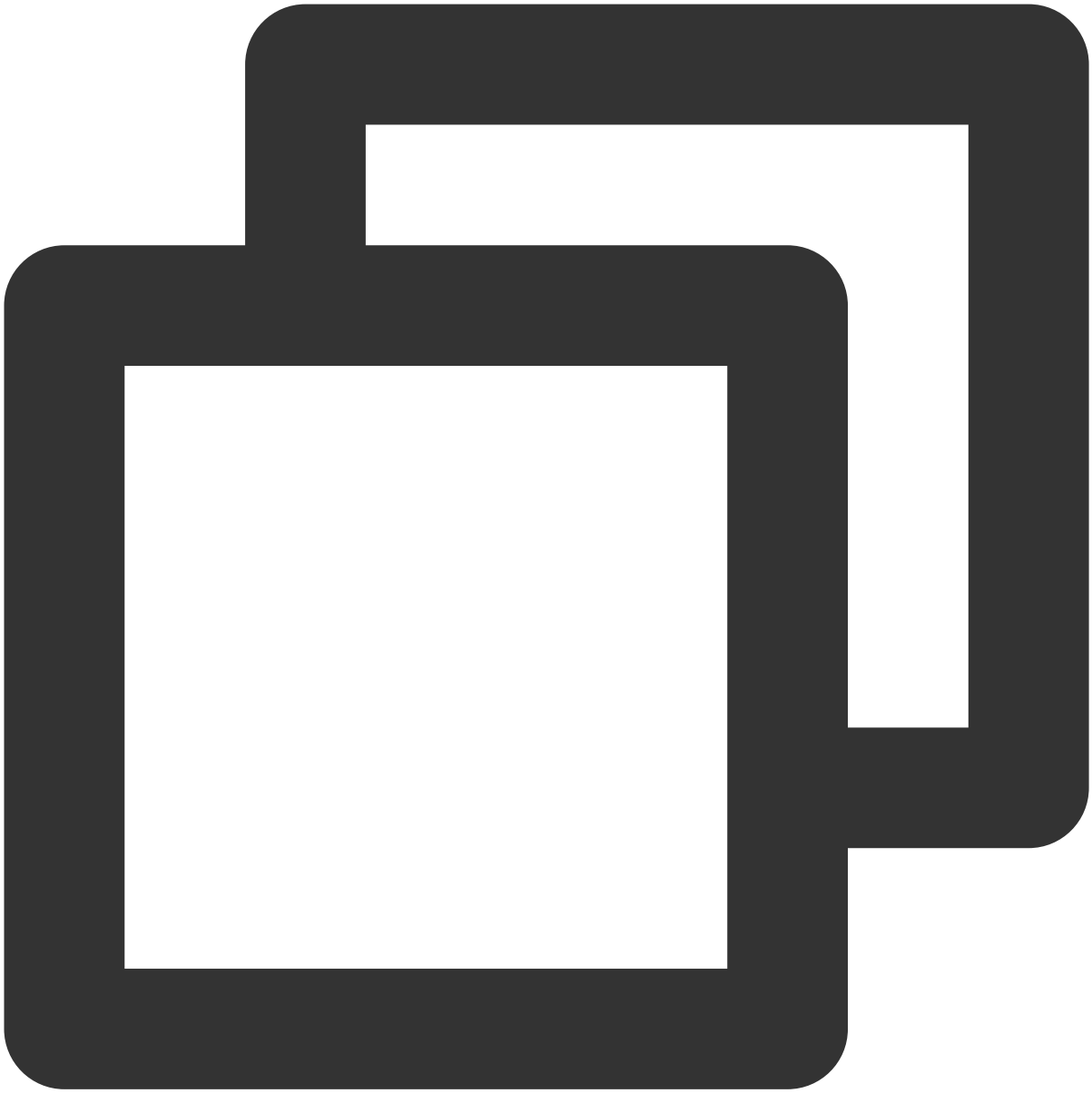
云上数据处理

注意

图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

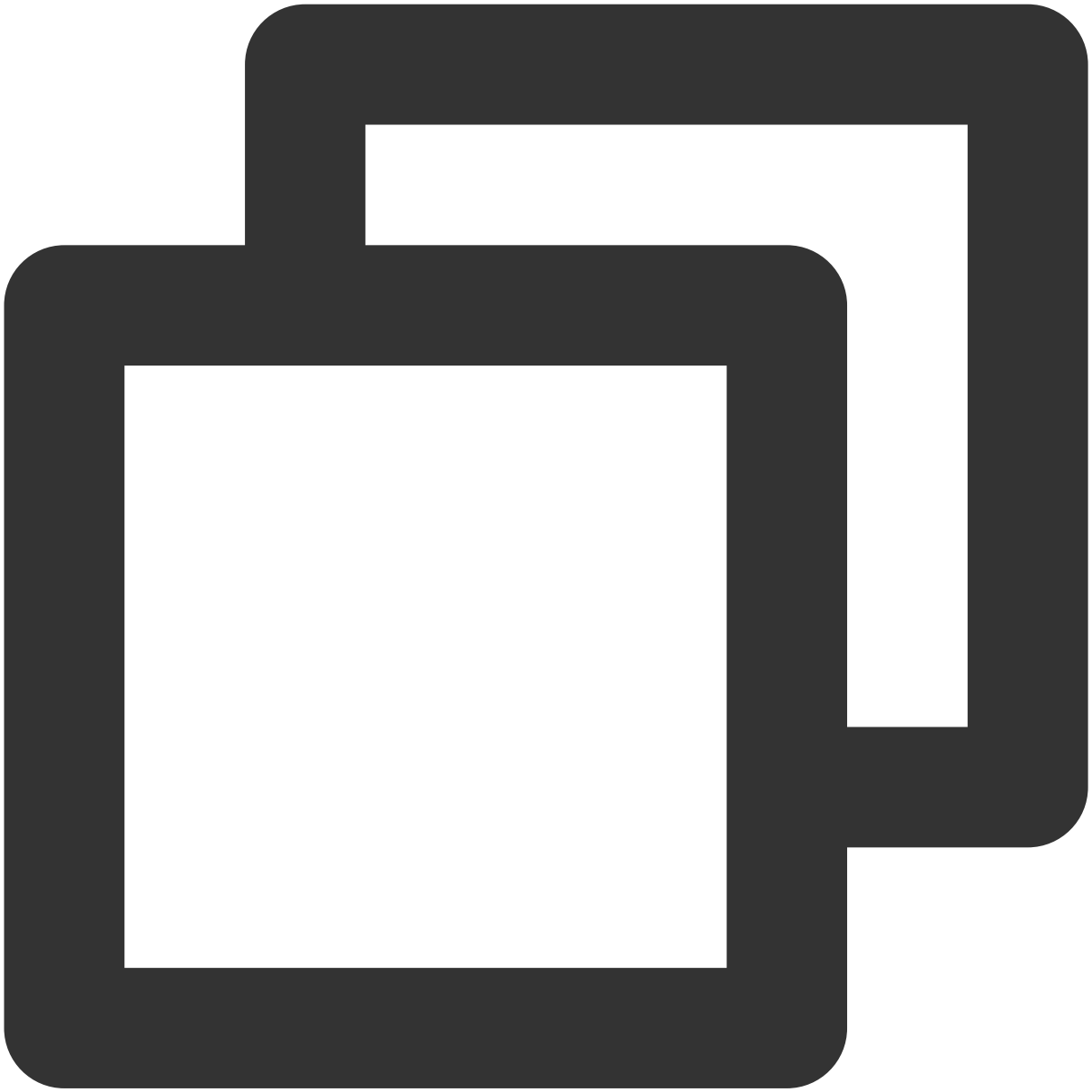
接口示例

1. 下载时处理



```
download_url?imageMogr2/thumbnail/<imageSizeAndOffsetGeometry>
```

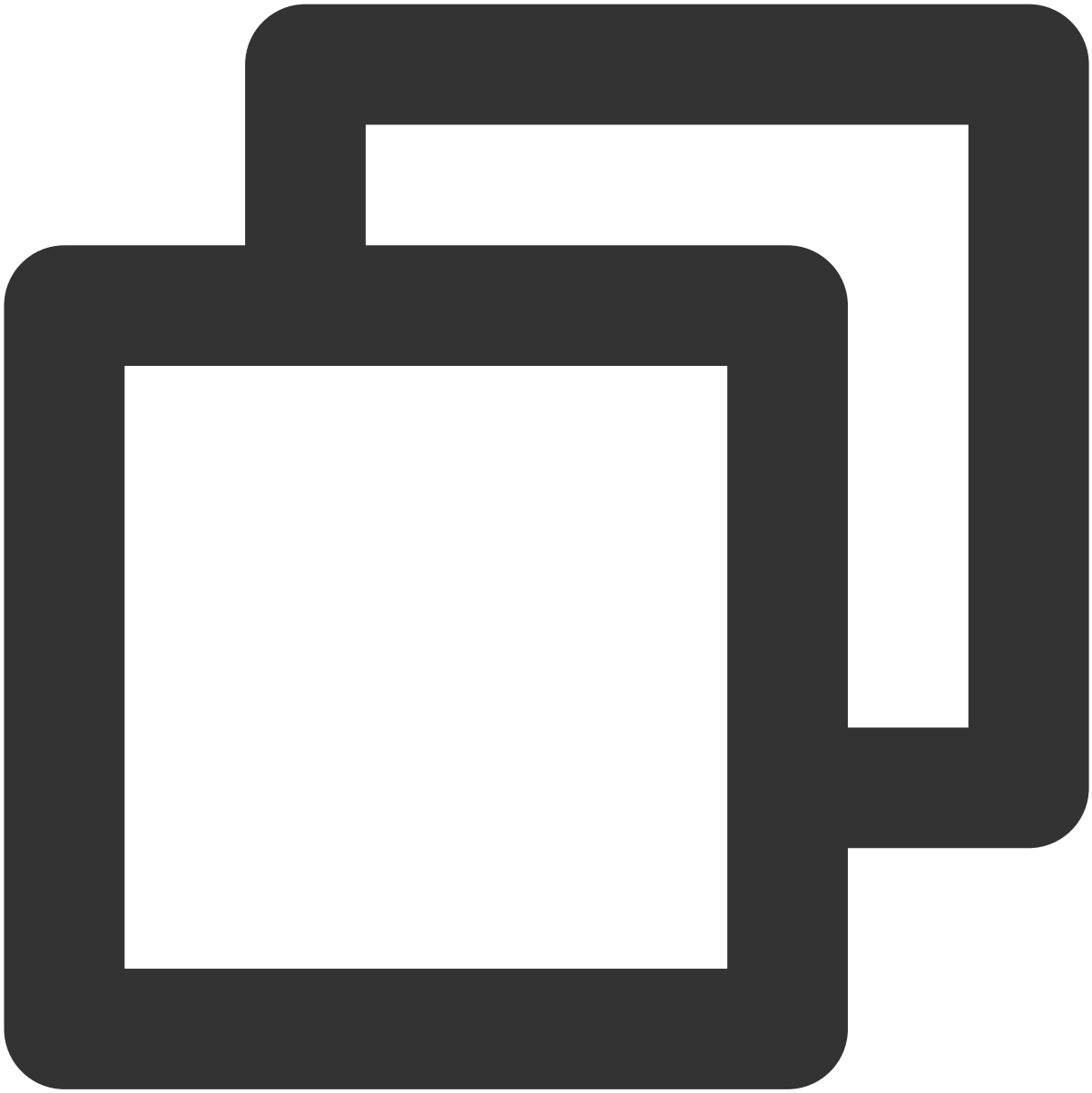
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/thumbnail/<imageSizeAndOffsetGeometry>"
  }]
```

```
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
```

```
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/thumbnail/<imageSizeAndOffsetGeometry>"
  }]
}
```

处理参数说明

操作名称：thumbnail。

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`
/thumbnail/!<Scale>p	指定图片的宽高为原图的 Scale%
/thumbnail/!<Scale>px	指定图片的宽为原图的 Scale%，高度不变
/thumbnail/!x<Scale>p	指定图片的高为原图的 Scale%，宽度不变
/thumbnail/<Width>x	指定目标图片宽度为 Width，高度等比缩放
/thumbnail/x<Height>	指定目标图片高度为 Height，宽度等比缩放
/thumbnail/<Width>x<Height>	限定缩略图的宽度和高度的最大值分别为 Width 和 Height，进行等比缩放
/thumbnail/<Width>x<Height>>	限定缩略图的宽度和高度的最大值分别为 Width 和 Height，进行等比缩小，比例值为宽缩放比和高缩放比的较小值，如果目标宽（高）都大于原图宽（高），则不变
/thumbnail/<Width>x<Height><	限定缩略图的宽度和高度的最大值分别为 Width 和 Height，进行等比放大，比例值为宽缩放比和高缩放比的较小值。如果目标宽(高)小于原图宽(高)，则不变
/thumbnail/!<Width>x<Height>r	限定缩略图的宽度和高度的最小值分别为 Width 和 Height，进行等比缩放
/thumbnail/<Width>x<Height>!	忽略原图宽高比例，指定图片宽度为 Width，高度为 Height，强行缩放图片，可能导致目标图片变形
/thumbnail/<Area>@	等比缩放图片，缩放后的图像，总像素数量不超过 Area
/pad/	将原图缩放为指定 Width 和 Height 的矩形内的最大图片，之后使用 color 参

版权所有：腾讯云计算（北京）有限责任公司

	数指定的颜色居中填充空白部分；取值0或1，0代表不使用 pad 模式，1代表使用 pad 模式
/color/	填充颜色，缺省为灰色，需设置为十六进制 RGB 格式（如 #FF0000），详情参考 RGB 编码表，需经过 URL 安全的 Base64 编码，默认值为 #3D3D3D
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

实际案例

说明

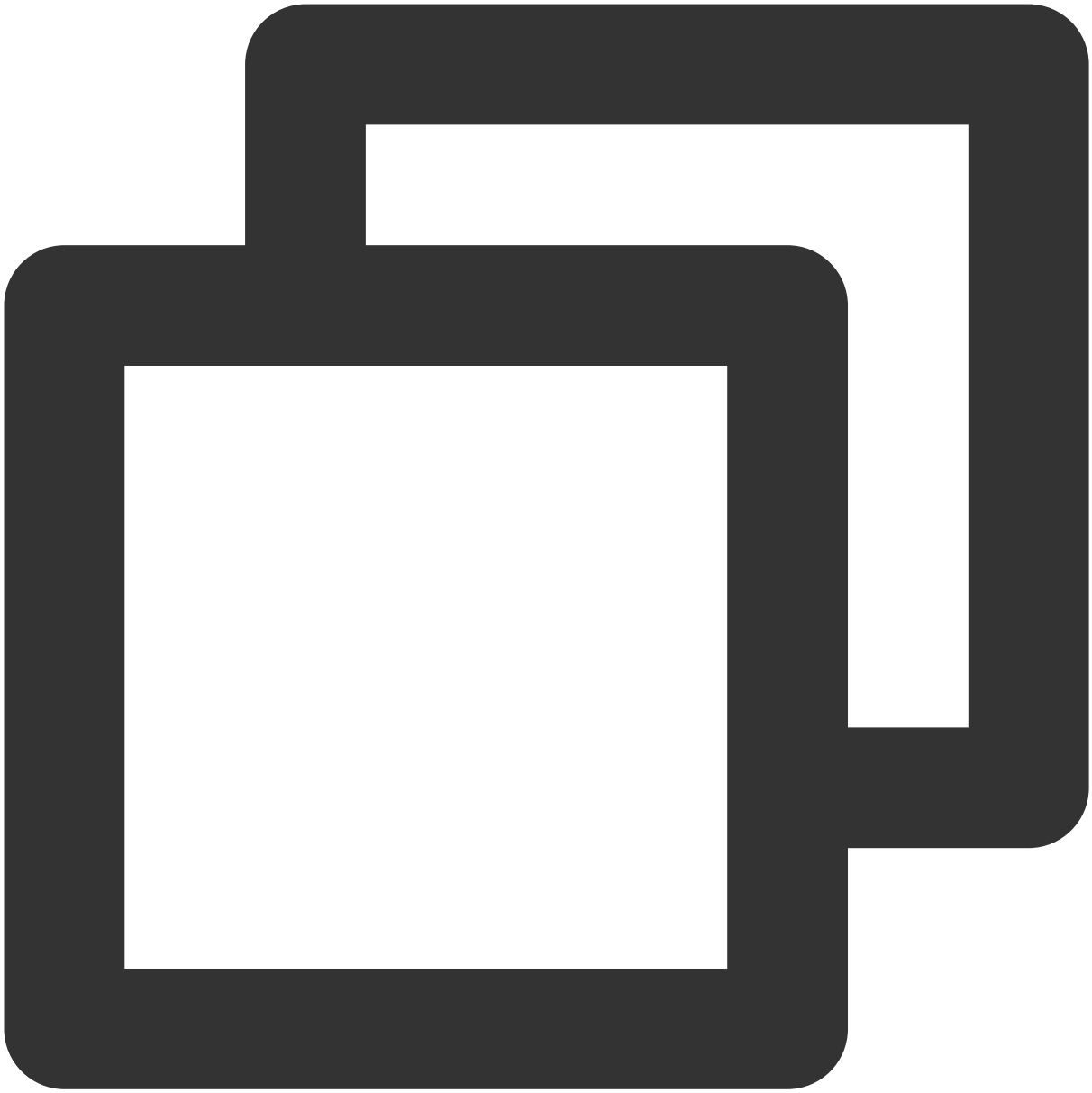
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

原图如下：



案例一：缩放宽高

假设缩放图片宽高为原图50%，示例如下：



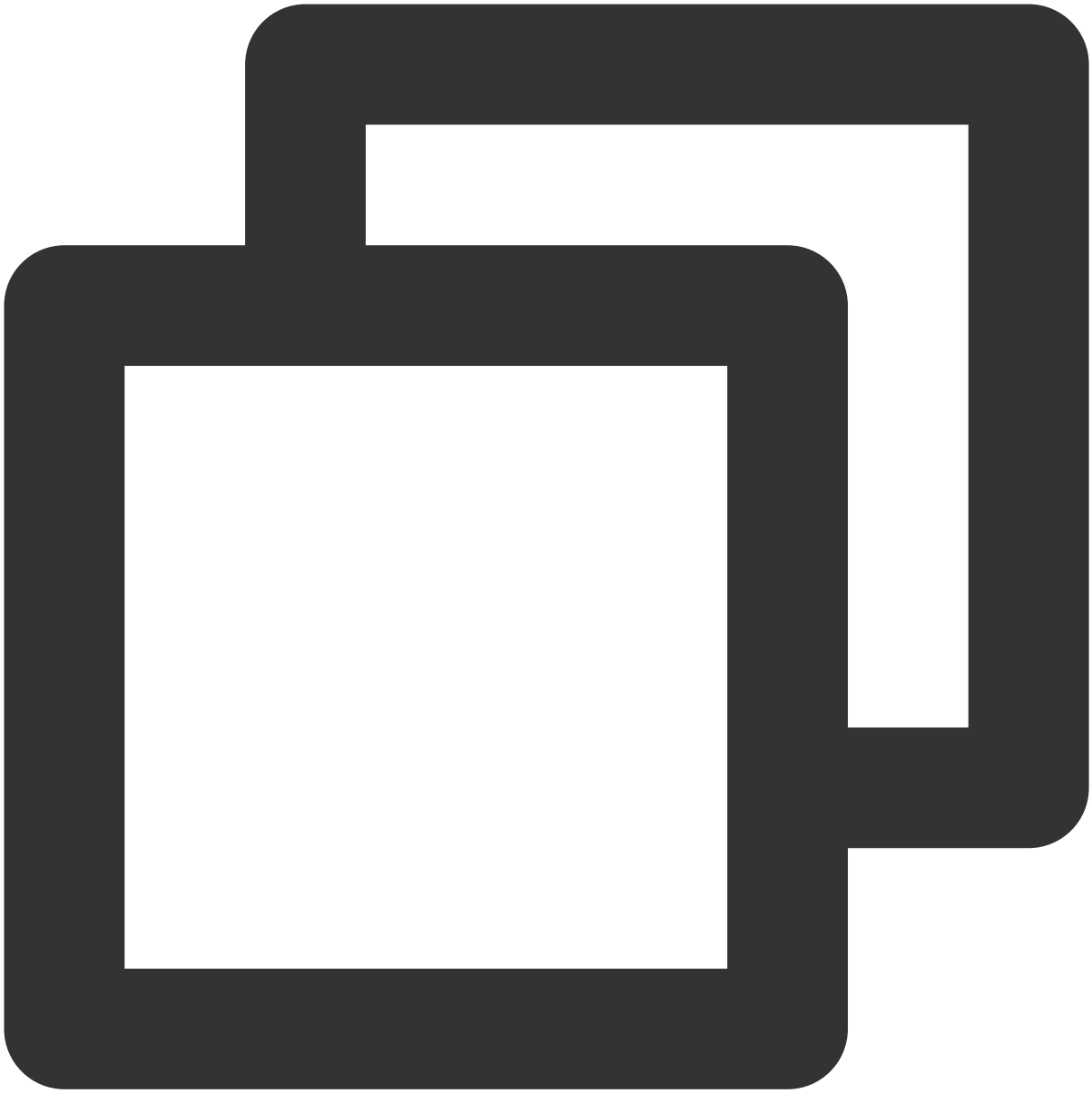
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

最终效果如下：



案例二：缩放宽度，高度不变

假设缩放指定图片宽度为原图50%，高度不变，示例如下：



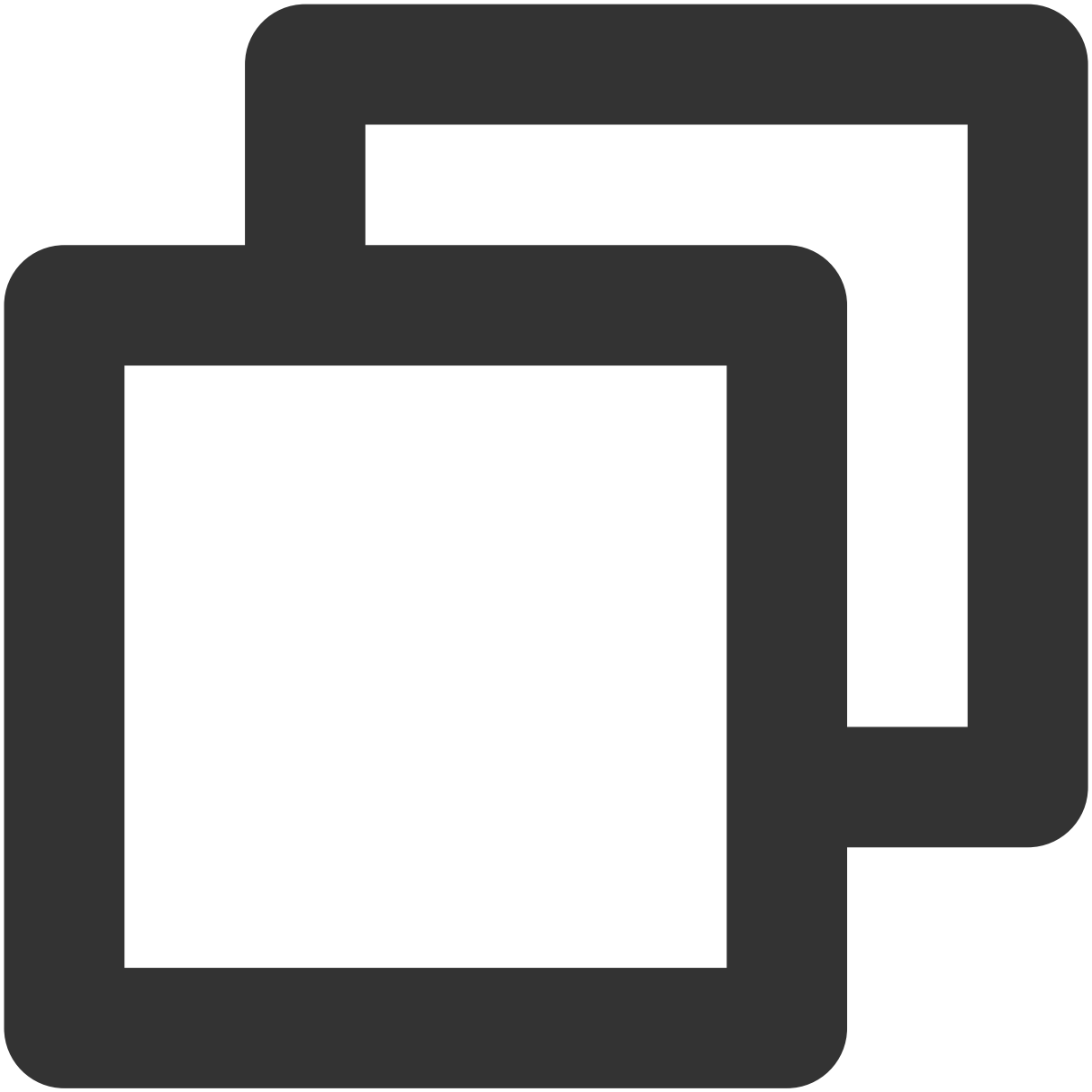
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

最终效果如下：



案例三：Pad 模式缩放

将原图缩放为600 x 600的矩形内的最大图片，并使用指定颜色填充空白部分，示例如下：



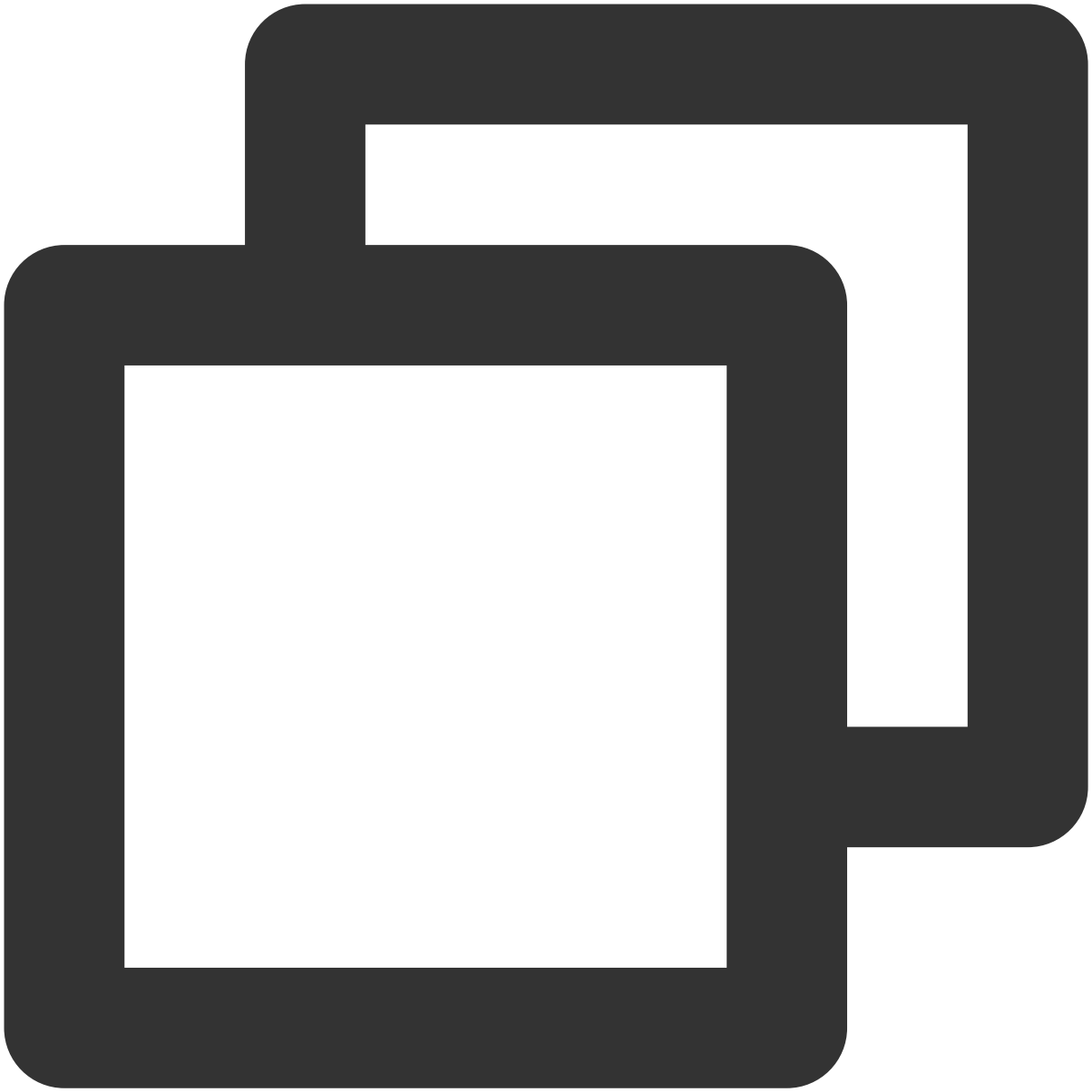
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

最终效果如下：



案例四：Pad 模式缩放并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



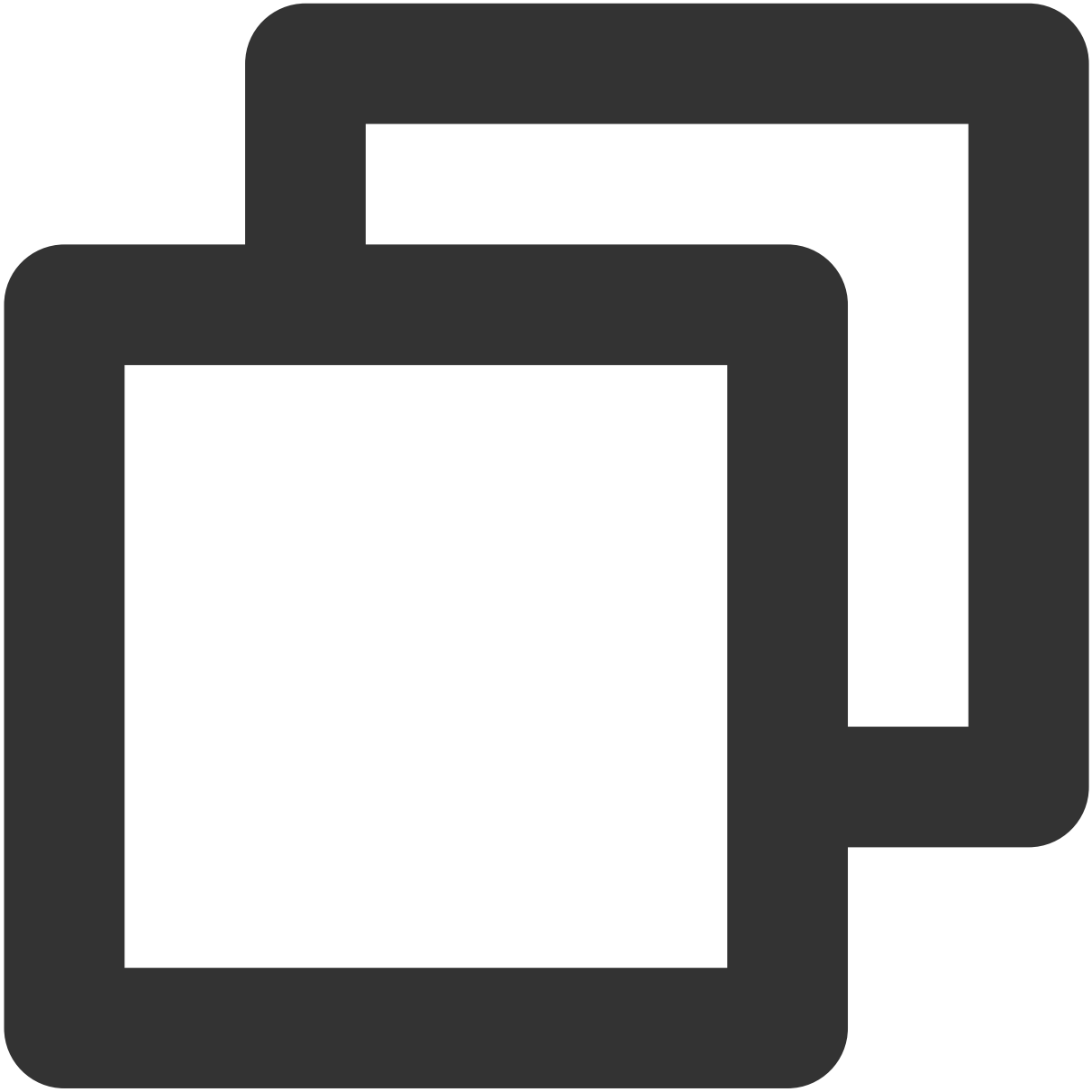
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

裁剪

最近更新时间：2023-12-21 17:55:08

功能概述

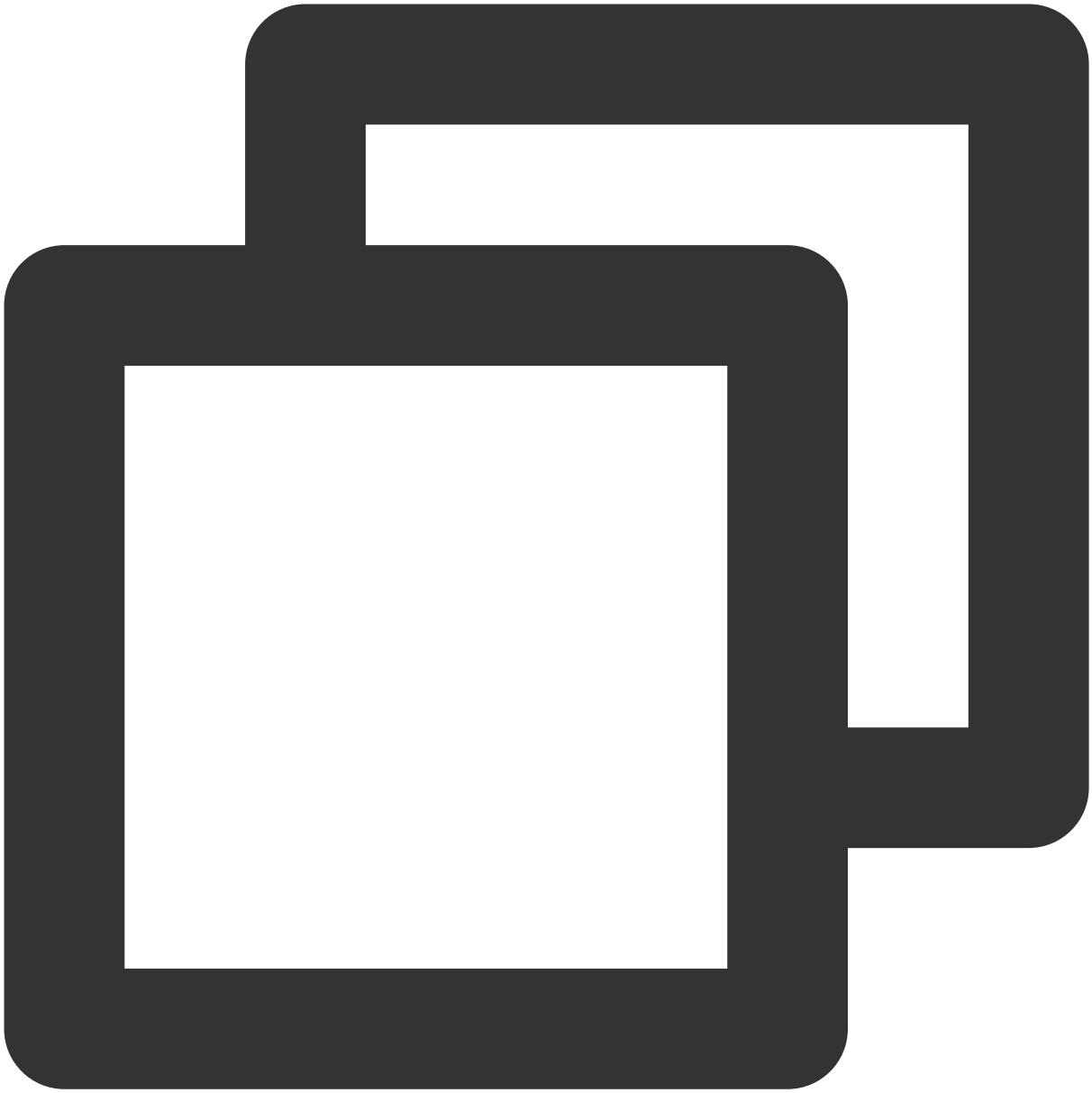
对象存储通过数据万象接口 **imageMogr2** 提供裁剪功能，包括普通裁剪、缩放裁剪、内切圆裁剪、圆角裁剪和人脸智能裁剪。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理
注意
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

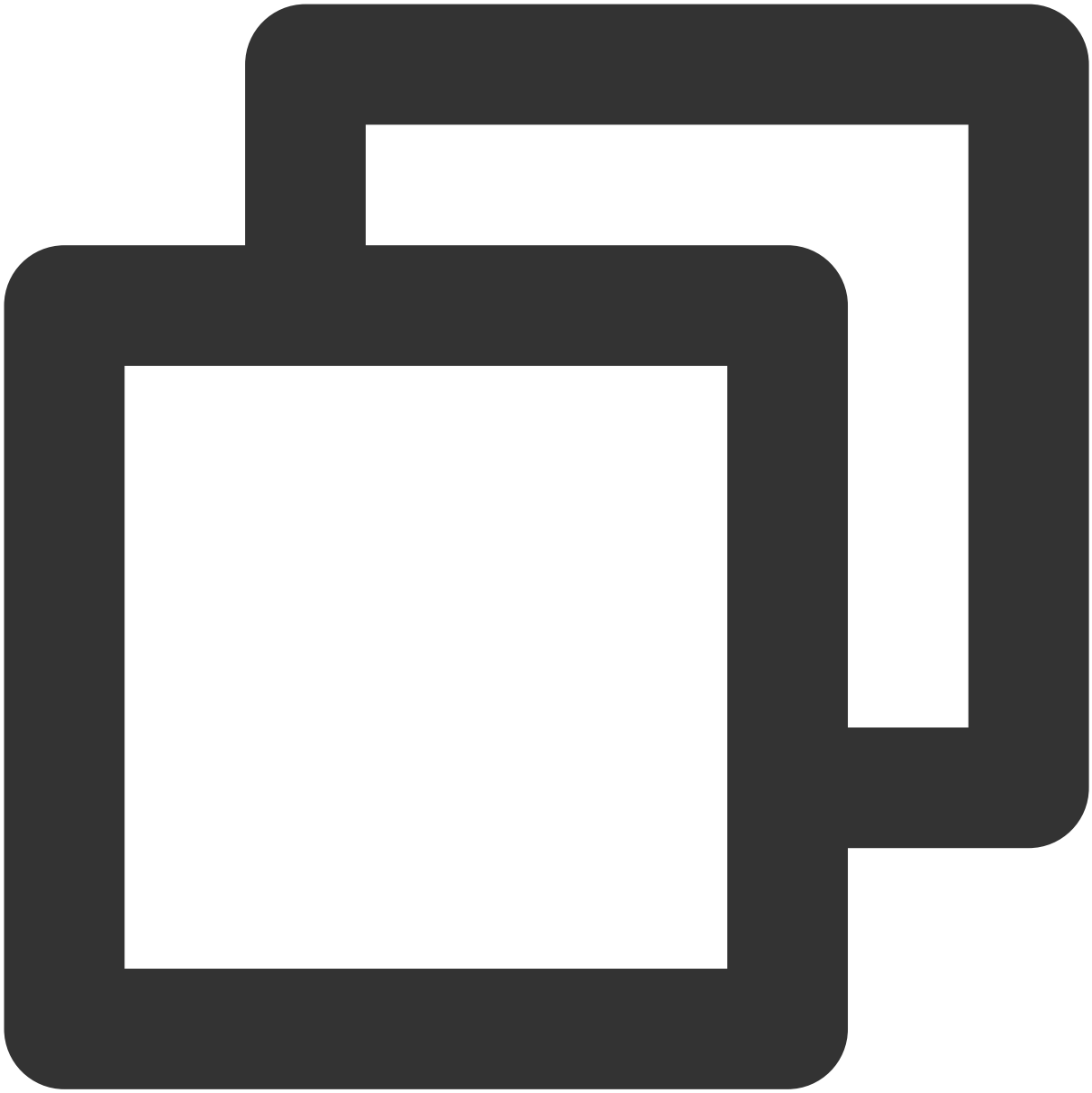


```
download_url?imageMogr2/cut/<width>x<height>x<dx>x<dy>
                        /crop/<imageSizeAndOffsetGeometry>
                        /iradius/<radius>
                        /rradius/<radius>
                        /scrop/<Width>x<Height>
```

说明

请忽略上面的空格与换行符。

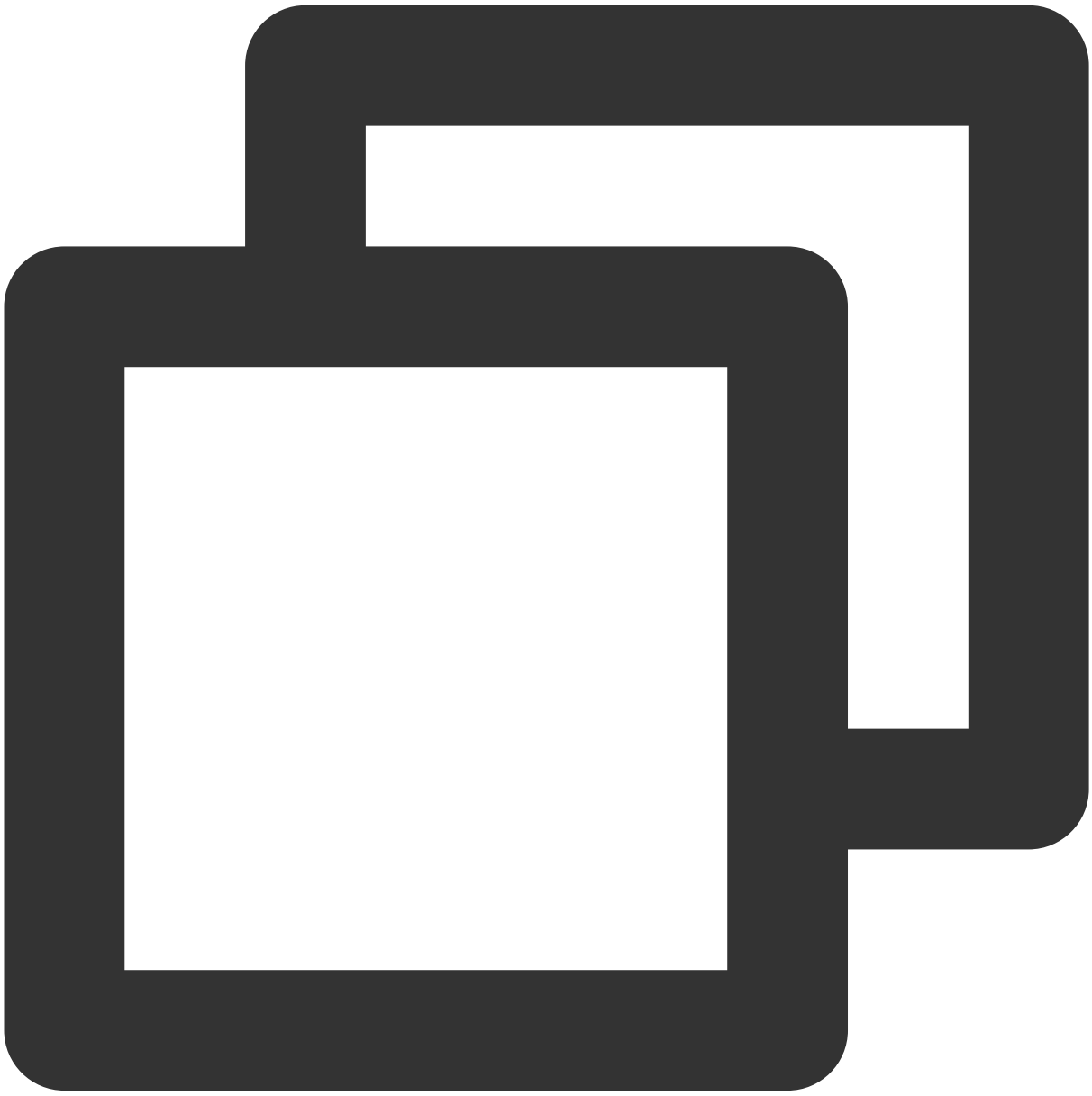
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/cut/<width>x<height>x<dx>x<dy>
                        /crop/<imageSizeAndOffsetGeometry>
```

```
                        /iradius/<radius>
                        /rradius/<radius>
                        /scrop/<Width>x<Height>"
    }]
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
```

```
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/cut/<width>x<height>x<dx>x<dy>
                        /crop/<imageSizeAndOffsetGeometry>
                        /iradius/<radius>
                        /rradius/<radius>
                        /scrop/<Width>x<Height>"
  }]
}
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

普通裁剪参数说明

操作名称：cut。

参数	含义
<width>	指定目标图片的宽为 width
<height>	指定目标图片的高为 height
<dx>	相对于图片左上顶点水平向右偏移 dx
<dy>	相对于图片左上顶点水平向下偏移 dy

说明

参数取值范围应大于0，小于原图宽高。

缩放裁剪参数说明

操作名称：crop。

参数	含义
/crop/<Width>x	指定目标图片宽度为 Width，高度不变。Width 取值范围应大于0，小于原图宽度
/crop/x<Height>	指定目标图片高度为 Height，宽度不变。Height 取值范围应大于0，小于原图高度
/crop/<Width>x<Height>	指定目标图片宽度为 Width，高度为 Height 。Width 和 Height 取值范围都应大于0，小于原图宽度/高度

在进行缩放裁剪操作时，您也可以使用 gravity 参数指定操作的起点位置，详见下文中 缩放裁剪操作示例。

内切圆裁剪参数说明

操作名称：iradius。

参数	含义
/iradius/<radius>	内切圆裁剪功能，radius 是内切圆的半径，取值范围为大于0且小于原图最小边一半的整数。内切圆的圆心为图片的中心。图片格式为 gif 时，不支持该参数。

圆角裁剪参数说明

操作名称：rradius。

参数	含义
/rradius/<radius>	圆角裁剪功能，radius 为图片圆角边缘的半径，取值范围为大于0且小于原图最小边一半的整数。圆角与原图边缘相切。图片格式为 gif 时，不支持该参数。

人脸智能裁剪参数说明

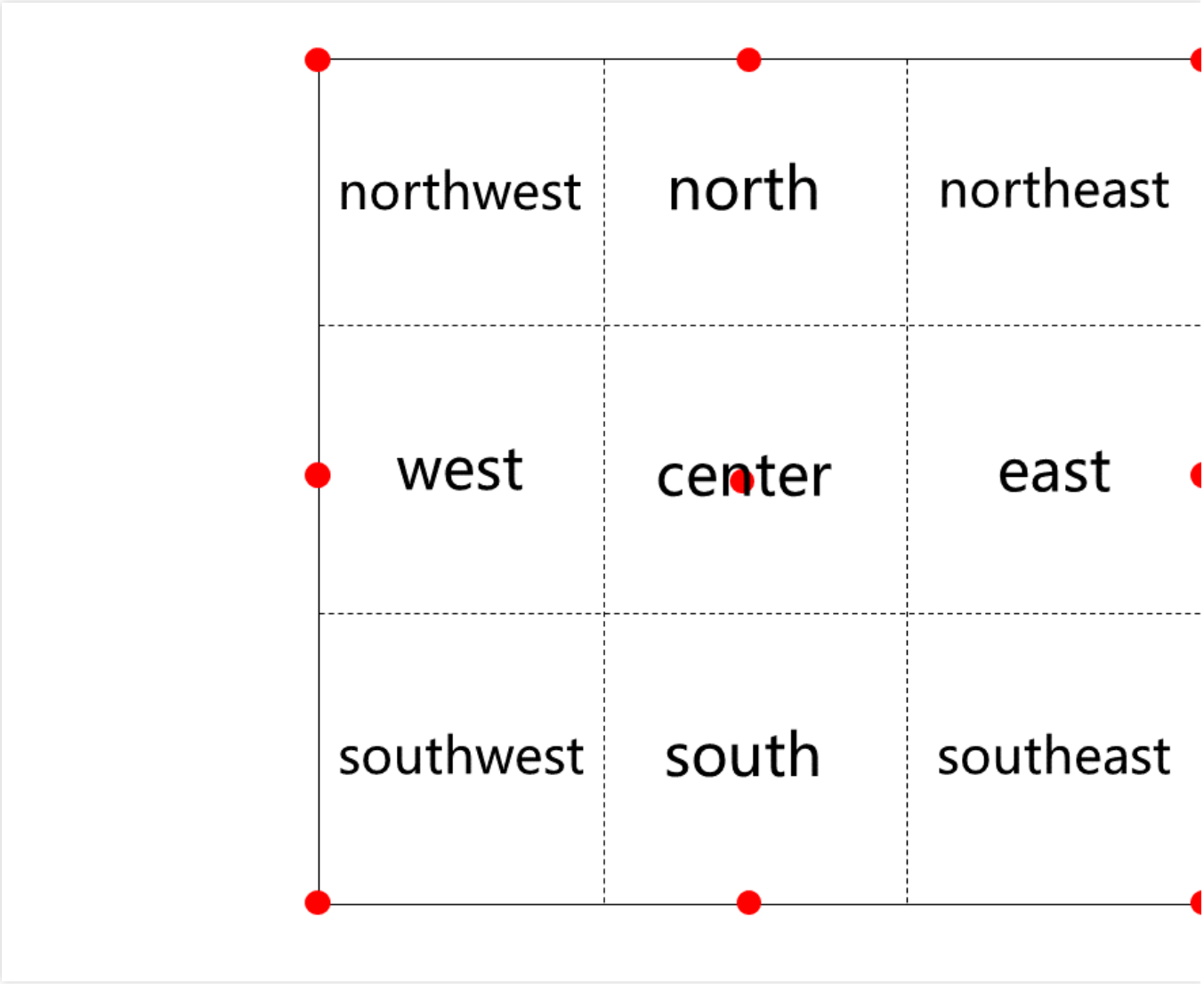
操作名称：scrop。

参数	含义
/scrop/<Width>x<Height>	基于图片中的人脸位置进行缩放裁剪。目标图片的宽度为 Width、高度为 Height。

九宫格方位图

版权所有：腾讯云计算（北京）有限责任公司

九宫格方位图可为图片的多种操作提供位置参考。红点为各区域位置的原点（通过 gravity 参数选定各区域后位移操作会以相应远点为参照）。



注意

当 gravity 参数设置为 center 时，dx、dy 参数无效。

当 gravity 参数设置为 north 或 south 时，dx 参数无效（水印会水平居中）。

当 gravity 参数设置为 west 或 east 时，dy 参数无效（水印会垂直居中）。

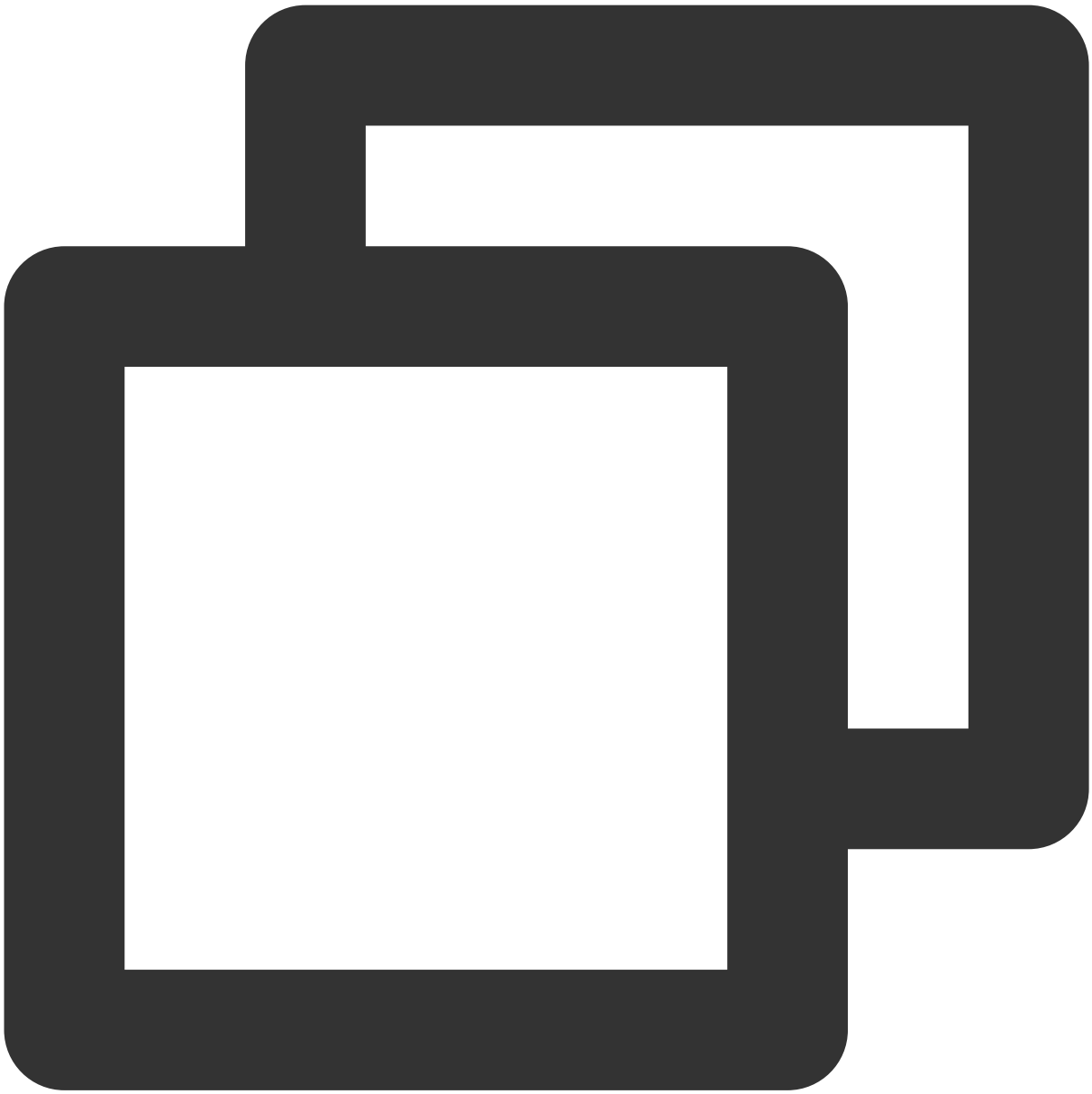
实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：普通裁剪

相对于图片左上顶点水平向右平移100像素，垂直向下平移10像素，指定目标图片大小为600×600进行裁剪：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/cut/
```

原图如下：



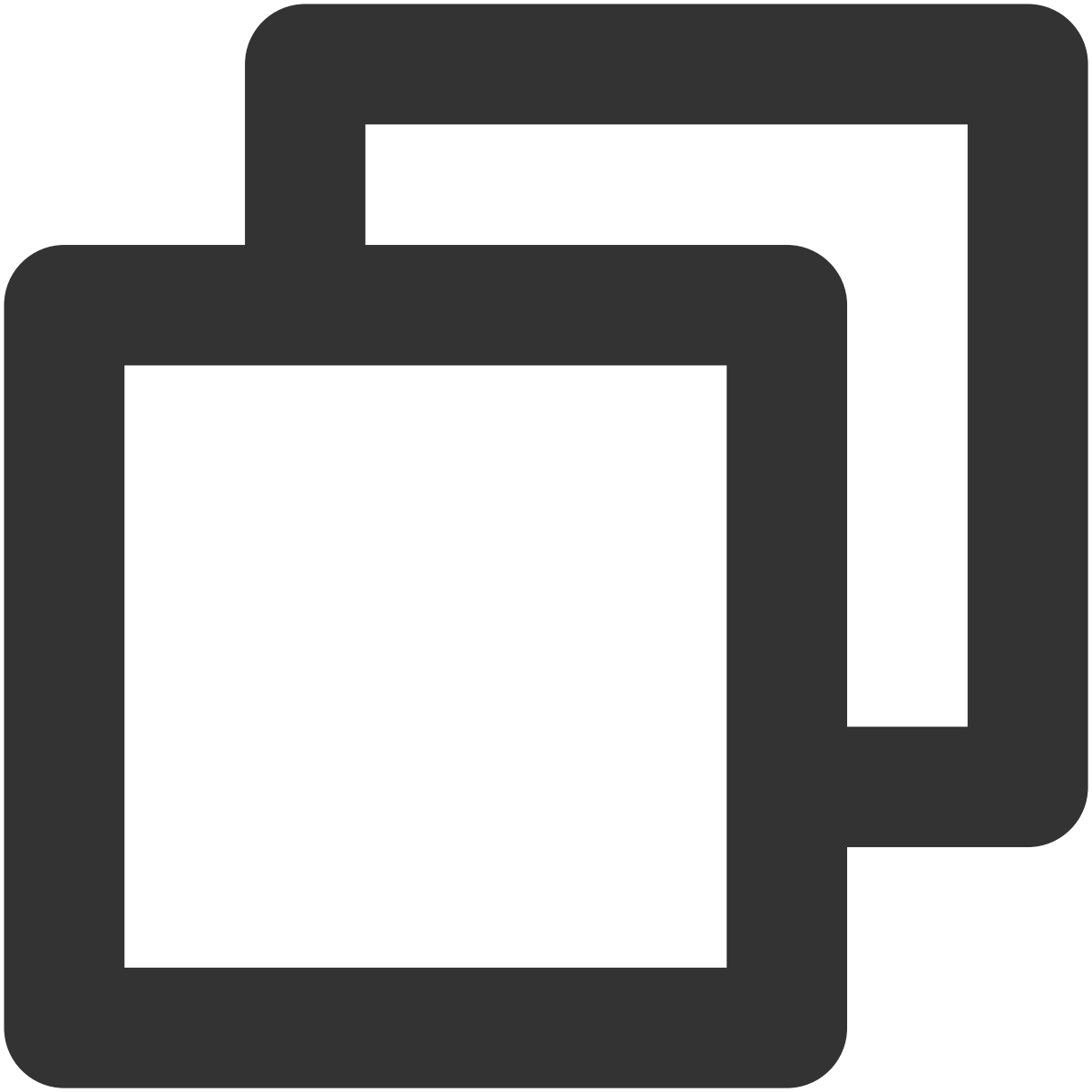
最终效果如下：

版权所有：腾讯云计算（北京）有限责任公司



案例二：缩放裁剪

假设以中心点 center 为参考点，缩放裁剪至300×400，示例如下：



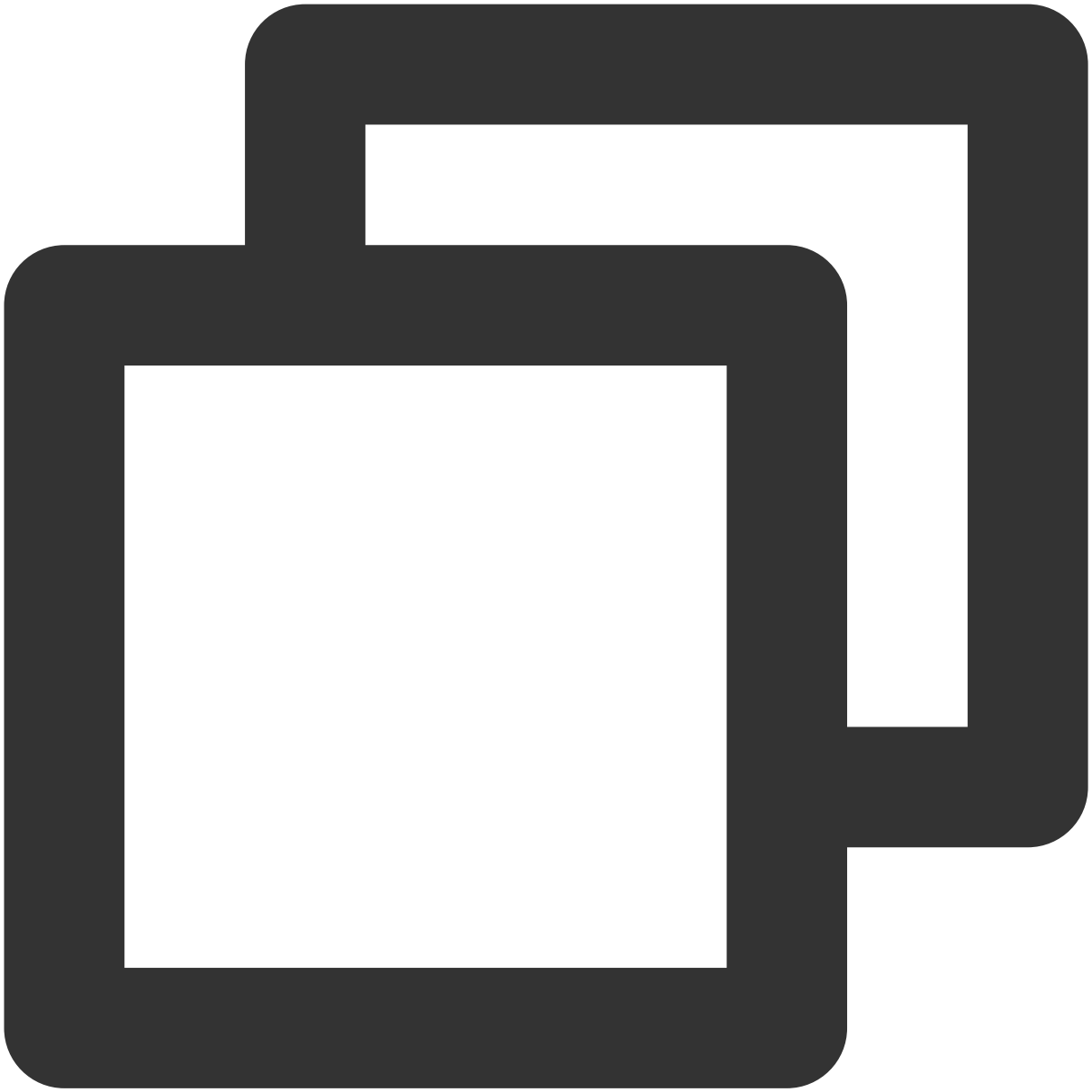
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/crop
```

最终效果如下：



案例三：内切圆裁剪

取半径为200，进行内切圆裁剪：



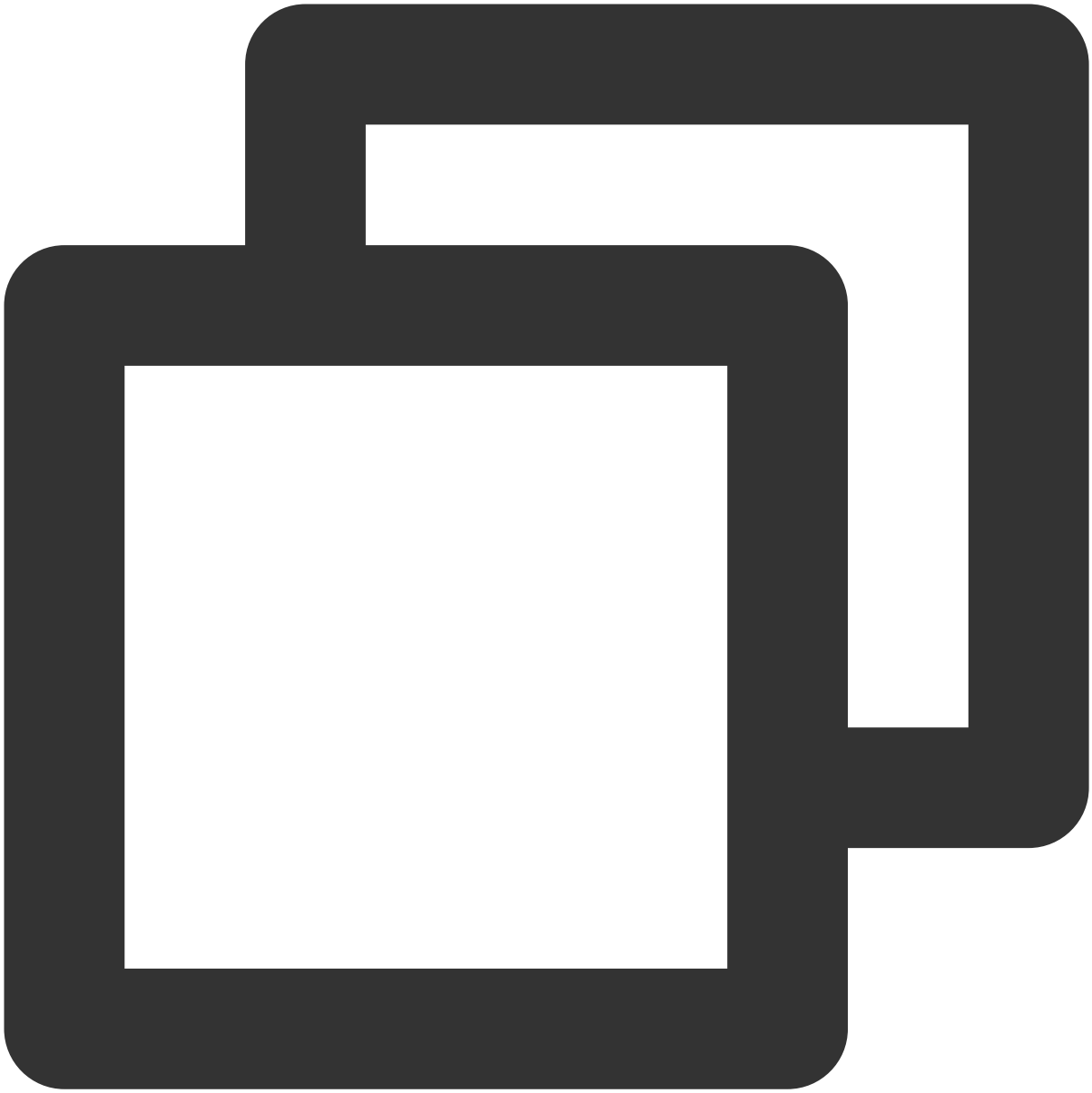
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/irad
```

最终效果如下：



案例四：圆角裁剪

取半径为100，进行圆角裁剪：



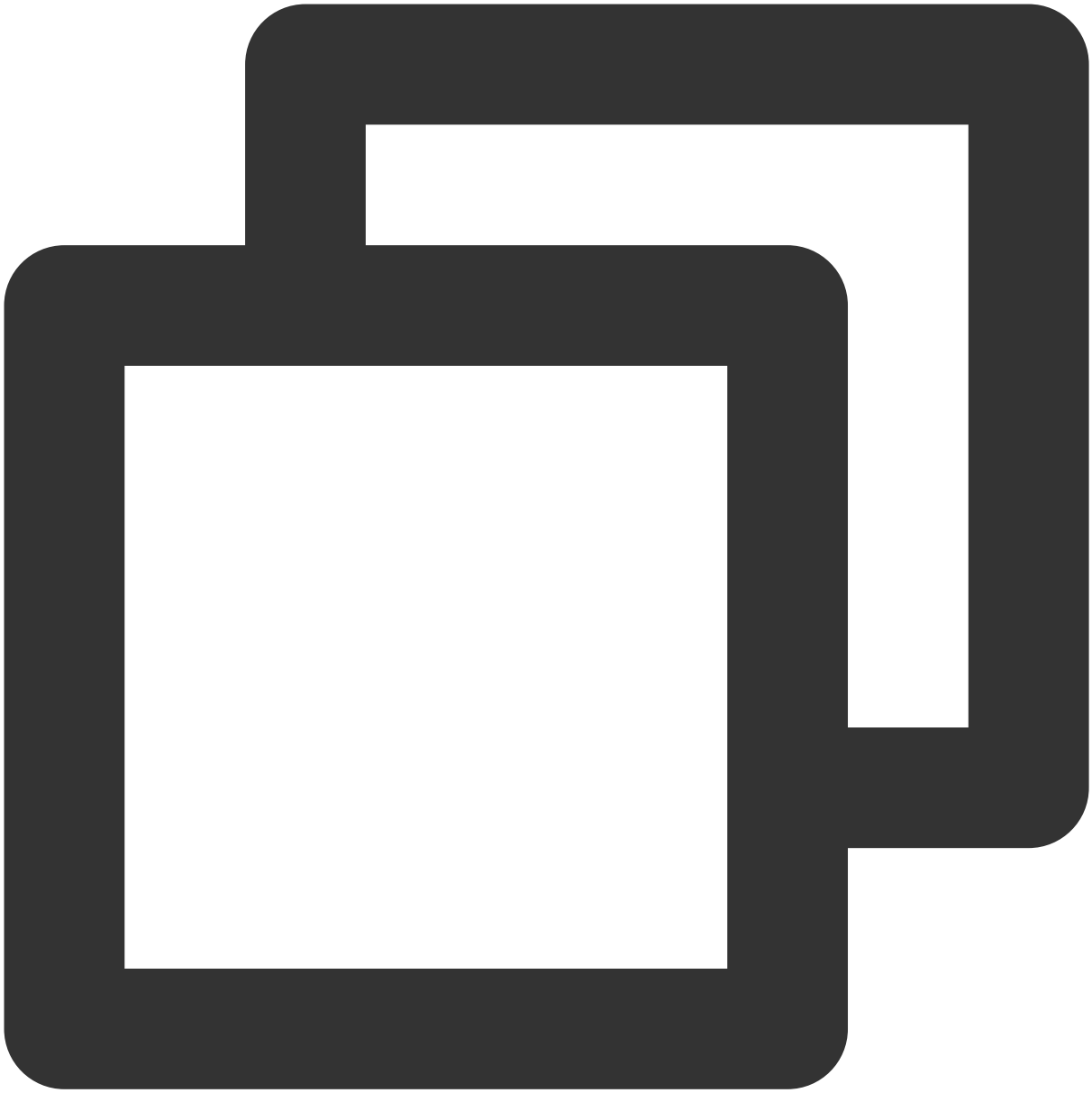
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/rrad
```

最终效果如下：



案例五：人脸智能裁剪

基于图片中的人脸位置进行缩放裁剪，目标图片大小为100×600，示例如下：



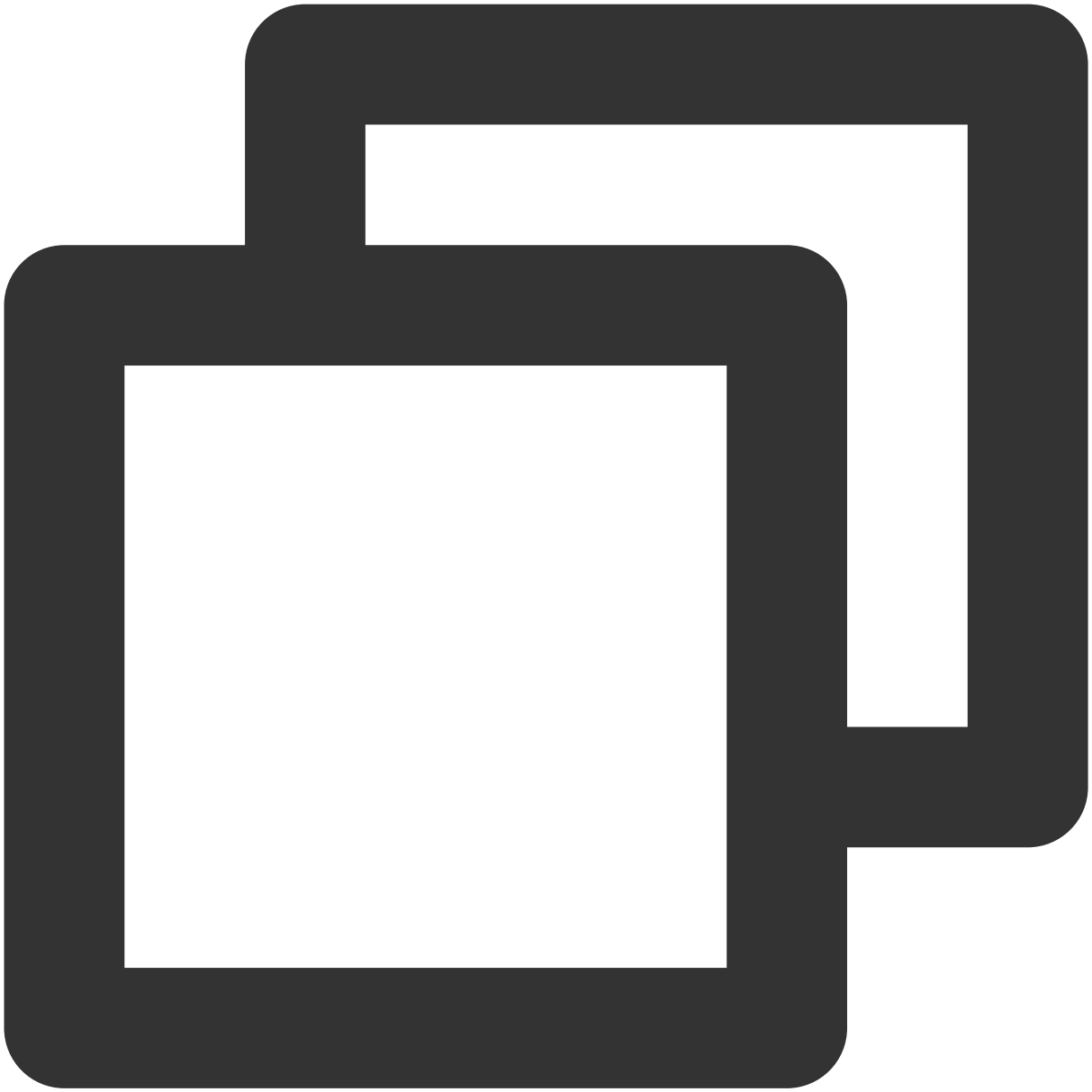
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/scro
```

说明

如果图片中没有识别到人脸，则返回原图。

案例六：人脸智能裁剪并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



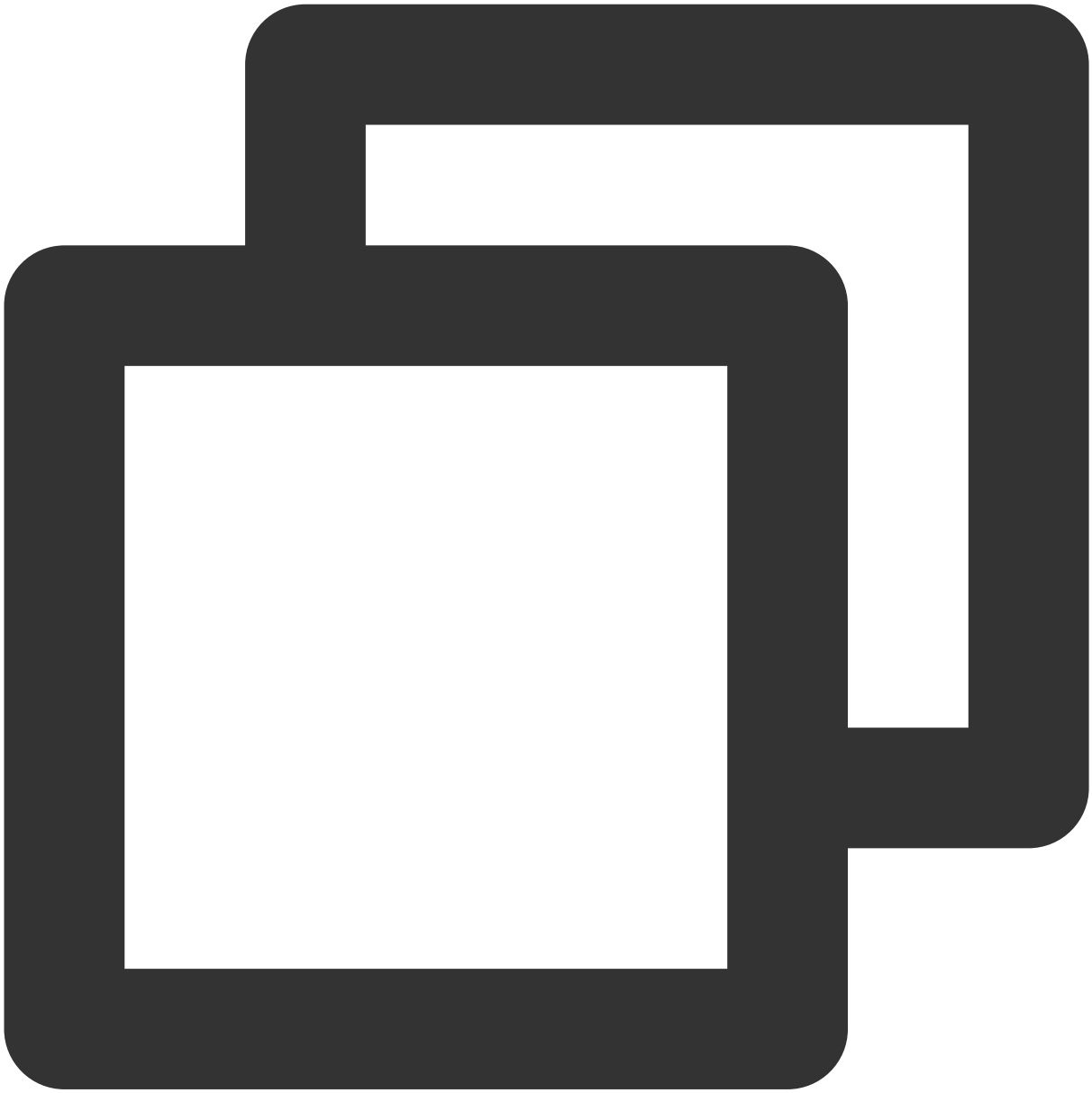
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

旋转

最近更新时间：2023-12-21 17:54:24

功能概述

对象存储通过数据万象 **imageMogr2** 接口提供旋转功能，包括普通旋转和自适应旋转。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理
注意
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

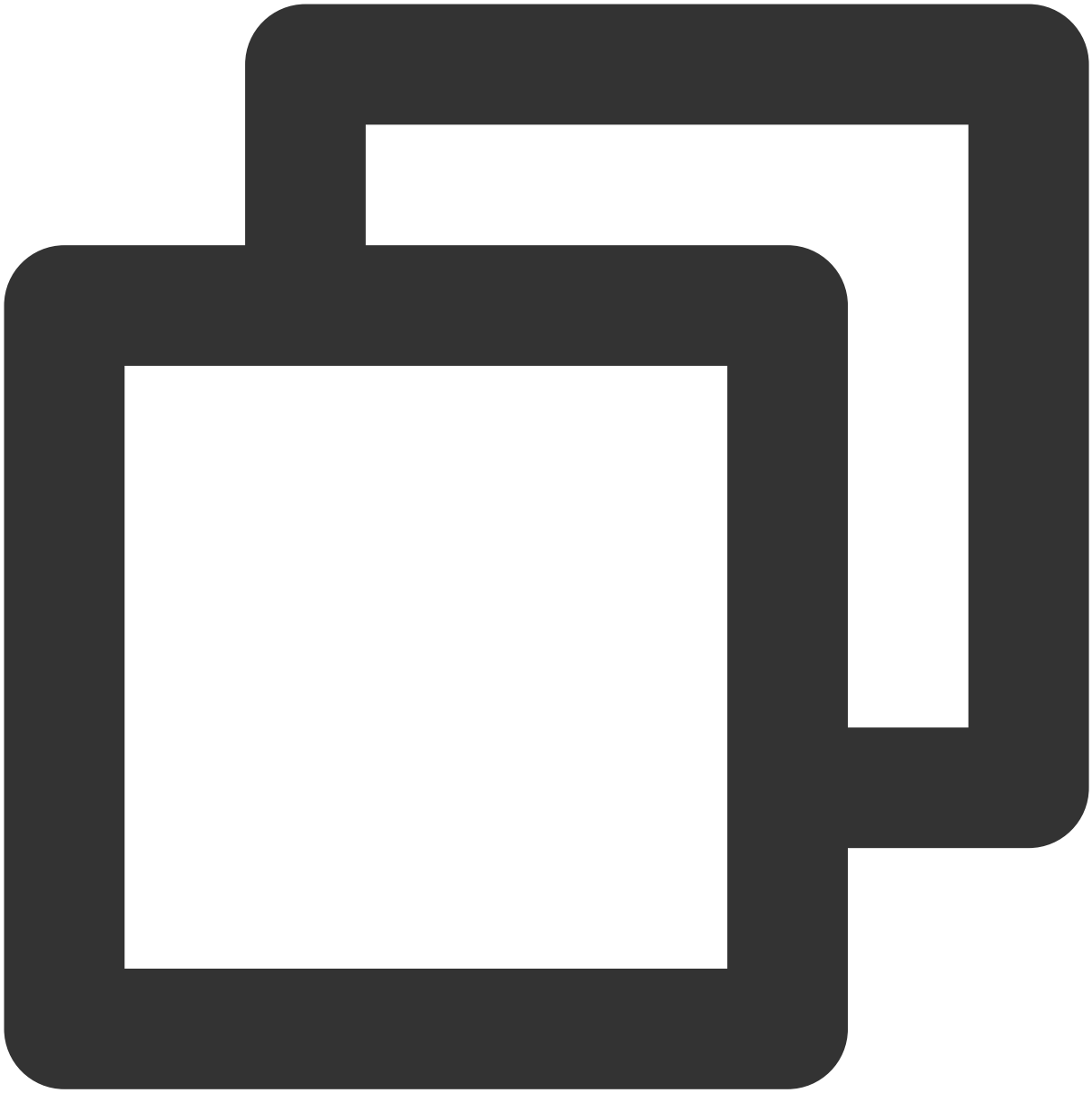
限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口形式

1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司

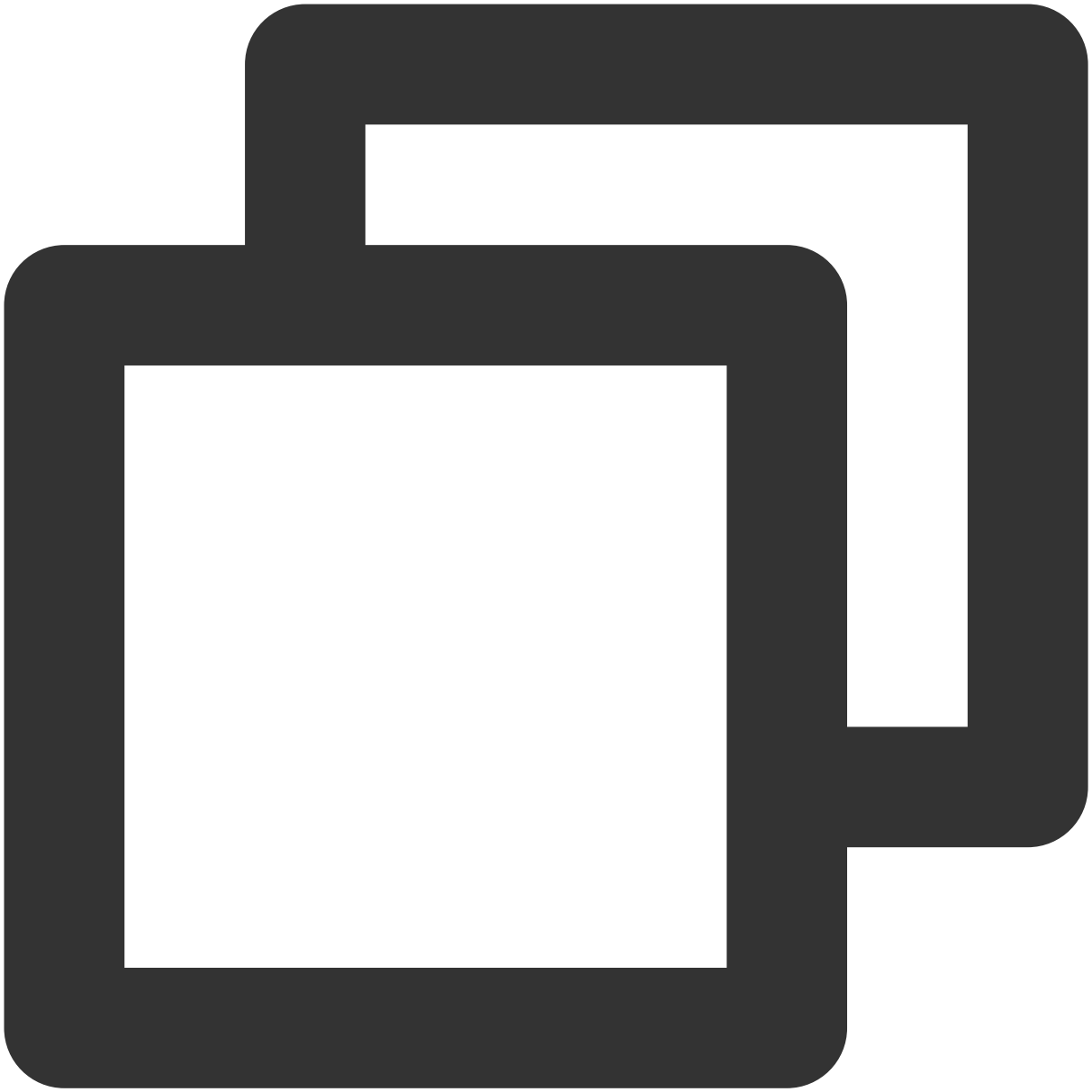


```
download_url?imageMogr2/rotate/<rotateDegree>
                        /auto-orient
```

注意

请忽略上面的空格与换行符。

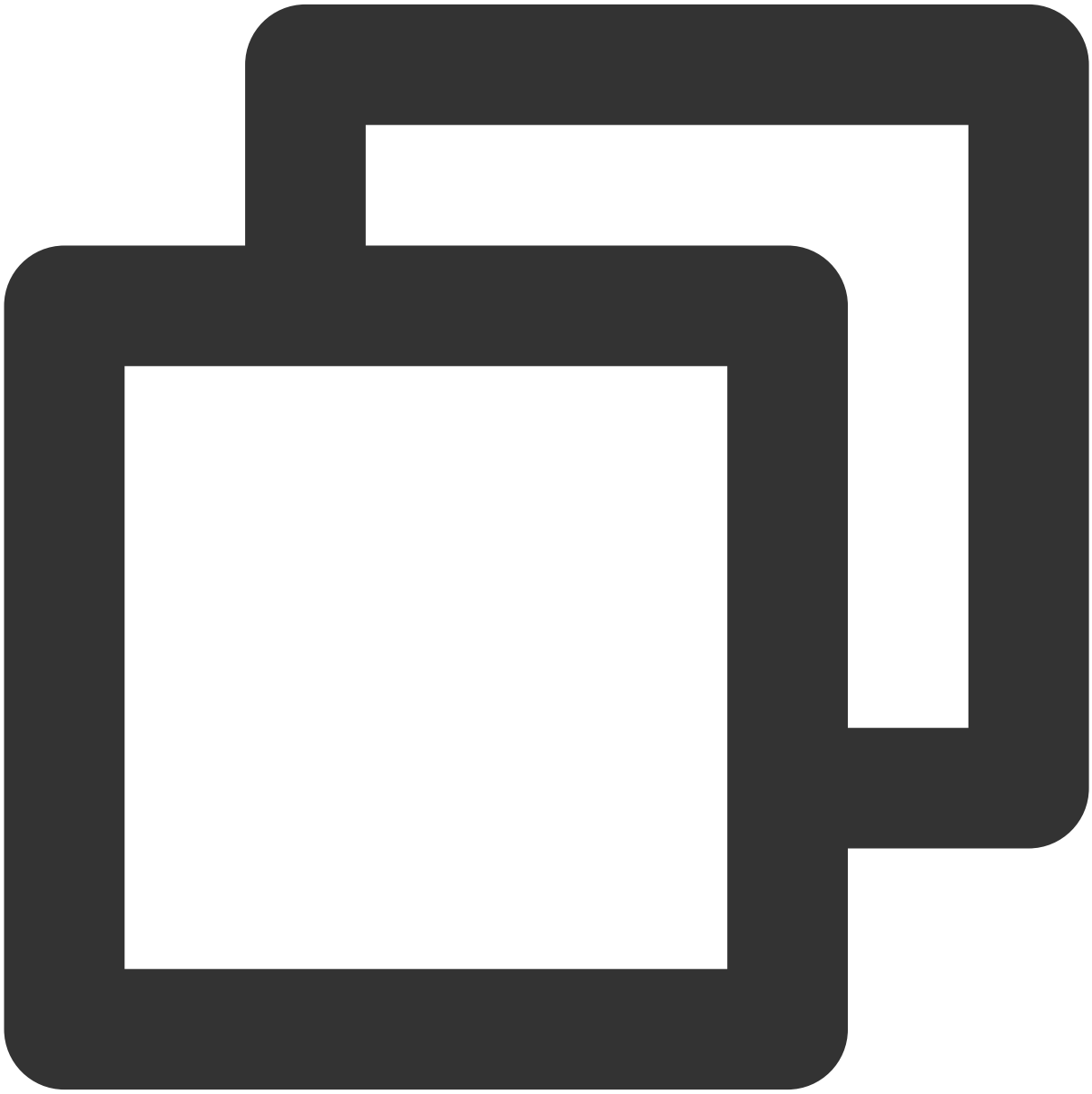
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/rotate/<rotateDegree>
                      /auto-orient"
```

```
  }]
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
```

```
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/rotate/<rotateDegree>
                        /auto-orient"
  }]
}
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/rotate/<rotateDegree>	普通旋转：图片顺时针旋转角度，取值范围0 - 360，默认不旋转。
/auto-orient	自适应旋转：根据原图 EXIF 信息将图片自适应旋转回正。
/flip/<flip>	镜像翻转：flip 值为 vertical 表示垂直翻转，horizontal 表示水平翻转
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

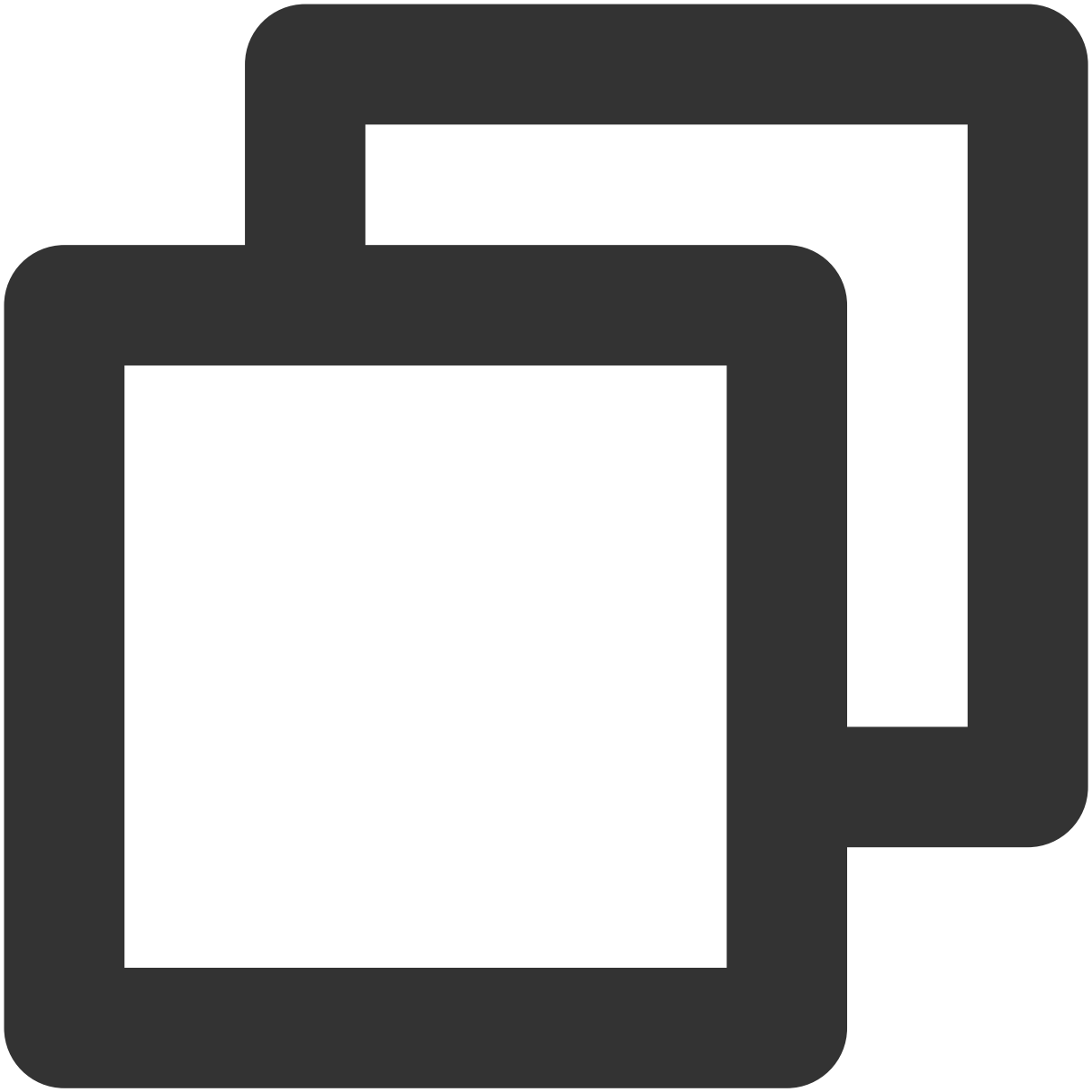
实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：普通旋转

顺时针旋转90度，示例如下：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/rota
```

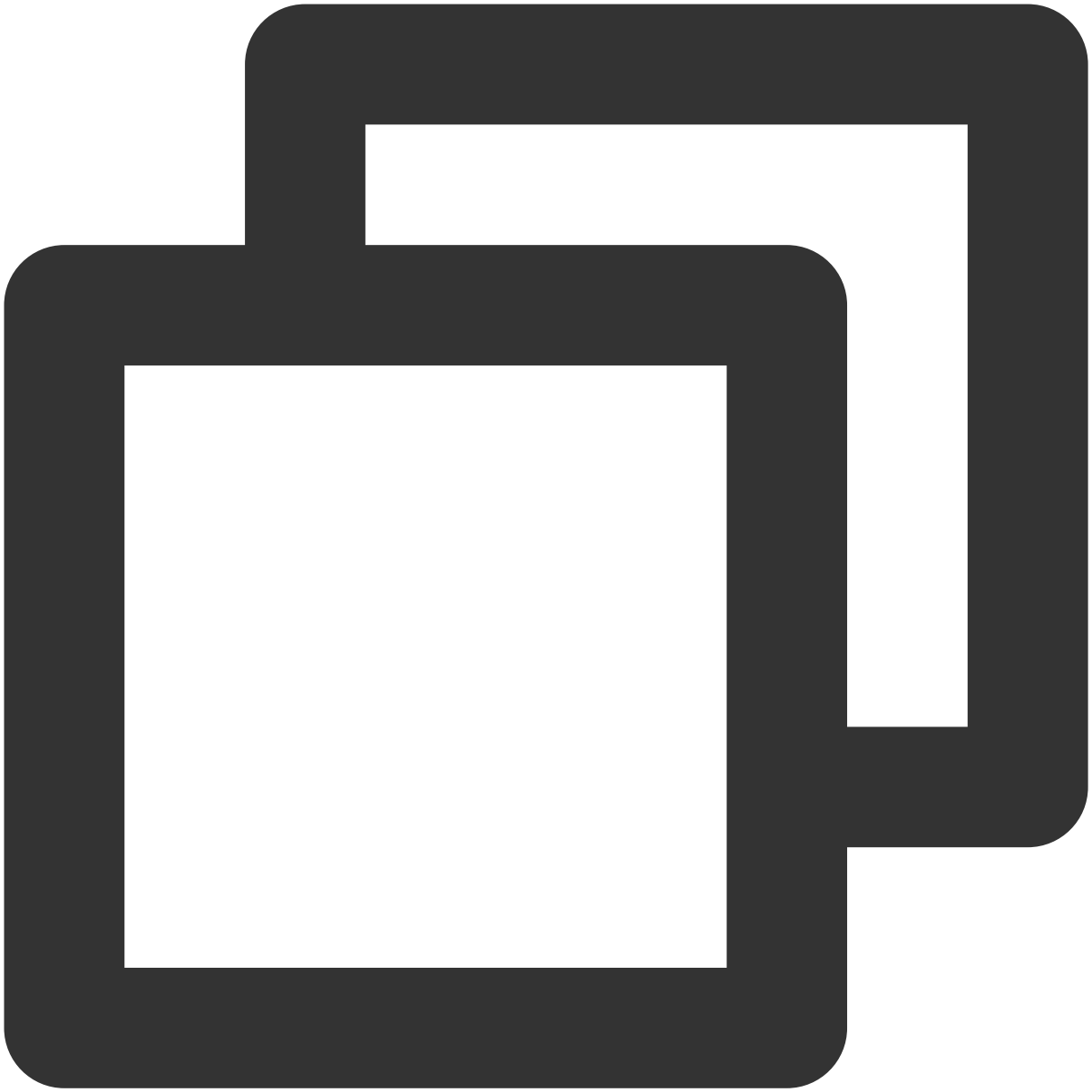
最终效果如下：





案例二：普通旋转并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



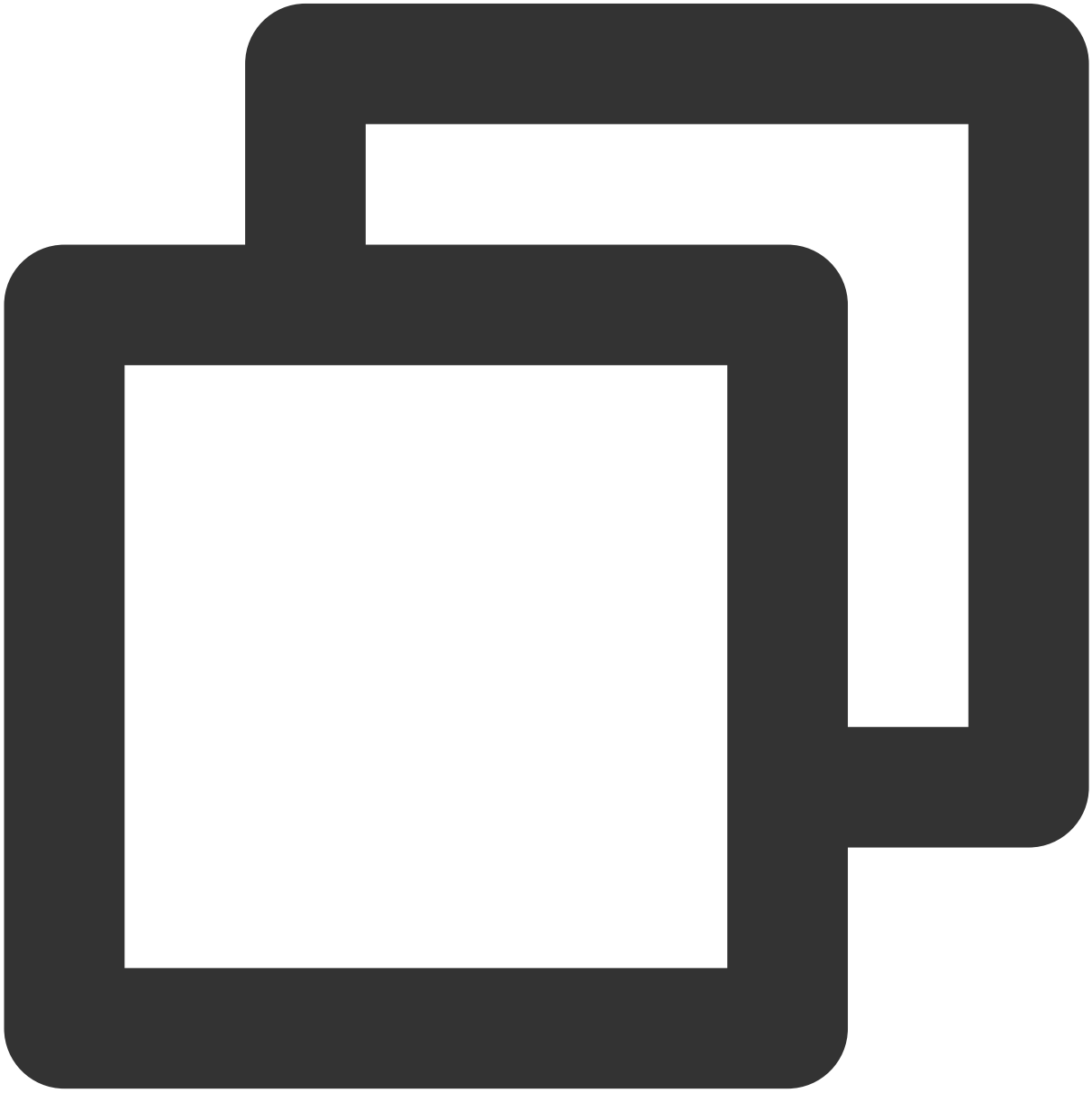
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了防止未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

格式转换

最近更新时间：2023-12-21 17:55:31

功能概述

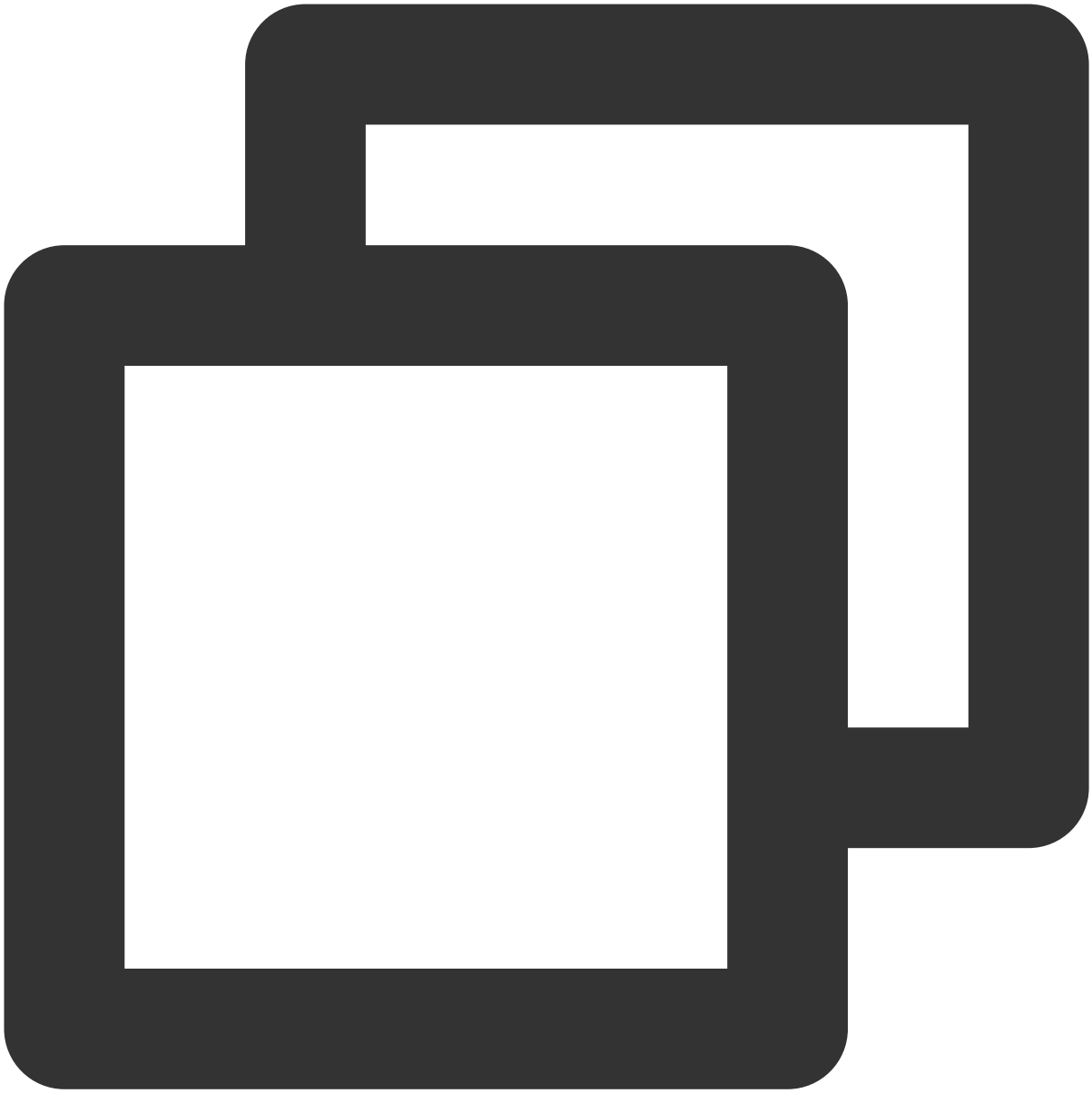
对象存储通过数据万象接口 **imageMogr2** 提供格式转换、gif 格式优化、渐进显示功能。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理
注意
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

限制说明

gif、webp、tpg 等动图，支持互相转换。
jpg、png、bmp、tpg、heif 等静态图片，支持互相转换。

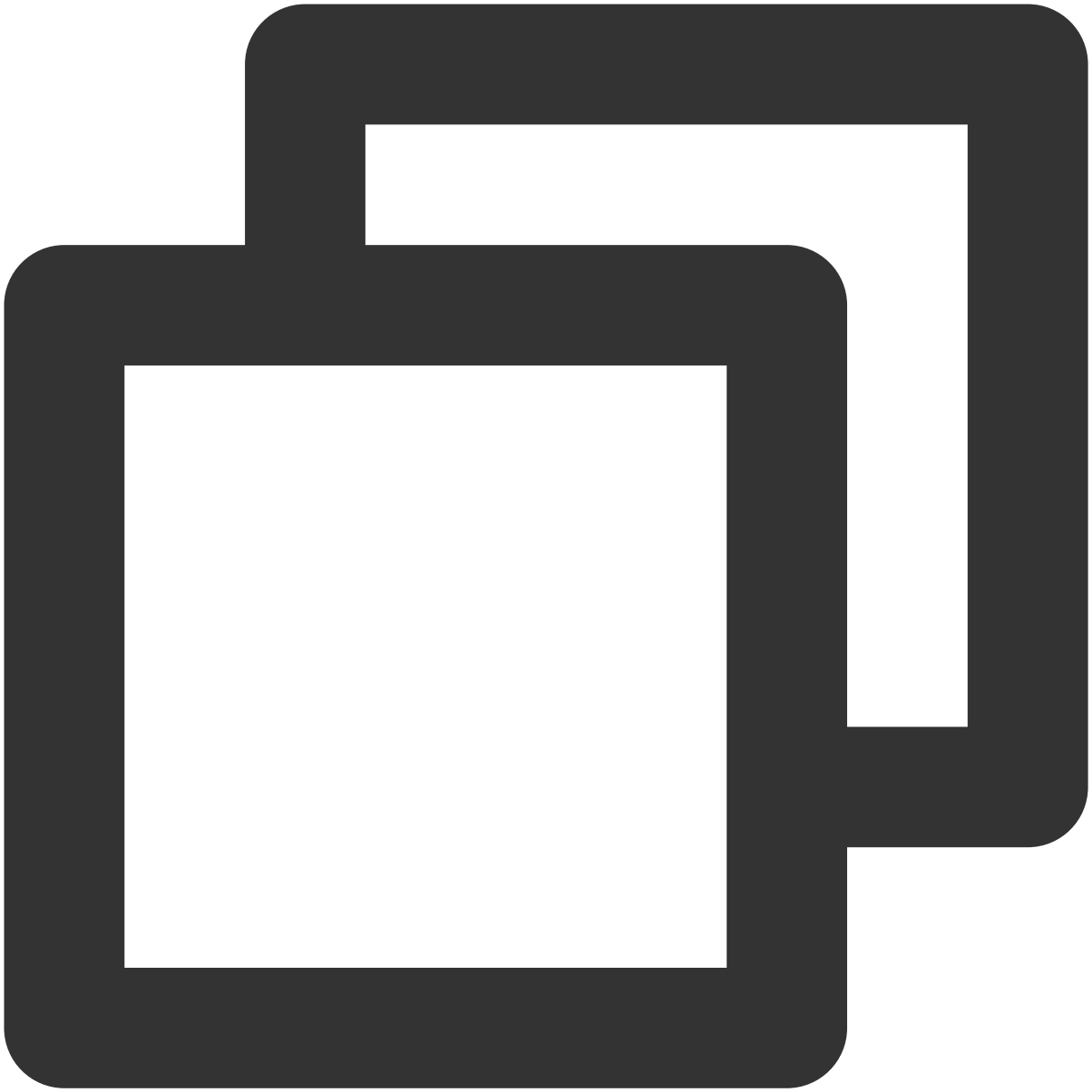
接口形式

1. 下载时处理



```
download_url?imageMogr2/format/<Format>
                       /cgif/<FrameNumber>
                       /interlace/<Mode>
```

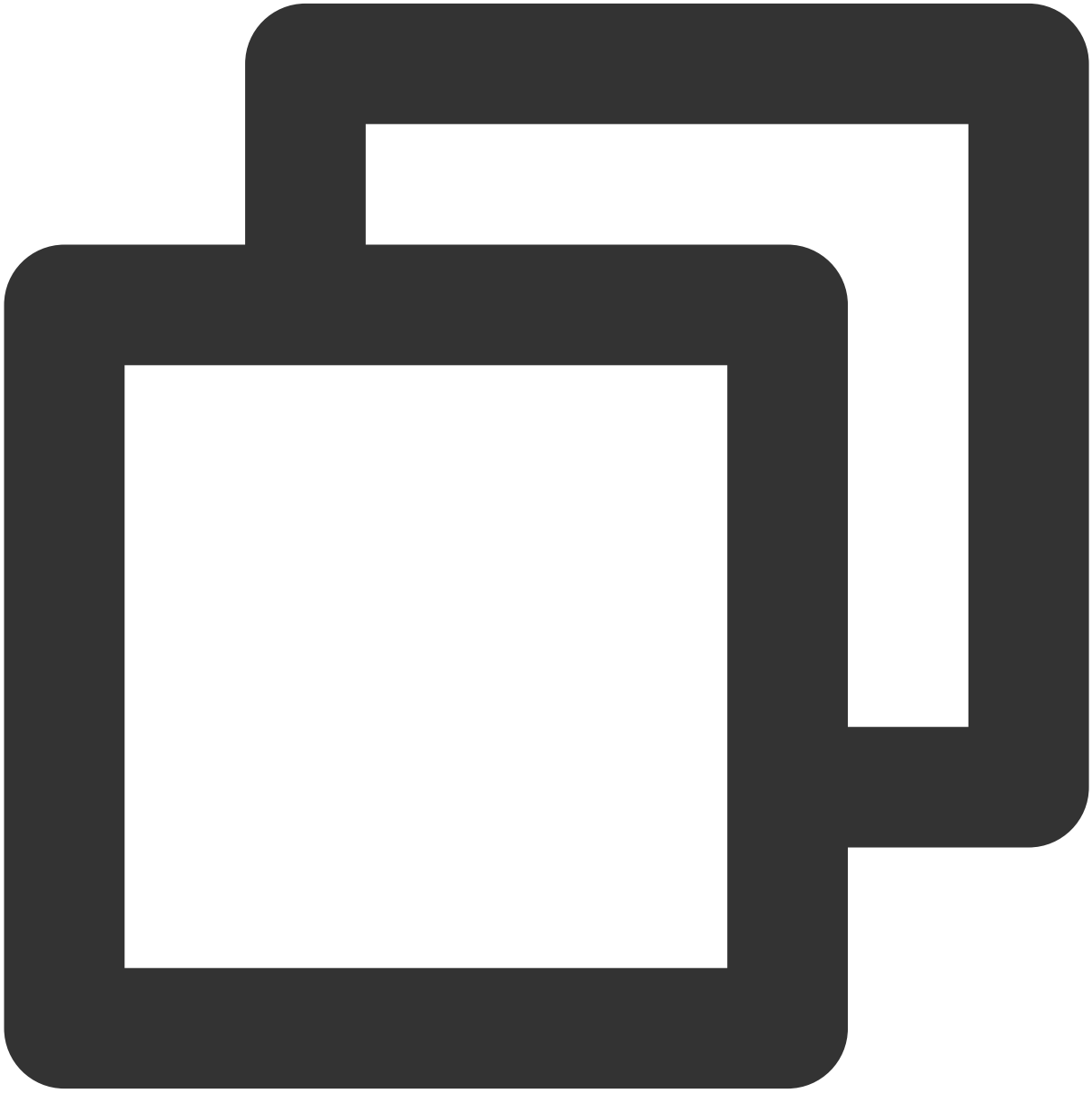
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/format/<Format>
                      /cgif/<FrameNumber>
```

```
                        /interlace/<Mode>"
  }]
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
```

```
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/format/<Format>
                      /cgif/<FrameNumber>
                      /interlace/<Mode>"
  }]
}
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

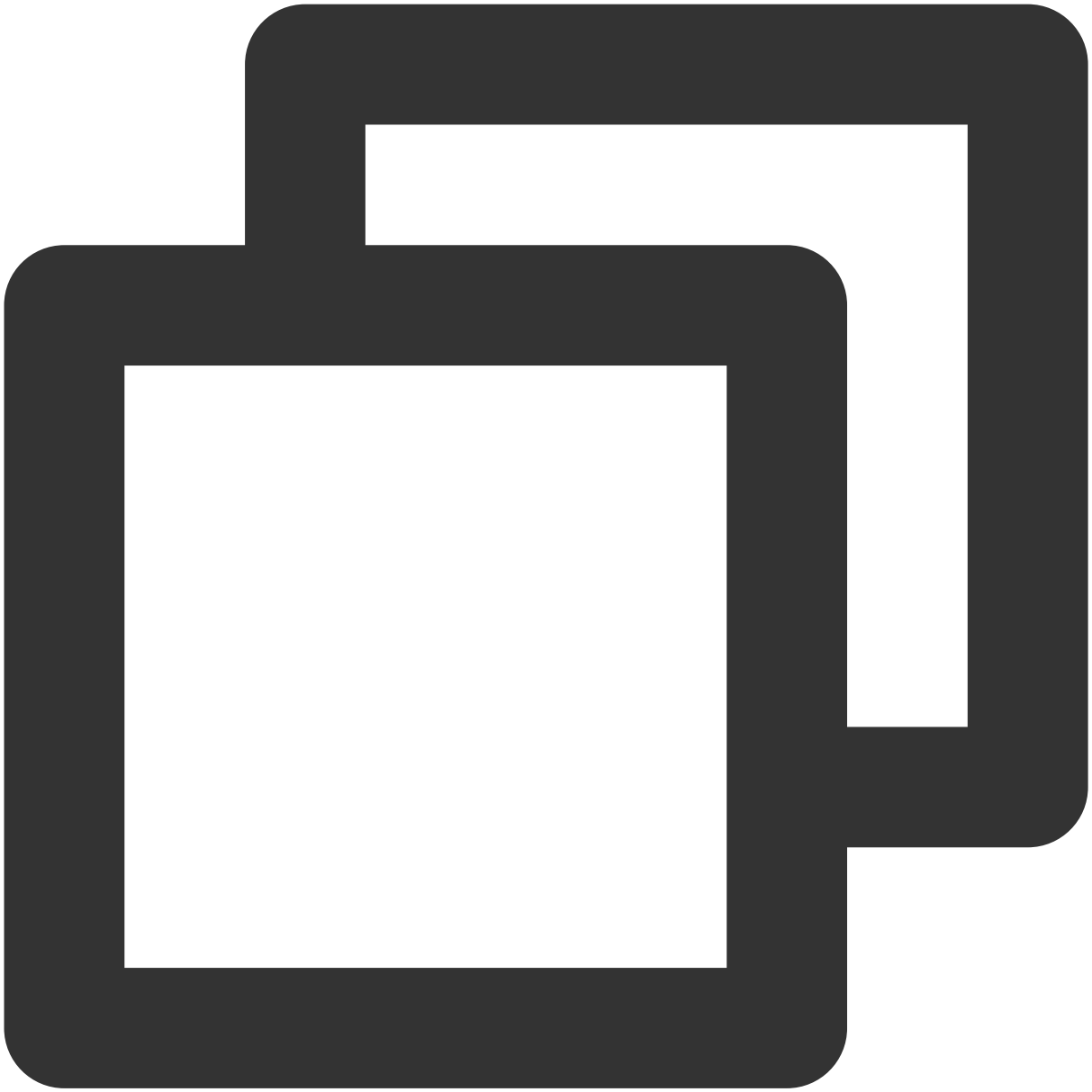
参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/format/<Format>	格式转换：目标缩略图的图片格式可为：jpg，bmp，gif，png，webp，yjpeg 等，其中 yjpeg 为数据万象针对 jpeg 格式进行的优化，本质为 jpg 格式；缺省为原图格式。
/cgif/<FrameNumber>	gif 格式优化：只针对原图为 gif 格式，对 gif 图片格式进行的优化，降帧降颜色。分为以下两种情况： FrameNumber=1，则按照默认帧数30处理，如果图片帧数大于该帧数则截取。 FrameNumber 取值(1,100]，则将图片压缩到指定帧数 （FrameNumber）。
/interlace/<Mode>	输出为渐进式 jpg 格式。Mode 可为0或1。0：表示不开启渐进式；1：表示开启渐进式。该参数仅在输出图片格式为 jpg 格式时有效。如果输出非 jpg 图片格式，会忽略该参数，默认值0。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

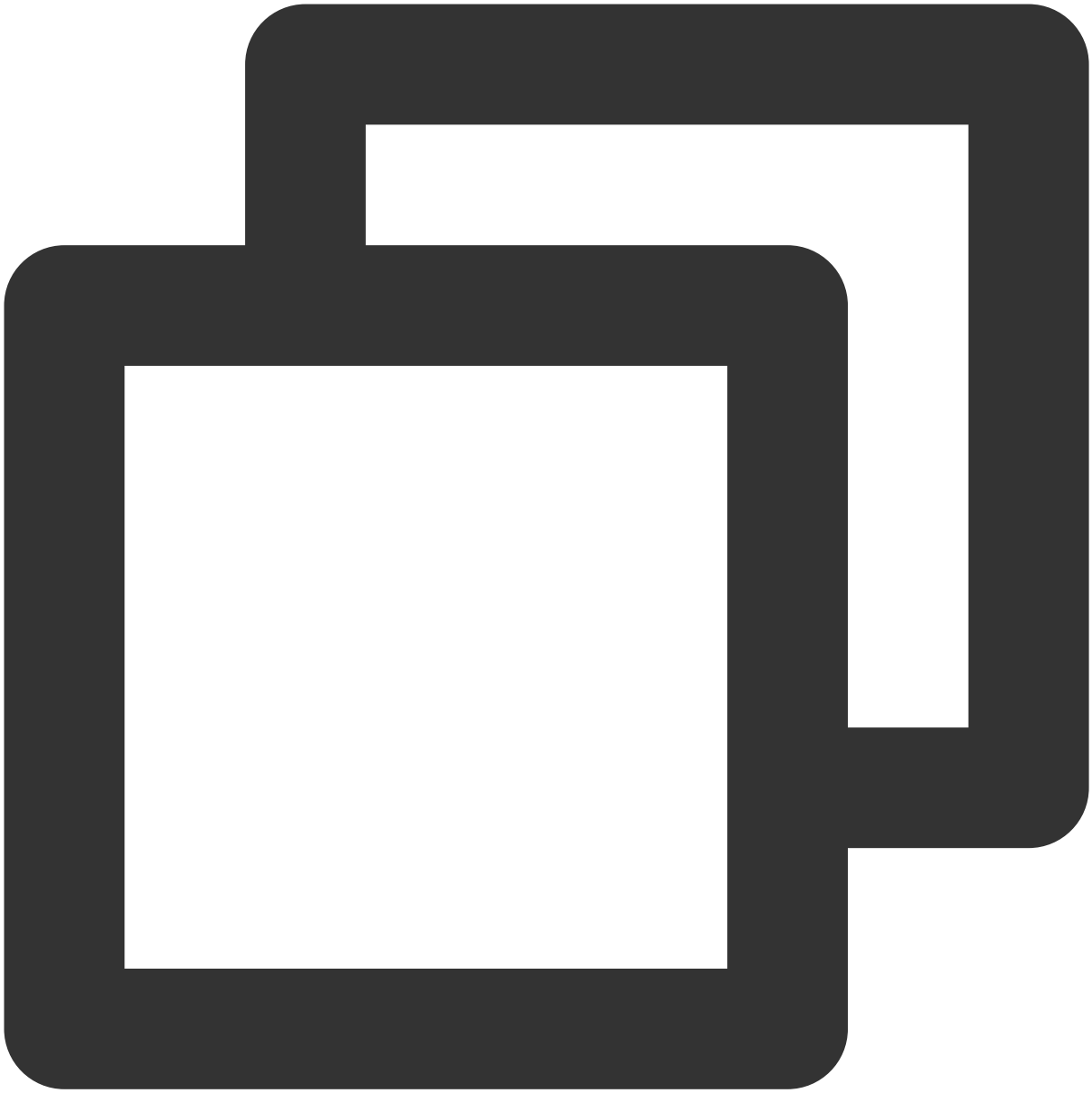
案例一：将 jpeg 格式的原图片转换为 png 格式



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/form
```

案例二：将 jpeg 格式的原图片转换为 png 格式并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



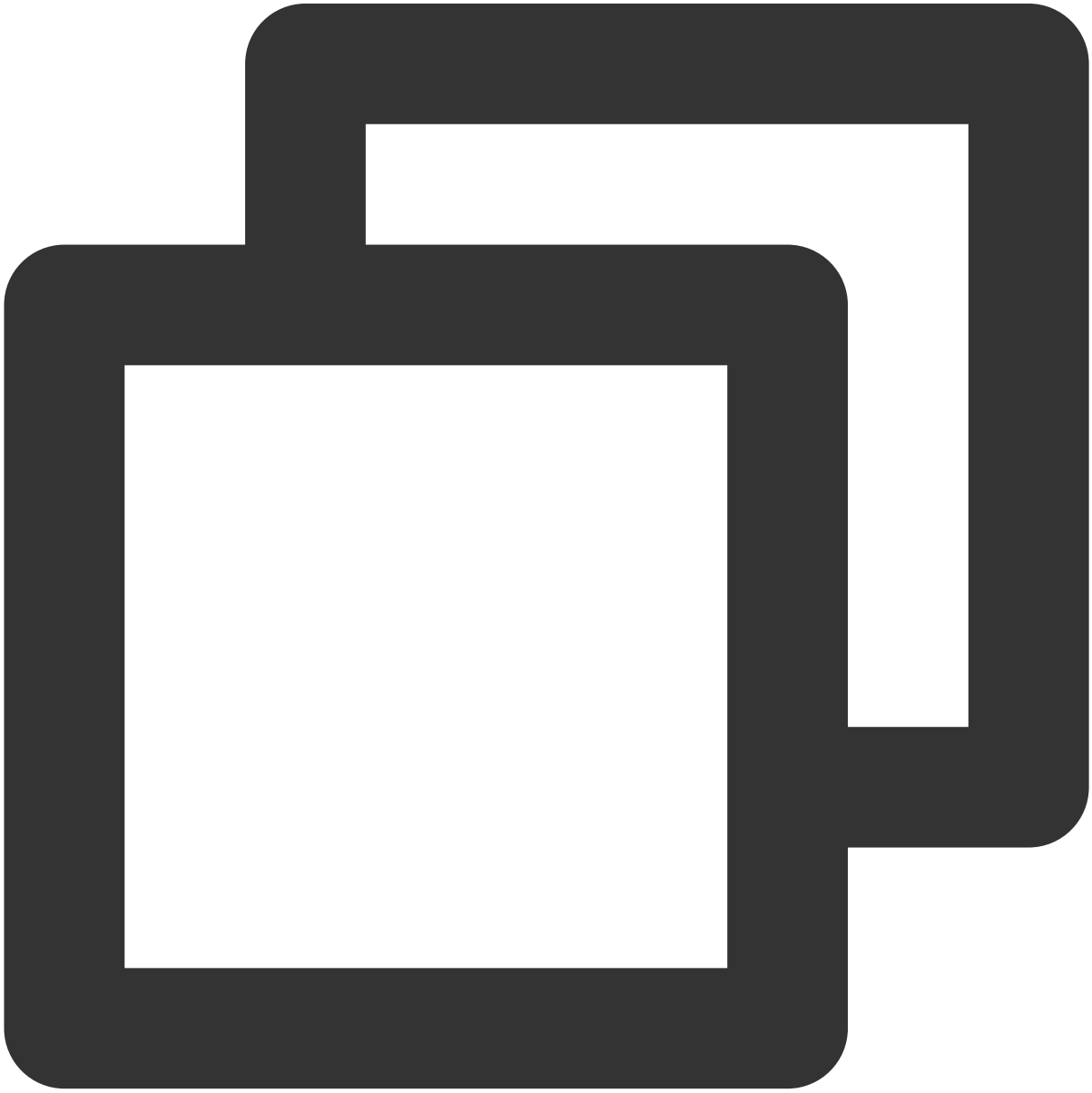
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

质量变换

最近更新时间：2023-12-21 17:55:40

功能概述

对象存储通过数据万象 **imageMogr2** 接口对图片质量进行调节。

该功能支持以下处理方式：

下载时处理

上传时处理

云上数据处理

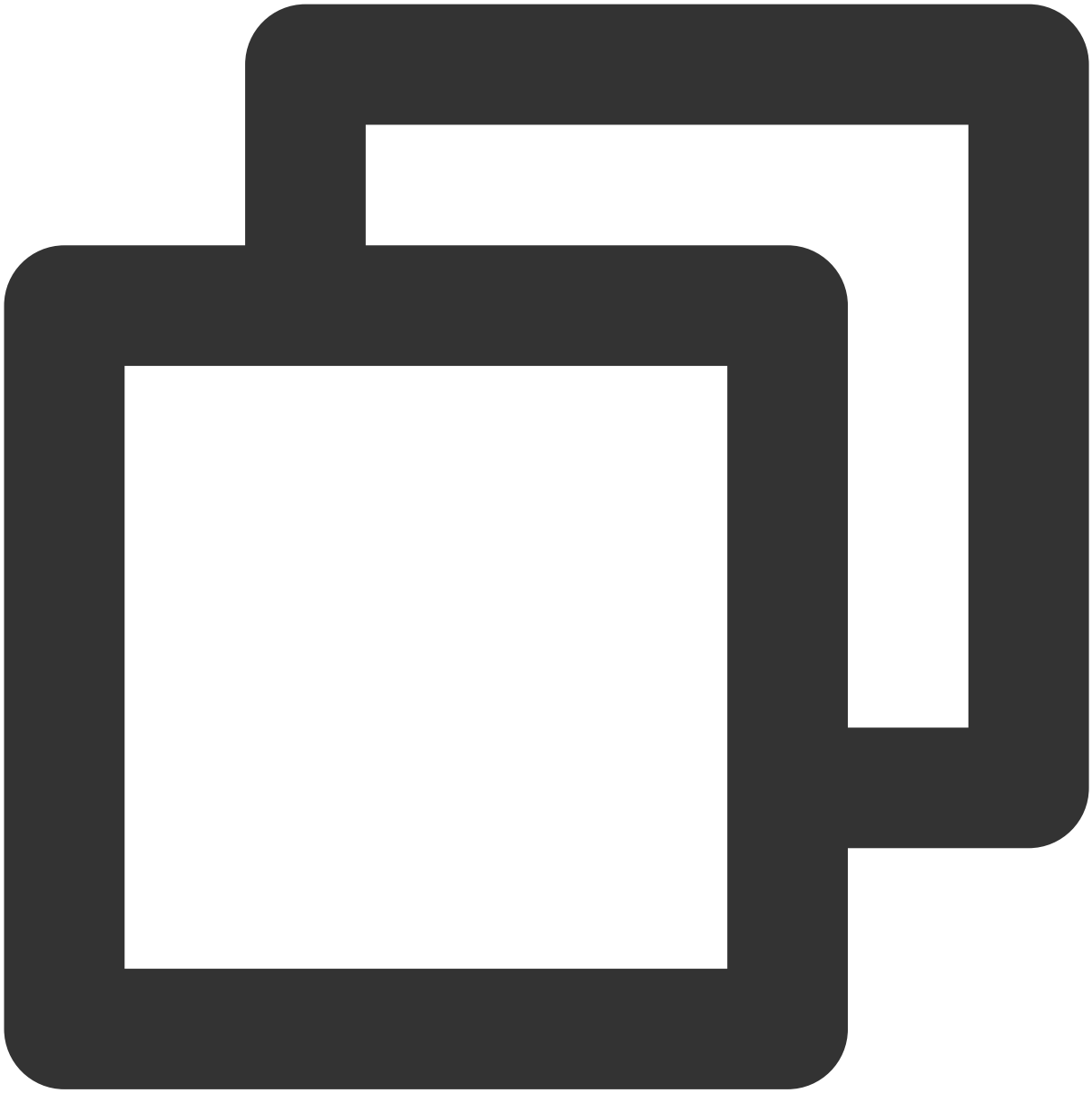
说明

图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。

该接口支持 jpg、webp、tpg、heif、avif 格式图片。

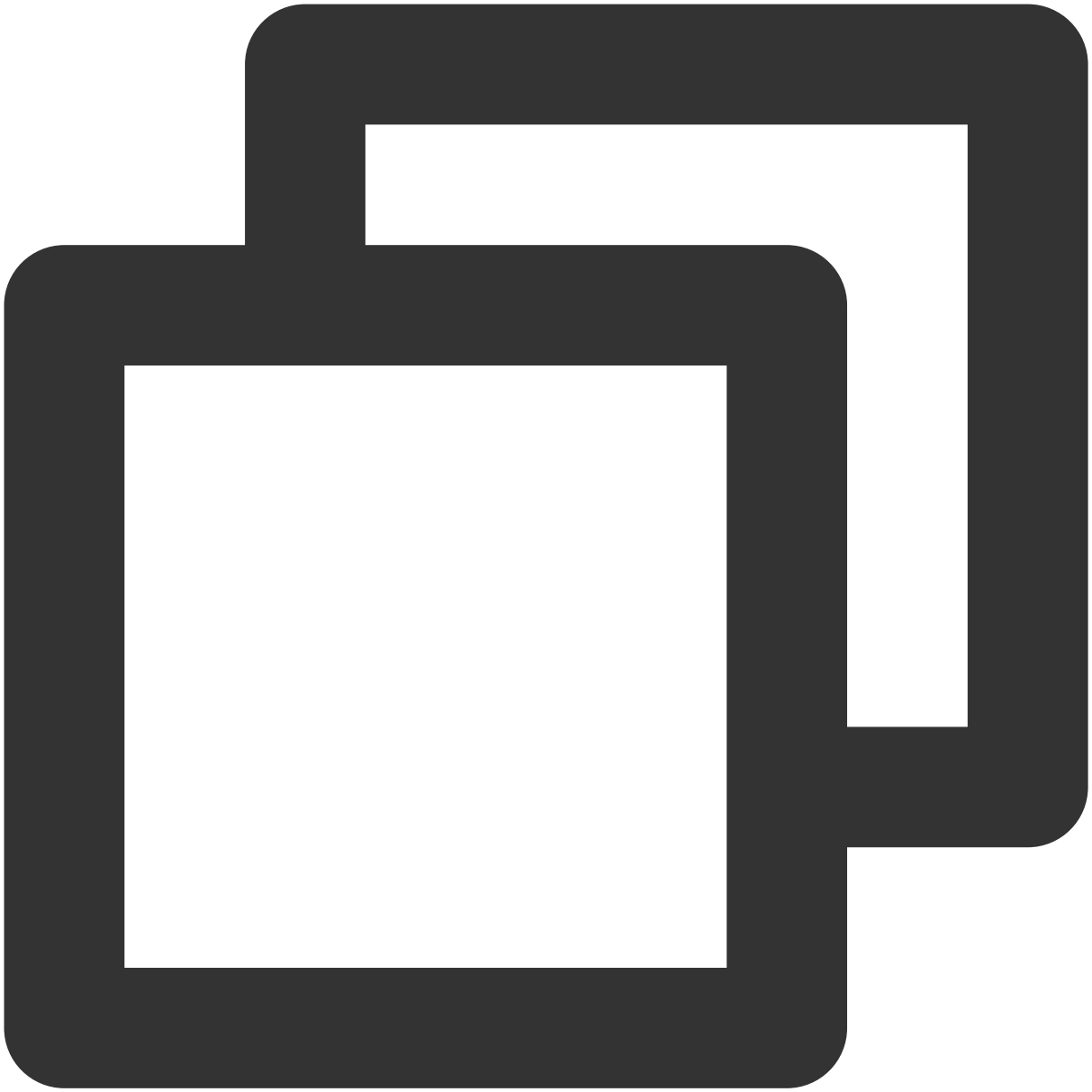
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/quality/<Quality>
                       /rquality/<quality>
                       /lquality/<quality> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



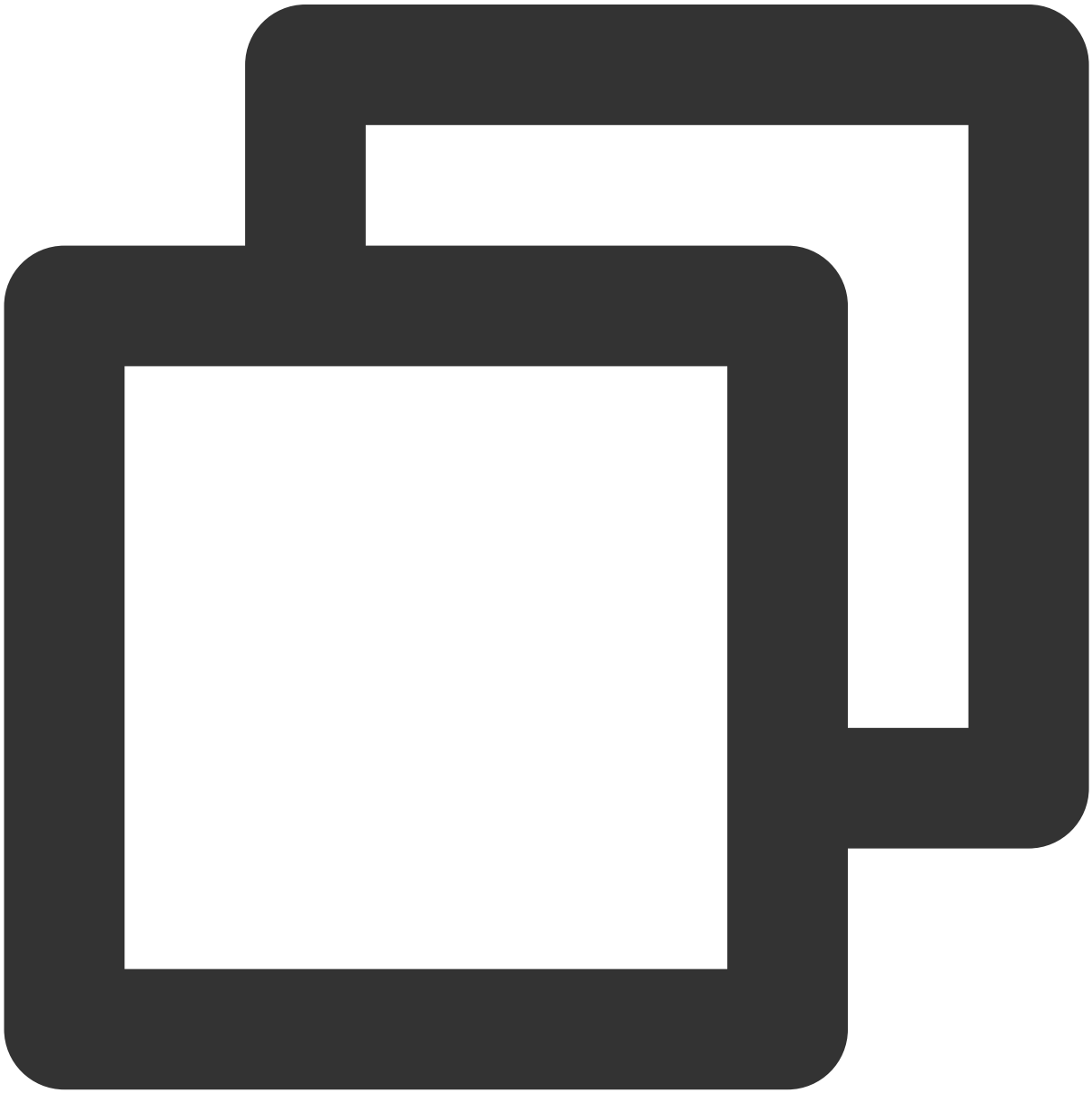
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/quality/<Quality>
                      /rquality/<quality>
```

```
                        /lquality/<quality>"
    }]
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
```

```
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/quality/<Quality>
                        /rquality/<quality>
                        /lquality/<quality>"
  }]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

操作名称：quality。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/quality/<Quality>	图片的绝对质量，取值范围0 - 100，默认值为原图质量；取原图质量和指定质量的最小值；<Quality>后面加“!”表示强制使用指定值，例如：`90!`。
/rquality/<quality>	图片的相对质量，取值范围0 - 100，数值以原图质量为标准。例如原图质量为80，将 rquality 设置为80后，得到处理结果图的图片质量为64（80x80%）。
/lquality/<quality>	图片的最低质量，取值范围0 - 100，设置结果图的质量参数最小值。 例如原图质量为85，将 lquality 设置为80后，处理结果图的图片质量为85。 例如原图质量为60，将 lquality 设置为80后，处理结果图的图片质量会被提升至80。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

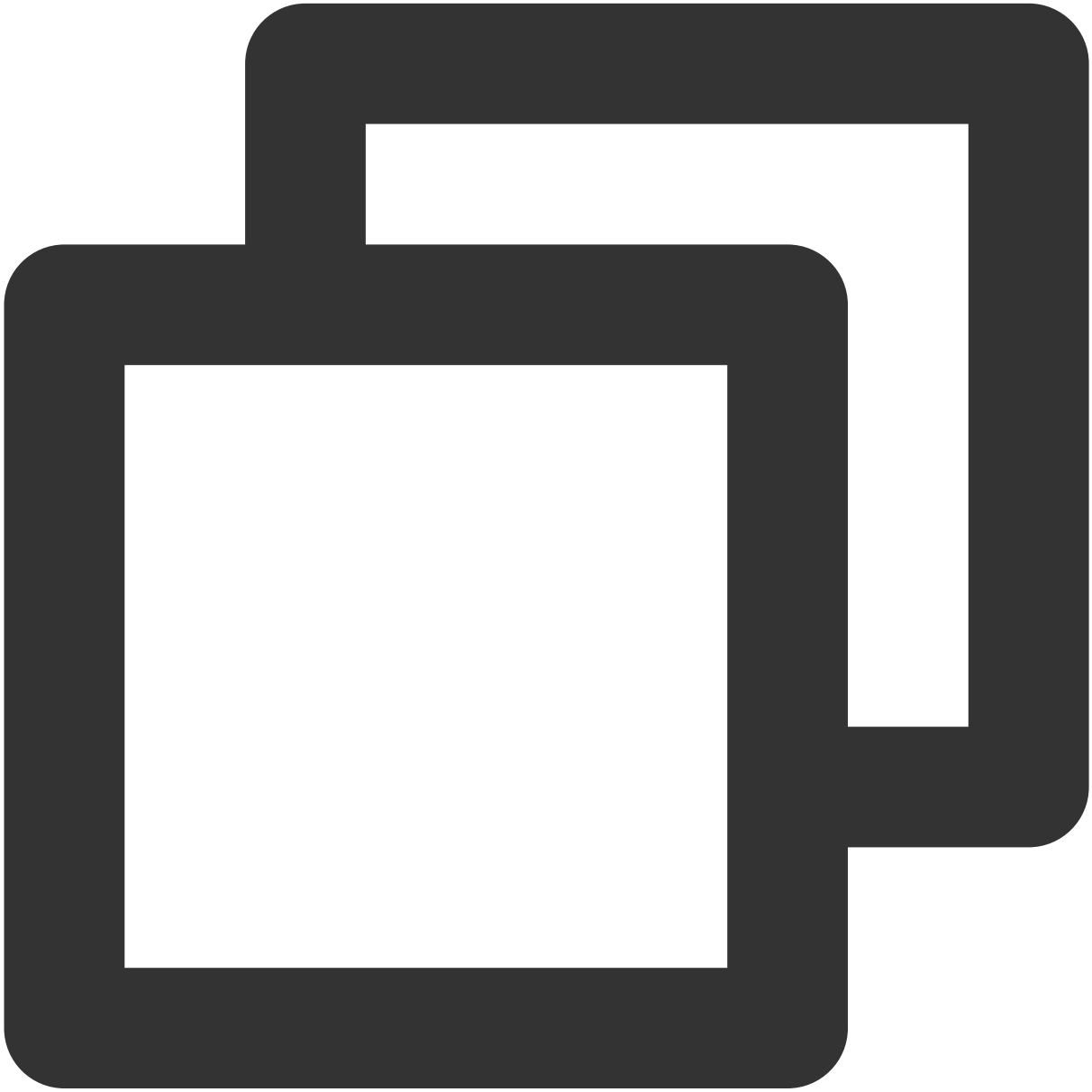
实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：设置绝对质量

假设设置**绝对质量**为60，示例如下：



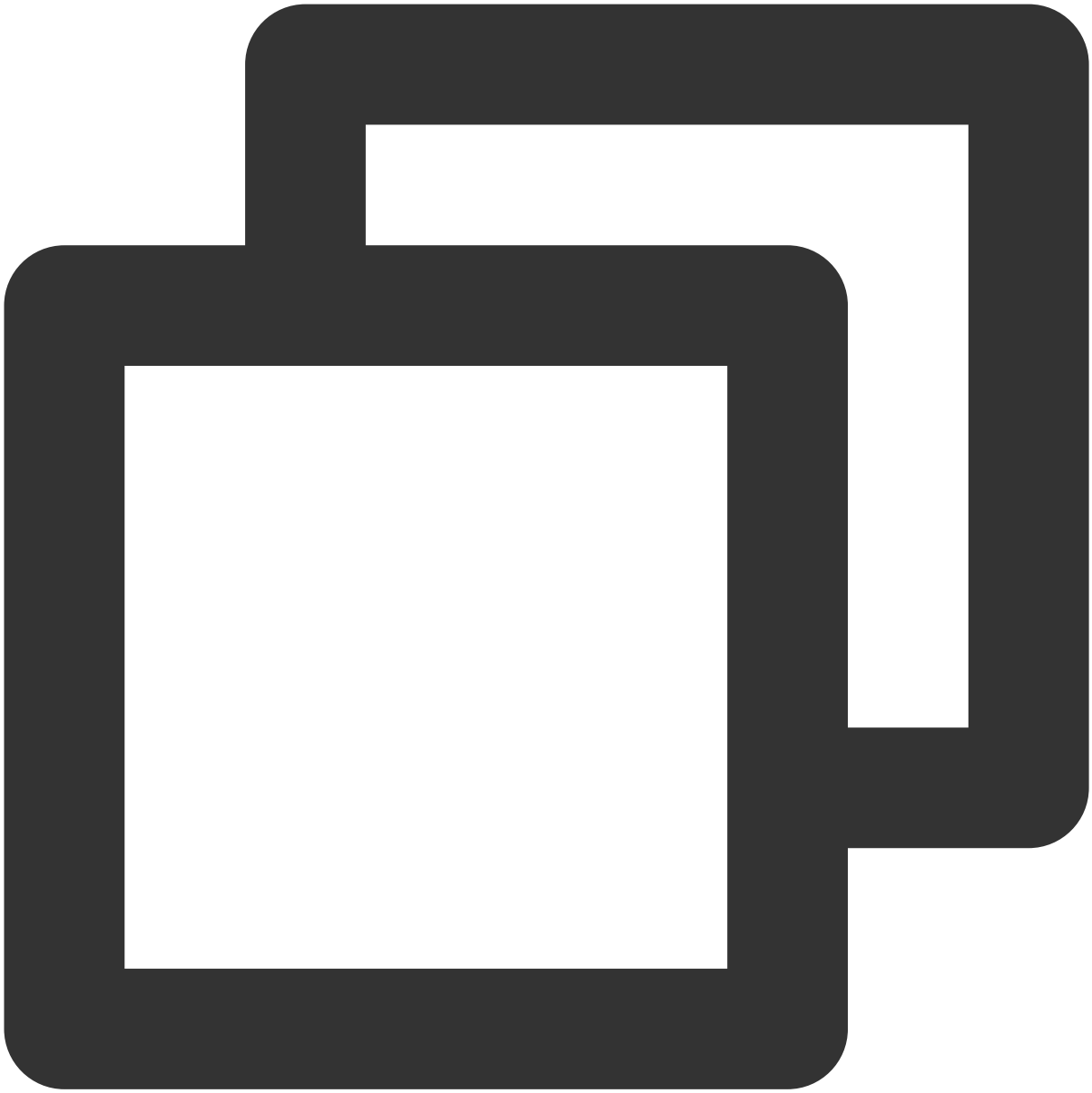
```
http://examples-1251000004.picsh.myqcloud.com/sample.jpeg?imageMogr2/quality/60
```

效果如下：



案例二：设置相对质量

假设设置**相对质量**为60，示例如下：



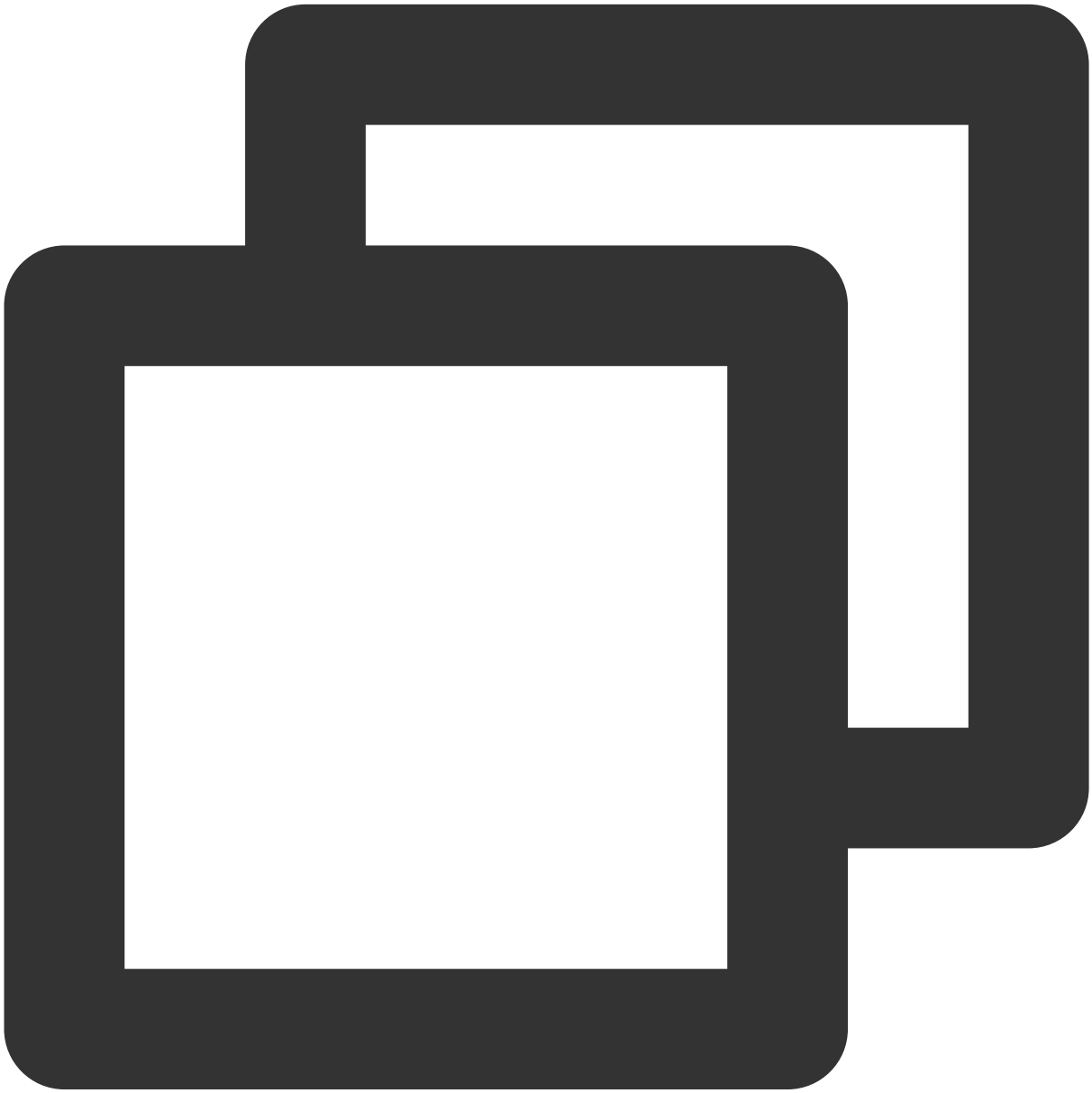
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/rqua
```

效果如下：



案例三：设置相对质量为60并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



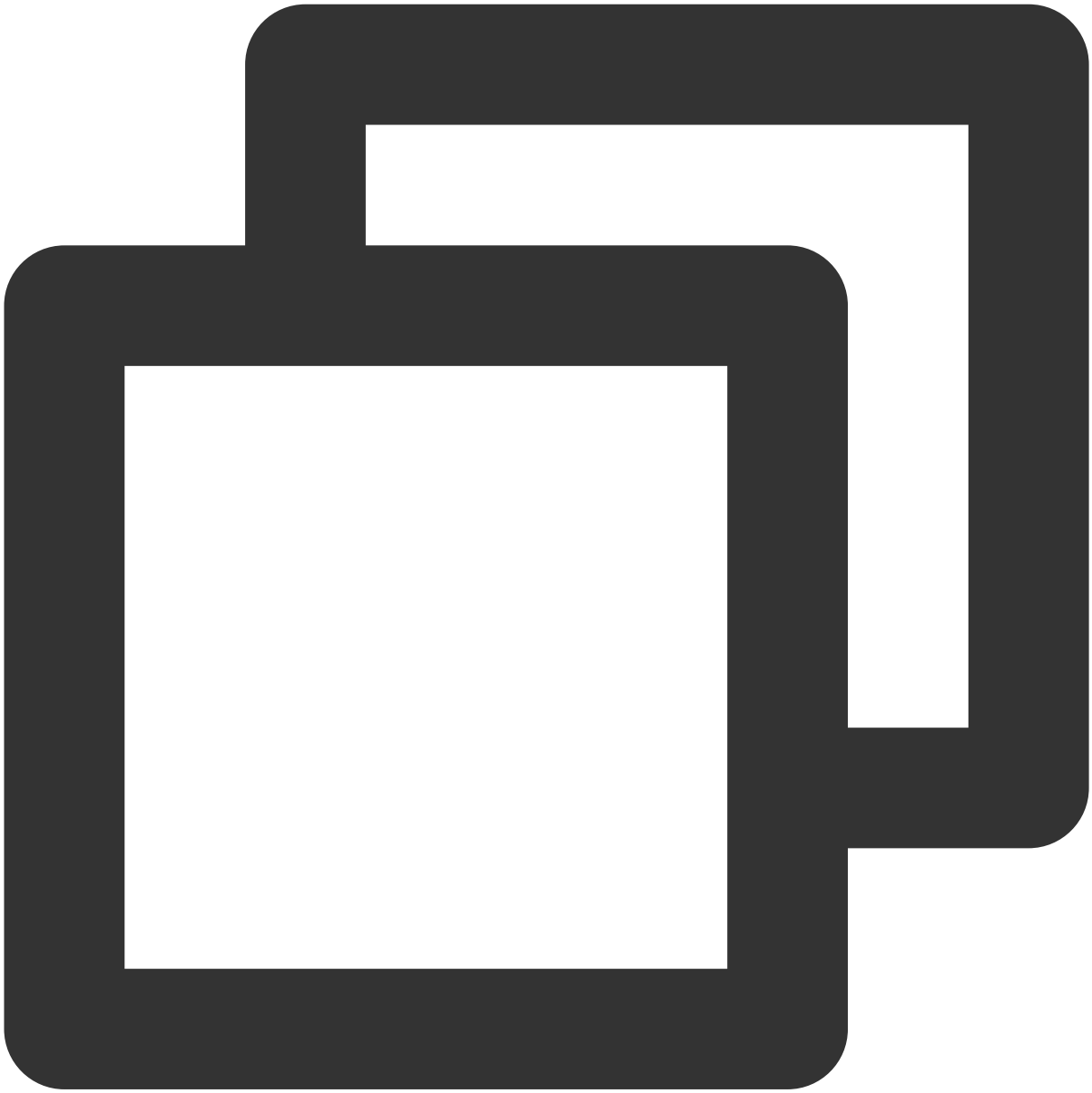
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

高斯模糊

最近更新时间：2023-12-21 17:56:11

功能概述

对象存储通过数据万象接口 **imageMogr2** 提供图片模糊处理功能。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理

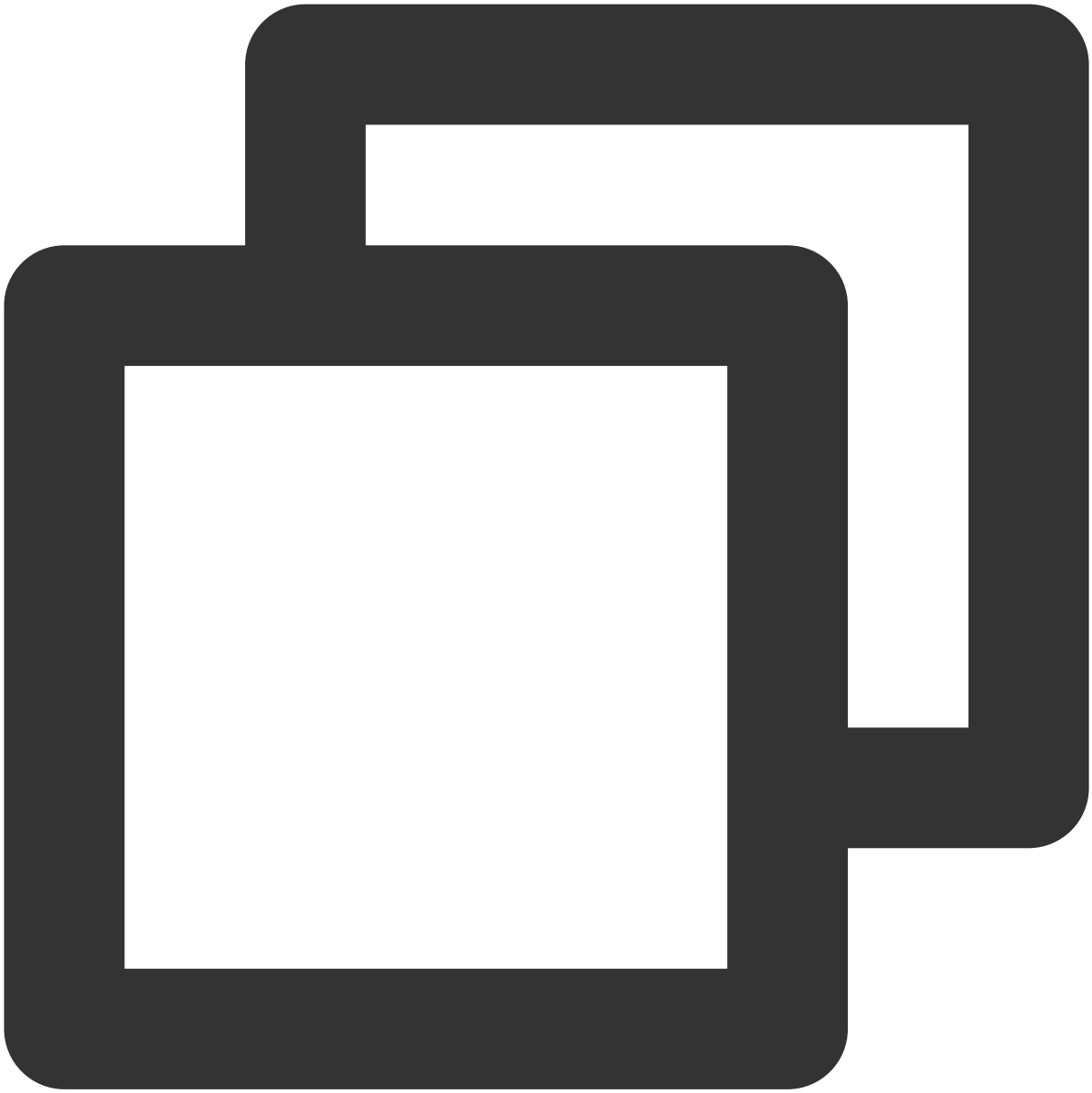
限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

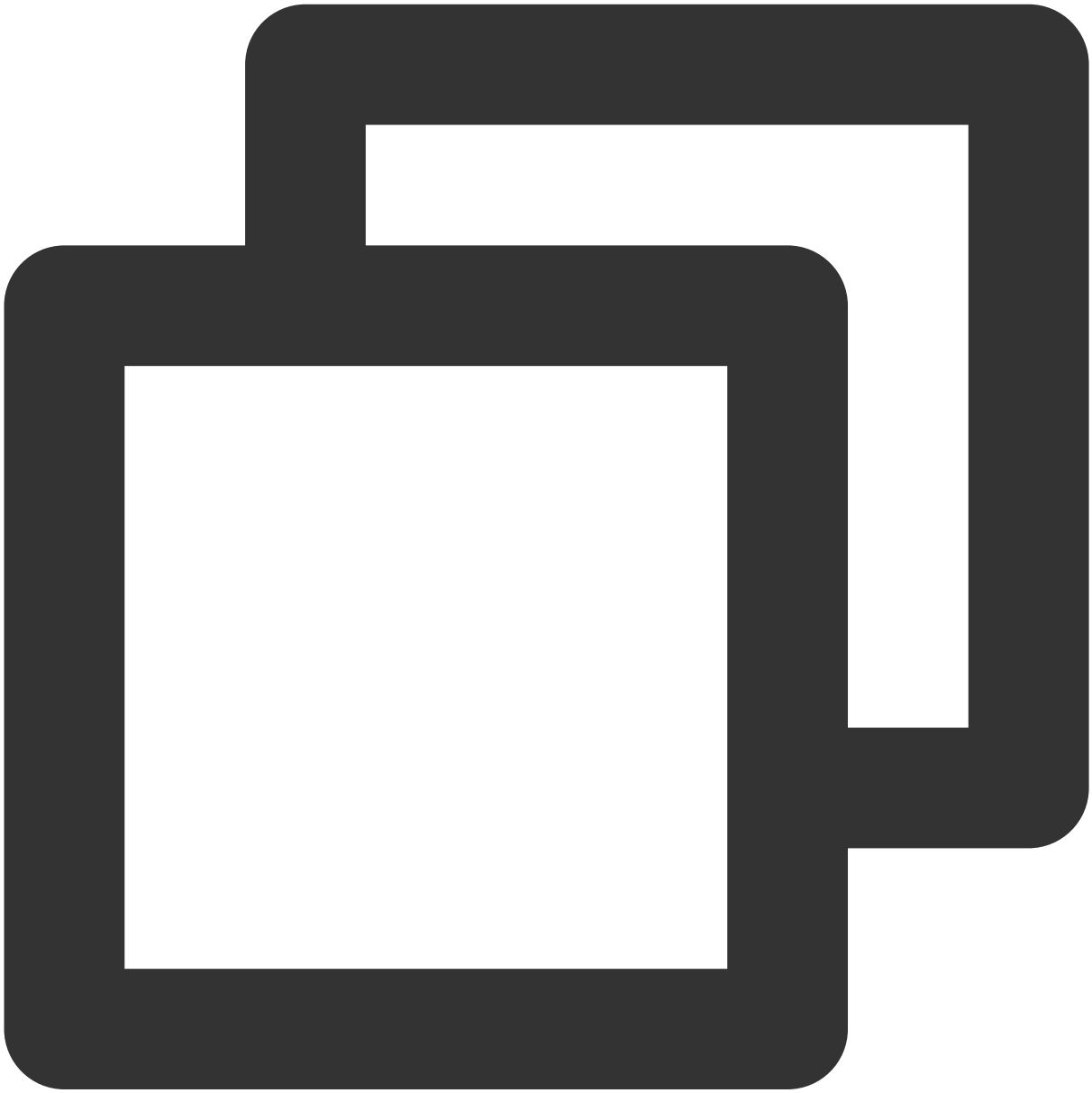
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
download_url?imageMogr2/blur/<radius>x<sigma>
```

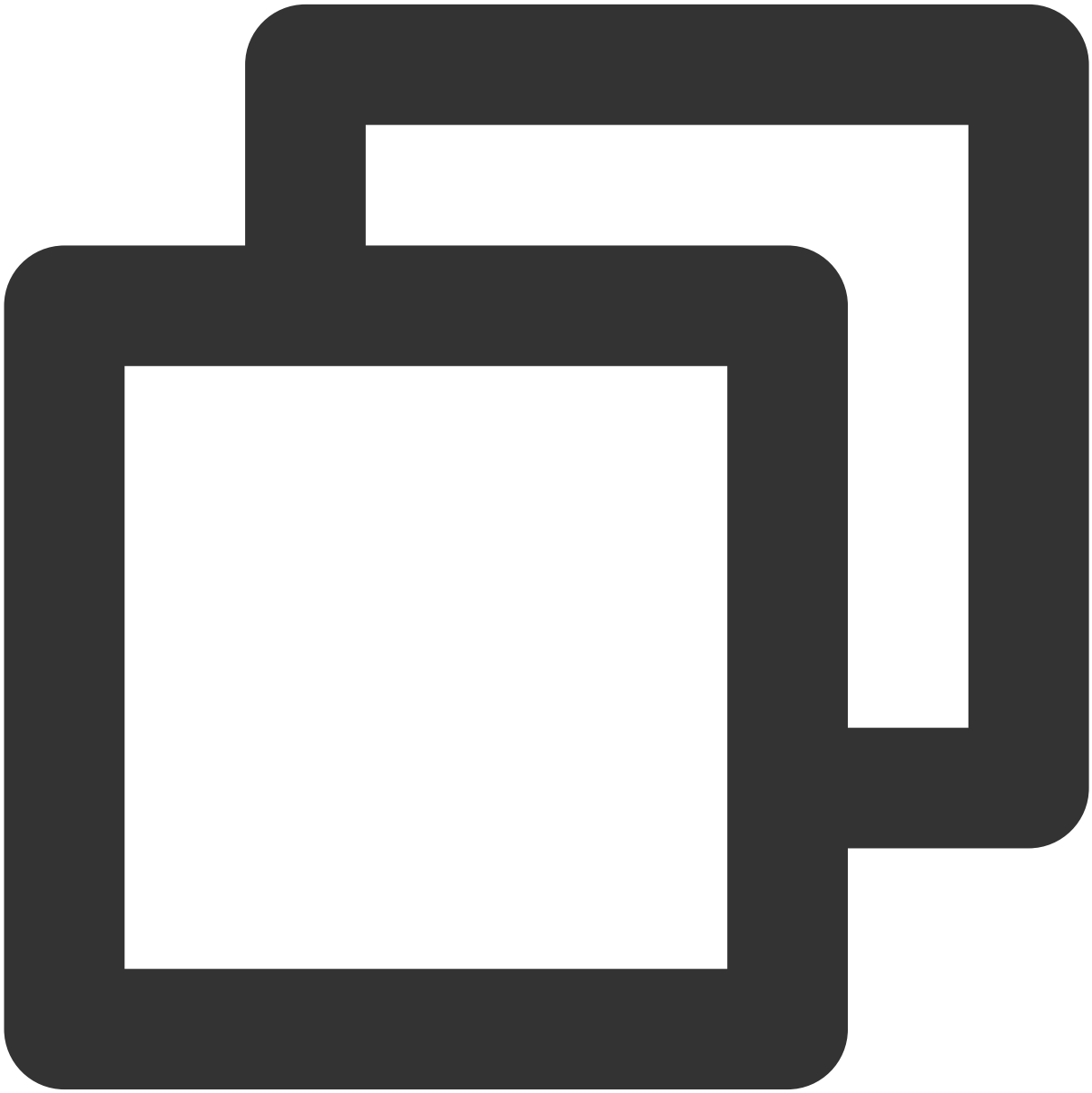
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/blur/<radius>x<sigma>"
  }]
```

```
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
```

```
    "is_pic_info": 1,
    "rules": [{
        "fileid": "exampleobject",
        "rule": "imageMogr2/blur/<radius>x<sigma>"
    }]
}
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

操作名称：blur。

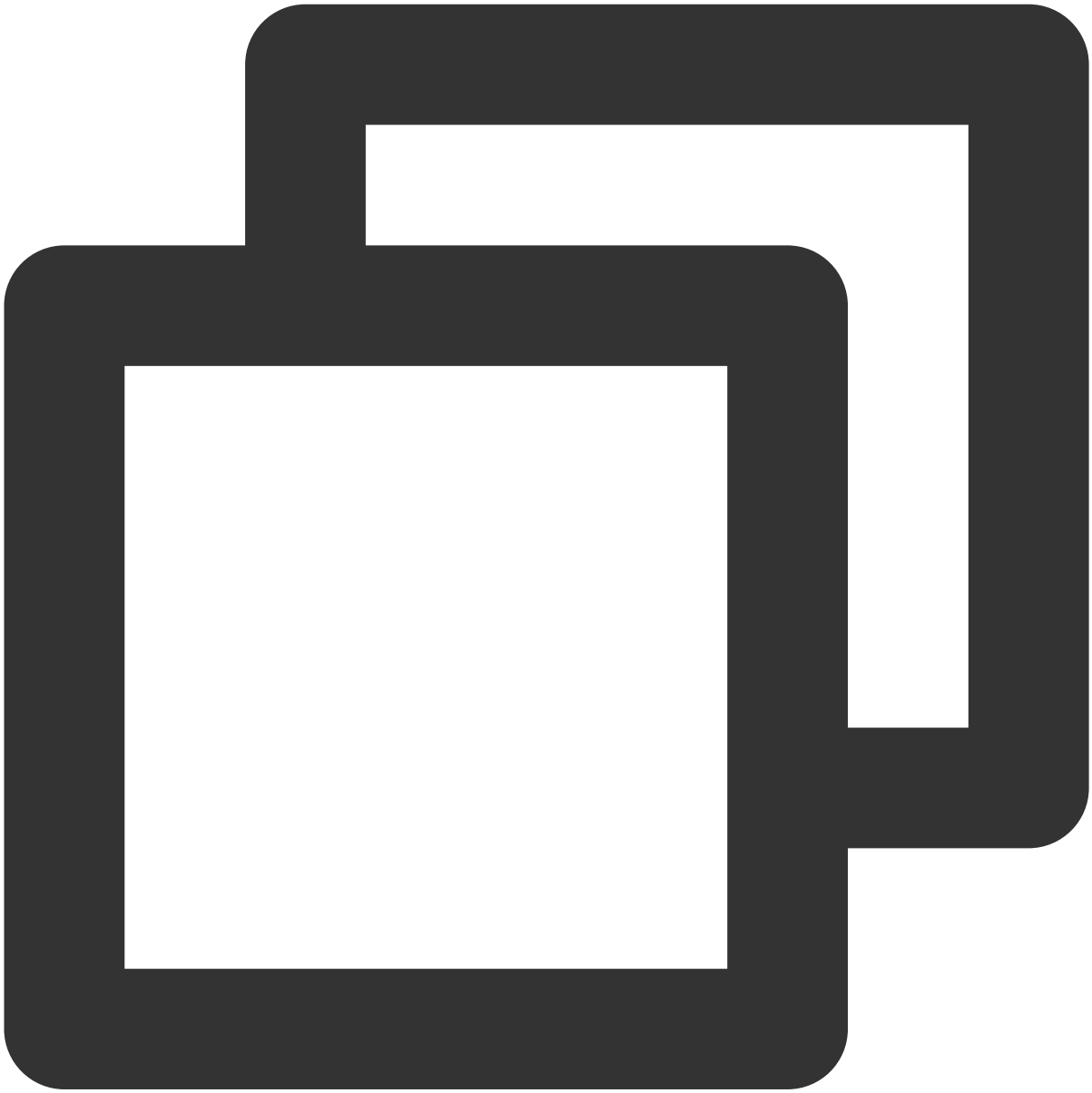
参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`
radius<radius>	模糊半径，取值范围为1 - 50
sigma<sigma>	正态分布的标准差，必须大于0
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：模糊半径取8，sigma 值取5，进行高斯模糊处理



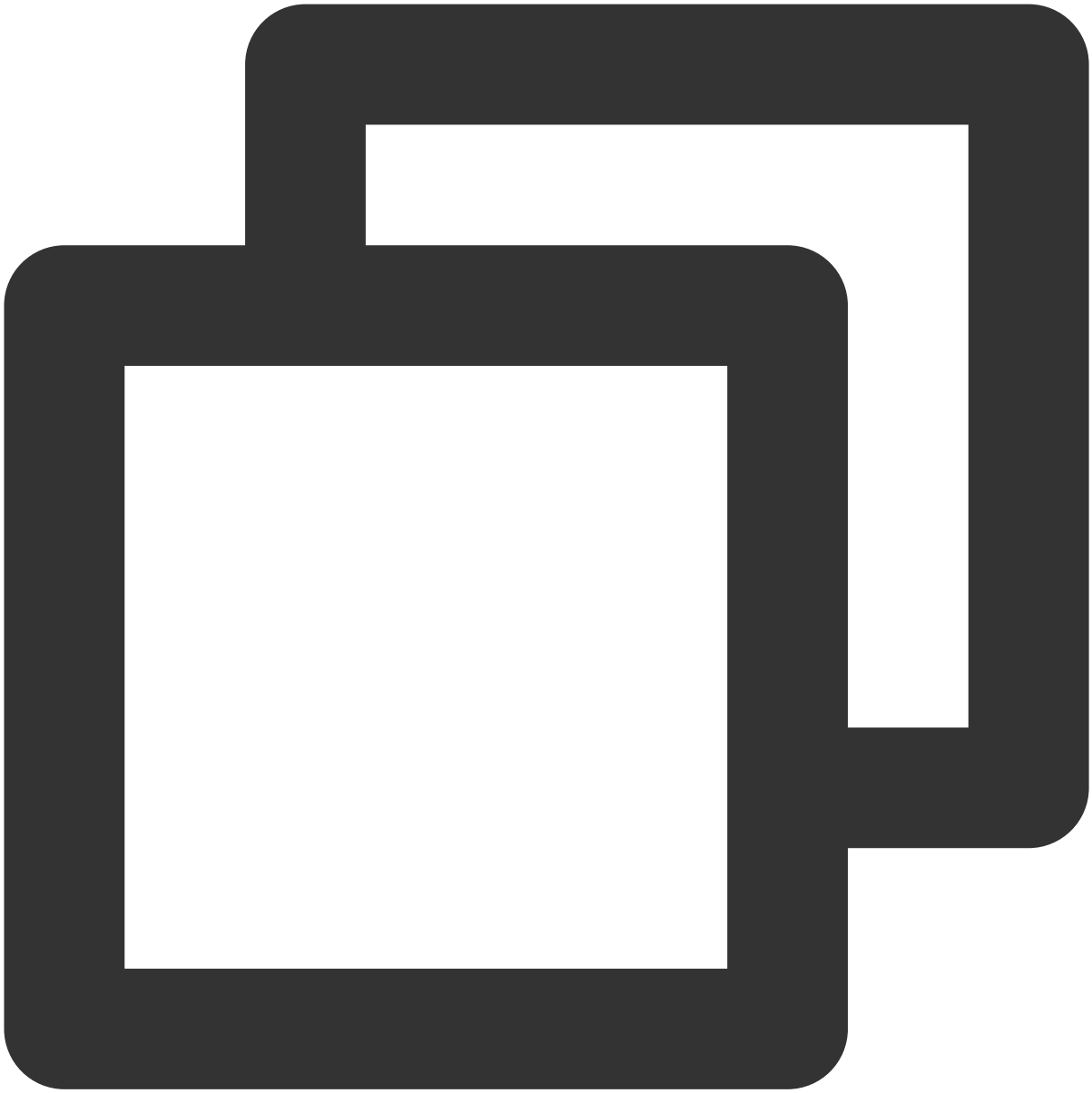
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/blur
```

高斯模糊处理后效果如下：



案例二：模糊半径取8，sigma 值取5，进行高斯模糊处理，并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



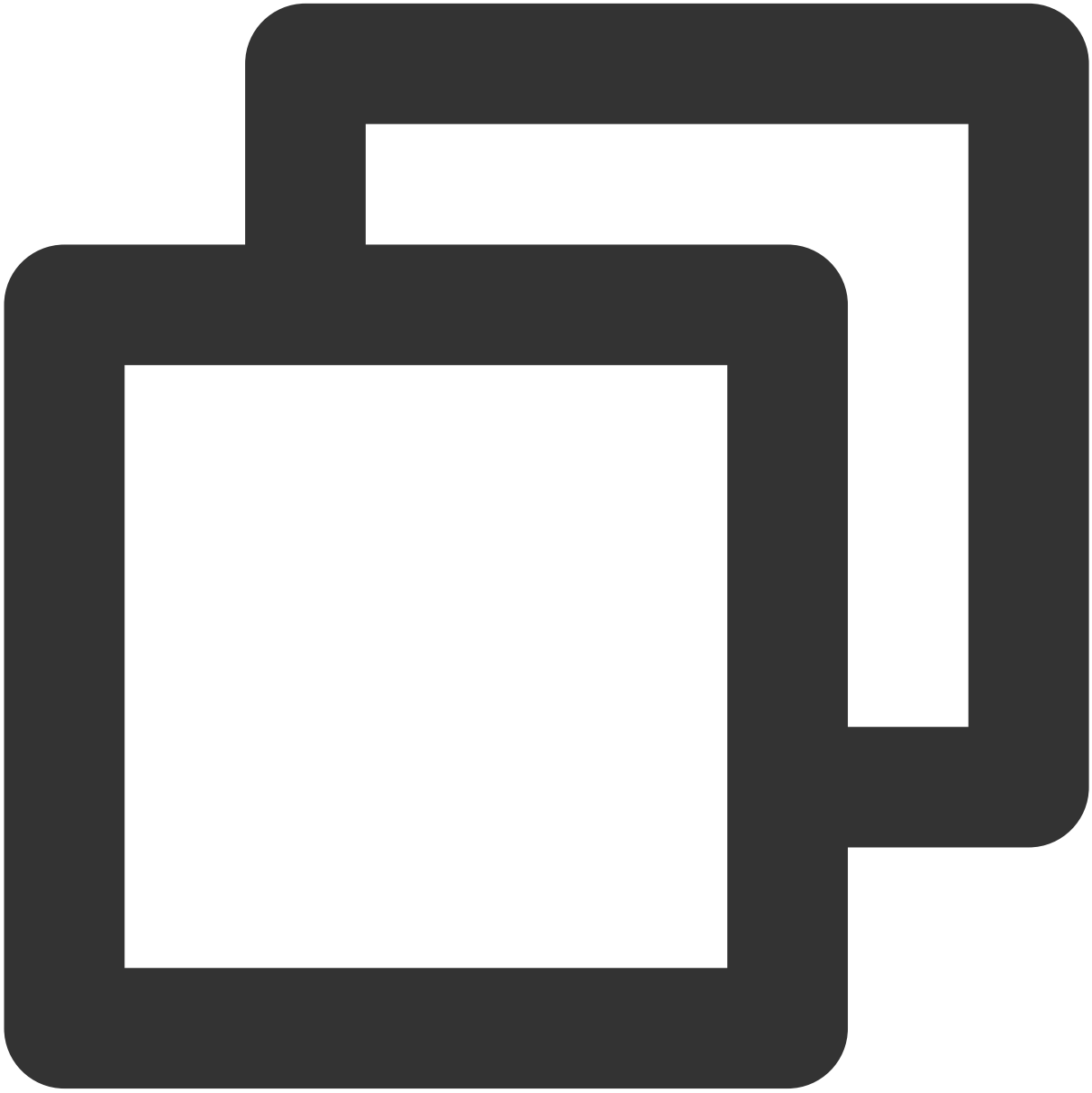
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

亮度

最近更新时间：2023-12-21 17:56:54

功能概述

对象存储通过数据万象 **imageMogr2** 接口调节图片亮度。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理

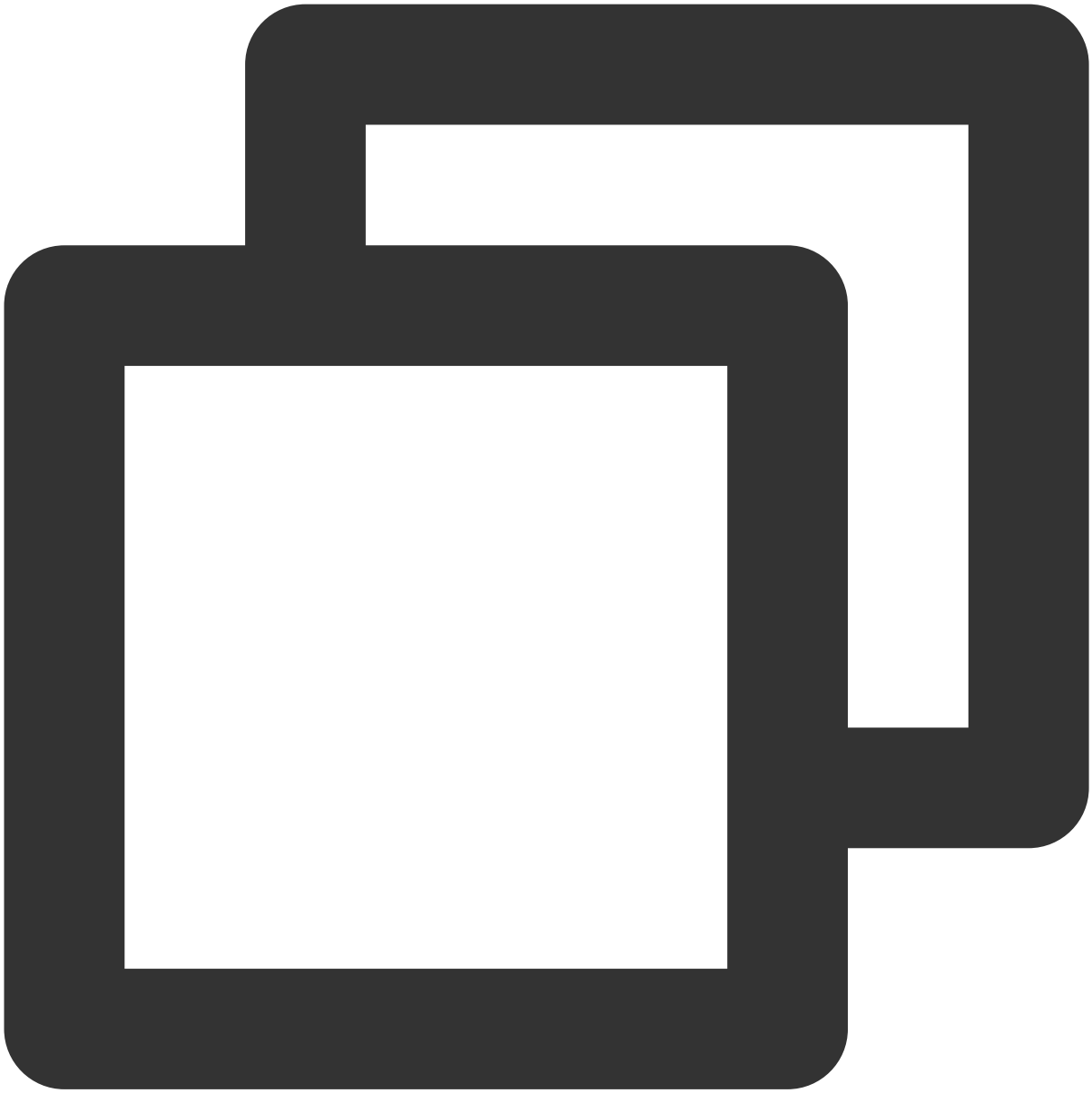
限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

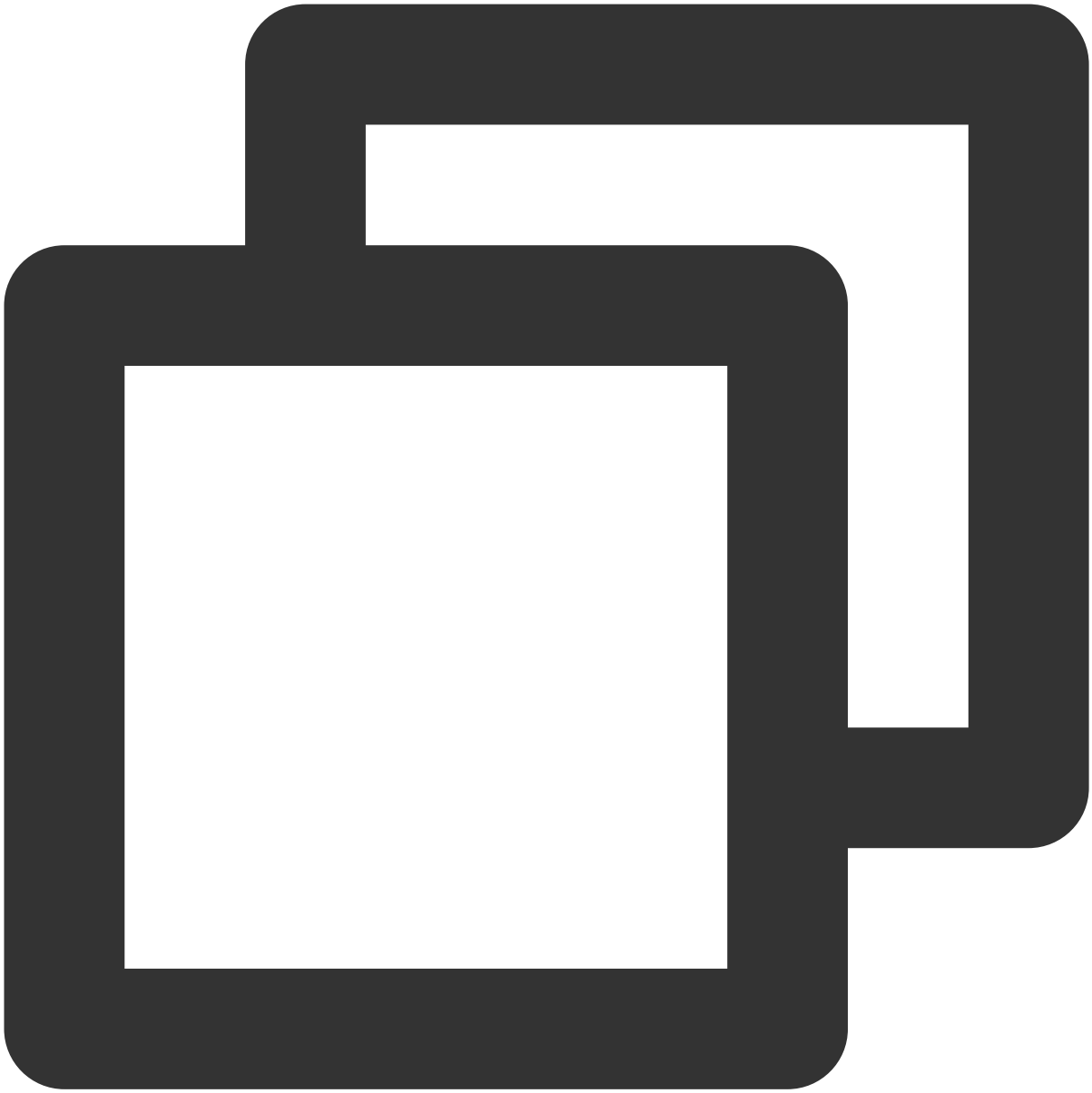
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
download_url?imageMogr2/bright/<value>
```

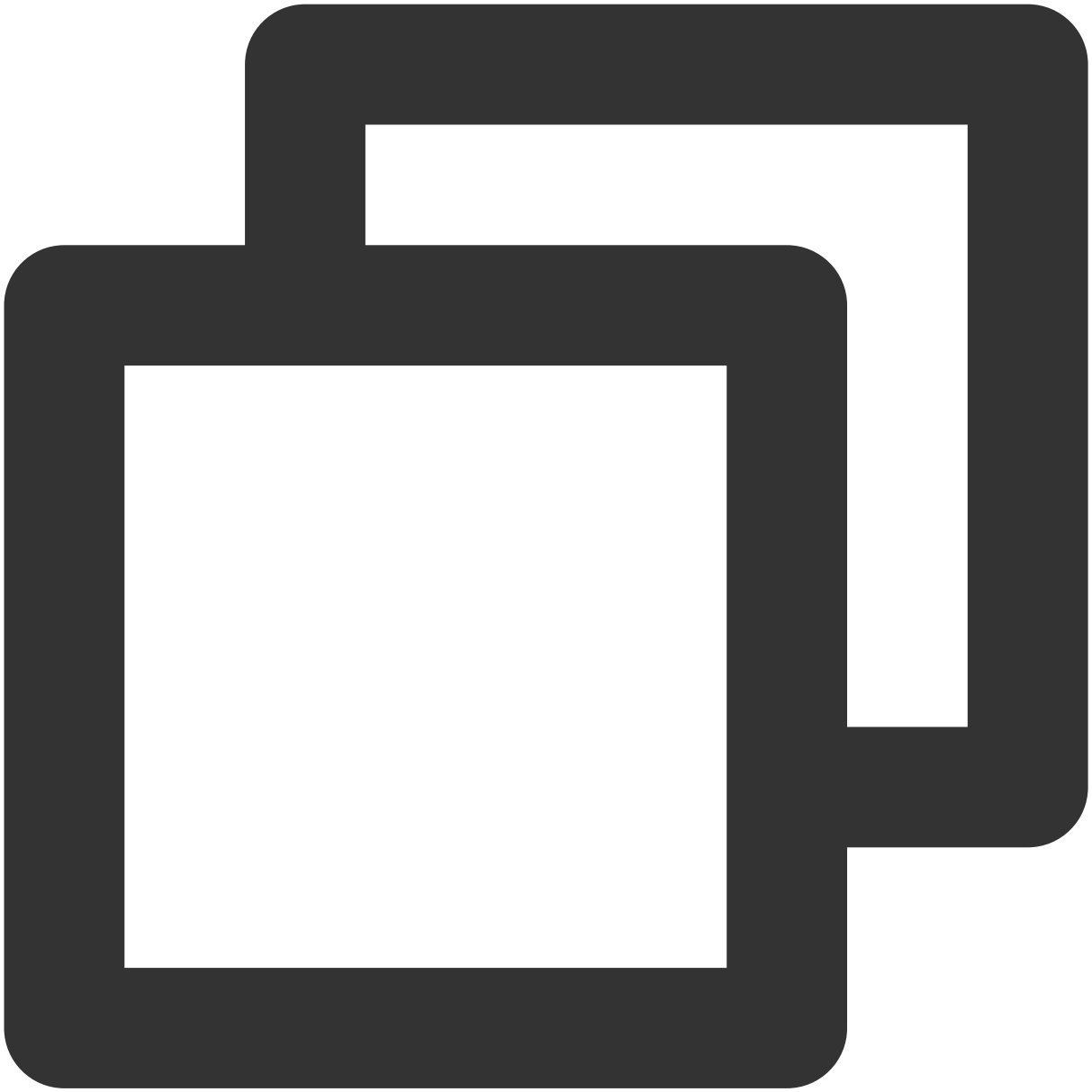
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/bright/<value>"
  }]
```

```
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
```

```
    "is_pic_info": 1,
    "rules": [{
        "fileid": "exampleobject",
        "rule": "imageMogr2/bright/<value>"
    }]
}
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

操作名称：bright。

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/bright/<value>	图片亮度调节功能，value 为亮度参数值，取值范围为[-100, 100]的整数。 取值＜0：降低图片亮度。 取值 = 0：不调整图片亮度。 取值＞0：提高图片亮度。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错。

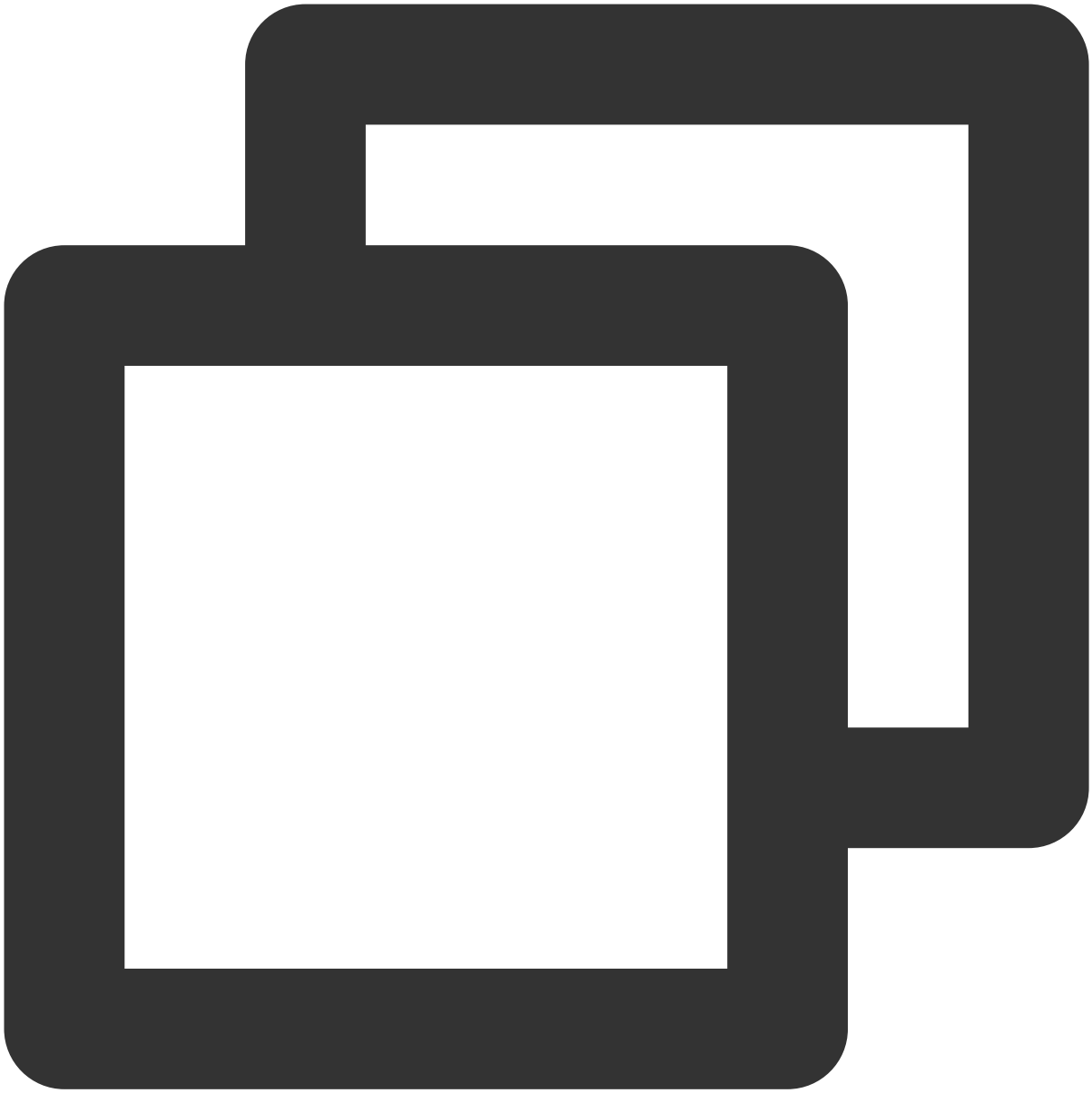
实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：亮度调节

将图片亮度提高70，示例如下：



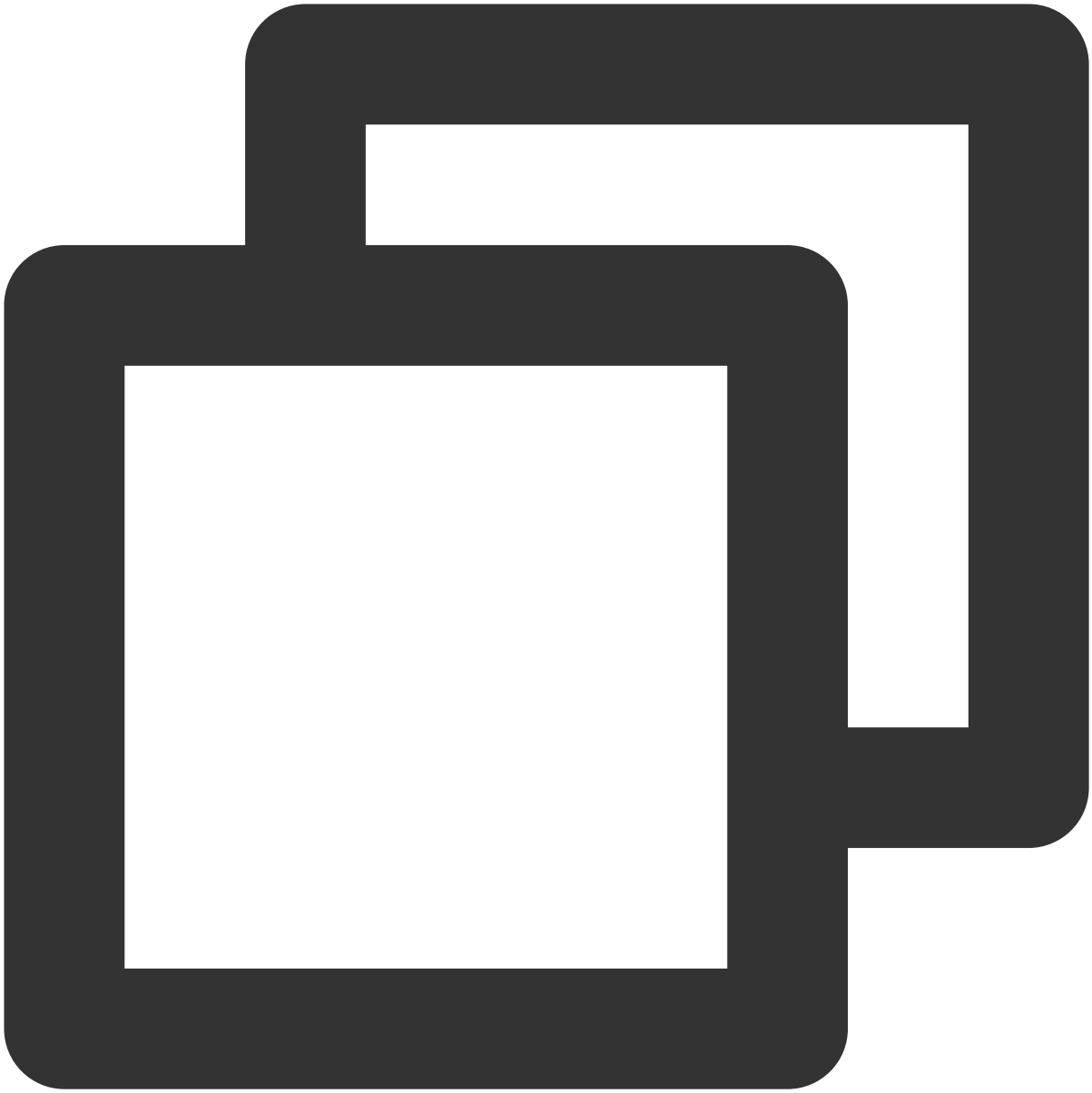
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/brig
```

最终效果如下：



案例二：亮度调节并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



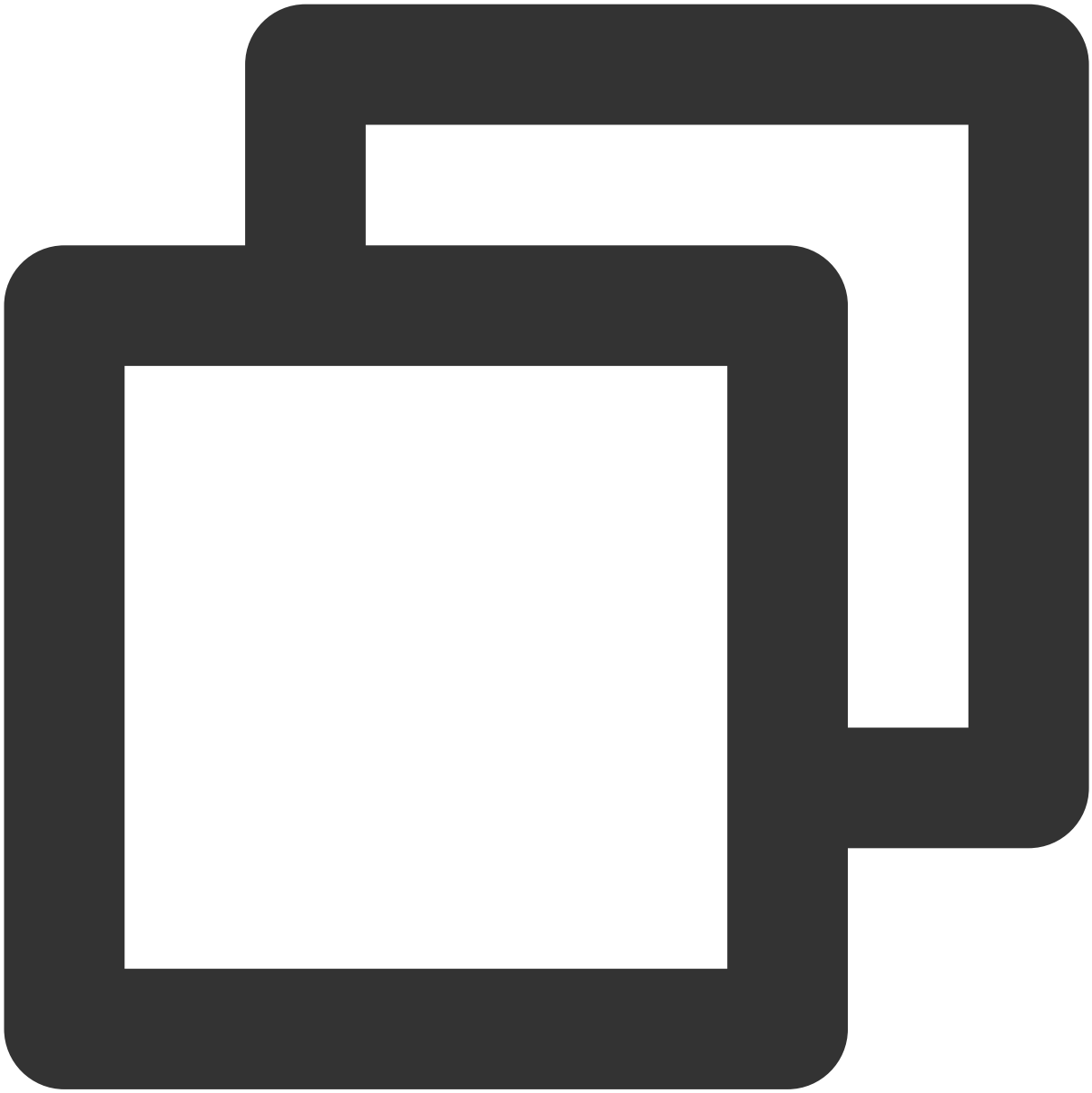
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了防止未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

对比度

最近更新时间：2021-10-22 14:17:33

功能概述

对比度指一幅图像中明暗区域最亮的白和最暗的黑之间不同亮度层级的测量，即指一幅图像灰度反差的大小。对象存储通过数据万象 **imageMogr2** 接口调节图片对比度。

该功能支持以下处理方式：

- 下载时处理
- 上传时处理
- 云上数据处理

限制说明

- 支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
- 体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
download_url?imageMogr2/contrast/<value>
```

2. 上传时处理

```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
"fileid": "exampleobject",
```

版权所有：腾讯云计算（北京）有限责任公司

```
"rule": "imageMogr2/contrast/<value>"
}]
}
```

3. 云上数据处理

```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
"fileid": "exampleobject",
"rule": "imageMogr2/contrast/<value>"
}]
}
```

> 说明：
> Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

操作名称：contrast。

参数	含义
download_url	文件的访问链接，具体构成为&dxlt;BucketName-APPID>.cos.&dxlt;Region>.myqcloud.com/&dxlt;picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/contrast/&dxlt;value>	图片对比度调节功能，value 为对比度参数值，取值范围为[-100, 100]的整数。 • 取值＜0：降低图片对比度。 • 取值 = 0：不调整图片对比度。 • 取值＞0：提高图片对比度。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错。

实际案例

案例一：对比度调节

> 说明：
> 本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

将图片对比度降低50，示例如下：

```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/contrast/-50
```

最终效果如下：



案例二：对比度调节并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：

```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorithm=<signature>&imageMogr2/contrast/-50
```

说明：
`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。

```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorithm=sha1&q-ak=AKID********************&q-sign-time=1593342360;1593342720&q-key-time=1593342360;1593342720&q-header-list=&q-url-param-list=watermark%252f1%252fimage%252fahr0cdovl2v4yw1wbgvzlteyntewmdawmdqucgljc2gubxlxy2xvdwquy29tl3nodwl5aw4uanbn%252fgravity%252fsoutheast&q-signature=26a429871963375c88081ef60247c5746e834a98&watermark/1/image/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQucGljc2gubXlxY2xvdWQuY29tL3NodWl5aW4uanBn/gravity/southeast
```

锐化

最近更新时间：2023-12-21 17:56:47

功能概述

对象存储通过数据万象 **imageMogr2** 接口对图片进行锐化处理。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理
注意
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

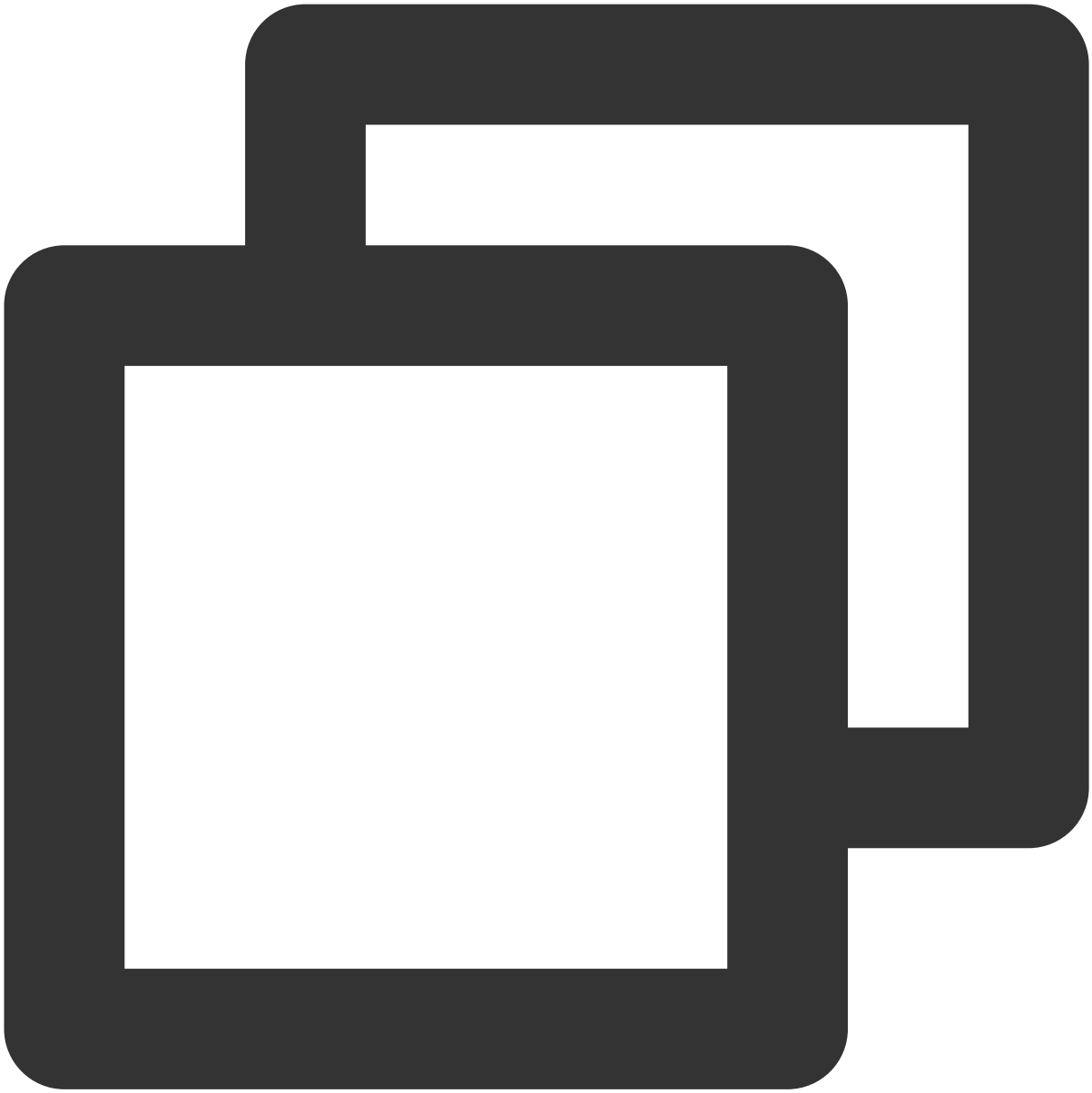
限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

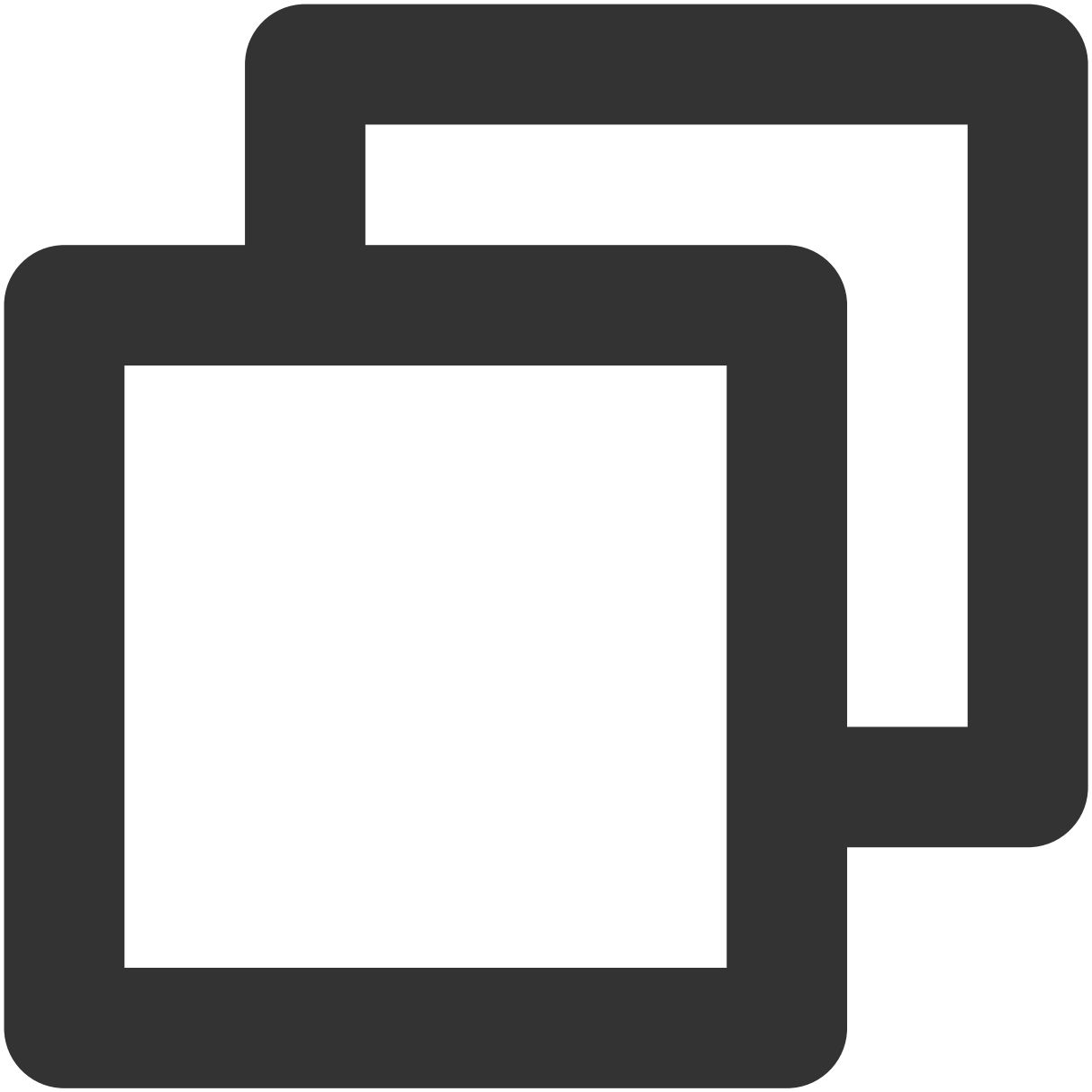
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
download_url?imageMogr2/sharpen/<value>
```

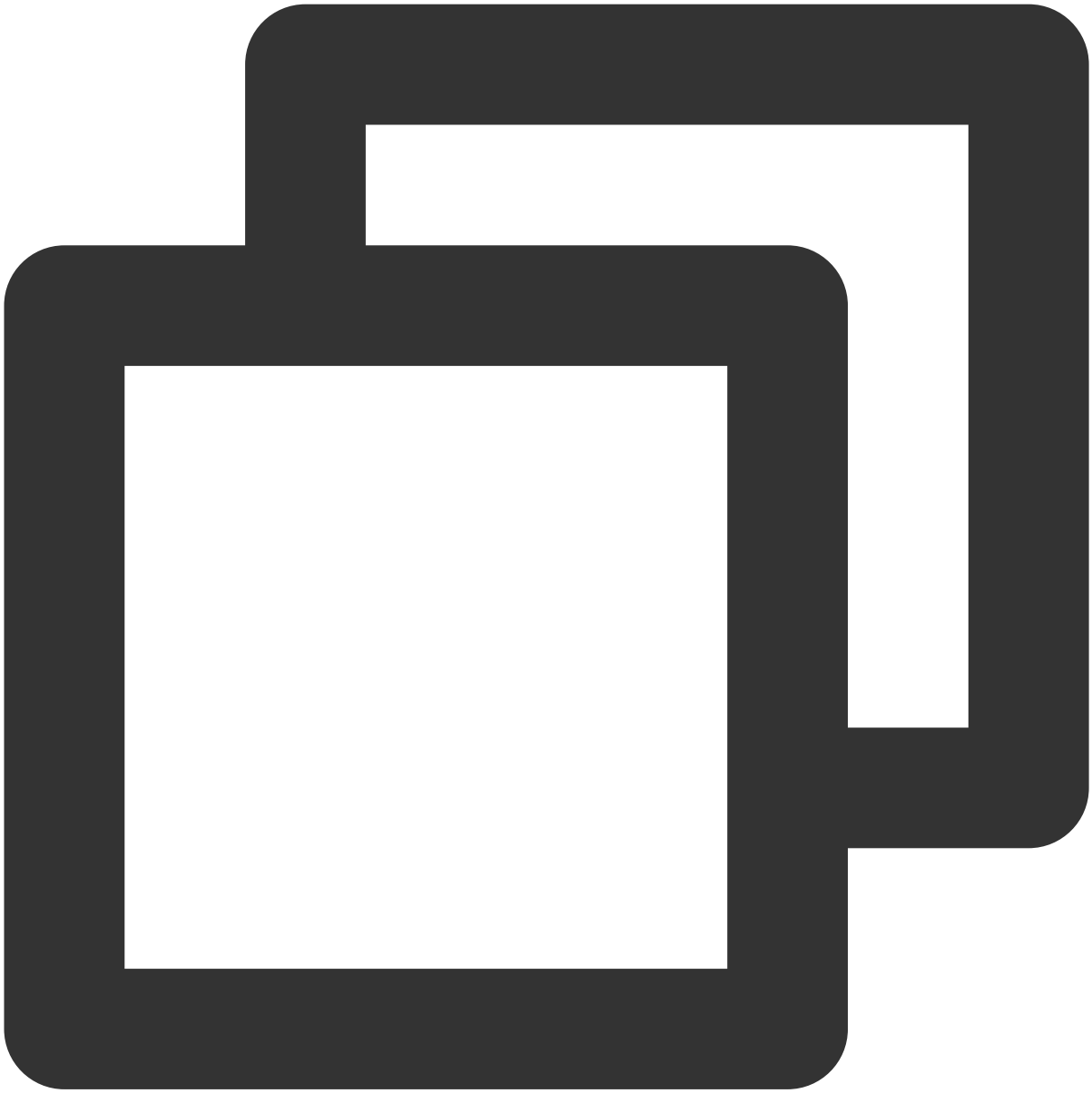
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/sharpen/<value>"
  }]
```

```
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
```

```
    "is_pic_info": 1,
    "rules": [{
        "fileid": "exampleobject",
        "rule": "imageMogr2/sharpen/<value>"
    }]
}
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

处理参数说明

操作名称：sharpen。

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/sharpen/<value>	图片锐化功能，value 为锐化参数值，取值范围为10 - 300间的整数（推荐使用70）。参数值越大，锐化效果越明显。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错。

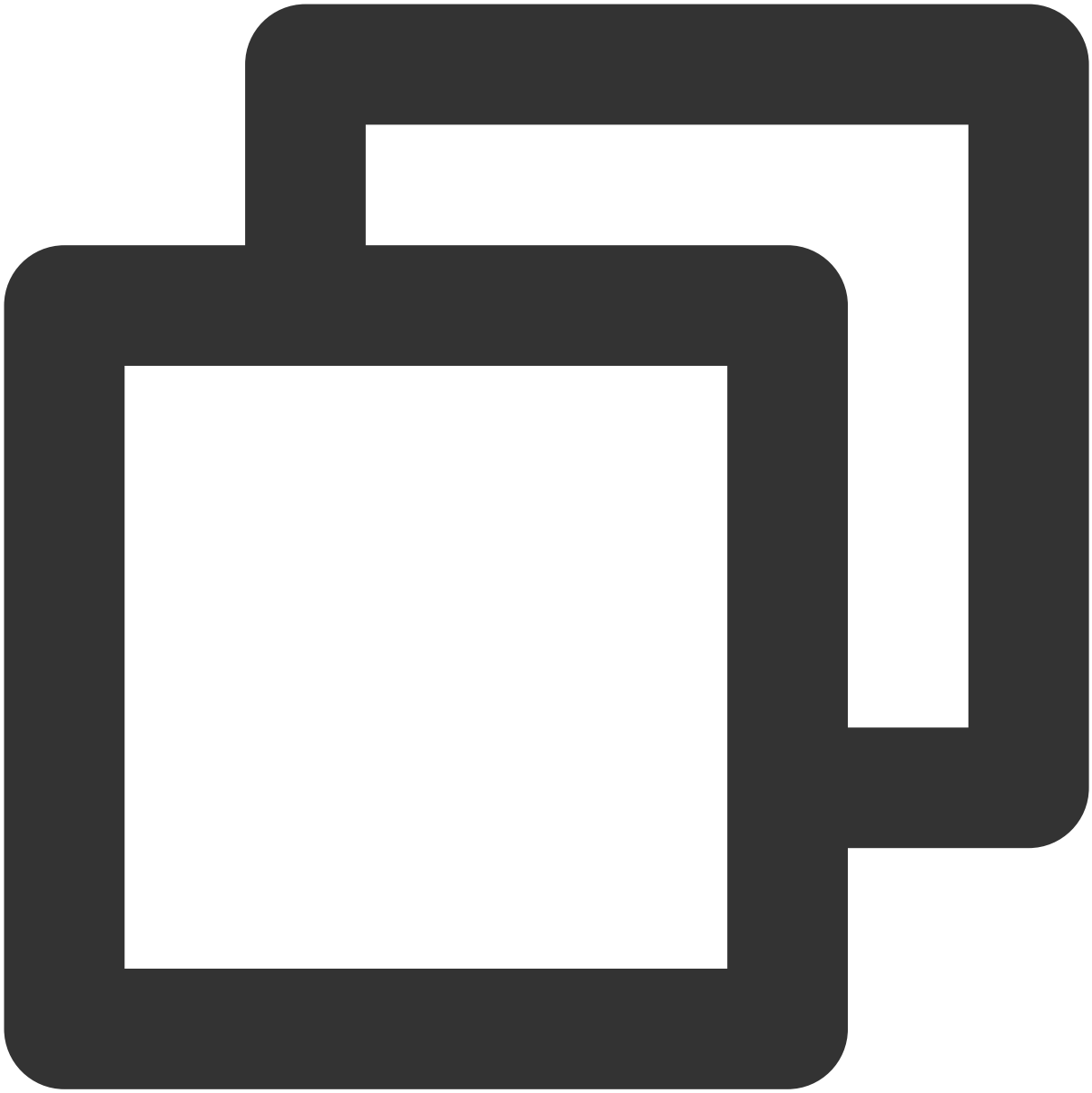
实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：锐化

假设设置锐化参数为70，进行锐化处理，示例如下：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/shar
```

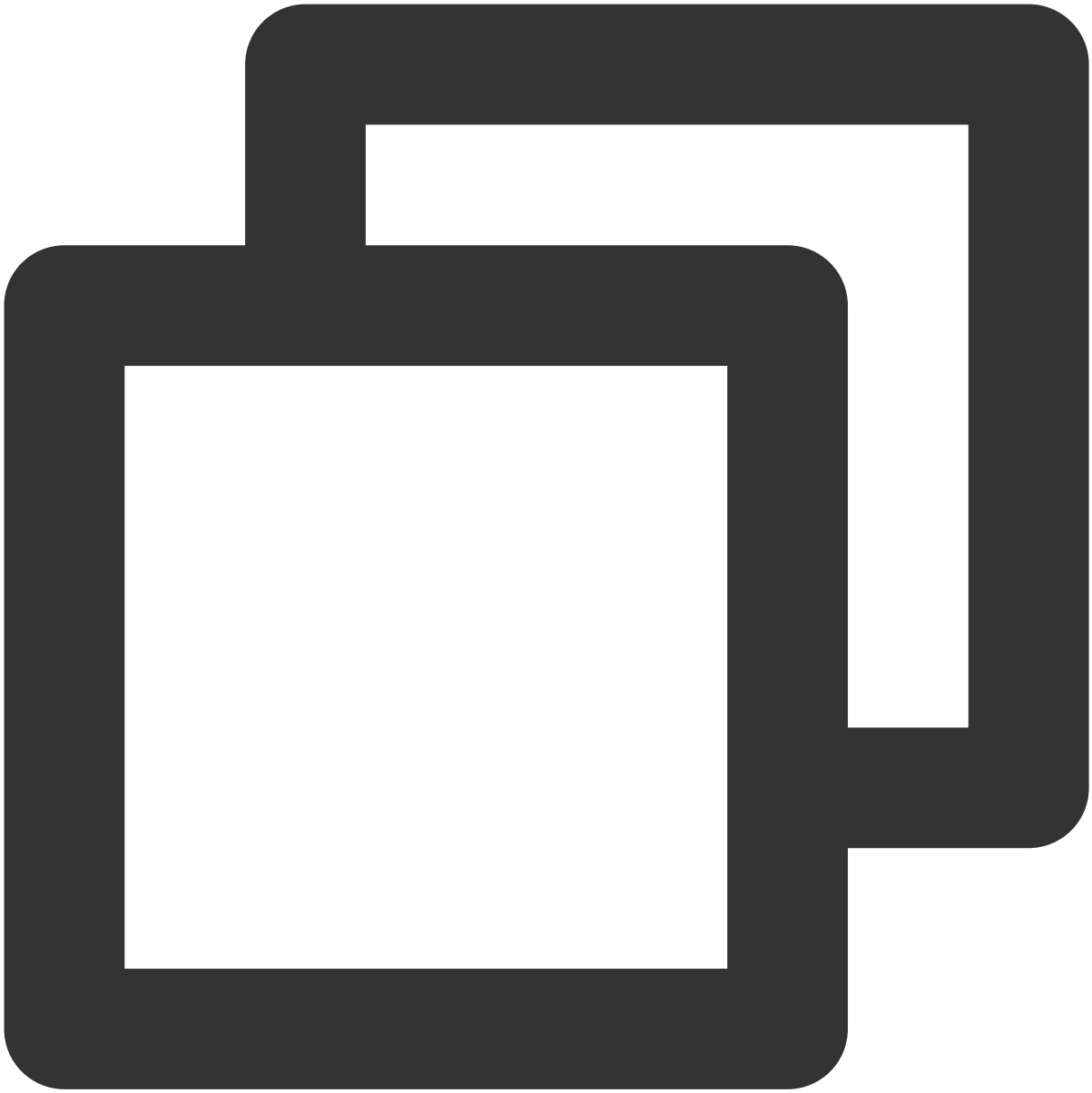
最终效果如下：



案例二：锐化并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：

版权所有：腾讯云计算（北京）有限责任公司



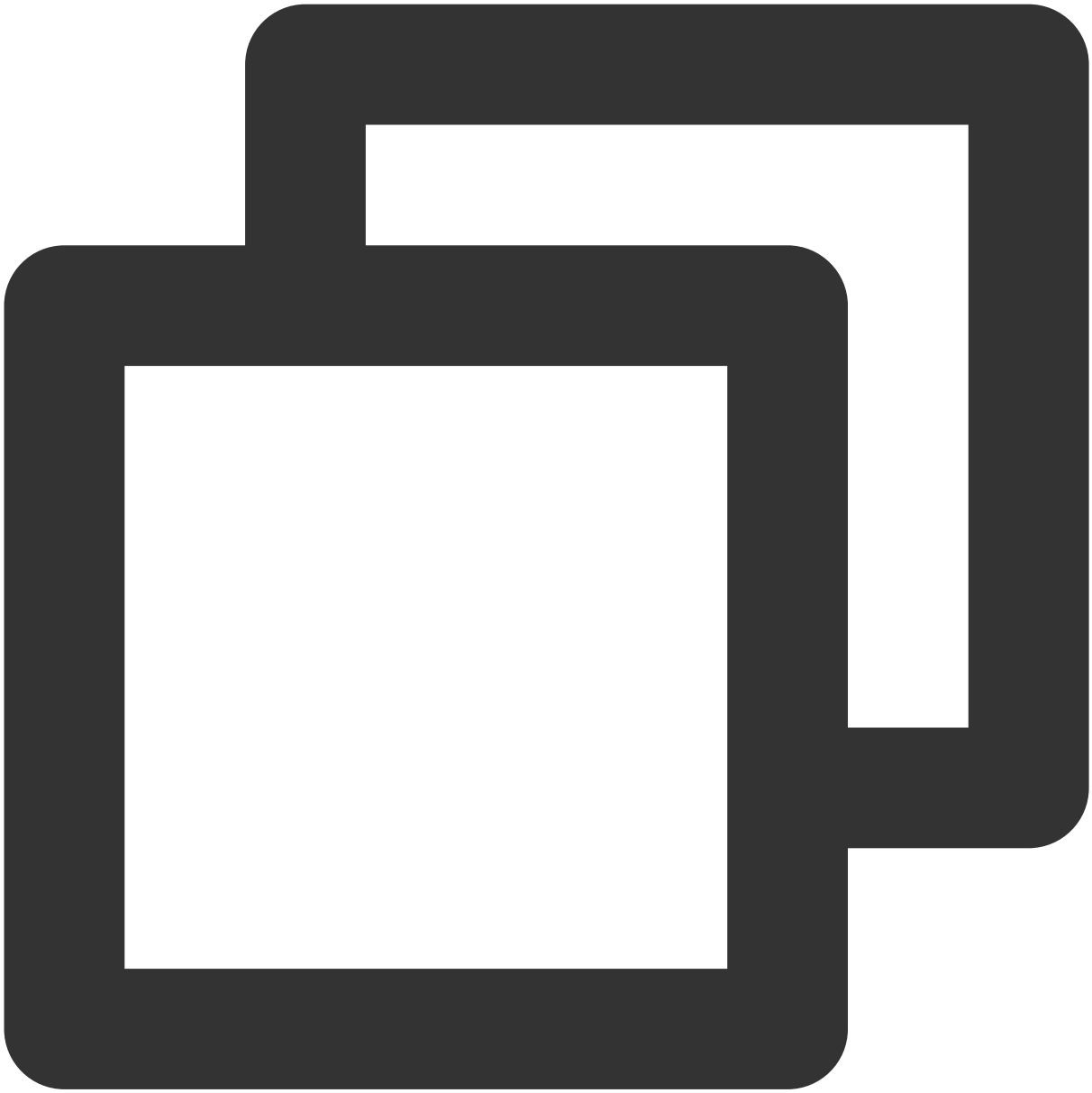
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

灰度图

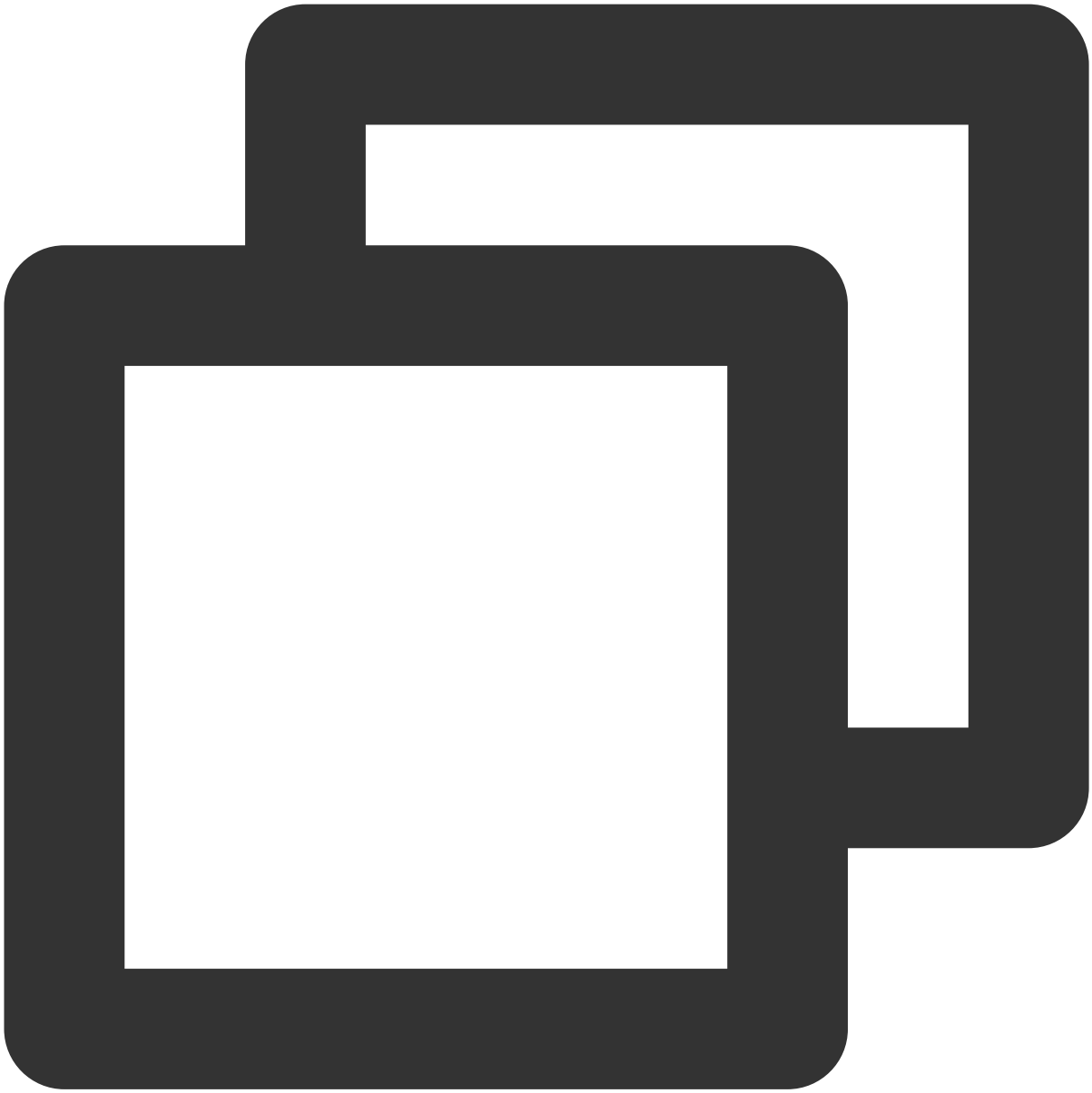
最近更新时间：2023-12-21 17:57:23

功能概述

对象存储通过数据万象 **imageMogr2/grayscale** 接口将图片设置为灰度图。

接口示例

版权所有：腾讯云计算（北京）有限责任公司



```
download_url?imageMogr2/grayscale/<value>
```

处理参数说明

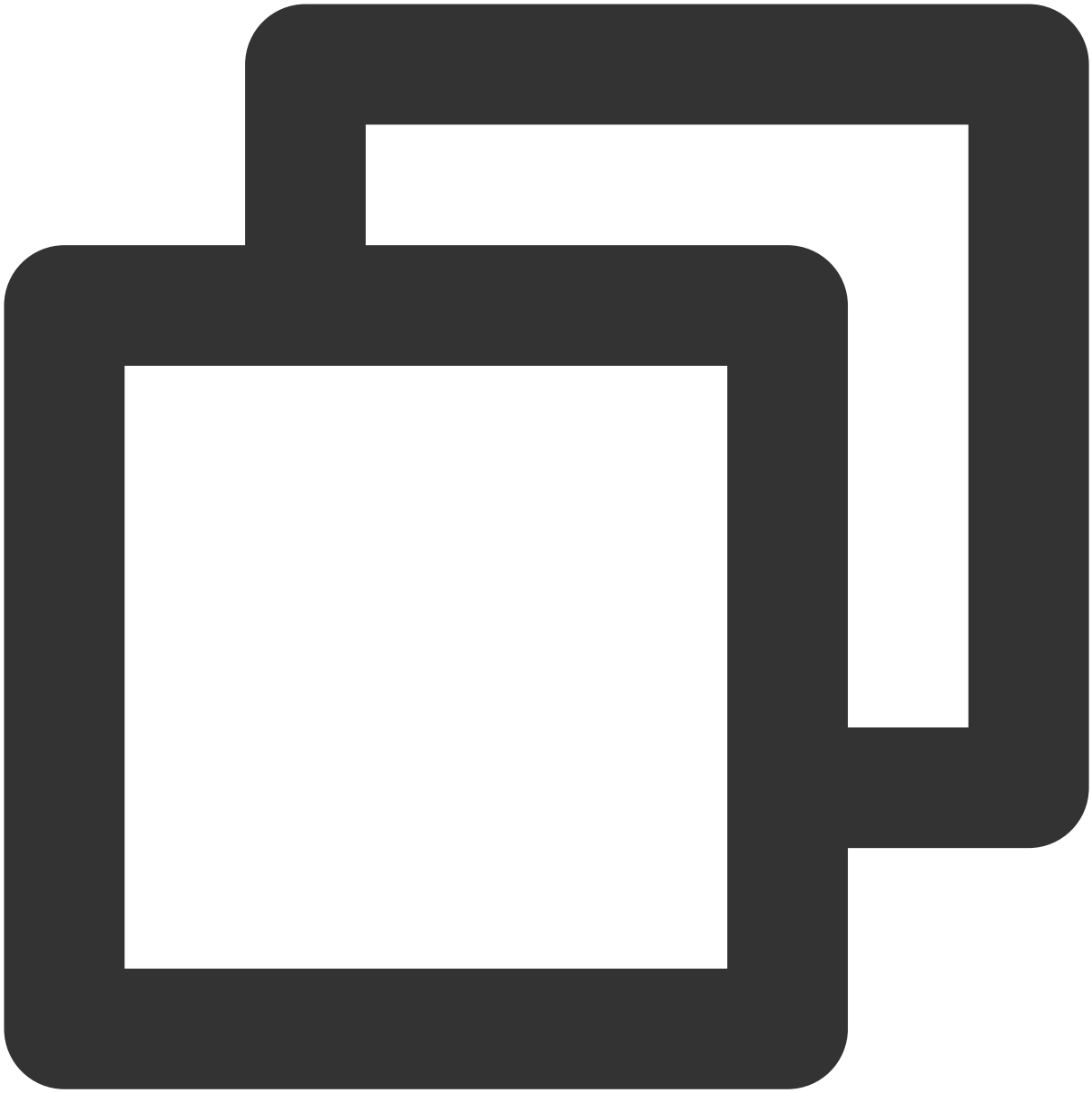
操作名称：grayscale。

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.

	<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/grayscale/<value>	将图片设置为灰度图。 value 取值为0表示不改变图片。 value 取值为1表示将图片变为灰度图。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错。

实际案例

将图片变为灰度图，示例如下：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/gray
```

最终效果如下：



版权所有：腾讯云计算（北京）有限责任公司

图片水印

最近更新时间：2023-12-21 17:57:28

功能概述

对象存储通过数据万象 **watermark** 接口提供图片水印处理功能。目前水印图片必须指定为已存储于数据万象绑定（或新建）的存储桶中的图片。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理

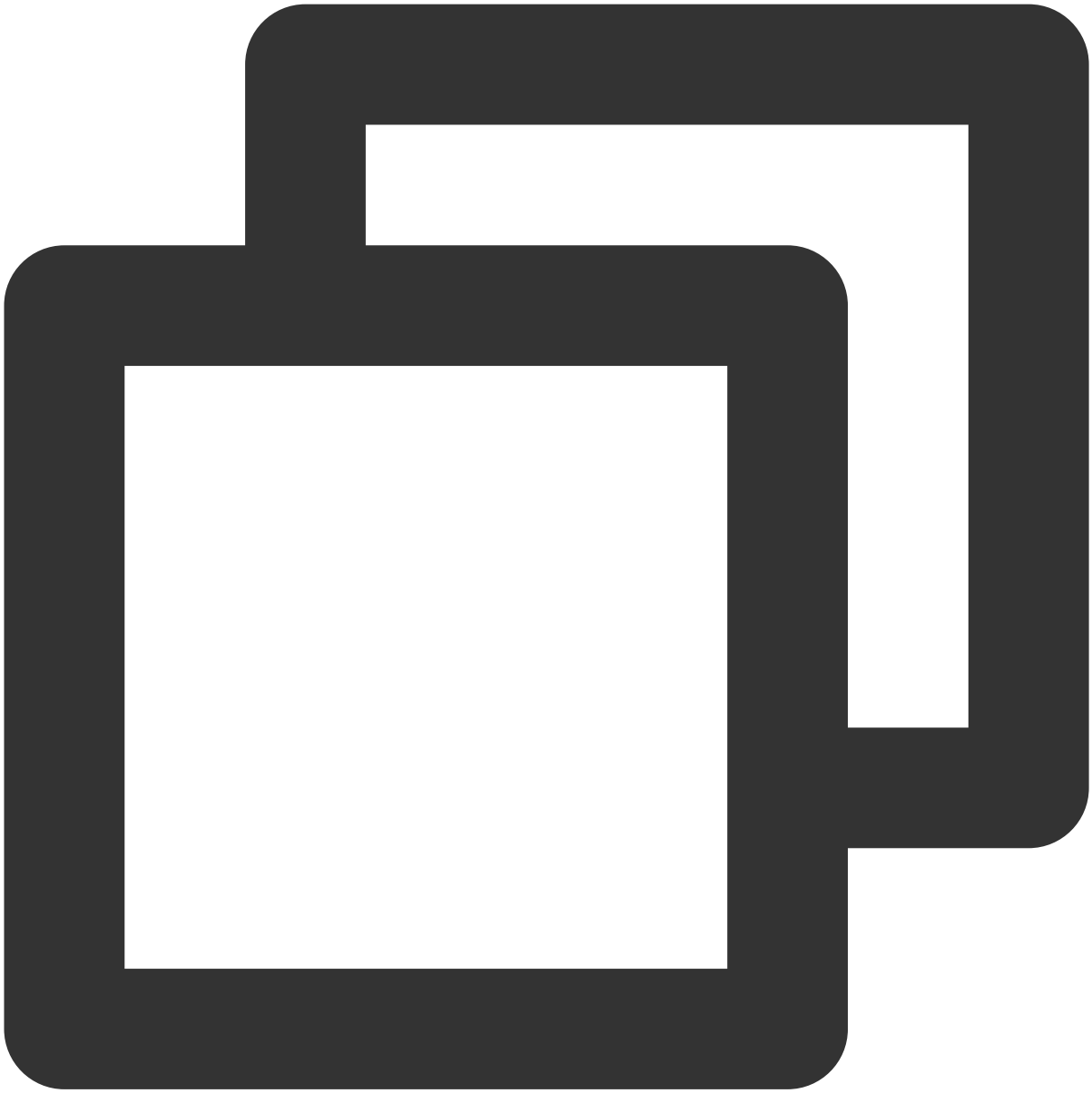
限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。
说明
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。
一张图片上，最多添加10张不同的图片水印。
不支持添加动图作为水印。

接口示例

1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司

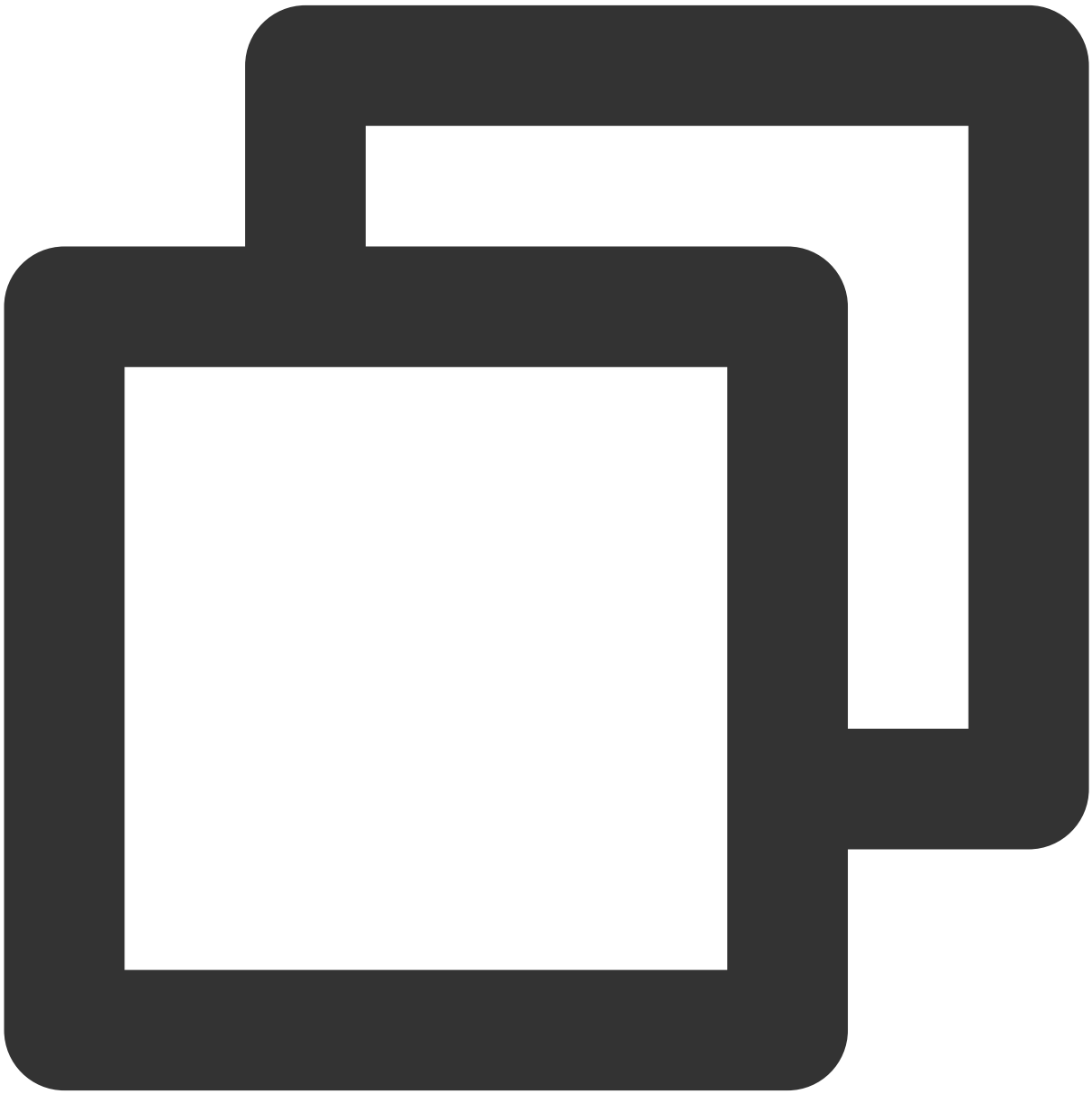


```
GET /<ObjectKey>?watermark/1/image/<encodedURL>
                        /gravity/<gravity>
                        /dx/<dx>
                        /dy/<dy>
                        /blogo/<type> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

请忽略上面的空格与换行符。

2. 上传时处理



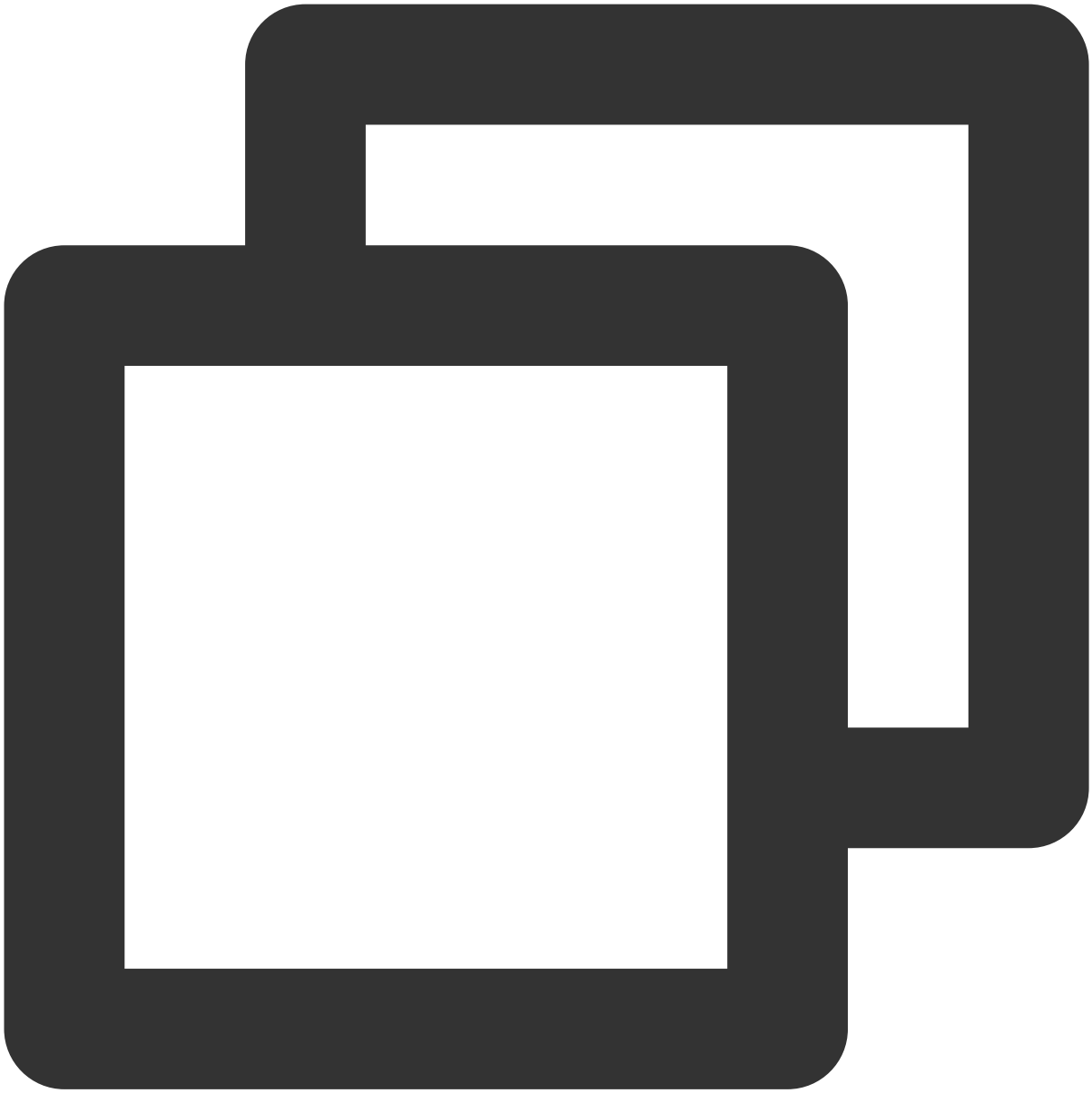
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
```

```
    "rules": [{
        "fileid": "exampleobject",
        "rule": "watermark/1/image/<encodedURL>
                            /gravity/<gravity>
                            /dx/<dx>
                            /dy/<dy>
                            /blogo/<type>"
    }]
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "watermark/1/image/<encodedURL>
```

```
                            /gravity/<gravity>
                            /dx/<dx>
                            /dy/<dy>
                            /blogo/<type>"
    }]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

操作名称：watermark，数字1，表示水印类型为图片水印。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/image/	水印图片地址，需要经过 URL 安全的 Base64 编码。例如，水印图片为 `http://examplebucke shanghai.myqcloud.com/shuiyin_2.png`，则该处编码后的字符串为 `aHR0cDovL2V4YW1wbGVidWNrZXQtMTI1MDAwMDAwMC5jb3MuYXAtc2hhbmdoYWkubXlxY2`
/gravity/	图片水印位置，九宫格位置（参考九宫格方位图），默认值为 SouthEast
/dx/	水平（横轴）边距，单位为像素，缺省值为0
/dy/	垂直（纵轴）边距，单位为像素，默认值为0
/blogo/	水印图适配功能，适用于水印图尺寸过大的场景（如水印墙）。一共有两种类型： 当 blogo 设置为1时，水印图会被缩放至与原图相似大小后添加 当 blogo 设置为2时，水印图会被直接裁剪至与原图相似大小后添加
/scatype/	根据原图的大小，缩放调整水印图的大小： 当 scatype 设置为1时，按原图的宽缩放 当 scatype 设置为2时，按原图的高缩放 当 scatype 设置为3时，按原图的整体面积缩放
/spcent/	与 scatype 搭配使用： 当 scatype 设置为1时，该有效值为[1, 1000]，单位为千分比 当 scatype 设置为2时，该有效值为[1, 1000]，单位为千分比 当 scatype 设置为3时，该有效值为[1, 250]，单位为千分比。 示例：`http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpe watermark/1/image/xxxxxxx/scatype/3/spcent/250`
/dissolve/	图片水印的透明度，取值为1 - 100，默认90（90%不透明度）

/batch/	平铺水印功能，可将图片水印平铺至整张图片。值为1时，表示开启平铺水印功能
/degree/	当 batch 值为1时生效。图片水印的旋转角度设置，取值范围为0 - 360，默认0

注意

指定的水印图片必须同时满足如下3个条件：

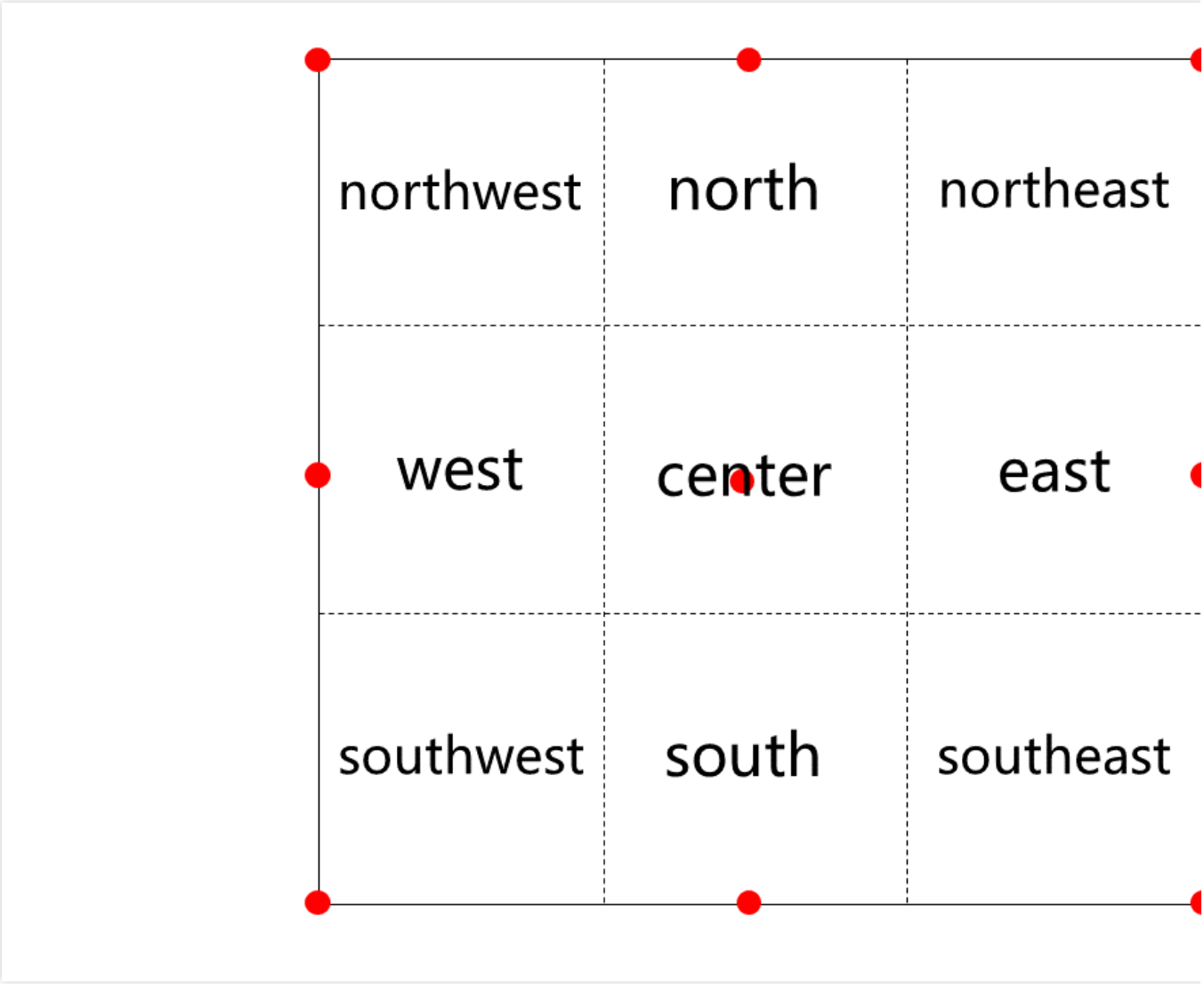
水印图片与源图片必须位于同一个存储桶下。

URL 需使用 COS 域名（不能使用 CDN 加速域名，例如 `examplebucket-1250000000.file.myqcloud.com/shuiyin_2.png` 不可用），且需保证水印图可访问（如果水印图读取权限为私有，则需要携带有效签名）。

URL 必须以 `http://` 开始，不能省略 HTTP 头，也不能填 HTTPS 头，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/shuiyin_2.png`，`https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/shuiyin_2.png` 为非法的水印 URL。

九宫格方位图

九宫格方位图可为图片的多种操作提供位置参考。红点为各区域位置的原点（通过 gravity 参数选定各区域后位移操作会以相应远点为参照）。



说明

当 gravity 参数设置为 center 时，dx、dy 参数无效。

当 gravity 参数设置为 north 或 south 时，dx 参数无效（水印会水平居中）。

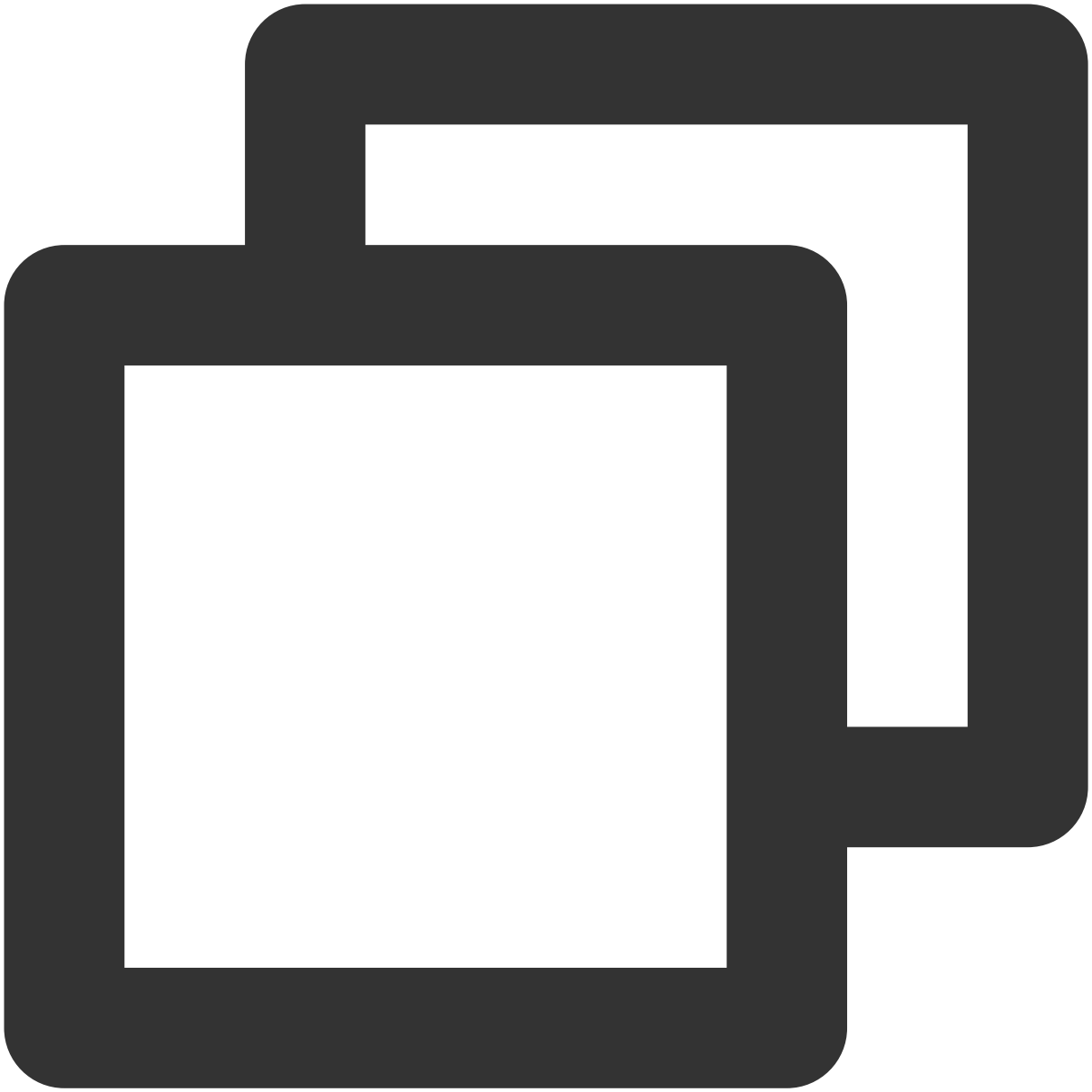
当 gravity 参数设置为 west 或 east 时，dy 参数无效（水印会垂直居中）。

实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：添加图片水印

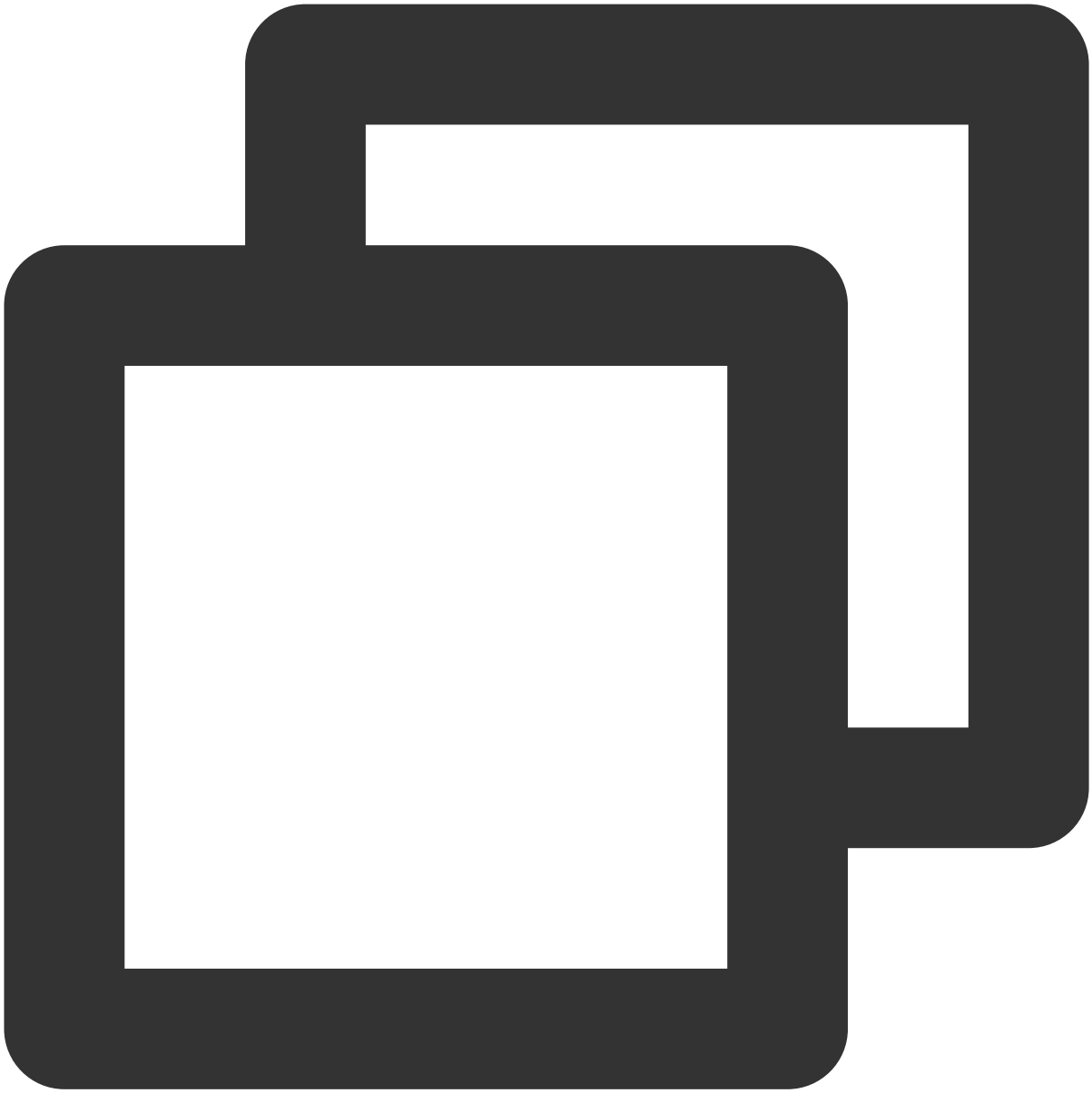


```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/1/ima
```

添加图片水印后效果如下：



案例二：添加图片水印并设置平铺、透明度



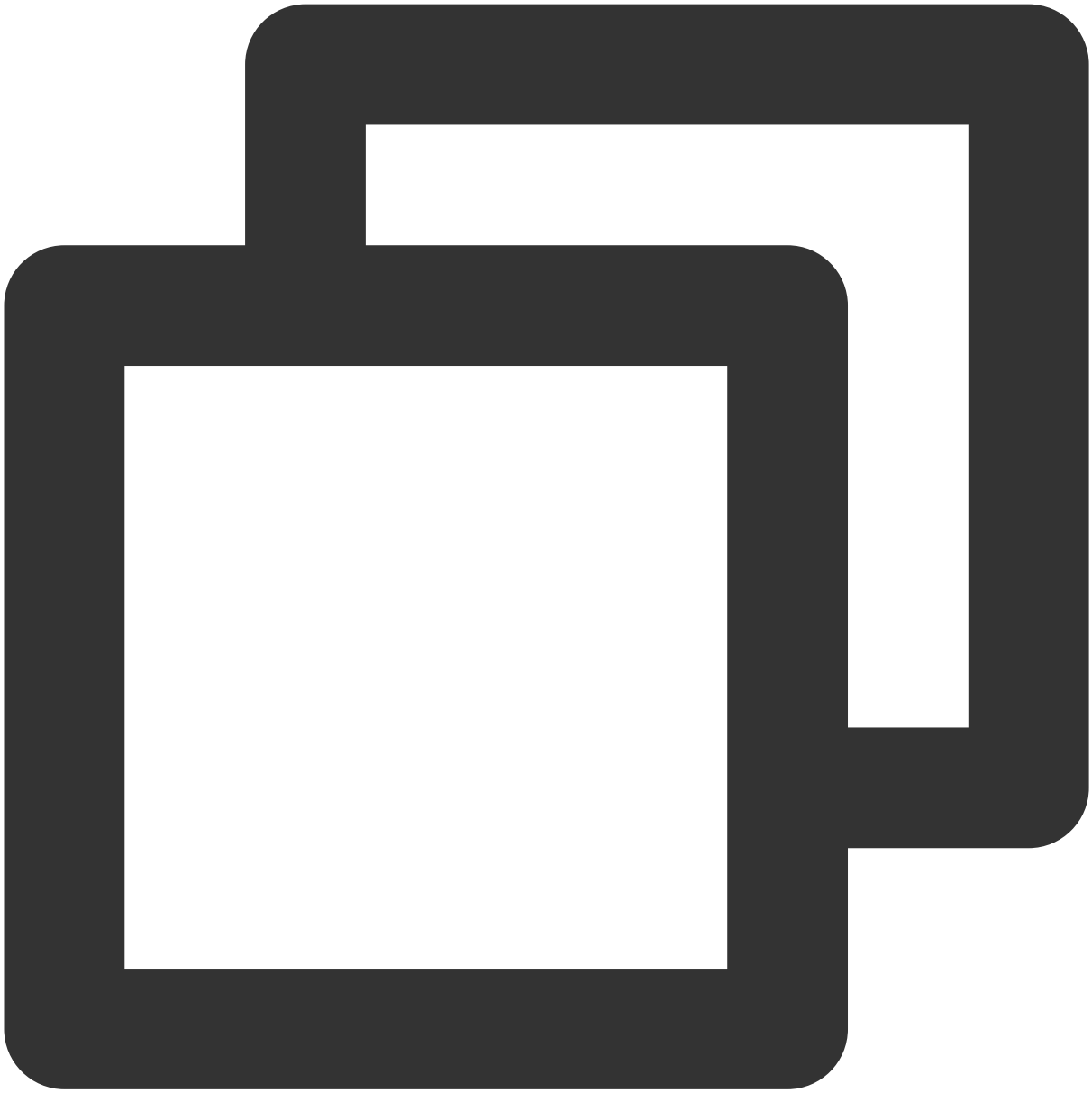
```
https://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/1/im
```

添加图片水印并设置旋转45度平铺、透明度为40%后，效果如下：



案例三：添加图片水印并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



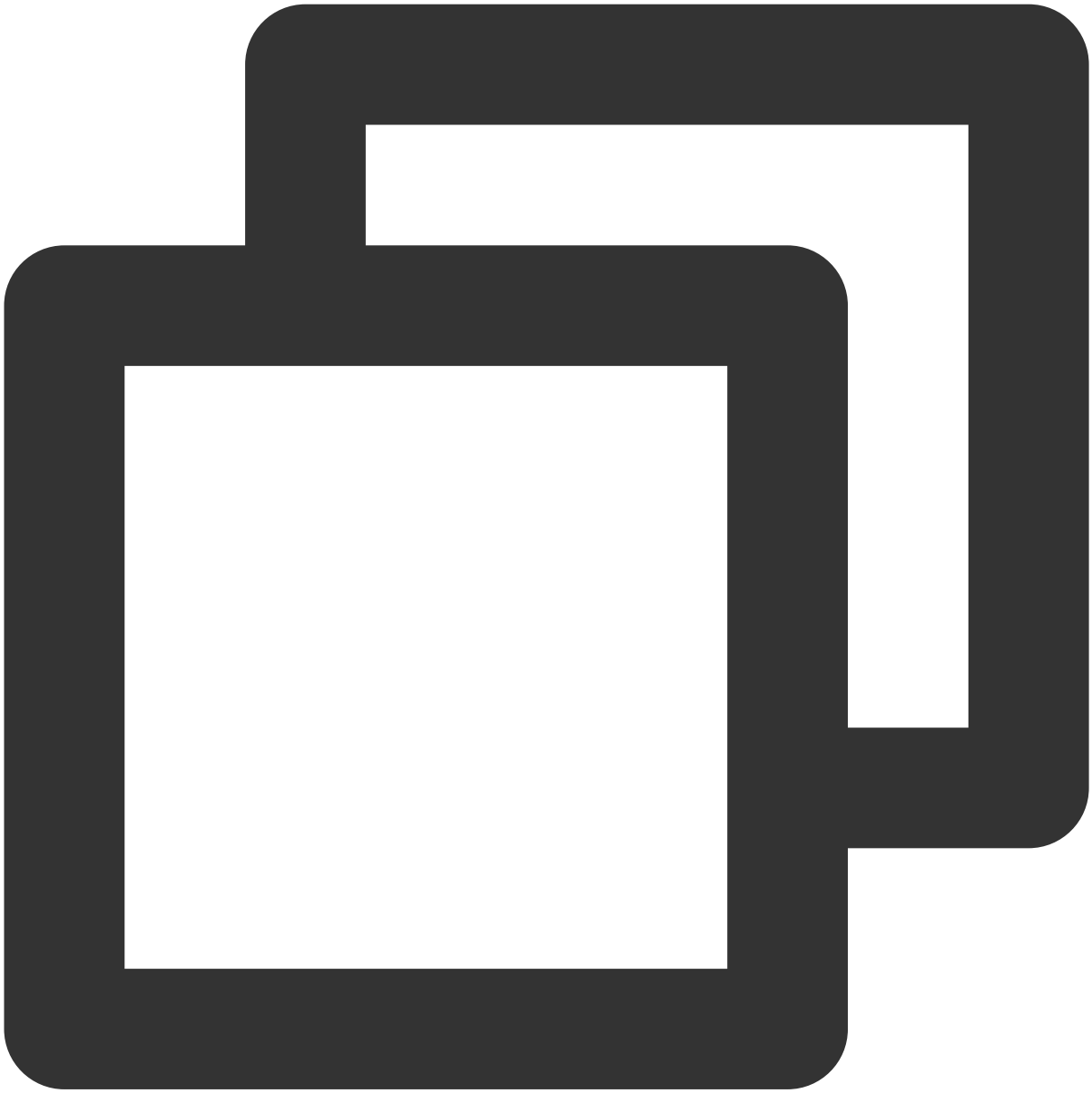
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

<signature>为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

文字水印

最近更新时间：2023-12-21 17:57:53

功能概述

对象存储通过数据万象 **watermark** 接口提供实时文字水印处理功能。

注意

图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。

该功能支持以下的处理方式：

下载时处理

上传时处理

云上数据处理

限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。

静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。

WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。

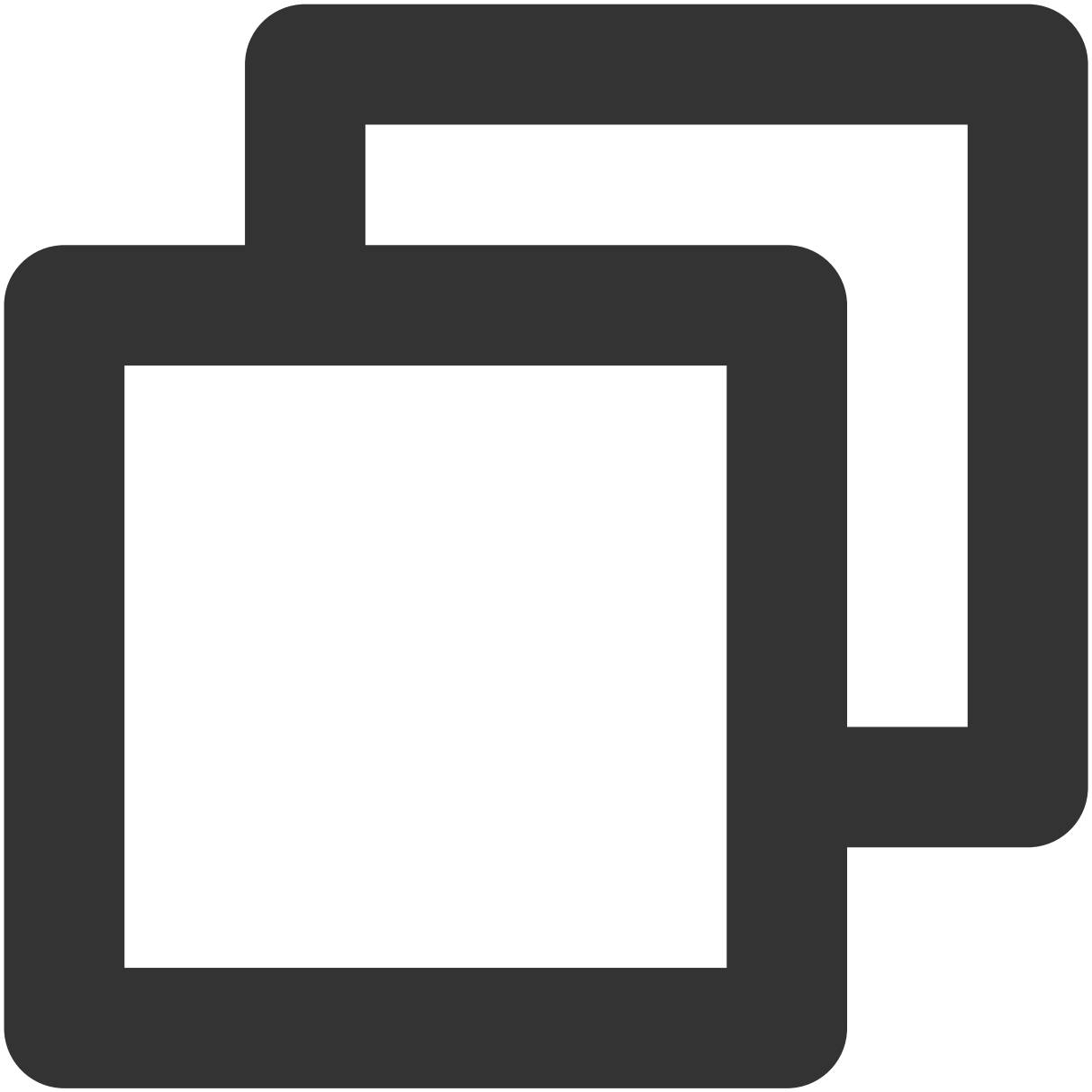
动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。

动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?watermark/2/text/<encodedText>
                        /font/<encodedFont>
                        /fontsize/<fontSize>
                        /fill/<encodedColor>
                        /dissolve/<dissolve>
                        /gravity/<gravity>
                        /dx/<dx>
                        /dy/<dy>
                        /batch/<type>
                        /degree/<degree>
                        /shadow/<shadow> HTTP/1.1
```

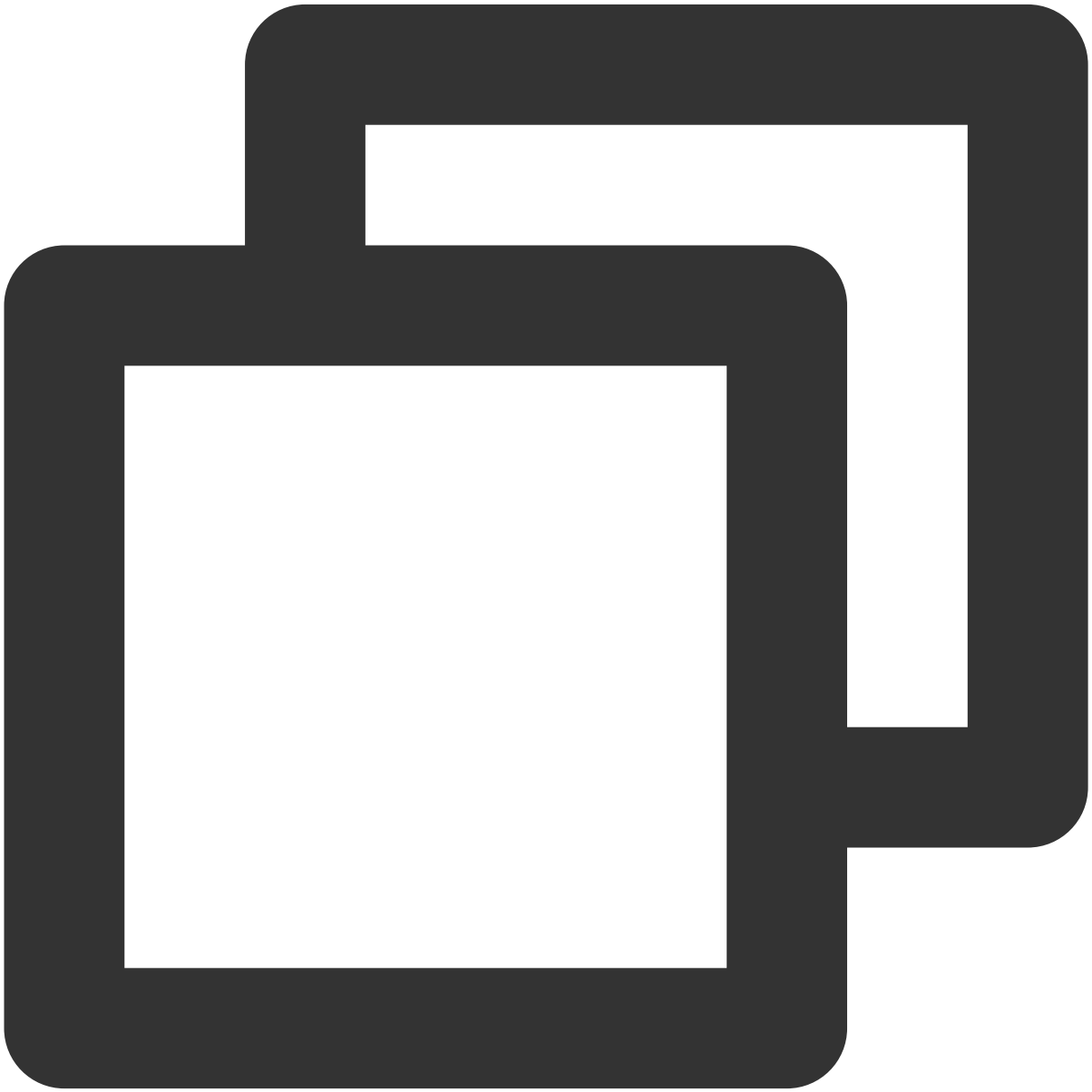
版权所有：腾讯云计算（北京）有限责任公司

```
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

请忽略上面的空格与换行符。

2. 上传时处理



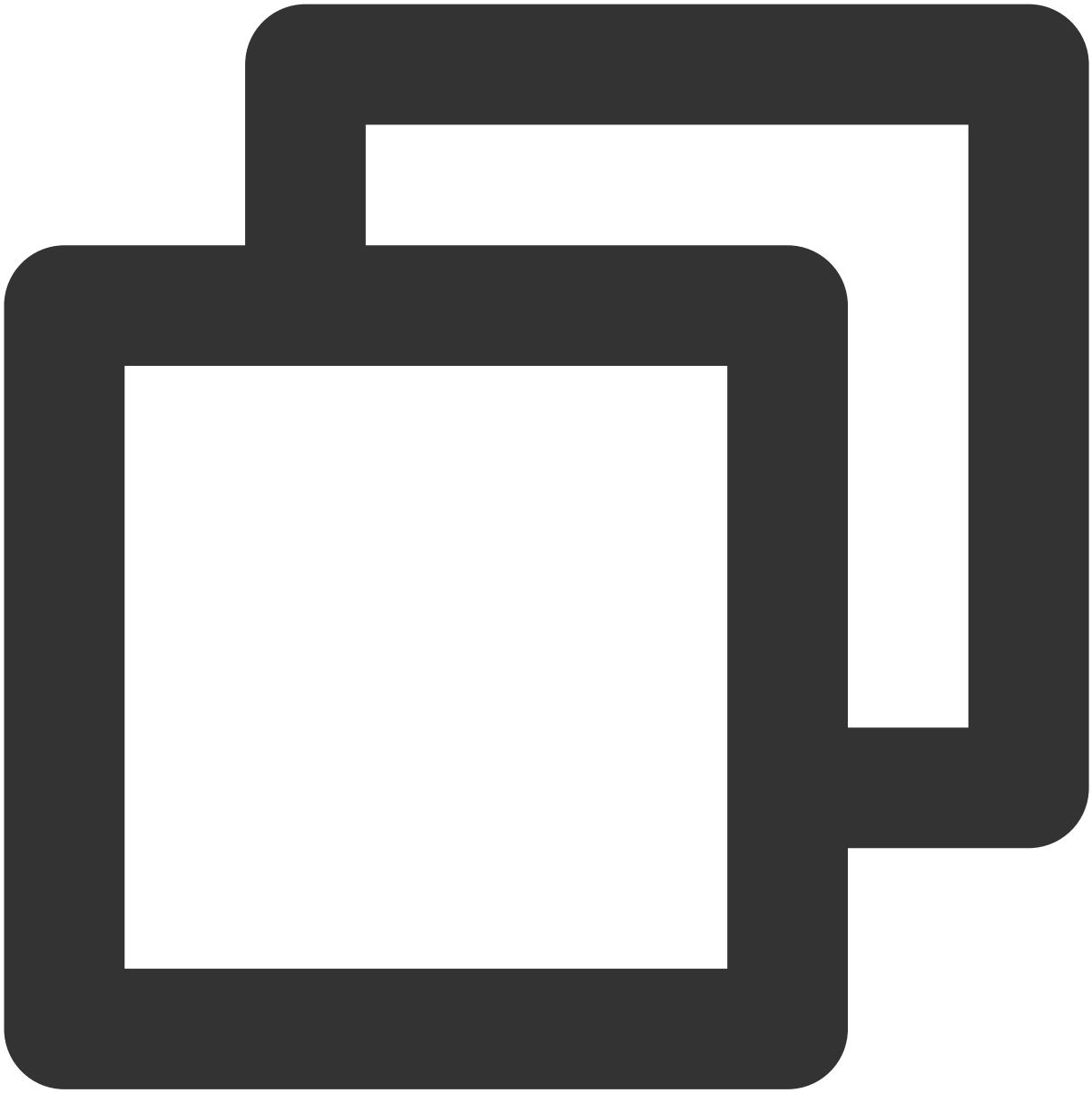
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
```

```
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "watermark/2/text/<encodedText>
                        /font/<encodedFont>
                        /fontsize/<fontSize>
                        /fill/<encodedColor>
                        /dissolve/<dissolve>
                        /gravity/<gravity>
                        /dx/<dx>
                        /dy/<dy>
                        /batch/<type>
                        /degree/<degree>
                        /shadow/<shadow>"
  }]
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "watermark/2/text/<encodedText>
```

```
                        /font/<encodedFont>
                        /fontsize/<fontSize>
                        /fill/<encodedColor>
                        /dissolve/<dissolve>
                        /gravity/<gravity>
                        /dx/<dx>
                        /dy/<dy>
                        /batch/<type>
                        /degree/<degree>
                        /shadow/<shadow>"
    }]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

操作名称：watermark，数字2，表示水印类型为文字水印。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/text/	水印内容，需要经过 URL 安全的 Base64 编码
/font/	水印字体，需要经过 URL 安全的 Base64 编码，默认值 tahoma.ttf 。水印字体列表参考 支持字体列表
/fontsize/	水印文字字体大小，单位为磅，缺省值13。如需按照原图片的大小比例缩放文字水印，请将文字水印转换为 PNG 图片，并参考 图片水印 文档进行配置
/fill/	字体颜色，缺省为灰色，需设置为十六进制 RGB 格式（例如 #FF0000），详情可参考 RGB 编码表，需经过 URL 安全的 Base64 编码，默认值为 #3D3D3D
/dissolve/	文字透明度，取值1 - 100，默认90（90%不透明度）
/gravity/	文字水印位置，九宫格位置（参见九宫格方位图），默认值 SouthEast
/dx/	水平（横轴）边距，单位为像素，缺省值为0
/dy/	垂直（纵轴）边距，单位为像素，默认值为0
/batch/	平铺水印功能，可将文字水印平铺至整张图片。值为1时，表示开启平铺水印功能

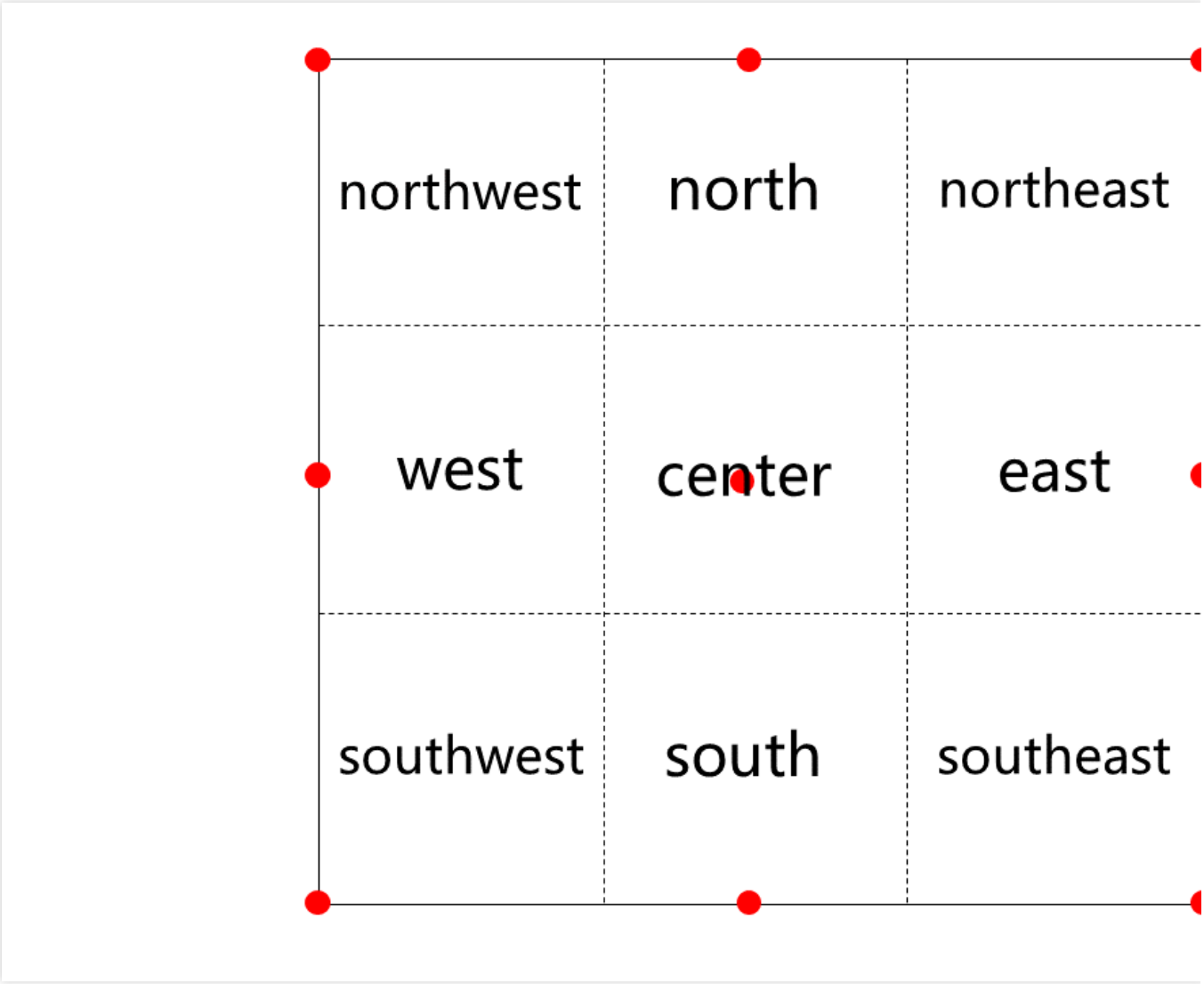
版权所有：腾讯云计算（北京）有限责任公司

/spacing/	平铺模式下的水平、垂直间距相对文字水印贴图的宽高百分比，范围为[0,100]，默认10
/degree/	当 batch 值为1时生效。文字水印的旋转角度设置，取值范围为0 - 360，默认0
/shadow/	文字阴影效果，有效值为[0,100]，默认为0，表示无阴影
/scatype/	根据原图的大小，缩放调整文字水印的大小： 当 scatype 设置为1时，按原图的宽缩放 当 scatype 设置为2时，按原图的高缩放 当 scatype 设置为3时，按原图的整体面积缩放
/spcent/	与 scatype 搭配使用： 当 scatype 设置为1时，该有效值为[1, 1000]，单位为千分比 当 scatype 设置为2时，该有效值为[1, 1000]，单位为千分比 当 scatype 设置为3时，该有效值为[1, 250]，单位为千分比。

九宫格方位图

九宫格方位图可为图片的多种操作提供位置参考。红点为各区域位置的原点（通过 gravity 参数选定各区域后位移操作会以相应远点为参照）。

版权所有：腾讯云计算（北京）有限责任公司



注意

当 gravity 参数设置为 center 时，dx、dy 参数无效。

当 gravity 参数设置为 north 或 south 时，dx 参数无效（水印会水平居中）。

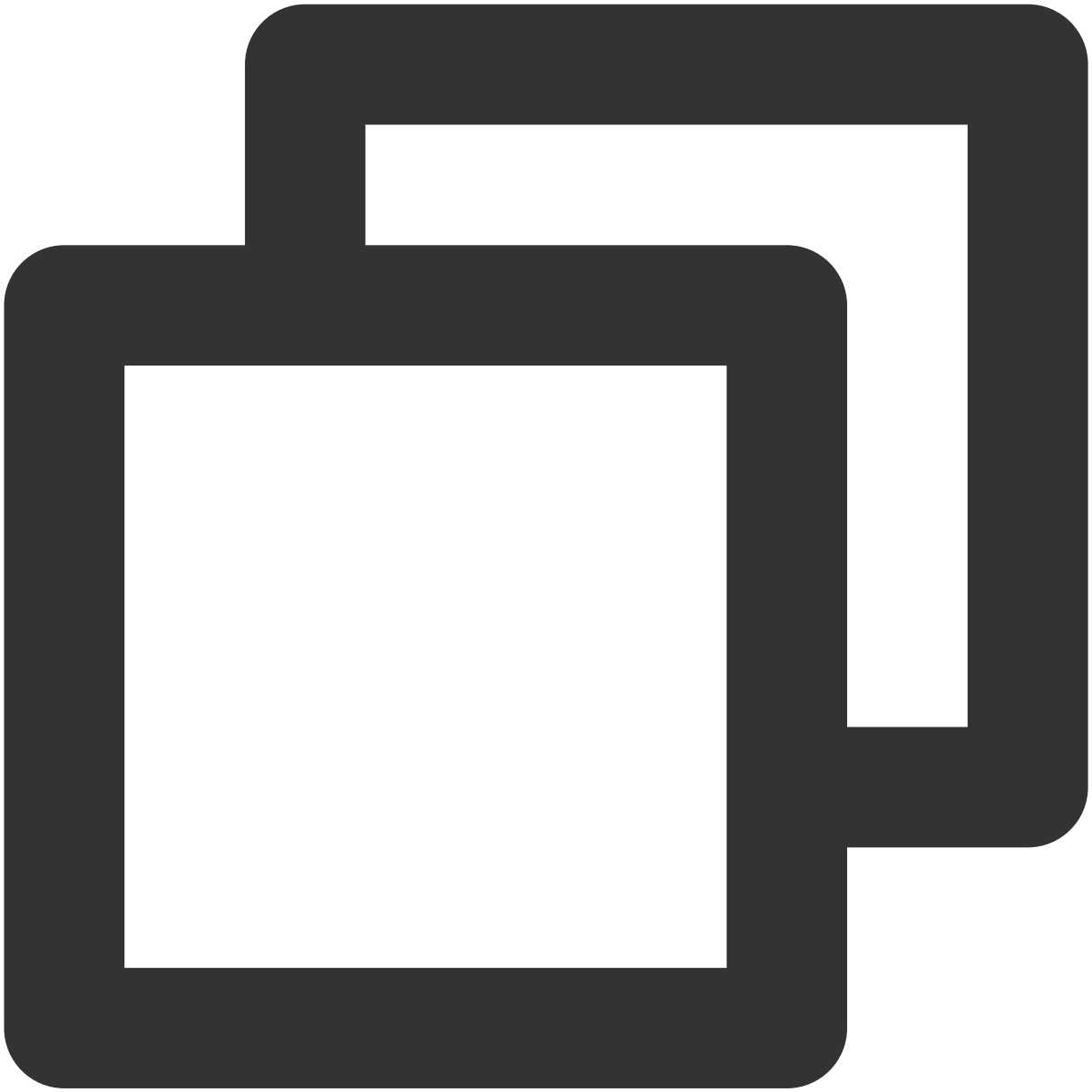
当 gravity 参数设置为 west 或 east 时，dy 参数无效（水印会垂直居中）。

实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：添加文字水印



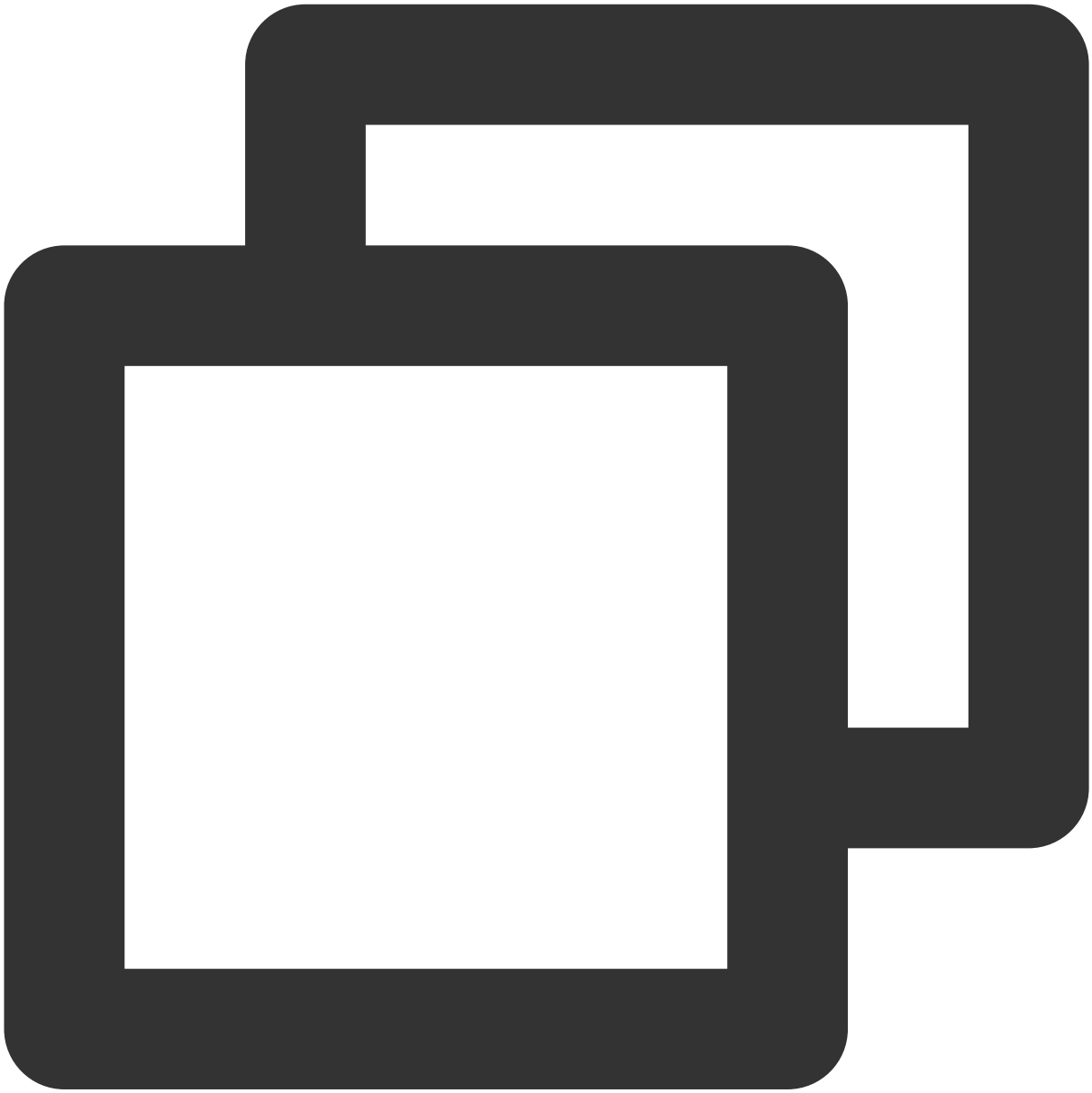
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/2/tex
```

添加文字水印后，效果如下：



案例二：添加文字水印并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



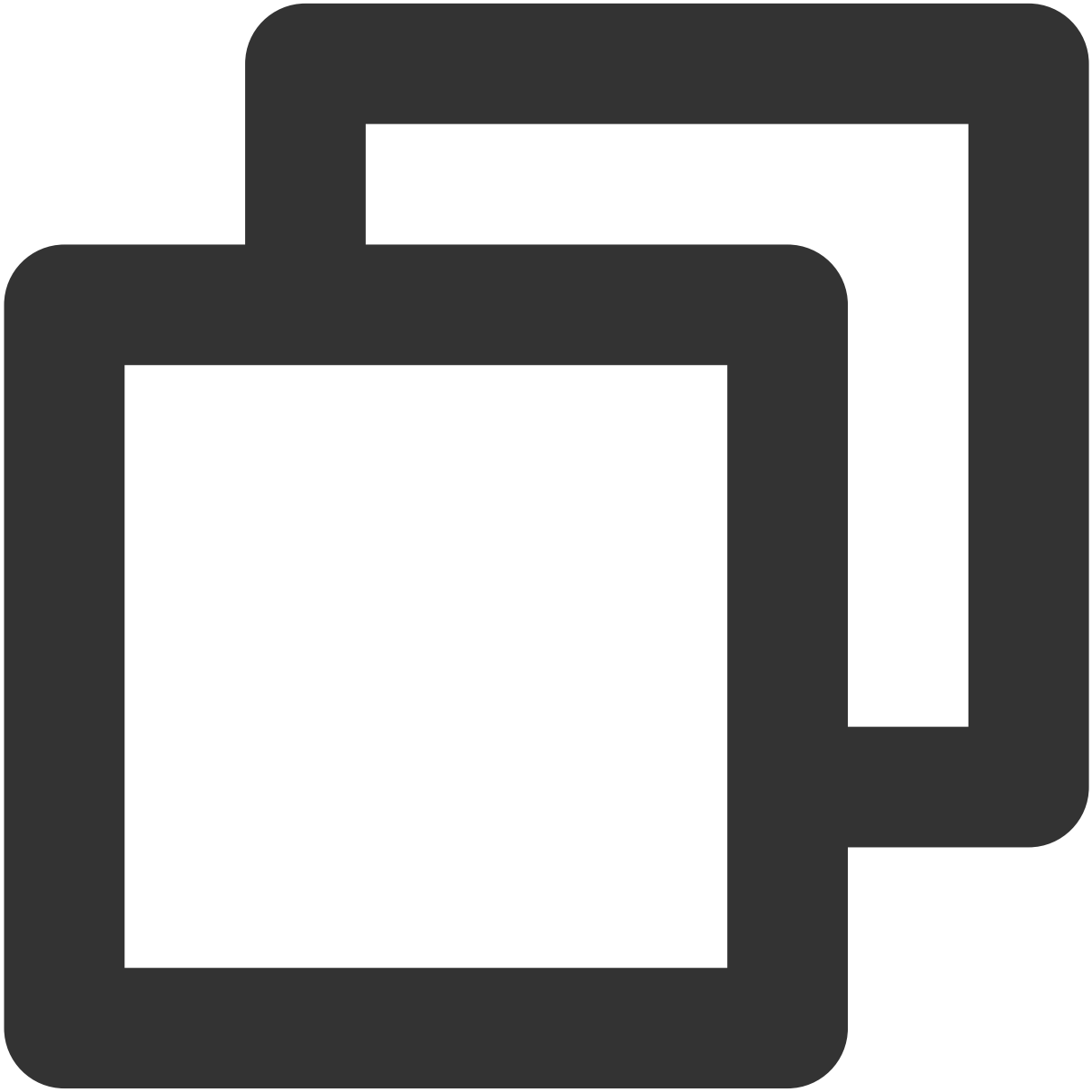
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

获取图片基本信息

最近更新时间：2023-12-21 17:57:47

功能概述

对象存储通过数据万象 **imageInfo** 接口查询图片基本信息，包括格式、长、宽等。

注意

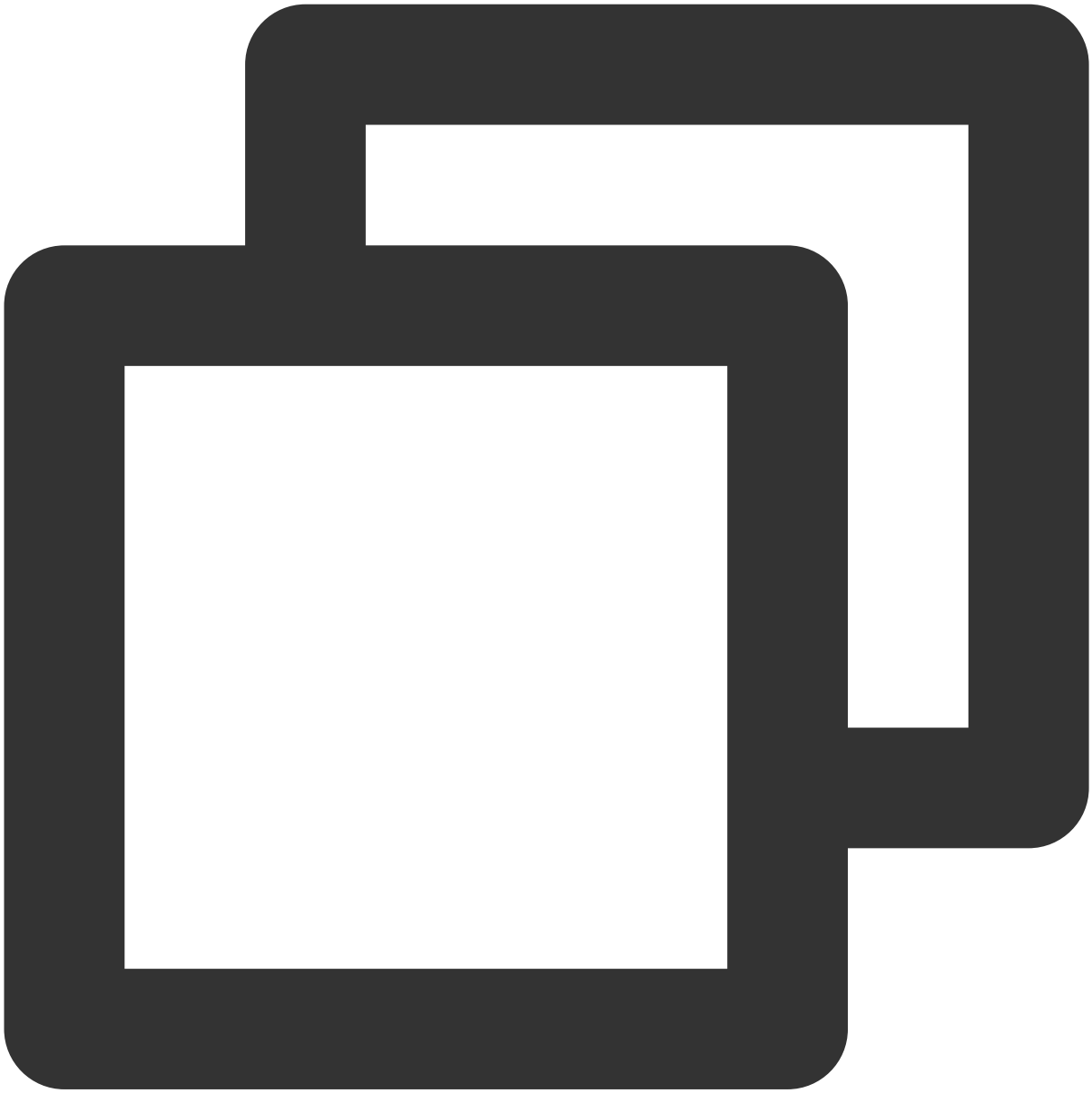
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。

使用限制

待处理的图片，原图大小不超过32MB。
宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素。
针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。

接口示例

版权所有：腾讯云计算（北京）有限责任公司



```
download_url?imageInfo
```

处理参数说明

请求参数

操作名称：imageInfo。

参数	含义

download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`

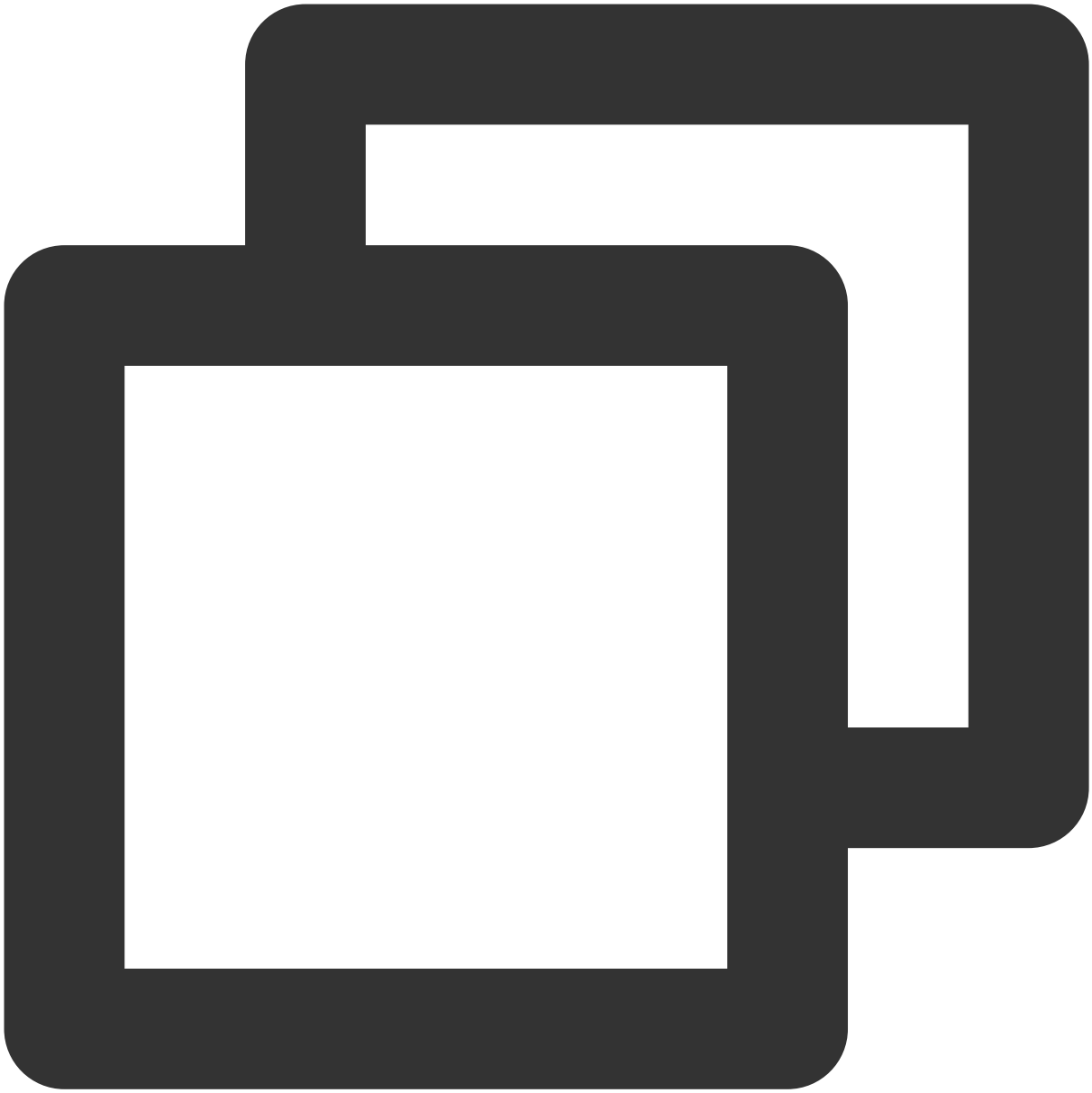
响应参数

参数	含义
format	图片类型。例如 png、gif 等
width	图片的宽度。单位为像素（px）
height	图片的高度。单位为像素（px）
size	图片的大小。单位为 Bytes
md5	图片的 md5 值
frame_count	图片的帧数。静态图为1，动图为对应的帧数

实际案例

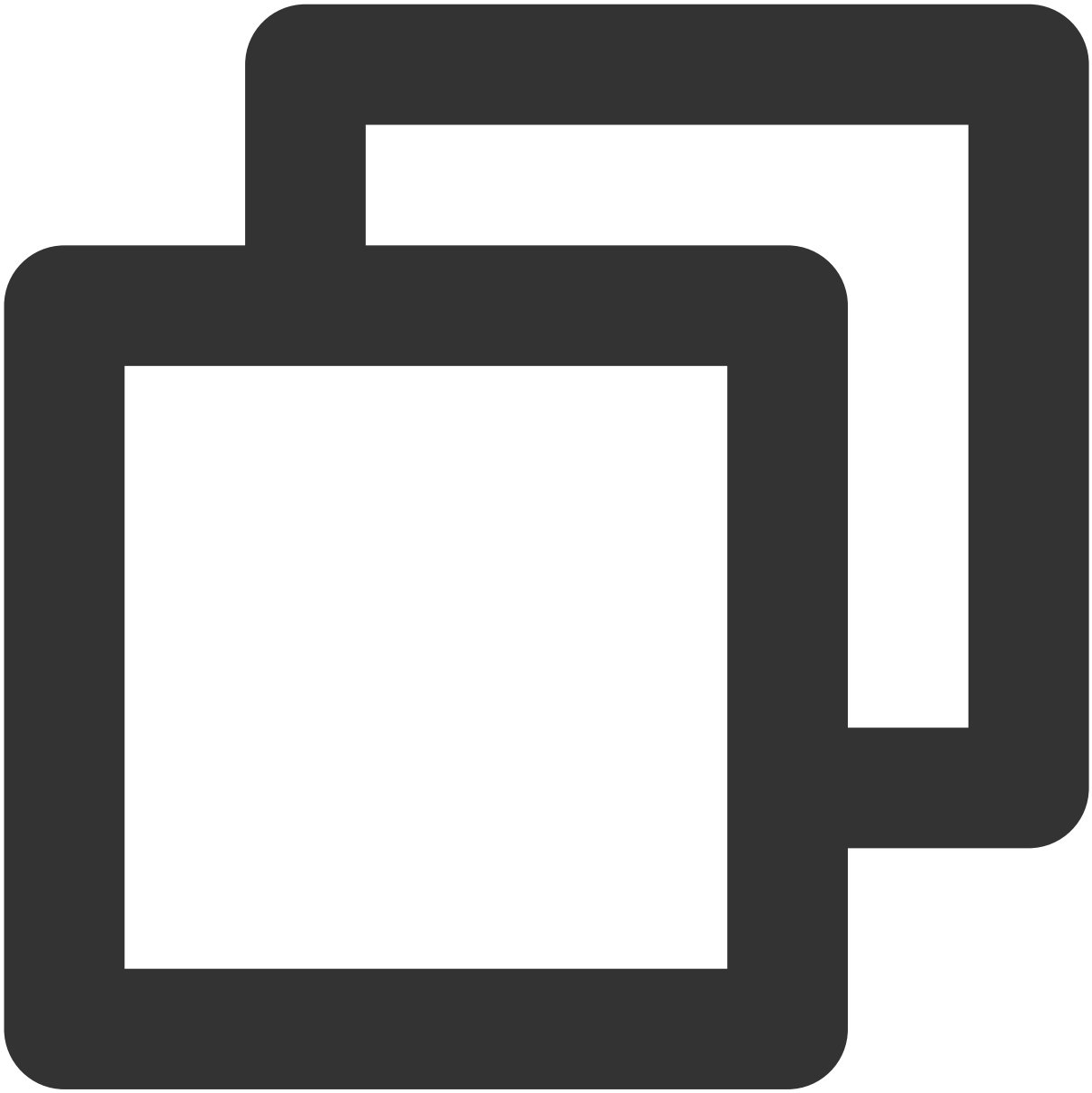
案例一：公有读

请求



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageInfo
```

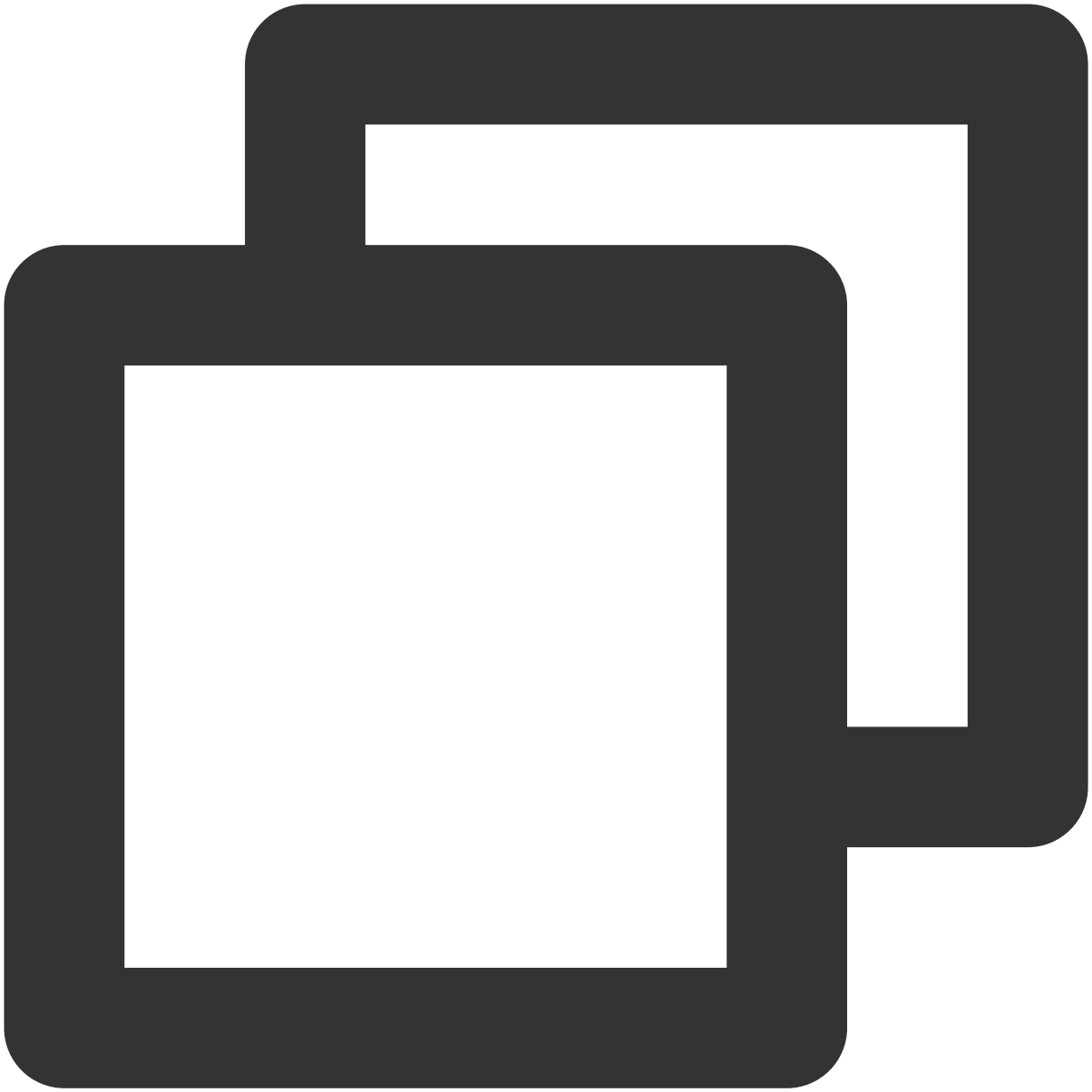
响应



```
{"format": "jpeg", "width": "960", "height": "540", "size": "158421", "md5": "77a16
```

案例二：私有读、携带签名

获取方式同上，仅增加签名部分，并与获取参数以“&”连接，示例如下：



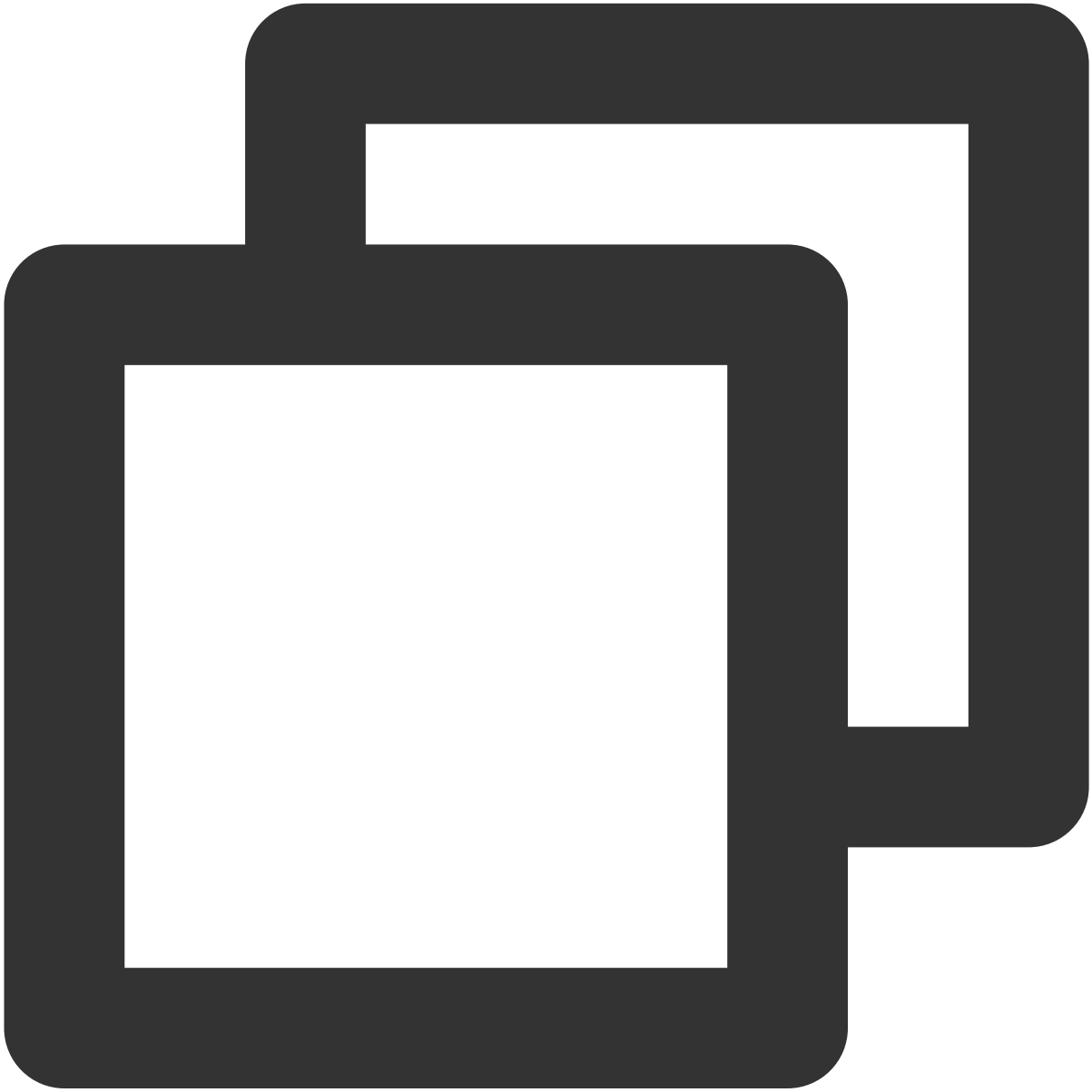
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

获取图片 EXIF

最近更新时间：2023-12-21 17:58:18

功能概述

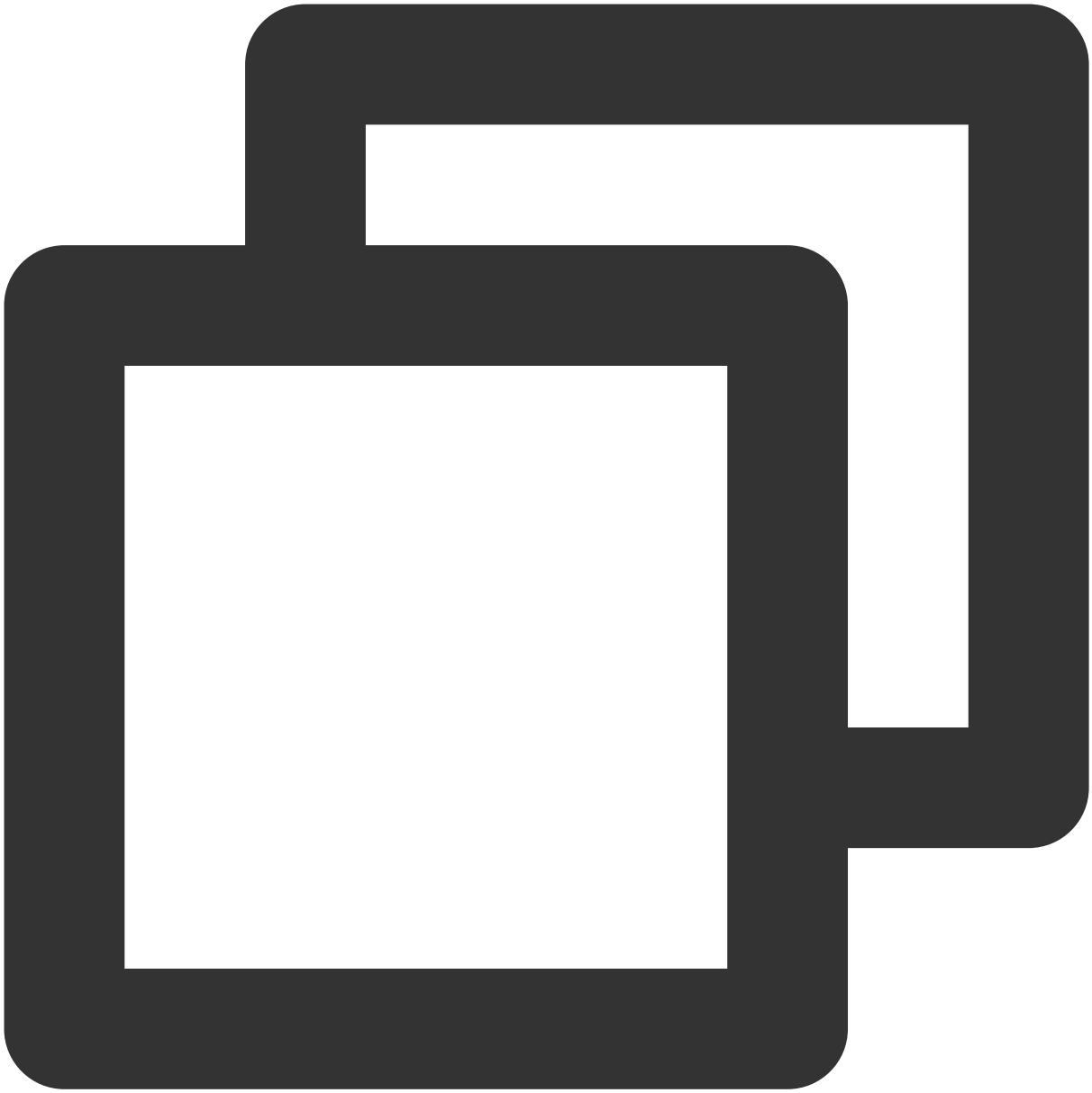
EXIF（Exchangeable Image File）全称为可交换图像文件，可记录数码照片的拍摄参数、缩略图及其他属性信息。对象存储通过数据万象的 **exif** 接口获取 EXIF 信息。处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。

注意

图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。

如图片无 exif 信息，将返回 `{"error" : "no exif data"}` 。

接口示例



```
GET /<ObjectKey>?exif HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

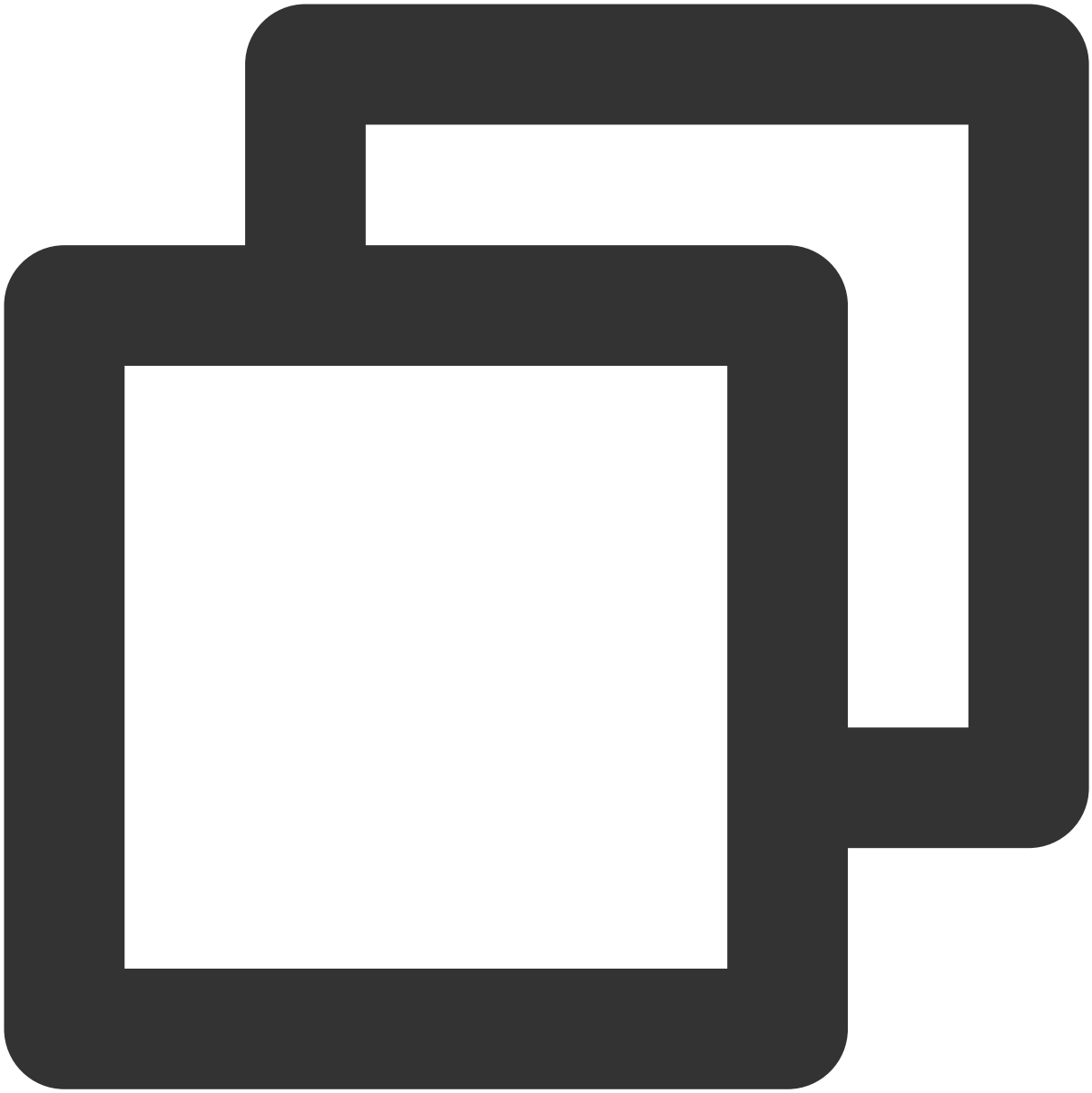
处理参数说明

操作名称：exif。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。

实际案例

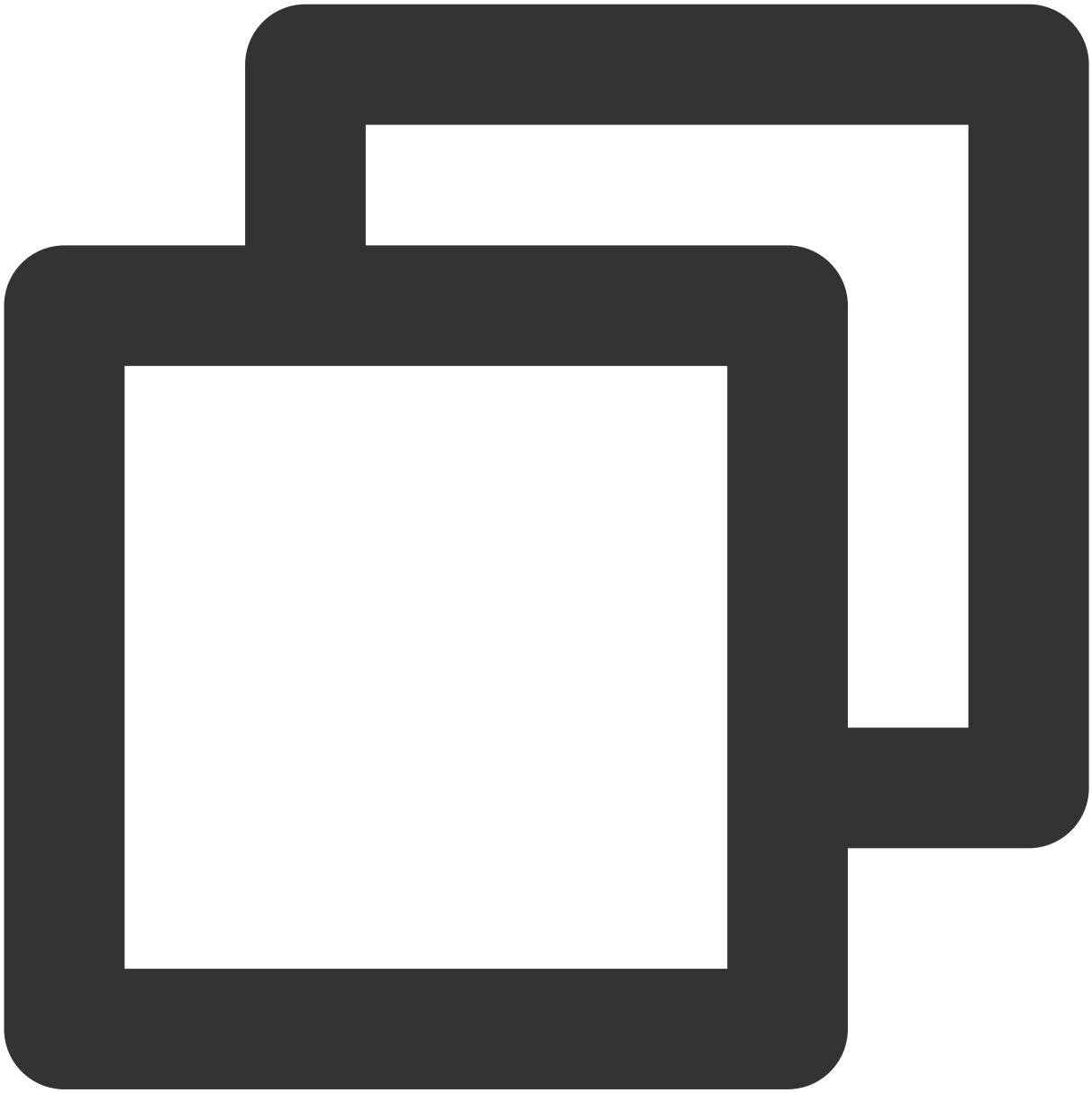
案例一：公有读



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?exif
```

案例二：私有读、携带签名

获取方式同上，仅增加签名部分，并与获取参数以“&”连接，示例如下：



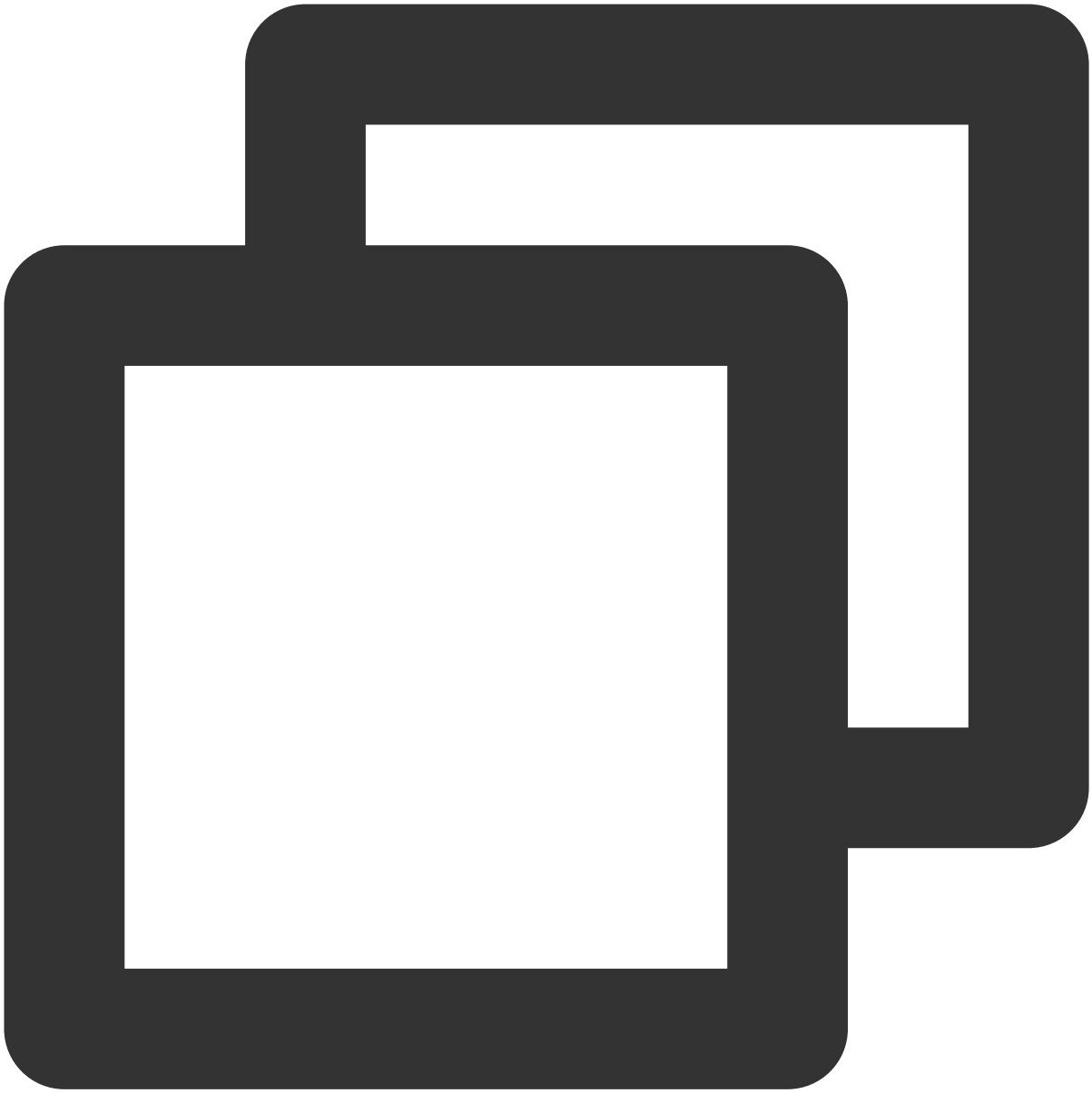
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

获取图片主色调

最近更新时间：2023-12-21 17:58:13

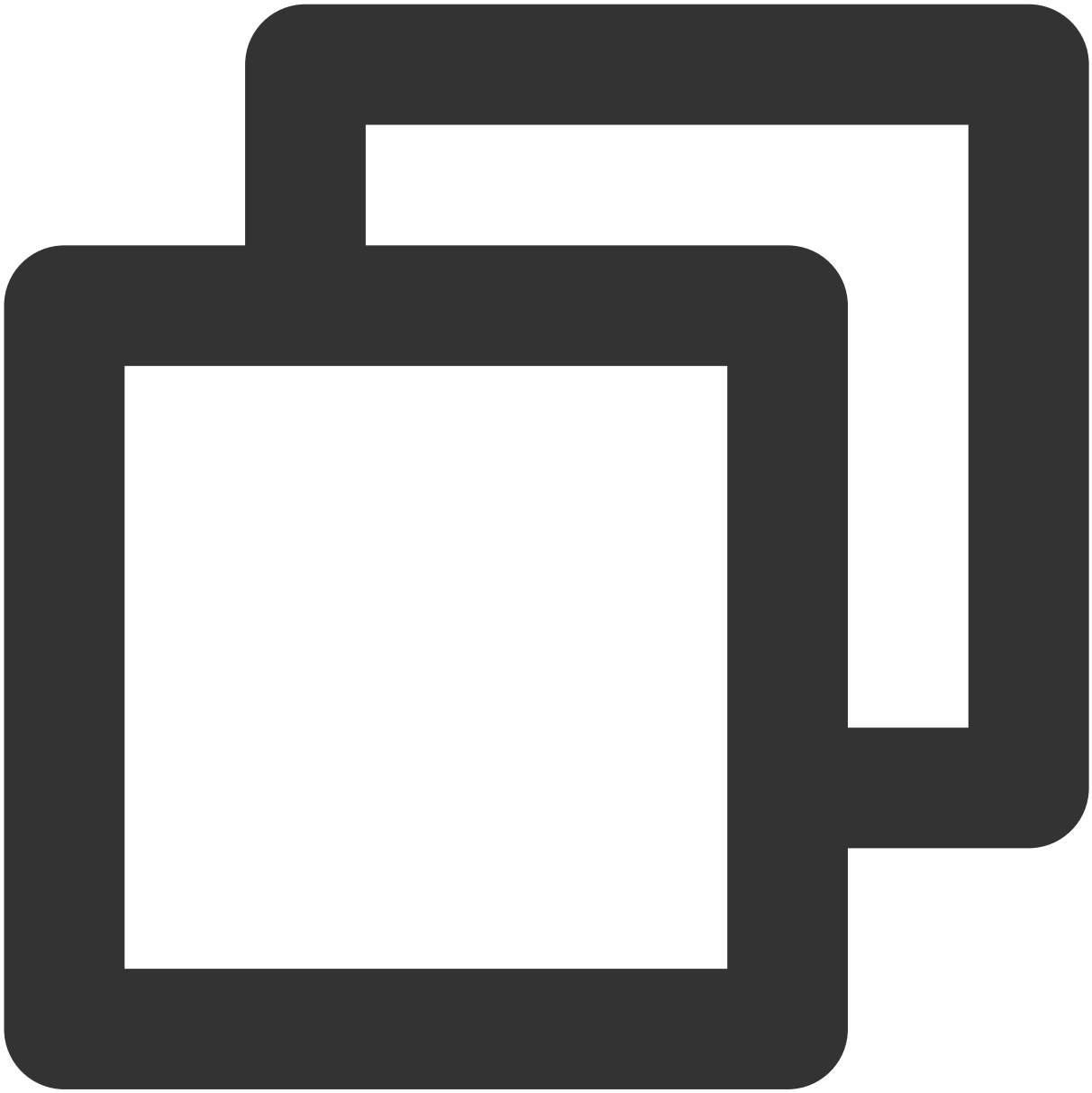
功能概述

对象存储通过数据万象 **imageAve** 接口获取图片主色调信息。处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。

注意

图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

接口示例



```
download_url?imageAve
```

参数说明

操作名称：imageAve

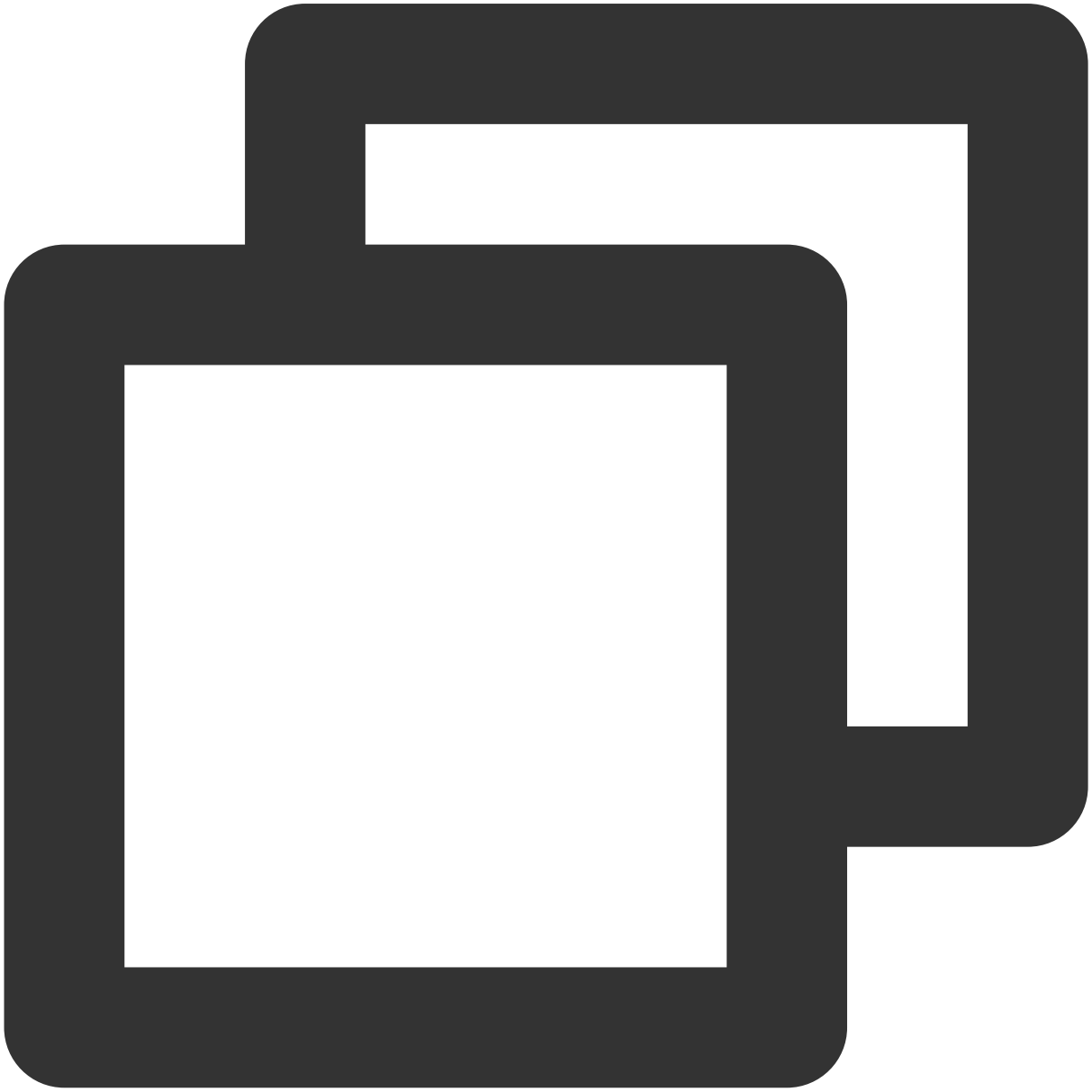
参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.

	<Region>.myqcloud.com/<picture name>，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`

实际案例

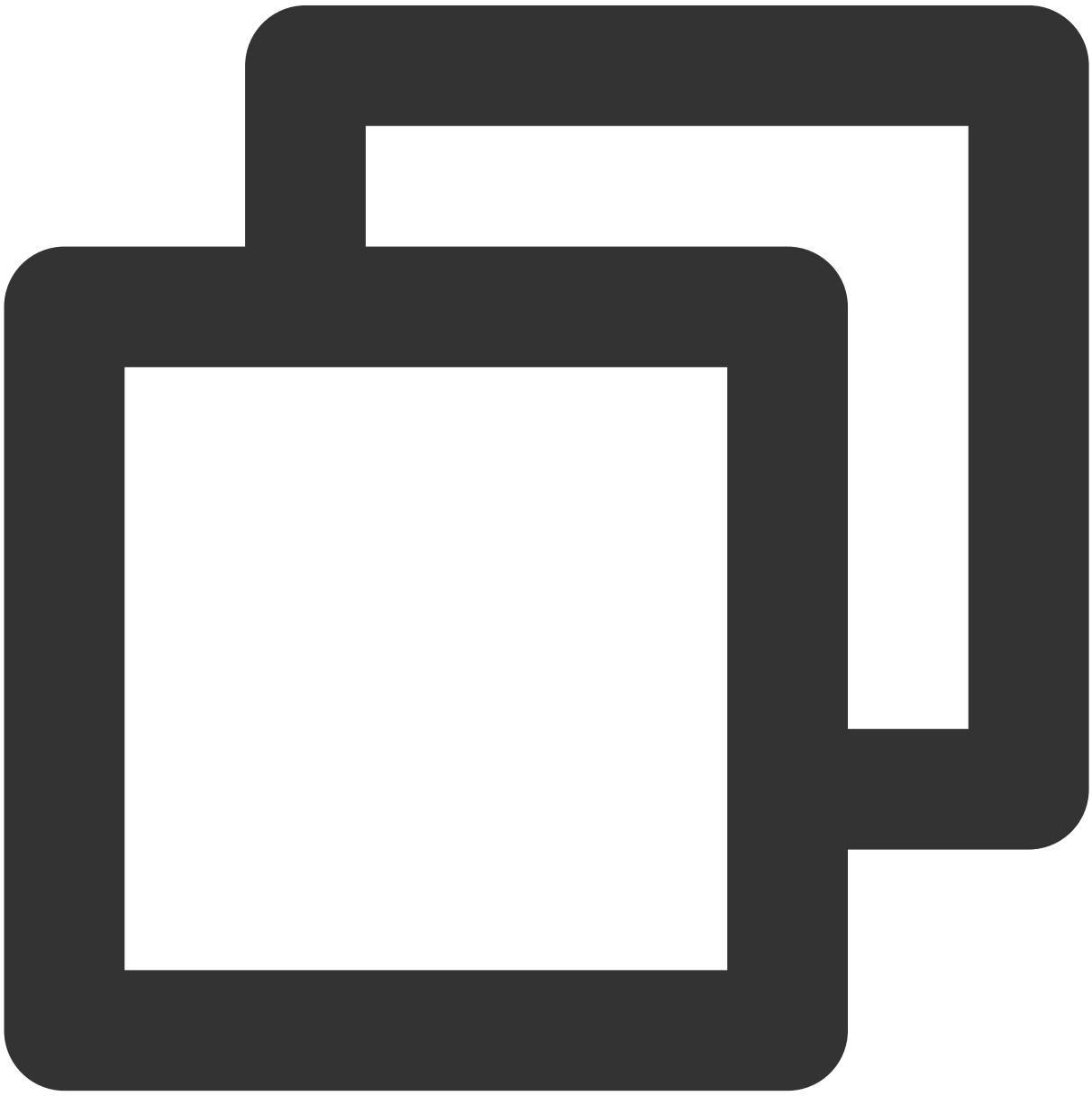
案例一：公有读

请求



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageAve
```

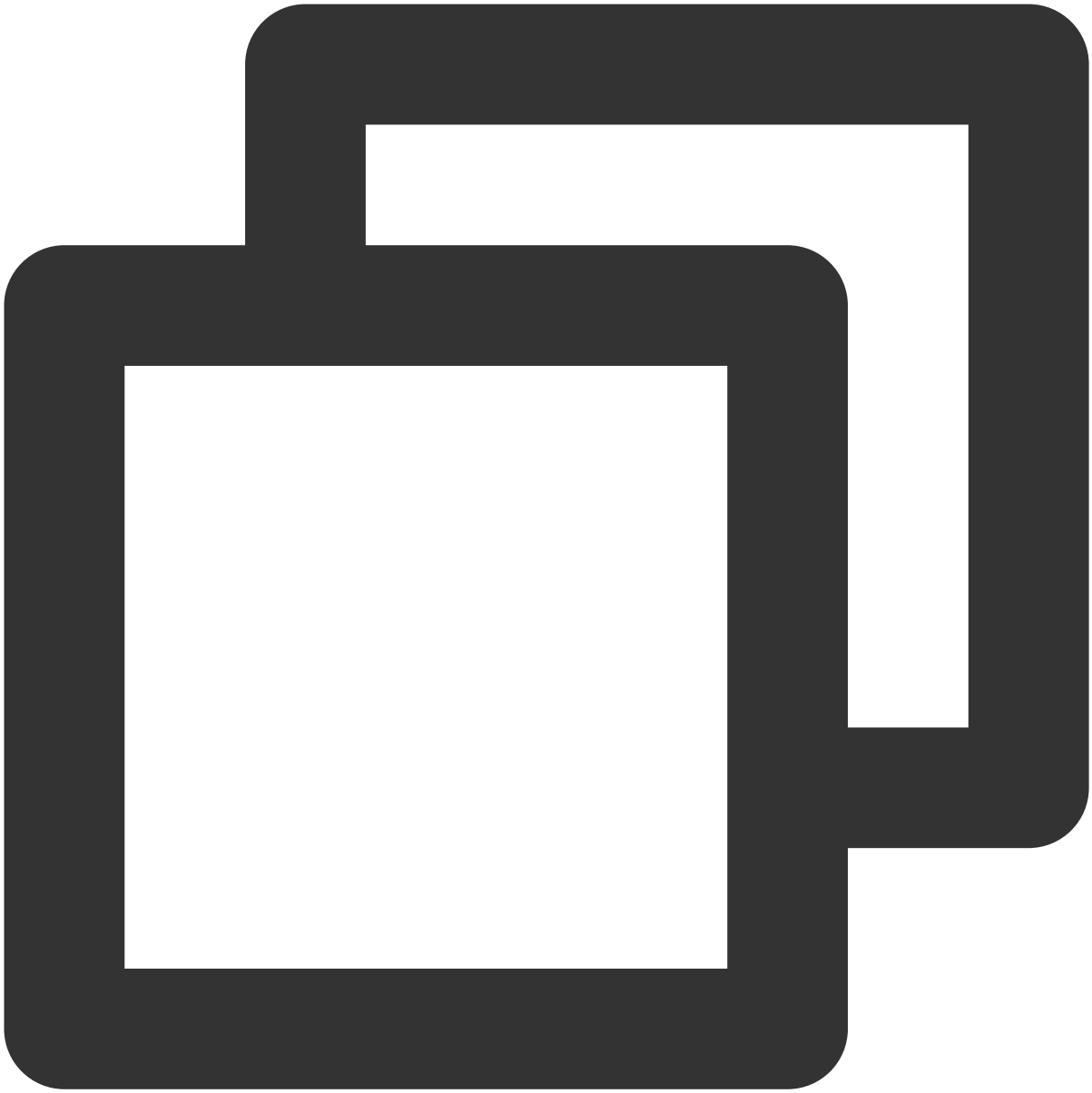
响应



```
{"RGB": "0x736246"}
```

案例二：私有读、携带签名

获取方式同上，仅增加签名部分，并与获取参数以“&”连接，示例如下：



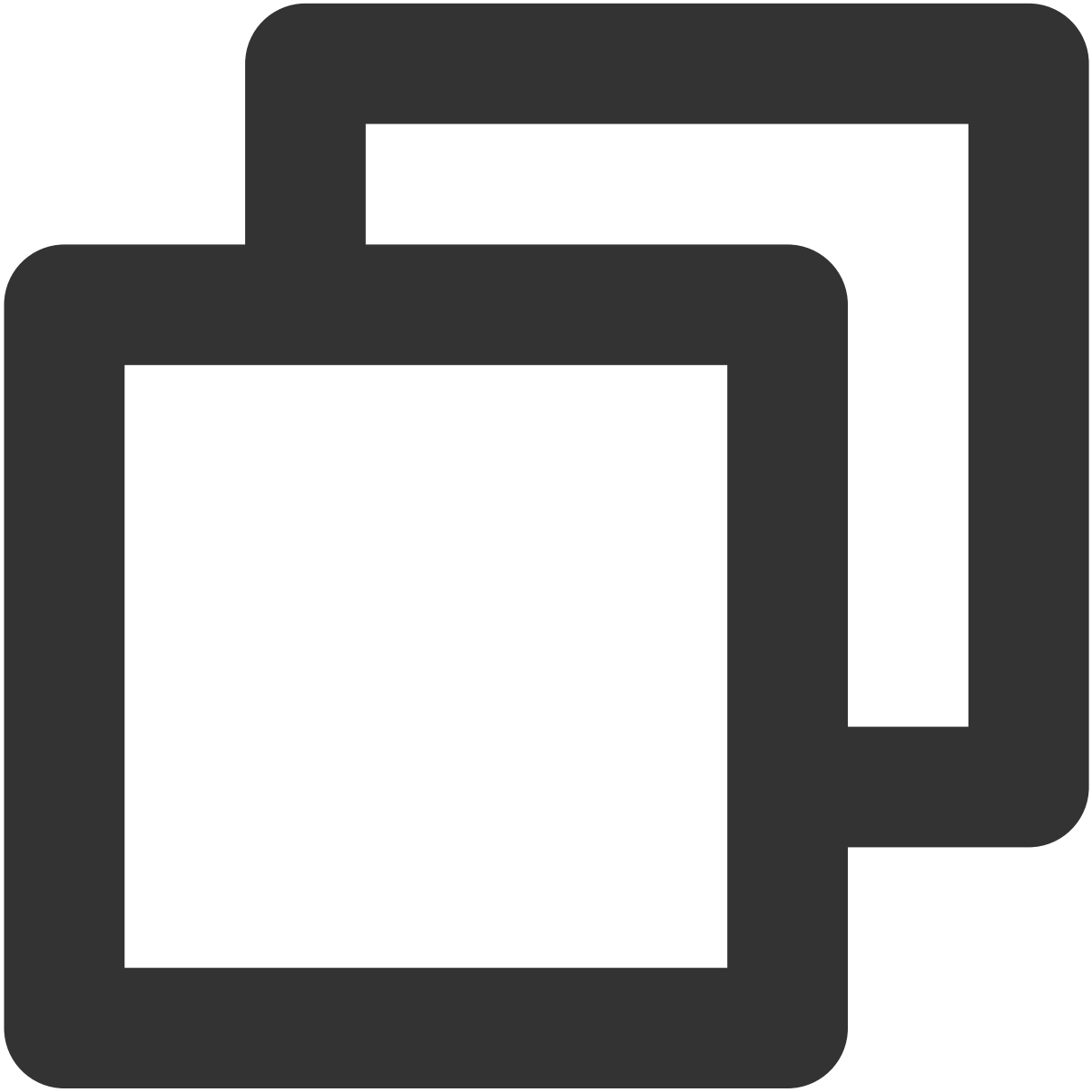
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

去除元信息

最近更新时间：2023-12-21 17:58:39

功能概述

对象存储通过数据万象 **imageMogr2** 接口可去除图片元信息，包括 exif 信息。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理
注意
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。

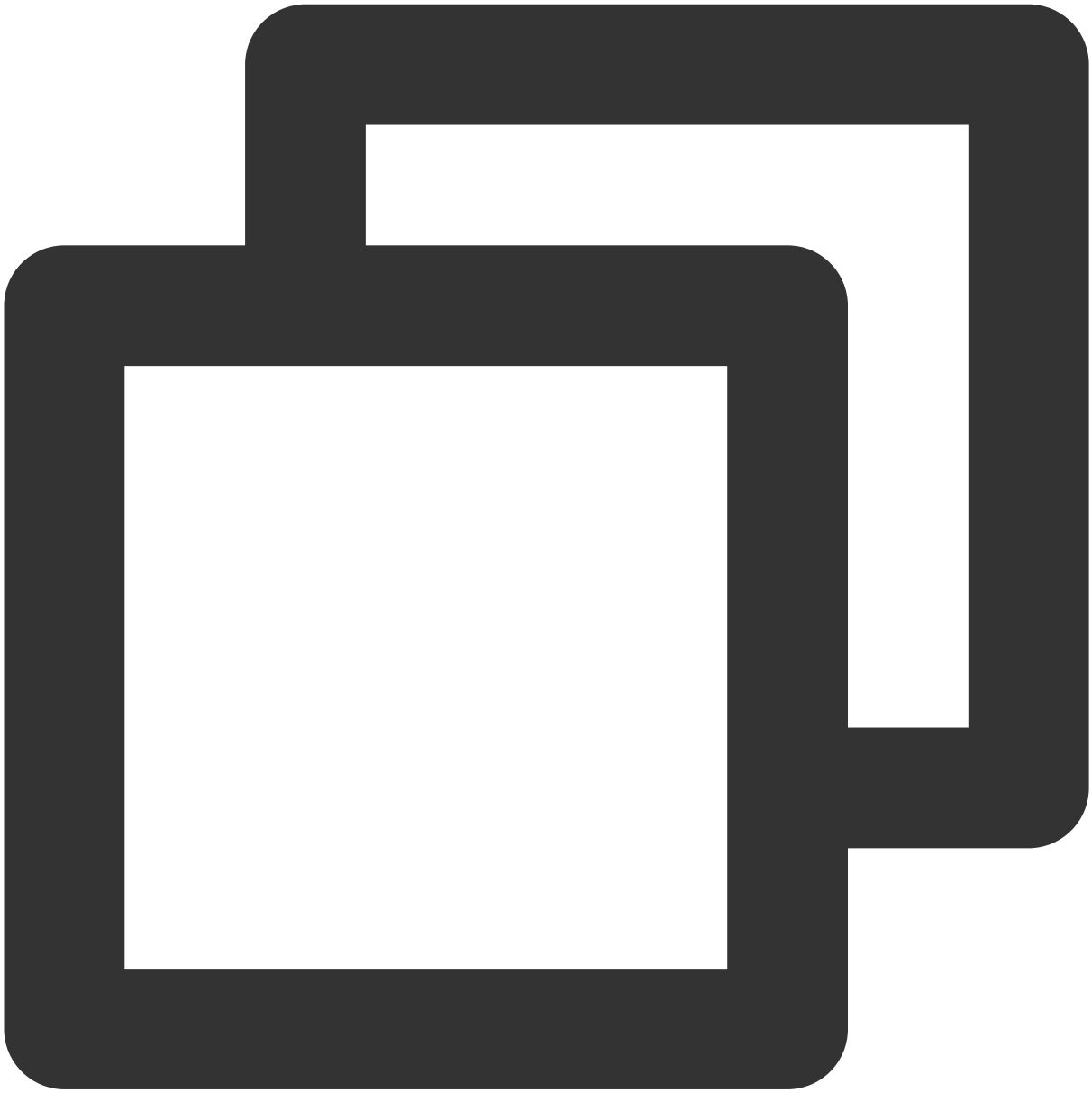
限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

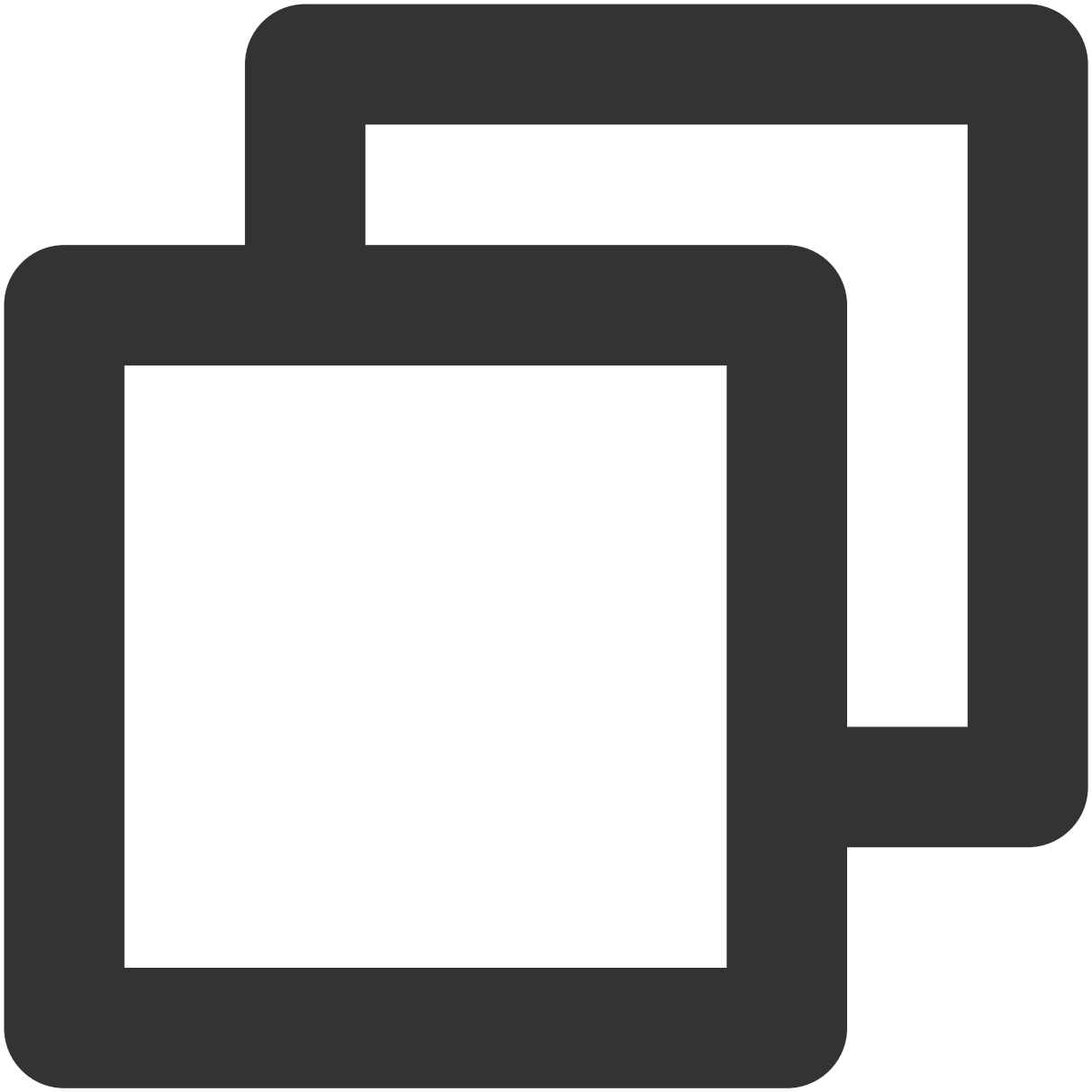
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/strip HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/strip"
  }]
```

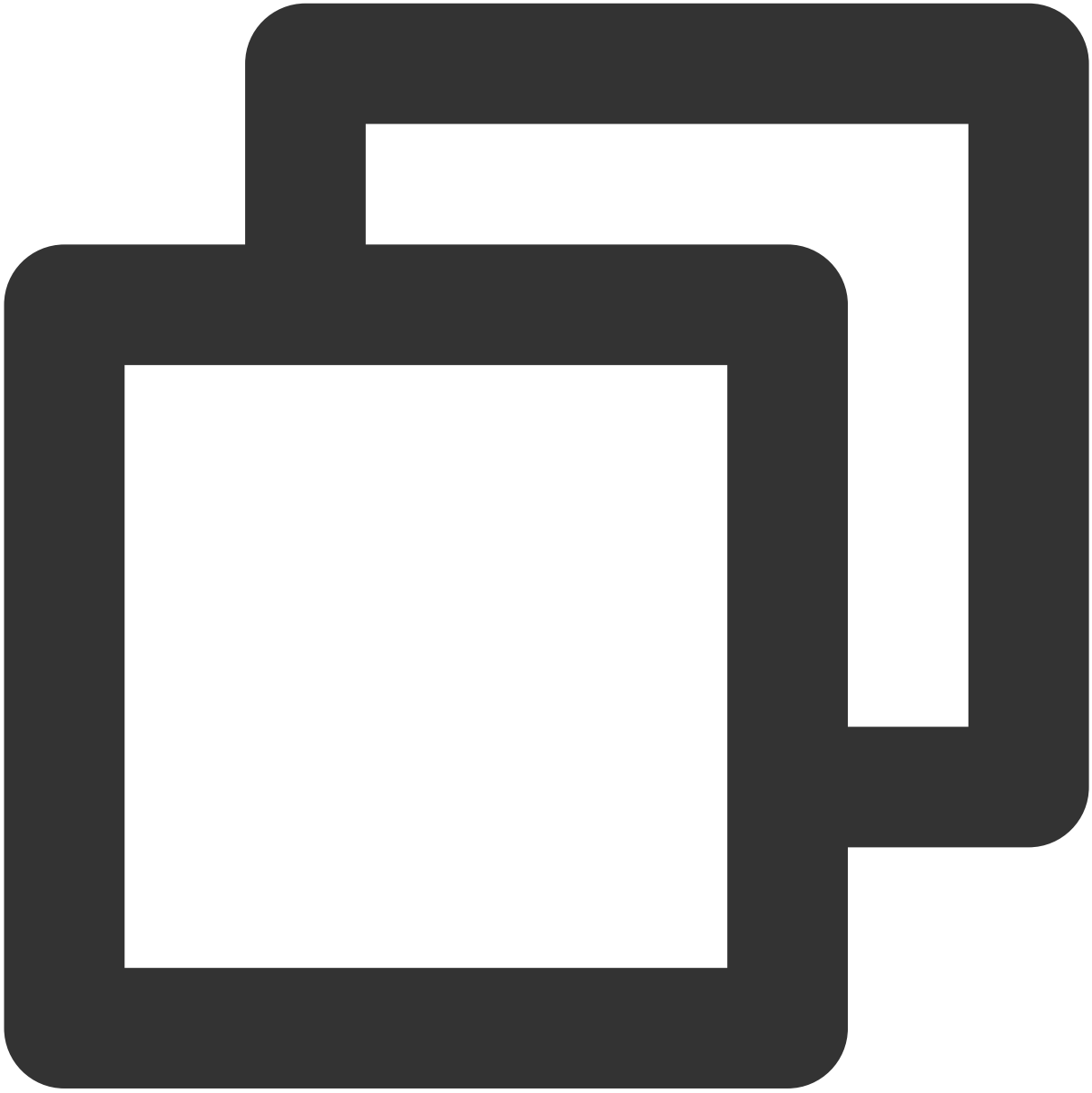
版权所有：腾讯云计算（北京）有限责任公司

```
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
```

```
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/strip"
  }]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

操作名称：strip。

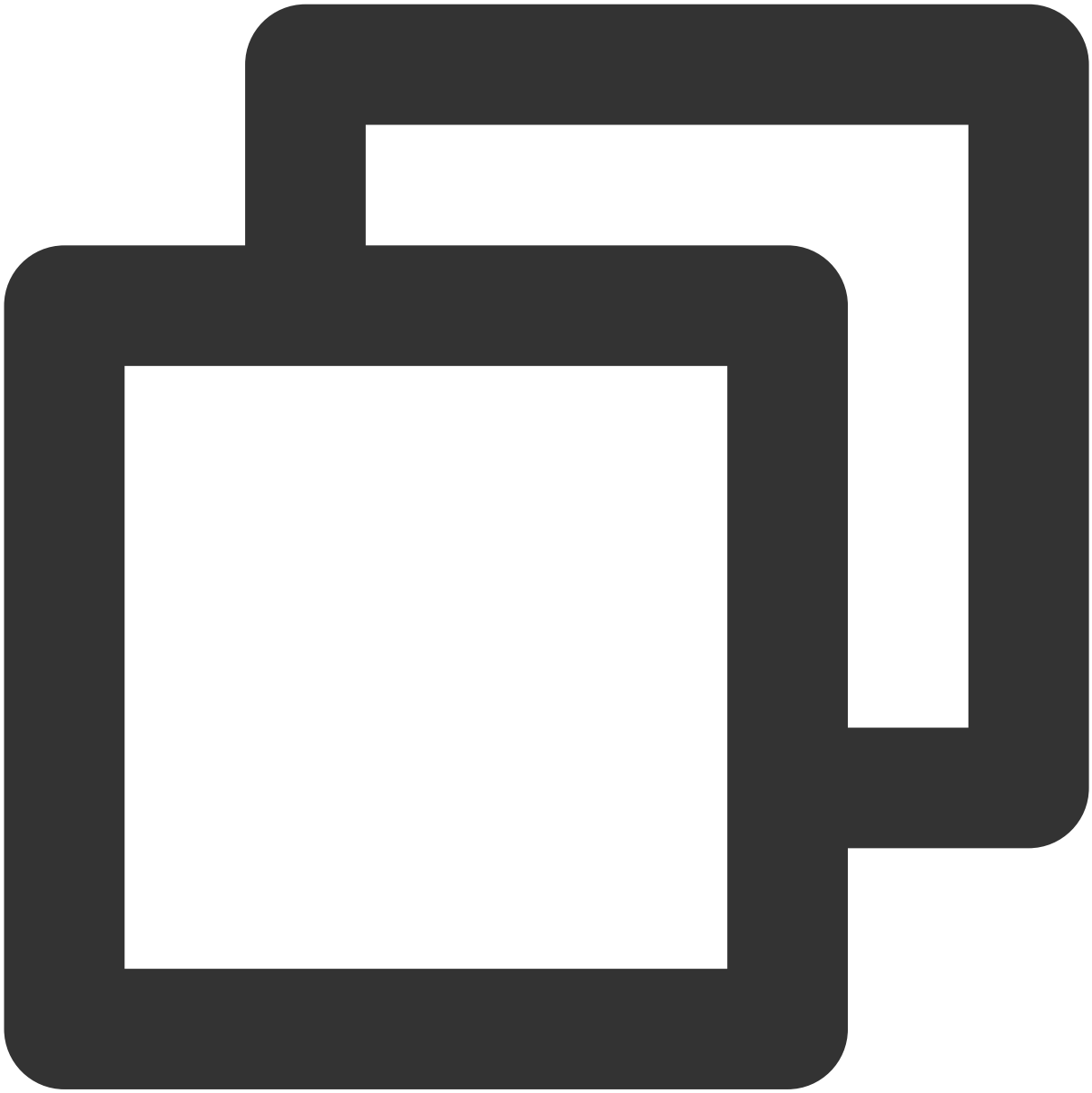
参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错。

实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

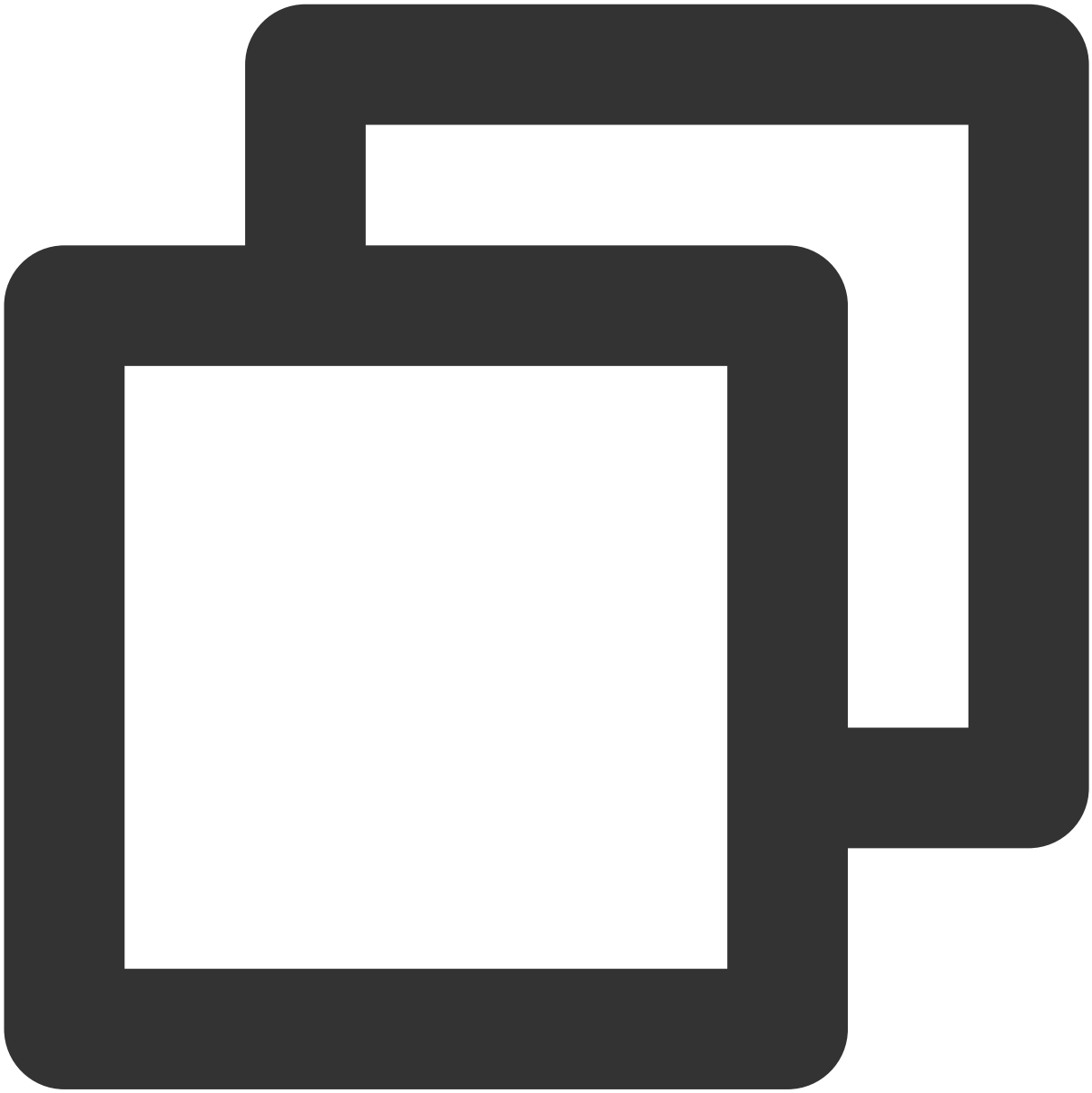
案例一：公有读



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/stri
```

案例二：私有读、携带签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：



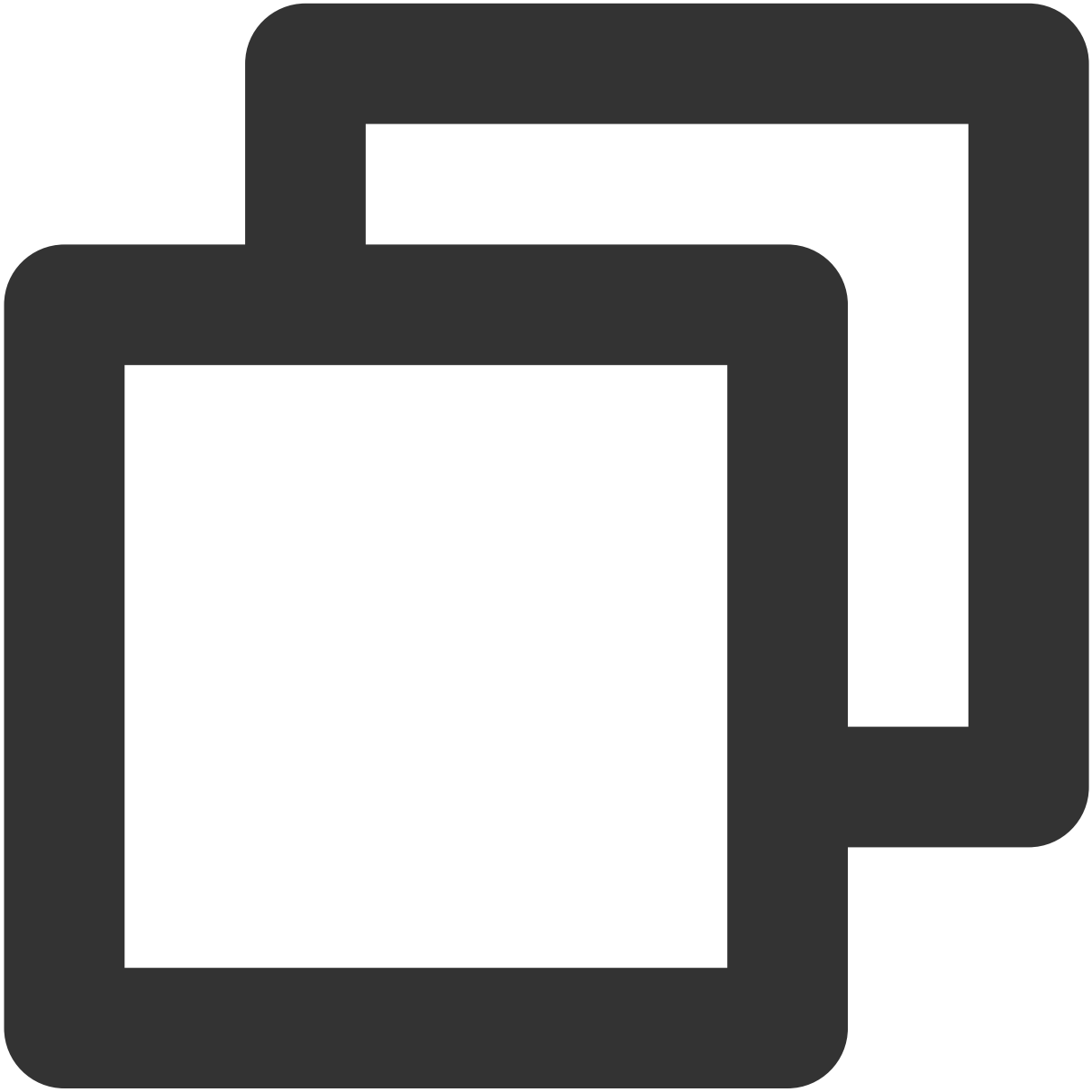
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了防止未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

快速缩略模板

最近更新时间：2023-12-21 17:58:44

功能概述

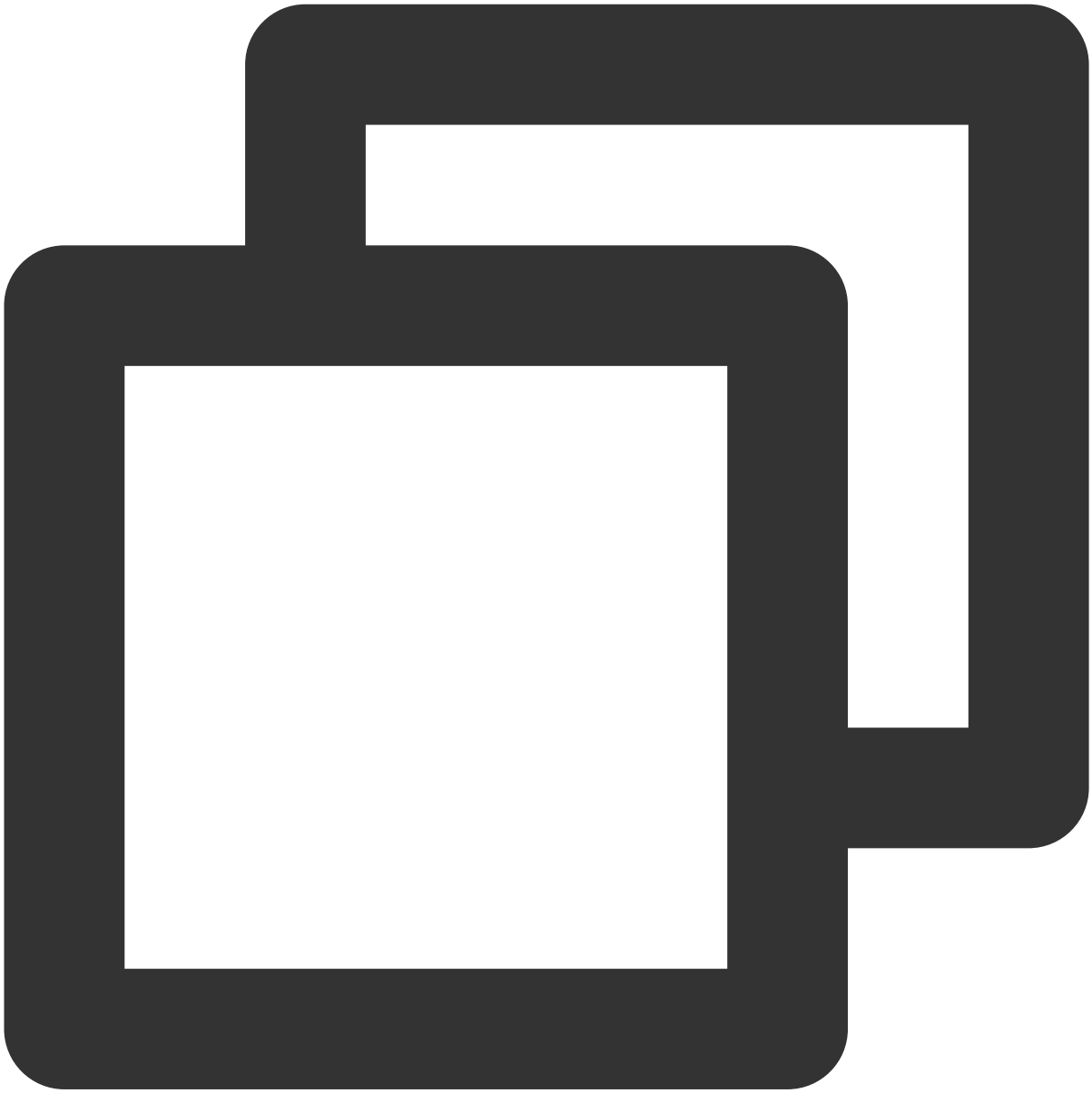
对象存储通过数据万象 **imageView2** 接口提供常用图片处理模板。开发者根据业务需求，只需在下载 URL 后面附加相应的参数，就可以生成相应的缩略图。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理
注意
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用。

限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

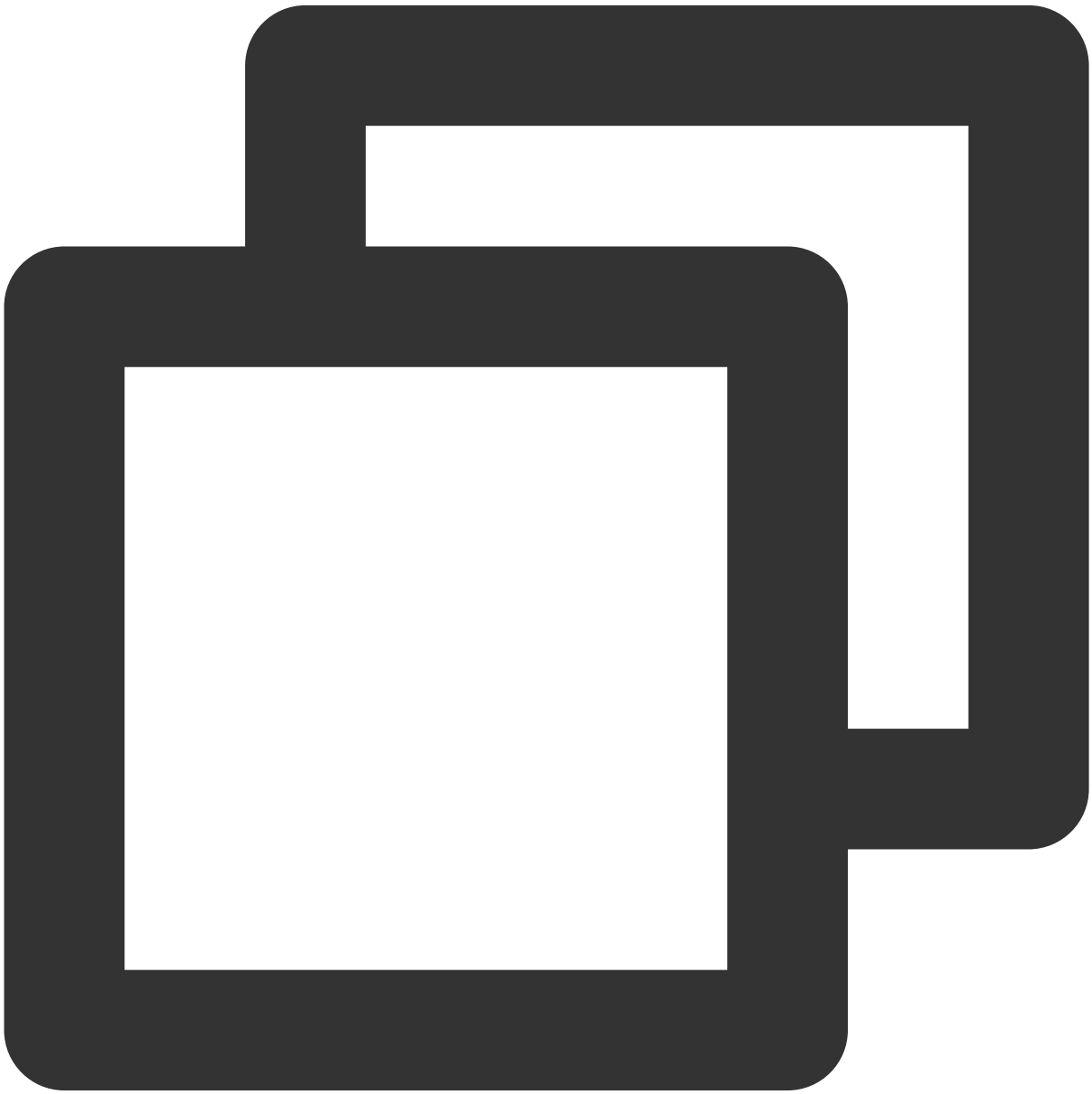


```
GET /<ObjectKey>?imageView2/<mode>/w/<Width>
                /h/<Height>
                /format/<Format>
                /q/<Quality>
                /rq/<Quality>
                /lq/<Quality> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

注意

质量变换参数仅针对 **jpg** 和 **webp** 格式图片。

2. 上传时处理



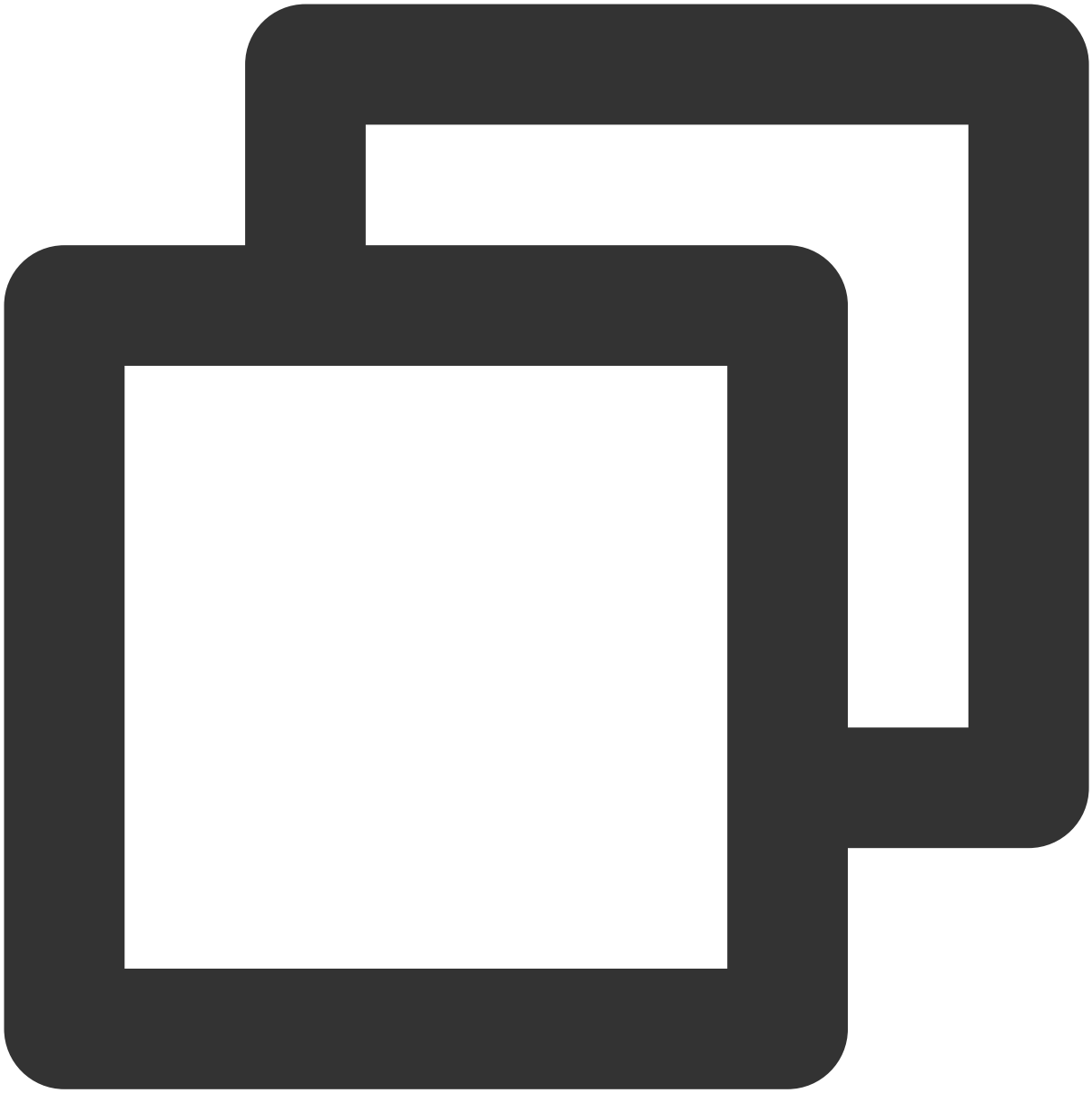
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
```

```
    "rules": [{
        "fileid": "exampleobject",
        "rule": "imageView2/<mode>/w/<Width>
                  /h/<Height>
                  /format/<Format>
                  /q/<Quality>
                  /rq/<Quality>
                  /lq/<Quality>"
    }]
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageView2/<mode>/w/<Width>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                    /h/<Height>
                    /format/<Format>
                    /q/<Quality>
                    /rq/<Quality>
                    /lq/<Quality>"
    }]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/1/w/<Width>/h/<Height>	限定缩略图的宽高最小值。该操作会将图像等比缩放直至某一边达到设定最小值，之后将另一边居中裁剪至设定值。若只指定一边，则表示宽高相等的正方形 例如，原图大小为1000x500，将参数设定为?imageView2/1/w/500/h/400后，图像会先等比缩放至800x400，之后左右各裁剪150，得到500x400大小的图像
/2/w/<Width>/h/<Height>	限定缩略图的宽高最大值。该操作会将图像等比缩放至宽高都小于设定最大值 例如，原图大小为 1000x500，将参数设定为?imageView2/2/w/500/h/400后，图像会等比缩放至500x250。如果只指定一边，则另一边自适应
/3/w/<Width>/h/<Height>	限定缩略图的宽高最小值。该操作会将图像等比缩放至宽高都大于设定最小值 例如，原图大小为 1000x500，将参数设定为?imageView2/3/w/500/h/400后，图像会等比缩放至800x400。如果只指定一边，则另一边设为相同值
/4/w/<LongEdge>/h/<ShortEdge>	限定缩略图的长边和短边的最小值分别为 LongEdge 和 ShortEdge，进行等比压缩；如果只指定一边，代表另外一边为同样的值
/5/w/<LongEdge>/h/<ShortEdge>	限定缩略图的长边和短边的最大值分别为 LongEdge 和 ShortEdge，进行等比压缩，居中裁剪；如果只指定一边，则表示宽高相等的正方形；缩放后其中一边多余的部分会被裁剪掉
/format/<Format>	目标缩略图的图片格式，Format 可为：jpg，bmp，gif，png，webp，缺省为原图格式

版权所有：腾讯云计算（北京）有限责任公司

/q/<Quality>	图片质量，取值范围0 - 100，默认值为原图质量；取原图质量和指定质量的最小值；<Quality> 后面加!（注意为英文字符），表示强制使用指定值
/rq/<quality>	图片的相对质量，取值范围0 - 100，数值以原图质量为标准。例如原图质量为80，将 rquality 设置为80后，得到处理结果图的图片质量为64（80x80%）
/lq/<quality>	图片的最低质量，取值范围0 - 100，设置结果图的质量参数最小值。 例如，原图质量为85，将 lquality 设置为80后，处理结果图的图片质量为85 例如，原图质量为60，将 lquality 设置为80后，处理结果图的图片质量会被提升至80

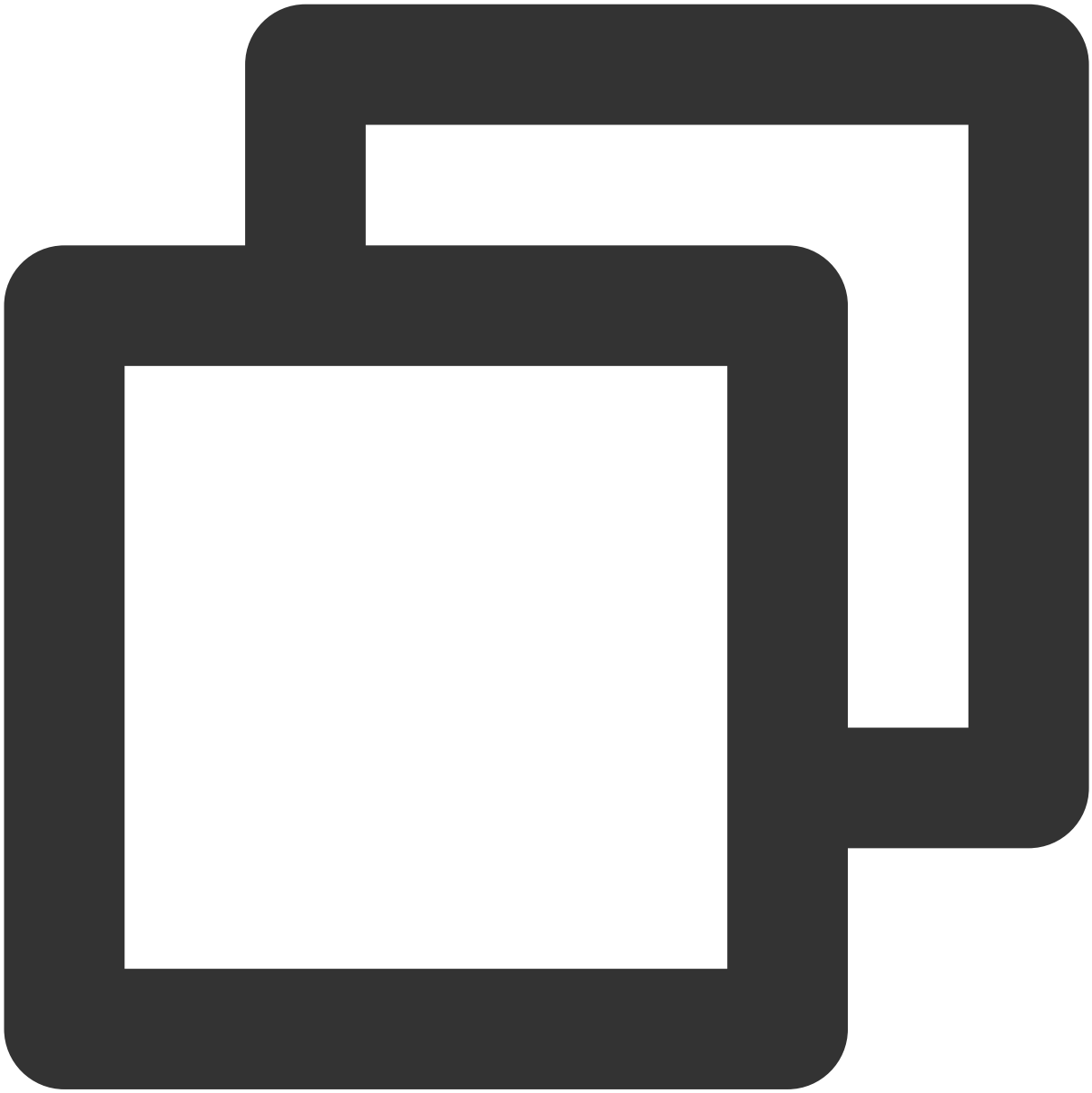
实际案例

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

案例一：选用模板样式1

选用样式1，并限定缩略图的宽高最小值为400 × 600，绝对质量为85：



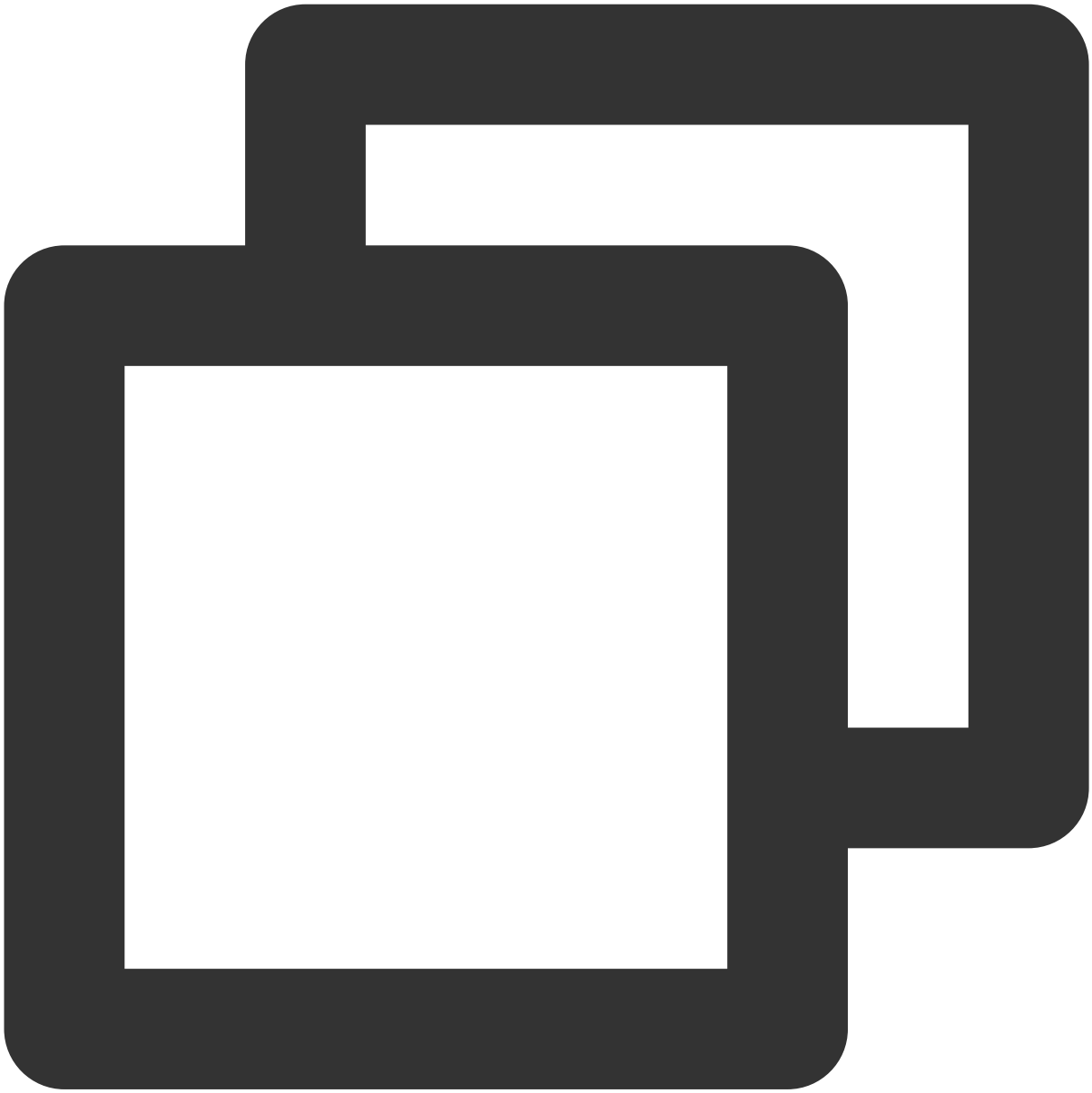
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageView2/1/w/
```

缩略图效果如下：



案例二：选用模板样式1并携带私有签名

处理方式同上，仅增加签名部分，并与处理参数以“&”连接，示例如下：



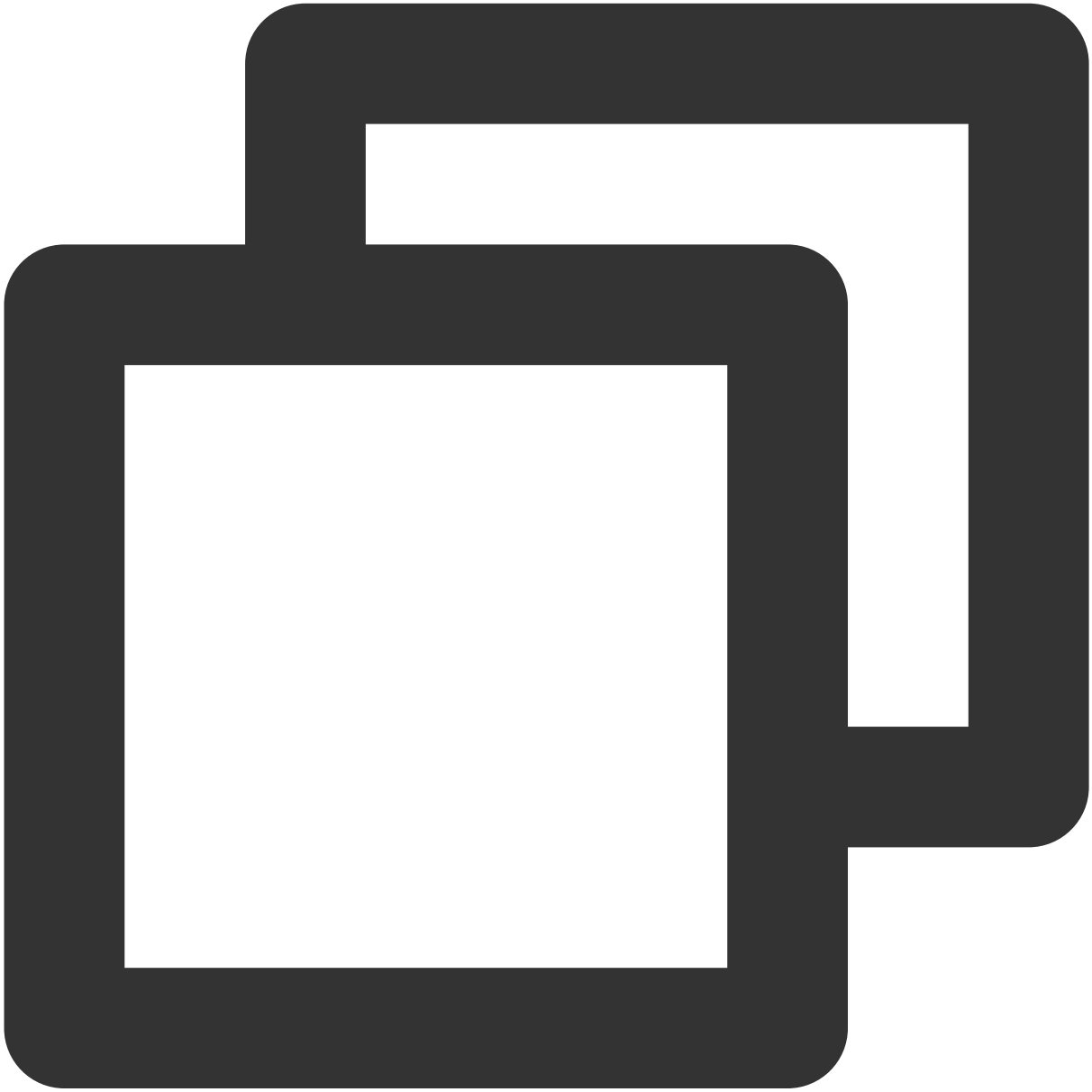
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

限制图片大小

最近更新时间：2023-12-21 17:59:20

功能概述

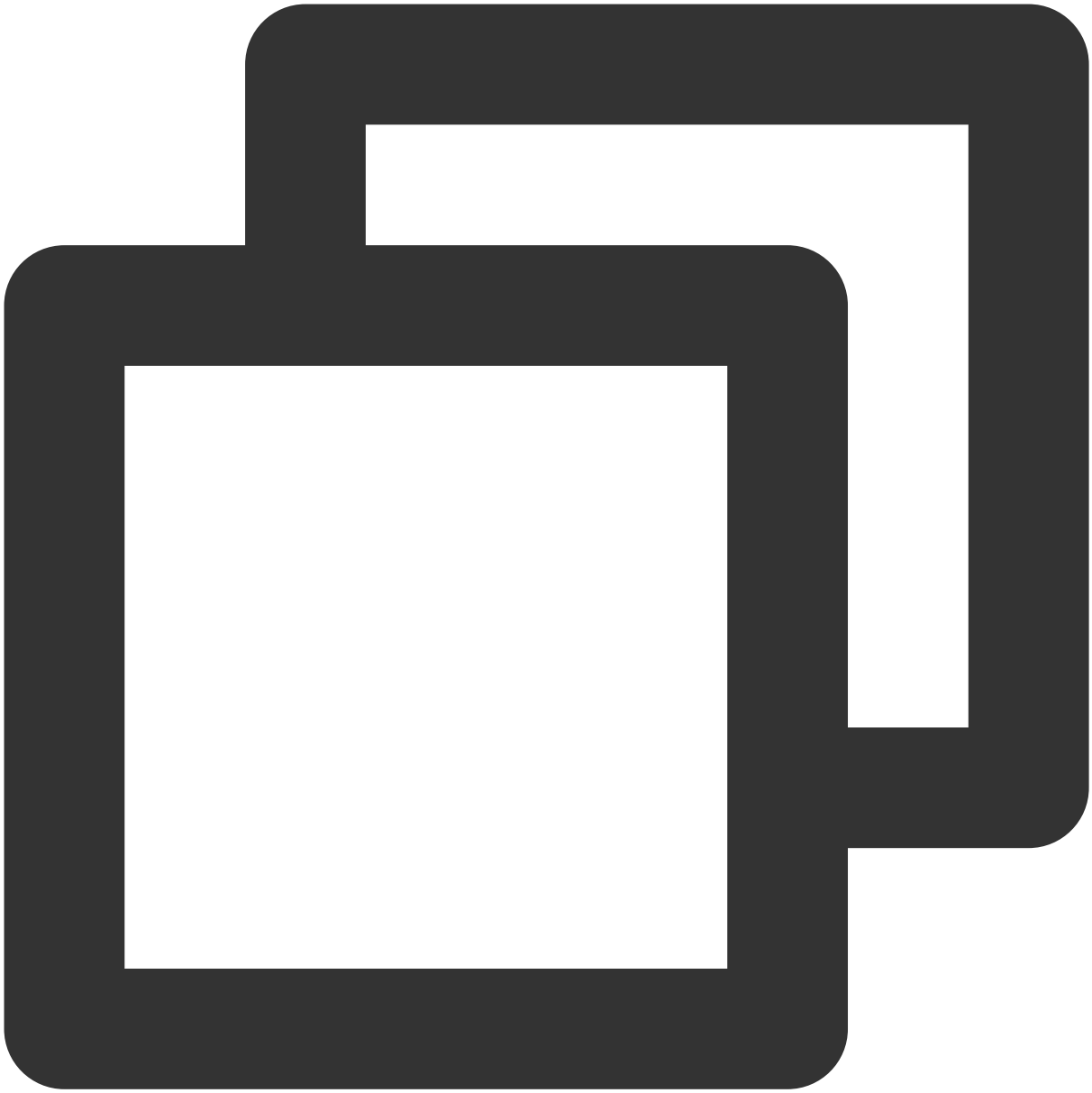
对象存储通过数据万象 **imageMogr2/size-limit** 接口可限制图片处理（例如缩放、压缩等）后的文件大小。

限制说明

支持格式：支持处理 JPG、BMP、GIF、PNG、WEBP 格式，并且支持 HEIF 格式的解码和处理操作。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

接口示例

版权所有：腾讯云计算（北京）有限责任公司



```
download_url?imageMogr2/size-limit
```

处理参数说明

操作名称：size-limit。

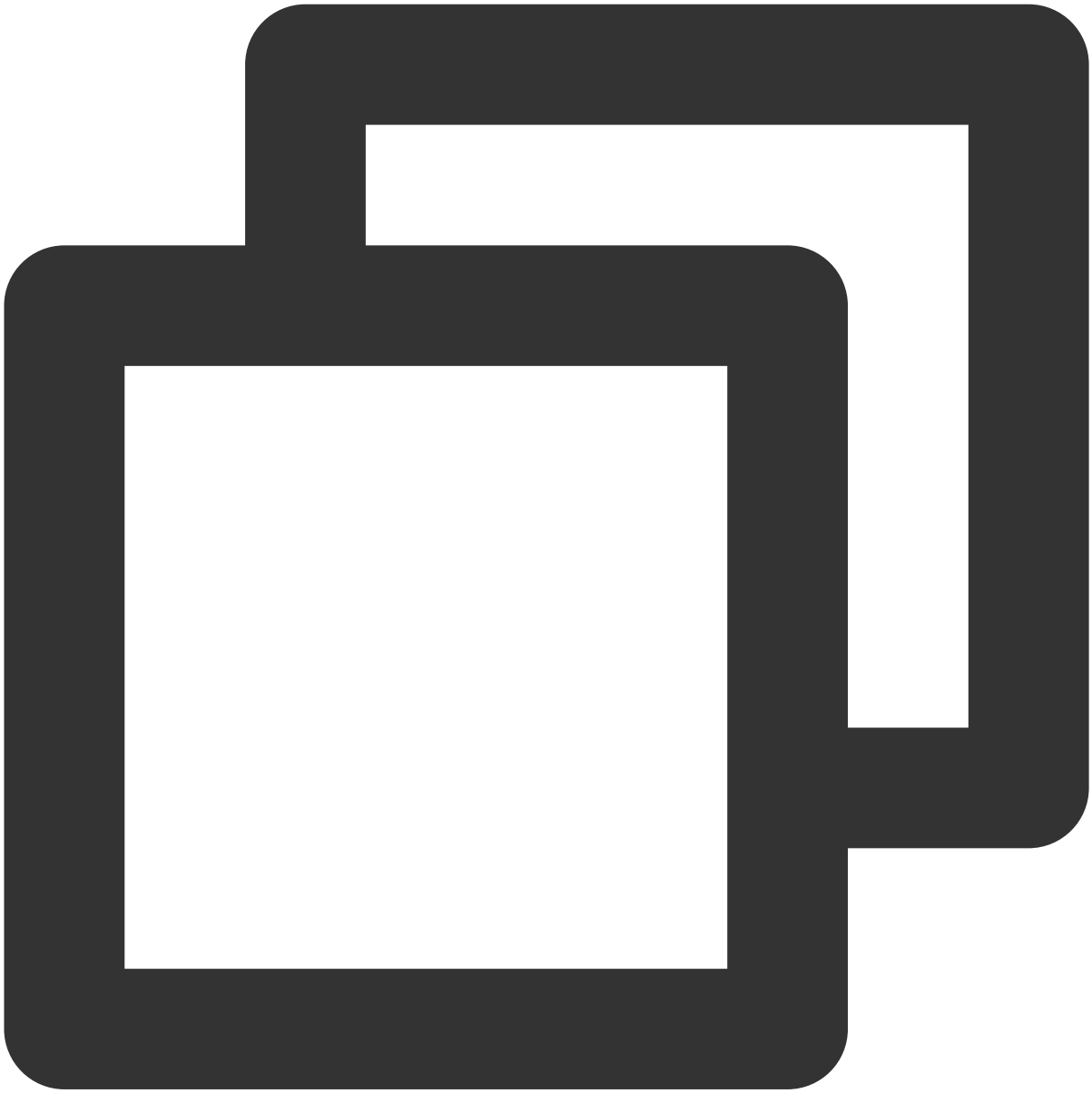
参数	含义
download_url	文件的访问链接，具体构成为 `<BucketName-APPID>.cos.`

	`<Region>.myqcloud.com/<picture name>`，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`
size-limit	限制图片转换后的大小，支持以兆字节和千字节为单位的图片 1. 仅支持 JPG 格式的图片，可以用于限制处理后图片的大小 2. 若在尾部加上!，表示用处理后的图片大小与原图大小做比较，如果处理后的图片比原图小，则返回处理后的图片，否则返回原图。例如：`examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpg?imageMogr2/size-limit/15k!` 3. 建议搭配strip参数使用，去除图片的一些冗余信息，会有更好的效果。例如：`examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpg?imageMogr2/strip/format/png/size-limit/15k!`
/ignore-error/1	当处理参数中携带此参数时，针对文件过大导致处理失败的场景，会直接返回原图而不报错

实际案例

案例一：转换图片格式并限制大小

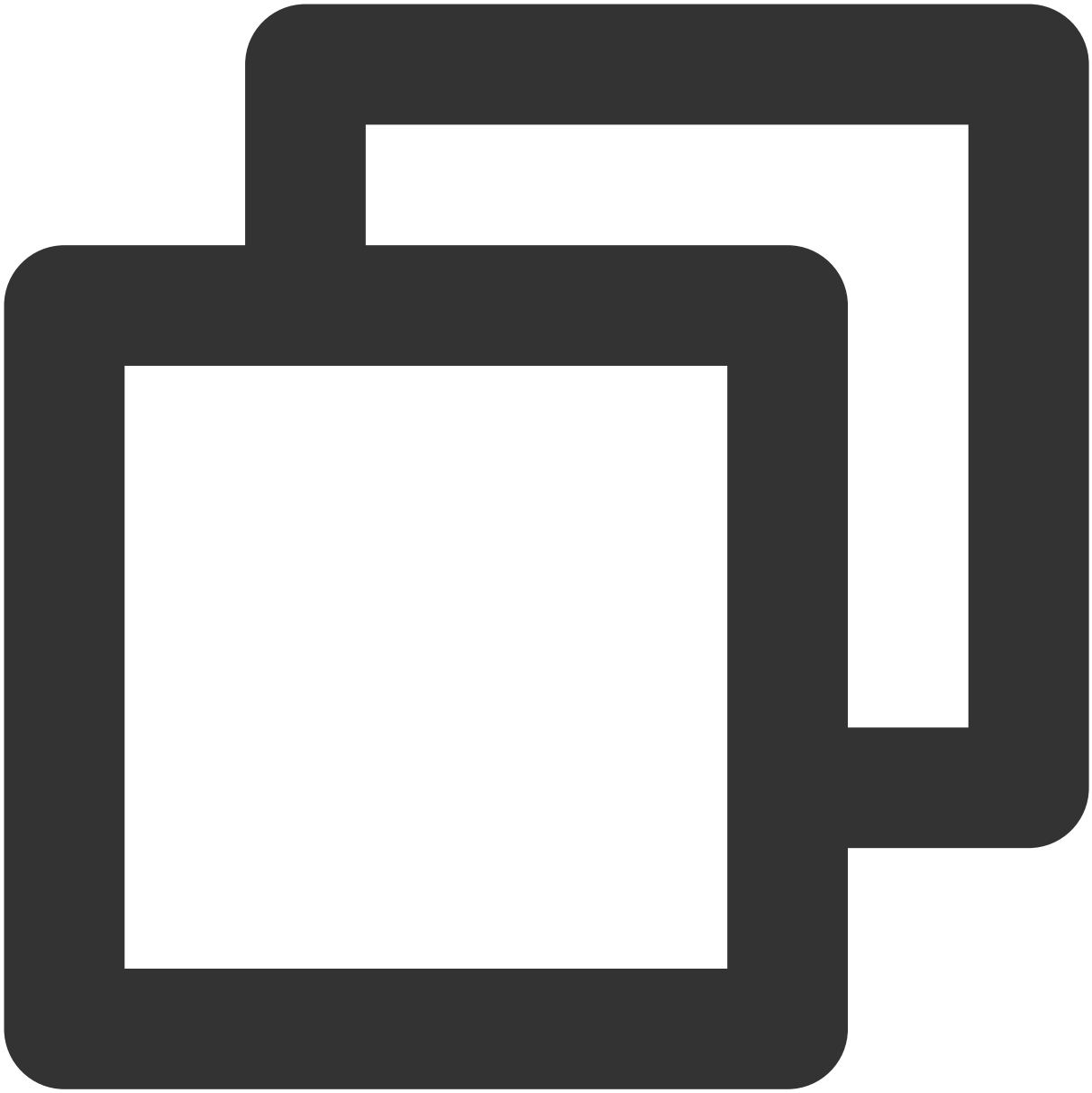
将 JPG 图片转换为 PNG 格式，并限制图片大小为15KB，示例如下：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/stri
```

案例二：转换图片格式、限制大小并携带私有文件签名

处理方式同上，仅增加签名部分，并与处理参数以“&”连接，示例如下：



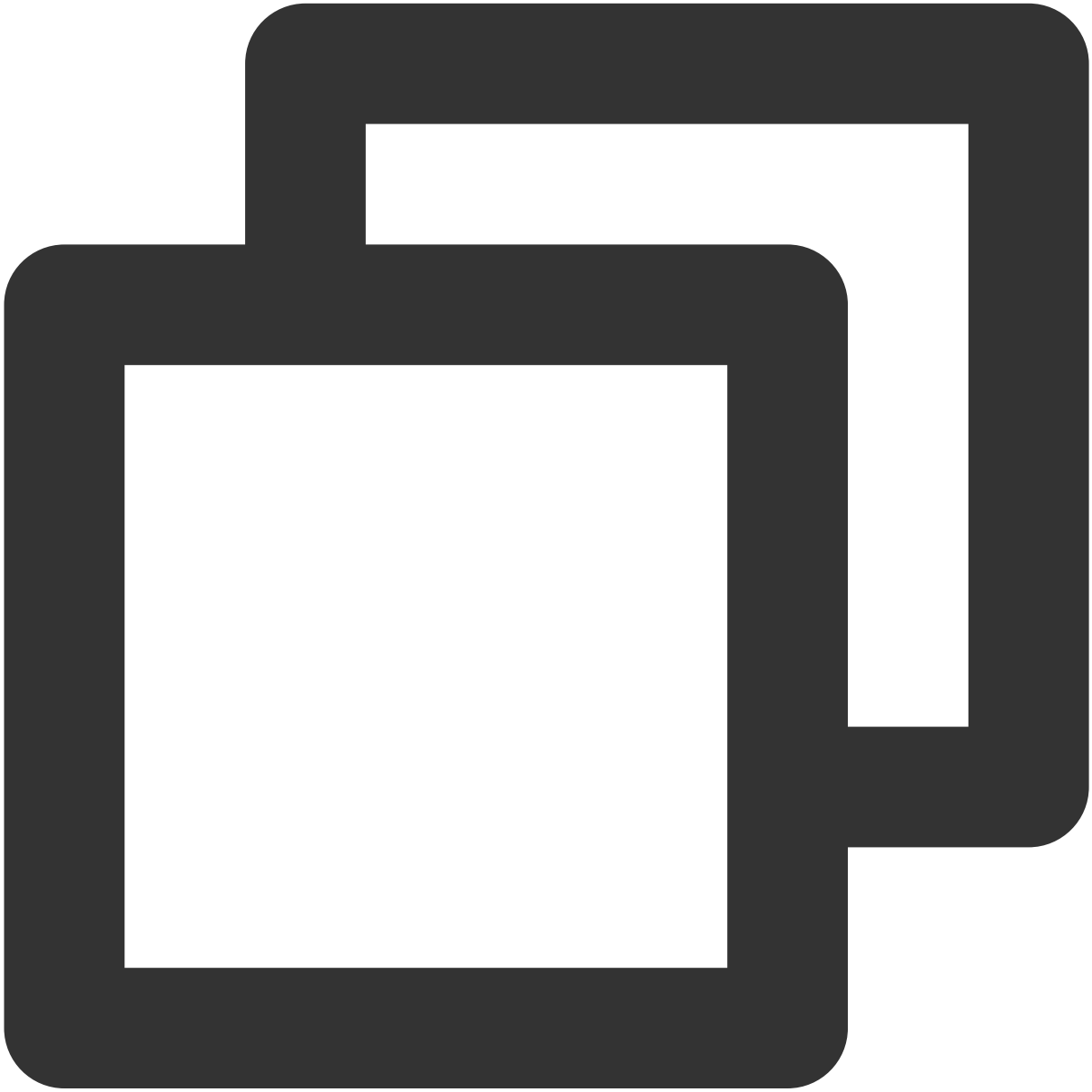
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

管道操作符

最近更新时间：2023-12-21 17:59:32

功能概述

对象存储通过数据万象的管道操作符 `|` 实现对图片按顺序进行多种处理。用户可以通过管道操作符将多个处理参数分隔开，从而实现在一次访问中按顺序对图片进行不同处理。处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。

注意

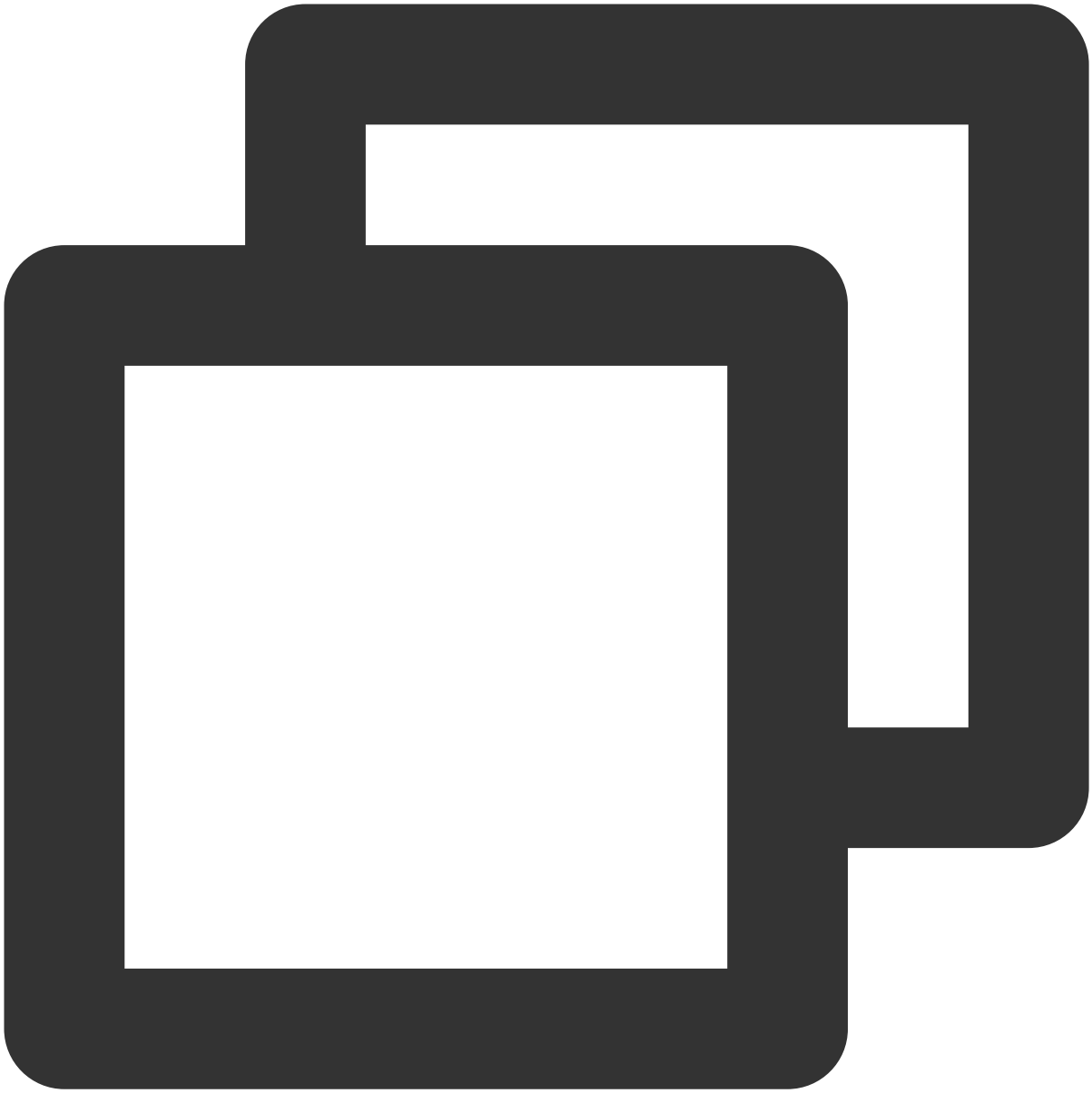
图片处理功能为收费项，由数据万象收取，详细的计费说明请参见数据万象图片处理费用。

接口示例

用户在图片 URL 链接后以样式分隔符 `?` 与处理样式相连接，多个处理样式之间以管道操作符 `|` 分隔，样式按照先后顺序执行，目前最多支持10层管道。

实际案例

本次案例先将原图缩放为50%，然后在图的右下角添加“数据万象”的文字水印。
假设原图 URL 为：



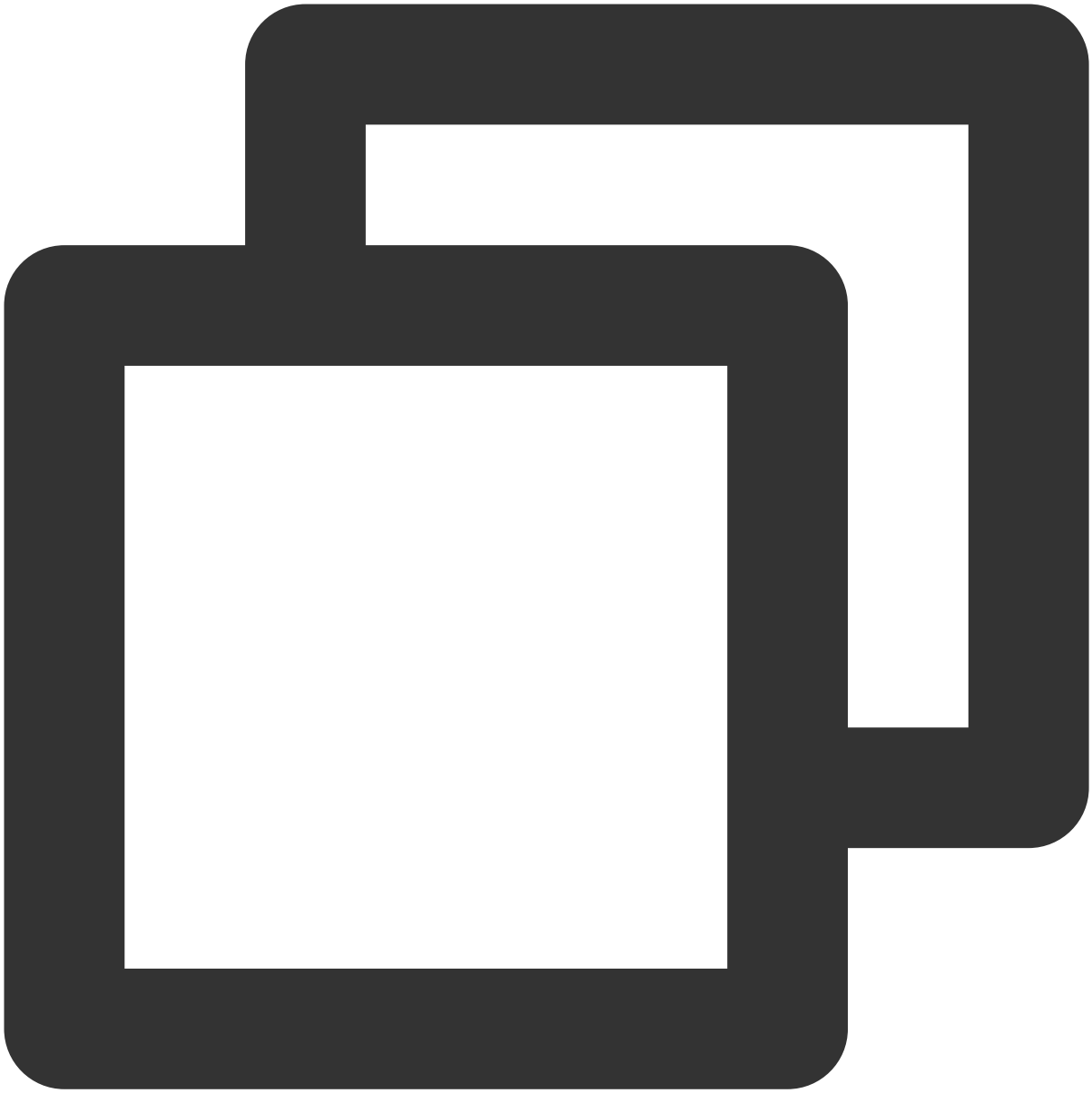
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg
```

原图显示如下：



案例一：对图片进行缩放并添加水印

首先增加缩放操作，则 URL 为：

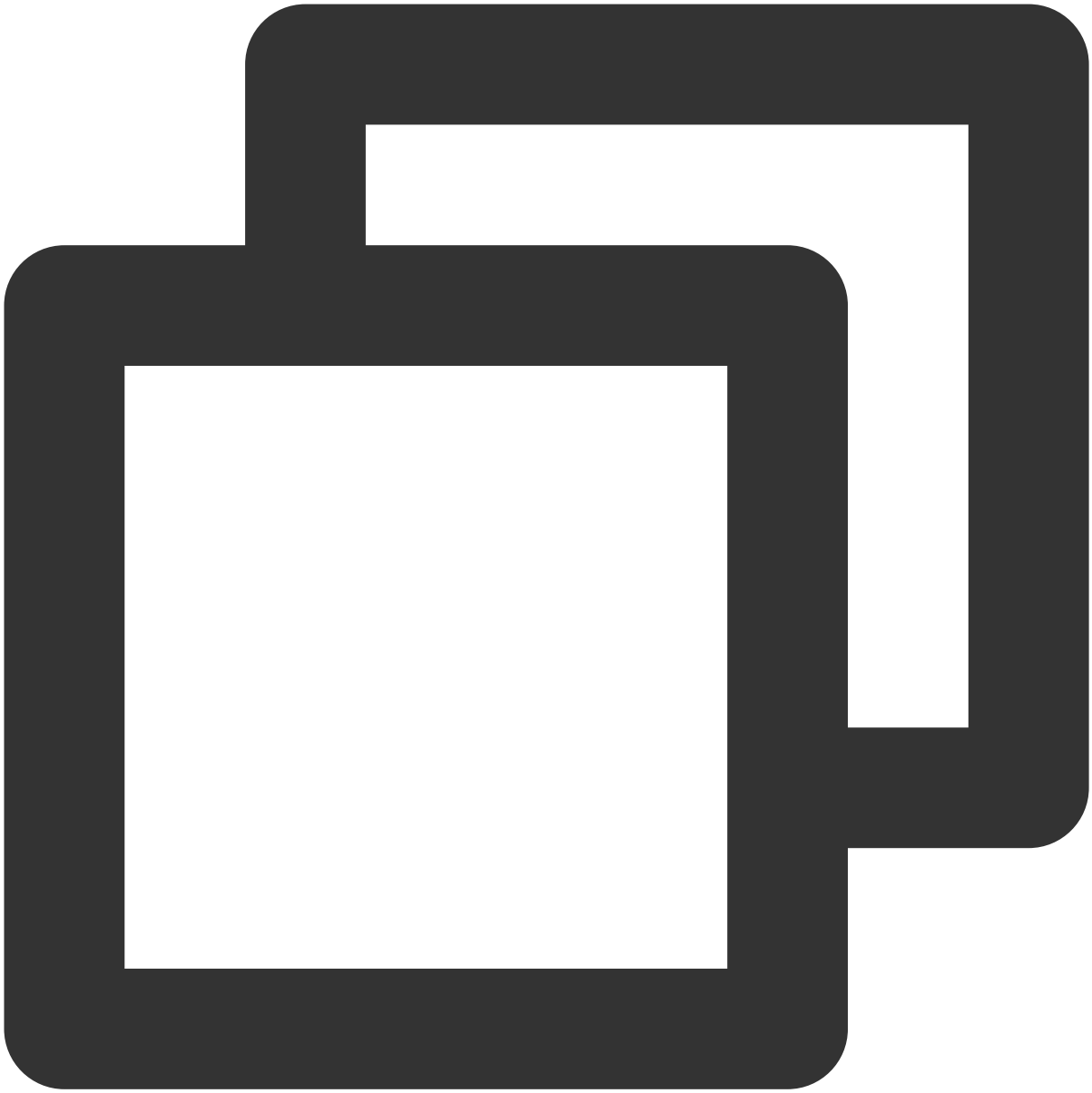


```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

图片显示如下：



使用管道操作符再添加文字水印的样式处理，则最终 URL 为：



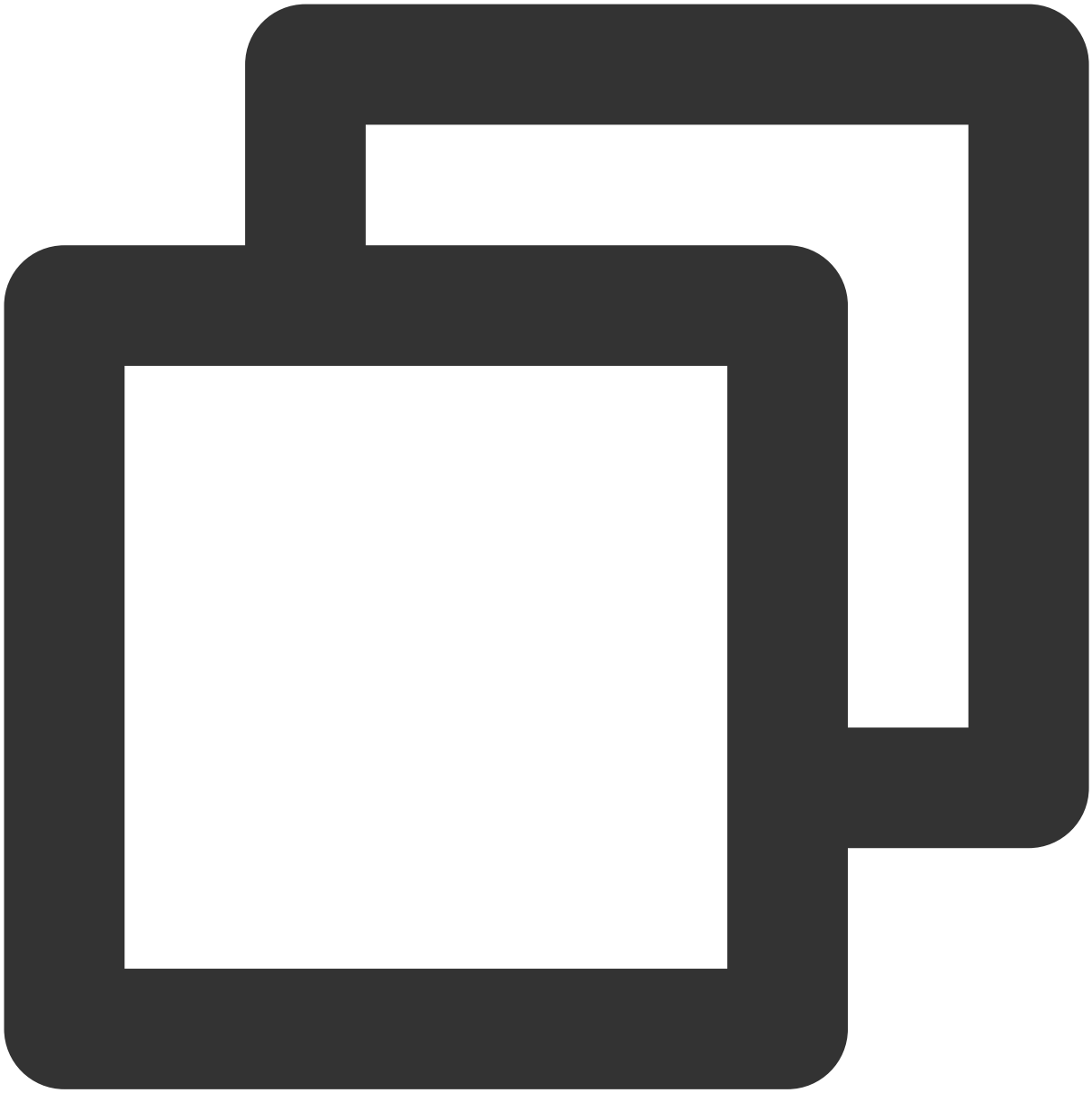
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

图片最终处理效果如下：



案例二：使用管道操作符并携带私有文件签名

处理方式同上，仅增加签名部分，并与处理参数以“&”连接，示例如下：



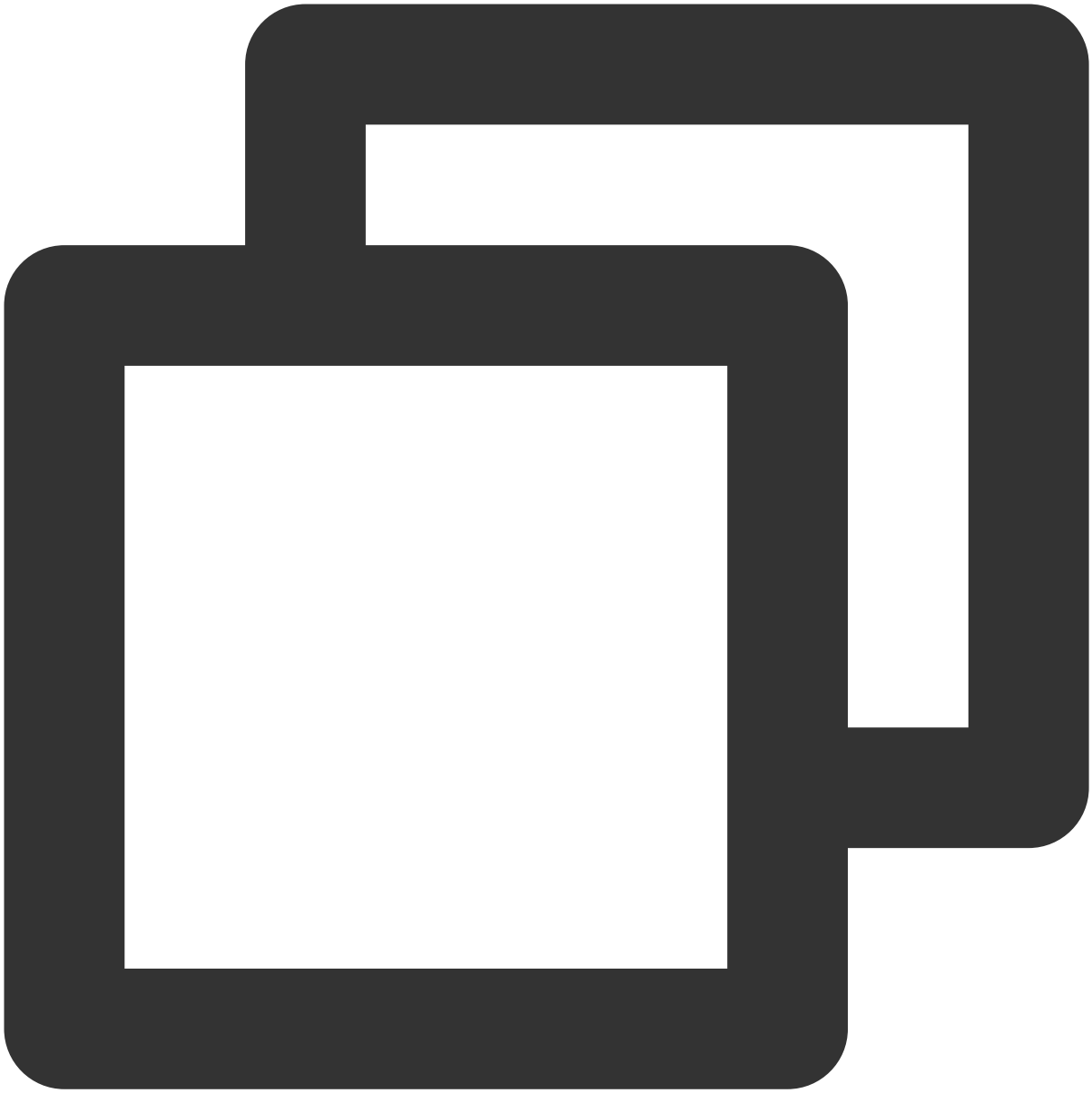
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明

`<signature>` 为签名部分，获取方式请参考 请求签名。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名。



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

图片高级压缩

最近更新时间：2023-12-21 18:08:50

功能描述

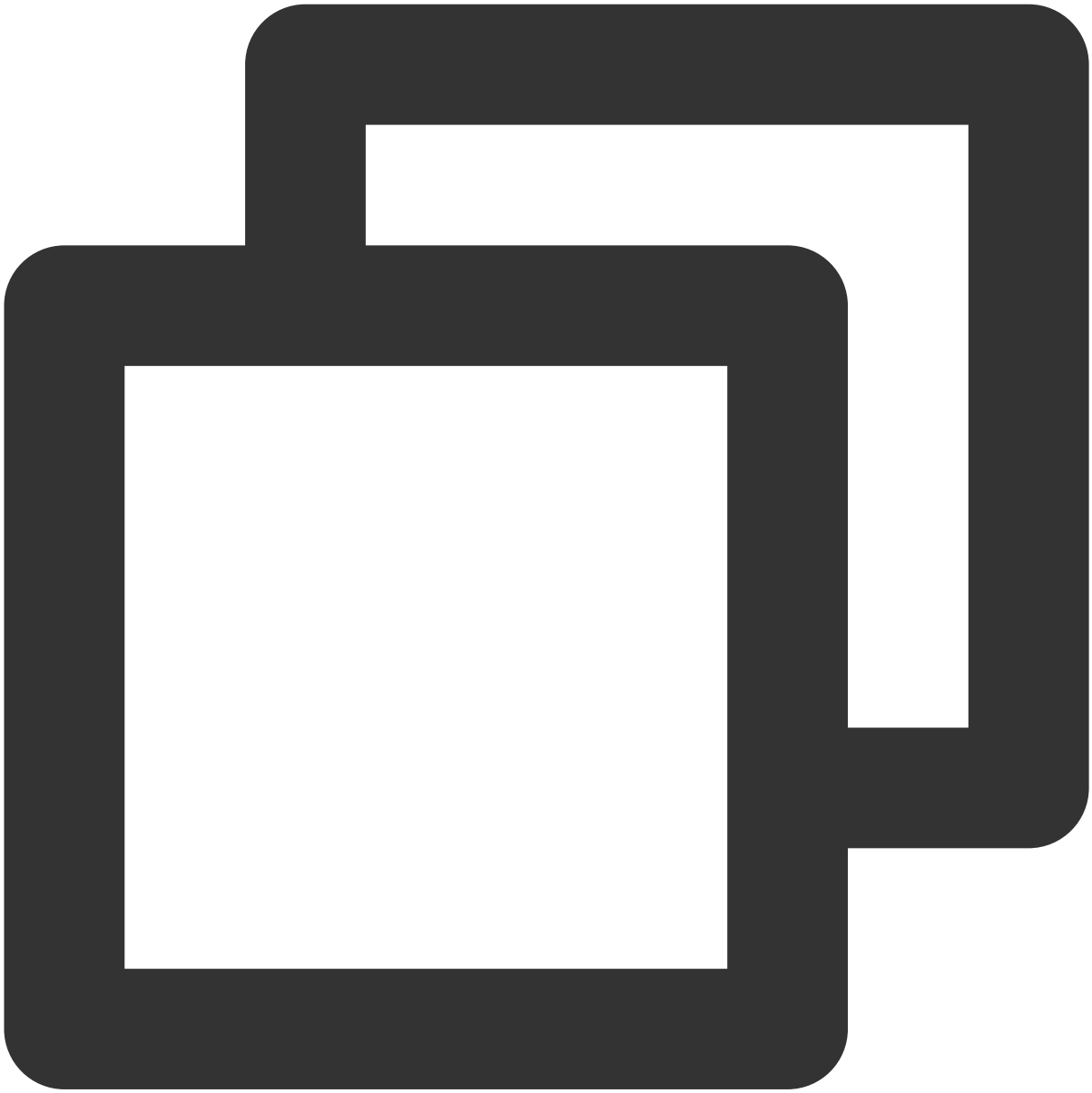
COS 通过数据万象 imageMogr2 接口提供图片高级压缩功能。图片高级压缩可以更加高效地将图片转换为 TPG 或 HEIF 等高压缩比格式，有效降低图片传输链路及加载耗时，降低带宽及流量成本。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理

功能	简介
TPG 压缩	TPG 是腾讯推出的自研图片格式，可将 JPG、PNG、WEBP 等格式图片转换为 TPG 格式，大幅减小图片大小。
HEIF 压缩	针对 iOS 环境的图片使用场景，可将 JPG、PNG、GIF、WEBP 等格式图片转换为 HEIF 格式，HEIF 格式有着超高压缩率。

说明
使用图片高级压缩功能时，您需要在相应的存储桶配置页中通过开关按钮开启服务，详情请参见 设置图片高级压缩。
TPG 是腾讯自研的图片格式，如需使用请确认**图片加载环境支持 TPG 解码**，腾讯云数据万象提供集成 TPG 解码器的 iOS、Android、Windows 终端 SDK，可帮助您快速接入和使用 TPG。
目前 iOS 11以上及 Android P 系统已原生支持 HEIF 格式。
图片高级压缩为付费服务，由数据万象收取，具体费用请参见数据万象的定价文档。

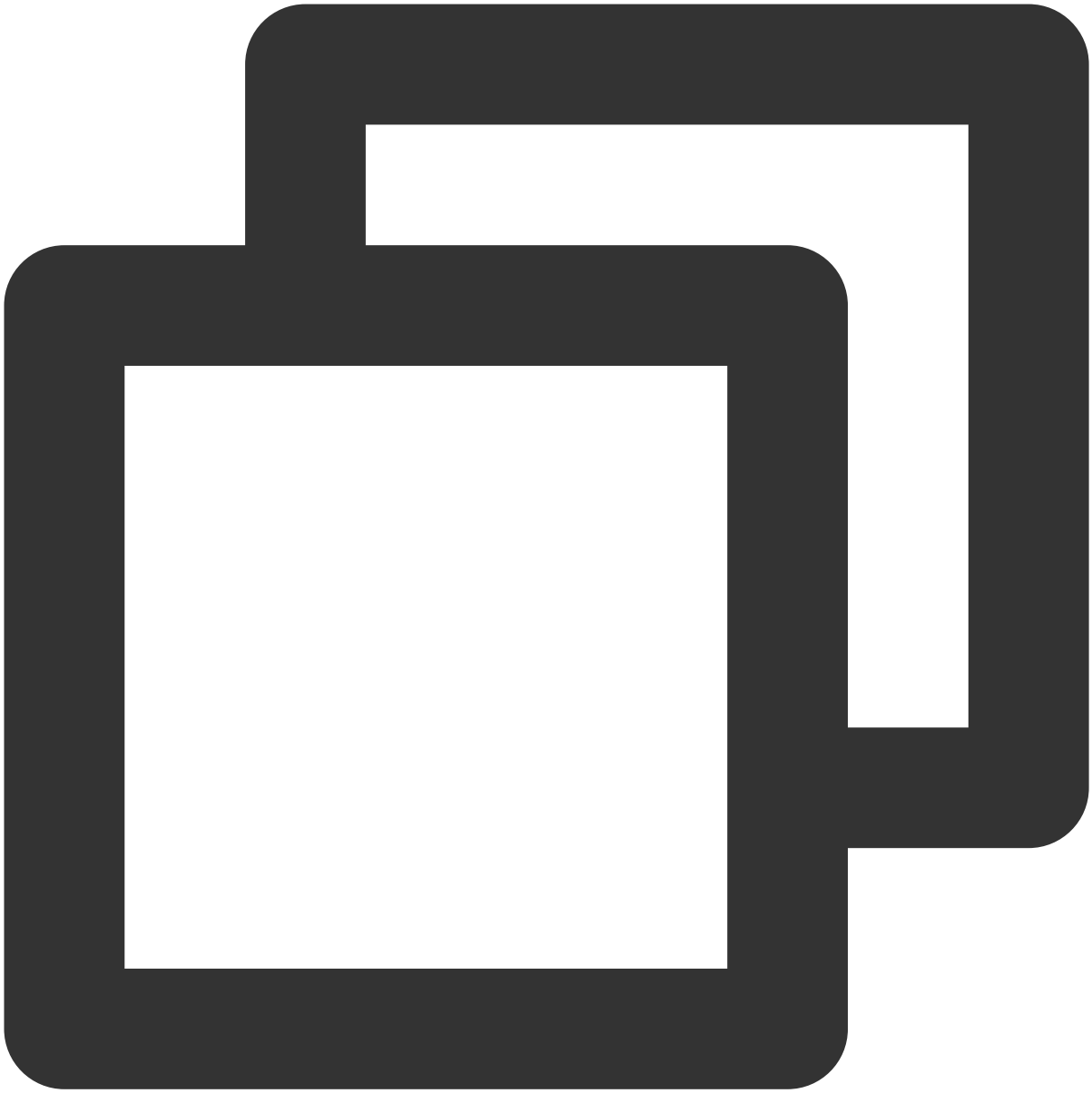
接口示例

1. 下载时处理



```
download_url?imageMogr2/format/<Format>
```

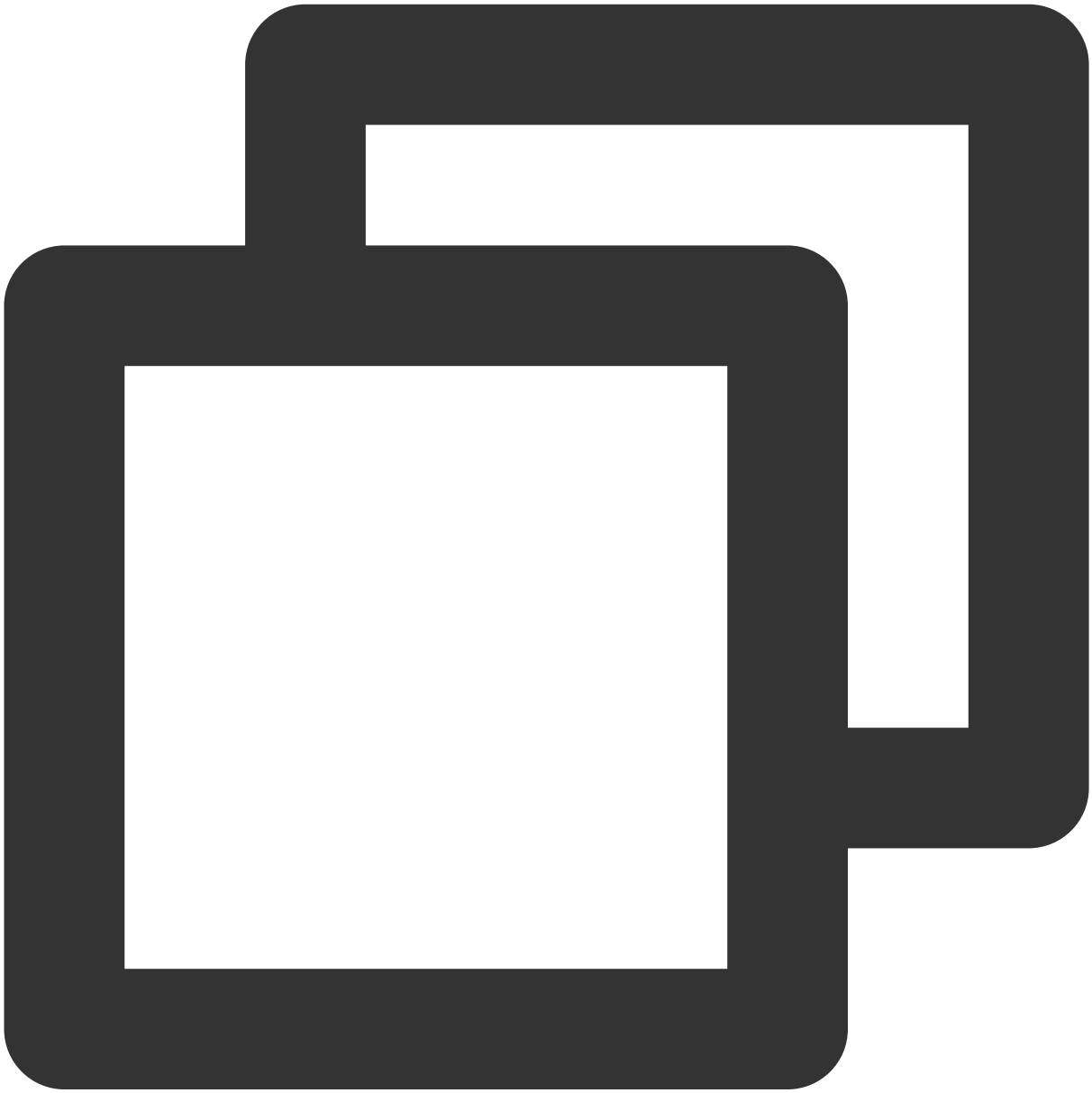
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/format/<Format>"
  }]
```

```
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
```

```
    "is_pic_info": 1,
    "rules": [{
        "fileid": "exampleobject",
        "rule": "imageMogr2/format/<Format>"
    }]
}
```

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅图片持久化处理文档并配置**上传时处理**或**云上数据处理**。

处理参数说明

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>， 例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/format/<Format>	压缩格式，目标缩略图的图片格式为 TPG 或 HEIF。

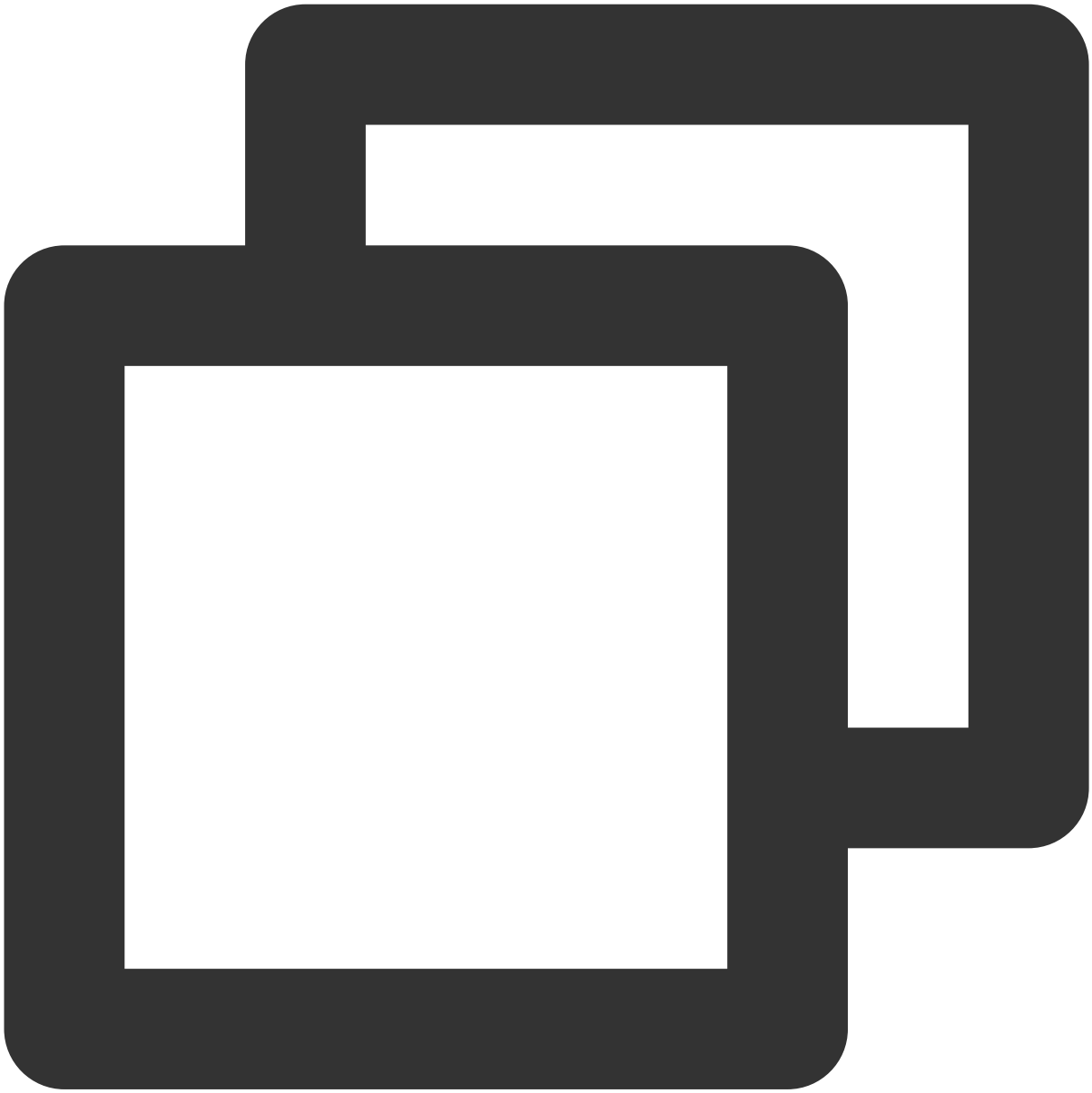
实际案例

假设原图格式为 PNG，图片大小为1335.2KB，如下图所示。



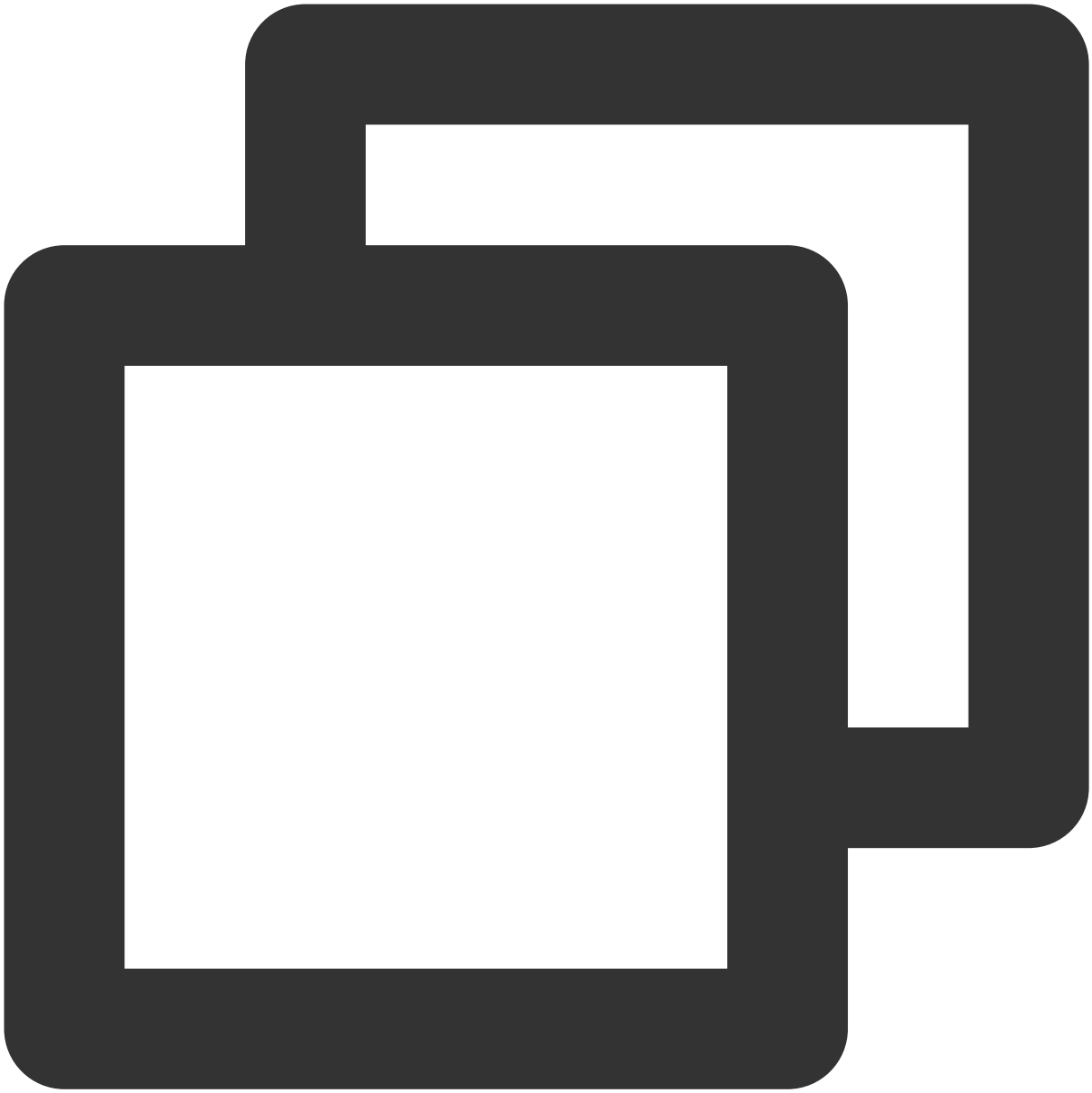
将原图转换为 TPG 格式，URL 地址如下：

版权所有：腾讯云计算（北京）有限责任公司



```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/format
```

将原图转换为 HEIF 格式，URL 地址如下：



```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/format
```

压缩率对比

格式	图片大小
PNG（原图）	1335.2KB
TPG	36.67KB（压缩率97.3%）
HEIF	52.87KB（压缩率96.0%）

版权所有：腾讯云计算（北京）有限责任公司

图片处理机制介绍

最近更新时间：2023-12-21 18:08:58

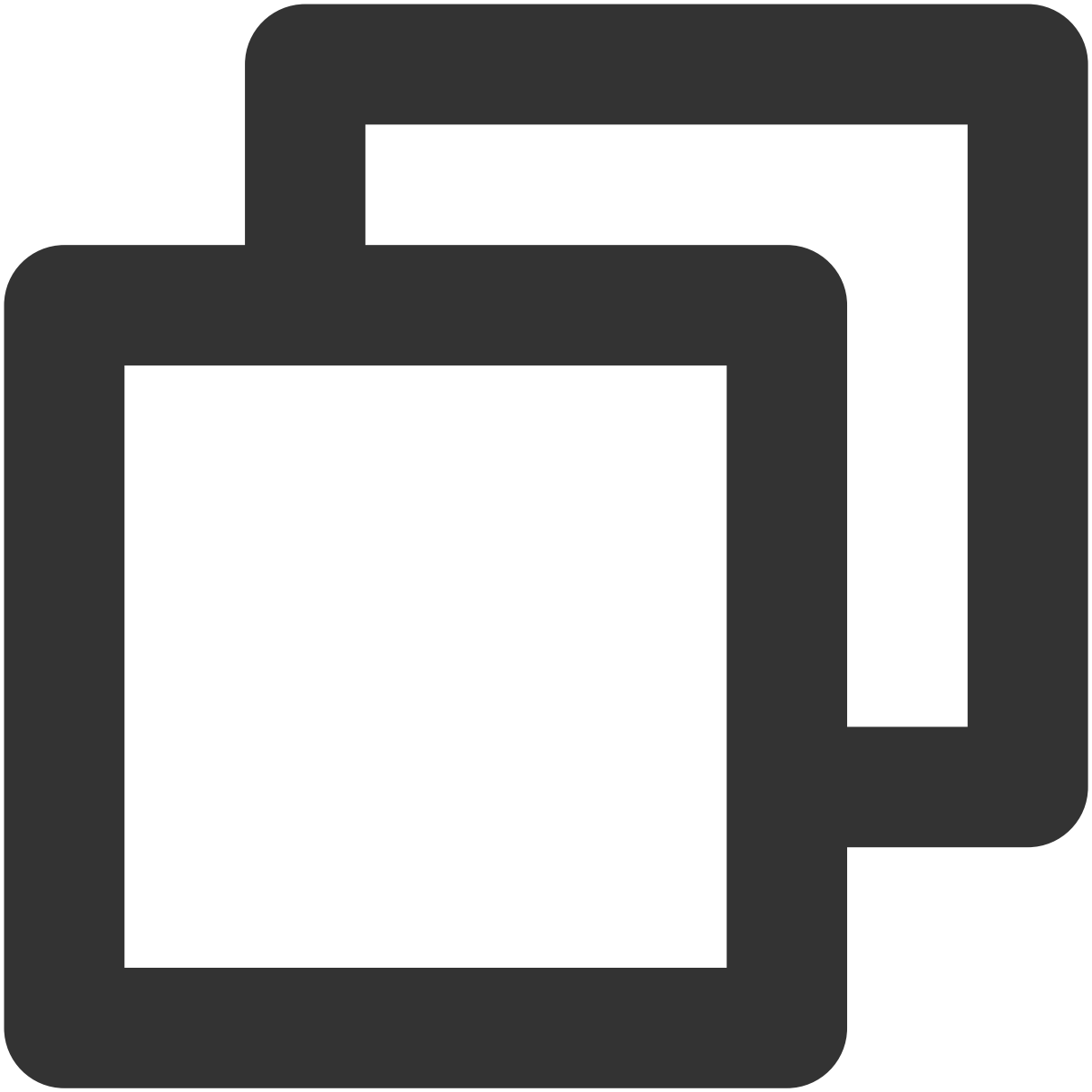
上传时处理

对象存储提供的上传时处理功能可以帮助使用者在上传时实现图片处理。您只需要在请求包头部中加入 Pic-Operations 项并设置好相应参数，就可在图片上传时实现相应的图片处理，并可将原图和处理结果存入到 COS。目前支持20M以内文件处理。

请求语法

图片上传的请求包与 COS V5 简单上传文件接口一致，只在请求包头部增加图片处理参数。

版权所有：腾讯云计算（北京）有限责任公司



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>
```

说明

COS 简单上传文件接口，请参见 COS PUT Object 文档。

Authorization: Auth String（详情请参见 请求签名 文档）。

持久化处理的 QPS 限制为100，如您有更高需求请 联系我们。

请求内容

Pic-Operations 为 json 格式的字符串，具体参数如下：

参数名称	类型	是否必选	描述
is_pic_info	Int	否	是否返回原图信息，0不返回原图信息，1返回原图信息，默认为0
rules	Array	否	处理规则，一条规则对应一个处理结果（目前支持五条规则），不填则不进行图片处理

rules（json 数组）中每一项具体参数如下：

参数名称	类型	是否必选	描述
bucket	String	否	存储结果的目标存储桶名称，格式为 BucketName-APPID，如果不指定的话默认保存到当前存储桶
fileid	String	是	处理后文件的保存路径及名称。名称规则说明： 例如源文件的路径及文件名为：/p1/test 1. jpg1. 以 `/` 开头为绝对路径，例如 fileid 值为 /p2/test2.jpg 时，表示在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 2. 不以 `/` 开头为相对路径，例如 fileid 值为 p2/test2.jpg 时，表示在 p1 文件夹中新建了一个 p2 文件夹，然后在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 3. 注意：请不要以 `/` 结尾，否则会产生空文件名
rule	String	是	处理参数，参见对象存储图片处理 API。若按指定样式处理，则以 `style/` 开头，后加样式名，如样式名为 `test`，则 rule 字段为 `style/test`

返回内容

响应包体具体数据内容如下：

参数名称	类型	描述
UploadResult	Container	原图信息

UploadResult 节点内容：

参数名称	类型	描述
OriginalInfo	Container	原图信息

ProcessResults	Container	图片处理结果

OriginalInfo 节点内容：

节点名称	类型	描述
Key	String	原图文件名
Location	String	图片路径
ImageInfo	Container	原图图片信息
ETag	String	原图 ETag 信息（若处理结果图覆盖原图则为结果图 ETag 信息）

ImageInfo 节点内容：

节点名称	类型	描述
Format	String	格式
Width	Int	图片宽度
Height	Int	图片高度
Quality	Int	图片质量
Ave	String	图片主色调
Orientation	Int	图片旋转角度

ProcessResults 节点内容：

节点名称	类型	描述
Object	Container	每一个图片处理结果

Object 节点内容：

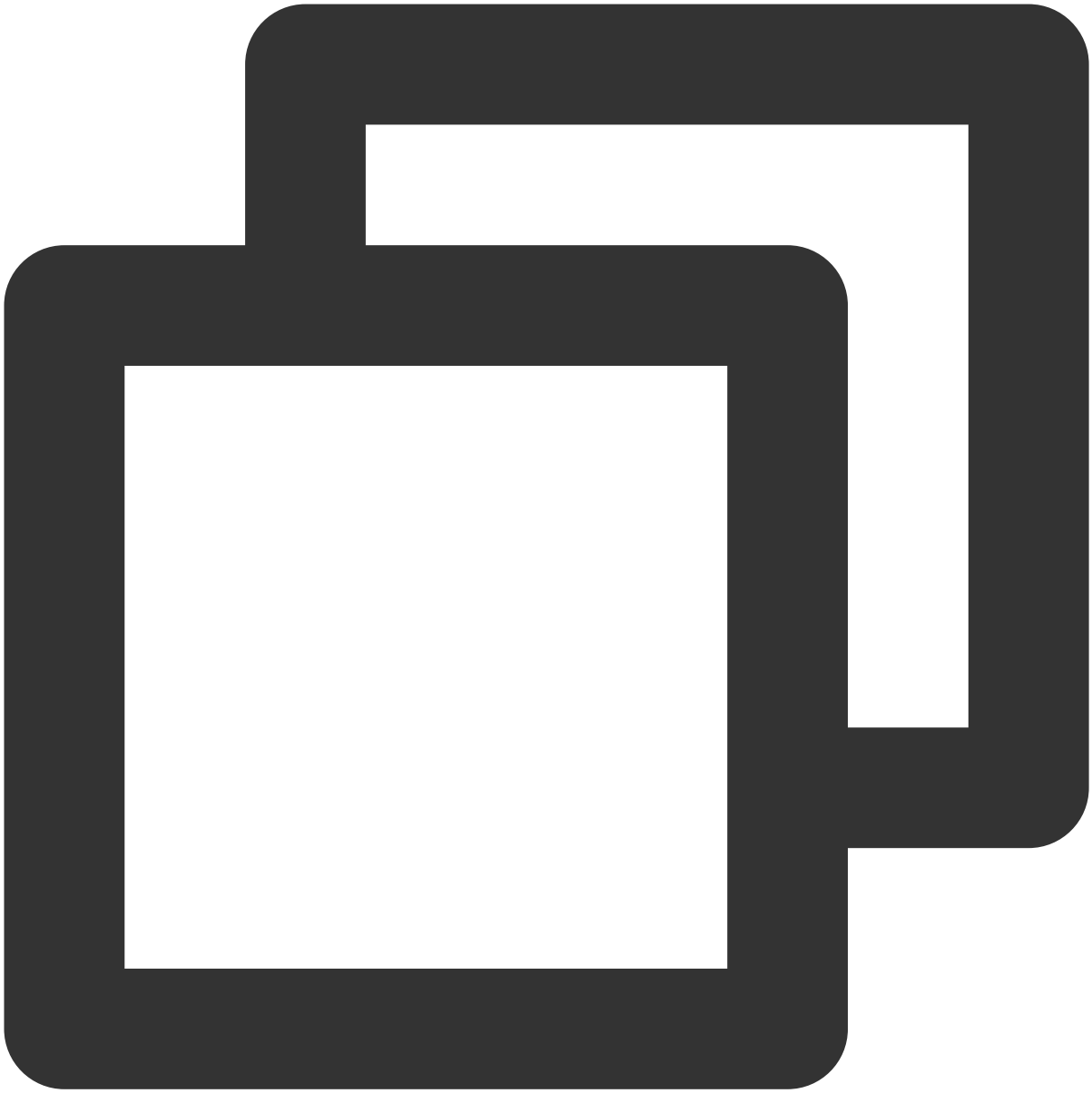
节点名称	类型	描述
Key	String	文件名
Location	String	图片路径
Format	String	图片格式

版权所有：腾讯云计算（北京）有限责任公司

Width	Int	图片宽度
Height	Int	图片高度
Size	Int	图片大小
Quality	Int	图片质量
ETag	String	处理结果图 ETag 信息

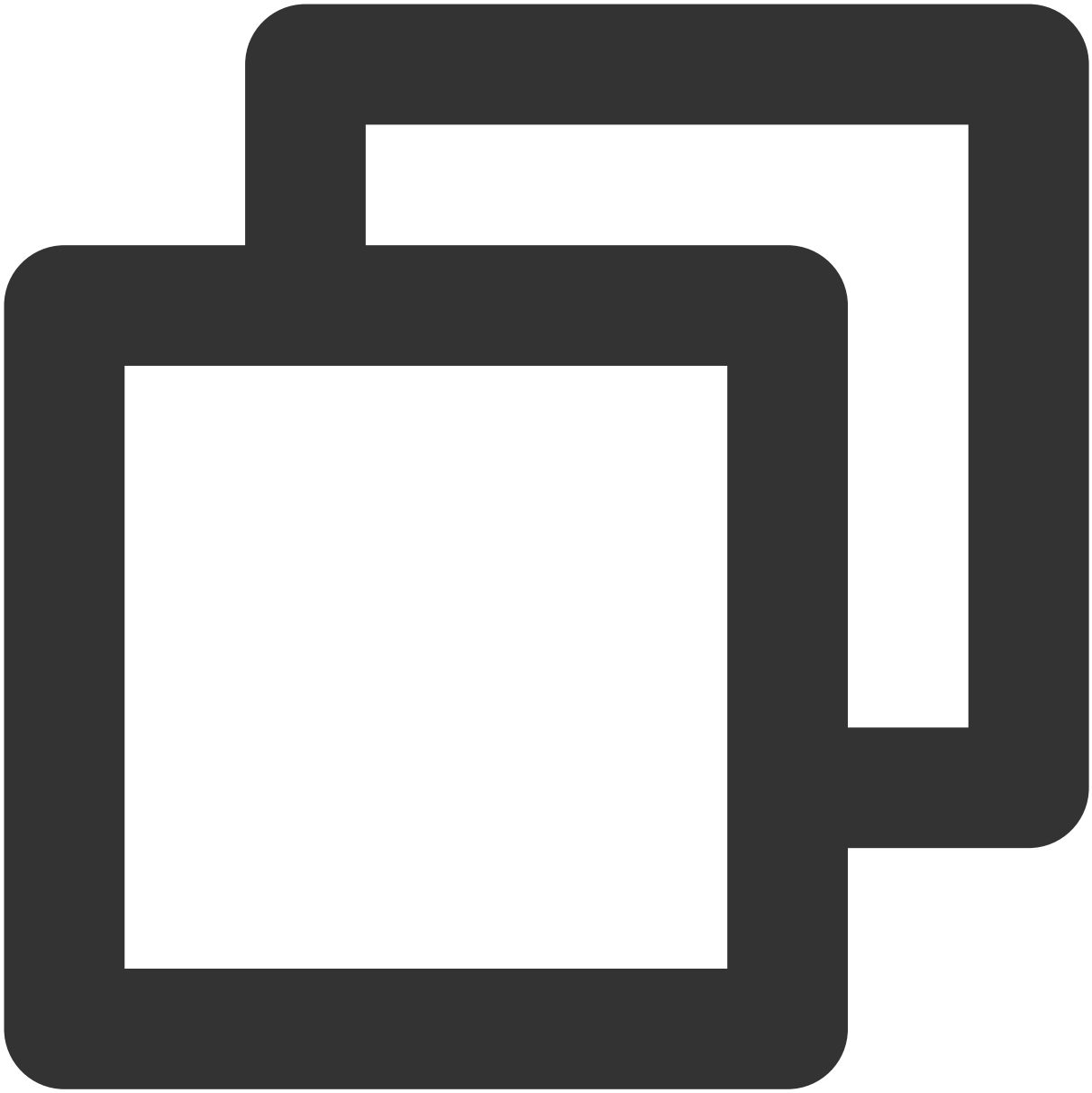
示例

请求



```
PUT /filename.jpg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Wed, 28 Oct 2015 20:32:00 GMT
Authorization:XXXXXXXXXXXX
Pic-Operations: {"is_pic_info":1,"rules":[{"fileid":"test.jpg","rule":"imageView2/f
Content-Length: 64

[Object]
```

返回

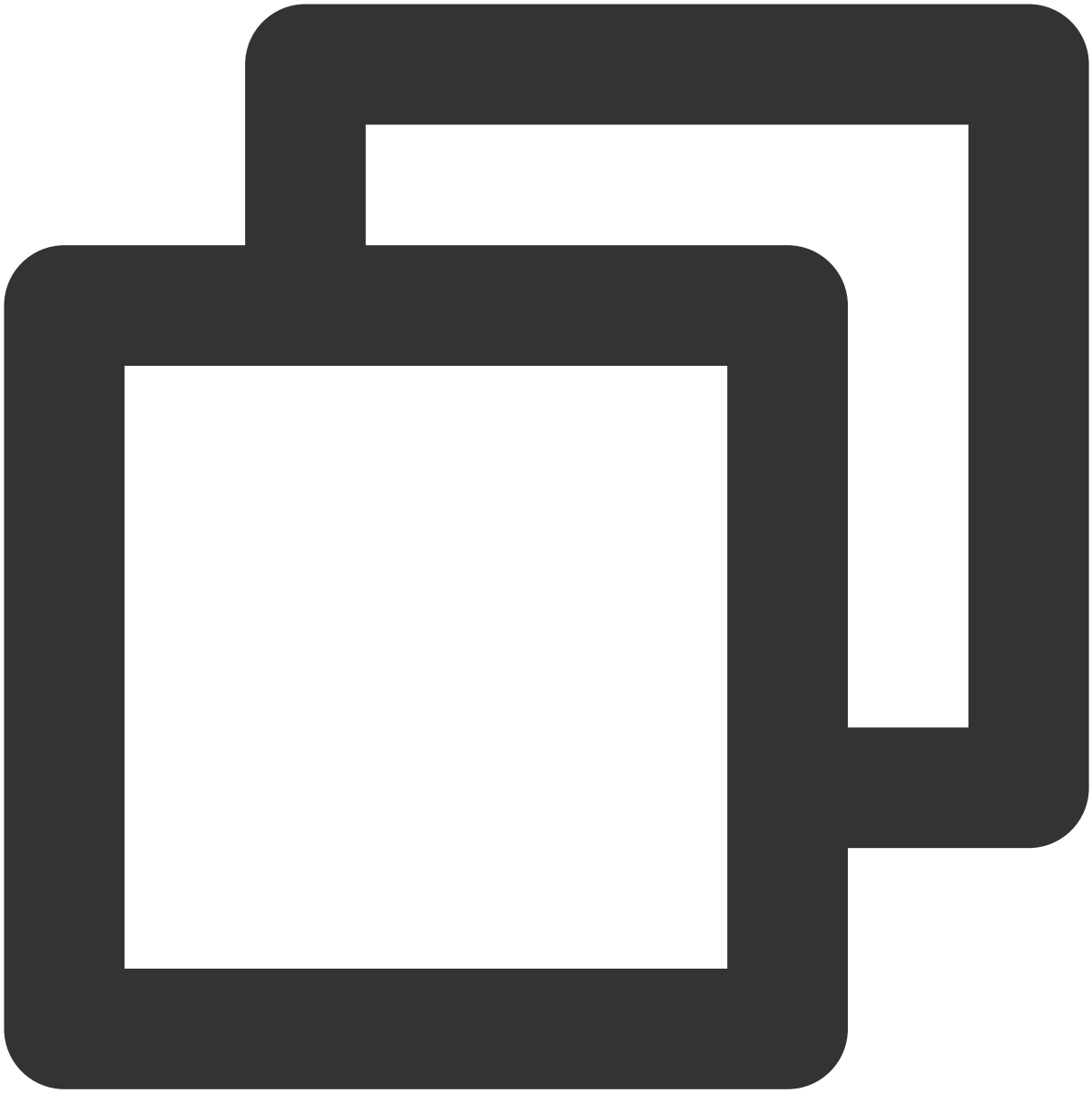


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
        <OriginalInfo>
            <Key>filename.jpg</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename
```

```
            <ETag>&amp;quot;580cd6930444576523c25f86ce2af9b1fc2d5484&amp;quot;</ETag>
            <ImageInfo>
                <Format>JPEG</Format>
                <Width>640</Width>
                <Height>427</Height>
                <Quality>100</Quality>
                <Ave>0xa18454</Ave>
                <Orientation>1</Orientation>
            </ImageInfo>
        </OriginalInfo>
        <ProcessResults>
            <Object>
                <Key>test.jpg</Key>
                <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test
                <Format>png</Format>
                <Width>640</Width>
                <Height>427</Height>
                <Size>463092</Size>
                <Quality>100</Quality>
                <ETag>&amp;quot;eaa4e3d8fd498bbc63be3b71c46b9c61f23e3f0c&amp;quot;</ETag>
            </Object>
        </ProcessResults>
</UploadResult>
```

注意

上传时处理支持 COS V5 的分块上传功能，您在使用 COS V5 的 Complete Multipart Upload 接口时只需在请求包头部中加入 Pic-Operations 项，即可实现图片处理。



```
POST /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations: <PicOperations>
```

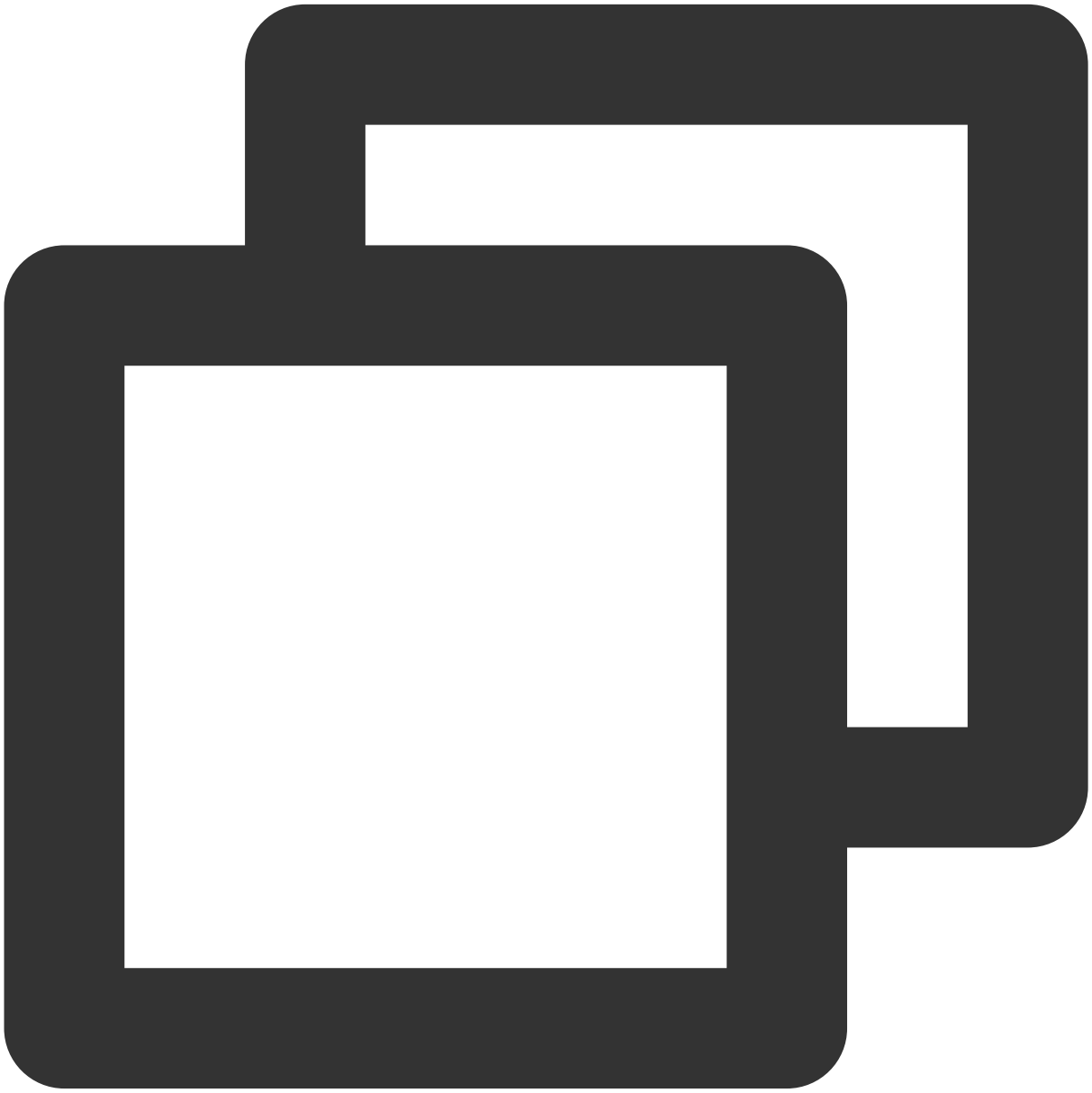
说明

COS 接口请参见 COS Complete Multipart Upload 文档。

云上数据处理

对象存储的图片处理 API 能够对已存储在 COS 的图片进行相应处理操作，并将结果存入到 COS。

请求语法



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
```

```
Authorization: Auth String
Pic-Operations: <PicOperations>
```

请求内容

Pic-Operations 为 json 格式的字符串，具体参数如下：

参数名称	类型	是否必选	描述
is_pic_info	Int	否	是否返回原图信息，0不返回原图信息，1返回原图信息，默认为0
rules	Array	否	处理规则，一条规则对应一个处理结果（目前支持五条规则），不填则不进行图片处理

rules（json 数组）中每一项具体参数如下：

参数名称	类型	是否必选	描述
bucket	String	否	存储结果的目标 bucket 名称，形如 BucketName-APPID，如果不指定的话默认保存到当前存储桶
fileid	String	是	处理后文件的保存路径及名称。名称规则说明： 例如源文件的路径及文件名为：/p1/test1.jpg 1. 以/开头为绝对路径，例如 fileid 值为 /p2/test2.jpg 时，表示在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 2. 不以/开头为相对路径，例如 fileid 值为 p2/test2.jpg 时，表示在 p1 文件夹中新建了一个 p2 文件夹，然后在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 3. 注意：请不要以/结尾，否则会产生空文件名
rule	String	是	处理参数，参见对象存储图片处理 API。若按指定样式处理，则以 `style/` 开头，后加样式名，如样式名为 `test`，则 rule 字段为 `style/test`

返回内容

响应包体具体数据内容如下：

参数名称	类型	描述
UploadResult	Container	原图信息

UploadResult 节点内容：

参数名称	类型	描述

OriginalInfo	Container	原图信息
ProcessResults	Container	图片处理结果

OriginalInfo 节点内容：

节点名称	类型	描述
Key	String	原图文件名
Location	String	图片路径
ImageInfo	Container	原图图片信息
ETag	String	原图 ETag 信息（若处理结果图覆盖原图则为结果图 ETag 信息）

ImageInfo 节点内容：

节点名称	类型	描述
Format	String	格式
Width	Int	图片宽度
Height	Int	图片高度
Quality	Int	图片质量
Ave	String	图片主色调
Orientation	Int	图片旋转角度

ProcessResults 节点内容：

节点名称	类型	描述
Object	Container	每一个图片处理结果

Object 节点内容：

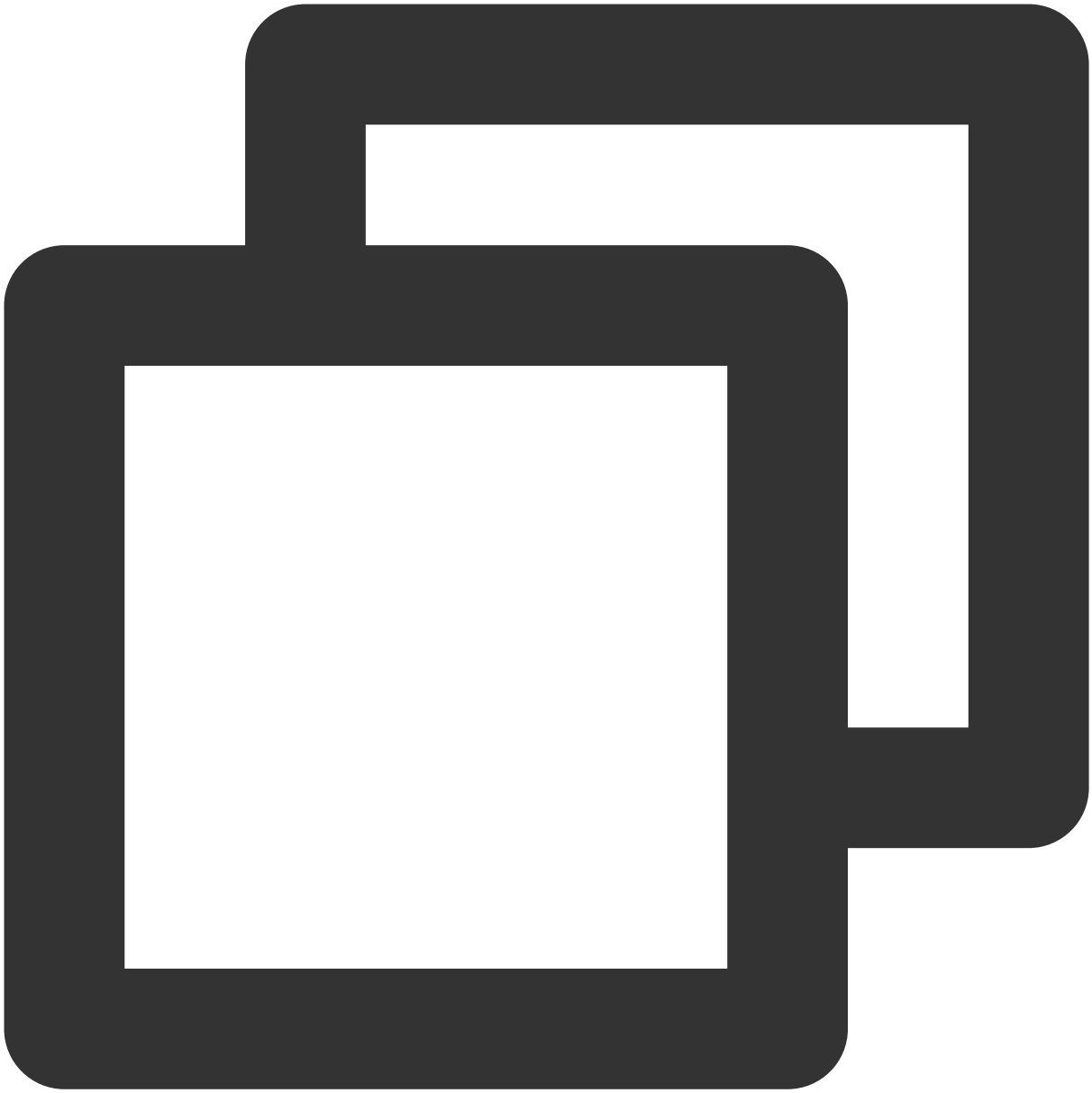
节点名称	类型	描述
Key	String	文件名
Location	String	图片路径
Format	String	图片格式

版权所有：腾讯云计算（北京）有限责任公司

Width	Int	图片宽度
Height	Int	图片高度
Size	Int	图片大小
Quality	Int	图片质量
ETag	String	处理结果图 ETag 信息

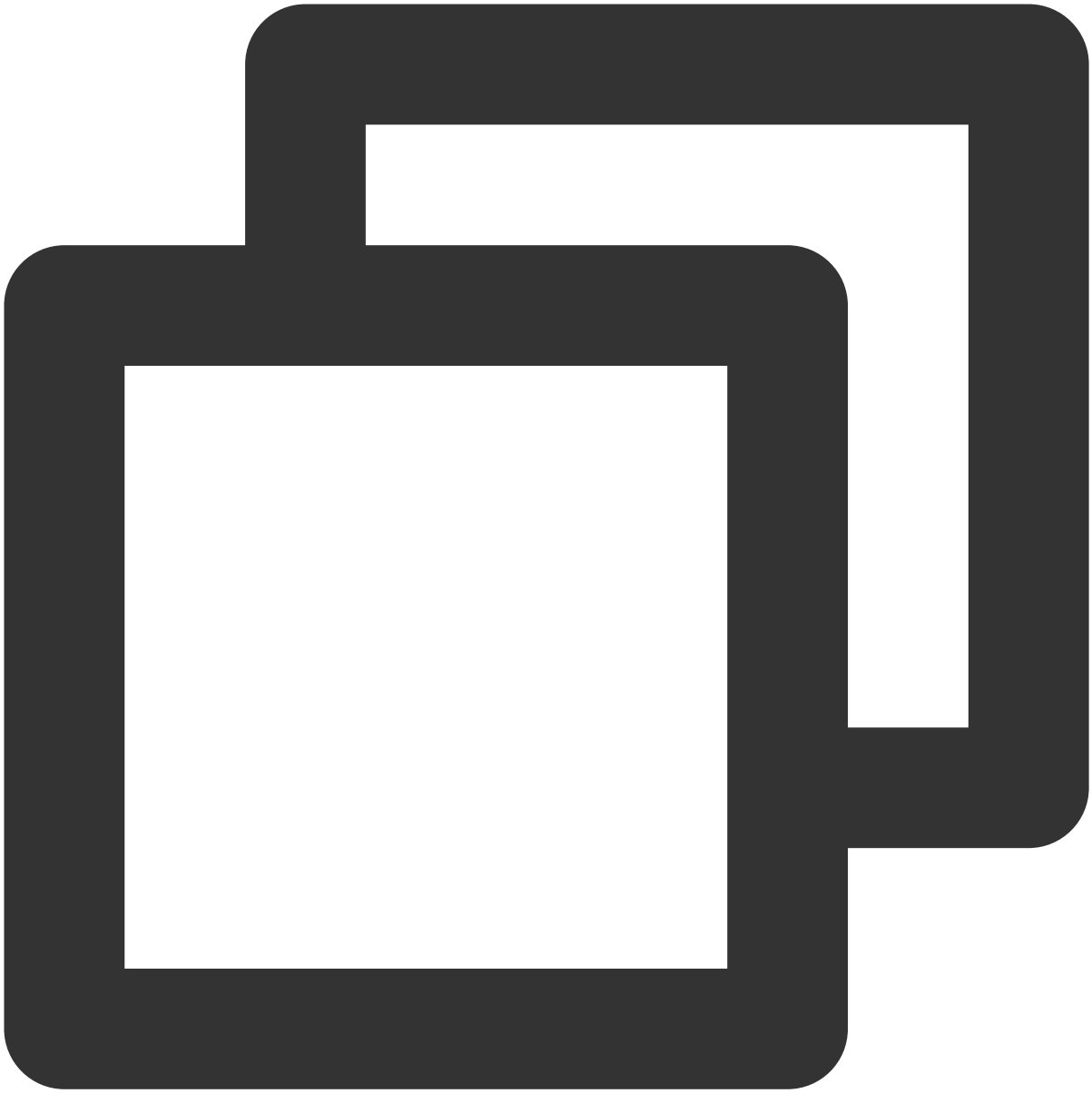
示例

请求



```
POST /filename.jpg?image_process HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization: XXXXXXXXXX
Pic-Operations: {"is_pic_info":1,"rules":[{"fileid":"test.jpg","rule":"imageView2/f
Content-Length: XX
```

返回



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
            <OriginalInfo>
                <Key>filename.jpg</Key>
                <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/file
```

```
                <ETag>&amp;quot;eaa4e3d8fd498bbc63be3b71c46b9c61f23e3f0c&amp;quot;</ETag>
                <ImageInfo>
                    <Format>JPEG</Format>
                    <Width>640</Width>
                    <Height>427</Height>
                    <Quality>100</Quality>
                    <Ave>0xa18454</Ave>
                    <Orientation>1</Orientation>
                </ImageInfo>
            </OriginalInfo>
            <ProcessResults>
                <Object>
                    <Key>test.jpg</Key>
                    <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/
                    <Format>png</Format>
                    <Width>640</Width>
                    <Height>427</Height>
                    <Size>463092</Size>
                    <Quality>100</Quality>
                    <ETag>&amp;quot;eaa4e3d8fd498bbc63be3b71c46b9c61f23e3f0c&amp;quot;</ETa
                </Object>
            </ProcessResults>
</UploadResult>
```

图片压缩
WebP 压缩

最近更新时间：2023-12-22 10:51:51

功能描述

图片压缩指在图片质量保持不变的情况，尽可能的减小图片大小，以达到节省图片存储空间、减少图片访问流量、提升图片访问速度的效果。
对象存储（Cloud Object Storage，COS）基于 数据万象（Cloud Infinite，CI） 产品推出了 WebP 压缩功能，可将图片转换为 webp 压缩图片格式，其在压缩方面相比 jpg 格式更优越。在相同图片质量的情况下，webp 格式图片要比 jpg 格式图片减小25%以上，可以适配多终端使用场景。

限制说明

支持格式：支持将 jpg、png、bmp、gif、tpg、heif、avif 等格式图片转换为 webp 格式。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

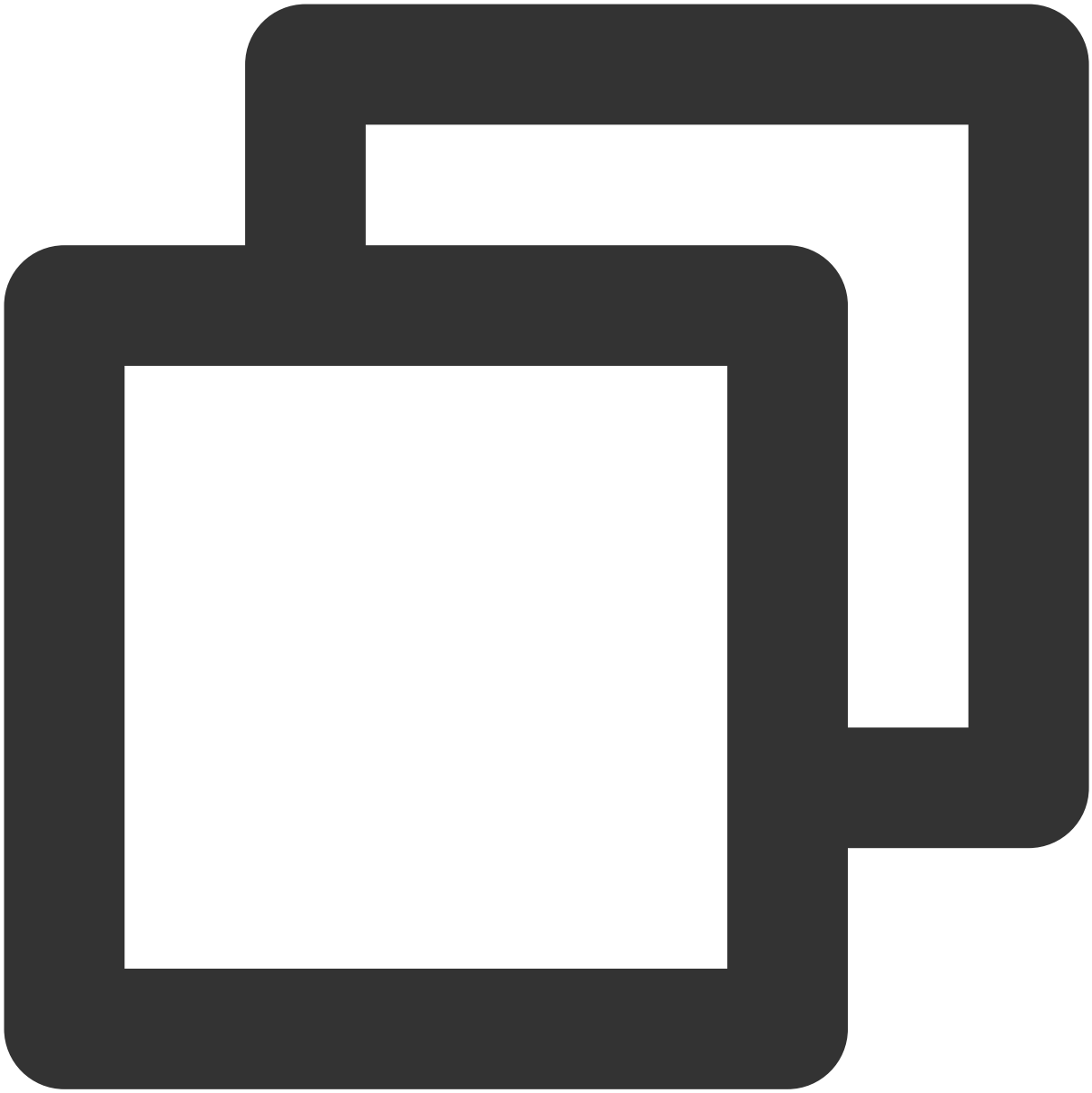
使用方式

COS 通过数据万象 imageMogr2 接口提供 WebP 压缩功能。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理
说明
WebP 压缩为付费服务，费用同基础图片处理，由数据万象收取，具体费用请参见数据万象 图片处理费用。
图片转换为 WebP 格式后，部分浏览器无法读取 WebP 图片的 exif 信息，导致没有旋转。您可参见 旋转 文档，增加 auto-orient 参数，对原图旋转后再进行压缩。
WebP 压缩默认继承原始图片的质量参数。您可参见 质量变换 文档，通过修改图片质量来调节压缩率。

接口示例

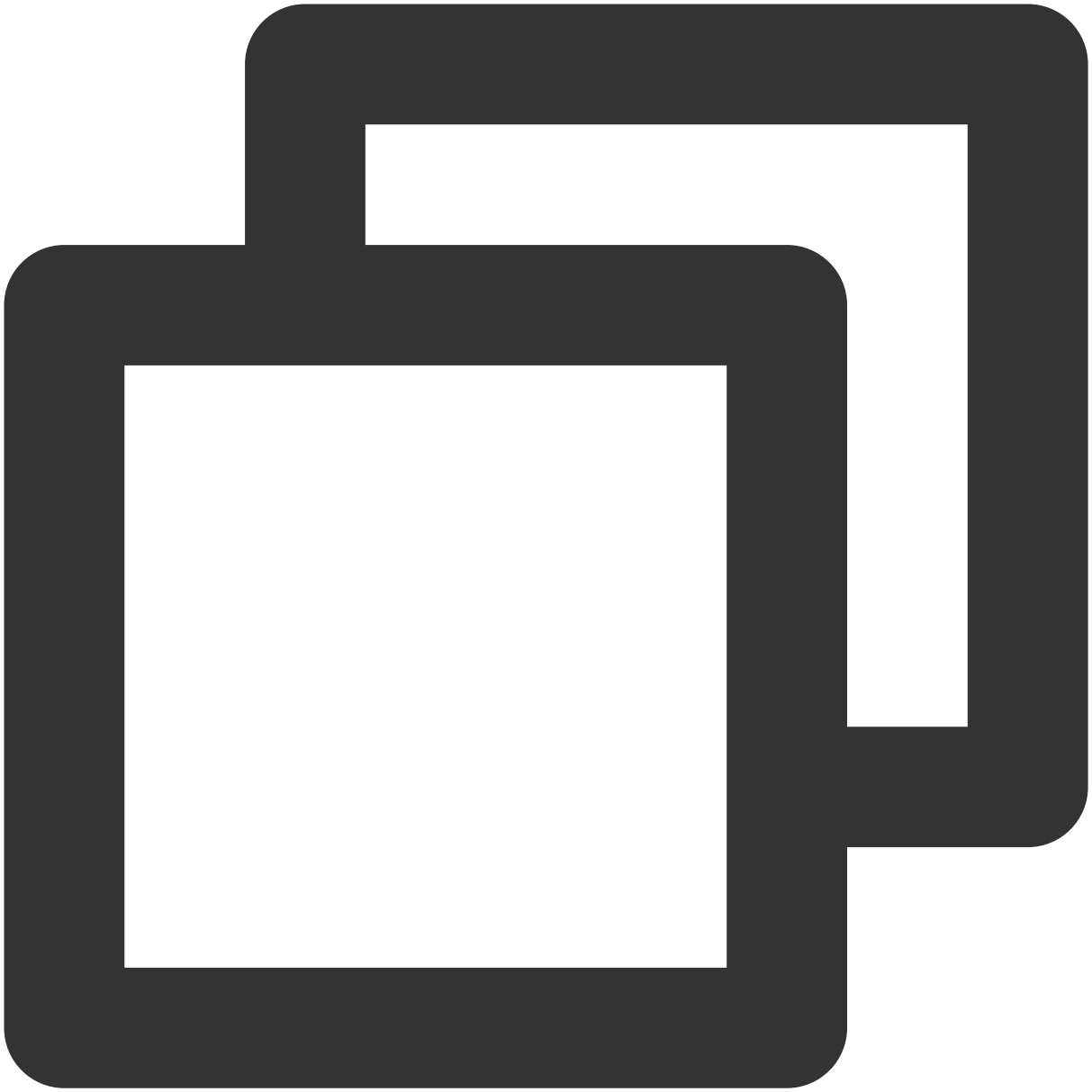
版权所有：腾讯云计算（北京）有限责任公司

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/format/webp HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



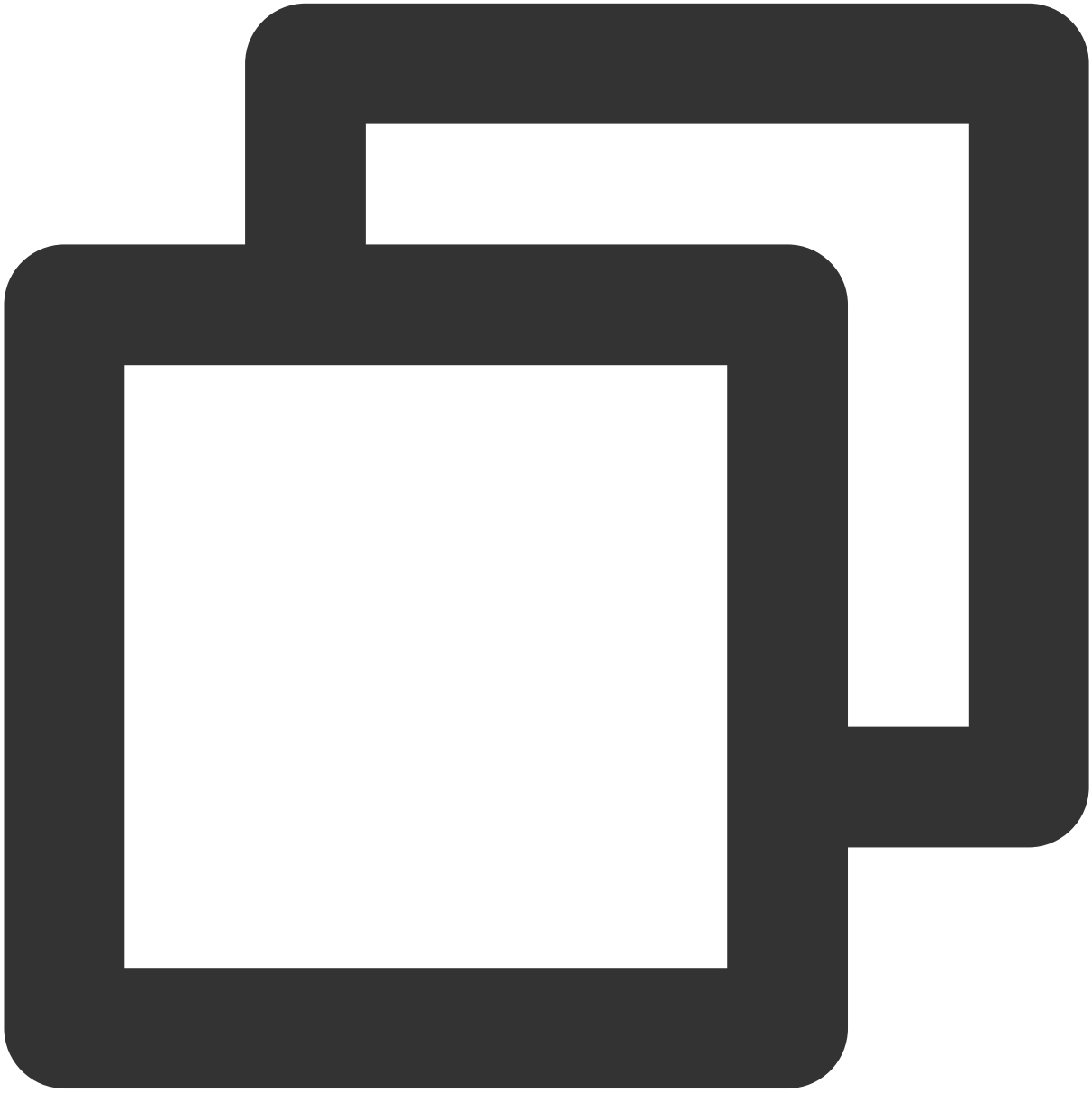
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/webp"
}]
```

```
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
```

```
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/webp"
}]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/format/<Format>	压缩格式，此处为 webp。

实际案例

说明

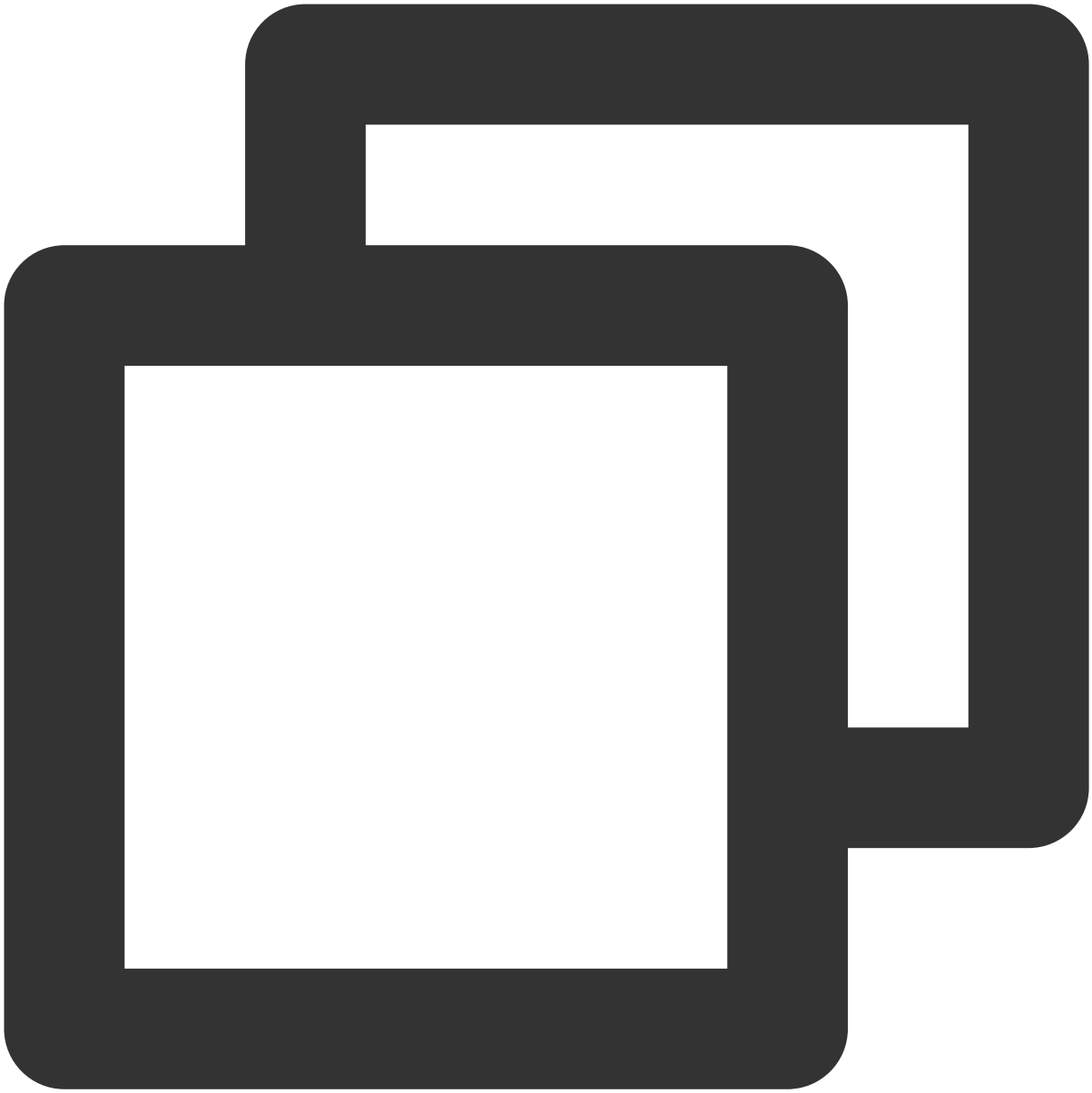
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，请使用**上传时处理**或**云上数据处理**方式。

假设原图格式为 png，图片大小为1335.2KB，如下图所示：



将原图转换为 webp 格式，请求 URL 如下：

版权所有：腾讯云计算（北京）有限责任公司



```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/format
```

效果如下：



压缩率对比

格式	图片大小
png（原图）	1335.2KB
webp	65KB（压缩率95.13%）

版权所有：腾讯云计算（北京）有限责任公司

HEIF 压缩

最近更新时间：2023-12-22 10:51:58

功能描述

图片压缩指在图片质量保持不变的情况，尽可能的减小图片大小，以达到节省图片存储空间、减少图片访问流量、提升图片访问速度的效果。
对象存储（Cloud Object Storage，COS）基于 数据万象（Cloud Infinite，CI）产品推出了 HEIF 压缩功能，可将图片转换为 heif 格式，它具有超高的压缩率，在图片质量相同的情况下，图片大小比 jpg 格式图片减小80%以上，iOS 系统已将 heif 图片作为相册默认格式，Android P 系统已原生支持 heif。

限制说明

支持格式：支持将 jpg、png、bmp、webp、tpg、avif 等格式图片转换为 heif 格式。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

前提条件

使用 HEIF 压缩，需要先开通存储桶的图片高级压缩功能，您需要在相应的存储桶配置页中通过开关按钮开启功能，详情请参见 设置图片高级压缩。

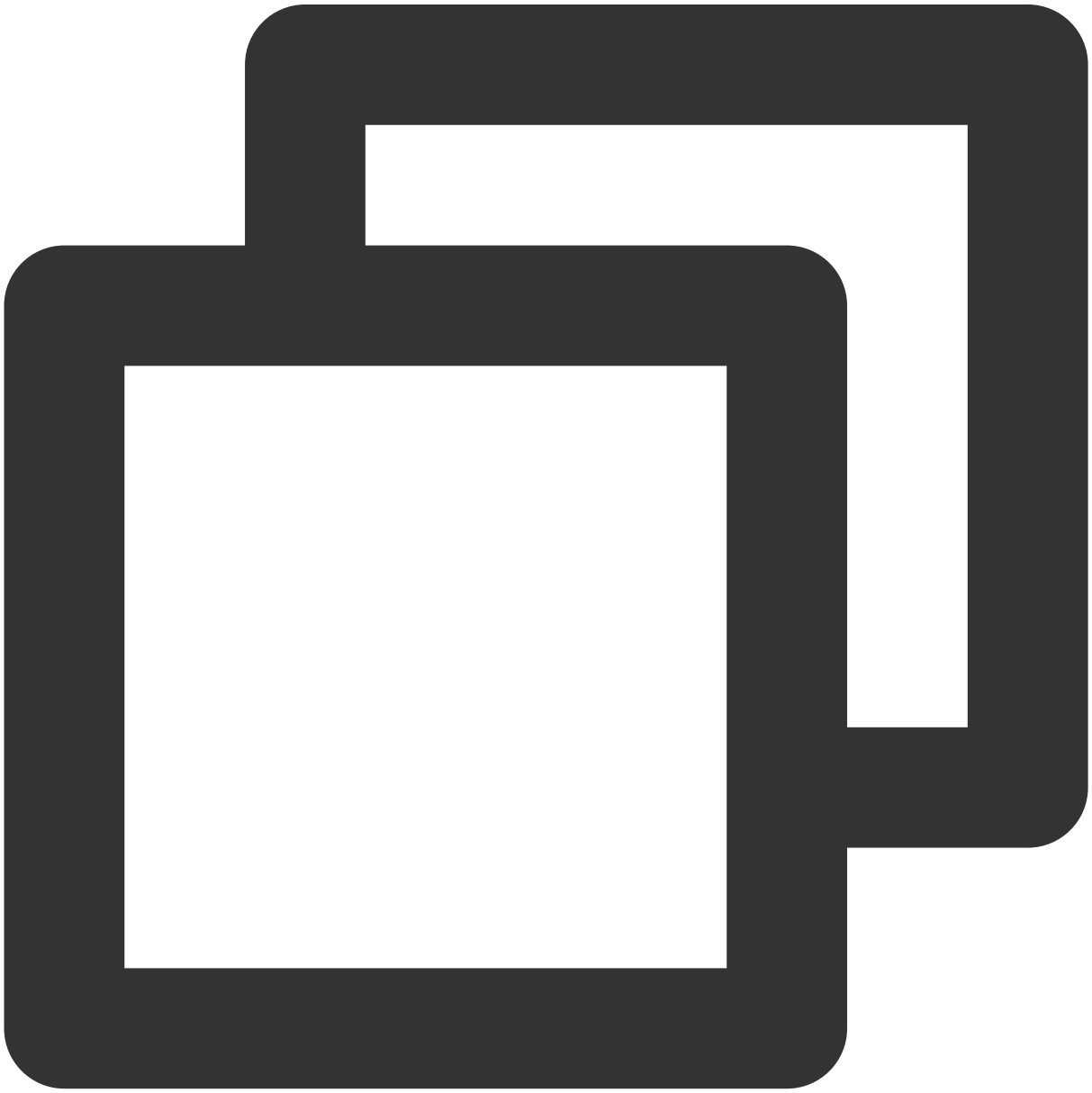
使用方式

COS 通过数据万象 imageMogr2 接口提供 HEIF 压缩功能。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理
说明
HEIF 压缩为付费服务，计费项为图片高级压缩费用，由数据万象收取，具体费用请参见数据万象图片处理费用。

版权所有：腾讯云计算（北京）有限责任公司

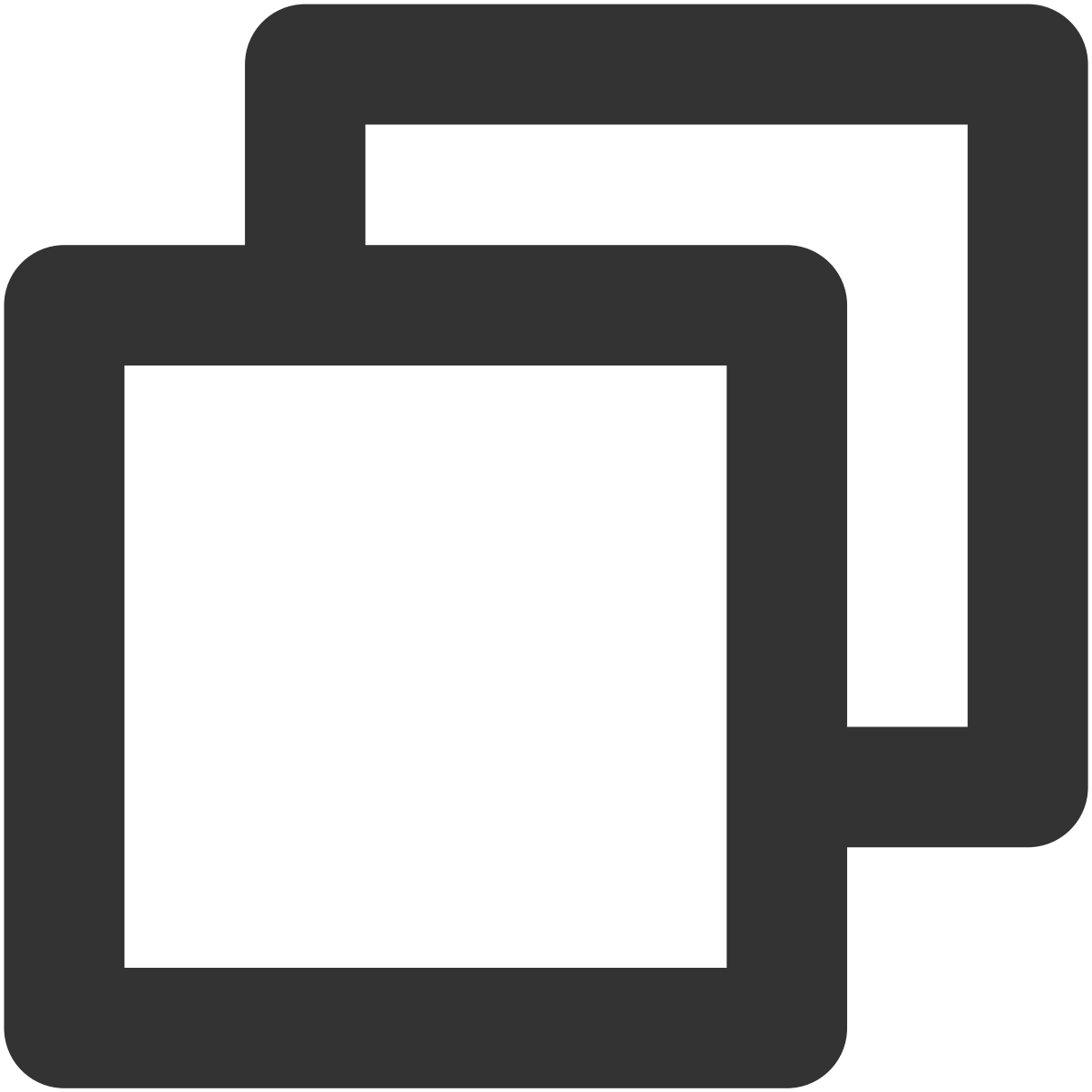
接口示例

1. 下载时处理



```
download_url?imageMogr2/format/heif
```

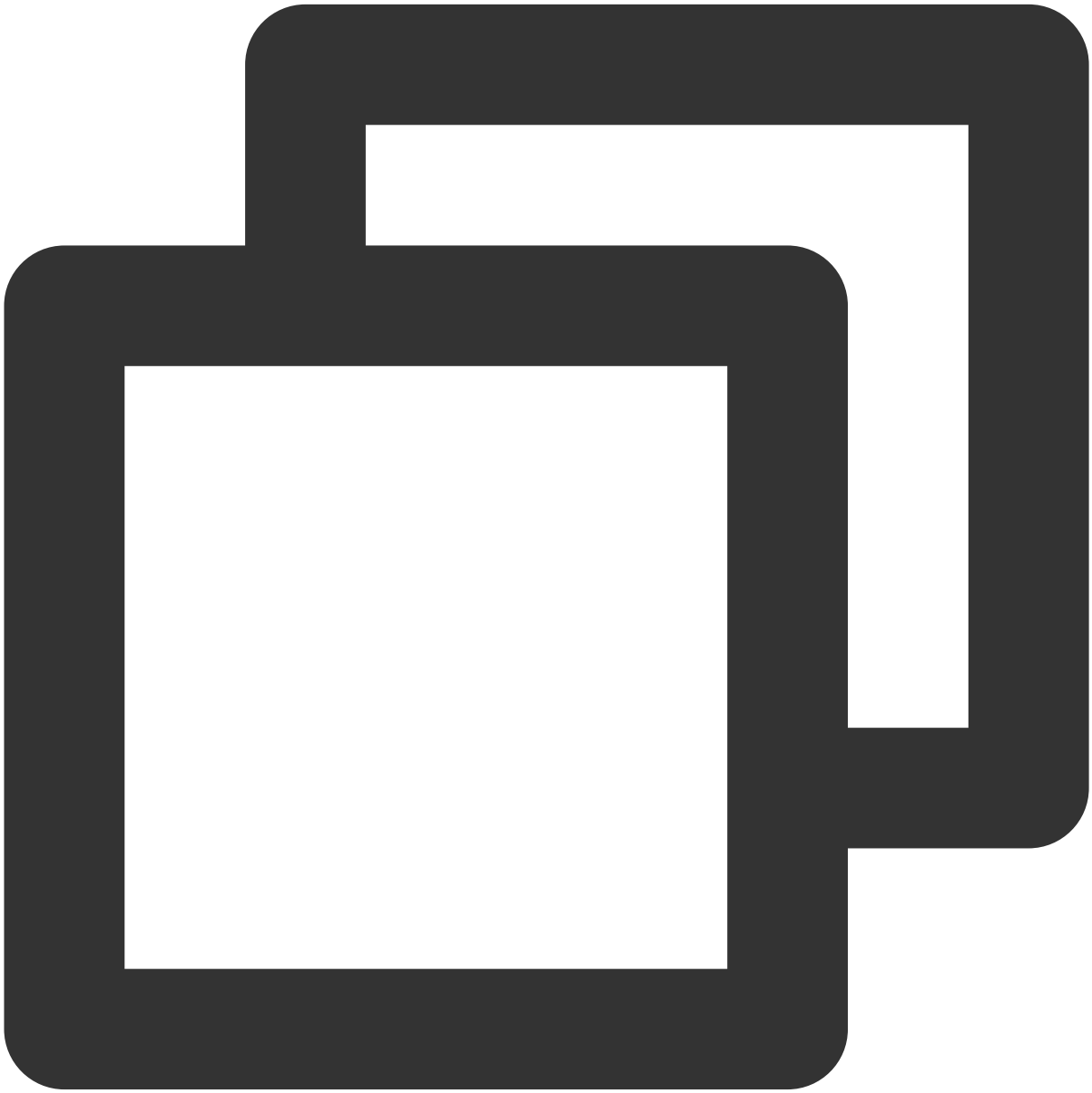
2. 上传时处理



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/heif"
}]
```

```
}
```

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
```

```
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/heif"
}]
}
```

说明

本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，请使用**上传时处理**或**云上数据处理**方式。

处理参数说明

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>， 例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/format/<Format>	压缩格式，此处为 heif。

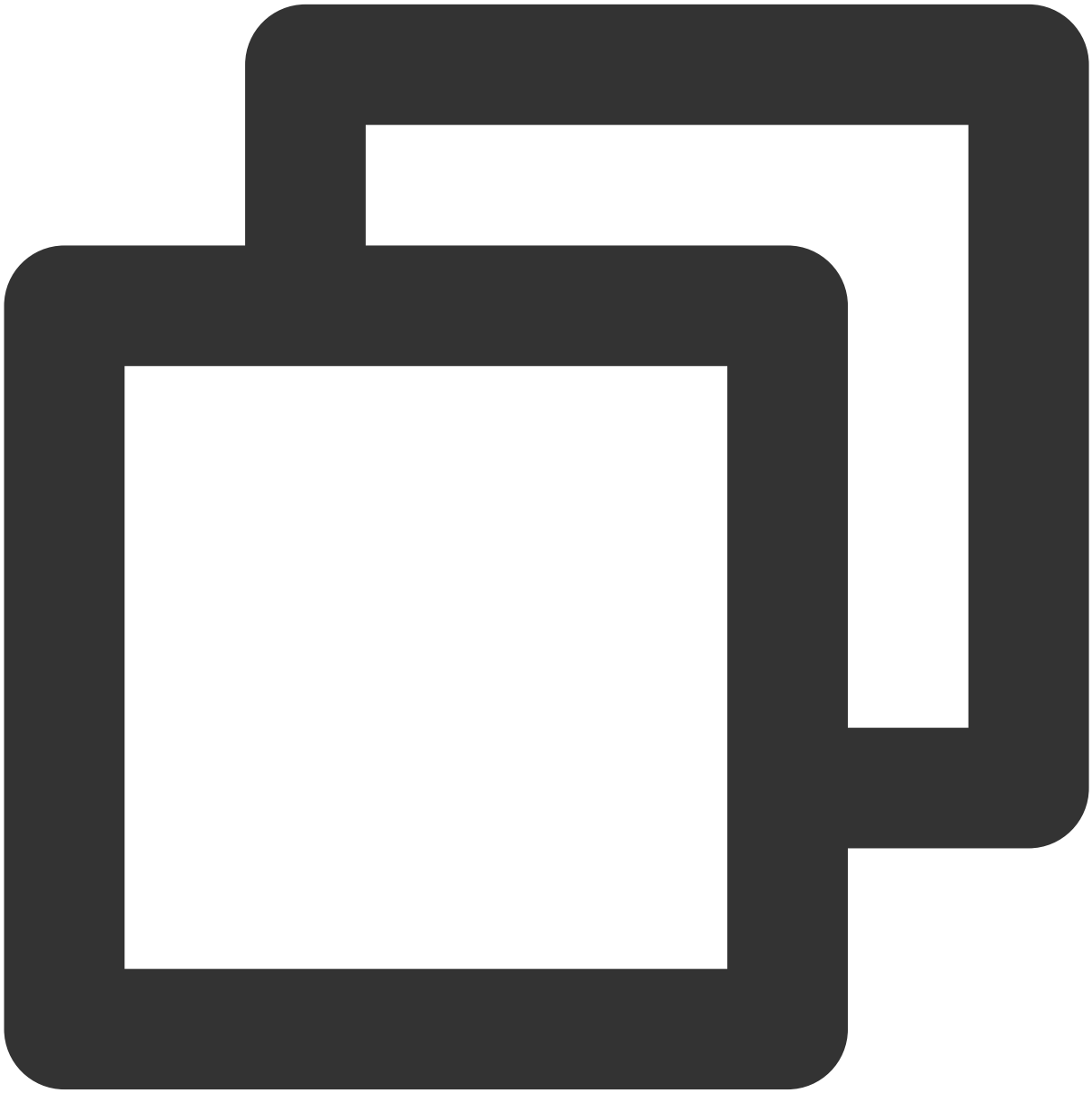
实际案例

假设原图格式为 png，图片大小为1335.2KB，如下图所示。



将原图转换为 heif 格式，请求 URL 如下：

版权所有：腾讯云计算（北京）有限责任公司



```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/format
```

压缩率对比

格式	图片大小
png（原图）	1335.2KB
heif	52.87KB（压缩率96.0%）

版权所有：腾讯云计算（北京）有限责任公司

TPG 压缩

最近更新时间：2023-12-22 10:52:20

功能描述

图片压缩指在图片质量保持不变的情况，尽可能的减小图片大小，以达到节省图片存储空间、减少图片访问流量、提升图片访问速度的效果。
对象存储（Cloud Object Storage，COS）基于 数据万象（Cloud Infinite，CI）产品推出了 TPG 压缩功能，可将图片转换为 tpg 格式，它是腾讯推出的自研图片格式，能支持动图，目前 QQ 浏览器、QQ 空间等产品已默认支持 tpg 图片，在图片质量相同的情况下，图片大小比 gif 格式图片减少90%以上，比 png 图片小50%以上。

限制说明

支持格式：支持将 jpg、png、bmp、gif、webp、heif、avif 等格式图片转换为 tpg 格式。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

前提条件

使用 TPG 压缩，需要先开通存储桶的图片高级压缩功能，您需要在相应的存储桶配置页中通过开关按钮开启功能，详情请参见 设置图片高级压缩。
TPG 是腾讯自研的图片格式，使用前请确认**图片加载环境已支持 TPG 解码**，腾讯云数据万象提供集成 TPG 解码器的 iOS、Android 终端 SDK，可帮助您快速接入和使用 TPG。

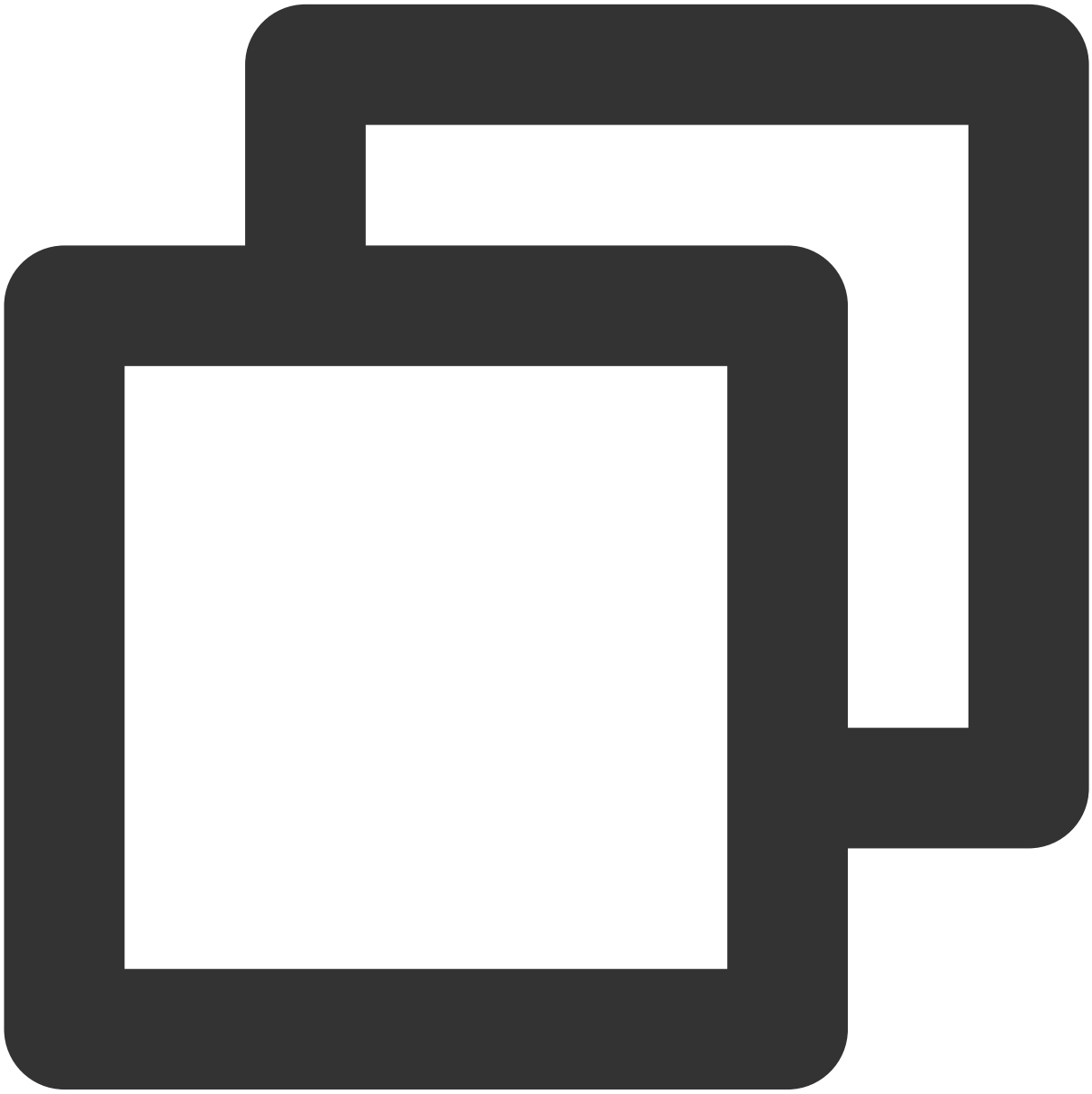
使用方式

COS 通过数据万象 imageMogr2 接口提供 tpg 压缩功能。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理
说明
TPG 压缩为付费服务，计费项为图片高级压缩费用，由数据万象收取，具体费用请参见数据万象 图片处理费用。

版权所有：腾讯云计算（北京）有限责任公司

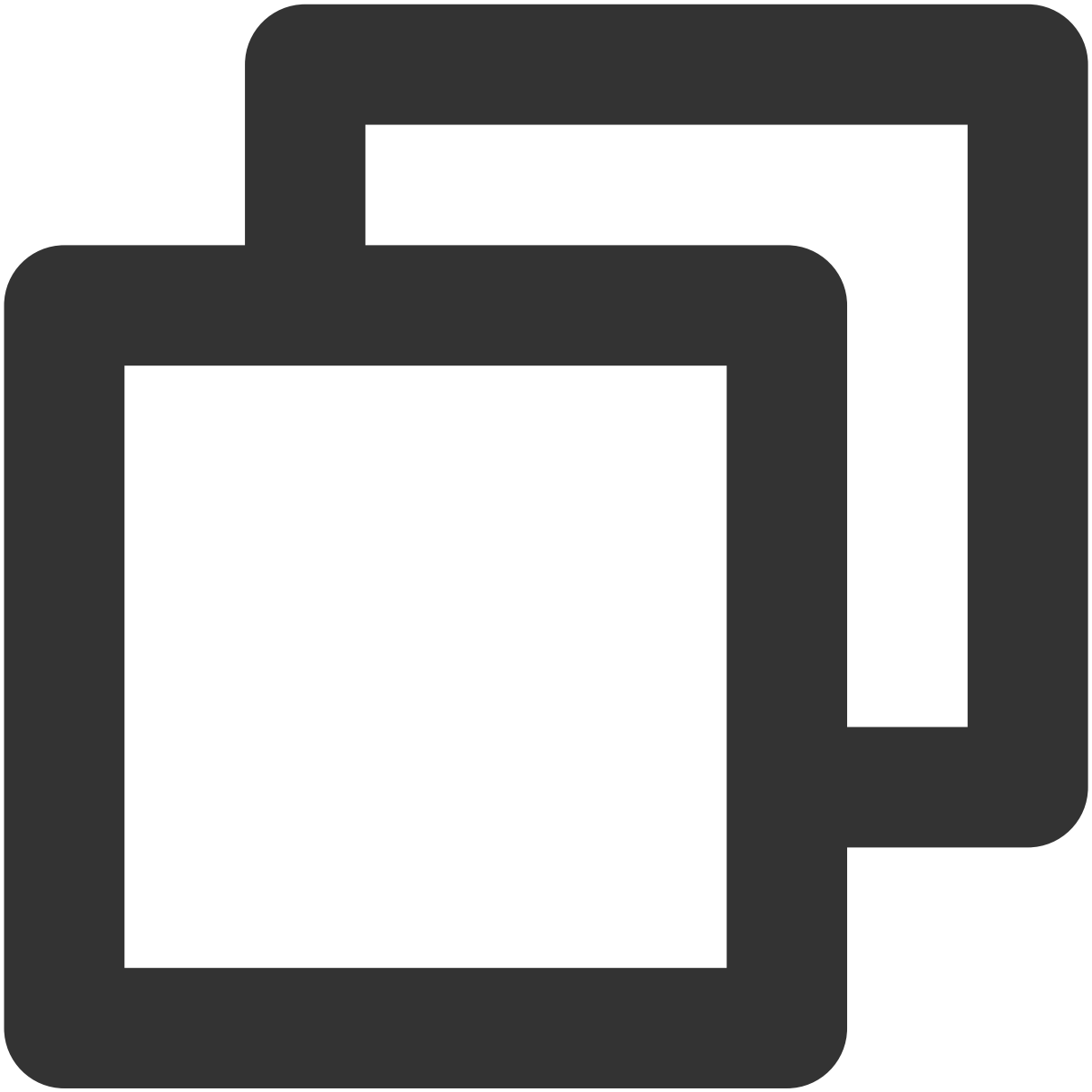
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/format/tpg HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



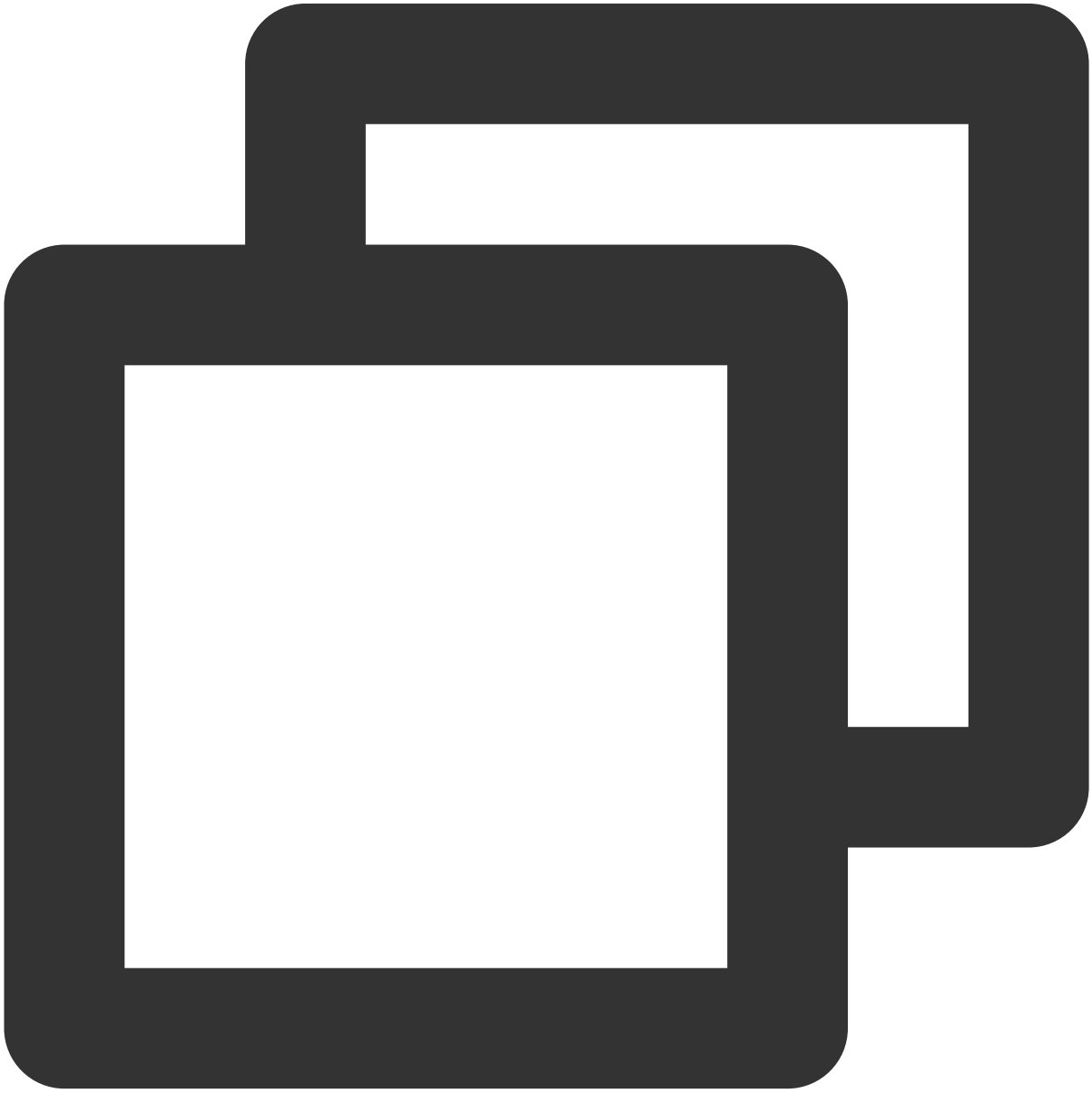
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/tpg"
}]
```

```
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
```

```
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/tpg"
}]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

参数	含义
download_url	文件的访问链接，具体构成为<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>， 例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` 。
/format/<Format>	压缩格式，此处为 tpg。

实际案例

说明

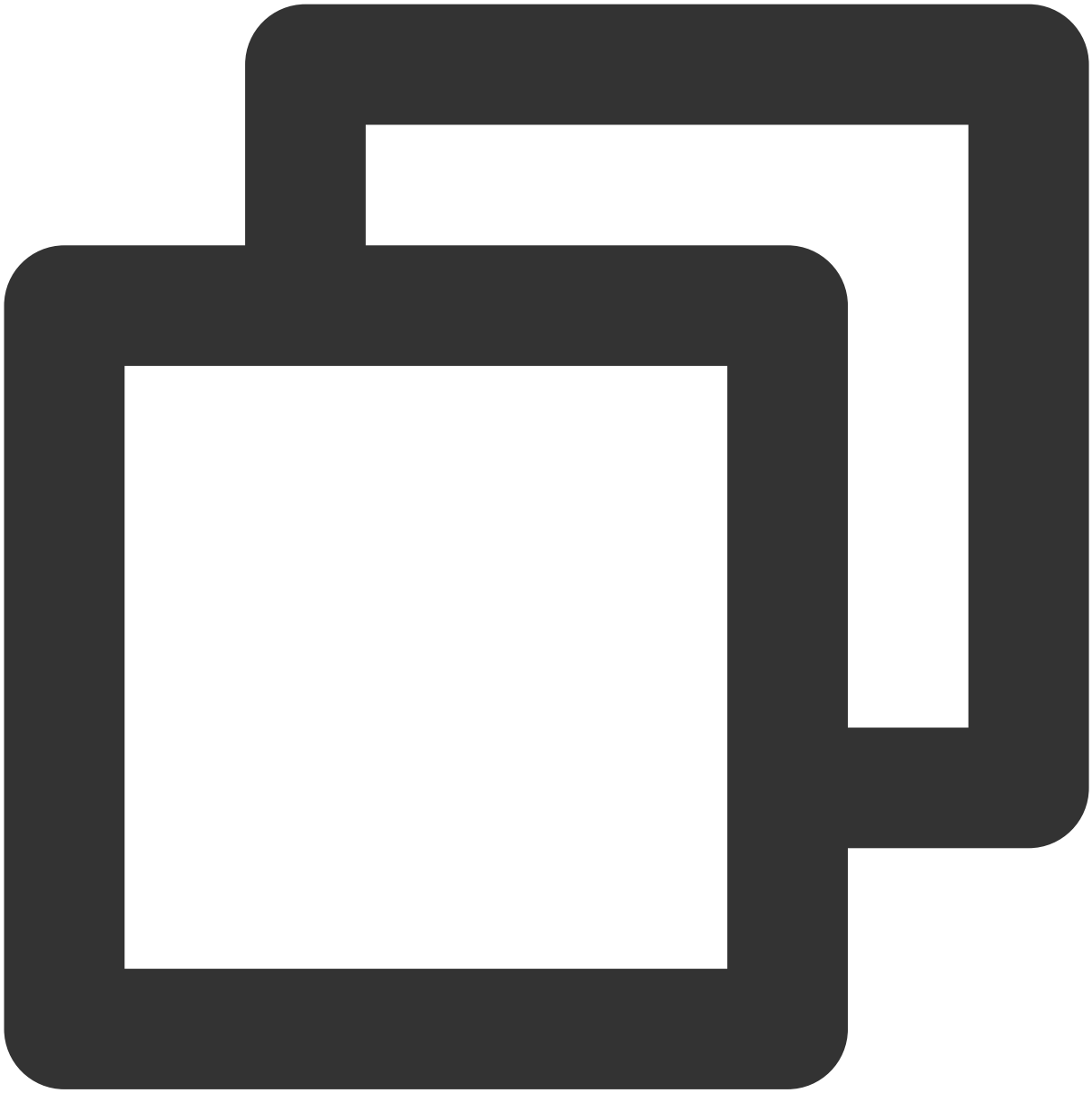
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，请使用**上传时处理**或**云上数据处理**方式。

假设原图格式为 png，图片大小为1335.2KB，如下图所示：



将原图转换为 tpg 格式，请求 URL 如下：

版权所有：腾讯云计算（北京）有限责任公司



```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/format
```

压缩率对比

格式	图片大小
png（原图）	1335.2KB
tpg	36.67KB（压缩率97.3%）

版权所有：腾讯云计算（北京）有限责任公司

AVIF 压缩

最近更新时间：2023-12-22 10:52:26

功能描述

图片压缩指在图片质量保持不变的情况，尽可能的减小图片大小，以达到节省图片存储空间、减少图片访问流量、提升图片访问速度的效果。
对象存储（Cloud Object Storage，COS）基于 数据万象（Cloud Infinite，CI） 产品推出了 AVIF 压缩功能，可将图片转换为 avif 格式，avif 是基于 av1 的一种全新图片格式，在2020年2月由 Netflix 首次公布于众，目前已支持 Chrome、Firefox 等浏览器。

限制说明

支持格式：支持将 jpg、png、bmp、gif、webp、tpg、heif 等格式图片转换为 avif 格式。
体积限制：处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。
动图帧数限制：gif 帧数限300帧。

前提条件

使用 AVIF 压缩，需要先开通存储桶的图片高级压缩功能，您需要在相应的存储桶配置页中通过开关按钮开启功能，详情请参见 设置图片高级压缩。

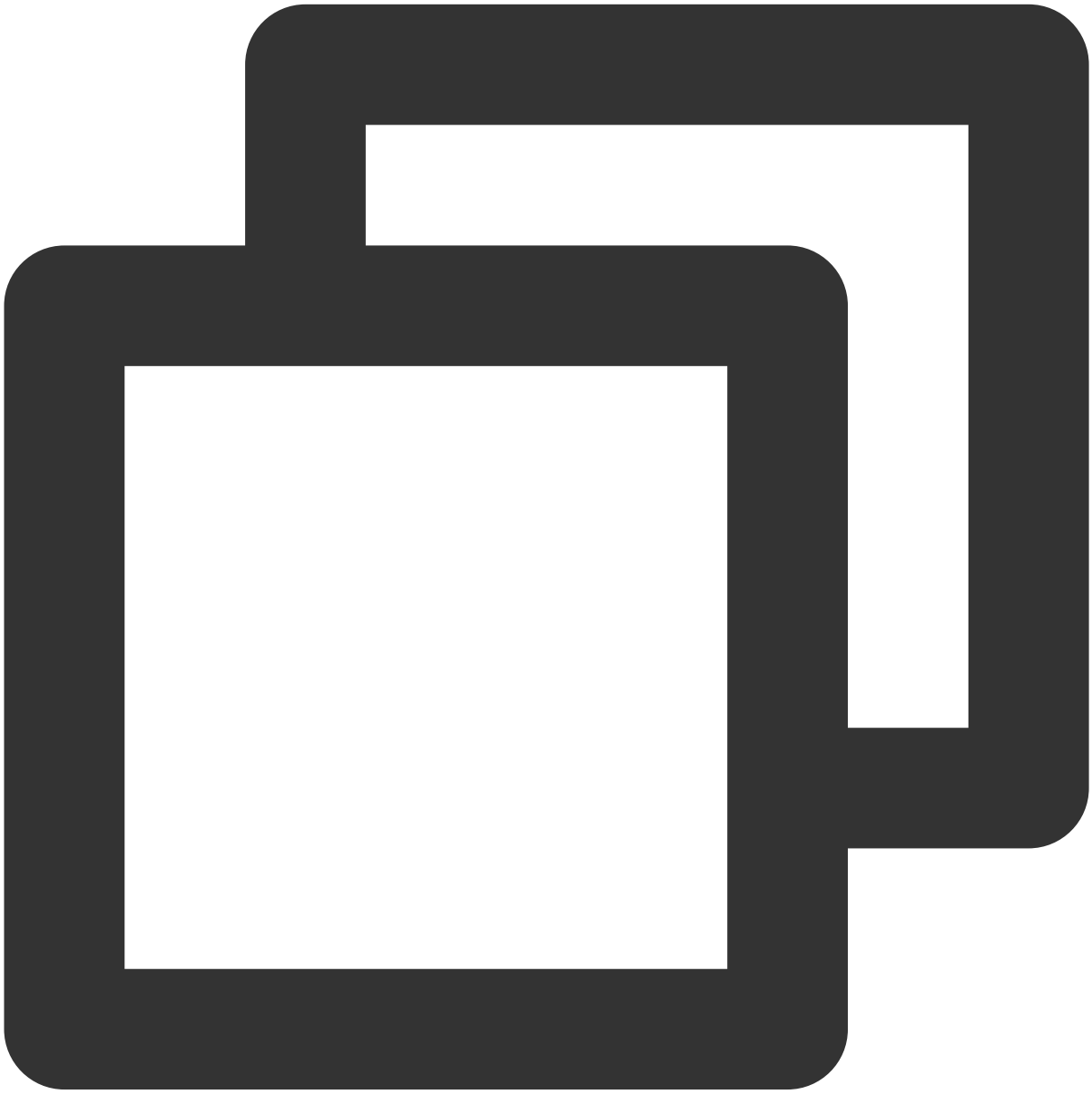
使用方式

COS 通过数据万象 imageMogr2 接口提供 avif 压缩功能。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理
说明
AVIF 压缩为付费服务，计费项为图片高级压缩费用，由数据万象收取，具体费用请参见数据万象 图片处理费用。

版权所有：腾讯云计算（北京）有限责任公司

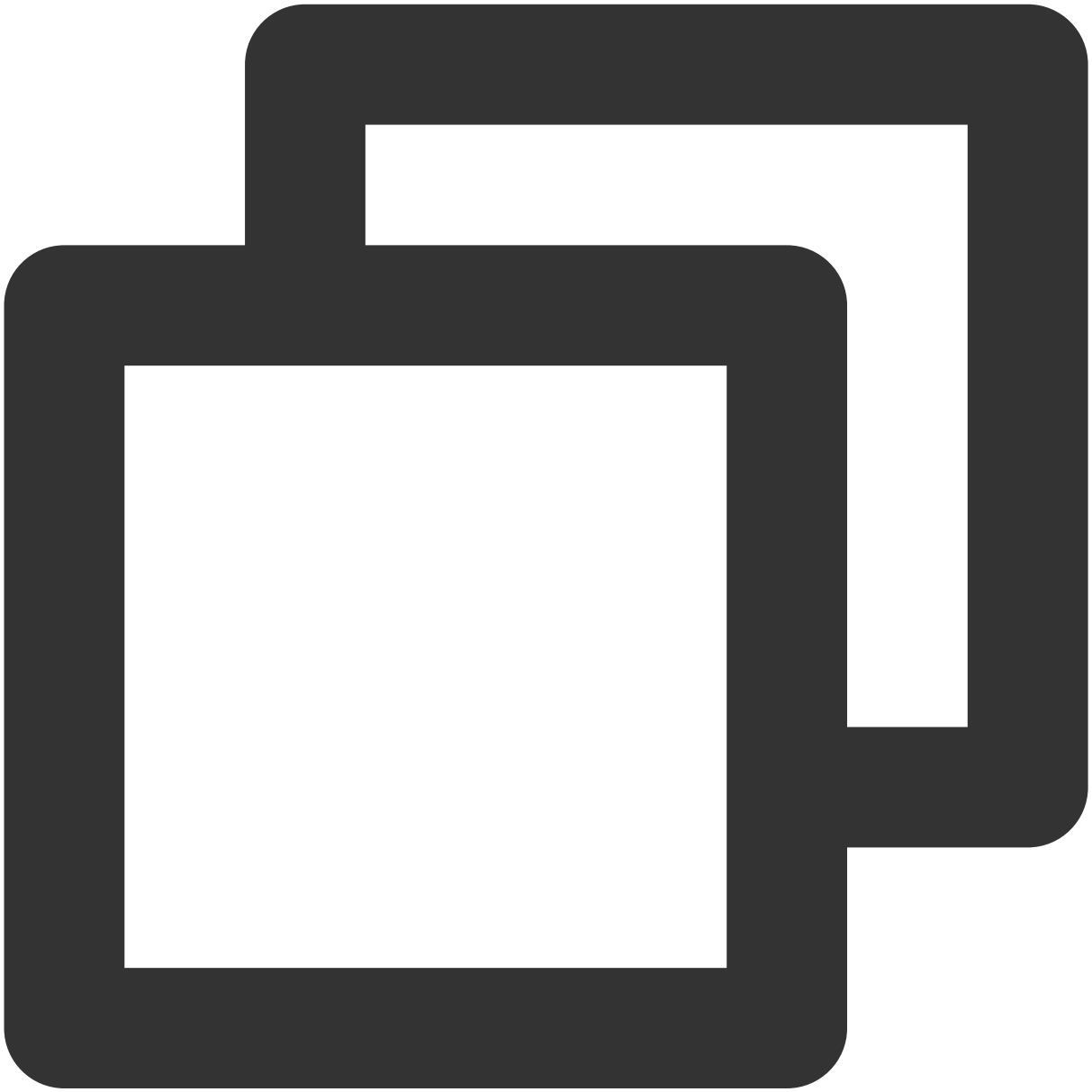
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/format/avif HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



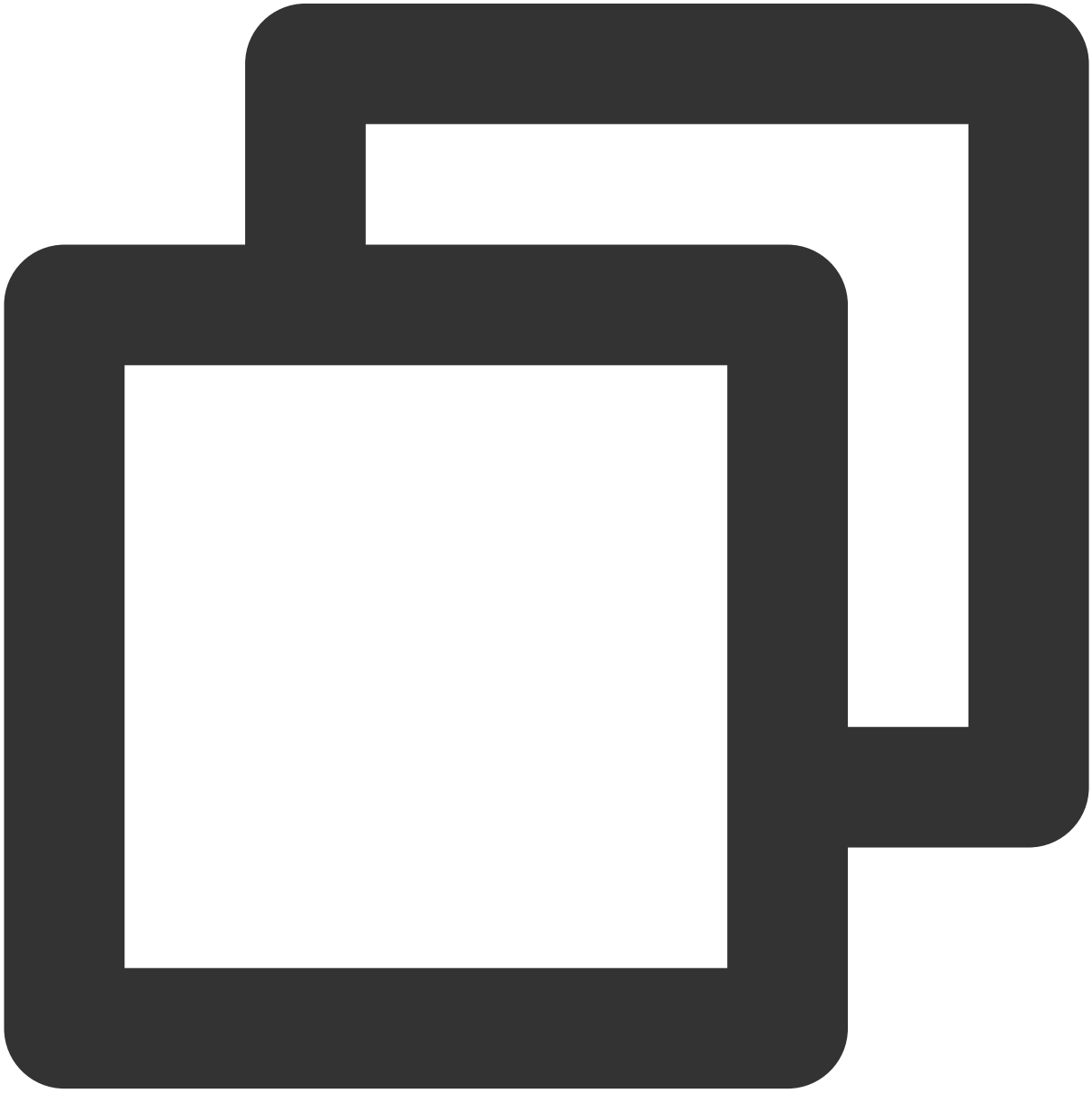
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/avif"
}]
```

```
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
```

```
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/avif"
}]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/format/<Format>	压缩格式，此处为 avif。

实际案例

说明

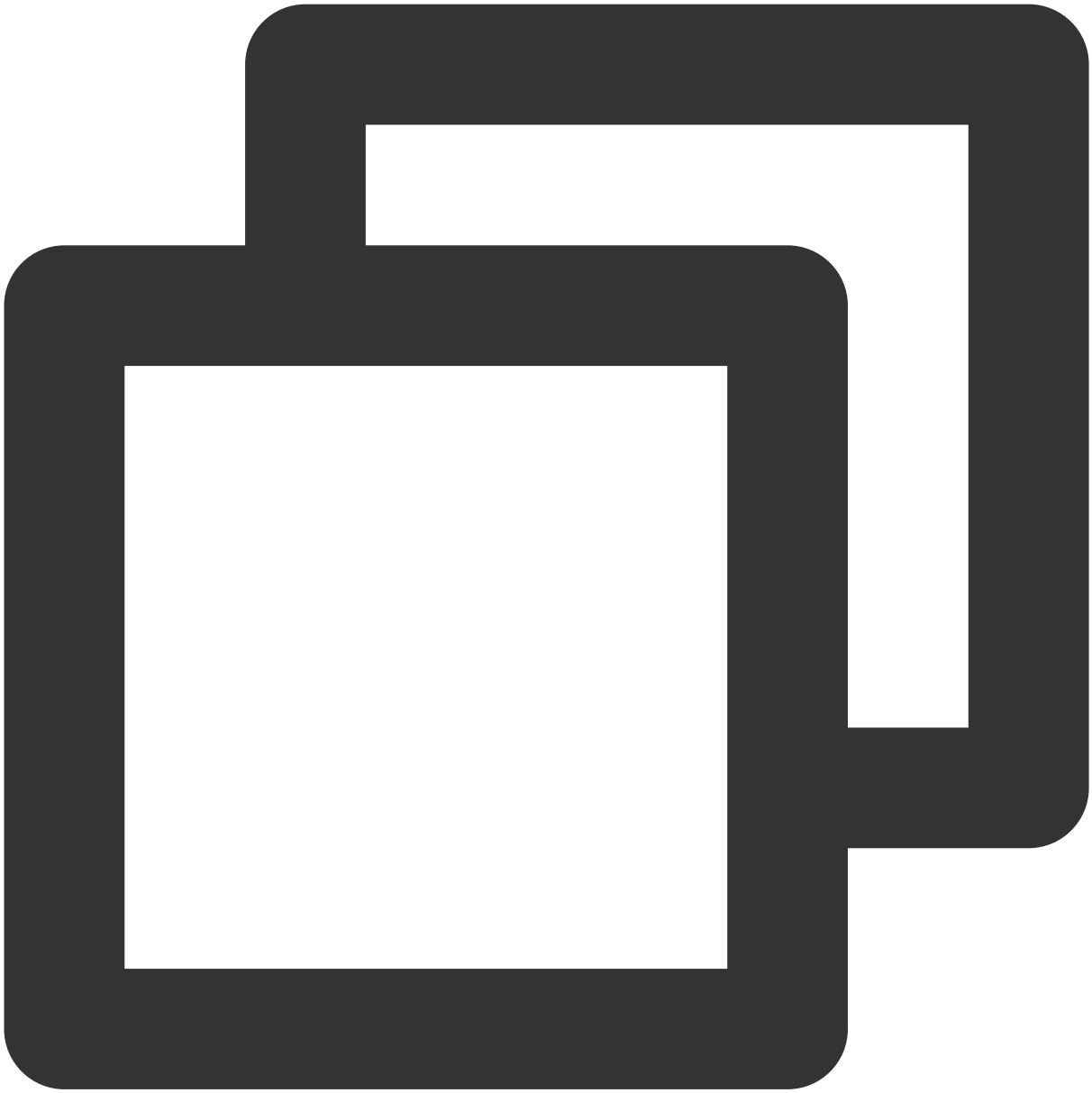
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，请使用**上传时处理**或**云上数据处理**方式。

假设原图格式为 png，图片大小为1335.2KB，如下图所示：



将原图转换为 avif 格式，请求 URL 如下：

版权所有：腾讯云计算（北京）有限责任公司



```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/format
```

压缩率对比

格式	图片大小
png（原图）	1335.2KB
avif	62.8KB（压缩率95.3%）

版权所有：腾讯云计算（北京）有限责任公司

SVG 压缩

最近更新时间：2023-12-22 10:52:57

功能描述

图片压缩指在图片质量保持不变的情况，尽可能的减小图片大小，以达到节省图片存储空间、减少图片访问流量、提升图片访问速度的效果。
对象存储（Cloud Object Storage，COS）基于 数据万象（Cloud Infinite，CI） 产品推出了 SVG 压缩功能，可以在不影响显示效果的情况下，通过一定的优化算法删除 SVG 文件中的冗余信息，最终达到减少SVG文件体积的目的。

限制说明

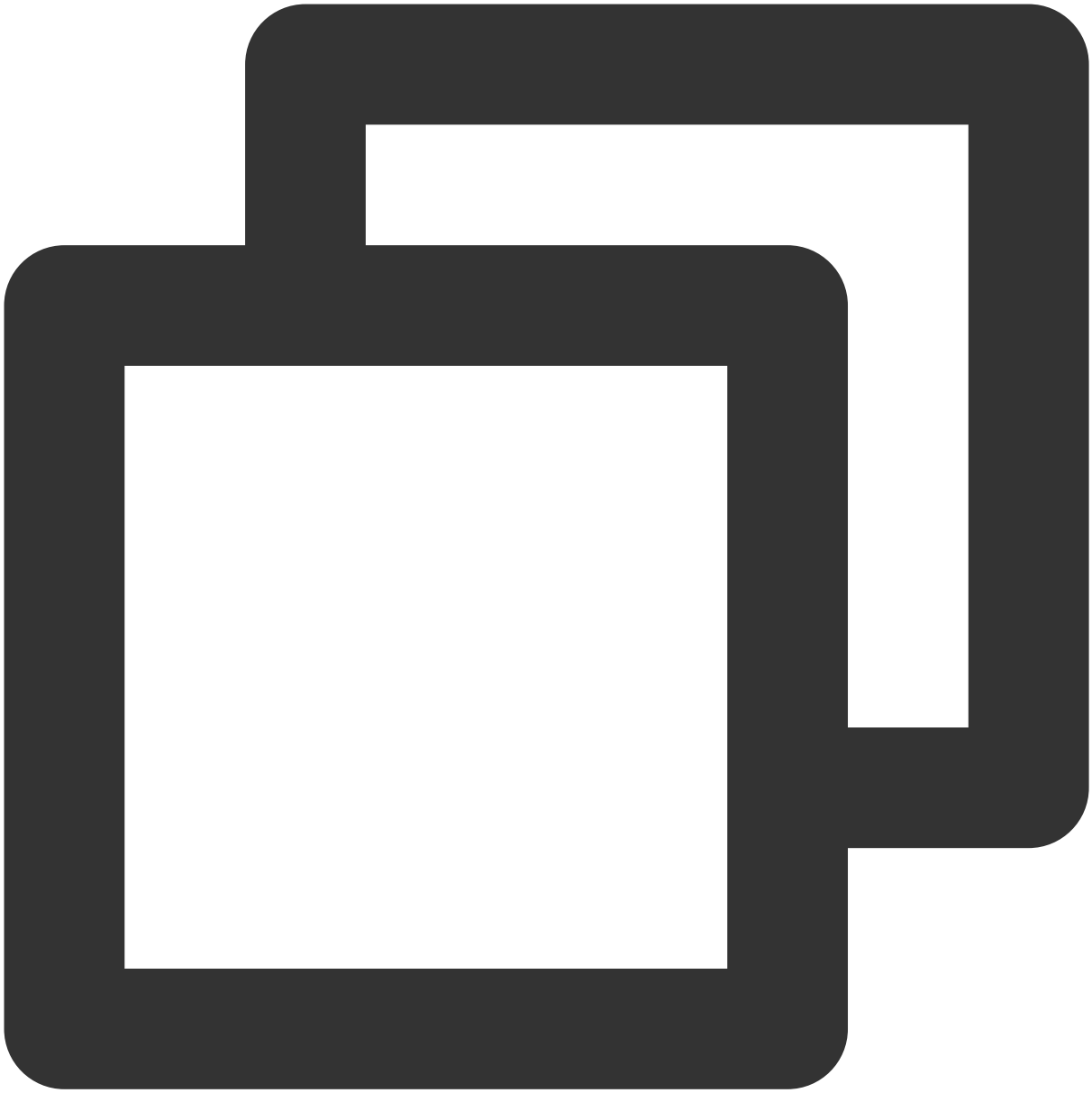
支持格式：仅支持将 svg 格式图片作为输入。
体积限制：处理图片原图大小不超过32MB。

使用方式

对象存储通过数据万象 imageMogr2 接口提供 SVG 压缩功能。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理
说明
SVG 压缩为付费服务，计费项为图片高级压缩费用，由数据万象收取，具体费用请参见数据万象 图片处理费用。

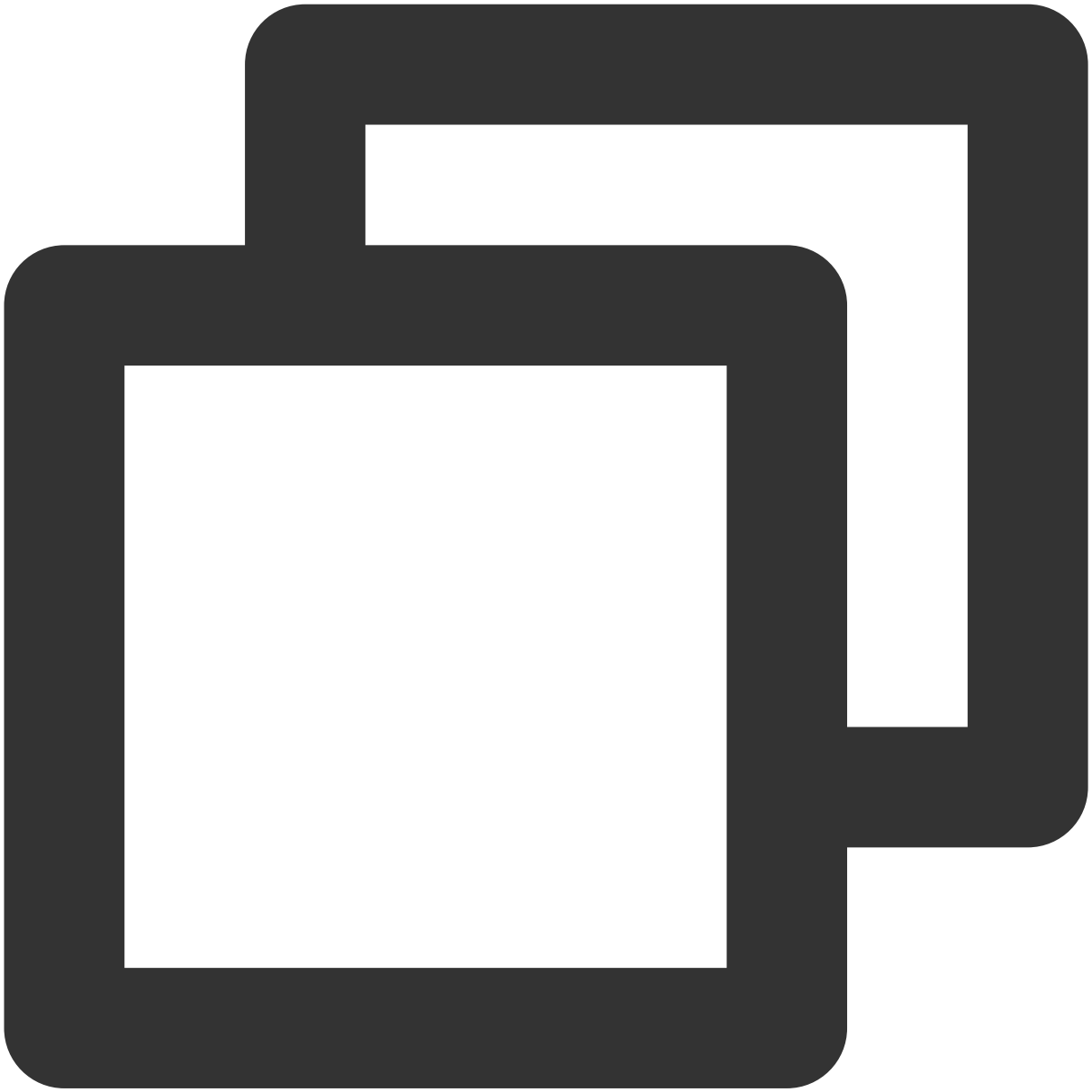
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/format/svgc HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



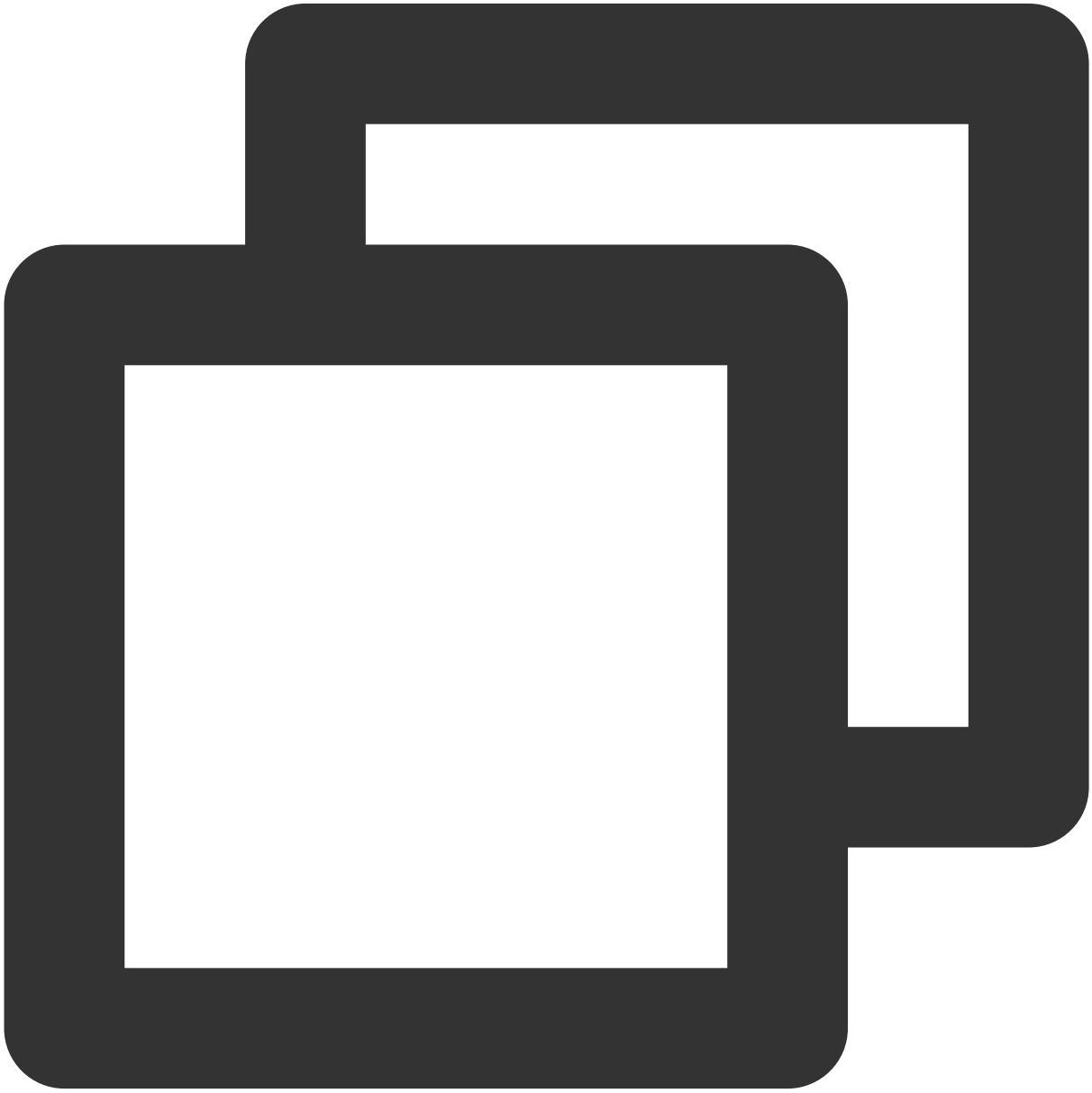
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/svgc"
}]
```

```
}
```

说明

Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片持久化处理。

3. 云上数据处理



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
```

```
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/svgc"
}]
}
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

处理参数说明

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/format/<Format>	压缩格式，此处为 svgc。

盲水印

最近更新时间：2023-12-22 10:53:47

功能描述

盲水印功能基于腾讯云数据万象，是一种全新的水印模式。通过该功能，您可将水印图以不可见的形式添加到原图信息中，并且不会对原图质量产生太大影响。当发现图片被盗取后，您可对疑似被盗取的资源进行盲水印提取，验证图片归属。

通过该 API 接口，您可以：

在上传时为图片添加盲水印

在下载时为图片添加盲水印

对 COS 上的图片添加盲水印

也可以对已添加盲水印的图片进行盲水印提取。

说明

暂不支持为动图（例如 GIF 图片）添加盲水印。

使用盲水印功能，水印图的宽高不得超过原图的1/8。

为保证盲水印效果，水印图请选用黑底白色图片。

文字盲水印当前支持数字[0 - 9]及英文大小写[A - Z,a - z]。

盲水印可抵抗裁剪、涂抹、变色等多种图片盗取攻击，防盗效果与原图大小及攻击程度相关。如需详细咨询请 提交工单。

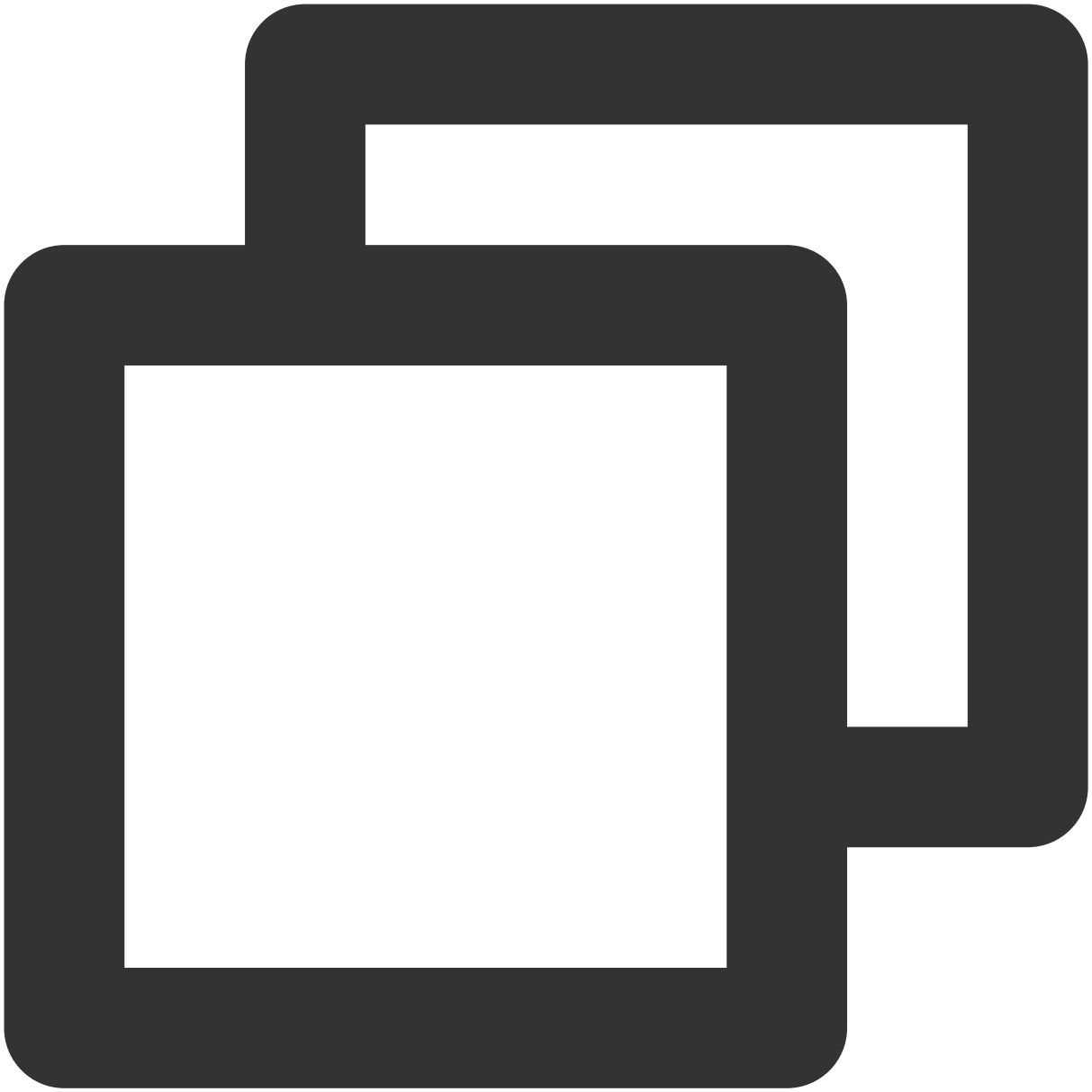
盲水印已支持数据万象所有地域，详情请查看 地域与域名。

添加盲水印

请求一：上传时添加盲水印

图片上传时添加盲水印的请求包与 PUT Object 接口一致，只需在请求包头部增加图片处理参数 Pic-Operations 并使用盲水印参数即可。

请求示例



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>

[Object Content]
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

请求头

此接口使用公共请求头部，详情请参见 公共请求头部 文档。

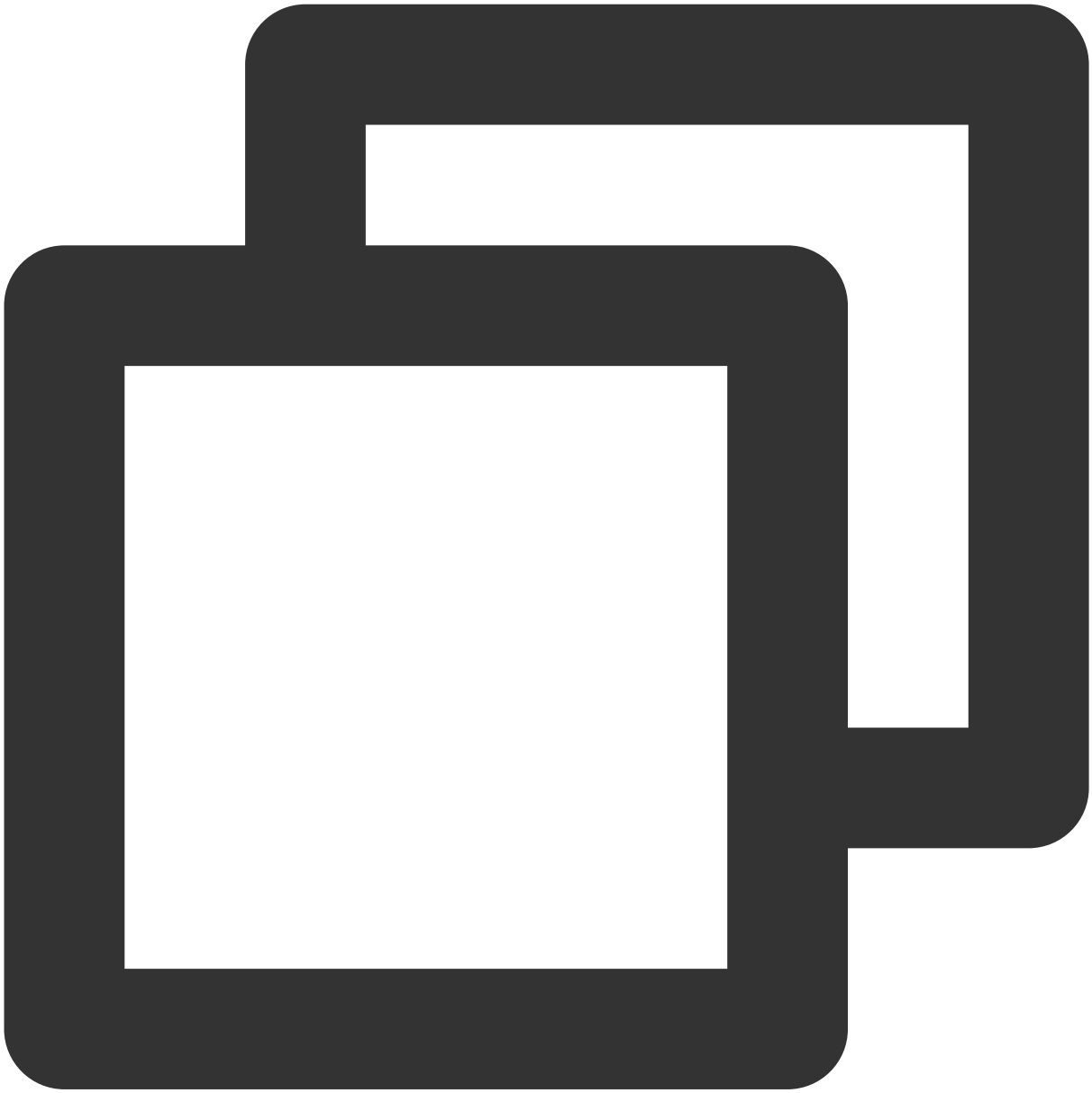
请求包头部增加图片处理参数 Pic-Operations，Pic-Operations 为 json 格式的字符串，具体参数如下：

参数名称	描述	类型	是否必选
is_pic_info	是否返回原图信息。0表示不返回原图信息，1表示返回原图信息，默认为0	Int	否
rules	处理规则，一条规则对应一个处理结果（目前最多支持五条规则），不填则不进行图片处理	Array	否

rules（json 数组）中每一项具体参数如下：

参数名称	描述	类型	是否必选
bucket	存储结果的目标存储桶名称，格式为：BucketName-APPID，如果不指定的话默认保存到当前存储桶	String	否
fileid	处理后文件的保存路径及名称。名称规则说明： 例如源文件的路径及文件名为：/p1/test1.jpg 1. 以/开头为绝对路径，例如 fileid 值为 /p2/test2.jpg 时，表示在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 2. 不以/开头为相对路径，例如 fileid 值为 p2/test2.jpg 时，表示在 p1 文件夹中新建了一个 p2 文件夹，然后在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 3. 注意：请不要以/结尾，否则会产生空文件名	String	是
rule	处理参数，可参见图片处理 API。 若按指定样式处理，则以 `style/` 开头，后加样式名，如样式名为“test”，则 rule 字段为 `style/test`	String	是

使用盲水印需在 rule 中添加水印图参数（watermark），示例格式如下：



```
watermark/3/type/<type>/image/<imageUrl>/text/<text>/level/<level>
```

示例参数说明如下：

参数名称	描述	类型	是否必选
type	盲水印类型，有效值：1为半盲水印；2为全盲水印；3为文字盲水印	Int	是
image	盲水印图片地址，需要经过 URL 安全的 Base64 编码。 当 type 为1或2时必填，type 为3时无效。 指定的水印图片必须同时满足如下条件： 1. 盲水印图片与原图片必须位于同一个存储桶下	String	否

版权所有：腾讯云计算（北京）有限责任公司

	2. URL 必须以http://开始，不能省略 http 头，也不能填 https 头，例如以下地址为非法水印地址：examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png		
text	盲水印文字，需要经过 URL 安全的 Base64 编码。当 type 为3时必填，type 为1或2时无效	String	否
level	只对全盲水印（type=2）有效。level 的取值范围为{1,2,3}，默认值为1，level 值越大则图片受影响程度越大、盲水印效果越好。	Int	否

请求参数

此接口无请求参数。

请求体

此接口请求的请求体为需要添加盲水印的原文件（图片）内容。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

响应体具体数据内容如下：

参数名称	描述	类型
UploadResult	原图信息	Container

UploadResult 节点内容：

参数名称	描述	类型
OriginalInfo	原图信息	Container
ProcessResults	图片处理结果	Container

OriginalInfo 节点内容：

节点名称	描述	类型
Key	原图文件名	String
Location	图片路径	String

版权所有：腾讯云计算（北京）有限责任公司

ImageInfo	原图图片信息	Container

ImageInfo 节点内容：

节点名称	描述	类型
Format	格式	String
Width	图片宽度	Int
Height	图片高度	Int
Quality	图片质量	Int
Ave	图片主色调	String
Orientation	图片旋转角度	Int

ProcessResults 节点内容：

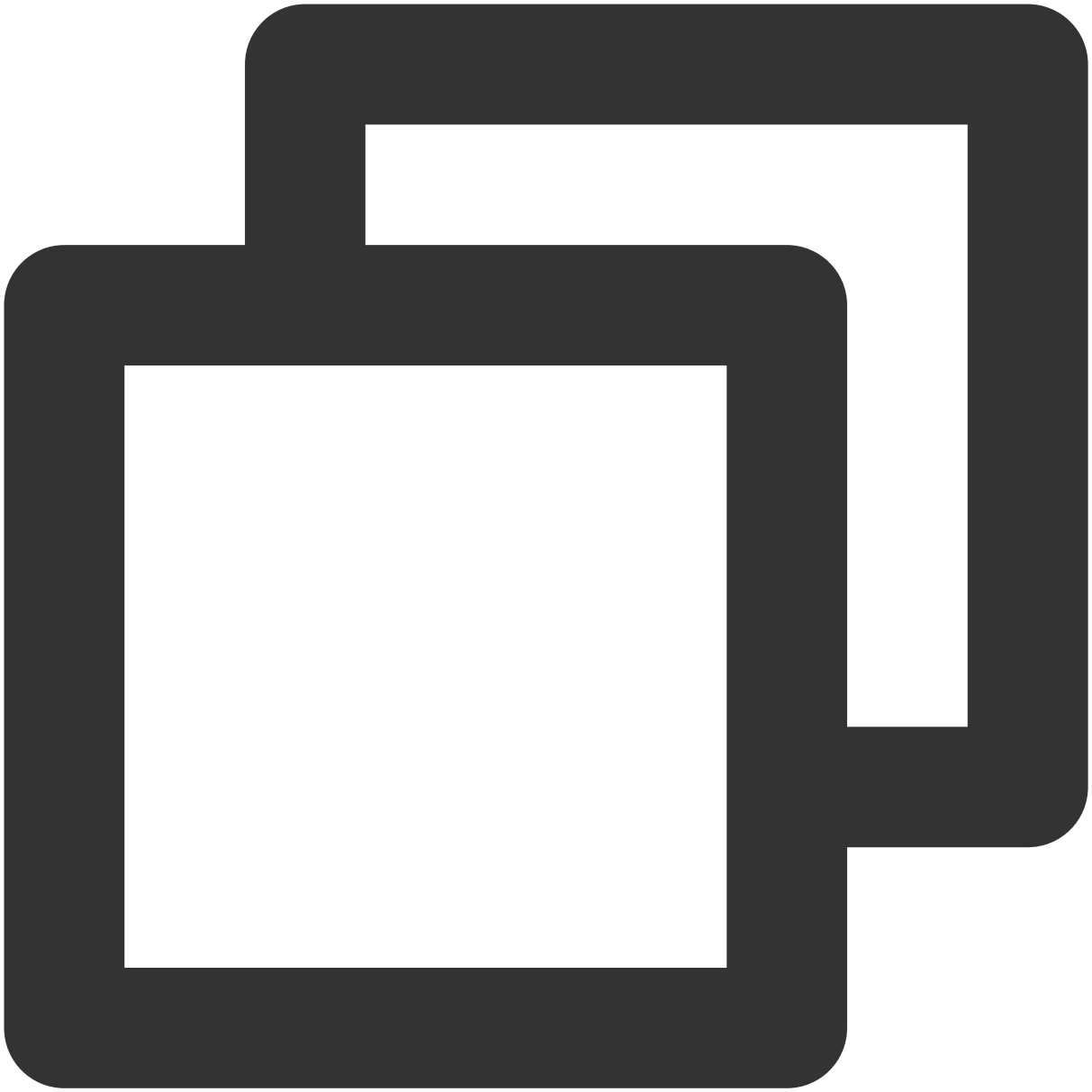
节点名称	描述	类型
Object	每一个图片处理结果	Container

Object 节点内容：

节点名称	描述	类型
Key	文件名	String
Location	图片路径	String
Format	图片格式	String
Width	图片宽度	Int
Height	图片高度	Int
Size	图片大小	Int
Quality	图片质量	Int

实际案例

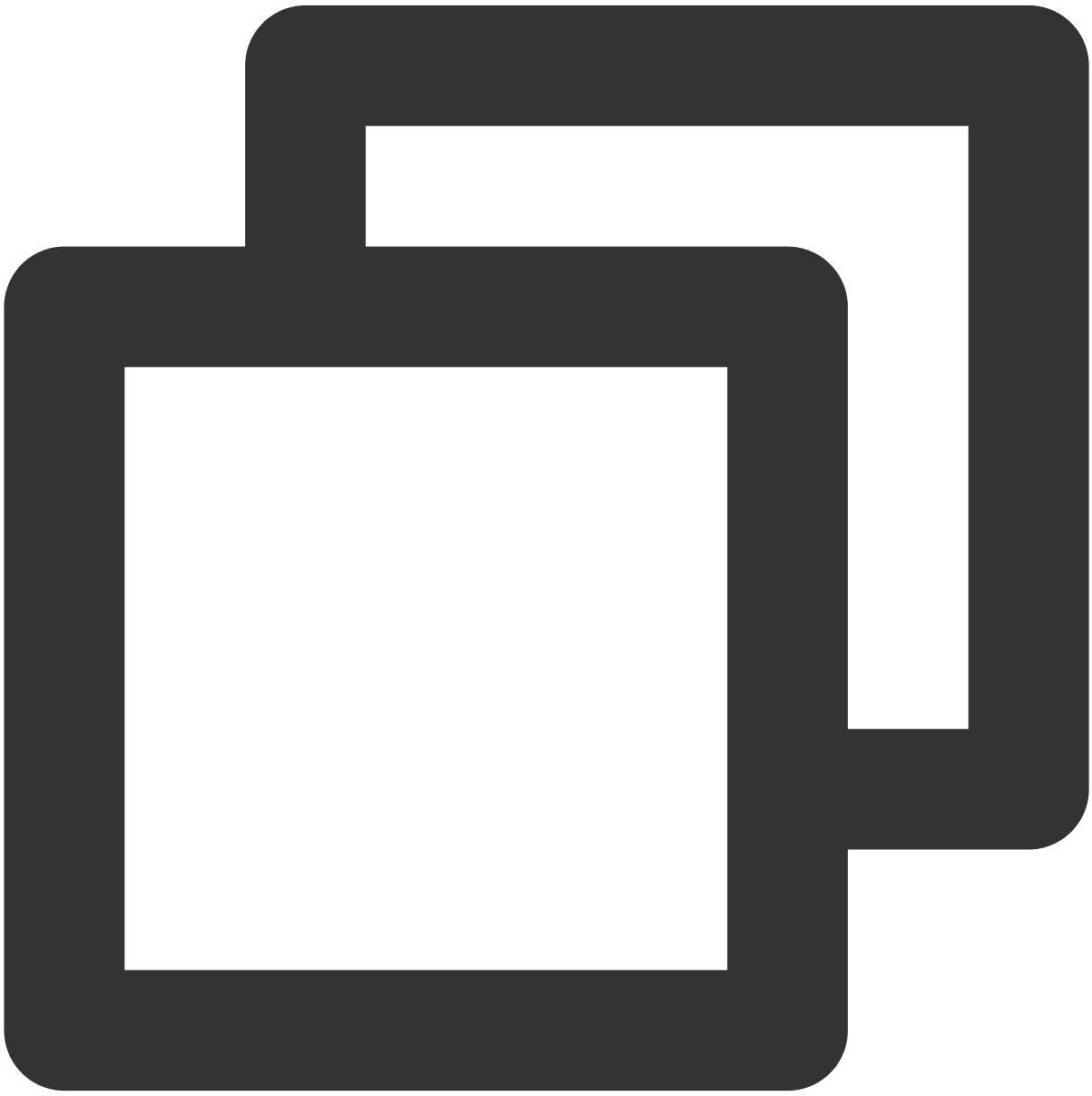
请求



```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2018 09:06:15 GMT
Authorization:XXXXXXXXXXXX

Pic-Operations:
{
    "is_pic_info": 1,
    "rules": [{
        "fileid": "exampleobject",
        "rule": "watermark/3/type/1/image/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQucGljc
```

```
    }]
}
Content-Length: 64

[Object Content]
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
```

```
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
          <OriginalInfo>
              <Key>exampleobject</Key>
              <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/exampl
              <ImageInfo>
                   <Format>JPEG</Format>
                   <Width>640</Width>
                   <Height>427</Height>
                   <Quality>100</Quality>
                   <Ave>0xa18454</Ave>
                   <Orientation>1</Orientation>
              </ImageInfo>
          </OriginalInfo>
          <ProcessResults>
              <Object>
                   <Key>exampleobject</Key>
                   <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/e
                   <Format>png</Format>
                   <Width>640</Width>
                   <Height>427</Height>
                   <Size>463092</Size>
                   <Quality>100</Quality>
              </Object>
          </ProcessResults>
</UploadResult>
```

注意

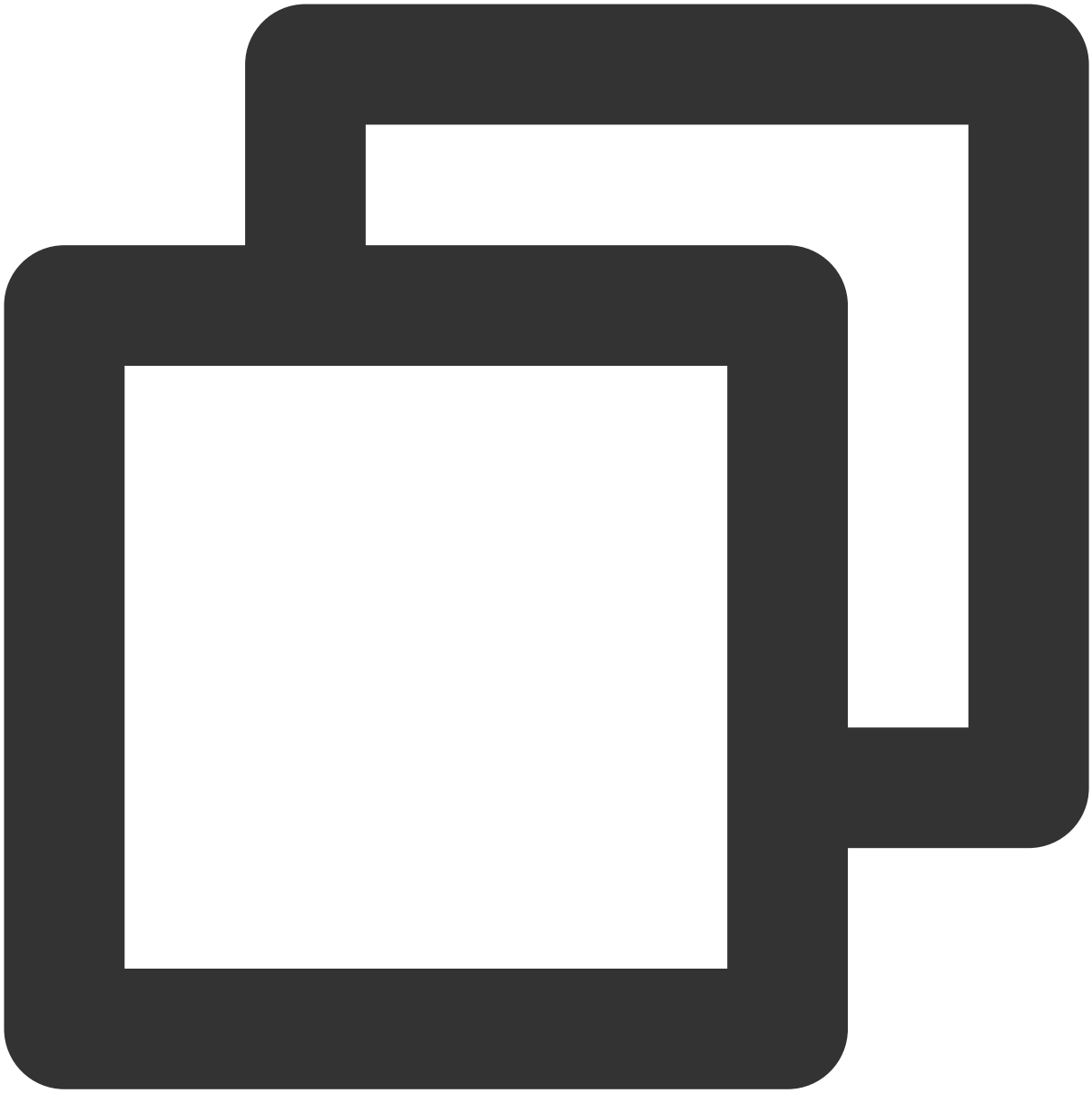
请求体的 [Object Content] 字段为原图的图片内容(二进制字符串)，请求头 Pic-Operations 的 rule 字段中，imageUrl 为水印图片的地址。

响应体的 ProcessResults 中的 Location 字段为处理后已经携带盲水印的图片的地址。

请求二：下载时添加盲水印

图片下载时添加盲水印与添加普通水印操作相同，只需在图片访问链接后使用 watermark 参数即可。

请求示例



```
GET /<ObjectKey>?watermark/3/type/<type>/image/<imageUrl>/text/<text> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

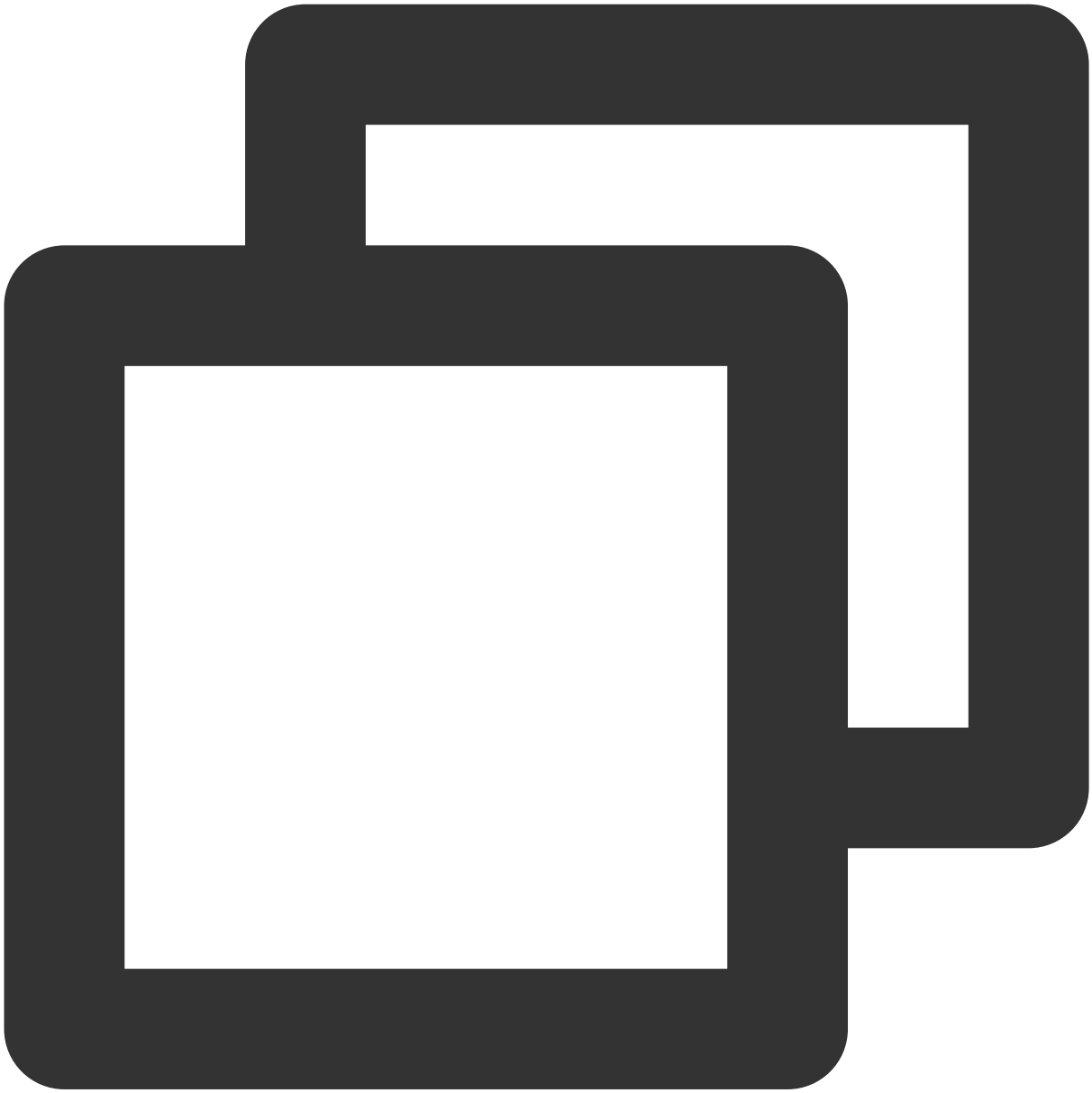
请求参数

请求参数说明如下：

参数	描述	类型	是否必选

download_url	文件的访问链接，具体构成为 http(s):// `<BucketName-APPID>.cos.<Region>.myqcloud.com/<picture name>`，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`	String	是
type	盲水印类型，有效值：1为半盲水印；2为全盲水印；3为文字盲水印	Int	是
image	盲水印图片地址，需要经过 URL 安全的 Base64 编码。 当 type 为1或2时必填，type 为3时无效。 指定的水印图片必须同时满足如下条件： 1. 盲水印图片与原图片必须位于同一个存储桶下 2. URL 必须以http://开始，不能省略 http 头，也不能填 https 头，例如以下地址为非法水印地址： examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png	String	否
text	盲水印文字，需要经过 URL 安全的 Base64 编码。当 type 为3时必填，type 为1或2时无效	String	否
level	只对全盲水印（type=2）有效。level 的取值范围为{1,2,3}，默认值为1，level 值越大则图片受影响程度越大、盲水印效果越好	Int	否

实际案例

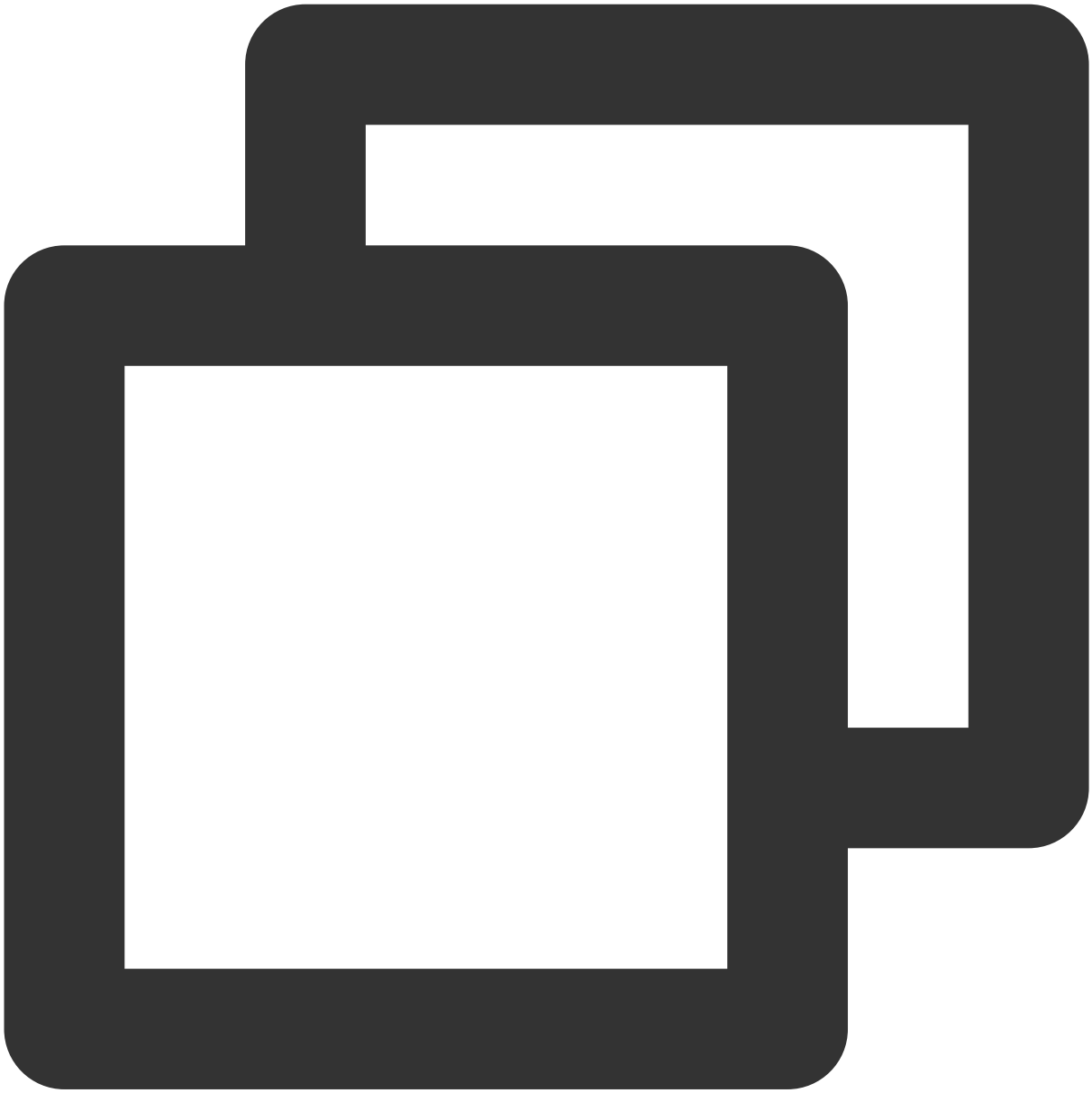


```
https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark
```

请求三：对 COS 上的图片添加盲水印

通过 API 能够对已存储在 COS 的图片进行添加盲水印的操作，并将结果存入到 COS。

请求示例



```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口使用公共请求头部，详情请参见 公共请求头部 文档。

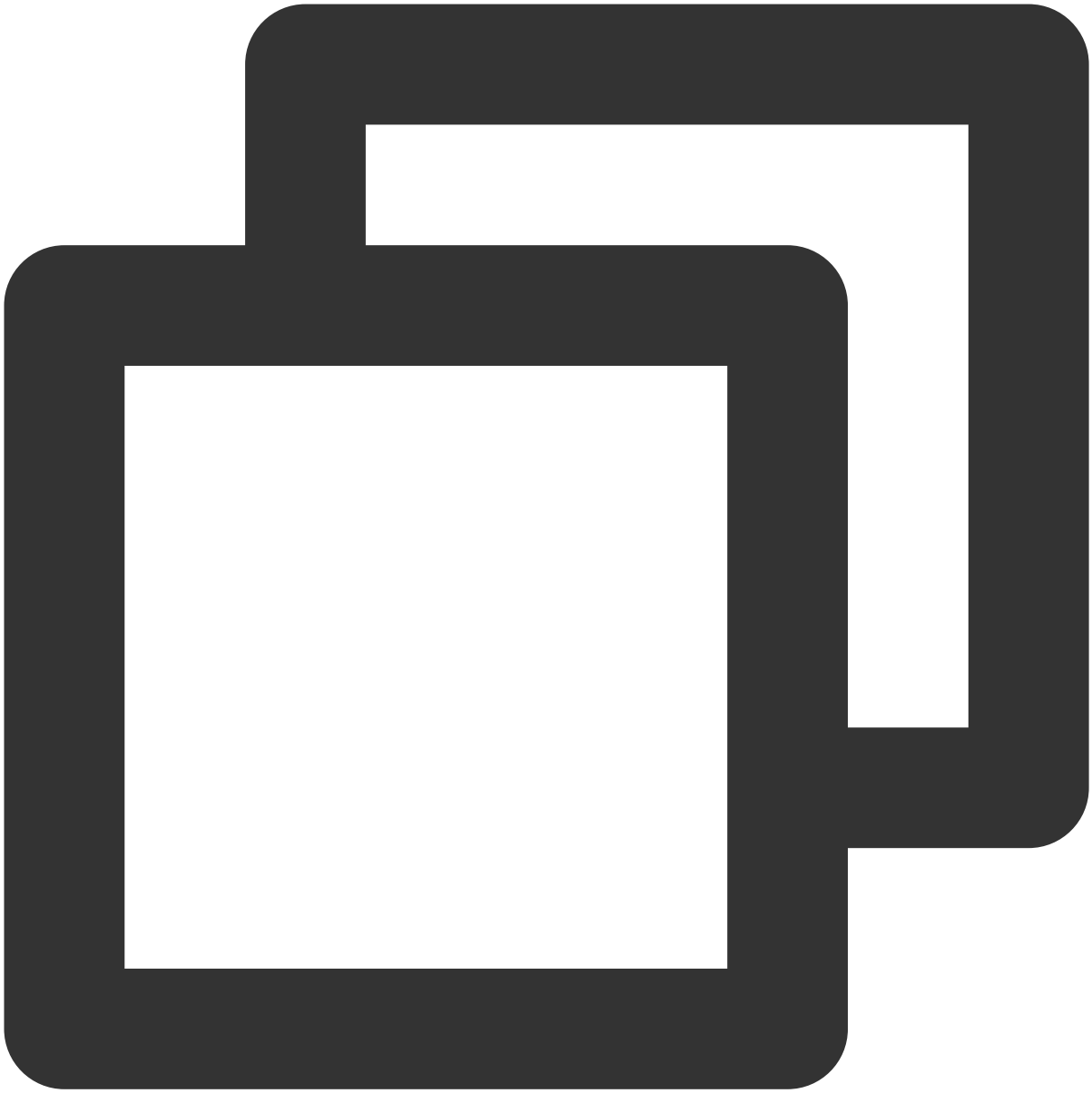
请求包头部增加图片处理参数 Pic-Operations，Pic-Operations 为 json 格式的字符串，具体参数如下：

参数名称	描述	类型	是否必选
is_pic_info	是否返回原图信息。0表示不返回原图信息，1表示返回原图信息，默认为0	Int	否
rules	处理规则，一条规则对应一个处理结果（目前最多支持五条规则），不填则不进行图片处理	Array	否

rules（json 数组）中每一项具体参数如下：

参数名称	描述	类型	是否必选
bucket	存储结果的目标存储桶名称，格式为：BucketName-APPID，如果不指定的话默认保存到当前存储桶	String	否
fileid	处理后文件的保存路径及名称。名称规则说明：例如源文件的路径及文件名为：/p1/test1.jpg 1. 以/开头为绝对路径，例如 fileid 值为/p2/test2.jpg时，表示在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 2. 不以/开头为相对路径，例如 fileid 值为 p2/test2.jpg 时，表示在 p1 文件夹中新建了一个 p2 文件夹，然后在p2文件夹中存储了一个文件名为 test2.jpg 的文件 3. 注意：请不要以/结尾，否则会产生空文件名	String	是
rule	处理参数，可参见图片处理 API。 若按指定样式处理，则以 `style/` 开头，后加样式名，如样式名为“test”，则 rule 字段为 `style/test`	String	是

使用盲水印需在 rule 中添加水印图参数（watermark），示例格式如下：



```
watermark/3/type/<type>/image/<imageUrl>/text/<text>/level/<level>
```

示例参数说明如下：

参数	描述	类型	是否必选
type	盲水印类型，有效值：1为半盲水印；2为全盲水印；3为文字盲水印	Int	是
image	盲水印图片地址，需要经过 URL 安全的 Base64 编码。当 type 为1或2时必填，type 为3时无效。指定的水印图片必须同时满足如下条件：	String	否

版权所有：腾讯云计算（北京）有限责任公司

	1. 盲水印图片与原图片必须位于同一个存储桶下 2. URL 必须以http://开始，不能省略 http 头，也不能填 https 头，例如以下地址为非法水印地址： examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png		
text	盲水印文字，需要经过 URL 安全的 Base64 编码。当 type 为3时必填，type 为1或2时无效。	String	否
level	只对全盲水印（type=2）有效。level 的取值范围为{1,2,3}，默认值为1，level 值越大则图片受影响程度越大、盲水印效果越好	Int	否

请求参数

此接口无请求参数。

请求体

此接口无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

响应体具体数据内容如下：

参数名称	描述	类型
UploadResult	原图信息	Container

UploadResult 节点内容：

参数名称	描述	类型
OriginalInfo	原图信息	Container
ProcessResults	图片处理结果	Container

OriginalInfo 节点内容：

节点名称	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

Key	原图文件名	String
Location	图片路径	String
ImageInfo	原图图片信息	Container

ImageInfo 节点内容：

节点名称	描述	类型
Format	格式	String
Width	图片宽度	Int
Height	图片高度	Int
Quality	图片质量	Int
Ave	图片主色调	String
Orientation	图片旋转角度	Int

ProcessResults 节点内容：

节点名称	描述	类型
Object	每一个图片处理结果	Container

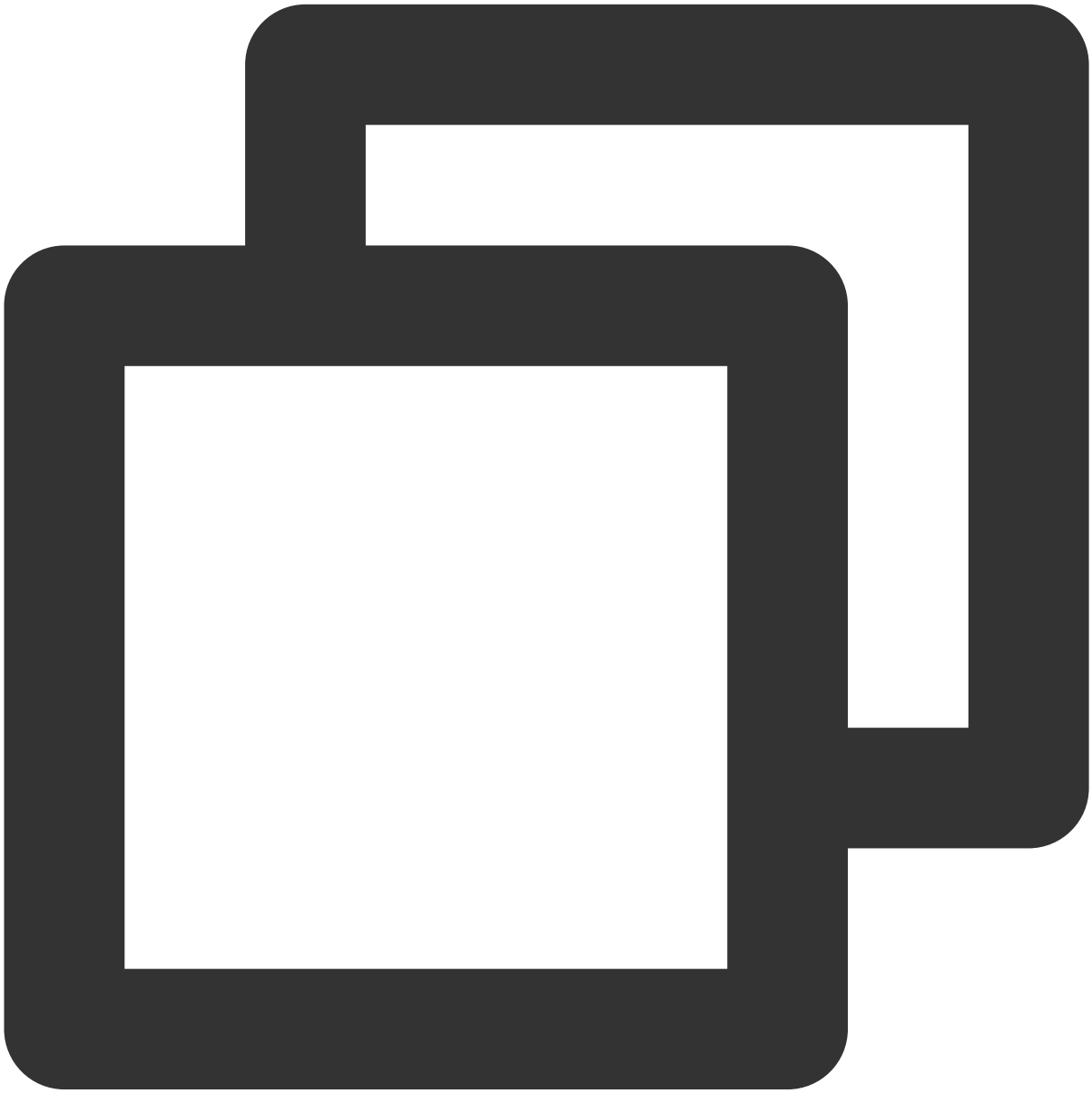
Object 节点内容：

节点名称	描述	类型
Key	文件名	String
Location	图片路径	String
Format	图片格式	String
Width	图片宽度	Int
Height	图片高度	Int
Size	图片大小	Int
Quality	图片质量	Int

实际案例

版权所有：腾讯云计算（北京）有限责任公司

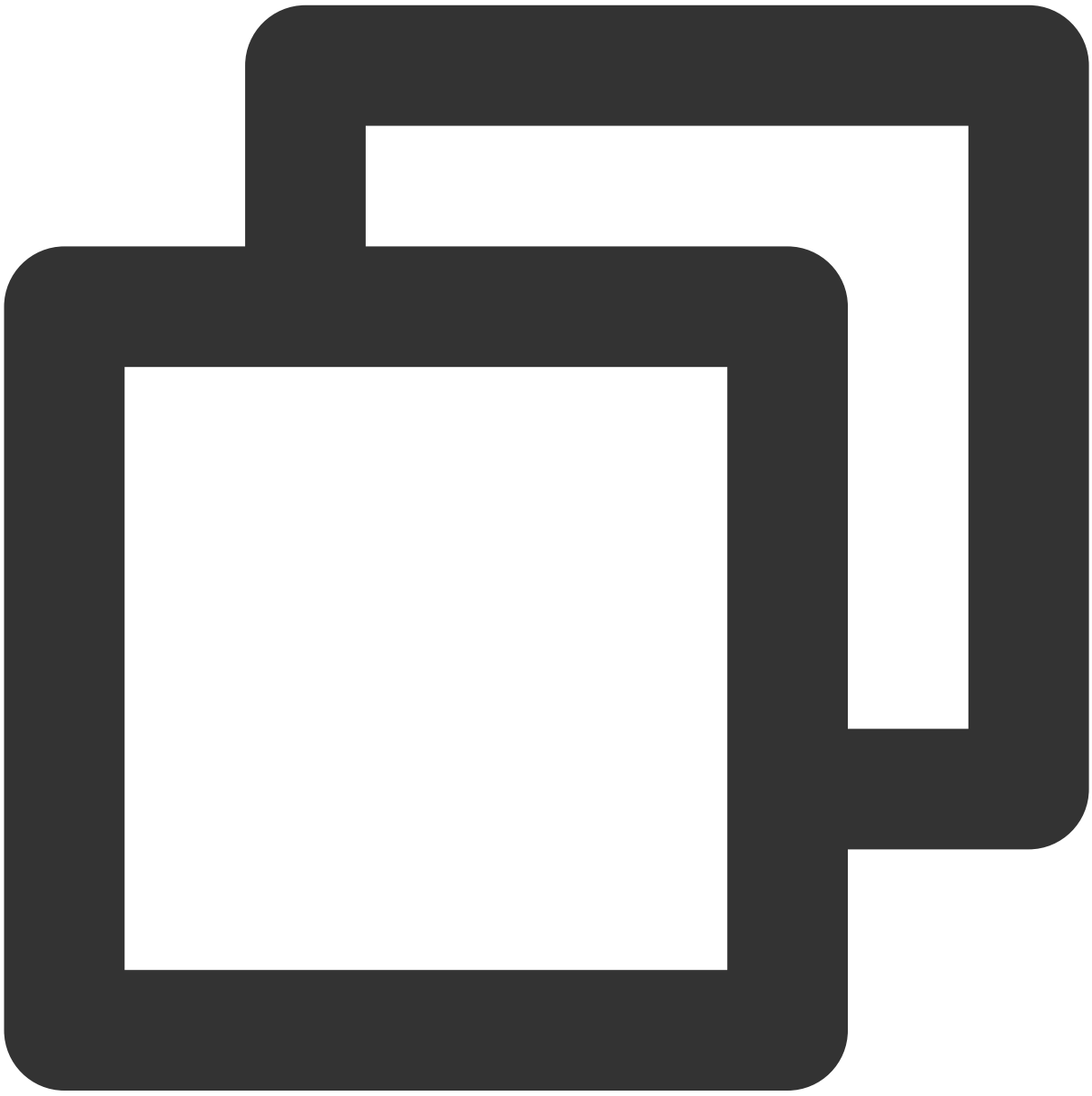
请求



```
POST /exampleobject?image_process HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2018 09:06:15 GMT
Authorization:XXXXXXXXXXXX

Pic-Operations:
{
    "is_pic_info": 1,
    "rules": [{
```

```
        "fileid": "exampleobject",
        "rule": "watermark/3/type/1/image/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQucGljc
    }]
}
Content-Length: 64
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
```

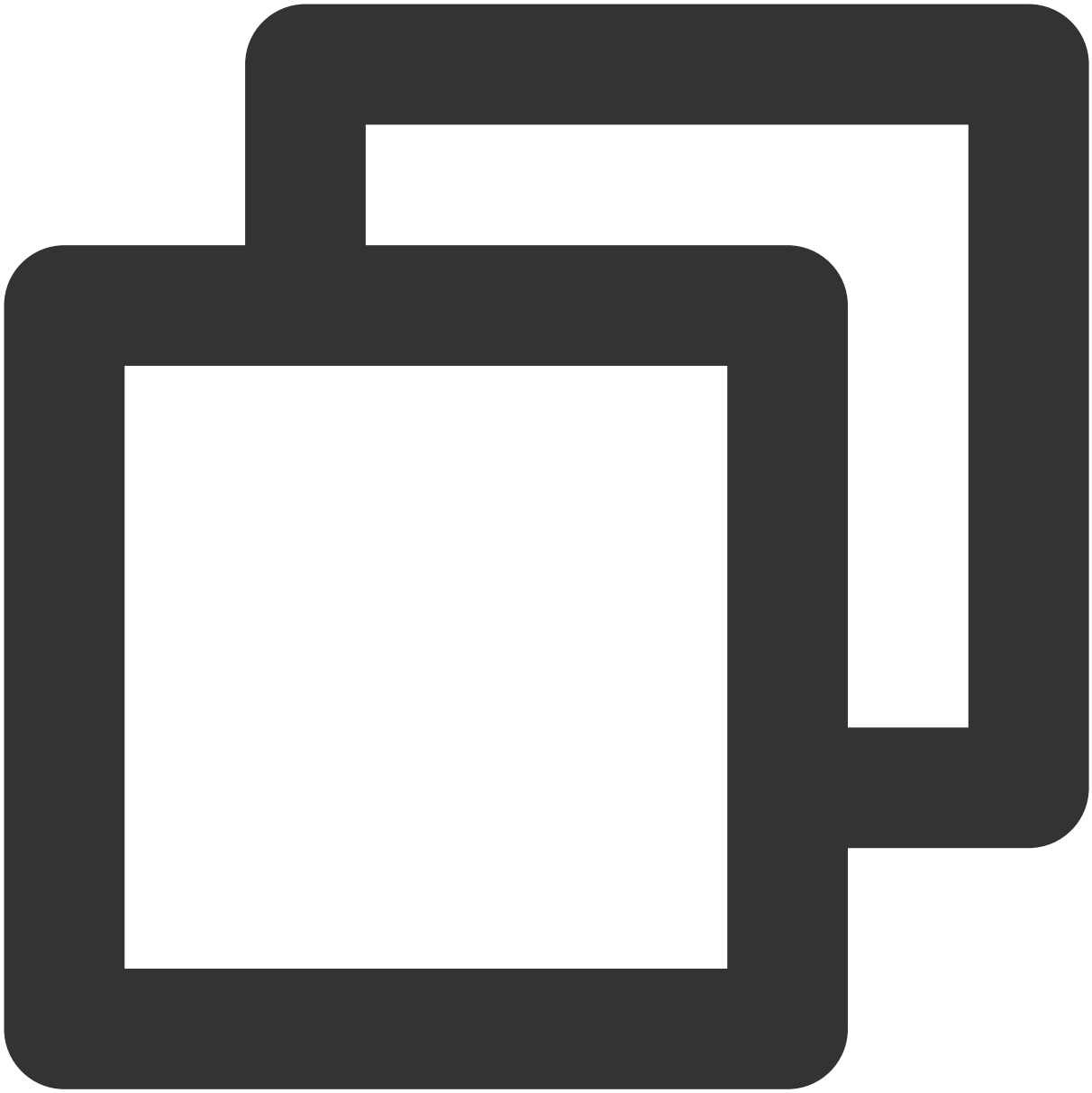
```
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
          <OriginalInfo>
              <Key>exampleobject</Key>
              <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/exampl
              <ImageInfo>
                   <Format>JPEG</Format>
                   <Width>640</Width>
                   <Height>427</Height>
                   <Quality>100</Quality>
                   <Ave>0xa18454</Ave>
                   <Orientation>1</Orientation>
              </ImageInfo>
          </OriginalInfo>
          <ProcessResults>
              <Object>
                   <Key>exampleobject</Key>
                   <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/e
                   <Format>png</Format>
                   <Width>640</Width>
                   <Height>427</Height>
                   <Size>463092</Size>
                   <Quality>100</Quality>
              </Object>
          </ProcessResults>
</UploadResult>
```

提取盲水印

请求一：上传时提取盲水印

请求示例

上传图片并提取盲水印的请求包与 PUT Object 接口一致，只需在请求包头部增加图片处理参数 Pic-Operations 并使用提取盲水印参数（watermark/4）即可。



```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization:XXXXXXXXXXXX
Pic-Operations: <PicOperations>

[Object Content]
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

请求头

此接口使用公共请求头部，详情请参见 公共请求头部 文档。

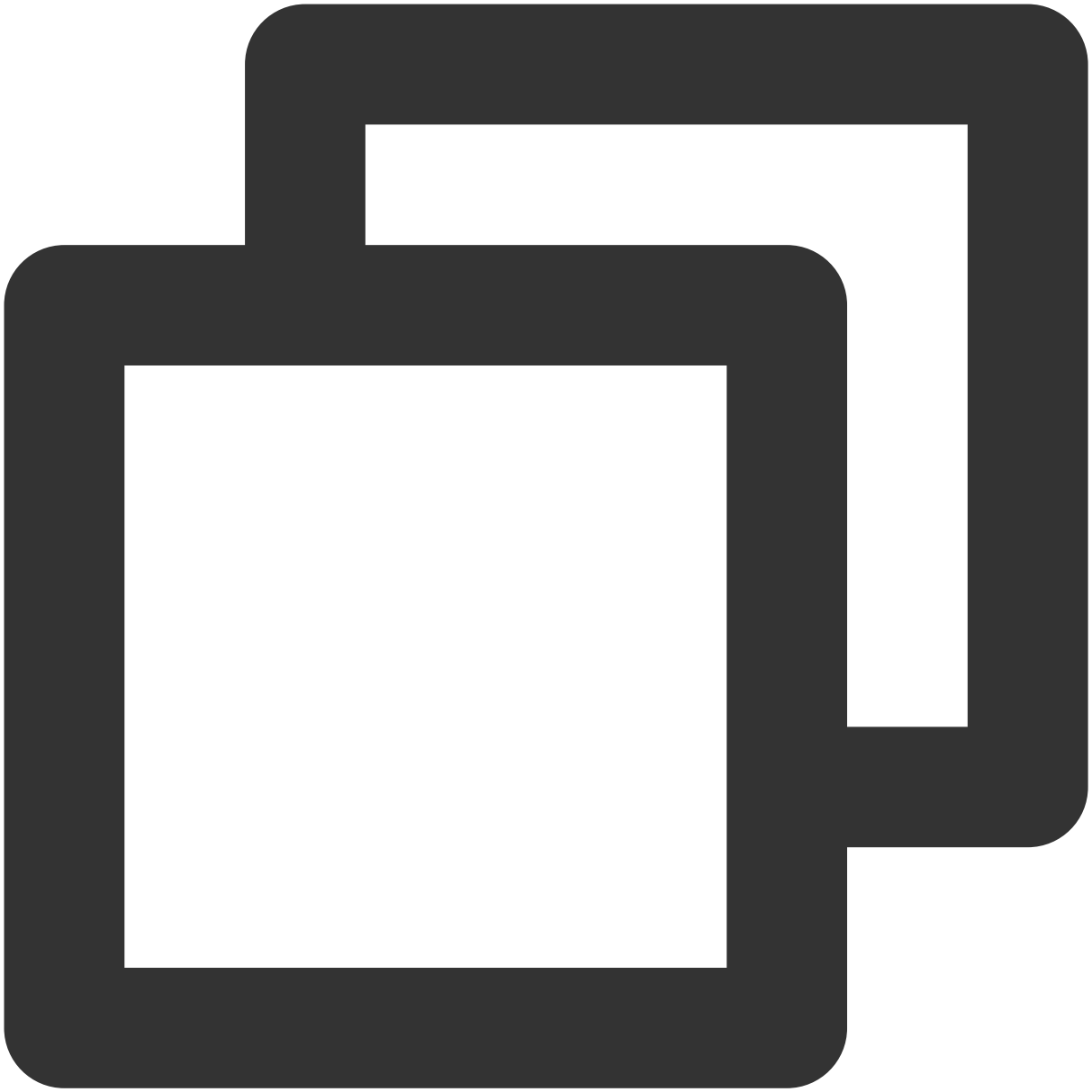
请求包头部增加图片处理参数 Pic-Operations，Pic-Operations 为 json 格式的字符串，具体参数如下：

参数名称	描述	类型	是否必选
is_pic_info	是否返回原图信息。0表示不返回原图信息，1表示返回原图信息，默认为0	Int	否
rules	处理规则，一条规则对应一个处理结果（目前最多支持五条规则），不填则不进行图片处理	Array	否

rules（json 数组）中每一项具体参数如下：

参数名称	描述	类型	是否必选
bucket	存储结果的目标存储桶名称，格式为：BucketName-APPID，如果不指定的话默认保存到当前存储桶	String	否
fileid	处理后文件的保存路径及名称。名称规则说明：例如源文件的路径及文件名为：/p1/test1.jpg 1. 以/开头为绝对路径，例如 fileid 值为 /p2/test2.jpg 时，表示在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 2. 不以/开头为相对路径，例如 fileid 值为 p2/test2.jpg 时，表示在 p1 文件夹中新建了一个 p2 文件夹，然后在 p2 文件夹中存储了一个文件名为 test2.jpg 的文件 3. 注意：请不要以/结尾，否则会产生空文件名	String	是
rule	处理参数，请参见图片处理 API。 若按指定样式处理，则以 `style/` 开头，后加样式名，如样式名为“test”，则 rule 字段为 `style/test`	String	是

提取盲水印需在 rule 中添加水印图参数（watermark），示例格式如下：



```
watermark/4/type/<type>/image/<imageUrl>
```

示例参数说明如下：

参数名称	描述	类型	是否必选
type	盲水印类型，有效值：1为半盲水印；2为全盲水印；3为文字盲水印，**必须跟添加盲水印时的 type 类型一致**。	Int	是
image	图片地址，根据 type 值填写：	String	否

	当 type 为1，则 image 必填，且为原图图片地址 当 type 为2，则 image 必填，且为水印图地址 当 type 为3，则 image 无需填写（无效） image 需要经过 URL 安全的 Base64 编码，指定的图片必须同时满足如下条件： 1. 图片与存在水印的图片必须位于同一个对象存储桶下； 2. URL 必须以http://开始，不能省略 http 头，也不能填 https 头，例如以下地址为非法水印地址： examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/watermark.png		

请求参数

此接口无请求参数。

请求体

此接口请求的请求体为图片内容。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

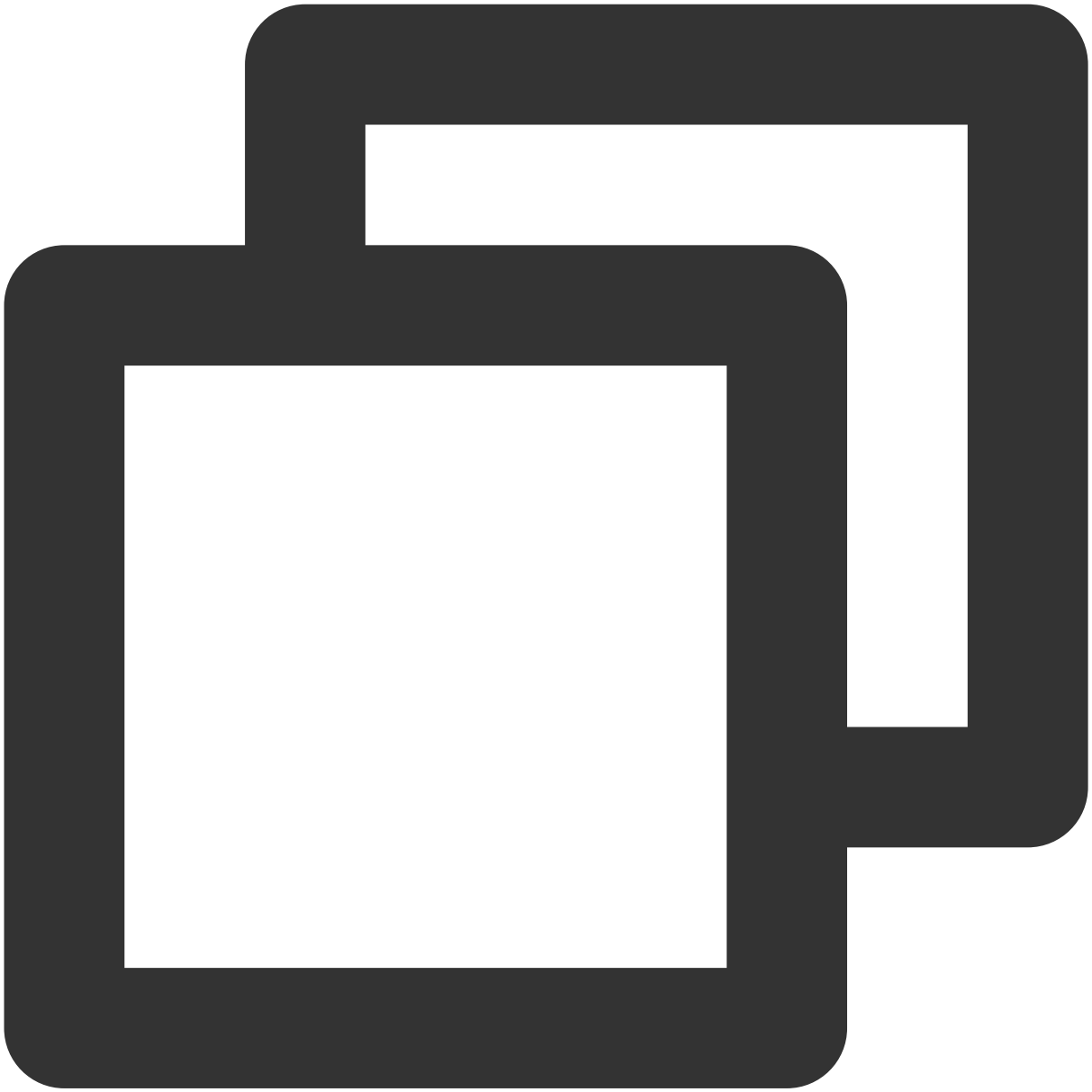
响应体

响应体的 UploadResult.ProcessResults.Object 字段中新增 WatermarkStatus 字段，当盲水印提取的请求体中 type 参数为2时携带 WatermarkStatus 参数，其他情况不返回该参数。

参数名称	父节点	描述	类型
WatermarkStatus	Object	当 type 为2时返回该字段，表示提取到全盲水印的可信度。具体为0-100的数字，75分以上表示确定有盲水印，60-75表示疑似有盲水印，60以下可认为未提取到盲水印	Int

实际案例

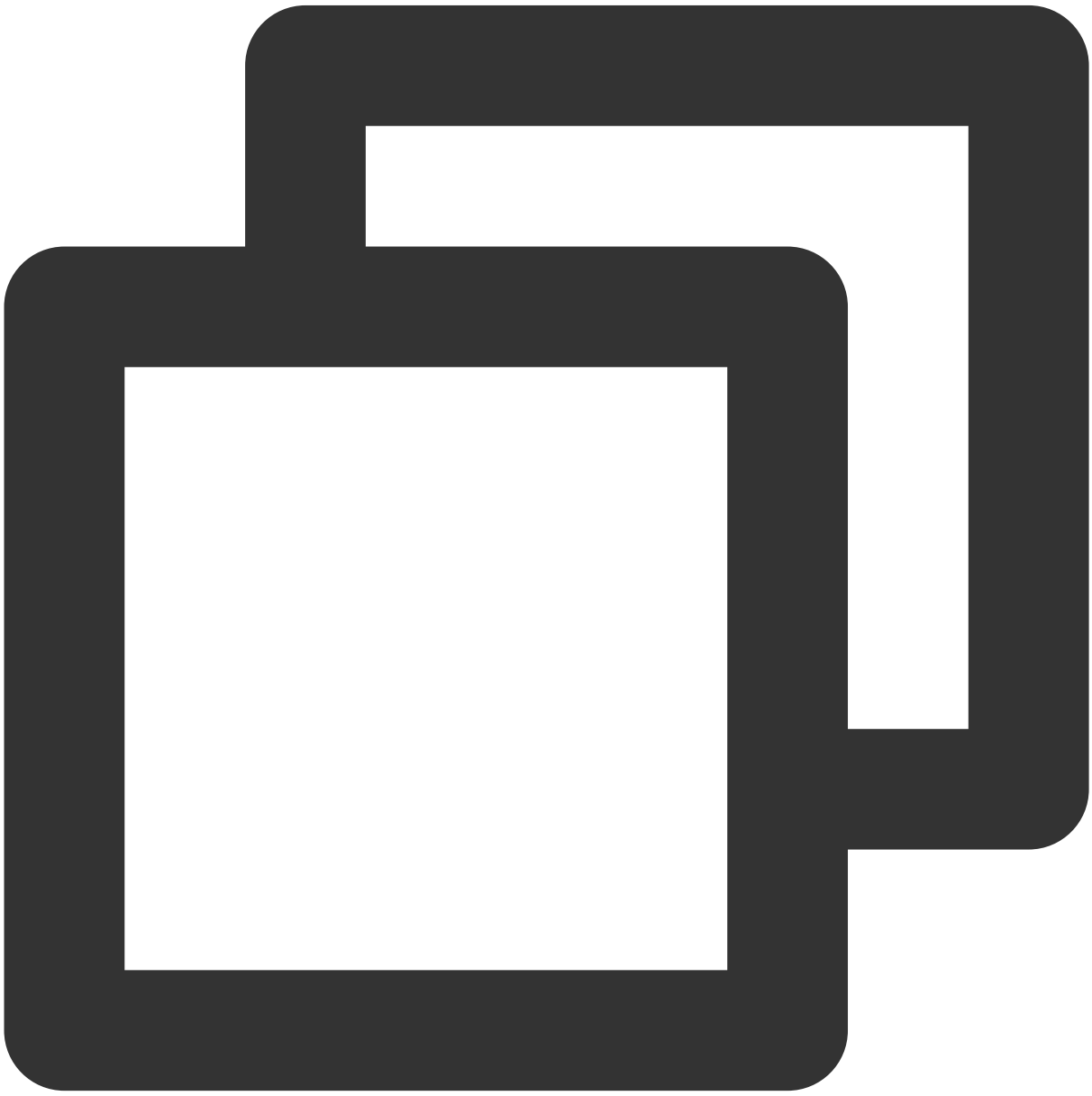
请求



```
PUT /exampleobject1 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2018 09:06:15 GMT
Authorization:XXXXXXXXXXXX

Pic-Operations:
{
    "is_pic_info": 1,
    "rules": [{
        "fileid": "exampleobject2",
        "rule": "watermark/4/type/2/image/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQucGljc
```

```
    }]
}
Content-Length: 64

[Object Content]
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
```

```
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
      <OriginalInfo>
          <Key>exampleobject1</Key>
          <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/exampleobj
          <ImageInfo>
                <Format>JPEG</Format>
                <Width>640</Width>
                <Height>427</Height>
                <Quality>100</Quality>
                <Ave>0xa18454</Ave>
                <Orientation>1</Orientation>
                <FrameCount>1</FrameCount>
          </ImageInfo>
      </OriginalInfo>
      <ProcessResults>
          <Object>
                <Key>exampleobject2</Key>
                <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/exam
                <Format>png</Format>
                <Width>640</Width>
                <Height>427</Height>
                <Size>463092</Size>
                <Quality>100</Quality>
                <FrameCount>1</FrameCount>
                <WatermarkStatus>86</WatermarkStatus>
          </Object>
      </ProcessResults>
</UploadResult>
```

注意

对于 type 值为1的请求，请求头 Pic-Operations 的 rule 字段中，imageUrl 为未添加盲水印的原图地址；对于 type 值为2的请求，包头 Pic-Operations 的 rule 字段中，imageUrl 为水印图的地址。

响应体 ProcessResults 字段的 url 为提取出来的水印图地址。

请求二：对 COS 上的图片提取盲水印

对 COS 上的图片提取盲水印的请求包与 云上数据处理 接口一致，只需在请求包头部增加图片处理参数 Pic-Operations 并使用提取盲水印参数（watermark/4）即可。

AI 内容识别
队列接口
查询队列

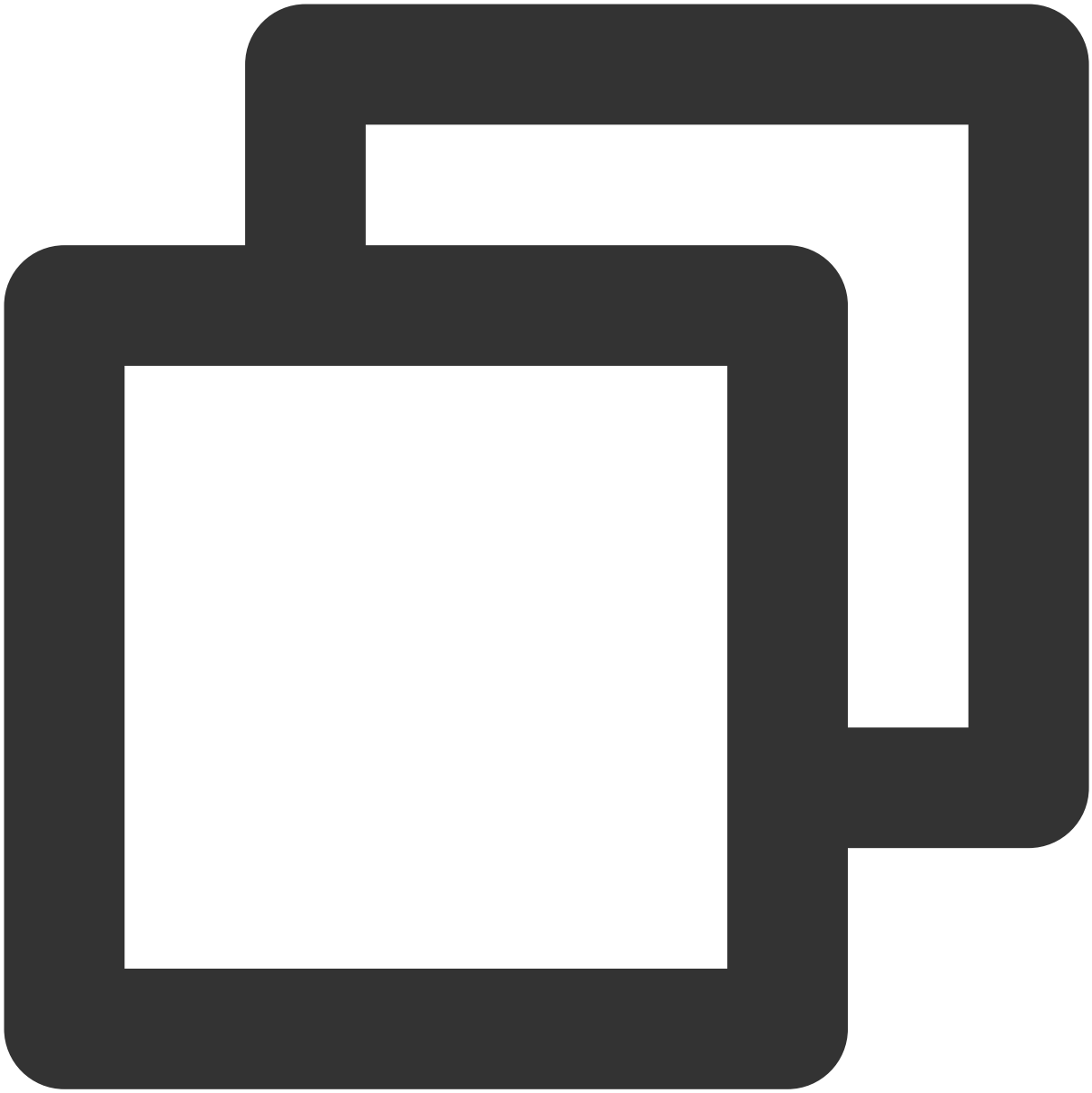
最近更新时间：2023-12-22 10:53:54

功能描述

搜索 AI 内容识别队列。

请求

请求示例



```
GET /ai_queue HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

参数名称（关键字）	描述	类型	是否必选
queueId	队列 ID，以“,”符号分割字符串	String	否
state	1. Active 表示队列内的作业会被媒体处理服务调度执行 2. Paused 表示队列暂停，作业不再会被媒体处理调度执行，队列内的所有作业状态维持在暂停状态，已经执行中的任务不受影响	String	否
pageNumber	第几页，默认值1	String	否
pageSize	每页个数，默认值10	String	否

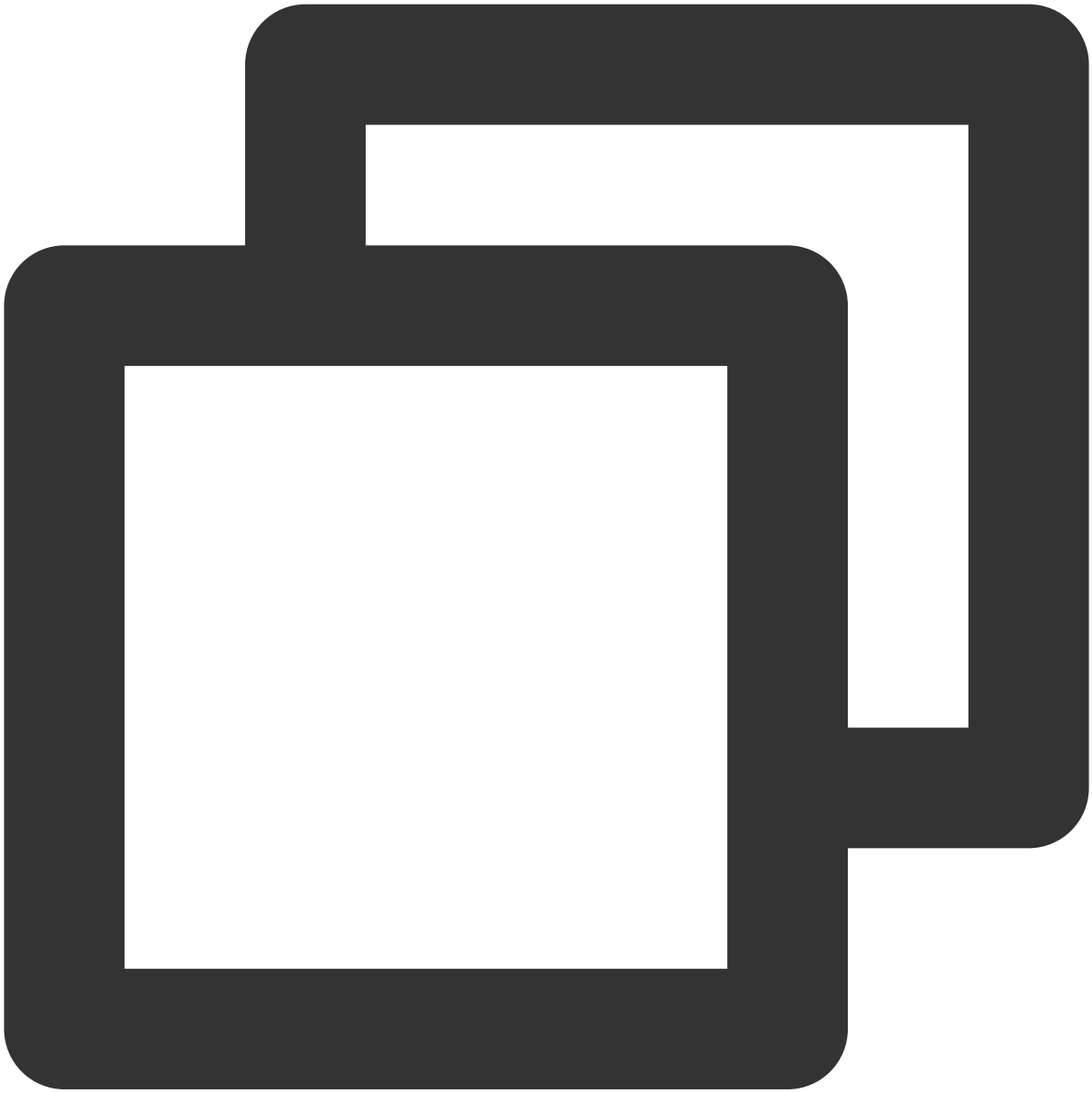
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，当指定 pageNumber 和 pageSize 时包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMzI5MmRfOTBmYTUwNjRfNjcwM18x</RequestId>
    <TotalCount>1</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <QueueList>
        <QueueId>p23f69f4779e64b18aa61e70d5416c071</QueueId>
        <Name>queue-ai-process-1</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
            <ResultFormat>JSON</ResultFormat>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:23:11+0800</CreateTime>
        <Category>AIProcess</Category>
    </QueueList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	队列总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
QueueList	Response	队列数组	Container数组
NonExistPIDs	Response	不存在的队列 ID 列表	String数组

Container 节点 QueueList 的内容：

节点名称（关键字）	父节点	描述	类型
QueueId	Response.QueueList	队列 ID	String
Name	Response.QueueList	队列名字	String
State	Response.QueueList	当前状态，Active 或者 Paused	String
NotifyConfig	Response.QueueList	回调配置	Container
MaxSize	Response.QueueList	队列最大长度	Int

版权所有：腾讯云计算（北京）有限责任公司

MaxConcurrent	Response.QueueList	当前队列最大并行执行的任务数	Int
Category	Response.QueueList	队列类型	String
UpdateTime	Response.QueueList	更新时间	String
CreateTime	Response.QueueList	创建时间	String

Container 节点 NotifyConfig 的内容：

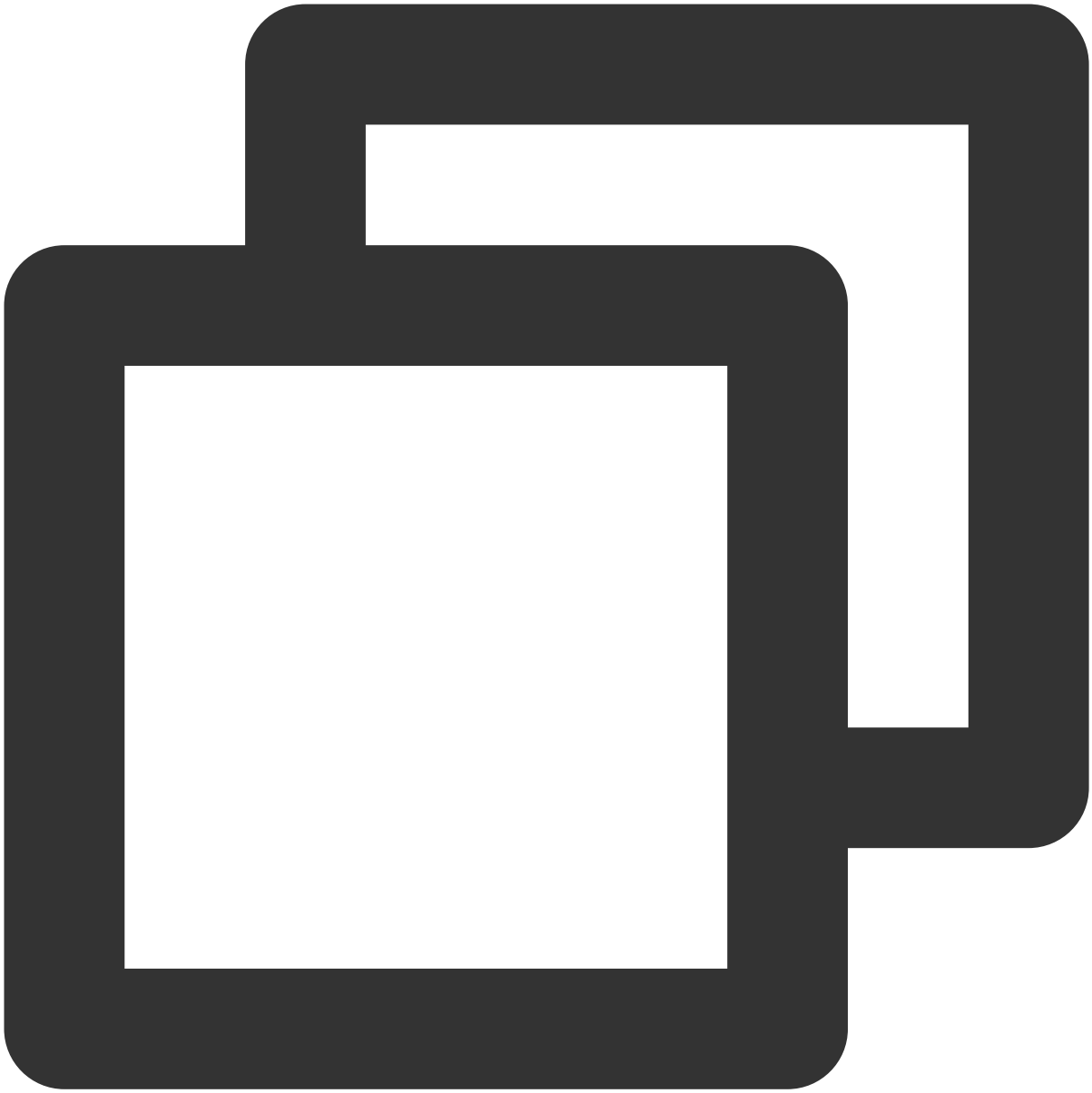
节点名称（关键字）	父节点	描述	类型
Url	Response.QueueList.NotifyConfig	回调地址	String
State	Response.QueueList.NotifyConfig	开关状态，On 或者 Off	String
Type	Response.QueueList.NotifyConfig	回调类型，Url	String
Event	Response.QueueList.NotifyConfig	任务完成：TaskFinish；工作流完成：WorkflowFinish	String
ResultFormat	Response.QueueList.NotifyConfig	回调类型，XML 或者 JSON	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

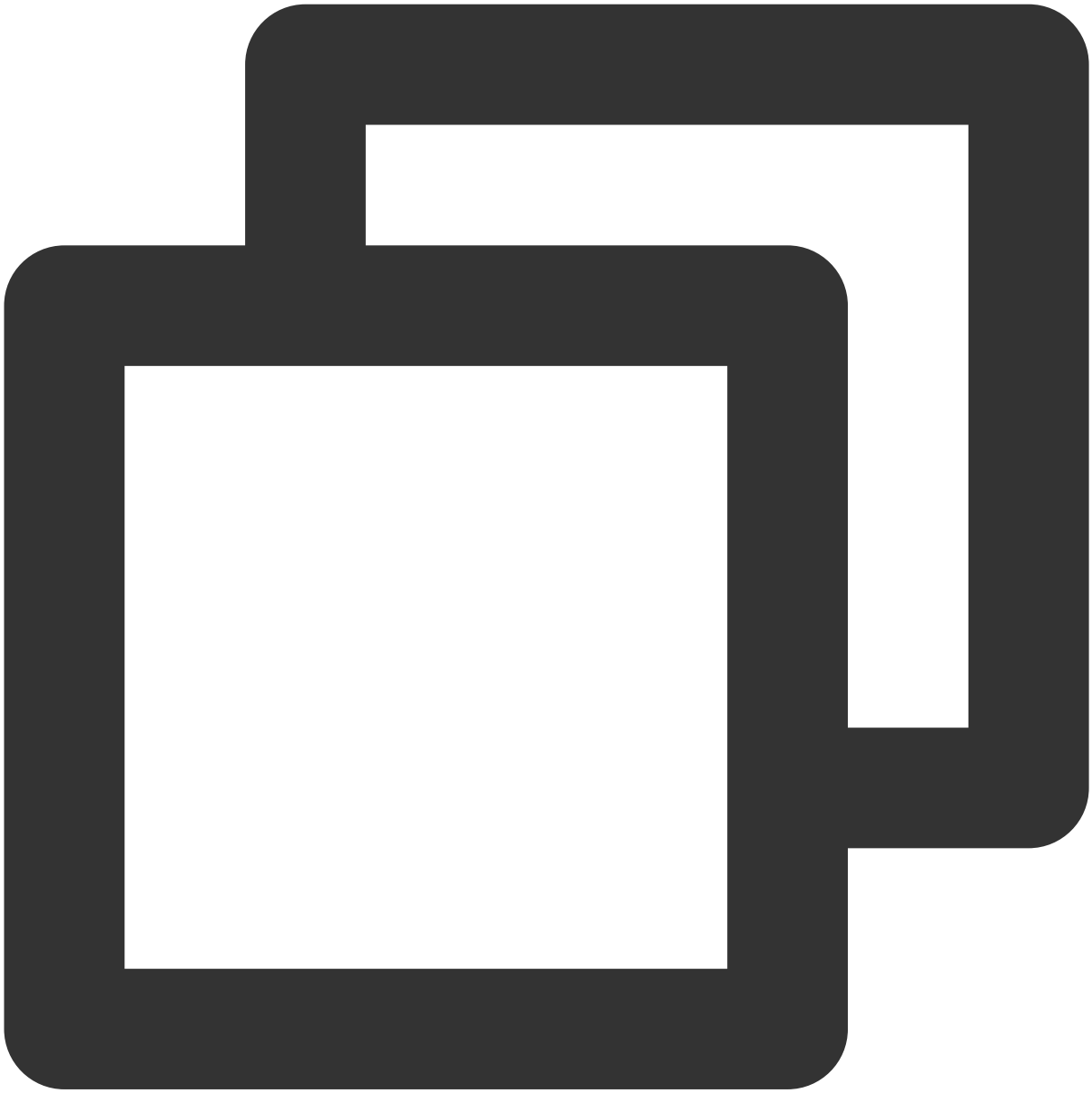
实际案例

请求1：搜索指定队列



```
GET /ai_queue?queueIds=p23f69f4779e64b18aa61e70d5416c071,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

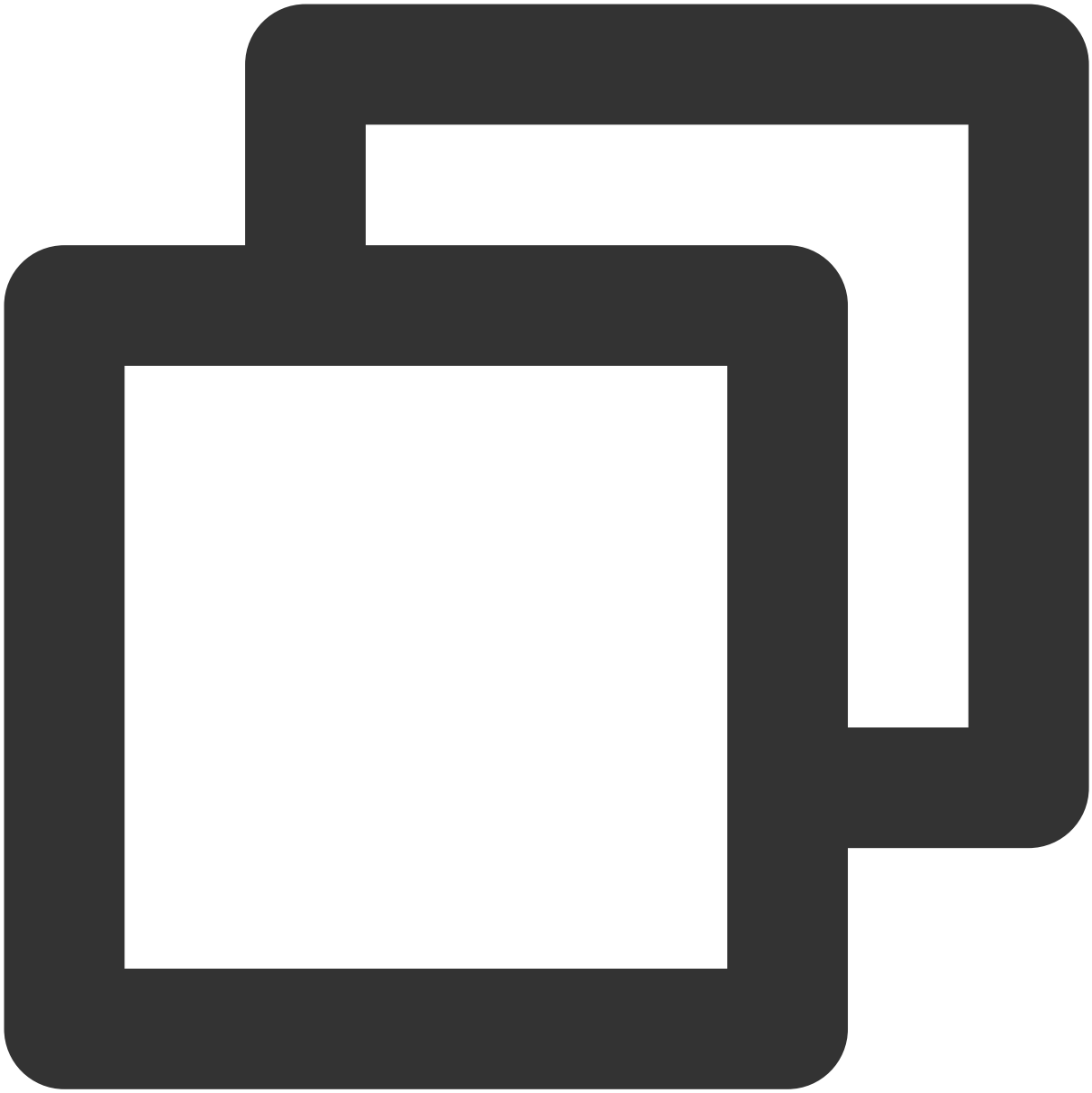
响应1



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x

<Response>
    <RequestId>NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x</RequestId>
    <QueueList>
```

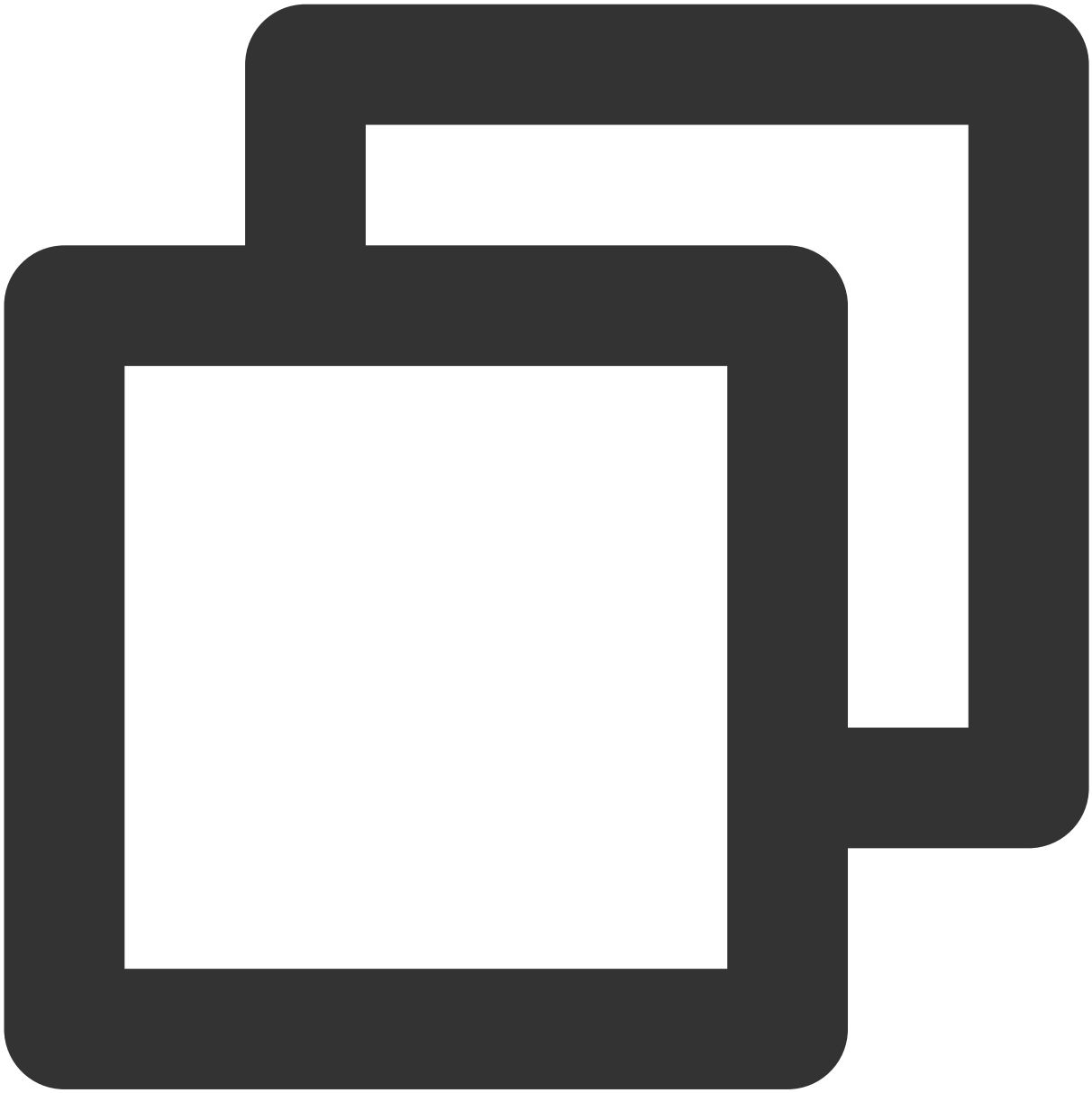
```
        <QueueId>p23f69f4779e64b18aa61e70d5416c071</QueueId>
        <Name>queue-ai-process-1</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
            <ResultFormat>JSON</ResultFormat>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:23:11+0800</CreateTime>
        <Category>AIProcess</Category>
    </QueueList>
    <NonExistPIDs>A</NonExistPIDs>
    <NonExistPIDs>B</NonExistPIDs>
</Response>
```

请求2：查询队列列表



```
GET /ai_queue?pageSize=10&pageNumber=1&category=CateAll HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应2



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMzI5MmRfOTBmYTUwNjRfNjcwM18x

<Response>
    <RequestId>NjJmMzI5MmRfOTBmYTUwNjRfNjcwM18x</RequestId>
    <TotalCount>1</TotalCount>
```

```
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <QueueList>
        <QueueId>p23f69f4779e64b18aa61e70d5416c071</QueueId>
        <Name>queue-ai-process-1</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
            <ResultFormat>JSON</ResultFormat>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:23:11+0800</CreateTime>
        <Category>AIProcess</Category>
    </QueueList>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新队列

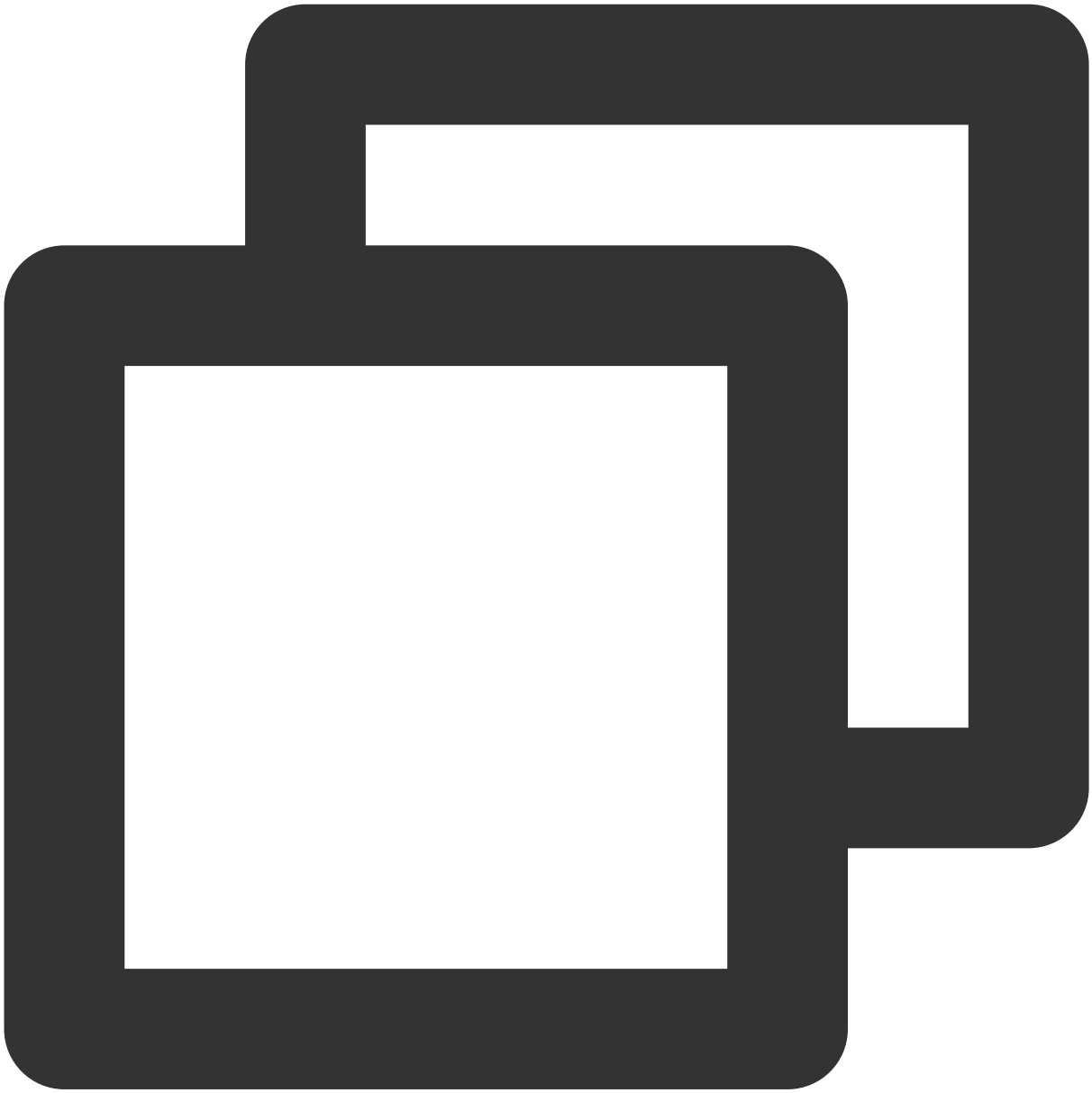
最近更新时间：2023-12-22 10:54:36

功能描述

更新 AI 内容识别队列。

请求

请求示例



```
PUT /ai_queue/<queueId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

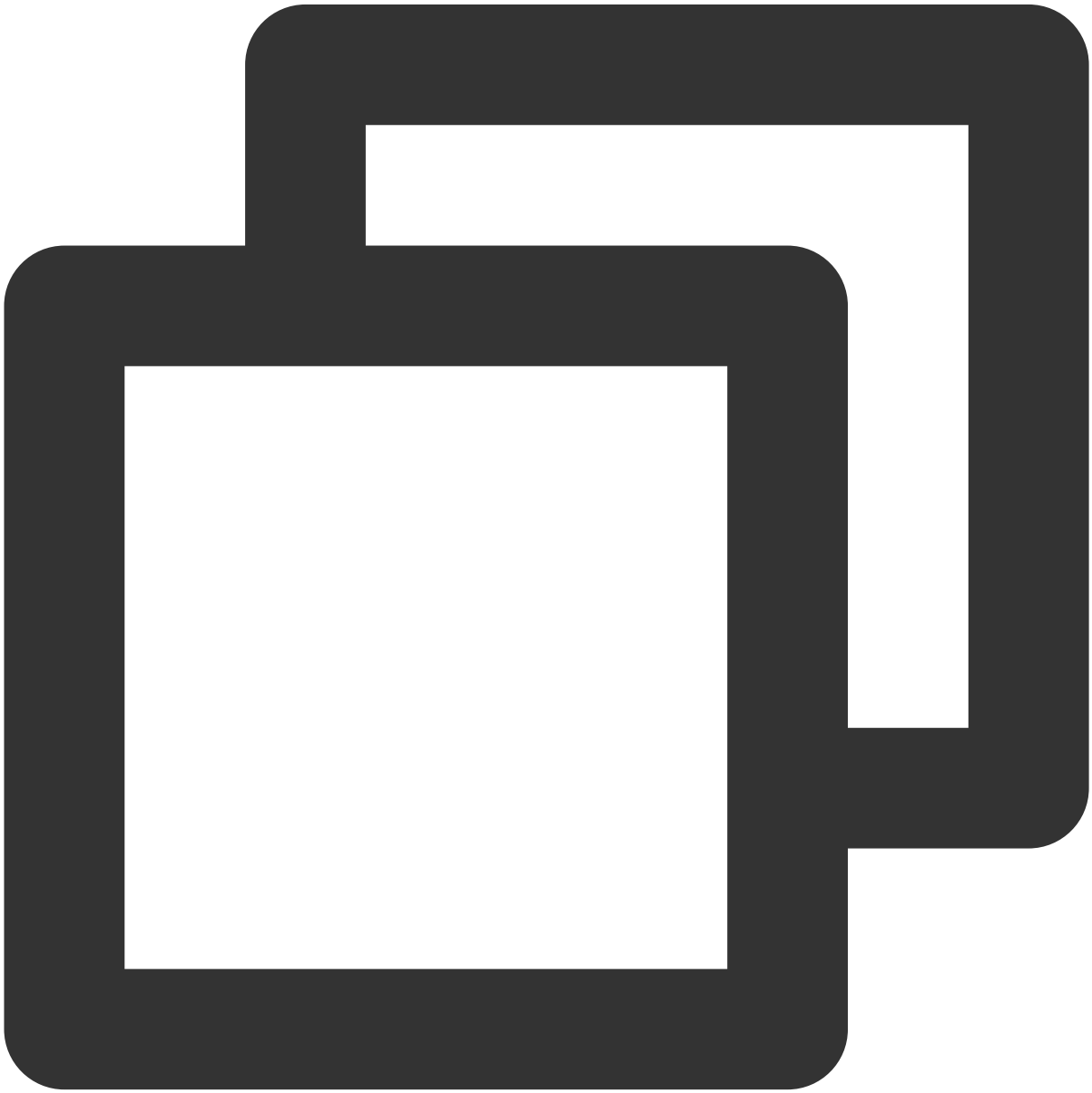
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有以下请求体。



```
<Request>
```

```
    <Name>My-Queue-AI</Name>
    <State>Active</State>
    <NotifyConfig>
        <State>On</State>
        <Url>http://callback.demo.com</Url>
        <Type>Url</Type>
        <Event>TaskFinish,WorkflowFinish</Event>
        <ResultFormat>JSON</ResultFormat>
    </NotifyConfig>
</Request>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Request	无	保存结果的容器	Container	是	无	无

Container 节点 Request 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Name	Request	队列名称，长度不超过128	String	是	无	无
State	Request	1. Active 表示队列内的作业会被媒体处理服务调度执行 2. Paused 表示队列暂停，作业不再会被媒体处理调度执行，队列内的所有作业状态维持在暂停状态，已经执行中的任务不受影响	String	是	无	无
NotifyConfig	Request	回调配置	Container	是	无	无

Container 节点 NotifyConfig 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
State	Request.NotifyConfig	回调开关，Off/On	String	否	Off	无
Type	Request.NotifyConfig	回调类型	String	当 State=On 时，必选	无	普通回调：Url
Url	Request.NotifyConfig	回调地址	String	当 State=On 时，必选	无	无

版权所有：腾讯云计算（北京）有限责任公司

Event	Request.NotifyConfig	回调事件	String	当 State=On 时，必选	无	任务完成：TaskFinish；工作流完成：WorkflowFinish
ResultFormat	Request.NotifyConfig	回调格式	String	否	XML	JSON/XML

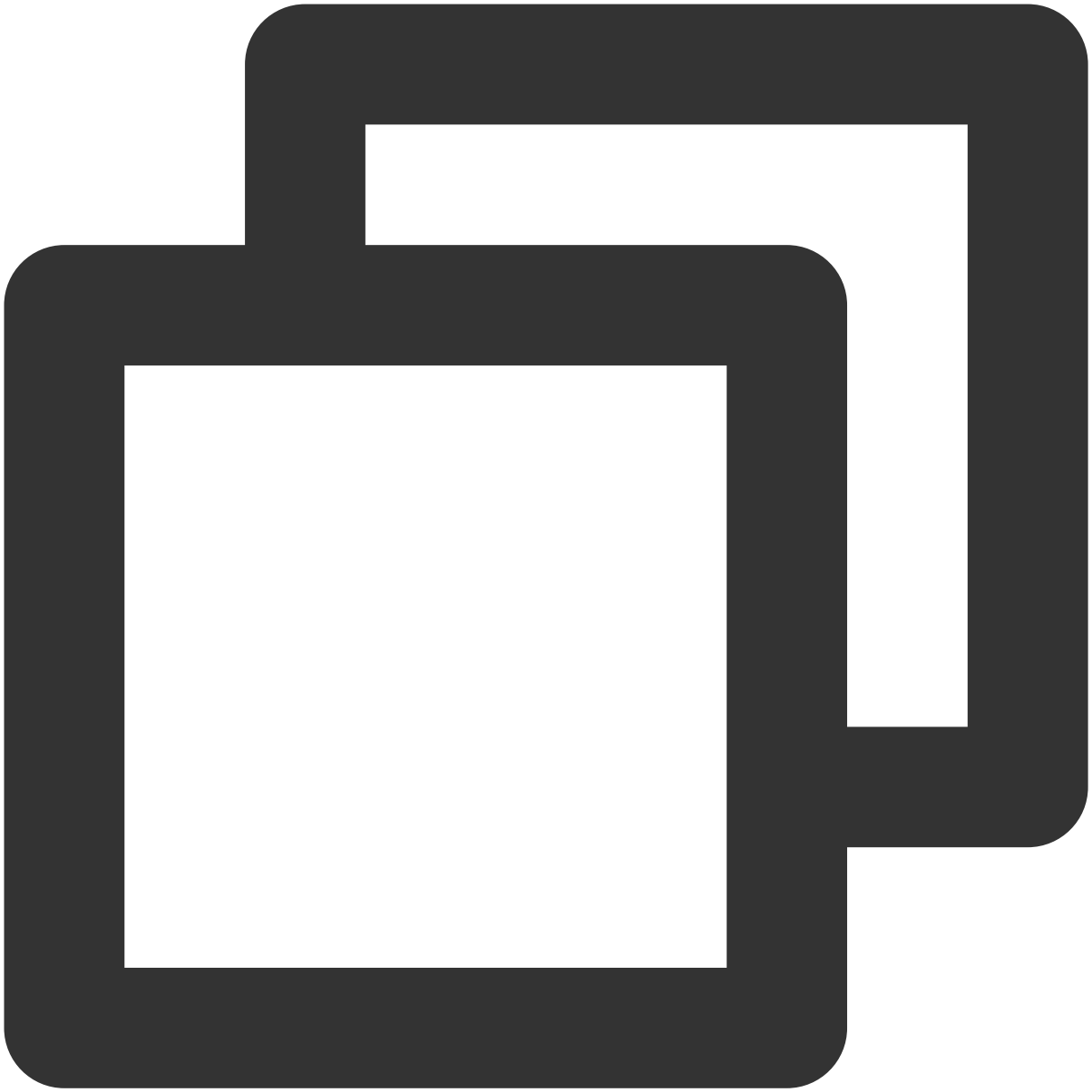
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <Queue>
        <QueueId>p23f69f4779e64b18aa61e70d5416c071</QueueId>
        <Name>My-Queue-AI</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
```

```
            <ResultFormat>JSON</ResultFormat>
        </NotifyConfig>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
Queue	Response	队列信息	Container

Container 节点 Queue 的内容：

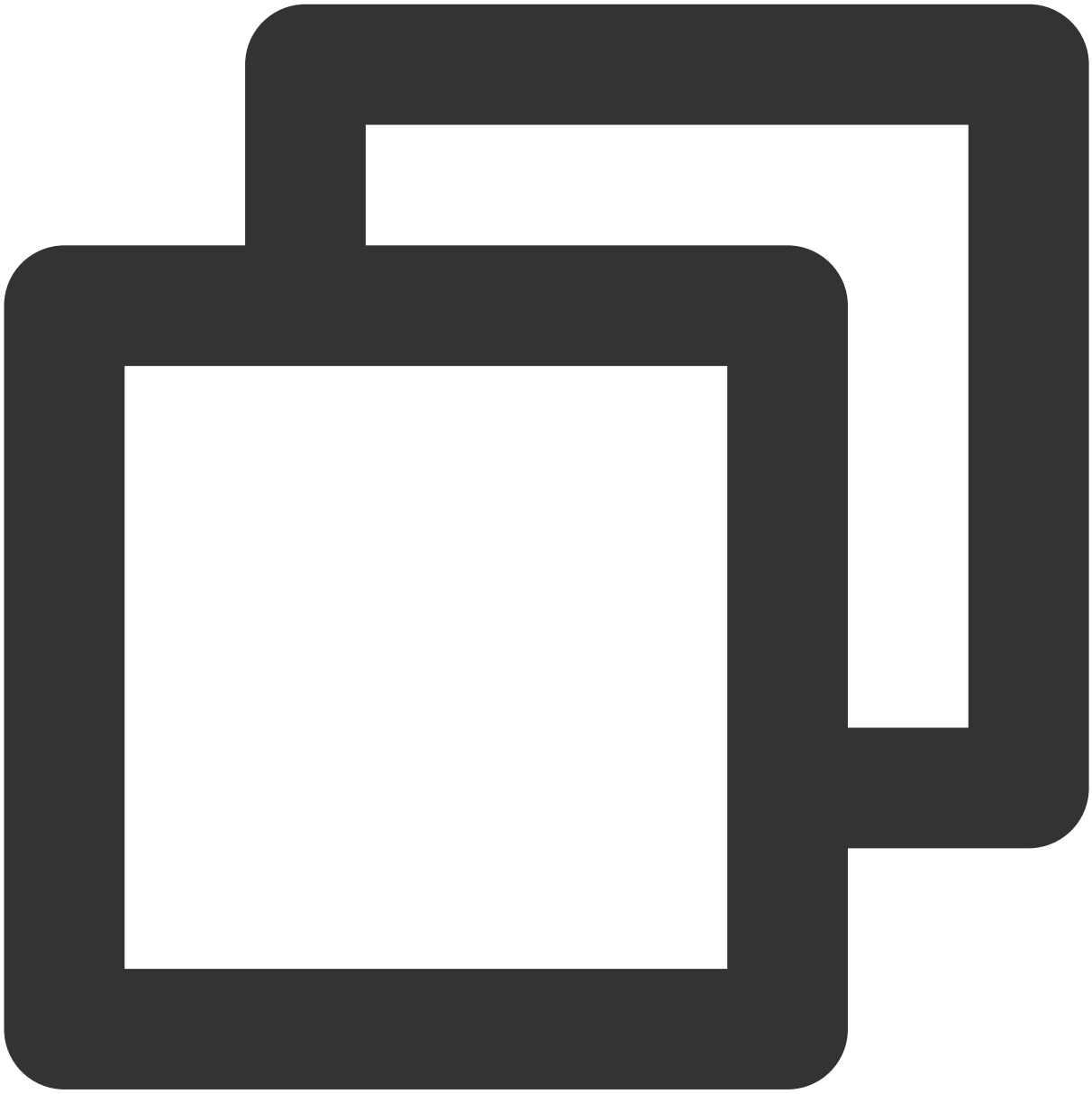
节点名称（关键字）	父节点	描述	类型
QueueId	Response.Queue	队列 ID	String
Name	Response.Queue	队列名字	String
State	Response.Queue	当前状态，Active 或者 Paused	String
NotifyConfig	Response.Queue	回调配置,同 Request.NotifyConfig	Container
MaxSize	Response.Queue	队列最大长度	Int
MaxConcurrent	Response.Queue	当前队列最大并行执行的任务数	Int
Category	Response.Queue	队列类型	String
UpdateTime	Response.Queue	更新时间	String
CreateTime	Response.Queue	创建时间	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

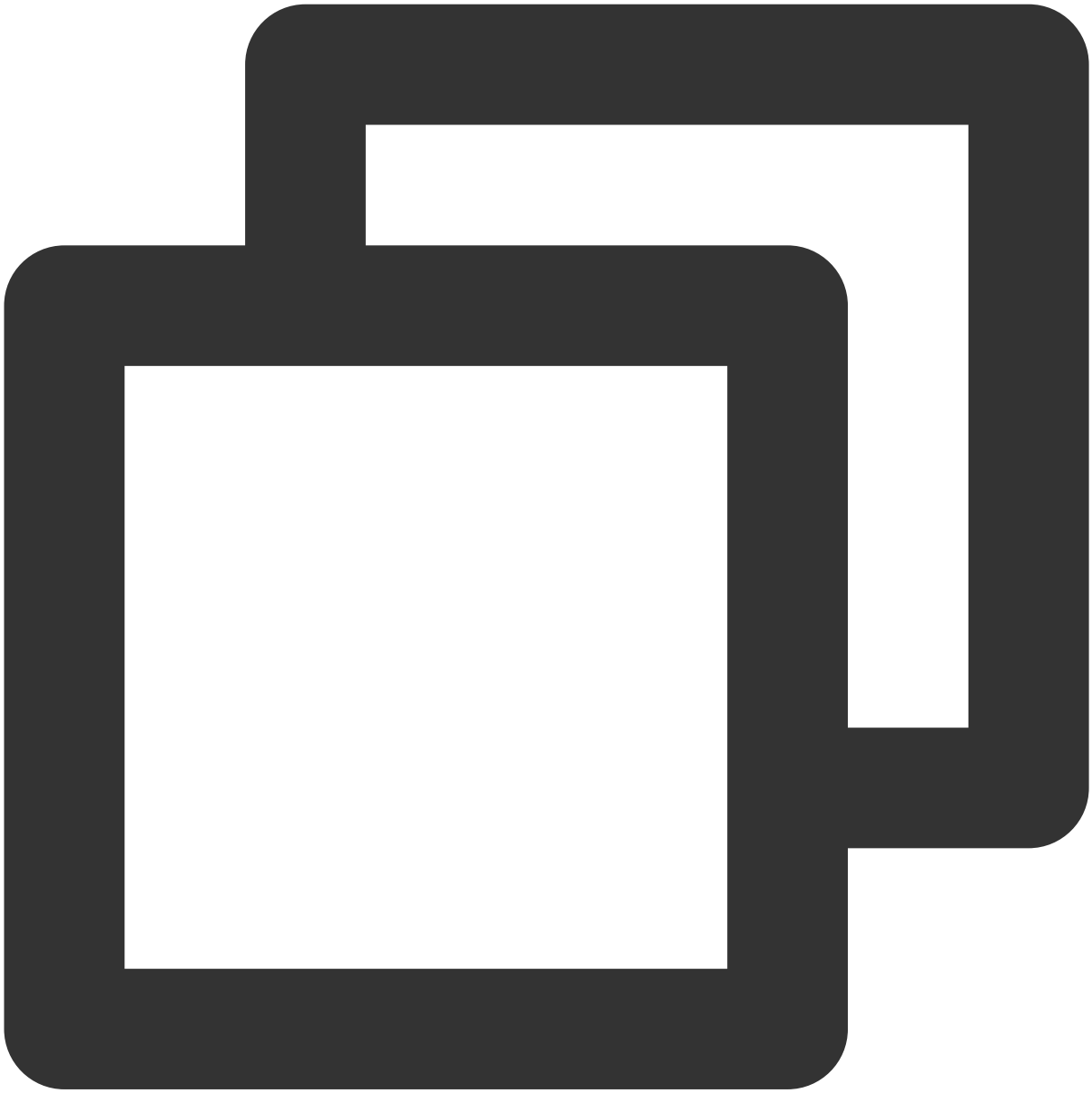
请求



```
PUT /ai_queue/p23f69f4779e64b18aa61e70d5416c071 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 162
Content-Type: application/xml
```

```
<Request>
    <Name>My-Queue-AI</Name>
    <State>Active</State>
    <NotifyConfig>
        <State>On</State>
        <Url>http://callback.demo.com</Url>
        <Type>Url</Type>
        <Event>TaskFinish,WorkflowFinish</Event>
        <ResultFormat>JSON</ResultFormat>
    </NotifyConfig>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85

<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <Queue>
```

```
        <QueueId>p23f69f4779e64b18aa61e70d5416c071</QueueId>
        <Name>My-Queue-AI</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
            <ResultFormat>JSON</ResultFormat>
        </NotifyConfig>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

服务配置
开通 AI 内容识别

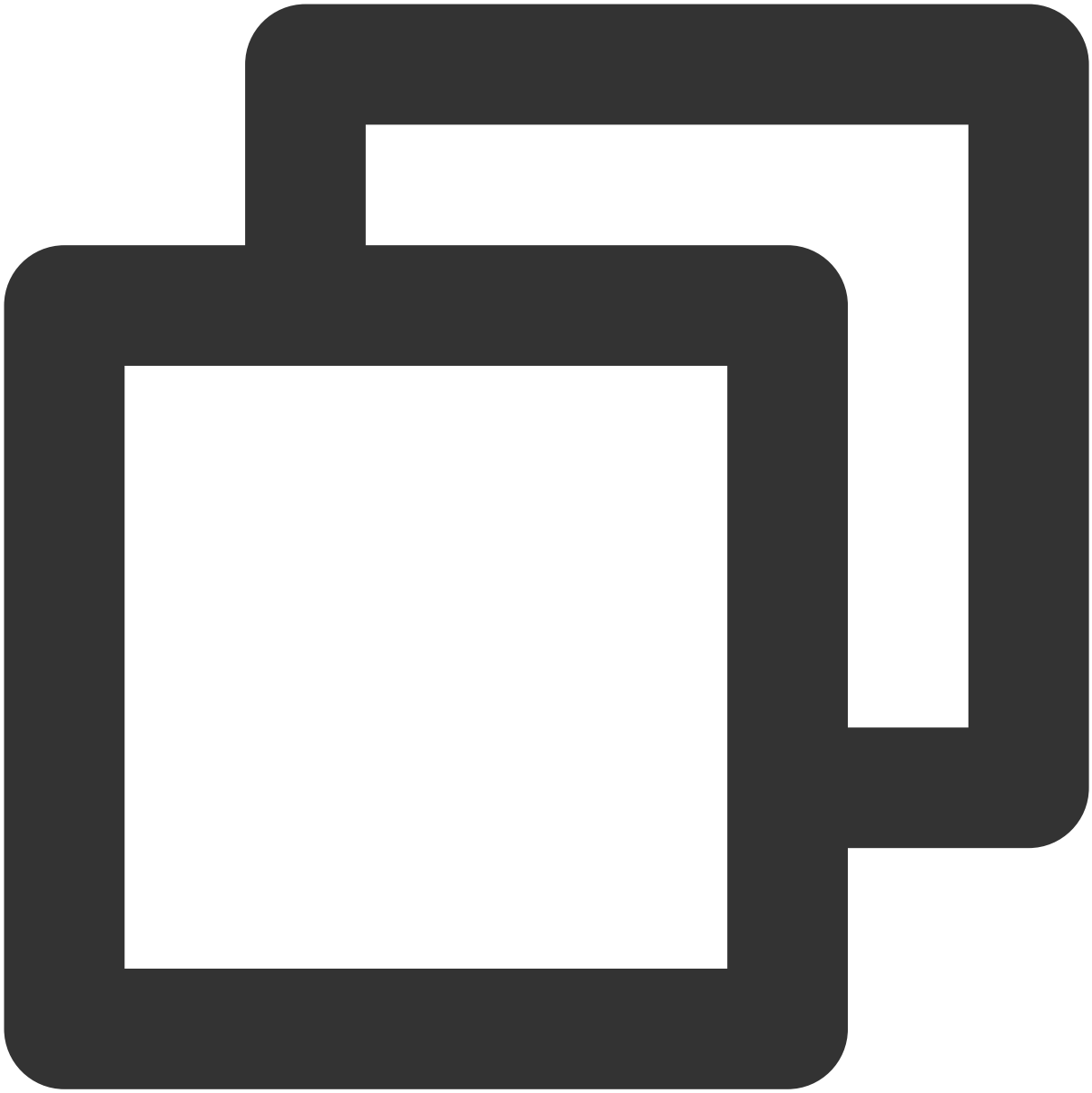
最近更新时间：2023-12-22 10:54:29

功能描述

开通 AI 内容识别服务并生成队列。

请求

请求示例



```
POST /ai_bucket HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见公共请求头部 文档。

请求体

该请求的请求体为空。

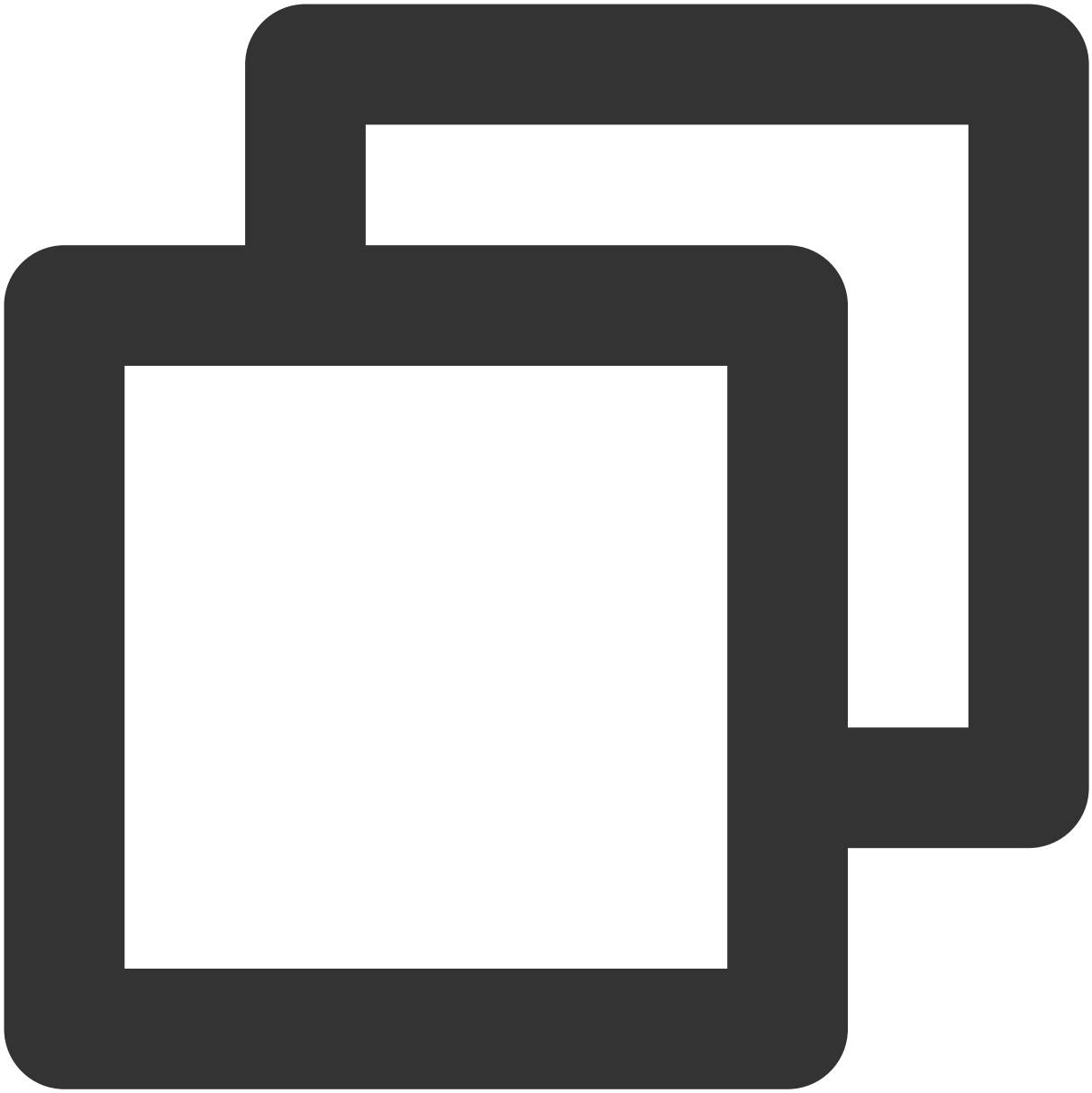
响应

响应头

此接口仅返回公共响应头部，详情请参见公共响应头部文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMzRlZGZfOTBmYTUwNjRfNzI0MV8x</RequestId>
    <AiBucket>
        <BucketId>test-1234567890</BucketId>
        <Name>test-1234567890</Name>
        <Region>ap-chongqing</Region>
        <CreateTime2022-08-09T16:23:11+0800></CreateTime>
    </AiBucket>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
AiBucket	Response	存储桶信息	Container

Container 节点 AiBucket 的内容：

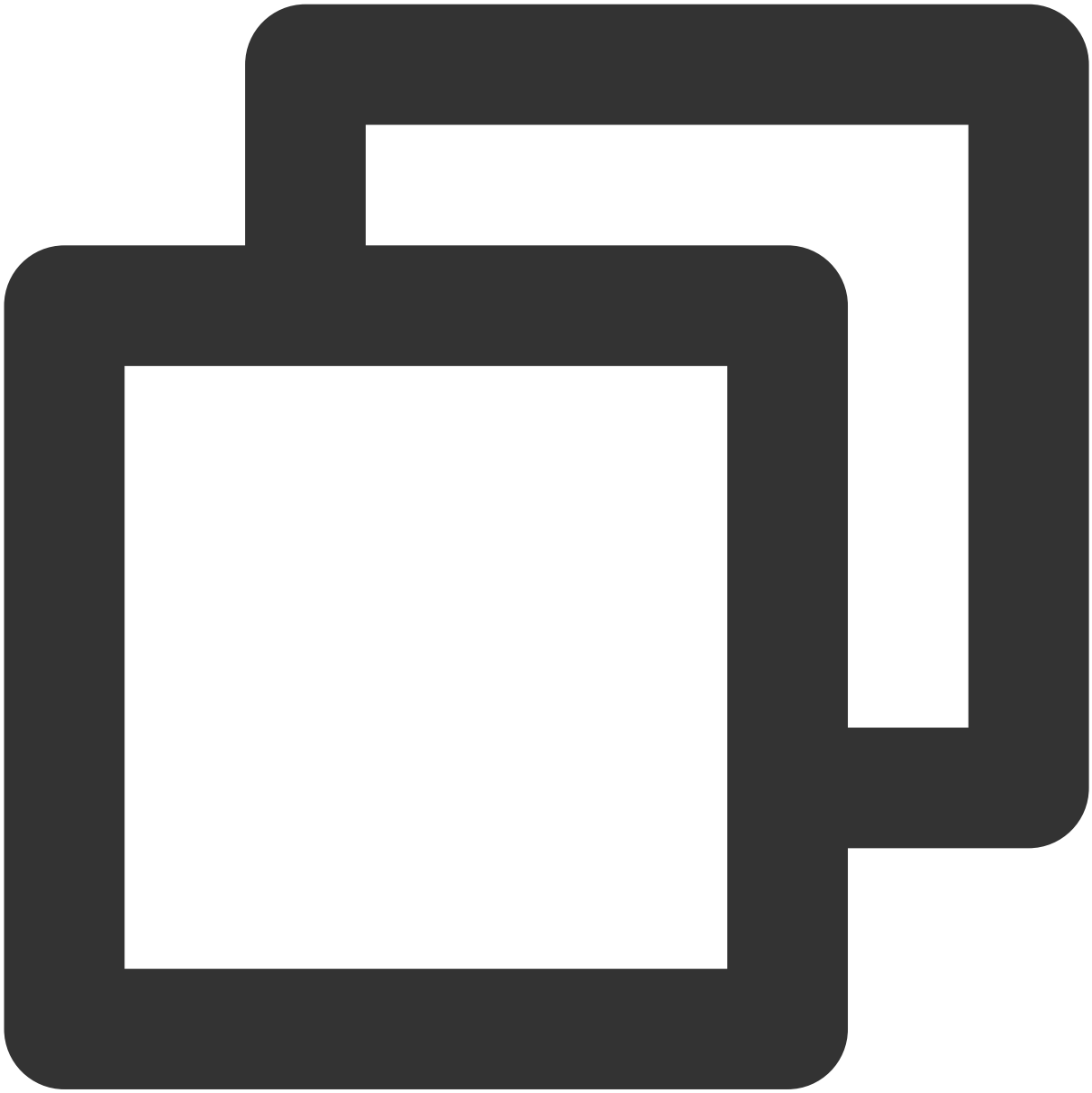
节点名称（关键字）	父节点	描述	类型
BucketId	Response.AiBucket	存储桶 ID	String
Name	Response.AiBucket	存储桶名称，同 BucketId	String
Region	Response.AiBucket	所在的地域	String
CreateTime	Response.AiBucket	创建时间	String

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

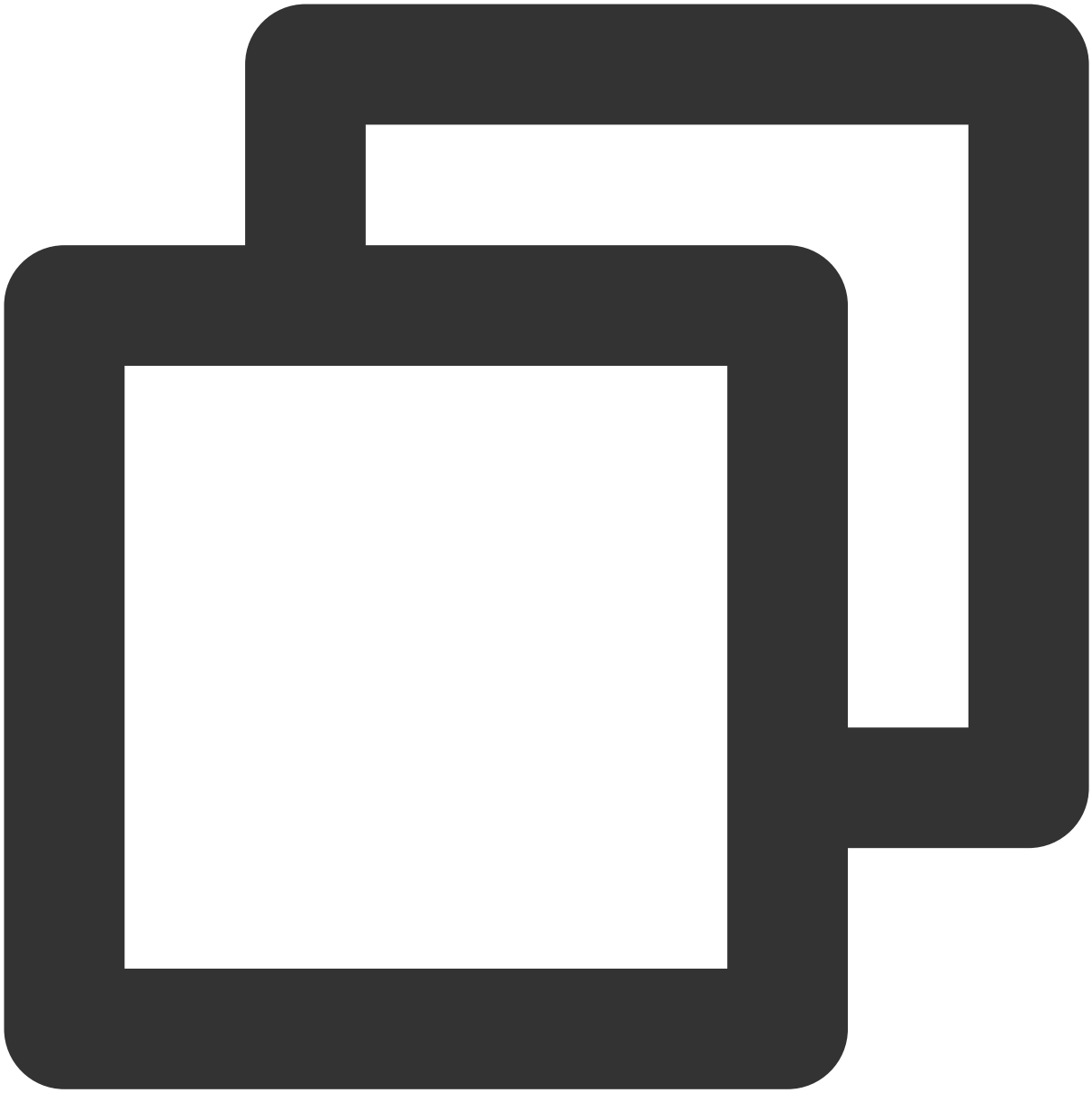
实际案例

请求



```
POST /ai_bucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMzRlZGZfOTBmYTUwNjRfNzI0MV8x

<Response>
    <RequestId>NjJmMzRlZGZfOTBmYTUwNjRfNzI0MV8x</RequestId>
    <AiBucket>
```

```
        <BucketId>test-1234567890</BucketId>
        <Name>test-1234567890</Name>
        <Region>ap-chongqing</Region>
        <CreateTime2022-08-09T16:23:11+0800></CreateTime>
    </AiBucket>
</Response>
```

查询 AI 内容识别

最近更新时间：2022-09-27 15:34:42

功能描述

查询 AI 内容识别服务状态。

推荐使用 API Explorer

点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

请求

请求示例

```
GET /ai_bucket HTTP/1.1
Host: ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明：

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

名称	描述	类型	是否必选

名称	描述	类型	是否必选
regions	地域信息，例如 ap-shanghai、ap-beijing，若查询多个地域以“,”分隔字符串，支持中国大陆地域，详情请参见 地域与域名	string	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	string	否
bucketName	存储桶名称前缀，前缀搜索	string	否
pageNumber	第几页，默认值1	String	否
pageSize	每页个数，默认值10	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

```
<Response>
<RequestId>NjJmMzY1Y2NfOTBmYTUwNjRfNTA5YV8y</RequestId>
<TotalCount>2</TotalCount>
<PageNumber>1</PageNumber>
<PageSize>5</PageSize>
<AiBucketList>
<BucketId>test-1234567890</BucketId>
<Name>test-1234567890</Name>
<Region>ap-chongqing</Region>
<CreateTime2022-08-08T16:23:11+0800></CreateTime>
</AiBucketList>
<AiBucketList>
<BucketId>test-1-1234567890</BucketId>
<Name>test-1-1234567890</Name>
<Region>ap-chongqing</Region>
<CreateTime2022-08-09T16:23:11+0800></CreateTime>
</AiBucketList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	媒体 Bucket 总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
AiBucketList	Response	媒体 Bucket 列表	Container

Container 节点 AiBucketList 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.AiBucketList	存储桶 ID	String
Name	Response.AiBucketList	存储桶名称，同 BucketId	String
Region	Response.AiBucketList	所在的地域	String
CreateTime	Response.AiBucketList	创建时间	String

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /ai_bucket?regions=ap-chongqing&pageSize=5&bucketName=t HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sign-time=1497530202;1497610202&q-key-time=1497530202;1497610202&q-header-list=&q-url-param-list=&q-signature=28e9a4986df11bed0255e97ff90500557e0e****
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

```
Content-Length: 0
Content-Type: application/xml
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMzY1Y2NfOTBmYTUwNjRfNTA5YV8y
<Response>
<RequestId>NjJmMzY1Y2NfOTBmYTUwNjRfNTA5YV8y</RequestId>
<TotalCount>2</TotalCount>
<PageNumber>1</PageNumber>
<PageSize>5</PageSize>
<AiBucketList>
<BucketId>test-1234567890</BucketId>
<Name>test-1234567890</Name>
<Region>ap-chongqing</Region>
<CreateTime2022-08-08T16:23:11+0800></CreateTime>
</AiBucketList>
<AiBucketList>
<BucketId>test-1-1234567890</BucketId>
<Name>test-1-1234567890</Name>
<Region>ap-chongqing</Region>
<CreateTime2022-08-09T16:23:11+0800></CreateTime>
</AiBucketList>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

媒体处理
媒体 bucket 接口
查询媒体处理开通状态

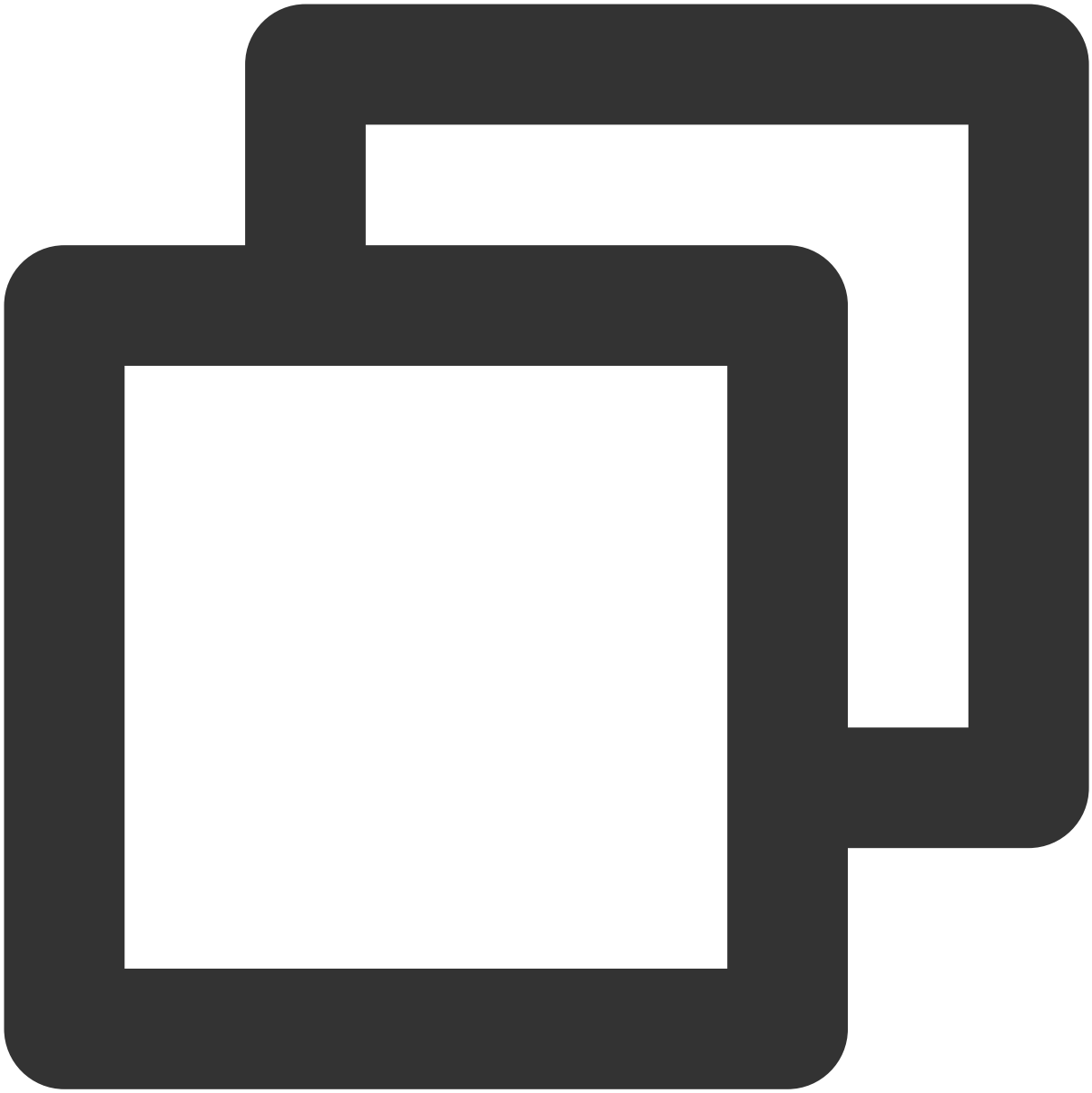
最近更新时间：2023-12-22 10:55:50

功能描述

DescribeMediaBuckets 接口用于查询已经开通媒体处理功能的存储桶。

请求

请求示例



```
GET /mediabucket HTTP/1.1
Host: ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

名称	描述	类型	是否必选
regions	地域信息，例如 ap-shanghai、ap-beijing，若查询多个地域以“,”分隔字符串，支持中国大陆地域，详情请参见 地域与域名	string	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	string	否
bucketName	存储桶名称前缀，前缀搜索	string	否
pageNumber	第几页	string	否
pageSize	每页个数	string	否

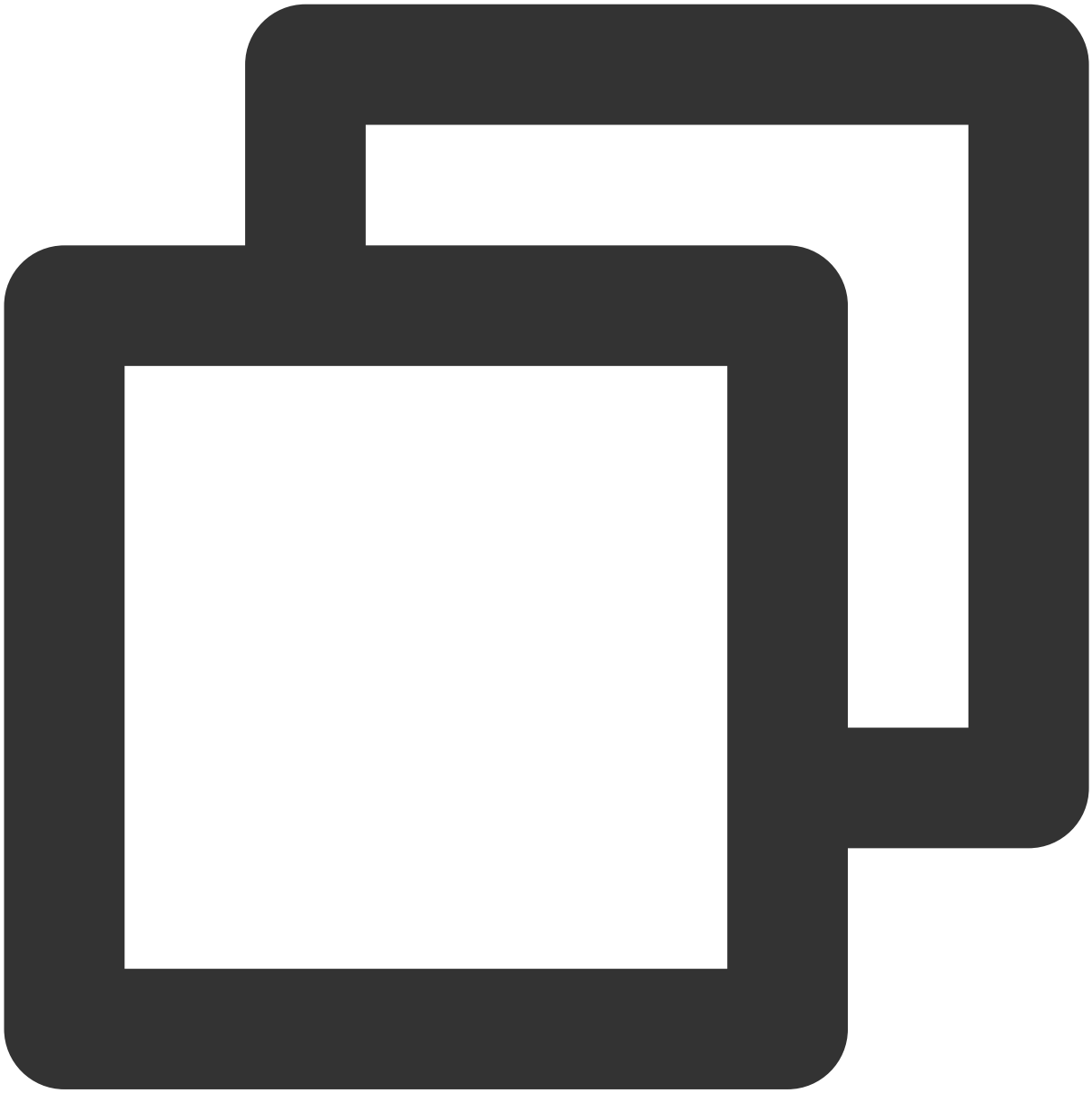
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TotalCount>1</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <MediaBucketList>
        <BucketId></BucketId>
        <Region></Region>
        <CreateTime></CreateTime>
    </MediaBucketList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	媒体 Bucket 总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
MediaBucketList	Response	媒体 Bucket 列表	Container

Container 节点 MediaBucketList 的内容：

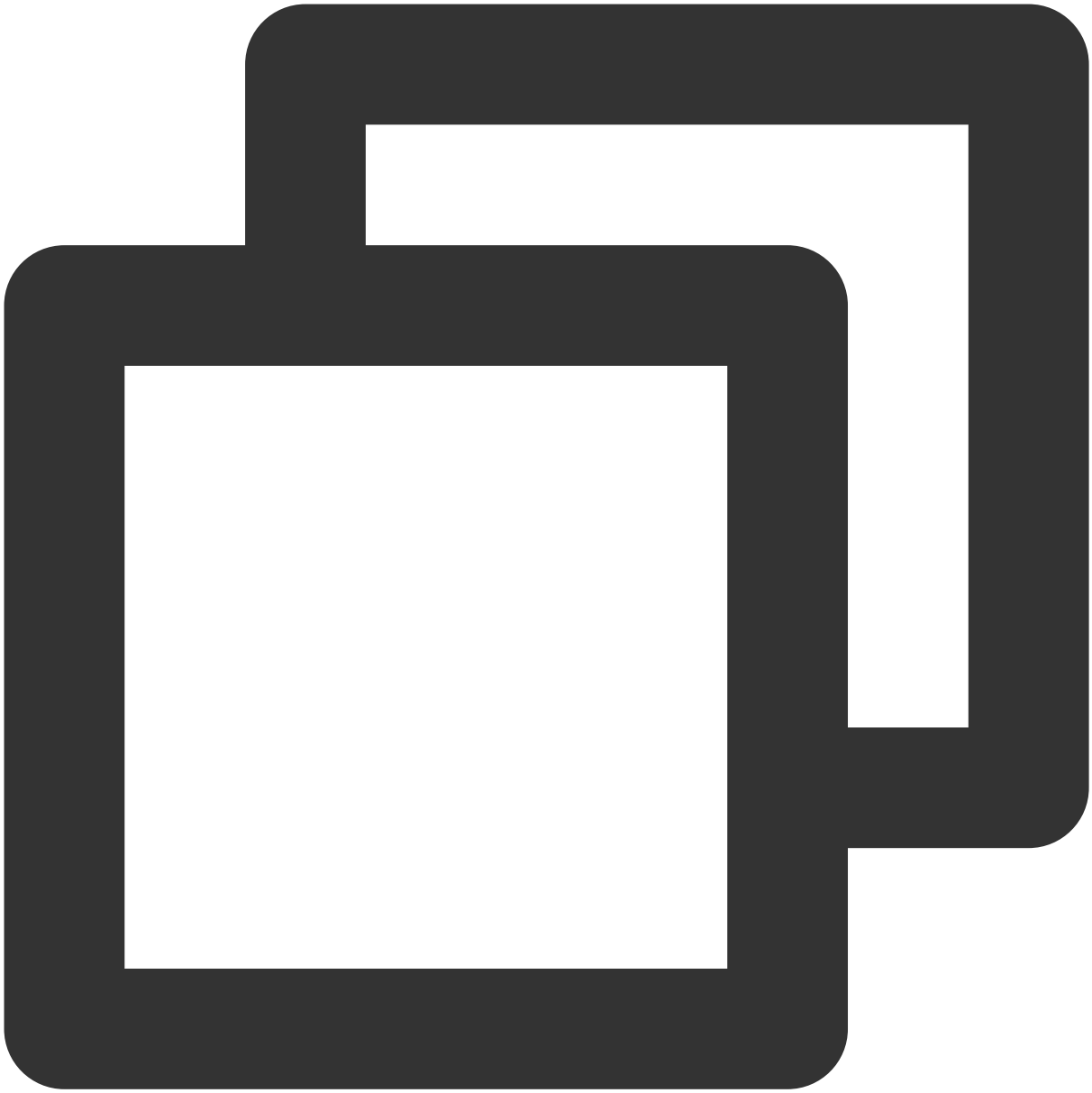
节点名称（关键字）	父节点	描述	类型
BucketId	Response.MediaBucketList	存储桶 ID	String
Name	Response.MediaBucketList	存储桶名称，同 BucketId	String
Region	Response.MediaBucketList	所在的地域	String
CreateTime	Response.MediaBucketList	创建时间	String

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

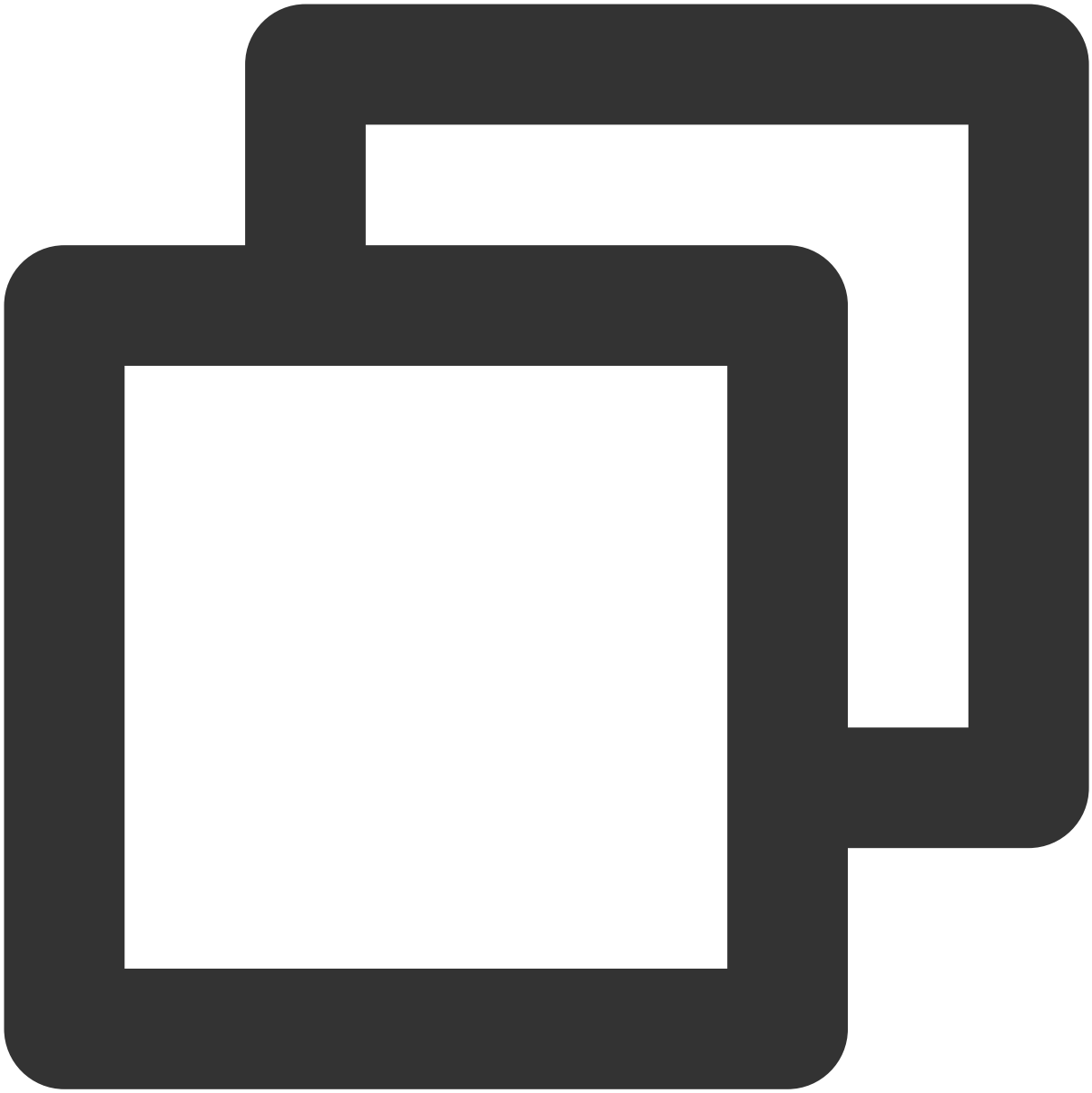
实际案例

请求



```
GET /mediabucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host: ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****



<Response>
```

```
        <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
        <TotalCount>1</TotalCount>
        <PageNumber>1</PageNumber>
        <PageSize>10</PageSize>
        <MediaBucketList>
              <BucketId></BucketId>
              <Region></Region>
              <CreateTime></CreateTime>
        </MediaBucketList>
        <MediaBucketList>
              <BucketId></BucketId>
              <Region></Region>
              <CreateTime></CreateTime>
        </MediaBucketList>
</Response>
```

开通媒体处理

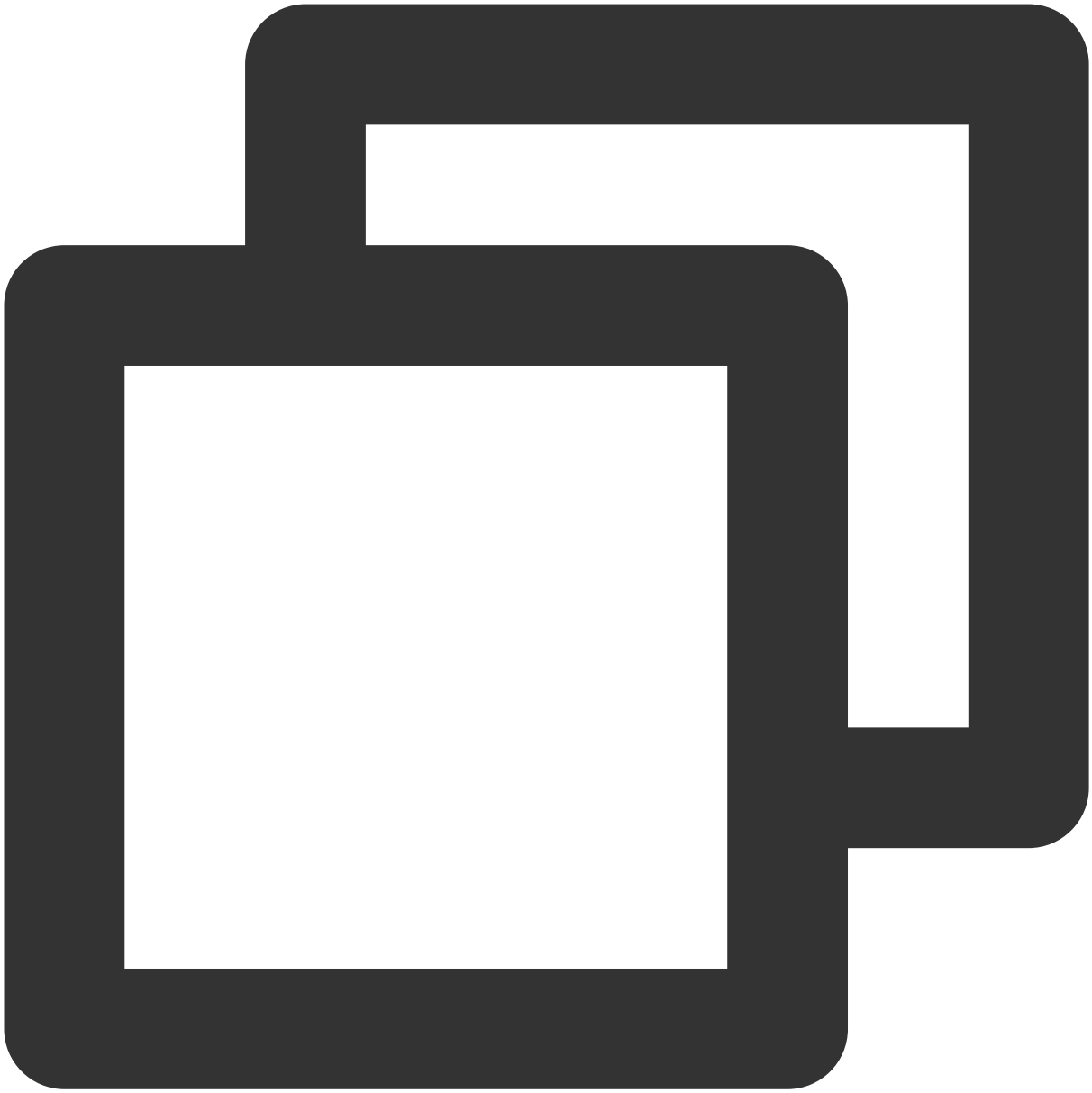
最近更新时间：2023-12-22 10:56:16

功能描述

CreateMediaBucket 接口用于绑定媒体存储桶操作（授权媒体处理服务角色访问存储桶）。

请求

请求示例



```
POST /mediabucket HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

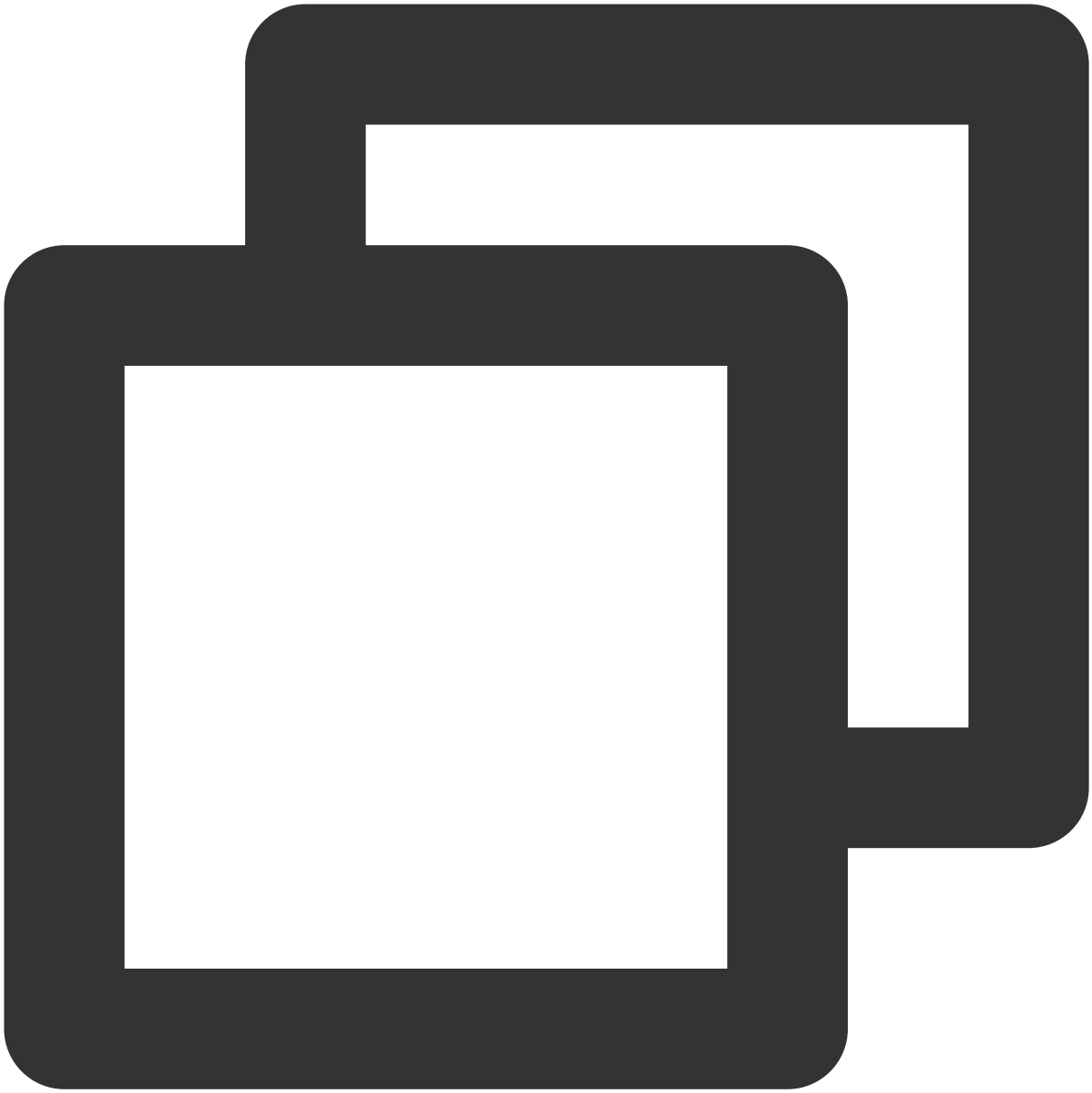
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <MediaBucket>
        <BucketId></BucketId>
        <Region></Region>
        <CreateTime></CreateTime>
    </MediaBucket>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

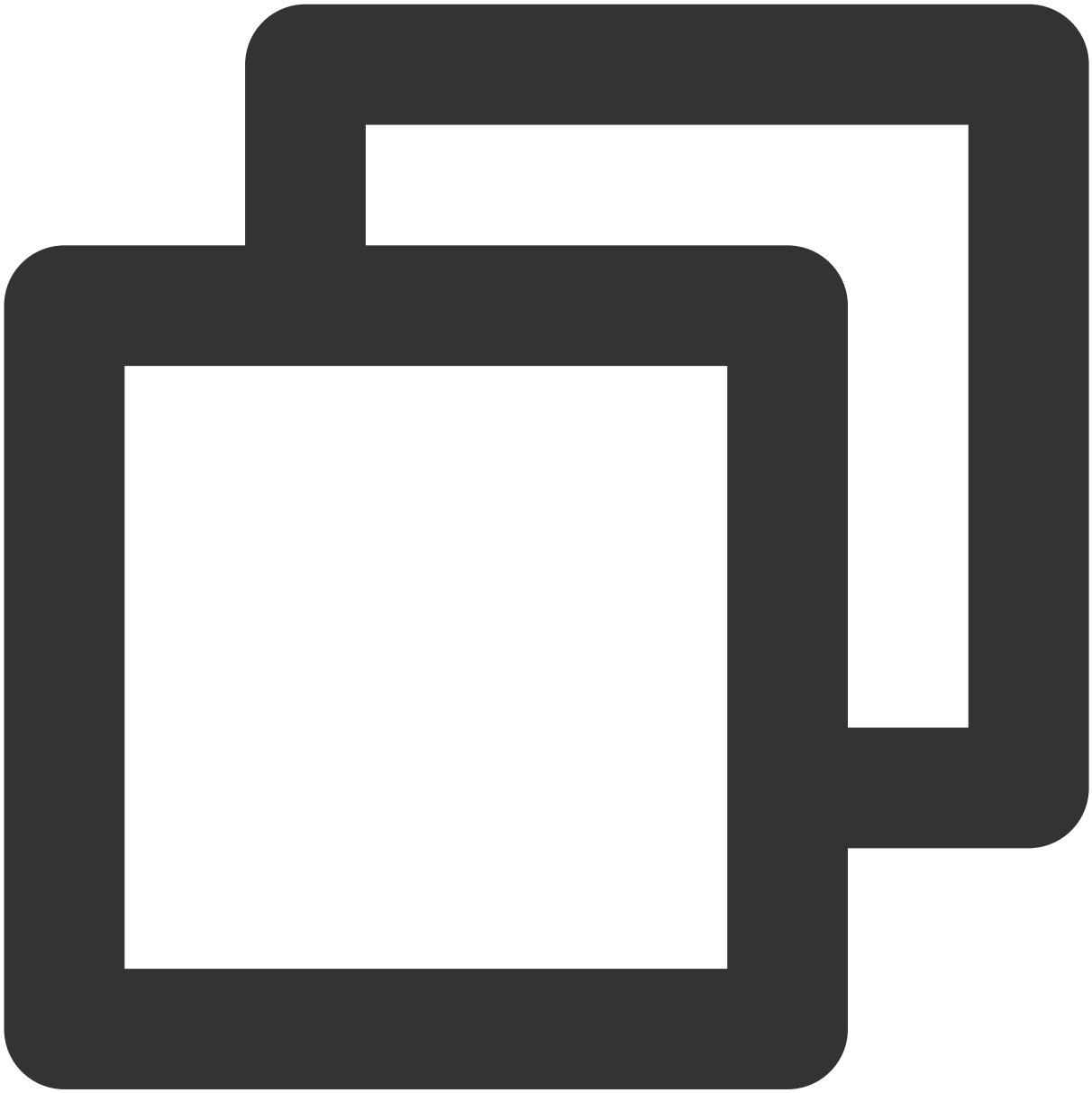
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
MediaBucket	Response	媒体 Bucket 详情，同 DescribeMediaBuckets 中的 Response.MediaBucketList	Container

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

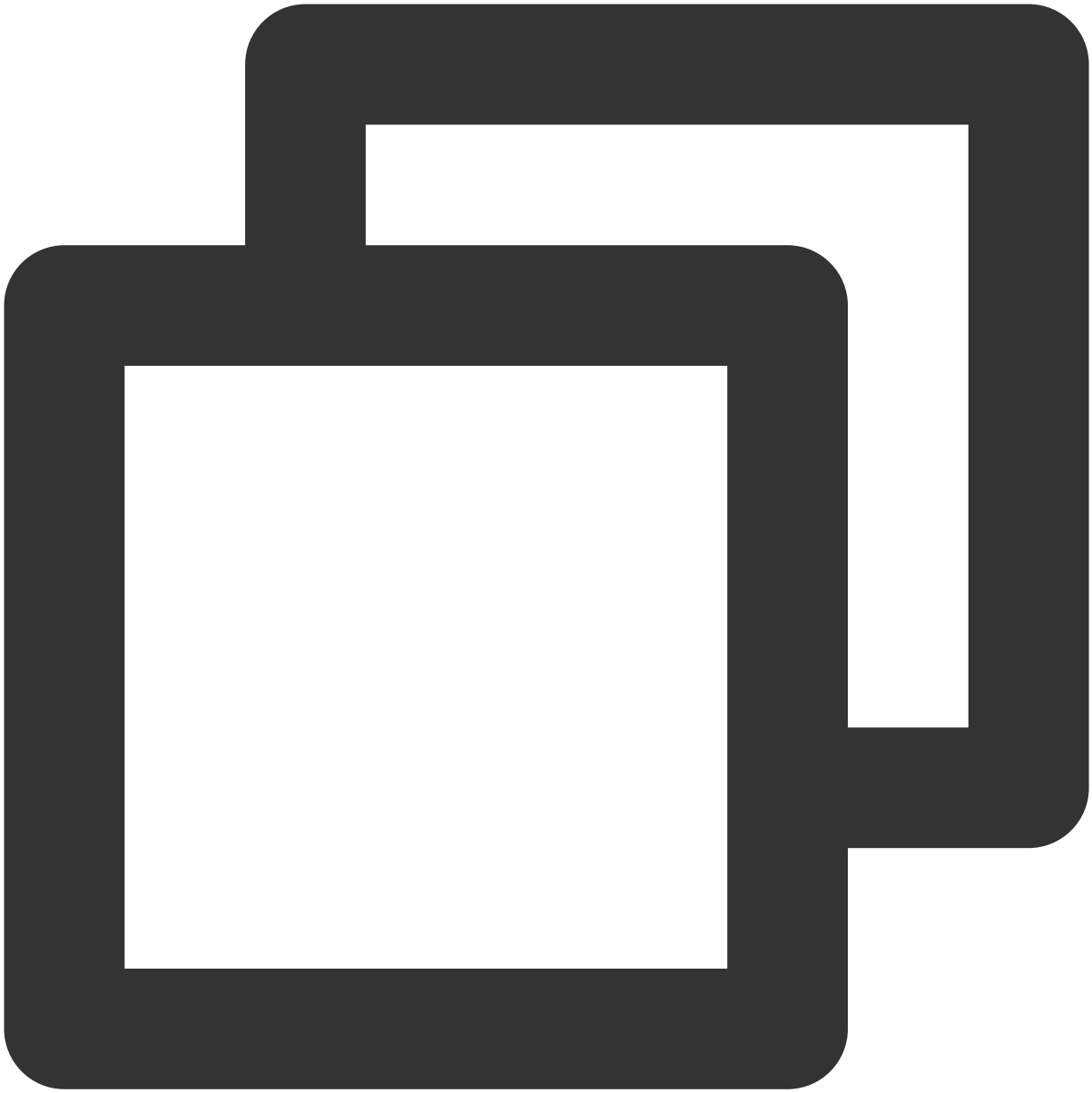
实际案例

请求



```
POST /mediabucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <MediaBucket>
```

```
        <BucketId>examplebucket-1250000000</BucketId>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </MediaBucket>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

媒体截图接口
获取媒体文件截图

最近更新时间：2023-12-22 10:55:56

功能描述

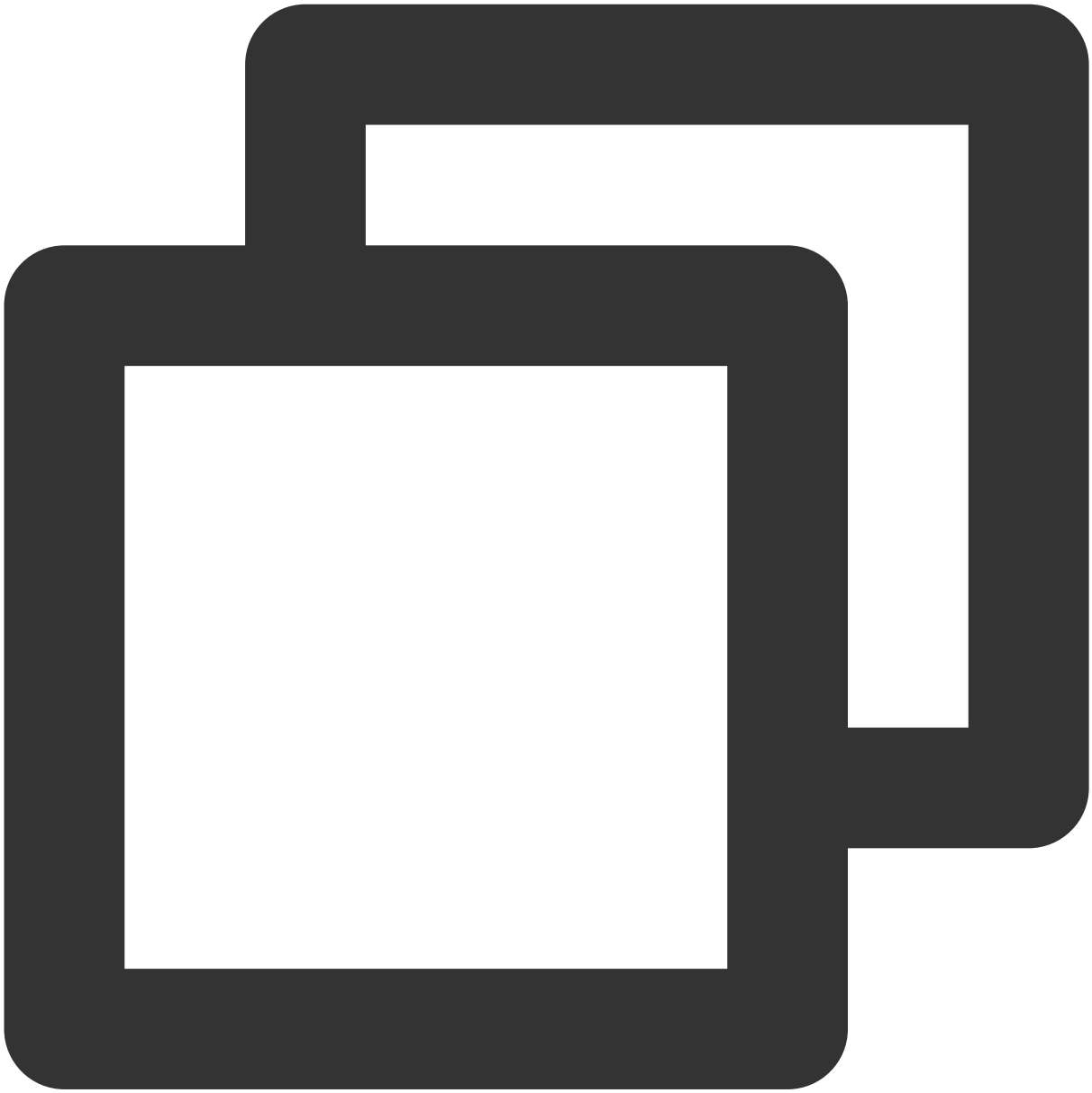
GetSnapshot 接口用于获取媒体文件某个时间的截图。

使用说明

此接口为同步模式，异步模式请参考任务接口-截图任务。

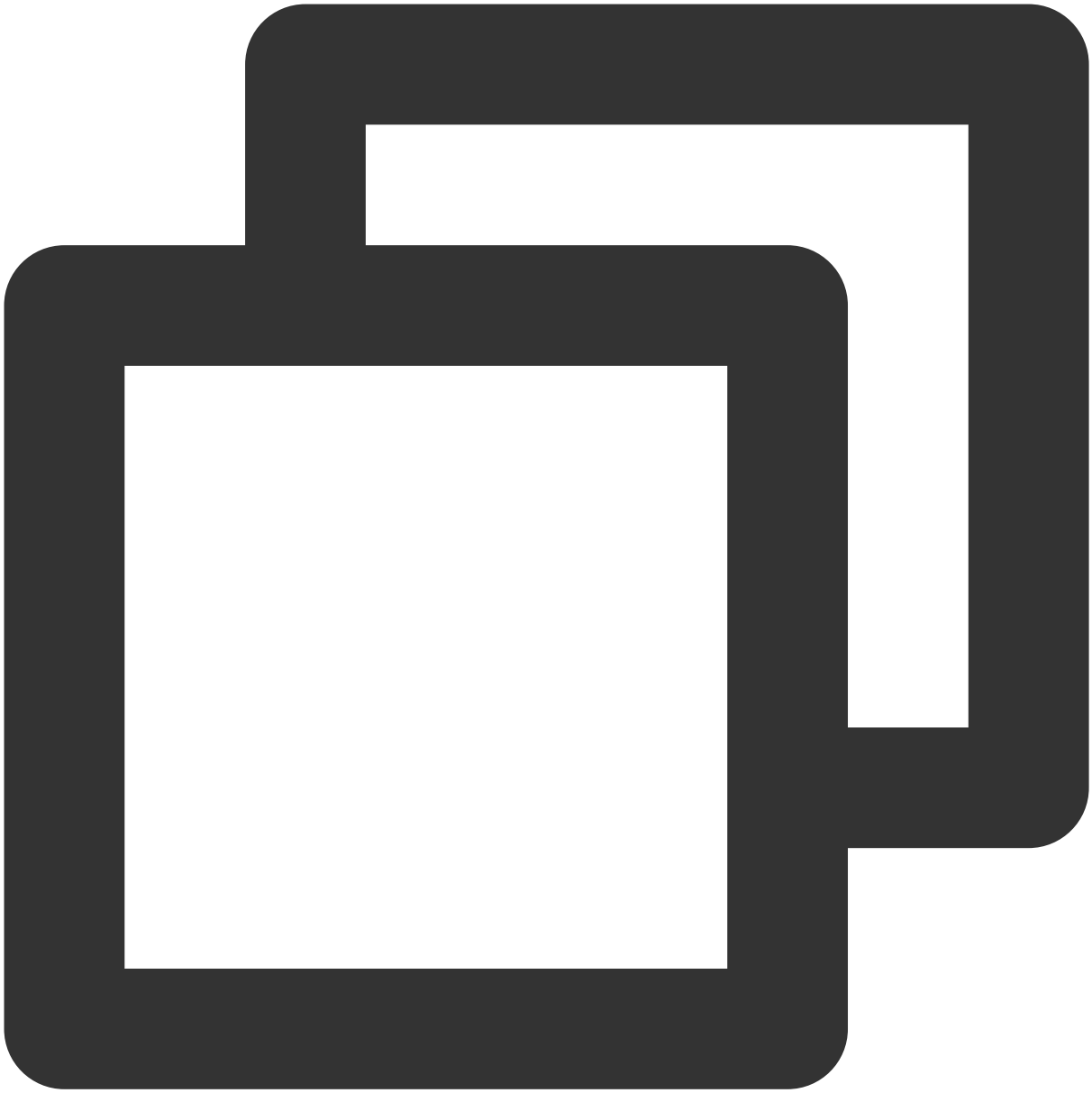
请求示例

1. 通过GET请求方式



```
GET /for-test.mp4?ci-process=snapshot&time=1&format=jpg HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
```

2. 通过URL地址添加参数方式



```
https://<BucketName-APPID>.cos.<Region>.myqcloud.com/for-test.mp4?ci-process=snapsh
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

参数说明如下：

参数名称	描述	类型	是否必选
ci-process	操作类型，固定使用 snapshot	String	是
time	截图的时间点，单位为秒	float	是
width	截图的宽。默认为0	Int	否
height	截图的高。默认为0 当 width 和 height 都为0时，表示使用视频的宽高。如果单个为0，则以另外一个值按视频宽高比例自动适应	Int	否
format	截图的格式，支持 jpg 和 png，默认 jpg	String	否
rotate	图片旋转方式 auto：按视频旋转信息进行自动旋转 off：不旋转 默认值为 auto	String	否
mode	截帧方式 keyframe：截取指定时间点之前的最近的一个关键帧 exactframe：截取指定时间点的帧 默认值为 exactframe	String	否

请求体

该请求无请求体。

响应

响应头

该响应包含公共响应头，详情请参见 公共响应头部 文档。

响应体

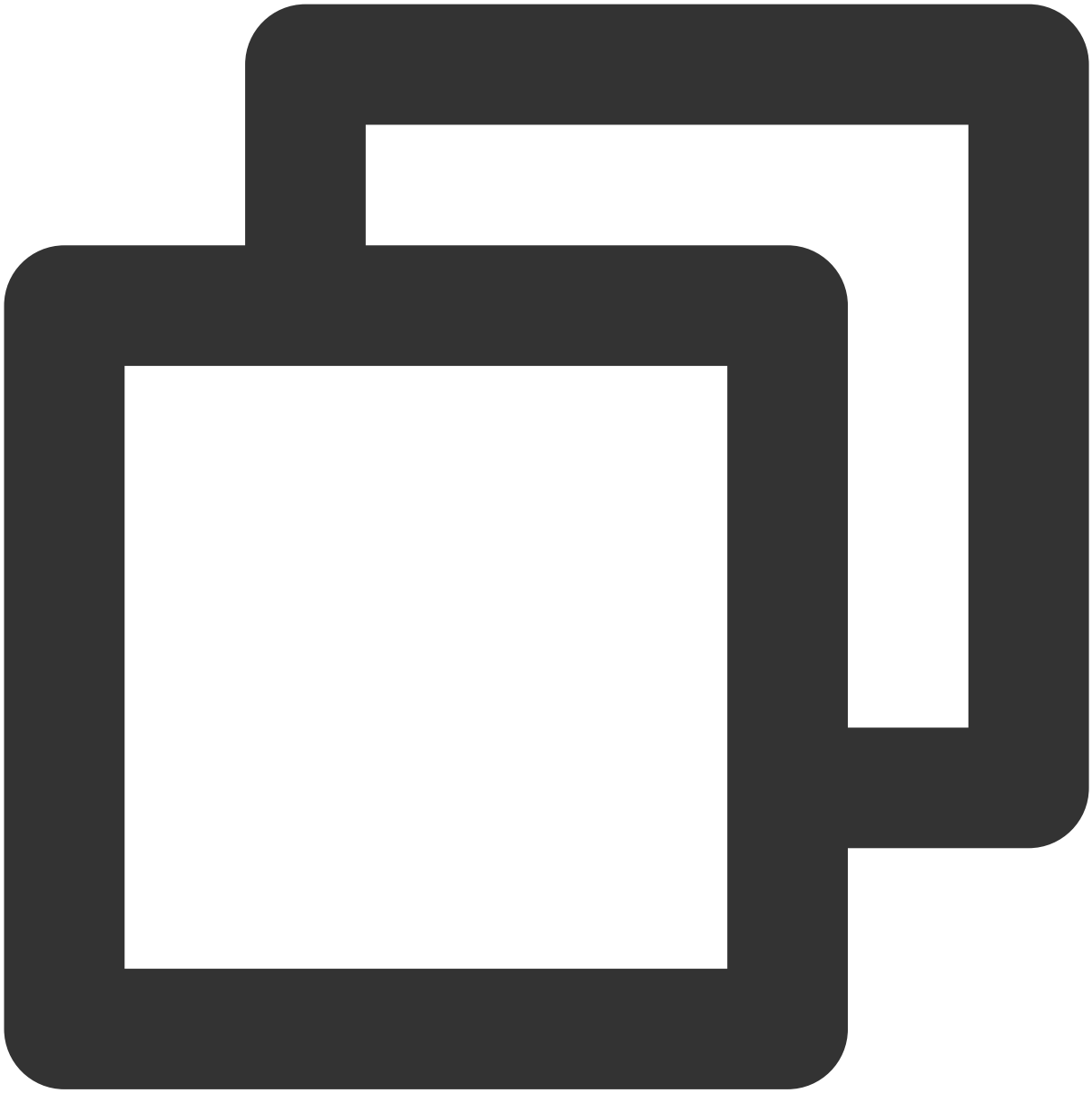
该响应体为截图文件内容。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见错误码文档。

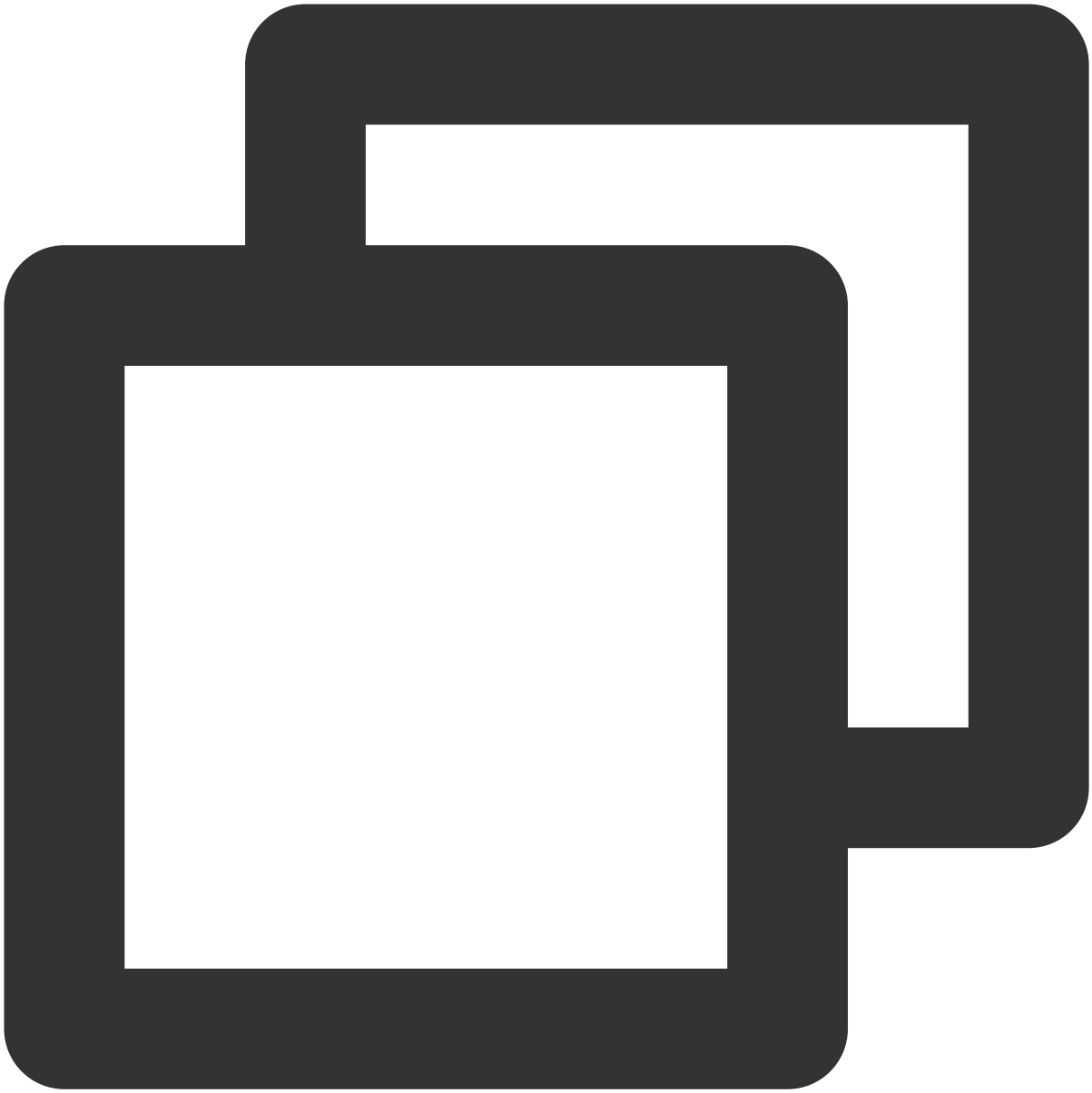
实际案例

请求1 通过GET请求方式



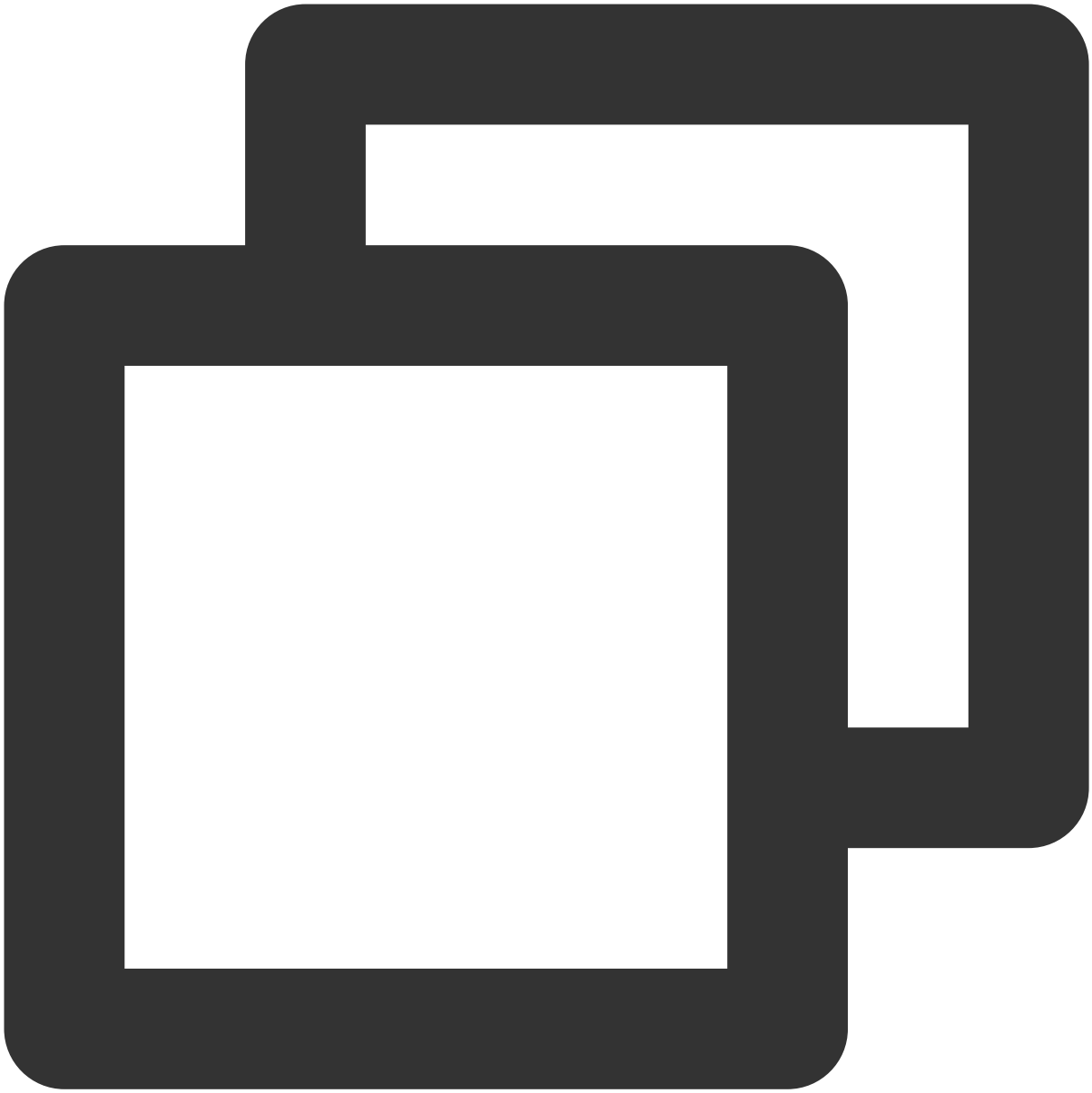
```
GET /for-test.mp4?ci-process=snapshot&time=1.5 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG****-si
Content-Length: 0
```

请求2 通过URL地址添加参数方式



```
https://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/for-test.mp4?ci-proces
```

响应



```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 266005
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==


<图片内容>
```

版权所有：腾讯云计算（北京）有限责任公司

获取媒体文件截图（万象域名）

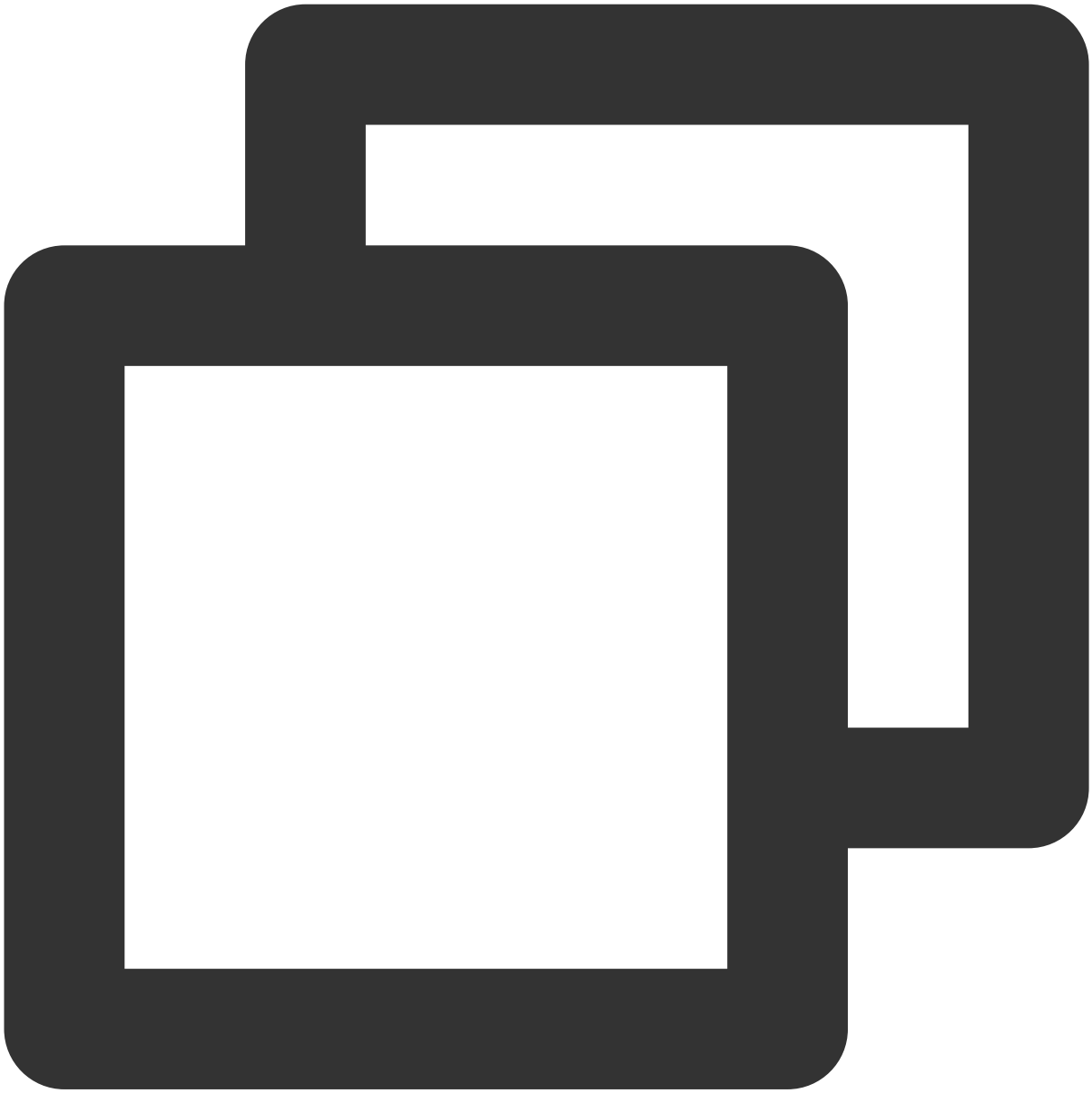
最近更新时间：2023-12-22 10:56:22

功能描述

GenerateSnapshot 接口用于获取媒体文件某个时间的截图，输出的截图统一为 jpeg 格式。

请求

请求示例



```
POST /snapshot HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-type: application/xml

<body>
```

说明

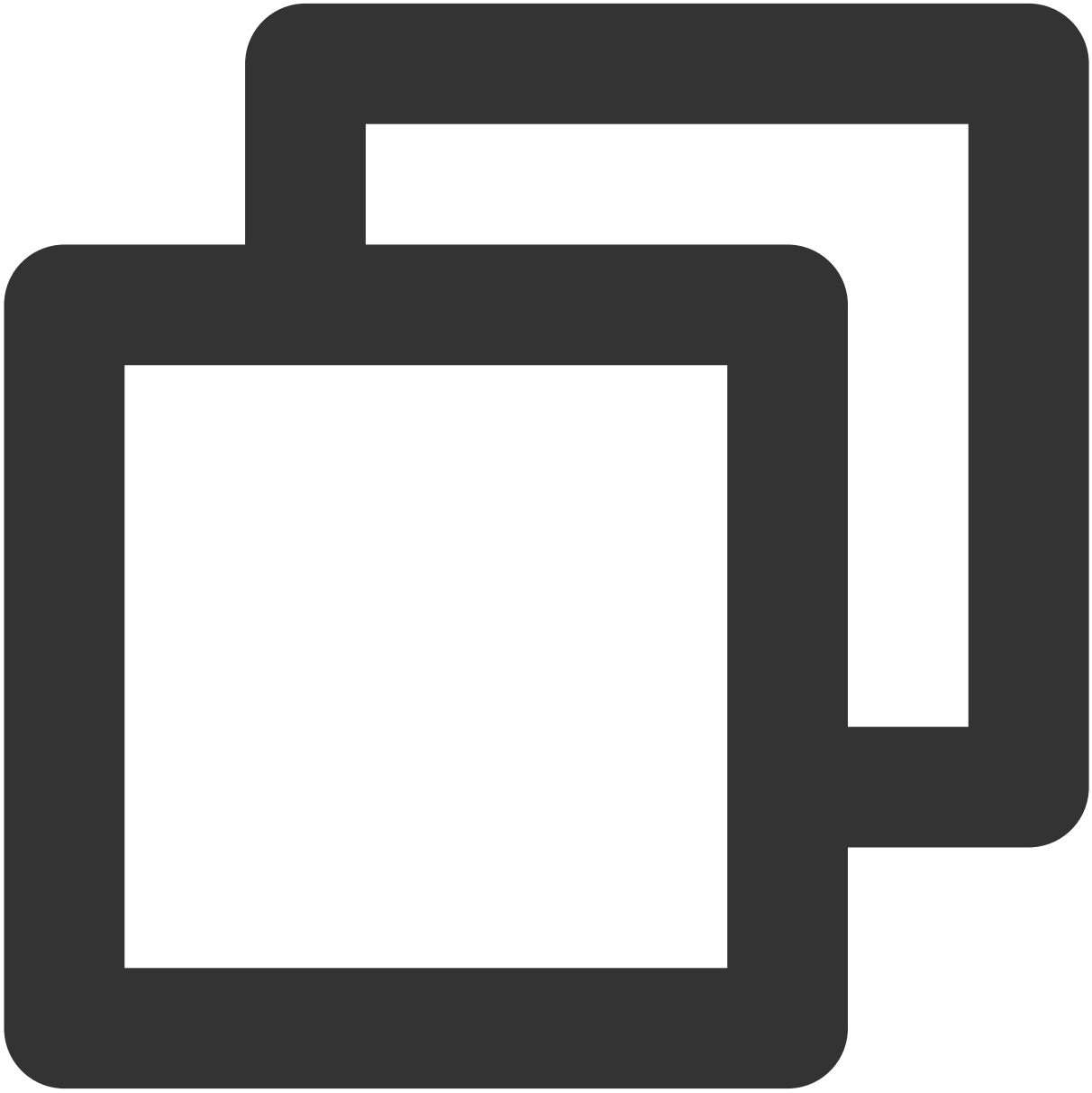
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该 API 接口请求的请求体具体节点内容为：



```
<Request>
  <Input>
```

```
    <Object></Object>
  </Input>
  <Time></Time>
  <Width></Width>
  <Height></Height>
  <Mode></Mode>
  <Rotate></Rotate>
  <Format></Format>
  <Output>
    <Region></Region>
    <Bucket></Bucket>
    <Object></Object>
  </Output>
</Request>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存 Generate Snapshot 请求的容器	Container	是

Container 节点 Request 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	媒体文件的位置信息	Container	是
Time	Request	截取哪个时间点的内容，单位为秒	Float	是
Output	Request	截图保存的位置信息	Container	是
Width	Request	截图的宽。默认为0	Int	否
Height	Request	截图的高。默认为0。 Width 和 Height 都为0时，表示使用视频的宽高。如果单个为0，则以另外一个值按视频宽高比例自动适应。	Int	否
Format	Request	截图的格式，支持 jpg 和 png，默认 jpg	String	否
Mode	Request	截帧方式： keyframe：截取指定时间点之前的最近的一个关键帧 exactframe：截取指定时间点的帧 默认值为 exactframe	String	否
Rotate	Request	图片旋转方式： auto：按视频旋转信息进行自动旋转	String	否

版权所有：腾讯云计算（北京）有限责任公司

		off：不旋转 默认值为 auto		

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	文件的名称	String	是

Container 节点 Output 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
Region	Request.Output	存储桶所在的地域	String	是
Bucket	Request.Output	文件所属的 COS 存储桶	String	是
Object	Request.Output	文件的名称	String	是

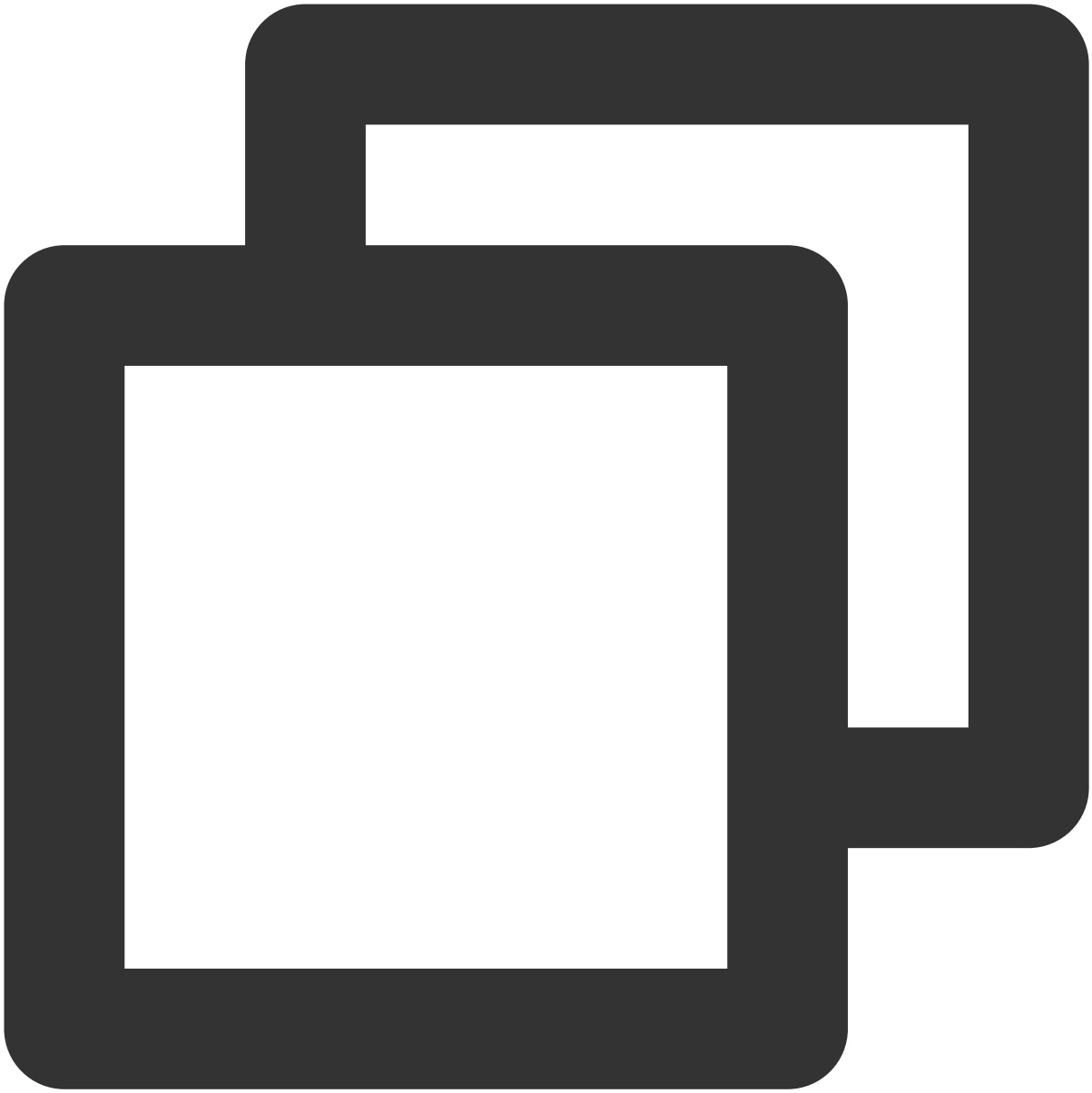
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <Output>
    <Region></Region>
    <Bucket></Bucket>
    <Object></Object>
  </Output>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
Output	Request	截图保存的位置信息	Container

Container 节点 Output 的内容：

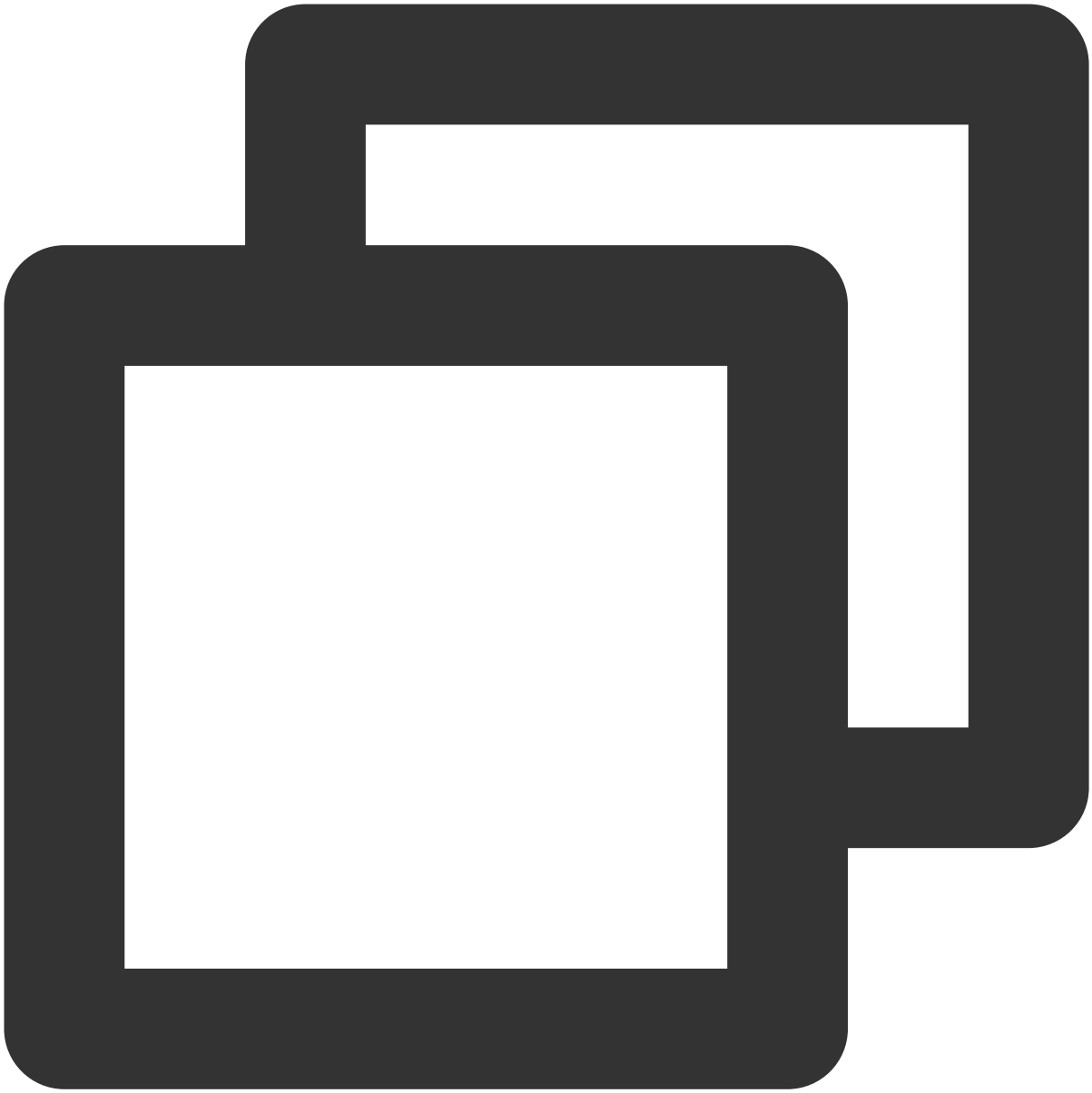
节点名称（关键字）	父节点	描述	类型
Region	Response.Output	存储桶所在的地域	String
Bucket	Response.Output	文件所属的 COS 存储桶	String
Object	Response.Output	文件的名称	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

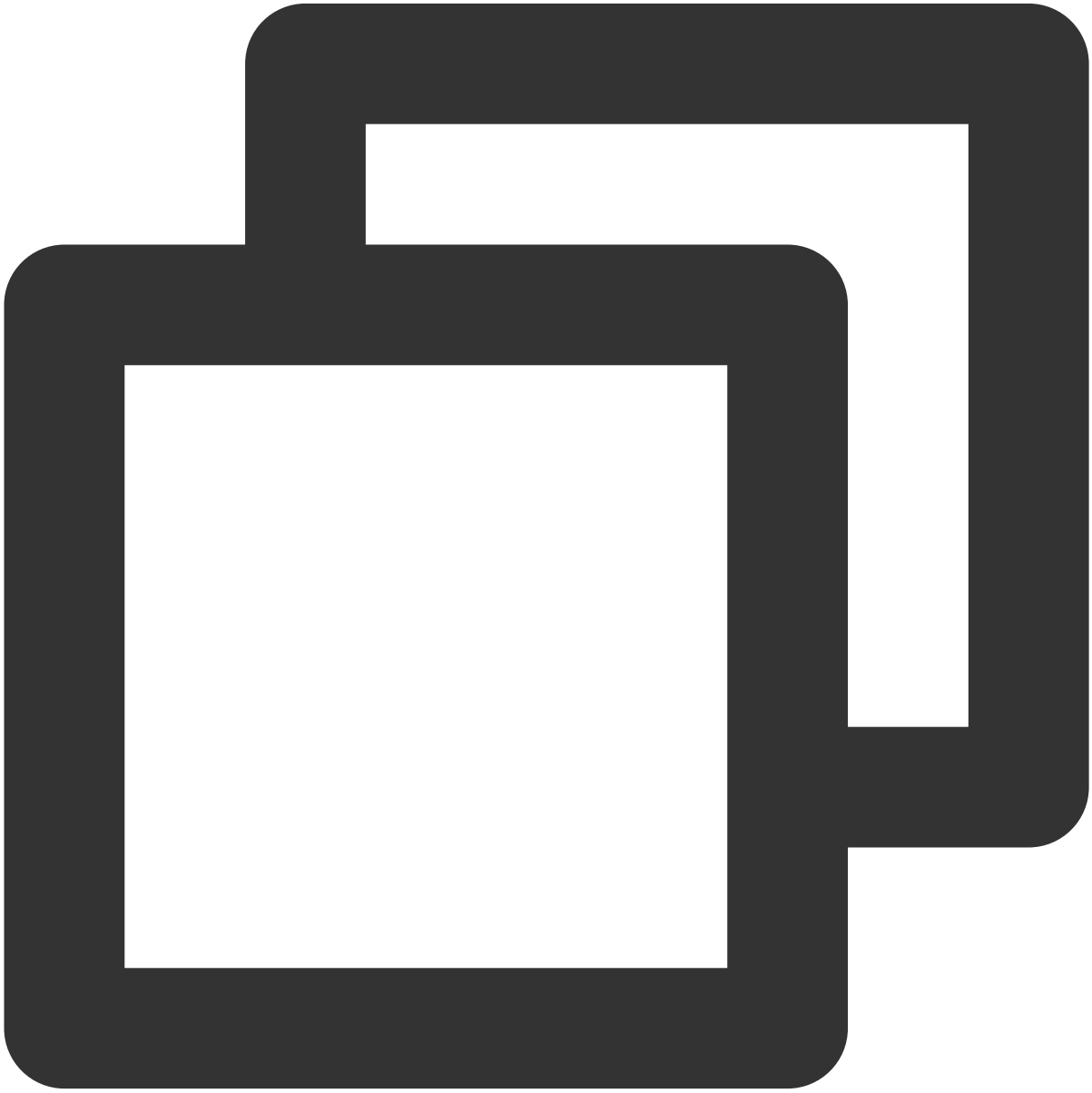
请求



```
POST /snapshot HTTP/1.1
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
Content-Length: 552
Content-Type: application/xml

<Request>
  <Input>
    <Object>video-for-test.mp4</Object>
  </Input>
```

```
    <Time>10</Time>
    <Output>
      <Region>ap-beijing</Region>
      <Bucket>ci-output-1250000000</Bucket>
      <Object>snapshot/video-for-test-snapshot.jpg</Object>
    </Output>
</Request>
```

响应



```
HTTP/1.1 200 OK
```

```
Content-Type: application/xml
Content-Length: 266
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-ci
x-ci-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZm****

<Response>
  <Output>
    <Region>ap-beijing</Region>
    <Bucket>ci-output-1250000000</Bucket>
    <Object>snapshot/video-for-test-snapshot.jpg</Object>
  </Output>
</Response>
```

媒体信息接口
获取媒体文件信息

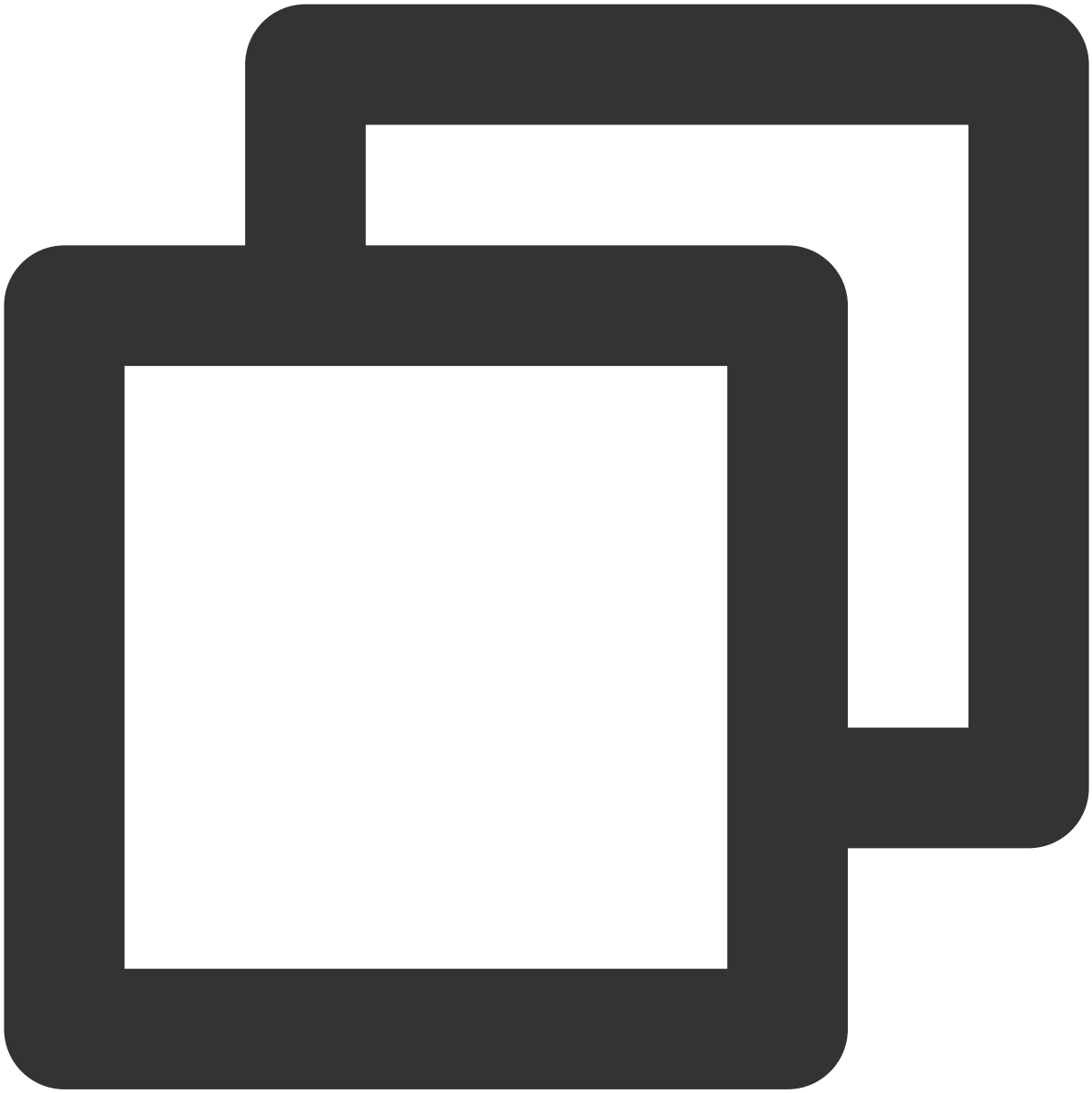
最近更新时间：2023-12-22 10:58:38

功能描述

GetMediainfo 接口用于获取媒体文件的信息。

请求

请求示例



```
GET /for-test.mp4?ci-process=videoinfo HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

参数说明如下：

节点名称（关键字）	父节点	描述	类型	是否必选
ci-process	无	操作类型，固定使用 videoinfo	String	是

请求体

该请求无请求体。

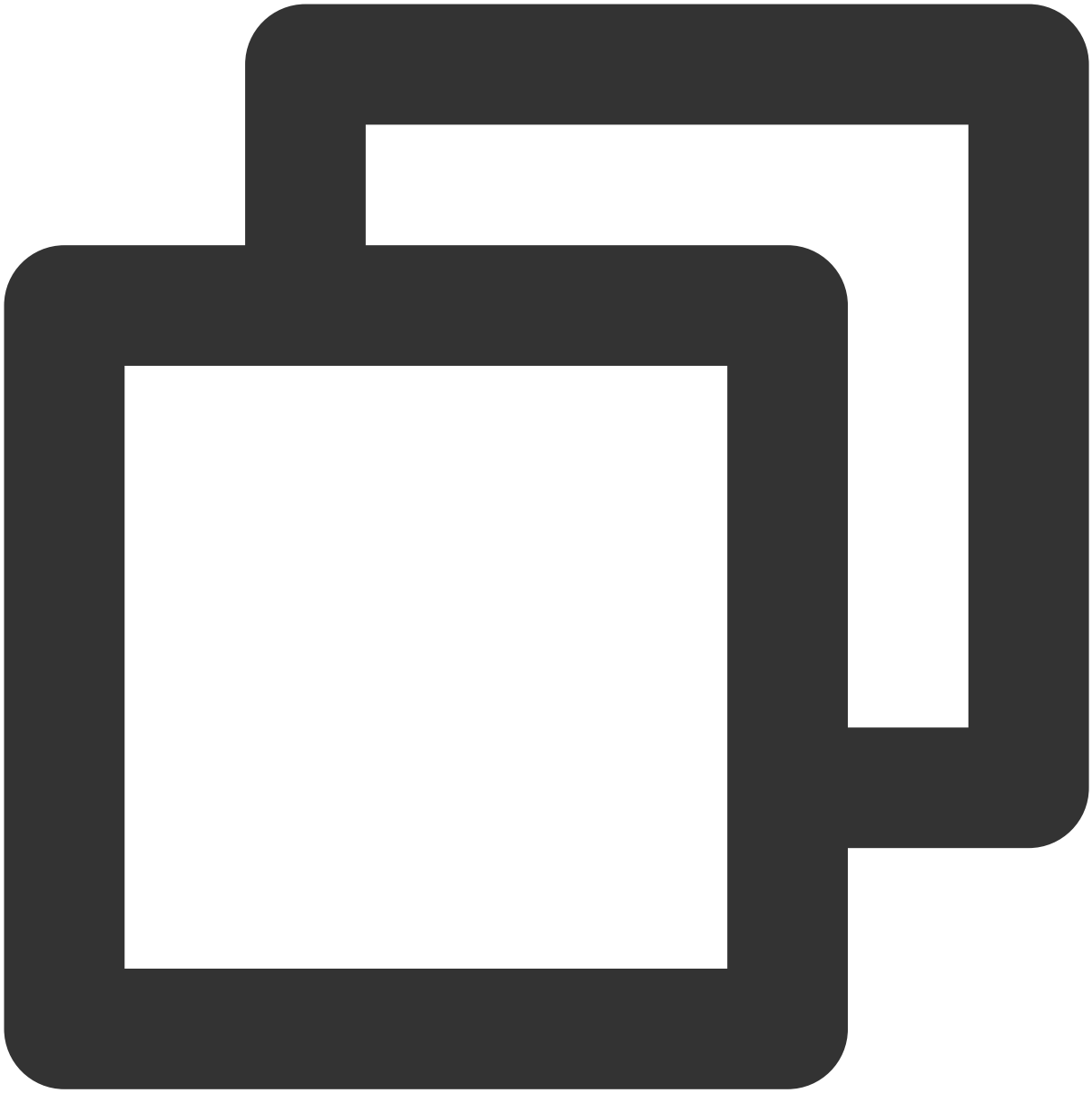
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
      <MediaInfo>
         <Stream>
            <Video>
                <Index></Index>
                <CodecName></CodecName>
                <CodecLongName></CodecLongName>
                <CodecTimeBase></CodecTimeBase>
                <CodecTagString></CodecTagString>
                <CodecTag></CodecTag>
                <ColorPrimaries></ColorPrimaries>
```

```
                <ColorRange></ColorRange>
                <ColorTransfer></ColorTransfer>
                <Profile></Profile>
                <Width></Width>
                <Height></Height>
                <HasBFrame></HasBFrame>
                <RefFrames></RefFrames>
                <Sar></Sar>
                <Dar></Dar>
                <PixFormat></PixFormat>
                <FieldOrder></FieldOrder>
                <Level></Level>
                <Fps></Fps>
                <AvgFps></AvgFps>
                <Timebase></Timebase>
                <StartTime></StartTime>
                <Duration></Duration>
                <Bitrate></Bitrate>
                <NumFrames></NumFrames>
                <Language></Language>
            </Video>
            <Audio>
                <Index></Index>
                <CodecName></CodecName>
                <CodecLongName></CodecLongName>
                <CodecTimeBase></CodecTimeBase>
                <CodecTagString></CodecTagString>
                <CodecTag></CodecTag>
                <SampleFmt></SampleFmt>
                <SampleRate></SampleRate>
                <Channel></Channel>
                <ChannelLayout></ChannelLayout>
                <Timebase></Timebase>
                <StartTime></StartTime>
                <Duration></Duration>
                <Bitrate></Bitrate>
                <Language></Language>
            </Audio>
            <Subtitle>
                <Index></Index>
                <Language></Language>
            </Subtitle>
        </Stream>
        <Format>
            <NumStream></NumStream>
            <NumProgram></NumProgram>
            <FormatName></FormatName>
```

```
            <FormatLongName></FormatLongName>
            <StartTime></StartTime>
            <Duration></Duration>
            <Bitrate></Bitrate>
            <Size></Size>
         </Format>
      </MediaInfo>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
MediaInfo	Response	媒体的详细信息	Container

Container 节点 MediaInfo 的内容：

节点名称（关键字）	父节点	描述	类型
Stream	Response.MediaInfo	流信息	Container
Format	Response.MediaInfo	格式信息	Container

Container 节点 Stream 的内容：

节点名称（关键字）	父节点	描述	类型
Video	Response.MediaInfo.Stream	视频信息	Container
Audio	Response.MediaInfo.Stream	音频信息	Container
Subtitle	Response.MediaInfo.Stream	字幕信息	Container

Container 节点 Format 的内容（查询视频信息时，可能部分字段未返回）：

节点名称（关键字）	父节点	描述	类型
NumStream	Response.MediaInfo.Format	Stream（包含 Video、Audio、Subtitle）的数量	Int

版权所有：腾讯云计算（北京）有限责任公司

NumProgram	Response.MediaInfo.Format	节目的数量	Int
FormatName	Response.MediaInfo.Format	容器格式名字	String
FormatLongName	Response.MediaInfo.Format	容器格式的详细名字	String
StartTime	Response.MediaInfo.Format	起始时间，单位为秒	Float
Duration	Response.MediaInfo.Format	时长，单位为秒	Float
Bitrate	Response.MediaInfo.Format	比特率，单位为 kbps	Int
Size	Response.MediaInfo.Format	大小，单位为 Byte	Int

Container 节点 Video 的内容（查询视频信息时，可能部分字段未返回）：

节点名称（关键字）	父节点	描述	类型
Index	Response.MediaInfo.Stream.Video	该流的编号	Int
CodecName	Response.MediaInfo.Stream.Video	编解码格式名字	String
CodecLongName	Response.MediaInfo.Stream.Video	编解码格式的详细名字	String
CodecTimeBase	Response.MediaInfo.Stream.Video	编码时基	String
CodecTagString	Response.MediaInfo.Stream.Video	编码标签名	String
CodecTag	Response.MediaInfo.Stream.Video	编码标签	String
ColorPrimaries	Response.MediaInfo.Stream.Video	色原	String
ColorRange	Response.MediaInfo.Stream.Video	色彩范围	String
ColorTransfer	Response.MediaInfo.Stream.Video	色彩通道	String
Profile	Response.MediaInfo.Stream.Video	视频编码档位	String
Height	Response.MediaInfo.Stream.Video	视频高，单位 px	Int
Width	Response.MediaInfo.Stream.Video	视频宽，单位 px	Int
HasBFrame	Response.MediaInfo.Stream.Video	是否有B帧。1表示有，0表示无	Int
RefFrames	Response.MediaInfo.Stream.Video	视频编码的参考帧个数	Int
Sar	Response.MediaInfo.Stream.Video	采样宽高比	String

版权所有：腾讯云计算（北京）有限责任公司

Dar	Response.MediaInfo.Stream.Video	显示宽高比	String
PixFormat	Response.MediaInfo.Stream.Video	像素格式	String
FieldOrder	Response.MediaInfo.Stream.Video	场的顺序	String
Level	Response.MediaInfo.Stream.Video	视频编码等级	Int
Fps	Response.MediaInfo.Stream.Video	视频帧率	Float
AvgFps	Response.MediaInfo.Stream.Video	平均帧率	String
Timebase	Response.MediaInfo.Stream.Video	时基	String
StartTime	Response.MediaInfo.Stream.Video	视频开始时间，单位为秒	Float
Duration	Response.MediaInfo.Stream.Video	视频时长，单位为秒	Float
Bitrate	Response.MediaInfo.Stream.Video	比特率，单位为 kbps	Float
NumFrames	Response.MediaInfo.Stream.Video	总帧数	Int
Language	Response.MediaInfo.Stream.Video	语言	String

Container 节点 Audio 的内容（查询视频信息时，可能部分字段未返回）：

节点名称（关键字）	父节点	描述	类型
Index	Response.MediaInfo.Stream.Audio	该流的编号	Int
CodecName	Response.MediaInfo.Stream.Audio	编解码格式名字	String
CodecLongName	Response.MediaInfo.Stream.Audio	编解码格式的详细名字	String
CodecTimeBase	Response.MediaInfo.Stream.Audio	编码时基	String
CodecTagString	Response.MediaInfo.Stream.Audio	编码标签名	String
CodecTag	Response.MediaInfo.Stream.Audio	编码标签	String
SampleFmt	Response.MediaInfo.Stream.Audio	采样格式	String
SampleRate	Response.MediaInfo.Stream.Audio	采样率	Int
Channel	Response.MediaInfo.Stream.Audio	通道数量	Int
ChannelLayout	Response.MediaInfo.Stream.Audio	通道格式	String

版权所有：腾讯云计算（北京）有限责任公司

Timebase	Response.MediaInfo.Stream.Audio	时基	String
StartTime	Response.MediaInfo.Stream.Audio	音频开始时间，单位秒	Float
Duration	Response.MediaInfo.Stream.Audio	音频时长，单位秒	Float
Bitrate	Response.MediaInfo.Stream.Audio	比特率，单位 kbps	Float
Language	Response.MediaInfo.Stream.Audio	语言	String

Container 节点 Subtitle 的内容：

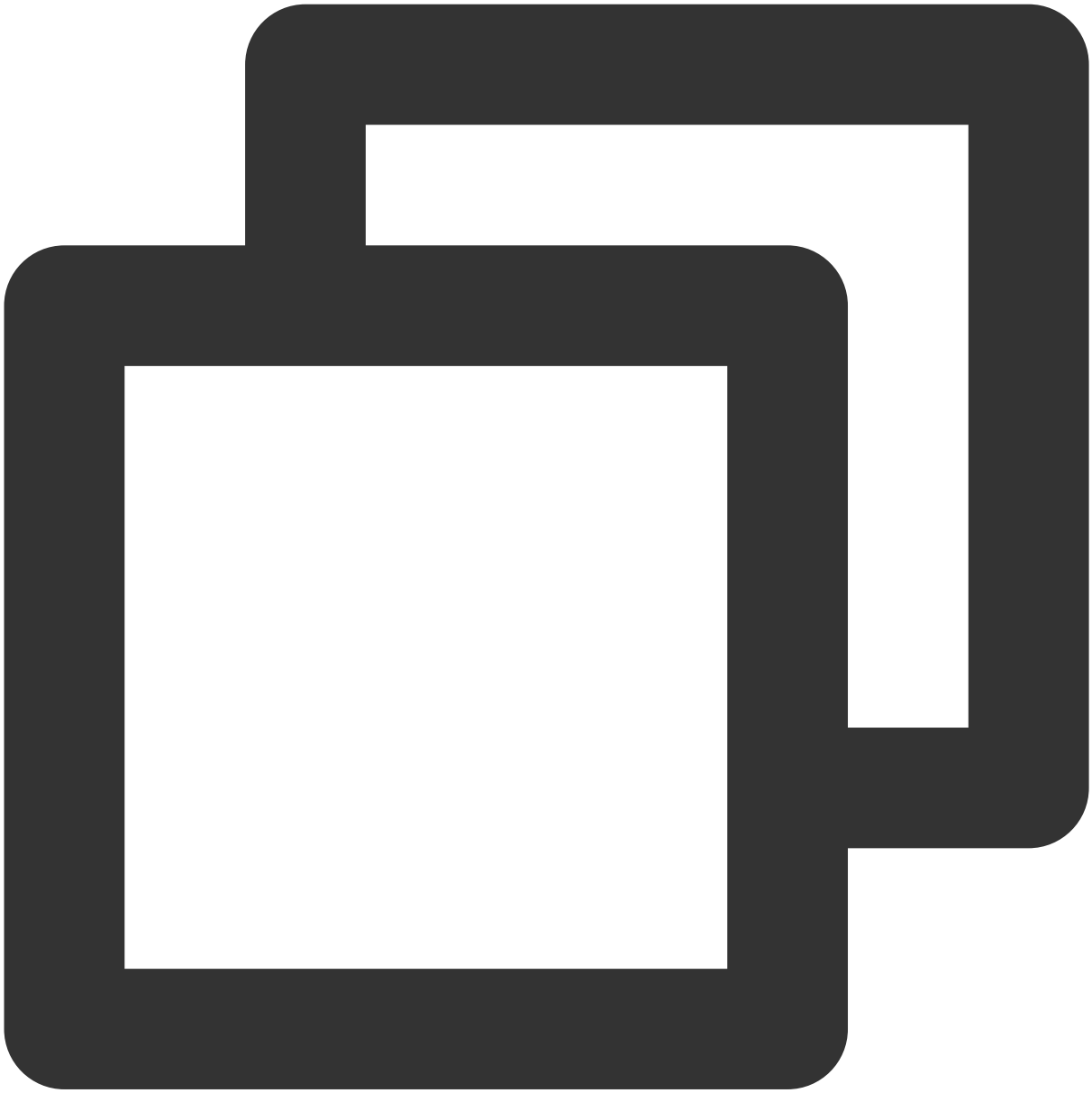
节点名称（关键字）	父节点	描述	类型
Index	Response.MediaInfo.Stream.Subtitle	该流的编号	Int
Language	Response.MediaInfo.Stream.Subtitle	语言，und 表示无查询结果	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见错误码 文档。

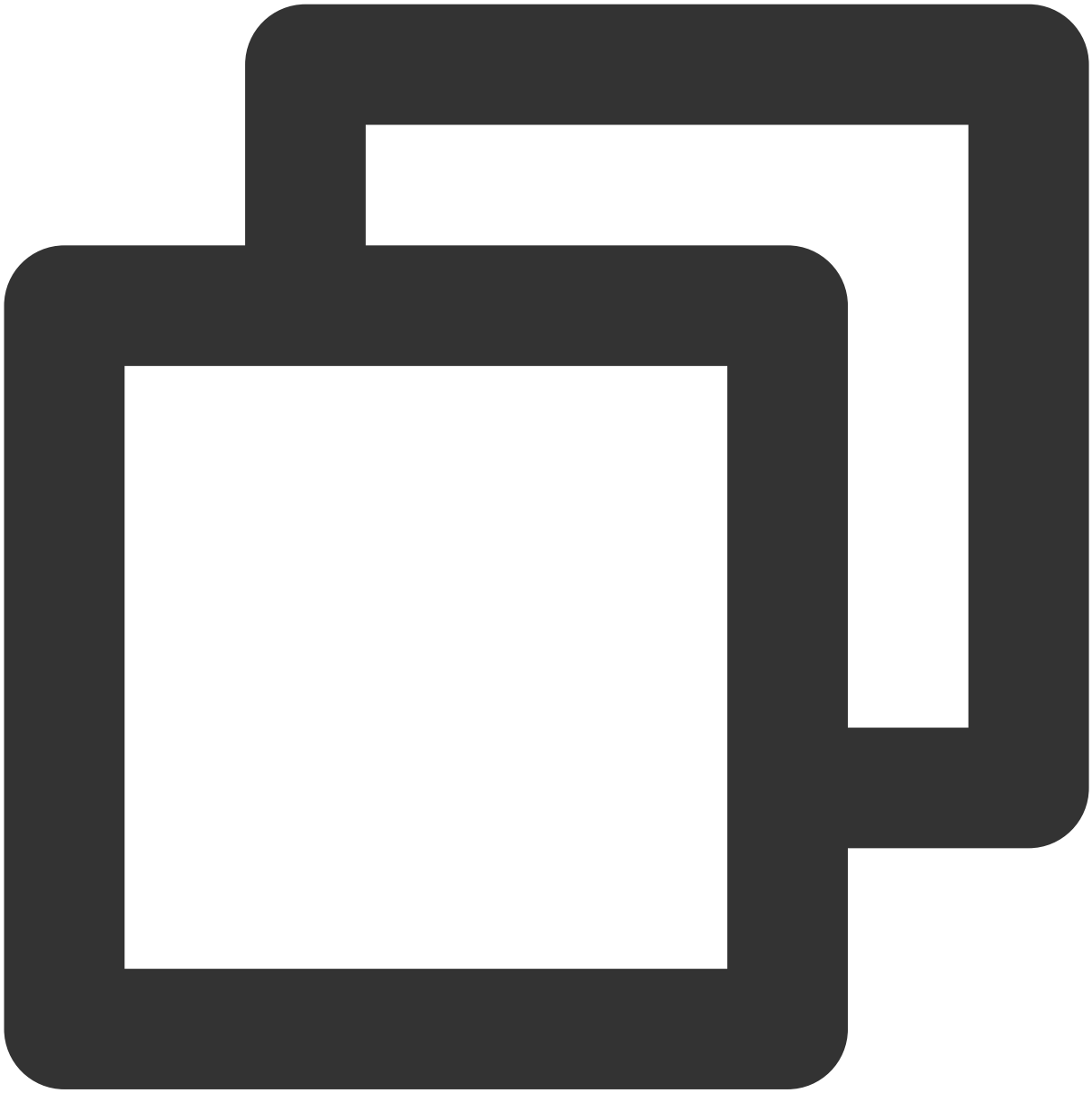
实际案例

请求



```
GET /for-test.mp4?ci-process=videoinfo HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG****-si
Content-Length: 0
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==


<Response>
```

```
        <MediaInfo>
                <Format>
                        <Bitrate>1014.950000</Bitrate>
                        <Duration>10.125000</Duration>
                        <FormatLongName>QuickTime / MOV</FormatLongName>
                        <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                        <NumProgram>0</NumProgram>
                        <NumStream>2</NumStream>
                        <Size>1284547</Size>
                        <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                        <Audio>
                                <Bitrate>70.451000</Bitrate>
                                <Channel>1</Channel>
                                <ChannelLayout>mono</ChannelLayout>
                                <CodecLongName>AAC (Advanced Audio Coding)</CodecLo
                                <CodecName>aac</CodecName>
                                <CodecTag>0x6134706d</CodecTag>
                                <CodecTagString>mp4a</CodecTagString>
                                <CodecTimeBase>1/44100</CodecTimeBase>
                                <Duration>0.440294</Duration>
                                <Index>1</Index>
                                <Language>und</Language>
                                <SampleFmt>fltp</SampleFmt>
                                <SampleRate>44100</SampleRate>
                                <StartTime>0.000000</StartTime>
                                <Timebase>1/44100</Timebase>
                        </Audio>
                        <Subtitle/>
                        <Video>
                                <AvgFps>24/1</AvgFps>
                                <Bitrate>938.164000</Bitrate>
                                <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 pa
                                <CodecName>h264</CodecName>
                                <CodecTag>0x31637661</CodecTag>
                                <CodecTagString>avc1</CodecTagString>
                                <CodecTimeBase>1/12288</CodecTimeBase>
                                <ColorPrimaries>unknown</ColorPrimaries>
                                <ColorRange>unknown</ColorRange>
                                <ColorTransfer>unknown</ColorTransfer>
                                <Dar>40:53</Dar>
                                <Duration>0.124416</Duration>
                                <Fps>24.500000</Fps>
                                <HasBFrame>2</HasBFrame>
                                <Height>1280</Height>
                                <Index>0</Index>
```

```
                                <Language>und</Language>
                                <Level>32</Level>
                                <NumFrames>243</NumFrames>
                                <PixFormat>yuv420p</PixFormat>
                                <Profile>High</Profile>
                                <RefFrames>1</RefFrames>
                                <Sar>25600:25599</Sar>
                                <StartTime>0.000000</StartTime>
                                <Timebase>1/12288</Timebase>
                                <Width>966</Width>
                        </Video>
                </Stream>
        </MediaInfo>
</Response>
```

私有 M3U8 接口
获取私有M3U8

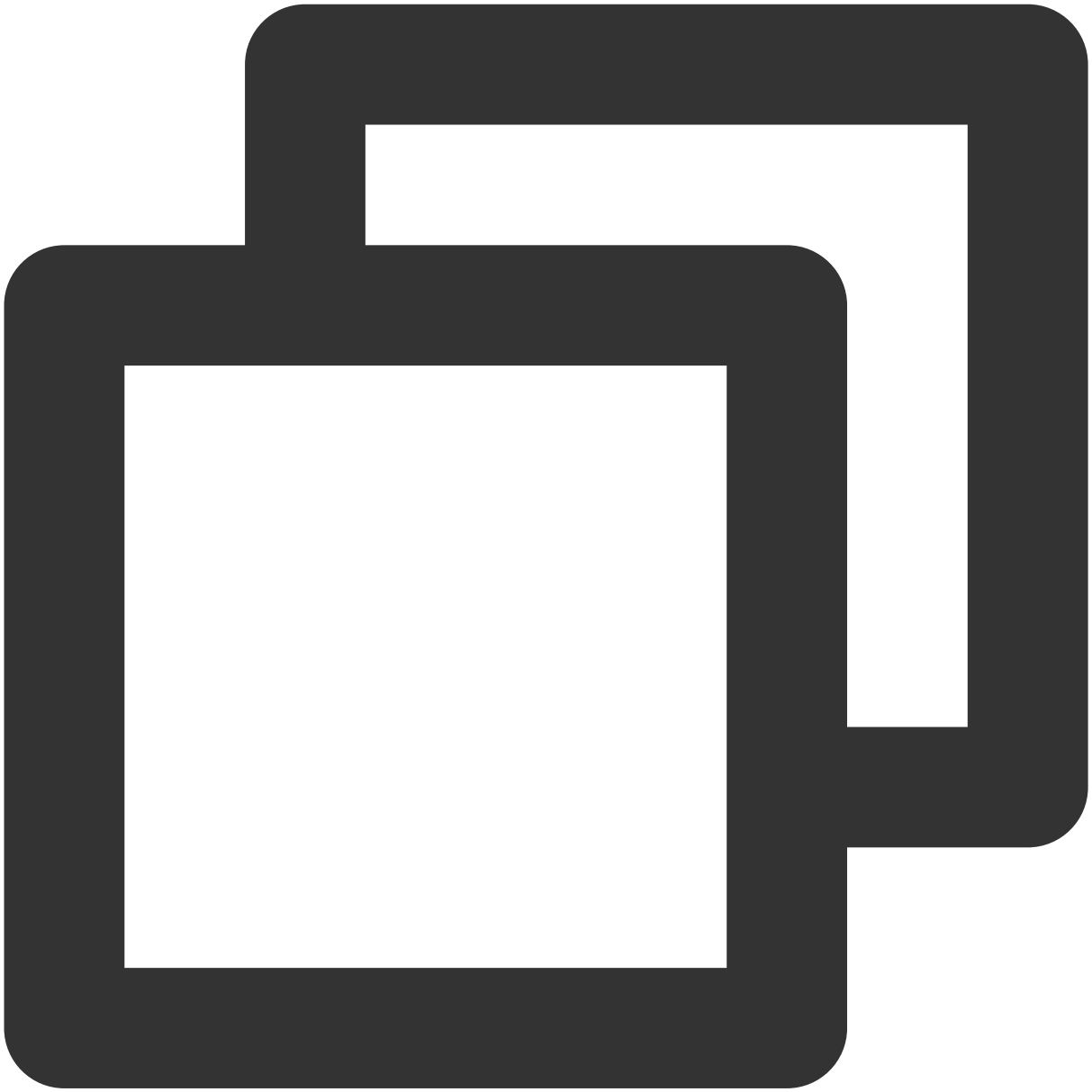
最近更新时间：2024-06-19 16:08:06

功能描述

GetPrivateM3U8 接口用于获取私有 M3U8 ts 资源的下载授权。（此方式通过对象存储转发请求至数据万象）。

请求

请求示例



```
GET /for-test.m3u8?ci-process=pm3u8&expires= HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

请求参数

参数的具体内容如下：

参数名称	描述	类型	是否必选
ci-process	操作类型，固定使用 pm3u8	String	是
expires	私有 ts 资源 url 下载凭证的相对有效期，单位为秒，范围为[3600, 43200]	String	是

请求头

此接口仅使用公共请求头部，详情请参见公共请求头部文档。

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见公共响应头部文档。

响应体

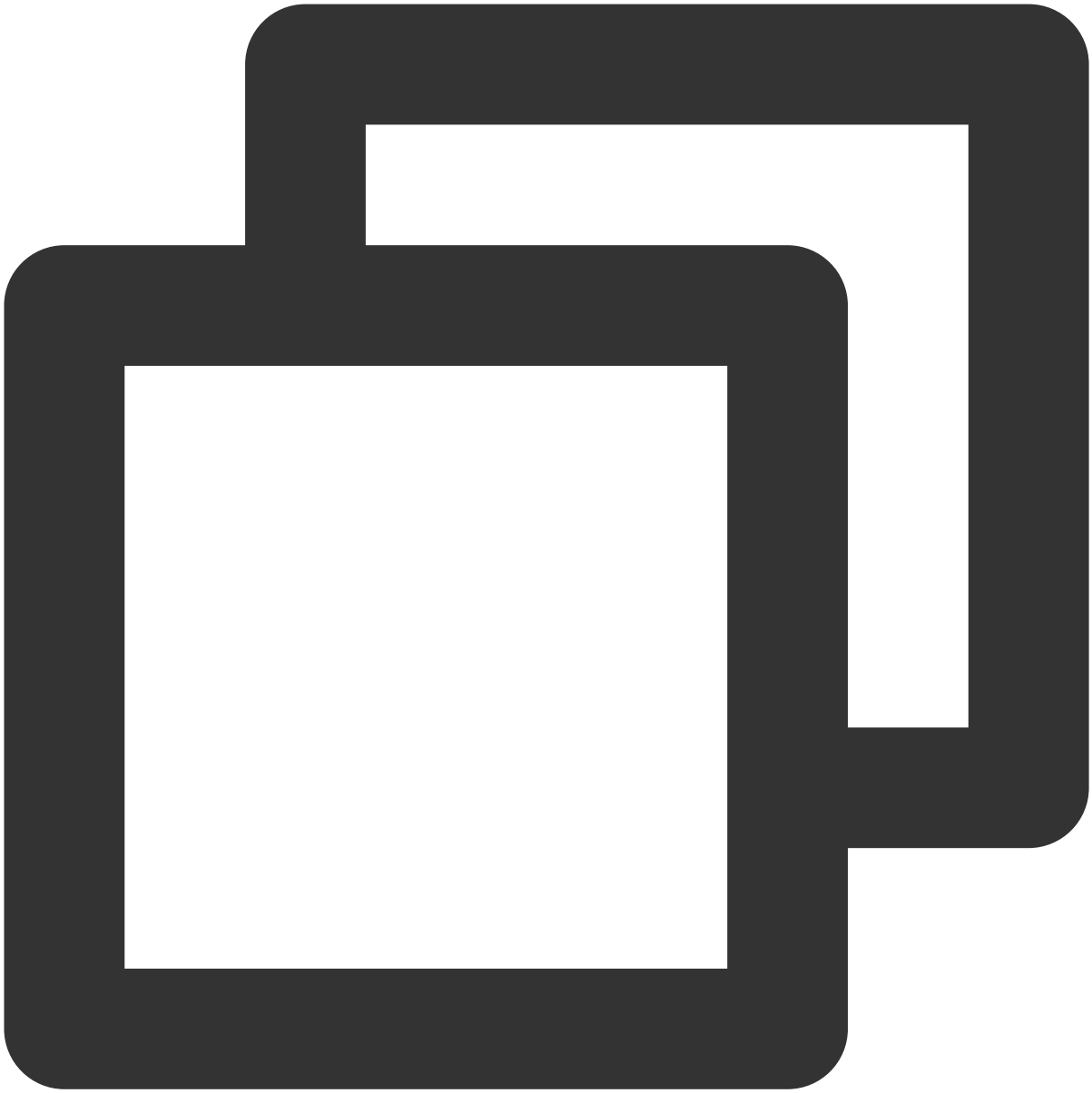
该响应体为截图文件内容。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见错误码文档。

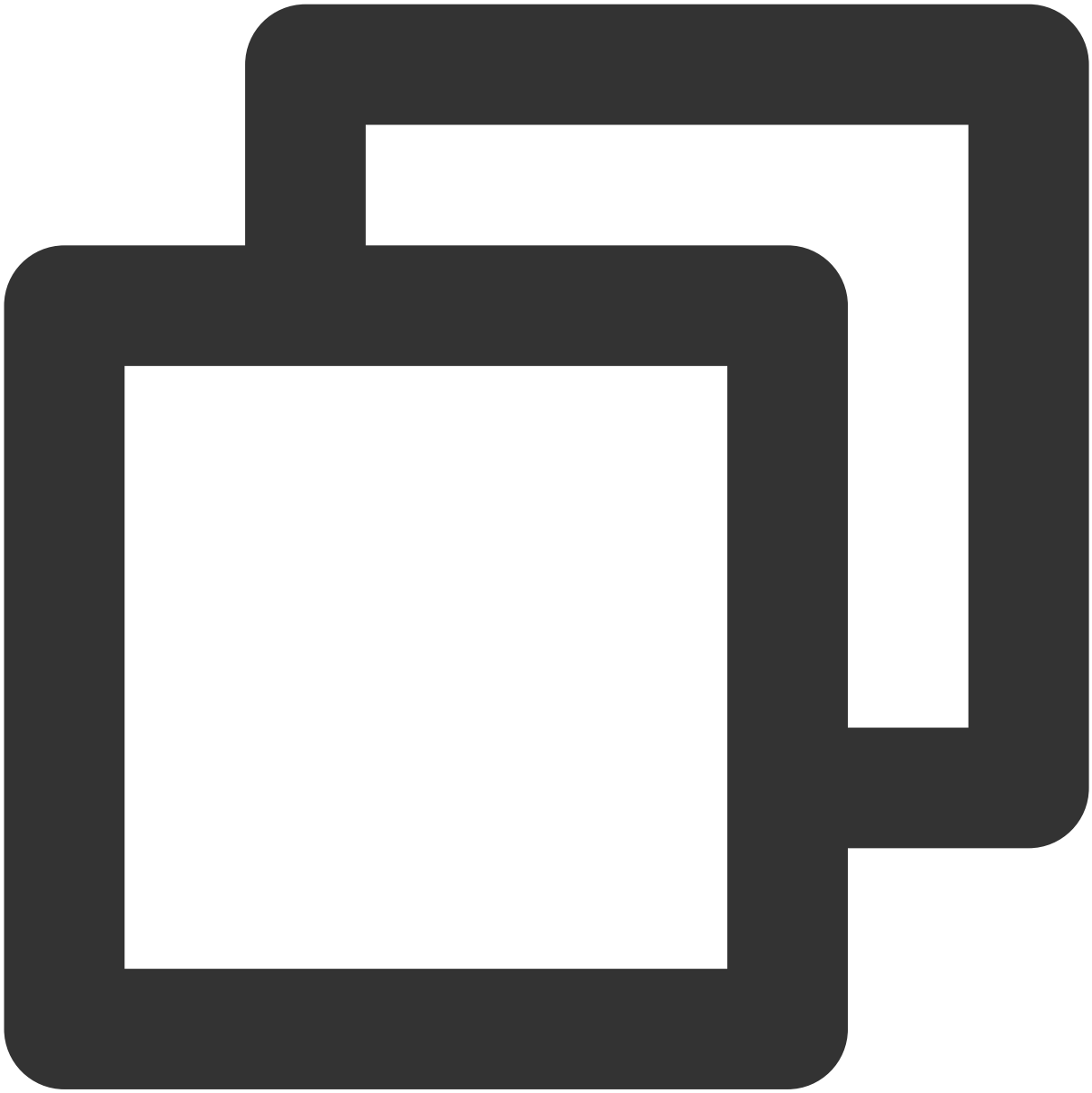
实际案例

请求



```
GET /for-test.pm3u8?ci-process=pm3u8&expires=3600 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG****-si
Content-Length: 0
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/x-mpegURL
Content-Length: 266005
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZm****

<M3U8文件内容>
```

处理方式接口

最近更新时间：2023-12-22 10:59:18

任务接口

任务是一个抽象的概念，媒体处理包含多种类型的任务，如转码任务、截图任务、媒体信息任务等。
在一个任务中，包含3个关键信息：输入、输出和参数。输入和输出设置任务执行时的输入文件以及执行后的目标文件，参数则用来设置执行具体功能的详细配置。
任务包含同步模式和异步模式：
同步任务：用户侧同步提交待处理任务。
异步任务：任务按提交顺序和优先级进入媒体处理队列中排队。

功能		任务接口
基础处理	音视频转码	提交音视频转码任务
	极速高清转码	提交极速高清任务
	视频转动图	提交视频转动图任务
	视频截帧	提交视频截帧任务
	音视频转封装	提交转封装任务
	音视频拼接	提交视频拼接任务
	获取媒体信息	提交获取媒体信息任务
画质增强	超分辨率	提交超分辨率任务
	视频增强	提交视频增强任务
	SDRtoHDR	提交SDRtoHDR任务
	视频质量评分	提交质量评分任务
版权保护	数字水印	提交添加数字水印任务
智能编辑	视频标签	提交视频标签任务
	精彩集锦	提交精彩集锦任务
	智能封面	提交智能封面任务
智能语音	音频降噪	提交音频降噪任务

版权所有：腾讯云计算（北京）有限责任公司

	人声分离	提交人声分离任务
	语音合成	提交语音合成任务
	语音识别	提交语音识别任务

工作流接口

媒体处理中的工作流是指通过配置多个单任务操作组合对指定的上传文件进行自动化处理的过程，工作流调用请参考 工作流接口。

批量任务处理接口

批量任务处理是对指定的存量文件进行单任务处理或按照工作流中配置的操作自动化处理，调用请参考 批量数据处理接口。

文档处理
查询文档预览开通状态

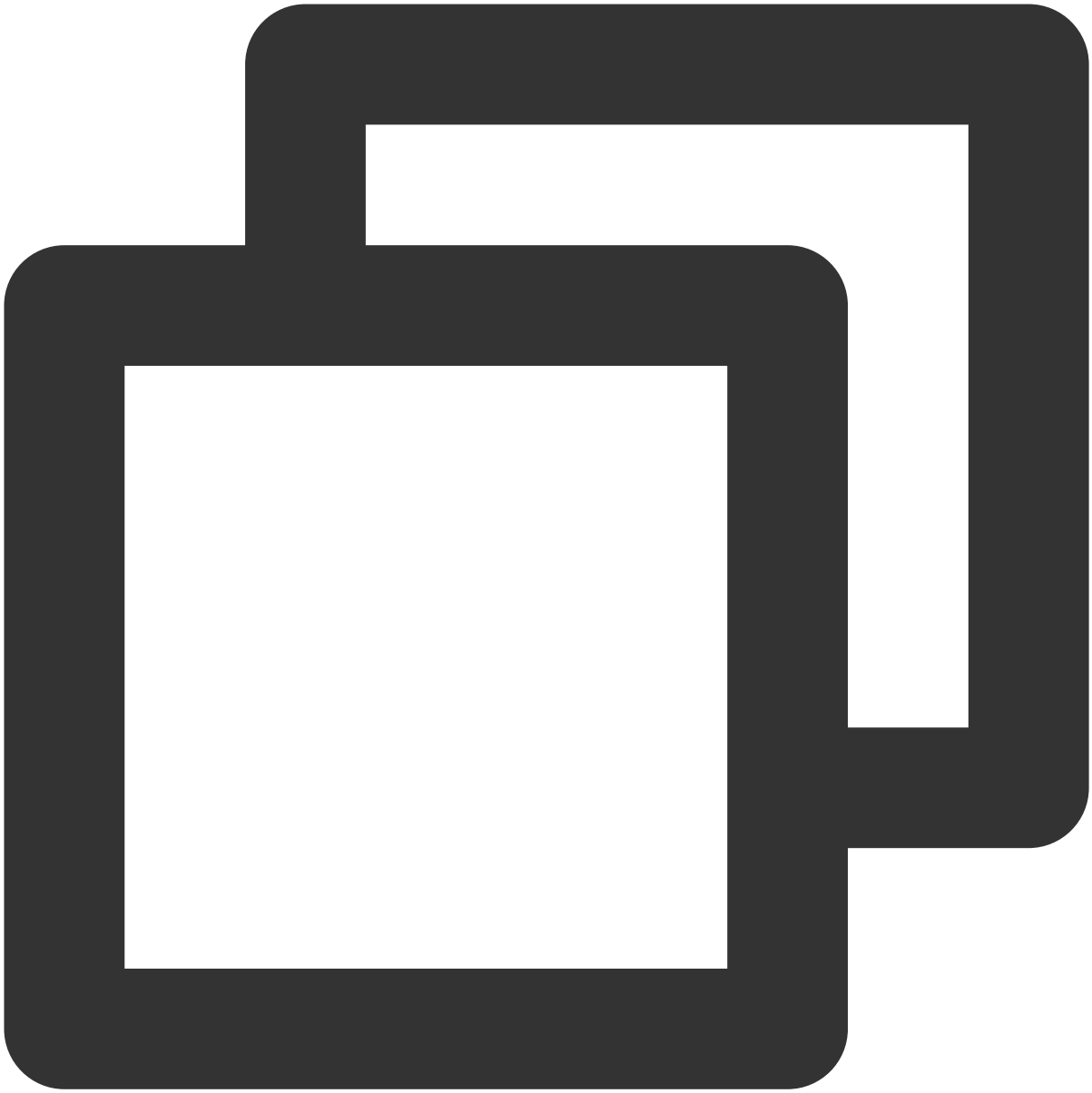
最近更新时间：2023-12-22 10:59:35

功能描述

DescribeDocProcessBuckets 接口用于查询已经开通文档预览功能的 Bucket。

请求

请求示例



```
GET /docbucket?pageNumber=1&pageSize=2 HTTP/1.1
Host: ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

名称	描述	类型	是否必选
regions	地域信息，以“,”分隔字符串，支持 All、ap-shanghai、ap-beijing	string	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	string	否
bucketName	存储桶名称前缀，前缀搜索	string	否
pageNumber	第几页	string	否
pageSize	每页个数	string	否

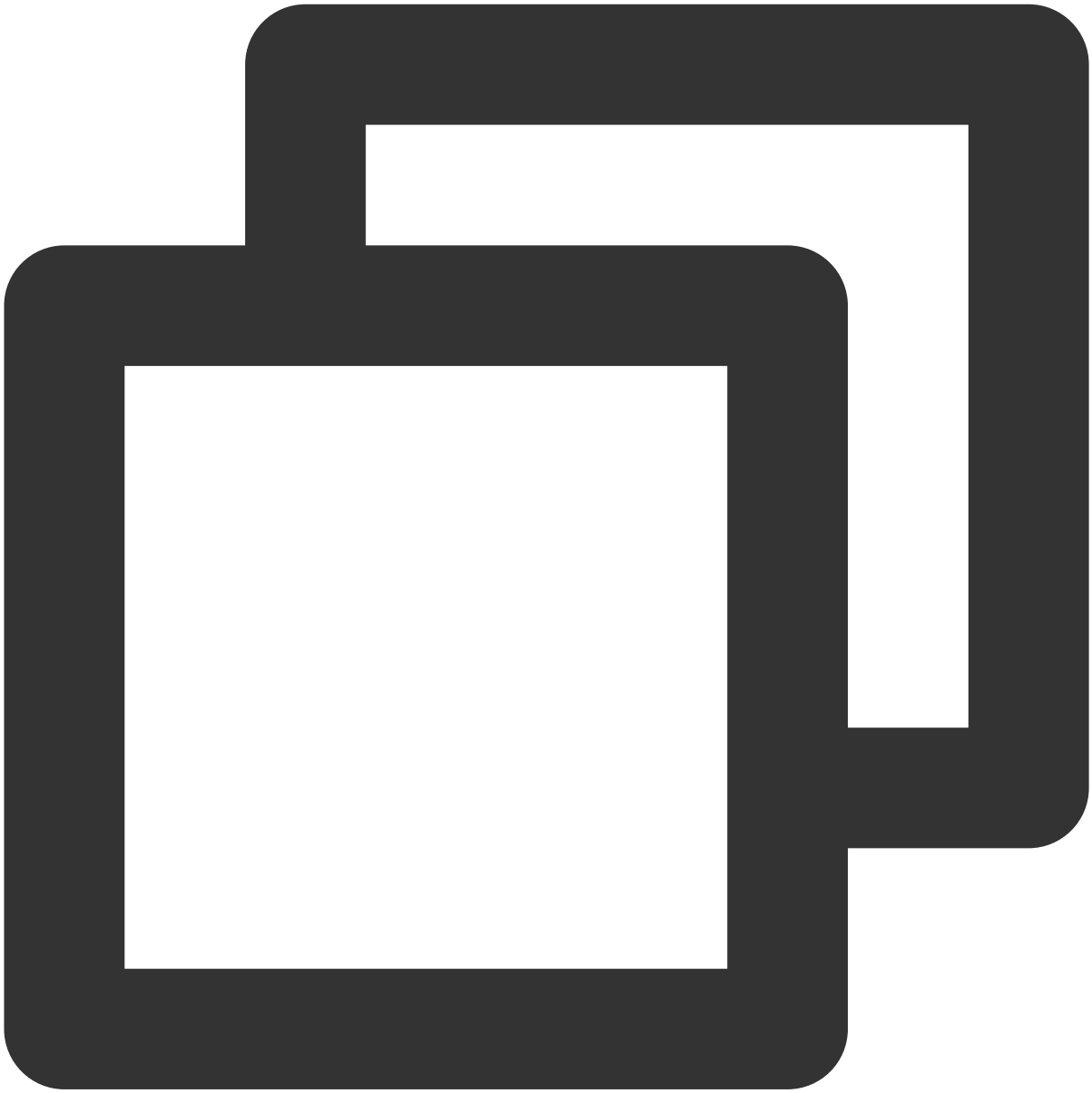
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
        <TotalCount>1</TotalCount>
        <RequestId>RequestId</RequestId>
        <PageNumber>1</PageNumber>
        <PageSize>2</PageSize>
        <DocBucketList>
                <Name>BucketName-APPID</Name>
                <CreateTime>Time</CreateTime>
                <Region>Region</Region>
                <AliasBucketId/>
                <BucketId>BucketName-APPID</BucketId>
```

```
        </DocBucketList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	文档预览 Bucket 总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
DocBucketList	Response	文档预览 Bucket 列表	Container

Container 节点 DocBucketList 的内容：

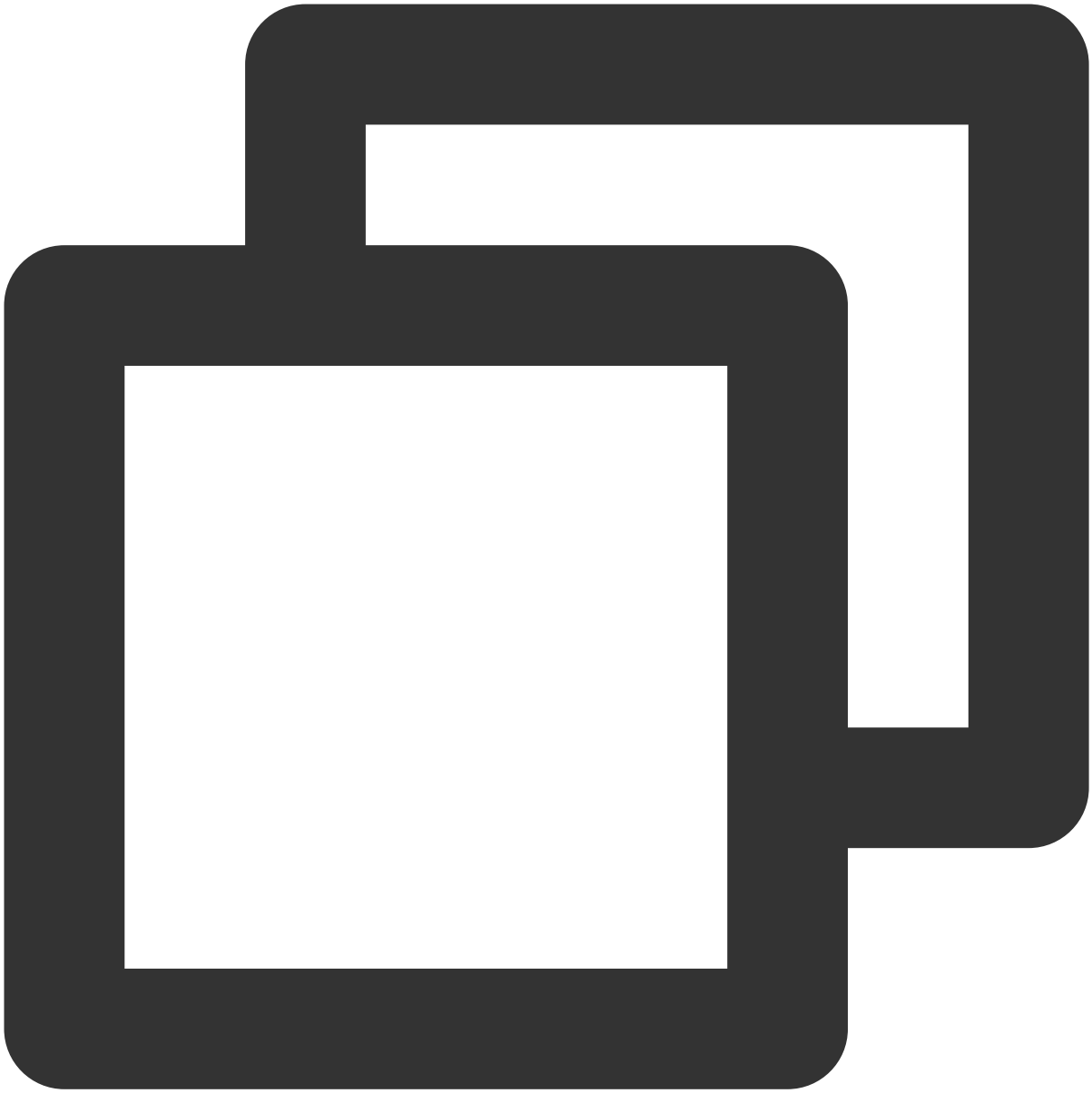
节点名称（关键字）	父节点	描述	类型
BucketId	Response.DocBucketList	存储桶 ID	String
Name	Response.DocBucketList	存储桶名称，同 BucketId	String
Region	Response.DocBucketList	所在的地域	String
CreateTime	Response.DocBucketList	创建时间	String
AliasBucketId	Response.DocBucketList	存储桶别名	String

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

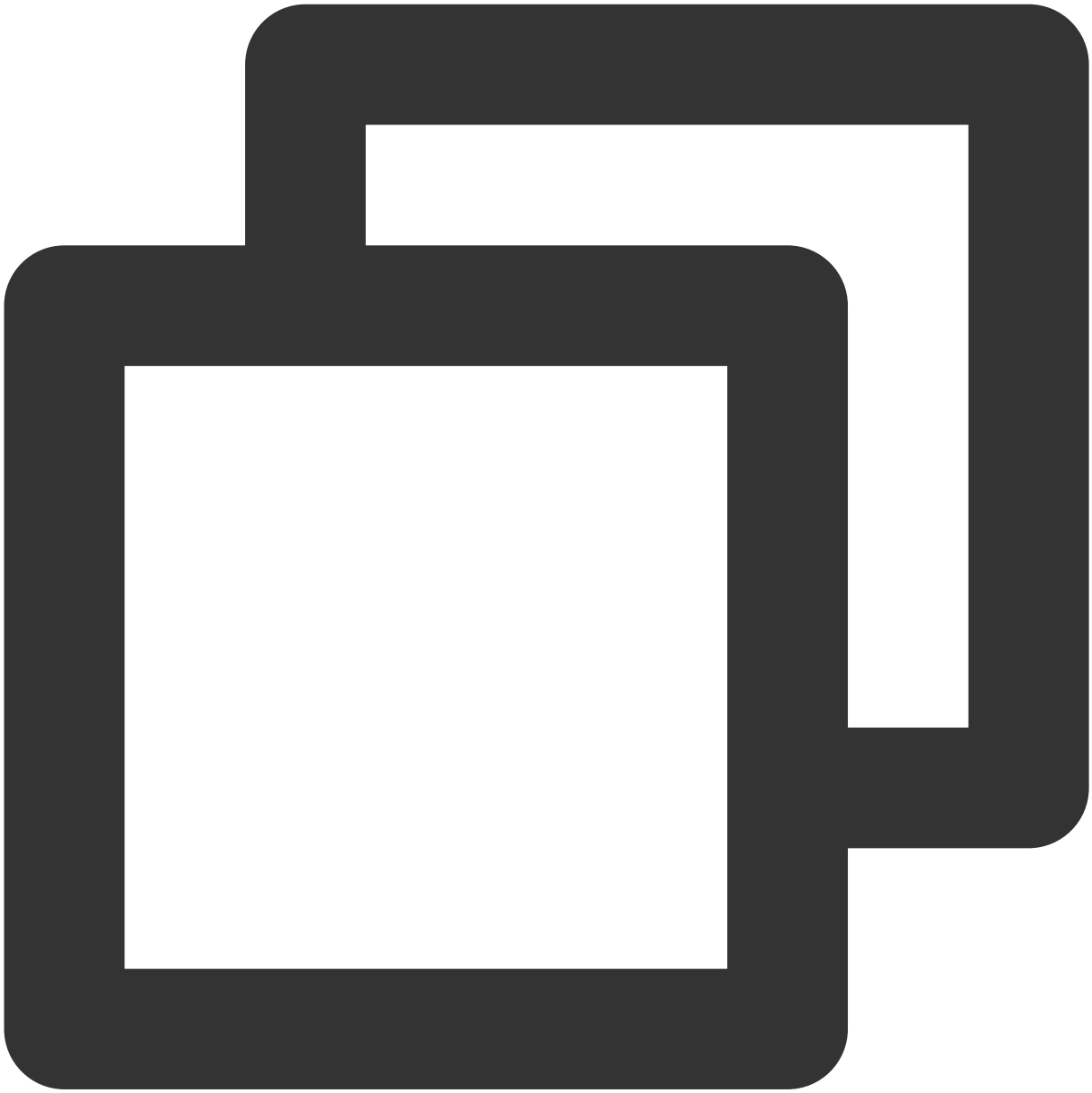
实际案例

请求



```
GET /docbucket?regions=ap-chongqing HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Authorization: Authorization
```

响应



```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 07:00:43 GMT
Content-Type: application/xml
Content-Length: 843
Connection: keep-alive
Server: tencent-ci
x-ci-request-id: NWYxZTdiOWJfYzc2OTQzNjRfM2Qz****

<?xml version="1.0" encoding="utf-8"?>
<Response>
        <TotalCount>3</TotalCount>
```

```
        <RequestId>NWYxZTdiOWJfYzc2OTQzNjRfM2Qz****</RequestId>
        <PageNumber>1</PageNumber>
        <PageSize>10</PageSize>
        <DocBucketList>
                <Name>examplebucket-1250000000</Name>
                <CreateTime>2020-07-27T10:54:42+0800</CreateTime>
                <Region>ap-chongqing</Region>
                <AliasBucketId/>
                <BucketId>examplebucket-1250000000</BucketId>
        </DocBucketList>
        <DocBucketList>
                <Name>examplebucket-1250000000</Name>
                <CreateTime>2020-07-24T22:42:26+0800</CreateTime>
                <Region>ap-chongqing</Region>
                <AliasBucketId/>
                <BucketId>examplebucket-1250000000</BucketId>
        </DocBucketList>
</Response>
```

文档转码
同步请求接口

最近更新时间：2023-12-22 11:00:14

功能描述

文档预览功能支持对多种文件类型生成图片格式预览，并可对转换后的图片同步进行基础处理操作，可以解决文档内容的页面展示问题，满足 PC、App 等多个用户端的文档在线浏览需求，适用于在线教育、企业 OA、网站转码等业务场景。

说明

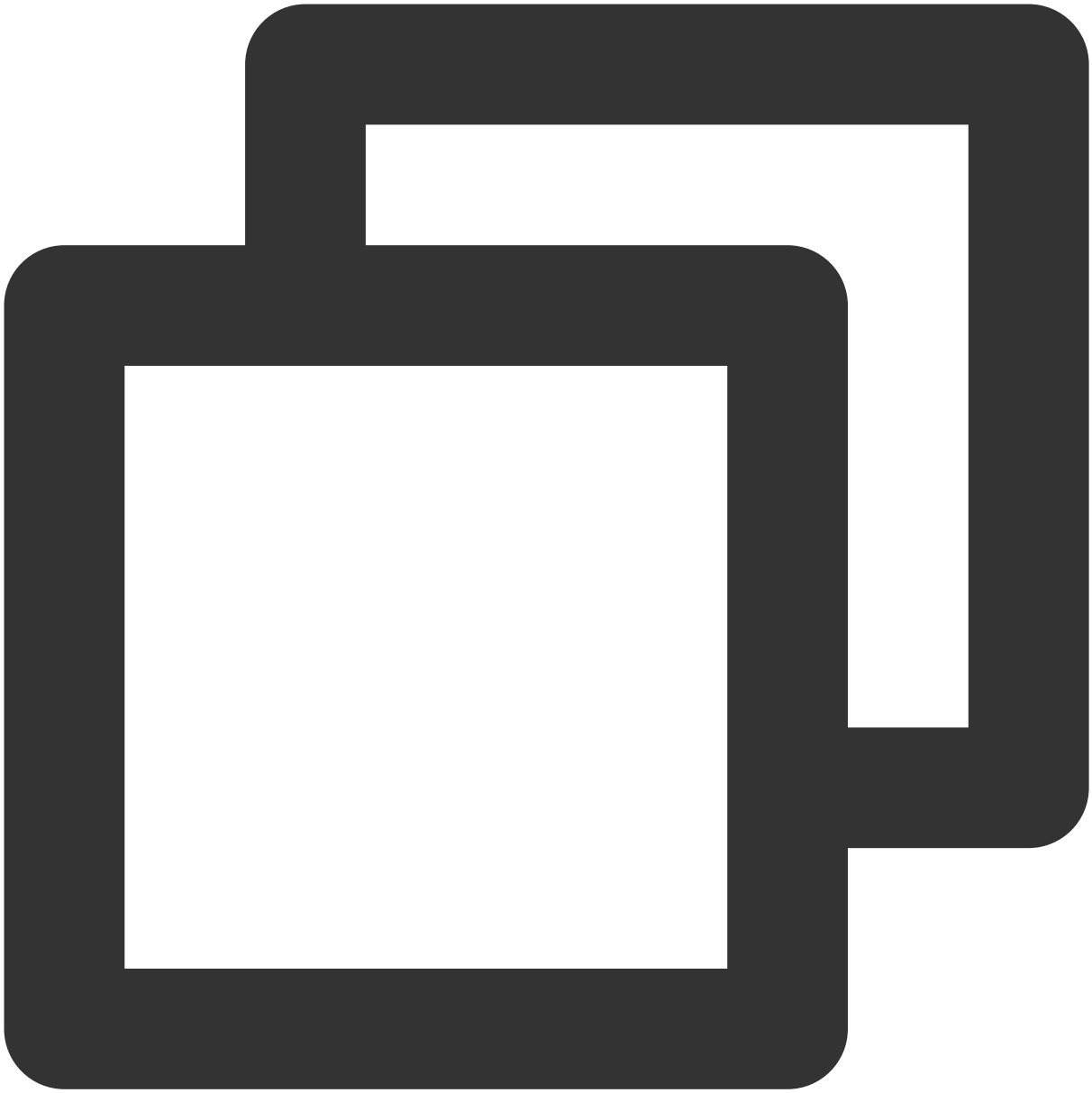
使用时您需要先开通服务，详情请参见设置文档预览。开通服务后，对于当前存储桶中的文档资源，使用相应的文档预览 API 接口即可实现同步或异步文档预览请求。

文档预览为付费功能，由数据万象收取，具体费用请参见文档处理费用。

文档转码的转换逻辑与本地的打印逻辑相同，如果您需要改变转换方向（例如将列数较多的 excel 文件横向输出到一页），需要更改源文件打印设置。

请求

请求示例



```
GET /<ObjectKey>?ci-process=doc-preview&page=<page>&srcType=<srcType>&ImageParams=<
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

私有存储桶需要携带下载文件的签名。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

通用请求参数

名称	描述	类型	是否必选
ObjectKey	对象文件名，例如 folder/document.pdf	String	是
ci-process	数据万象处理能力，文档预览固定为 doc-preview	String	是
srcType	源数据的后缀类型，当前文档转换根据 COS 对象的后缀名来确定源数据类型。当 COS 对象没有后缀名时，可以设置该值	String	否
page	需转换的文档页码，默认从1开始计数；表格文件中 page 表示转换的第 X 个 sheet 的第 X 张图	Int	否
dstType	转换输出目标文件类型： png，转成 png 格式的图片文件 jpg，转成 jpg 格式的图片文件 pdf，转成 pdf 格式的图片文件。无法选择页码，page 参数不生效 txt，转成 txt 格式文件 如果传入的格式未能识别，默认使用 jpg 格式	String	否
password	Office 文档的打开密码，如果需要转换有密码的文档，请设置该字段	String	否
comment	是否隐藏批注和修订，默认为 0 0：隐藏批注，应用修订 1：显示批注和修订	Int	否

适用于表格文件（Excel）的参数

名称	描述	类型	是否必选
sheet	表格文件参数，转换第 X 个表，默认为1	Int	否
excelPaperDirection	表格文件转换纸张方向，0代表垂直方向，非0代表水平方向，默认为0	Int	否
excelRow	值为 `1` 表示将所有列放到 1 页进行排版，默认值为 0	Int	否
excelCol	值为 `1` 表示将所有行放到 1 页进行排版，默认值为 0	Int	否
excelPaperSize	设置纸张（画布）大小，对应信息为： `0 → A4` 、 `1 → A2` 、 `2 → A0` ，默认 A4 纸张 （需配合 `excelRow` 或 `excelCol` 一起使用）	Int	否

版权所有：腾讯云计算（北京）有限责任公司

适用于转码成 txt 的参数

名称	描述	类型	是否必选
txtPagination	是否转换成长文本，设置为 `true` 时，可以将需要导出的页中的文字合并导出，分页范围可以通过 Ranges 控制。默认值为 `false` ，按页导出 txt。（ `ExportType="txt"` 时生效)	Bool	否

适用于转码成 png/jpg 图片的参数

名称	描述	类型	是否必选
ImageParams	转换后的图片处理参数，支持 基础图片处理 所有处理参数，多个处理参数可通过 管道操作符 分隔，从而实现在一次访问中按顺序对图片进行不同处理	String	否
quality	生成预览图的图片质量，取值范围为 [1, 100]，默认值100。 例如取值为100，代表生成图片质量为100%	Int	否
scale	预览图片的缩放参数，取值范围为 [10, 200]， 默认值100。 例如取值为200，代表图片缩放比例为200% 即放大两倍	Int	否
imageDpi	按指定 dpi 渲染图片，该参数与 `scale` 共同作用，取值范围 `96-600` ，默认值为 `96` 。转码后的图片单边宽度需小于65500像素	Int	否

注意

目前支持的输入文件类型包含如下格式：

演示文件：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。

文字文件：doc、dot、wps、wpt、docx、dotx、docm、dotm。

表格文件：xls、xlt、et、ett、xlsx、xltx、csv、xlsb、xlsm、xltm、ets。

表格文件，一张表可能分割为多页转换，生成多张图片。

其他格式文件： pdf、 lrc、 c、 cpp、 h、 asm、 s、 java、 asp、 bat、 bas、 prg、 cmd、 rtf、 txt、 log、 xml、 htm、 html。

输入文件大小限制在200MB之内。

输入文件页数限制在5000页之内。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应参数说明

参数名称	描述	类型
X-Total-Page	在 http header 中返回文档总页数（表格文件表示当前 sheet 转换后的总图片数），异常时为空	Int
Content-Type	根据图片实际的格式，返回不同的值，例如 image/jpeg、image/webp，异常时保持现有值即可	String
X-ErrNo	当异常时返回错误码，可在 http header 中查看	String
X-Total-Sheet	表格文件返回参数，在 http header 中返回文档中总表数	Int
X-Sheet-Name	表格文件返回参数，在 http header 中返回当前 sheet 名	String

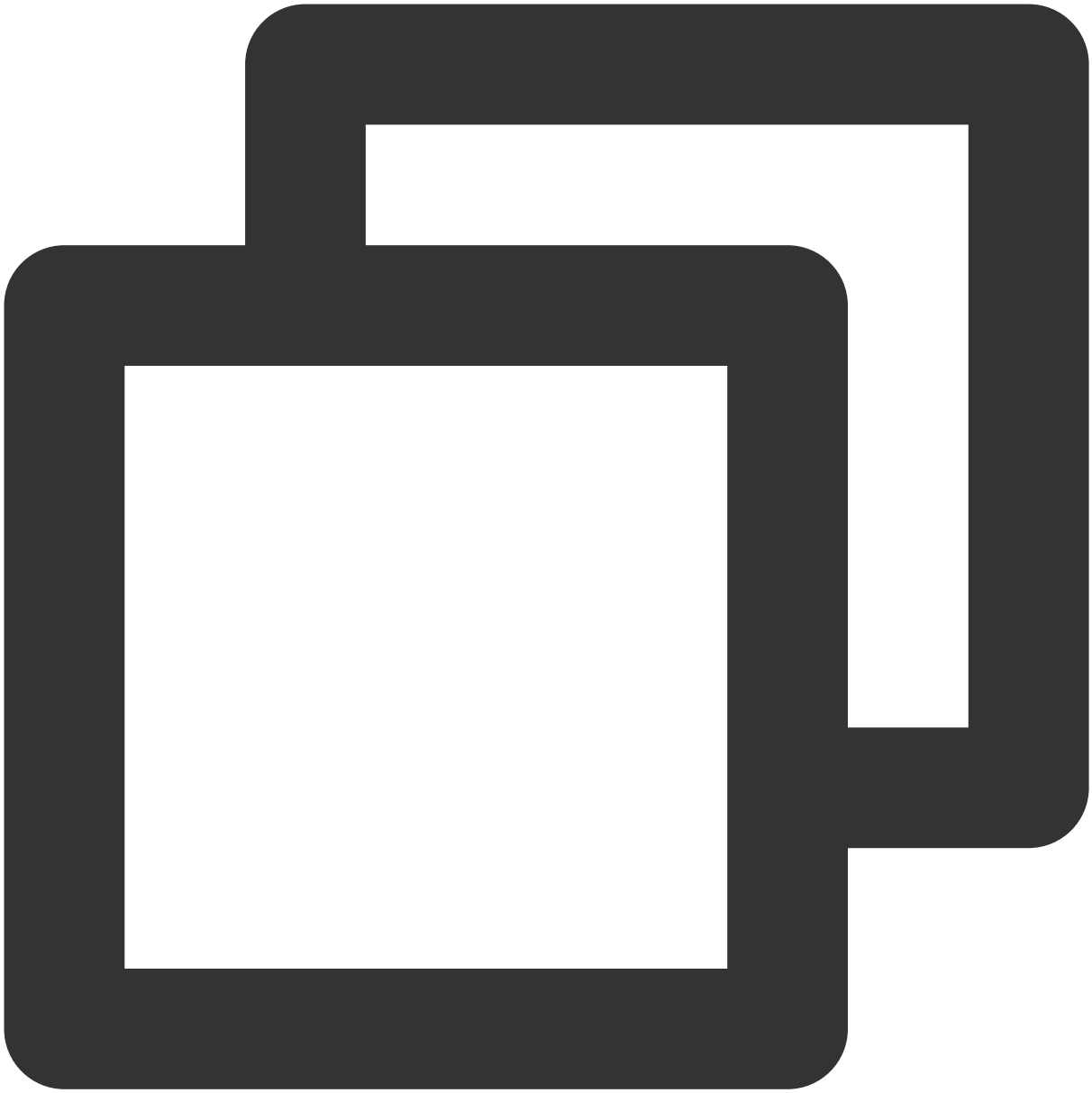
错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

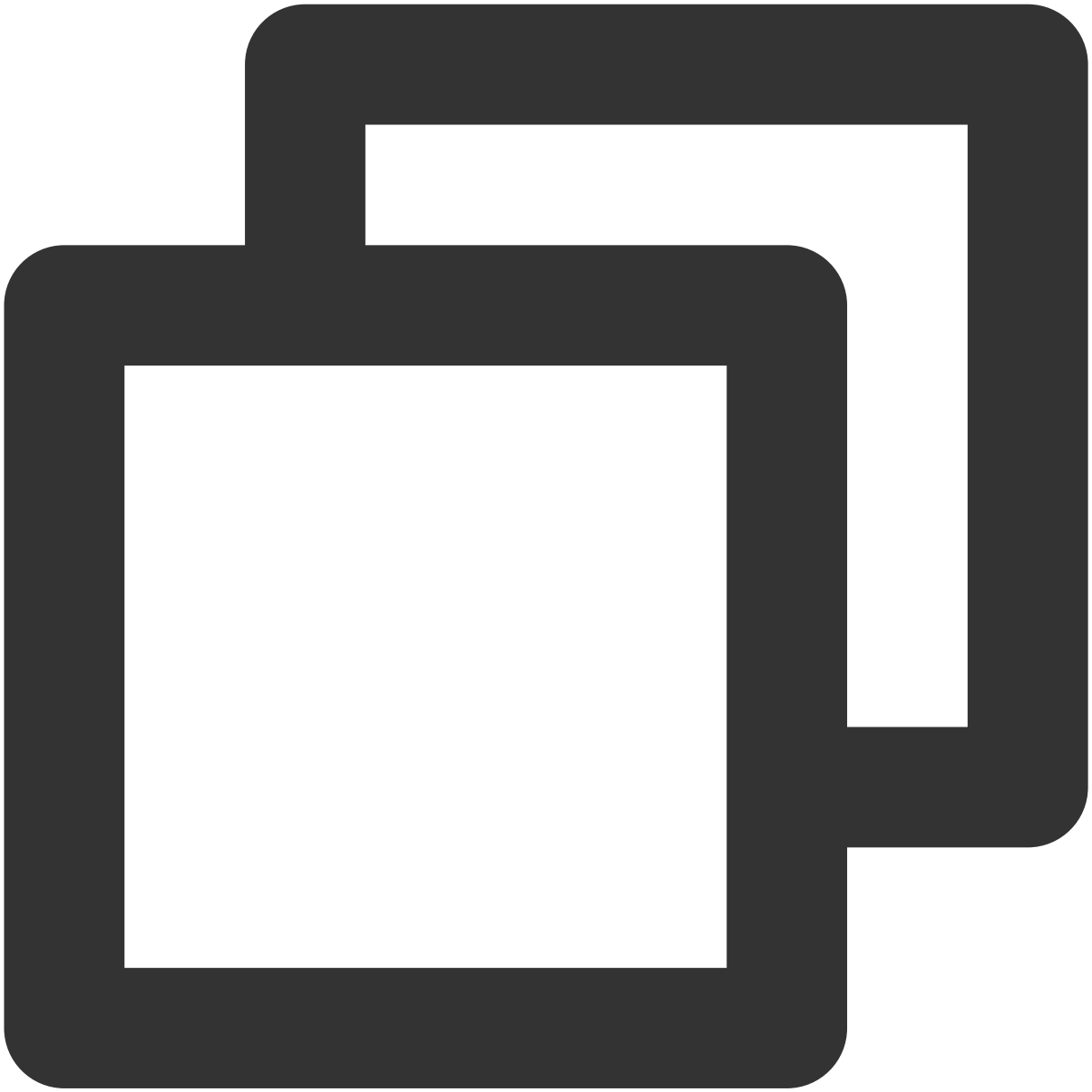
案例一：普通文档预览

请求



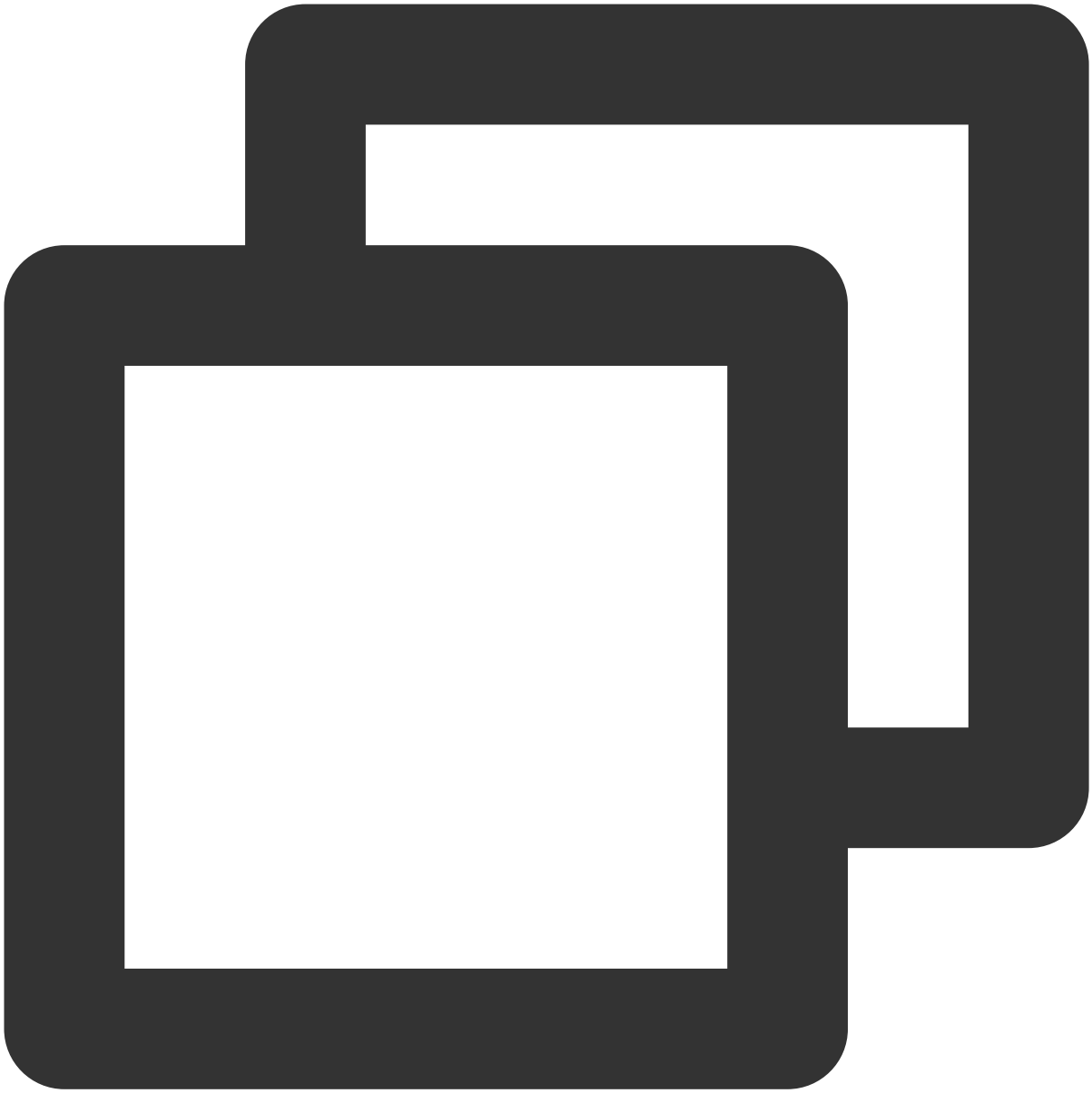
```
GET /sample.pdf?ci-process=doc-preview&page=1 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
```

响应1：成功预览



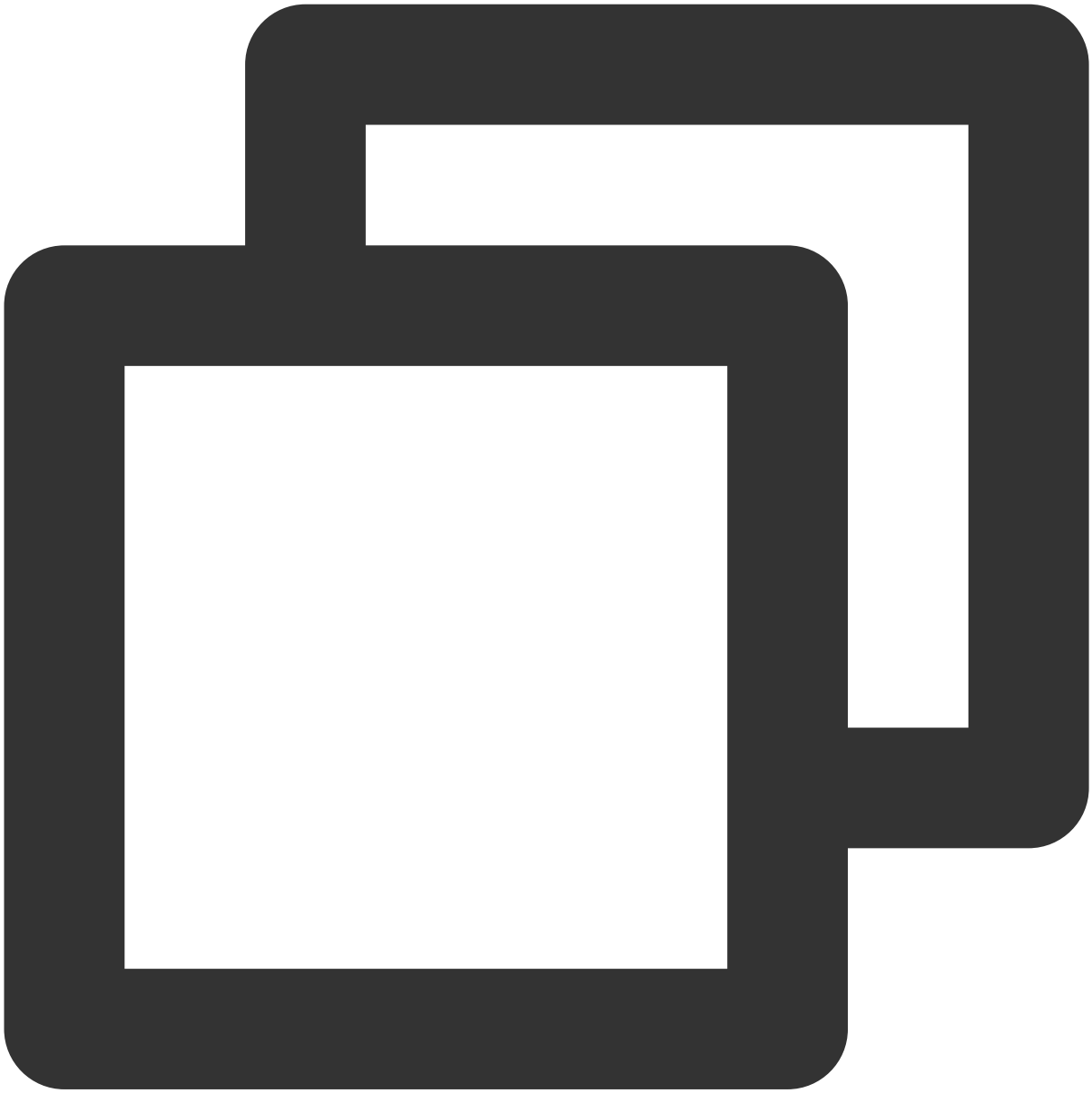
```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
X-Total-Page: 100

[转码后图片data]
```

响应2：文件不存在



```
HTTP/1.1 404 OK
Connection: close
Content-Length: 455
Content-Type: application/xml
Date: Thu, 09 Apr 2020 13:44:52 GMT
X-ErrNo: -3004
```

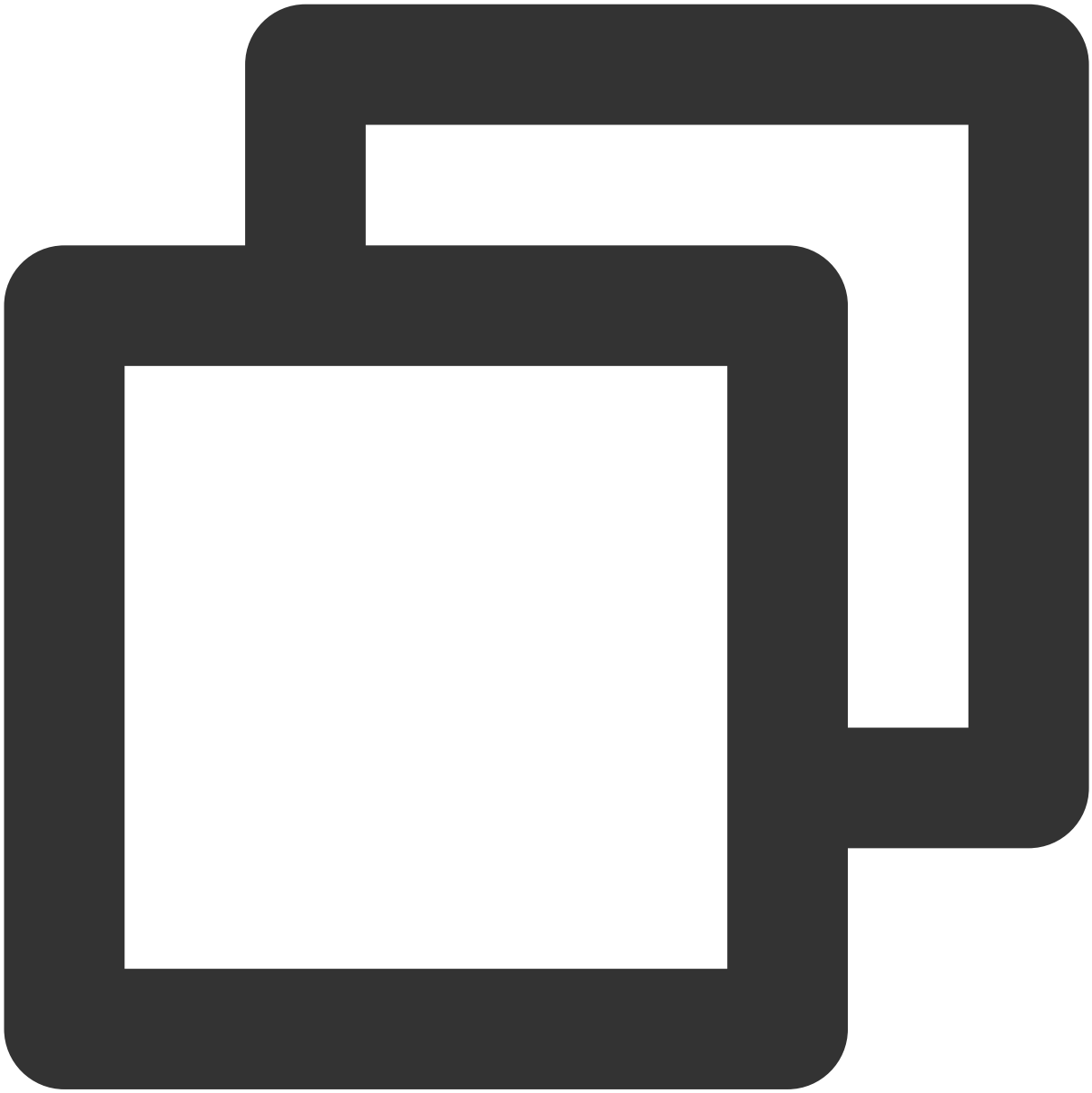
响应3：指定页码不存在



```
HTTP/1.1 404 OK
Connection: close
Content-Length: 455
Content-Type: application/xml
Date: Thu, 09 Apr 2020 13:44:52 GMT
X-ErrNo: -3013
```

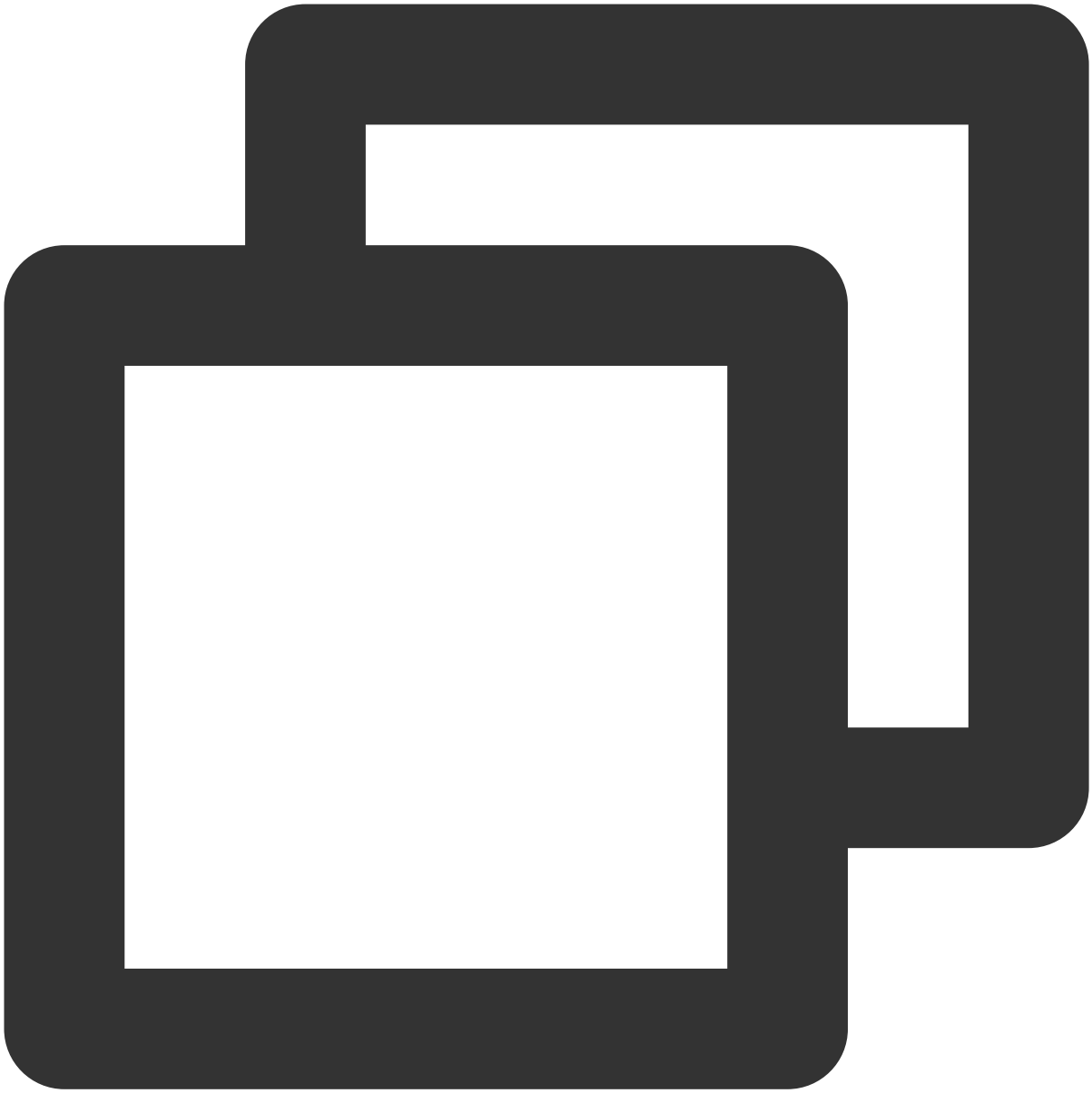
案例二：文档预览并进行缩放和添加文字水印的图片处理

请求



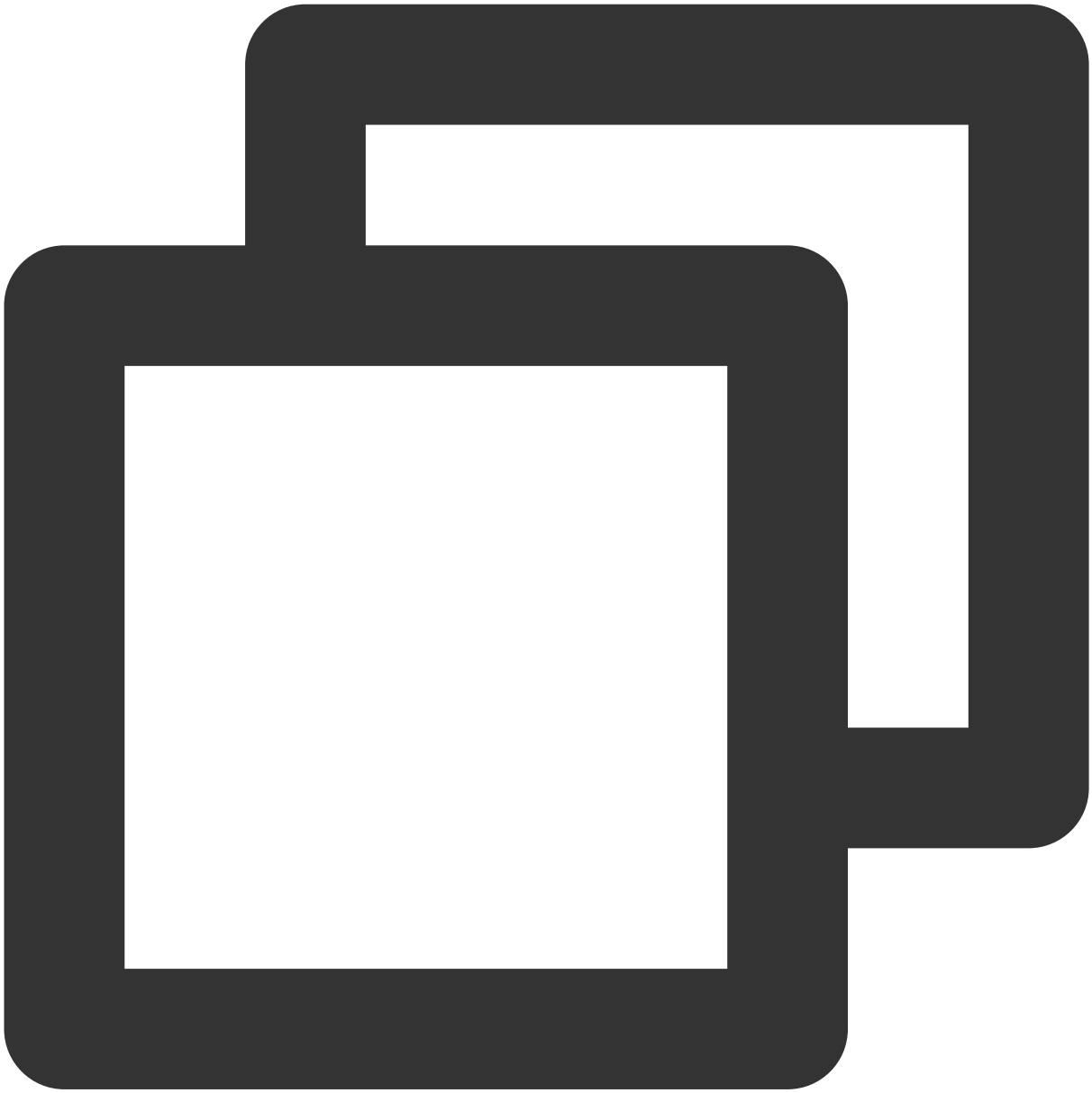
```
GET /sample.pdf?ci-process=doc-preview&page=1&ImageParams=imageMogr2/thumbnail/!50p
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
```

响应1：成功预览



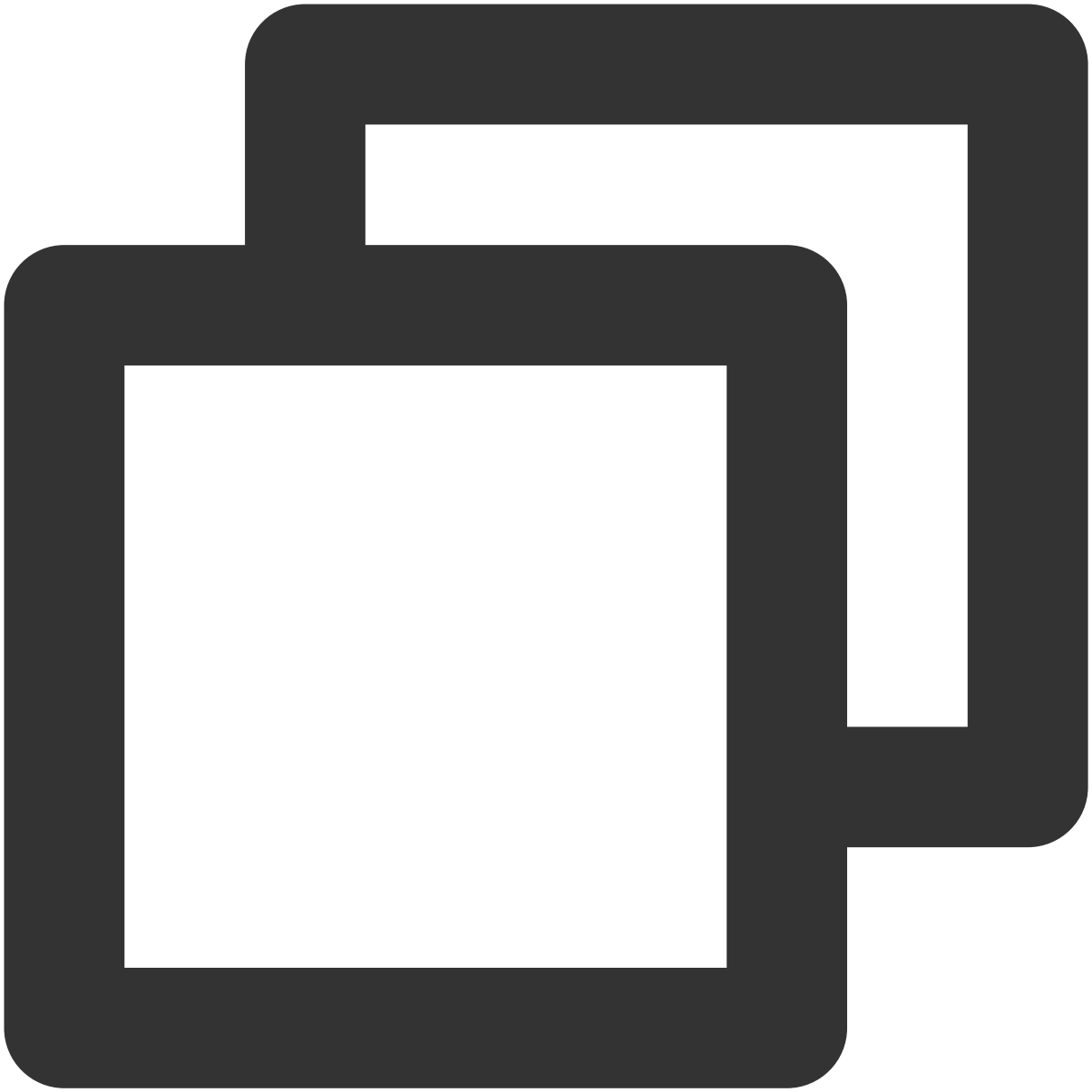
```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
X-Total-Page: 100

[转码后图片data]
```

响应2：文件不存在



```
HTTP/1.1 404 OK
Connection: close
Content-Length: 455
Content-Type: application/xml
Date: Thu, 09 Apr 2020 13:44:52 GMT
X-ErrNo: -3004
```

响应3：指定页码不存在



```
HTTP/1.1 404 OK
Connection: close
Content-Length: 455
Content-Type: application/xml
Date: Thu, 09 Apr 2020 13:44:52 GMT
X-ErrNo: -3013
```

异步处理任务接口
提交文档转码任务

最近更新时间：2023-12-22 11:00:34

功能描述

CreateDocProcessJobs 接口用于提交一个文档转码任务。

注意

目前支持的输入文件类型包含如下格式：

演示文件：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。

文字文件：doc、dot、wps、wpt、docx、dotx、docm、dotm。

表格文件：xls、xlt、et、ett、xlsx、xltx、csv、xlsb、xlsm、xltm、ets。

表格文件，一张表可能分割为多页转换，生成多张图片。

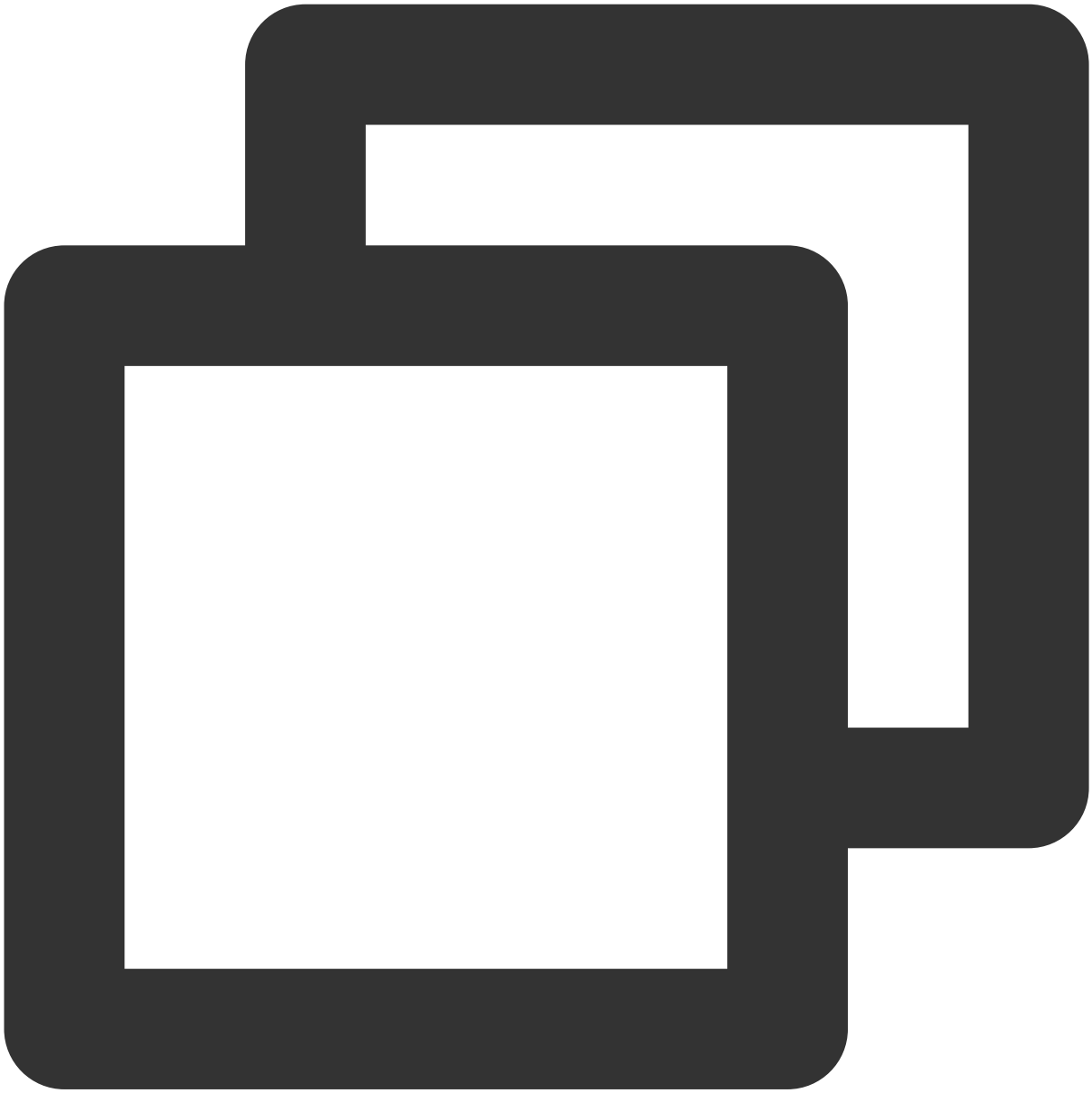
其他格式文件：pdf、lrc、c、cpp、h、asm、s、java、asp、bat、bas、prg、cmd、rtf、txt、log、xml、htm、html。

输入文件大小限制在200M之内。

输入文件页数限制在5000页之内。

请求

请求示例



```
POST /doc_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

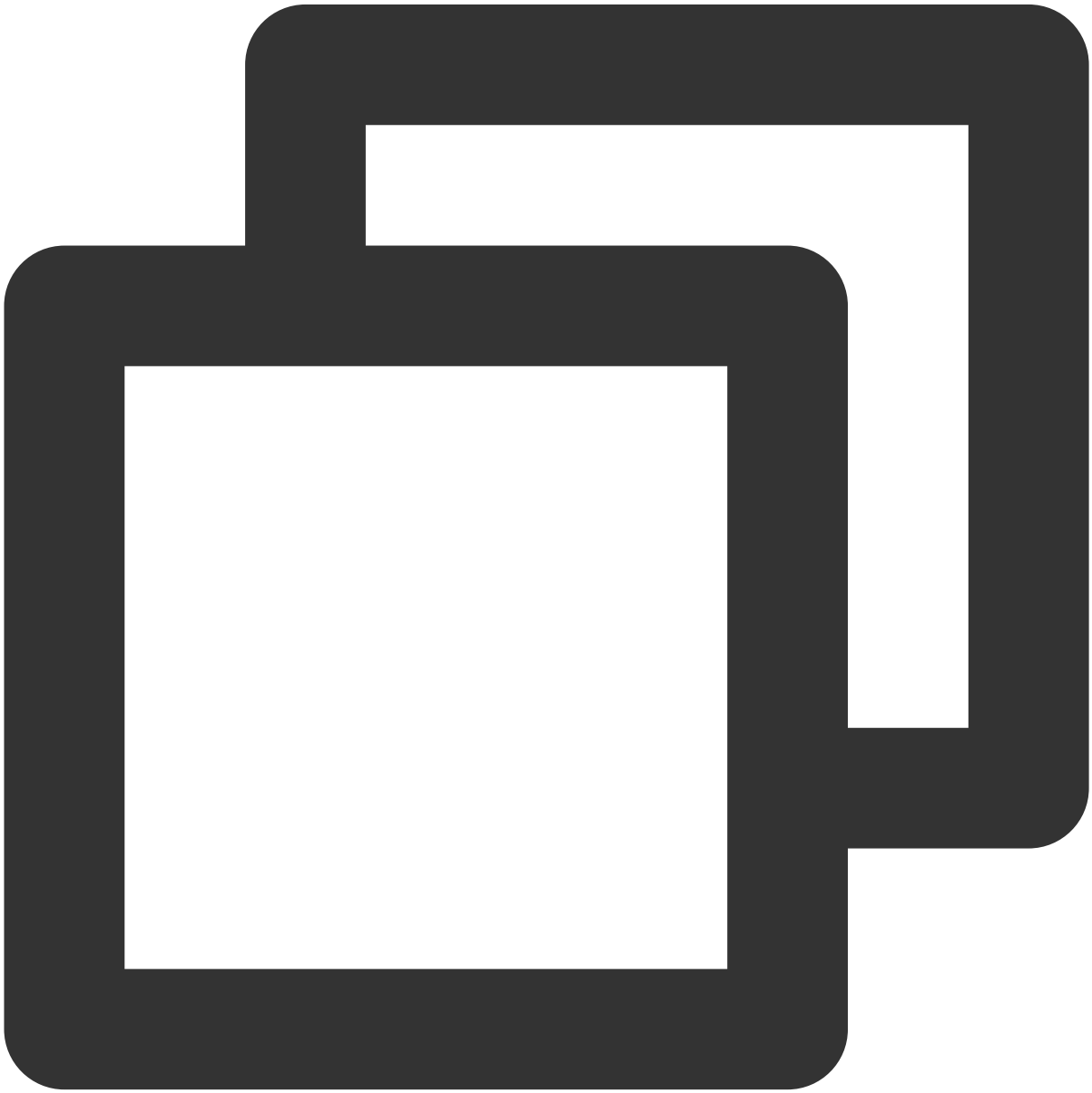
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
  <Tag></Tag>
  <Input>
    <Object></Object>
  </Input>
  <Operation>
    <Output>
      <Region></Region>
      <Bucket></Bucket>
      <Object></Object>
    </Output>
    <DocProcess>
       <StartPage></StartPage>
       <EndPage></EndPage>
       <TgtType></TgtType>
    </DocProcess>
  </Operation>
  <QueueId></QueueId>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	创建任务的 Tag，目前仅支持：DocProcess	String	是
Input	Request	待操作的文件对象	Container	是
Operation	Request	操作规则	Container	是
QueueId	Request	任务所在的队列 ID，开通预览服务后自动生成，请使用 查询队列 获取或前往 万象控制台 在存储桶中查询	String	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选

Object	Request.Input	文件在 COS 上的文件路径，Bucket 由 Host 指定	String	是

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
DocProcess	Request.Operation	当 Tag 为 DocProcess 时有效，指定该任务的参数	Container	是
Output	Request.Operation	结果输出地址	Container	是

Container 类型 DocProcess 的具体数据描述如下：

通用请求参数

节点名称（关键字）	父节点	描述	类型	是否必选
SrcType	Request.Operation.DocProcess	源数据的后缀类型，当前文档转换根据 cos 对象的后缀名来确定源数据类型，当 cos 对象没有后缀名时，可以设置该值	String	否
TgtType	Request.Operation.DocProcess	转换输出目标文件类型： jpg，转成 jpg 格式的图片文件；如果传入的格式未能识别，默认使用 jpg 格式 png，转成 png 格式的图片文件 pdf，转成 pdf 格式文件（暂不支持指定页数）	String	否
StartPage	Request.Operation.DocProcess	从第 X 页开始转换；在表格文件中，一张表可能分割为多页转换，生成多张图片。StartPage 表示从指定 SheetId 的第 X 页开始转换。默认为1	Int	否
EndPage	Request.Operation.DocProcess	转换至第 X 页；在表格文件中，一张表可能分割为多页转换，生成多张图片。EndPage 表示转换至指定 SheetId 的第 X 页。默认为-1，即转换全部页	Int	否

版权所有：腾讯云计算（北京）有限责任公司

适用于表格文件（Excel）的参数

节点名称（关键字）	父节点	描述	类型	是否必选
SheetId	Request.Operation.DocProcess	表格文件参数，转换第 X 个表，默认为0；设置 SheetId 为0，即转换文档中全部表	Int	否
PaperDirection	Request.Operation.DocProcess	表格文件转换纸张方向，0代表垂直方向，非0代表水平方向，默认为0	Int	否
PaperSize	Request.Operation.DocProcess	设置纸张（画布）大小，对应信息为：`0 → A4` 、 `1 → A2` 、 `2 → A0` ，默认 A4 纸张	Int	否

适用于转码成 png/jpg 图片的参数

节点名称（关键字）	父节点	描述	类型	是否必选
ImageParams	Request.Operation.DocProcess	转换后的图片处理参数，支持 基础图片处理 所有处理参数，多个处理参数可通过 管道操作符 分隔，从而实现在一次访问中按顺序对图片进行不同处理	String	否
Quality	Request.Operation.DocProces	生成预览图的图片质量，取值范围 [1-100]，默认值100。 例：值为100，代表生成图片质量为100%	Int	否
Zoom	Request.Operation.DocProces	预览图片的缩放参数，取值范围[10-200]， 默认值100。 例：值为200，代表图片缩放比例为200% 即放大两倍	Int	否
ImageDpi	Request.Operation.DocProcess	按指定 dpi 渲染图片，该参数与 `Zoom` 共同作用，取值范围 `96-600` ，默认值为 `96` 。转码后的图片单边宽度需小于65500像素	Int	否
PicPagination	Request.Operation.DocProcess	是否转换成单张长图，设置为 `1` 时，最多仅支持将 20 标准页面合成单张长图，超过可能会报错，分页范围	Int	否

版权所有：腾讯云计算（北京）有限责任公司

		可以通过 StartPage、EndPage 控制。默认值为 `0` ，按页导出图片， `TgtType="png"/"jpg"` 时生效		

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Region	Request.Operation.Output	存储桶的地域	String	是
Bucket	Request.Operation.Output	存储结果的存储桶	String	是
Object	Request.Operation.Output	输出文件路径。 **非表格文件输出文件名需包含 ${Number} 或 ${Page} 参数。**多个输出文件，${Number} 表示序号从1开始，${Page} 表示序号与预览页码一致。 ${Number} 表示多个输出文件，序号从1开始，例如输入 abc_${Number}.jpg，预览某文件5 - 6页，则输出文件名为 abc_1.jpg，abc_2.jpg ${Page} 表示多个输出文件，序号与预览页码一致，例如输入 abc_${Page}.jpg，预览某文件5-6页，则输出文件名为 abc_5.jpg，abc_6.jpg **表格文件输出路径需包含 ${SheetID} 占位符，输出文件名必须包含 ${Number} 参数。**例如 `/${SheetID}/abc_${Number}.jpg` ，先根据 excel 转换的表格数，生成对应数量的文件夹，再在对应的文件夹下，生成对应数量的图片文件	String	是

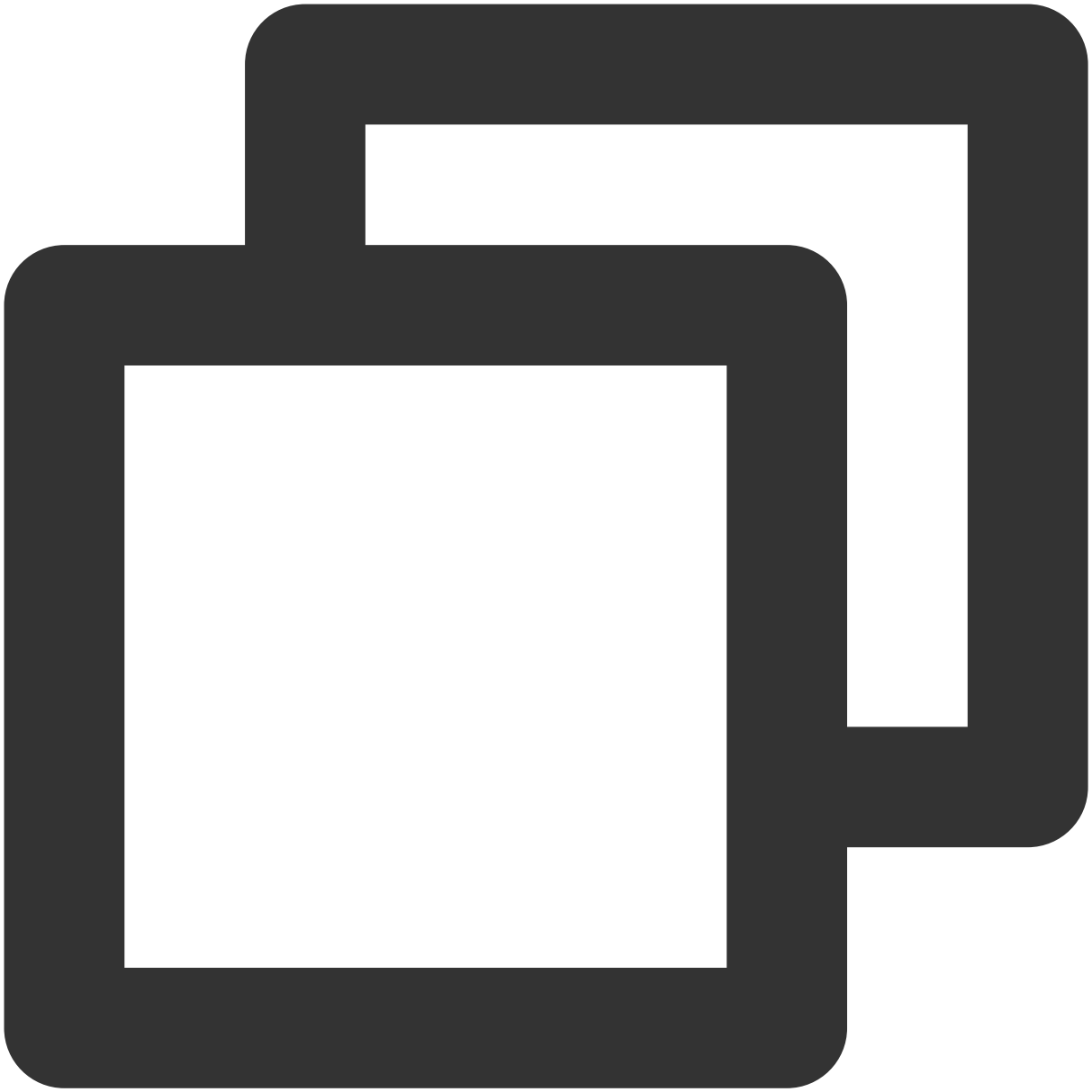
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
        <JobsDetail>
                <Code></Code>
                <CreationTime></CreationTime>
                <EndTime></EndTime>
                <Input>
                    <Object></Object>
                </Input>
                <JobId></JobId>
                <Message/>
                <Operation>
```

```
                    <DocProcess>
                        <SrcType></SrcType>
                        <TgtType></TgtType>
                        <StartPage></StartPage>
                        <EndPage></EndPage>
                        <ImageParams></ImageParams>
                    </DocProcess>
                    <Output>
                        <Bucket></Bucket>
                        <Object></Object>
                        <Region></Region>
                    </Output>
                </Operation>
                <QueueId></QueueId>
                <State></State>
                <Tag></Tag>
        </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
JobId	Response.JobsDetail	新创建任务的 ID	String
Tag	Response.JobsDetail	新创建任务的 Tag：DocProcess	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String

QueueId	Response.JobsDetail	任务所属的队列 ID	String
Input	Response.JobsDetail	该任务的输入文件路径	Container
Operation	Response.JobsDetail	该任务的规则	Container

Container 节点 Input 的内容：

同上面请求中的 Request.Input 节点。

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
DocProcess	Response.JobsDetail.Operation	文档预览任务参数	Container
Output	Response.JobsDetail.Operation	文件的输出地址	Container

Container 节点 DocProcess 的内容：

同上面请求中的 Request.Operation.DocProcess 节点。

Container 节点 Output 的内容：

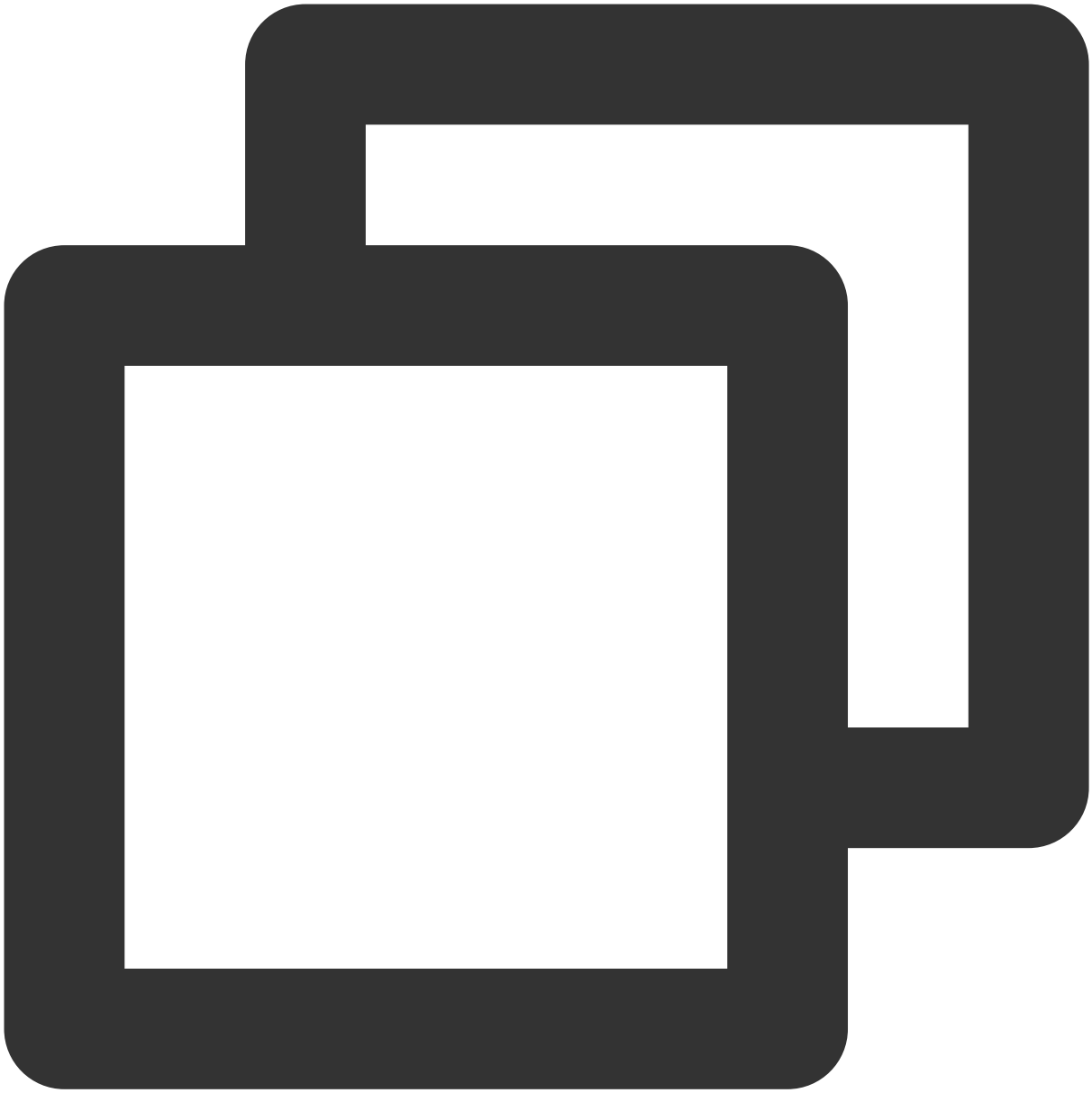
同上面请求中的 Request.Operation.Output 节点。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

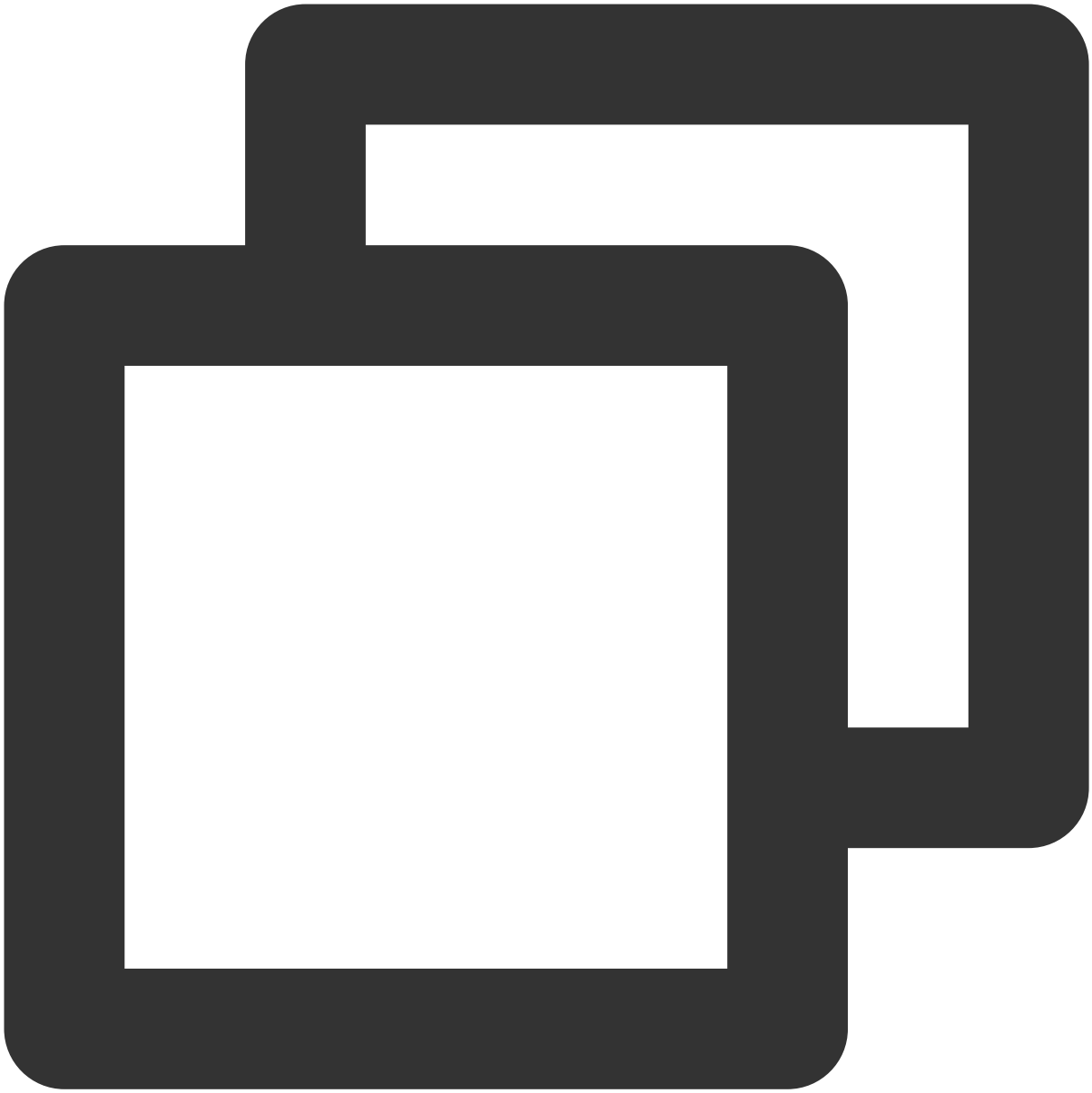
请求



```
POST /doc_jobs HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: examplebucket-1250000000.ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Content-Length: 546
Authorization: Authorization

<?xml version="1.0" encoding="UTF-8" ?>
```

```
<Request>
    <Input>
        <Object>1.doc</Object>
    </Input>
    <Operation>
      <Output>
          <Region>ap-chongqing</Region>
          <Object>big/test-${Number}</Object>
          <Bucket>examplebucket-1250000000</Bucket>
      </Output>
      <DocProcess>
          <TgtType>png</TgtType>
          <StartPage>1</StartPage>
          <EndPage>-1</EndPage>
          <ImageParams>watermark/1/image/aHR0cDovL3Rlc3QwMDUtMTI1MTcwNDcwOC5jb3MuYX
      </DocProcess>
    </Operation>
    <Tag>DocProcess</Tag>
    <QueueId>p532fdead78444e649e1a4467c1cd19d3</QueueId>
</Request>[!http]
```

响应



```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 07:20:08 GMT
Content-Type: application/xml
Content-Length: 863
Connection: keep-alive
Server: tencent-ci
x-ci-request-id: NWYxZTgwMjhfYzc2OTQzNjRfMzUx****

<?xml version="1.0" encoding="utf-8"?>
<Response>
        <JobsDetail>
```

```
                <Code>Success</Code>
                <CreationTime>2020-07-27T15:20:08+0800</CreationTime>
                <EndTime>-</EndTime>
                <Input>
                        <Object>1.doc</Object>
                </Input>
                <JobId>d99b3127ecfd911eab5e60dedb7c395dd</JobId>
                <Message/>
                <Operation>
                        <DocProcess>
                                <EndPage>5001</EndPage>
                                <ImageParams>watermark/1/image/aHR0cDovL3Rlc3QwMDUt
                                <SrcType/>
                                <StartPage>1</StartPage>
                                <TgtType>png</TgtType>
                        </DocProcess>
                        <Output>
                                <Bucket>examplebucket-1250000000</Bucket>
                                <Object>big/test-${Number}</Object>
                                <Region>ap-chongqing</Region>
                        </Output>
                </Operation>
                <QueueId>p532fdead78444e649e1a4467c1cd19d3</QueueId>
                <State>Submitted</State>
                <Tag>DocProcess</Tag>
        </JobsDetail>
</Response>
```

查询指定的文档转码任务

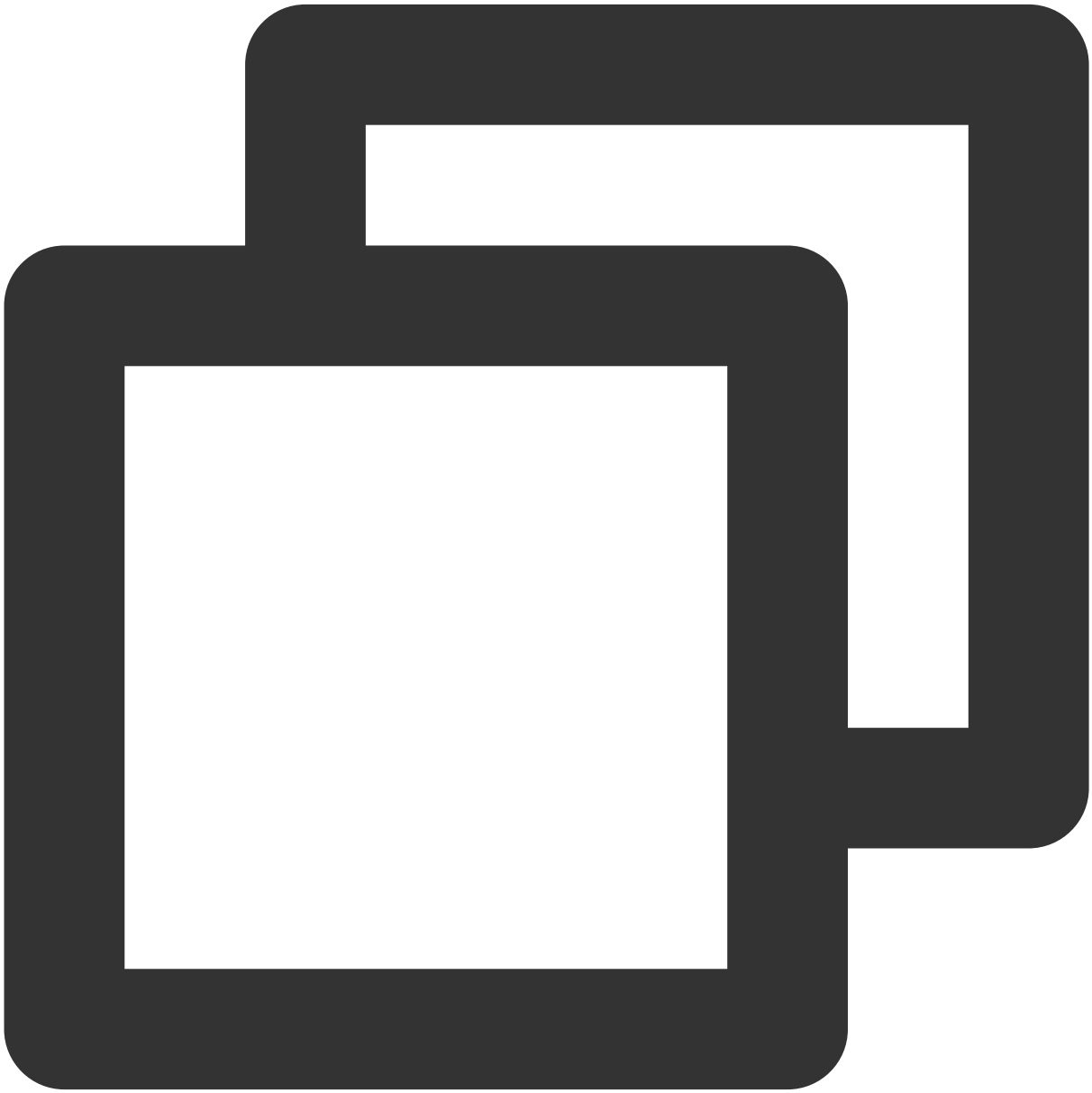
最近更新时间：2023-12-22 11:01:06

功能描述

DescribeDocProcessJob 用于查询指定的文档转码任务。

请求

请求示例



```
GET /doc_jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

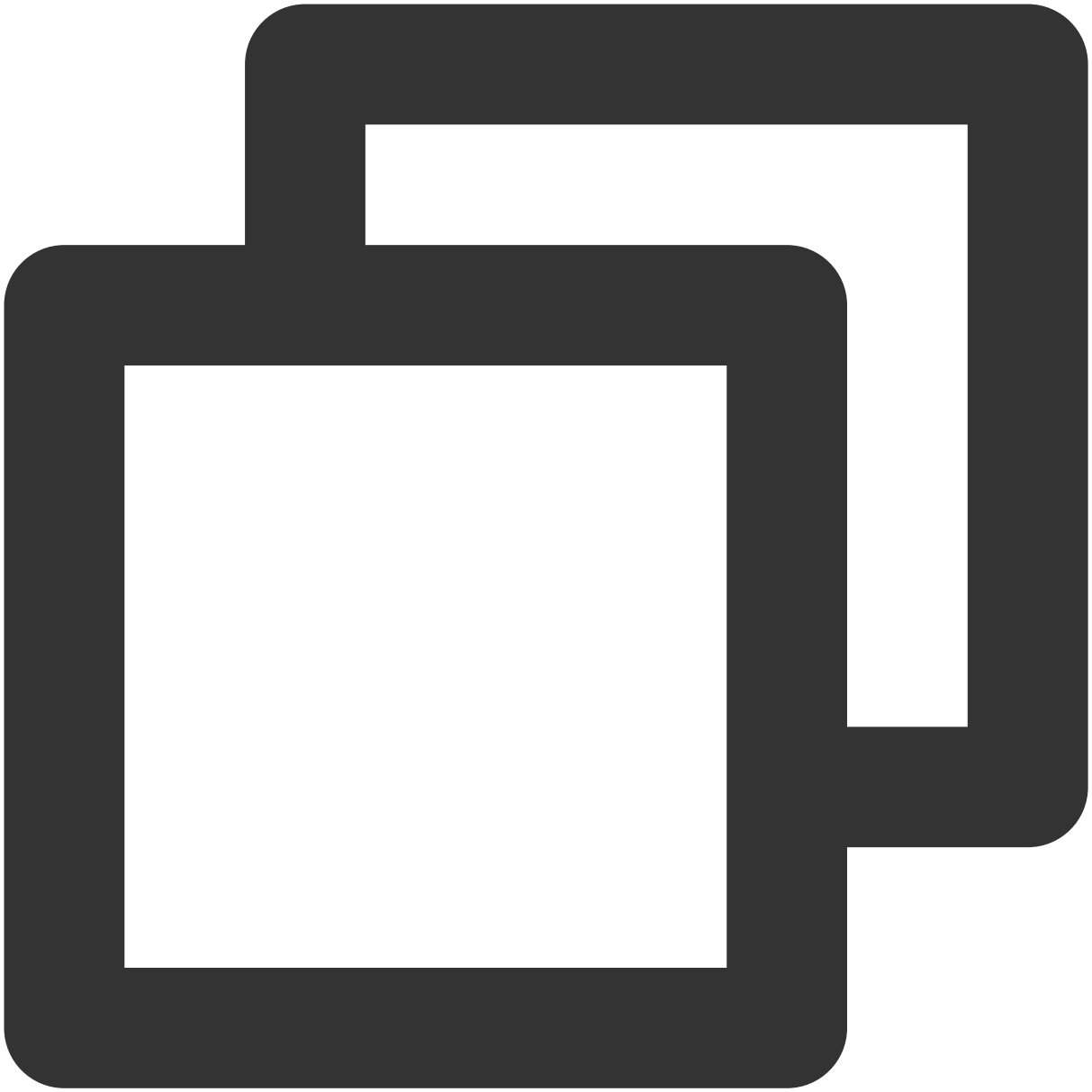
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
        <JobsDetail>
                <Code></Code>
                <CreationTime></CreationTime>
                <EndTime></EndTime>
                <Input>
                        <Object></Object>
                </Input>
                <JobId></JobId>
                <Message> </Message>
                <Operation>
```

```
                        <DocProcess>
                                <EndPage></EndPage>
                                <ImageParams></ImageParams>
                                <SrcType></SrcType>
                                <StartPage></StartPage>
                                <TgtType></TgtType>
                        </DocProcess>
                        <DocProcessResult>
                                <FailPageCount></FailPageCount>
                                <PageInfo>
                                        <PageNo></PageNo>
                                        <TgtUri></TgtUri>
                                </PageInfo>
                                <SuccPageCount></SuccPageCount>
                                <TaskId></TaskId>
                                <TgtType></TgtType/>
                                <TotalPageCount></TotalPageCount>
                        </DocProcessResult>
                        <Output>
                                <Bucket></Bucket>
                                <Object></Object>
                                <Region></Region>
                        </Output>
                </Operation>
                <QueueId></QueueId>
                <State></State>
                <Tag></Tag>
        </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息，同 CreateDocProcessJobs 接口的 Response.JobsDetail 节点	Container
NonExistJobIds	Response	查询的 ID 中不存在的任务，所有任务都存在时不返回	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
DocProcess	Response.JobsDetail.Operation	文档预览任务参数，同 CreateDocProcessJobs 接口的 Request.Operation.DocProcess 节点	Container
DocProcessResult	Response.JobsDetail.Operation	在 job 的类型为 DocProcess 且 job 状态为 success 时，返回文档预览任务结果详情	Container
Output	Response.JobsDetail.Operation	结果输出地址，同 CreateDocProcessJobs 接口的 Request.Operation.Output 节点	Container

Container 节点 DocProcessResult 节点的内容

节点名称（关键字）	父节点	描述	类型
PageInfo	Response.JobsDetail.Operation.DocProcessResult	预览任务产物详情	Container
TgtType	Response.JobsDetail.Operation.DocProcessResult	预览产物目标格式	String
TotalPageCount	Response.JobsDetail.Operation.DocProcessResult	预览任务产物的总数	Int
SuccPageCount	Response.JobsDetail.Operation.DocProcessResult	预览任务产物的成功数	Int
FailPageCount	Response.JobsDetail.Operation.DocProcessResult	预览任务产物的失败数	Int
TotalSheetCount	Response.JobsDetail.Operation.DocProcessResult	预览任务的 Sheet 总数（源文件为 Excel 特有参数）	Int

Container 节点 PageInfo 节点的内容

节点名称（关键字）	父节点	描述	类型
PageNo	Response.JobsDetail.Operation.DocProcessResult.PageInfo	预览产物页码,源文件为 Excel 格式时表示 SheetId	Container

版权所有：腾讯云计算（北京）有限责任公司

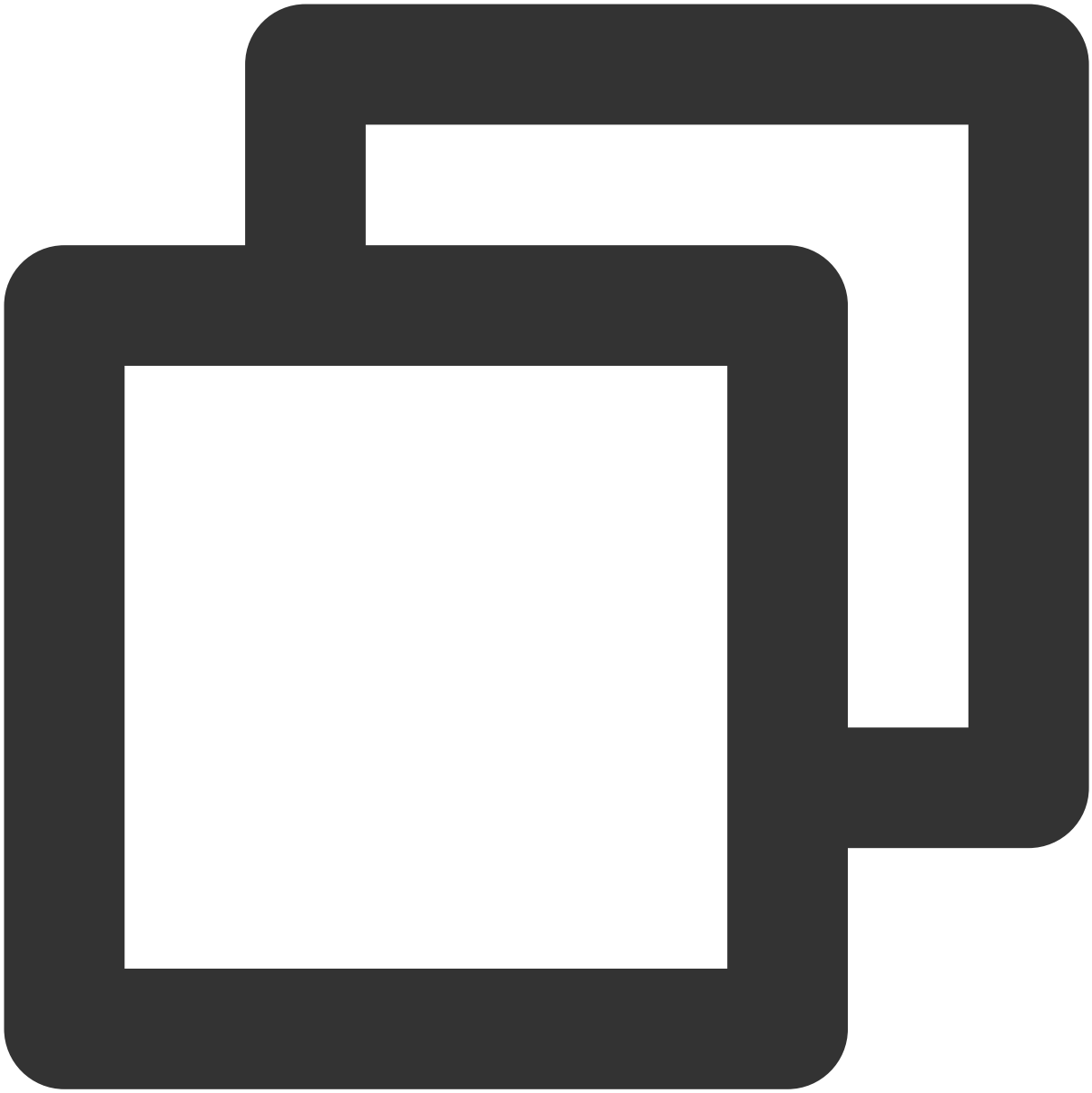
TgtUri	Response.JobsDetail.Operation.DocProcessResult.PageInfo	预览产物生成的 cos 桶路径	Int
X-SheetPics	Response.JobsDetail.Operation.DocProcessResult.PageInfo	当前 Sheet 生成的图片总数（源文件为 Excel 特有参数）	Int
PicIndex	Response.JobsDetail.Operation.DocProcessResult.PageInfo	当前预览产物在整个源文件中的序号（源文件为 Excel 特有参数）	Int
PicNum	Response.JobsDetail.Operation.DocProcessResult.PageInfo	当前预览产物在 Sheet 中的序号（源文件为 Excel 特有参数）	Int

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

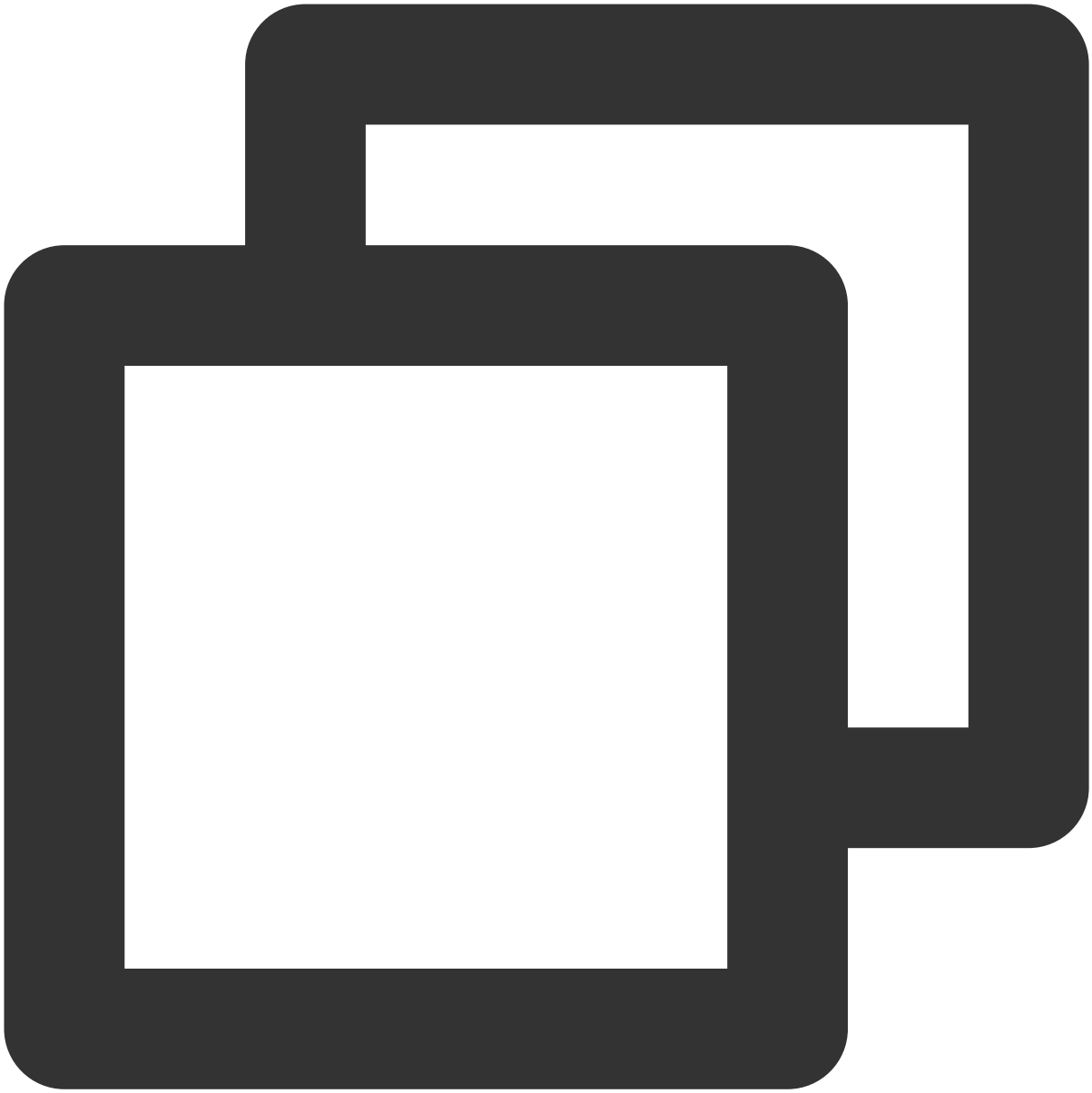
实际案例

请求



```
GET /doc_jobs/d13cfd584cd9011ea820b597ad1785a2f HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

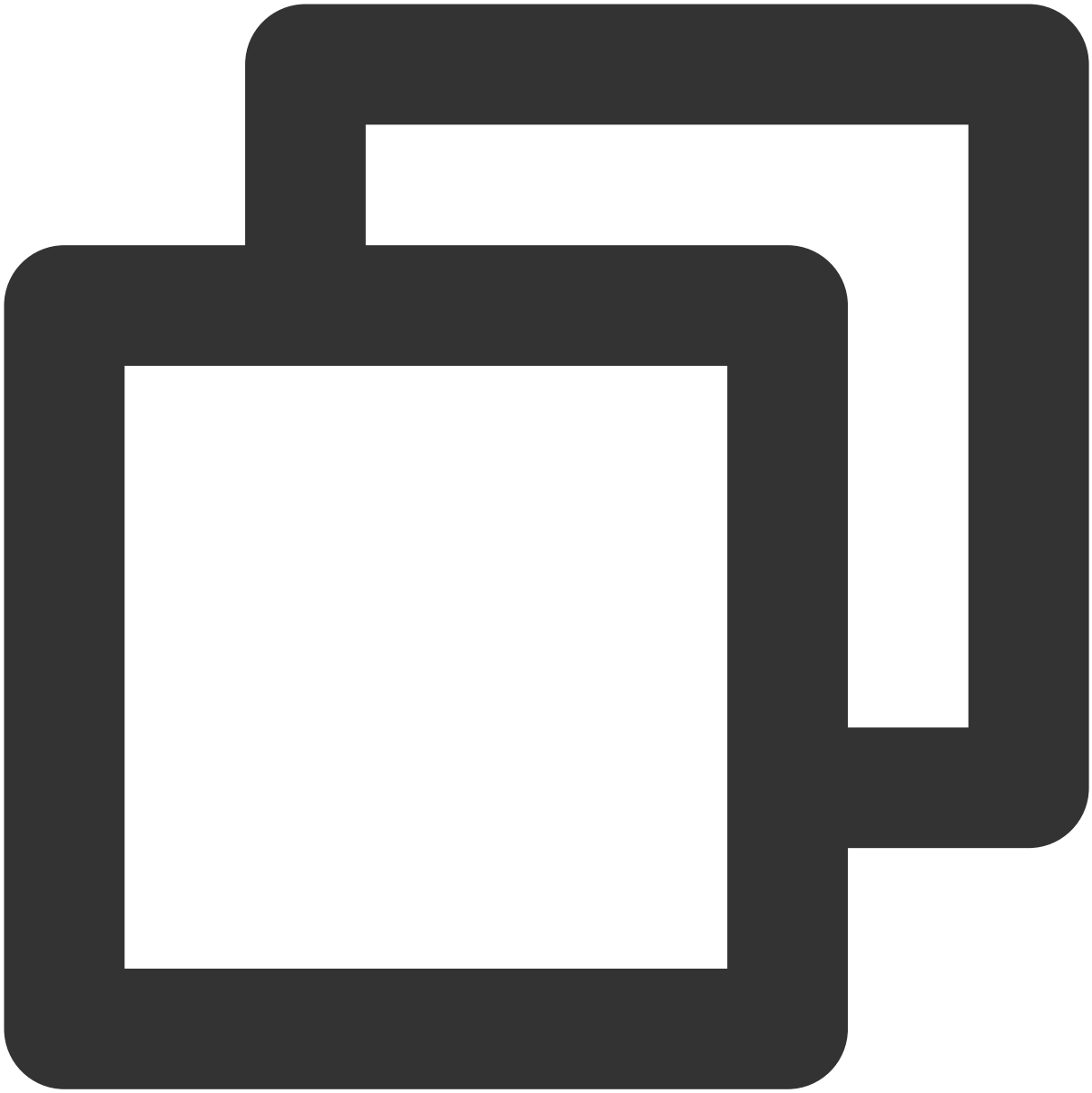
1.非 Excel 文档请求响应



```
<Response>
        <JobsDetail>
                <Code>Success</Code>
                <CreationTime>2020-07-24T17:28:47+0800</CreationTime>
                <EndTime>2020-07-24T17:28:49+0800</EndTime>
                <Input>
                        <Object>1.doc</Object>
                </Input>
                <JobId>d13cfd584cd9011ea820b597ad1785a2f</JobId>
                <Message/>
                <Operation>
```

```
                        <DocProcess>
                                <EndPage>0</EndPage>
                                <ImageParams/>
                                <SrcType/>
                                <StartPage>2</StartPage>
                                <TgtType>png</TgtType>
                        </DocProcess>
                        <DocProcessResult>
                                <FailPageCount>0</FailPageCount>
                                <PageInfo>
                                        <PageNo>2</PageNo>
                                        <TgtUri>big/test-1</TgtUri>
                                </PageInfo>
                                <SuccPageCount>1</SuccPageCount>
                                <TaskId/>
                                <TgtType/>
                                <TotalPageCount>2</TotalPageCount>
                        </DocProcessResult>
                        <Output>
                                <Bucket>examplebucket-1250000000</Bucket>
                                <Object>big/test-${Number}</Object>
                                <Region>ap-chongqing</Region>
                        </Output>
                </Operation>
                <QueueId>p50882922b848464fadd222d771438328</QueueId>
                <State>Success</State>
                <Tag>DocProcess</Tag>
        </JobsDetail>
</Response>
```

2. Excel 格式响应



```
<Response>
        <JobsDetail>
                <Code>Success</Code>
                <CreationTime>2020-12-03T19:54:10+0800</CreationTime>
                <EndTime>2020-12-03T19:54:13+0800</EndTime>
                <Input>
                        <Object>1.xlsx</Object>
                </Input>
                <JobId>d13cfd584cd9011ea820b597ad1785a2f</JobId>
                <Message/>
                <Operation>
```

```
                        <DocProcess>
                                <Comments>0</Comments>
                                <DocPassword/>
                                <EndPage>2</EndPage>
                                <ImageParams/>
                                <PaperDirection>0</PaperDirection>
                                <PaperSize>0</PaperSize>
                                <Quality>100</Quality>
                                <SheetId>0</SheetId>
                                <SrcType/>
                                <StartPage>1</StartPage>
                                <TgtType/>
                                <Zoom>100</Zoom>
                        </DocProcess>
                        <DocProcessResult>
                                <FailPageCount>0</FailPageCount>
                                <PageInfo>
                                        <PageNo>1</PageNo>
                                        <PicIndex>1</PicIndex>
                                        <PicNum>1</PicNum>
                                        <TgtUri>mark2/1/test-1.jpg</TgtUri>
                                        <X-SheetPics>2</X-SheetPics>
                                </PageInfo>
                                <PageInfo>
                                        <PageNo>1</PageNo>
                                        <PicIndex>2</PicIndex>
                                        <PicNum>2</PicNum>
                                        <TgtUri>mark2/1/test-2.jpg</TgtUri>
                                        <X-SheetPics>2</X-SheetPics>
                                </PageInfo>
                                <SuccPageCount>6</SuccPageCount>
                                <TaskId/>
                                <TgtType/>
                                <TotalPageCount>6</TotalPageCount>
                                <TotalSheetCount>3</TotalSheetCount>
                        </DocProcessResult>
                        <Output>
                                <Bucket>markjrzhang-1251704708</Bucket>
                                <Object>mark/${SheetID}/pic-${Page}.jpg</Object>
                                <Region>ap-chongqing</Region>
                        </Output>
                </Operation>
                <QueueId>p5fdbba9a9b83479f84538d5beb*****</QueueId>
                <State>Success</State>
                <Tag>DocProcess</Tag>
        </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

拉取符合条件的文档转码任务

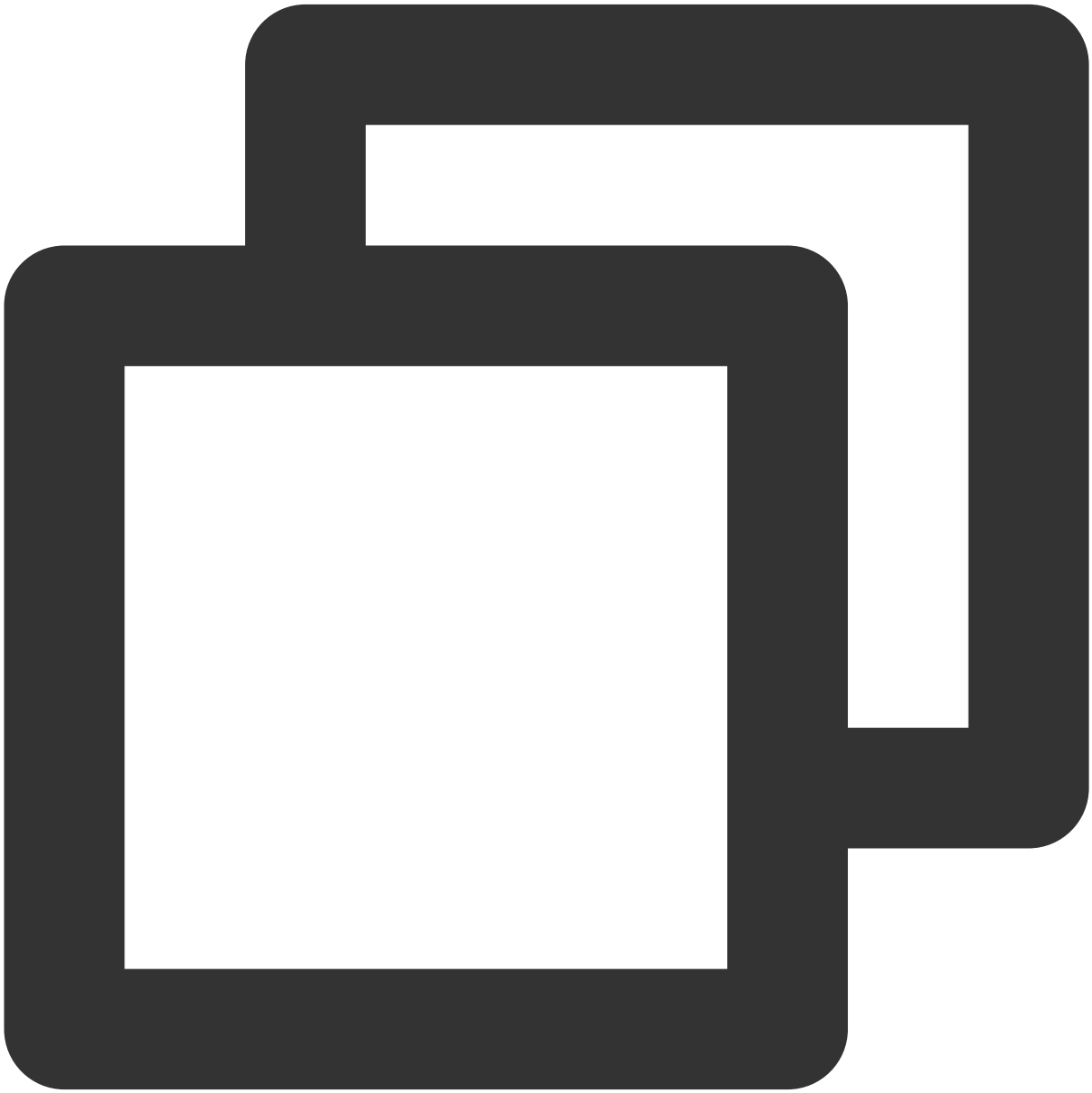
最近更新时间：2023-12-22 11:01:09

功能描述

DescribeDocProcessJobs 用于拉取符合条件（状态、创建时间等）的文档转码任务。

请求

请求示例



```
GET /doc_jobs?queueId=<queueId>&tag=DocProcess HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	是否必选
queueId	无	拉取该队列 ID 下的任务。	String	是
tag	无	任务的 Tag：DocProcess。	String	是
orderByTime	无	Desc 或者 Asc。默认为 Desc。	String	否
nextToken	无	请求的上下文，用于翻页。上次返回的值。	String	否
size	无	拉取的最大任务数。默认为10。最大为100。	Integer	否
states	无	拉取该状态的任务，以 `,` 分割，支持多状态：All、Submitted、Running、Success、Failed、Pause、Cancel。默认为 All。	String	否
startCreationTime	无	拉取创建时间大于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否
endCreationTime	无	拉取创建时间小于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否

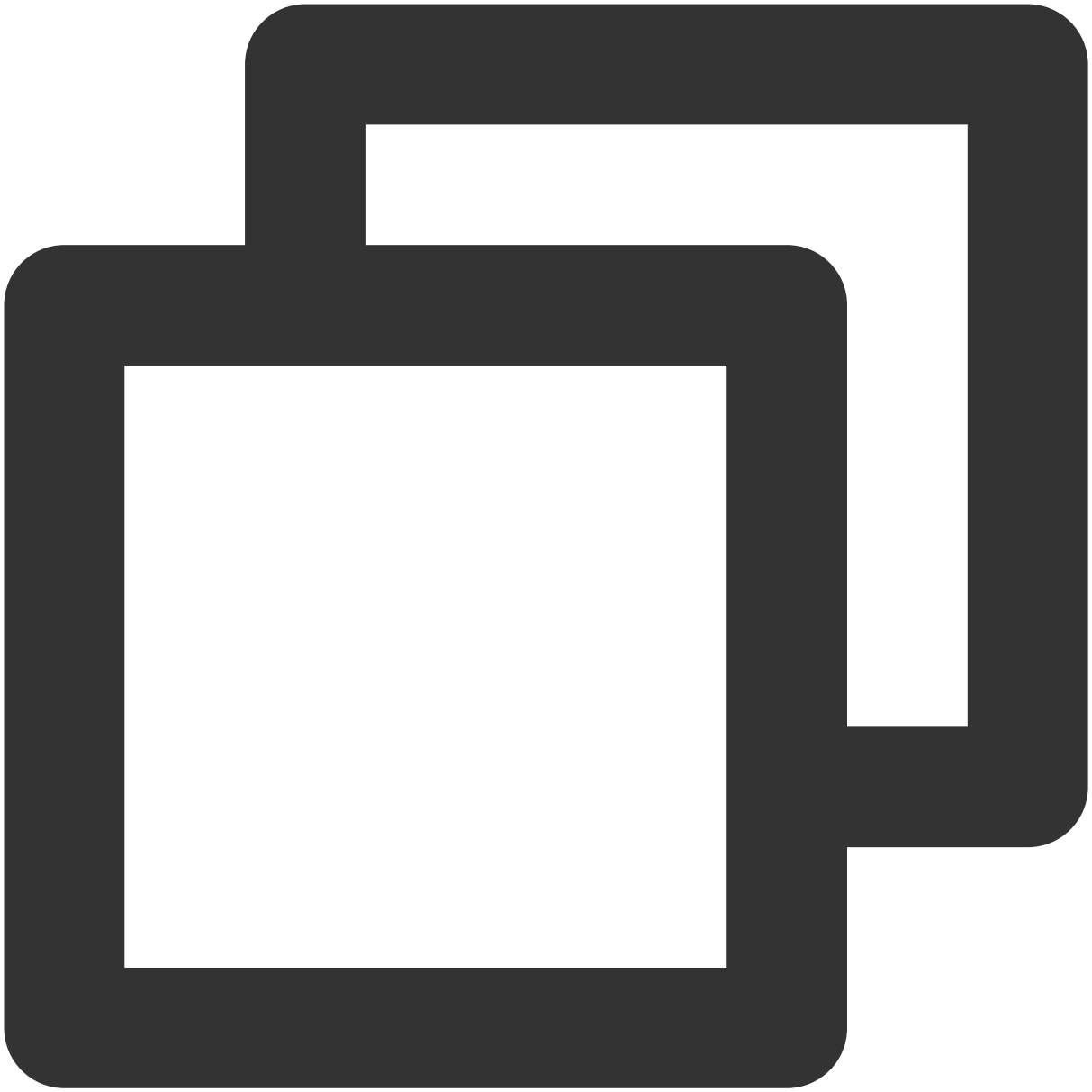
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail></JobsDetail>
  <NextToken></NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

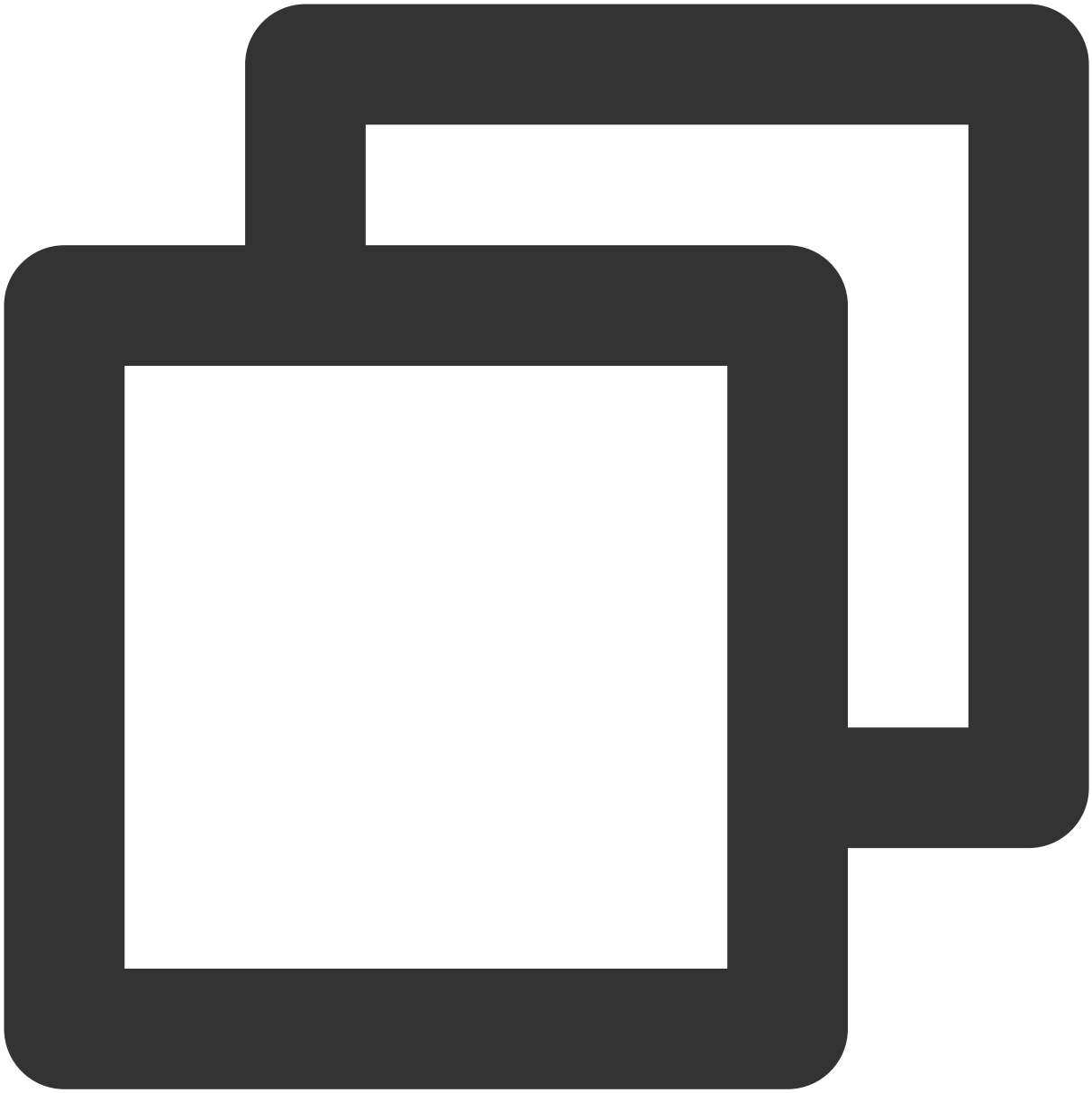
节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息，同 CreateDocProcessJobs 接口中的 Response.JobsDetail 节点	Container
NextToken	Response	翻页的上下文 Token	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

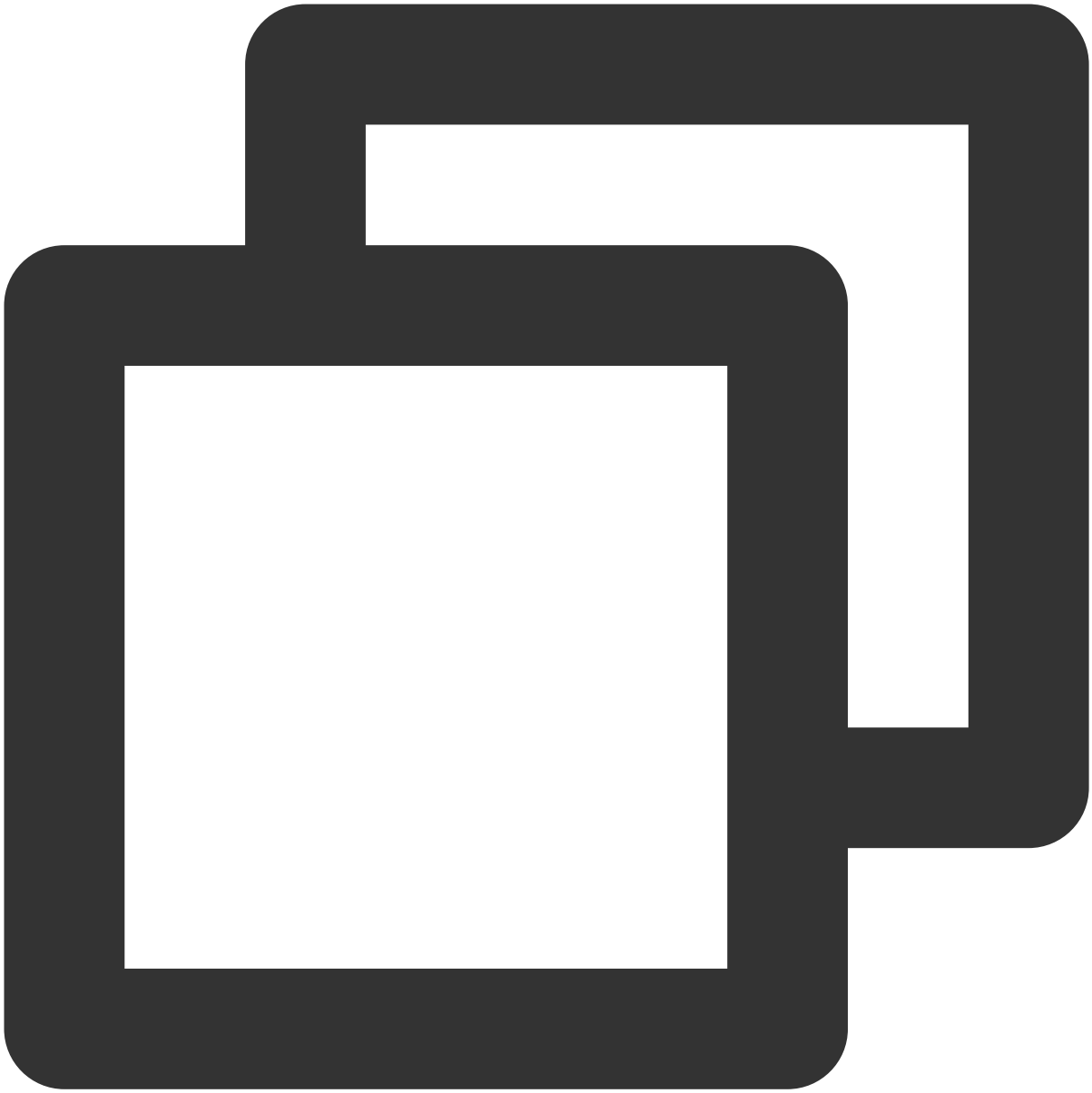
实际案例

请求



```
GET /doc_jobs/queueId=QueueId&tag=DocProcess HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
<Response>
        <JobsDetail>
                <Code>Success</Code>
                <CreationTime>2020-07-20T10:43:17+0800</CreationTime>
                <EndTime>-</EndTime>
                <Input>
                        <Object>1.docx</Object>
                </Input>
                <JobId>JobId</JobId>
                <Message/>
                <Operation>
```

```
                        <DocProcess>
                                <EndPage>1</EndPage>
                                <ImageParams>ImageParams</ImageParams>
                                <SrcType/>
                                <StartPage>1</StartPage>
                                <TgtType>png</TgtType>
                        </DocProcess>
                        <Output>
                                <Bucket>BucketId</Bucket>
                                <Object>file/test-${Page}.jpg</Object>
                                <Region>Region</Region>
                        </Output>
                </Operation>
                <QueueId>QueueId</QueueId>
                <State>Submitted</State>
                <Tag>DocProcess</Tag>
        </JobsDetail>
        <JobsDetail>
                <Code>Success</Code>
                <CreationTime>2020-07-20T10:43:17+0800</CreationTime>
                <EndTime>-</EndTime>
                <Input>
                        <Object>test.docx</Object>
                </Input>
                <JobId>JobId</JobId>
                <Message/>
                <Operation>
                        <DocProcess>
                                <EndPage>1</EndPage>
                                <ImageParams>ImageParams</ImageParams>
                                <SrcType/>
                                <StartPage>1</StartPage>
                                <TgtType>png</TgtType>
                        </DocProcess>
                        <Output>
                                <Bucket>BucketId</Bucket>
                                <Object>file/test-${Page}.jpg</Object>
                                <Region>Region</Region>
                        </Output>
                </Operation>
                <QueueId>QueueId</QueueId>
                <State>Submitted</State>
                <Tag>DocProcess</Tag>
        </JobsDetail>
        <NextToken>19193</NextToken>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

异步处理队列接口
查询文档转码队列

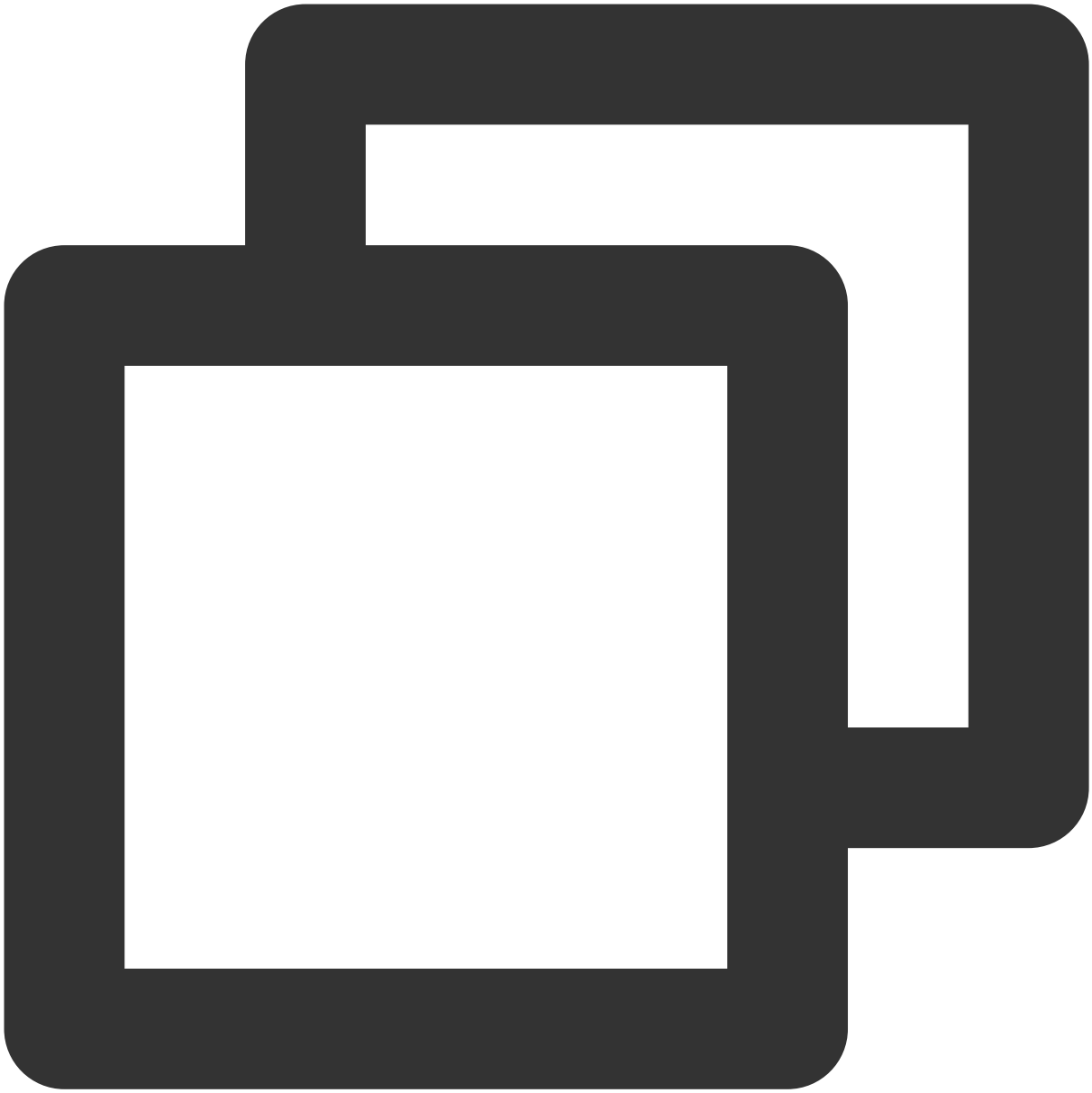
最近更新时间：2023-12-22 11:42:07

功能描述

DescribeDocProcessQueues 接口用于查询文档转码队列。

请求

请求示例



```
GET /docqueue?pageNumber=1&pageSize=2 HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

名称	描述	类型	是否必选
queueIds	队列 ID，以“,”符号分割字符串	string	否
state	1. Active 表示队列内的作业会被文档预览服务调度执行 2. Paused 表示队列暂停，作业不再会被文档预览服务调度执行，队列内的所有作业状态维持在暂停状态，已经处于执行中的任务将继续执行，不受影响	string	否
pageNumber	第几页，默认第一页	string	否
pageSize	每页个数，默认10个	string	否

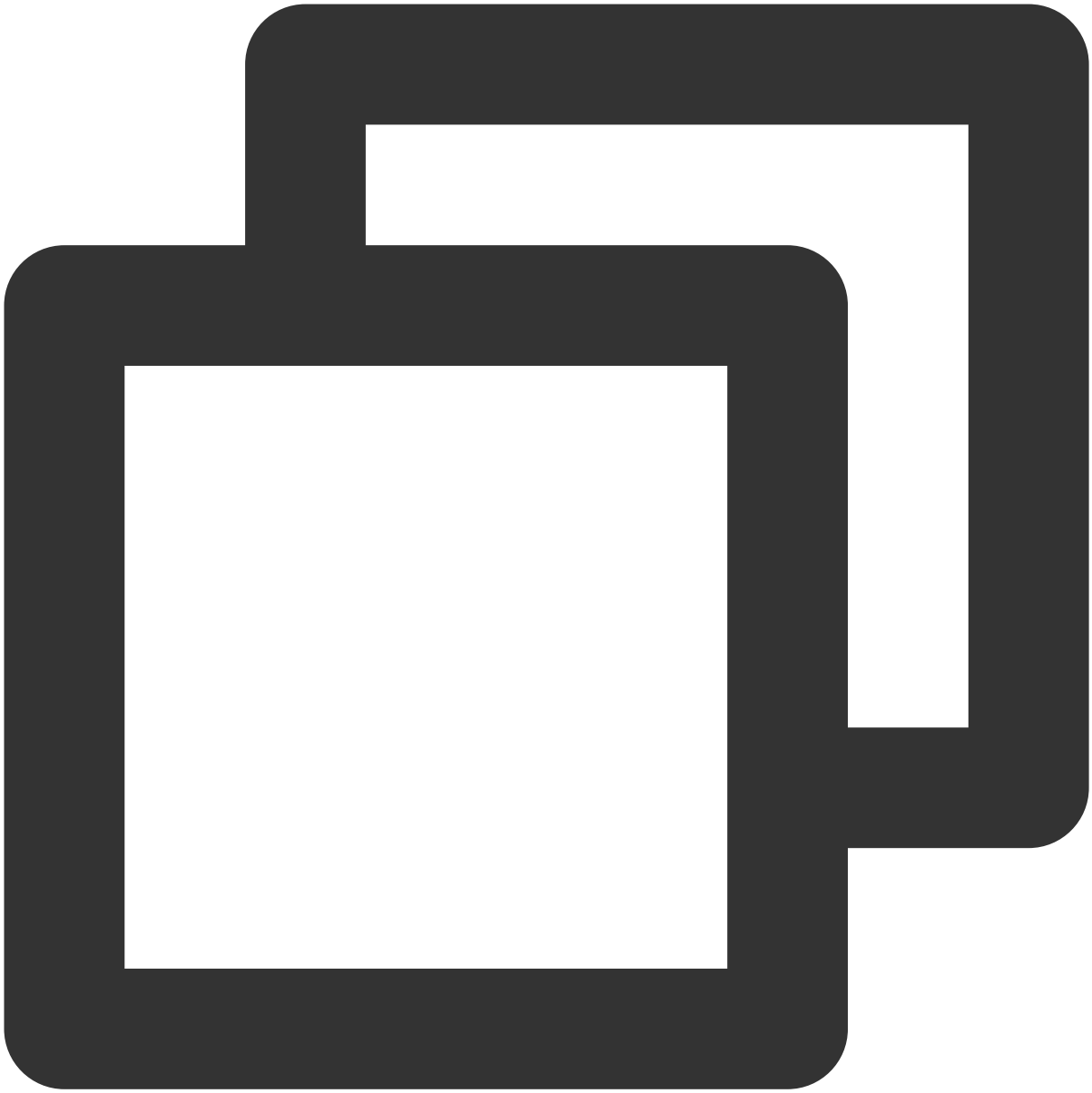
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据。
响应体1：当指定 pageNumber 和 pageSize 时包含完整节点数据的内容展示如下。



```
<Response>
        <TotalCount>1</TotalCount>
        <RequestId>RequestId</RequestId>
        <PageNumber>1</PageNumber>
        <PageSize>2</PageSize>
        <QueueList>
                <QueueId>QueueId</QueueId>
                <Name>QueueName</Name>
                <State>Active</State>
                <NotifyConfig>
                        <Url>url</Url>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                        <Event>Event</Event>
                        <Type/></Type>
                        <State>Off</State>
                </NotifyConfig>
                <MaxSize>10000</MaxSize>
                <MaxConcurrent>10</MaxConcurrent>
                <CreateTime>CreateTime</CreateTime>
                <UpdateTime>UpdateTime</UpdateTime>
                <BucketId>BucketId</BucketId>
                <Category>DocProcess</Category>
        </QueueList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	队列总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
QueueList	Response	队列数组	Container

Container 节点 QueueList 的内容：

节点名称（关键字）	父节点	描述	类型
QueueId	Response.QueueList	队列 ID	String
Name	Response.QueueList	队列名字	String
State	Response.QueueList	当前状态，Active 或者 Paused	String
NotifyConfig	Response.QueueList	回调配置	Container
MaxSize	Response.QueueList	队列最大长度	Int

版权所有：腾讯云计算（北京）有限责任公司

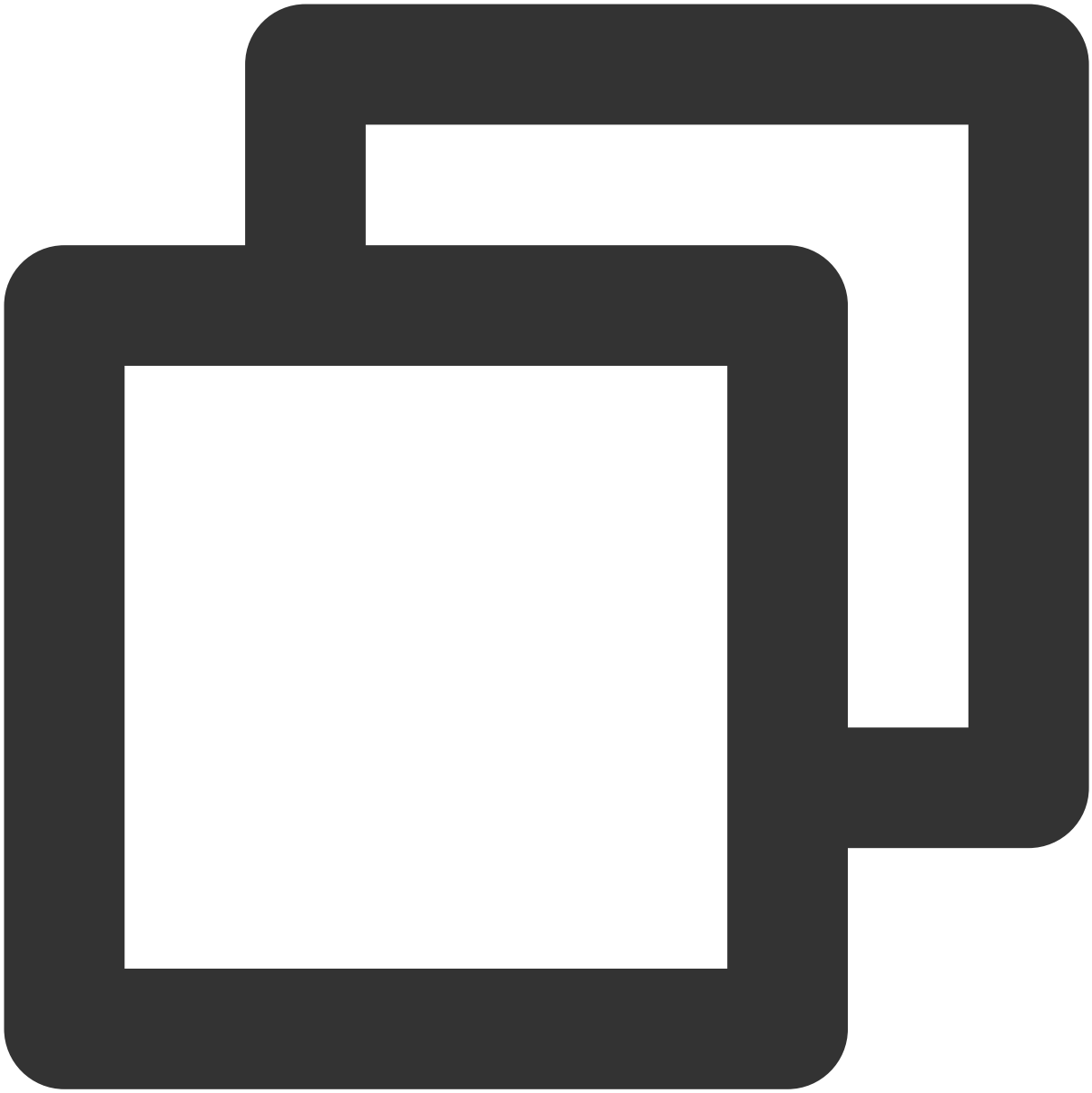
MaxConcurrent	Response.QueueList	当前队列最大并行执行的任务数	Int
UpdateTime	Response.QueueList	更新时间	String
CreateTime	Response.QueueList	创建时间	String

Container 节点 NotifyConfig 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.QueueList.NotifyConfig	回调地址	String
State	Response.QueueList.NotifyConfig	开关状态，On 或者 Off	String
Type	Response.QueueList.NotifyConfig	回调类型，Url	String
Event	Response.QueueList.NotifyConfig	触发回调的事件	String

响应体2：当指定 queueIds 时包含完整节点数据的内容展示如下。

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <QueueList>
        <QueueId>A</QueueId>
        <Name></Name>
        <State>Active</State>
        <NotifyConfig>
            <Url></Url>
        </NotifyConfig>
    </QueueList>
    <NonExistPIDs>
```

```
        <QueueID>B</QueueID>
        <QueueID>C</QueueID>
    </NonExistPIDs>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
QueueList	Response	队列数组，同上面解释的 QueueList	Container
NonExistPIDs	Response	不存在的队列 ID 列表	Container

Container 节点 NonExistPIDs 的内容：

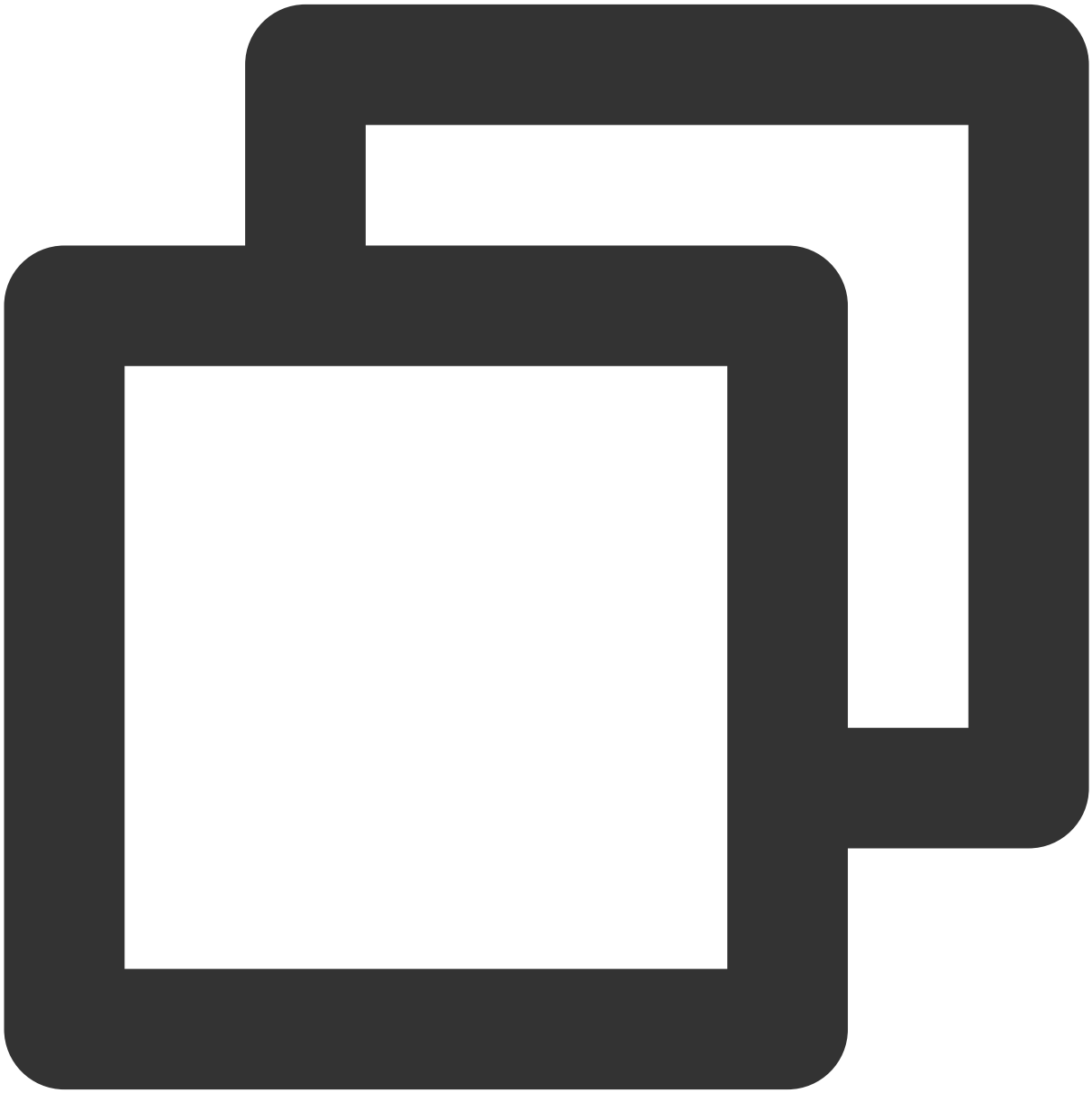
节点名称（关键字）	父节点	描述	类型
QueueID	Response.NonExistPIDs	队列 ID	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

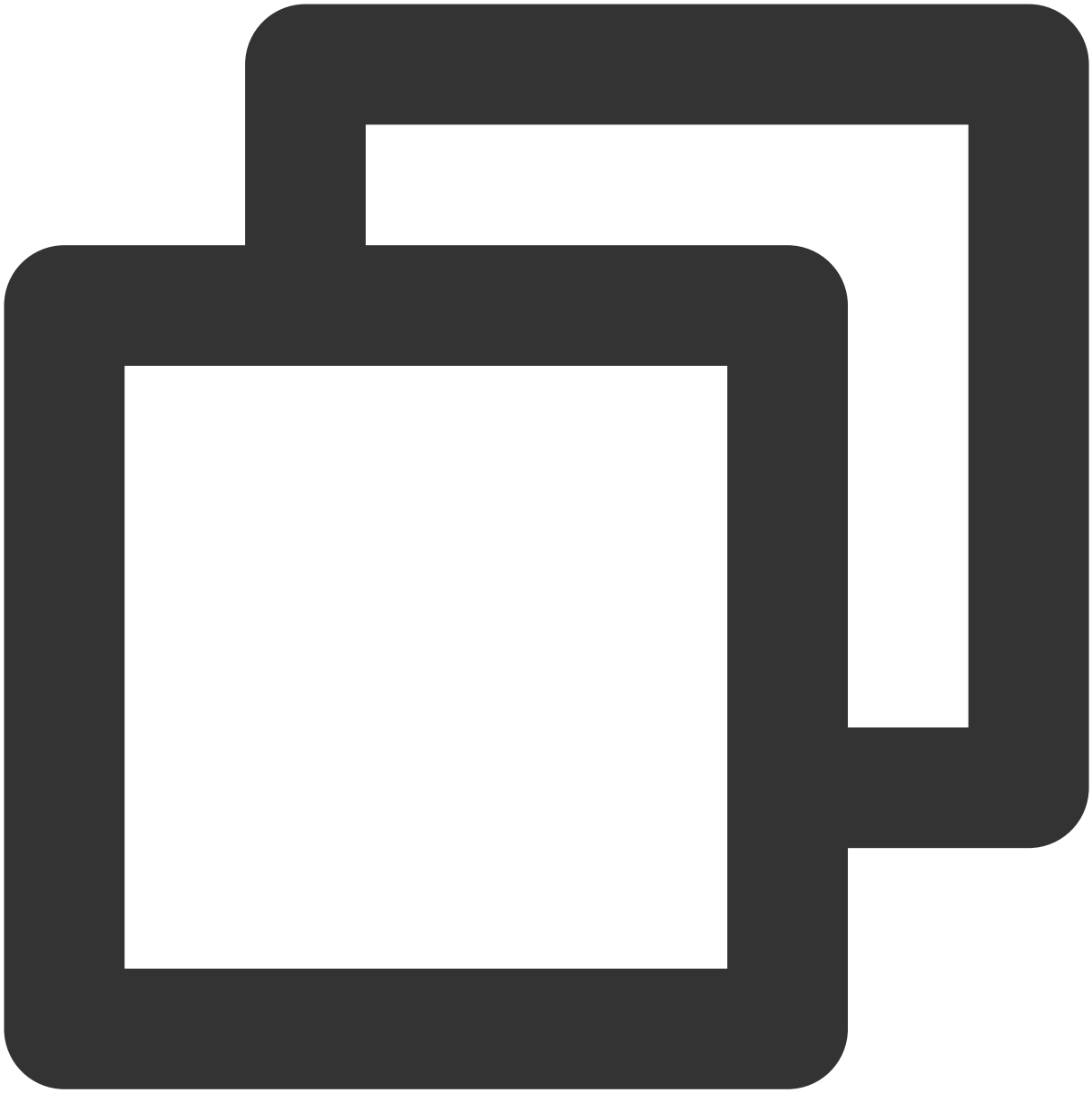
实际案例

请求



```
GET /docqueue?pageNumber=1&pageSize=2 HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: examplebucket-1250000000.ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Authorization: Authorization
```

响应



```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 07:29:28 GMT
Content-Type: application/xml
Content-Length: 677
Connection: keep-alive
Server: tencent-ci
x-ci-request-id: NWYxZTgyNTdfYzc2OTQzNjRfMzUx****

<?xml version="1.0" encoding="utf-8"?>
<Response>
        <TotalCount>1</TotalCount>
```

```
        <RequestId>NWYxZTgyNTdfYzc2OTQzNjRfMzUx****</RequestId>
        <PageNumber>1</PageNumber>
        <PageSize>2</PageSize>
        <QueueList>
                <QueueId>p2505d57bdf4c4329804b58a6a5fb1572</QueueId>
                <Name>queue-doc-process-1</Name>
                <State>Active</State>
                <NotifyConfig>
                        <Url/>
                        <Event/>
                        <Type/>
                        <State>Off</State>
                </NotifyConfig>
                <MaxSize>10000</MaxSize>
                <MaxConcurrent>10</MaxConcurrent>
                <CreateTime>2020-07-24T22:42:27+0800</CreateTime>
                <UpdateTime>2020-07-24T22:42:27+0800</UpdateTime>
                <BucketId>examplebucket-1250000000.</BucketId>
                <Category>DocProcessing</Category>
        </QueueList>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新文档转码队列

最近更新时间：2022-10-26 15:23:04

功能描述

UpdateDocProcessQueue 接口用于更新文档转码队列。

请求

请求示例

```
PUT /docqueue/p8eb46b8cc1a94bc09512d16c5c4f4d3a HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
<body>
```

说明：

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有以下请求体。

```
<Request>
<Name>Queue Name</Name>
<QueueID></QueueID>
<State></State>
<NotifyConfig>
<Type></Type>
<Url></Url>
```

```
<Event></Event>
</NotifyConfig>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Name	Request	队列名称	String	是	长度限制100字符
QueueID	Request	队列ID	String	是	无
State	Request	队列状态	String	是	1. Active 表示队列内的作业会被文档预览服务调度转码执行 2. Paused 表示队列暂停，作业不再会被文档预览服务调度执行，队列内的所有作业状态维持在已提交状态，已经处于执行中的任务将继续执行，不受影响
NotifyConfig	Request	通知渠道	Container	是	第三方回调 Url

Container 类型 NotifyConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Url	Request.NotifyConfig	回调配置	String	否	长度限制100字符
Type	Request.NotifyConfig	回调类型，普通回调：Url	String	否	长度限制100字符
Event	Request.NotifyConfig	回调事件，文档预览任务完成	String	否	长度限制100字符
State	Request.NotifyConfig	回调开关，Off，On	String	否	长度限制100字符

版权所有：腾讯云计算（北京）有限责任公司

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

```
<Response>
<RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
<Queue>
<QueueId></QueueId>
<Name></Name>
<State>Active</State>
<NotifyConfig>
<Url>mts-topic-1</Url>
<Type></Type>
<Event></Event>
</NotifyConfig>
<CreateTime></CreateTime>
<UpdateTime></UpdateTime>
</Queue>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
Queue	Response	队列信息，详情请参见 DescribeDocProcessQueues 中的 Response.QueueList	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
PUT /docqueue/p2505d57bdf4c4329804b58a6a5fb1572 HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: examplebucket-1250000000.ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Content-Length: 279
Authorization: Authorization
<?xml version="1.0" encoding="UTF-8" ?>
<Request>
<QueueId>p2505d57bdf4c4329804b58a6a5fb1572</QueueId>
<State>Active</State>
<Name>markjrzhang4</Name>
<NotifyConfig>
<Url>http://google.com/</Url>
<State>On</State>
<Type>Url</Type>
<Event>TransCodingFinish</Event>
</NotifyConfig>
</Request>[!http]
```

响应

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 08:22:41 GMT
Content-Type: application/xml
Content-Length: 641
Connection: keep-alive
Server: tencent-ci
x-ci-request-id: NWYxZThlZDBfYzc2OTQzNjRfMzUxYV8x****
<?xml version="1.0" encoding="utf-8"?>
<Response>
<RequestId>NWYxZThlZDBfYzc2OTQzNjRfMzUxYV8x****</RequestId>
<Queue>
<QueueId>p2505d57bdf4c4329804b58a6a5fb1572</QueueId>
<Name>markjrzhang4</Name>
<State>Active</State>
```

版权所有：腾讯云计算（北京）有限责任公司

```
<NotifyConfig>
<Url>http://example.com/</Url>
<Event>TransCodingFinish</Event>
<Type>Url</Type>
<State>On</State>
</NotifyConfig>
<MaxSize>10000</MaxSize>
<MaxConcurrent>10</MaxConcurrent>
<CreateTime>2020-07-24T22:42:27+0800</CreateTime>
<UpdateTime>2020-07-27T16:22:40+0800</UpdateTime>
<BucketId>test007-1251704708</BucketId>
<Category>DocProcessing</Category>
</Queue>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

支持的功能和效果

最近更新时间：2023-12-22 11:42:47

以下是文档转图片功能支持的功能和效果：

类型	文件名后缀	支持的功能和效果
演示	ppt、pptx、pptm、ppsx、ppsm、pps、potx、potm、dpt、dps	幻灯片背景、对象边框/填充、对象阴影/倒影、文本（横/竖向文本框、占位符、形状）、图片、链接、艺术字、形状、表格、图片、图表、公式、符号、链接、脑图、流程图
文字	doc、dot、wps、wpt、docx、dotx、docm、dotm	字属性、段属性、图片、文字表、链接、形状、代码块、日期、水印、评论、页眉页脚
表格	xls、 xlt、et、xlsx、xltx、xlsm、xltm	字属性、段属性、自选图形、艺术字、公式、图片、流程图、脑图
PDF	pdf	背景、文字（字体、字号、字色、加粗、斜体）、画笔、高亮、下划线、删除线

版权所有：腾讯云计算（北京）有限责任公司

文档转 HTML
快速入门

最近更新时间：2023-12-22 11:42:53

功能描述

文档转 HTML 功能支持对多种文档类型的文件生成 HTML 格式预览，满足 PC、App 等多个用户端的文档在线浏览需求，适用于在线教育、企业 OA、在线知识库、网盘文档预览等业务场景。

说明

对象存储封装预览 JS SDK，只需要一条 URL 即可接入 HTML 预览服务。

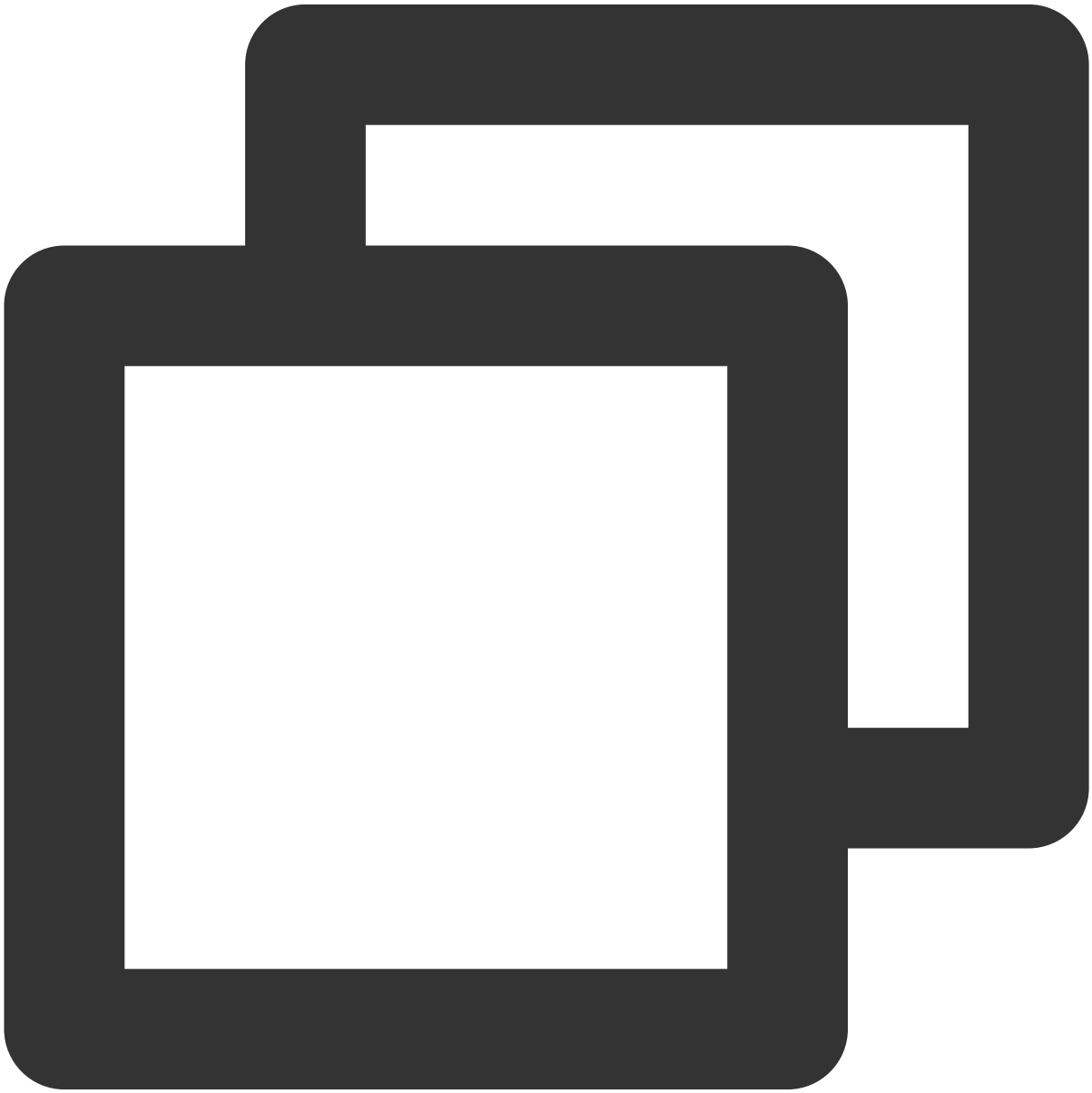
如需在小程序内使用，请参考小程序配置业务域名。

如需接入 CDN 域名，需要打开 CDN 的**高级缓存过期设置**，详情请参见 节点缓存过期配置 CDN 文档。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?ci-process=doc-preview&dstType=html HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
```

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

名称	描述	类型	是否必

			选
ObjectKey	对象文件名，例如 folder/document.pdf	String	是
ci-process	对象存储处理能力，文档 HTML 预览固定为 doc-preview	String	是
dstType	转换输出目标文件类型，文档 HTML 预览固定为 html（需为小写字母）	String	是
srcType	指定目标文件类型，支持的文件类型请见下方	String	否
sign	对象下载签名，如果预览的对象为私有读时，需要传入签名，详情请参见 请求签名 文档（注意：需要进行 urlencode）	String	否
copyable	是否可复制。默认为可复制，填入值为1；不可复制，填入值为0	String	否
htmlParams	自定义配置参数，JSON 结构，需要经过 URL 安全 的 Base64 编码，默认配置为：{ commonOptions: { isShowTopArea: true, isShowHeader: true } }，支持的配置参考 自定义配置项说明	String	否
htmlwaterword	水印文字，需要经过 URL 安全 的 Base64 编码，默认为空	String	否
htmlfillstyle	水印 RGBA（颜色和透明度），需要经过 URL 安全 的 Base64 编码，默认为：rgba(192,192,192,0.6)	String	否
htmlfront	水印文字样式，需要经过 URL 安全 的 Base64 编码，默认为：bold 20px Serif	String	否
htmlrotate	水印文字旋转角度，0 - 360，默认315度	String	否
htmlhorizontal	水印文字水平间距，单位 px，默认为50	String	否
htmlvertical	水印文字垂直间距，单位 px，默认为100	String	否

注意

目前支持的输入文件类型包含如下格式：

演示文件：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。

文字文件：doc、dot、wps、wpt、docx、dotx、docm、dotm。

表格文件：xls、xlt、et、ett、xlsx、xltx、csv、xlsb、xlsm、xltm、ets。

其他格式文件：pdf、lrc、c、cpp、h、asm、s、java、asp、bat、bas、prg、cmd、rtf、txt、log、xml、htm、html。

输入文件大小限制在200MB之内。

输入文件页数限制在5000页之内。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口请求的响应体为html预览内容，展示效果可参见下方实际案例。

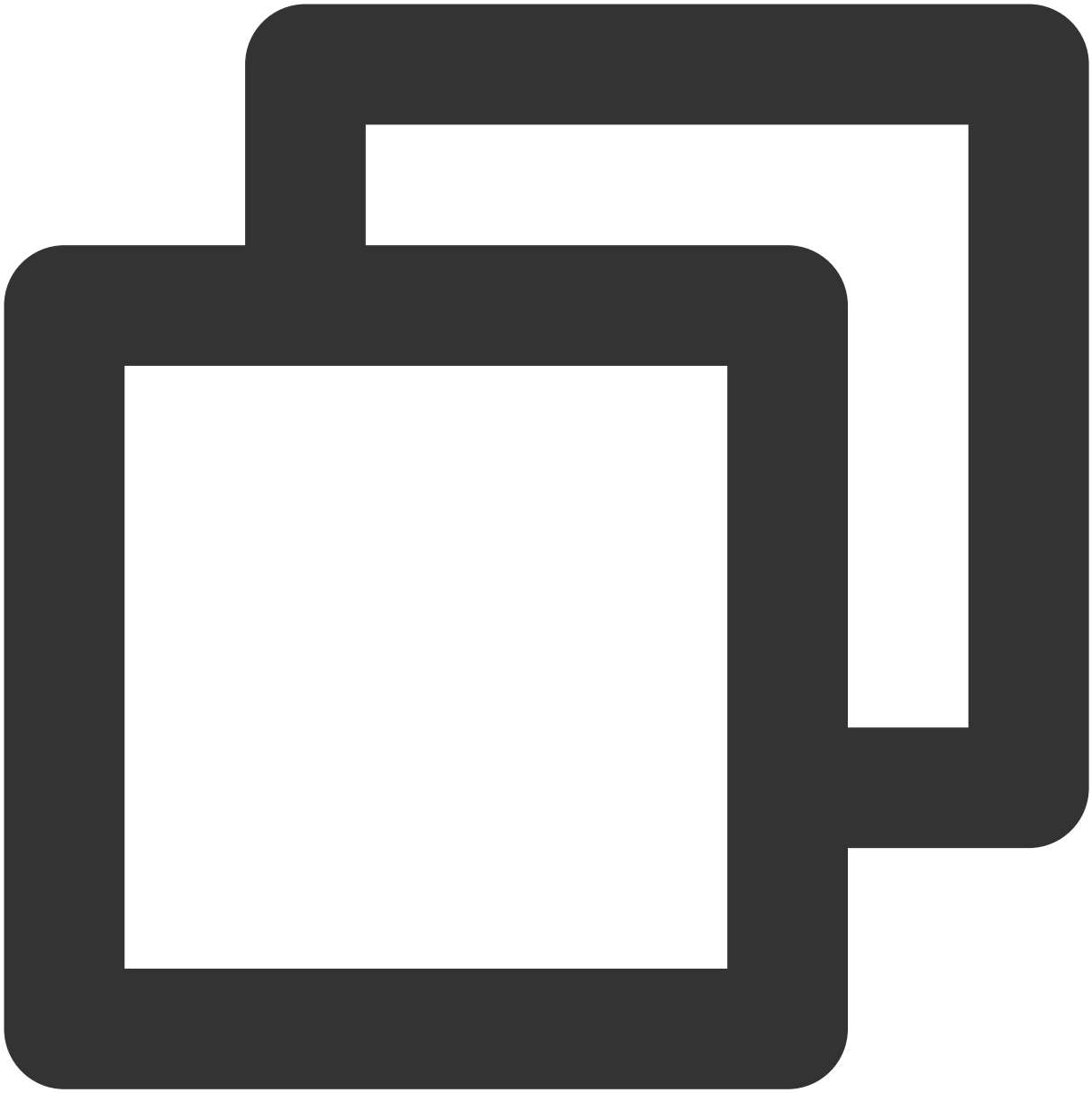
错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

案例一：简单预览

将存储桶 examples-1258125638 中的 ObjectKey 为 test.pptx 的 PPT 文件进行在线预览，最终 URL 如下：



```
https://examples-1258125638.cos.ap-guangzhou.myqcloud.com/test.pptx?ci-process=doc-
```

说明

您还可通过 JS-SDK 实现更多功能的定制，详情请参见 自定义配置概述。

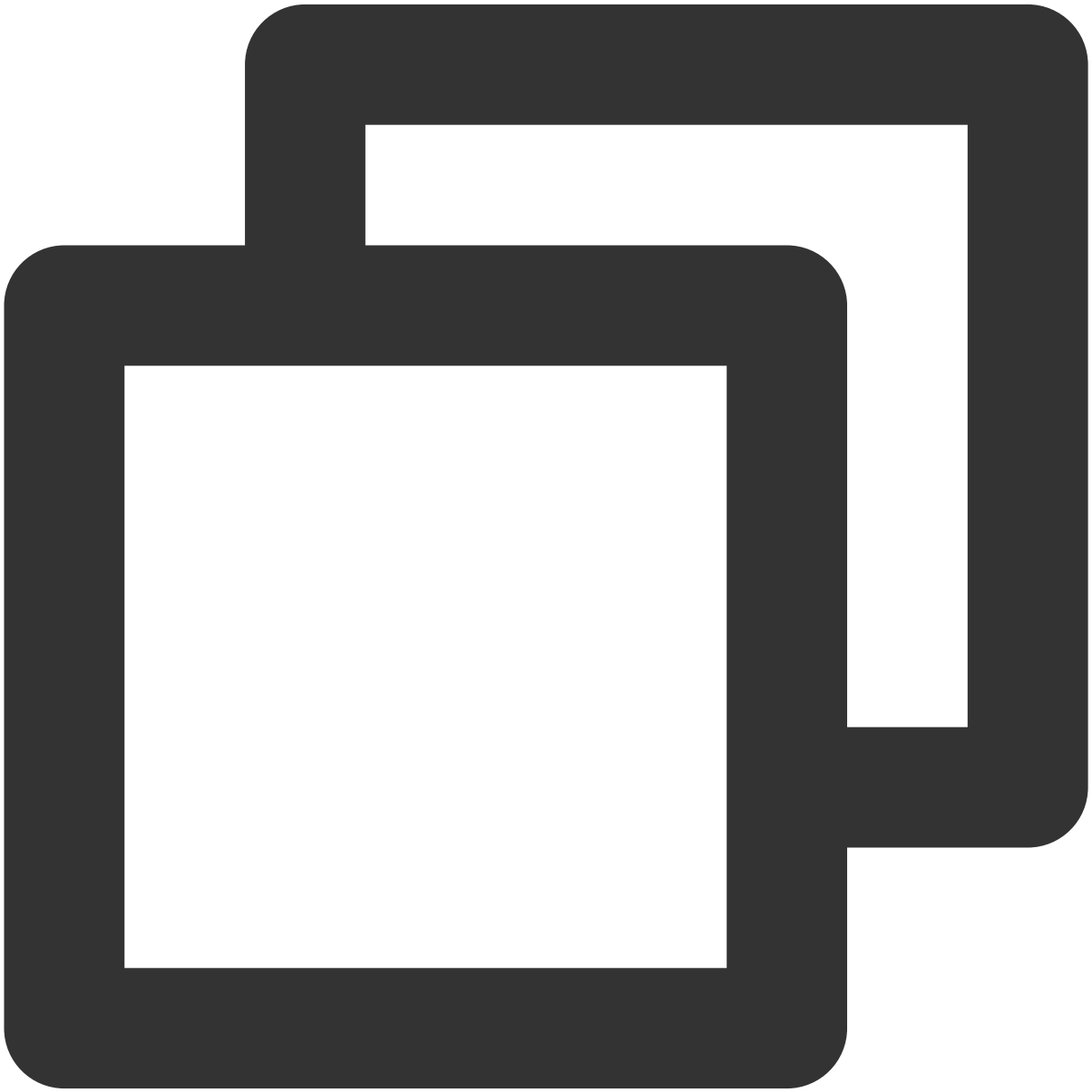
直接在浏览器中打开链接，生成的预览页面如下图所示，对于幻灯片文档，还可进行全屏的幻灯片放映，支持**动画、触发器**等效果，并提供**虚拟激光笔、演讲者模式**等功能。

案例二：水印 + 自定义配置预览

将存储桶 examples-1258125638 中的 ObjectKey 为 test.pptx 的 PPT 文件进行在线预览。

1. 设置**不可复制**（copyable=0）
2. 设置**test** 水印（通过 URL 安全的 Base64编码得到 htmlwaterword=dGVzdAog）
3. 设置**自定义配置参数**（htmlParams为 {"mode":"normal","commonOptions":{"isShowHeader":false,"isShowTopArea":true},"pptOptions":{"isSlidesStatusPlay": true}}，通过 URL 安全的 Base64编码得到 htmlParams=eyJtb2RlIjoibm9ybWFsIiwiY29tbW9uT3B0aW9ucyI6eyJpc1Nob3dIZWFkZXIiOmZhbHNlLCJpc1Nob3dUb3BBcmVhIjp0cnVlfSwicHB0T3B0aW9ucyI6eyJpc1NsaWRlc1N0YXR1c1BsYXkiOiB0cnVlfX0=）。

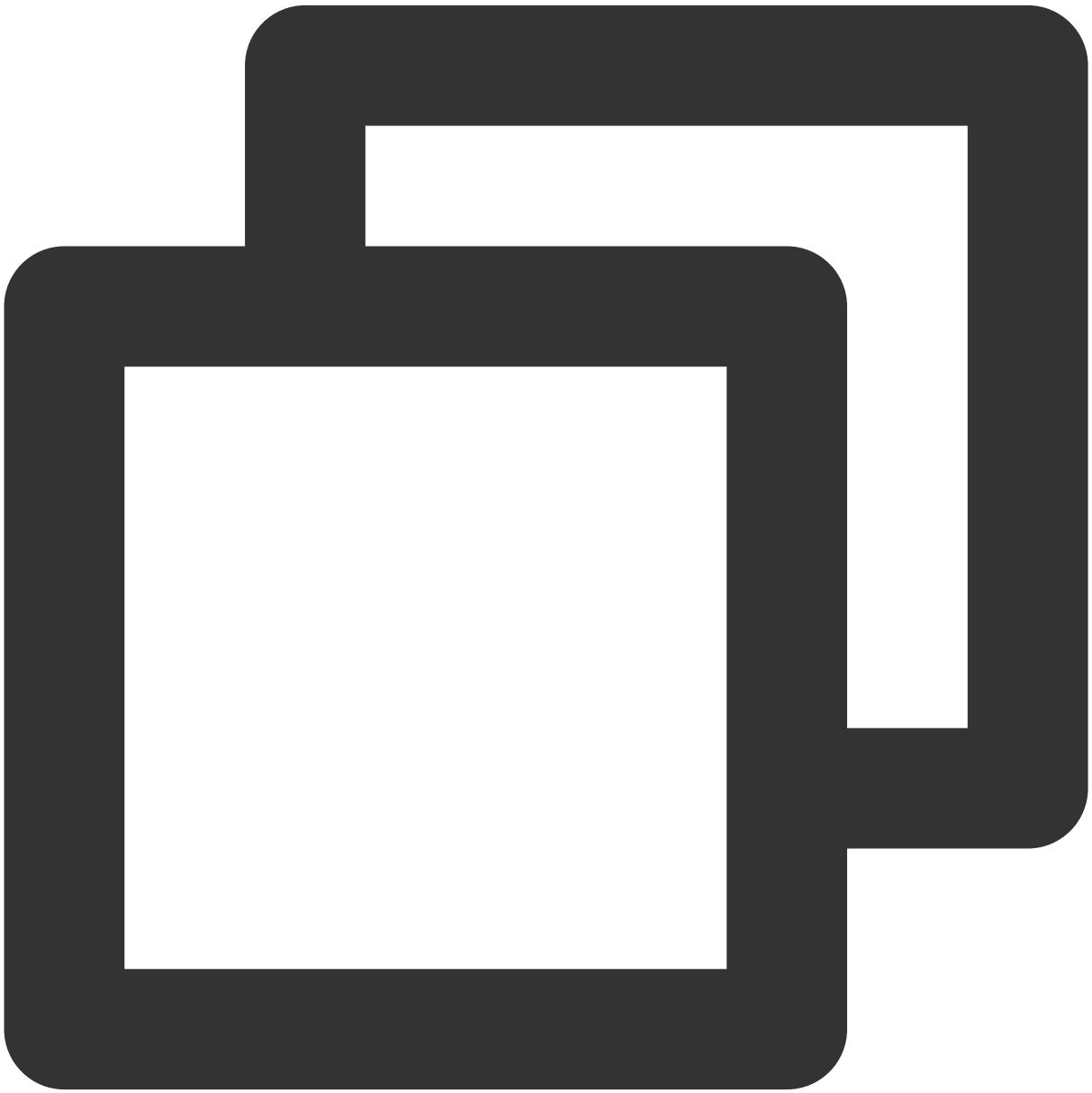
则最终 URL 如下：



```
https://examples-1258125638.cos.ap-guangzhou.myqcloud.com/test.pptx?ci-process=doc-
```

直接在浏览器中打开链接，生成的预览页面如下图所示，对于幻灯片文档，还可进行全屏的幻灯片放映，支持**动画、触发器**等效果，并提供**虚拟激光笔、演讲者模式**等功能。

预览链接也可以作为iframe嵌入到业务页面中，如下所示：



```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport"
```

```
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">
    <meta name="keywords" content="对象存储,文档预览">
    <meta name="description" content="">
    <title>腾讯云-对象存储-文档预览</title>
    <style>
        html,
        body {
            padding: 0;
            margin: 0;
            height: 100%;
            touch-action: manipulation;
        }
    </style>
</head>
<body>
    <iframe src="https://examples-1258125638.cos.ap-guangzhou.myqcloud.com/test.ppt
</body>
</html>
```

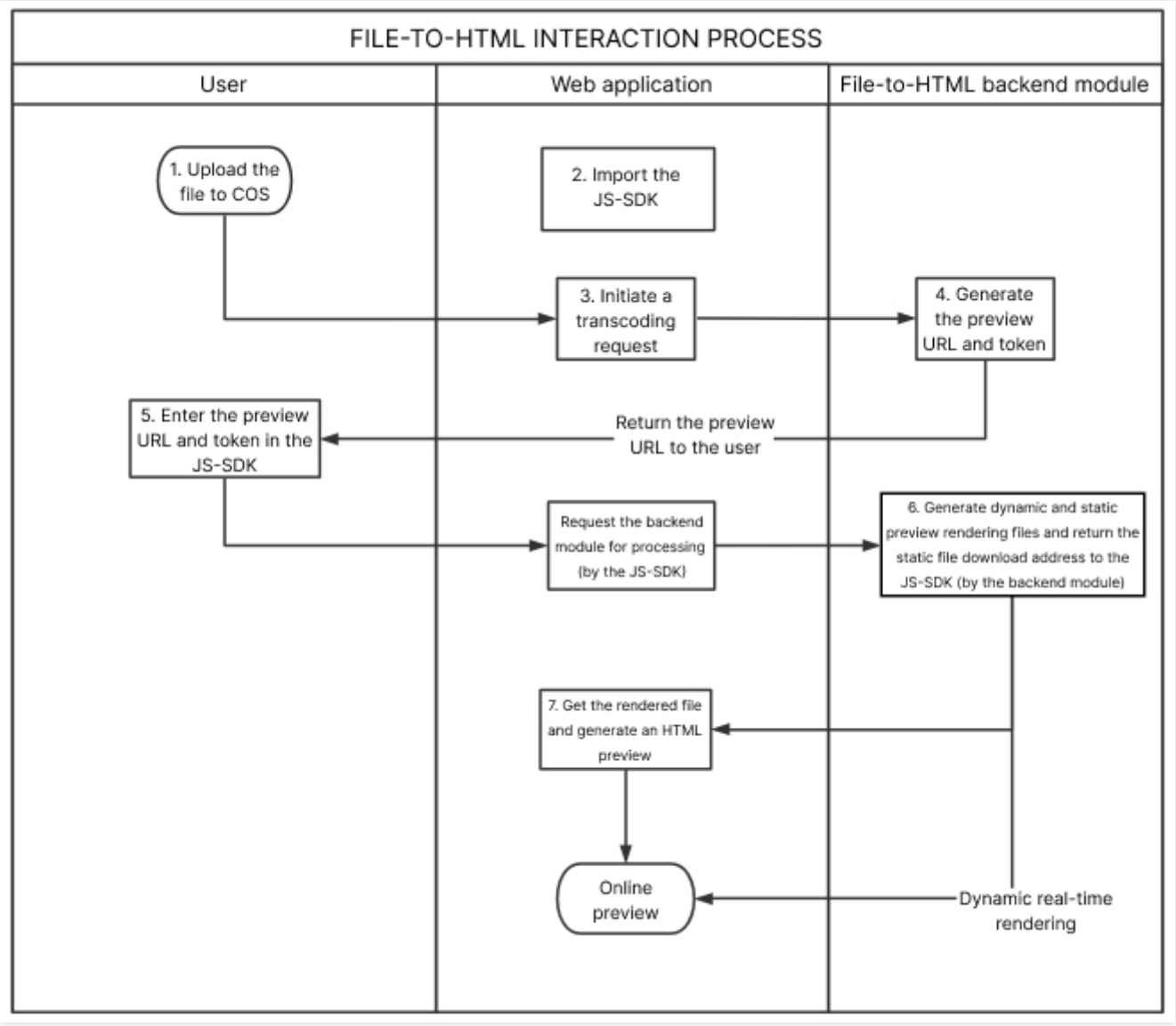
自定义配置概述

最近更新时间：2023-12-22 11:43:16

功能说明

通过使用 JS-SDK，网页开发者可以自定义界面显示、页面状态、监听事件等能力，为用户提供更优质的网页体验。

架构图



使用环境

平台	支持浏览器
iOS	Safari、QQ 内置浏览器、QQ 小程序、微信内置浏览器、微信小程序
Android	QQ 内置浏览器、QQ 小程序、微信内置浏览器、微信小程序
Windows	Chrome、QQ 浏览器（非兼容模式）、EDGE、火狐、IE11
Mac OS X	Chrome、Safari、QQ 浏览器、EDGE、火狐

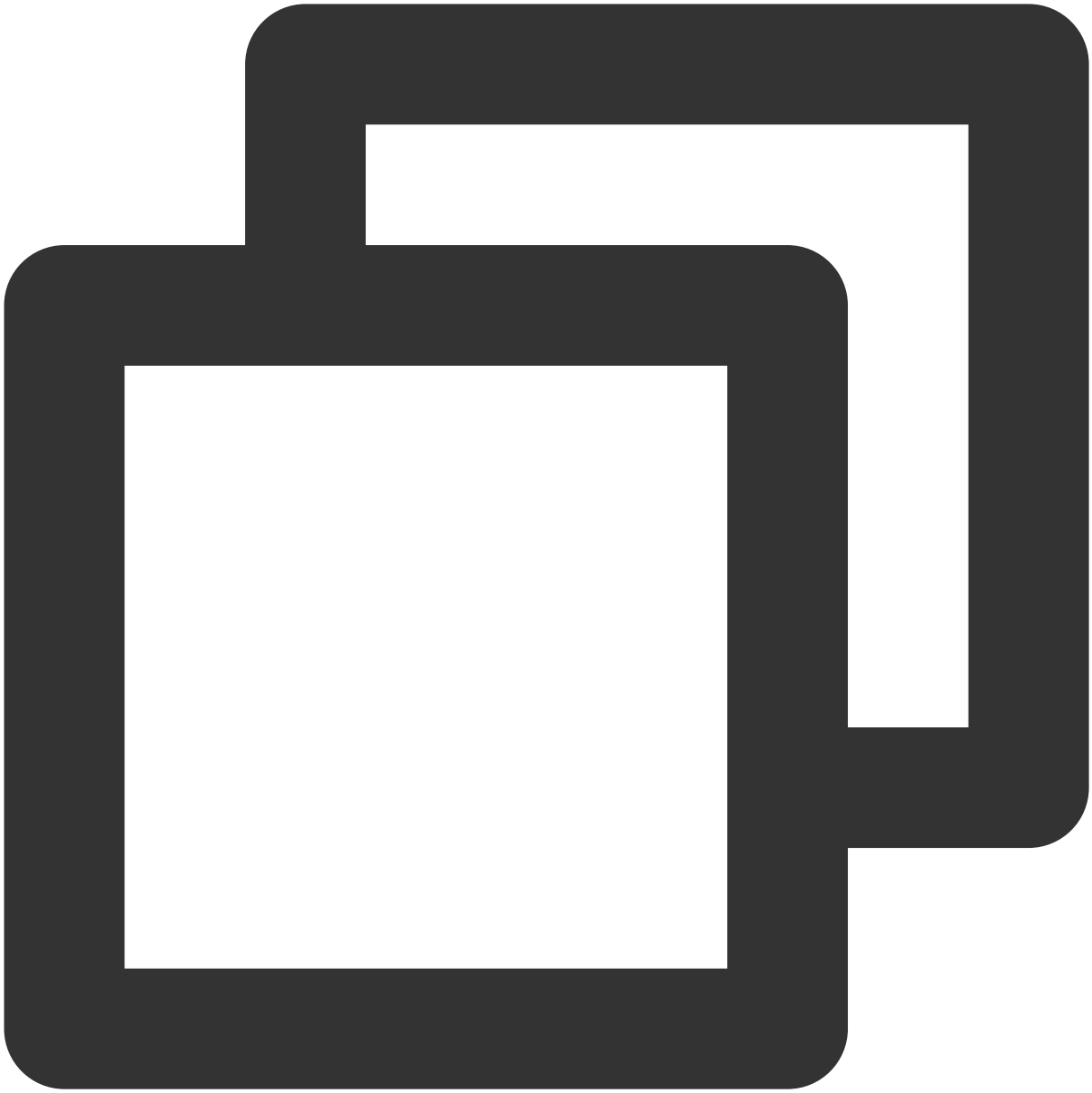
使用说明

1. 下载

请下载最新版本的 JS-SDK 包。历史版本前往 更新日志。

2. 引入 JS-SDK

在需要调用的页面引入如下 JS-SDK。示例中 `${version}` 表示版本号，请根据下载的实际版本号填写。

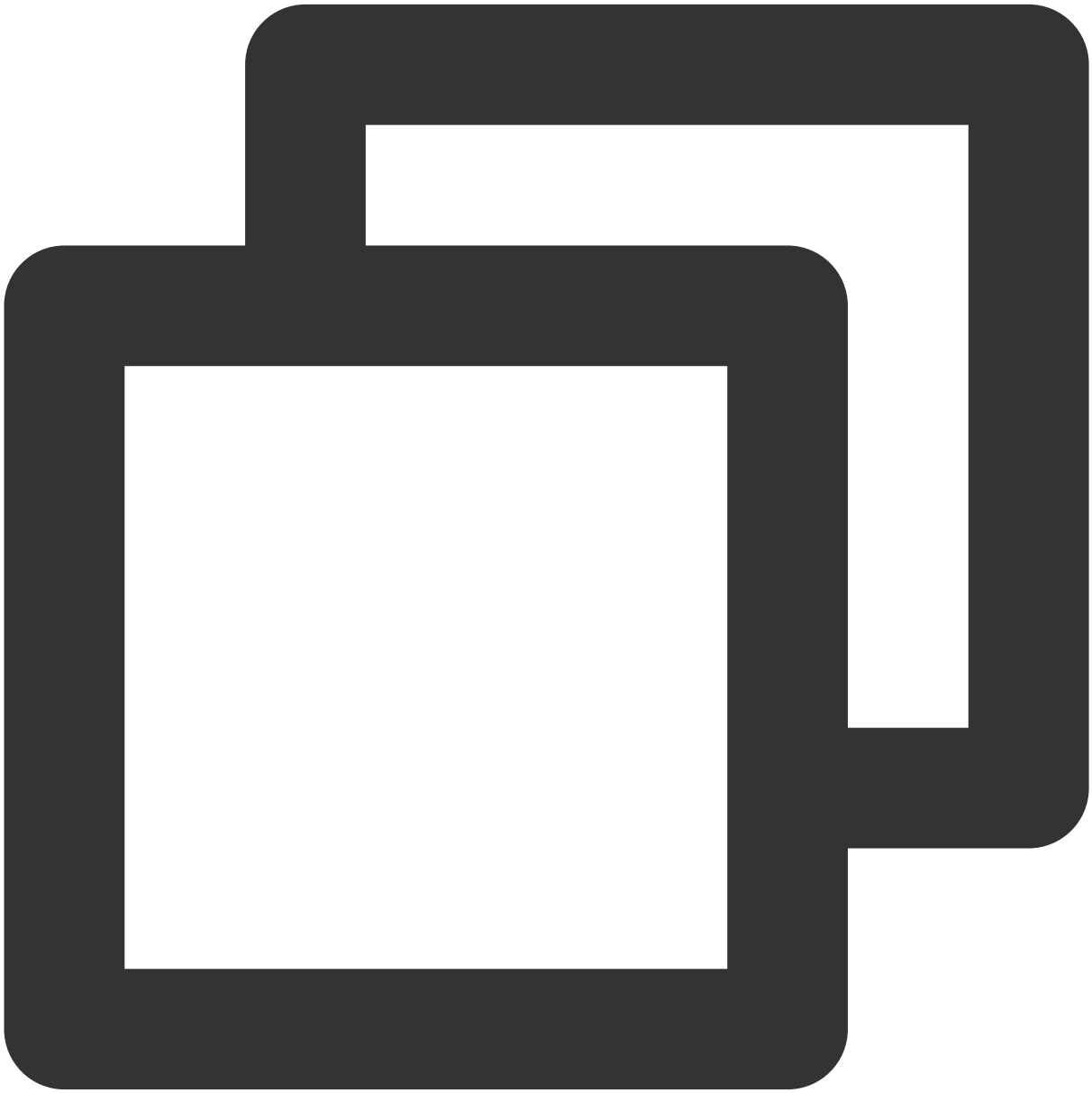


```
<script src="sdk-${version}.js"></script>
```

3. 获取在线文档预览地址

方法一：使用 JS-SDK 获取（支持 sdk-v0.1.1及以上版本）

通过 JS-SDK（v0.1.1及以上版本）获取在线文档预览地址，示例如下：



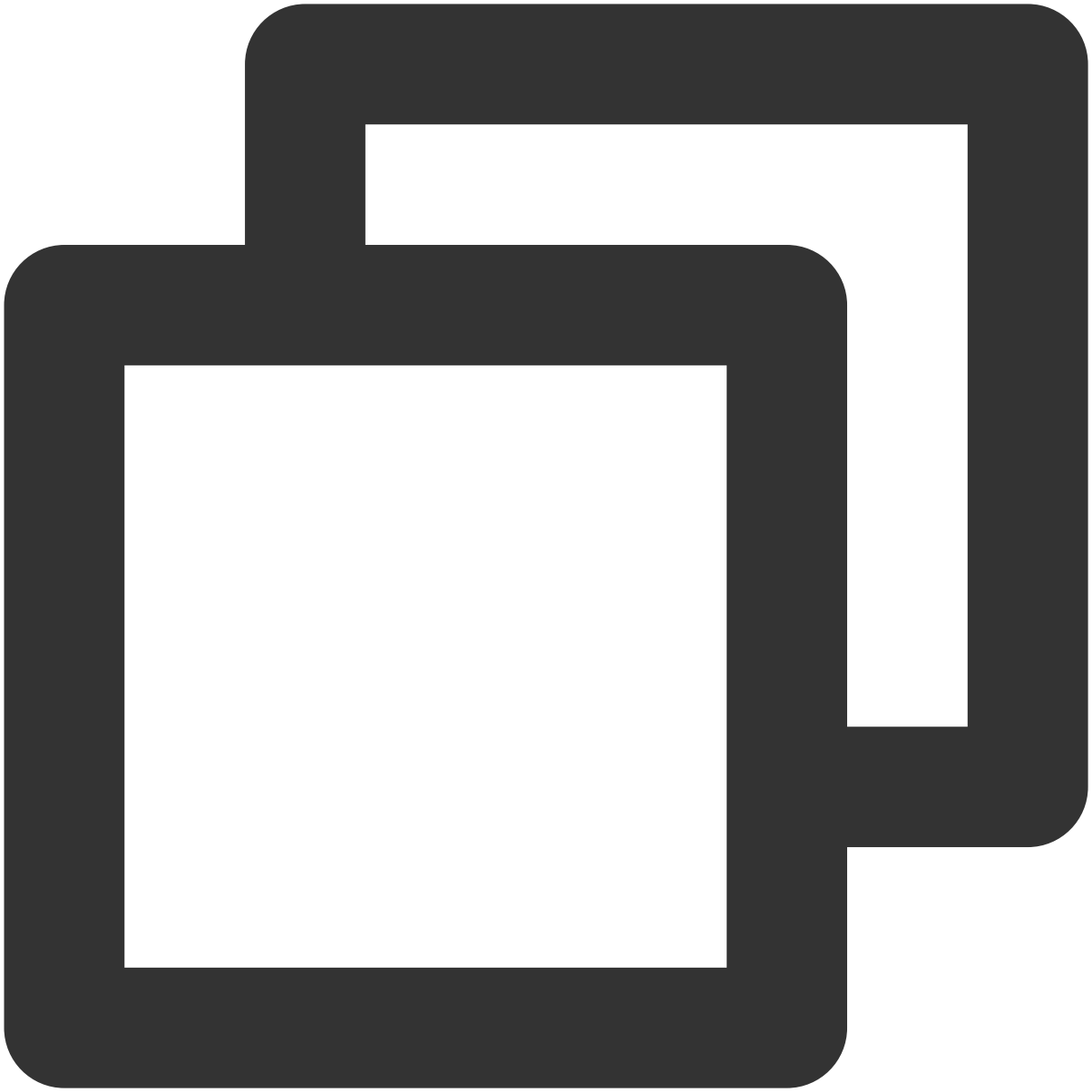
```
var url = await COSDocPreviewSDK.getPreviewUrl( {
      objectUrl: 'https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/tes
      credentials: {       // 如果预览文件是私有的，则需要携带访问凭证。
          secretId: '',
          secretKey: '',
          authorization: 'q-sign-algorithmxxxxxxxxxxxxxxx',
      },
      copyable: 0,  // 是否可以复制文件内容
      htmlwaterword: ''  // 水印内容
})
console.log(url)
```

配置项说明如下：

配置项	说明	是否必选
objectUrl	需要预览的对象地址，例如：`https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/test.pptx`	是
credentials	如果预览的对象是私有，则需要传入 credentials 作为访问凭证。可以传入计算好的 请求签名 authorization，也可以传入用户的 secretId 和 secretKey（固定密钥放在前端会有安全风险，不推荐此方式）	否
credentials.authorization	对象下载签名，详情请参见 请求签名 文档	否
credentials.secretId	用户的 secretId，可前往访问管理控制台中的 API 密钥管理 页面获取	否
credentials.secretKey	用户的 secretKey，可前往访问管理控制台中的 API 密钥管理 页面获取	否
copyable	是否可复制。默认为可复制，填入值为1；不可复制，填入值为0	否
htmlwaterword	水印文字，需要经过 URL 安全 的 Base64 编码，默认为空	否
htmlfillstyle	水印 RGBA（颜色和透明度），需要经过 URL 安全 的 Base64 编码，默认为：rgba(192,192,192,0.6)	否
htmlfront	水印文字样式，需要经过 URL 安全 的 Base64 编码，默认为：bold 20px Serif	否
htmlrotate	水印文字旋转角度，0 - 360，默认315度	否
htmlhorizontal	水印文字水平间距，单位 px，默认为50	否
htmlvertical	水印文字垂直间距，单位 px，默认为100	否

方法二：直接请求获取

请求示例



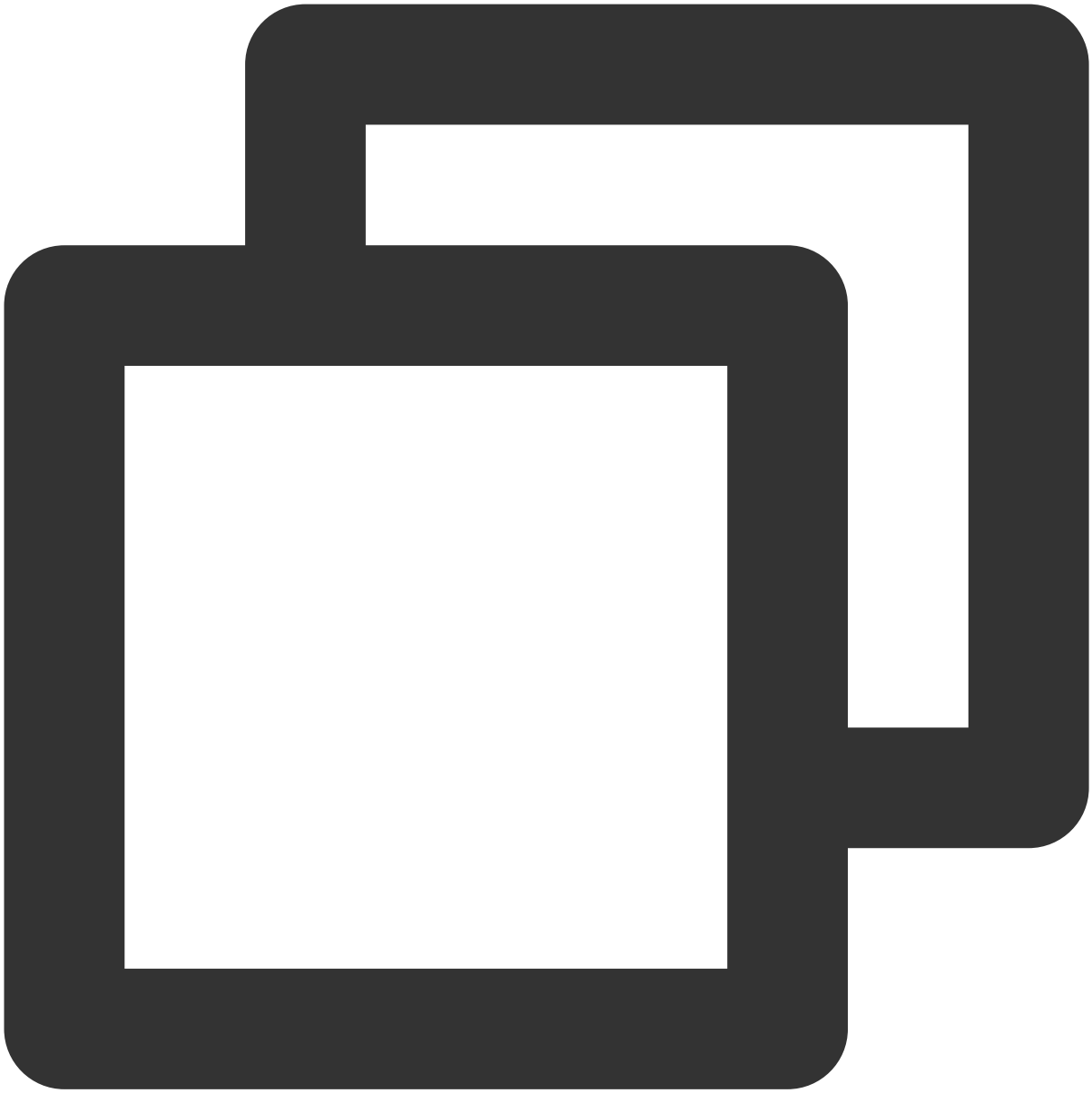
```
GET /<ObjectKey>?ci-process=doc-preview&dstType=html&weboffice_url=1&sign=AuthStrin
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
```

请求参数

参数名称	描述	是否必选
ci-process	预览固定参数，值为 doc-preview	是

dstType	预览类型，如需预览生成类型为 html 则填入 html	是
weboffice_url	是否获取预览链接。填入值为1会返回预览链接和 Token 信息；填入值为2只返回 Token 信息；不传会直接预览	否
tokenuid	是否获取 Token。根据传入的 tokenuid、文件信息等生成 Token，返回 Token 和有效期；如果不传则不会返回 Token 信息	否
sign	对象下载签名，如果预览的对象为私有读时，需要传入签名，详情请参见 请求签名 文档 注意：需要进行 urlencode	否
copyable	是否可复制。默认为可复制，填入值为1；不可复制，填入值为0	否
htmlwaterword	水印文字，需要经过 URL 安全 的 Base64 编码，默认为空	否
htmlfillstyle	水印 RGBA（颜色和透明度），需要经过 URL 安全 的 Base64 编码，默认为：rgba(192,192,192,0.6)	否
htmlfront	水印文字样式，需要经过 URL 安全 的 Base64 编码，默认为：bold 20px Serif	否
htmlrotate	水印文字旋转角度，0 - 360，默认315度	否
htmlhorizontal	水印文字水平间距，单位 px，默认为50	否
htmlvertical	水印文字垂直间距，单位 px，默认为100	否

响应示例

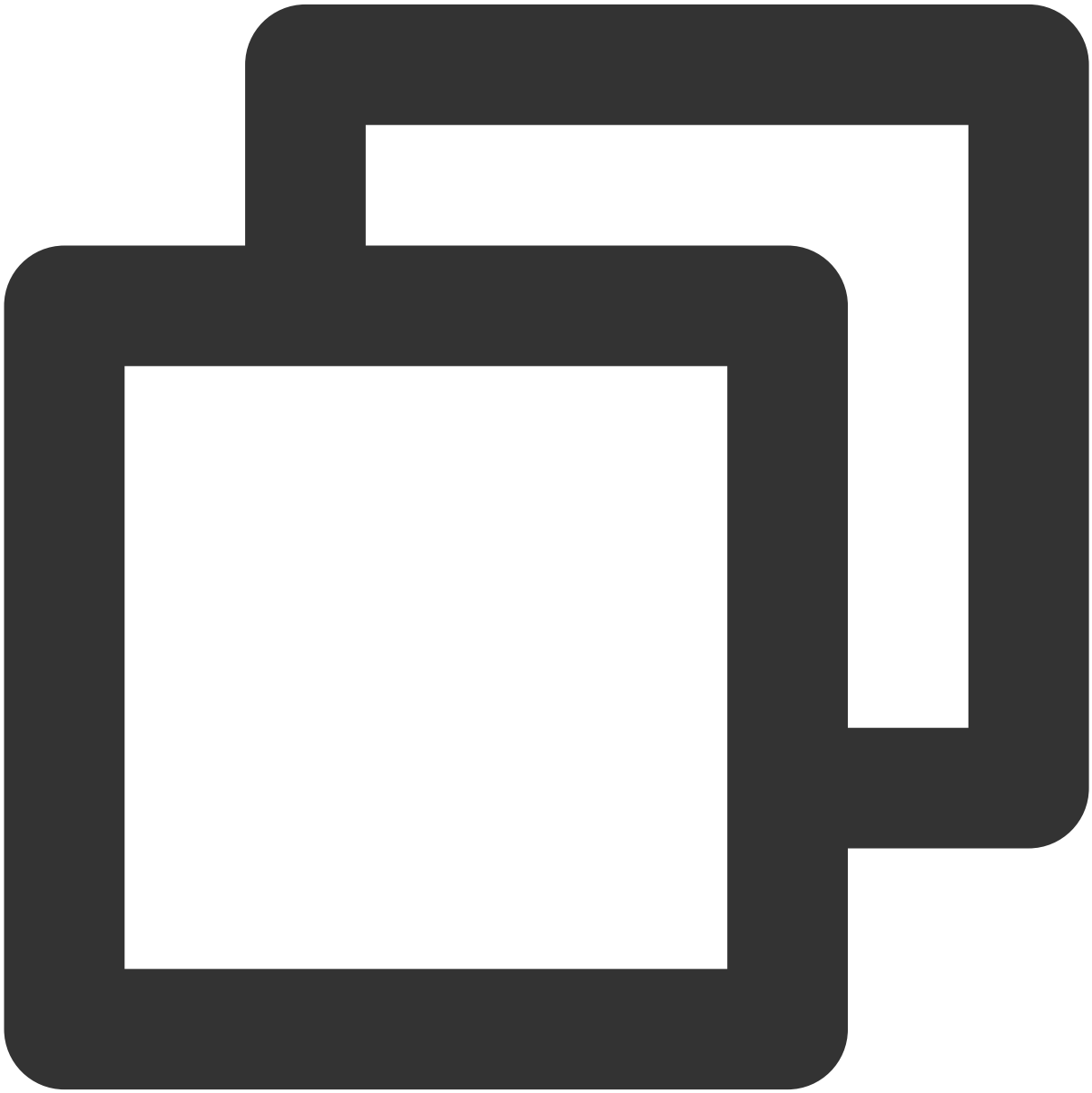


```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 478
Date: Thu, 25 Feb 2021 07:44:37 GMT
Status: 200 OK
x-cos-request-id: NjAzNzU1NjVfY2MxZTIzMDlfNzc5NF8yMDQ5****



{
        "PreviewUrl": "https://prvsh.myqcloud.com/office/p/xxxx?_w_provider=XXX"
```

```
}
```

4. 初始化

通过初始化配置，JS-SDK 会在配置的 mount 节点下创建一个 iframe，并自动初始化相关数据和事件。

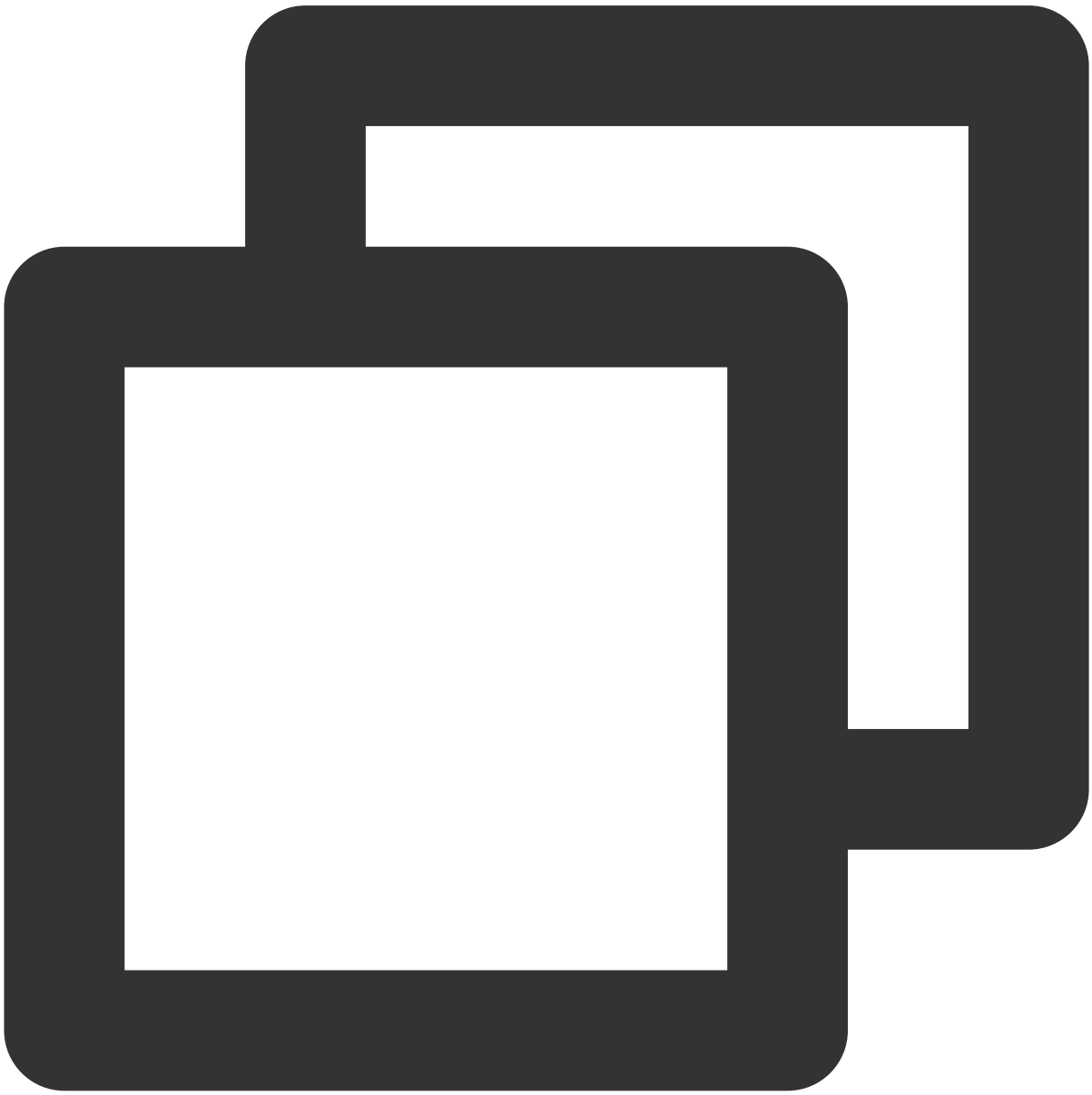


```
var demo = COSDocPreviewSDK.config({
    url: '在线文档预览地址',  // 步骤3中获取到的在线文档预览地址
})
// 如果需要对 iframe 进行特殊的处理，可以通过以下方式拿到 iframe 的 dom 对象
console.log(demo.iframe)
```

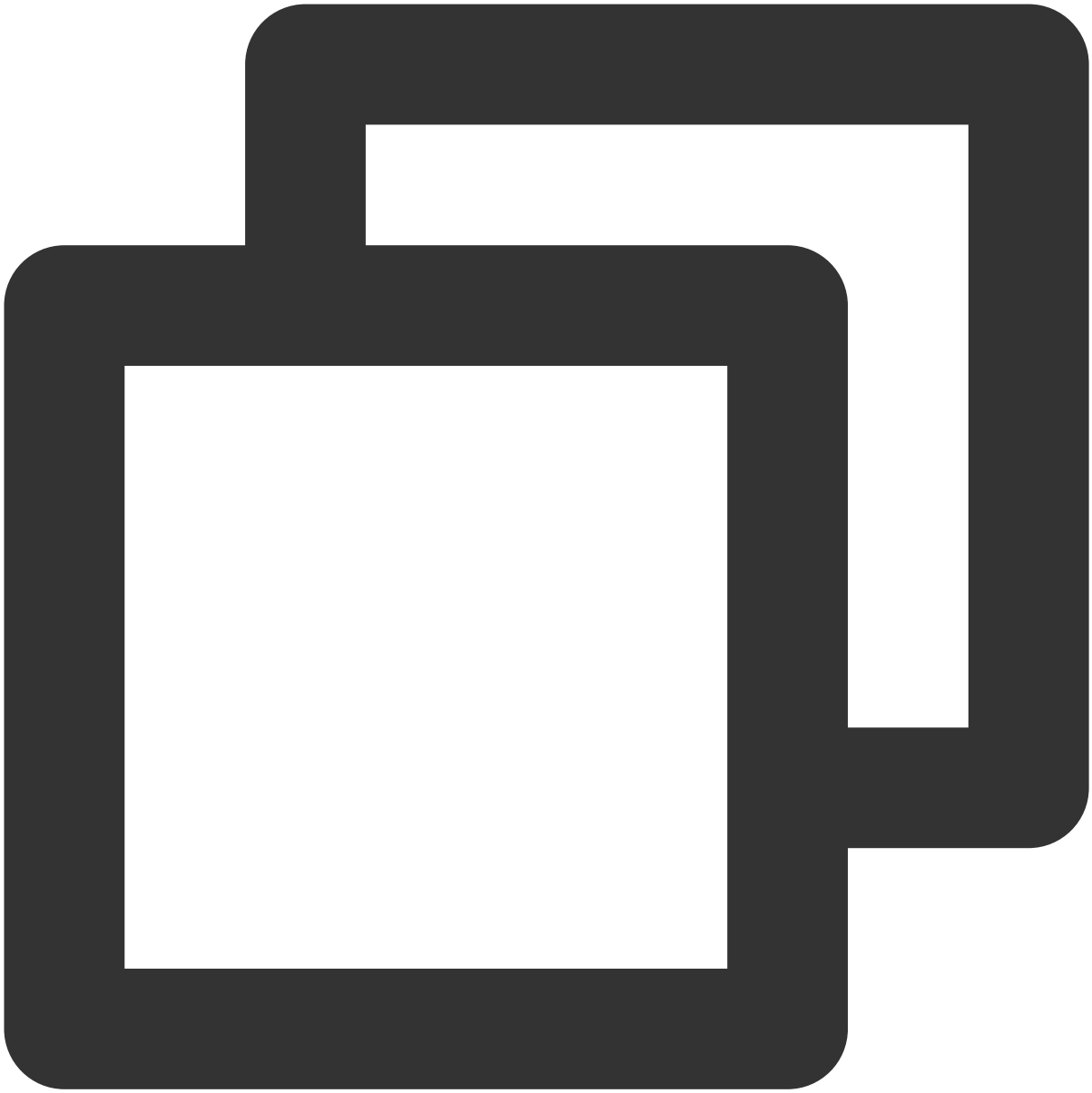
```
// 打开文档结果
demo.on('fileOpen', function(data) {
    console.log(data.success)
})
```

5. 挂载节点

挂载节点指 JS-SDK 插入 iframe 时挂载的节点。在初始化时，JS-SDK 挂载节点为可配置项。

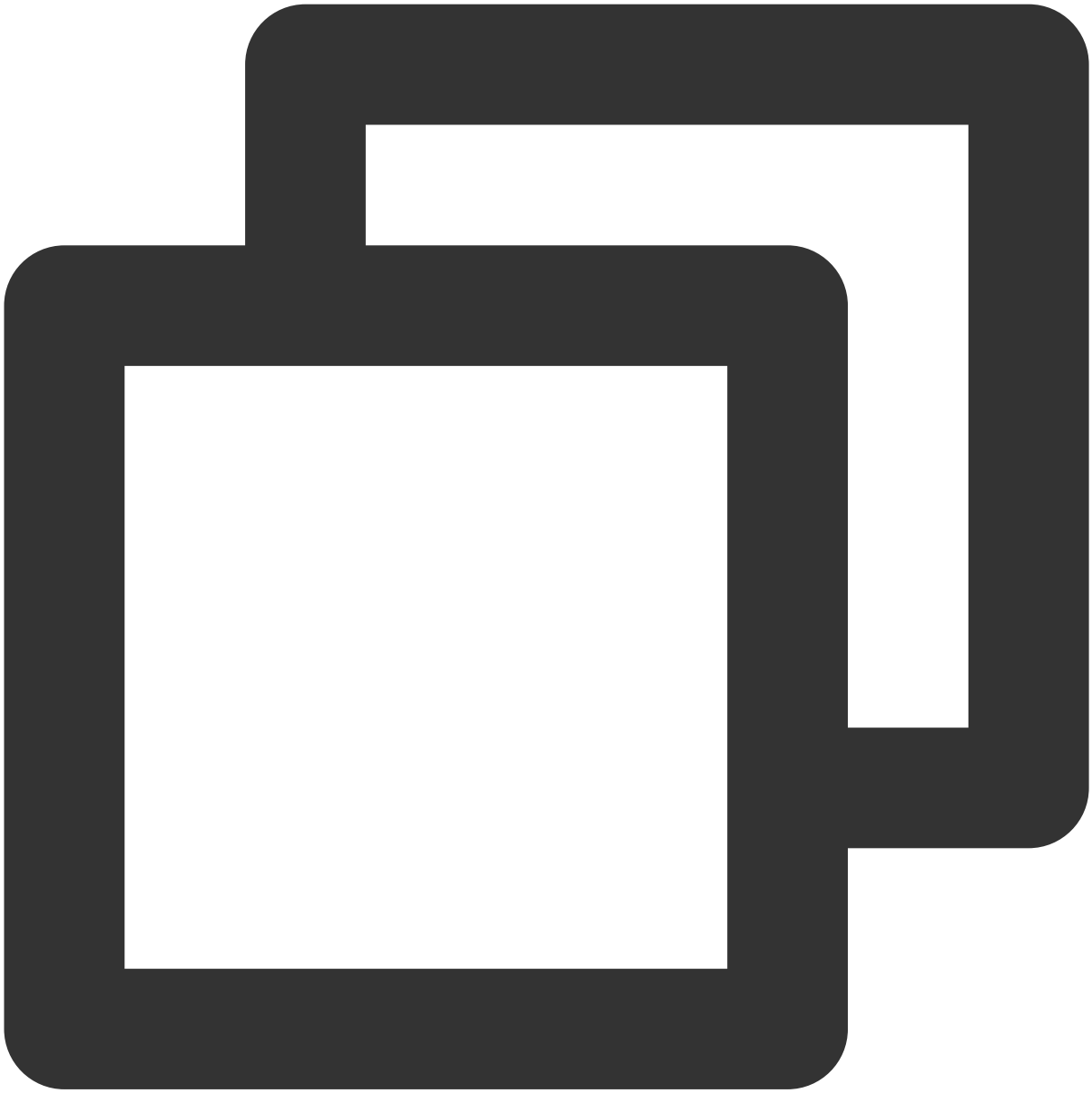


```
<div class="custom-mount"></div>
```



```
var demo = COSDocPreviewSDK.config({
    url: '在线文档预览地址',     // 步骤3中获取到的在线文档预览地址
    mount: document.querySelector('.custom-mount')     // 挂载节点，默认会挂载到 body 下
})
```

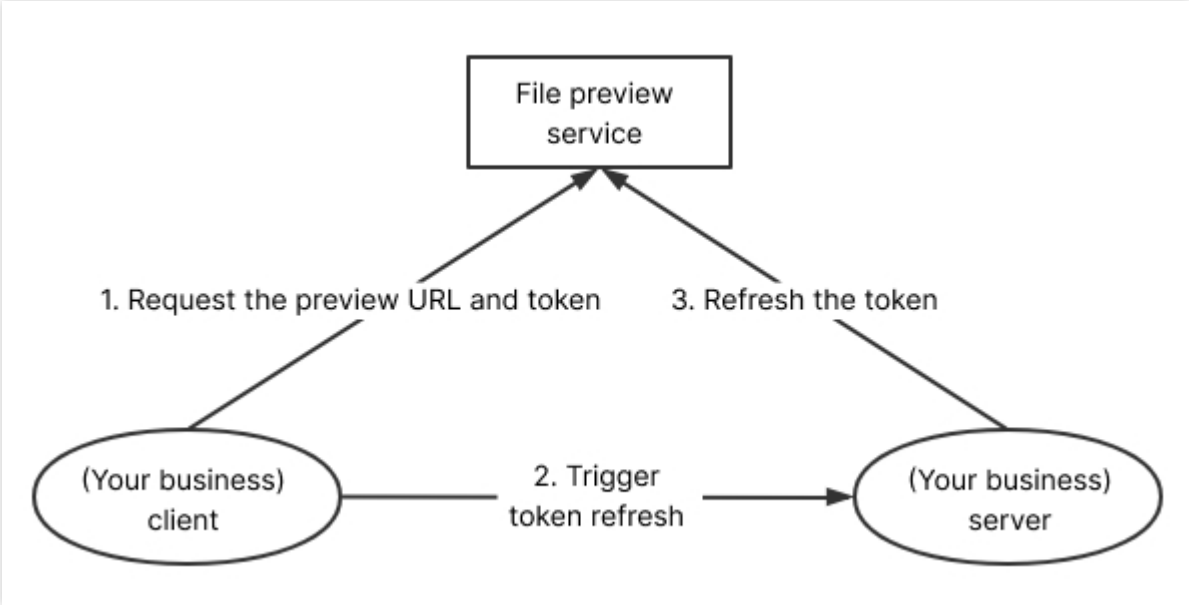
JS-SDK 初始化完成后会自动在挂载节点下面插入一个 iframe。



```
<div class="custom-mount">
    <iframe src="..."></iframe>
</div>
```

6. Token 鉴权

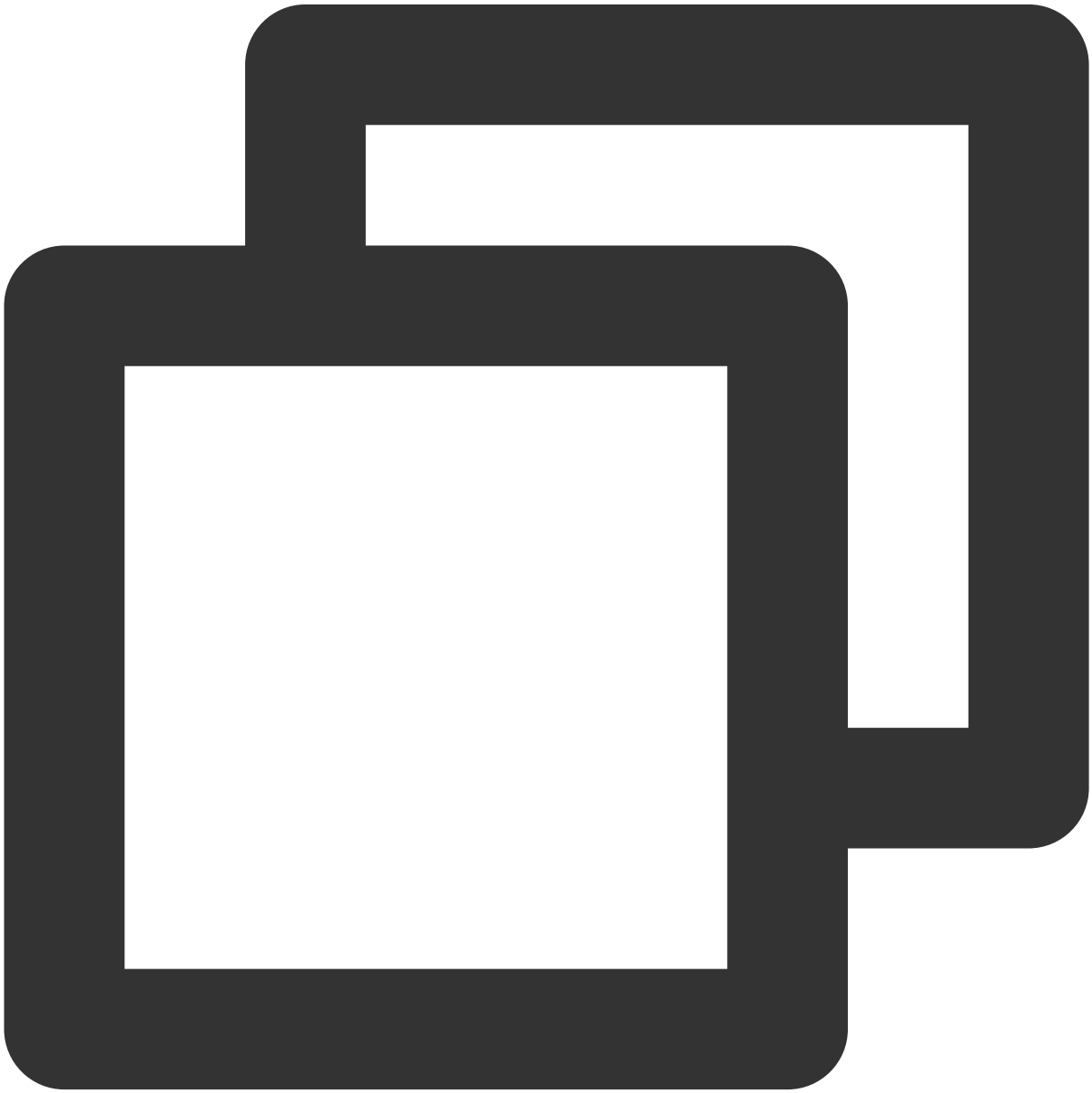
不使用 Token 时，默认使用文档的签名有效期鉴权，过期则无法预览文件。
使用 Token 鉴权可以在预览期间持续进行 Token 鉴权和 Token 续期，由业务侧自主控制预览授权。



Token 鉴权分为三个步骤：

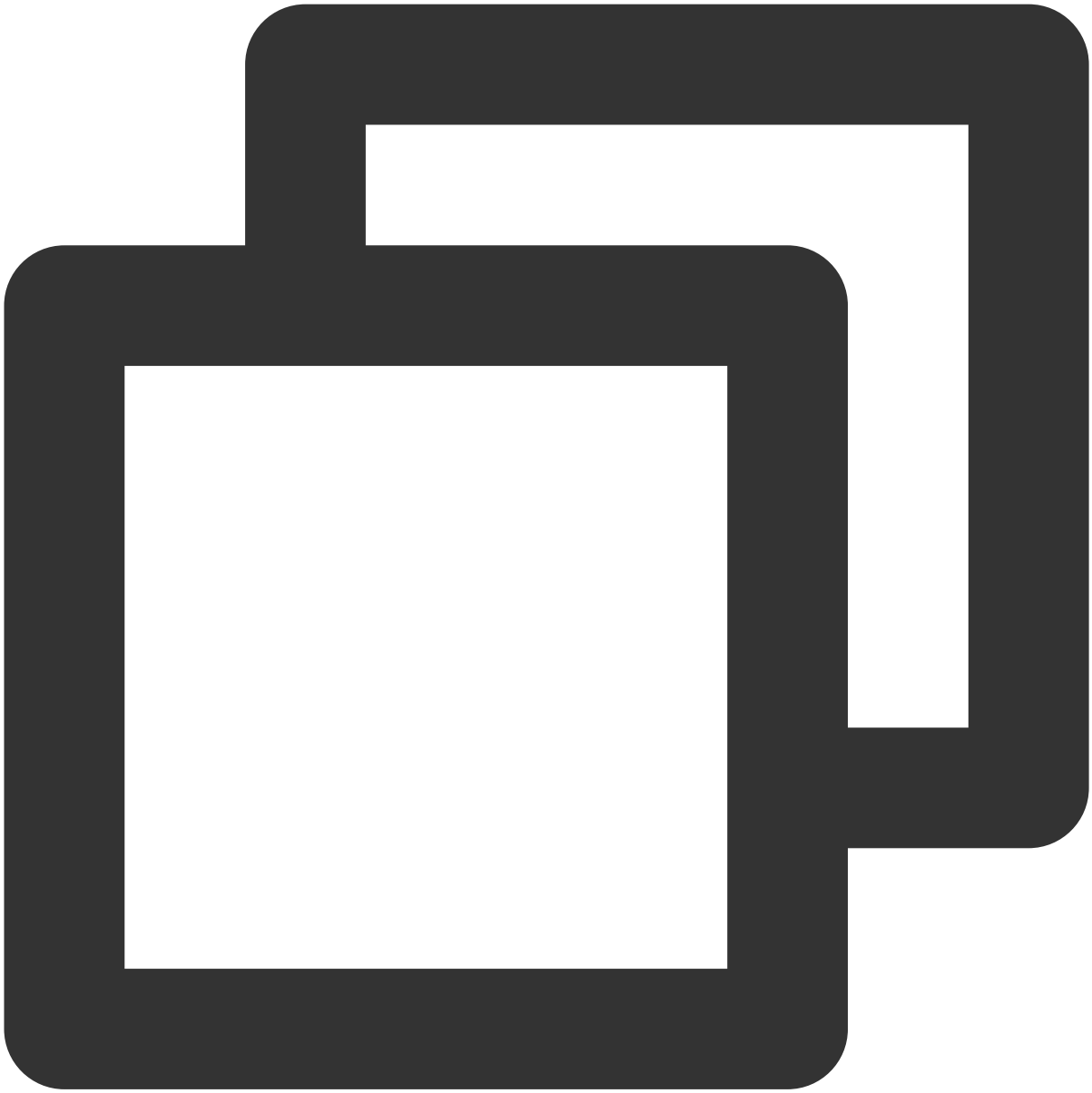
1. 获取 Token

请求示例



```
GET /<ObjectKey>?ci-process=doc-preview&dstType=html&weboffice_url=2&tokenuid=test&
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
```

响应示例



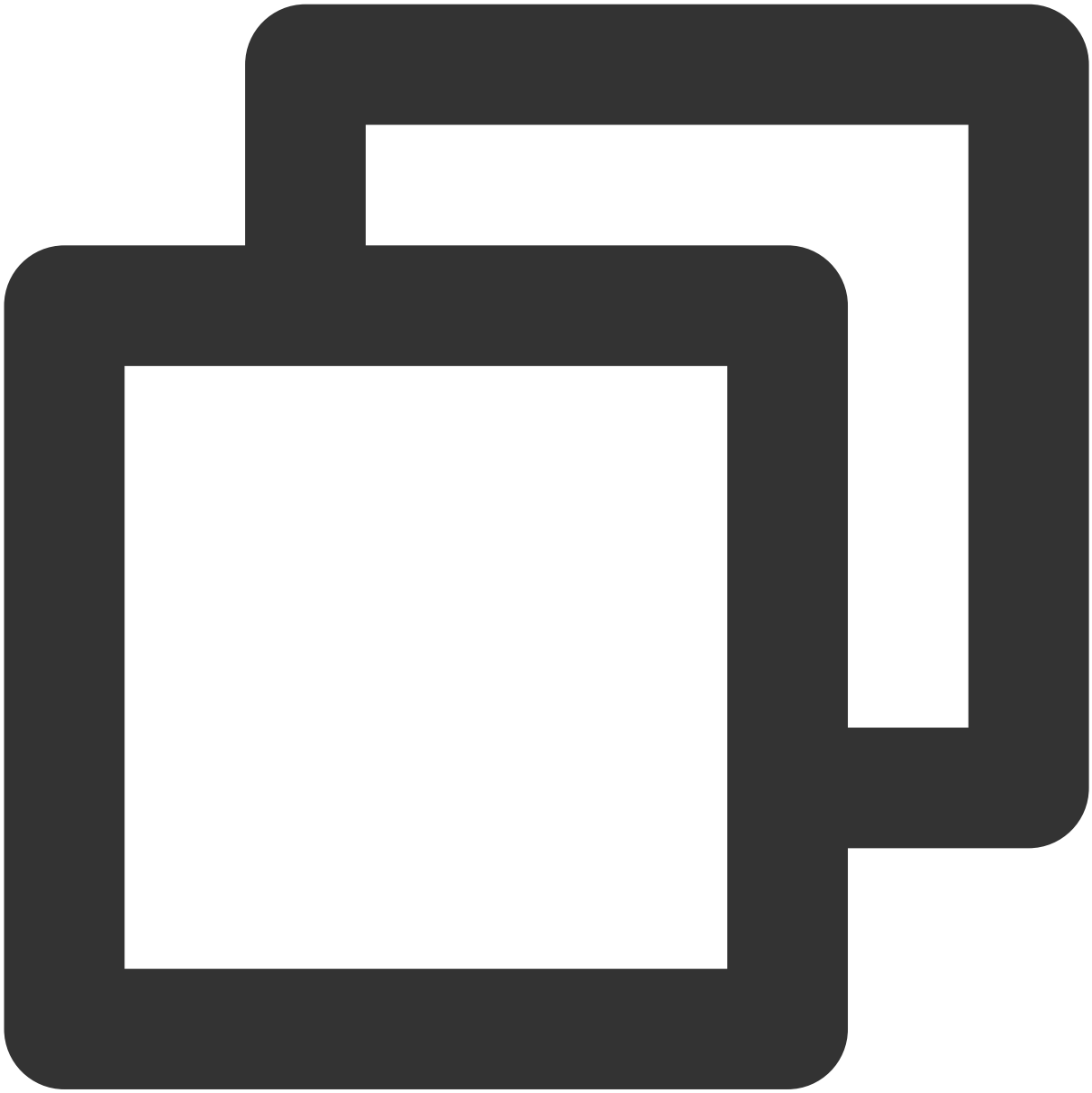
```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 108
Date: Tue, 21 Jun 2022 10:41:10 GMT
Status: 200 OK
x-cos-request-id: NjJiMWEwNDZfMTgyMjYyNjRfYmRlYl9iNTAw****



{
 Token: "ChAJj2vNRiHTc8reToMmJ7T2EJ7FxpUGGhS1mPJanpN03u8PSB17/YKBA/Vo****"
```

```
 TokenExpireTime: "1655808670"
}
```

2. 设置 Token

说明

为避免 Token 刷新过于频繁，推荐设置的超时时间（timeout）为 10 分钟以上；为避免 refresh 请求超时，JS-SDK 会提前几分钟去更新 Token。



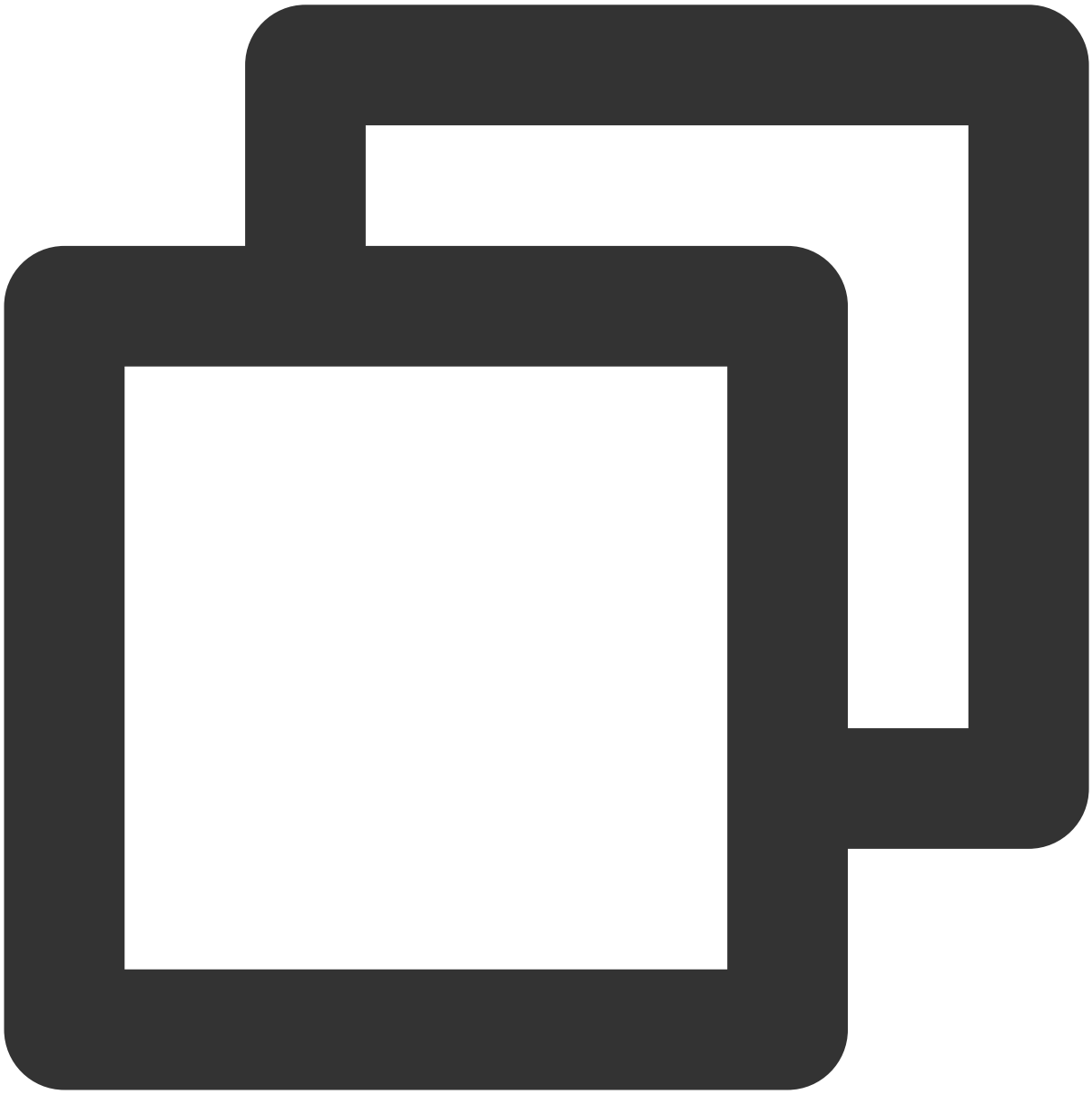
```
var demo = COSDocPreviewSDK.config({
 url: '在线文档预览地址',  // 步骤3中获取到的在线文档预览地址
})
```

```
// 设置Token
demo.setToken({
  token: 'yourToken', // 填入获取到的token
  timeout: 10 * 60 * 1000, // token超时时间，单位为毫秒，为避免频繁续期，建议设置 10 分钟及以
});
```

3. 配置 refreshToken 方法，超时自动续期 Token

refreshToken 方法需要请求**您的服务端封装的更新 Token 的方法**，服务端可以封装上述的获取 Token 的方法（参考获取 Token）。

以服务端方法 `http://your.server.com/refreshToken` 为例，配置 refreshToken 函数如下：



```
// 获取 Token 函数
const refreshToken = () => {
  // 调用服务端封装的方法，获取token信息
  const tokenInfo = await $.get('http://your.server.com/refreshToken')

  // 可以返回 Promise 或者 return { token, timeout }
  return {
 token: tokenInfo.token, // 必需：你需要设置的 Token
 timeout: 10 * 60 * 1000, //  必需：Token 超时时间，单位毫秒
  };
};

var demo = COSDocPreviewSDK.config({
 url: '在线文档预览地址',
 refreshToken    // Token自动更新方法
})

// 设置Token
demo.setToken({
  token: 'yourToken', // 填入获取到的token
  timeout: 10 * 60 * 1000   // token超时时间，单位为毫秒
});
```

实际案例

文字文件在线预览案例，参考 DOC 预览案例。
演示文件在线预览案例，参考 PPT 预览案例。
表格文件在线预览案例，参考 Excel 预览案例。
PDF文件在线预览案例，参考 PDF 预览案例。

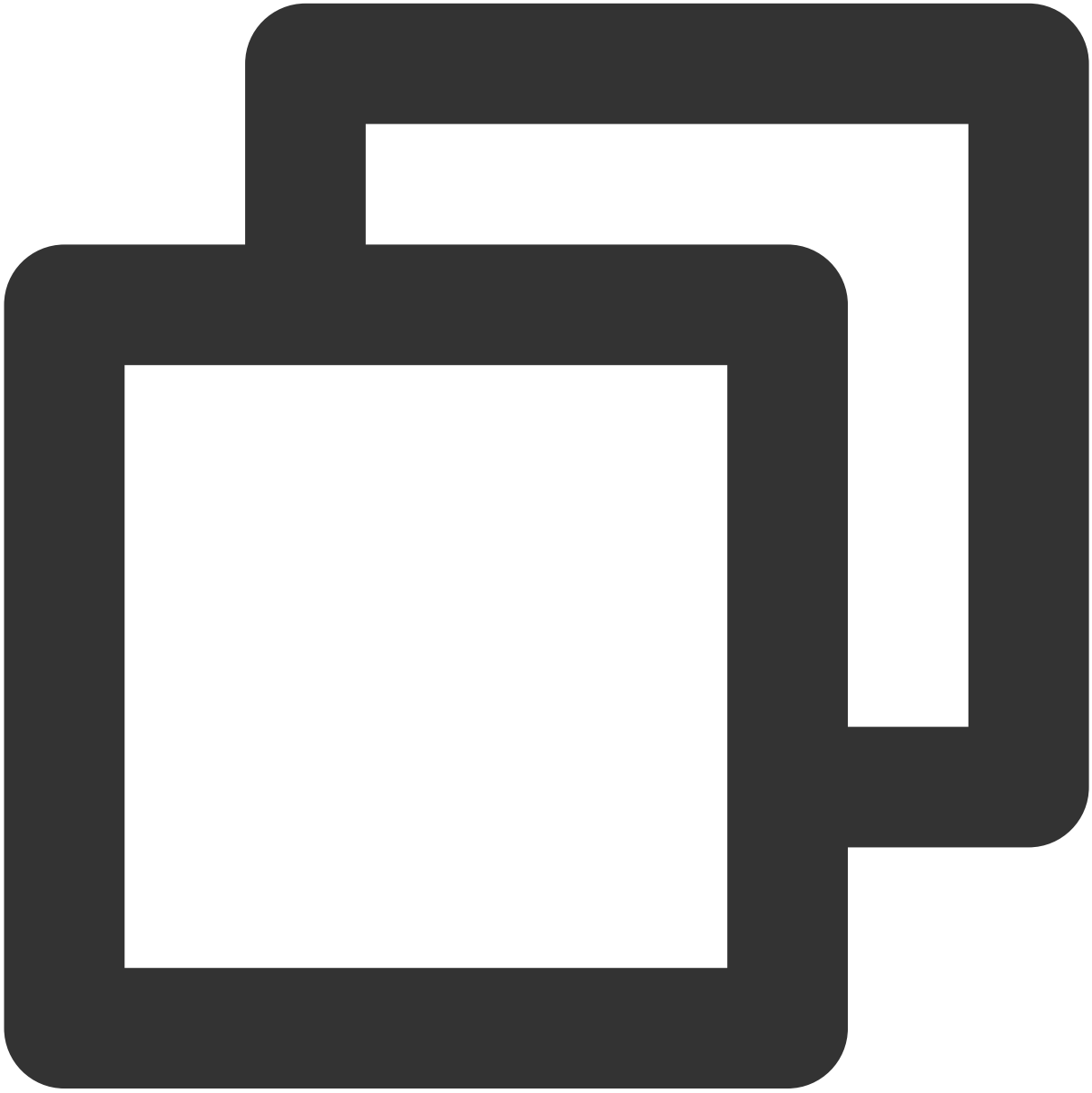
版权所有：腾讯云计算（北京）有限责任公司

配置项说明

最近更新时间：2023-12-22 11:44:08

显示模式

通过初始化配置选项类，开启不同的显示模式：



```
var demo = COSDocPreviewSDK.config({
```

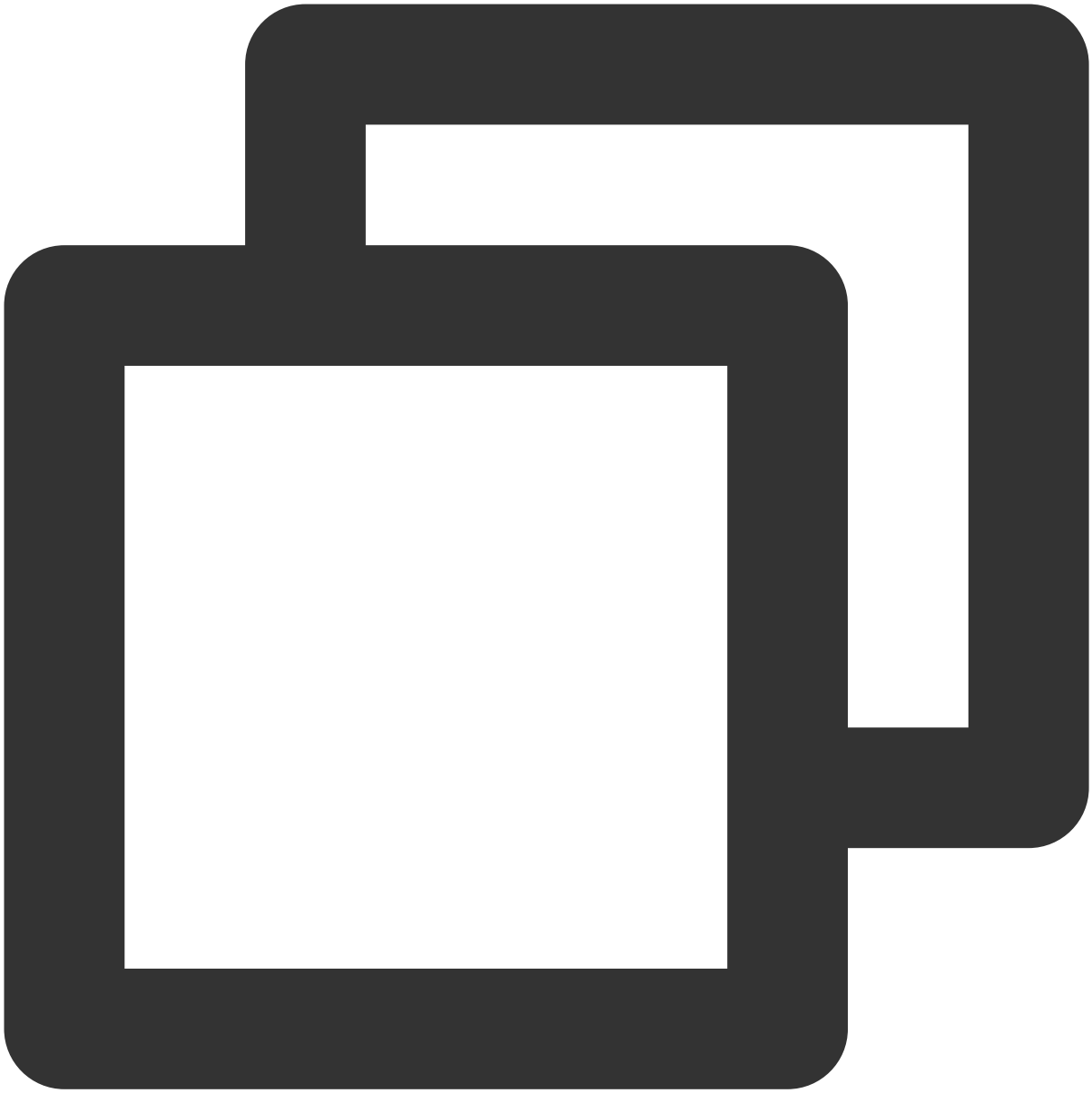
```
    mode: 'normal',
})
```

配置选项说明如下：

mode 选项值	说明	是否默认选项
normal	普通模式，展示所有功能界面	是
simple	极简模式，不显示头部和工具栏	否

自定义配置选项

初始化 JS-SDK 时，通过分别配置不同文档对应的配置选项，可以开启或关闭文档中的特定配置以及控制文档打开时的状态。



```
COSDocPreviewSDK.config({
    //通用选项，适用于所有类型的文档。
    commonOptions: {
      isShowTopArea: false, //隐藏顶部区域（头部和工具栏）。
      isShowHeader: false //隐藏头部区域。

    },
    //WORD文档相关配置。
    wordOptions: {
      isShowDocMap: false,
      isBestScale: true
```

```
    },
    //PDF文档相关配置。
    pdfOptions: {
      isShowComment: true,
      isInSafeMode: false
    },
    //演示文档相关配置。
    pptOptions: {
      isShowBottomStatusBar: true
    }
})
```

通用配置项：commonOptions

配置项	说明	默认值
isShowTopArea	是否显示顶部区域（头部和工具栏）	false
isShowHeader	是否显示头部区域	false
isBrowserViewFullscreen	是否在浏览器区域全屏	false
isIframeViewFullscreen	是否在 iframe 区域内全屏	false

文字文件配置项：wordOptions

配置项	说明	默认值
isShowDocMap	是否开启目录功能，默认开启	true
isBestScale	打开文档时，默认以最佳比例显示（适用于 PC）	true
isShowBottomStatusBar	是否展示底部状态栏（适用于 PC）	false

PDF 配置项：pdfOptions

配置项	说明	默认值
isShowComment	是否显示注解，默认显示	true
isInSafeMode	是否处于安全模式（安全模式下不能划选文字，不能复制以及不能通过链接跳转），默认不是安全模式	false
isShowBottomStatusBar	是否展示底部状态栏（适用于 PC）	false

版权所有：腾讯云计算（北京）有限责任公司

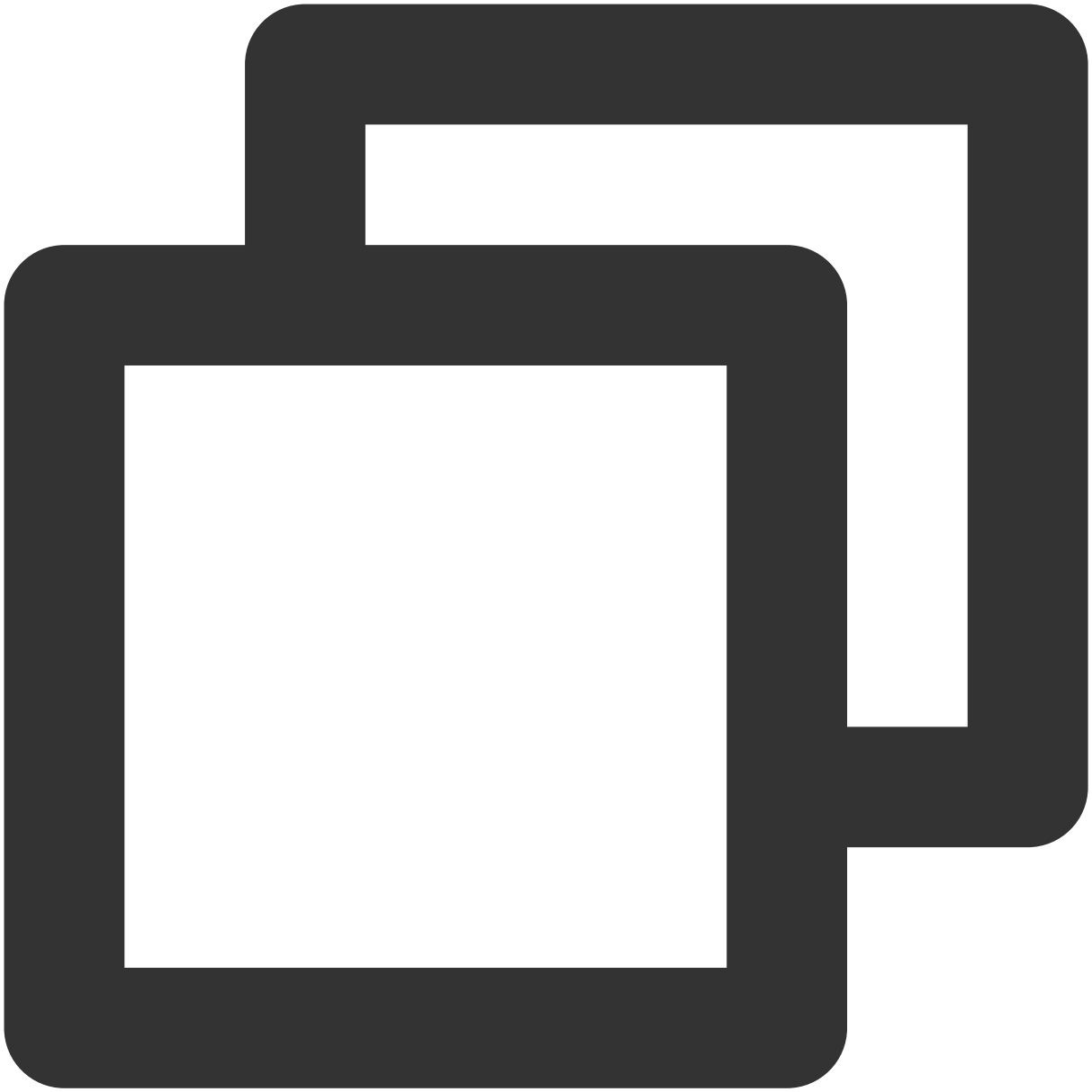
演示文件配置项：pptOptions

配置项	说明	默认值
isShowBottomStatusBar	是否展示底部状态栏（适用于PC）	false

组件状态设置

可以通过 commandBars 选项来隐藏或禁用页面的组件。

版权所有：腾讯云计算（北京）有限责任公司



```
COSDocPreviewSDK.config({
    //通用选项，适用于所有类型的文档。
    commandBars: [
      {
        cmbId: 'Print', // 组件ID
        attributes: {
          visible: false, // 组件是否显示，默认为true。false：隐藏组件；true：显示组件
          enable: false, // 禁用或者开启组件，默认为true
        },
      },
    ]
```

```
})
```

目前可用的组件列表如下：

组件ID	说明	适用范围
Print	打印功能，和 ctrl+P 打印快捷键	所有文档
MobileHeader	客户端的顶部工具栏	所有文档
WriterHoverToolbars	客户端的底部工具栏	文字文件
DownloadImg	双击图片下载按钮图标	演示文件
PlayComponentToolbar	全屏播放时右上角工具栏	演示文件
PlayContextCheckImage	播放时右键 - 查看原图	演示文件
FloatMenuCheckImage	选中图片时浮动菜单栏 - 查看原图	演示文件
WPPMobileCommentButton	客户端底部工具栏的评论按钮	演示文件
PDFMobilePageBar	客户端的页码	PDF文件

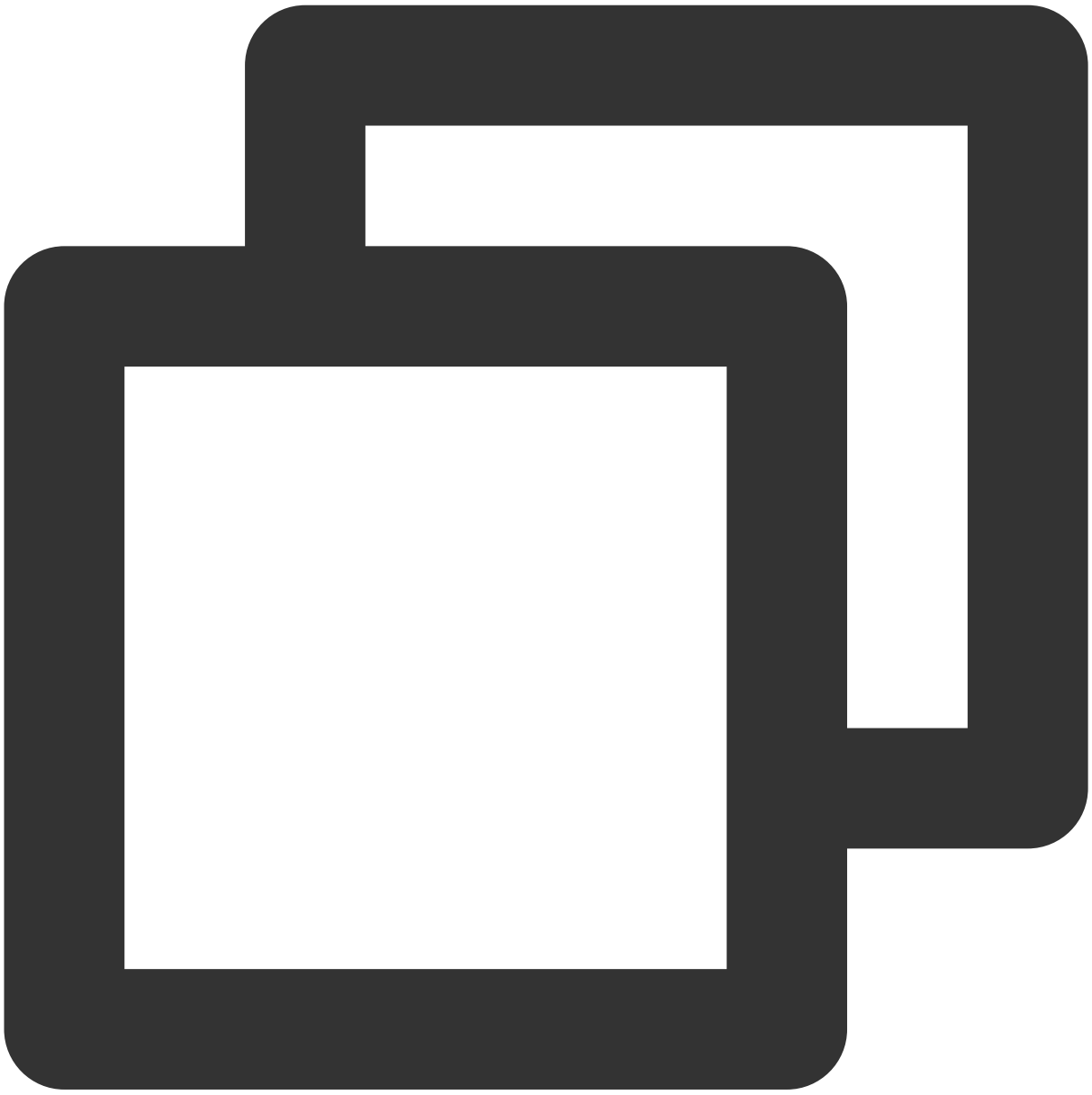
版权所有：腾讯云计算（北京）有限责任公司

高级功能

最近更新时间：2023-12-30 10:44:30

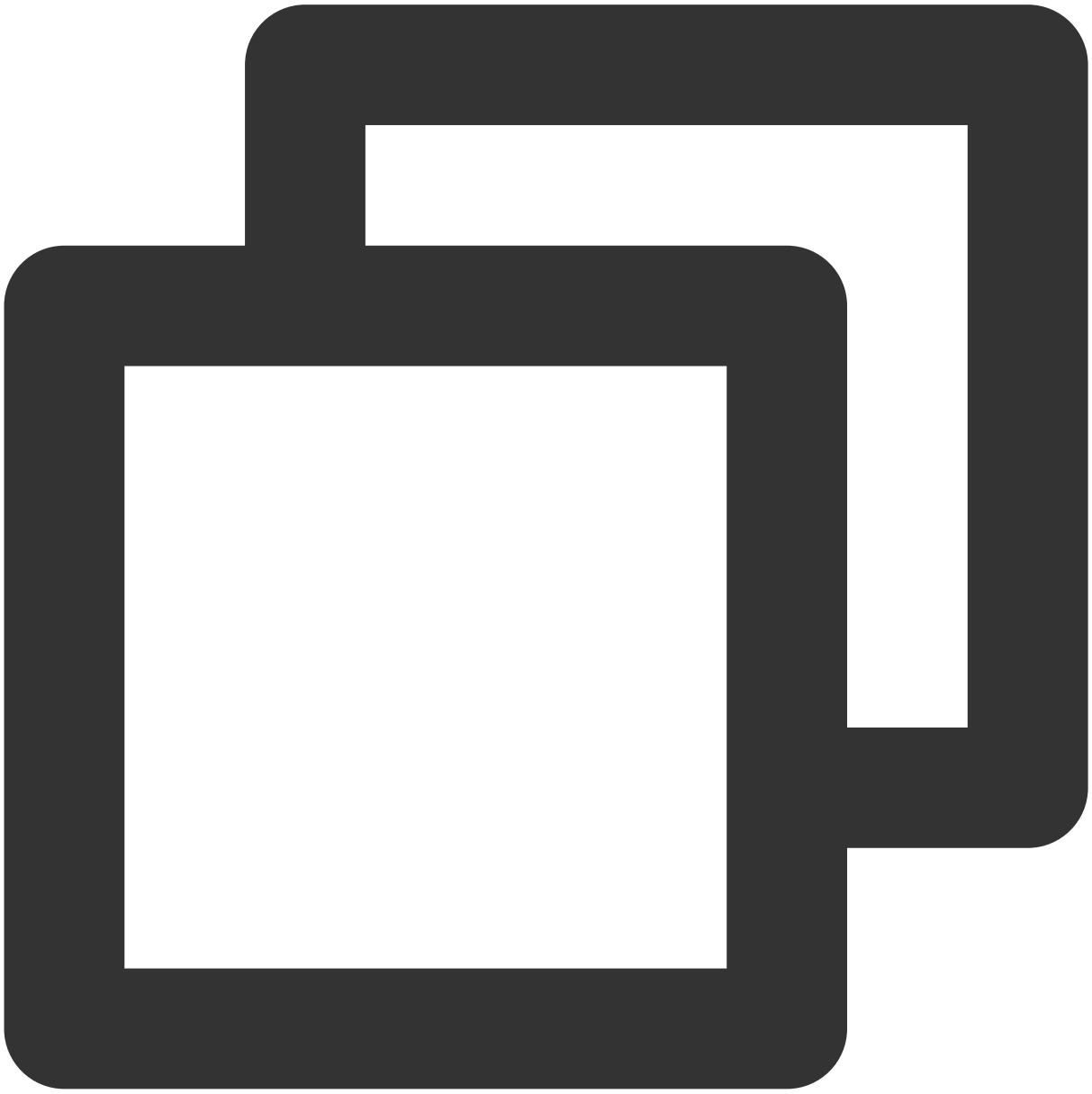
JS-SDK 提供了高级功能可直接对文档进行操作，以下是高级功能的调用流程。

1. 等待文档加载完成



```
await demo.ready() // 务必等待 demo ready 之后再调用高级 api, demo 对象是指 JS-SDK 实例化
```

2. 获取文档对象



```
// 文字
const wordApp = demo.WordApplication()
// 表格
```

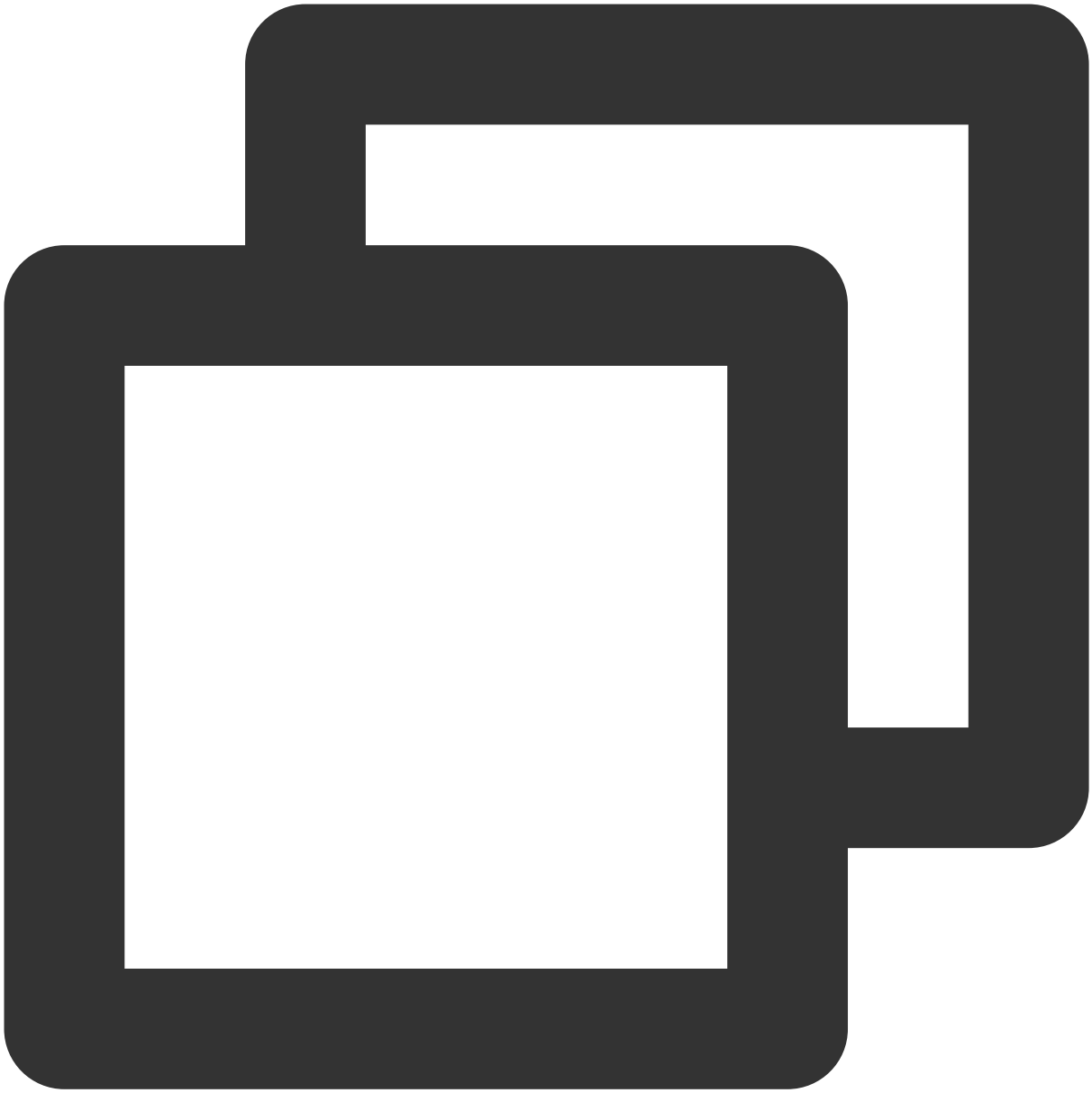
```
const excelApp = demo.ExcelApplication()
// 演示
const pptApp = demo.PPTApplication()
// PDF
const pdfApp = demo.PDFApplication()

// 自动识别
const app = demo.Application
```

3. 使用高级功能

以下对文字文件、表格文件、演示文件、PDF 文件等类型使用高级功能进行列举介绍，如需了解上述类型支持的文件格式，请参见 规则与限制。

文字文件

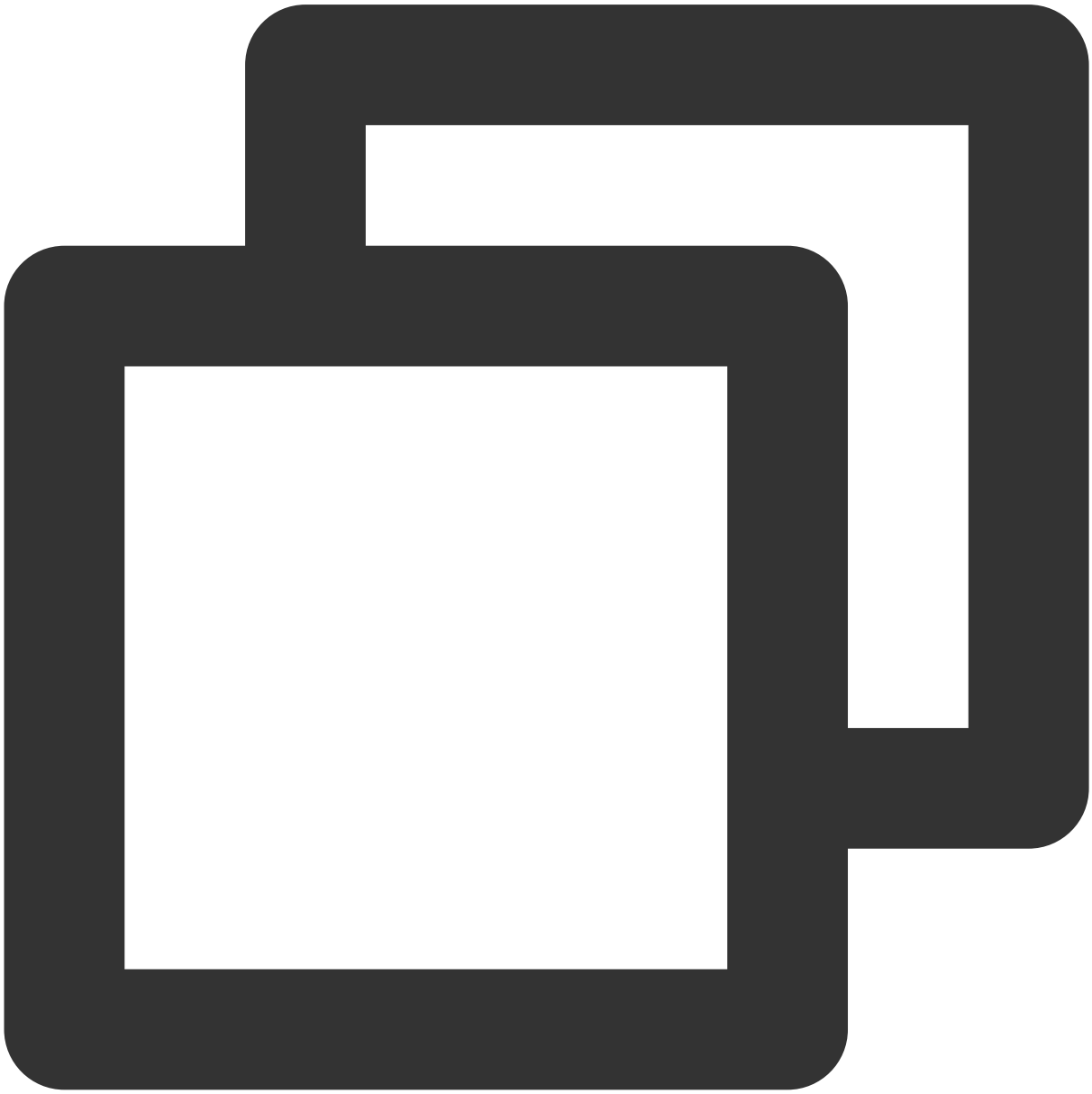
获取总页数



```
/*
* @param: WdInformation: {
*       wdNumberOfPagesInDocument: 4
*   }
* @return: {PagesCount: number, End: boolean}
*/
const app = demo.WordApplication()
const {Enum} = app
let totalPages = await app.ActiveDocument.Range.Information(Enum.WdInformation.wd
if (totalPages.End) {
  // 获取总页数是异步过程，仅当totalPages.End为true时，获取到的总页数才是准确的
```

```
    console.log("加载完了！一共", totalPages.PagesCount, "页")
}
```

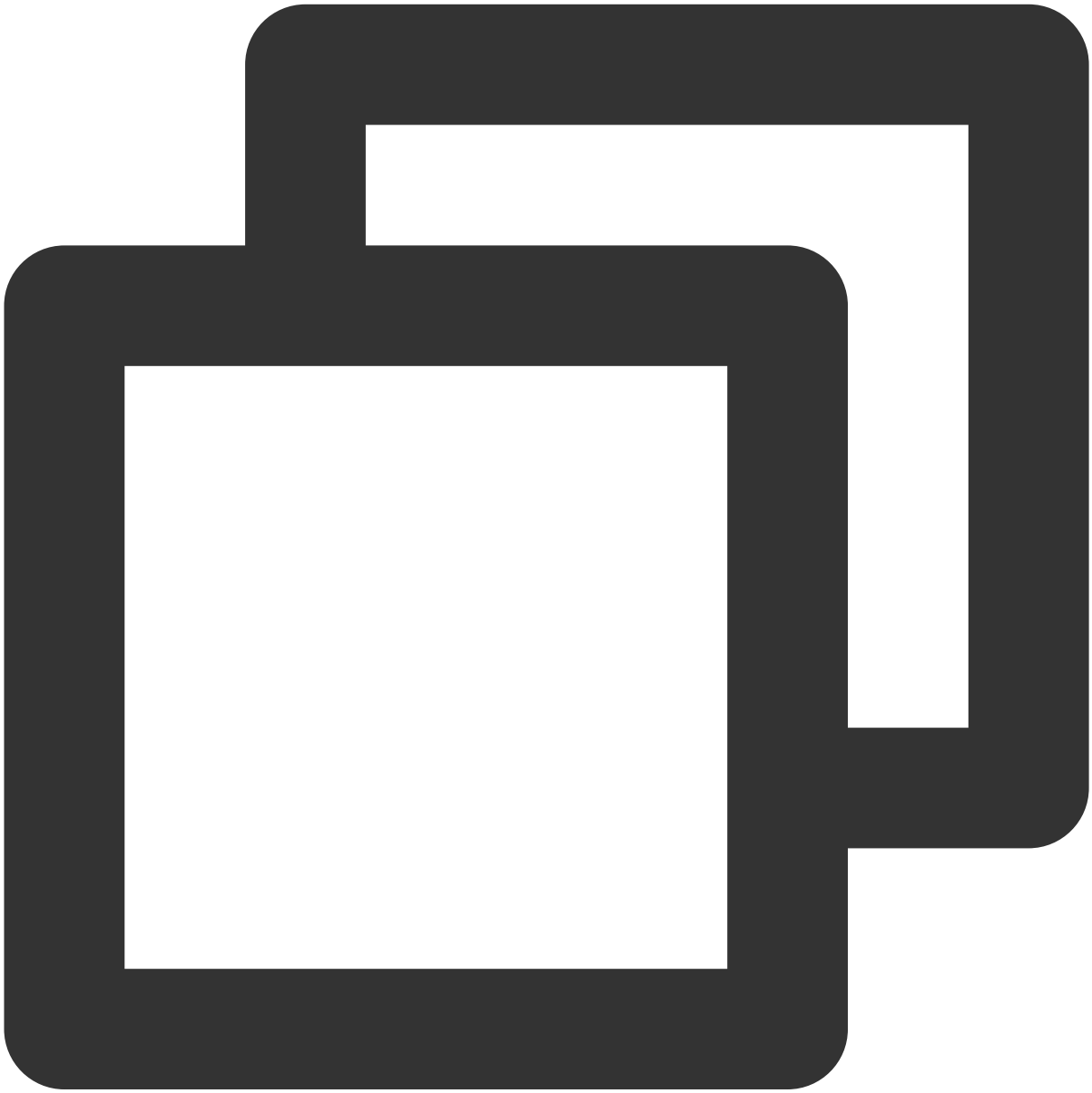
获取当前页



```
/*
* @param: WdInformation: {
*       wdActiveEndPageNumber: 3
*   }
* @return: number
*/
const app = demo.WordApplication()
```

```
const {Enum} = app
let currentPage = await app.ActiveDocument.Selection.Information(Enum.WdInformati
```

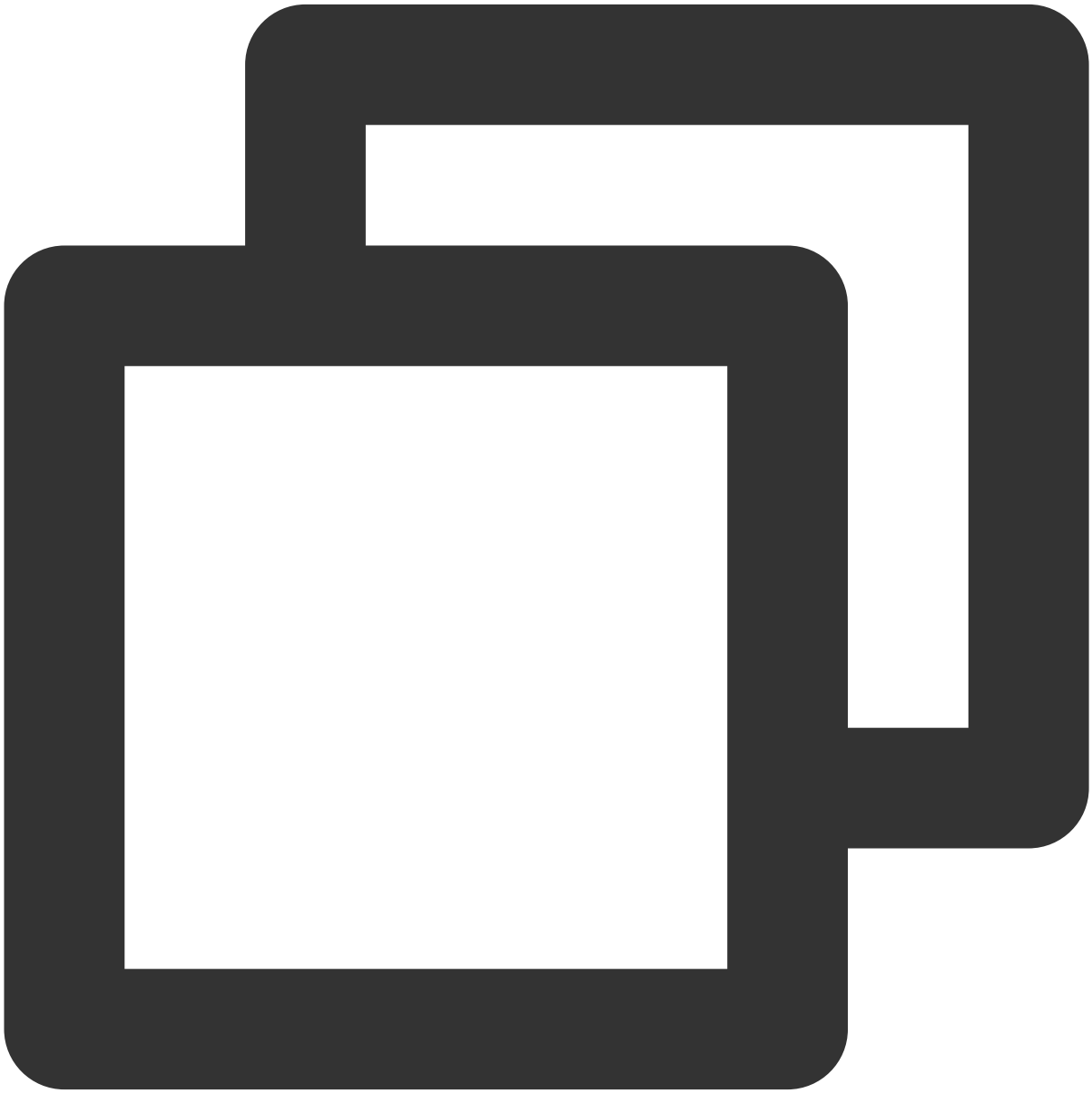
跳转到指定页



```
/*
* @param: { What?: WdGoToItem, Which?: WdGoToDirection.wdGoToAbsolute, Count?: nu
* WdGoToItem: {
*      wdGoToPage: 1,
*  }
*  WdGoToDirection: {
*      wdGoToAbsolute: 1
```

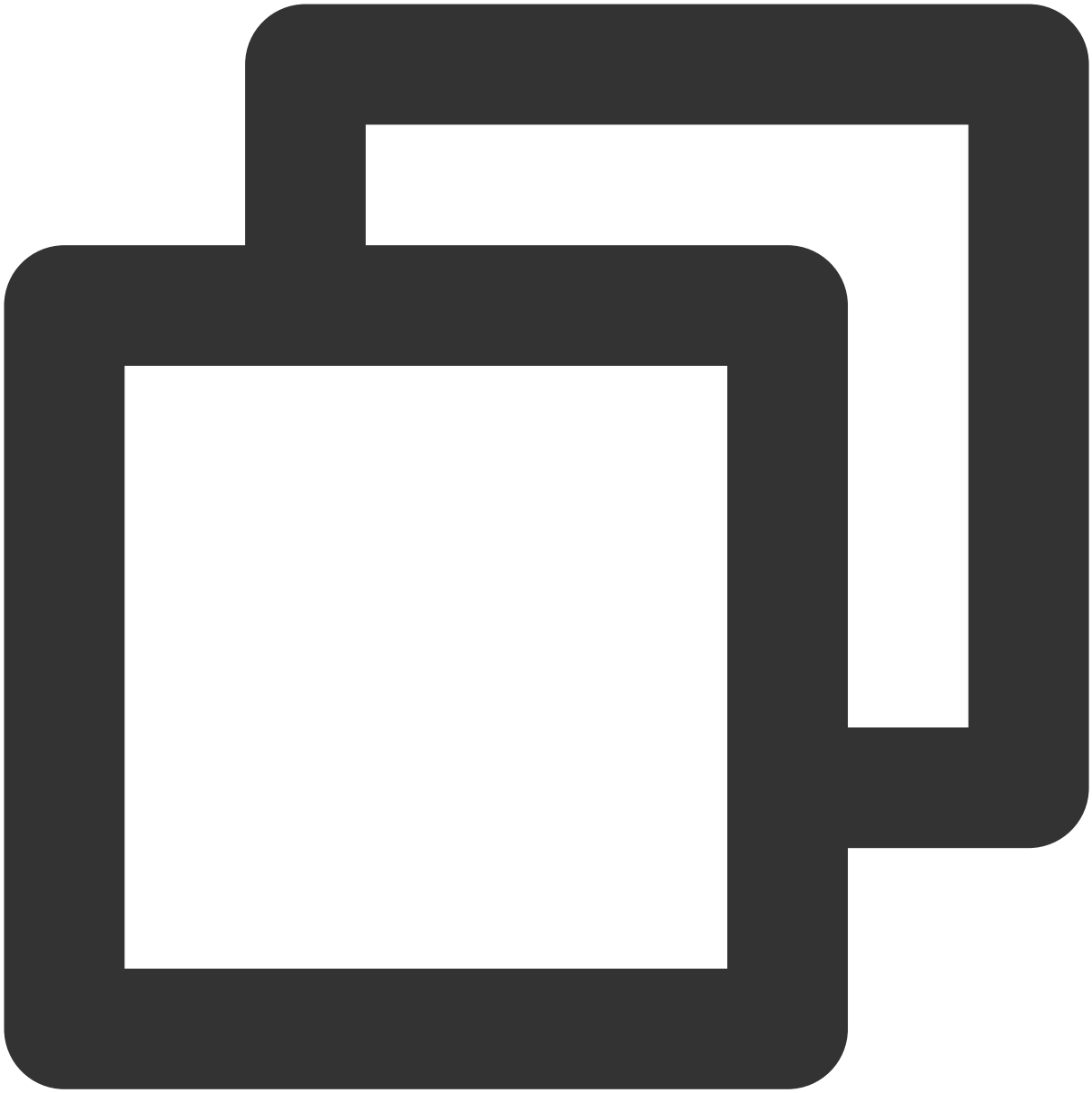
```
    *  }
    * @return: number 返回跳转页
    */
    const app = demo.WordApplication()
    const {Enum} = app
    const page = await app.ActiveDocument.Selection.GoTo(Enum.WdGoToItem.wdGoToPage,
    // 或者
    const page = await app.ActiveDocument.Selection.GoTo({
      What: Enum.WdGoToItem.wdGoToPage,
      Which: Enum.WdGoToDirection.wdGoToAbsolute,
      Count: 10
    })
```

设置评论显示



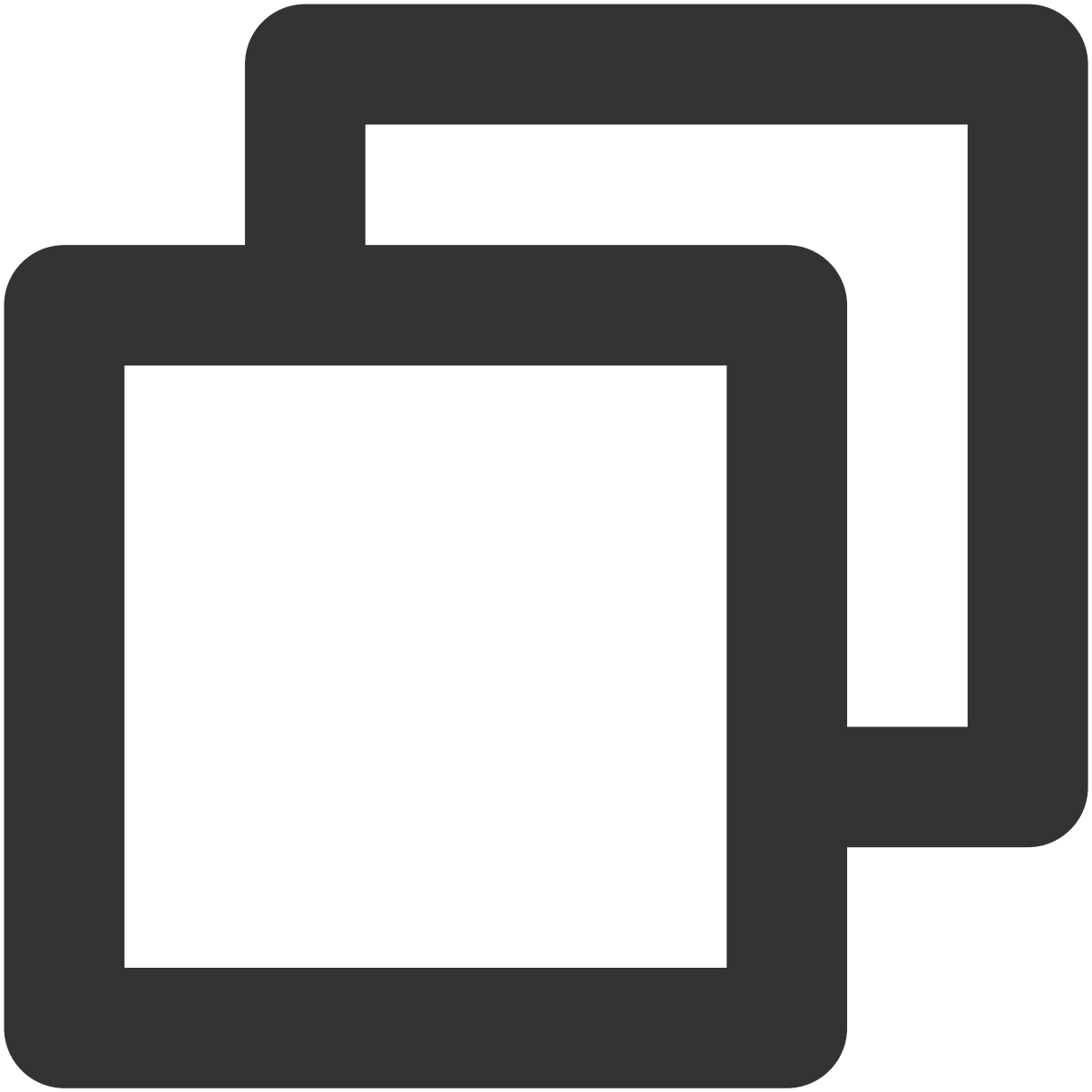
```
/*
* @param: bool
* true 为显示,  false 为隐藏
*/

// 隐藏评论
demo.WordApplication().ActiveDocument.ActiveWindow.View.ShowComments = false
```

设置目录显示



```
/*
* @param: bool
* true 为显示,  false 为隐藏
*/

// 隐藏目录
demo.WordApplication().ActiveDocument.ActiveWindow.DocumentMap = false
```

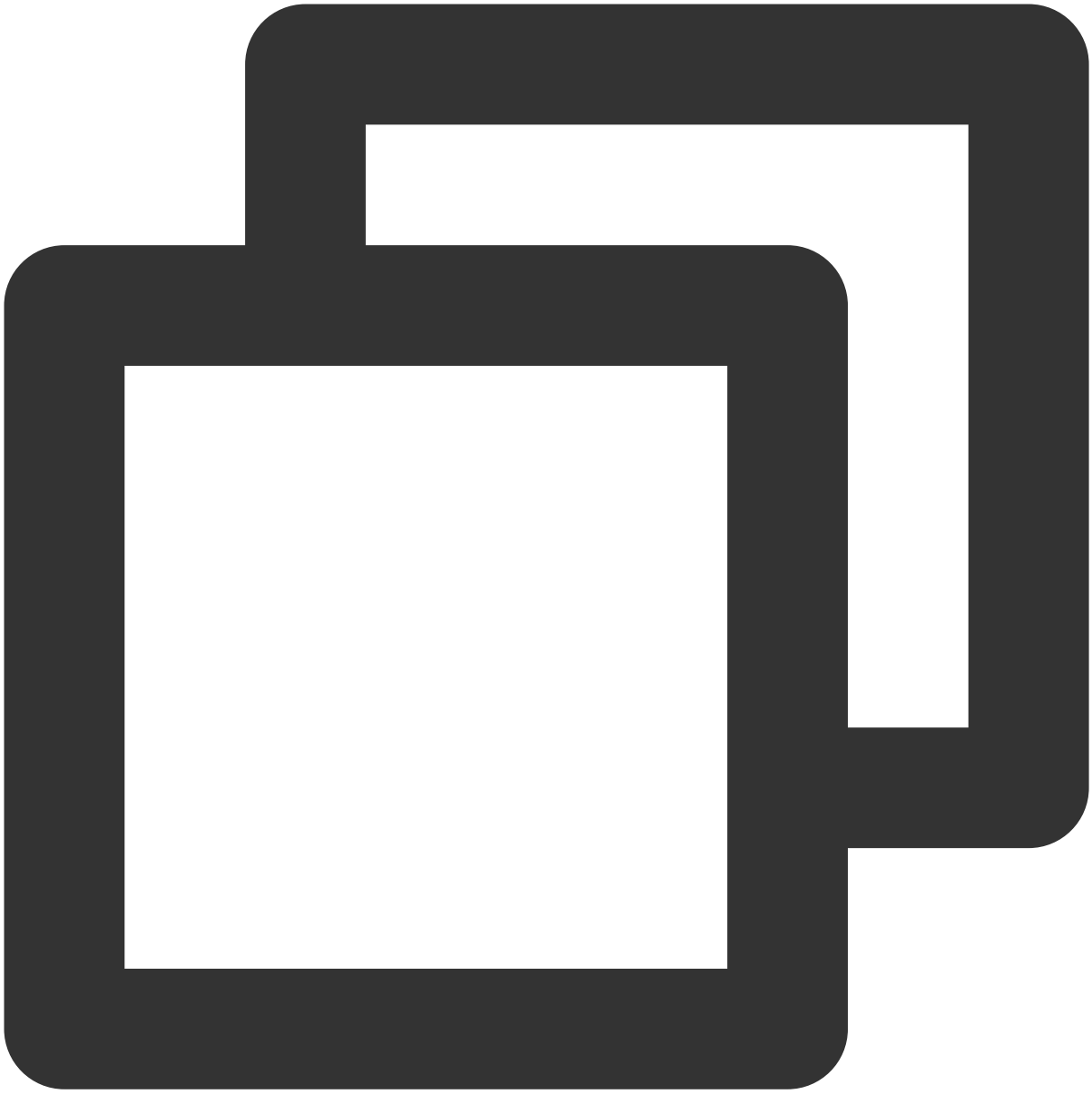
视图缩放



```
/*
* @return: number
*/
await demo.WordApplication().ActiveDocument.ActiveWindow.View.Zoom.Percentage  //

/*
* @param : 50 <= number <= 300 (缩放属性值在50%到 300%之间)
*/
demo.WordApplication().ActiveDocument.ActiveWindow.View.Zoom.Percentage = 100  //
```

表格文件

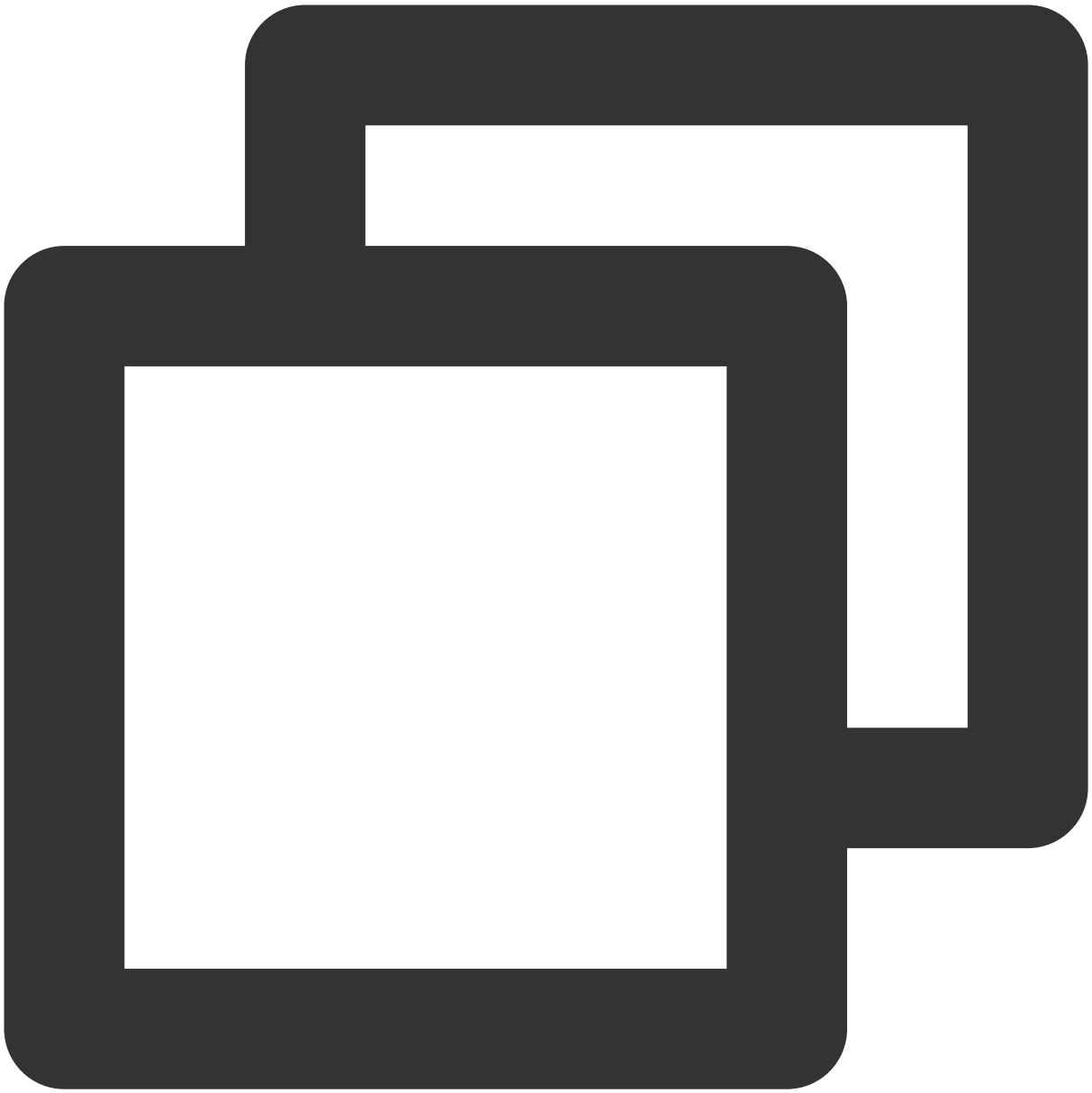
获取所有 sheet 名称



```
const app = demo.ExcelApplication()
const Names = []
// For(start, end, step, handle)
await app.For(1, app.Sheets.Count, 1, async (Index) => {
    Names.push(await app.Sheets.Item(Index).Name)
})
console.log(Names)
```

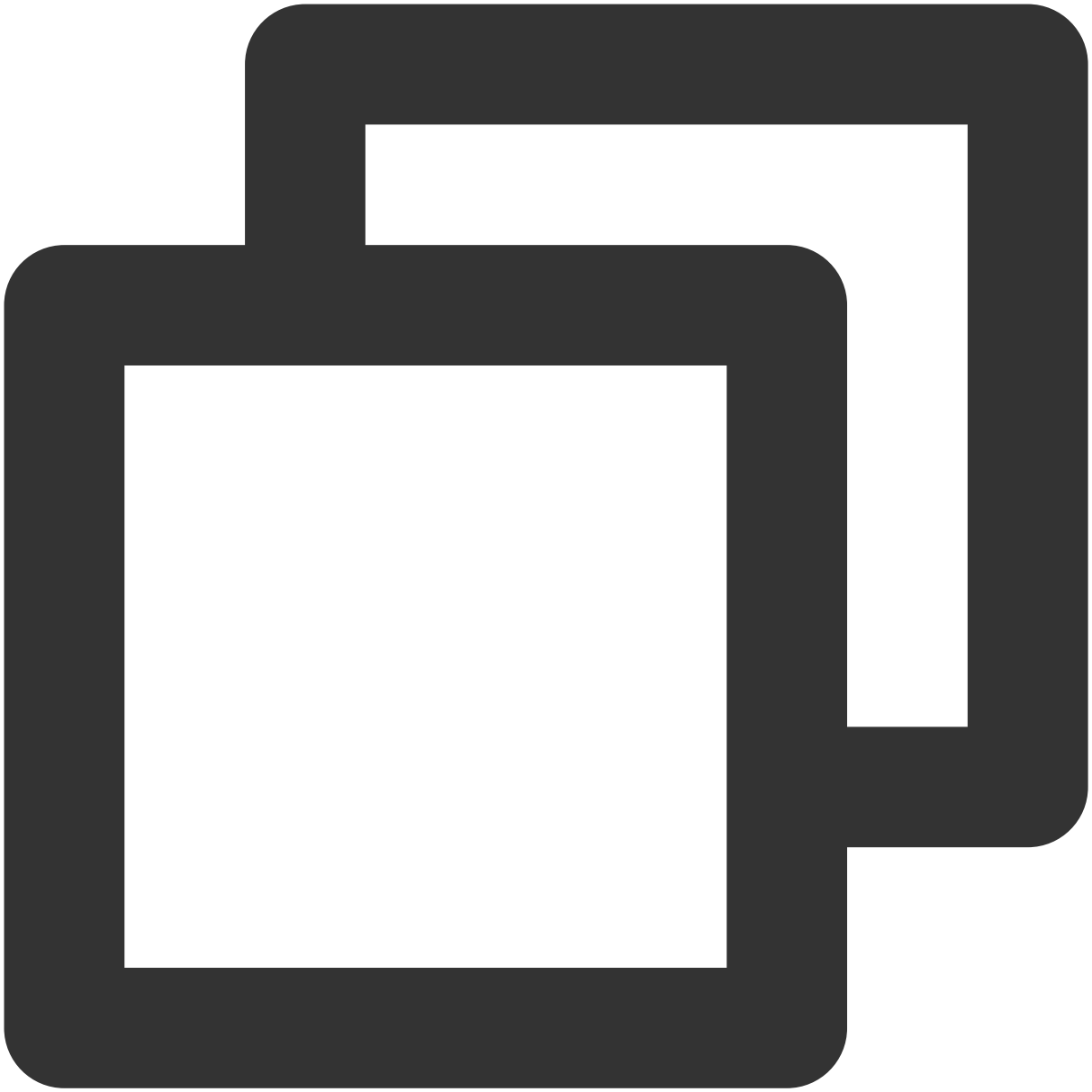
版权所有：腾讯云计算（北京）有限责任公司

获取当前 sheet 名称



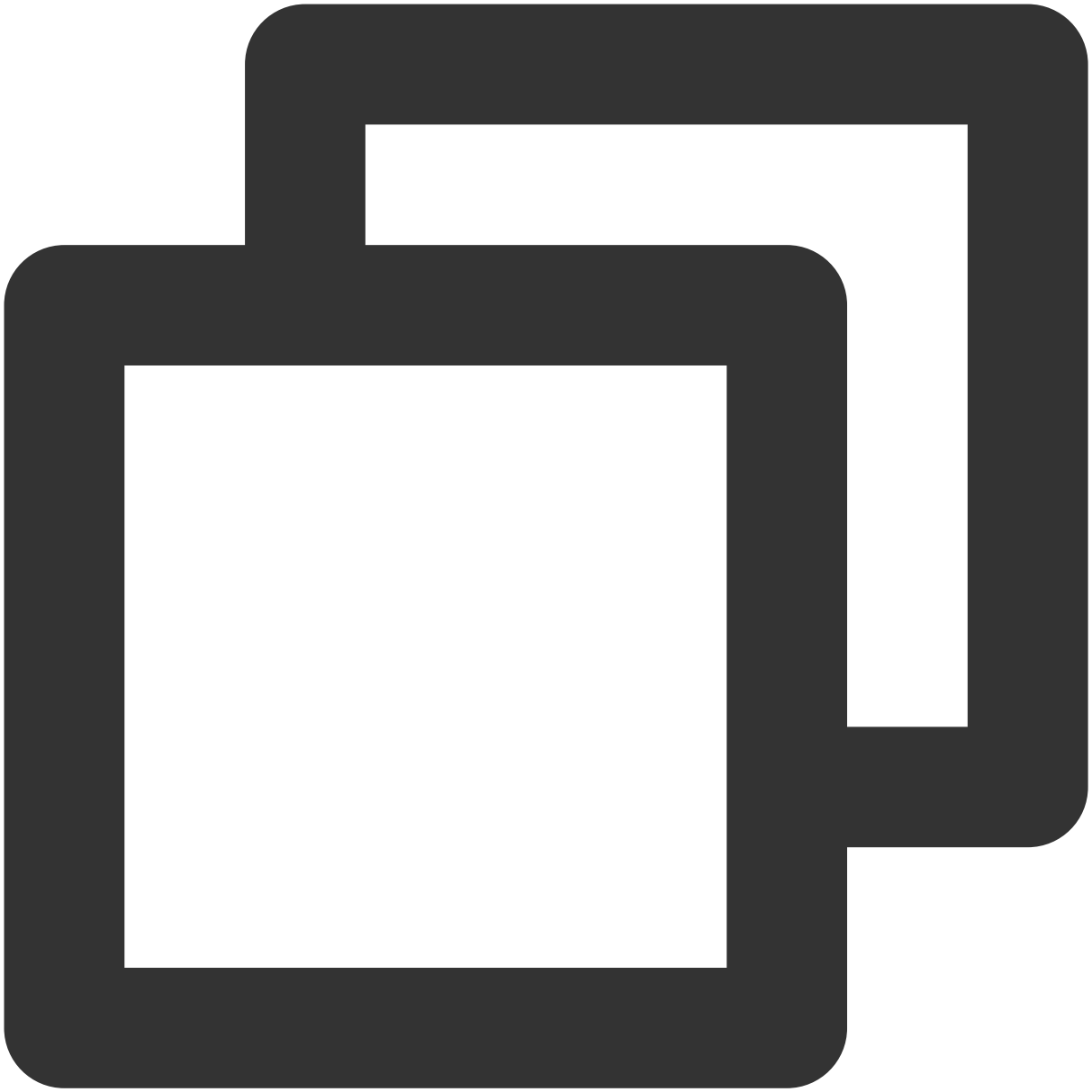
```
const app = demo.ExcelApplication()
const name = await app.ActiveSheet.Name
```

切换到指定 sheet



```
const app = demo.ExcelApplication()
const sheetIndex = 1 // sheets序号,  从1开始
app.Sheets.Item(sheetIndex).Activate() // 切换sheet
```

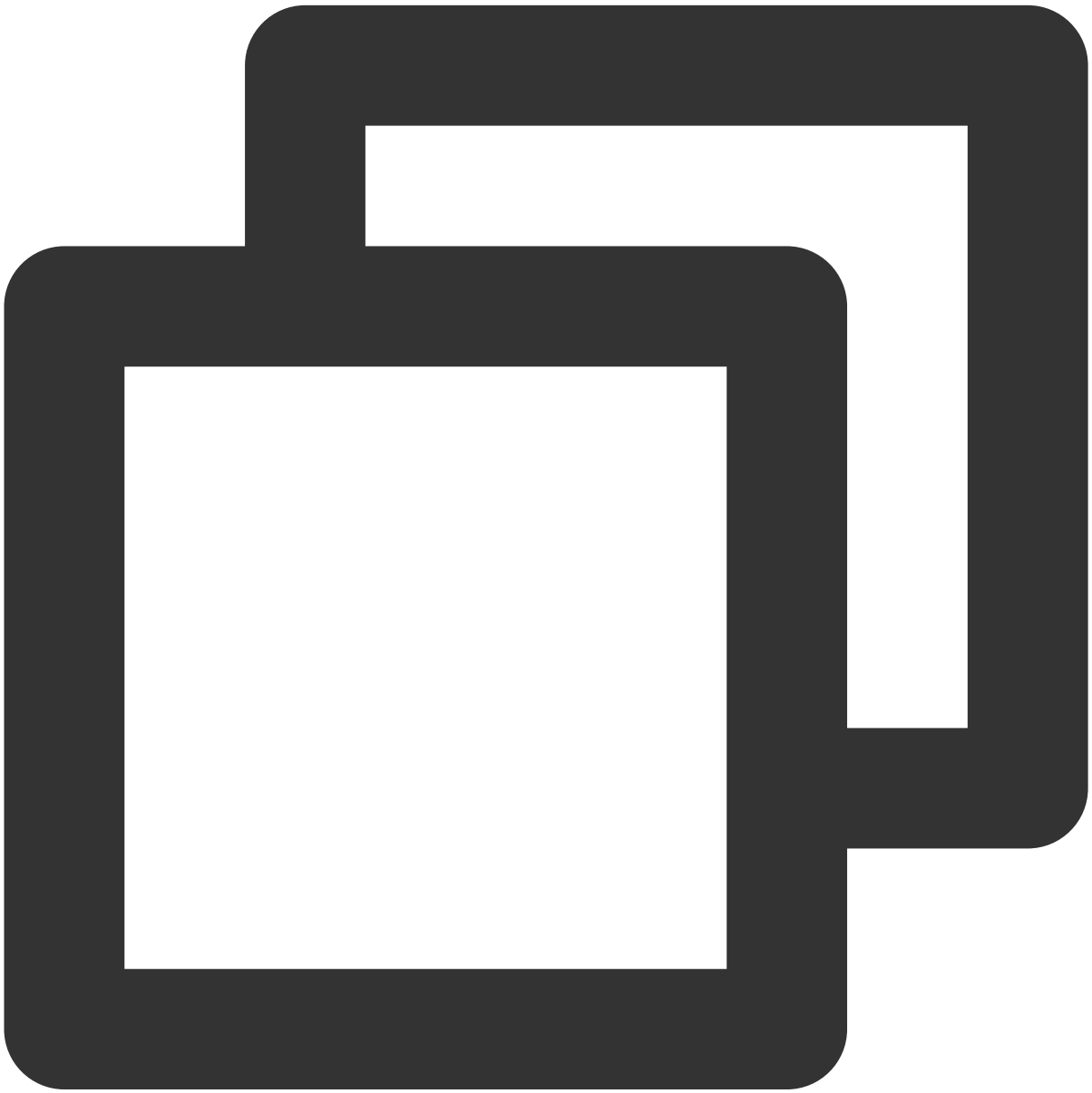
视图缩放



```
/*
* @return: number
*/
await demo.ExcelApplication().ActiveWorkbook.ActiveSheetView.Zoom      // 读取视图缩
/*
* @param : 10 <= number <= 500 (缩放属性值在10%到500%之间)
*/
demo.ExcelApplication().ActiveWorkbook.ActiveSheetView.Zoom = 100      // 设置视图缩
```

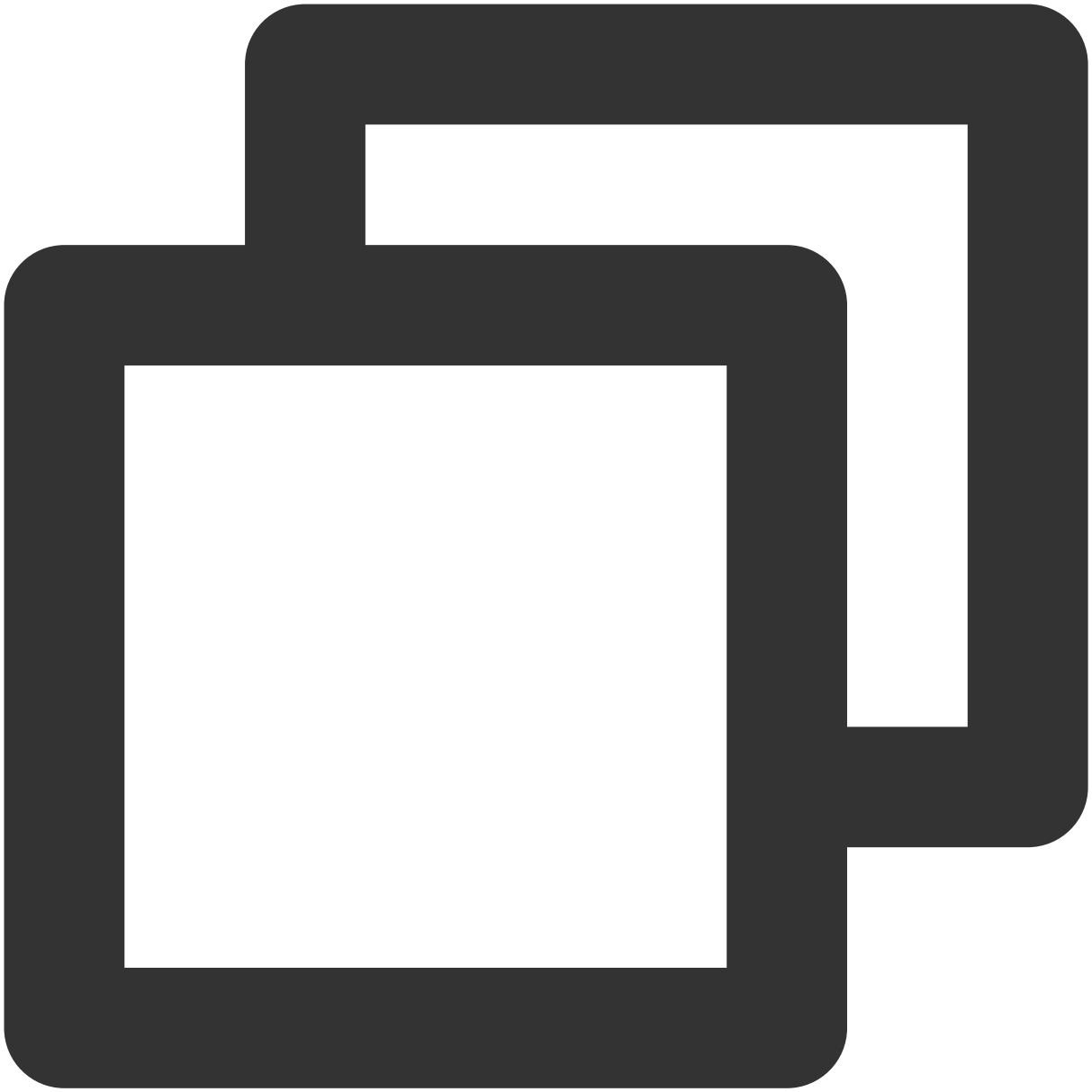
演示文件

获取总页数



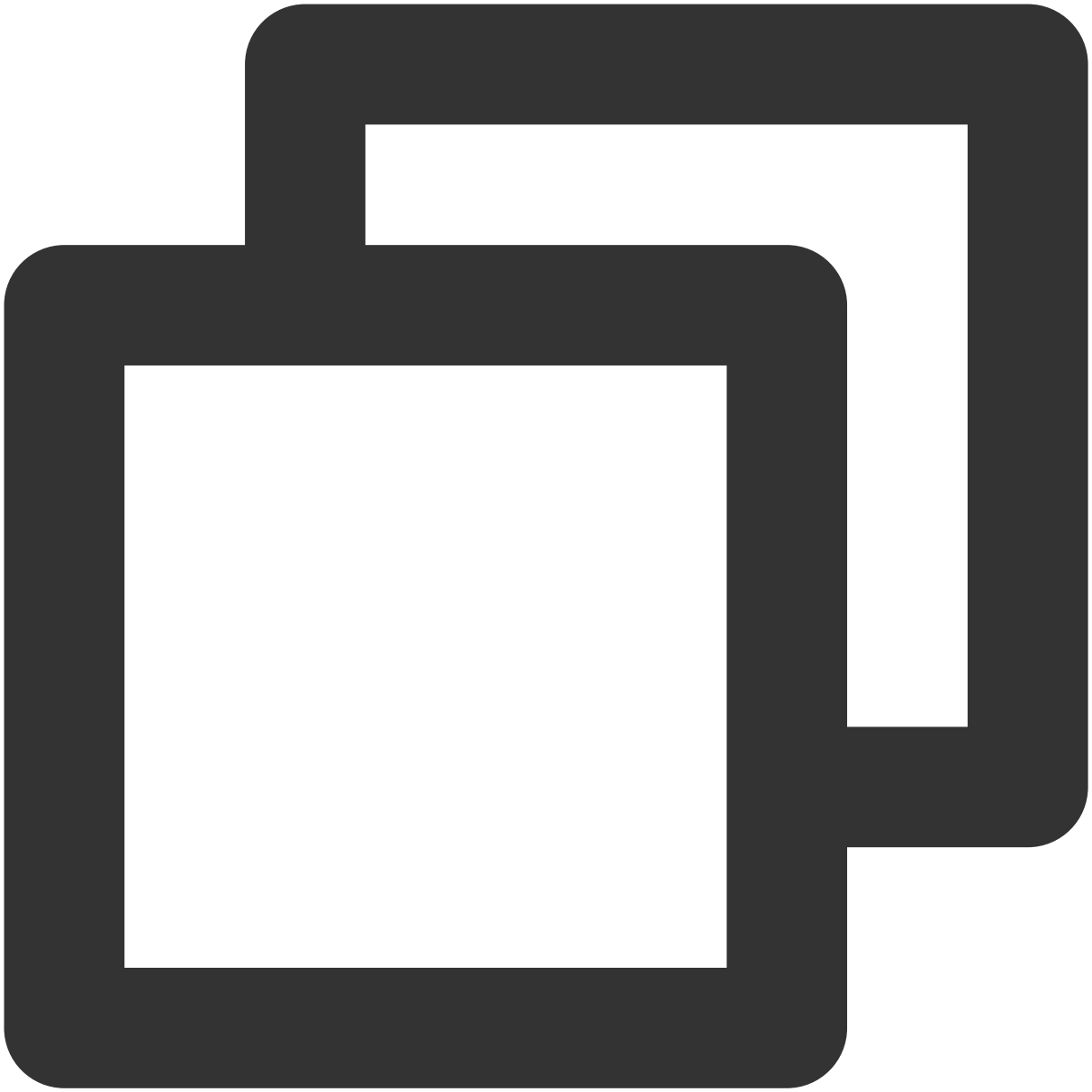
```
/*
* @return: number
*/
let totalPages = await demo.PPTApplication().ActivePresentation.Slides.Count
```

获取当前页



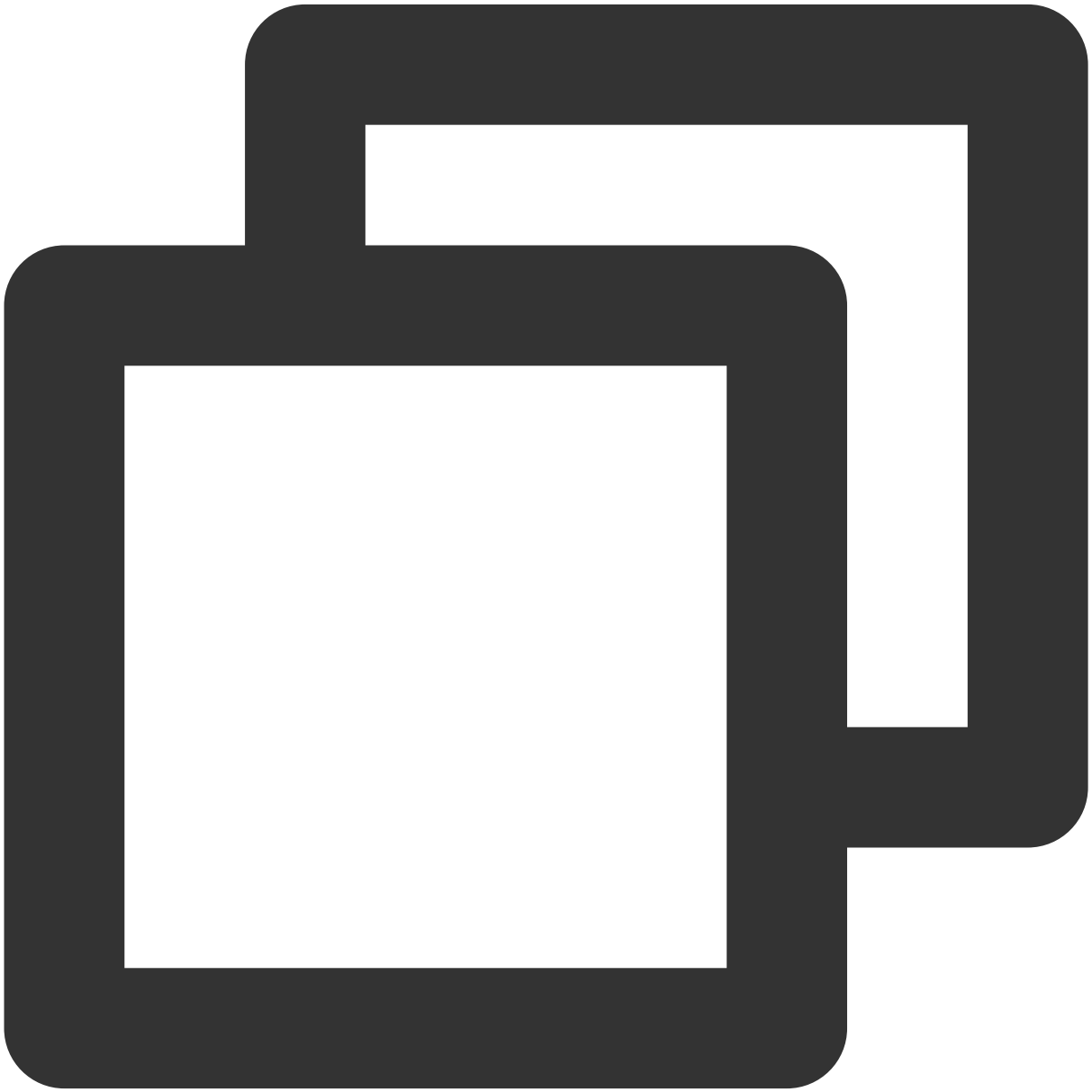
```
/*
* @return: number
*/
let totalPages = await demo.PPTApplication().ActivePresentation.SlideShowWindow.V
```

跳转到指定页



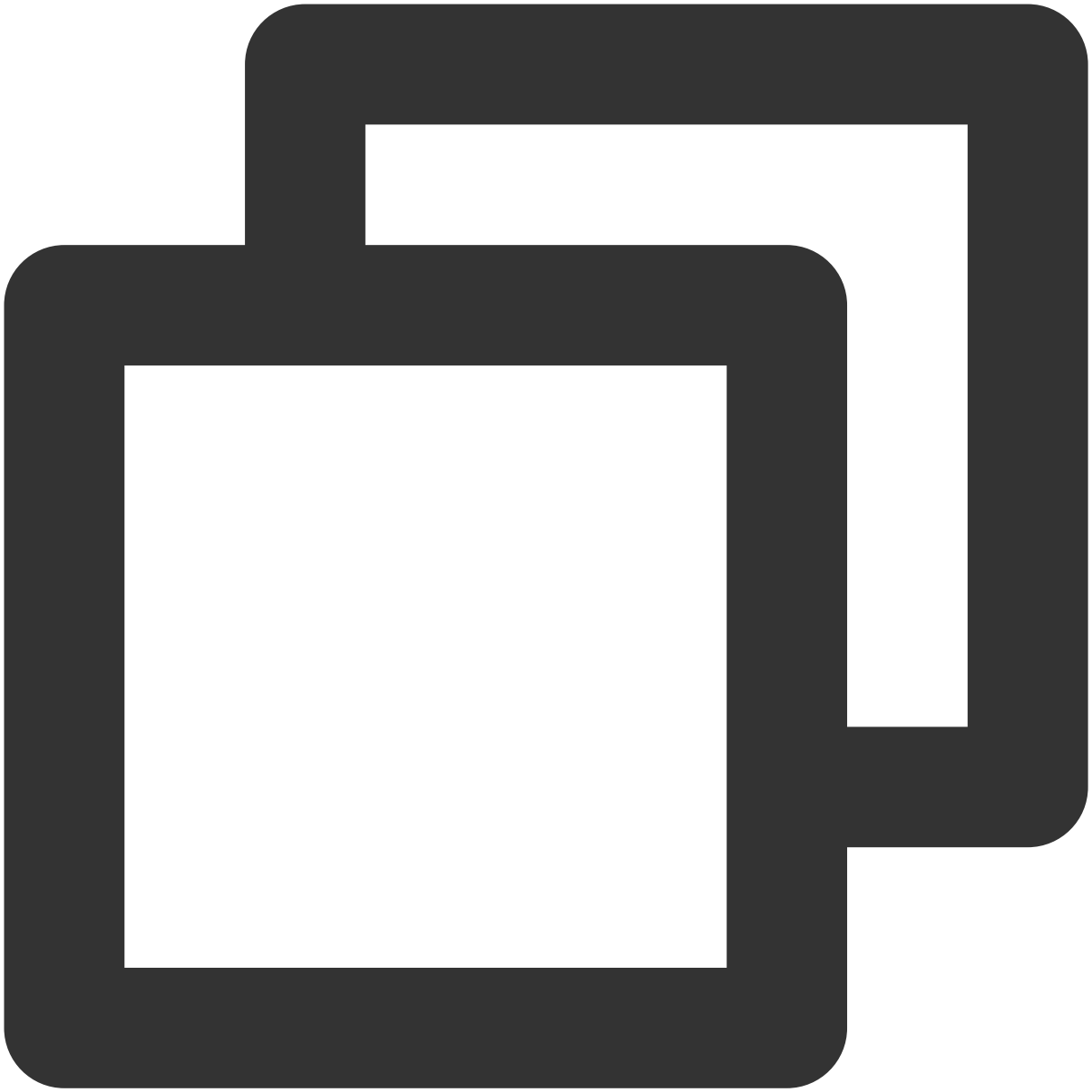
```
/*
* @param: number
*/
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.GotoSlide(3)
```

播放幻灯片



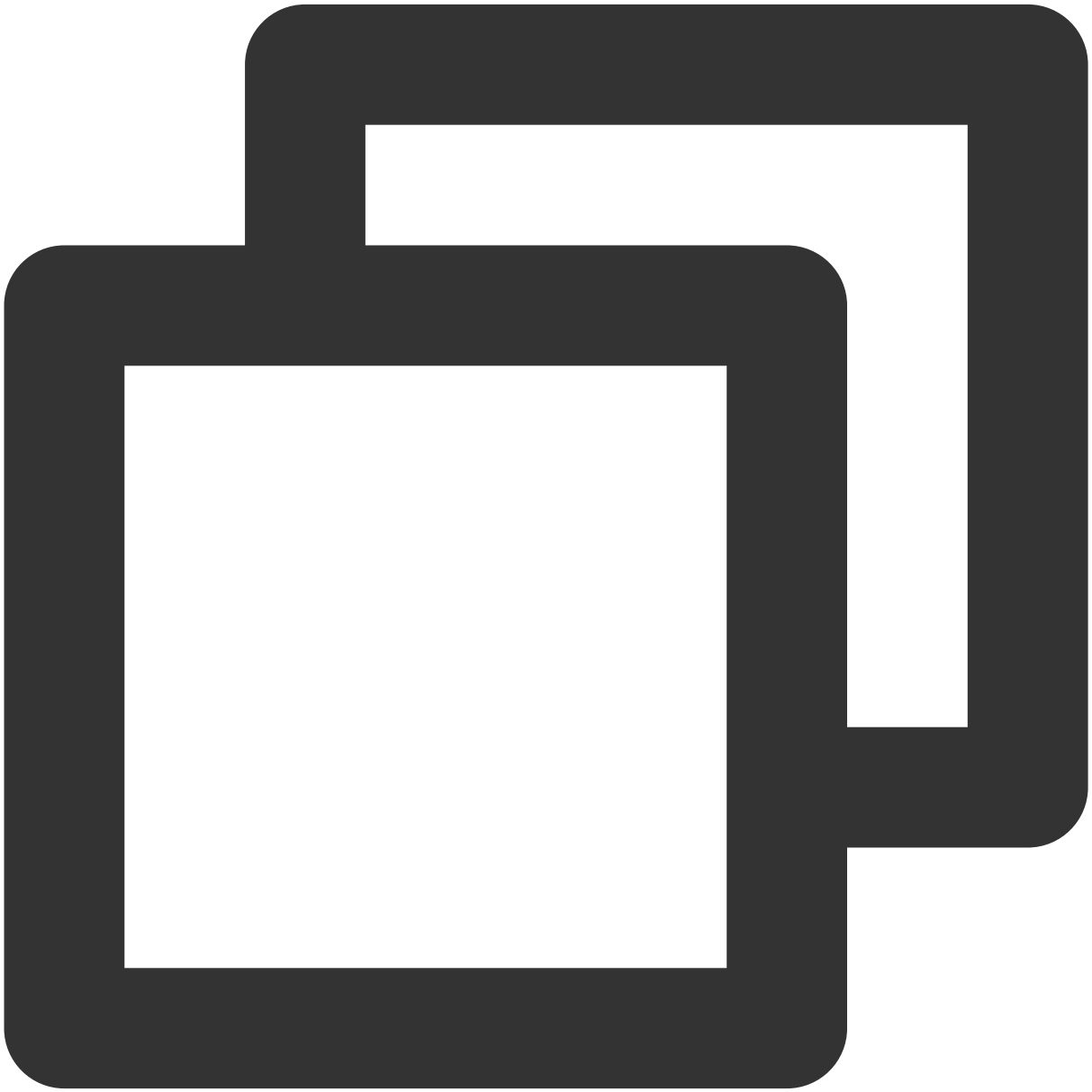
```
await demo.PPTApplication().ActivePresentation.SlideShowSettings.Run()
```

退出播放



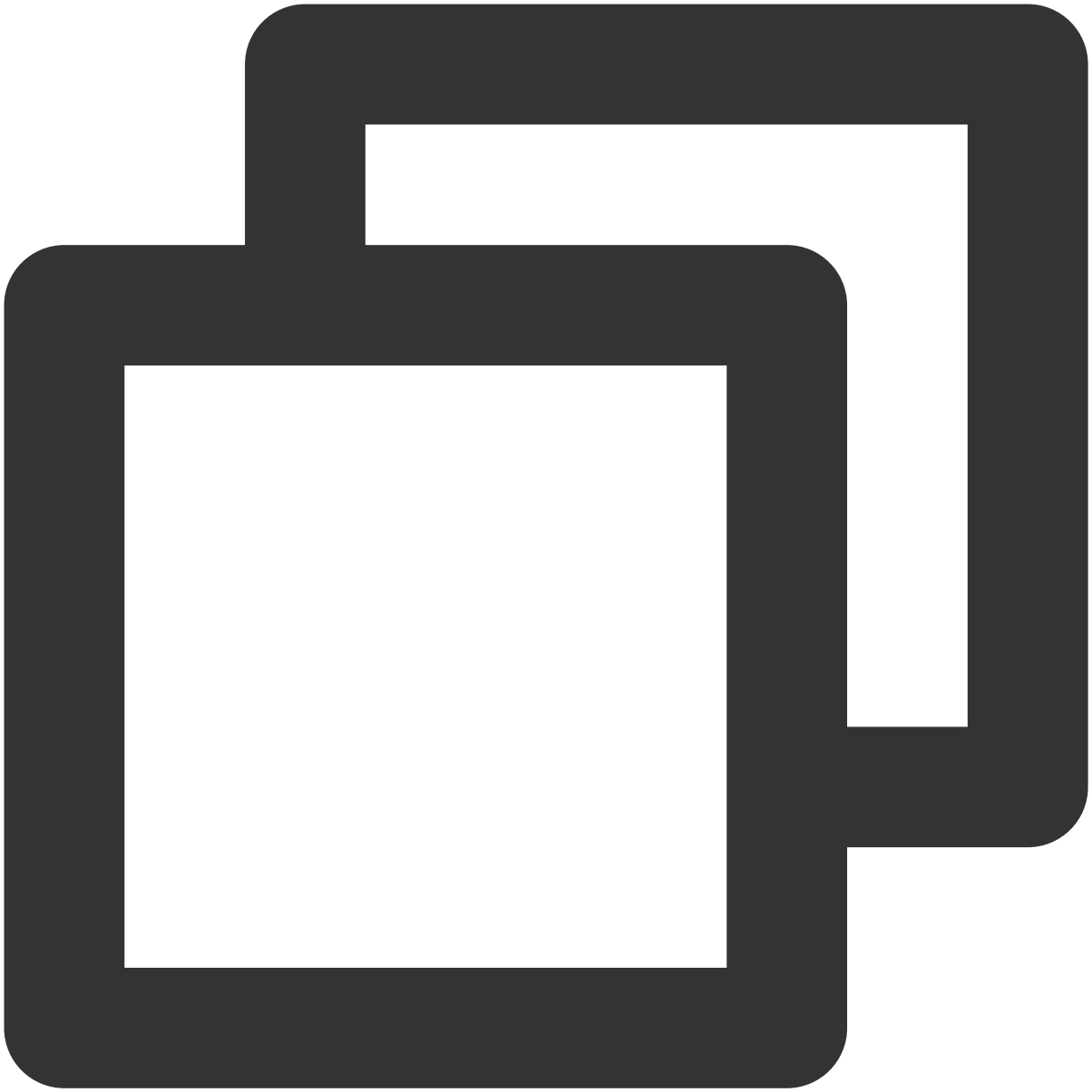
```
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.Exit()
```

幻灯片下一步动画



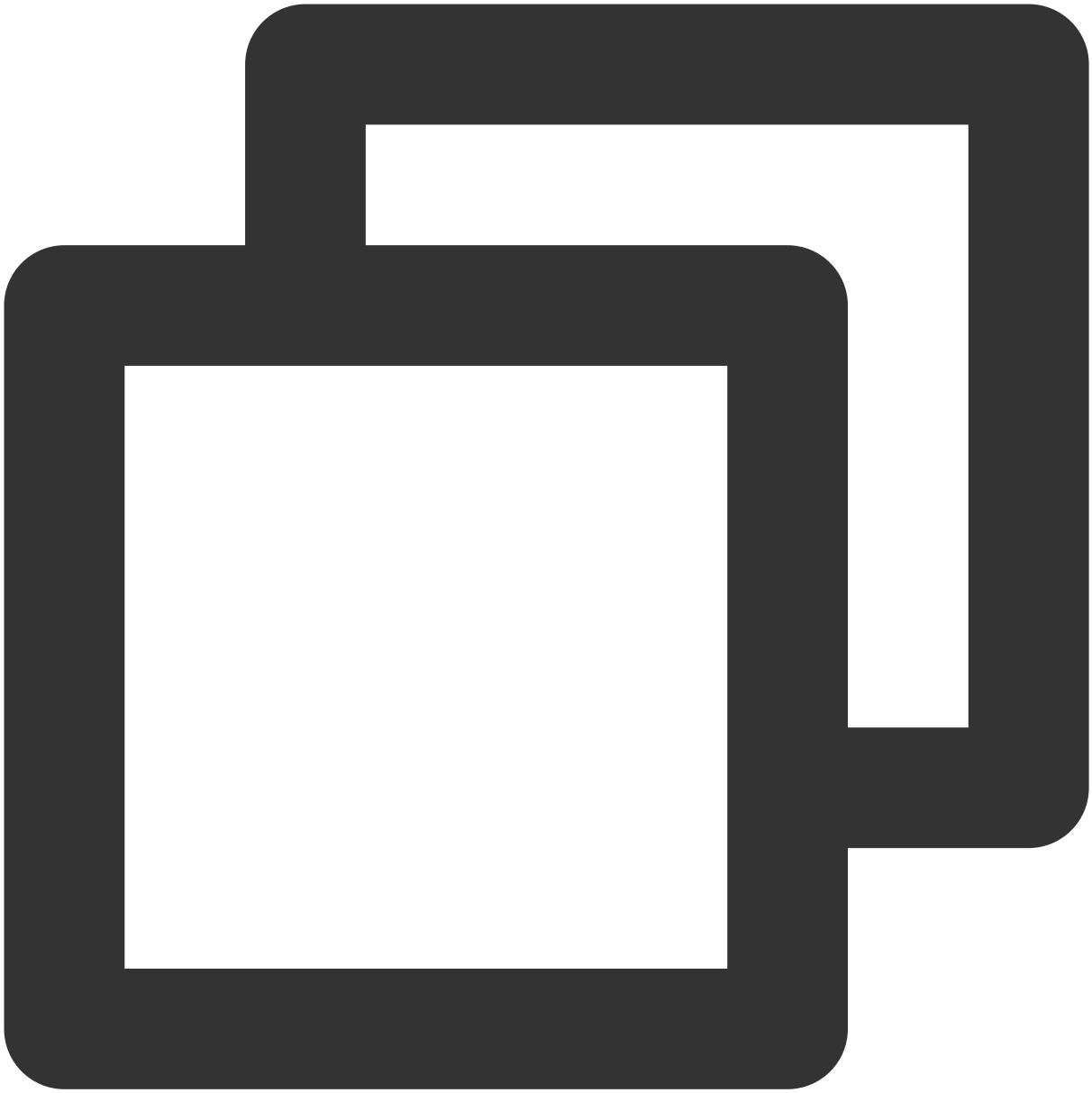
```
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.GotoNextClick
```

幻灯片上一步动画



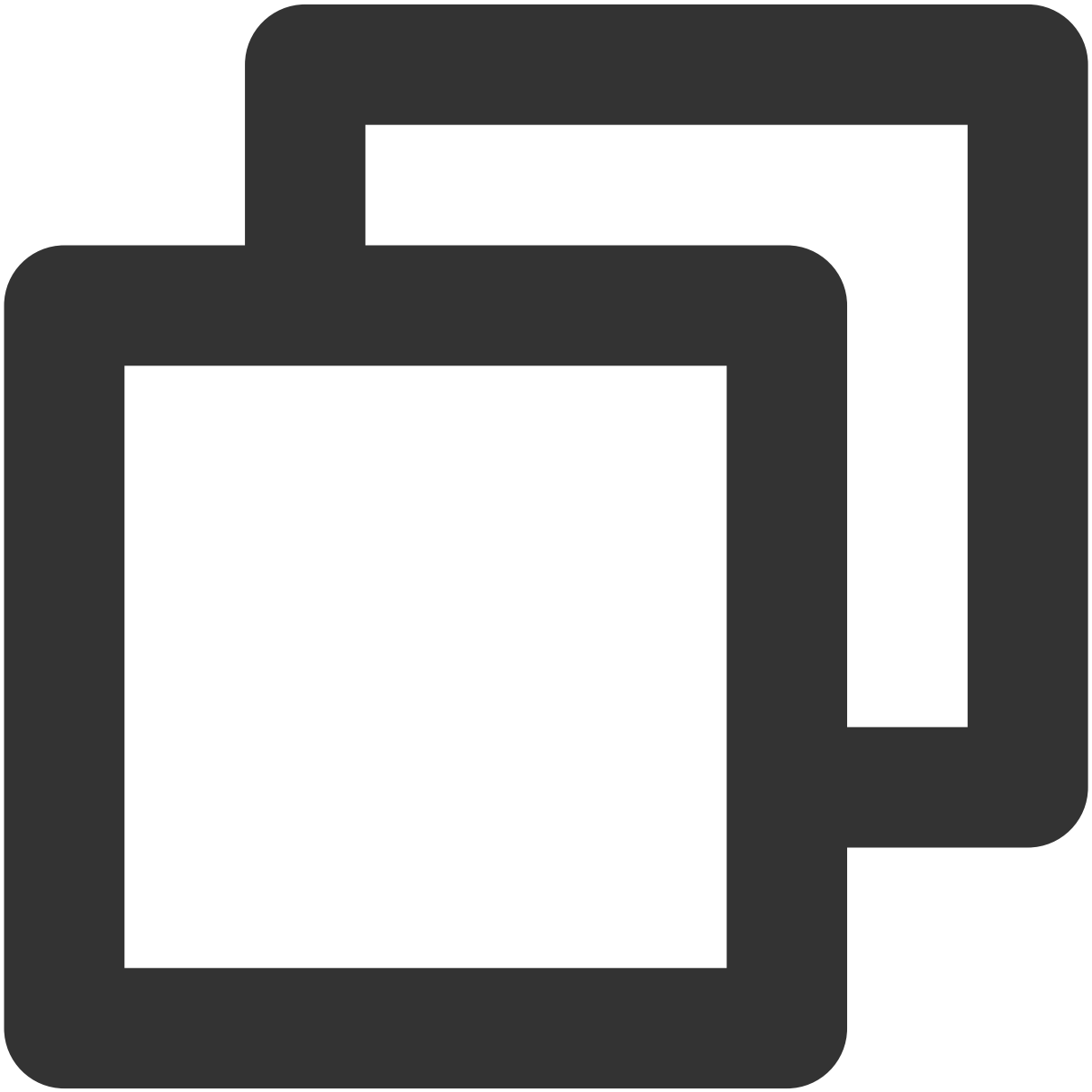
```
    await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.GotoPreClick(
```

获取播放状态



```
/*
* @return: string ('preview' | 'play')
*/
let currentState = await demo.PPTApplication().ActivePresentation.SlideShowWindow
```

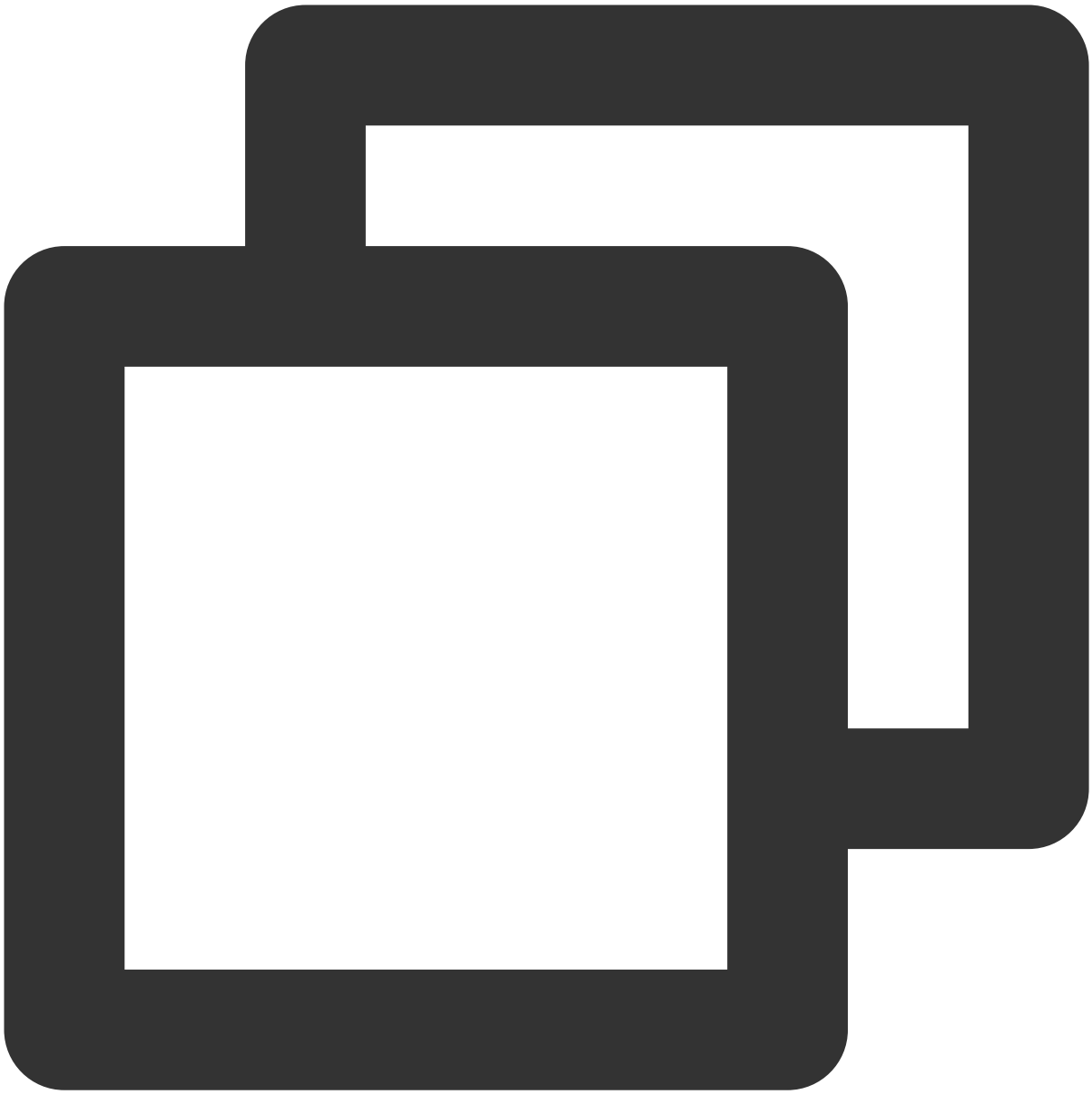
视图缩放



```
/*
* @return: number
*/
await demo.PPTApplication().ActivePresentation.View.Zoom       // 读取视图缩放比例
/*
* @param : 1 <= number <= 400 (缩放属性值在1%到400%之间)
*/
demo.PPTApplication().ActivePresentation.View.Zoom = 100       // 设置视图缩放比例（非
```

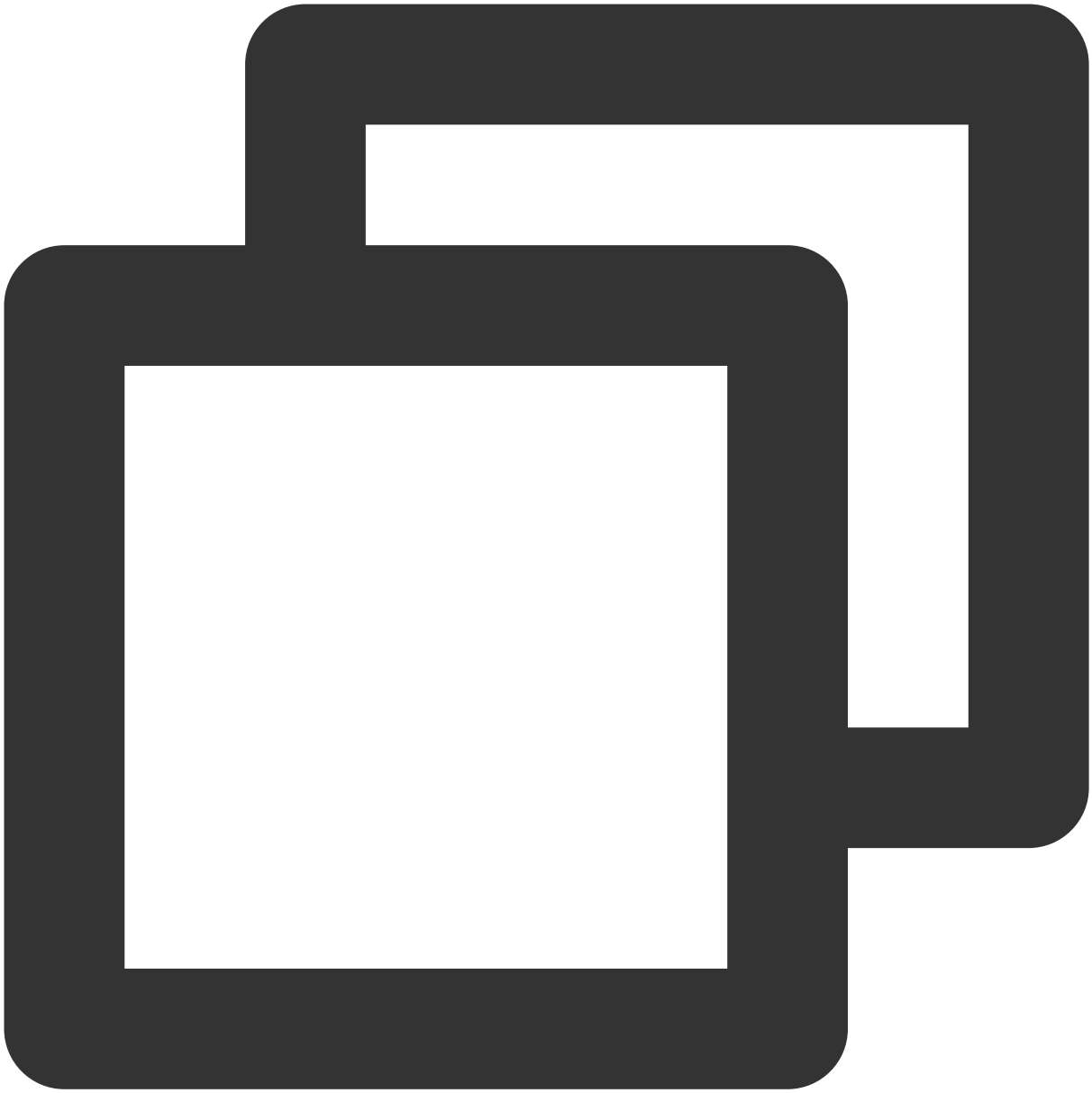
PDF 文件

获取总页数



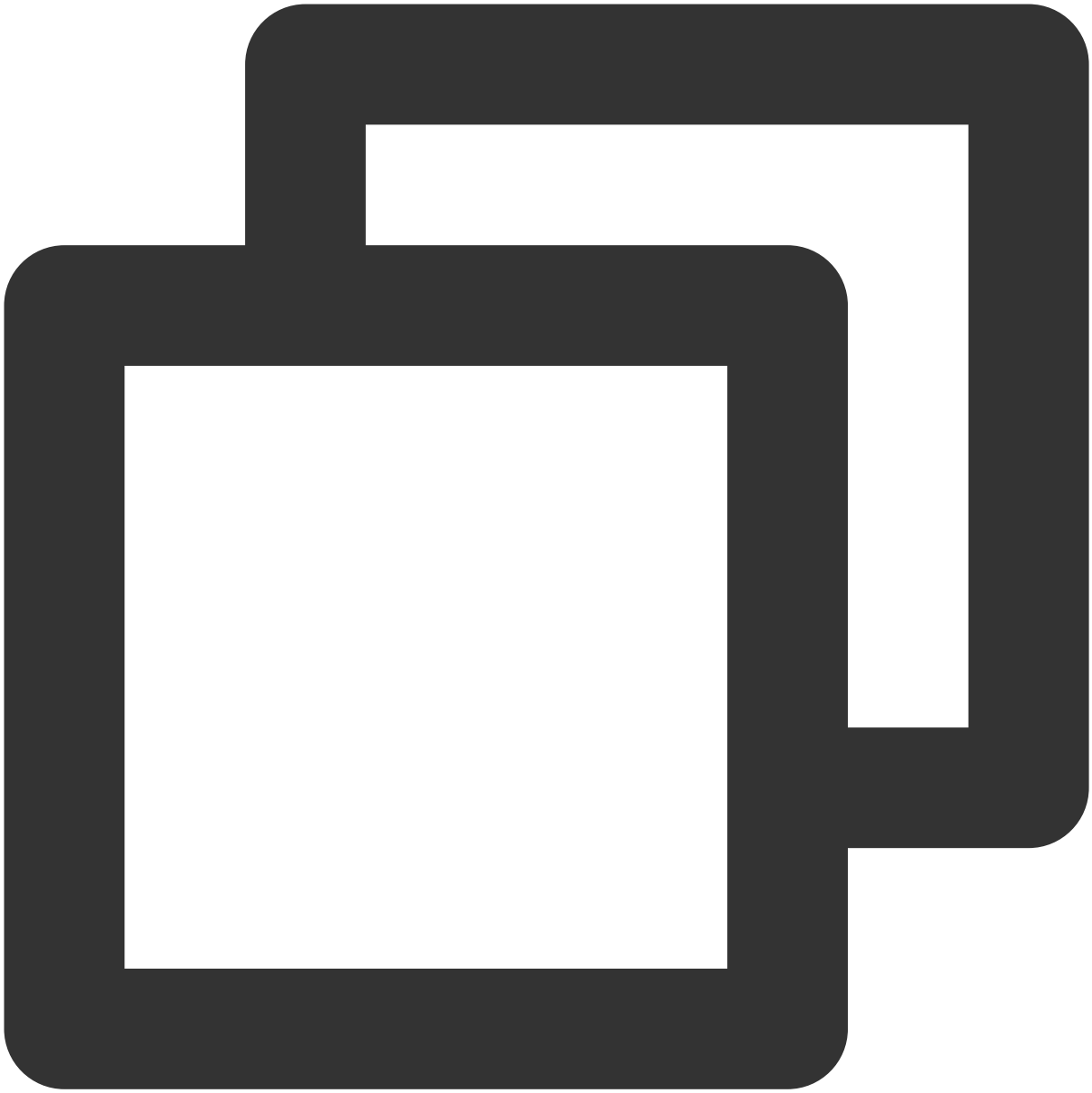
```
/*
* @return: number
*/
let totalPages = await demo.PDFApplication().ActivePDF.PagesCount
```

获取当前页



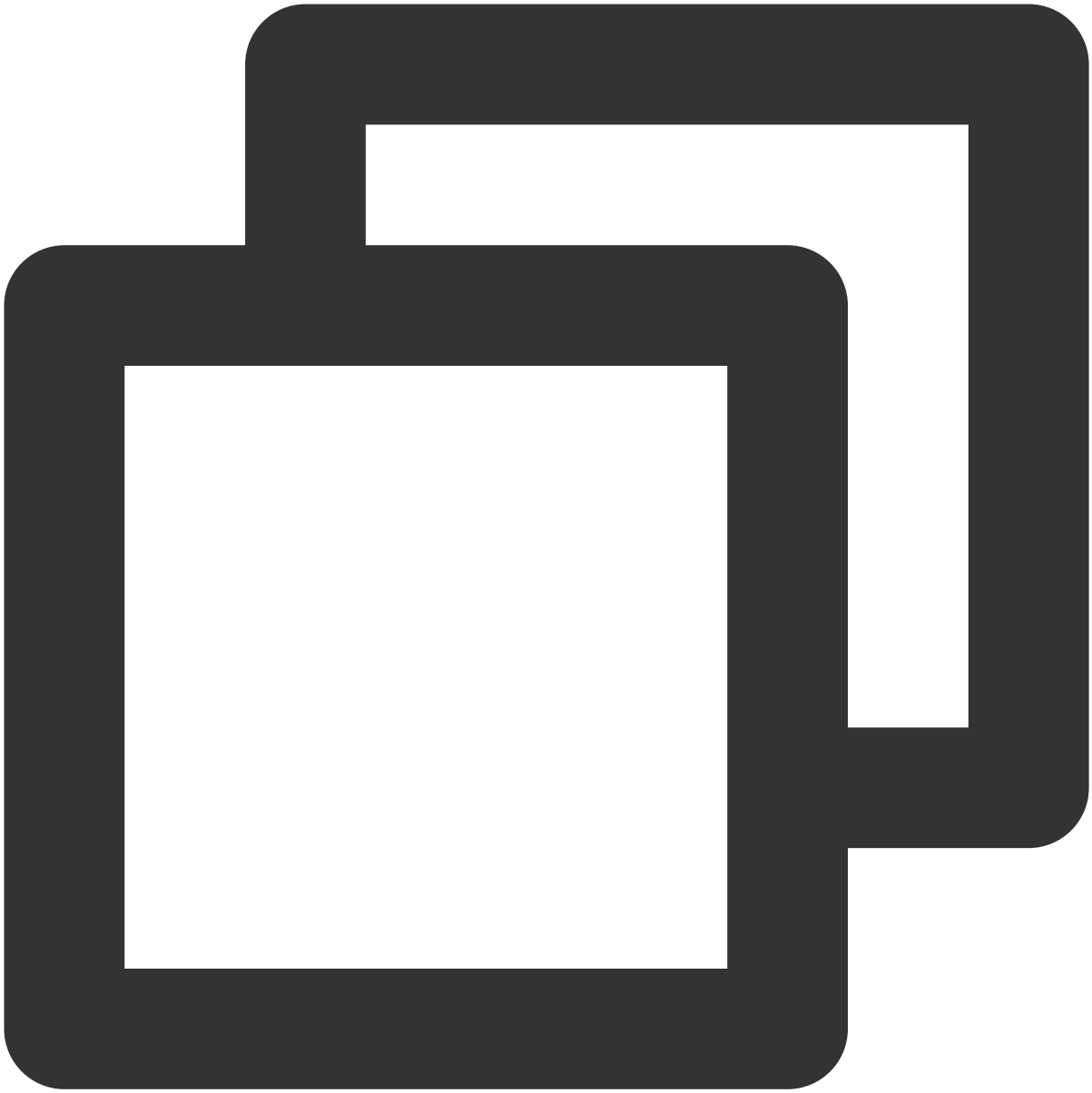
```
/*
* @return: number
*/
let totalPages = await demo.PDFApplication().ActivePDF.CurrentPage
```

跳转到指定页



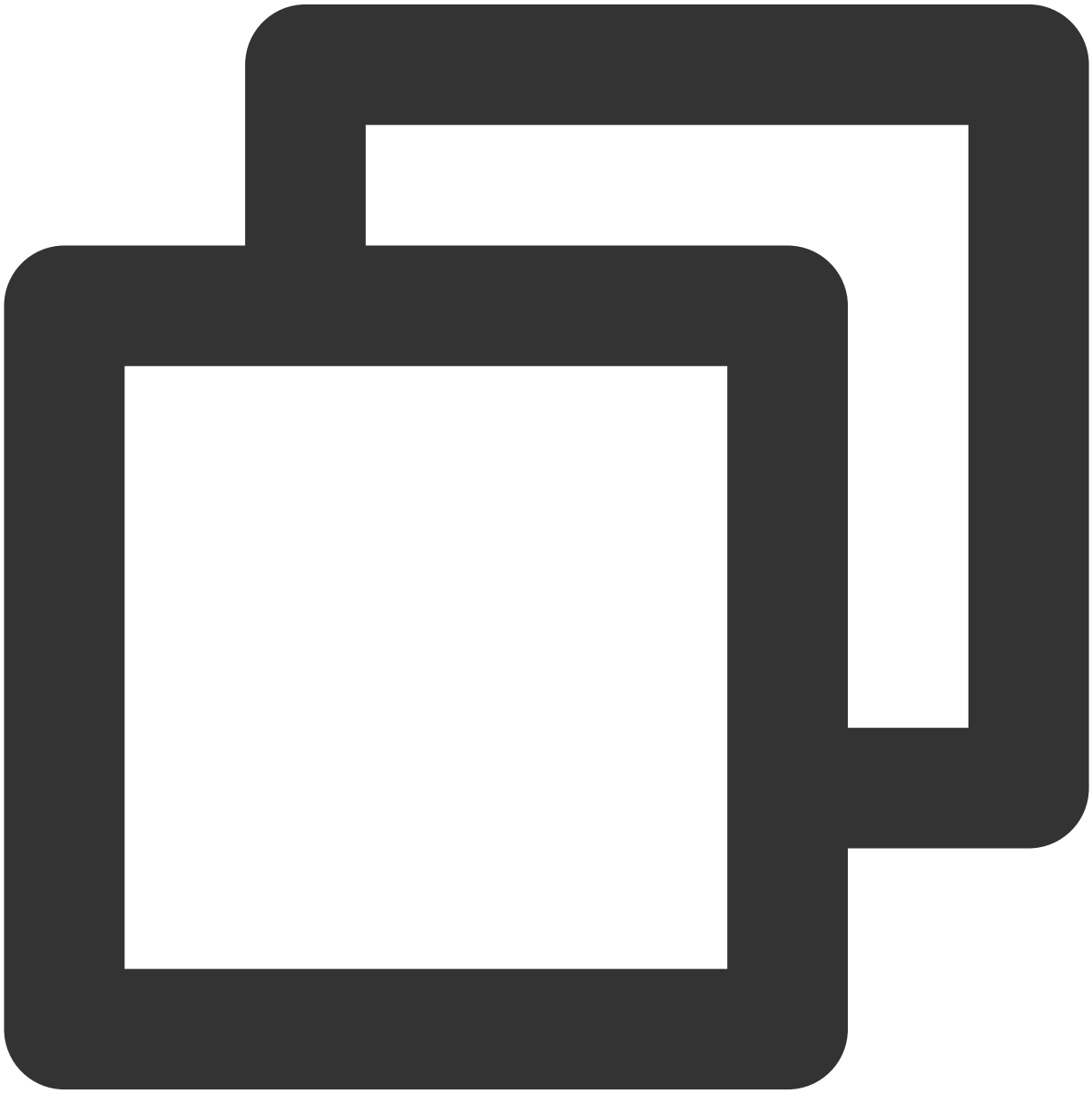
```
/*
* @param : { PageNum: number }
*/
let PageNum = 10
await demo.PDFApplication().ActivePDF.JumpToPage({PageNum})
```

设置页面渲染模式



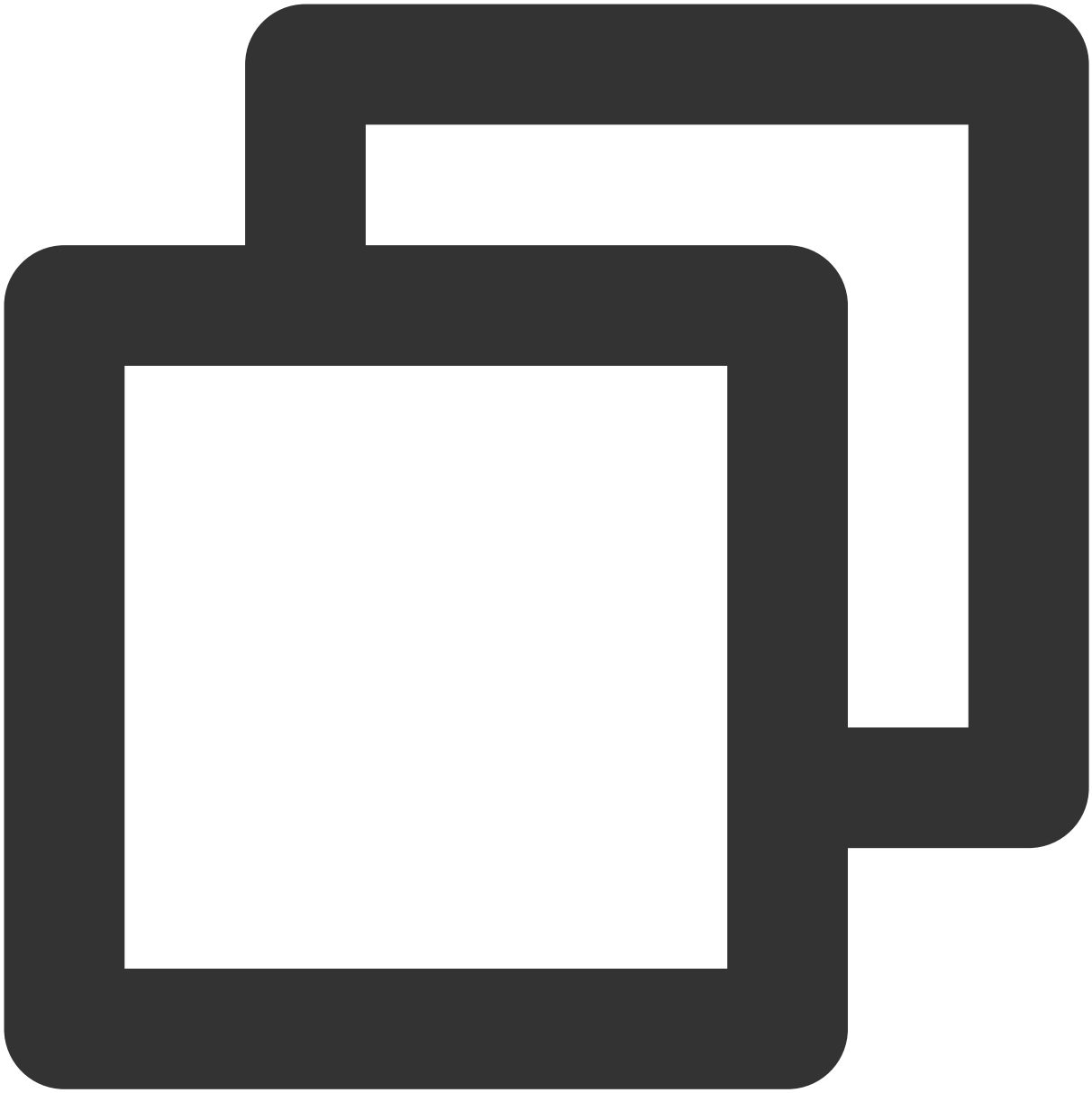
```
/*
* 1：单页模式
* 0：多页模式
*/

demo.PDFApplication().ActivePDF.PageMode = 1     // 设置为单页模式
```

设置目录显示



```
/*
* true：显示目录
* false：隐藏目录
*/
demo.PDFApplication().ActivePDF.DocumentMap = true     // 显示目录
```

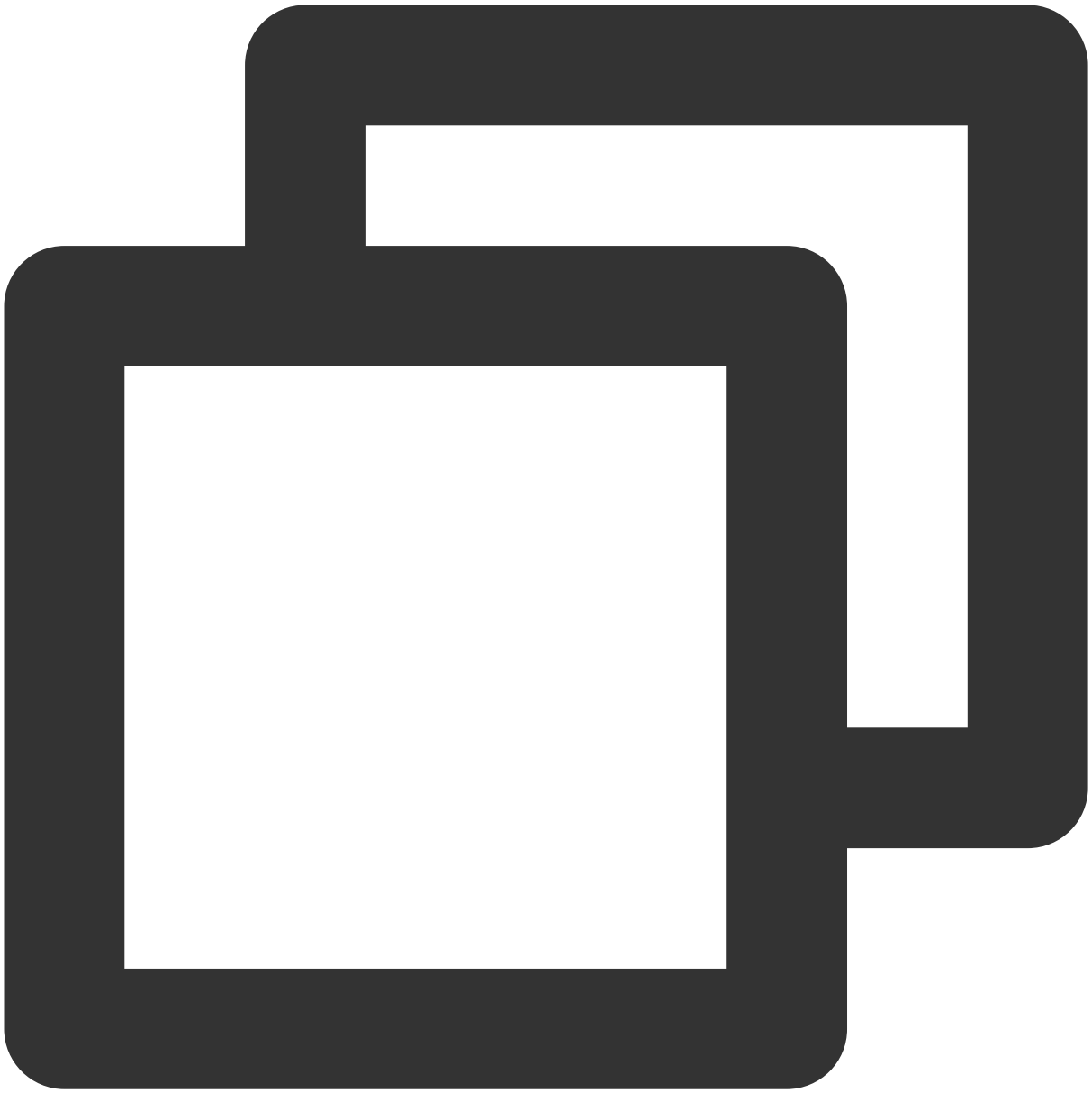
视图缩放



```
/*
* @return: number
*/
await demo.PDFApplication().ActivePDF.Zoom  // 读取视图缩放比例
/*
* @param : 1 <= number <= 400 (缩放属性值在1%到400%之间)
*/
demo.PDFApplication().ActivePDF.Zoom = 100  // 设置视图缩放比例
```

事件监听

可以通过实例的 on 方法监听事件。



```
demo.on('事件名', function(data) {
    // do something...
})
```

事件名	说明
fileOpen	文件打开成功或者失败时的事件回调
error	错误发生时的事件回调，会返回错误码

fileStatus	文件保存的事件回调，会返回状态 status 值：0表示文档无更新，1表示版本保存成功，2表示暂不支持保存空文件，3表示空间已满，4表示保存中请勿频繁操作，5表示保存失败，6表示文件更新保存中，触发场景：修改文档内容触发的保存，7表示文档内容修改保存成功

版权所有：腾讯云计算（北京）有限责任公司

支持的字体和动画

最近更新时间：2023-12-22 14:17:02

简介

文档转 HTML 功能支持预览多种字体，并可预览演示文件中的动画效果，具体请参见下方的字体支持表和动画支持表。

说明

使用文档转 HTML 功能，实际的预览效果可能因文档格式的不同而有所差异。

支持的字体

文档预览支持如下所示的字体，部分字体**仅支持 Word 文档**。不支持的字体会在展示时匹配相近字体，可能会出现字体效果与本地不符的情况。

中文字体（简/繁）

仅支持 Word 文档

字体名称	字体名称	字体名称
幼圆	华文行楷	方正姚体
新宋体	华文仿宋	方正舒体
微软雅黑	华文彩云	等线
微软雅黑 Light	黑体	等线 Light
腾讯体 W3	汉仪智楷繁	PMingLiU-ExtB （新細明體-ExtB）
腾讯体 W7	汉仪智草繁	MingLiU-ExtB （細明體-ExtB）
宋体	汉仪赵楷繁	MingLiU_HKSCS-ExtB （細明體_HKSCS-ExtB）
隶书	汉仪王行繁	Microsoft YaHei UI
楷体	汉仪唐隶繁	Microsoft YaHei UI Light
华文中宋	汉仪欧行繁	Microsoft JhengHei UI

华文新魏	汉仪陆行繁	Microsoft JhengHei UI Light
华文细黑	汉仪柳楷繁	Microsoft JhengHei
华文宋体	汉仪衡方碑繁	Microsoft JhengHei Light
华文隶书	汉仪褚楷繁	Adobe 宋体 Std
华文楷体	汉仪曹隶繁	-
华文琥珀	仿宋	-

支持全部文档类型

字体名称	字体名称
汉仪中黑 KW	汉仪楷体 KW
汉仪中等线 KW	汉仪君黑 KW 55J
汉仪书宋二 KW	汉仪君黑 KW 85J
汉仪旗黑 KW 85S	汉仪仿宋 KW
汉仪旗黑 KW 55S	-

其他语言字体

仅支持 Word 文档

字体名称	字体名称	字体名称
Agency FB	Gadugi	Palatino Linotype
Algerian	Garamond	Papyrus
Arial Black	Gigi	Parchment
Arial Narrow	Gill Sans MT	Perpetua
Arial Rounded MT Bold	Gill Sans MT Condensed	Perpetua Titling MT
Bahnschrift	Gill Sans MT Ext Condensed Bold	Playbill
Baskerville Old Face	Gill Sans Ultra Bold	Poor Richard
Bauhaus 93	Gill Sans Ultra Bold Condensed	Pristina

版权所有：腾讯云计算（北京）有限责任公司

Bell MT	Gloucester MT Extra Condensed	Rage Italic
Berlin Sans FB	Goudy Old Style	Raleway
Berlin Sans FB Demi	Goudy Stout	Raleway Black
Bernard MT Condensed	Haettenschweiler	Raleway ExtraBold
Blackadder ITC	Harlow Solid Italic	Raleway ExtraLight
Bodoni MT	Harrington	Raleway Light
Bodoni MT Black	High Tower Text	Raleway Medium
Bodoni MT Condensed	HoloLens MDL2 Assets	Raleway SemiBold
Bodoni MT Poster Compressed	Impact	Raleway Thin
Book Antiqua	Imprint MT Shadow	Ravie
Bookman Old Style	Informal Roman	Roboto
Bookshelf Symbol 7	Ink Free	Roboto Black
Bradley Hand ITC	Javanese Text	Roboto Light
Britannic Bold	Jokerman	Roboto Medium
Broadway	Juice ITC	Roboto Thin
Brush Script MT	Kristen ITC	Roboto Slab
C059	Kunstler Script	Rockwell
Calibri	Leelawadee	Rockwell Condensed
Calibri Light	Leelawadee UI	Rockwell Extra Bold
Californian FB	Leelawadee UI Semilight	Script MT Bold
Calisto MT	Lucida Bright	Segoe MDL2 Assets
Cambria	Lucida Calligraphy	Segoe Print
Cambria Math	Lucida Console	Segoe Script
Candara	Lucida Fax	Segoe UI
Candara Light	Lucida Handwriting	Segoe UI Black

版权所有：腾讯云计算（北京）有限责任公司

Castellar	Lucida Sans	Segoe UI Light
Centaur	Lucida Sans Typewriter	Segoe UI Semibold
Century	Lucida Sans Unicode	Segoe UI Semilight
Century Gothic	Magneto	Segoe UI Emoji
Century Schoolbook	Maiandra GD	Segoe UI Historic
Chiller	Malgun Gothic (맑은 고딕)	Segoe UI Symbol
Colonna MT	Malgun Gothic Semilight (맑은 고딕 Semilight)	Showcard Gothic
Consolas	Matura MT Script Capitals	SimSun-ExtB
Constantia	Merriweather	Sitka Banner
Cooper Black	Merriweather Black	Sitka Display
Copperplate Gothic Bold	Merriweather Light	Sitka Heading
Copperplate Gothic Light	Microsoft Himalaya	Sitka Small
Corbel	Microsoft New Tai Lue	Sitka Subheading
Corbel Light	Microsoft PhagsPa	Sitka Text
Courier	Microsoft Sans Serif	Small Fonts
Curlz MT	Microsoft Tai Le	Snap ITC
D050000L	Microsoft Uighur	Standard Symbols PS
DejaVu Sans	Microsoft Yi Baiti	Stencil
DejaVu Sans Condensed	Mistral	Sylfaen
DejaVu Sans Light	Modern No. 20	Symbol
Dubai	Mongolian Baiti	Symbol Tiger
Dubai Light	Monotype Corsiva	Symbol Tiger Expert
Dubai Medium	MS Gothic (ＭＳ ゴシック)	System
Ebrima	MS Outlook	Tahoma
Edwardian Script ITC	MS Pgothic (ＭＳ Ｐゴシック)	Tempus Sans ITC

版权所有：腾讯云计算（北京）有限责任公司

Elephant	MS Reference Sans Serif	TencentSans Korean W3 (텐센트체 W3)
Engravers MT	MS Reference Specialty	TencentSans Korean W7 (텐센트체 W7)
Eras Bold ITC	MS Sans Serif	Terminal
Eras Demi ITC	MS Serif	Tiger
Eras Light ITC	MS UI Gothic	Tiger Expert
Eras Medium ITC	MT Extra	Trebuchet MS
Euclid	MT Extra Tiger	Tw Cen MT
Euclid Extra	MV Boli	Tw Cen MT Condensed
Euclid Fraktur	Myanmar Text	Tw Cen MT Condensed Extra Bold
Euclid Math One	Niagara Engraved	URW Bookman
Euclid Math Two	Niagara Solid	URW Gothic
Euclid Symbol	Nimbus Mono PS	Viner Hand ITC
Felix Titling	Nimbus Roman	Vivaldi
Fences	Nimbus Sans	Vladimir Script
Fixedsys	Nimbus Sans Narrow	Wide Latin
Footlight MT Light	Nirmala UI	Wingdings
Forte	Nirmala UI Semilight	Wingdings 2
Franklin Gothic Book	OCR A Extended	Wingdings 3
Franklin Gothic Demi	Old English Text MT	Yu Gothic (游ゴシック)
Franklin Gothic Demi Cond	Onyx	Yu Gothic Light (游ゴシック Light)
Franklin Gothic Heavy	Open Sans	Yu Gothic Medium (游ゴシック Medium)
Franklin Gothic Medium	Open Sans Extrabold	Yu Gothic UI

版权所有：腾讯云计算（北京）有限责任公司

Franklin Gothic Medium Cond	Open Sans Light	Yu Gothic UI Light
Freestyle Script	Open Sans Semibold	Yu Gothic UI Semibold
French Script MT	P052	Yu Gothic UI Semilight
Gabriola	Palace Script MT	Z003

支持全部文档类型

字体名称	字体名称	字体名称
Arial	Noto Sans Gujarati UI	Noto Sans Shavian
Comic Sans MS	Noto Sans Gurmukhi	Noto Sans Sinhala
Courier New	Noto Sans Gurmukhi UI	Noto Sans Sinhala UI
Georgia	Noto Sans Hanunoo	Noto Sans Sundanese
Kingsoft Confetti	Noto Sans Hebrew	Noto Sans Syloti Nagri
Kingsoft Extra	Noto Sans Imperial Aramaic (Noto Sans ImpAramaic)	Noto Sans Symbols
Kingsoft Mark	Noto Sans Inscriptional Pahlavi (Noto Sans InsPahlavi)	Noto Sans Symbols2
Kingsoft Math	Noto Sans Inscriptional Parthian (Noto Sans InsParthi)	Noto Sans Syriac Eastern
Kingsoft Sign	Noto Sans Javanese	Noto Sans Syriac Estrangela
Kingsoft Stress	Noto Sans Kaithi	Noto Sans Syriac Western
Noto Emoji	Noto Sans Kannada	Noto Sans Tagalog
Noto Kufi Arabic	Noto Sans Kannada UI	Noto Sans Tagbanwa
Noto Mono	Noto Sans Kayah Li	Noto Sans Tai Le
Noto Naskh Arabic	Noto Sans Kharoshthi	Noto Sans Tai Tham
Noto Naskh Arabic UI	Noto Sans Khmer	Noto Sans Tai Viet
Noto Nastaliq Urdu	Noto Sans Khmer UI	Noto Sans Tamil
Noto Sans	Noto Sans Lao	Noto Sans Tamil UI

版权所有：腾讯云计算（北京）有限责任公司

Noto Sans Adlam	Noto Sans Lao UI	Noto Sans Telugu
Noto Sans Adlam Unjoined	Noto Sans Lepcha	Noto Sans Telugu UI
Noto Sans Anatolian Hieroglyphs (Noto Sans AnatoHiero)	Noto Sans Limbu	Noto Sans Thaana
Noto Sans Arabic	Noto Sans Linear B	Noto Sans Thai
Noto Sans Arabic UI	Noto Sans Lisu	Noto Sans Thai UI
Noto Sans Armenian	Noto Sans Lycian	Noto Sans Tibetan
Noto Sans Avestan	Noto Sans Lydian	Noto Sans Tifinagh
Noto Sans Balinese	Noto Sans Malayalam	Noto Sans Ugaritic
Noto Sans Bamum	Noto Sans Malayalam UI	Noto Sans Vai
Noto Sans Batak	Noto Sans Mandaic	Noto Sans Yi
Noto Sans Bengali	Noto Sans Meetei Mayek	Noto Serif
Noto Sans Bengali UI	Noto Sans Mongolian	Noto Serif Armenian
Noto Sans Brahmi	Noto Sans Mono	Noto Serif Bengali
Noto Sans Buginese	Noto Sans Myanmar	Noto Serif CJK SC
Noto Sans Buhid	Noto Sans Myanmar UI	Noto Serif CJK TC
Noto Sans Canadian Aboriginal	Noto Sans New Tai Lue	Noto Serif Devanagari
Noto Sans Carian	Noto Sans Nko (Noto Sans N'Ko)	Noto Serif Display (Noto Serif Disp)
Noto Sans Chakma	Noto Sans Ogham	Noto Serif Ethiopic
Noto Sans Cham	Noto Sans Ol Chiki	Noto Serif Georgian
Noto Sans Cherokee	Noto Sans Old Italic	Noto Serif Gujarati
Noto Sans Coptic	Noto Sans Old Persian	Noto Serif Hebrew
Noto Sans Cuneiform	Noto Sans Old South Arabian (Noto Sans OldSouArab)	Noto Serif Kannada

版权所有：腾讯云计算（北京）有限责任公司

Noto Sans Cypriot	Noto Sans Old Turkic	Noto Serif Khmer
Noto Sans Deseret	Noto Sans Oriya	Noto Serif Lao
Noto Sans Devanagari	Noto Sans Oriya UI	Noto Serif Malayalam
Noto Sans Devanagari UI	Noto Sans Osage	Noto Serif Myanmar
Noto Sans Display (Noto Sans Disp)	Noto Sans Osmanya	Noto Serif Sinhala
Noto Sans Egyptian Hieroglyphs (Noto Sans EgyptHiero)	Noto Sans Phags Pa	Noto Serif Tamil
Noto Sans Ethiopic	Noto Sans Phoenician	Noto Serif Telugu
Noto Sans Georgian	Noto Sans Rejang	Noto Serif Thai
Noto Sans Glagolitic	Noto Sans Runic	Times New Roman
Noto Sans Gothic	Noto Sans Samaritan	Verdana
Noto Sans Gujarati	Noto Sans Saurashtra	Webdings

支持的演示文件动画

演示文件包含的格式有：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。以下表中的动画，除**随机效果**外，均支持**播放时机**、**速度**、**重复**和**延迟**的调节。

进入和退出动画

动画名称	属性	动画名称	属性	动画名称	属性
百叶窗	支持方向	圆形扩展	支持方向	放大	不支持
擦除	支持方向	渐变	不支持	飞旋	不支持
出现	不支持	渐变式回旋	不支持	浮动	不支持
飞入、飞出	支持方向	渐变式缩放	不支持	光速	不支持
盒状	支持方向	展开-细微型	不支持	滑翔	不支持
缓慢进入、缓慢退出	支持方向	翻转式由远及近	不支持	挥鞭式	不支持

版权所有：腾讯云计算（北京）有限责任公司

阶梯状	支持方向	回旋	不支持	挥舞	不支持
菱形	支持方向	渐入,渐出	不支持	空翻	不支持
轮子	支持辐射状	上升	不支持	螺旋飞入、螺旋飞出	不支持
劈裂	支持方向	伸展	支持方向	曲线向上	不支持
棋盘	支持方向	升起	不支持	投掷	不支持
切入	支持方向	缩放	支持缩放	玩具风车	不支持
闪烁一次	不支持	下降	不支持	线形	不支持
扇形展开	不支持	压缩	不支持	旋转	支持方向
十字形扩展	支持方向	颜色打字机	不支持	折叠	不支持
随机线条	支持方向	展开-温和型	不支持	字幕式	不支持
随机效果	不支持	中心旋转	不支持	-	-
向内溶解	不支持	弹跳	不支持	-	-

强调动画

动画名称	属性	动画名称	属性
放大/缩小	尺寸	忽明忽暗	不支持
更改填充颜色	充填色	混色	颜色
更改线条颜色	线条色	加粗闪烁	不支持
更改字号	字号	加深	不支持
更改字体	字体	添加下划线	不支持
更改字体颜色	字体颜色	着色	颜色
更改字形	字形	温和型	不支持
透明	数量	彩色延伸	颜色
陀螺旋	数量	跷跷板	不支持
细微型	不支持	闪动	颜色

版权所有：腾讯云计算（北京）有限责任公司

变淡	不支持	闪现	不支持
补色	不支持	华丽型	不支持
补色2	不支持	爆炸	颜色
不饱和	不支持	波浪型	不支持
彩色波纹	颜色	加粗展示	不支持
垂直突出显示	颜色	闪烁	不支持
对比色	不支持	样式强调	颜色

版权所有：腾讯云计算（北京）有限责任公司

更新日志

最近更新时间：2023-12-30 10:43:41

V0.2.1

下载地址：sdk-v0.2.1.zip
支持 COS V5 签名鉴权
支持 token 鉴权和续期
支持监听 PPT 中的视频播放状态

V0.1.1

下载地址：sdk-v0.1.1.zip
支持在线预览
支持 COS V4 签名鉴权

版权所有：腾讯云计算（北京）有限责任公司

文件处理
文件哈希值计算
哈希值计算同步请求

最近更新时间：2023-03-09 15:17:42

功能描述

以同步请求的方式进行文件哈希值计算，实时返回计算得到的哈希值，该接口属于 GET 请求。

费用说明

- 调用接口会产生哈希值计算费用 和 COS 读请求费用。
- 如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
- 不支持处理 COS 上的归档存储类型和深度归档存储类型的文件，如果需要处理此类型文件，请先 恢复归档文件。

限制说明

- 文件大小：支持计算小于 128 MB 大小的文件，更大文件请使用 哈希值计算异步接口。
- 请求超时时间：10秒
- 调用接口需携带签名，具体规则请参见 请求签名 文档。
- 本接口当前支持以下地域：北京、上海、广州、成都、中国香港、新加坡、硅谷。

请求

请求示例

```
GET /for-test.mp4?ci-process=filehash&type=md5 HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
```

说明：

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

参数说明如下：

节点名称（关键字）	描述	类型	是否必选
ci-process	操作类型，哈希值计算固定为：filehash	String	是
type	支持的哈希算法类型，有效值：md5、sha1、sha256	String	是
addtoheader	是否将计算得到的哈希值，自动添加至文件的自定义header，格式为：x-cos-meta-md5/sha1/sha256; 有效值： true、false，不填则默认为false。	String	否

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

```
<Response>
<FileHashCodeResult>
<MD5>0fe771ba515b0525552f59290a50c0ef</MD5>
<FileSize>1048576</FileSize>
</FileHashCodeResult>
<Input>
<Bucket>test-1234567890</Bucket>
```

版权所有：腾讯云计算（北京）有限责任公司

```
<Object>wg.txt</Object>
<Region>ap-chongqing</Region>
</Input>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
FileHashCodeResult	Response	文件哈希值的结果。	Container
Input	Response	输入文件的基本信息。	Container

Container 节点 FileHashCodeResult 的内容：

节点名称（关键字）	父节点	描述	类型
MD5	Response.FileHashCodeResult	请求时 `type` 为 md5 时返回	String
SHA1	Response.FileHashCodeResult	请求时 `type` 为 sha1 时返回	String
SHA256	Response.FileHashCodeResult	请求时 `type` 为 sha256 时返回	String
FileSize	Response.FileHashCodeResult	文件的大小。	Int

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Input	存储桶所在地域。	String
Bucket	Response.JobsDetail.Input	输入文件所在的存储桶。	String
Object	Response.JobsDetail.Input	输入文件在存储桶中的名称。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /for-test.mp4?ci-process=filehash&type=md5 HTTP/1.1
Host: test-1234567890.cos.ap-beijing.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG****-
sign-time=1484213027;32557109027&q-key-time=1484213027;32557109027&q-header-list=
host&q-url-param-list=acl&q-signature=dcc1eb2022b79cb2a780bf062d3a40e120b4****
Content-Length: 0
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==

<Response>
<FileHashCodeResult>
<MD5>0fe771ba515b0525552f59290a50c0ef</MD5>
<FileSize>1048576</FileSize>
<LastModified>2022-06-27T15:23:12+0800</LastModified>
<Etag>"ee8de918d05640145b18f70f4c3aa602"</Etag>
</FileHashCodeResult>
<Input>
<Bucket>test-1234567890</Bucket>
<Object>wg.txt</Object>
<Region>ap-chongqing</Region>
</Input>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

提交哈希值计算任务

最近更新时间：2023-12-30 10:42:53

功能描述

以提交任务的方式进行文件哈希值计算，异步返回计算得到的哈希值，该接口属于 POST 请求。

费用说明

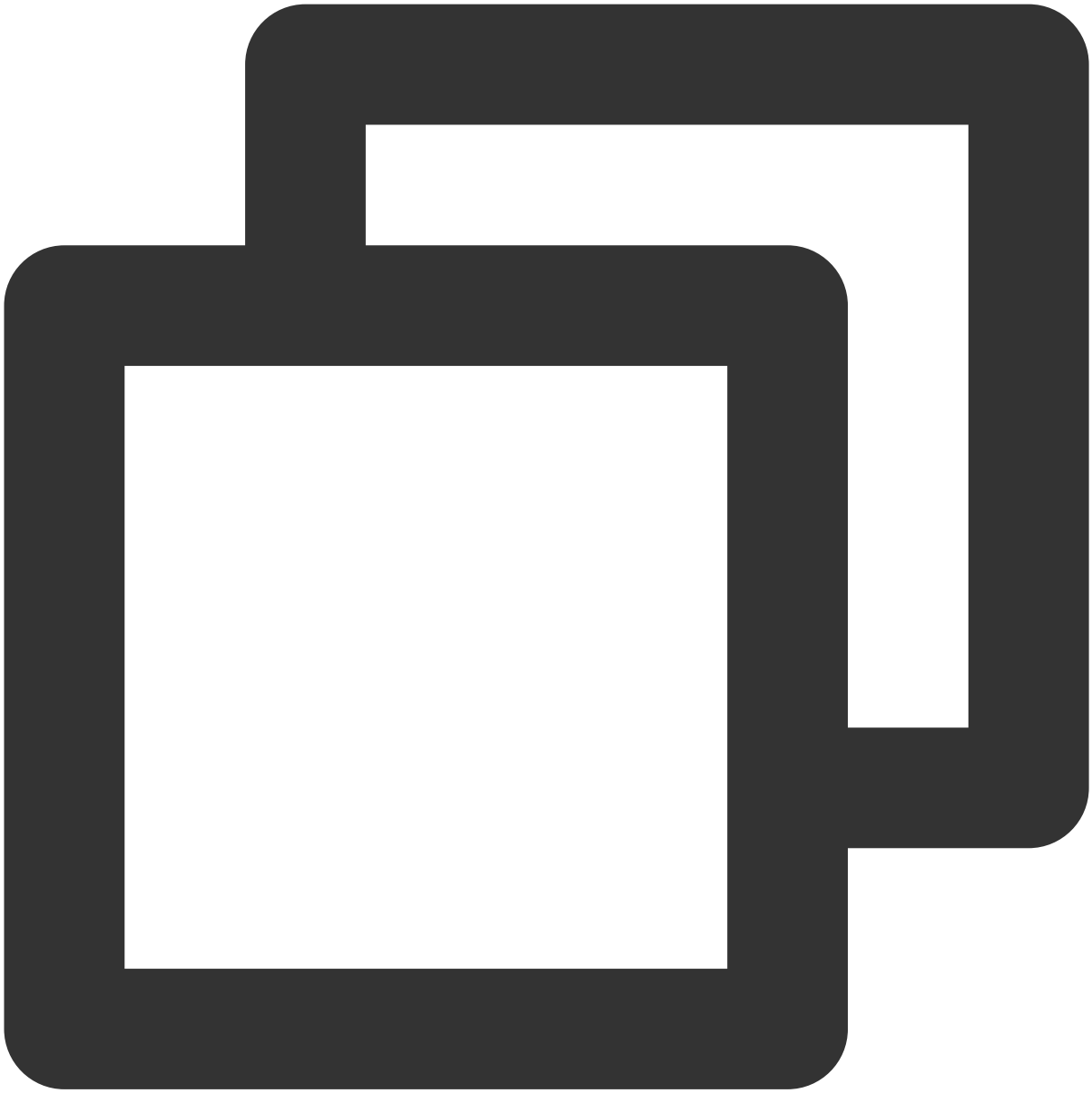
调用接口会产生 哈希值计算费用 和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持计算 COS 上的归档存储类型和深度归档存储类型的文件哈希值，如果需要计算此类型文件的哈希值，请先 恢复归档文件。

限制说明

文件大小支持：支持计算小于 50 GB 大小的文件。
调用接口需携带签名，具体规则请参见 请求签名 文档。
本接口当前支持以下地域：北京、上海、广州、成都、中国香港、新加坡、硅谷。

请求

请求示例



```
POST /file_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

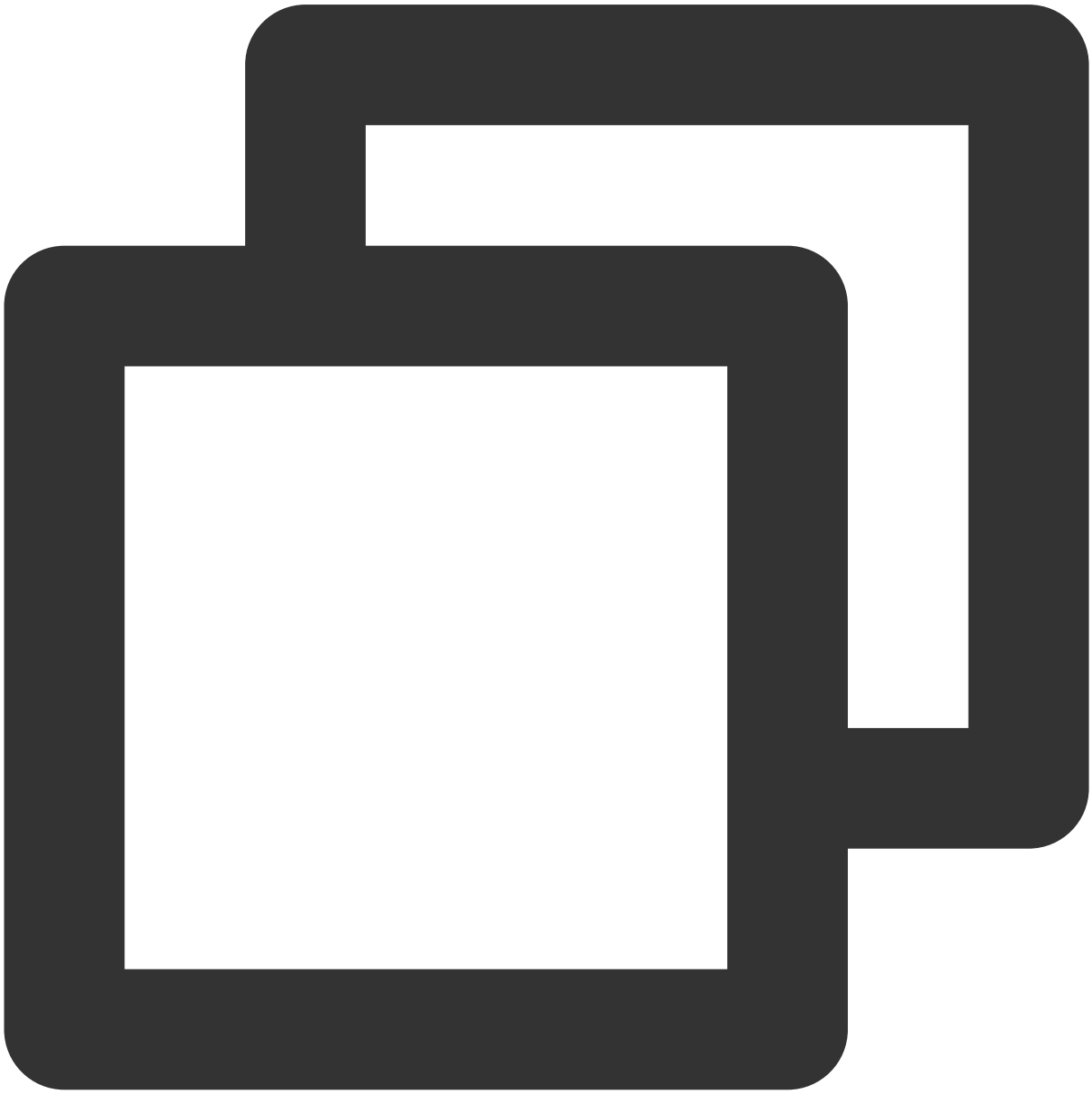
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>FileHashCode</Tag>
    <Input>
        <Object>input/test.mp4</Object>
    </Input>
    <Operation>
        <FileHashCodeConfig>
            <Type>MD5</Type>
            <AddToHeader>true</AddToHeader>
        </FileHashCodeConfig>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://test.test.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	表示任务的类型，哈希值计算默认为：FileHashCode。	String	是
Input	Request	包含待操作的文件信息。	Container	是
Operation	Request	包含哈希值计算的处理规则。	Container	是
QueueId	Request	任务所在的队列 ID。	String	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式。	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型。	String	否
CallBack	Request	任务回调的地址，优先级高于队列的回调地址。	String	否
CallBackMqConfig	Request	任务回调 TDMQ 配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	文件名，取值为文件在当前存储桶中的完整名称。	String	是

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
FileHashCodeConfig	Request.Operation	指定哈希值计算的处理规则。	Container	是
UserData	Request.Operation	透传用户信息, 可打印的 ASCII 码, 长度不超过1024	String	否

Container 类型 FileHashCodeConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.Operation.FileHashCodeConfig	哈希值的算法类型，有效值：MD5、SHA1、SHA256。	String	是
AddToHeader	Request.Operation.FileHashCodeConfig	是否将计算得到的哈希值添加至文件自定义header, 有效值：true、false，默认值为false。自定义header根据 `Type` 的值变化，例如 `Type` 值为MD5时，自定义header为 x-cos-meta-md5。	String	否

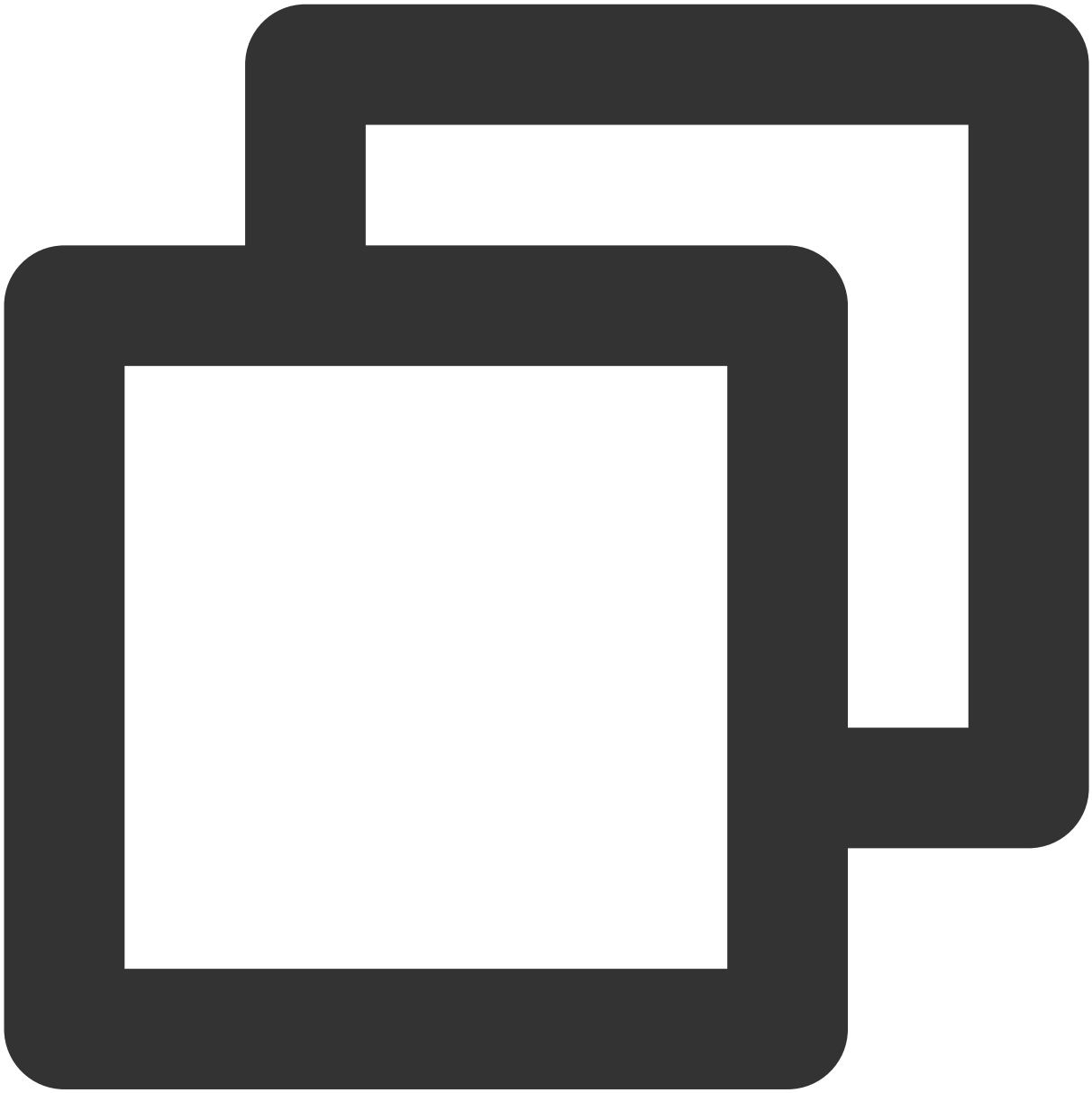
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileHashCode</Tag>
```

```
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileHashCodeConfig>
                <Type>MD5</Type>
                <AddToHeader>true</AddToHeader>
            </FileHashCodeConfig>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义。	String
JobId	Response.JobsDetail	新创建任务的 ID。	String
Tag	Response.JobsDetail	表示任务的类型，哈希值计算默认为：FileHashCode。	String
State	Response.JobsDetail	任务的状态，为 Submitted(已提交)、Running(进行中)、Success(成功)、Failed(失败)、Pause(已暂停)、Cancel(已取消) 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间。	String
StartTime	Response.JobsDetail	任务的开始时间。	String

版权所有：腾讯云计算（北京）有限责任公司

EndTime	Response.JobsDetail	任务的结束时间。	String
QueueId	Response.JobsDetail	任务所属的队列 ID。	String
Input	Response.JobsDetail	被计算哈希值的文件信息。	Container
Operation	Response.JobsDetail	哈希值计算的处理规则。	Container

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Input	存储桶所在地域。	String
Bucket	Response.JobsDetail.Input	文件所在的存储桶。	String
Object	Response.JobsDetail.Input	被计算哈希值的文件名。	String

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
UserData	Response.JobsDetail.Operation	透传用户信息。	String
FileHashCodeConfig	Response.JobsDetail.Operation	同请求中的 Request.Operation.FileHashCodeConfig。	Contai
FileHashCodeResult	Response.JobsDetail.Operation	计算得到的文件 hash 值信息，任务未完成时不返回。	Contai

Container 节点 FileHashCodeResult 的内容：

节点名称（关键字）	父节点	描述	类型
MD5	Response.JobsDetail.Operation.FileHashCodeResult	MD5 计算结果。	String
SHA1	Response.JobsDetail.Operation.FileHashCodeResult	SHA1 计算结果。	String
SHA256	Response.JobsDetail.Operation.FileHashCodeResult	SHA256 计算结果。	String
FileSize	Response.JobsDetail.Operation.FileHashCodeResult	文件大小。	Int
LastModified	Response.JobsDetail.Operation.FileHashCodeResult	文件的最后修改时间。	String

版权所有：腾讯云计算（北京）有限责任公司

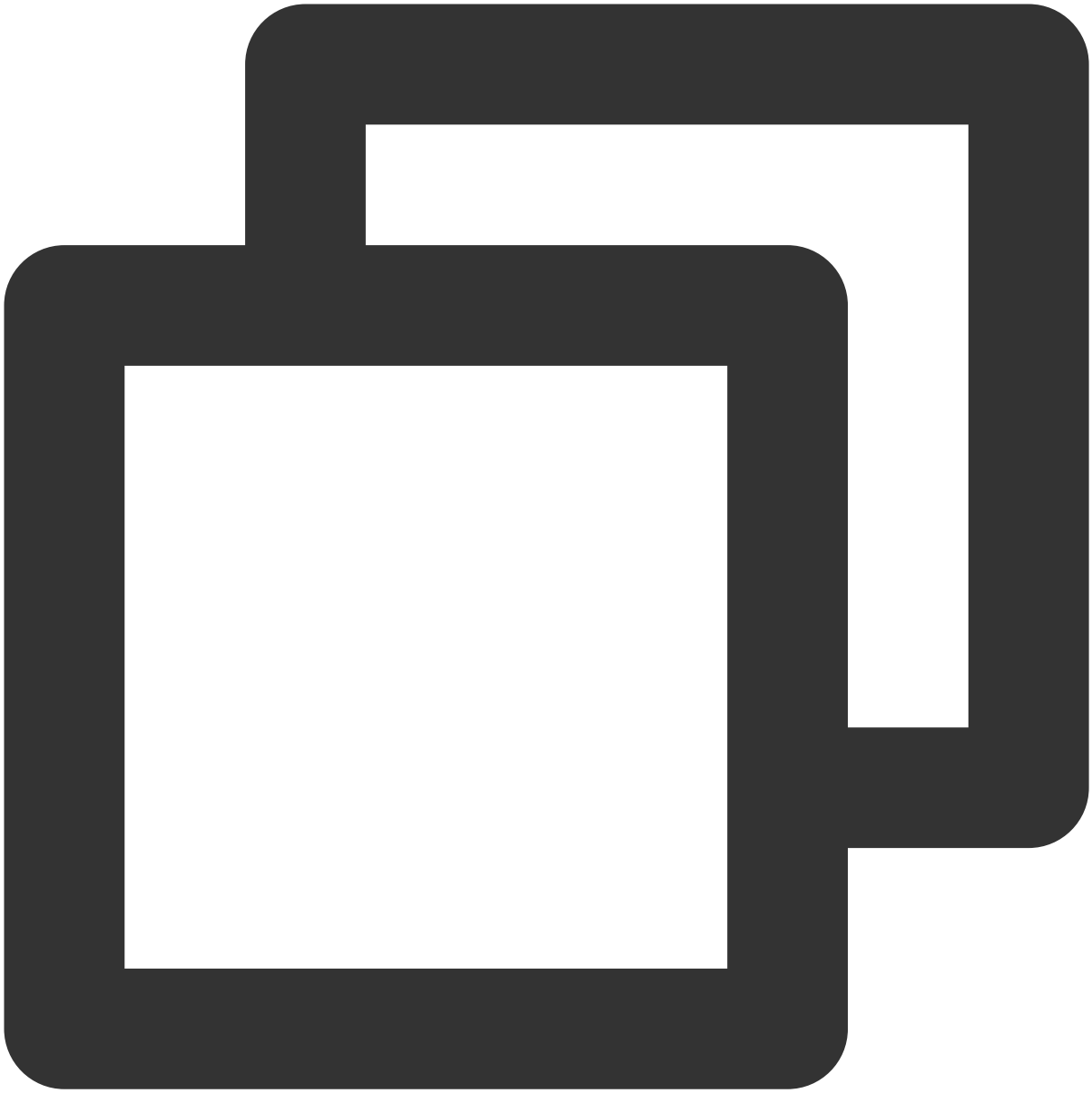
Etag	Response.JobsDetail.Operation.FileHashCodeResult	文件的Etag。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

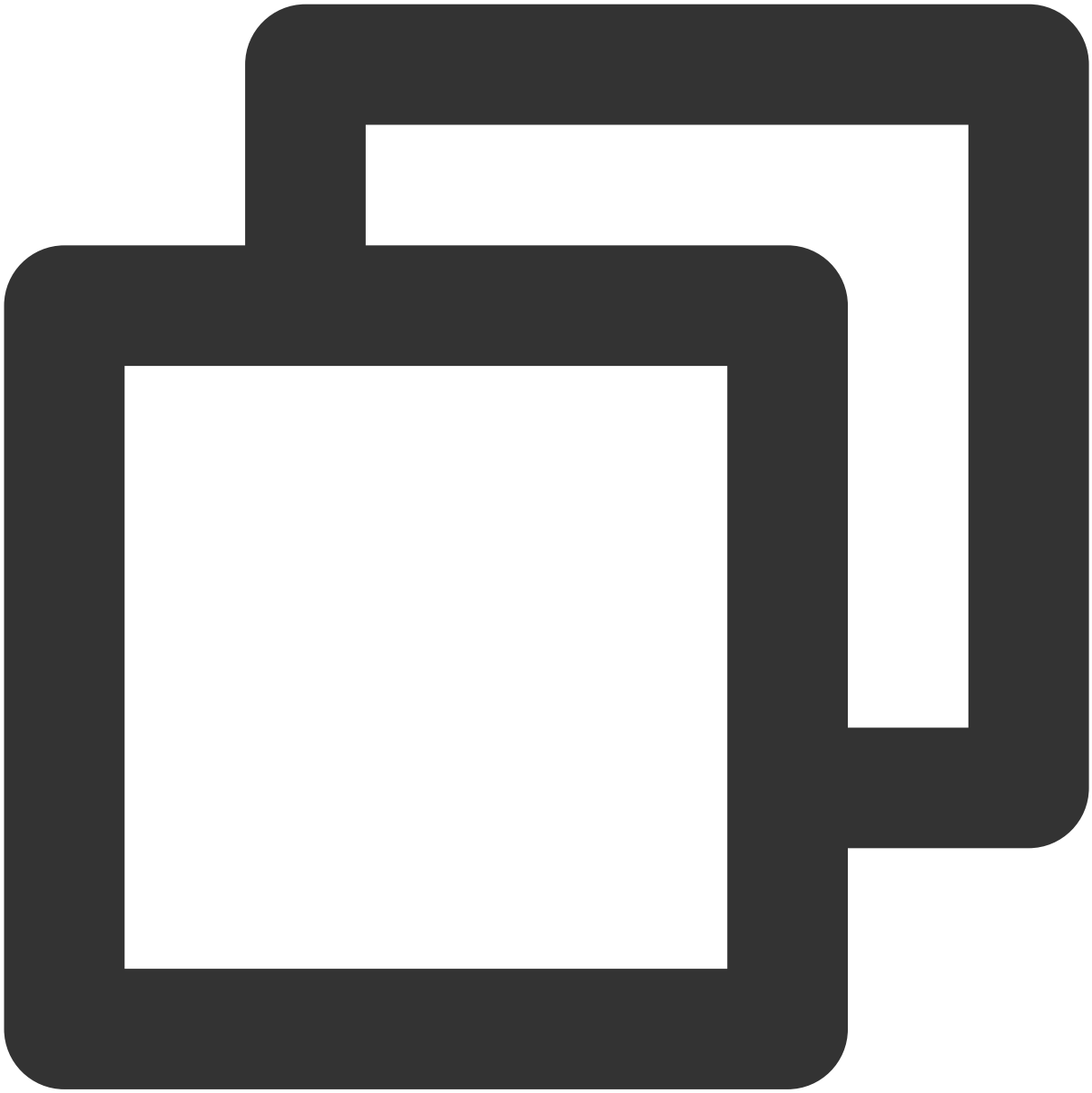
请求



```
POST /file_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>FileHashCode</Tag>
    <Input>
        <Object>input/test.mp4</Object>
    </Input>
```

```
    <Operation>
        <FileHashCodeConfig>
            <Type>MD5</Type>
            <AddToHeader>true</AddToHeader>
        </FileHashCodeConfig>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileHashCode</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileHashCodeConfig>
                <Type>MD5</Type>
                <AddToHeader>true</AddToHeader>
            </FileHashCodeConfig>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

查询哈希值计算结果

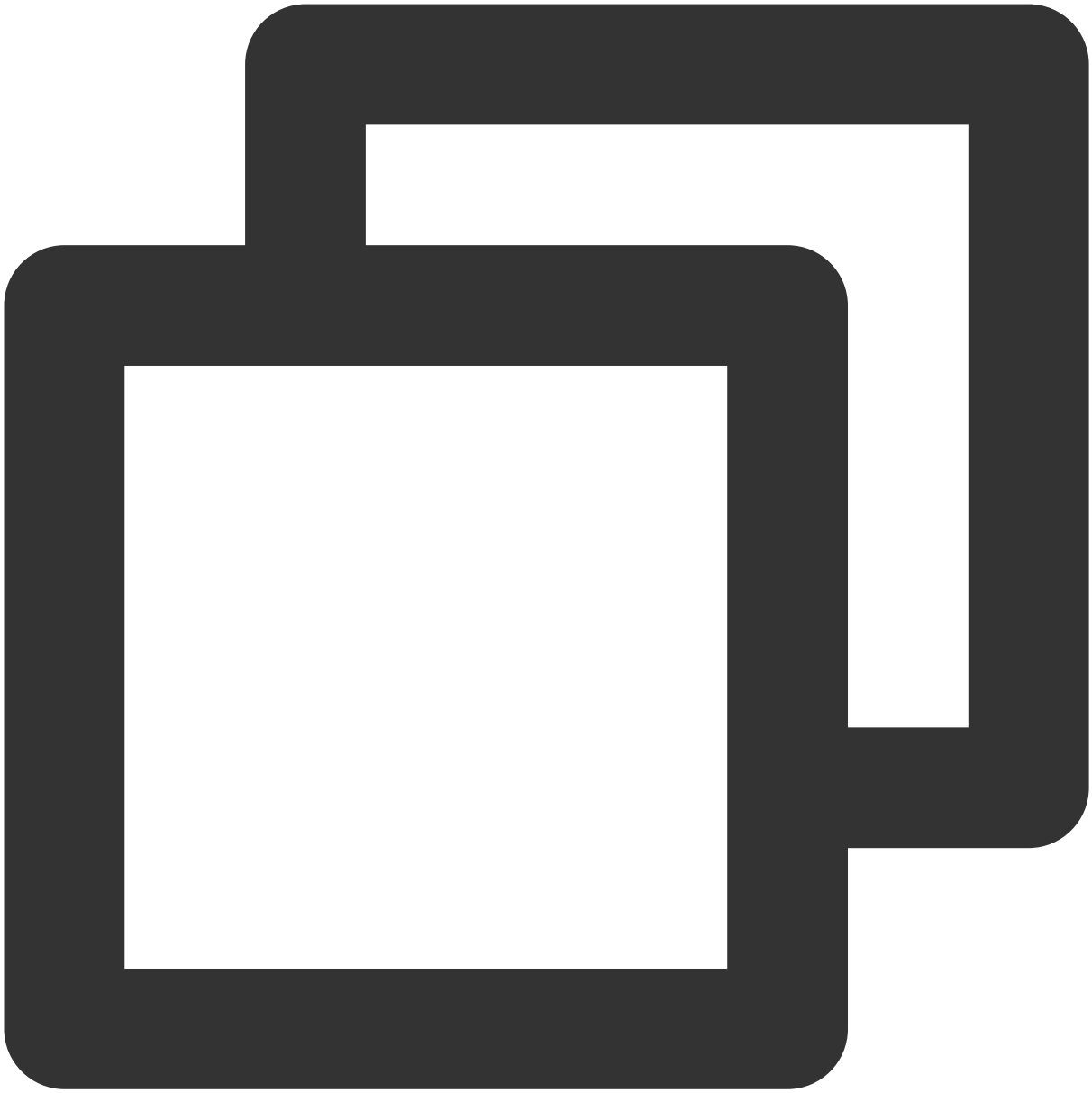
最近更新时间：2023-12-30 10:42:44

功能描述

本接口用于主动查询指定的文件哈希值计算任务结果。

请求

请求示例



```
GET /file_jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

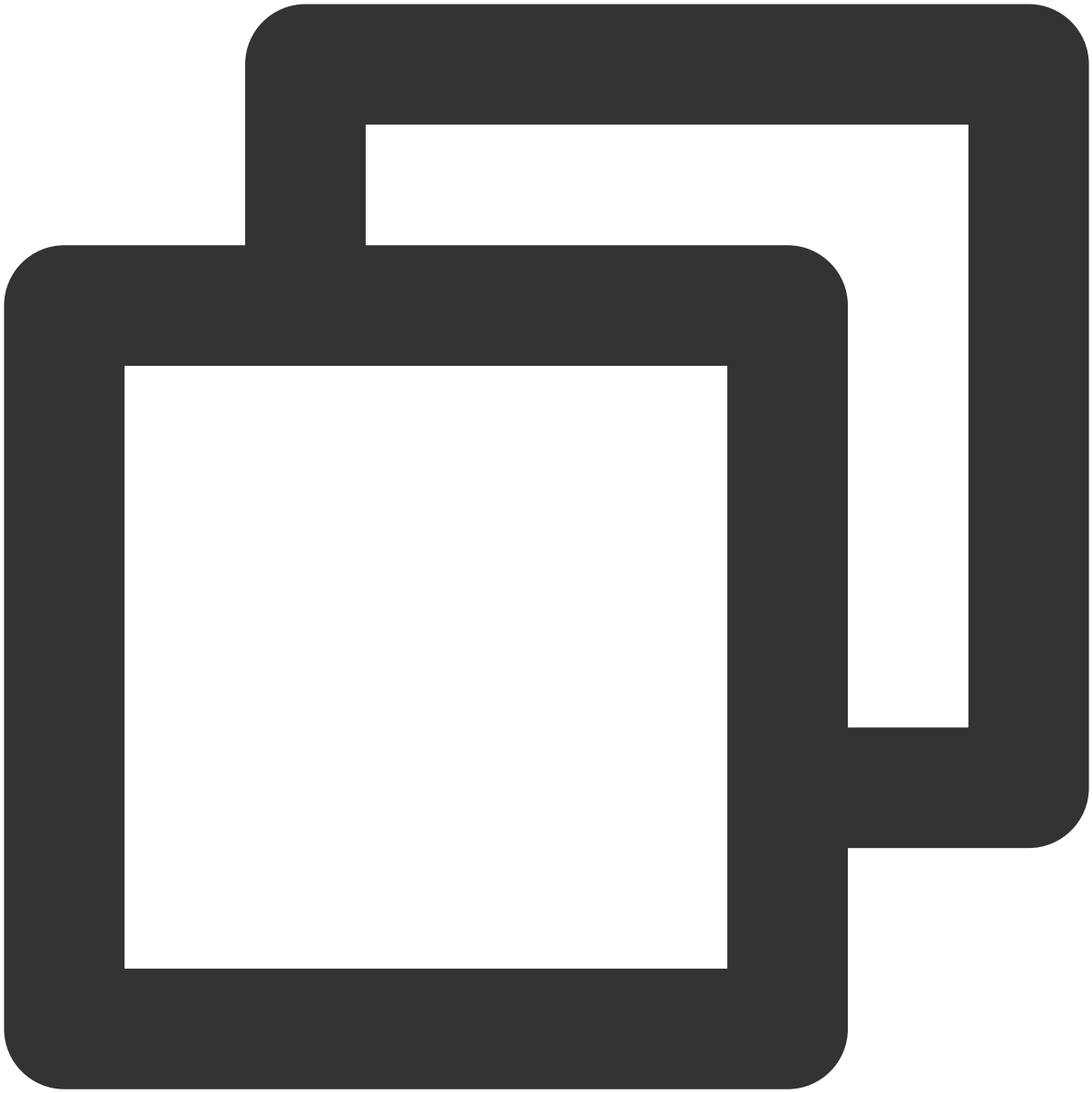
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileHashCode</Tag>
```

```
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileHashCodeConfig>
                <Type>MD5</Type>
                <AddToHeader>true</AddToHeader>
            </FileHashCodeConfig>
            <FileHashCodeResult>
                <MD5>xxxxxxxx</MD5>
                <FileSize>123</FileSize>
                <LastModified>2022-06-27T15:23:11+0800</LastModified>
                <Etag>xxxxxx</Etag>
            </FileHashCodeResult>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息。	Container
NonExistJobIds	Response	查询的 ID 中不存在任务，所有任务都存在时不返回。	String

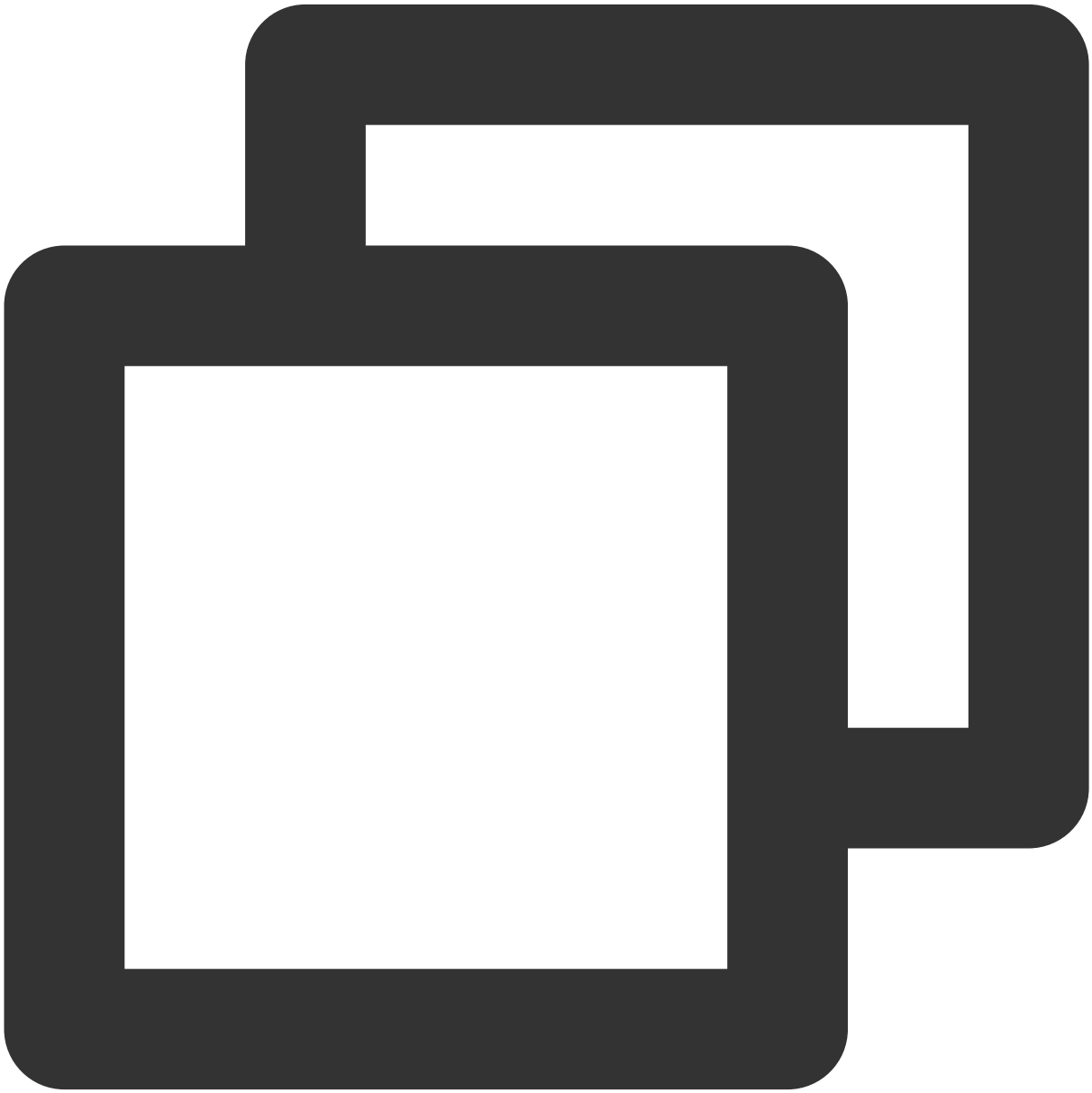
具体参数值，请前往 文件哈希值计算 `JobsDetail` 查看。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

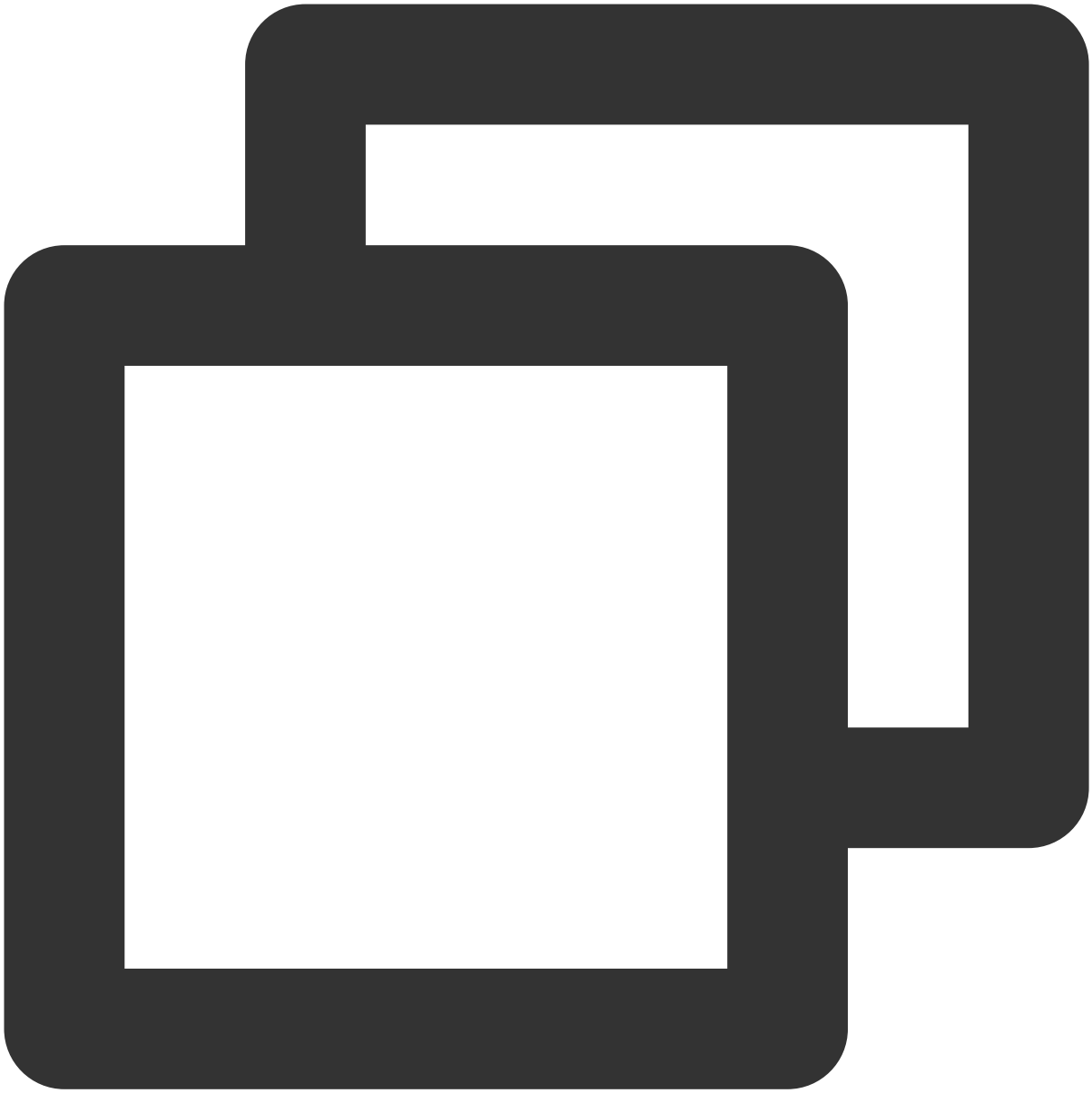
实际案例

请求



```
GET /file_jobs/f93984788066911ed89ed352d4d9d2084 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileHashCode</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileHashCodeConfig>
                <Type>MD5</Type>
                <AddToHeader>true</AddToHeader>
            </FileHashCodeConfig>
            <FileHashCodeResult>
                <MD5>xxxxxxxx</MD5>
                <FileSize>123</FileSize>
                <LastModified>2022-06-27T15:23:11+0800</LastModified>
                <Etag>xxxxxx</Etag>
            </FileHashCodeResult>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

文件解压
提交文件解压任务

最近更新时间：2023-12-30 10:42:04

功能描述

以提交任务的方式进行文件解压缩，异步返回压缩包内被解压出来的文件，该接口属于 POST 请求。

费用说明

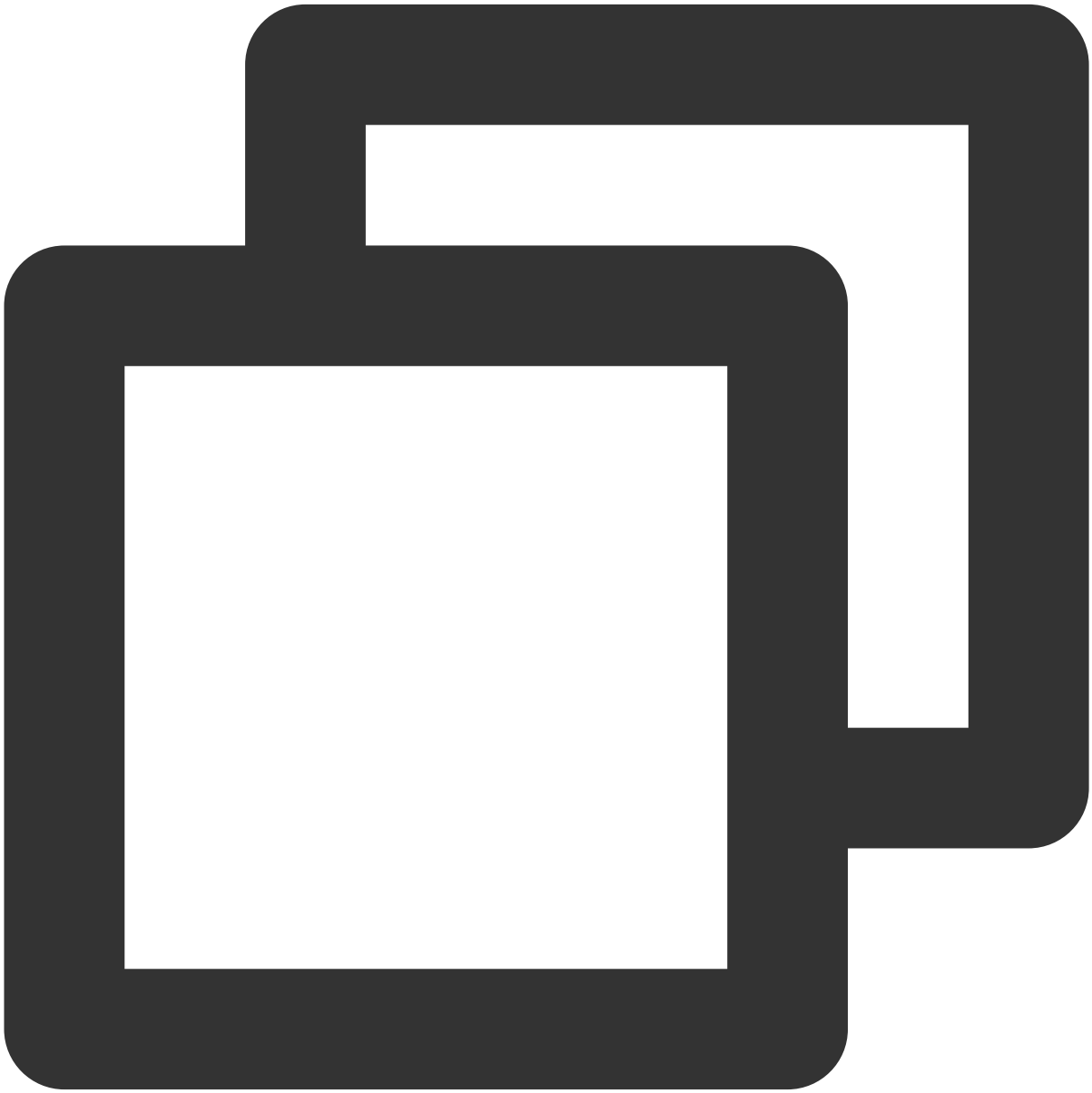
调用接口会产生文件解压费用和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持解压 COS 上的归档存储类型和深度归档存储类型的文件，如果解压需要此类型文件，请先 恢复归档文件。

限制说明

文件大小支持：支持解压小于 5 TB 的压缩包文件。
文件解压的处理方式为流式解压，会边解压边输出解压文件，如果解压过程中有部分文件损坏导致解压操作中止，会保留已解压的部分文件。
调用接口需携带签名，具体规则请参见 请求签名 文档。
本接口当前支持以下地域：北京、上海、广州、成都、中国香港、新加坡、硅谷。

请求

请求示例



```
POST /file_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

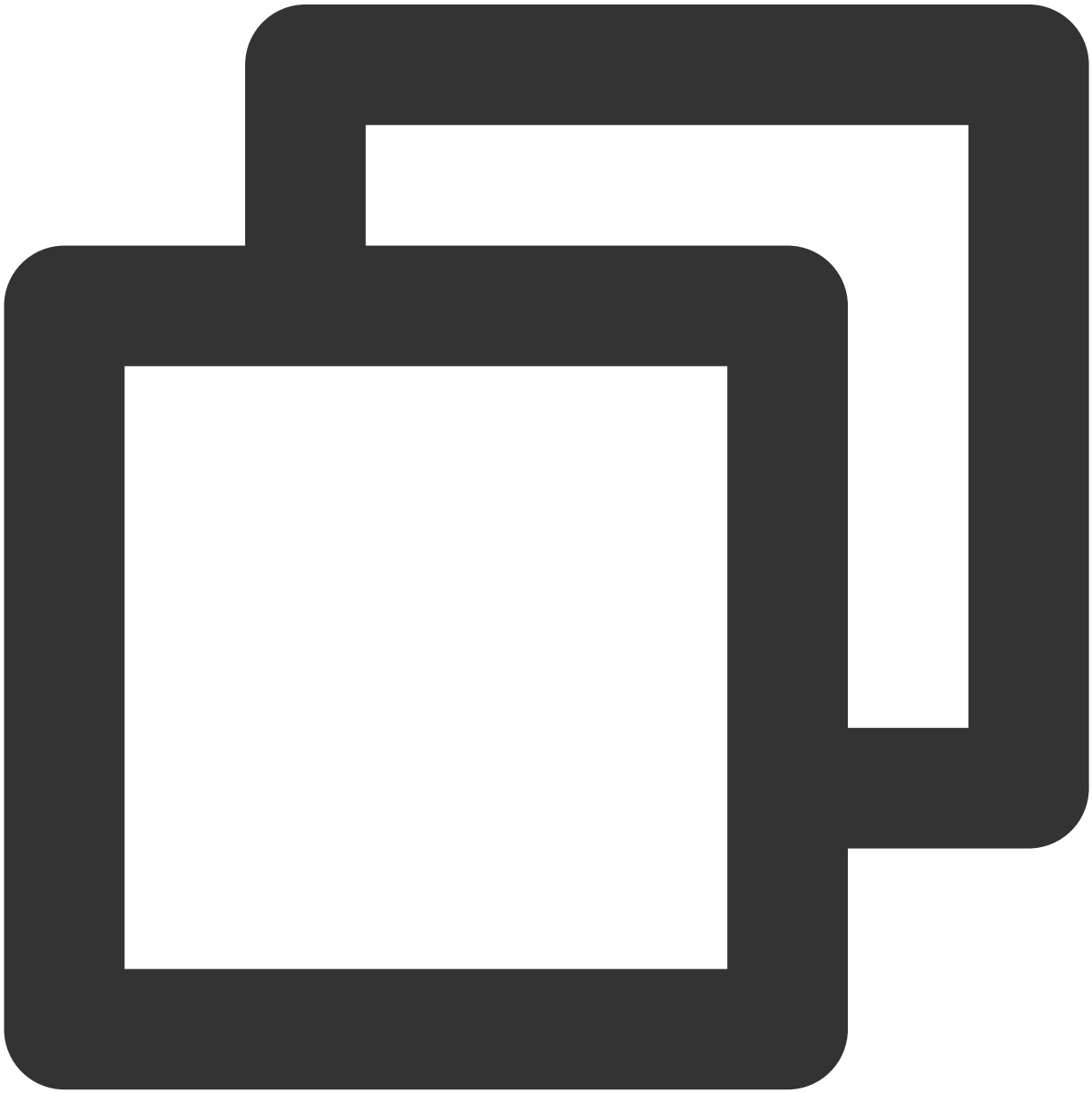
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>FileUncompress</Tag>
    <Input>
        <Object>input/test.zip</Object>
    </Input>
    <Operation>
        <FileUncompressConfig>
            <Prefix>output/</Prefix>
            <PrefixReplaced>1</PrefixReplaced>
        </FileUncompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	表示任务的类型，文件解压默认为：FileUncompress。	String	是
Input	Request	包含待操作的文件信息。	Container	是
Operation	Request	包含文件解压的处理规则。	Container	是
QueueId	Request	任务所在的队列 ID。	String	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式。	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型。	String	否
CallBack	Request	任务回调的地址，优先级高于队列的回调地	String	否

版权所有：腾讯云计算（北京）有限责任公司

		址。		
CallBackMqConfig	Request	任务回调 TDMQ 配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig	Container	否

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	文件名，取值为文件在当前存储桶中的完整名称。	String	是

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
FileUncompressConfig	Request.Operation	指定文件解压的处理规则。	Container	是
UserData	Request.Operation	透传用户信息, 可打印的 ASCII 码, 长度不超过1024	String	否
Output	Request.Operation	指定解压后的文件保存的存储桶信息。	Container	是

Container 类型 FileUncompressConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Prefix	Request.Operation.FileUncompressConfig	指定解压后输出文件的前缀，不填则默认保存在存储桶根路径。	String	否
PrefixReplaced	Request.Operation.FileUncompressConfig	指定解压后的文件路径是否需要替换前缀，有效值： - 0：不添加额外的前缀，解压缩将保存在Prefix指定的路径下（不会保留压缩包的名称，仅将压缩包内的文件保存至指定的路径）。	String	否

版权所有：腾讯云计算（北京）有限责任公司

		- 1：以压缩包本身的名称作为前缀，解压缩将保存在Prefix指定的路径下。 - 2：以压缩包完整路径作为前缀，此时如果不指定Prefix，就是解压到压缩包所在的当前路径（包含压缩包本身名称）。 - 默认值为0。		

示例：

假设压缩包名为 test.zip，压缩包内的文件为“图片.jpg”，存储在了存储桶 A 的 123 目录下，则压缩包的完整路径为：123/test.zip。

对该压缩包进行解压缩，指定解压缩到存储桶A，并指定 `Prefix` 为 456，则不同的 `PrefixReplaced` 值对应解压后的文件保存情况如下：

0：“图片.jpg”会保存在 456 目录下，“图片.jpg”的完整路径为 456/图片.jpg；

1：“图片.jpg”会以 test 为前缀保存在 456 目录下，“图片.jpg”的完整路径为 456/test/图片.jpg；

2：“图片.jpg”会以 123/test 为前缀保存在 456 目录下，“图片.jpg”的完整路径为 456/123/test/图片.jpg。

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Region	Request.Operation.Output	存储桶的地域。	String	是
Bucket	Request.Operation.Output	保存解压后文件的存储桶。	String	是

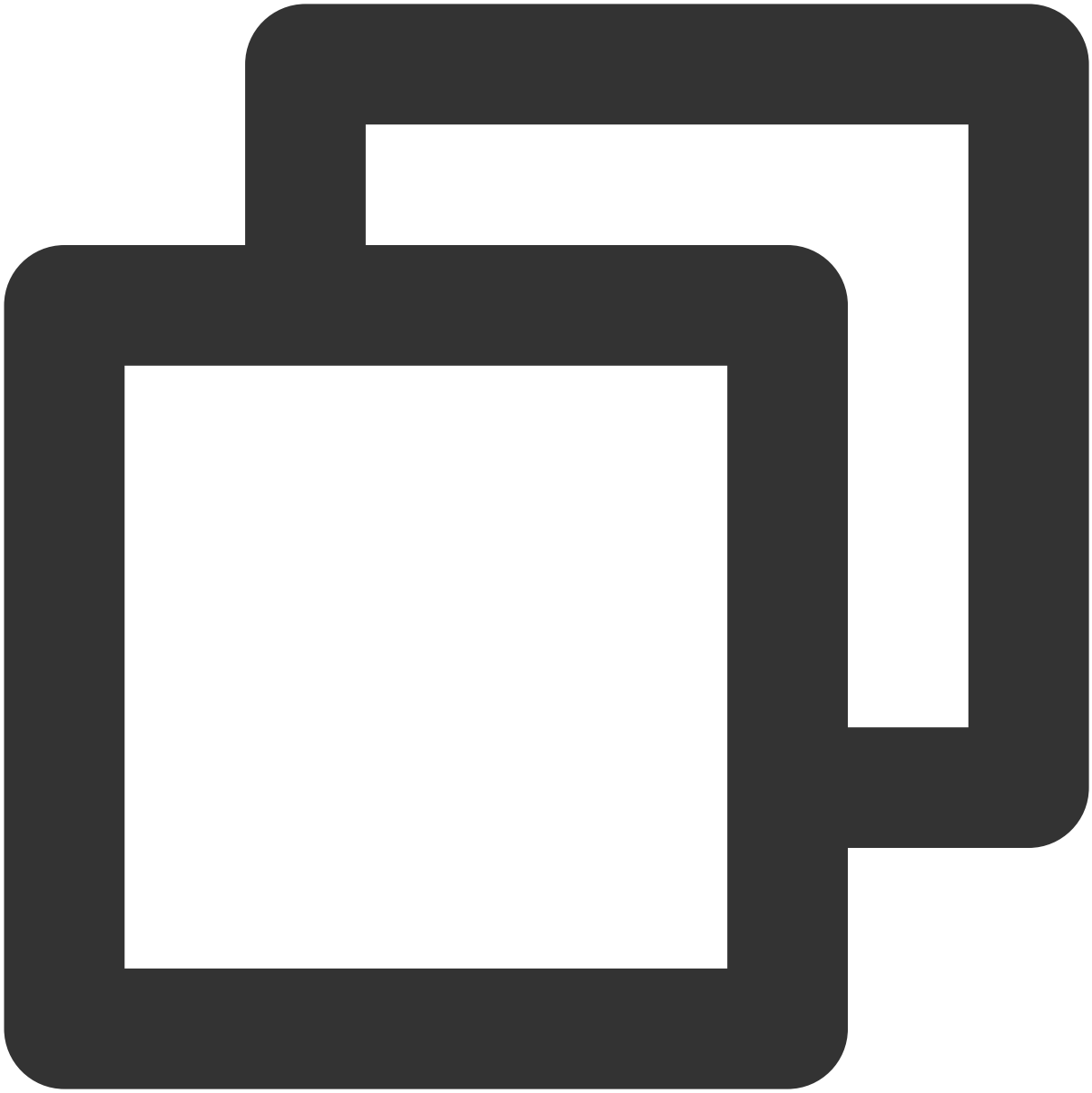
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileUncompress</Tag>
```

```
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.zip</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileUncompressConfig>
                <Prefix>output/</Prefix>
                <PrefixReplaced>1</PrefixReplaced>
            </FileUncompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义。	String
JobId	Response.JobsDetail	新创建任务的 ID。	String
Tag	Response.JobsDetail	表示任务的类型，文件解压默认为：FileUncompress。	String
State	Response.JobsDetail	任务的状态，为 Submitted(已提交)、Running(进行中)、Success(成功)、Failed(失败)、Pause(已暂	String

版权所有：腾讯云计算（北京）有限责任公司

		停)、Cancel(已取消) 其中一个	
CreationTime	Response.JobsDetail	任务的创建时间。	String
StartTime	Response.JobsDetail	任务的开始时间。	String
EndTime	Response.JobsDetail	任务的结束时间。	String
QueueId	Response.JobsDetail	任务所属的队列 ID。	String
Input	Response.JobsDetail	被解压文件的信息。	Container
Operation	Response.JobsDetail	文件解压的处理规则。	Container

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Input	存储桶所在地域。	String
Bucket	Response.JobsDetail.Input	文件所在的存储桶。	String
Object	Response.JobsDetail.Input	被解压文件的文件名。	String

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类
UserData	Response.JobsDetail.Operation	透传用户信息。	St
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Co
FileUncompressConfig	Response.JobsDetail.Operation	同请求中的 Request.Operation.FileUncompressConfig。	Co
FileUncompressResult	Response.JobsDetail.Operation	文件解压的结果，任务未完成时不返回。	Co

Container 节点 FileUncompressResult 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Operation.FileUncompressResult	解压后文件保存的存储桶的地域	String
Bucket	Response.JobsDetail.Operation.FileUncompressResult	解压后文件保存的存储桶	String

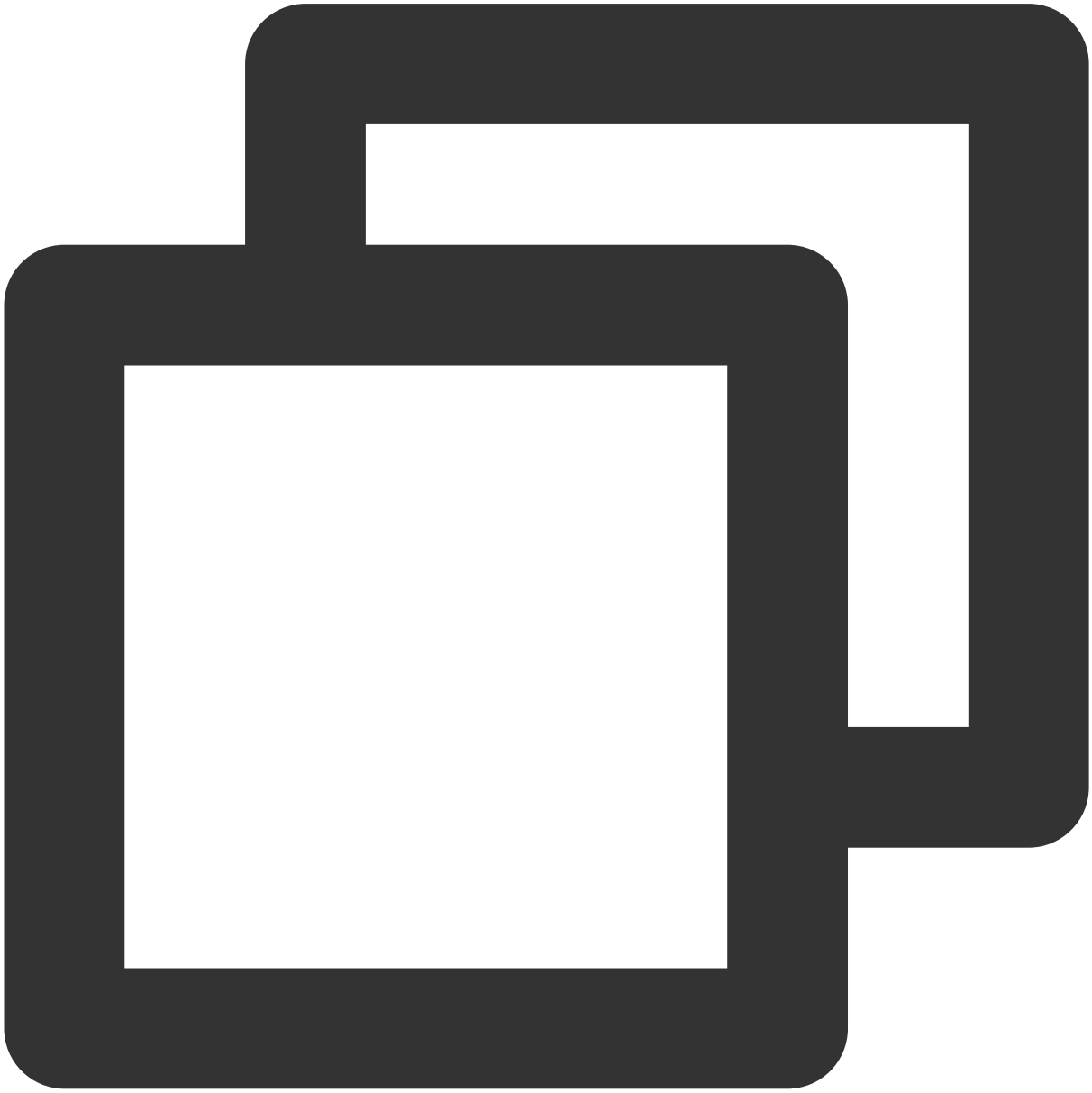
FileCount	Response.JobsDetail.Operation.FileUncompressResult	解压后文件的个数	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

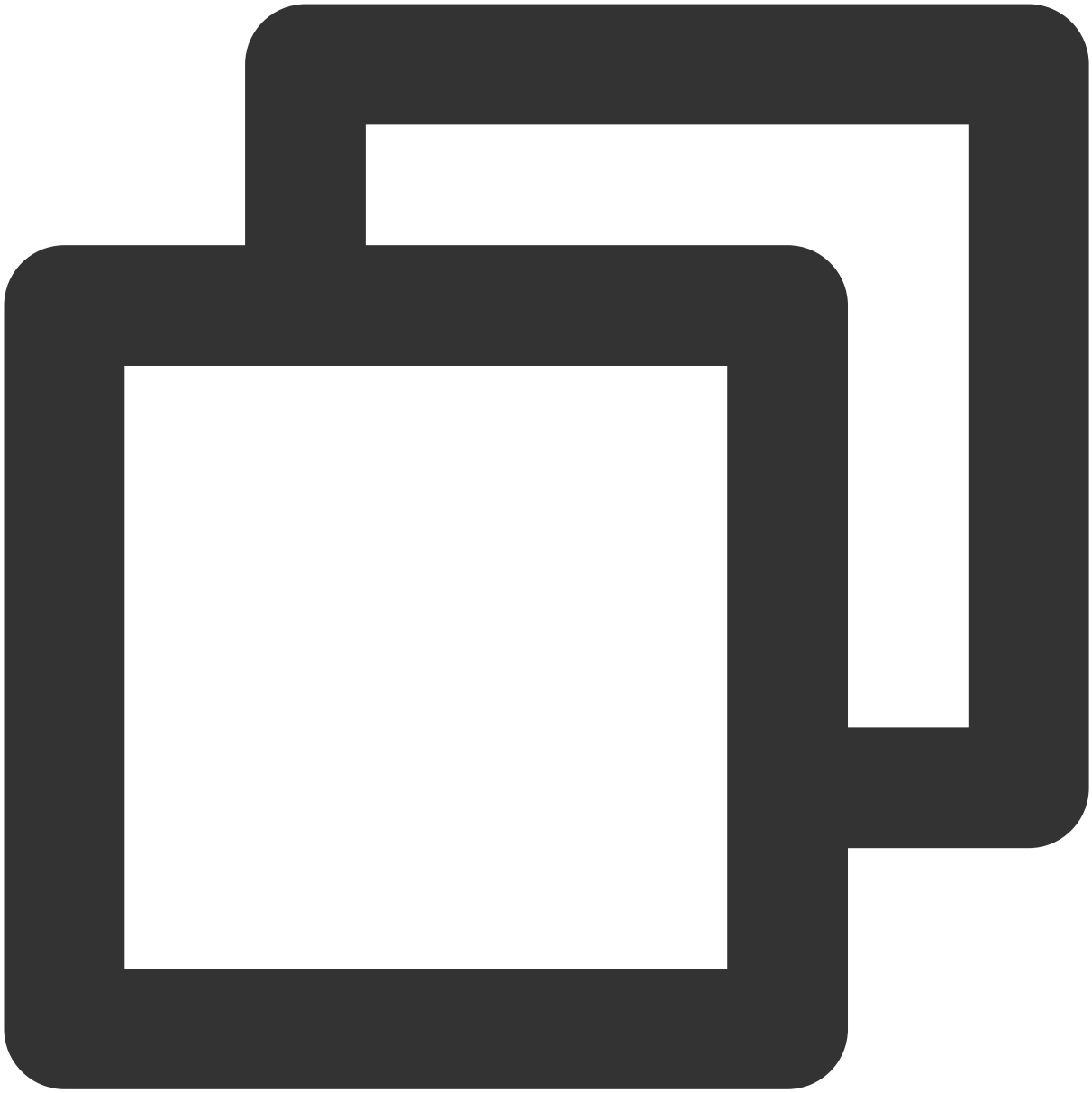
实际案例

请求



```
POST /file_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>FileUncompress</Tag>
    <Input>
        <Object>input/test.zip</Object>
    </Input>
    <Operation>
        <FileUncompressConfig>
            <Prefix>output/</Prefix>
            <PrefixReplaced>1</PrefixReplaced>
        </FileUncompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileUncompress</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.zip</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileUncompressConfig>
                <Prefix>output/</Prefix>
                <PrefixReplaced>1</PrefixReplaced>
            </FileUncompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

查询文件解压结果

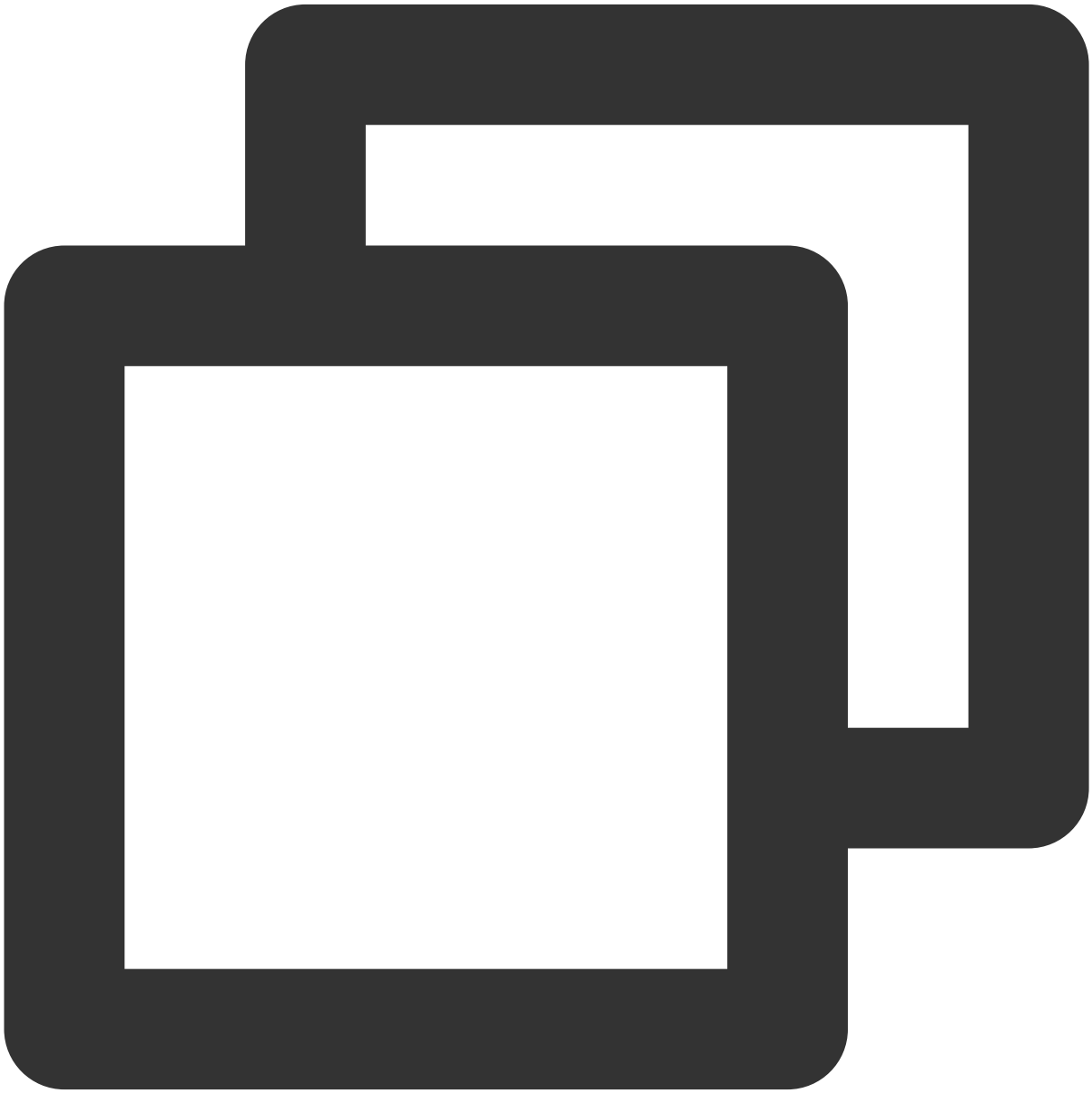
最近更新时间：2023-12-30 10:41:33

功能描述

本接口用于主动查询指定的文件解压任务结果。

请求

请求示例



```
GET /file_jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

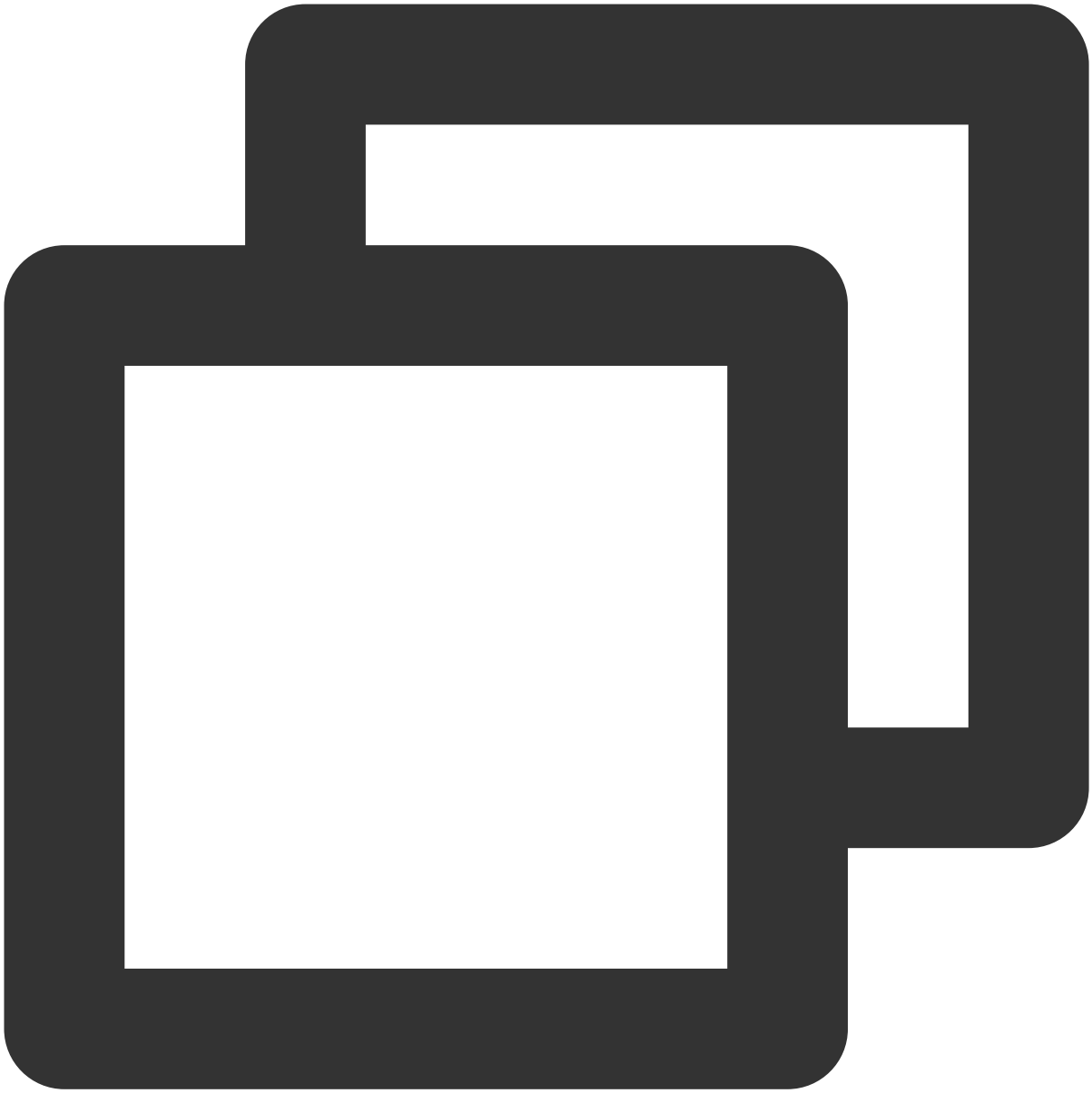
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileUncompress</Tag>
```

```
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.zip</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileUncompressConfig>
                <Prefix>output/</Prefix>
                <PrefixReplaced>1</PrefixReplaced>
            </FileUncompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息。	Container
NonExistJobIds	Response	查询的 ID 中不存在任务，所有任务都存在时不返回。	String

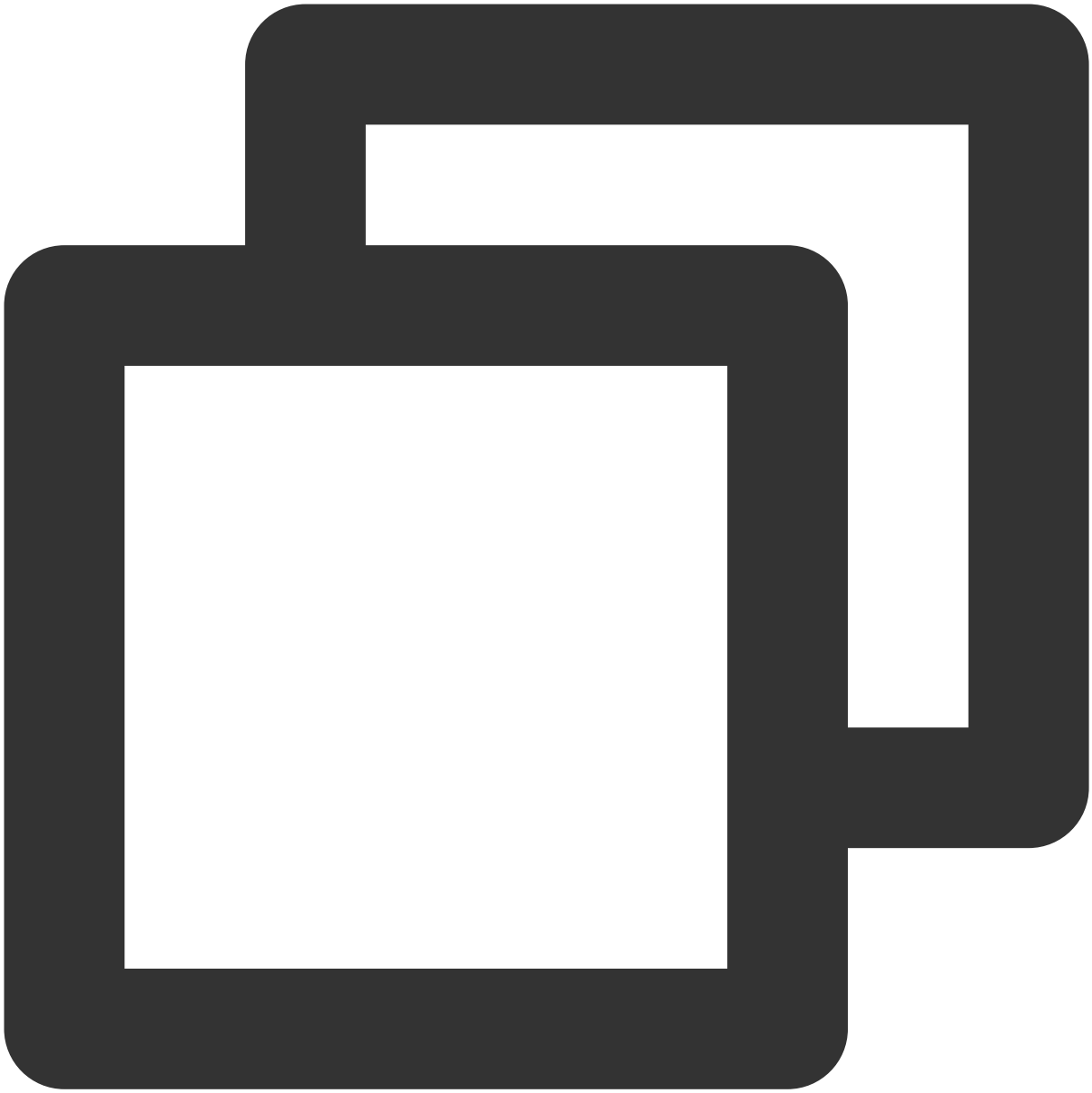
具体参数值，请前往 文件解压 `JobsDetail` 查看。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

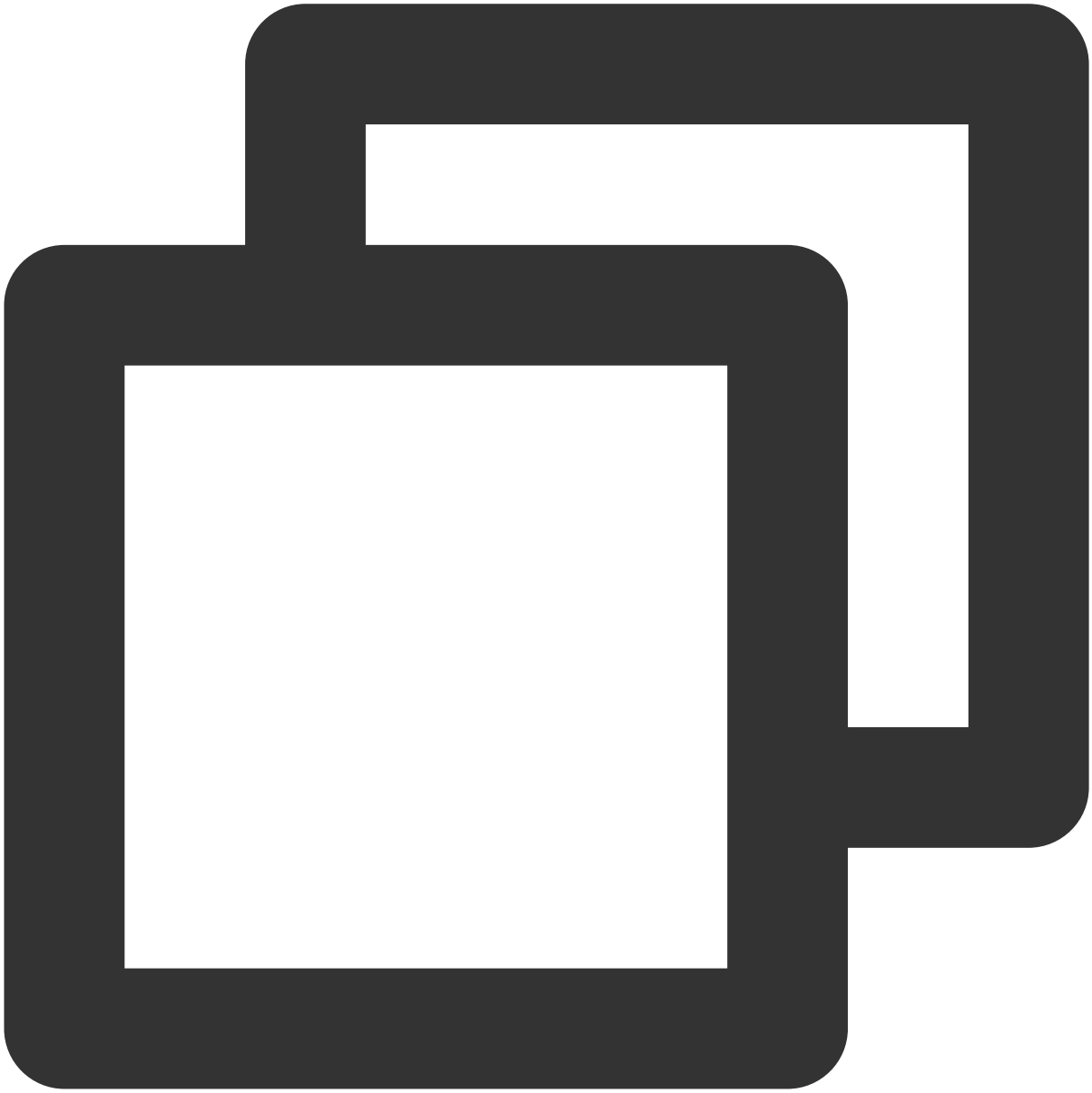
实际案例

请求



```
GET /file_jobs/f93984788066911ed89ed352d4d9d2084 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileUncompress</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.zip</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileUncompressConfig>
                <Prefix>output/</Prefix>
                <PrefixReplaced>1</PrefixReplaced>
            </FileUncompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

多文件打包压缩
提交多文件打包压缩任务

最近更新时间：2023-12-30 10:41:13

功能描述

多文件打包压缩功能可以将您的多个文件，打包为 zip 等压缩包格式，以提交任务的方式进行多文件打包压缩，异步返回打包后的文件，该接口属于 POST 请求。

费用说明

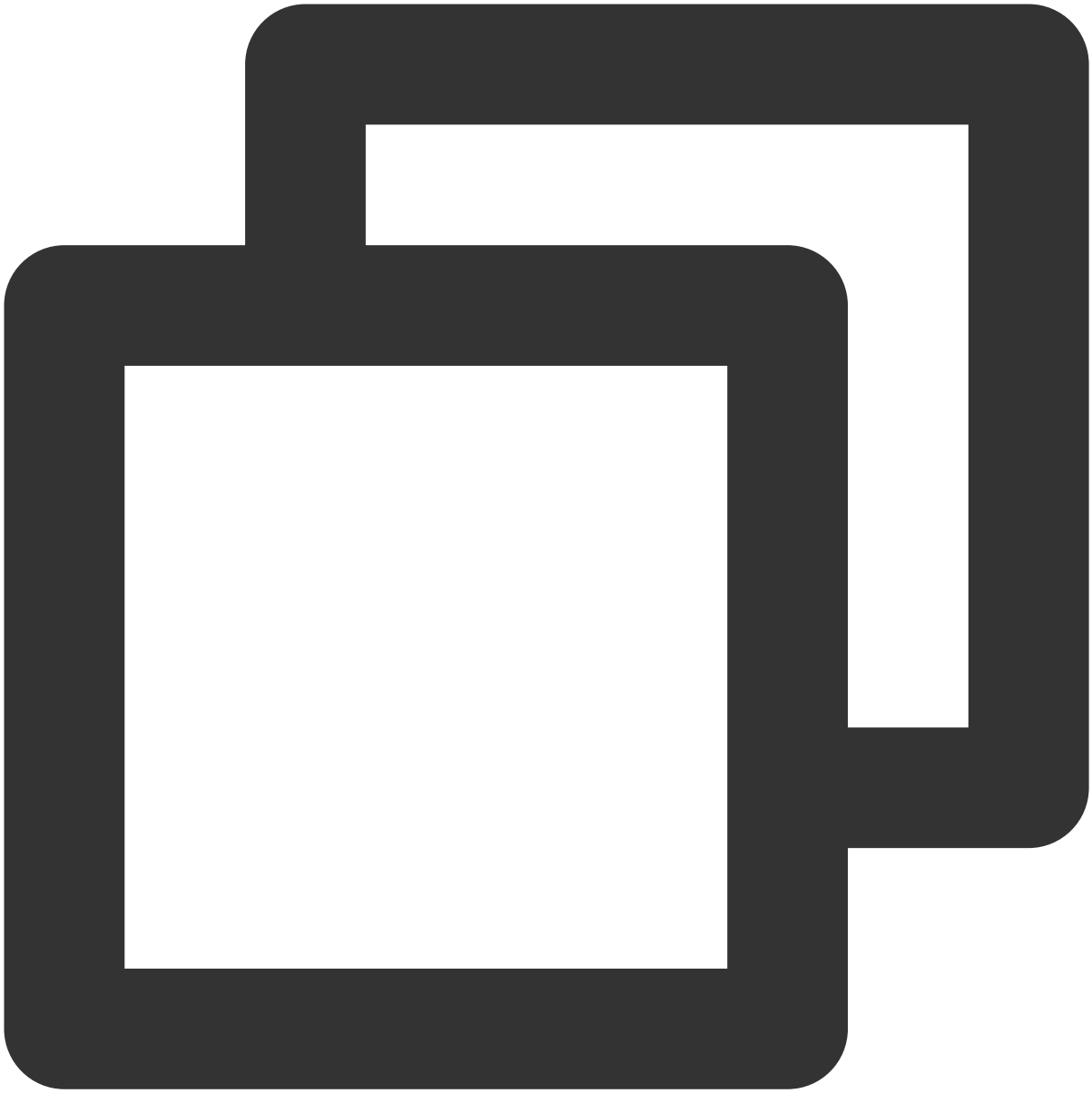
调用接口会产生多文件打包压缩费用和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持打包 COS 上的归档存储类型和深度归档存储类型的文件，如果打包需要此类型文件，请先 恢复归档文件。

限制说明

文件数量支持：最多支持打包 1 万个文件；
文件大小支持：打包压缩后总大小小于 50GB；
调用接口需携带签名，具体规则请参见 请求签名 文档。

请求

请求示例



```
POST /file_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

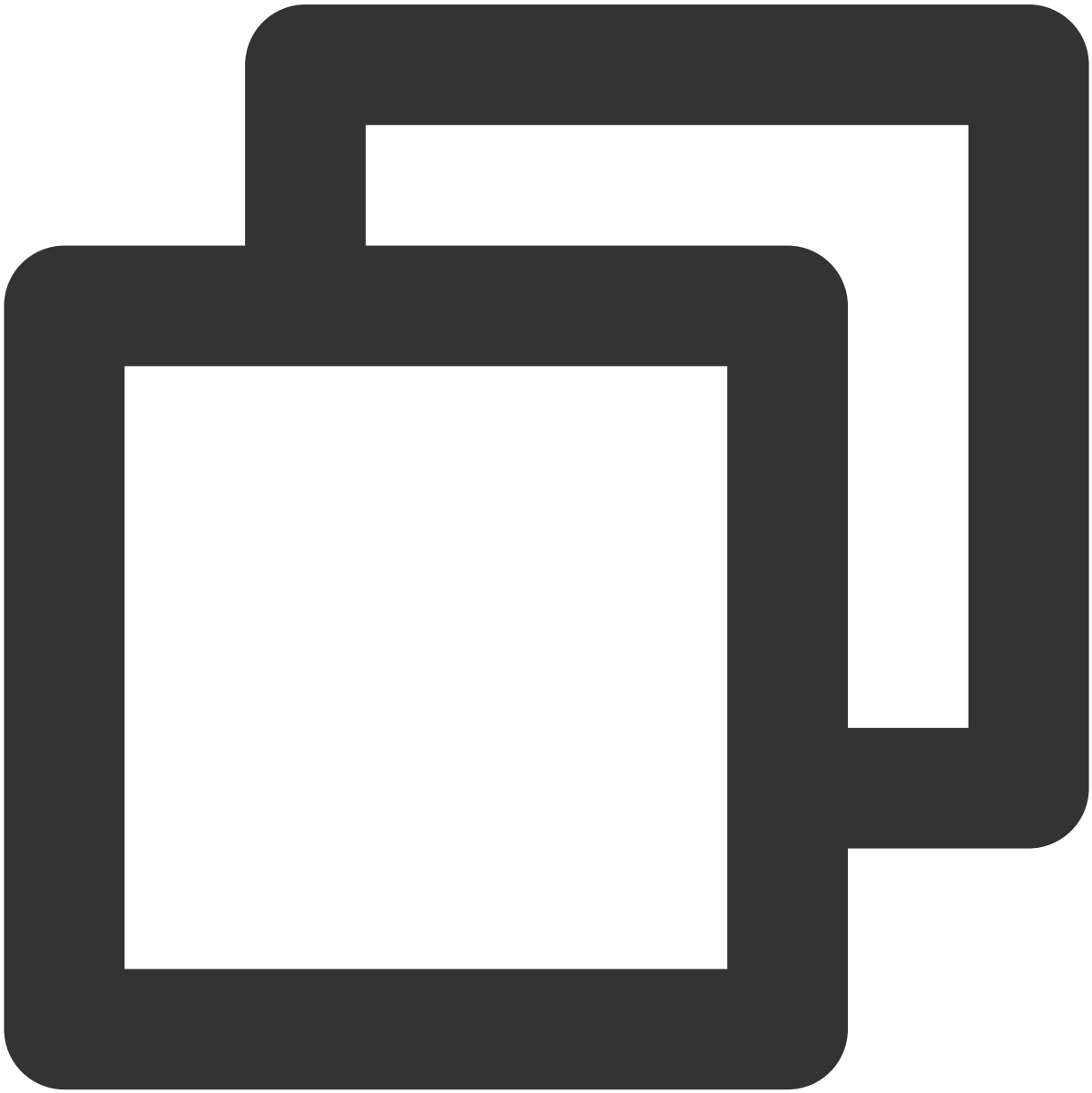
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>FileCompress</Tag>
    <Operation>
        <FileCompressConfig>
            <Prefix>test/</Prefix>
            <Format>zip</Format>
            <Flatten>0</Flatten>
        </FileCompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/test.zip</Object>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	表示任务的类型，文件解压默认为：FileCompress。	String	是
Operation	Request	包含文件打包压缩的处理规则。	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式。	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型。	String	否
CallBack	Request	任务回调的地址，优先级高于队列的回调地址。	String	否
CallBackMqConfig	Request	任务回调 TDMQ 配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
FileCompressConfig	Request.Operation	指定文件打包压缩的处理规则。	Container	是
UserData	Request.Operation	透传用户信息, 可打印的 ASCII 码, 长度不超过1024	String	否
Output	Request.Operation	指定打包压缩后的文件保存的地址信息。	Container	是

Container 类型 FileCompressConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Flatten	Request.Operation.FileCompressConfig	文件打包时，是否需要去除源文件已有的目录结构，有效值： - 0：不需要去除目录结构，打包后压缩包中的文件会保留原有的目录结构； - 1：需要，打包后压缩包内的文件会去除原有的目录结构，所有文件都在同一层级。 例如：源文件 URL 为 `https://domain/source/test.mp4`，则源文件路径为 `source/test.mp4`，如果为 1，则 ZIP 包中该文件路径为 `test.mp4`；如果为0， ZIP 包中该文件路径为 `source/test.mp4`。	String
Format	Request.Operation.FileCompressConfig	打包压缩的类型，有效值：zip、tar、tar.gz。	String
UrlList	Request.Operation.FileCompressConfig	支持将需要打包的文件整理成索引文件，后台将根据索引文件内提供的文件url，打包为一个压缩包文件。索引文件需要保存在当前存储桶中，本字段需要提供索引文件的对象地址，例如：/test/index.csv。 索引文件格式：仅支持 CVS文件，一行一条URL（仅支持本存储桶文件），如有多列字段，默认取第一列作为URL。最多不超过	String

		10000个文件, 总大小不超过50G, 否则会导致任务失败。	
Prefix	Request.Operation.FileCompressConfig	支持对存储桶中的某个前缀进行打包，如果需要对某个目录进行打包，需要加/，例如test目录打包，则值为：test/。最多不超过10000个文件，总大小不超过50G，否则会导致任务失败。	String
Key	Request.Operation.FileCompressConfig	支持对存储桶中的多个文件进行打包，个数不能超过 1000, 总大小不超过50G，否则会导致任务失败。	String 数组

注意

UrlList、Prefix、Key三者仅能选择一个，不能都为空，也不会同时生效。如果填了多个，会按优先级UrlList > Prefix > Key取最高优先级执行。

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Region	Request.Operation.Output	存储桶的地域。	String	是
Bucket	Request.Operation.Output	保存压缩后文件的存储桶。	String	是
Object	Request.Operation.Output	压缩后文件的文件名	String	是

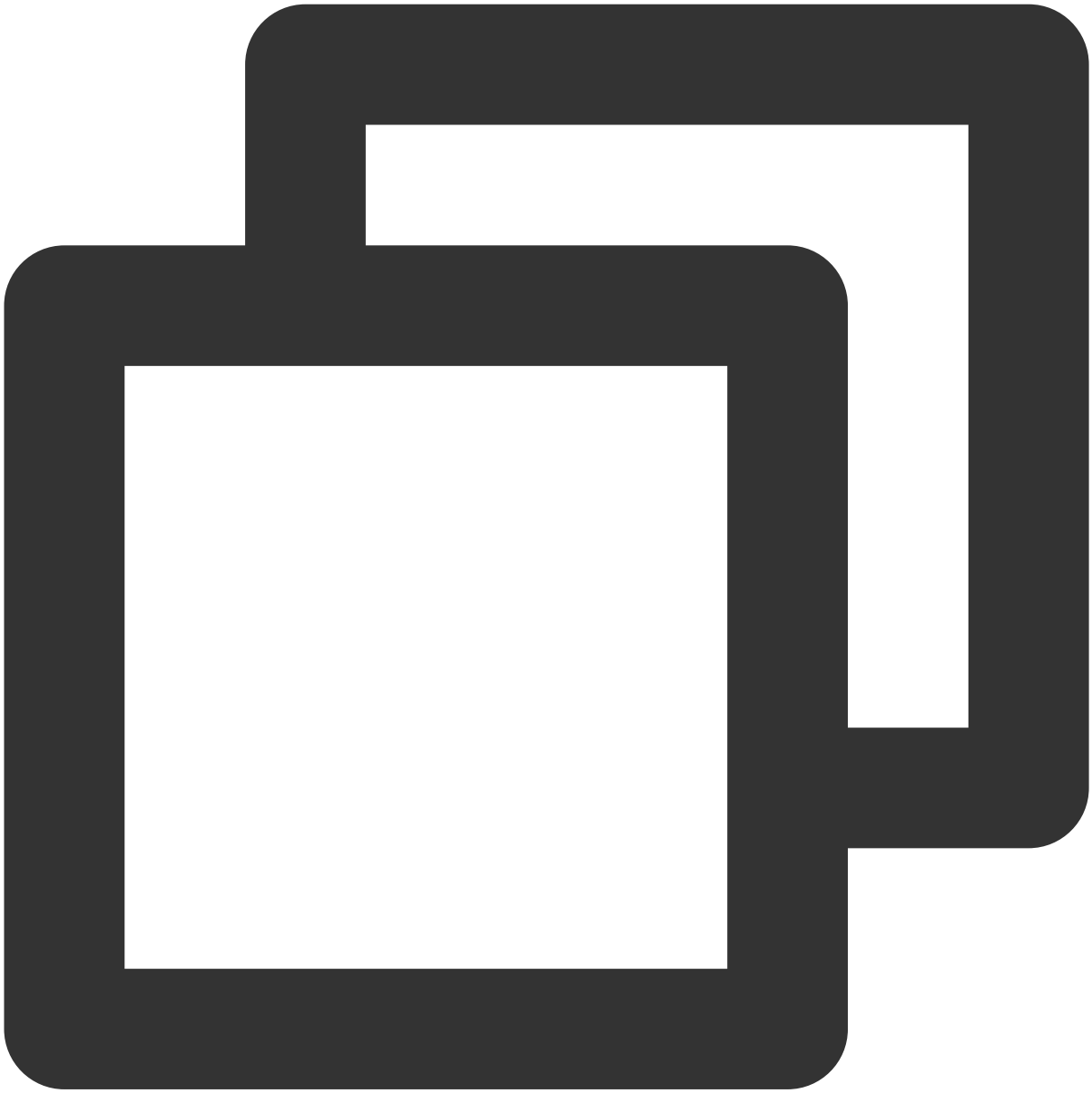
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileCompress</Tag>
```

```
        <Operation>
            <FileCompressConfig>
                <Prefix>test/</Prefix>
                <Format>zip</Format>
                <Flatten>0</Flatten>
            </FileCompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
                <Object>output/test.zip</Object>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义。	String
JobId	Response.JobsDetail	新创建任务的 ID。	String
Tag	Response.JobsDetail	表示任务的类型，多文件打包压缩默认为：FileCompress。	String
State	Response.JobsDetail	任务的状态，为 Submitted(已提交)、Running(进行中)、Success(成功)、Failed(失败)、Pause(已暂停)、Cancel(已取消) 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间。	String

StartTime	Response.JobsDetail	任务的开始时间。	String
EndTime	Response.JobsDetail	任务的结束时间。	String
QueueId	Response.JobsDetail	任务所属的队列 ID。	String
Operation	Response.JobsDetail	多文件打包压缩的处理规则。	Container

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
UserData	Response.JobsDetail.Operation	透传用户信息。	String
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Contair
FileCompressConfig	Response.JobsDetail.Operation	同请求中的 Request.Operation.FileCompressConfig。	Contair
FileCompressResult	Response.JobsDetail.Operation	多文件打包压缩的结果，任务未完成时不返回。	Contair

Container 节点 FileCompressResult 的内容：

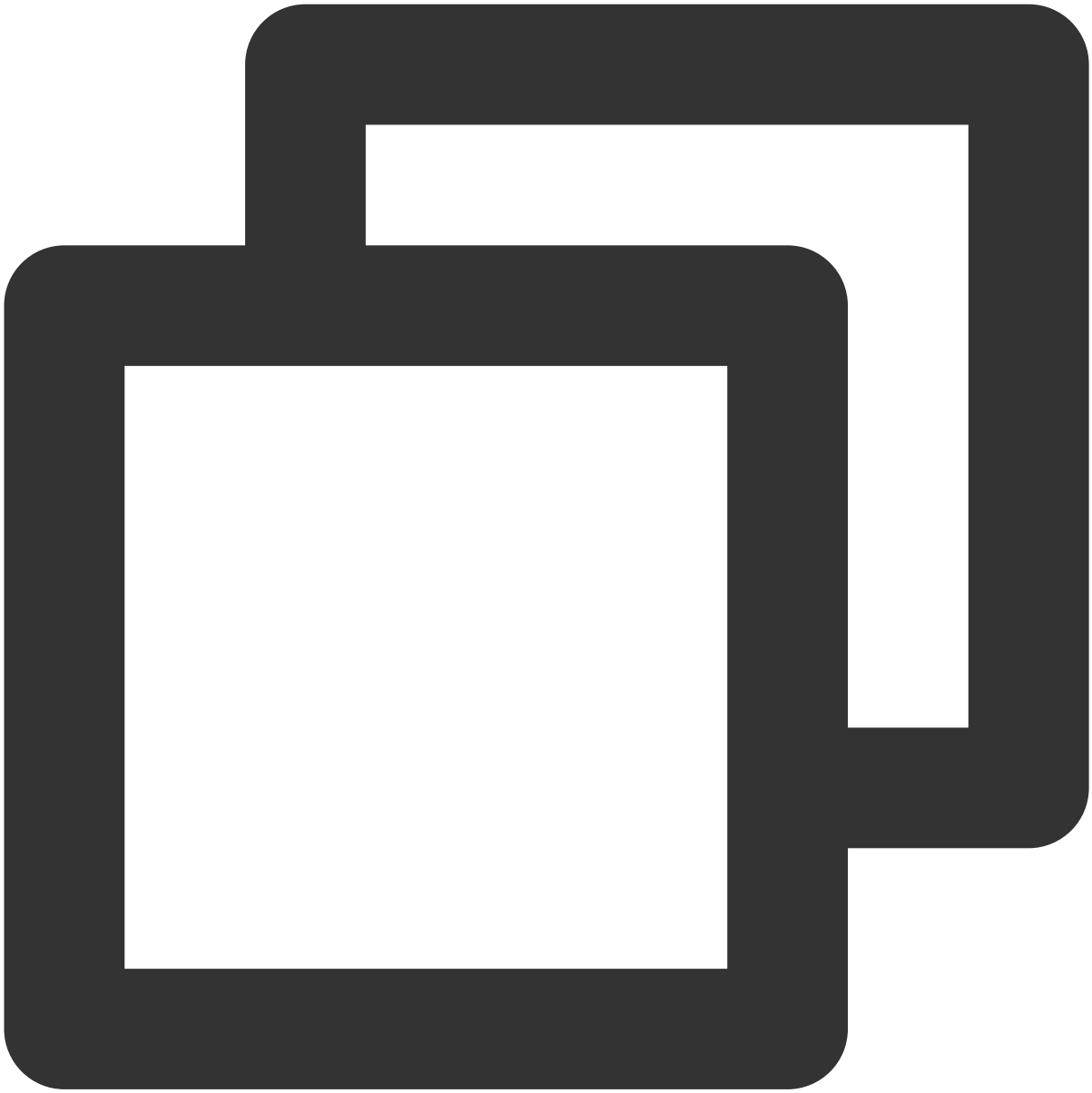
节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Operation.FileCompressResult	打包压缩后文件保存的存储桶的地域。	String
Bucket	Response.JobsDetail.Operation.FileCompressResult	打包压缩后文件保存的存储桶。	String
Object	Response.JobsDetail.Operation.FileCompressResult	打包压缩后文件的名称。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

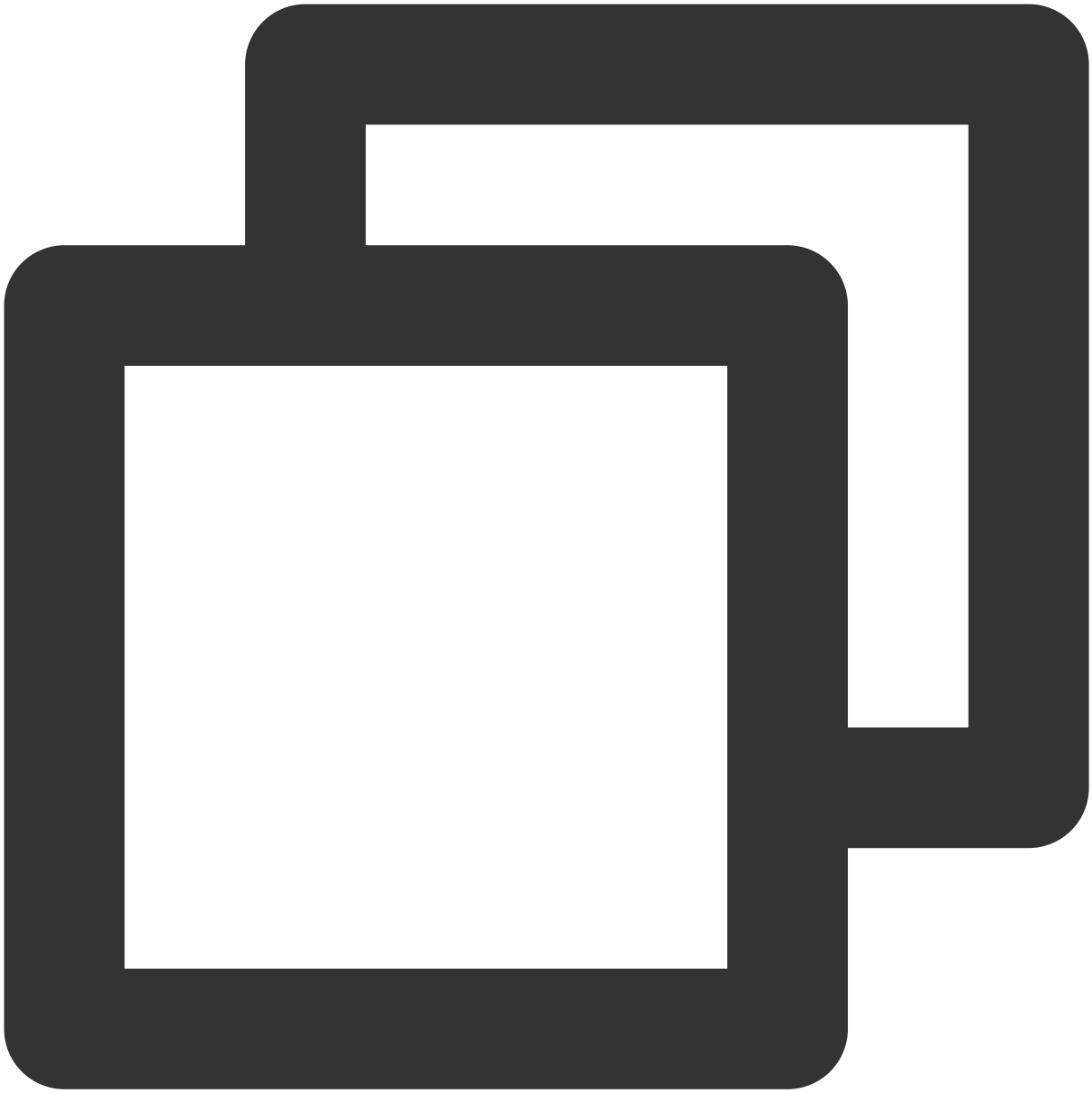
请求



```
POST /file_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>FileCompress</Tag>
    <Operation>
        <FileCompressConfig>
            <Prefix>test/</Prefix>
```

```
            <Format>zip</Format>
            <Flatten>0</Flatten>
        </FileCompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/test.zip</Object>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileCompress</Tag>
        <Operation>
            <FileCompressConfig>
                <Prefix>test/</Prefix>
                <Format>zip</Format>
                <Flatten>0</Flatten>
            </FileCompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
                <Object>output/test.zip</Object>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

查询多文件打包压缩结果

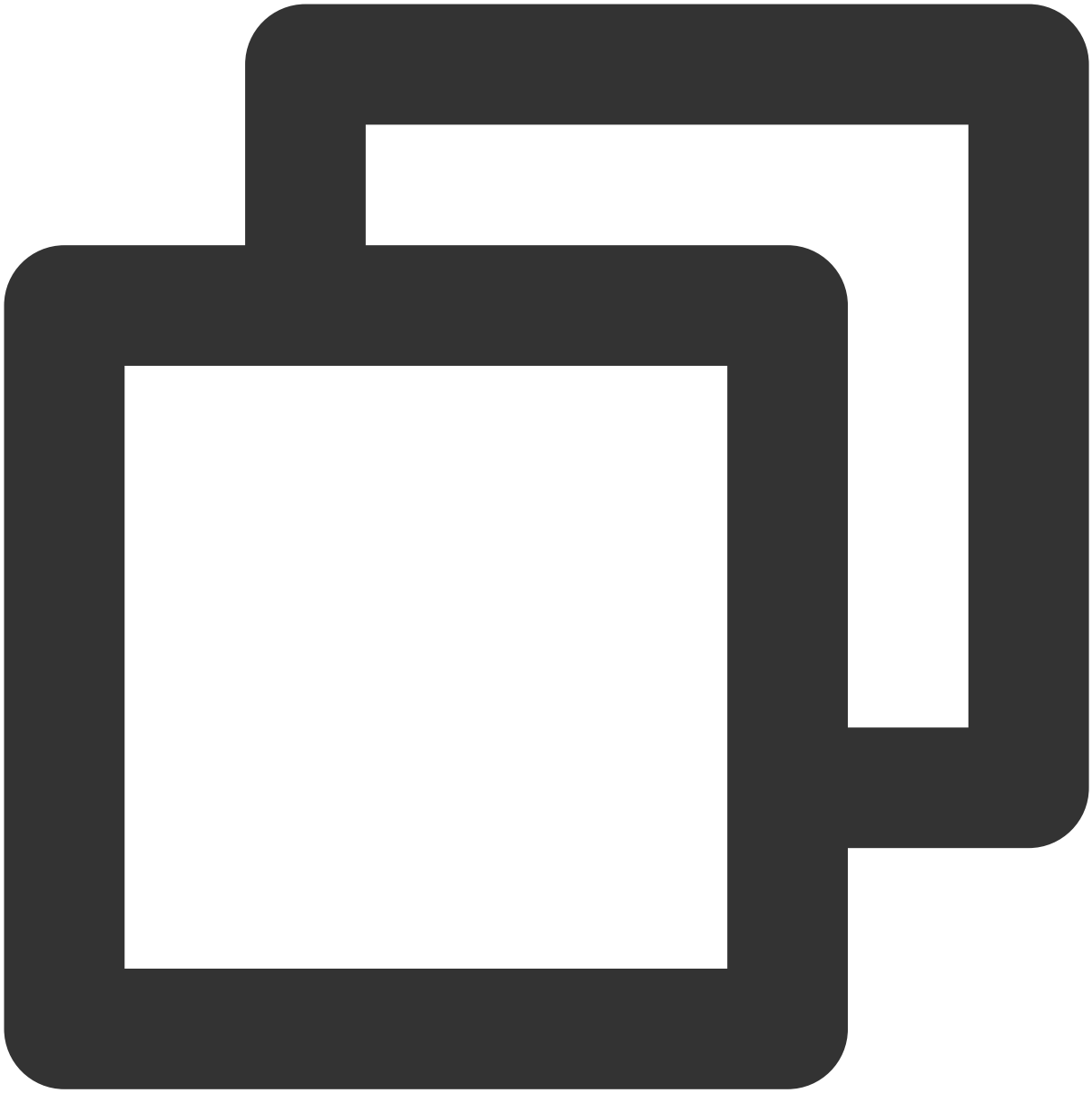
最近更新时间：2023-12-30 10:40:43

功能描述

本接口用于主动查询指定的多文件打包压缩任务结果。

请求

请求示例



```
GET /file_jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

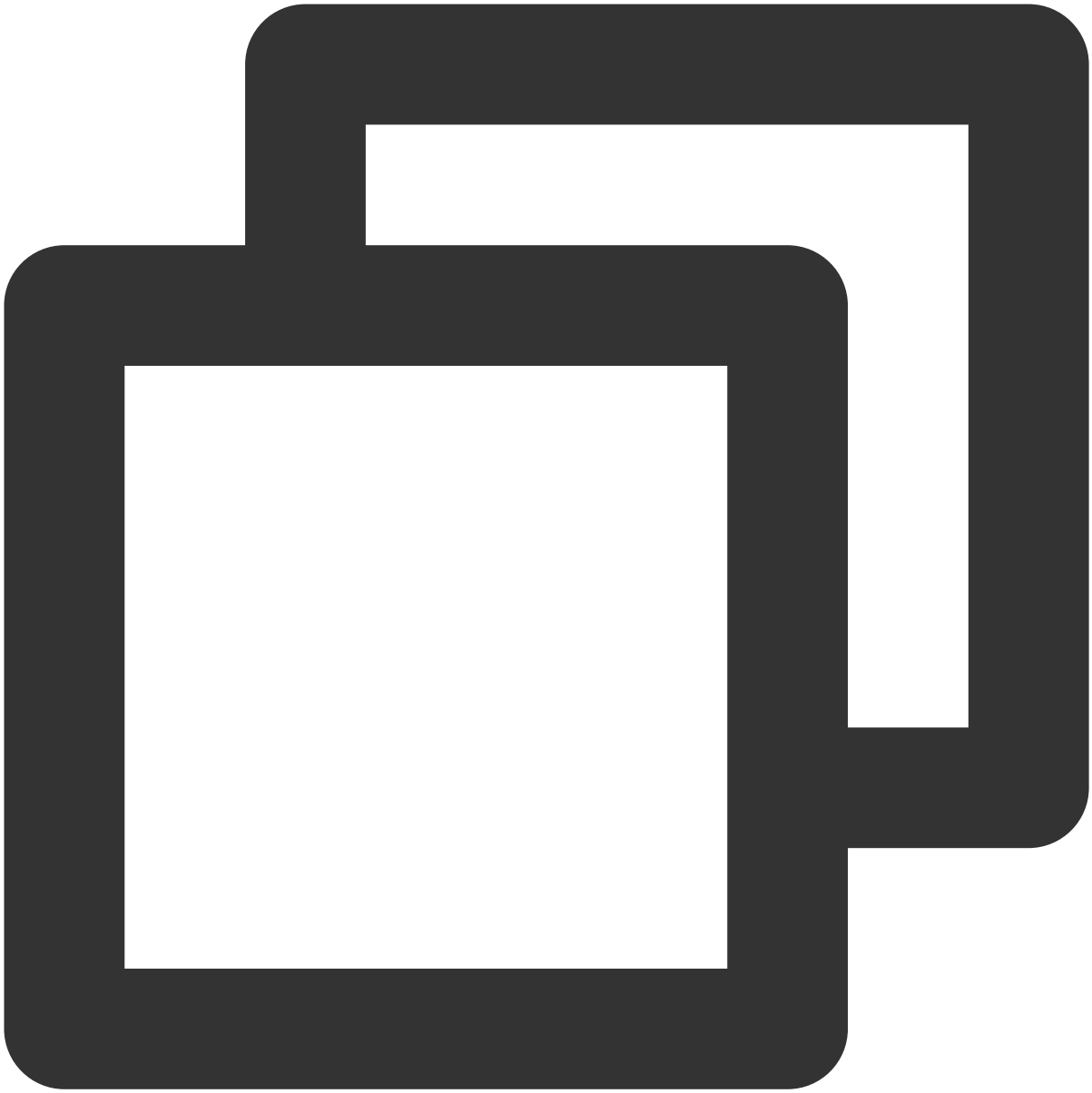
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileCompress</Tag>
```

```
        <Operation>
            <FileCompressConfig>
                <Prefix>test/</Prefix>
                <Format>zip</Format>
                <Flatten>0</Flatten>
            </FileCompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
                <Object>output/test.zip</Object>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息。	Container
NonExistJobIds	Response	查询的 ID 中不存在任务，所有任务都存在时不返回。	String

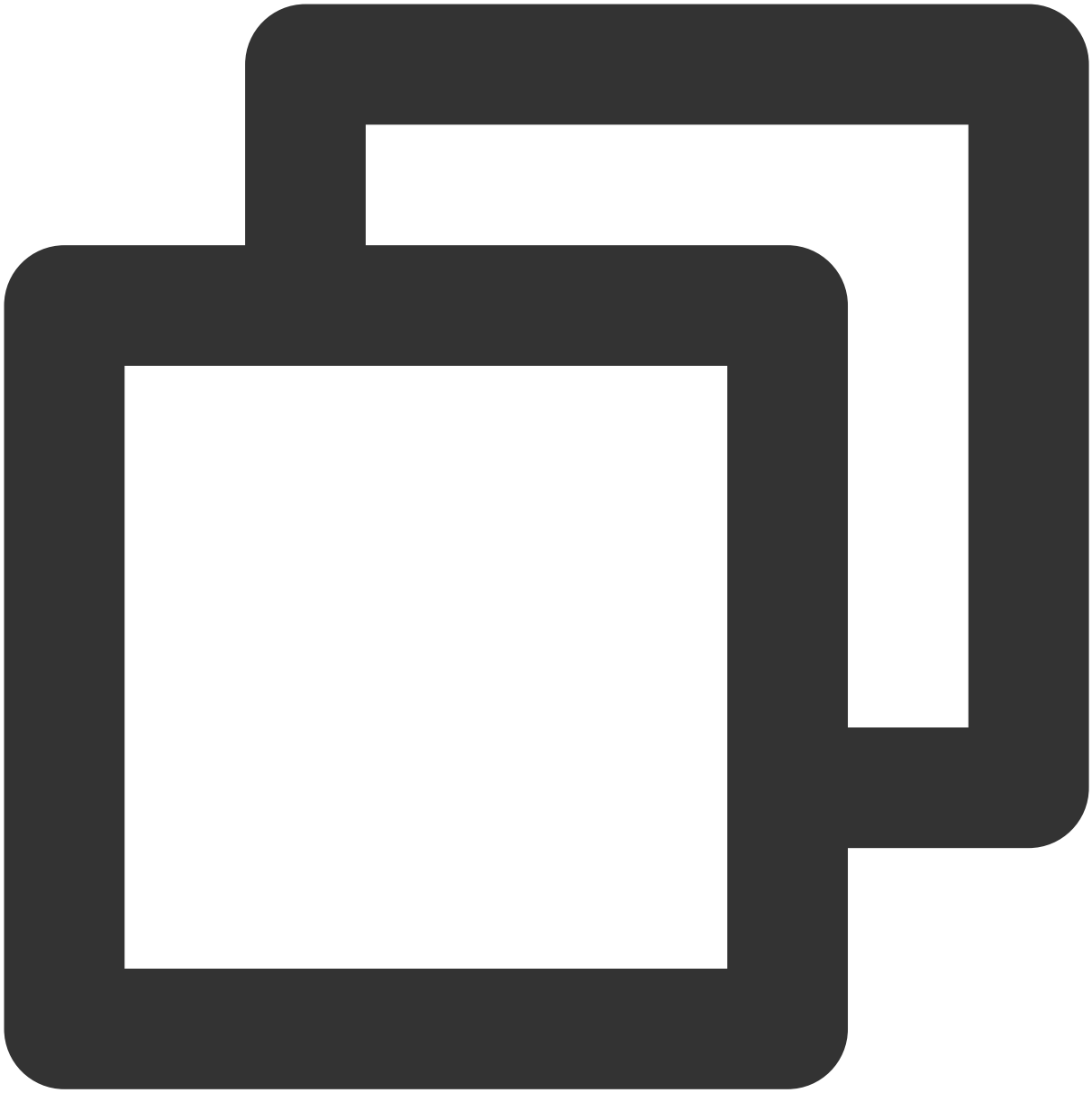
具体参数值，请前往 多文件打包压缩 `JobsDetail` 查看。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

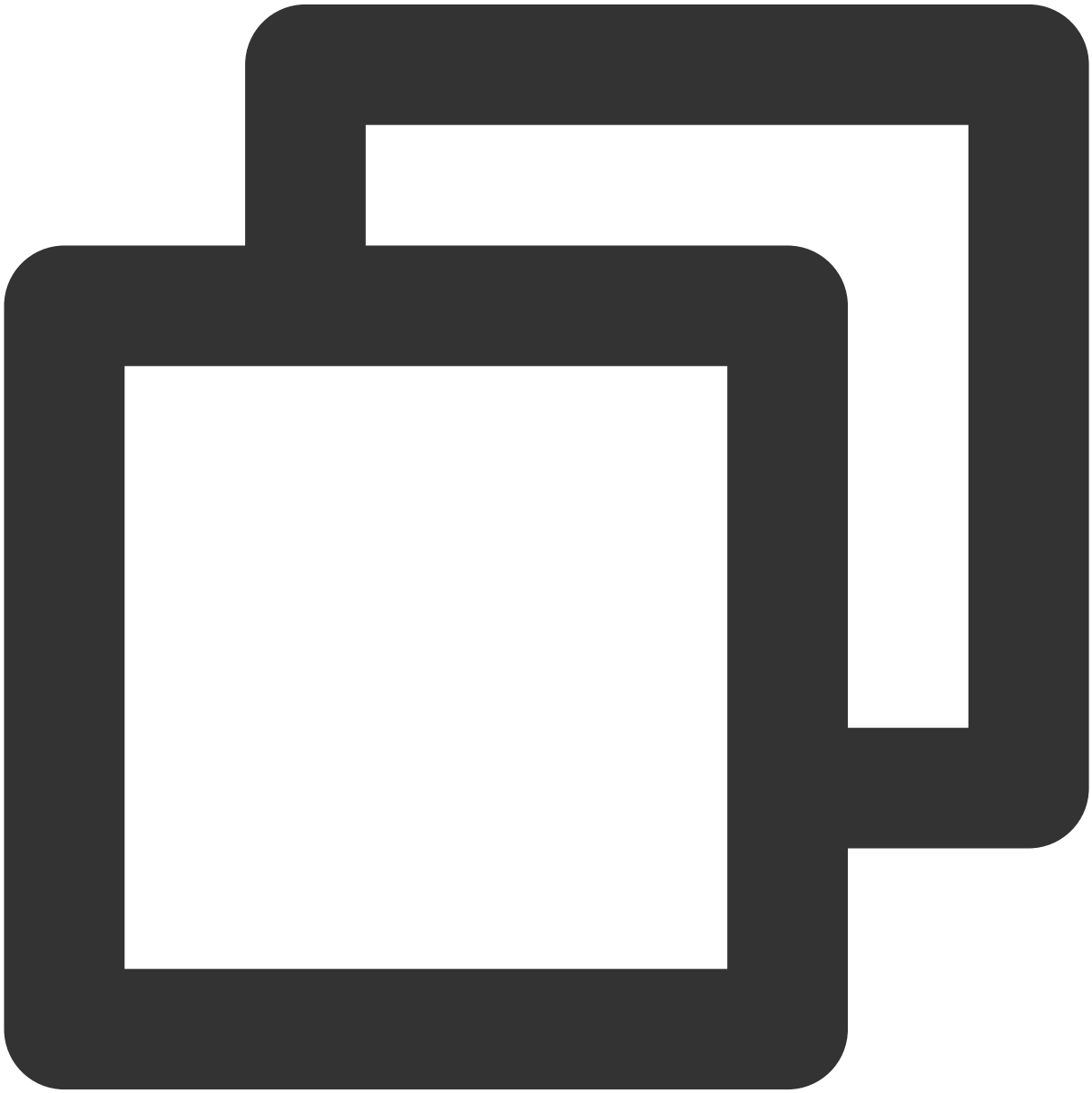
实际案例

请求



```
GET /file_jobs/f93984788066911ed89ed352d4d9d2084 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileCompress</Tag>
        <Operation>
            <FileCompressConfig>
                <Prefix>test/</Prefix>
                <Format>zip</Format>
                <Flatten>0</Flatten>
            </FileCompressConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
                <Object>output/test.zip</Object>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

任务与工作流
公共请求头部

最近更新时间：2023-12-26 17:13:14

描述

此篇文档将为您介绍在使用 API 时候会使用到的公共请求头部（Request Header），下文提到的头部在之后的具体 API 文档中不再赘述。

请求头部列表

Header 名称	描述	类型	是否必选
Authorization	携带鉴权信息，用以验证请求合法性的签名信息。	string	是。
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）。	integer	否。针对 PUT 和 POST 请求，此头部是必选项。
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型（MIME），例如 application/xml。	string	否。针对有请求体的 PUT 和 POST 请求，此头部是必选项。
Content-MD5	RFC 1864 中定义的经过 Base64 编码的请求体内容 MD5 哈希值，例如 ZzD3iDJdrMAAb00lgLLeig==。此头部用于完整性检查，验证请求体在传输过程中是否发生变化。	string	否。
Date	RFC 1123 中定义的 GMT 格式当前时间，例如 Wed, 29 May 2019 04:10:12 GMT。	string	否。
Host	请求的主机，形式为 <BucketName-APPID>.ci.<Region>.myqcloud.com	string	是。
x-ci-security-	使用临时安全凭证时需要传入的安	string	否。当使用临时密钥并

版权所有：腾讯云计算（北京）有限责任公司

token	全令牌字段。		通过 Authorization 携带鉴权信息时，此头部为必选项。

公共响应头部

最近更新时间：2023-12-26 17:12:42

描述

此篇文档将为您介绍在使用 API 时候会出现的公共返回头部（Response Header），下文提到的头部会在之后的具体 API 文档中不再赘述。

响应头部列表

Header 名称	描述	类型
Content-Length	RFC 2616 中定义的 HTTP 响应内容长度（字节）。	string
Content-Type	RFC 2616 中定义的 HTTP 响应内容类型（MIME）。	string
Connection	RFC 2616 中定义，表明响应完成后是否会关闭网络连接。枚举值：keep-alive，close。	Enum
Date	RFC 1123 中定义的 GMT 格式服务端响应时间，例如Wed, 29 May 2019 04:10:12 GMT。	string
Server	接受请求并返回响应的服务器的名称，默认值：tencent-ci。	string
x-ci-request-id	每次请求发送时，服务端将会自动为请求生成一个ID。	string
x-ci-trace-id	每次请求出错时，服务端将会自动为这个错误生成一个ID。仅当请求出错时才会在响应中包含此头部。	string

版权所有：腾讯云计算（北京）有限责任公司

错误码

最近更新时间：2023-12-26 17:12:35

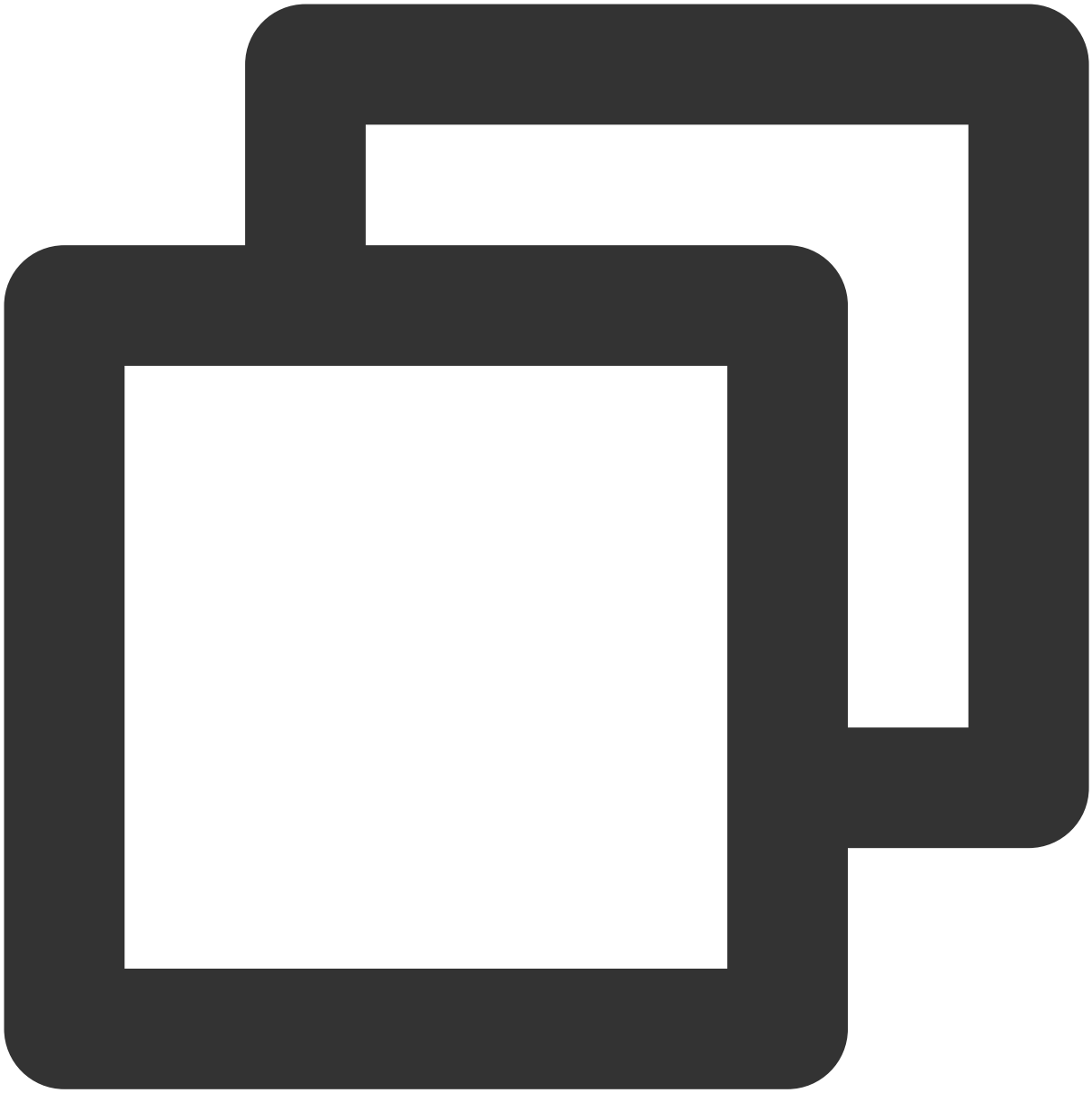
概述

本文将为您介绍请求出错时返回的错误码和对应错误信息。

错误响应

Content-Type：application/xml
对应 HTTP 状态码：3XX，4XX，5XX。

响应体



```
<?xml version='1.0' encoding='utf-8' ?>
<Error>
    <Code>string</Code>
    <Message>string</Message>
    <RequestId>string</RequestId>
    <TraceId>string</TraceId>
</Error>
```

具体的节点描述如下：

节点名称（关键字）	父节点	描述	类型

Error	无	包含所有的错误信息	Container

Container 节点 Error 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Error	错误码，用来定位唯一的错误条件和确定错误场景，具体错误码见下文	string
Message	Error	具体的错误信息	string
RequestId	Error	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题	string
TraceId	Error	每次请求出错时，服务端将会自动为这个错误生成一个ID，遇到问题时，该 ID 能更快地协助定位问题	string

错误码列表

3XX 类型错误

错误码	描述	HTTP状态码
PermanentRedirect	该资源已经被永久改变了位置，请利用 HTTP Location 响应头部来重定向到正确的新位置	301 Moved Permanently
TemporaryRedirect	该资源已经被临时改变了位置，请利用 HTTP Location 响应头部来重定向到正确的新位置	307 Temporary Redirect

4XX 类型错误

错误码	描述	HTTP 状态码
InvalidArgument	请求的部分参数不符合要求	400 Bad Request
MissingContent	缺少包体	400 Bad Request
InvalidContent	发送的包体格式不正确或者与 Content-MD5 值不同	400 Bad Request
InvalidHost	Host 验证失败	400 Bad Request
ResourceNotFound	指定的输入资源不存在	400 Bad Request

版权所有：腾讯云计算（北京）有限责任公司

ResourceAccessDenied	没有权限操作指定的资源，请检查是否赋权	400 Bad Request
ResourceDownloadFailed	指定的输入资源下载失败	400 Bad Request
CIRoleNotExist	万象角色CI_QCSRole不存在	400 Bad Request
ResourceNotSupported	文件有问题或不支持	400 Bad Request
MissingAuthorization	缺少 Authorization	401 Unauthorized
InvalidAuthorization	Authorization 格式不正确	401 Unauthorized
AccessDenied	没有权限	401 Unauthorized
SecretidNotExist	该密钥不存在	401 Unauthorized
SignatureExpired	签名已过期	401 Unauthorized
AccessDenied	禁止此操作	403 Forbidden
MediaBucketAlreadyBinded	该存储桶已经被绑定	403 Forbidden
QueueSaturated	该队列已满	403 Forbidden
QueuePaused	该队列已暂停	403 Forbidden
InvalidUrl	请求的 Url 格式不正确	404 Not Found
AuditingJobNotFound	审核Job不存在	404 Not Found
MissingContentLength	Content-Length 请求头部缺失	411 Length Required
InvalidContent	发送的包体过大	413 Payload Too Large
SlowDown	请降低访问频率	429 Too Many Requests
UnavailableForLegalReasons	因法律原因不可用。请 提交工单	451 Unavailable For Legal Reasons

5XX 类型错误

错误码	描述	HTTP 状态码
InternalError	服务端内部错误	500 Internal Server
NotImplemented	请求尚未实现	501 Not Implemented
ServiceUnavailable	服务暂不可用，请重试	503 Service Unavailable

版权所有：腾讯云计算（北京）有限责任公司

版权所有：腾讯云计算（北京）有限责任公司

工作流接口
创建工作流

最近更新时间：2023-12-26 17:08:25

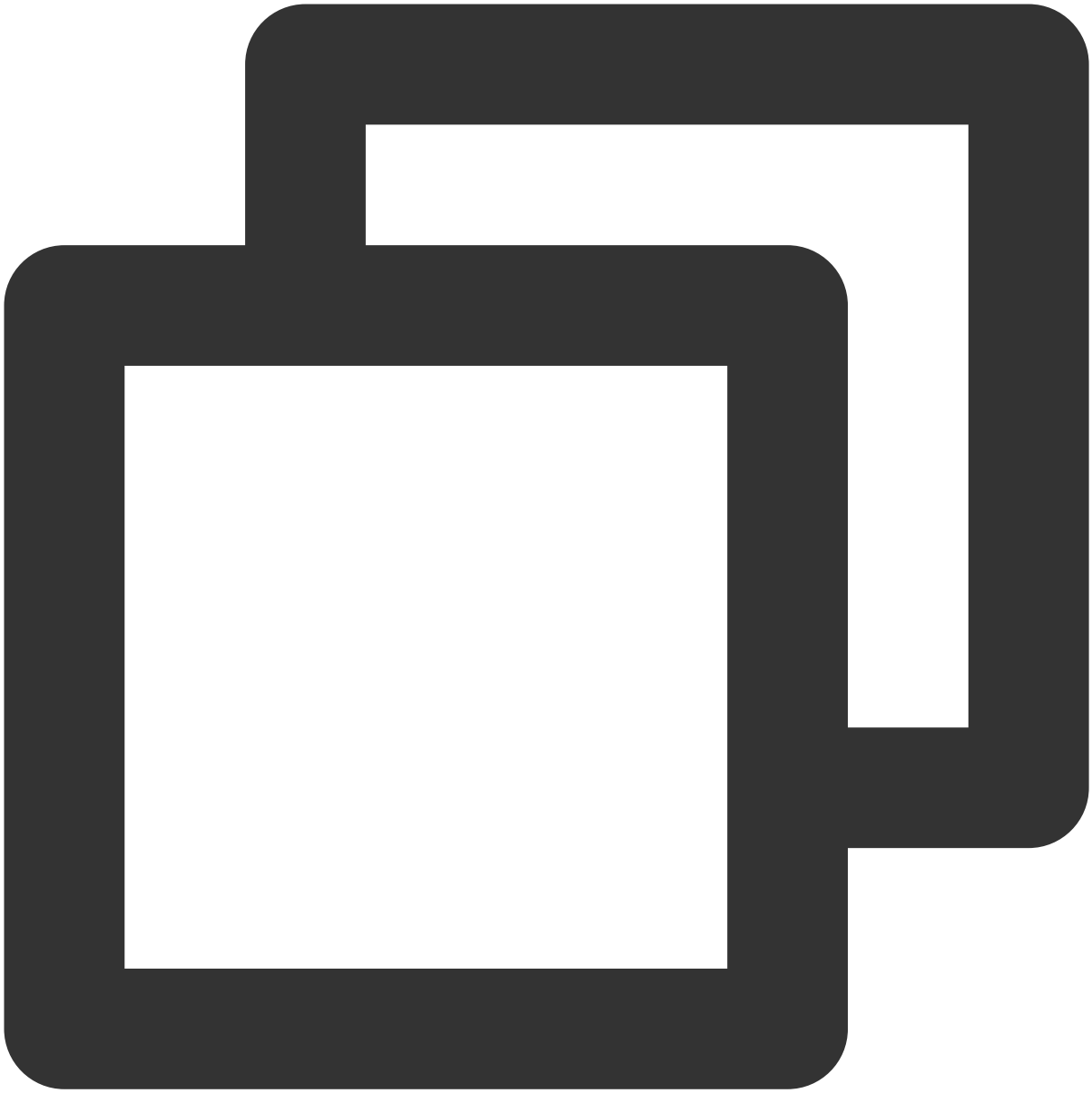
功能描述

创建工作流。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
POST /workflow HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

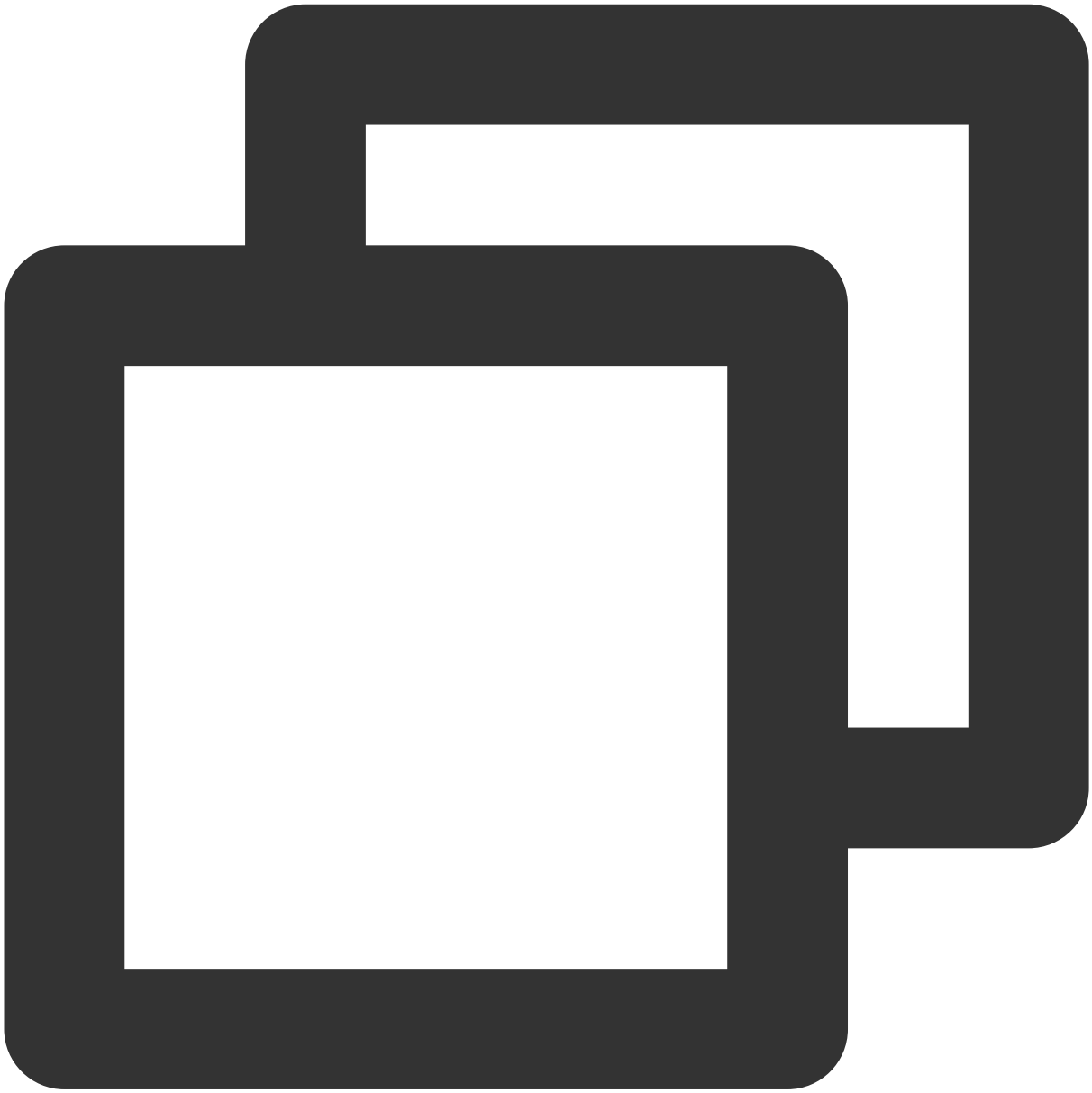
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：
请求体1
请求体2
对于输入的视频文件进行截图和转码处理。视频转码后，同时进行转封装处理和生成智能封面，然后对生成的封面进行图片处理。



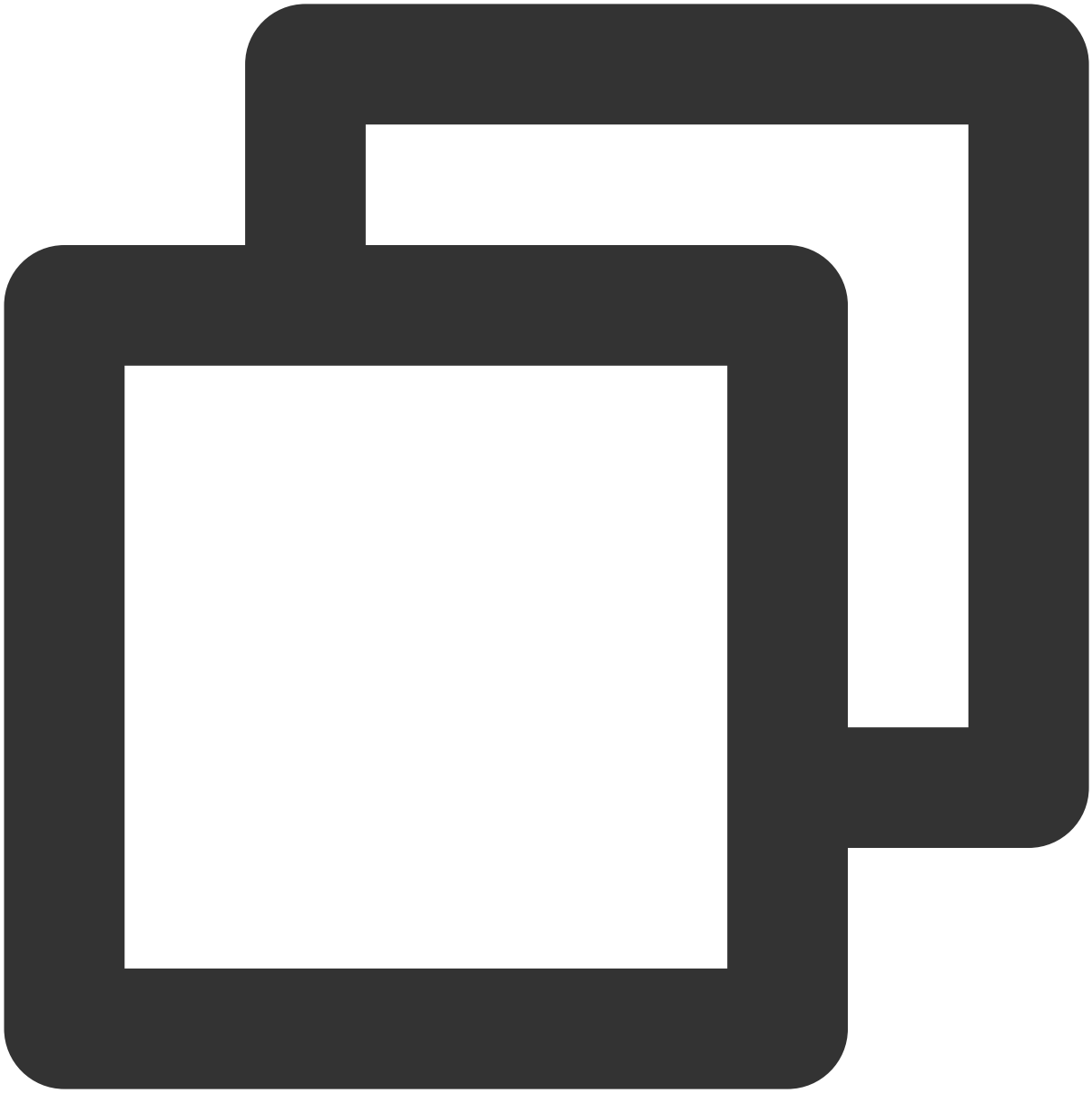
```
<Request>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <Topology>
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <PicProcess_15816659605668>End</PicProcess_15816659605668>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                    <ObjectPrefix>input/workflow-1</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <Snapshot_1581665960536>
                <Type>Snapshot</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                        <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                    </Output>
                </Operation>
            </Snapshot_1581665960536>
            <Transcode_1581665960538>
                <Type>Transcode</Type>
                <Operation>
                    <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>bcd/${RunId}/trans.{Ext}</Object>
                    </Output>
```

```
                    </Operation>
                </Transcode_1581665960538>
                <Segment_15816659605667>
                    <Type>Segment</Type>
                    <Operation>
                        <Segment>
                            <Format>mkv</Format>
                            <Duration>20</Duration>
                        </Segment>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>test-trans${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </Segment_15816659605667>
                <SmartCover_1581665960539>
                    <Type>SmartCover</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </SmartCover_1581665960539>
                <PicProcess_15816659605668>
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Request>
```

自适应码流



```
<Request>
    <MediaWorkflow>
        <Name>workflow-2</Name>
        <State>Active</State>
        <Topology>
            <Dependencies>
                <Start>StreamPackConfig_1581665960532</Start>
                <StreamPackConfig_1581665960532>VideoStream_1581665960536,VideoStre
                <VideoStream_1581665960536>StreamPack_1581665960538</VideoStream_15
                <VideoStream_1581665960537>StreamPack_1581665960538</VideoStream_15
                <StreamPack_1581665960538>End</StreamPack_1581665960538>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <ObjectPrefix>input/workflow-2</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <StreamPackConfig_1581665960532>
                    <Type>StreamPackConfig</Type>
                    <Operation>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${InputPath}/${InputName}._${RunId}.${ext}</Obj
                        </Output>
                        <StreamPackConfig>
                            <PackType>HLS</PackType>
                            <IgnoreFailedStream>true</IgnoreFailedStream>
                        </StreamPackConfig>
                    </Operation>
                </StreamPackConfig_1581665960532>
                <VideoStream_1581665960536>
                    <Type>VideoStream</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_1/video.m3u8</Object>
                        </Output>
```

```
                    </Operation>
                </VideoStream_1581665960536>
                <VideoStream_1581665960537>
                    <Type>VideoStream</Type>
                    <Operation>
                        <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_2/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960537>
                <StreamPack_1581665960538>
                    <Type>StreamPack</Type>
                    <Operation>
                        <StreamPackInfo>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960536</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960537</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                        </StreamPackInfo>
                    </Operation>
                </StreamPack_1581665960538>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
MediaWorkflow	Request	工作流节点	Container	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 MediaWorkflow 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Name	Request.MediaWorkflow	工作流名称	String	是	支持中文、英文、数字、—和_，长度限制128字符
State	Request.MediaWorkflow	工作流状态	String	否	Paused/Active
Topology	Request.MediaWorkflow	拓扑信息	Container	是	无

Container 类型 Topology 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Dependencies	Request.MediaWorkflow.Topology	节点依赖关系	Container	是	无
Nodes	Request.MediaWorkflow.Topology	节点列表	Container	是	无

Container 类型 Nodes 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限
Start	Request.MediaWorkflow.Topology.Nodes	开始节点	Container	是	只节
Animation_***	Request.MediaWorkflow.Topology.Nodes	动图类型节点	Container	否	节A节点
Snapshot_***	Request.MediaWorkflow.Topology.Nodes	截图类型节点	Container	否	节S节点
SmartCover_***	Request.MediaWorkflow.Topology.Nodes	智能封面节点	Container	否	节S参自
Transcode_***	Request.MediaWorkflow.Topology.Nodes	转码节点	Container	否	节

版权所有：腾讯云计算（北京）有限责任公司

					T 约 在
Concat_***	Request.MediaWorkflow.Topology.Nodes	音视频拼接节点	Container	否	节 入 /
VoiceSeparate_***	Request.MediaWorkflow.Topology.Nodes	人声节点	Container	否	节 V 前 人
VideoMontage_***	Request.MediaWorkflow.Topology.Nodes	精彩集锦节点	Container	否	节 V 约 采
StreamPackConfig_***	Request.MediaWorkflow.Topology.Nodes	自适应码流节点	Container	否	节 S 入 / 点 点 长 百 流
VideoStream_***	Request.MediaWorkflow.Topology.Nodes	视频子流节点	Container	否	节 V 约 步 在 S 节 前 节
StreamPack_***	Request.MediaWorkflow.Topology.Nodes	自适应码流打包节点	Container	否	节 S 约 适 ， 节 前
SDRtoHDR_***	Request.MediaWorkflow.Topology.Nodes	SDRtoHDR	Container	否	节

		节点			S 约 S
VideoProcess_***	Request.MediaWorkflow.Topology.Nodes	视频处理节点	Container	否	节 V 约 步
SCF_***	Request.MediaWorkflow.Topology.Nodes	SCF 函数节点	Container	否	节 自 S
SuperResolution_***	Request.MediaWorkflow.Topology.Nodes	超分辨率节点	Container	否	节 S 自 超
Segment_***	Request.MediaWorkflow.Topology.Nodes	音视频转封装节点	Container	否	节 S 音 转
PicProcess_***	Request.MediaWorkflow.Topology.Nodes	图片处理节点	Container	否	节 P 约 片
Tts_***	Request.MediaWorkflow.Topology.Nodes	语音合成节点	Container	否	节 自 语
SpeechRecognition_***	Request.MediaWorkflow.Topology.Nodes	语音识别节点	Container	否	节 S 文 /
VideoEnhance_***	Request.MediaWorkflow.Topology.Nodes	画质增强节点	Container	否	节 V 约 音
Condition_***	Request.MediaWorkflow.Topology.Nodes	条件判断节点	Container	否	节 C 音 断

MediaInfo_***	Request.MediaWorkflow.Topology.Nodes	媒体信息获取节点	Container	否	节 M 信 息
AIRecognition_***	Request.MediaWorkflow.Topology.Nodes	AI识别节点	Container	否	节 A 结
SegmentVideoBody_***	Request.MediaWorkflow.Topology.Nodes	视频人像抠图	Container	否	节 S 为

Container 类型 Start 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Start	节点类型	String	是	Start
Input	Request.MediaWorkflow.Topology.Nodes.Start	输入信息, 详情见 Input	Container	是	无

Container 类型 Animation_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Animation_***	节点类型	String	是	Anim
Operation	Request.MediaWorkflow.Topology.Nodes.Animation_***	操作规则, 详情见 Animation	Container	是	无

Container 类型 Snapshot_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Snapshot_***	节点类型	String	是	Snap

版权所有：腾讯云计算（北京）有限责任公司

Operation	Request.MediaWorkflow.Topology.Nodes.Snapshot_***	操作规则, 详情见 Snapshot	Container	是	无

Container 类型 SmartCover_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	
Type	Request.MediaWorkflow.Topology.Nodes.SmartCover_***	节点类型	String	是	
Operation	Request.MediaWorkflow.Topology.Nodes.SmartCover_***	操作规则, 详情见 SmartCover	Container	是	

Container 类型 Transcode_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限
Type	Request.MediaWorkflow.Topology.Nodes.Transcode_***	节点类型	String	是	Tra
Operation	Request.MediaWorkflow.Topology.Nodes.Transcode_***	操作规则, 详情见 Transcode	Container	是	无

Container 类型 Concat_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Concat_***	节点类型	String	是	Concat
Operation	Request.MediaWorkflow.Topology.Nodes.Concat_***	操作规则, 详情见 Concat	Container	是	无

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 VoiceSeparate_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.MediaWorkflow.Topology.Nodes.VoiceSeparate_***	节点类型	String	是
Operation	Request.MediaWorkflow.Topology.Nodes.VoiceSeparate_***	操作规则，详情见 VoiceSeparate	Container	是

Container 类型 VideoMontage_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.MediaWorkflow.Topology.Nodes.VideoMontage_***	节点类型	String	是
Operation	Request.MediaWorkflow.Topology.Nodes.VideoMontage_***	操作规则，详情见 VideoMontage	Container	是

Container 类型 StreamPackConfig_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Type	Request.MediaWorkflow.Topology.Nodes.StreamPackConfig_***	节点类型	String
Operation	Request.MediaWorkflow.Topology.Nodes.StreamPackConfig_***	操作规则，详情见 StreamPackConfig	Contain

Container 类型 VideoStream_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.MediaWorkflow.Topology.Nodes.VideoStream_***	节点类型	String	是

Operation	Request.MediaWorkflow.Topology.Nodes.VideoStream_***	操作规则，详情见 VideoStream	Container	是

Container 类型 StreamPack_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.MediaWorkflow.Topology.Nodes.StreamPack_***	节点类型	String	是
Operation	Request.MediaWorkflow.Topology.Nodes.StreamPack_***	操作规则，详情见 StreamPack	Container	是

Container 类型 SDRtoHDR_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限
Type	Request.MediaWorkflow.Topology.Nodes.SDRtoHDR_***	节点类型	String	是	S
Operation	Request.MediaWorkflow.Topology.Nodes.SDRtoHDR_***	操作规则，详情见 SDRtoHDR	Container	是	无

Container 类型 VideoProcess_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.MediaWorkflow.Topology.Nodes.VideoProcess_***	节点类型	String	是
Operation	Request.MediaWorkflow.Topology.Nodes.VideoProcess_***	操作规则，详情见 VideoProcess	Container	是

Container 类型 SCF_*** 的具体数据描述如下：

节点名称	父节点	描述	类型	是	限制

版权所有：腾讯云计算（北京）有限责任公司

（关键字）				否必选	
Type	Request.MediaWorkflow.Topology.Nodes.SCF_***	节点类型	String	是	SCF
Operation	Request.MediaWorkflow.Topology.Nodes.SCF_***	操作规则，详情见 SCF	Container	是	无

Container 类型 SuperResolution_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Type	Request.MediaWorkflow.Topology.Nodes.SuperResolution_***	节点类型	String
Operation	Request.MediaWorkflow.Topology.Nodes.SuperResolution_***	操作规则，详情见 SuperResolution	Container

Container 类型 Segment_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Segment_***	节点类型	String	是	Segme
Operation	Request.MediaWorkflow.Topology.Nodes.Segment_***	操作规则，详情见 Segment	Container	是	无

Container 类型 PicProcess_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限

版权所有：腾讯云计算（北京）有限责任公司

Type	Request.MediaWorkflow.Topology.Nodes.PicProcess_***	节点类型	String	是	P
Operation	Request.MediaWorkflow.Topology.Nodes.PicProcess_***	操作规则，详情见 PicProcess	Container	是	无

Container 类型 Tts_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Tts_***	节点类型	String	是	Tts
Operation	Request.MediaWorkflow.Topology.Nodes.Tts_***	操作规则，详情见 Tts	Container	是	无

Container 类型 SpeechRecognition_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Type	Request.MediaWorkflow.Topology.Nodes.SpeechRecognition_***	节点类型	String
Operation	Request.MediaWorkflow.Topology.Nodes.SpeechRecognition_***	操作规则，详情见 SpeechRecognition	Contai

Container 类型 VideoEnhance_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Type	Request.MediaWorkflow.Topology.Nodes.VideoEnhance_***	节点类型	String	是
Operation	Request.MediaWorkflow.Topology.Nodes.VideoEnhance_***	操作规则，详情见 VideoEnhance	Container	是

Container 类型 Condition_*** 的具体数据描述如下：

节点名称	父节点	描述	类型	是	限制

（关键字）				否 必 选	
Type	Request.MediaWorkflow.Topology.Nodes.Condition_***	节点类型	String	是	Cond
Operation	Request.MediaWorkflow.Topology.Nodes.Condition_***	操作规则, 详情见 Condition	Container	是	无

Container 类型 MediaInfo_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.MediaInfo_***	节点类型	String	是	MediaInfo

Container 类型 AIRecognition_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.AIRecognition_***	节点类型	String	是	AIRecognition

Container 类型 SegmentVideoBody_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Type	Request.MediaWorkflow.Topology.Nodes.SegmentVideoBody_***	节点类型	String
Operation	Request.MediaWorkflow.Topology.Nodes.SegmentVideoBody_***	操作规则, 详情见	Container

版权所有：腾讯云计算（北京）有限责任公司

		VideoEnhance	

工作流节点依赖关系如下：

工作流节点类型	后面可连接的节点类型
Start	Snapshot,Animation,SmartCover,Transcode,Concat,VoiceSeparate,VideoMontage,Stream
Snapshot	PicProcess,SCF,End
Animation	SCF,End
SmartCover	PicProcess,SCF,End
Transcode	Snapshot,Animation,SmartCover,Concat,VideoMontage,SDRtoHDR,VideoProcess,SCF,Se
Concat	Transcode,Snapshot,Animation,SmartCover,SDRtoHDR,VideoProcess,SCF,Segment,Vide
VoiceSeparate	SCF,SpeechRecognition,End
VideoMontage	Snapshot,Animation,SDRtoHDR,VideoProcess,SCF,SuperResolution,VideoEnhance,Segm
StreamPackConfig	VideoStream
VideoStream	StreamPack
StreamPack	End
SDRtoHDR	Transcode,Concat,VideoMontage,VoiceSeparate,SmartCover,Animation,VideoProcess,SC
VideoProcess	Transcode,Concat,VideoMontage,VoiceSeparate,SmartCover,Animation,SDRtoHDR,SCF,
SCF	SDRtoHDR,Snapshot,Transcode,Concat,VideoMontage,VoiceSeparate,SmartCover,Anima
SuperResolution	VideoMontage,Transcode,SmartCover,Animation,Snapshot,SCF,Segment,SDRtoHDR,Vid
Segment	VideoEnhance,SegmentVideoBody, End
PicProcess	SCF,End
Tts	SCF,Transcode,End
SpeechRecognition	SCF,Tts,End
VideoEnhance	Transcode,VideoMontage,SmartCover,Animation,Snapshot,SCF,Segment,SpeechRecogni
MediaInfo	Condition,End
Condition	Transcode,HighSpeedHd,SuperResolution,VideoProcess,Animation,Snapshot,VideoMonta

AIRecognition	End
SegmentVideoBody	End

工作流支持的通配符：

通配符	含义
${InputPath}	输入文件的路径(不包括文件名)
${InputName}	输入文件的文件名(不包括后缀)
${InputNameAndExt}	输入文件的文件名(包括后缀)
${RunId}	实例 Id
${Ext}	封装格式后缀
${Jobid}	任务Id
${Number}	产物索引, 从0开始计数, 仅支持 Snapshot、SmartCover、Segment 节点
${Time}	截图时间点(单位:毫秒), 仅支持 Snapshot 节点

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

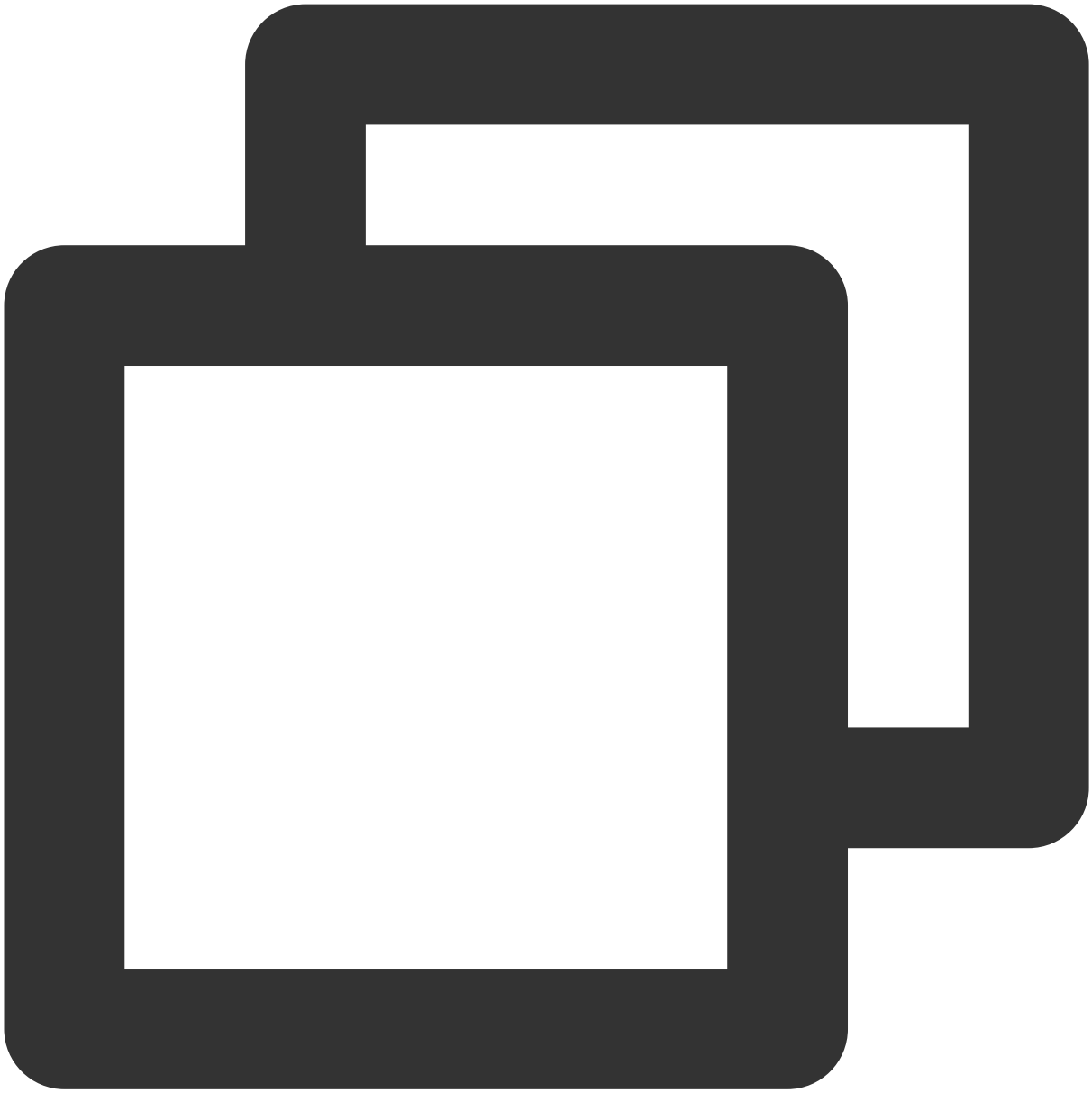
响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

响应体1

响应体2

对于输入的视频文件进行截图和转码处理。视频转码后, 同时进行转封装处理和生成智能封面, 然后对生成的封面进行图片处理。

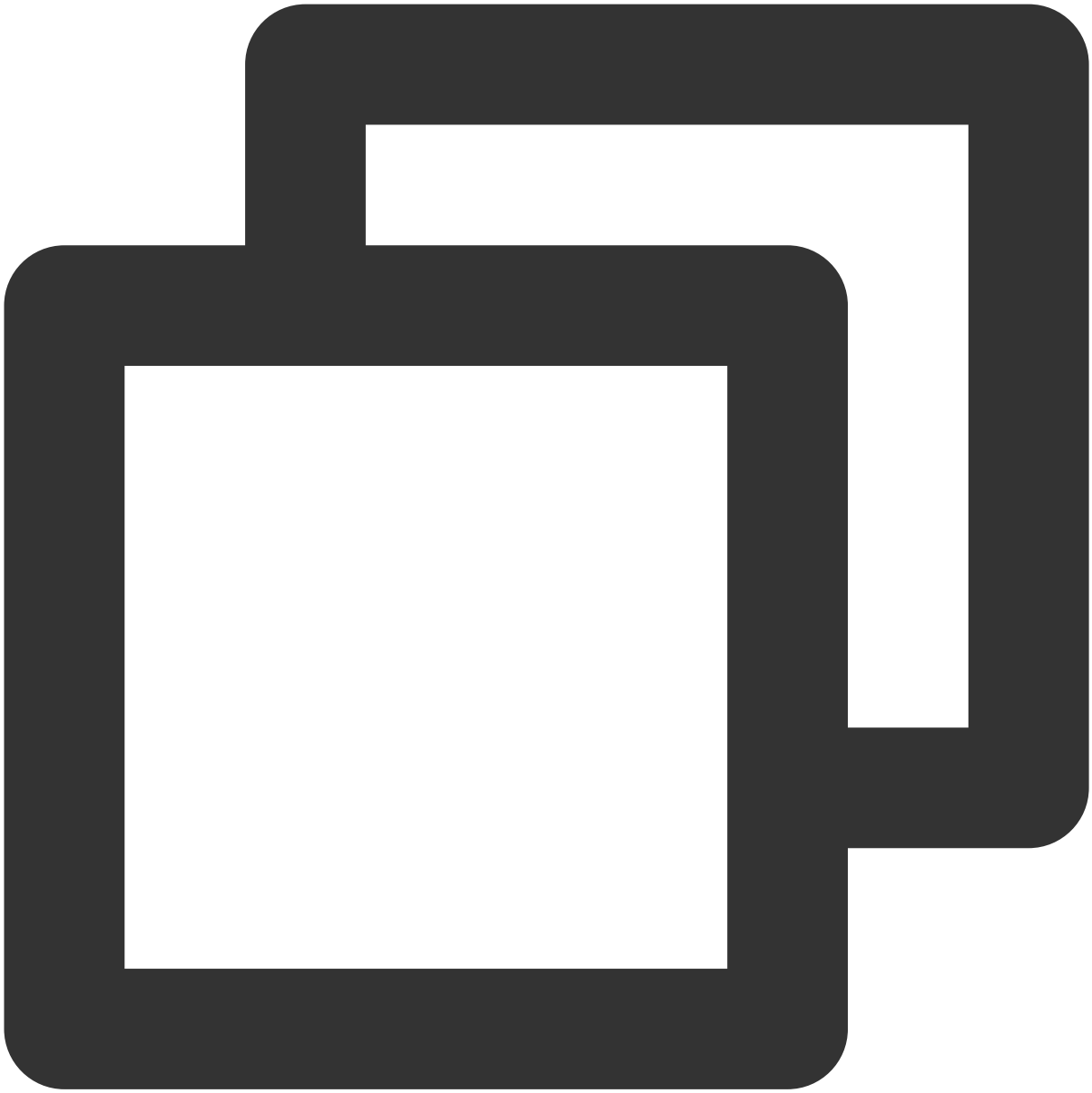


```
<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-07-14T12:37:28+0800</CreateTime>
        <UpdateTime>2022-07-14T12:37:28+0800</UpdateTime>
        <Topology>
            <Dependencies>
```

```
            <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
            <Snapshot_1581665960536>End</Snapshot_1581665960536>
            <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
            <Segment_15816659605667>End</Segment_15816659605667>
            <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
            <PicProcess_15816659605668>End</PicProcess_15816659605668>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                    <ObjectPrefix>input/workflow-1</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <Snapshot_1581665960536>
                <Type>Snapshot</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                        <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                    </Output>
                </Operation>
            </Snapshot_1581665960536>
            <Transcode_1581665960538>
                <Type>Transcode</Type>
                <Operation>
                    <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
```

```
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/trans.{Ext}</Object>
                        </Output>
                    </Operation>
                </Transcode_1581665960538>
                <Segment_15816659605667>
                    <Type>Segment</Type>
                    <Operation>
                        <Segment>
                            <Format>mkv</Format>
                            <Duration>20</Duration>
                        </Segment>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>test-trans${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </Segment_15816659605667>
                <SmartCover_1581665960539>
                    <Type>SmartCover</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </SmartCover_1581665960539>
                <PicProcess_15816659605668>
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Response>
```

自适应码流



```
<Response>
    <MediaWorkflow>
        <Name>workflow-2</Name>
        <State>Active</State>
        <WorkflowId>w93aa43ba105347169fa093ed857b2a90</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-07-14T12:37:28+0800</CreateTime>
        <UpdateTime>2022-07-14T12:37:28+0800</UpdateTime>
        <Topology>
```

```
        <Dependencies>
            <Start>StreamPackConfig_1581665960532</Start>
            <StreamPackConfig_1581665960532>VideoStream_1581665960536,VideoStre
            <VideoStream_1581665960536>StreamPack_1581665960538</VideoStream_15
            <VideoStream_1581665960537>StreamPack_1581665960538</VideoStream_15
            <StreamPack_1581665960538>End</StreamPack_1581665960538>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <ObjectPrefix>input/workflow-2</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <StreamPackConfig_1581665960532>
                <Type>StreamPackConfig</Type>
                <Operation>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>${InputPath}/${InputName}._${RunId}.${ext}</Obj
                    </Output>
                    <StreamPackConfig>
                        <PackType>HLS</PackType>
                        <IgnoreFailedStream>true</IgnoreFailedStream>
                    </StreamPackConfig>
                </Operation>
            </StreamPackConfig_1581665960532>
            <VideoStream_1581665960536>
                <Type>VideoStream</Type>
                <Operation>
```

```
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_1/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960536>
                <VideoStream_1581665960537>
                    <Type>VideoStream</Type>
                    <Operation>
                        <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_2/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960537>
                <StreamPack_1581665960538>
                    <Type>StreamPack</Type>
                    <Operation>
                        <StreamPackInfo>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960536</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960537</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                        </StreamPackInfo>
                    </Operation>
                </StreamPack_1581665960538>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
MediaWorkflow	Response	工作流数组	Container

Container节点 MediaWorkflow 的内容：

节点名称（关键字）	父节点	描述	类型
Name	Response.MediaWorkflow	工作流名称	String
WorkflowId	Response.MediaWorkflow	工作流 ID	String
State	Response.MediaWorkflow	工作流状态	String
CreateTime	Response.MediaWorkflow	创建时间	String
UpdateTime	Response.MediaWorkflow	更新时间	String
Topology	Response.MediaWorkflow	拓扑信息，同请求中的 Request.MediaWorkflow.Topology	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

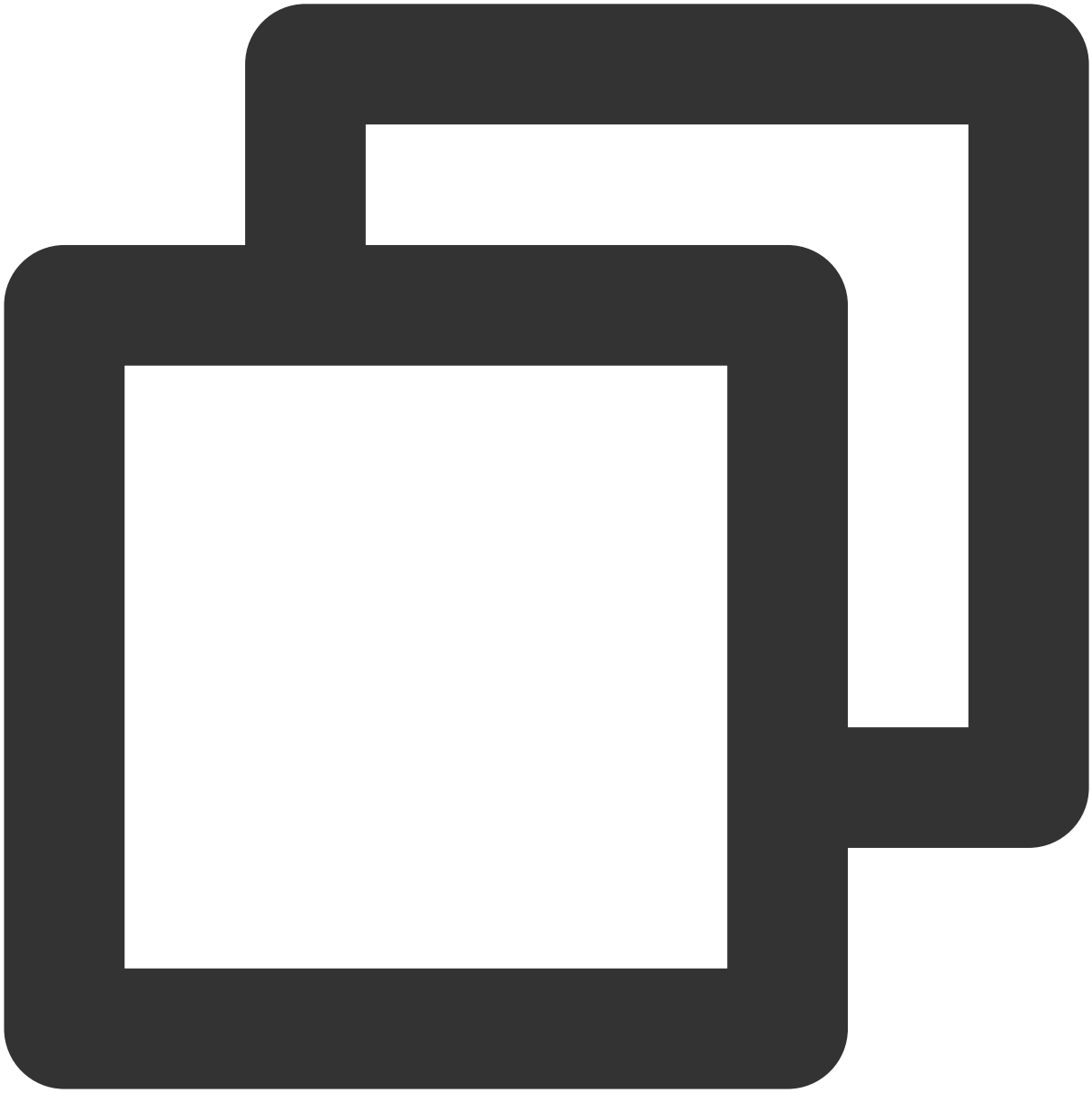
请求1

响应1

请求2

响应2

对于输入的视频文件进行截图和转码处理。视频转码后，同时进行转封装处理和生成智能封面，然后对生成的封面进行图片处理。

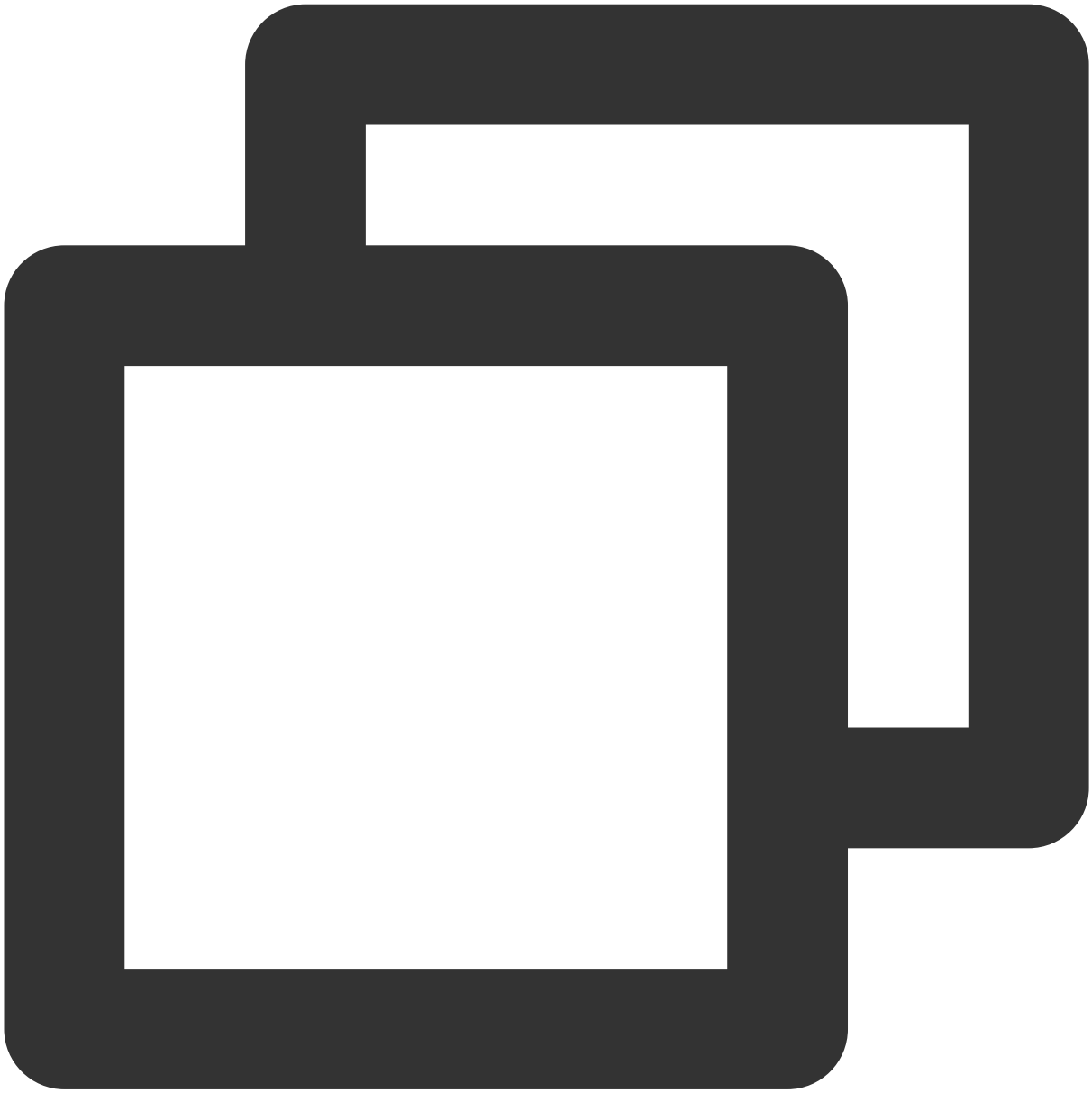


```
POST /workflow HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <Topology>
```

```
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                        <ObjectPrefix>input/workflow-1</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                            <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                        </Output>
                    </Operation>
                </Snapshot_1581665960536>
                <Transcode_1581665960538>
                    <Type>Transcode</Type>
                    <Operation>
```

```
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/trans.{Ext}</Object>
                        </Output>
                    </Operation>
                </Transcode_1581665960538>
                <Segment_15816659605667>
                    <Type>Segment</Type>
                    <Operation>
                        <Segment>
                            <Format>mkv</Format>
                            <Duration>20</Duration>
                        </Segment>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>test-trans${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </Segment_15816659605667>
                <SmartCover_1581665960539>
                    <Type>SmartCover</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </SmartCover_1581665960539>
                <PicProcess_15816659605668>
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
```

```
</Request>
```



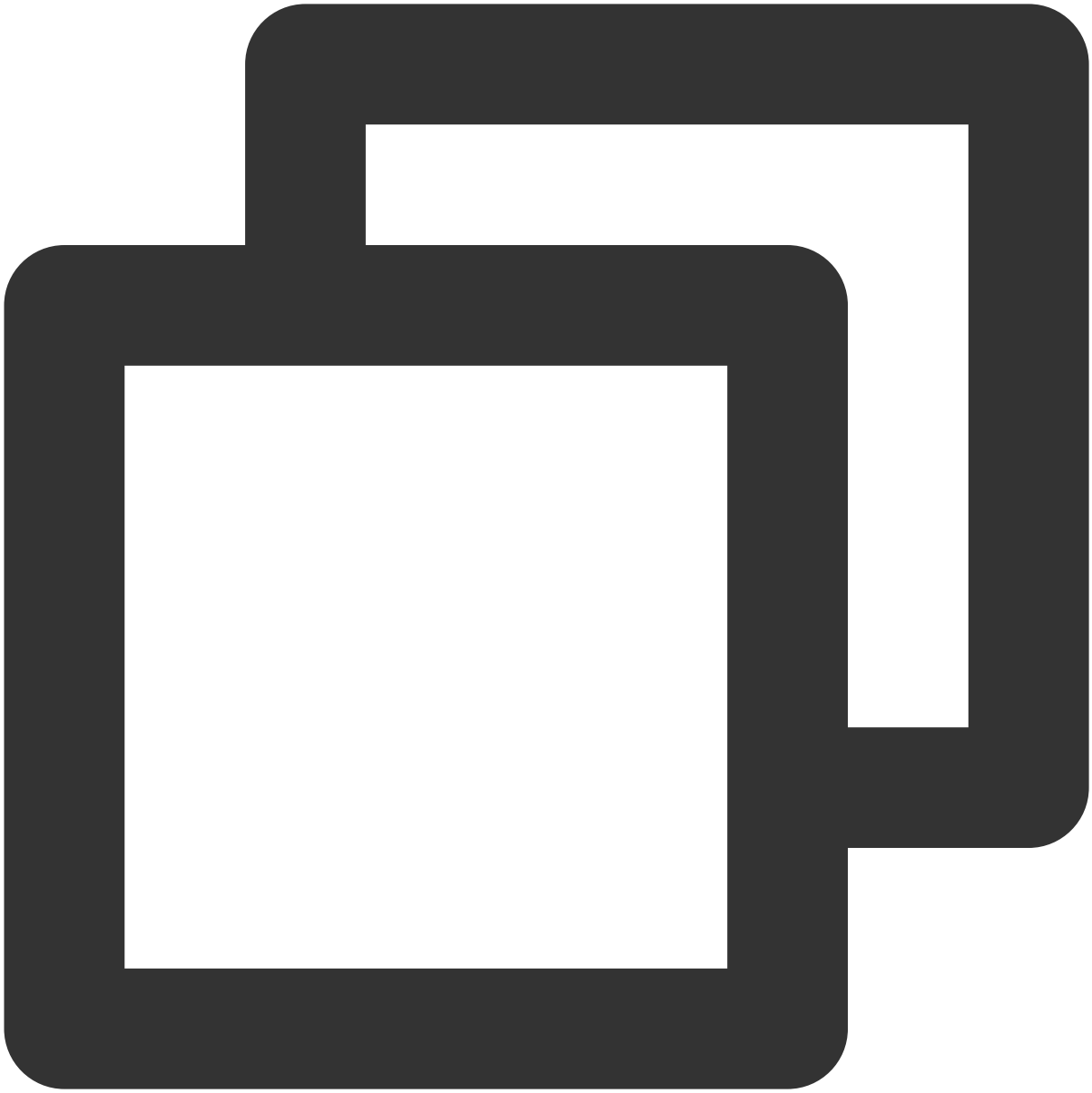
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x

<Response>
```

```
<RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
<MediaWorkflow>
    <Name>workflow-1</Name>
    <State>Active</State>
    <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
    <BucketId>test-1234567890</BucketId>
    <CreateTime>2022-07-14T12:37:28+0800</CreateTime>
    <UpdateTime>2022-07-14T12:37:28+0800</UpdateTime>
    <Topology>
        <Dependencies>
            <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
            <Snapshot_1581665960536>End</Snapshot_1581665960536>
            <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
            <Segment_15816659605667>End</Segment_15816659605667>
            <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
            <PicProcess_15816659605668>End</PicProcess_15816659605668>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                    <ObjectPrefix>input/workflow-1</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <Snapshot_1581665960536>
                <Type>Snapshot</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
```

```
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                    <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                </Output>
            </Operation>
        </Snapshot_1581665960536>
        <Transcode_1581665960538>
            <Type>Transcode</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>bcd/${RunId}/trans.{Ext}</Object>
                </Output>
            </Operation>
        </Transcode_1581665960538>
        <Segment_15816659605667>
            <Type>Segment</Type>
            <Operation>
                <Segment>
                    <Format>mkv</Format>
                    <Duration>20</Duration>
                </Segment>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>test-trans${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </Segment_15816659605667>
        <SmartCover_1581665960539>
            <Type>SmartCover</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </SmartCover_1581665960539>
        <PicProcess_15816659605668>
            <Type>PicProcess</Type>
            <Operation>
                <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                <Output>
```

```
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Response>
```

自适应码流

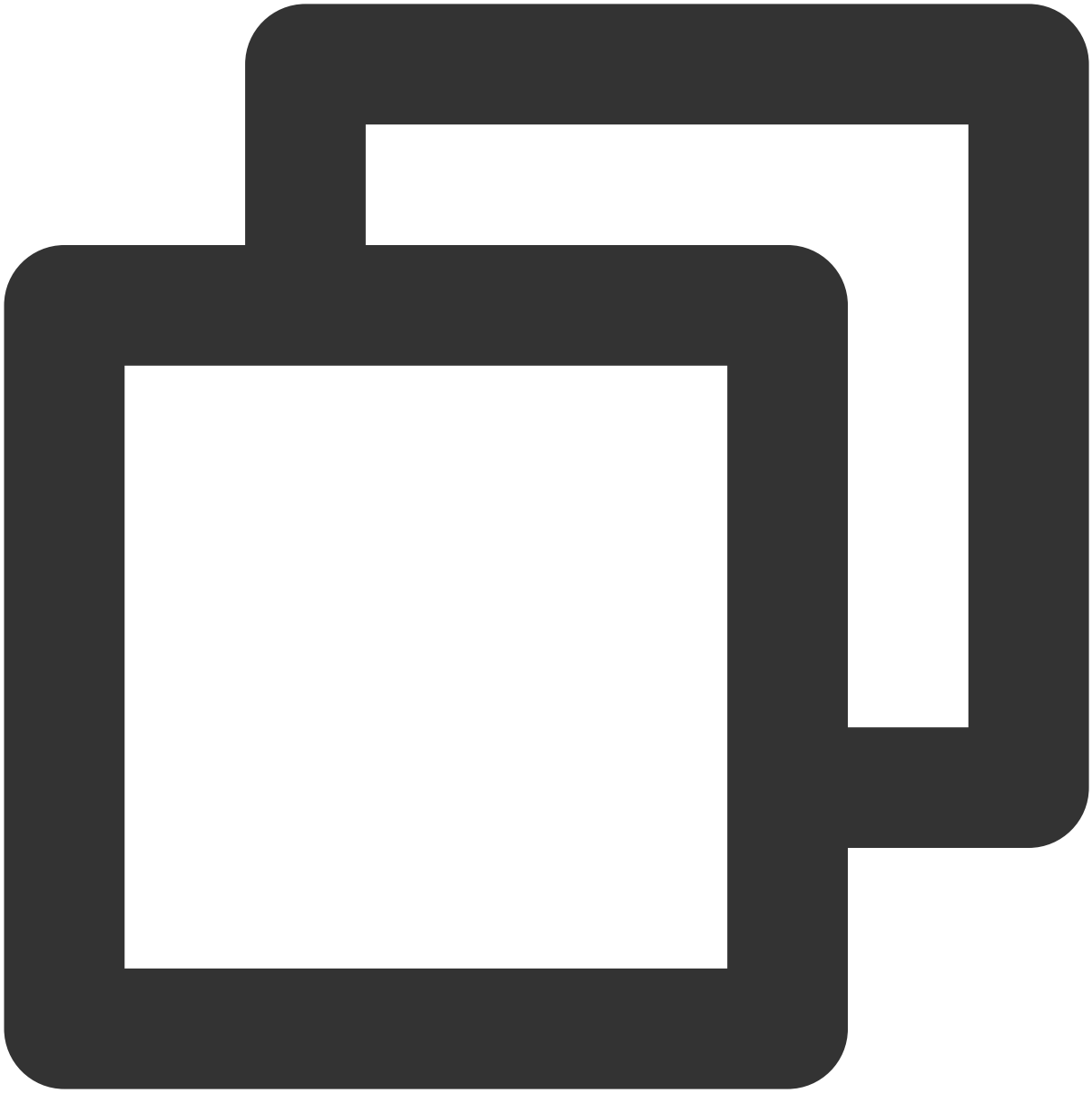


```
POST /workflow HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <MediaWorkflow>
        <Name>workflow-2</Name>
        <State>Active</State>
        <Topology>
```

```
        <Dependencies>
            <Start>StreamPackConfig_1581665960532</Start>
            <StreamPackConfig_1581665960532>VideoStream_1581665960536,VideoStre
            <VideoStream_1581665960536>StreamPack_1581665960538</VideoStream_15
            <VideoStream_1581665960537>StreamPack_1581665960538</VideoStream_15
            <StreamPack_1581665960538>End</StreamPack_1581665960538>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <ObjectPrefix>input/workflow-2</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <StreamPackConfig_1581665960532>
                <Type>StreamPackConfig</Type>
                <Operation>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>${InputPath}/${InputName}._${RunId}.${ext}</Obj
                    </Output>
                    <StreamPackConfig>
                        <PackType>HLS</PackType>
                        <IgnoreFailedStream>true</IgnoreFailedStream>
                    </StreamPackConfig>
                </Operation>
            </StreamPackConfig_1581665960532>
            <VideoStream_1581665960536>
                <Type>VideoStream</Type>
                <Operation>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_1/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960536>
                <VideoStream_1581665960537>
                    <Type>VideoStream</Type>
                    <Operation>
                        <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_2/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960537>
                <StreamPack_1581665960538>
                    <Type>StreamPack</Type>
                    <Operation>
                        <StreamPackInfo>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960536</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960537</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                        </StreamPackInfo>
                    </Operation>
                </StreamPack_1581665960538>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Request>
```



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf95s=

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf95s=</RequestId>
    <MediaWorkflow>
```

```
        <Name>workflow-2</Name>
        <State>Active</State>
        <WorkflowId>w93aa43ba105347169fa093ed857b2a90</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-07-14T12:37:28+0800</CreateTime>
        <UpdateTime>2022-07-14T12:37:28+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>StreamPackConfig_1581665960532</Start>
                <StreamPackConfig_1581665960532>VideoStream_1581665960536,VideoStre
                <VideoStream_1581665960536>StreamPack_1581665960538</VideoStream_15
                <VideoStream_1581665960537>StreamPack_1581665960538</VideoStream_15
                <StreamPack_1581665960538>End</StreamPack_1581665960538>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <ObjectPrefix>input/workflow-2</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <StreamPackConfig_1581665960532>
                    <Type>StreamPackConfig</Type>
                    <Operation>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${InputPath}/${InputName}._${RunId}.${ext}</Obj
                        </Output>
                        <StreamPackConfig>
                            <PackType>HLS</PackType>
```

```
                        <IgnoreFailedStream>true</IgnoreFailedStream>
                    </StreamPackConfig>
                </Operation>
            </StreamPackConfig_1581665960532>
            <VideoStream_1581665960536>
                <Type>VideoStream</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>${RunId}_Substream_1/video.m3u8</Object>
                    </Output>
                </Operation>
            </VideoStream_1581665960536>
            <VideoStream_1581665960537>
                <Type>VideoStream</Type>
                <Operation>
                    <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>${RunId}_Substream_2/video.m3u8</Object>
                    </Output>
                </Operation>
            </VideoStream_1581665960537>
            <StreamPack_1581665960538>
                <Type>StreamPack</Type>
                <Operation>
                    <StreamPackInfo>
                        <VideoStreamConfig>
                            <VideoStreamName>VideoStream_1581665960536</VideoSt
                            <BandWidth>200000000</BandWidth>
                        </VideoStreamConfig>
                        <VideoStreamConfig>
                            <VideoStreamName>VideoStream_1581665960537</VideoSt
                            <BandWidth>200000000</BandWidth>
                        </VideoStreamConfig>
                    </StreamPackInfo>
                </Operation>
            </StreamPack_1581665960538>
        </Nodes>
    </Topology>
</MediaWorkflow>
</Response>
```

查询工作流

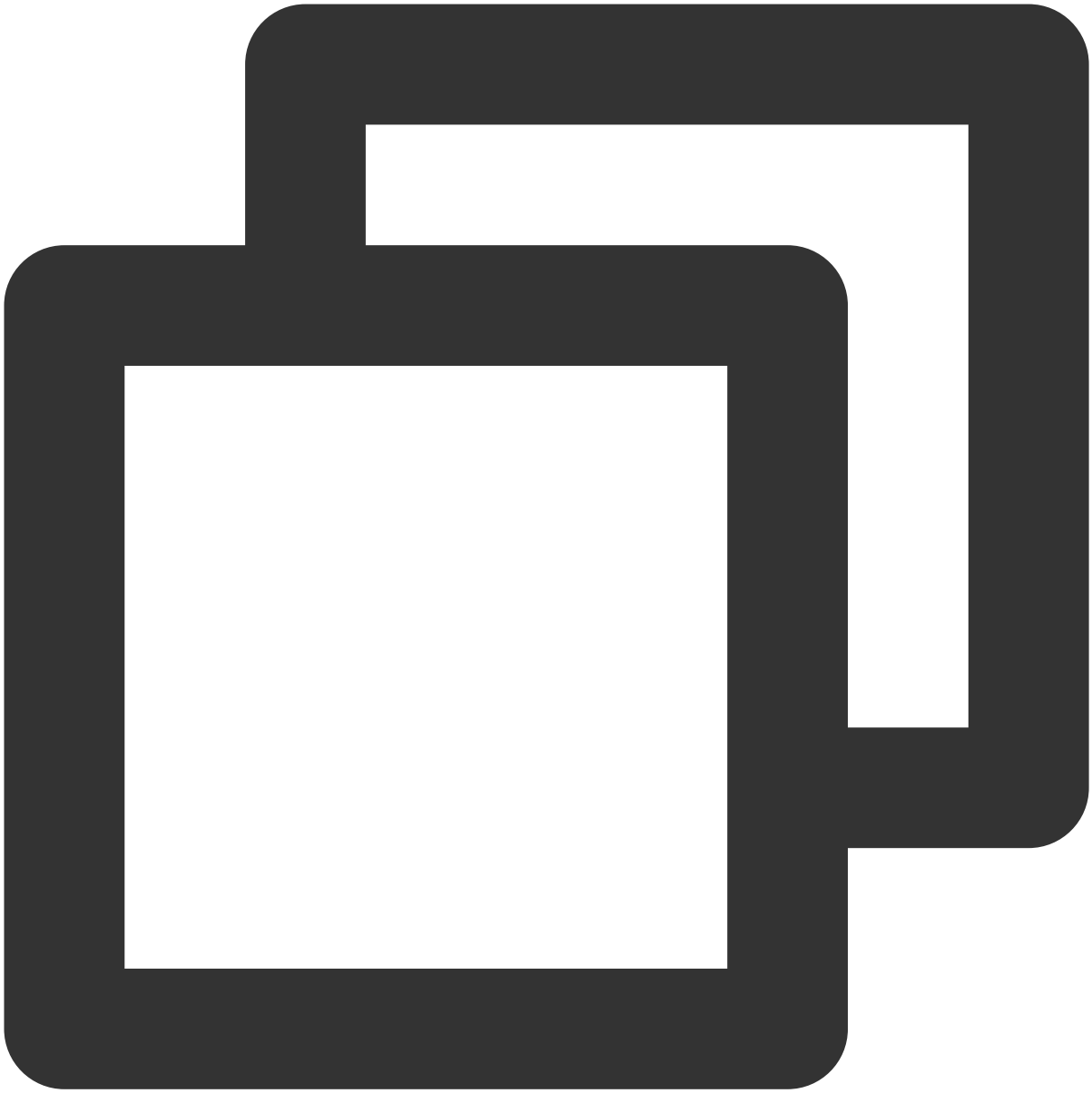
最近更新时间：2023-12-26 17:08:42

功能描述

查询工作流。

请求

请求示例



```
GET /workflow HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

参数名称（关键字）	描述	类型	是否必选
ids	工作流 ID，以 `,` 符号分割字符串	string	否
name	工作流名称	string	否
pageNumber	第几页	string	否
pageSize	每页个数	string	否

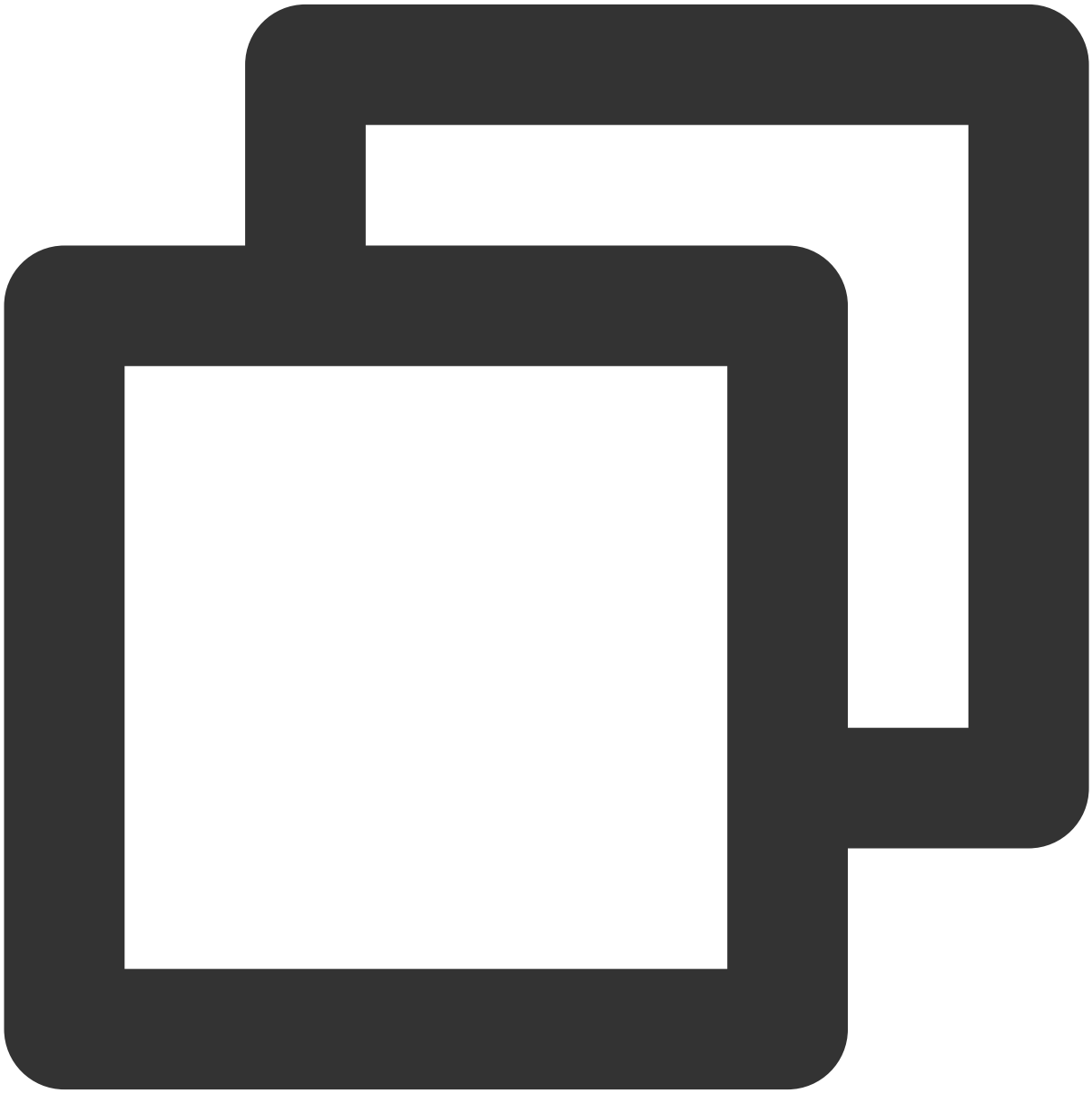
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMWZiZTFfOTBmYTUwNjRfNjVmNl8x</RequestId>
    <TotalCount>2</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <MediaWorkflowList>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
```

```
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                        <ObjectPrefix>input/workflow-1</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish,WorkflowStart</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                            <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                        </Output>
                    </Operation>
                </Snapshot_1581665960536>
                <Transcode_1581665960538>
```

```
                <Type>Transcode</Type>
                <Operation>
                    <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>bcd/${RunId}/trans.{Ext}</Object>
                    </Output>
                </Operation>
            </Transcode_1581665960538>
            <Segment_15816659605667>
                <Type>Segment</Type>
                <Operation>
                    <Segment>
                        <Format>mkv</Format>
                        <Duration>20</Duration>
                    </Segment>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>test-trans${Number}.{Ext}</Object>
                    </Output>
                </Operation>
            </Segment_15816659605667>
            <SmartCover_1581665960539>
                <Type>SmartCover</Type>
                <Operation>
                    <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                    </Output>
                </Operation>
            </SmartCover_1581665960539>
            <PicProcess_15816659605668>
                <Type>PicProcess</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>bcd/${RunId}/pic.{Ext}</Object>
                    </Output>
                </Operation>
            </PicProcess_15816659605668>
        </Nodes>
```

```
        </Topology>
    </MediaWorkflowList>
    <MediaWorkflowList>
        <Name>workflow-2</Name>
        <State>Active</State>
        <WorkflowId>w93aa43ba105347169fa093ed857b2a90</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>StreamPackConfig_1581665960532</Start>
                <StreamPackConfig_1581665960532>VideoStream_1581665960536,VideoStre
                <VideoStream_1581665960536>StreamPack_1581665960538</VideoStream_15
                <VideoStream_1581665960537>StreamPack_1581665960538</VideoStream_15
                <StreamPack_1581665960538>End</StreamPack_1581665960538>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <ObjectPrefix>input/workflow-2</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish,WorkflowStart</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <StreamPackConfig_1581665960532>
                    <Type>StreamPackConfig</Type>
                    <Operation>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${InputPath}/${InputName}._${RunId}.${ext}</Obj
```

```
                    </Output>
                    <StreamPackConfig>
                        <PackType>HLS</PackType>
                        <IgnoreFailedStream>true</IgnoreFailedStream>
                    </StreamPackConfig>
                </Operation>
            </StreamPackConfig_1581665960532>
            <VideoStream_1581665960536>
                <Type>VideoStream</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>${RunId}_Substream_1/video.m3u8</Object>
                    </Output>
                </Operation>
            </VideoStream_1581665960536>
            <VideoStream_1581665960537>
                <Type>VideoStream</Type>
                <Operation>
                    <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>${RunId}_Substream_2/video.m3u8</Object>
                    </Output>
                </Operation>
            </VideoStream_1581665960537>
            <StreamPack_1581665960538>
                <Type>StreamPack</Type>
                <Operation>
                    <StreamPackInfo>
                        <VideoStreamConfig>
                            <VideoStreamName>VideoStream_1581665960536</VideoSt
                            <BandWidth>200000000</BandWidth>
                        </VideoStreamConfig>
                        <VideoStreamConfig>
                            <VideoStreamName>VideoStream_1581665960537</VideoSt
                            <BandWidth>200000000</BandWidth>
                        </VideoStreamConfig>
                    </StreamPackInfo>
                </Operation>
            </StreamPack_1581665960538>
        </Nodes>
    </Topology>
</MediaWorkflowList>
```

```
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	工作流总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
MediaWorkflowList	Response	工作流数组	Container

Container节点 MediaWorkflowList 的内容：

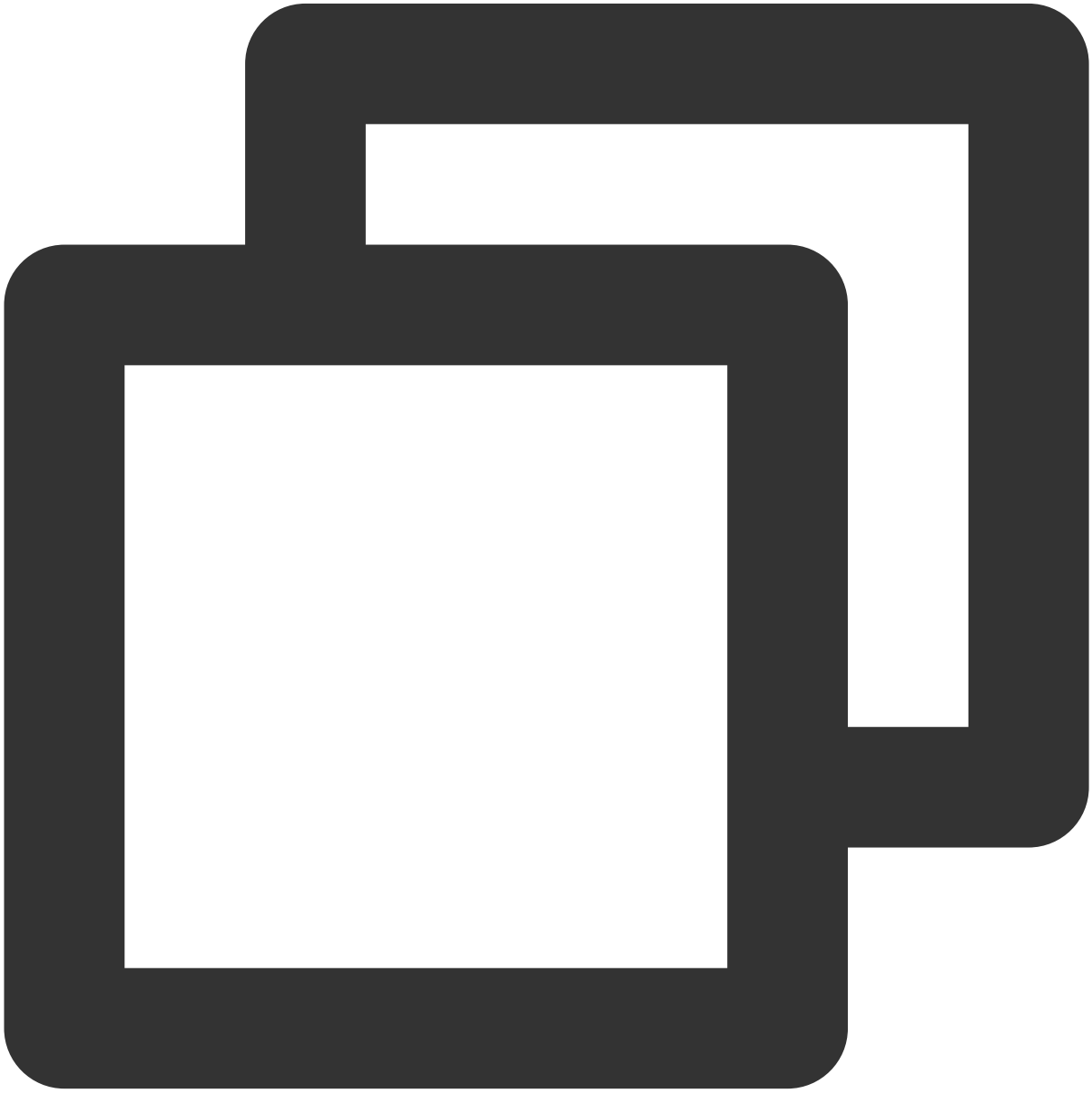
节点名称（关键字）	父节点	描述	类型	是否必选	限制
Name	Response.MediaWorkflowList	工作流名称	String	是	支持中文、英文、数字、—和_，长度限制128字符
WorkflowId	Response.MediaWorkflowList	工作流 ID	String	是	工作流唯一 ID
State	Response.MediaWorkflowList	工作流状态	String	是	1. Active 2. Paused
CreateTime	Response.MediaWorkflowList	创建时间	String	是	无
UpdateTime	Response.MediaWorkflowList	更新时间	String	是	无
Topology	Response.MediaWorkflowList	拓扑信息	Container	是	同创建工作流接口的 Topology

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

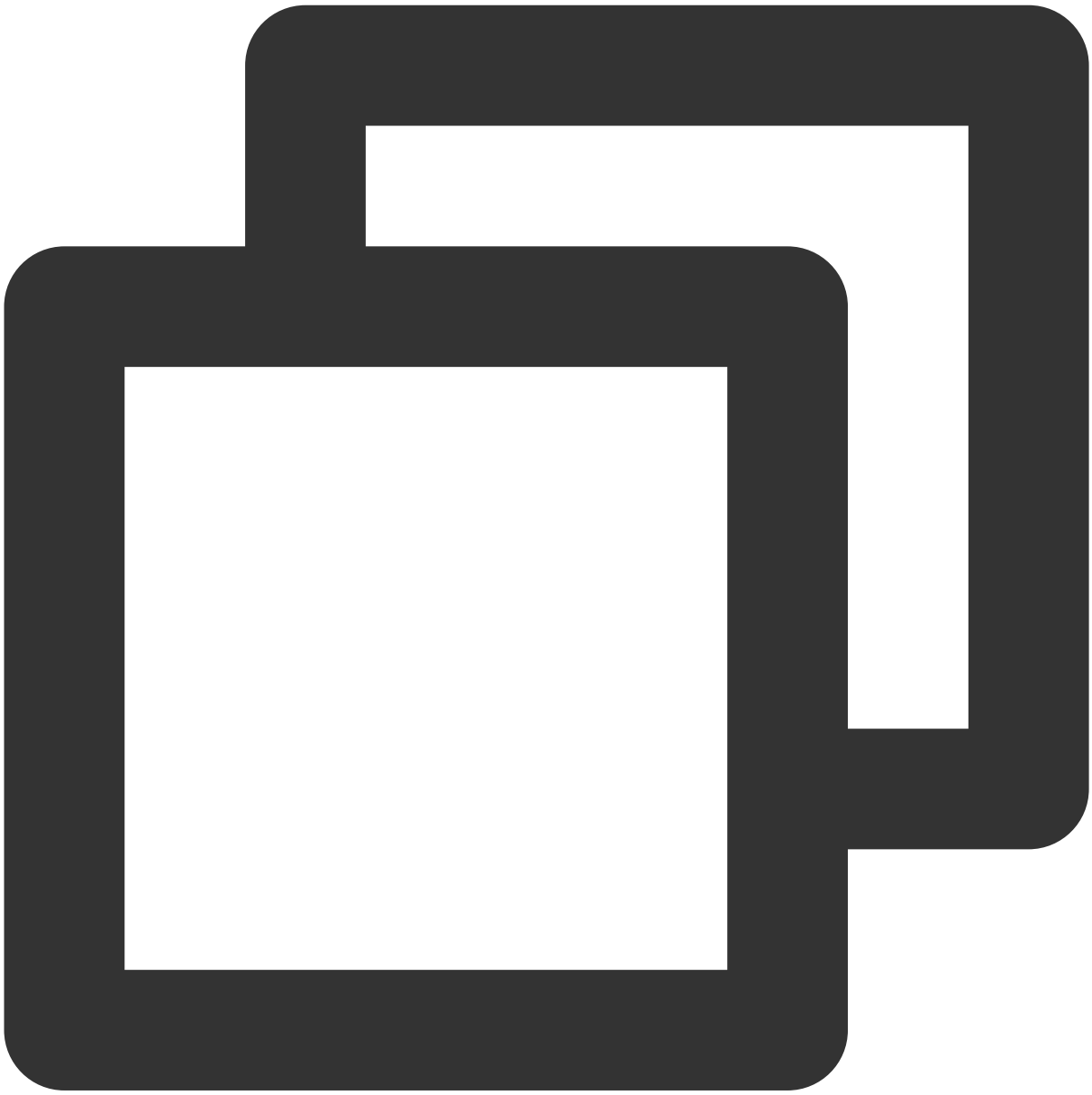
请求1：工作流 ID



```
GET /workflow?ids=wc666d0b9f9dd47ae9137a096252d49f7,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
```

```
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应1



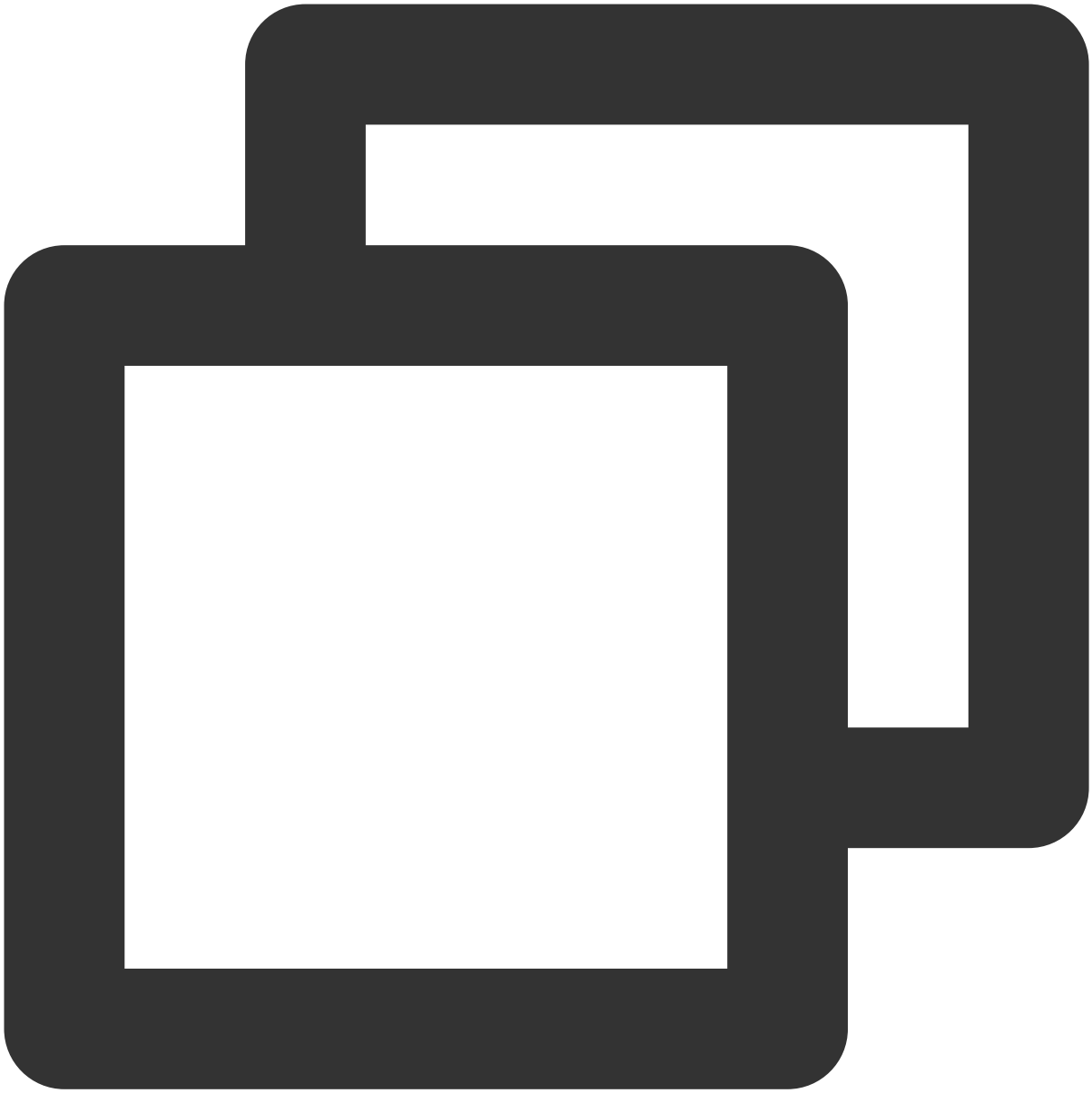
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
```

```
Server: tencent-ci
x-ci-request-id: NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x

<Response>
    <RequestId>NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x</RequestId>
        <MediaWorkflowList>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                        <ObjectPrefix>input/workflow-1</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish,WorkflowStart</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
```

```
            <Operation>
                <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                    <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                </Output>
            </Operation>
        </Snapshot_1581665960536>
        <Transcode_1581665960538>
            <Type>Transcode</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>bcd/${RunId}/trans.{Ext}</Object>
                </Output>
            </Operation>
        </Transcode_1581665960538>
        <Segment_15816659605667>
            <Type>Segment</Type>
            <Operation>
                <Segment>
                    <Format>mkv</Format>
                    <Duration>20</Duration>
                </Segment>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>test-trans${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </Segment_15816659605667>
        <SmartCover_1581665960539>
            <Type>SmartCover</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </SmartCover_1581665960539>
        <PicProcess_15816659605668>
```

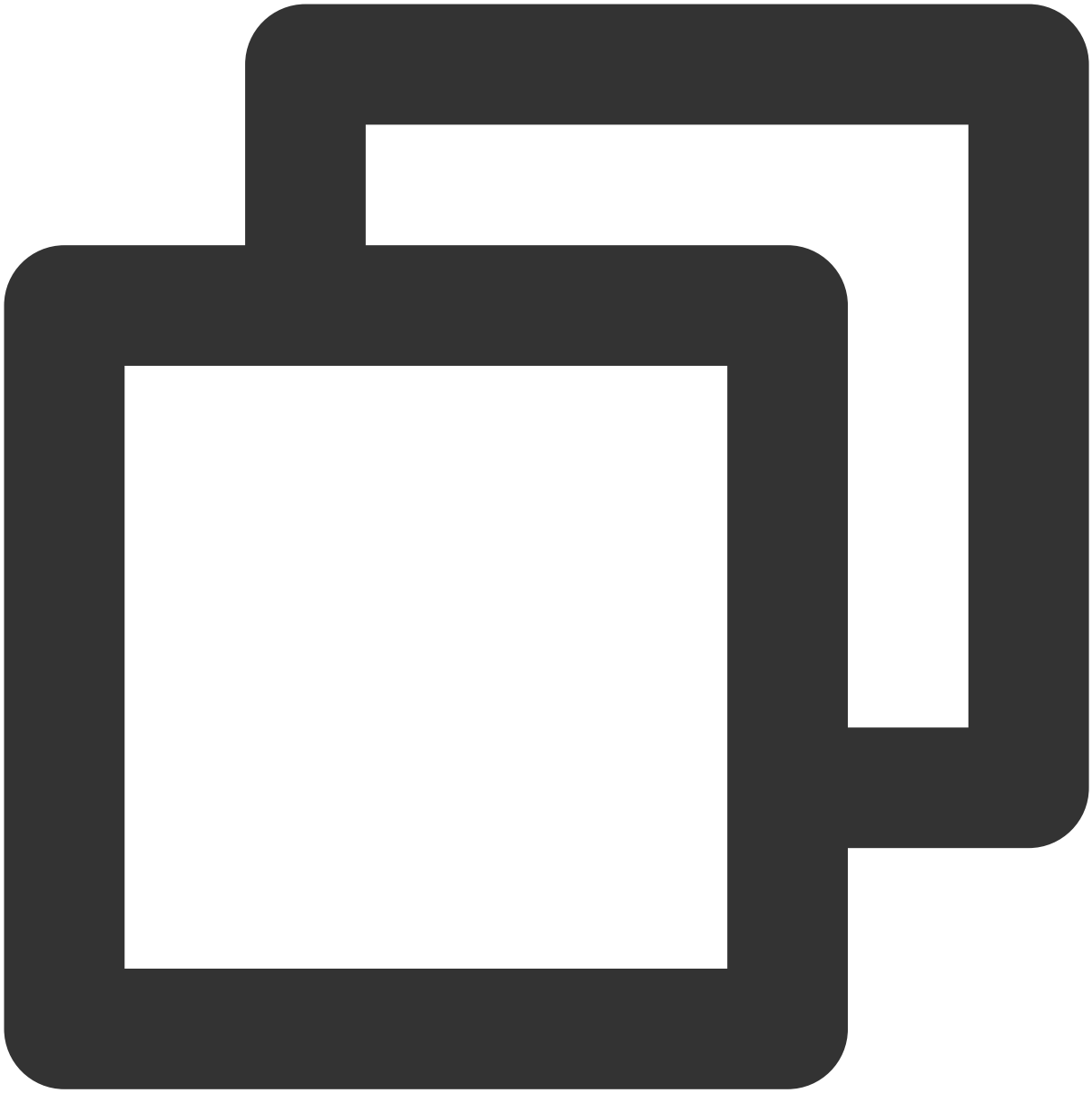
```
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflowList>
    <NonExistIDs>A</NonExistIDs>
    <NonExistIDs>B</NonExistIDs>
</Response>
```

请求2：工作流列表



```
GET /workflow?pageNumber=1&pageSize=10 HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应2



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TotalCount>2</TotalCount>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <MediaWorkflowList>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                        <ObjectPrefix>input/workflow-1</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish,WorkflowStart</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
```

```
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                    <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                </Output>
            </Operation>
        </Snapshot_1581665960536>
        <Transcode_1581665960538>
            <Type>Transcode</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>bcd/${RunId}/trans.{Ext}</Object>
                </Output>
            </Operation>
        </Transcode_1581665960538>
        <Segment_15816659605667>
            <Type>Segment</Type>
            <Operation>
                <Segment>
                    <Format>mkv</Format>
                    <Duration>20</Duration>
                </Segment>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>test-trans${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </Segment_15816659605667>
        <SmartCover_1581665960539>
            <Type>SmartCover</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </SmartCover_1581665960539>
        <PicProcess_15816659605668>
            <Type>PicProcess</Type>
            <Operation>
                <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflowList>
    <MediaWorkflowList>
        <Name>workflow-2</Name>
        <State>Active</State>
        <WorkflowId>w93aa43ba105347169fa093ed857b2a90</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>StreamPackConfig_1581665960532</Start>
                <StreamPackConfig_1581665960532>VideoStream_1581665960536,VideoStre
                <VideoStream_1581665960536>StreamPack_1581665960538</VideoStream_15
                <VideoStream_1581665960537>StreamPack_1581665960538</VideoStream_15
                <StreamPack_1581665960538>End</StreamPack_1581665960538>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <ObjectPrefix>input/workflow-2</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish,WorkflowStart</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
```

```
                </Start>
                <StreamPackConfig_1581665960532>
                    <Type>StreamPackConfig</Type>
                    <Operation>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${InputPath}/${InputName}._${RunId}.${ext}</Obj
                        </Output>
                        <StreamPackConfig>
                            <PackType>HLS</PackType>
                            <IgnoreFailedStream>true</IgnoreFailedStream>
                        </StreamPackConfig>
                    </Operation>
                </StreamPackConfig_1581665960532>
                <VideoStream_1581665960536>
                    <Type>VideoStream</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_1/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960536>
                <VideoStream_1581665960537>
                    <Type>VideoStream</Type>
                    <Operation>
                        <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>${RunId}_Substream_2/video.m3u8</Object>
                        </Output>
                    </Operation>
                </VideoStream_1581665960537>
                <StreamPack_1581665960538>
                    <Type>StreamPack</Type>
                    <Operation>
                        <StreamPackInfo>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960536</VideoSt
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                            <VideoStreamConfig>
                                <VideoStreamName>VideoStream_1581665960537</VideoSt
```

```
                                <BandWidth>200000000</BandWidth>
                            </VideoStreamConfig>
                        </StreamPackInfo>
                    </Operation>
                </StreamPack_1581665960538>
            </Nodes>
        </Topology>
    </MediaWorkflowList>
</Response>
```

删除工作流

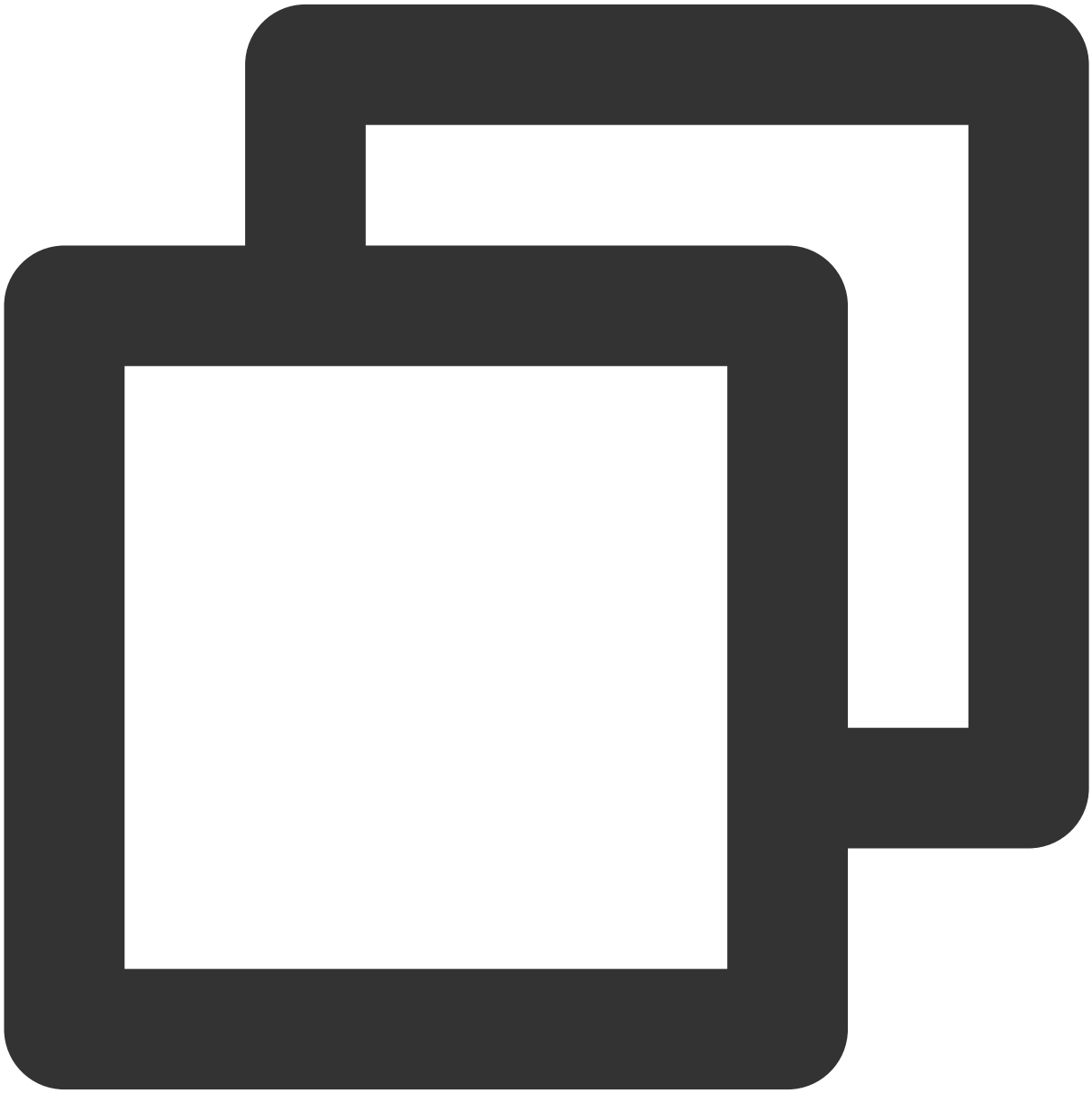
最近更新时间：2023-12-26 17:08:52

功能描述

删除工作流。

请求

请求示例



```
DELETE /workflow/<WorkflowId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

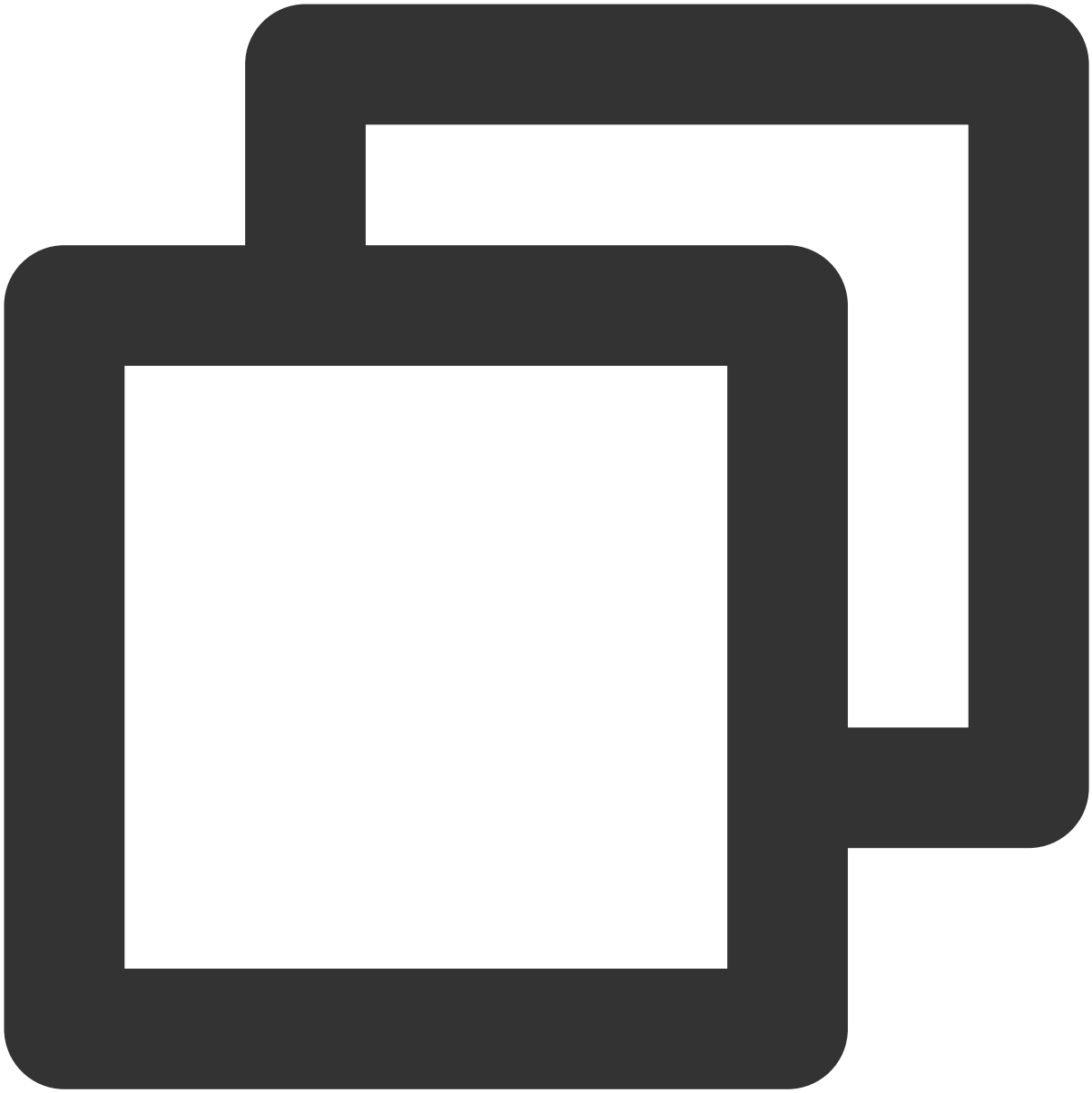
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x</RequestId>
    <WorkflowId>A</WorkflowId>
<Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

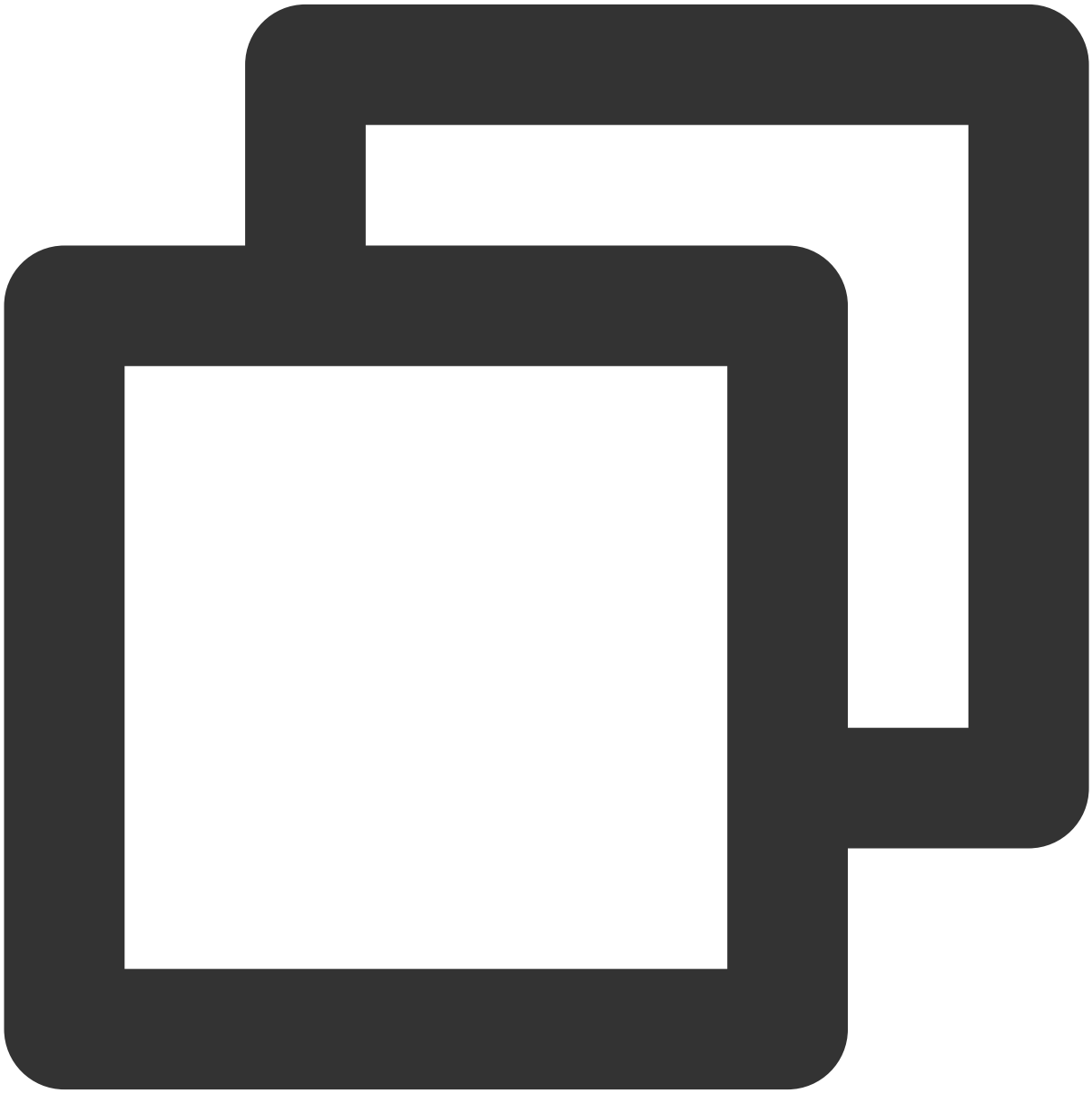
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
WorkflowId	Response	被删除的工作流 ID	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

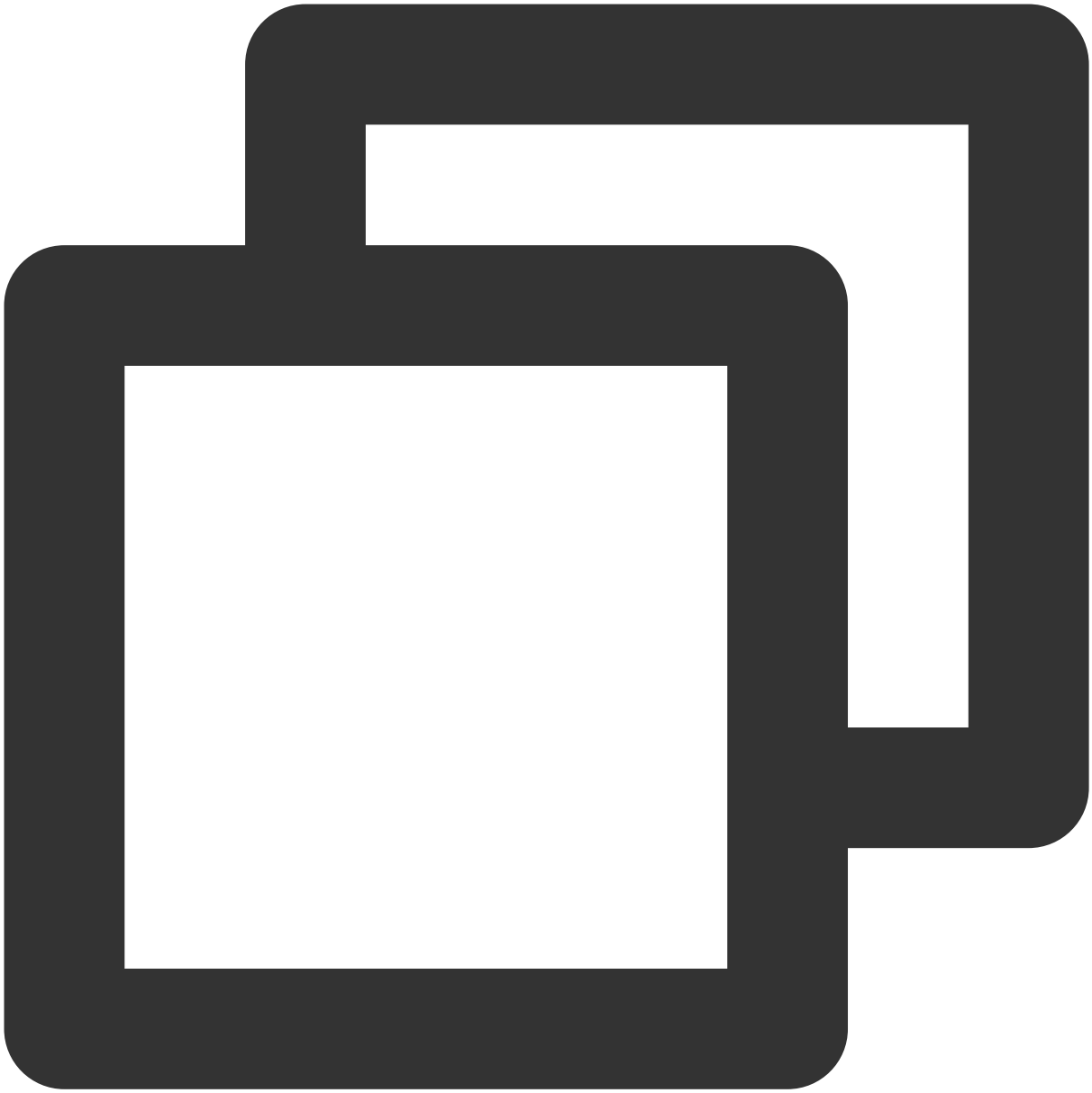
实际案例

请求



```
DELETE /workflow/we59a9648e62e48ffb25e4b41f3721799 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x

<Response>
    <RequestId>NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x</RequestId>
    <WorkflowId>we59a9648e62e48ffb25e4b41f3721799</WorkflowId>
```

```
<Response>
```

更新工作流

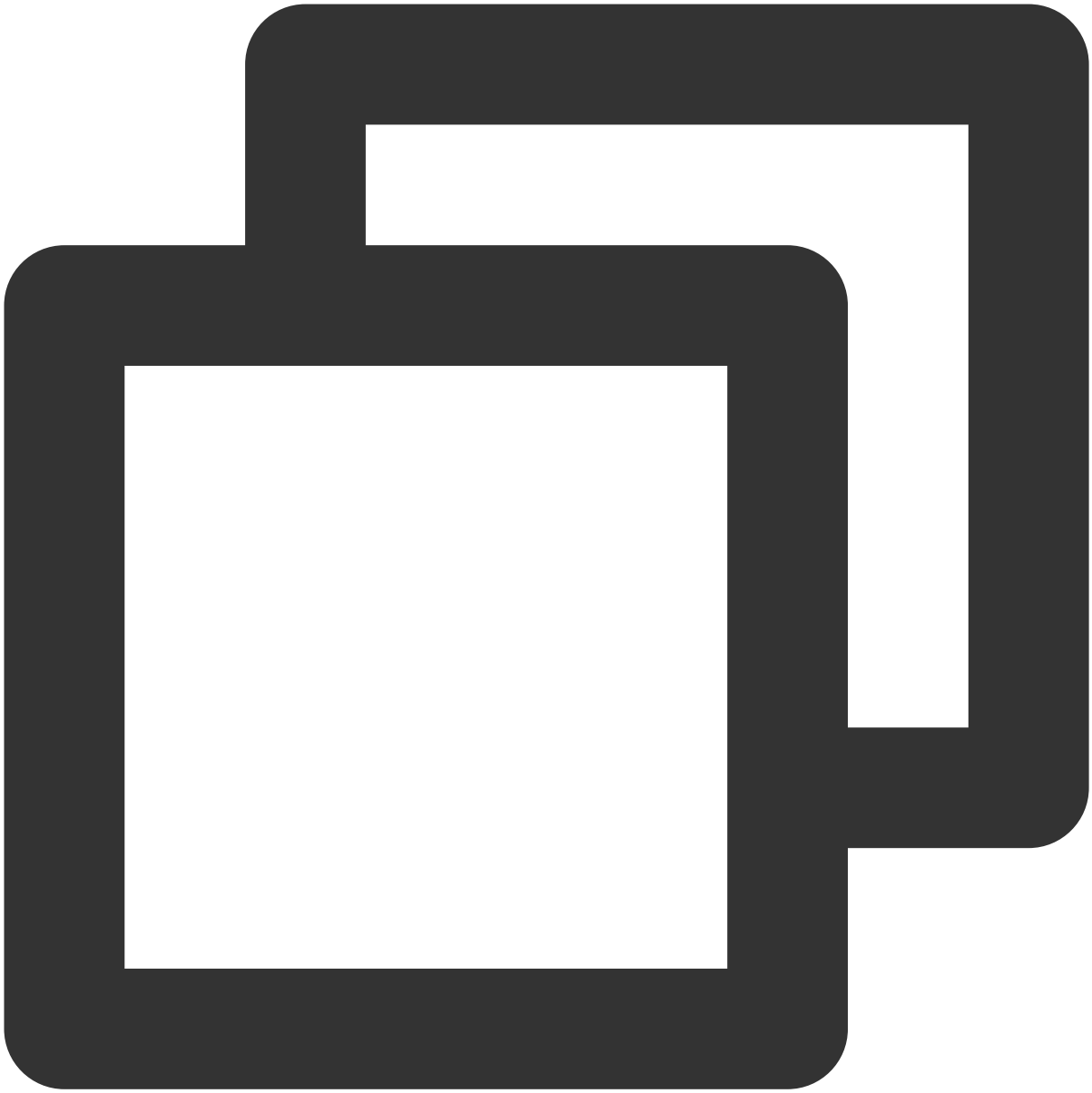
最近更新时间：2023-12-26 17:09:18

功能描述

更新工作流。

请求

请求示例



```
PUT /workflow/<WorkflowId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

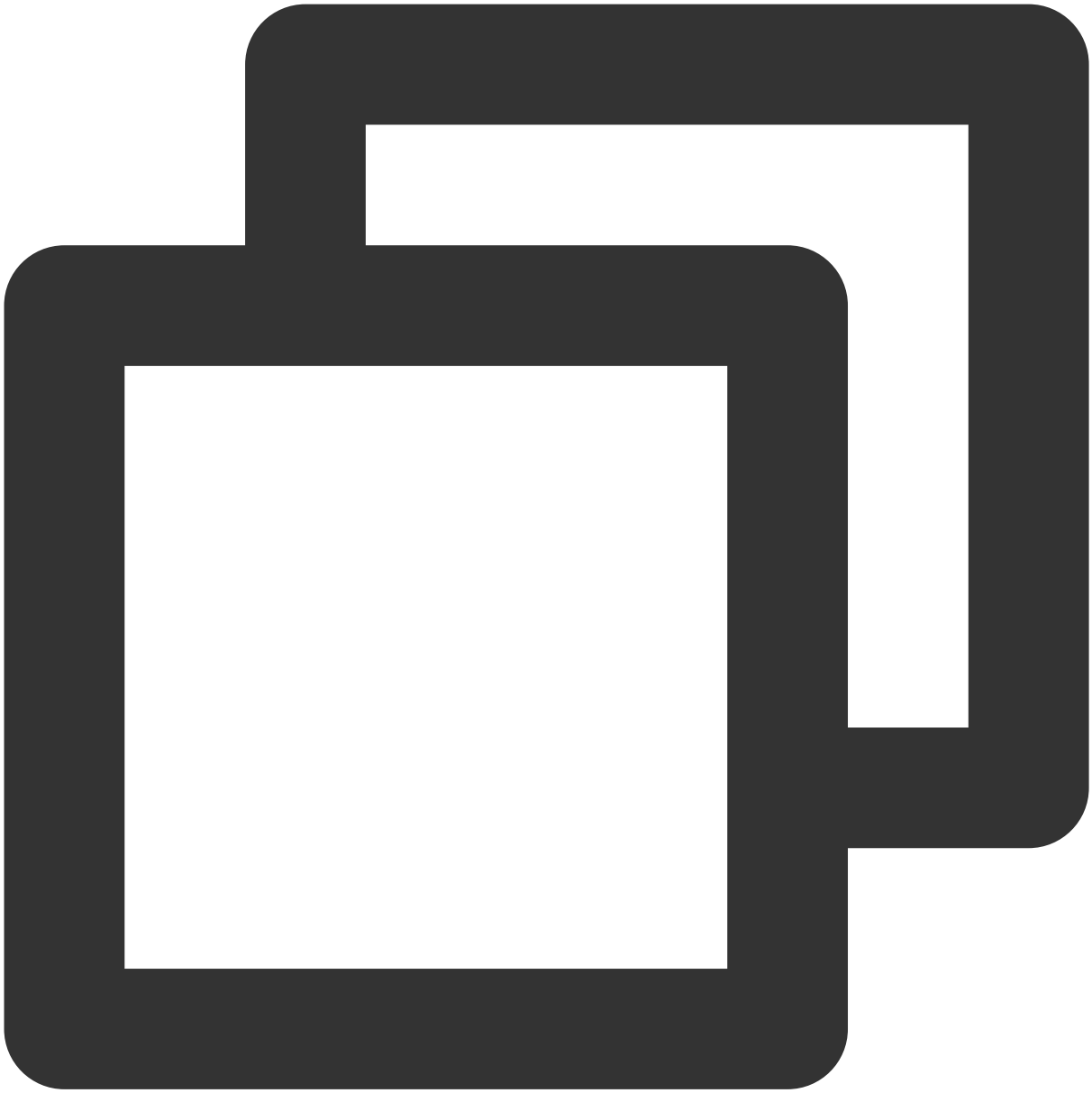
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体:



```
<Request>
```

```
<MediaWorkflow>
    <Name>workflow-1</Name>
    <State>Active</State>
    <Topology>
        <Dependencies>
            <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
            <Snapshot_1581665960536>End</Snapshot_1581665960536>
            <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
            <Segment_15816659605667>End</Segment_15816659605667>
            <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
            <PicProcess_15816659605668>End</PicProcess_15816659605668>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                    <ObjectPrefix>input/workflow-1</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <Snapshot_1581665960536>
                <Type>Snapshot</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                        <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                    </Output>
                </Operation>
```

```
                </Snapshot_1581665960536>
                <Transcode_1581665960538>
                    <Type>Transcode</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/trans.{Ext}</Object>
                        </Output>
                    </Operation>
                </Transcode_1581665960538>
                <Segment_15816659605667>
                    <Type>Segment</Type>
                    <Operation>
                        <Segment>
                            <Format>mkv</Format>
                            <Duration>25</Duration>
                        </Segment>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>test-trans${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </Segment_15816659605667>
                <SmartCover_1581665960539>
                    <Type>SmartCover</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </SmartCover_1581665960539>
                <PicProcess_15816659605668>
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
```

```
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建工作流接口的 Request	Container	是

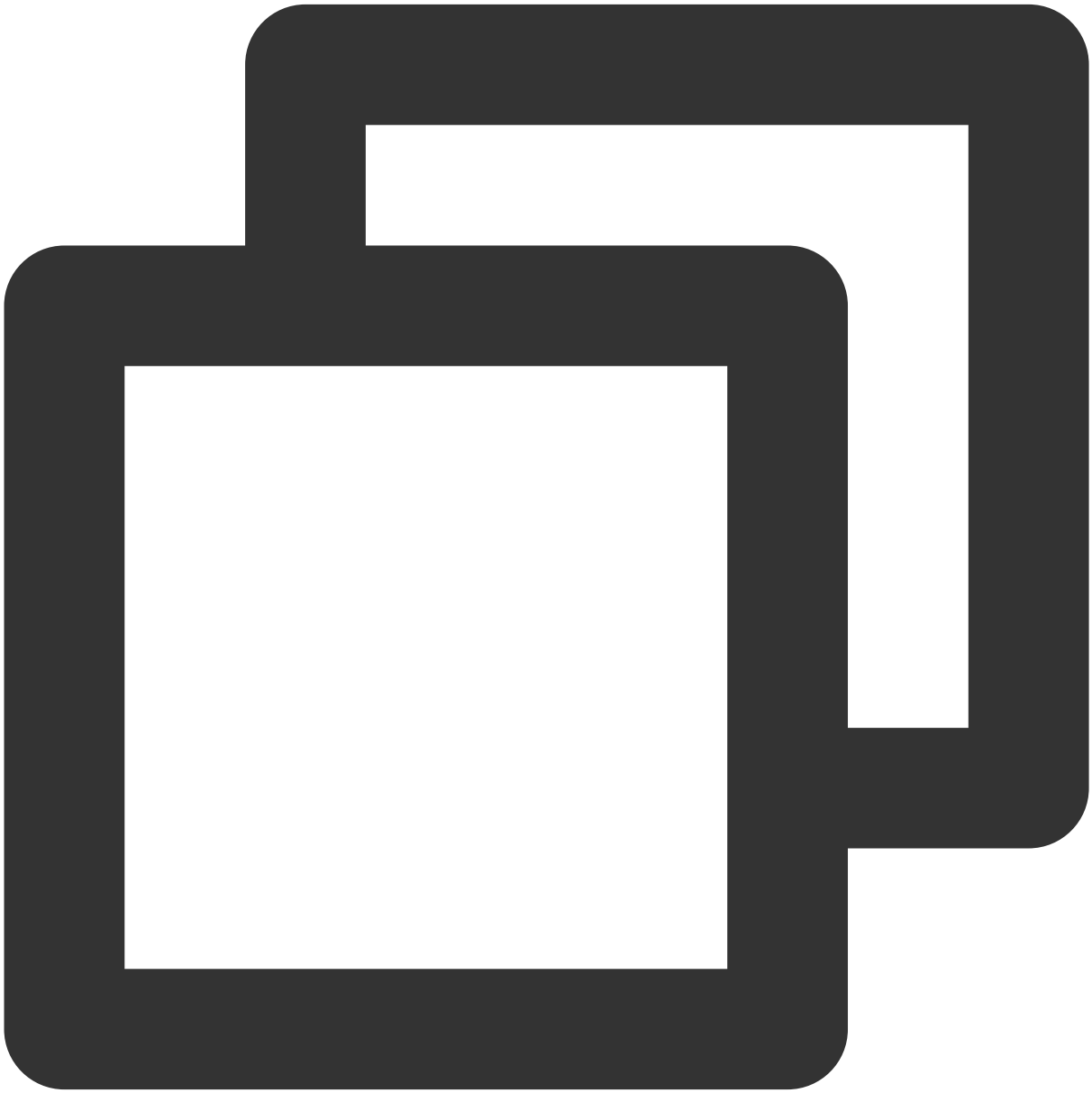
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-07-14T12:37:28+0800</CreateTime>
        <UpdateTime>2022-07-14T12:37:28+0800</UpdateTime>
        <Topology>
            <Dependencies>
```

```
            <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
            <Snapshot_1581665960536>End</Snapshot_1581665960536>
            <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
            <Segment_15816659605667>End</Segment_15816659605667>
            <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
            <PicProcess_15816659605668>End</PicProcess_15816659605668>
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                    <ObjectPrefix>input/workflow-1</ObjectPrefix>
                    <NotifyConfig>
                        <State>On</State>
                        <Url>http://www.callback.com</Url>
                        <Event>TaskFinish,WorkflowFinish</Event>
                        <Type>Url</Type>
                        <ResultFormat>JSON</ResultFormat>
                    </NotifyConfig>
                    <ExtFilter>
                        <State>On</State>
                        <Video>true</Video>
                        <Audio>false</Audio>
                        <Image>false</Image>
                        <Custom>false</Custom>
                        <AllFile>false</AllFile>
                    </ExtFilter>
                </Input>
            </Start>
            <Snapshot_1581665960536>
                <Type>Snapshot</Type>
                <Operation>
                    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                        <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                    </Output>
                </Operation>
            </Snapshot_1581665960536>
            <Transcode_1581665960538>
                <Type>Transcode</Type>
                <Operation>
                    <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
```

```
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/trans.{Ext}</Object>
                        </Output>
                    </Operation>
                </Transcode_1581665960538>
                <Segment_15816659605667>
                    <Type>Segment</Type>
                    <Operation>
                        <Segment>
                            <Format>mkv</Format>
                            <Duration>25</Duration>
                        </Segment>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>test-trans${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </Segment_15816659605667>
                <SmartCover_1581665960539>
                    <Type>SmartCover</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </SmartCover_1581665960539>
                <PicProcess_15816659605668>
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Response>
```

具体的数据内容如下：

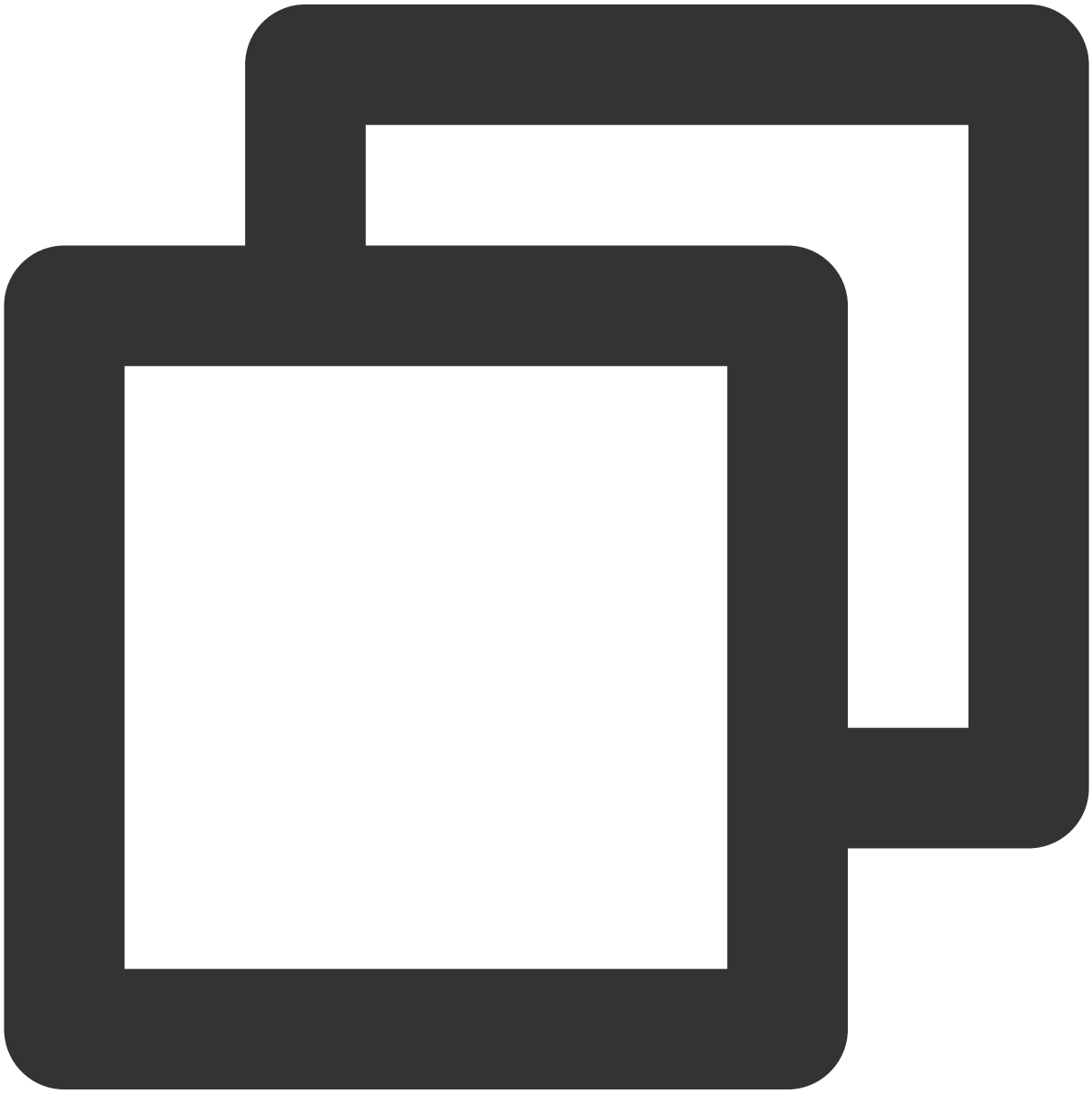
节点名称（关键字）	父节点	描述	类型	是否必选
Response	无	同创建工作流接口的 Response	Container	是

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求1：更新工作流配置



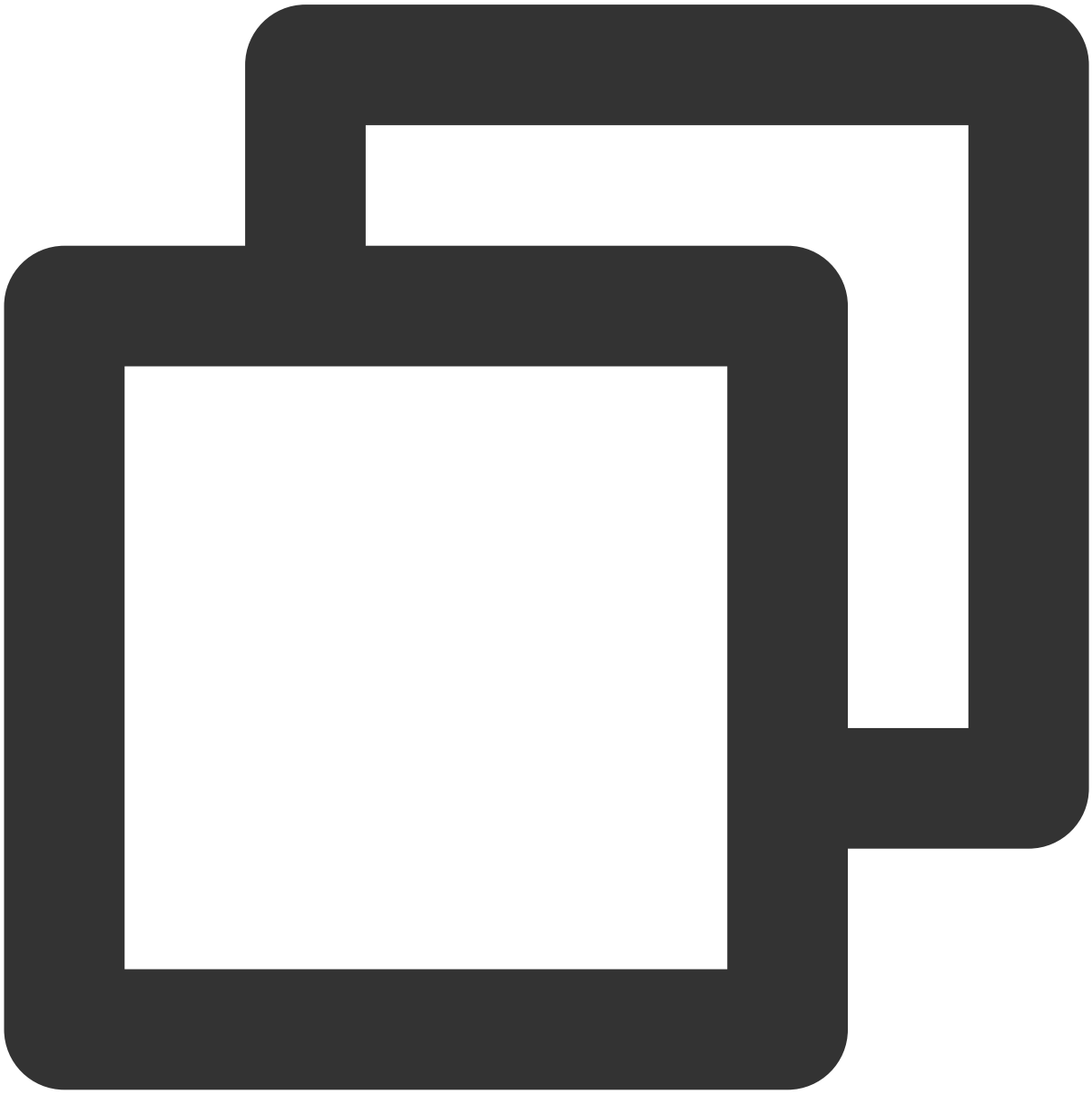
```
PUT /workflow/wc666d0b9f9dd47ae9137a096252d49f7 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <Topology>
```

```
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                        <ObjectPrefix>input/workflow-1</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                            <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                        </Output>
                    </Operation>
                </Snapshot_1581665960536>
                <Transcode_1581665960538>
                    <Type>Transcode</Type>
                    <Operation>
```

```
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/trans.{Ext}</Object>
                        </Output>
                    </Operation>
                </Transcode_1581665960538>
                <Segment_15816659605667>
                    <Type>Segment</Type>
                    <Operation>
                        <Segment>
                            <Format>mkv</Format>
                            <Duration>25</Duration>
                        </Segment>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>test-trans${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </Segment_15816659605667>
                <SmartCover_1581665960539>
                    <Type>SmartCover</Type>
                    <Operation>
                        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                        </Output>
                    </Operation>
                </SmartCover_1581665960539>
                <PicProcess_15816659605668>
                    <Type>PicProcess</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
```

```
</Request>
```

响应1



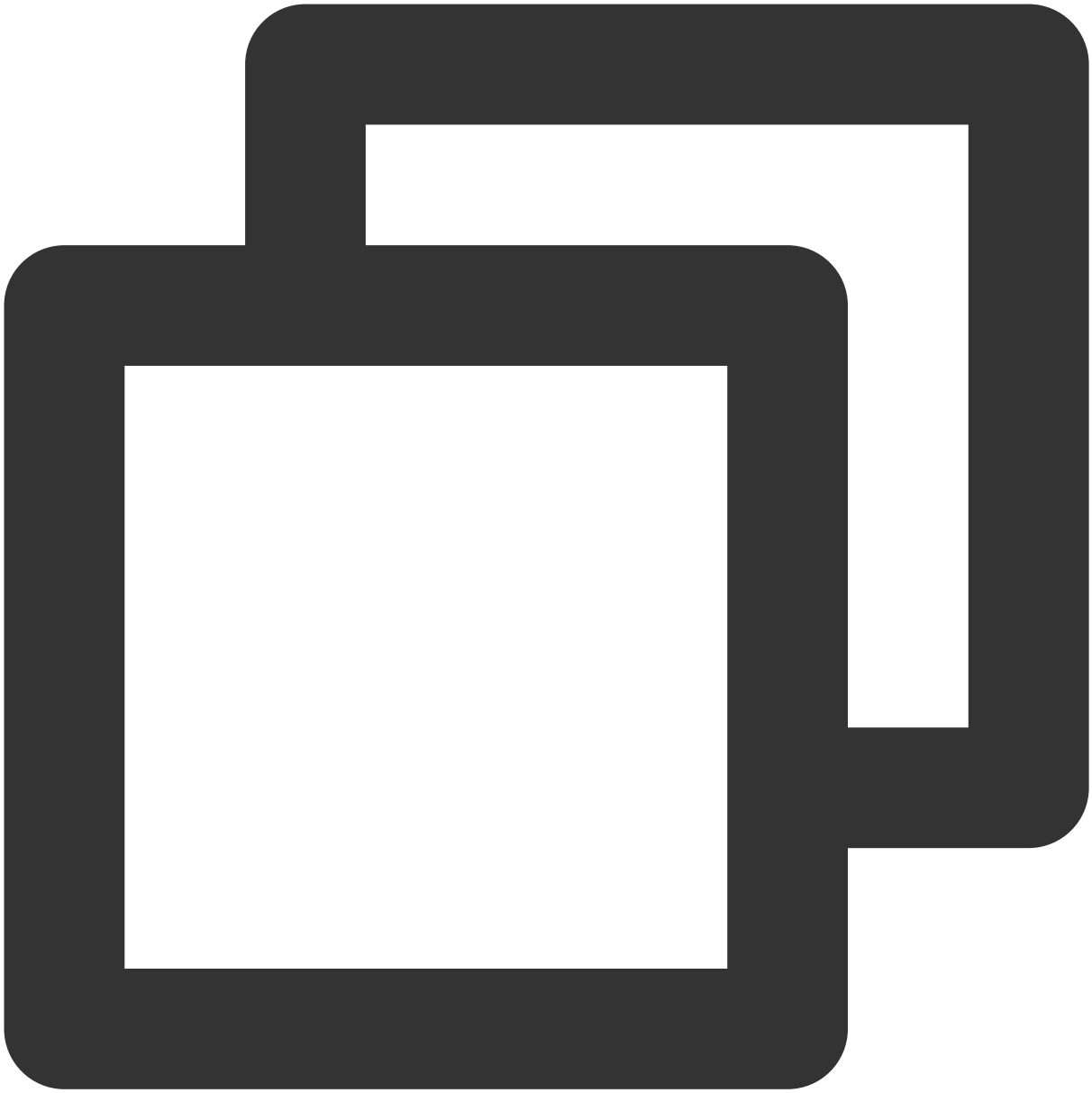
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x
```

```
<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-07-14T12:37:28+0800</CreateTime>
        <UpdateTime>2022-07-14T12:37:28+0800</UpdateTime>
        <Topology>
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                        <ObjectPrefix>input/workflow-1</ObjectPrefix>
                        <NotifyConfig>
                            <State>On</State>
                            <Url>http://www.callback.com</Url>
                            <Event>TaskFinish,WorkflowFinish</Event>
                            <Type>Url</Type>
                            <ResultFormat>JSON</ResultFormat>
                        </NotifyConfig>
                        <ExtFilter>
                            <State>On</State>
                            <Video>true</Video>
                            <Audio>false</Audio>
                            <Image>false</Image>
                            <Custom>false</Custom>
                            <AllFile>false</AllFile>
                        </ExtFilter>
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                    <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</Spr
                </Output>
            </Operation>
        </Snapshot_1581665960536>
        <Transcode_1581665960538>
            <Type>Transcode</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>bcd/${RunId}/trans.{Ext}</Object>
                </Output>
            </Operation>
        </Transcode_1581665960538>
        <Segment_15816659605667>
            <Type>Segment</Type>
            <Operation>
                <Segment>
                    <Format>mkv</Format>
                    <Duration>25</Duration>
                </Segment>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>test-trans${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </Segment_15816659605667>
        <SmartCover_1581665960539>
            <Type>SmartCover</Type>
            <Operation>
                <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
                <Output>
                    <Region>ap-chongqing</Region>
                    <Bucket>test-1234567890</Bucket>
                    <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
                </Output>
            </Operation>
        </SmartCover_1581665960539>
        <PicProcess_15816659605668>
            <Type>PicProcess</Type>
            <Operation>
```

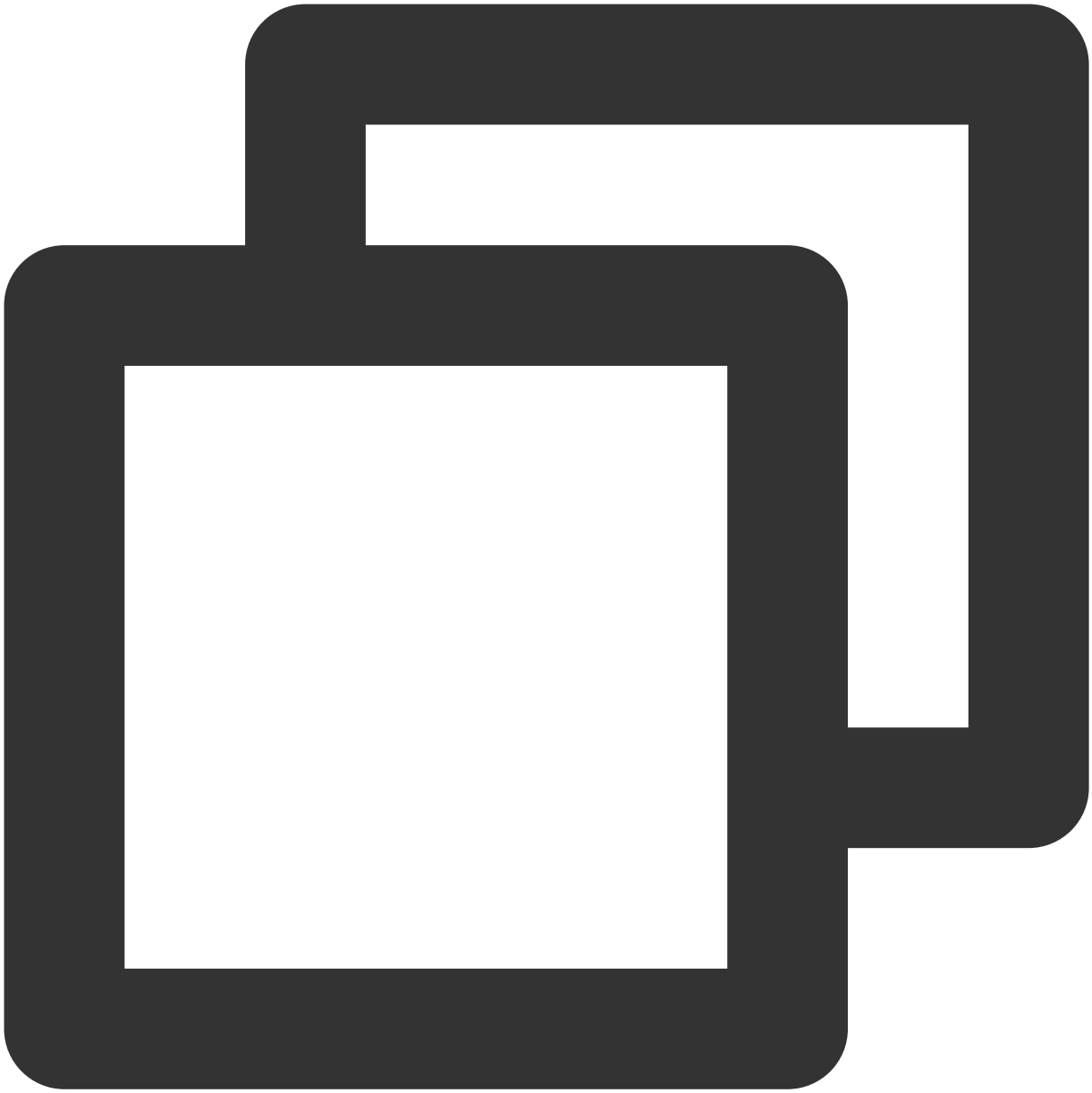
```
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Response>
```

请求2：停用工作流



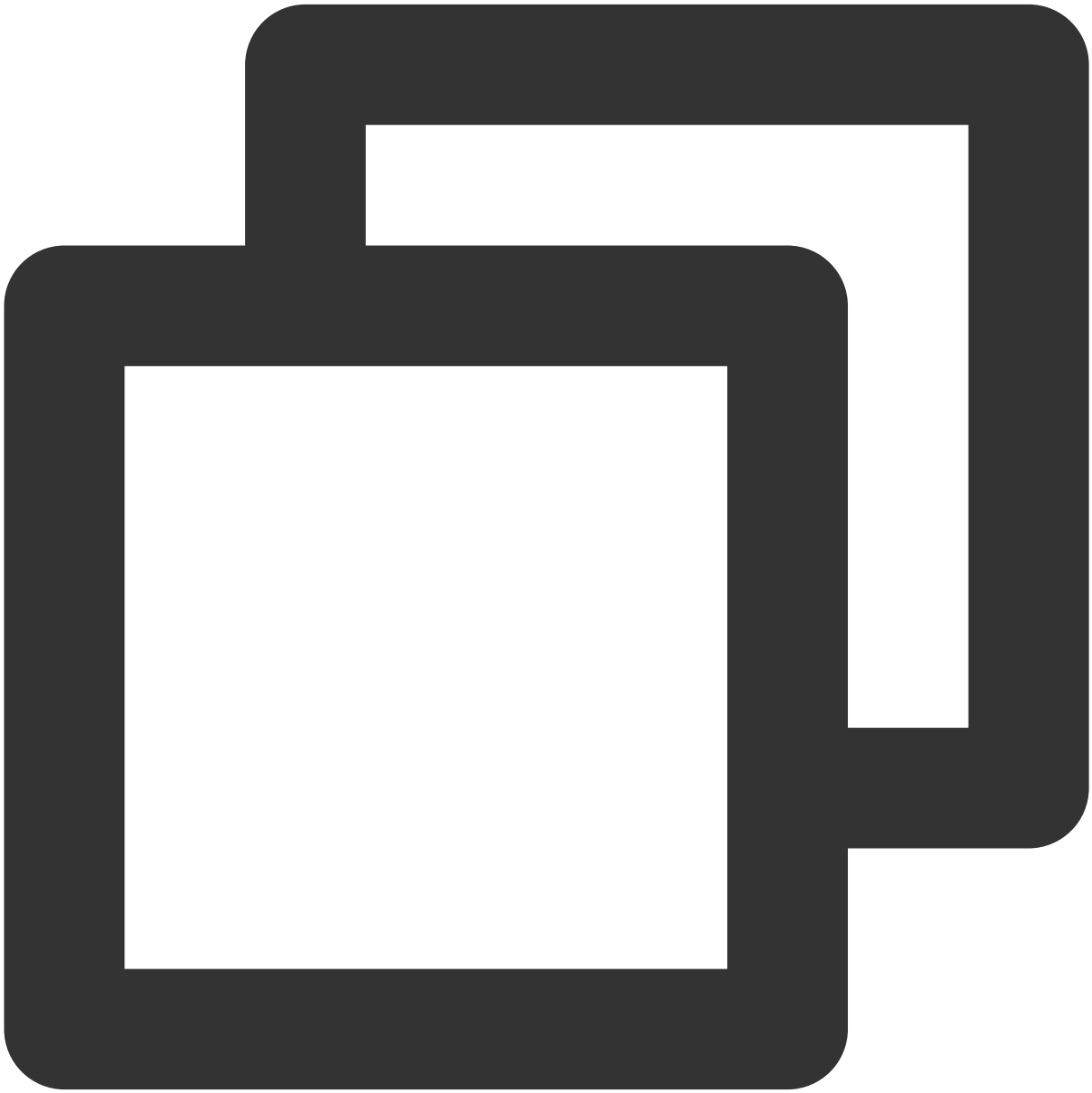
```
PUT /workflow/wc666d0b9f9dd47ae9137a096252d49f7?paused HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应2



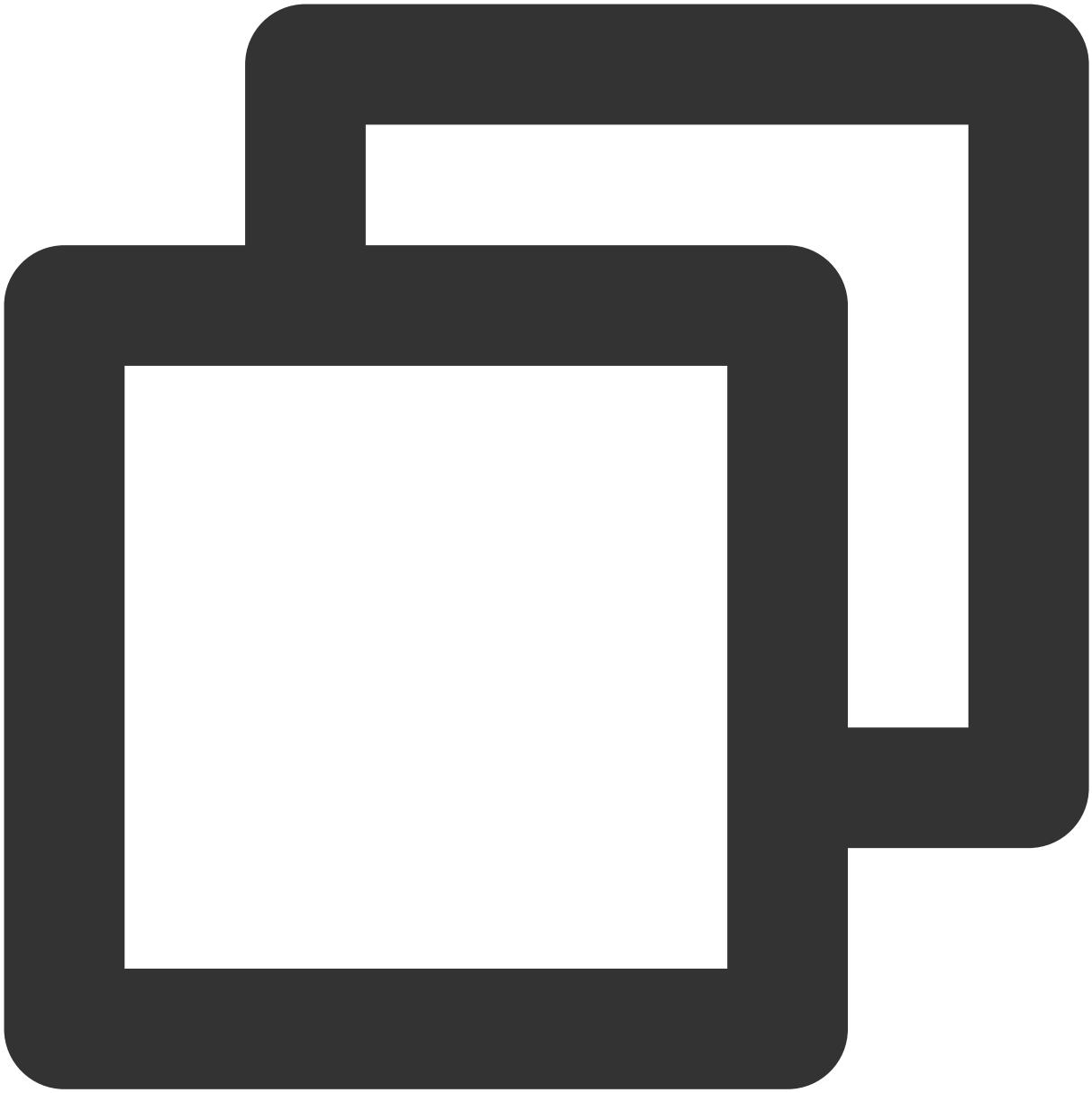
```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWRiMDlfOTBmYTUwNjRfNWYzMl80
```

请求3：启用工作流



```
PUT /workflow/wc666d0b9f9dd47ae9137a096252d49f7?active HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应3



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWRiMDlfOTBmYTUwNjRfNWYzMl80
```

测试工作流

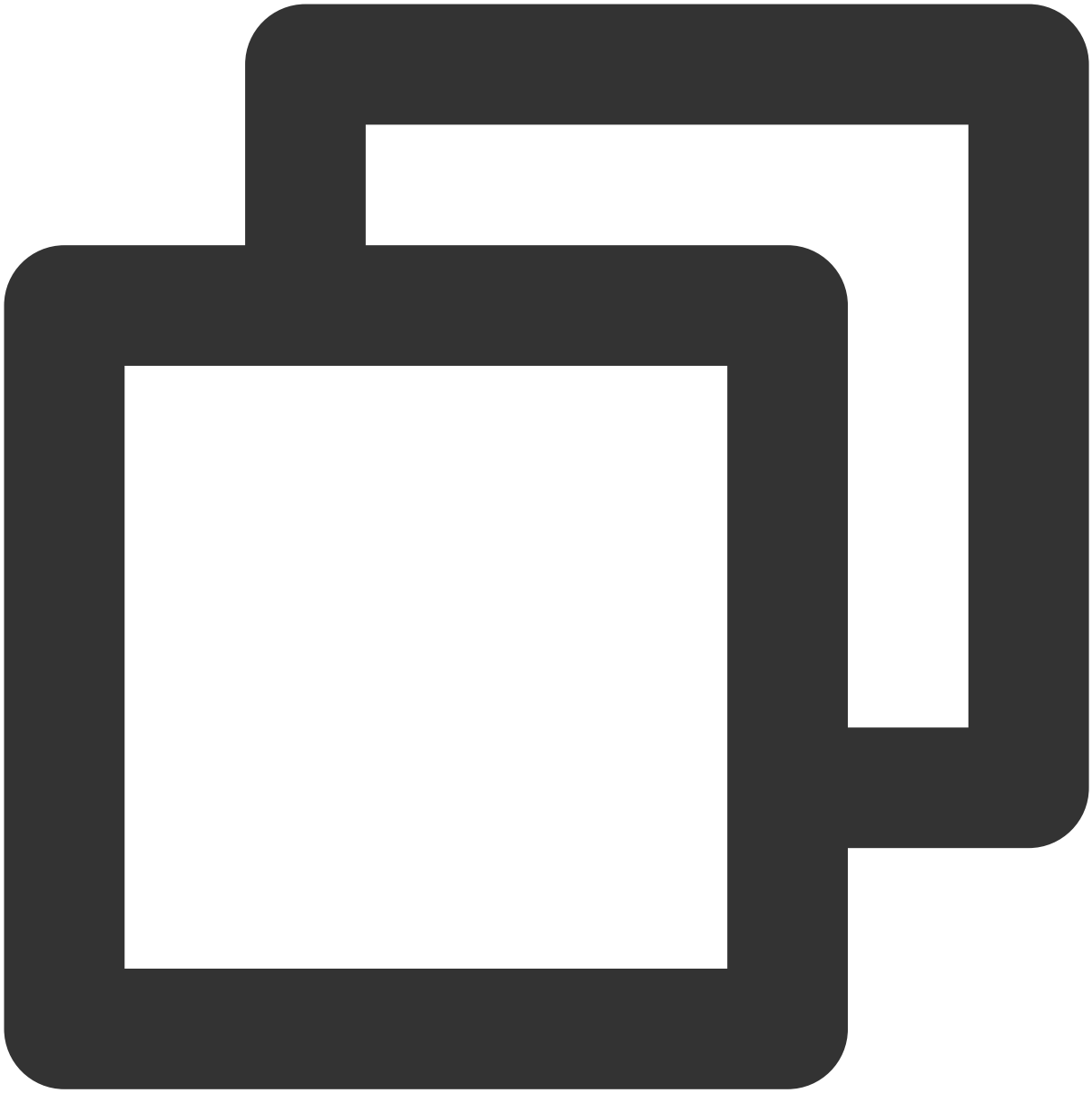
最近更新时间：2023-12-26 17:09:25

功能描述

测试工作流。

请求

请求示例



```
POST /triggerworkflow?workflowId=xxx&object=xxx HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求参数

名称	描述	类型	是否必选
workflowId	需要触发的工作流 ID	String	是
object	需要进行工作流处理的对象名称, 需要进行 url 编码	String	是
name	存量触发任务名称，支持中文、英文、数字、—和_，长度限制128字符，默认为空	String	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

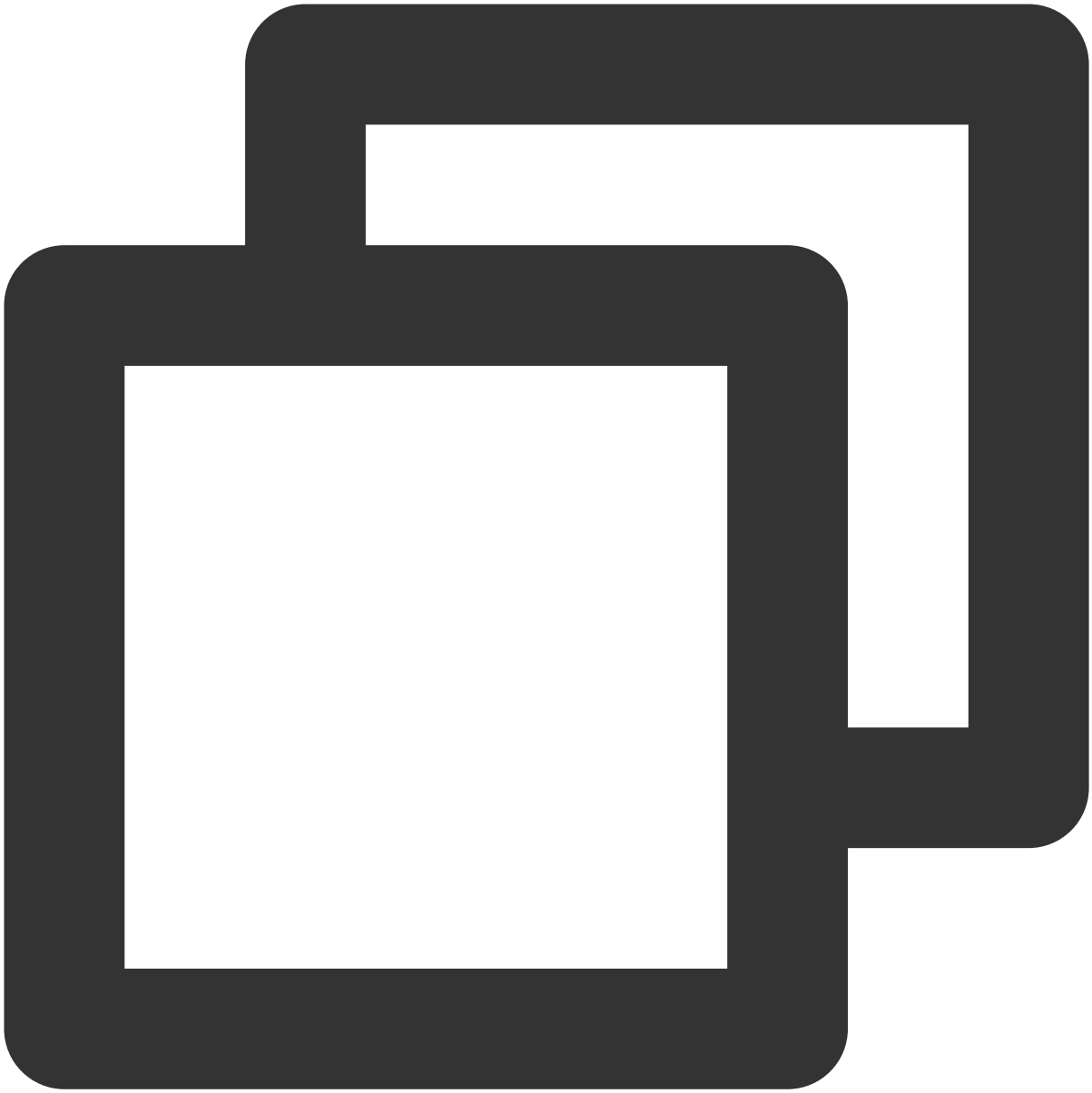
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x</RequestId>
    <InstanceId>i6fc78ca77d6011eba0ac5254008618d9=</InstanceId>
<Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

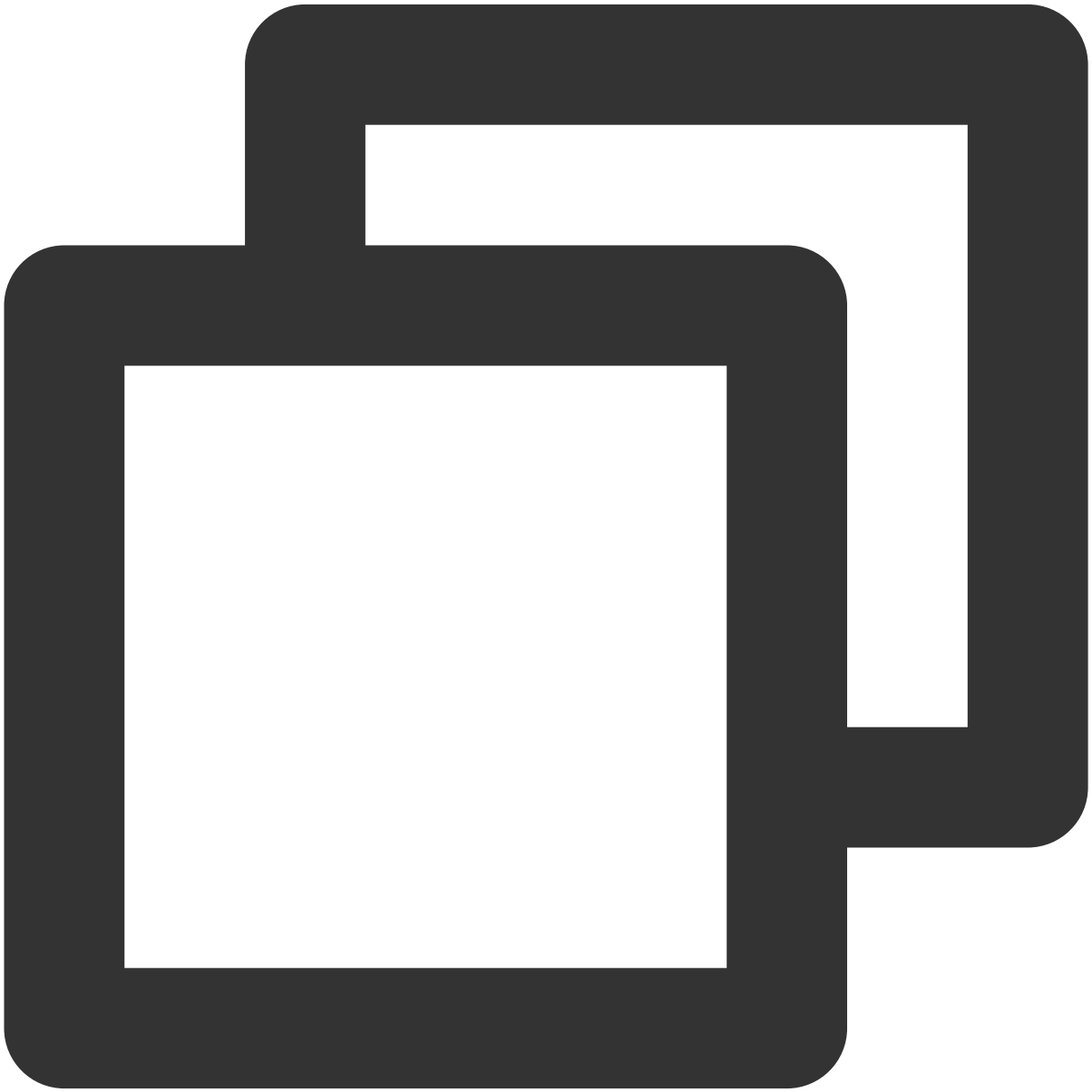
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
InstanceId	Response	实例 ID	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

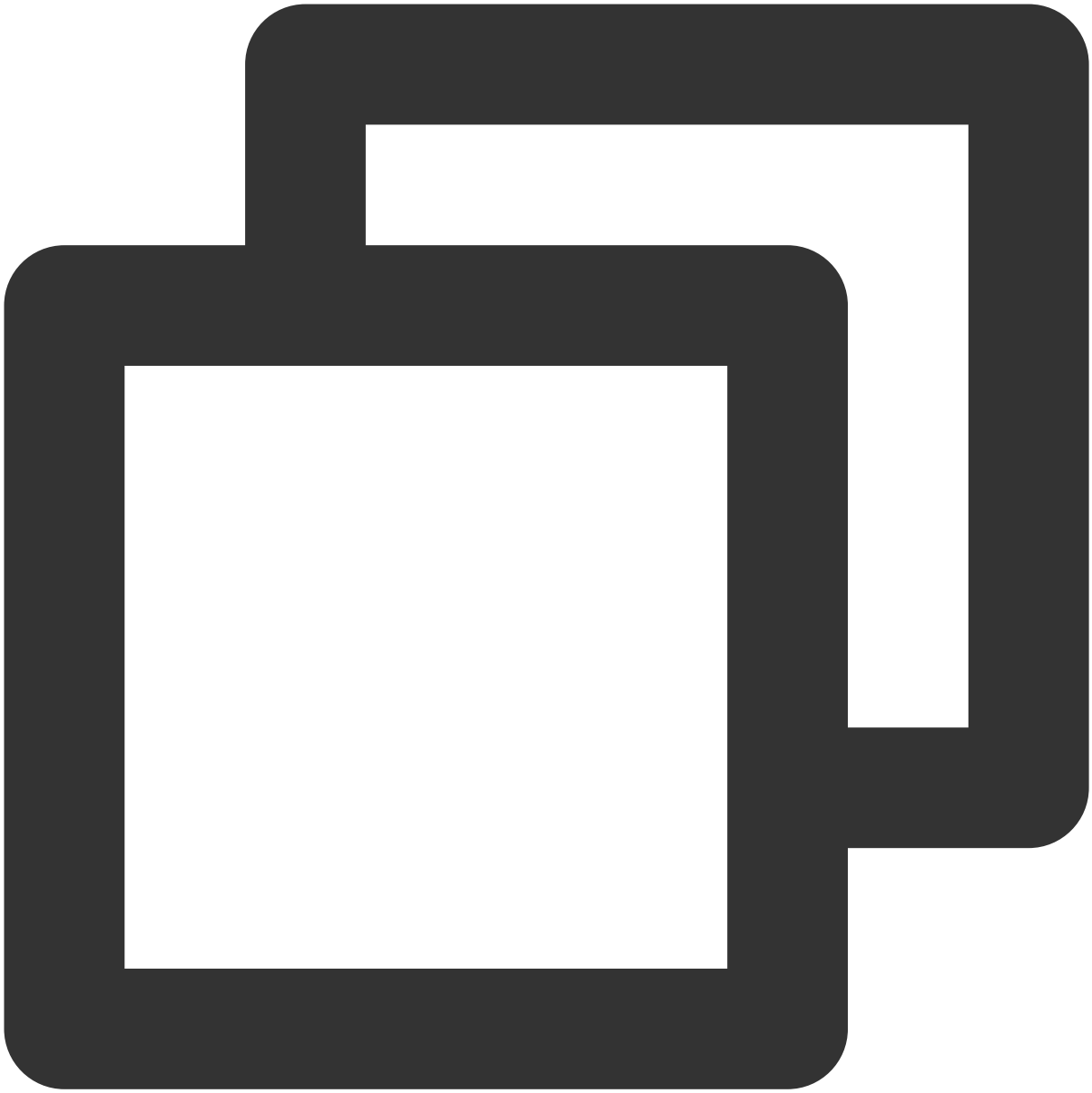
实际案例

请求



```
POST /triggerworkflow?workflowId=we59a9648e62e48ffb25e4b41f3721799&object=test.mp4
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x

<Response>
    <RequestId>NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x</RequestId>
    <InstanceId>i6fc78ca77d6011eba0ac5254008618d9=</InstanceId>
```

```
<Response>
```

工作流回调

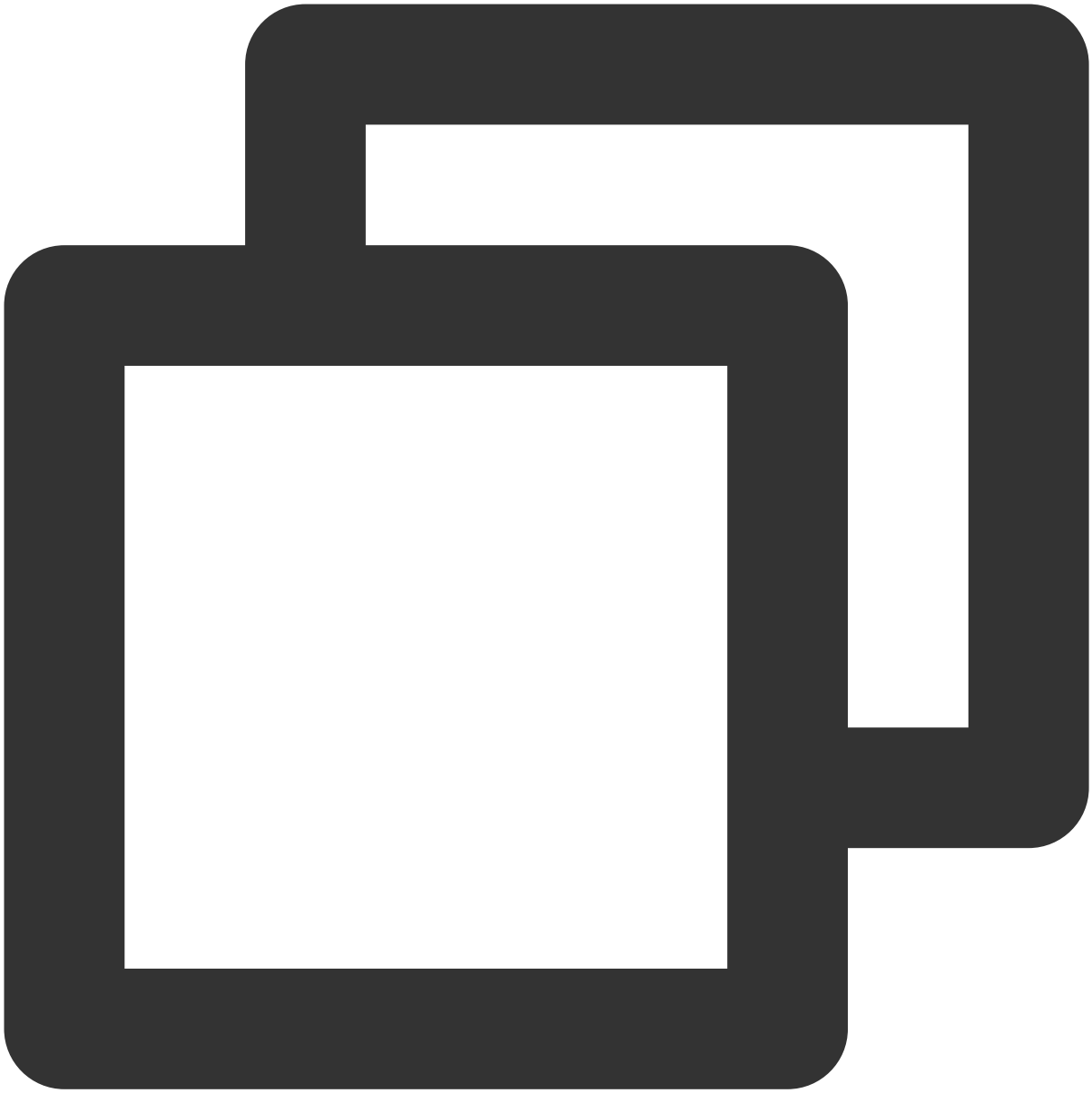
最近更新时间：2023-12-26 17:11:47

功能说明

数据万象支持自定义设置回调 URL，在工作流开启状态下，每执行一个实例后，系统会向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解工作流实例处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>WorkflowFinish</EventName>
    <WorkflowExecution>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <WorkflowName>workflow-3</WorkflowName>
        <RunId>i166ee19017b011eda8a5525400c540df</RunId>
        <State>Success</State>
        <BucketId>test-1234567890</BucketId>
        <Object>wk-test/game.mp4</Object>
        <CreateTime>2022-08-09 14:54:17+0800</CreateTime>
        <CosHeaders>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Key>Content-Type</Key>
            <Value>video/mp4</Value>
        </CosHeaders>
        <CosHeaders>
            <Key>x-cos-request-id</Key>
            <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
        </CosHeaders>
        <Tasks>
            <Type>Snapshot</Type>
            <CreateTime>2022-08-09 14:54:40+0800</CreateTime>
            <EndTime>2022-08-09 14:54:42+0800</EndTime>
            <State>Success</State>
            <JobId>j23c11e1e17b011edaab4ab15ec33d076</JobId>
            <Name>Snapshot_1581665960536</Name>
            <TemplateId>t07740e32081b44ad7a0aea03adcffd54a</TemplateId>
            <TemplateName>snapshot_1280*720_3</TemplateName>
        </Tasks>
        <Tasks>
            <Type>Transcode</Type>
            <CreateTime>2022-08-09 14:54:18+0800</CreateTime>
            <EndTime>2022-08-09 14:54:39+0800</EndTime>
            <State>Success</State>
            <JobId>j168668b217b011ed8efb27bb229e2d11</JobId>
            <Name>Transcode_1581665960537</Name>
            <TemplateId>t01e57db1c2d154d2fb57aa5de9313a897</TemplateId>
            <TemplateName>MP4-265-SD</TemplateName>
        </Tasks>
    </WorkflowExecution>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 WorkflowFinish	String
WorkflowExecution	Response	工作流的详细信息	Container

Container 节点 WorkflowExecution 的内容：

节点名称（关	父节点	描述	类型

键字）			
WorkflowId	Response.WorkflowExecution	工作流 ID	String
WorkflowName	Response.WorkflowExecution	工作流的名称	String
RunId	Response.WorkflowExecution	工作流实例的 ID	String
State	Response.WorkflowExecution	实例的状态，值为 Success、Failed 其中一个。Tasks 存在一个 Failed，则此实例视为 Failed	String
BucketId	Response.WorkflowExecution	被处理文件所属的存储桶	String
Object	Response.WorkflowExecution	该实例处理的文件名	String
CreateTime	Response.WorkflowExecution	实例的创建时间	String
CosHeaders	Response.WorkflowExecution	该 Object 上传时设定的自定义 Header 列表，没有设置时无此内容	Container 数组
Tasks	Response.WorkflowExecution	任务信息列表	Container 数组

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.WorkflowExecution.CosHeaders	自定义 Header 的名称	String
Value	Response.WorkflowExecution.CosHeaders	自定义 Header 的值	String

Container 节点 Tasks 的内容：

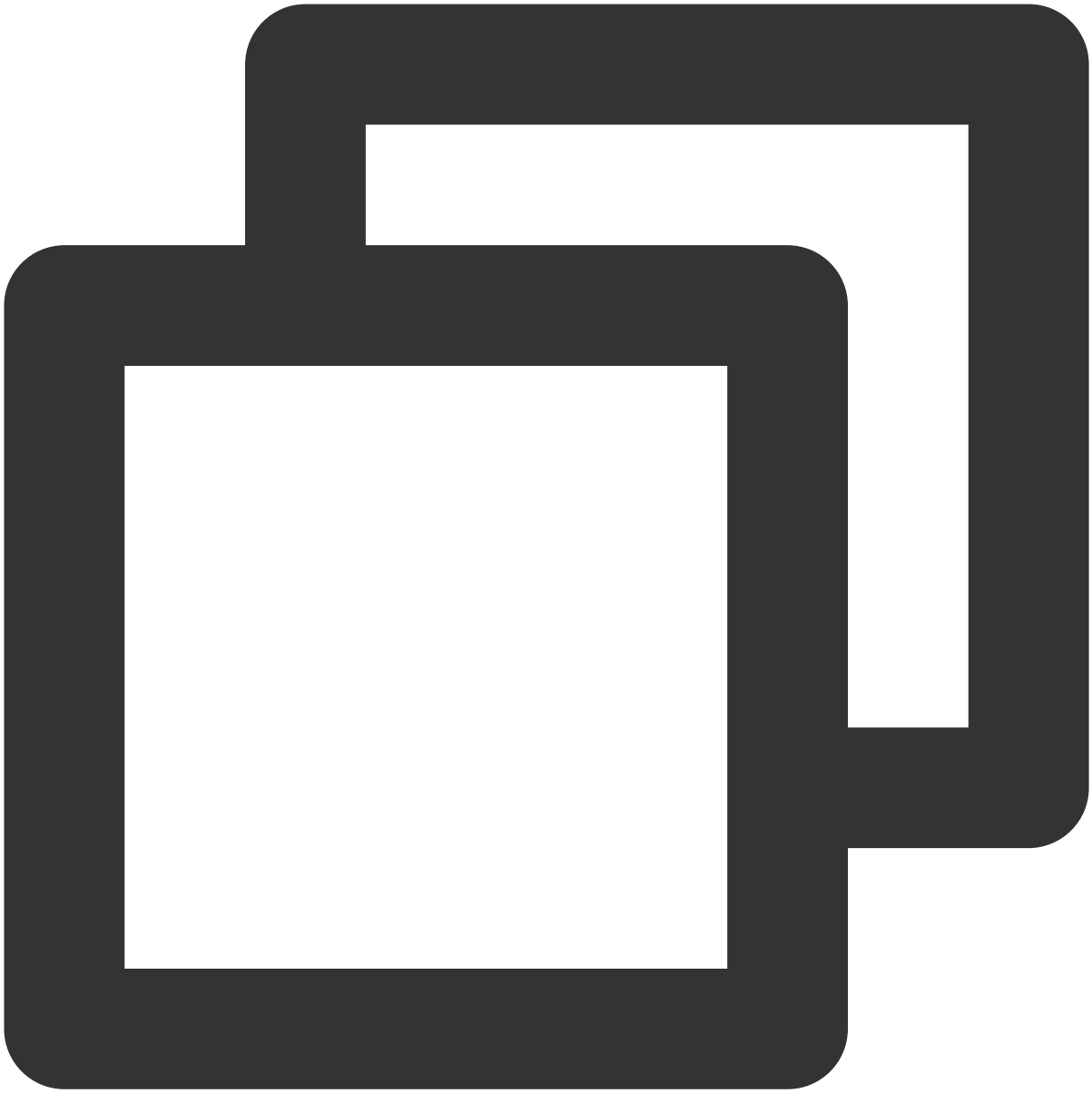
节点名称（关键字）	父节点	描述	类型
Type	Response.WorkflowExecution.Tasks	任务类型	String
Createtime	Response.WorkflowExecution.Tasks	任务的创建时间	String
StartTime	Response.WorkflowExecution.Tasks	任务开始时间	String
EndTime	Response.WorkflowExecution.Tasks	任务的完成时间	String
State	Response.WorkflowExecution.Tasks	任务的状态，值为 Success、Failed 等	String

版权所有：腾讯云计算（北京）有限责任公司

JobId	Response.WorkflowExecution.Tasks	任务的 ID	String
Name	Response.WorkflowExecution.Tasks	任务在工作流中的节点名称	String
TemplateId	Response.WorkflowExecution.Tasks	任务使用的模板 ID	String
TemplateName	Response.WorkflowExecution.Tasks	任务使用的模板名称	String

实际案例

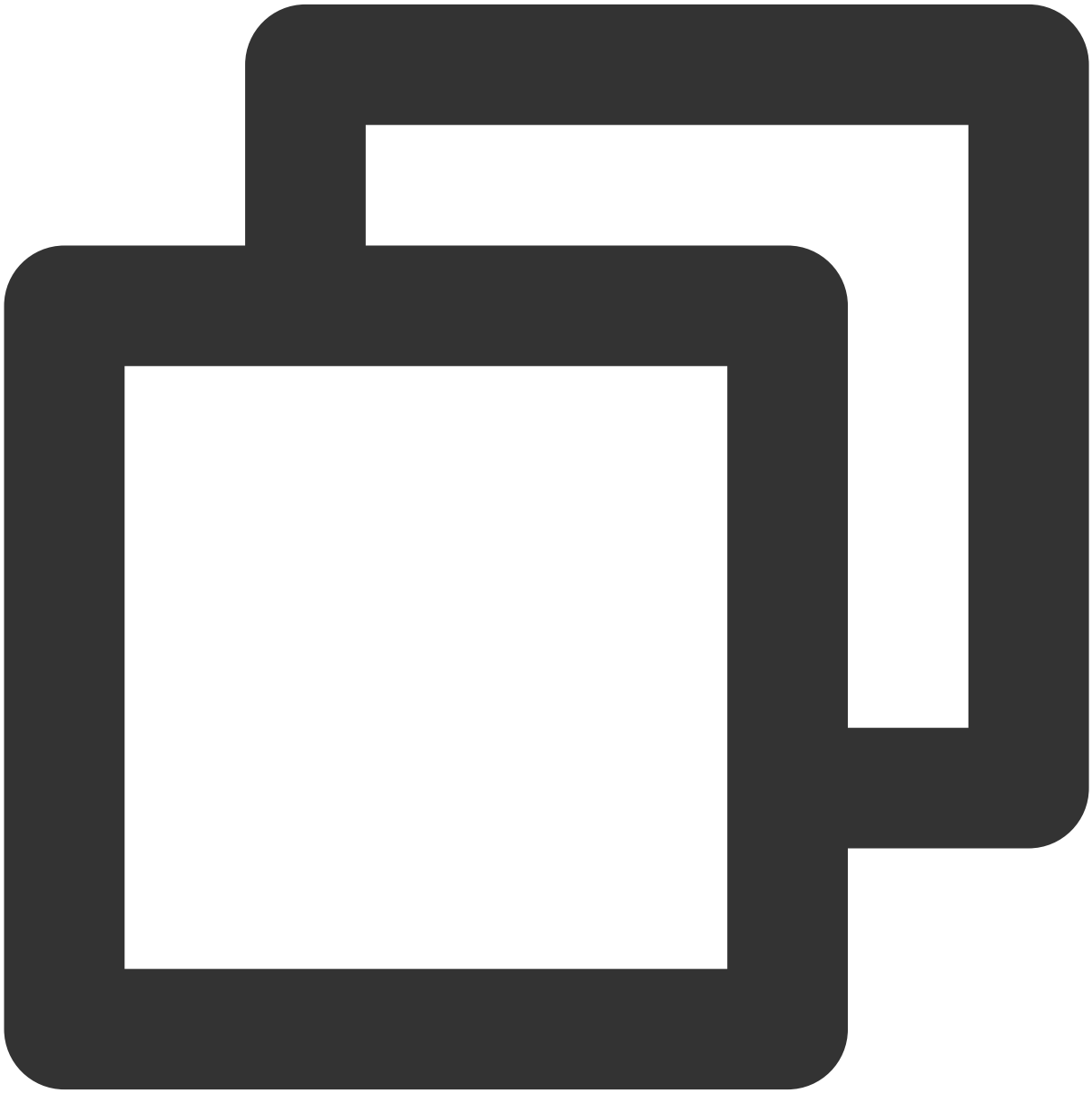
案例 1 (XML)



```
<Response>
    <EventName>WorkflowFinish</EventName>
    <WorkflowExecution>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <WorkflowName>workflow-3</WorkflowName>
        <RunId>i166ee19017b011eda8a5525400c540df</RunId>
        <State>Success</State>
        <BucketId>test-1234567890</BucketId>
        <Object>wk-test/game.mp4</Object>
        <CreateTime>2022-08-09 14:54:17+0800</CreateTime>
        <CosHeaders>
```

```
            <Key>Content-Type</Key>
            <Value>video/mp4</Value>
        </CosHeaders>
        <CosHeaders>
            <Key>x-cos-request-id</Key>
            <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
        </CosHeaders>
        <Tasks>
            <Type>Snapshot</Type>
            <CreateTime>2022-08-09 14:54:40+0800</CreateTime>
            <EndTime>2022-08-09 14:54:42+0800</EndTime>
            <State>Success</State>
            <JobId>j23c11e1e17b011edaab4ab15ec33d076</JobId>
            <Name>Snapshot_1581665960536</Name>
            <TemplateId>t07740e32081b44ad7a0aea03adcffd54a</TemplateId>
            <TemplateName>snapshot_1280*720_3</TemplateName>
        </Tasks>
        <Tasks>
            <Type>Transcode</Type>
            <CreateTime>2022-08-09 14:54:18+0800</CreateTime>
            <EndTime>2022-08-09 14:54:39+0800</EndTime>
            <State>Success</State>
            <JobId>j168668b217b011ed8efb27bb229e2d11</JobId>
            <Name>Transcode_1581665960537</Name>
            <TemplateId>t01e57db1c2d154d2fb57aa5de9313a897</TemplateId>
            <TemplateName>MP4-265-SD</TemplateName>
        </Tasks>
    </WorkflowExecution>
</Response>
```

案例 2 (JSON)




```
{
    "EventName": "WorkflowFinish",
    "WorkflowExecution": {
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-3",
        "RunId": "i166ee19017b011eda8a5525400c540df",
        "State": "Success",
        "BucketId": "test-1234567890",
        "Object": "wk-test/game.mp4",
        "CreateTime": "2022-08-09 14:54:17+0800",
        "CosHeaders": [{
```


版权所有：腾讯云计算（北京）有限责任公司

```
                "Key": "Content-Type",
                "Value": "video/mp4"
            },
            {
                "Key": "x-cos-request-id",
                "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
            }
        ],
        "Tasks": [{
                "Type": "Snapshot",
                "CreateTime": "2022-08-09 14:54:40+0800",
                "EndTime": "2022-08-09 14:54:42+0800",
                "State": "Success",
                "JobId": "j23c11e1e17b011edaab4ab15ec33d076",
                "Name": "Snapshot_1581665960536",
                "TemplateId": "t07740e32081b44ad7a0aea03adcffd54a",
                "TemplateName": "snapshot_1280*720_3"
            },
            {
                "Type": "Transcode",
                "CreateTime": "2022-08-09 14:54:18+0800",
                "EndTime": "2022-08-09 14:54:39+0800",
                "State": "Success",
                "JobId": "j168668b217b011ed8efb27bb229e2d11",
                "Name": "Transcode_1581665960537",
                "TemplateId": "t01e57db1c2d154d2fb57aa5de9313a897",
                "TemplateName": "MP4-265-SD"
            }
        ]
    }
}
```

工作流实例
获取工作流实例详情

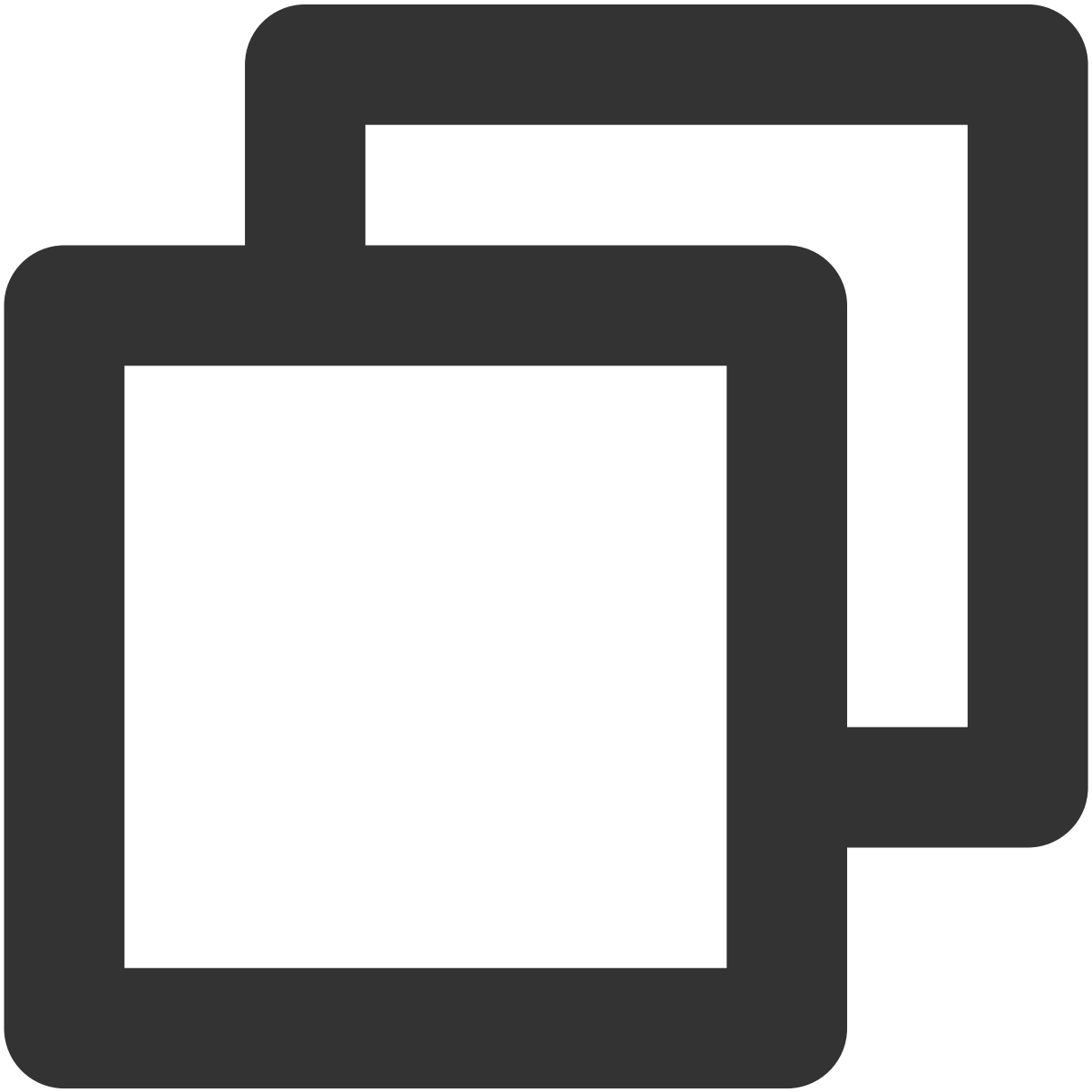
最近更新时间：2023-12-30 10:49:08

功能描述

获取工作流实例详情。

请求

请求示例



```
GET /workflowexecution/<RunId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

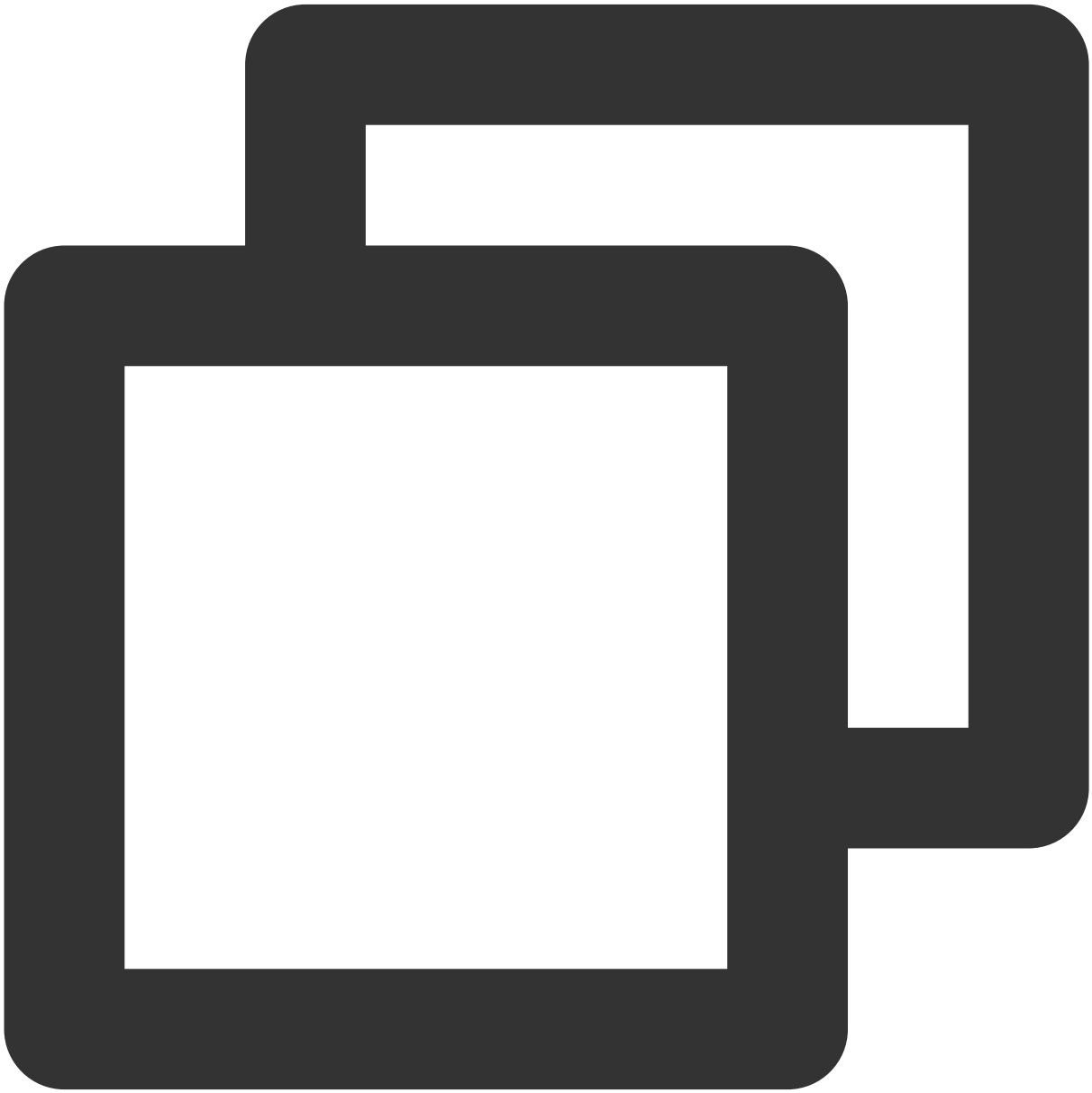
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMjA5MjZfZWM0YTYyNjRfN2U3ZF8yNzk1</RequestId>
    <WorkflowExecution>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <Name>workflow-1</Name>
        <RunId>i166ee19017b011eda8a5525400c540df</RunId>
        <CreateTime>2022-08-09T14:54:17+08:00</CreateTime>
        <Object>wk-test/game.mp4</Object>
        <State>Success</State>
        <Topology>
            <Dependencies>
```

```
            <Start>Transcode_1581665960537</Start>
            <Snapshot_1581665960536>End</Snapshot_1581665960536>
            <Transcode_1581665960537>Snapshot_1581665960536</Transcode_15816659
        </Dependencies>
        <Nodes>
            <Start>
                <Type>Start</Type>
                <Input>
                    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                    <ObjectPrefix>/wk-test</ObjectPrefix>
                    <ExtFilter>
                        <State>On</State>
                        <Video>false</Video>
                        <Audio>false</Audio>
                        <ContentType>false</ContentType>
                        <Custom>true</Custom>
                        <CustomExts>mp4</CustomExts>
                        <AllFile>false</AllFile>
                        <Image>false</Image>
                    </ExtFilter>
                    <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                </Input>
            </Start>
            <Snapshot_1581665960536>
                <Type>Snapshot</Type>
                <Operation>
                    <TemplateId>t07740e32081b44ad7a0aea03adcffd54a</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>snapshot-${number}.jpg</Object>
                    </Output>
                </Operation>
            </Snapshot_1581665960536>
            <Transcode_1581665960537>
                <Type>Transcode</Type>
                <Operation>
                    <TemplateId>t01e57db1c2d154d2fb57aa5de9313a897</TemplateId>
                    <Output>
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>trans1.mp4</Object>
                    </Output>
                </Operation>
            </Transcode_1581665960537>
        </Nodes>
    </Topology>
```

```
        <Tasks>
            <Type>Snapshot</Type>
            <JobId>j23c11e1e17b011edaab4ab15ec33d076</JobId>
            <CreateTime>2022-08-09T14:54:40+08:00</CreateTime>
            <Name>Snapshot_1581665960536</Name>
            <State>Success</State>
            <StartTime>2022-08-09T14:54:40+08:00</StartTime>
            <EndTime>2022-08-09T14:54:42+08:00</EndTime>
            <Code>Success</Code>
            <Message></Message>
        </Tasks>
        <Tasks>
            <Type>Transcode</Type>
            <JobId>j168668b217b011ed8efb27bb229e2d11</JobId>
            <CreateTime>2022-08-09T14:54:18+08:00</CreateTime>
            <Name>Transcode_1581665960537</Name>
            <State>Success</State>
            <StartTime>2022-08-09T14:54:18+08:00</StartTime>
            <EndTime>2022-08-09T14:54:39+08:00</EndTime>
            <Code>Success</Code>
            <Message>success</Message>
        </Tasks>
    </WorkflowExecution>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求 ID	String
WorkflowExecution	Response	工作流实例详细信息	Container

Container 节点 WorkflowExecution 的内容：

节点名称（关键字）	父节点	描述	类型
WorkflowId	Response.WorkflowExecution	工作流 ID	String
WorkflowName	Response.WorkflowExecution	工作流名称	String

版权所有：腾讯云计算（北京）有限责任公司

RunId	Response.WorkflowExecution	工作流实例 ID	String
CreateTime	Response.WorkflowExecution	创建时间	String
Object	Response.WorkflowExecution	COS 对象地址	String
State	Response.WorkflowExecution	工作流实例状态	String
Topology	Response.WorkflowExecution	同创建工作流接口的 Topology	Container
Tasks	Response.WorkflowExecution	工作流实例任务	Container 数组

Container 节点 Tasks 的内容：

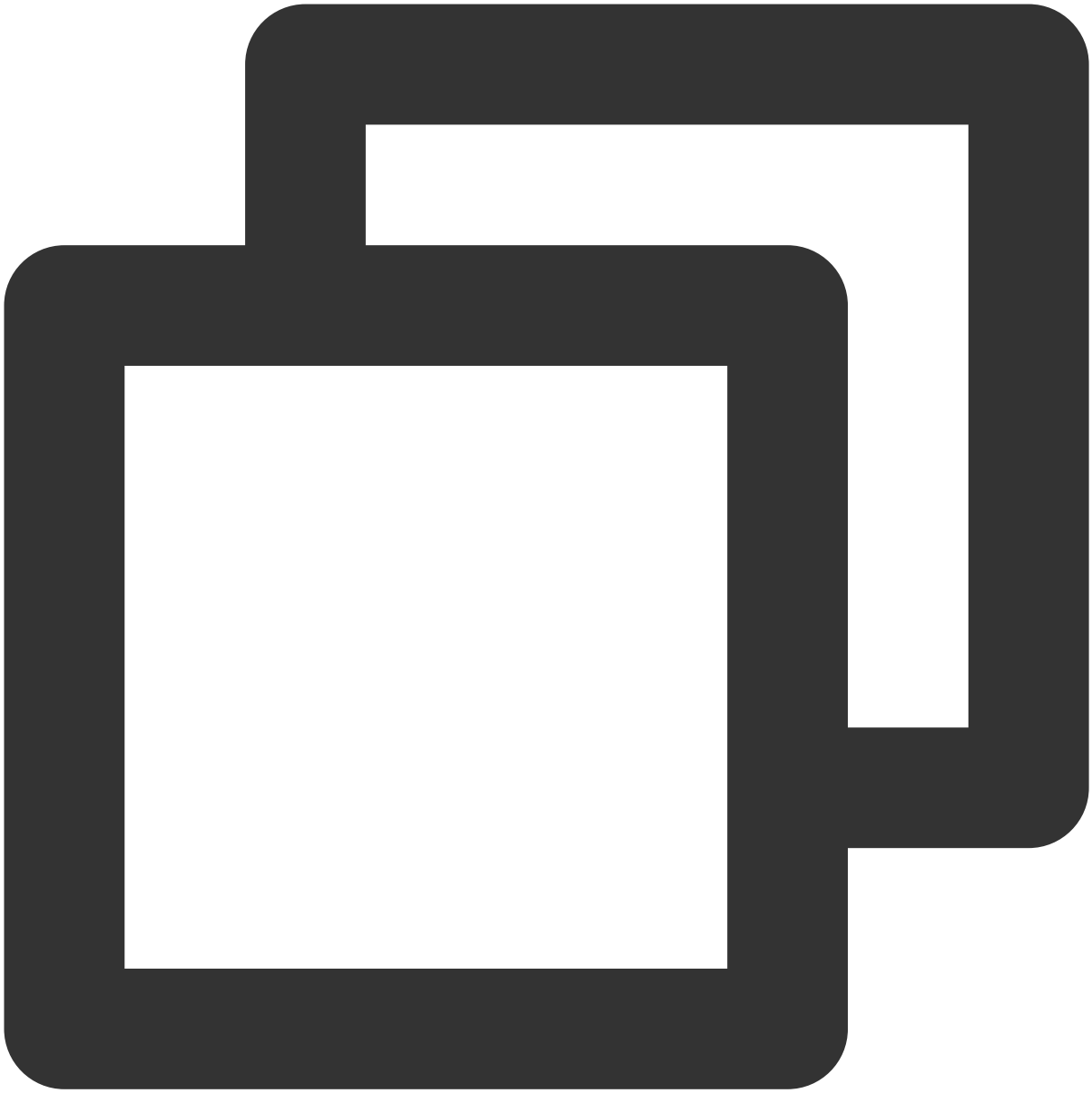
节点名称（关键字）	父节点	描述	类型
Type	Response.WorkflowExecution.Tasks	任务所属节点类型	String
State	Response.WorkflowExecution.Tasks	任务状态	String
JobId	Response.WorkflowExecution.Tasks	任务 ID	String
CreateTime	Response.WorkflowExecution.Tasks	任务创建时间	String
StartTime	Response.WorkflowExecution.Tasks	任务开始时间	String
EndTime	Response.WorkflowExecution.Tasks	任务结束时间	String
Code	Response.WorkflowExecution.Tasks	任务的错误码	String
Message	Response.WorkflowExecution.Tasks	任务的错误信息	String
Name	Response.WorkflowExecution.Tasks	工作流节点的节点名称	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

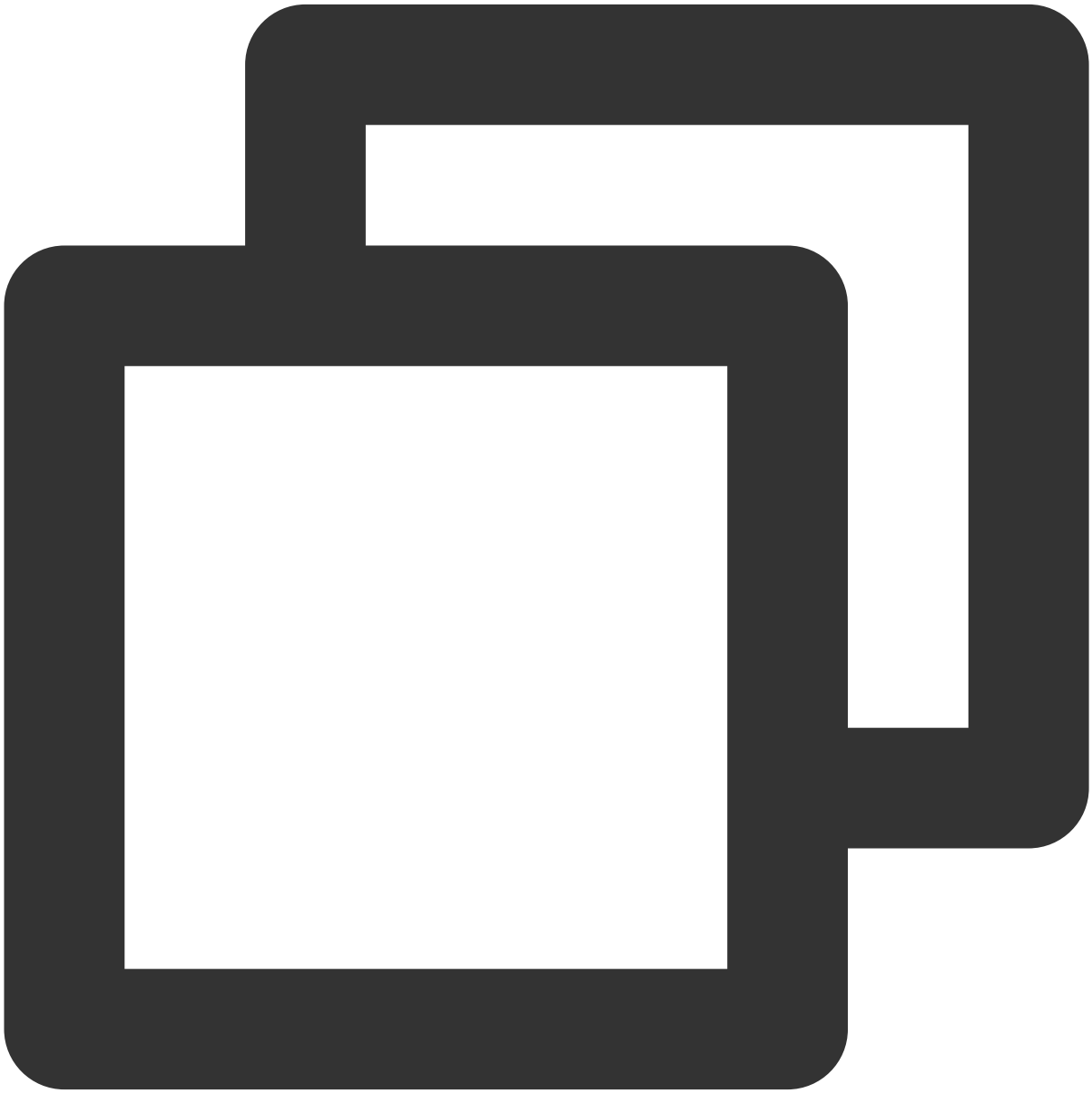
实际案例

请求



```
GET /workflowexecution/i166ee19017b011eda8a5525400c540df HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NjJmMjA5MjZfZWM0YTYyNjRfN2U3ZF8yNzk1</RequestId>
    <WorkflowExecution>
```

```
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <Name>workflow-1</Name>
        <RunId>i166ee19017b011eda8a5525400c540df</RunId>
        <CreateTime>2022-08-09T14:54:17+08:00</CreateTime>
        <Object>wk-test/game.mp4</Object>
        <State>Success</State>
        <Topology>
            <Dependencies>
                <Start>Transcode_1581665960537</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960537>Snapshot_1581665960536</Transcode_15816659
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <ObjectPrefix>/wk-test</ObjectPrefix>
                        <ExtFilter>
                            <State>On</State>
                            <Video>false</Video>
                            <Audio>false</Audio>
                            <ContentType>false</ContentType>
                            <Custom>true</Custom>
                            <CustomExts>mp4</CustomExts>
                            <AllFile>false</AllFile>
                            <Image>false</Image>
                        </ExtFilter>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicPr
                    </Input>
                </Start>
                <Snapshot_1581665960536>
                    <Type>Snapshot</Type>
                    <Operation>
                        <TemplateId>t07740e32081b44ad7a0aea03adcffd54a</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>snapshot-${number}.jpg</Object>
                        </Output>
                    </Operation>
                </Snapshot_1581665960536>
                <Transcode_1581665960537>
                    <Type>Transcode</Type>
                    <Operation>
                        <TemplateId>t01e57db1c2d154d2fb57aa5de9313a897</TemplateId>
                        <Output>
```

```
                        <Region>ap-chongqing</Region>
                        <Bucket>test-1234567890</Bucket>
                        <Object>trans1.mp4</Object>
                    </Output>
                </Operation>
            </Transcode_1581665960537>
        </Nodes>
    </Topology>
    <Tasks>
        <Type>Snapshot</Type>
        <JobId>j23c11e1e17b011edaab4ab15ec33d076</JobId>
        <CreateTime>2022-08-09T14:54:40+08:00</CreateTime>
        <Name>Snapshot_1581665960536</Name>
        <State>Success</State>
        <StartTime>2022-08-09T14:54:40+08:00</StartTime>
        <EndTime>2022-08-09T14:54:42+08:00</EndTime>
        <Code>Success</Code>
        <Message></Message>
    </Tasks>
    <Tasks>
        <Type>Transcode</Type>
        <JobId>j168668b217b011ed8efb27bb229e2d11</JobId>
        <CreateTime>2022-08-09T14:54:18+08:00</CreateTime>
        <Name>Transcode_1581665960537</Name>
        <State>Success</State>
        <StartTime>2022-08-09T14:54:18+08:00</StartTime>
        <EndTime>2022-08-09T14:54:39+08:00</EndTime>
        <Code>Success</Code>
        <Message>success</Message>
    </Tasks>
</WorkflowExecution>
```

获取工作流实例详情列表

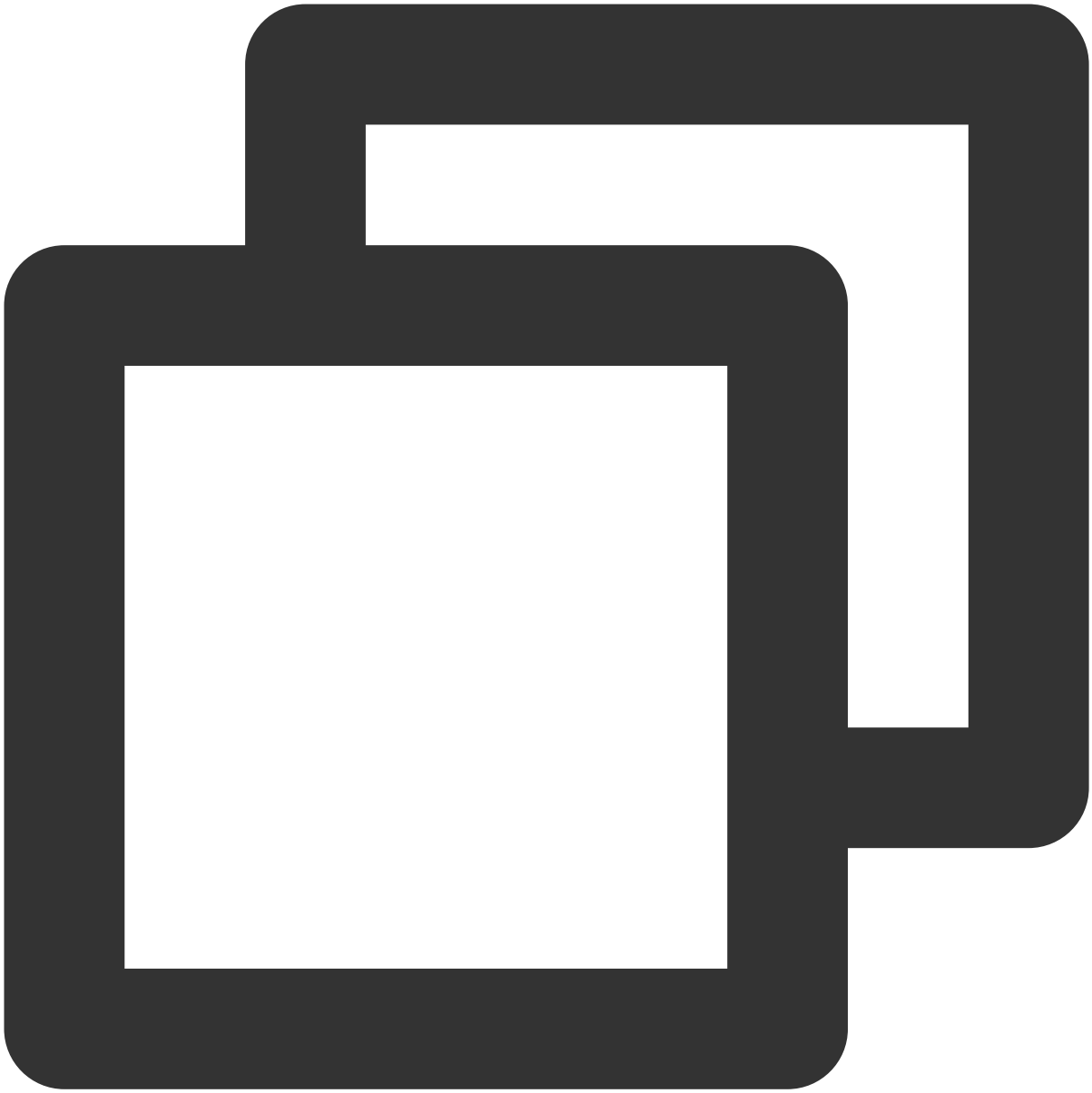
最近更新时间：2023-12-30 10:49:21

功能描述

获取工作流实例列表。

请求

请求示例



```
GET /workflowexecution HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

参数名称（关键字）	描述	类型	是否必选
workflowId	工作流 ID	String	是
name	文件名称	String	否
orderByTime	Desc 或者 Asc。默认为 Desc	String	否
size	拉取的最大任务数。默认为10。最大为100	String	否
states	工作流实例状态，以 `,` 分割支持多状态 All，Success，Failed，Running，Cancel。默认为 All	String	否
startCreationTime	拉取创建时间大于等于该时间。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否
endCreationTime	拉取创建时间小于等于该时间。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否
nextToken	请求的上下文，用于翻页。下一页输入 token	String	否
jobId	批量触发工作流任务 ID，用于扫描出批量触发任务 ID 对应执行的工作流实例	String	否

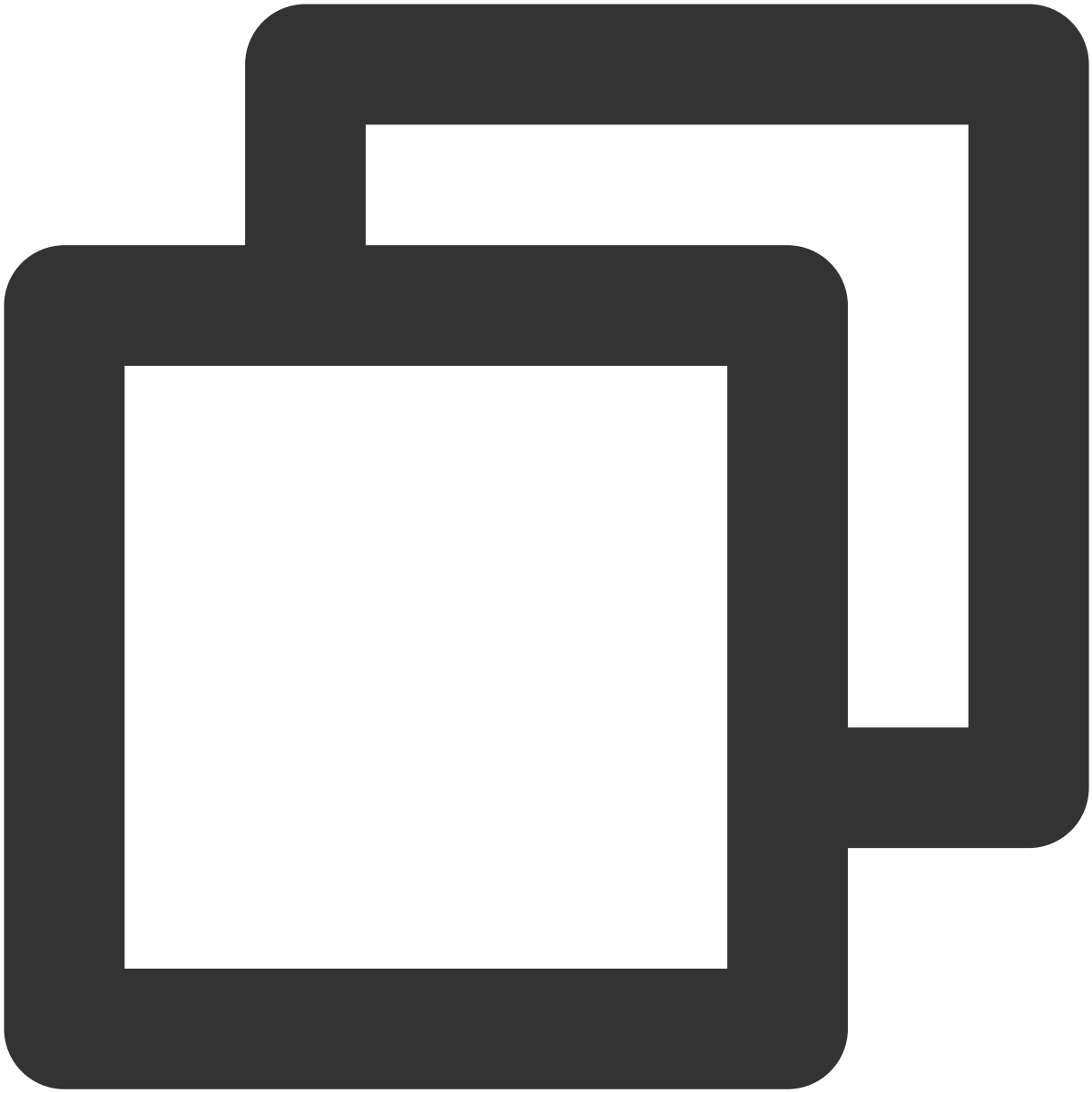
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJmMjEyMmFfZWM0YTYyNjRfN2U2NV8yNmI0</RequestId>
    <WorkflowExecutionList>
        <RunId>i166ee19017b011eda8a5525400c540df</RunId>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <Object>wk-test/game.mp4</Object>
        <CreateTime>2022-08-09T14:54:17+08:00</CreateTime>
        <State>Success</State>
    </WorkflowExecutionList>
    <WorkflowExecutionList>
        <RunId>i769bd26517af11ed917f525400a3c59f</RunId>
```

```
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <Object>wk-test/game.mp4</Object>
        <CreateTime>2022-08-09T14:49:49+08:00</CreateTime>
        <State>Success</State>
    </WorkflowExecutionList>
    <NextToken>115716</NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
WorkflowExecutionList	Response	工作流实例详细信息	Container
RequestId	Response	请求 ID	String
NextToken	Response	翻页的上下文 Token	String

Container 节点 WorkflowExecutionList 的内容：

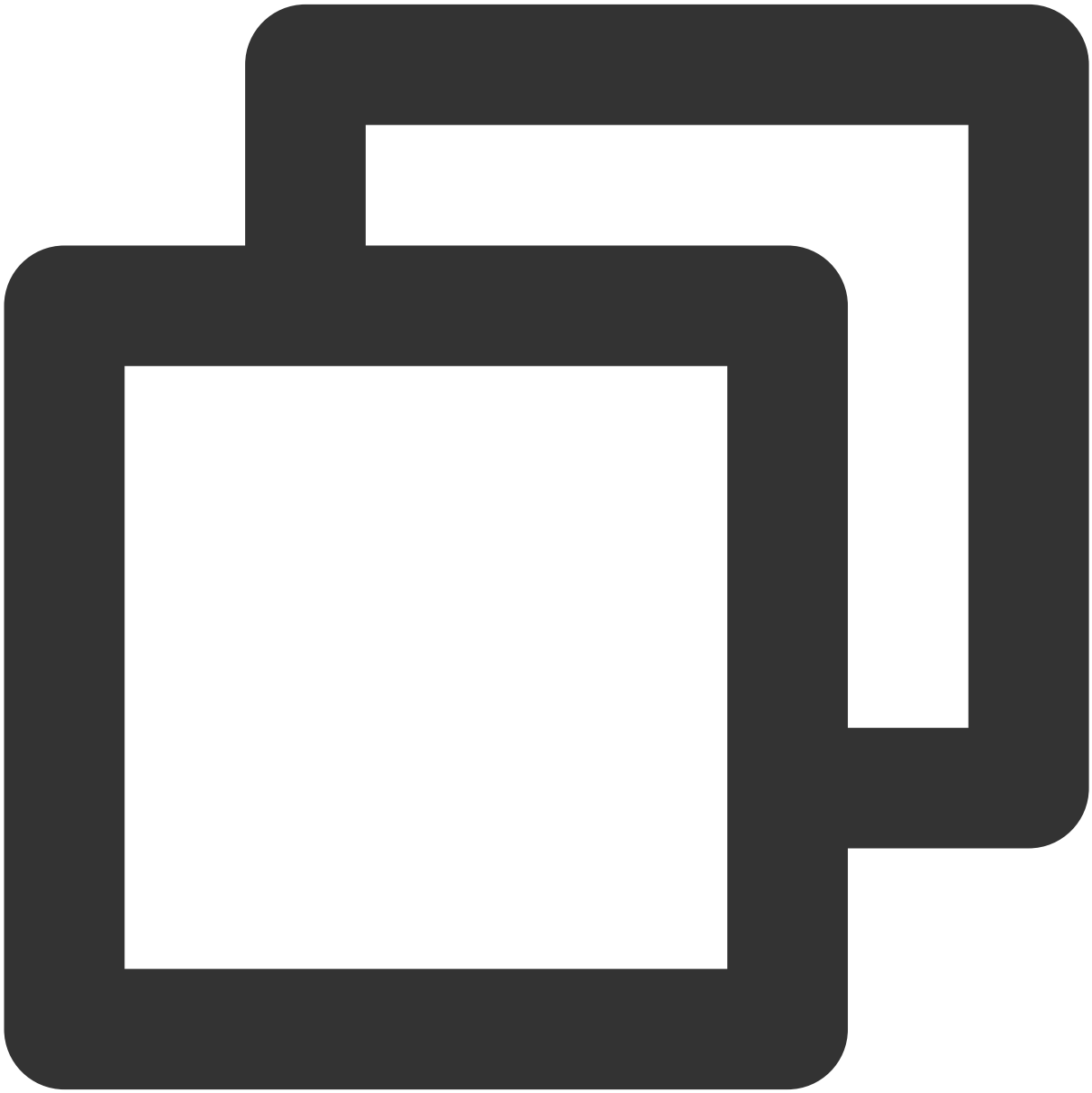
节点名称（关键字）	父节点	描述	类型
RunId	Response.WorkflowExecutionList	工作流实例 ID	String
WorkflowId	Response.WorkflowExecutionList	工作流 ID	String
State	Response.WorkflowExecutionList	工作流实例状态	String
CreateTime	Response.WorkflowExecutionList	创建时间	String
Object	Response.WorkflowExecutionList	COS 对象地址	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

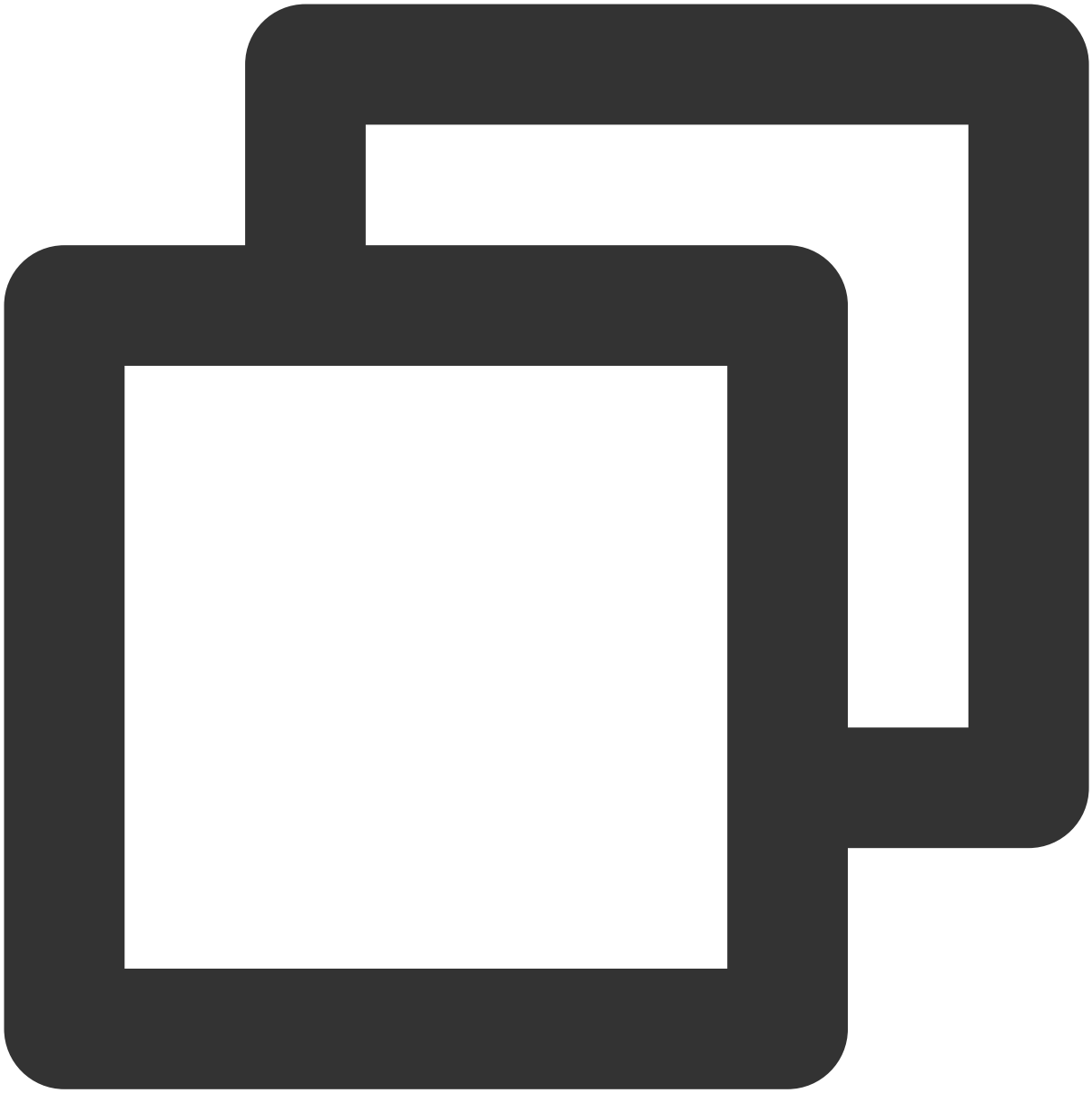
实际案例

请求



```
GET /workflowexecution?workflowId=web6ac56c1ef54dbfa44d7f4103203be9 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMjEyMmFfZWM0YTYyNjRfN2U2NV8yNmI0

<Response>
    <RequestId>NjJmMjEyMmFfZWM0YTYyNjRfN2U2NV8yNmI0</RequestId>
    <WorkflowExecutionList>
```

```
        <RunId>i166ee19017b011eda8a5525400c540df</RunId>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <Object>wk-test/game.mp4</Object>
        <CreateTime>2022-08-09T14:54:17+08:00</CreateTime>
        <State>Success</State>
    </WorkflowExecutionList>
    <WorkflowExecutionList>
        <RunId>i769bd26517af11ed917f525400a3c59f</RunId>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <Object>wk-test/game.mp4</Object>
        <CreateTime>2022-08-09T14:49:49+08:00</CreateTime>
        <State>Success</State>
    </WorkflowExecutionList>
    <NextToken>115716</NextToken>
</Response>
```

任务接口
媒体处理
取消媒体处理任务
CancelMediaJob

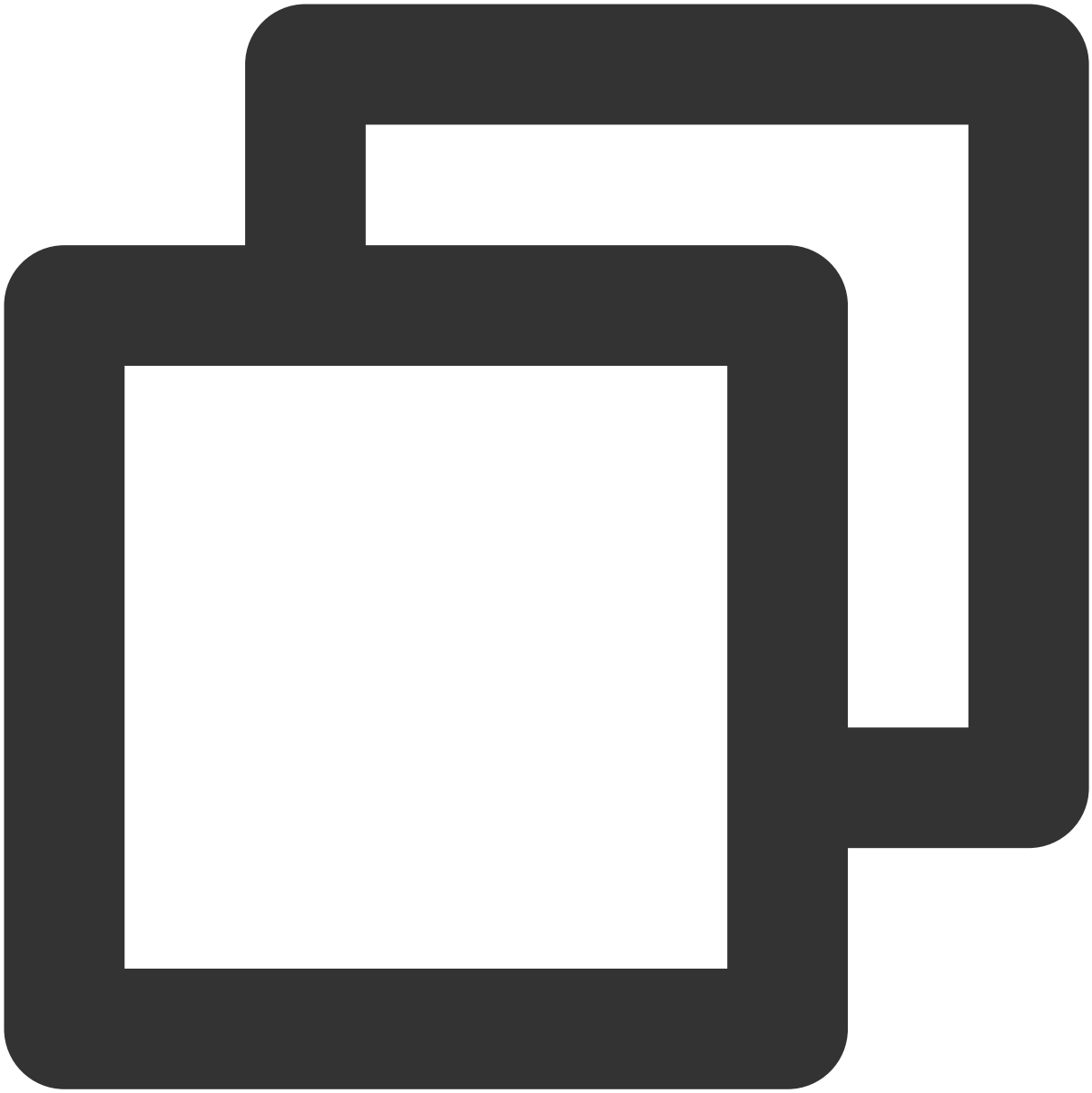
最近更新时间：2023-12-30 16:24:54

功能描述

取消一个任务，只有未执行的任务才可以被取消。

请求

请求示例



```
PUT /jobs/<jobId>?cancel HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

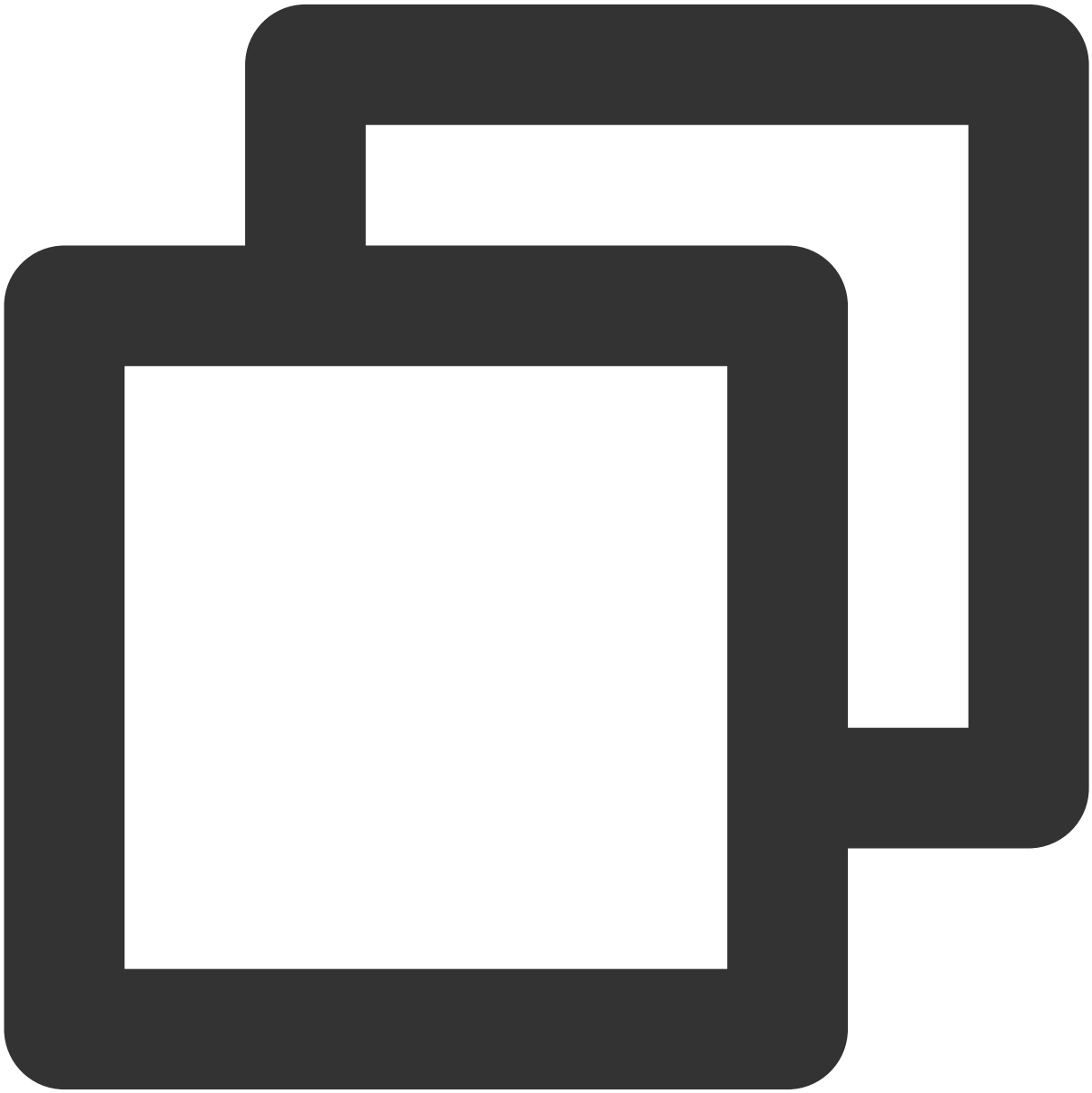
该请求无响应体。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

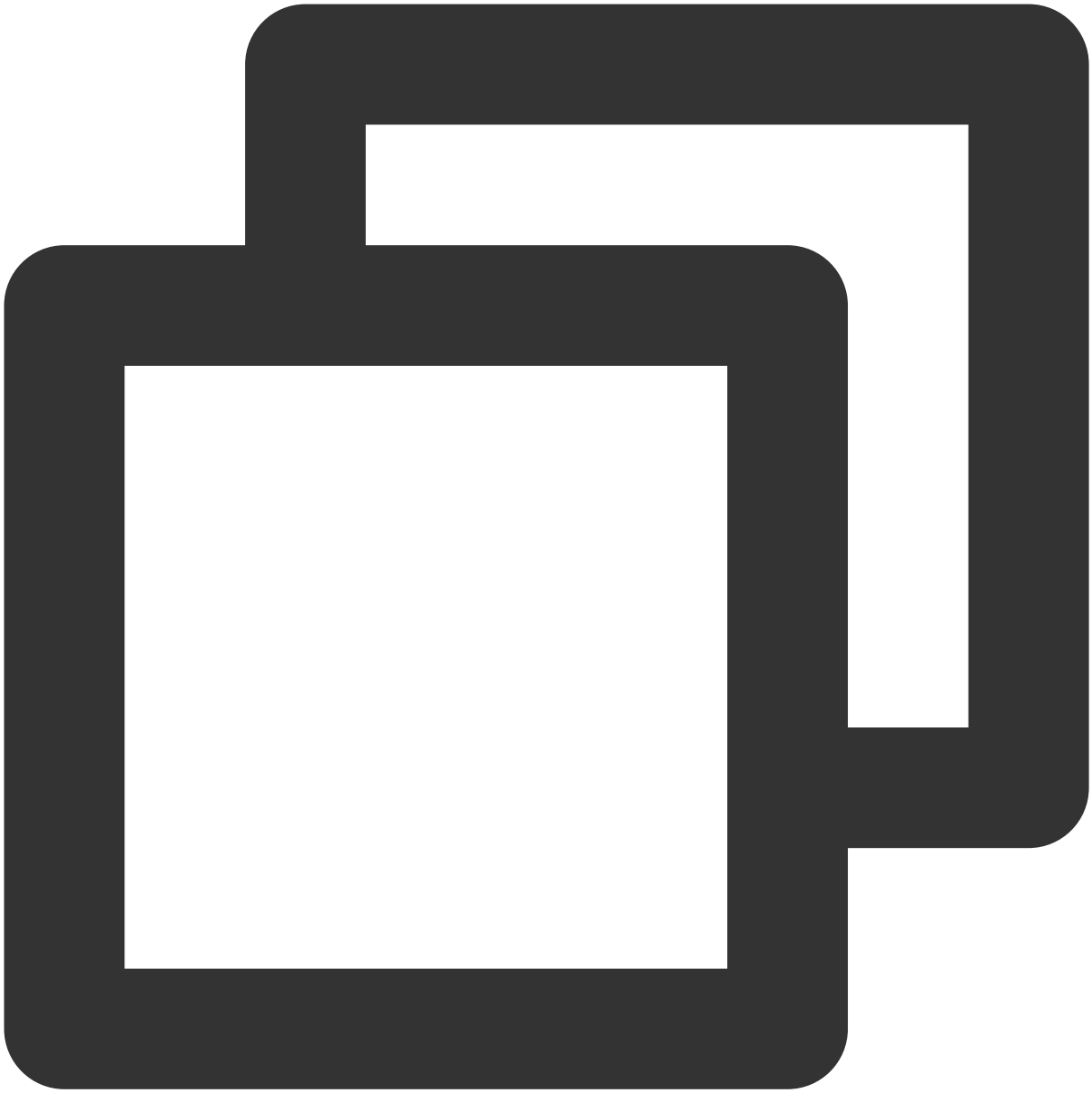
实际案例

请求



```
PUT /jobs/jb57da17e29c111eda218713081fe2b05?cancel HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjY=
```

查询媒体处理任务

DescribeMediaJob

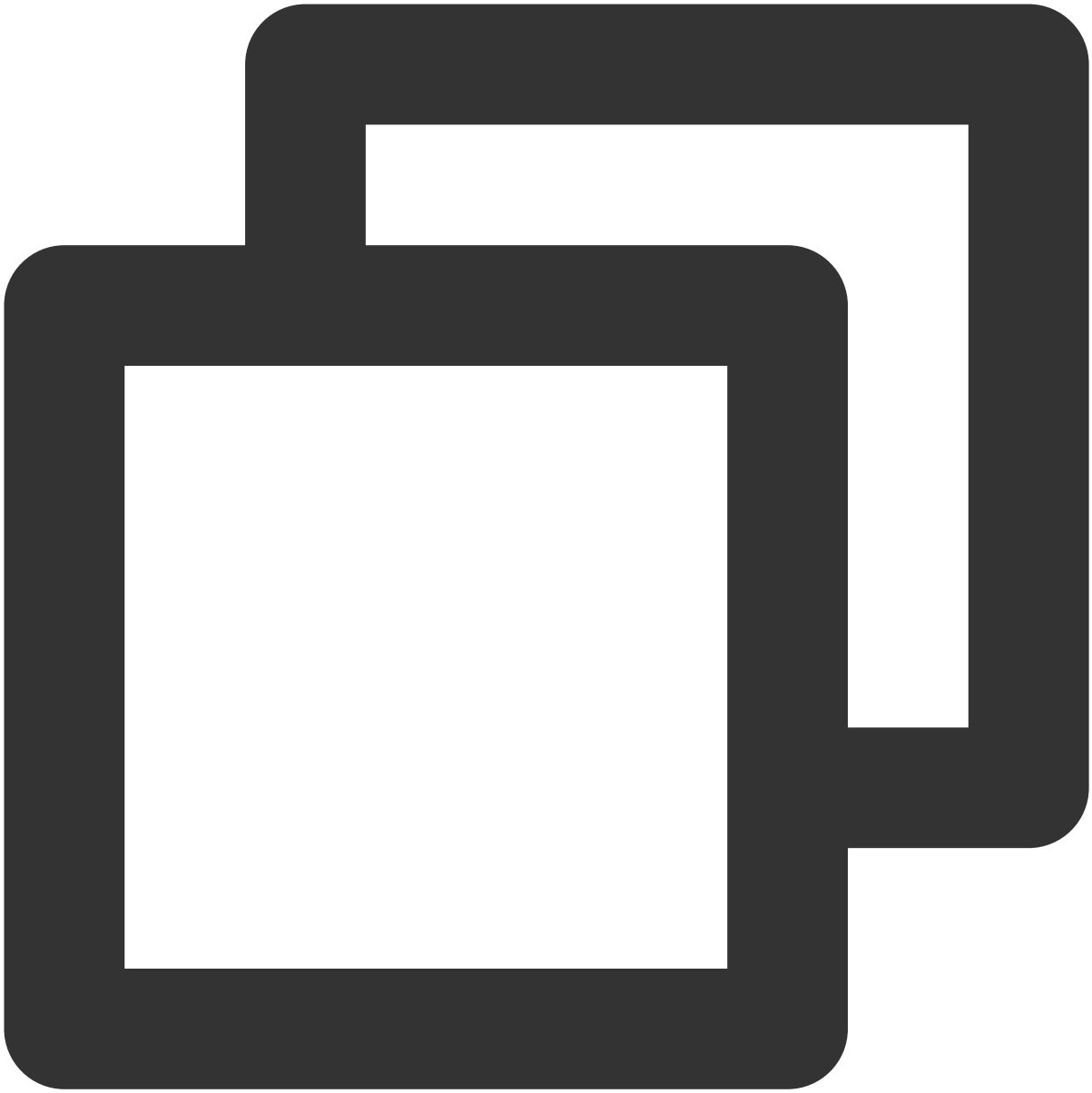
最近更新时间：2023-12-30 16:25:11

功能描述

查询指定的任务。

请求

请求示例



```
GET /jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

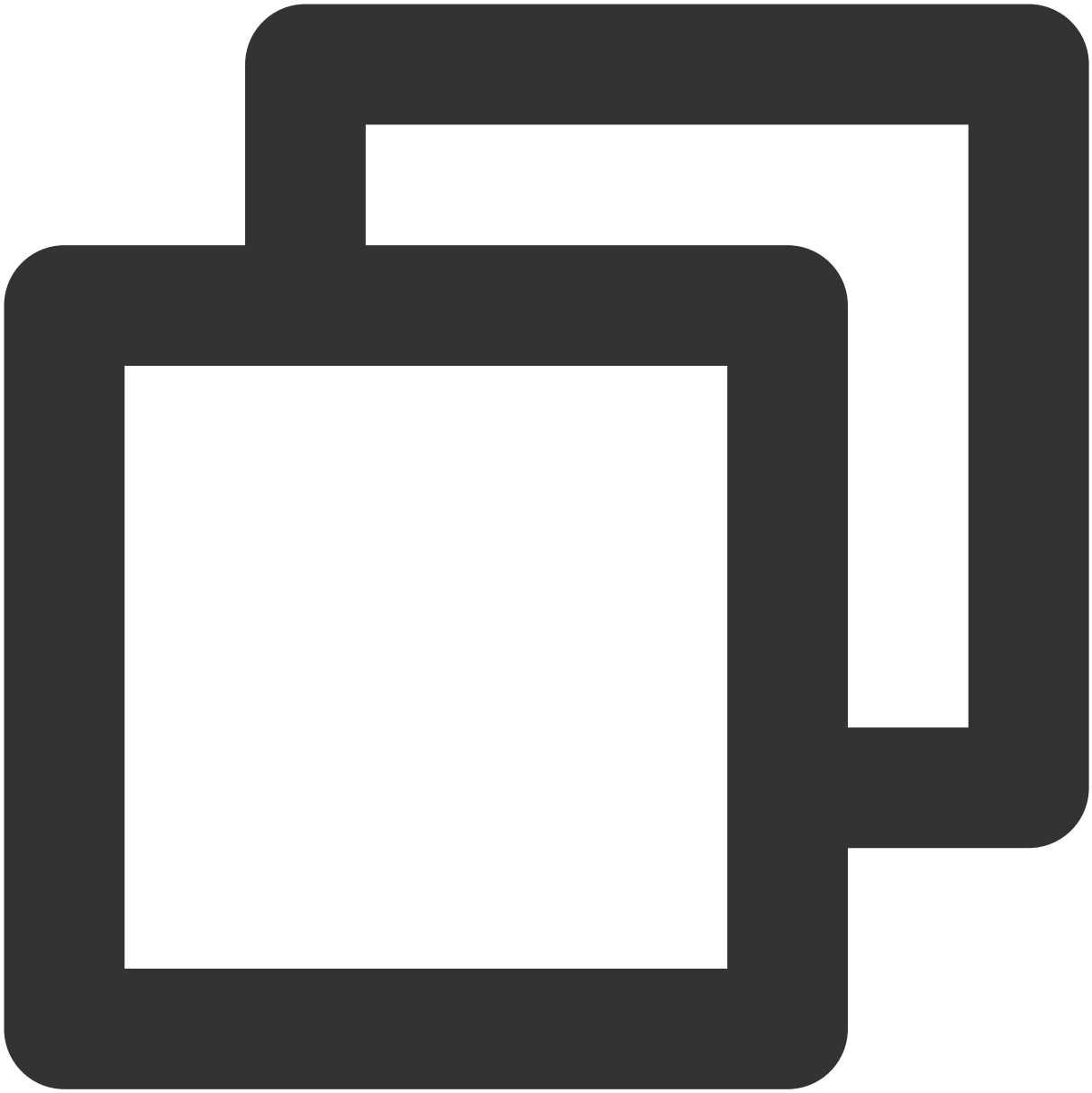
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
    </JobsDetail>
    <NonExistJobIds></NonExistJobIds>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container
NonExistJobIds	Response	查询的 ID 中不存在任务，所有任务都存在时不返回	String

对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：

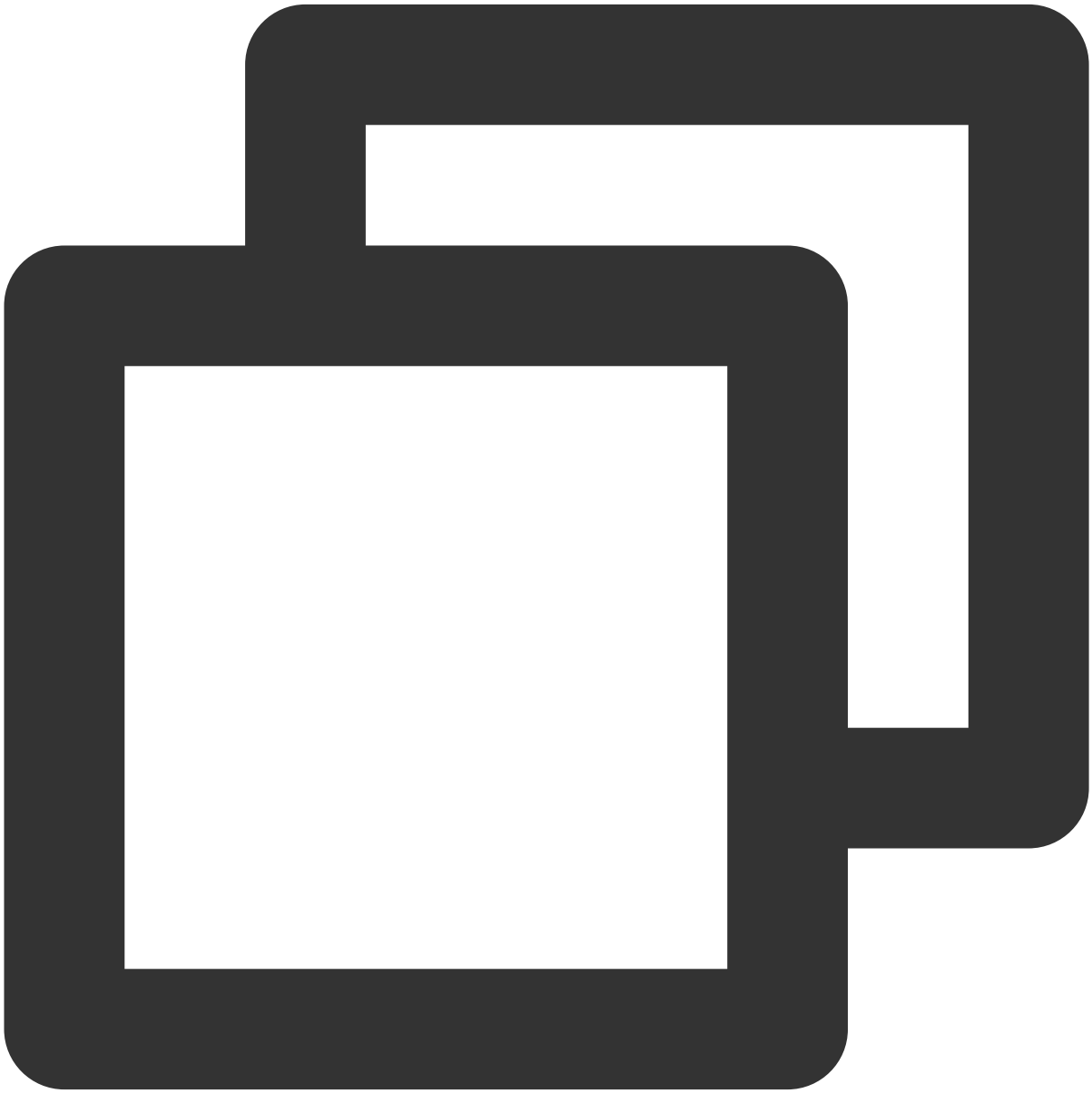
音视频转码
极速高清转码
视频转动图
视频截帧
智能封面
音视频拼接
人声分离
精彩集锦
SDR to HDR
视频增强
超分辨率
音视频转封装
数字水印
提取数字水印
视频标签
获取媒体信息
音视频流分离
视频质量分析
语音合成
音频降噪

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

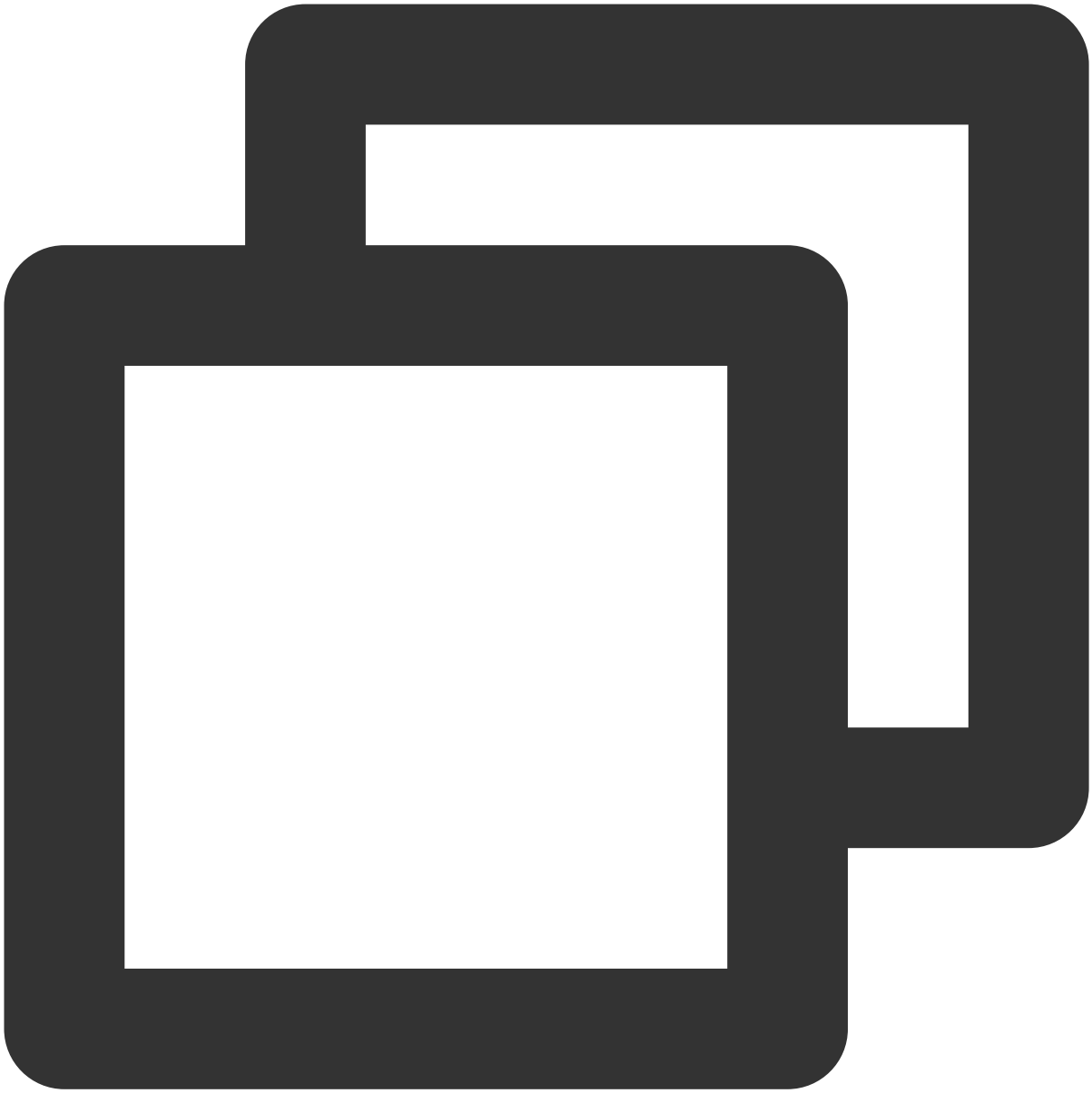
实际案例

请求1：查询一个任务，该任务正在执行中



```
GET /jobs/j8d121820f5e411ec926ef19d53ba9c6f HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

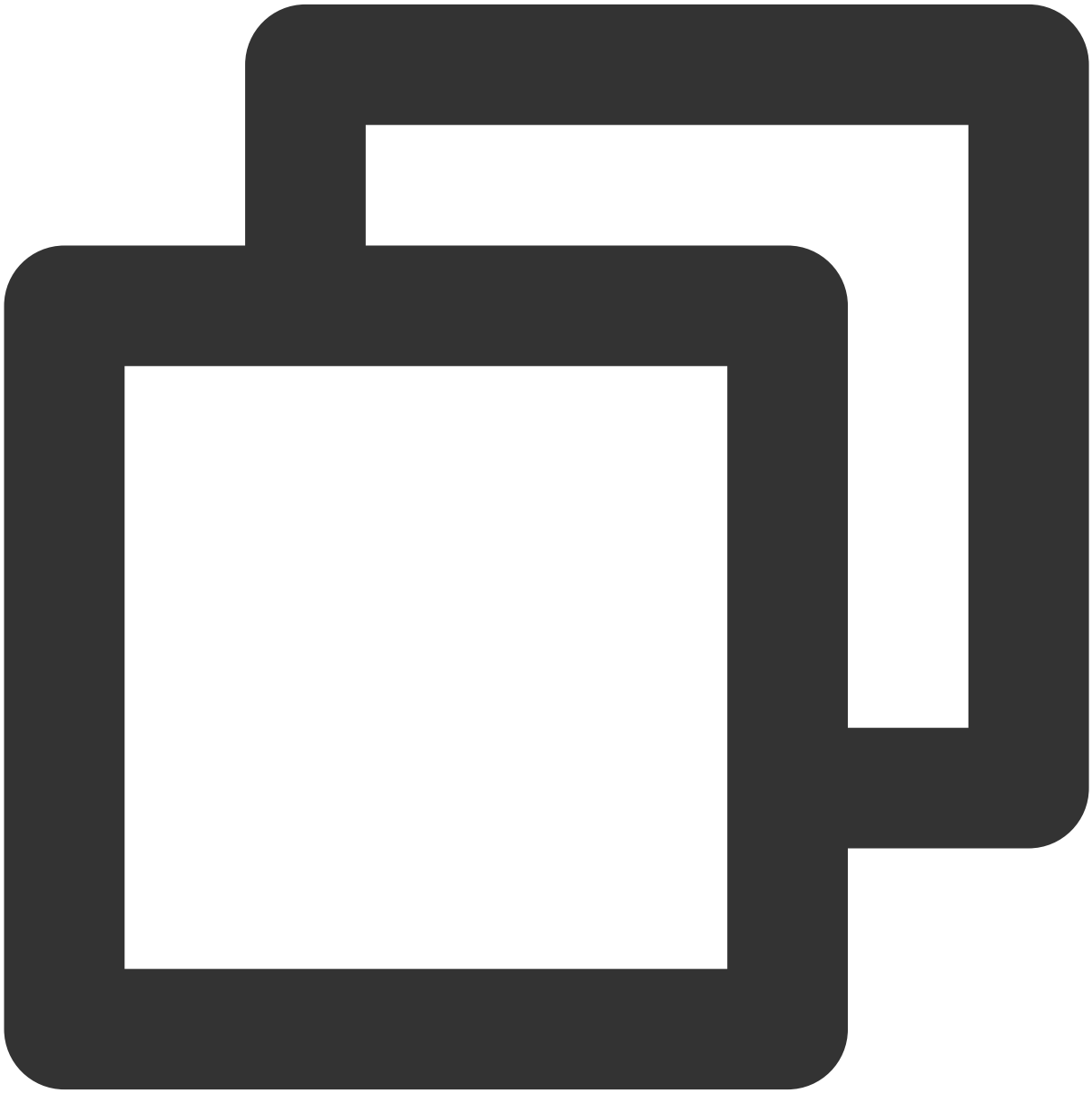
响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

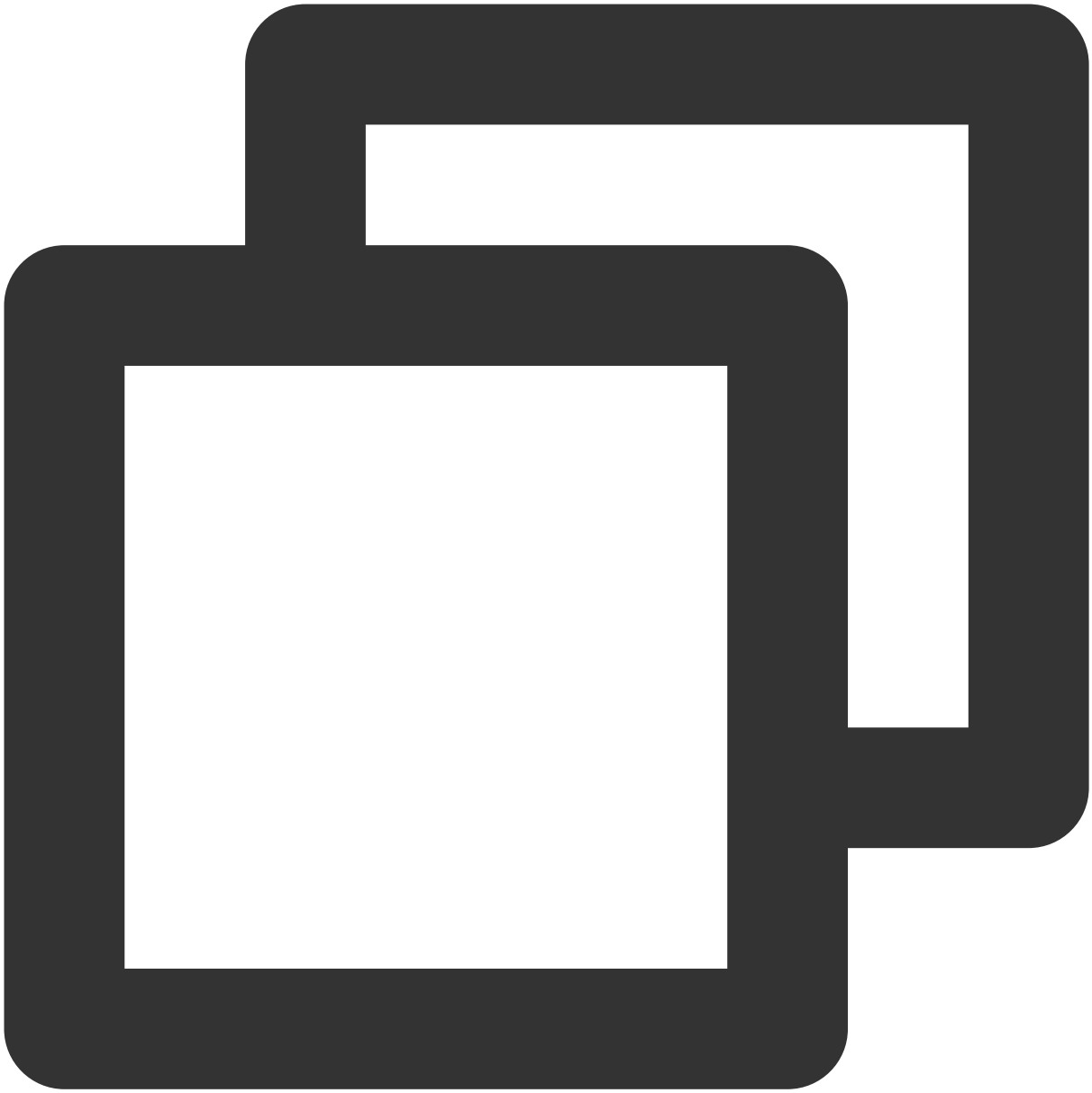
```
        </Message>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Running</State>
        <Progress>30</Progress>
        <CreationTime>2022-06-27T15:23:12+0800</CreationTime>
        <StartTime>2022-06-27T15:23:13+0800</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Transcode</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <RemoveWatermark>
                <Dx>150</Dx>
                <Dy>150</Dy>
                <Width>75</Width>
                <Height>75</Height>
            </RemoveWatermark>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
                <State>Running</State>
            </DigitalWatermark>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out.mp4</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求2：查询一个任务，该任务执行成功



```
GET /jobs/j9c0a4726f6ac11ec96aaa9b64ab18d00 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:25:22 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjY=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        </Message>
        <JobId>j9c0a4726f6ac11ec96aaa9b64ab18d00</JobId>
        <State>Success</State>
        <Progress>100</Progress>
        <CreationTime>2022-06-27T15:23:12+0800</CreationTime>
        <StartTime>2022-06-27T15:23:13+0800</StartTime>
        <EndTime>2022-06-27T15:24:33+0800</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Transcode</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <RemoveWatermark>
                <Dx>150</Dx>
                <Dy>150</Dy>
                <Width>75</Width>
                <Height>75</Height>
            </RemoveWatermark>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
                <State>Running</State>
            </DigitalWatermark>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out.mp4</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
            <MediaInfo>
                <Format>
                    <Bitrate>834.736000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
```

```
            <Size>1424687</Size>
            <StartTime>0.000000</StartTime>
        </Format>
        <Stream>
            <Audio>
                <Bitrate>104.047000</Bitrate>
                <Channel>2</Channel>
                <ChannelLayout>stereo</ChannelLayout>
                <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                <CodecName>aac</CodecName>
                <CodecTag>0x6134706d</CodecTag>
                <CodecTagString>mp4a</CodecTagString>
                <CodecTimeBase>1/44100</CodecTimeBase>
                <Duration>13.653311</Duration>
                <Index>1</Index>
                <Language>und</Language>
                <SampleFmt>fltp</SampleFmt>
                <SampleRate>44100</SampleRate>
                <StartTime>0.000000</StartTime>
                <Timebase>1/44100</Timebase>
            </Audio>
            <Subtitle/>
            <Video>
                <AvgFps>25.000000</AvgFps>
                <Bitrate>763.774000</Bitrate>
                <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                <CodecName>h264</CodecName>
                <CodecTag>0x31637661</CodecTag>
                <CodecTagString>avc1</CodecTagString>
                <CodecTimeBase>1/12800</CodecTimeBase>
                <ColorPrimaries>bt470bg</ColorPrimaries>
                <ColorRange>tv</ColorRange>
                <ColorTransfer>smpte170m</ColorTransfer>
                <Duration>12.960000</Duration>
                <Fps>25.000000</Fps>
                <HasBFrame>0</HasBFrame>
                <Height>960</Height>
                <Index>0</Index>
                <Language>und</Language>
                <Level>10</Level>
                <NumFrames>324</NumFrames>
                <PixFormat>yuv420p</PixFormat>
                <Profile>High</Profile>
                <RefFrames>1</RefFrames>
                <Rotation>0.000000</Rotation>
                <StartTime>0.000000</StartTime>
                <Timebase>1/12800</Timebase>
```


版权所有：腾讯云计算（北京）有限责任公司

```
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Region>ap-chongqing</Region>
                    <ObjectName>output/out.mp4</ObjectName>
                    <Md5Info>
                            <Md5>3df1f845d2ffd20a525a93ec40014d90</Md5>
                            <ObjectName>output/out.mp4</ObjectName>
                    </Md5Info>
                </OutputFile>
            </MediaResult>
        </Operation>
    </JobsDetail>
</Response>
```

DescribeMediaJobs

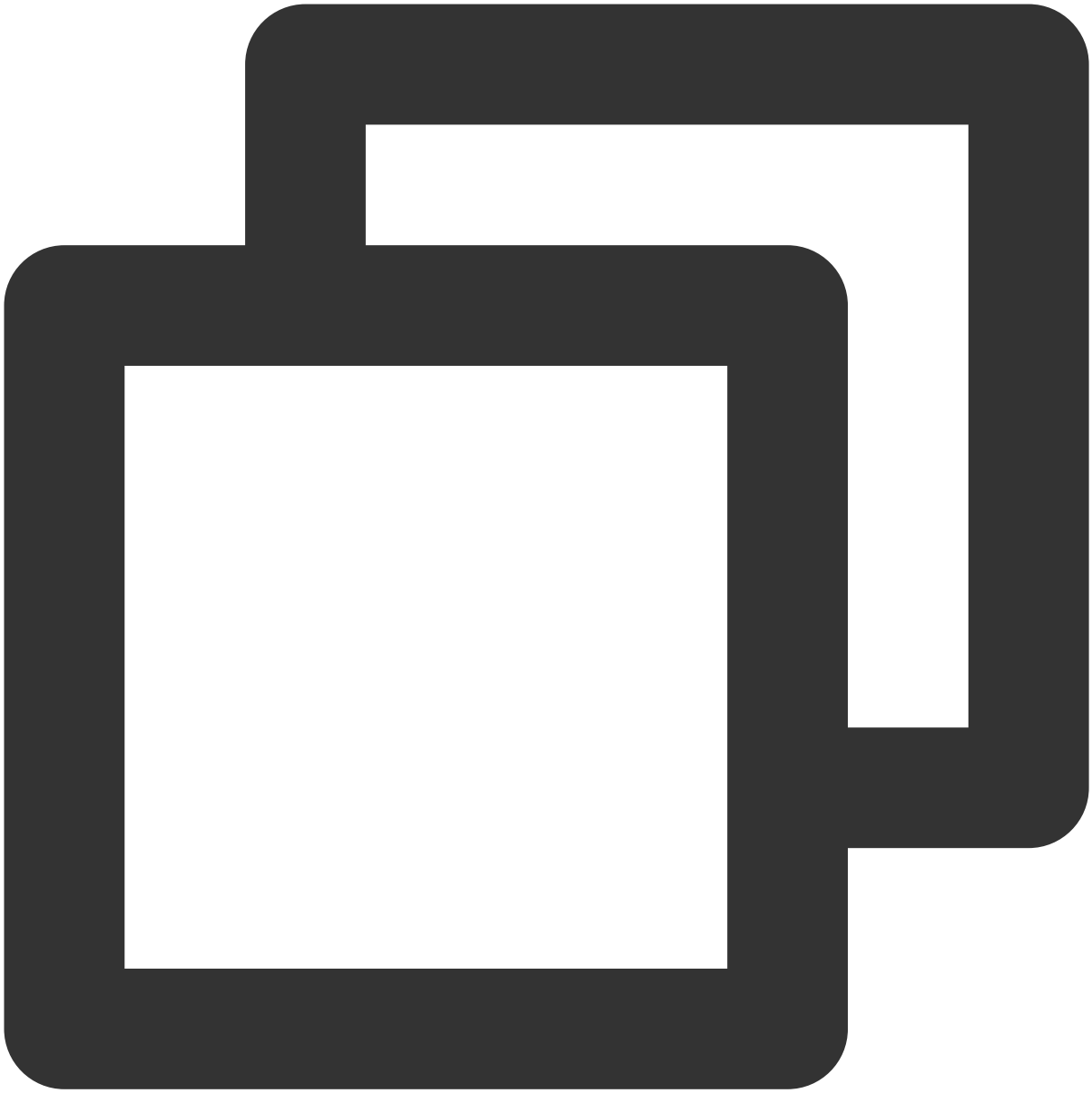
最近更新时间：2023-12-30 16:25:42

功能描述

DescribeMediaJobs 用于拉取符合条件的任务。

请求

请求示例



```
GET /jobs?size=&states=&queueId=&startCreationTime=&endCreationTime= HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	是否必选
queueId	无	拉取该队列 ID 下的任务	String	是
tag	无	任务的 Tag	String	是
workflowId	无	触发该任务的工作流 ID	String	否
inventoryTriggerJobId	无	触发该任务的存量触发任务 ID	String	否
inputObject	无	该任务的输入文件名，暂仅支持精确匹配	String	否
orderByTime	无	Desc 或者 Asc。默认为 Desc	String	否
nextToken	无	请求的上下文，用于翻页。上次返回的值	String	否
size	无	拉取的最大任务数。默认为10。最大为100	Integer	否
states	无	拉取该状态的任务，以 `,` 分割，支持多状态：All、Submitted、Running、Success、Failed、Pause、Cancel。默认为 All	String	否
startCreationTime	无	拉取创建时间大于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`，示例：2001-01-01T00:00:00+0800	String	否
endCreationTime	无	拉取创建时间小于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`，示例：2001-01-01T23:59:59+0800	String	否

tag 支持以下几种类型:

任务类型	tag
音视频转码	Transcode
视频转动图	Animation
视频截图	Snapshot
智能封面	SmartCover
音视频拼接	Concat

版权所有：腾讯云计算（北京）有限责任公司

人声分离	VoiceSeparate
精彩集锦	VideoMontage
SDR to HDR	SDRtoHDR
视频增强	VideoProcess
超级分辨率	SuperResolution
音视频转封装	Segment
数字水印	DigitalWatermark
提取数字水印	ExtractDigitalWatermark
视频标签	VideoTag
获取媒体信息	MediaInfo
音视频流分离	StreamExtract
视频质量分析	QualityEstimate
语音合成	Tts
音频降噪	NoiseReduction

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

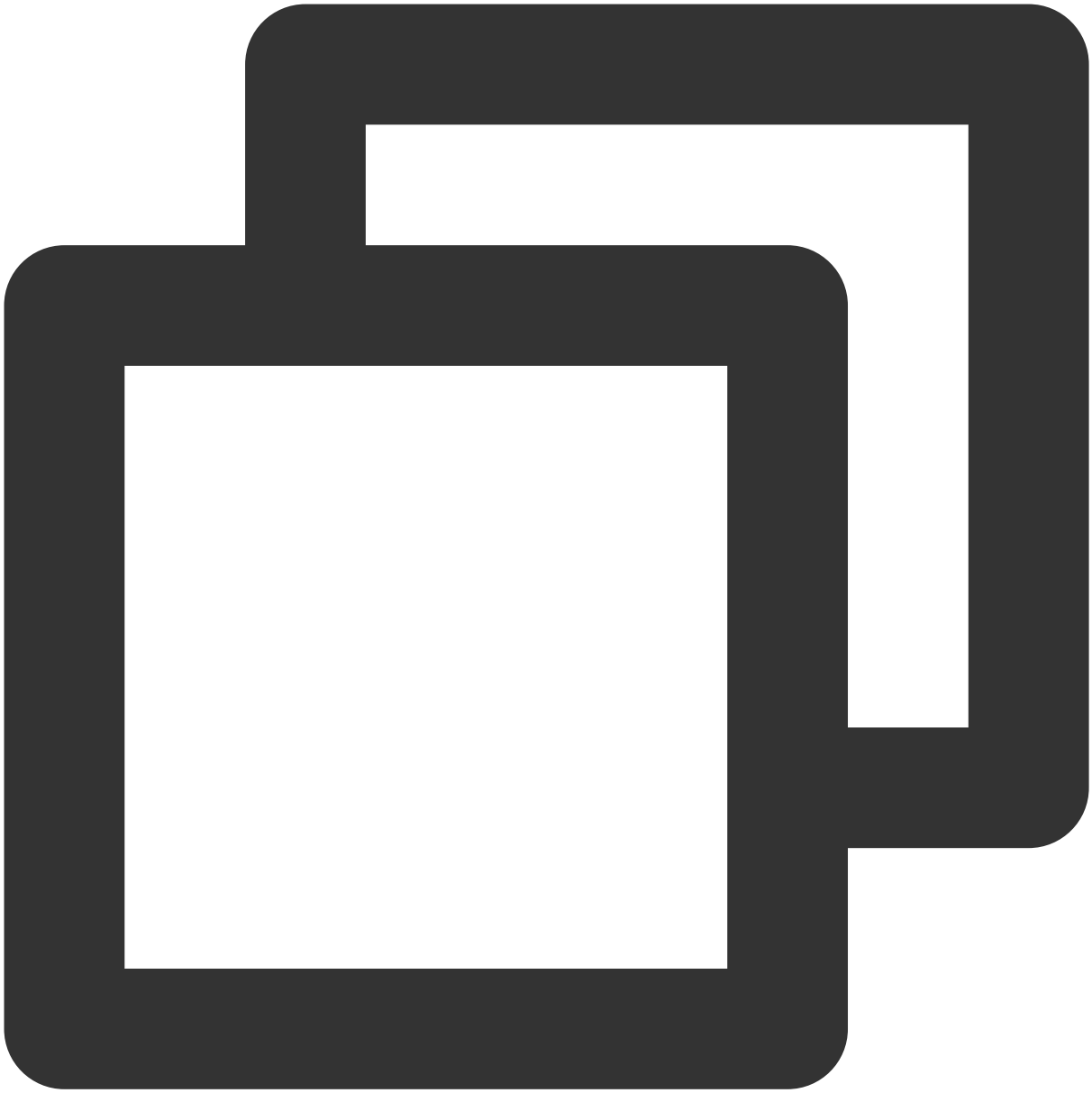
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        ...
    </JobsDetail>
    <NextToken></NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container数组
NextToken	Response	翻页的上下文 Token	String

对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：

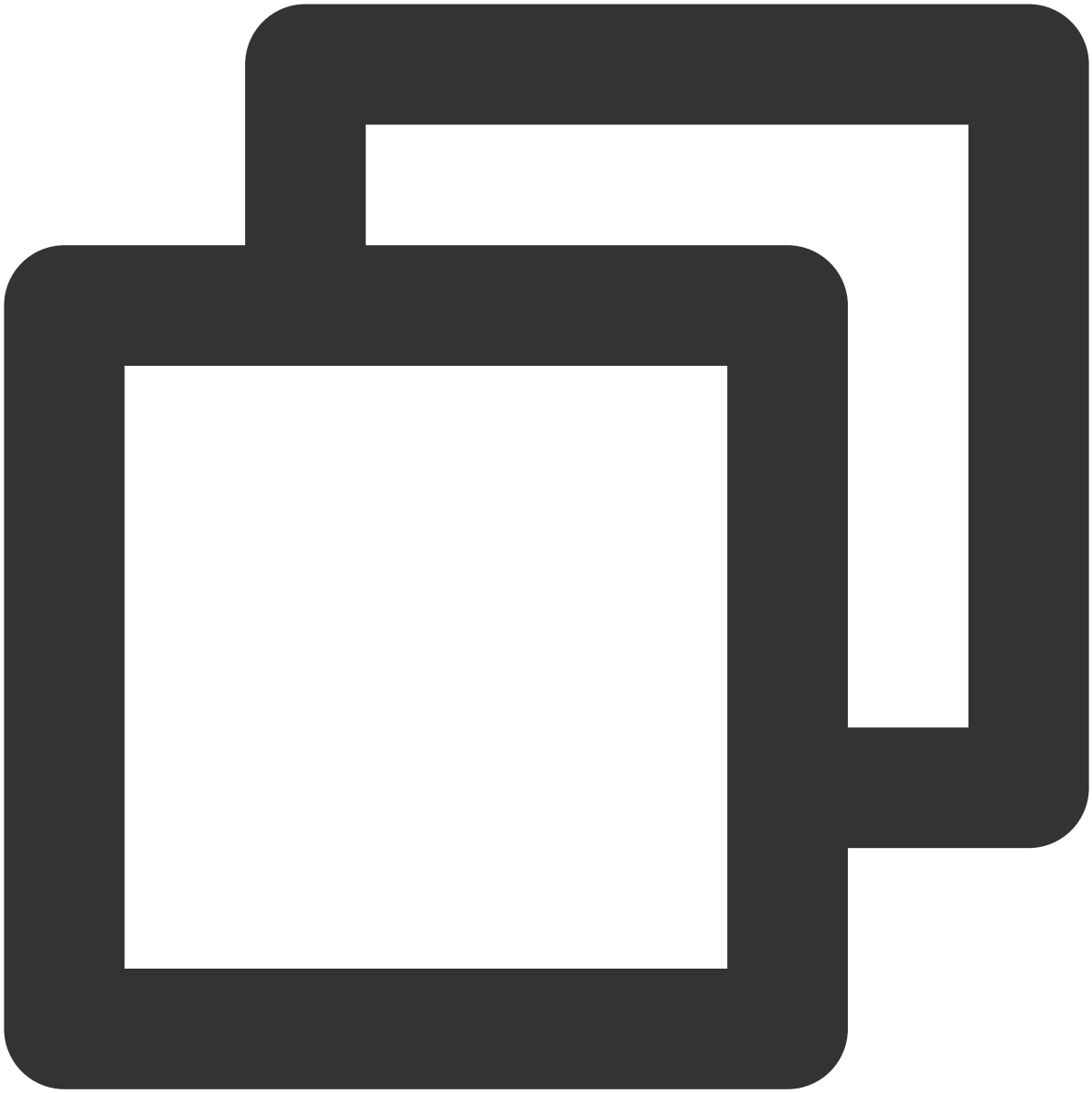
音视频转码
视频转动图
视频截帧
智能封面
音视频拼接
人声分离
精彩集锦
SDR to HDR
视频增强
超分辨率
音视频转封装
数字水印
提取数字水印
视频标签
获取媒体信息
音视频流分离
视频质量分析
语音合成
音频降噪

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

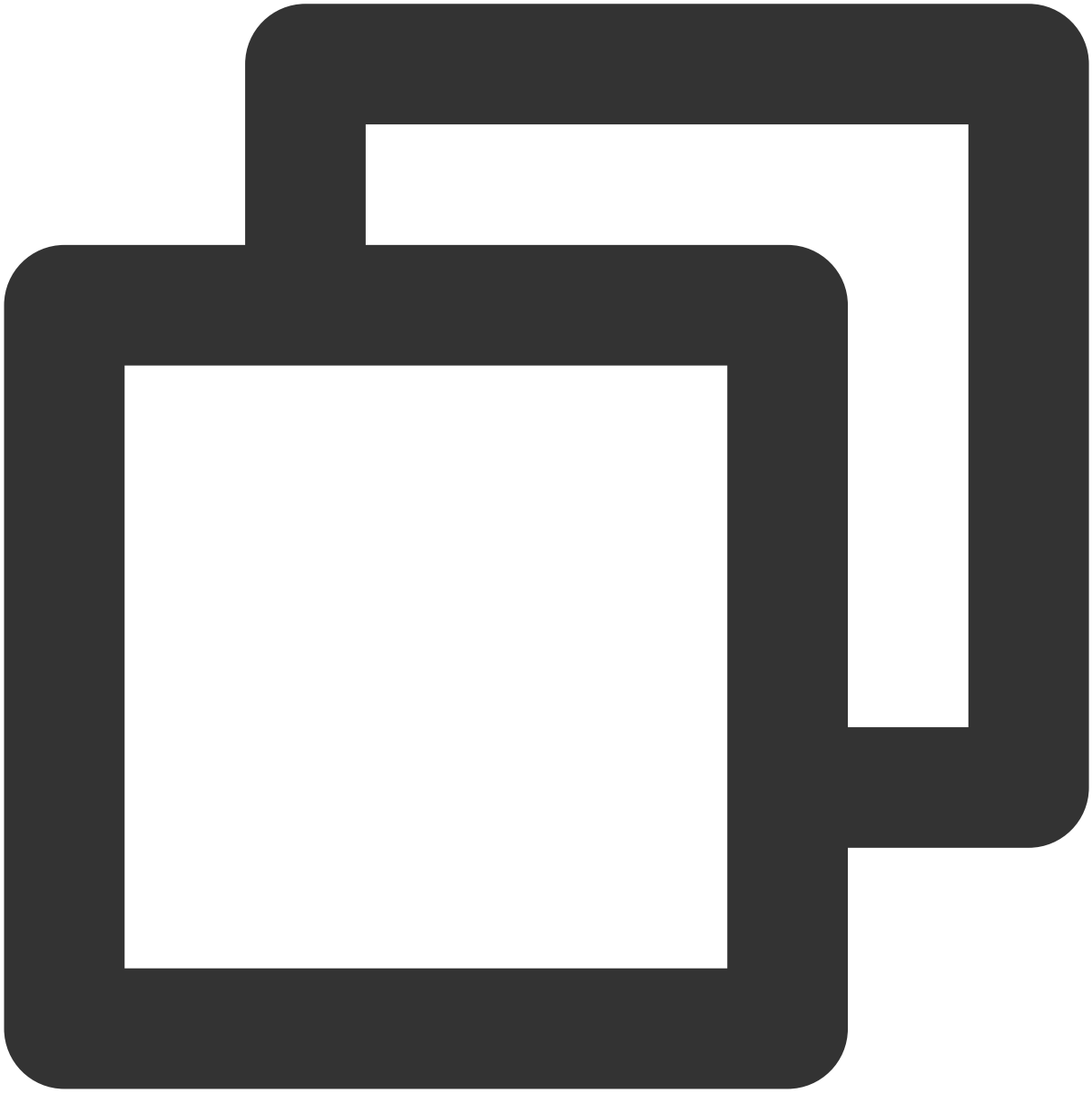
实际案例

请求



```
GET /jobs?queueId=p2242ab62c7c94486915508540933a2c6&tag=Transcode HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        </Message>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Running</State>
        <Progress>30</Progress>
        <CreationTime>2022-06-27T15:23:12+0800</CreationTime>
        <StartTime>2022-06-27T15:23:13+0800</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Transcode</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <RemoveWatermark>
                <Dx>150</Dx>
                <Dy>150</Dy>
                <Width>75</Width>
                <Height>75</Height>
            </RemoveWatermark>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
                <State>Running</State>
            </DigitalWatermark>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out.mp4</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
    <JobsDetail>
        <Code>Success</Code>
        </Message>
        <JobId>j9c0a4726f6ac11ec96aaa9b64ab18d00</JobId>
        <State>Success</State>
        <Progress>100</Progress>
```

```
        <CreationTime>2022-06-27T15:23:12+0800</CreationTime>
        <StartTime>2022-06-27T15:23:13+0800</StartTime>
        <EndTime>2022-06-27T15:24:33+0800</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Transcode</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <RemoveWatermark>
                <Dx>150</Dx>
                <Dy>150</Dy>
                <Width>75</Width>
                <Height>75</Height>
            </RemoveWatermark>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
                <State>Running</State>
            </DigitalWatermark>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out.mp4</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
            <MediaInfo>
                <Format>
                    <Bitrate>834.736000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>1424687</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
```

```
                <Audio>
                    <Bitrate>104.047000</Bitrate>
                    <Channel>2</Channel>
                    <ChannelLayout>stereo</ChannelLayout>
                    <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                    <CodecName>aac</CodecName>
                    <CodecTag>0x6134706d</CodecTag>
                    <CodecTagString>mp4a</CodecTagString>
                    <CodecTimeBase>1/44100</CodecTimeBase>
                    <Duration>13.653311</Duration>
                    <Index>1</Index>
                    <Language>und</Language>
                    <SampleFmt>fltp</SampleFmt>
                    <SampleRate>44100</SampleRate>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/44100</Timebase>
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>25.000000</AvgFps>
                    <Bitrate>763.774000</Bitrate>
                    <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                    <CodecName>h264</CodecName>
                    <CodecTag>0x31637661</CodecTag>
                    <CodecTagString>avc1</CodecTagString>
                    <CodecTimeBase>1/12800</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>12.960000</Duration>
                    <Fps>25.000000</Fps>
                    <HasBFrame>0</HasBFrame>
                    <Height>960</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>10</Level>
                    <NumFrames>324</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>High</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>544</Width>
                </Video>
            </Stream>
        </MediaInfo>
```

```
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Region>ap-chongqing</Region>
                    <ObjectName>output/out.mp4</ObjectName>
                    <Md5Info>
                            <Md5>3df1f845d2ffd20a525a93ec40014d90</Md5>
                            <ObjectName>output/out.mp4</ObjectName>
                    </Md5Info>
                </OutputFile>
            </MediaResult>
        </Operation>
    </JobsDetail>
    <NextToken>225508</NextToken>
</Response>
```

媒体处理任务回调
视频转动图

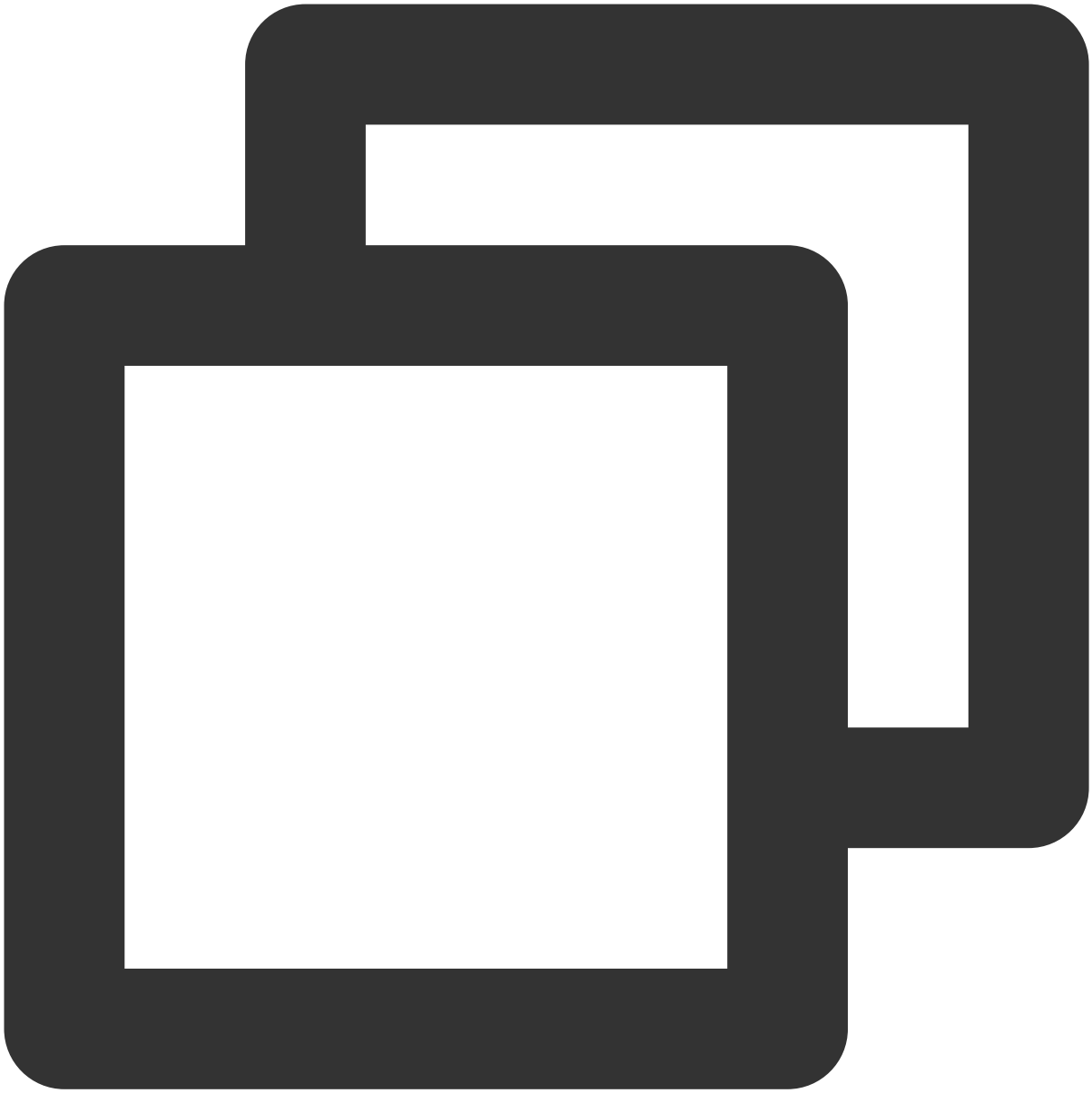
最近更新时间：2023-12-30 16:25:32

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>97624.424000</Bitrate>
                    <Duration>0.870000</Duration>
                    <FormatLongName>CompuServe Graphics Interchange Format (GIF)</F
                    <FormatName>gif</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>10616656</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video>
                        <AvgFps>14.666667</AvgFps>
                        <CodecLongName>CompuServe GIF (Graphics Interchange Format)
                        <CodecName>gif</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/100</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>0.870000</Duration>
                        <Fps>15.083333</Fps>
                        <HasBFrame>0</HasBFrame>
                        <Height>2258</Height>
                        <Index>0</Index>
                        <Level>-99</Level>
                        <NumFrames>7</NumFrames>
                        <PixFormat>bgra</PixFormat>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/100</Timebase>
                        <Width>1280</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
```

```
                        <Md5>3df1f845d2ffd20a525a93ec40014d90</Md5>
                        <ObjectName>output/animation.gif</ObjectName>
                    </Md5Info>
                    <ObjectName>output/animation.gif</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/animation.gif</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>template_name1512456</TemplateName>
            <UserData>This is my Animation job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>Animation</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交动图任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

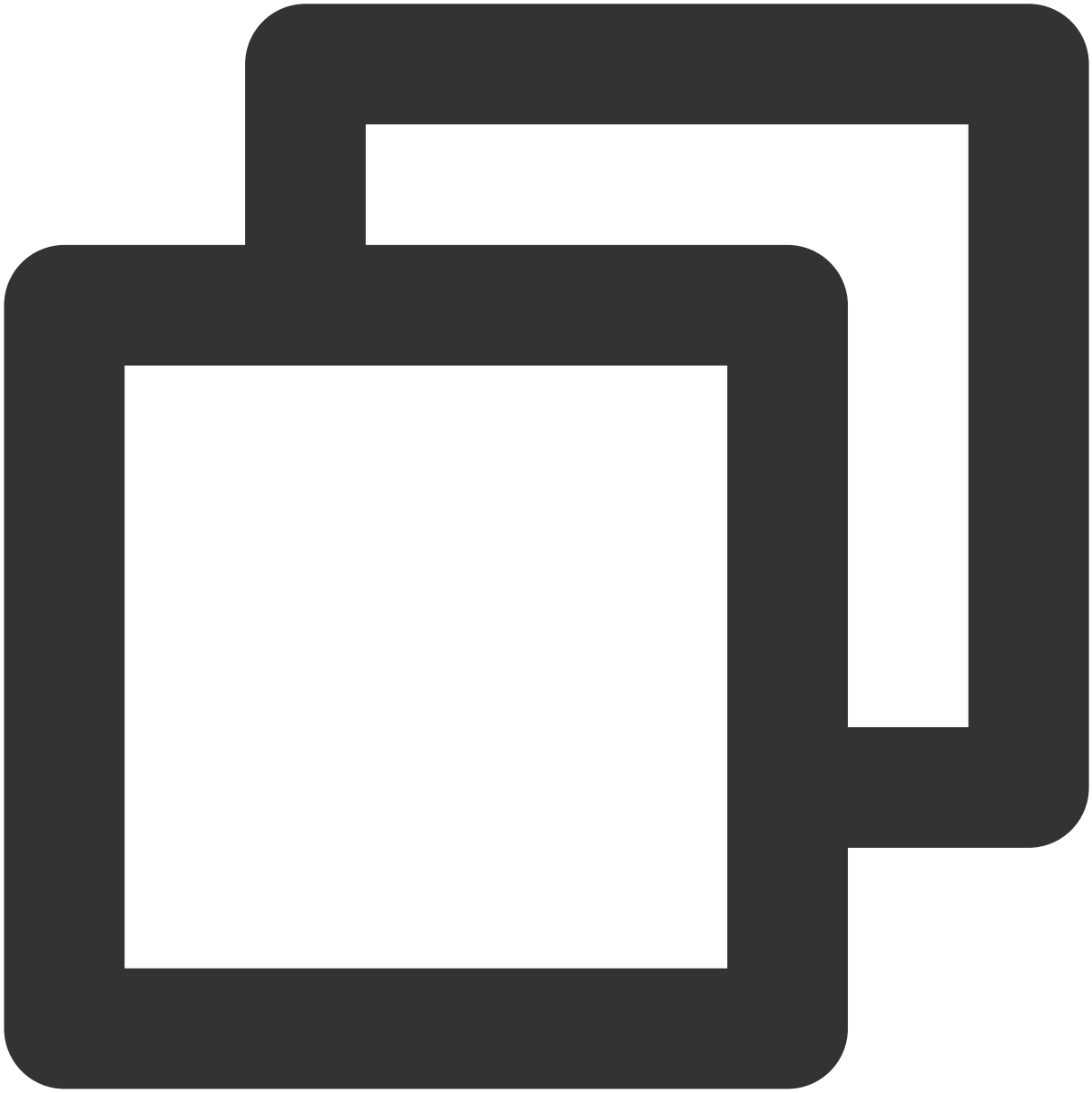
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

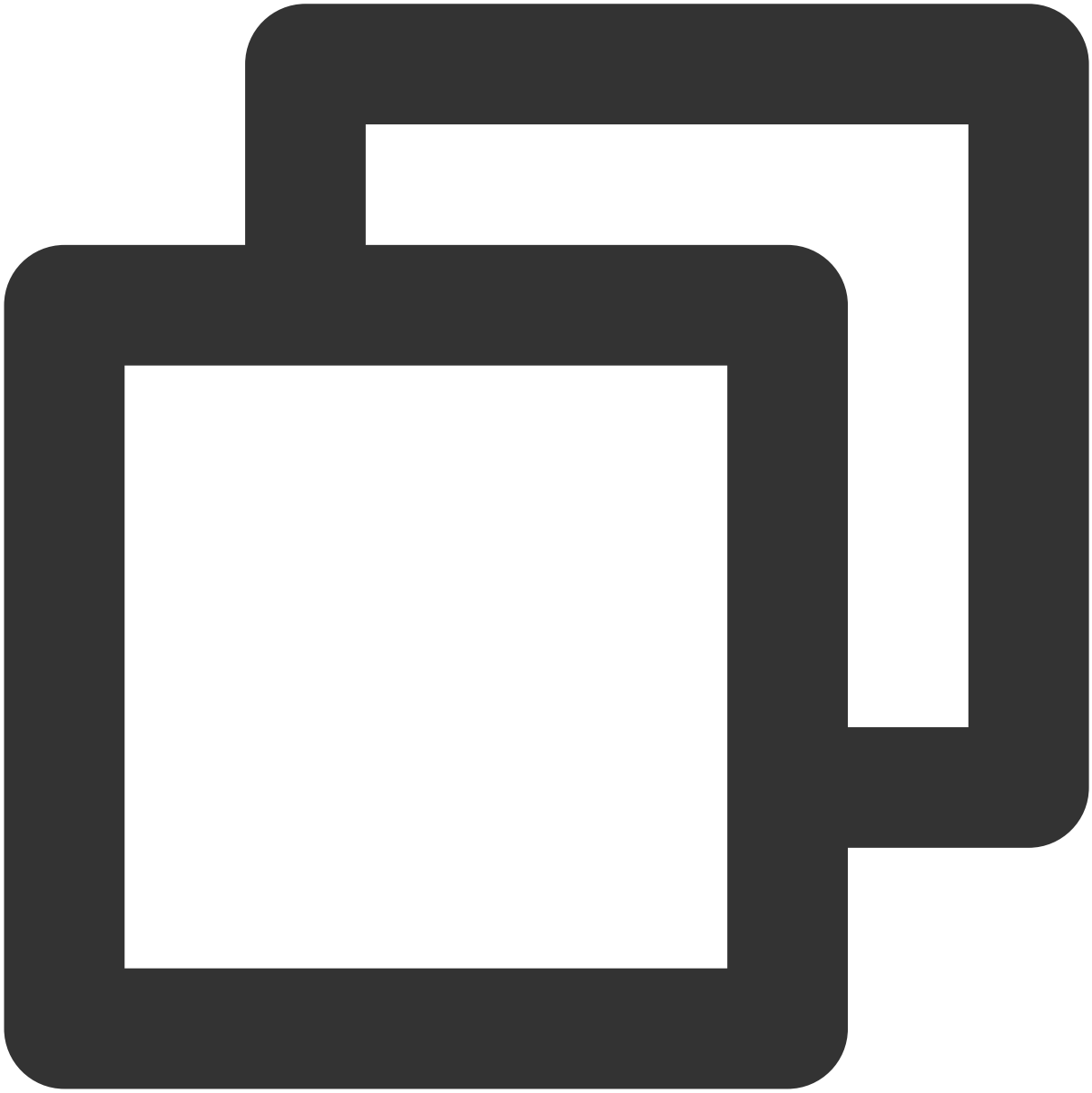


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>97624.424000</Bitrate>
                    <Duration>0.870000</Duration>
                    <FormatLongName>CompuServe Graphics Interchange Format (GIF)</F
                    <FormatName>gif</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>10616656</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video>
                        <AvgFps>14.666667</AvgFps>
                        <CodecLongName>CompuServe GIF (Graphics Interchange Format)
                        <CodecName>gif</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/100</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>0.870000</Duration>
                        <Fps>15.083333</Fps>
                        <HasBFrame>0</HasBFrame>
                        <Height>2258</Height>
                        <Index>0</Index>
                        <Level>-99</Level>
                        <NumFrames>7</NumFrames>
                        <PixFormat>bgra</PixFormat>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/100</Timebase>
                        <Width>1280</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
```

```
                        <Md5>3df1f845d2ffd20a525a93ec40014d90</Md5>
                        <ObjectName>output/animation.gif</ObjectName>
                    </Md5Info>
                    <ObjectName>output/animation.gif</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/animation.gif</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>template_name1512456</TemplateName>
            <UserData>This is my Animation job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>Animation</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调

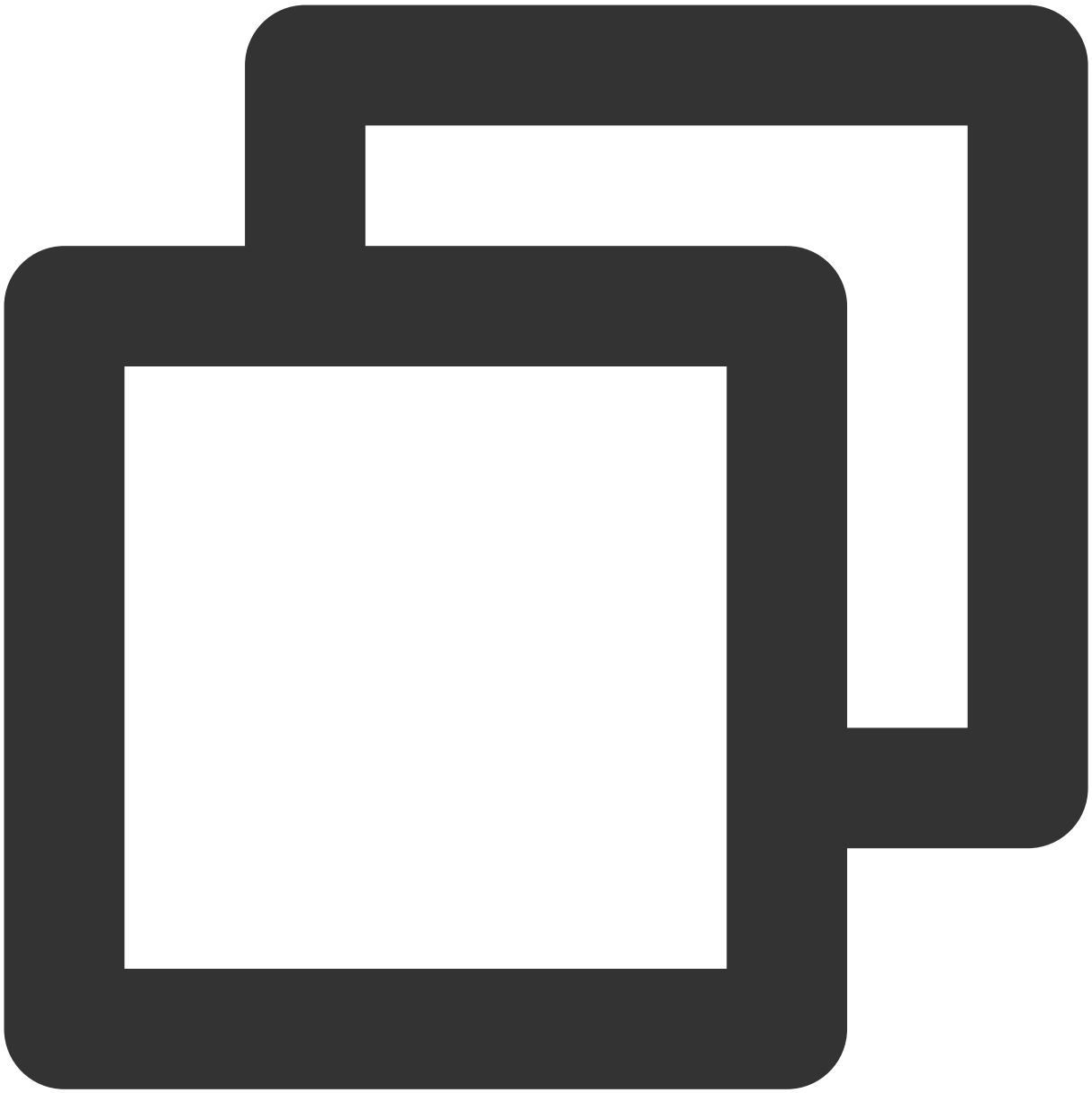


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>97624.424000</Bitrate>
                    <Duration>0.870000</Duration>
                    <FormatLongName>CompuServe Graphics Interchange Format (GIF)</F
                    <FormatName>gif</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>10616656</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video>
                        <AvgFps>14.666667</AvgFps>
                        <CodecLongName>CompuServe GIF (Graphics Interchange Format)
                        <CodecName>gif</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/100</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>0.870000</Duration>
                        <Fps>15.083333</Fps>
                        <HasBFrame>0</HasBFrame>
                        <Height>2258</Height>
```

```
                        <Index>0</Index>
                        <Level>-99</Level>
                        <NumFrames>7</NumFrames>
                        <PixFormat>bgra</PixFormat>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/100</Timebase>
                        <Width>1280</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>3df1f845d2ffd20a525a93ec40014d90</Md5>
                        <ObjectName>output/animation.gif</ObjectName>
                    </Md5Info>
                    <ObjectName>output/animation.gif</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/animation.gif</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>template_name1512456</TemplateName>
            <UserData>This is my Animation job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>Animation</Tag>
        <Workflow>
            <Name>Animation_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T15:29:42+0800",
        "EndTime": "2022-06-30T15:29:48+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
```

```
            "Region": "ap-chongqing",
            "CosHeaders": [{
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j682b9662f84611ecb8546d80f2baf56f",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "97624.424000",
                    "Duration": "0.870000",
                    "FormatLongName": "CompuServe Graphics Interchange Format (GIF)
                    "FormatName": "gif",
                    "NumProgram": "0",
                    "NumStream": "1",
                    "Size": "10616656",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Video": {
                        "AvgFps": "14.666667",
                        "CodecLongName": "CompuServe GIF (Graphics Interchange Form
                        "CodecName": "gif",
                        "CodecTag": "0x0000",
                        "CodecTagString": "[0][0][0][0]",
                        "CodecTimeBase": "1/100",
                        "ColorPrimaries": "unknown",
                        "ColorRange": "unknown",
                        "ColorTransfer": "unknown",
                        "Duration": "0.870000",
                        "Fps": "15.083333",
                        "HasBFrame": "0",
                        "Height": "2258",
```

```
                        "Index": "0",
                        "Level": "-99",
                        "NumFrames": "7",
                        "PixFormat": "bgra",
                        "RefFrames": "1",
                        "Rotation": "0.000000",
                        "StartTime": "0.000000",
                        "Timebase": "1/100",
                        "Width": "1280"
                    }
                }
            },
            "MediaResult": {
                "OutputFile": {
                    "Bucket": "test-123456789",
                    "Md5Info": {
                        "Md5": "3df1f845d2ffd20a525a93ec40014d90",
                        "ObjectName": "output/animation.gif"
                    },
                    "ObjectName": "output/animation.gif",
                    "Region": "ap-chongqing"
                }
            },
            "Output": {
                "Bucket": "test-123456789",
                "Object": "output/animation.gif",
                "Region": "ap-chongqing"
            },
            "TemplateId": "t1f16e1dfbdc994105b31292d45710642a",
            "TemplateName": "template_name1512456",
            "UserData": "This is my Animation job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T15:29:43+0800",
        "State": "Success",
        "Tag": "Animation",
        "Workflow": {
            "Name": "Animation_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

音视频拼接

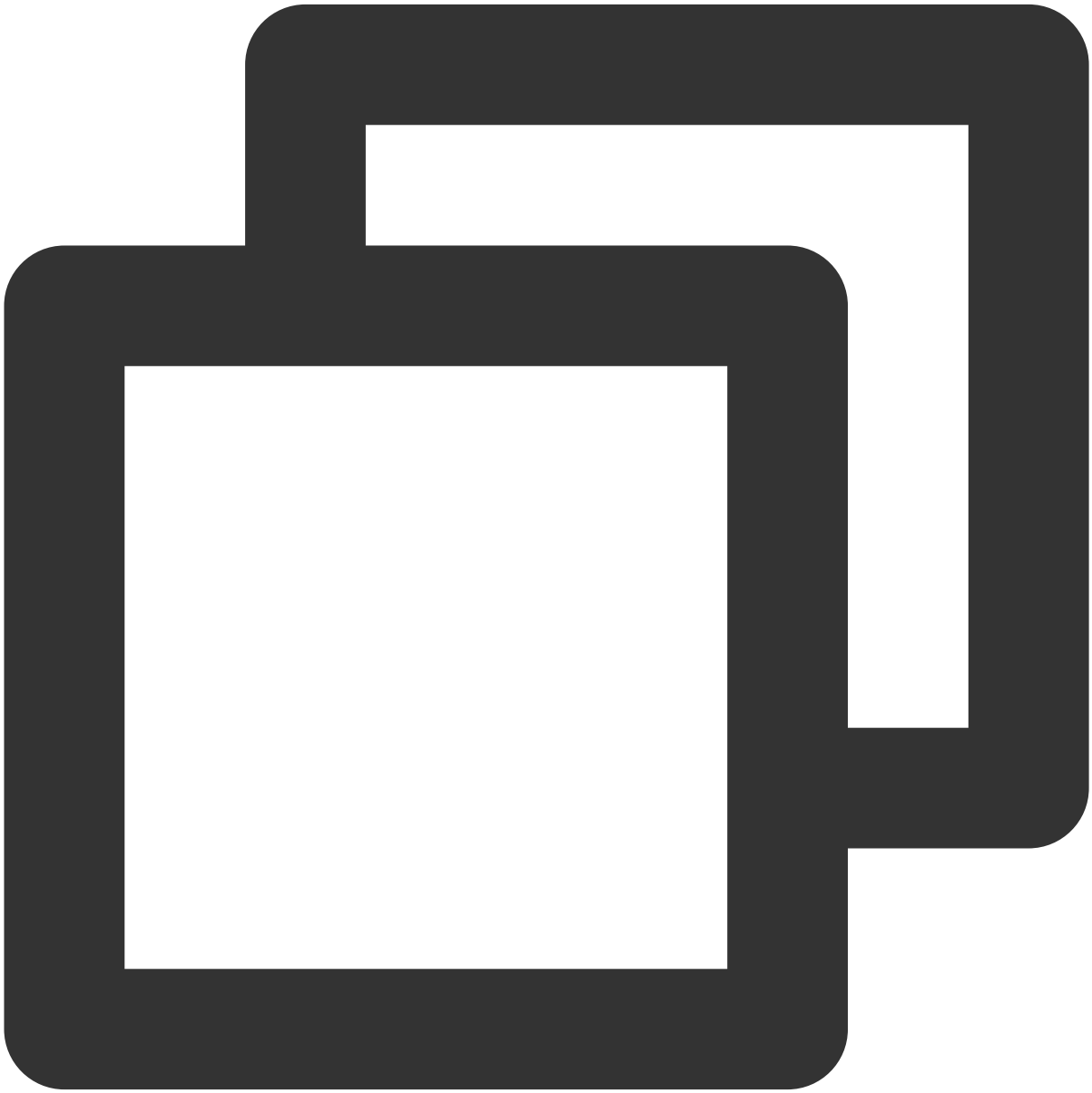
最近更新时间：2023-12-30 16:26:05

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:33:07+0800</CreationTime>
        <EndTime>2022-06-30T18:33:20+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j07f6263af86011ecbdf419856486c0ab</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>830.482000</Bitrate>
                    <Duration>70.413000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>7309593</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>32.020000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>70.412857</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>14.990016</AvgFps>
                        <Bitrate>803.112000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/90000</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Dar>4:3</Dar>
                        <Duration>69.779778</Duration>
                        <Fps>15.000000</Fps>
```

```
                        <HasBFrame>2</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>32</Level>
                        <NumFrames>1046</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <Sar>1:1</Sar>
                        <StartTime>0.046000</StartTime>
                        <Timebase>1/90000</Timebase>
                        <Width>1280</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>96322f8638d38480683e069e6b2ba1e9</Md5>
                        <ObjectName>output/concat.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/concat.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/concat.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
            <TemplateName>template_name3544697</TemplateName>
            <UserData>This is my Concat job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:33:08+0800</StartTime>
        <State>Success</State>
        <Tag>Concat</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交拼接任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

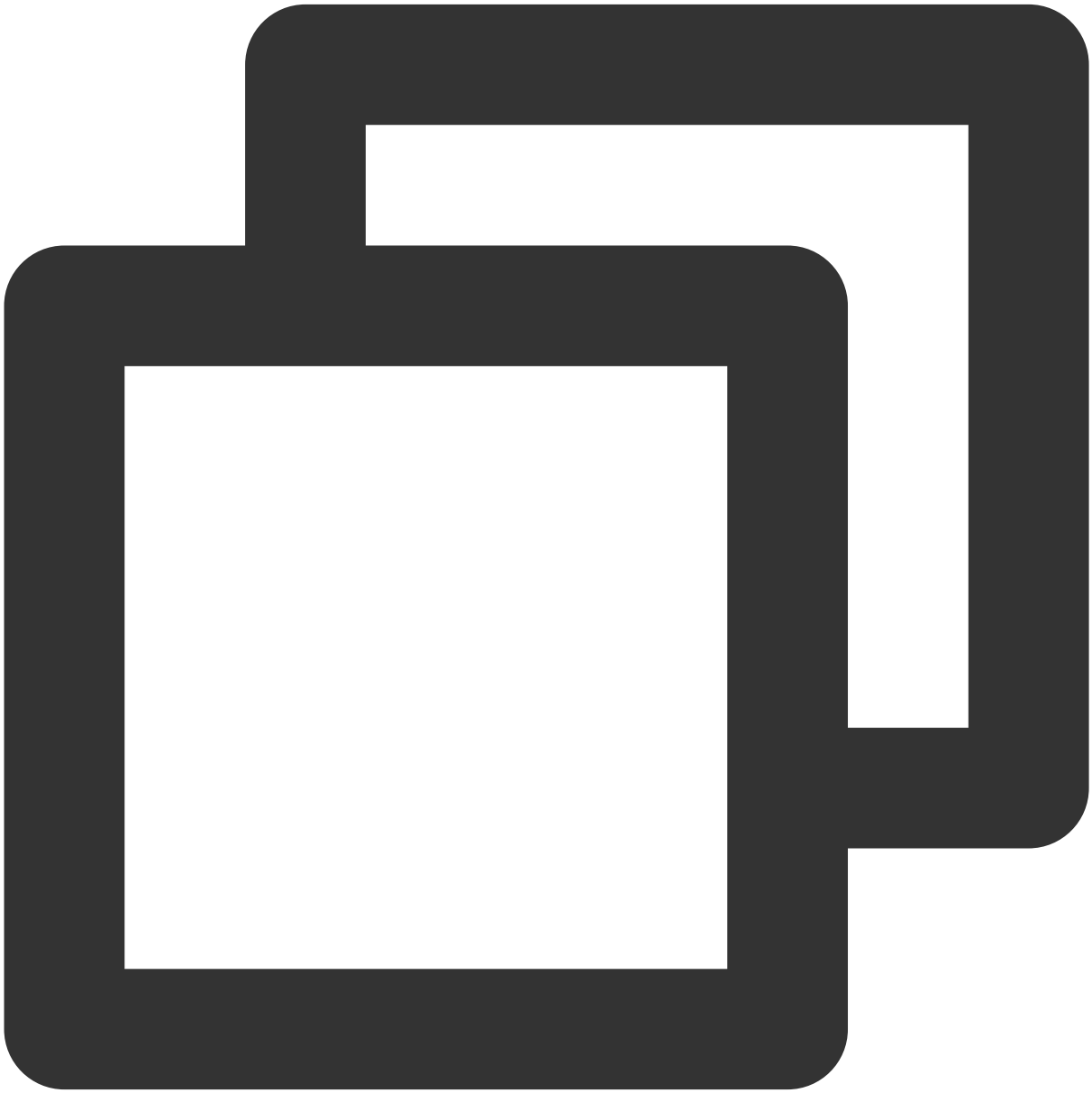
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

版权所有：腾讯云计算（北京）有限责任公司

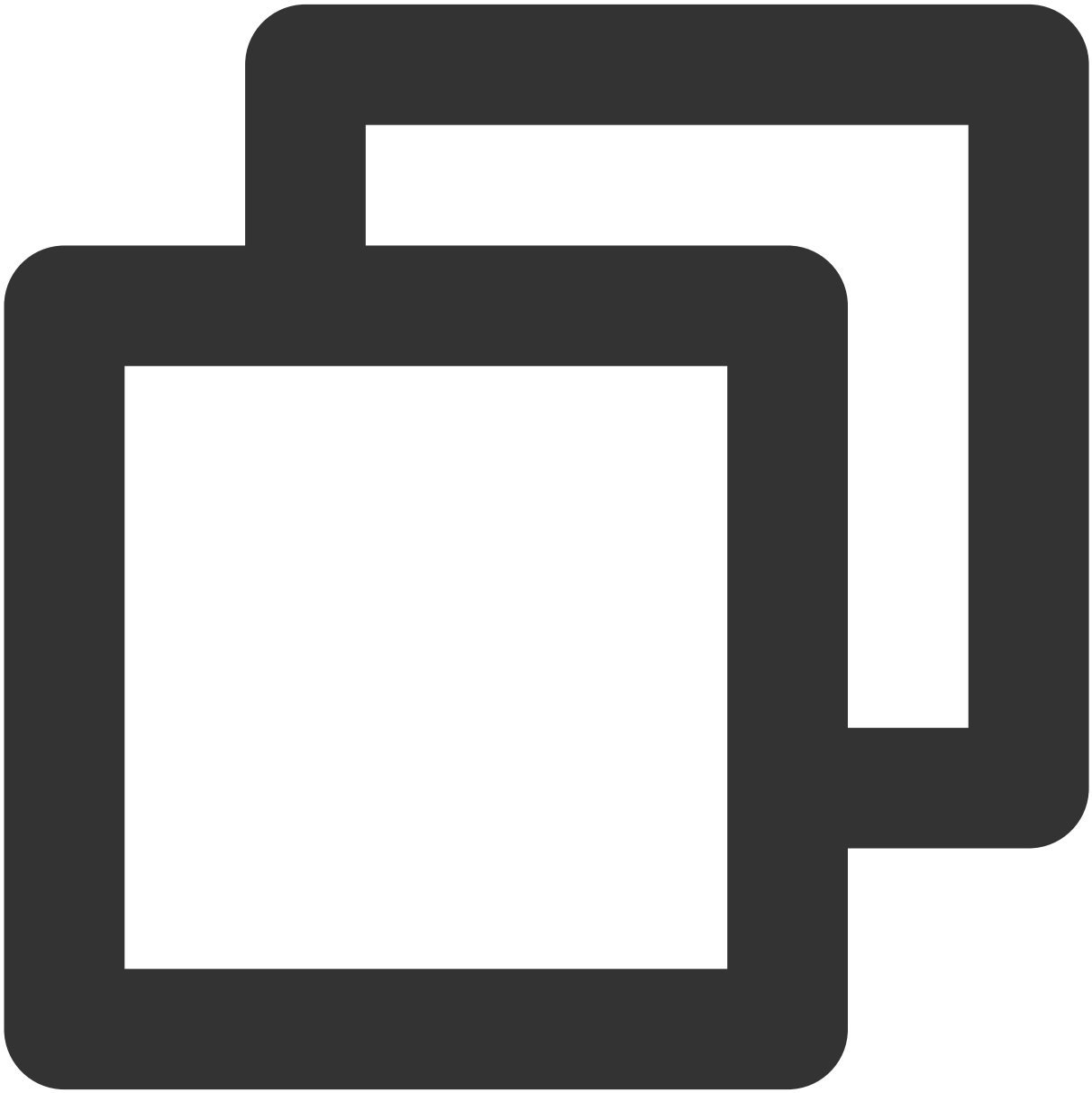


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:33:07+0800</CreationTime>
        <EndTime>2022-06-30T18:33:20+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j07f6263af86011ecbdf419856486c0ab</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>830.482000</Bitrate>
                    <Duration>70.413000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>7309593</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>32.020000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>70.412857</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>14.990016</AvgFps>
                        <Bitrate>803.112000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/90000</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Dar>4:3</Dar>
                        <Duration>69.779778</Duration>
                        <Fps>15.000000</Fps>
```

```
                        <HasBFrame>2</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>32</Level>
                        <NumFrames>1046</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <Sar>1:1</Sar>
                        <StartTime>0.046000</StartTime>
                        <Timebase>1/90000</Timebase>
                        <Width>1280</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>96322f8638d38480683e069e6b2ba1e9</Md5>
                        <ObjectName>output/concat.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/concat.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/concat.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
            <TemplateName>template_name3544697</TemplateName>
            <UserData>This is my Concat job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:33:08+0800</StartTime>
        <State>Success</State>
        <Tag>Concat</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:33:07+0800</CreationTime>
        <EndTime>2022-06-30T18:33:20+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

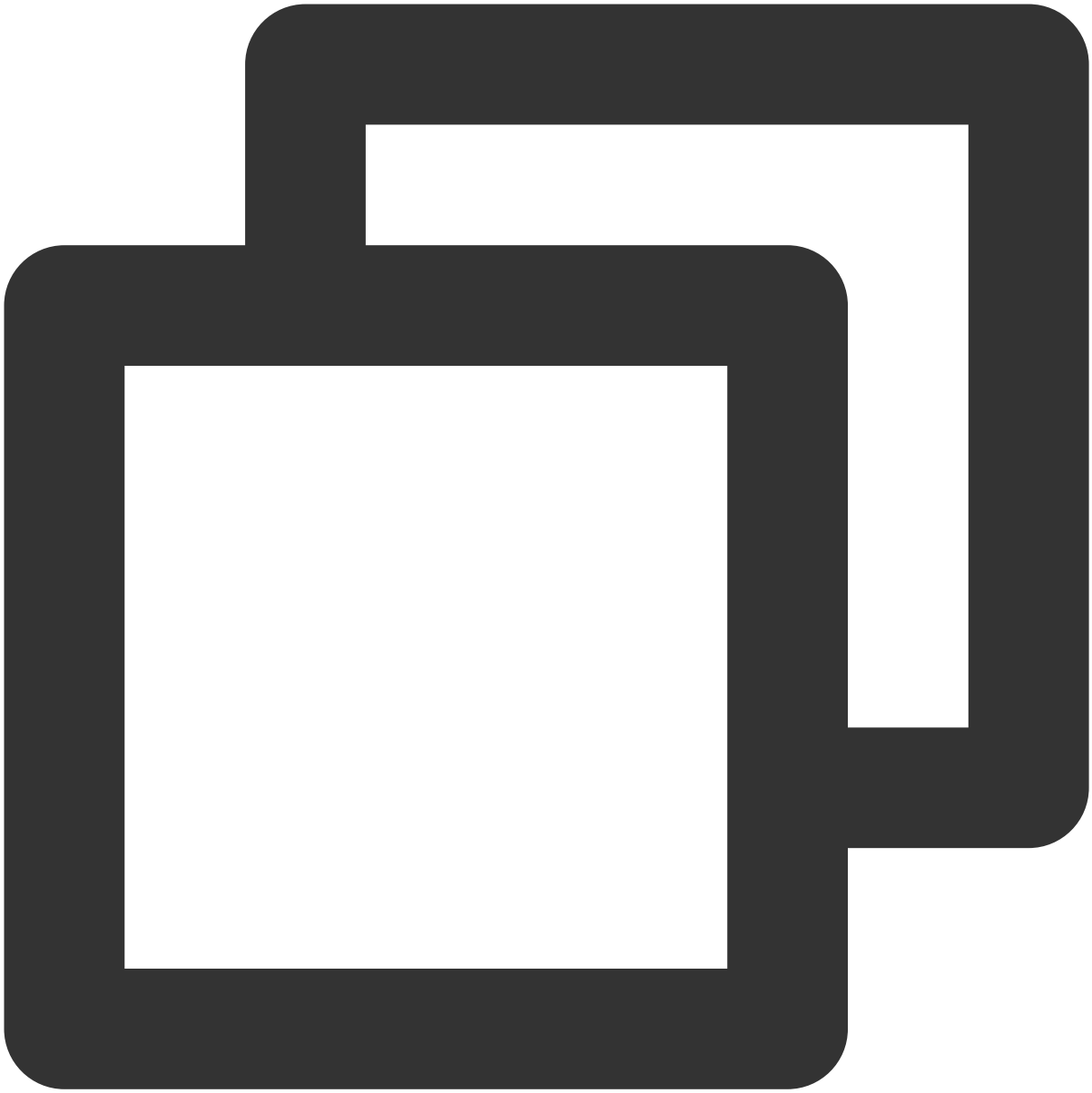
```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j07f6263af86011ecbdf419856486c0ab</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>830.482000</Bitrate>
                    <Duration>70.413000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>7309593</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>32.020000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>70.412857</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
```

```
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>14.990016</AvgFps>
                    <Bitrate>803.112000</Bitrate>
                    <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                    <CodecName>h264</CodecName>
                    <CodecTag>0x31637661</CodecTag>
                    <CodecTagString>avc1</CodecTagString>
                    <CodecTimeBase>1/90000</CodecTimeBase>
                    <ColorPrimaries>unknown</ColorPrimaries>
                    <ColorRange>unknown</ColorRange>
                    <ColorTransfer>unknown</ColorTransfer>
                    <Dar>4:3</Dar>
                    <Duration>69.779778</Duration>
                    <Fps>15.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>960</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>32</Level>
                    <NumFrames>1046</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>High</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <Sar>1:1</Sar>
                    <StartTime>0.046000</StartTime>
                    <Timebase>1/90000</Timebase>
                    <Width>1280</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>96322f8638d38480683e069e6b2ba1e9</Md5>
                    <ObjectName>output/concat.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/concat.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
```


版权所有：腾讯云计算（北京）有限责任公司

```
                <Object>output/concat.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
            <TemplateName>template_name3544697</TemplateName>
            <UserData>This is my Concat job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:33:08+0800</StartTime>
        <State>Success</State>
        <Tag>Concat</Tag>
        <Workflow>
            <Name>Concat_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T18:33:07+0800",
        "EndTime": "2022-06-30T18:33:20+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j07f6263af86011ecbdf419856486c0ab",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "830.482000",
                    "Duration": "70.413000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "7309593",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "32.020000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/44100",
                        "Duration": "70.412857",
                        "Index": "1",
                        "Language": "und",
                        "SampleFmt": "fltp",
                        "SampleRate": "44100",
                        "StartTime": "0.000000",
                        "Timebase": "1/44100"
```

```
                },
                "Video": {
                    "AvgFps": "14.990016",
                    "Bitrate": "803.112000",
                    "CodecLongName": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10
                    "CodecName": "h264",
                    "CodecTag": "0x31637661",
                    "CodecTagString": "avc1",
                    "CodecTimeBase": "1/90000",
                    "ColorPrimaries": "unknown",
                    "ColorRange": "unknown",
                    "ColorTransfer": "unknown",
                    "Dar": "4:3",
                    "Duration": "69.779778",
                    "Fps": "15.000000",
                    "HasBFrame": "2",
                    "Height": "960",
                    "Index": "0",
                    "Language": "und",
                    "Level": "32",
                    "NumFrames": "1046",
                    "PixFormat": "yuv420p",
                    "Profile": "High",
                    "RefFrames": "1",
                    "Rotation": "0.000000",
                    "Sar": "1:1",
                    "StartTime": "0.046000",
                    "Timebase": "1/90000",
                    "Width": "1280"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": {
                    "Md5": "96322f8638d38480683e069e6b2ba1e9",
                    "ObjectName": "output/concat.mp4"
                },
                "ObjectName": "output/concat.mp4",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/concat.mp4",
            "Region": "ap-chongqing"
```

```
            },
            "TemplateId": "t140325e8918ac423da53b7d78dbbab564",
            "TemplateName": "template_name3544697",
            "UserData": "This is my Concat job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T18:33:08+0800",
        "State": "Success",
        "Tag": "Concat",
        "Workflow": {
            "Name": "Concat_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

添加数字水印

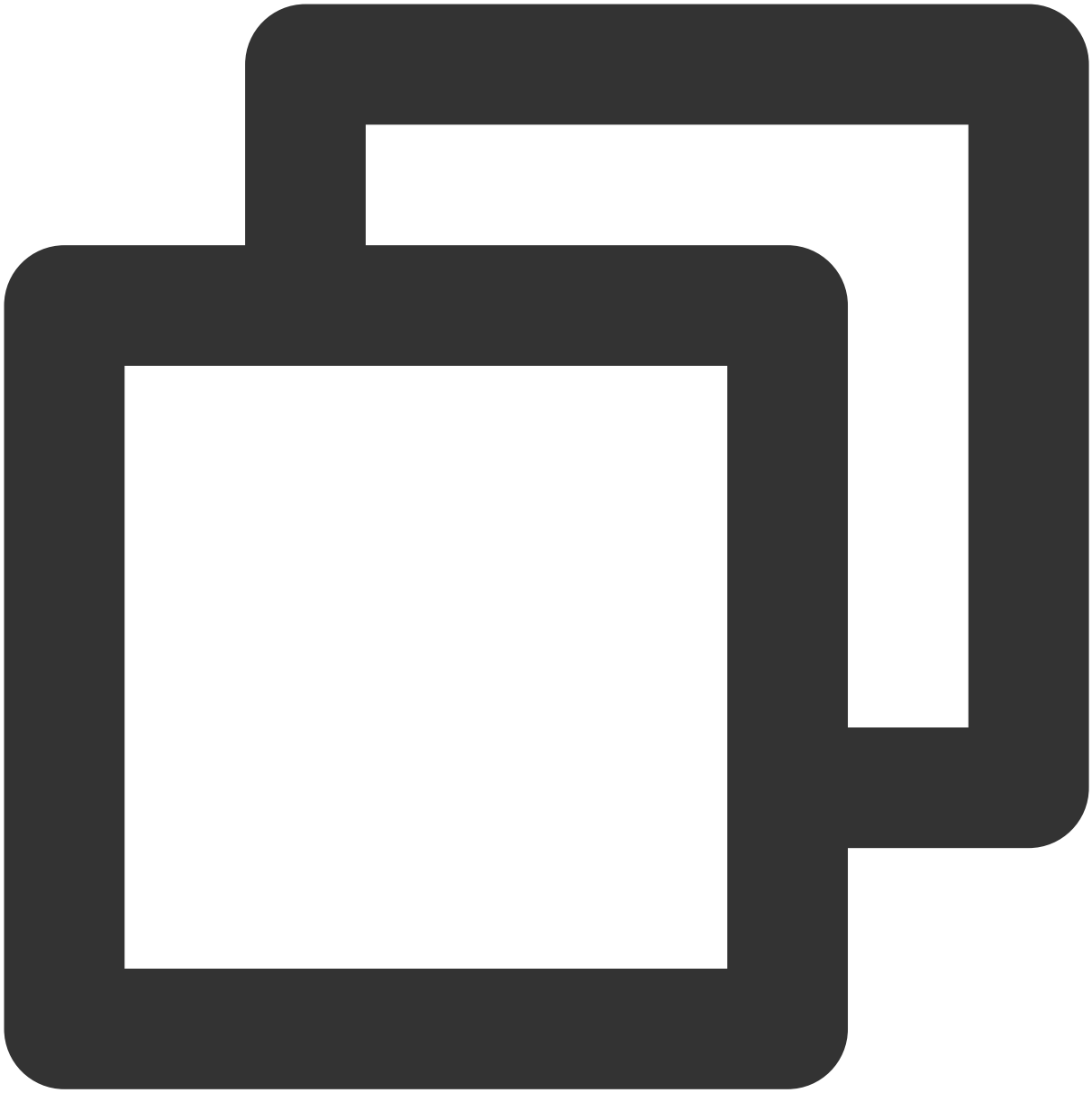
最近更新时间：2023-12-30 16:26:10

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:37:37+0800</CreationTime>
        <EndTime>2022-06-30T18:39:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>ja8de3b46f86011ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <DigitalWatermark>
                <Message>123456789ab</Message>
                <Type>Text</Type>
                <Version>V1</Version>
            </DigitalWatermark>
            <MediaInfo>
                <Format>
                    <Bitrate>2533.526000</Bitrate>
                    <Duration>215.638000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>68290577</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>256.050000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/48000</CodecTimeBase>
                        <Duration>215.637000</Duration>
                        <Index>1</Index>
                        <Language>eng</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>48000</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/48000</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.000000</AvgFps>
                        <Bitrate>2272.528000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/12288</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
```

```
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>215.583333</Duration>
                        <Fps>24.000000</Fps>
                        <HasBFrame>2</HasBFrame>
                        <Height>1080</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>40</Level>
                        <NumFrames>5174</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12288</Timebase>
                        <Width>1920</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>043221daee3f4e15ff13f42329f15eb1</Md5>
                        <ObjectName>output/digital.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/digital.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/digital.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my DigitalWatermark job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:37:38+0800</StartTime>
        <State>Success</State>
        <SubTag>DigitalWatermark</SubTag>
        <Tag>DigitalWatermark</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交数字水印任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

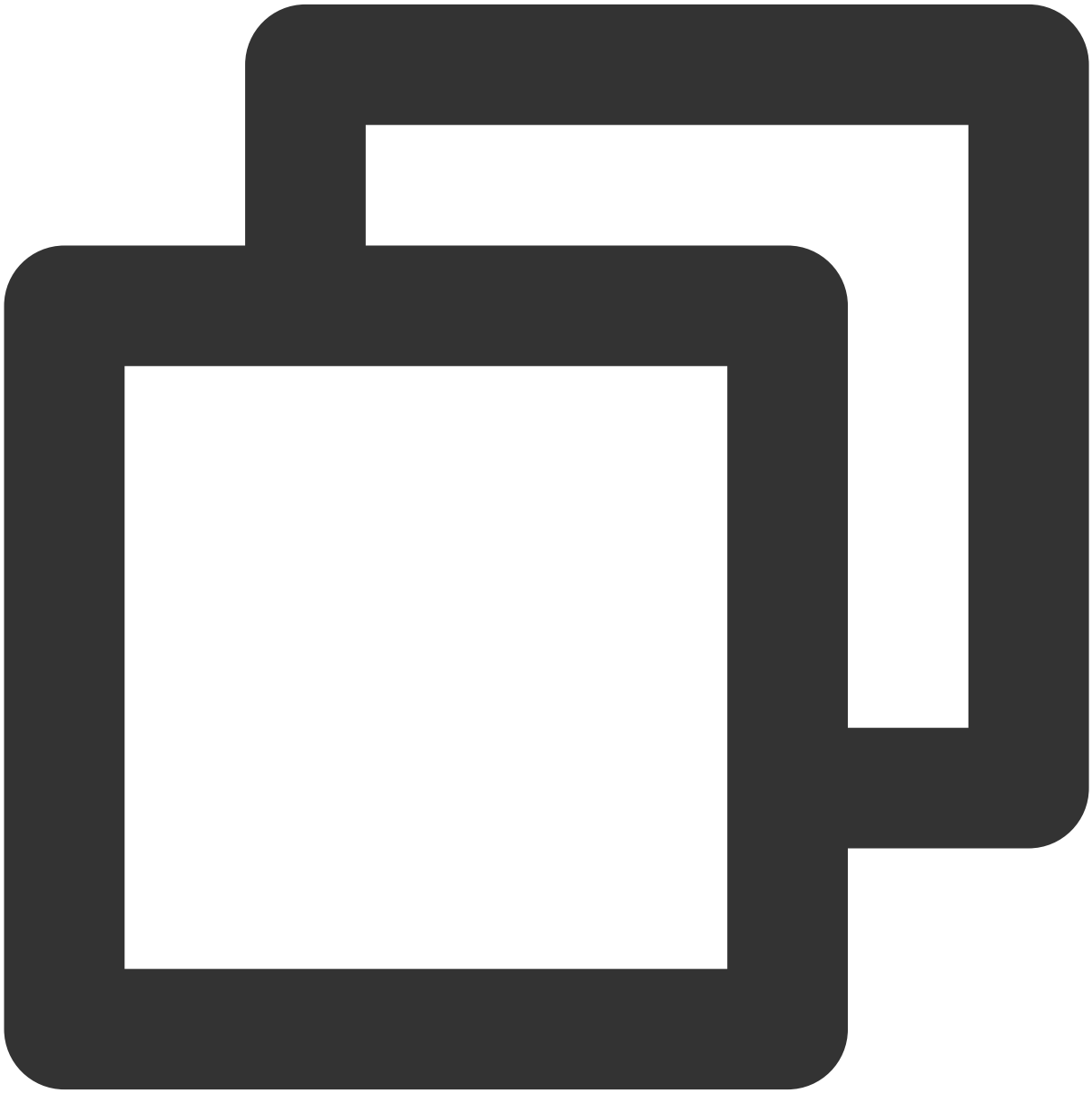
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

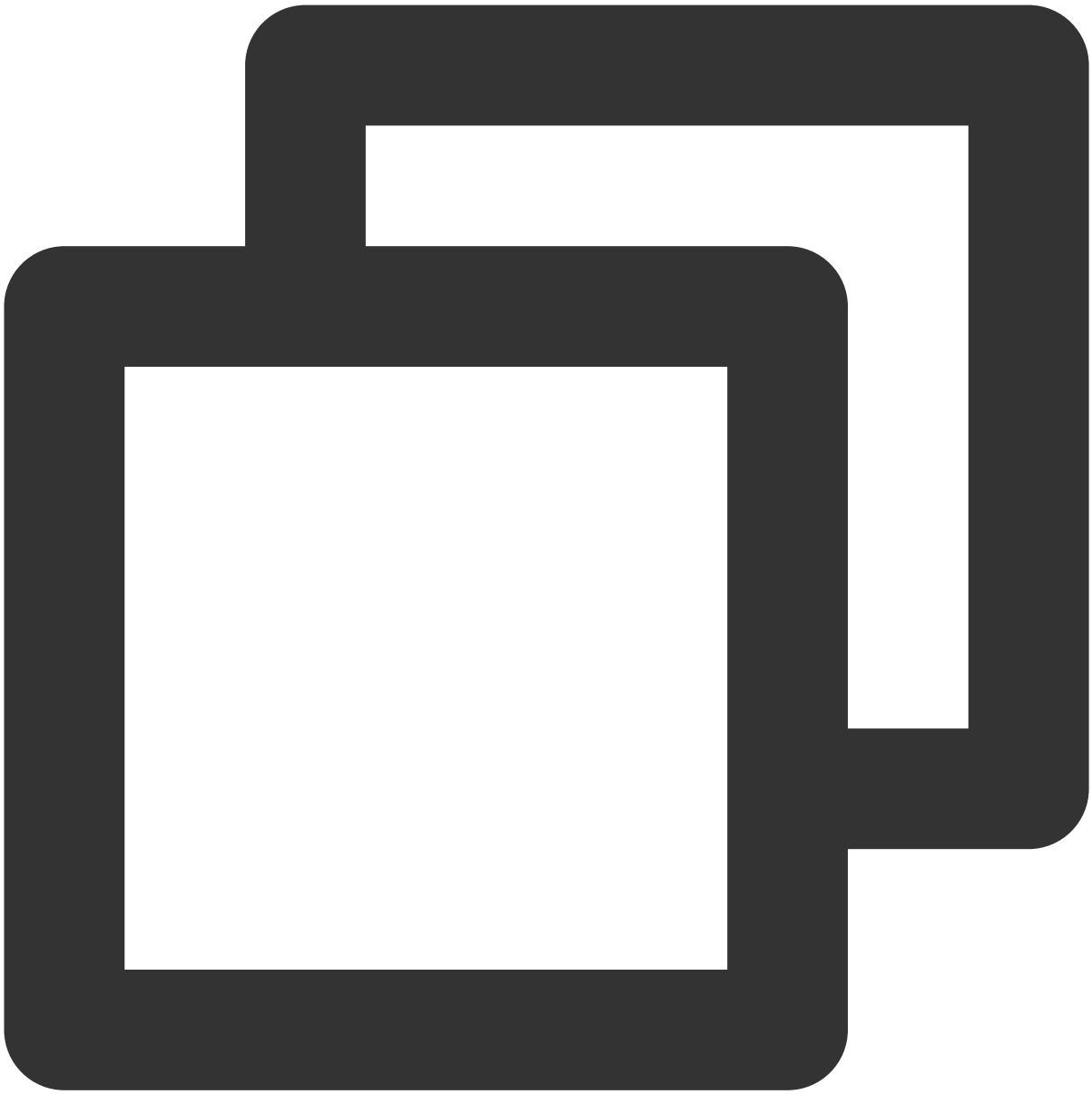


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:37:37+0800</CreationTime>
        <EndTime>2022-06-30T18:39:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>ja8de3b46f86011ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <DigitalWatermark>
                <Message>123456789ab</Message>
                <Type>Text</Type>
                <Version>V1</Version>
            </DigitalWatermark>
            <MediaInfo>
                <Format>
                    <Bitrate>2533.526000</Bitrate>
                    <Duration>215.638000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>68290577</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>256.050000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/48000</CodecTimeBase>
                        <Duration>215.637000</Duration>
                        <Index>1</Index>
                        <Language>eng</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>48000</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/48000</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.000000</AvgFps>
                        <Bitrate>2272.528000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/12288</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
```

```
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>215.583333</Duration>
                        <Fps>24.000000</Fps>
                        <HasBFrame>2</HasBFrame>
                        <Height>1080</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>40</Level>
                        <NumFrames>5174</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12288</Timebase>
                        <Width>1920</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>043221daee3f4e15ff13f42329f15eb1</Md5>
                        <ObjectName>output/digital.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/digital.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/digital.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my DigitalWatermark job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:37:38+0800</StartTime>
        <State>Success</State>
        <SubTag>DigitalWatermark</SubTag>
        <Tag>DigitalWatermark</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T18:37:37+0800",
        "EndTime": "2022-06-30T18:39:08+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
```

```
            "Region": "ap-chongqing"
        },
        "JobId": "ja8de3b46f86011ec9452c1c27f56a659",
        "Operation": {
            "DigitalWatermark": {
                "Message": "123456789ab",
                "Type": "Text",
                "Version": "V1"
            },
            "MediaInfo": {
                "Format": {
                    "Bitrate": "2533.526000",
                    "Duration": "215.638000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "68290577",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "256.050000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/48000",
                        "Duration": "215.637000",
                        "Index": "1",
                        "Language": "eng",
                        "SampleFmt": "fltp",
                        "SampleRate": "48000",
                        "StartTime": "0.000000",
                        "Timebase": "1/48000"
                    },
                    "Video": {
                        "AvgFps": "24.000000",
                        "Bitrate": "2272.528000",
                        "CodecLongName": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10
                        "CodecName": "h264",
                        "CodecTag": "0x31637661",
                        "CodecTagString": "avc1",
                        "CodecTimeBase": "1/12288",
                        "ColorPrimaries": "unknown",
```

```
                        "ColorRange": "unknown",
                        "ColorTransfer": "unknown",
                        "Duration": "215.583333",
                        "Fps": "24.000000",
                        "HasBFrame": "2",
                        "Height": "1080",
                        "Index": "0",
                        "Language": "und",
                        "Level": "40",
                        "NumFrames": "5174",
                        "PixFormat": "yuv420p",
                        "Profile": "High",
                        "RefFrames": "1",
                        "Rotation": "0.000000",
                        "StartTime": "0.000000",
                        "Timebase": "1/12288",
                        "Width": "1920"
                    }
                }
            },
            "MediaResult": {
                "OutputFile": {
                    "Bucket": "test-123456789",
                    "Md5Info": {
                        "Md5": "043221daee3f4e15ff13f42329f15eb1",
                        "ObjectName": "output/digital.mp4"
                    },
                    "ObjectName": "output/digital.mp4",
                    "Region": "ap-chongqing"
                }
            },
            "Output": {
                "Bucket": "test-123456789",
                "Object": "output/digital.mp4",
                "Region": "ap-chongqing"
            },
            "UserData": "This is my DigitalWatermark job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T18:37:38+0800",
        "State": "Success",
        "SubTag": "DigitalWatermark",
        "Tag": "DigitalWatermark"
    }
}
```

提取数字水印

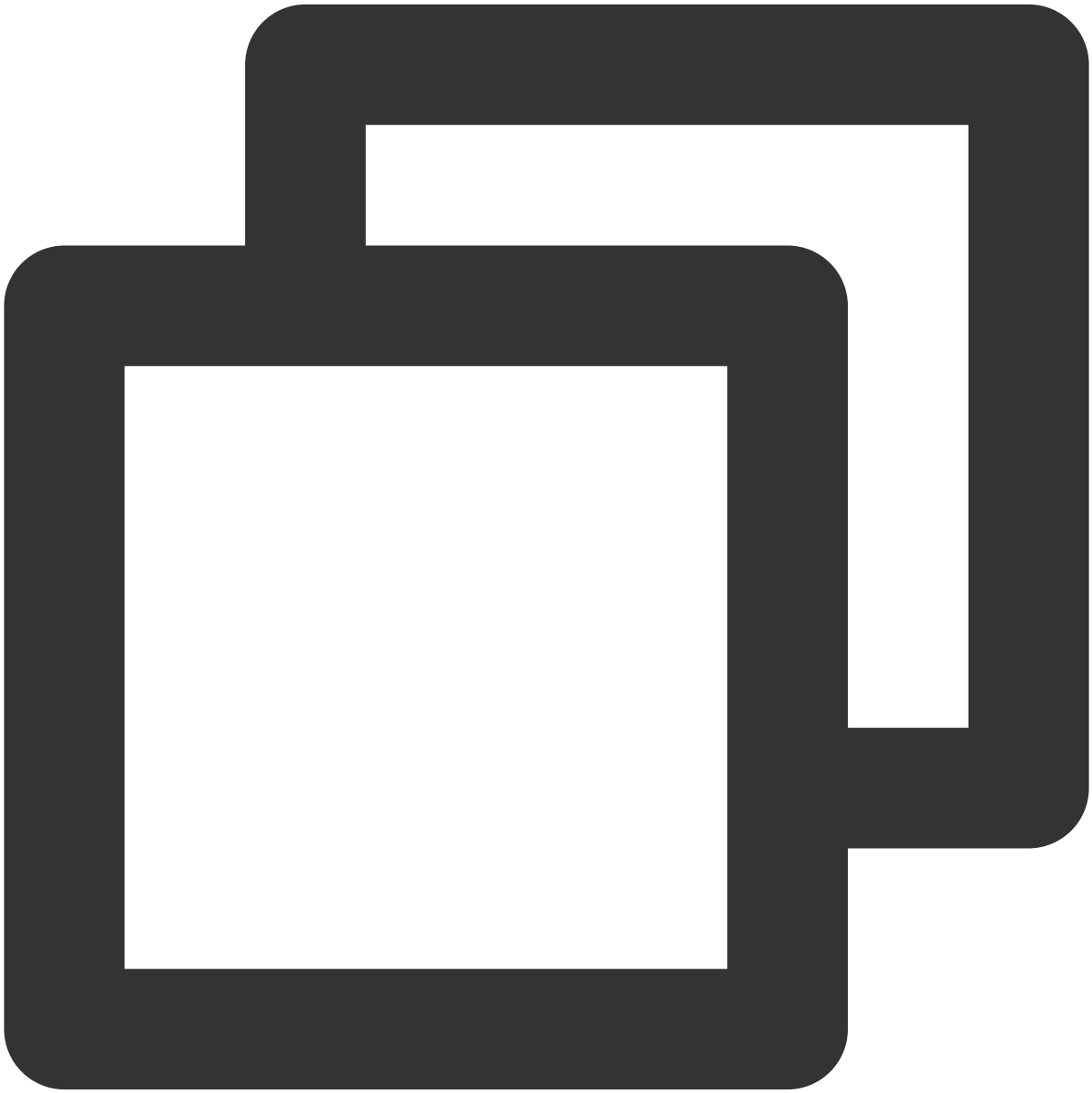
最近更新时间：2023-12-30 16:26:34

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:42:20+0800</CreationTime>
        <EndTime>2022-06-30T18:42:54+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>output/digital.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j515a494af86111ecb8546d80f2baf56f</JobId>
        <Message>done</Message>
        <Operation>
            <ExtractDigitalWatermark>
                <Message>123456789ab</Message>
                <Type>Text</Type>
                <Version>V1</Version>
            </ExtractDigitalWatermark>
            <UserData>This is my ExtractDigitalWatermark job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:42:21+0800</StartTime>
        <State>Success</State>
        <Tag>ExtractDigitalWatermark</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交提取数字水印任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

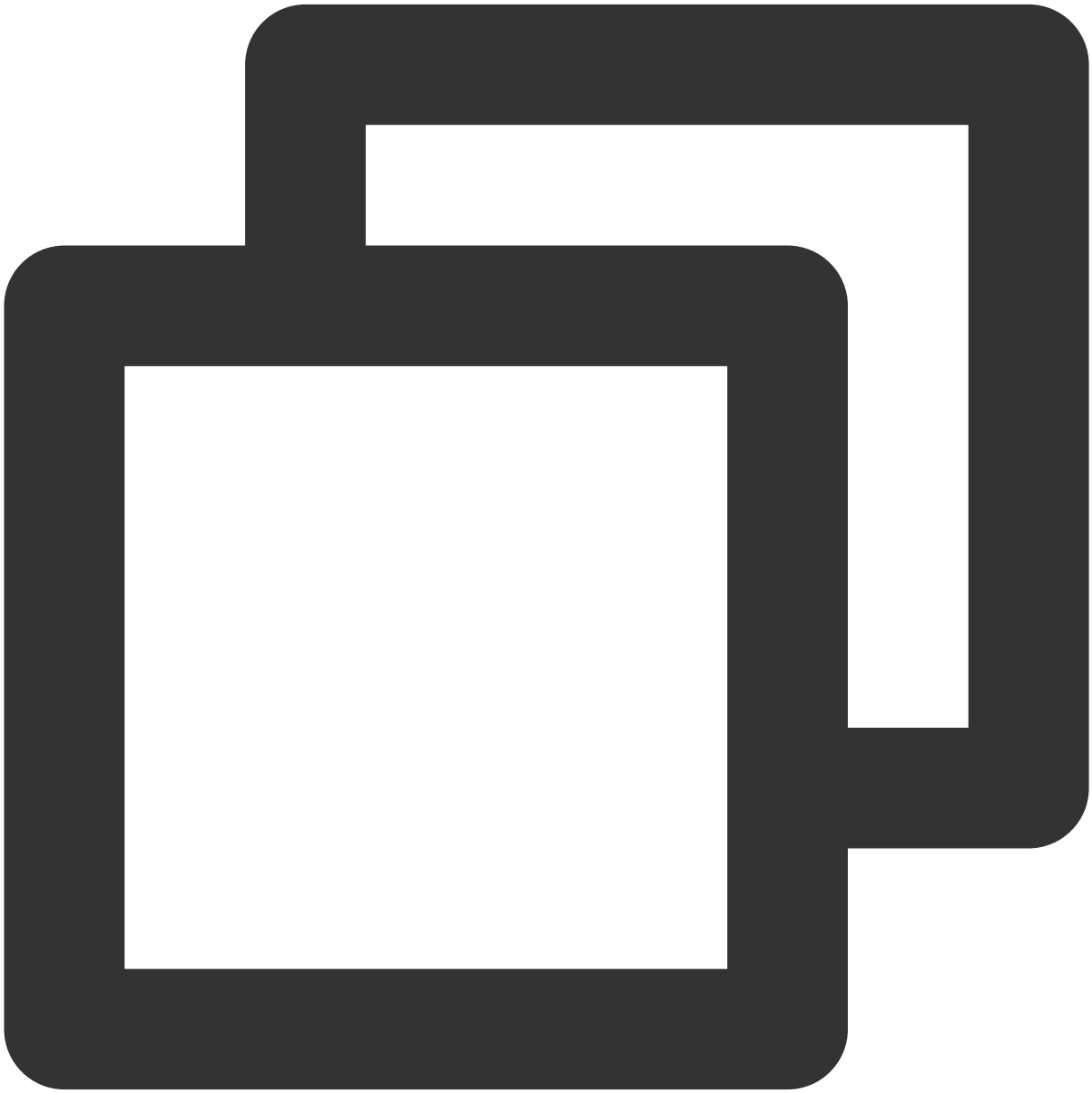
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

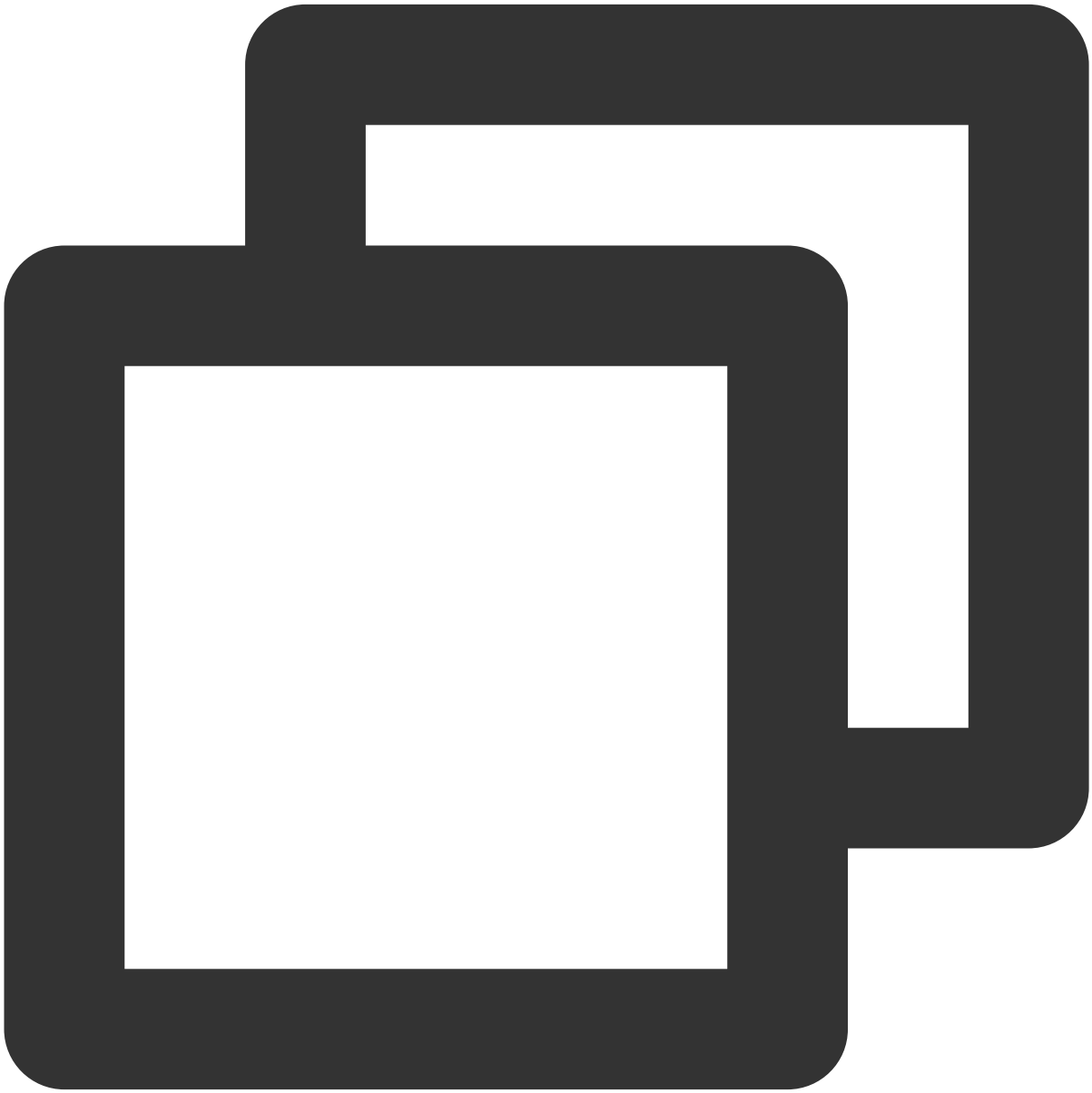
案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:42:20+0800</CreationTime>
        <EndTime>2022-06-30T18:42:54+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>output/digital.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j515a494af86111ecb8546d80f2baf56f</JobId>
        <Message>done</Message>
        <Operation>
            <ExtractDigitalWatermark>
                <Message>123456789ab</Message>
                <Type>Text</Type>
                <Version>V1</Version>
            </ExtractDigitalWatermark>
            <UserData>This is my ExtractDigitalWatermark job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:42:21+0800</StartTime>
        <State>Success</State>
        <Tag>ExtractDigitalWatermark</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T18:42:20+0800",
        "EndTime": "2022-06-30T18:42:54+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "output/digital.mp4",
            "Region": "ap-chongqing"
        },
```

```
        "JobId": "j515a494af86111ecb8546d80f2baf56f",
        "Message": "done",
        "Operation": {
            "ExtractDigitalWatermark": {
                "Message": "123456789ab",
                "Type": "Text",
                "Version": "V1"
            },
            "UserData": "This is my ExtractDigitalWatermark job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T18:42:21+0800",
        "State": "Success",
        "Tag": "ExtractDigitalWatermark"
    }
}
```

获取媒体信息

最近更新时间：2023-12-30 16:27:04

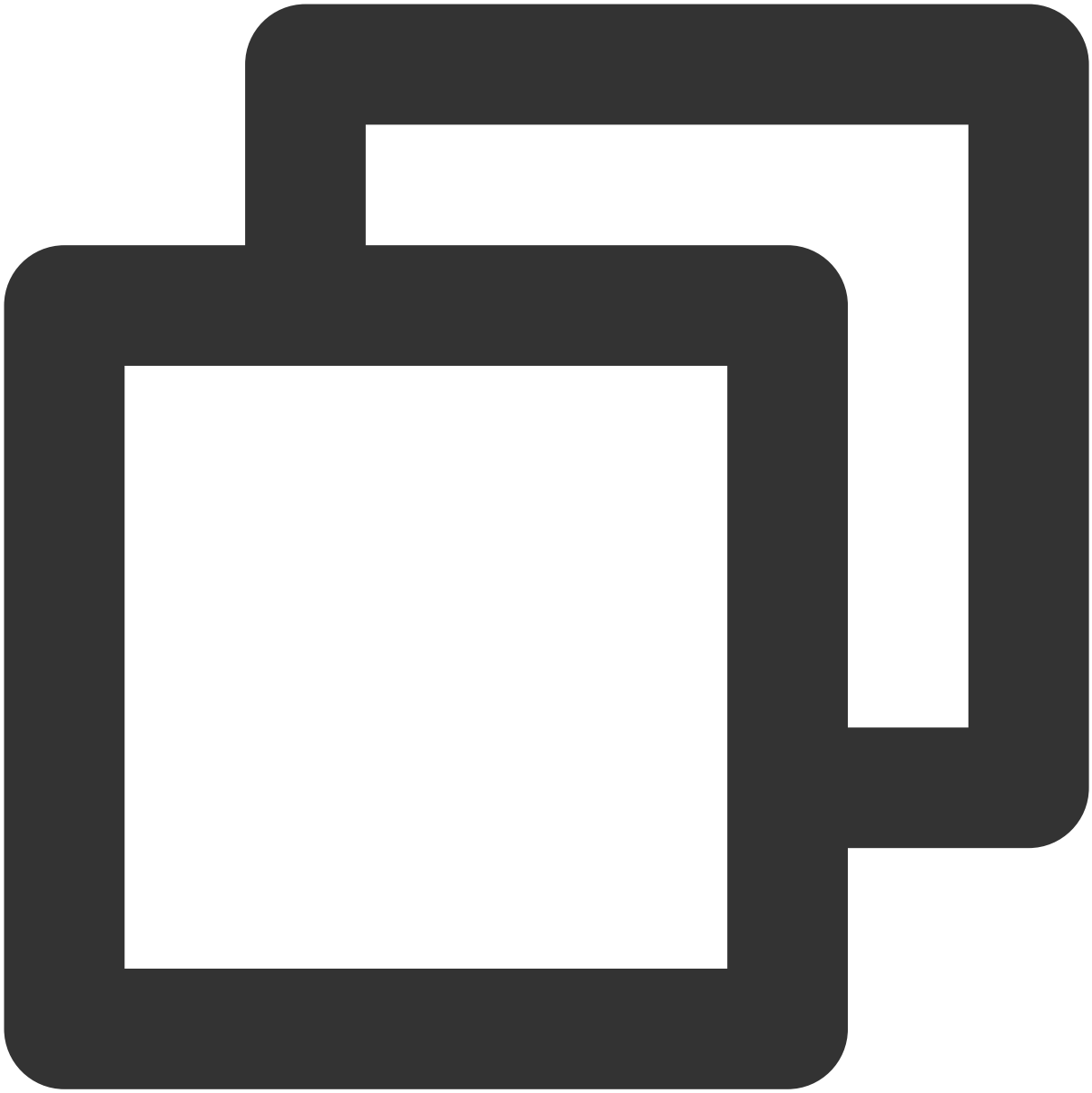
功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:45:14+0800</CreationTime>
        <EndTime>2022-06-30T18:45:15+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jb976c9e0f86111ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>834.736000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>1424687</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>104.047000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.653311</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000000</AvgFps>
                        <Bitrate>763.774000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/12800</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960000</Duration>
                        <Fps>25.000000</Fps>
                        <HasBFrame>0</HasBFrame>
```


版权所有：腾讯云计算（北京）有限责任公司

```
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>10</Level>
                        <NumFrames>324</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <UserData>This is my MediaInfo job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:45:15+0800</StartTime>
        <State>Success</State>
        <Tag>MediaInfo</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交获取媒体信息任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型

Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

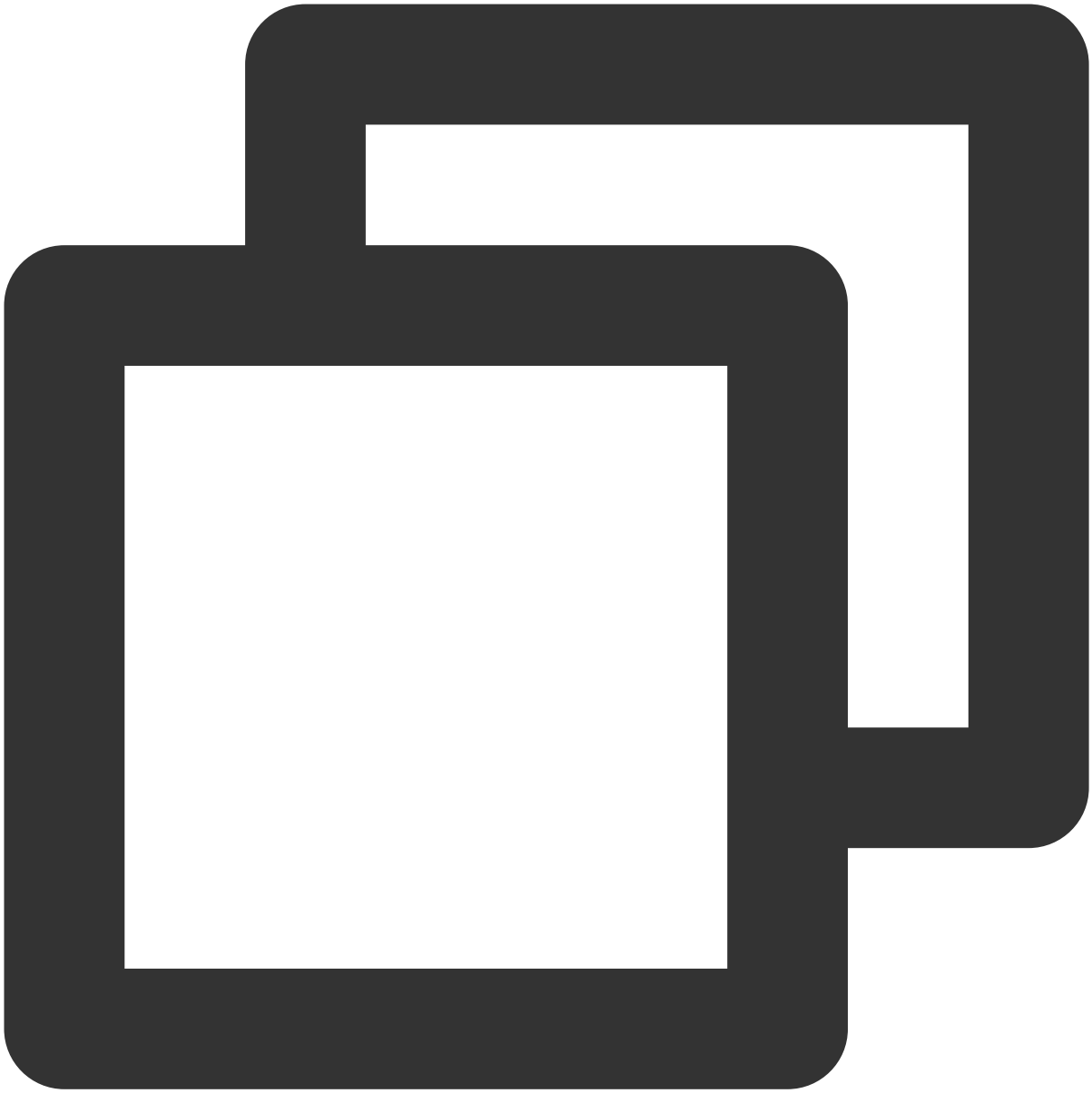
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

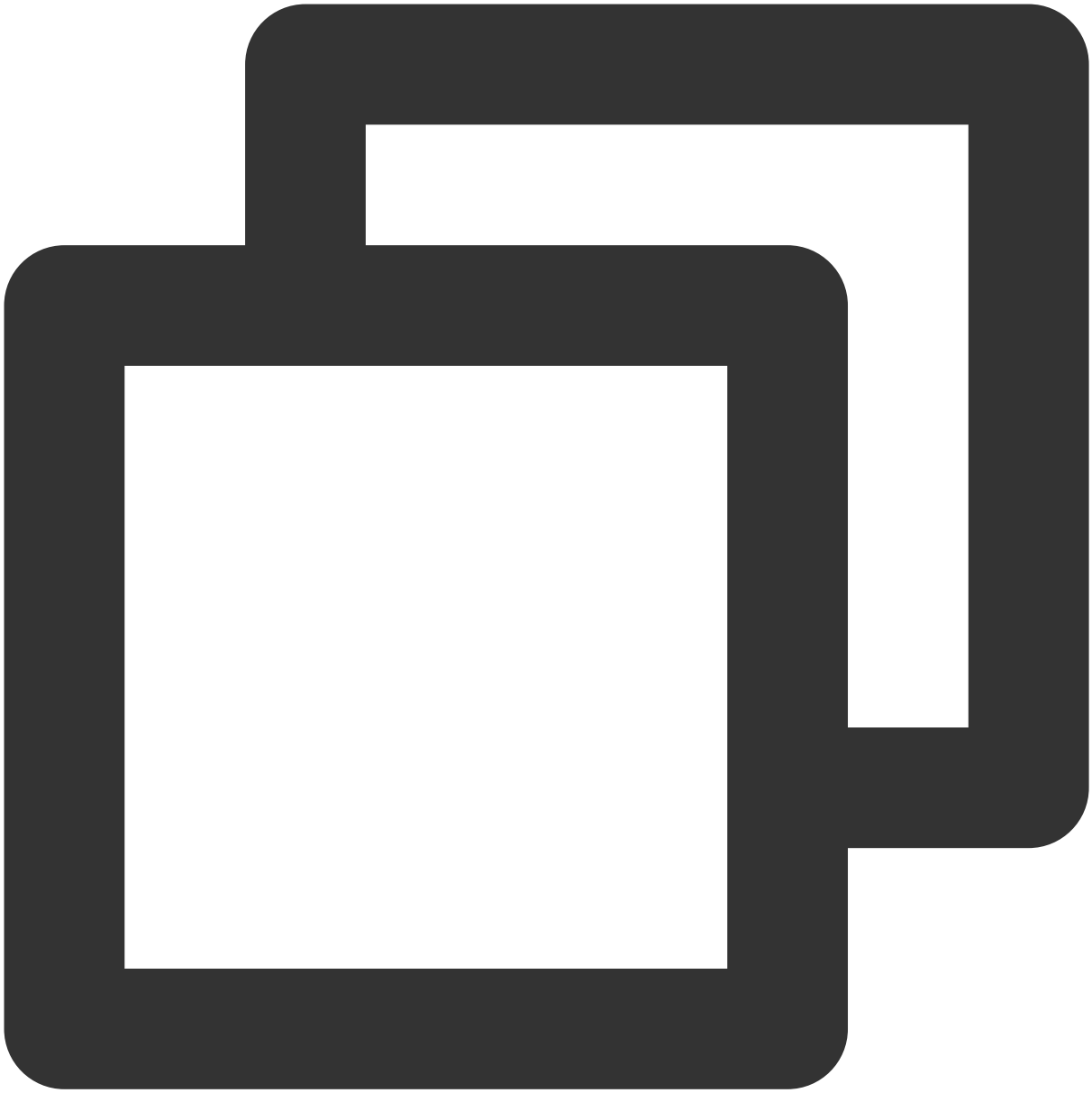


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:45:14+0800</CreationTime>
        <EndTime>2022-06-30T18:45:15+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jb976c9e0f86111ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>834.736000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>1424687</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>104.047000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.653311</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000000</AvgFps>
                        <Bitrate>763.774000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/12800</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960000</Duration>
                        <Fps>25.000000</Fps>
                        <HasBFrame>0</HasBFrame>
```

```
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>10</Level>
                        <NumFrames>324</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <UserData>This is my MediaInfo job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:45:15+0800</StartTime>
        <State>Success</State>
        <Tag>MediaInfo</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T18:45:14+0800",
        "EndTime": "2022-06-30T18:45:15+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing"
        },
```

```
        "JobId": "jb976c9e0f86111ec9452c1c27f56a659",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "834.736000",
                    "Duration": "13.654000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "1424687",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "104.047000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/44100",
                        "Duration": "13.653311",
                        "Index": "1",
                        "Language": "und",
                        "SampleFmt": "fltp",
                        "SampleRate": "44100",
                        "StartTime": "0.000000",
                        "Timebase": "1/44100"
                    },
                    "Video": {
                        "AvgFps": "25.000000",
                        "Bitrate": "763.774000",
                        "CodecLongName": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10
                        "CodecName": "h264",
                        "CodecTag": "0x31637661",
                        "CodecTagString": "avc1",
                        "CodecTimeBase": "1/12800",
                        "ColorPrimaries": "bt470bg",
                        "ColorRange": "tv",
                        "ColorTransfer": "smpte170m",
                        "Duration": "12.960000",
                        "Fps": "25.000000",
                        "HasBFrame": "0",
                        "Height": "960",
                        "Index": "0",
```

```
                        "Language": "und",
                        "Level": "10",
                        "NumFrames": "324",
                        "PixFormat": "yuv420p",
                        "Profile": "High",
                        "RefFrames": "1",
                        "Rotation": "0.000000",
                        "StartTime": "0.000000",
                        "Timebase": "1/12800",
                        "Width": "544"
                    }
                }
            },
            "UserData": "This is my MediaInfo job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T18:45:15+0800",
        "State": "Success",
        "Tag": "MediaInfo"
    }
}
```

降噪

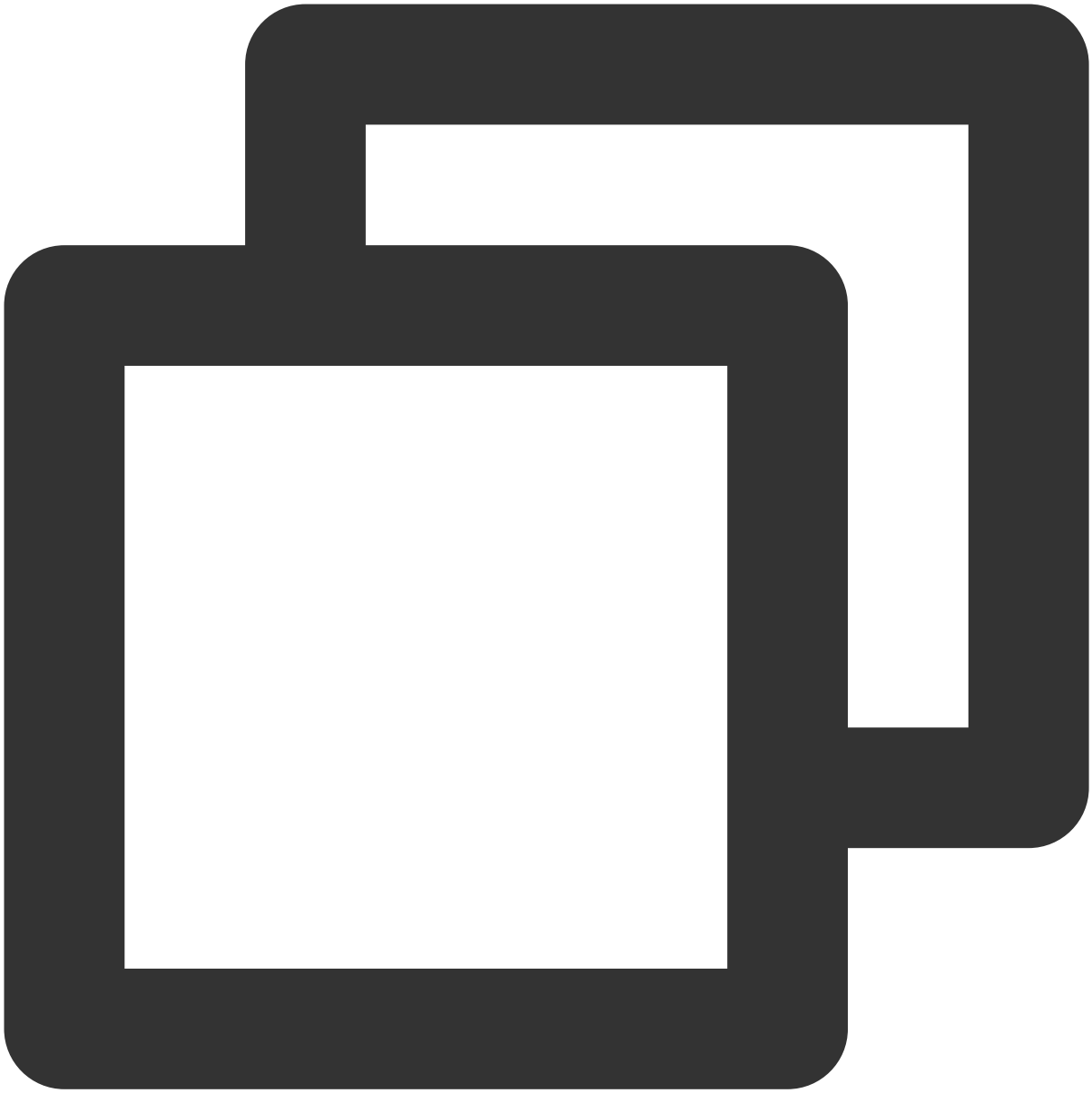
最近更新时间：2023-12-30 16:27:11

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:47:18+0800</CreationTime>
        <EndTime>2022-06-30T18:47:30+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j035aca66f86211ec8a5a87e016101404</JobId>
        <Message>SUCCESS</Message>
        <Operation>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/nr.wav</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my NoiseReduction job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:47:20+0800</StartTime>
        <State>Success</State>
        <Tag>NoiseReduction</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交音频降噪任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

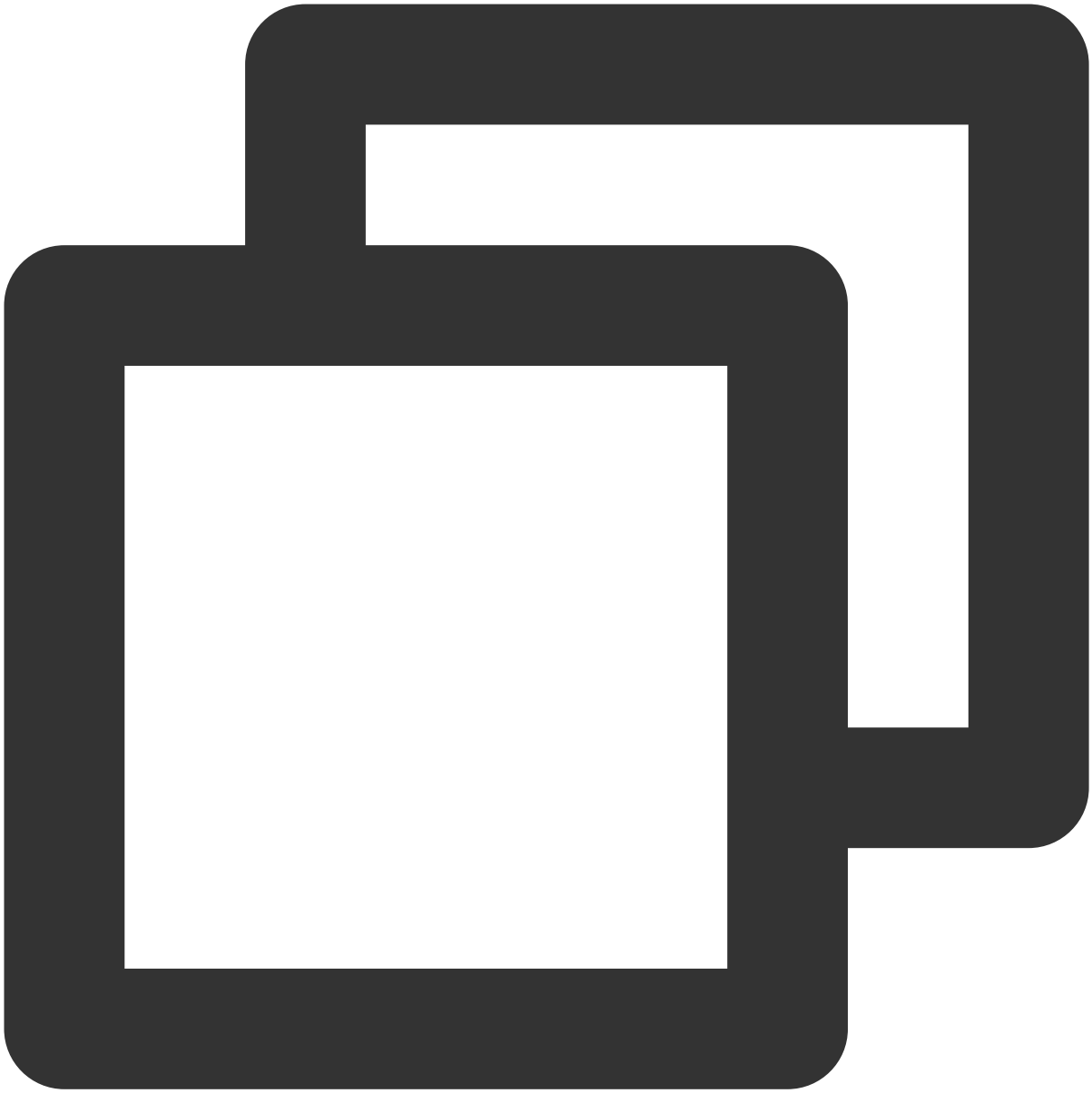
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

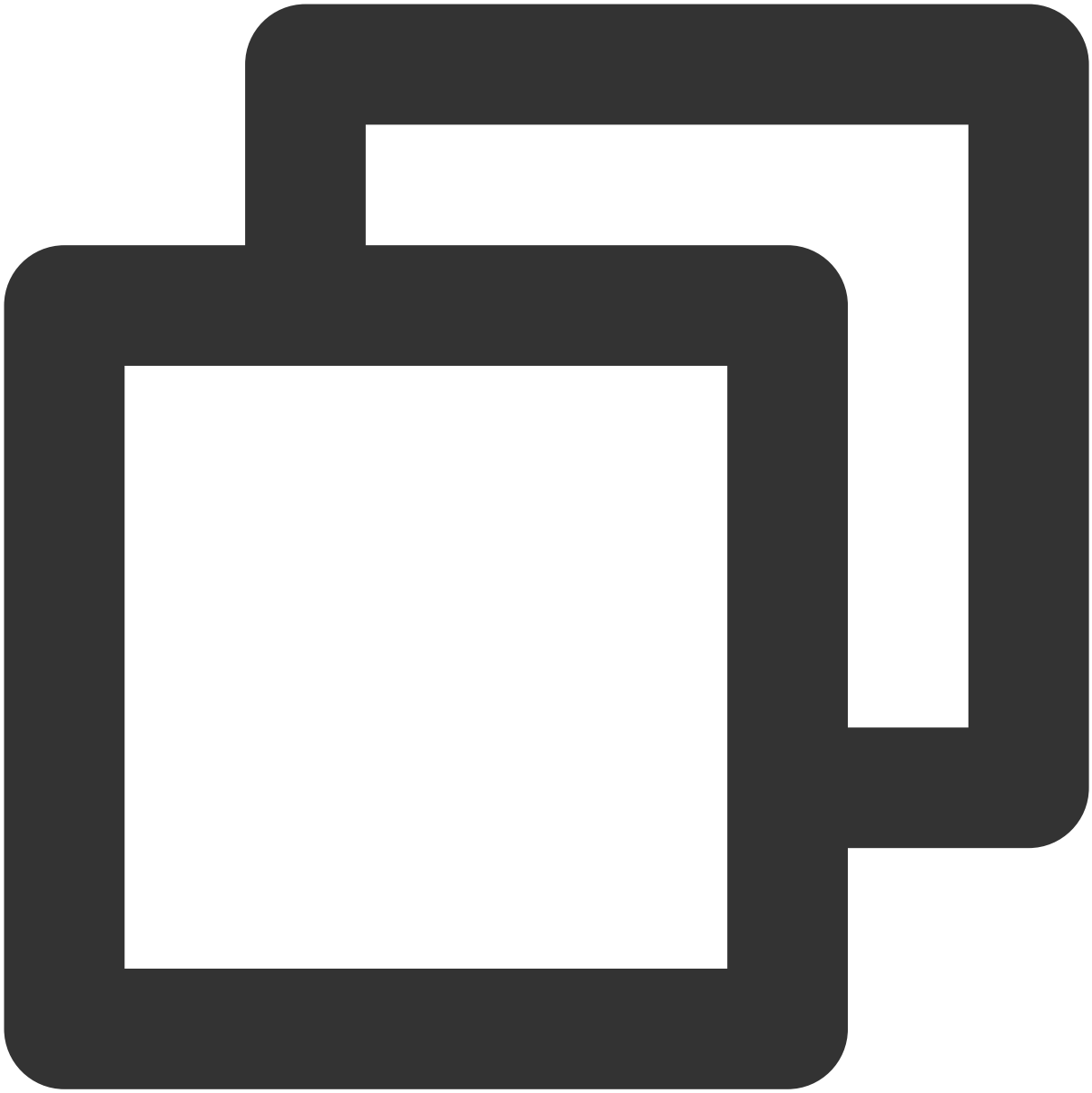
案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:47:18+0800</CreationTime>
        <EndTime>2022-06-30T18:47:30+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j035aca66f86211ec8a5a87e016101404</JobId>
        <Message>SUCCESS</Message>
        <Operation>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/nr.wav</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my NoiseReduction job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T18:47:20+0800</StartTime>
        <State>Success</State>
        <Tag>NoiseReduction</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T18:47:18+0800",
        "EndTime": "2022-06-30T18:47:30+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp3",
            "Region": "ap-chongqing"
        },
```

```
        "JobId": "j035aca66f86211ec8a5a87e016101404",
        "Message": "SUCCESS",
        "Operation": {
            "Output": {
                "Bucket": "test-123456789",
                "Object": "output/nr.wav",
                "Region": "ap-chongqing"
            },
            "UserData": "This is my NoiseReduction job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T18:47:20+0800",
        "State": "Success",
        "Tag": "NoiseReduction"
    }
}
```

视频质量评分

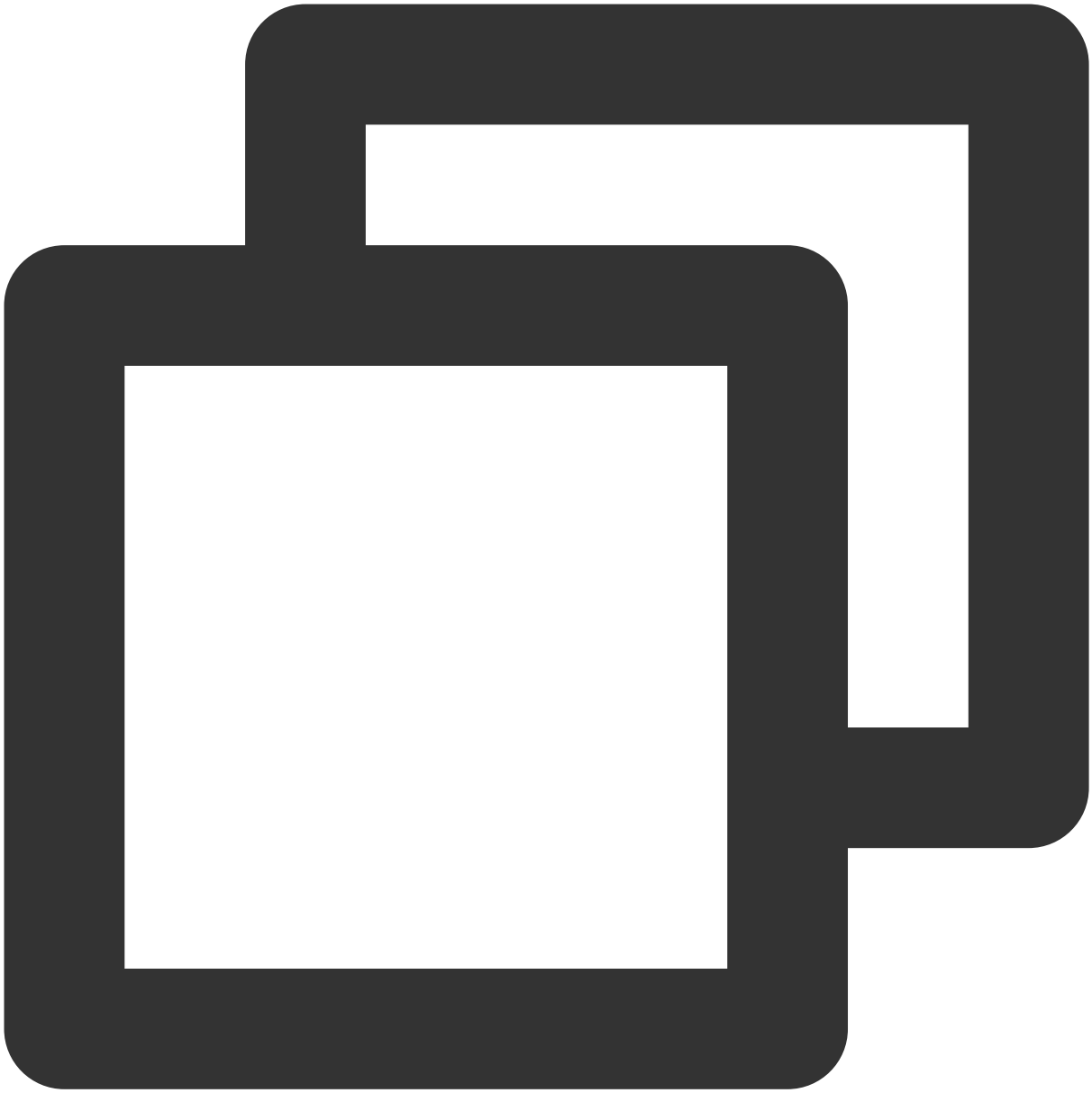
最近更新时间：2023-12-30 16:27:45

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:06:27+0800</CreationTime>
        <EndTime>2022-06-30T19:06:34+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jb01fe2f2f86411ec9ca1073b78d316d3</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>834736</Bitrate>
                    <Duration>13.654</Duration>
                    <NumStream>1</NumStream>
                    <Size>1424687</Size>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>104047</Bitrate>
                        <Channel>2</Channel>
                        <CodecName>aac</CodecName>
                        <Duration>13.653</Duration>
                        <Index>1</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000</AvgFps>
                        <Bitrate>763774</Bitrate>
                        <CodecName>h264</CodecName>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960</Duration>
                        <Fps>25.000</Fps>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Level>1.000</Level>
                        <NumFrames>324</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <Rotation>0</Rotation>
                        <StartTime>0.000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile/>
            </MediaResult>
            <QualityEstimate>
```

```
                <Score>6.460358</Score>
            </QualityEstimate>
            <UserData>This is my QualityEstimate job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:06:28+0800</StartTime>
        <State>Success</State>
        <Tag>QualityEstimate</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交视频质量分析任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

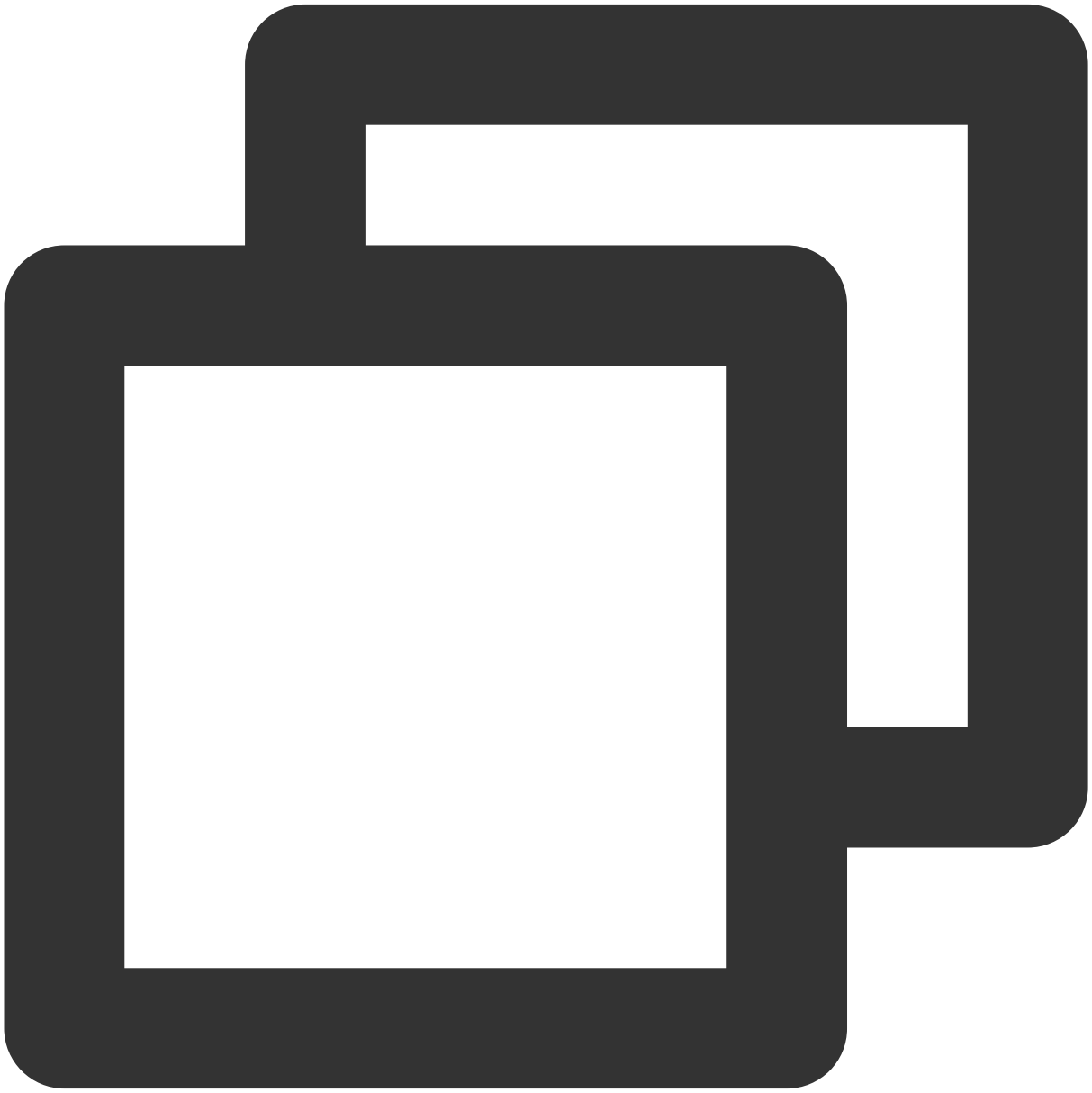
节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String

版权所有：腾讯云计算（北京）有限责任公司

WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

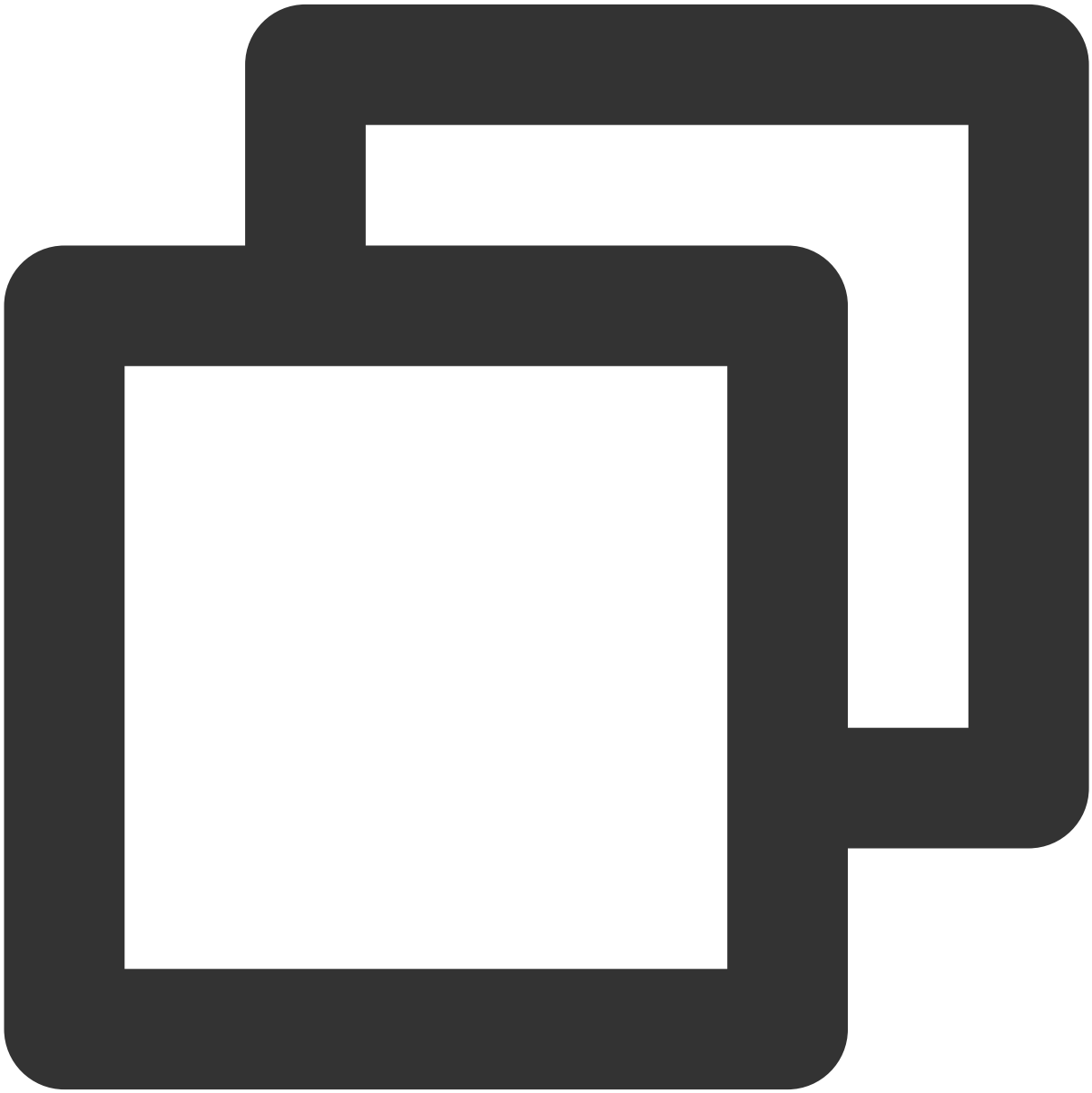


```
<Response>
```

```
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:06:27+0800</CreationTime>
        <EndTime>2022-06-30T19:06:34+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <JobId>jb01fe2f2f86411ec9ca1073b78d316d3</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>834736</Bitrate>
                    <Duration>13.654</Duration>
                    <NumStream>1</NumStream>
                    <Size>1424687</Size>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>104047</Bitrate>
                        <Channel>2</Channel>
                        <CodecName>aac</CodecName>
                        <Duration>13.653</Duration>
                        <Index>1</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000</AvgFps>
                        <Bitrate>763774</Bitrate>
                        <CodecName>h264</CodecName>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960</Duration>
                        <Fps>25.000</Fps>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Level>1.000</Level>
                        <NumFrames>324</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <Rotation>0</Rotation>
```

```
                        <StartTime>0.000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile/>
            </MediaResult>
            <QualityEstimate>
                <Score>6.460358</Score>
            </QualityEstimate>
            <UserData>This is my QualityEstimate job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:06:28+0800</StartTime>
        <State>Success</State>
        <Tag>QualityEstimate</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:06:27+0800",
        "EndTime": "2022-06-30T19:06:34+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing"
        },
```

```
        "JobId": "jb01fe2f2f86411ec9ca1073b78d316d3",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "834736",
                    "Duration": "13.654",
                    "NumStream": "1",
                    "Size": "1424687"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "104047",
                        "Channel": "2",
                        "CodecName": "aac",
                        "Duration": "13.653",
                        "Index": "1",
                        "SampleFmt": "fltp",
                        "SampleRate": "44100"
                    },
                    "Video": {
                        "AvgFps": "25.000",
                        "Bitrate": "763774",
                        "CodecName": "h264",
                        "ColorPrimaries": "unknown",
                        "ColorRange": "tv",
                        "ColorTransfer": "smpte170m",
                        "Duration": "12.960",
                        "Fps": "25.000",
                        "Height": "960",
                        "Index": "0",
                        "Level": "1.000",
                        "NumFrames": "324",
                        "PixFormat": "yuv420p",
                        "Profile": "High",
                        "Rotation": "0",
                        "StartTime": "0.000",
                        "Timebase": "1/12800",
                        "Width": "544"
                    }
                }
            },
            "MediaResult": {

            },
            "QualityEstimate": {
                "Score": "6.460358"
            },
```

```
            "UserData": "This is my QualityEstimate job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:06:28+0800",
        "State": "Success",
        "Tag": "QualityEstimate"
    }
}
```

SDRtoHDR

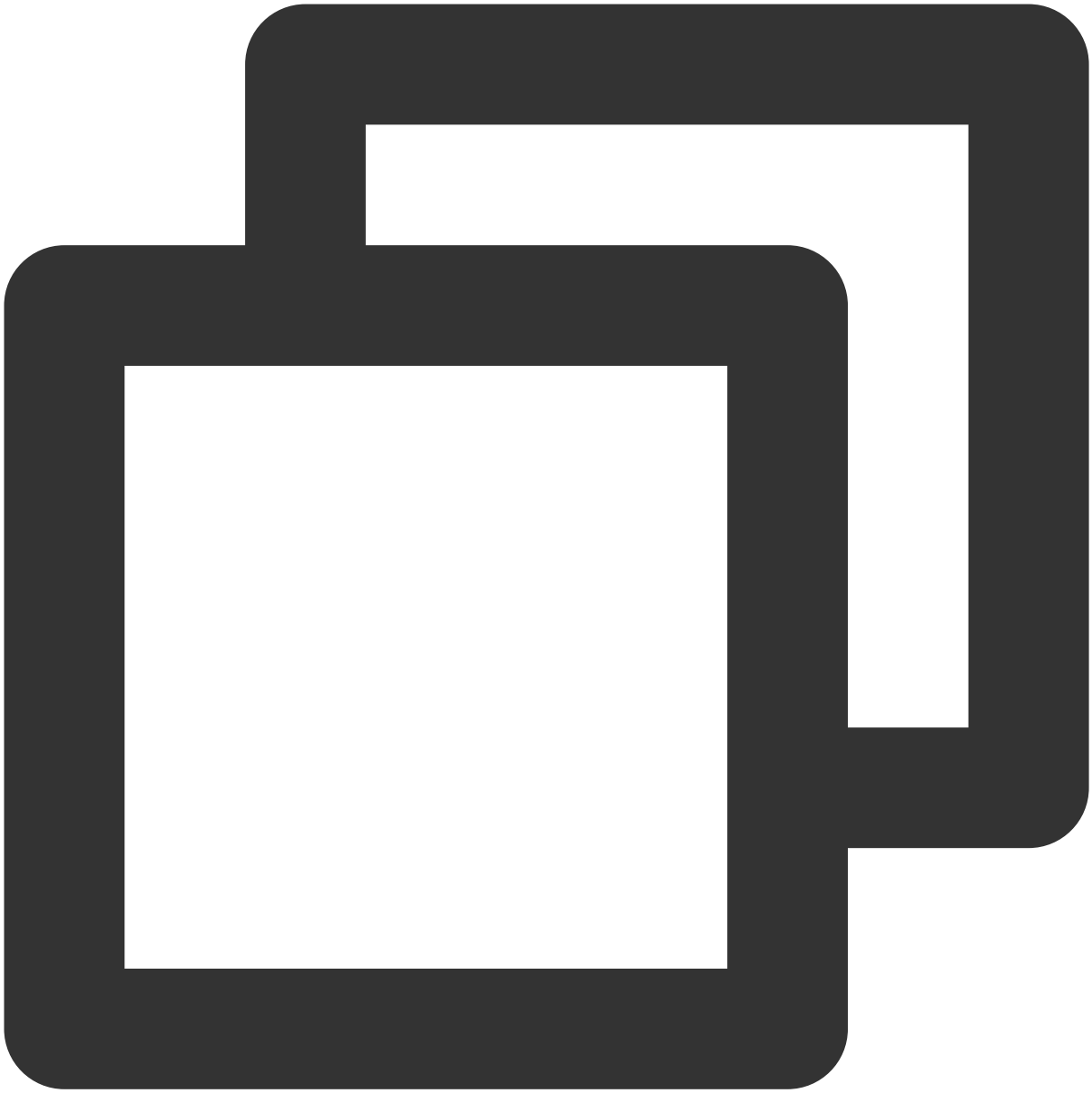
最近更新时间：2023-12-30 16:27:38

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:20:45+0800</CreationTime>
        <EndTime>2022-06-30T19:21:06+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jafa709a2f86611ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>9514.268000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>16238478</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.922840</AvgFps>
                        <Bitrate>9878.900000</Bitrate>
                        <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                        <CodecName>hevc</CodecName>
                        <CodecTag>0x31637668</CodecTag>
                        <CodecTagString>hvc1</CodecTagString>
                        <CodecTimeBase>1/12800</CodecTimeBase>
                        <ColorPrimaries>bt2020</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte2084</ColorTransfer>
                        <Duration>12.960000</Duration>
                        <FieldOrder>progressive</FieldOrder>
                        <Fps>25.000000</Fps>
```

```
                        <HasBFrame>2</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>93</Level>
                        <NumFrames>323</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>Main</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>ed8b5615687d789d6293bd58fa1ea1c0</Md5>
                        <ObjectName>output/sdr2hdr.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/sdr2hdr.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/sdr2hdr.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SDRtoHDR>
                <HdrMode>HDR10</HdrMode>
            </SDRtoHDR>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <UserData>This is my SDRtoHDR job.</UserData>
            <WatermarkTemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</WatermarkTempl
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:20:46+0800</StartTime>
        <State>Success</State>
        <Tag>SDRtoHDR</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交SDR to HDR任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

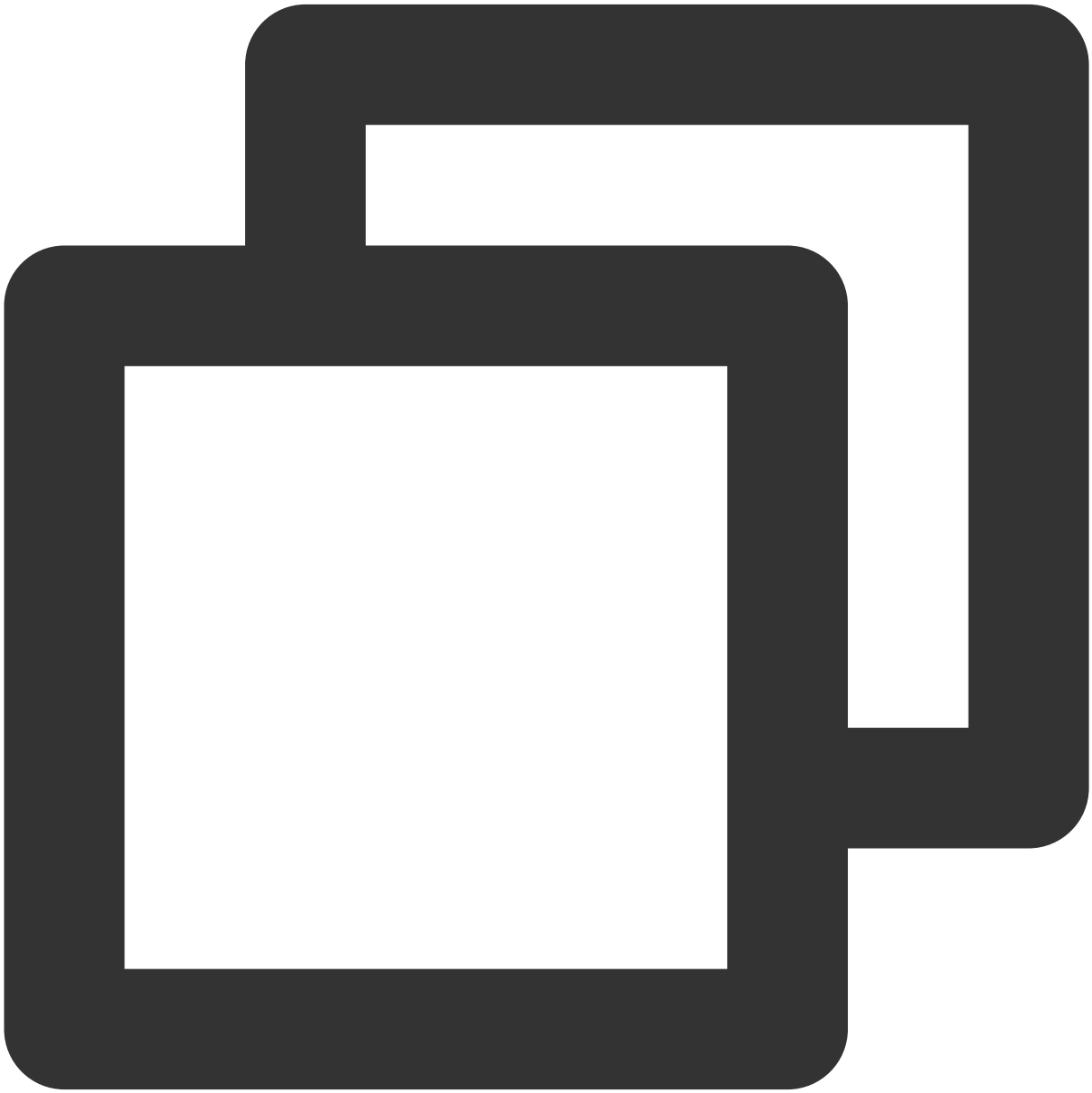
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

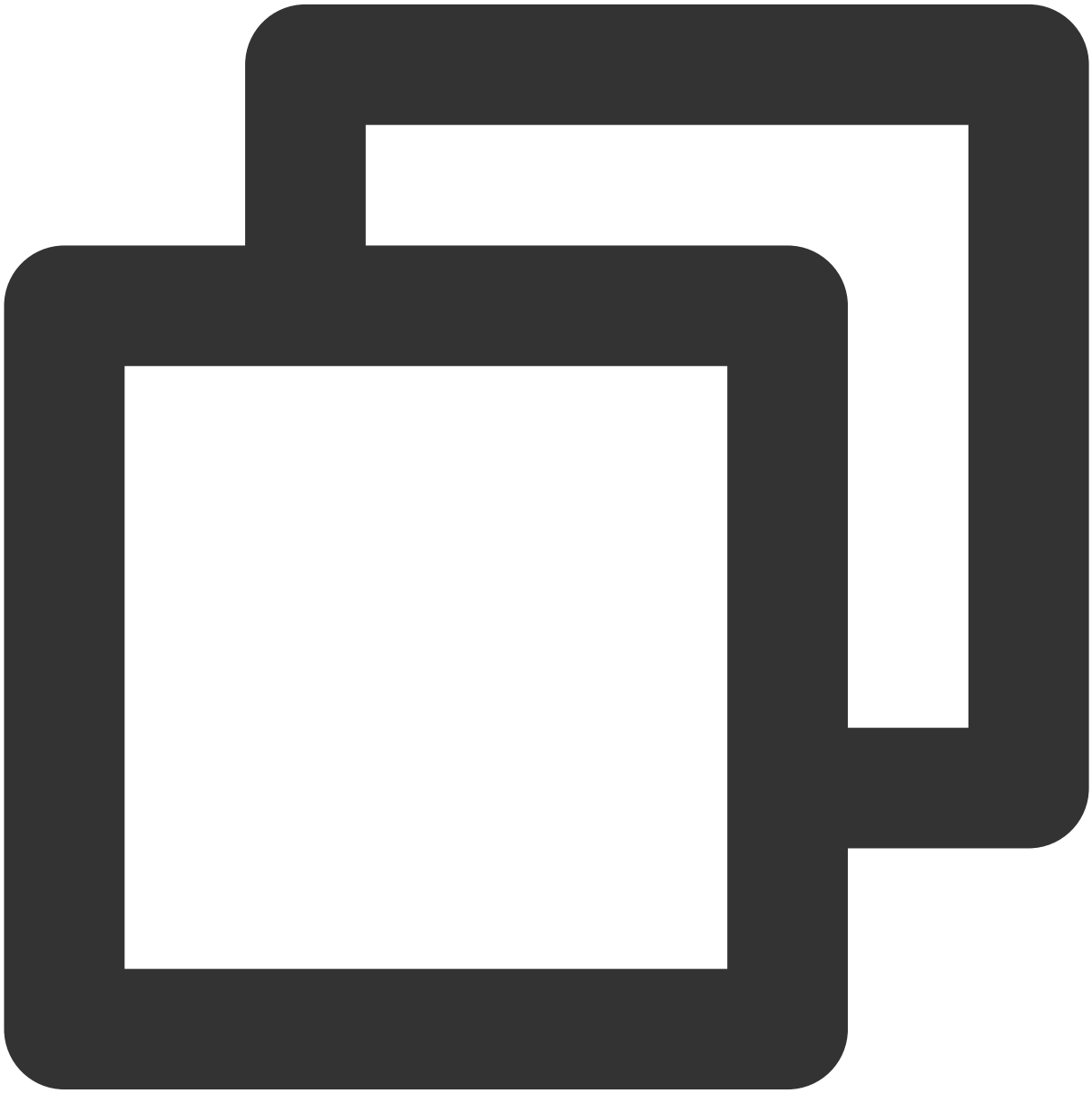


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:20:45+0800</CreationTime>
        <EndTime>2022-06-30T19:21:06+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jafa709a2f86611ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>9514.268000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>16238478</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.922840</AvgFps>
                        <Bitrate>9878.900000</Bitrate>
                        <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                        <CodecName>hevc</CodecName>
                        <CodecTag>0x31637668</CodecTag>
                        <CodecTagString>hvc1</CodecTagString>
                        <CodecTimeBase>1/12800</CodecTimeBase>
                        <ColorPrimaries>bt2020</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte2084</ColorTransfer>
                        <Duration>12.960000</Duration>
                        <FieldOrder>progressive</FieldOrder>
                        <Fps>25.000000</Fps>
```

```
                        <HasBFrame>2</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>93</Level>
                        <NumFrames>323</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>Main</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>ed8b5615687d789d6293bd58fa1ea1c0</Md5>
                        <ObjectName>output/sdr2hdr.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/sdr2hdr.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/sdr2hdr.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SDRtoHDR>
                <HdrMode>HDR10</HdrMode>
            </SDRtoHDR>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <UserData>This is my SDRtoHDR job.</UserData>
            <WatermarkTemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</WatermarkTempl
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:20:46+0800</StartTime>
        <State>Success</State>
        <Tag>SDRtoHDR</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



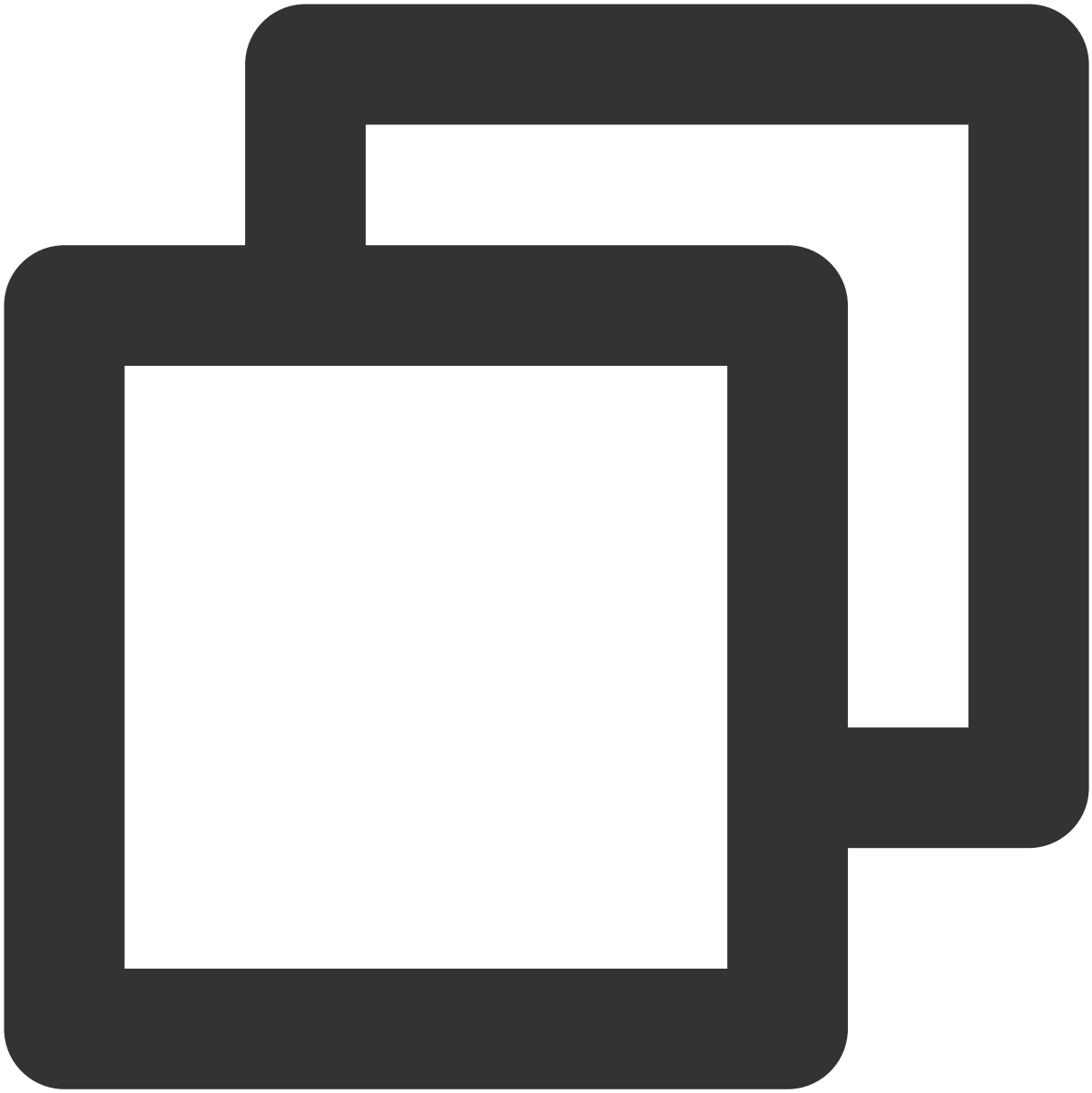
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:20:45+0800</CreationTime>
        <EndTime>2022-06-30T19:21:06+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
```

```
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>jafa709a2f86611ec9452c1c27f56a659</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>9514.268000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>16238478</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
```

```
                    <SampleRate>44100</SampleRate>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/44100</Timebase>
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>24.922840</AvgFps>
                    <Bitrate>9878.900000</Bitrate>
                    <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                    <CodecName>hevc</CodecName>
                    <CodecTag>0x31637668</CodecTag>
                    <CodecTagString>hvc1</CodecTagString>
                    <CodecTimeBase>1/12800</CodecTimeBase>
                    <ColorPrimaries>bt2020</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte2084</ColorTransfer>
                    <Duration>12.960000</Duration>
                    <FieldOrder>progressive</FieldOrder>
                    <Fps>25.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>960</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>93</Level>
                    <NumFrames>323</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>Main</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>544</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>ed8b5615687d789d6293bd58fa1ea1c0</Md5>
                    <ObjectName>output/sdr2hdr.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/sdr2hdr.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
```

```
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/sdr2hdr.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SDRtoHDR>
                <HdrMode>HDR10</HdrMode>
            </SDRtoHDR>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <UserData>This is my SDRtoHDR job.</UserData>
            <WatermarkTemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</WatermarkTempl
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:20:46+0800</StartTime>
        <State>Success</State>
        <Tag>SDRtoHDR</Tag>
        <Workflow>
            <Name>SDRtoHDR_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:20:45+0800",
        "EndTime": "2022-06-30T19:21:06+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "jafa709a2f86611ec9452c1c27f56a659",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "9514.268000",
                    "Duration": "13.654000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "16238478",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "128.726000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/44100",
                        "Duration": "13.652993",
                        "Index": "1",
                        "Language": "und",
                        "SampleFmt": "fltp",
                        "SampleRate": "44100",
                        "StartTime": "0.000000",
                        "Timebase": "1/44100"
```

```
                },
                "Video": {
                    "AvgFps": "24.922840",
                    "Bitrate": "9878.900000",
                    "CodecLongName": "H.265 / HEVC (High Efficiency Video Codin
                    "CodecName": "hevc",
                    "CodecTag": "0x31637668",
                    "CodecTagString": "hvc1",
                    "CodecTimeBase": "1/12800",
                    "ColorPrimaries": "bt2020",
                    "ColorRange": "tv",
                    "ColorTransfer": "smpte2084",
                    "Duration": "12.960000",
                    "FieldOrder": "progressive",
                    "Fps": "25.000000",
                    "HasBFrame": "2",
                    "Height": "960",
                    "Index": "0",
                    "Language": "und",
                    "Level": "93",
                    "NumFrames": "323",
                    "PixFormat": "yuv420p",
                    "Profile": "Main",
                    "RefFrames": "1",
                    "Rotation": "0.000000",
                    "StartTime": "0.000000",
                    "Timebase": "1/12800",
                    "Width": "544"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": {
                    "Md5": "ed8b5615687d789d6293bd58fa1ea1c0",
                    "ObjectName": "output/sdr2hdr.mp4"
                },
                "ObjectName": "output/sdr2hdr.mp4",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/sdr2hdr.mp4",
            "Region": "ap-chongqing"
        },
```

```
            "SDRtoHDR": {
                "HdrMode": "HDR10"
            },
            "TranscodeTemplateId": "t156c107210e7243c5817354565d81b578",
            "UserData": "This is my SDRtoHDR job.",
            "WatermarkTemplateId": [
                "t143ae6e040af6431aa772c9ec3f0a3f36",
                "t12a74d11687d444deba8a6cc52051ac27"
            ]
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:20:46+0800",
        "State": "Success",
        "Tag": "SDRtoHDR",
        "Workflow": {
            "Name": "SDRtoHDR_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

转封装（音视频分段）

最近更新时间：2023-12-30 16:28:17

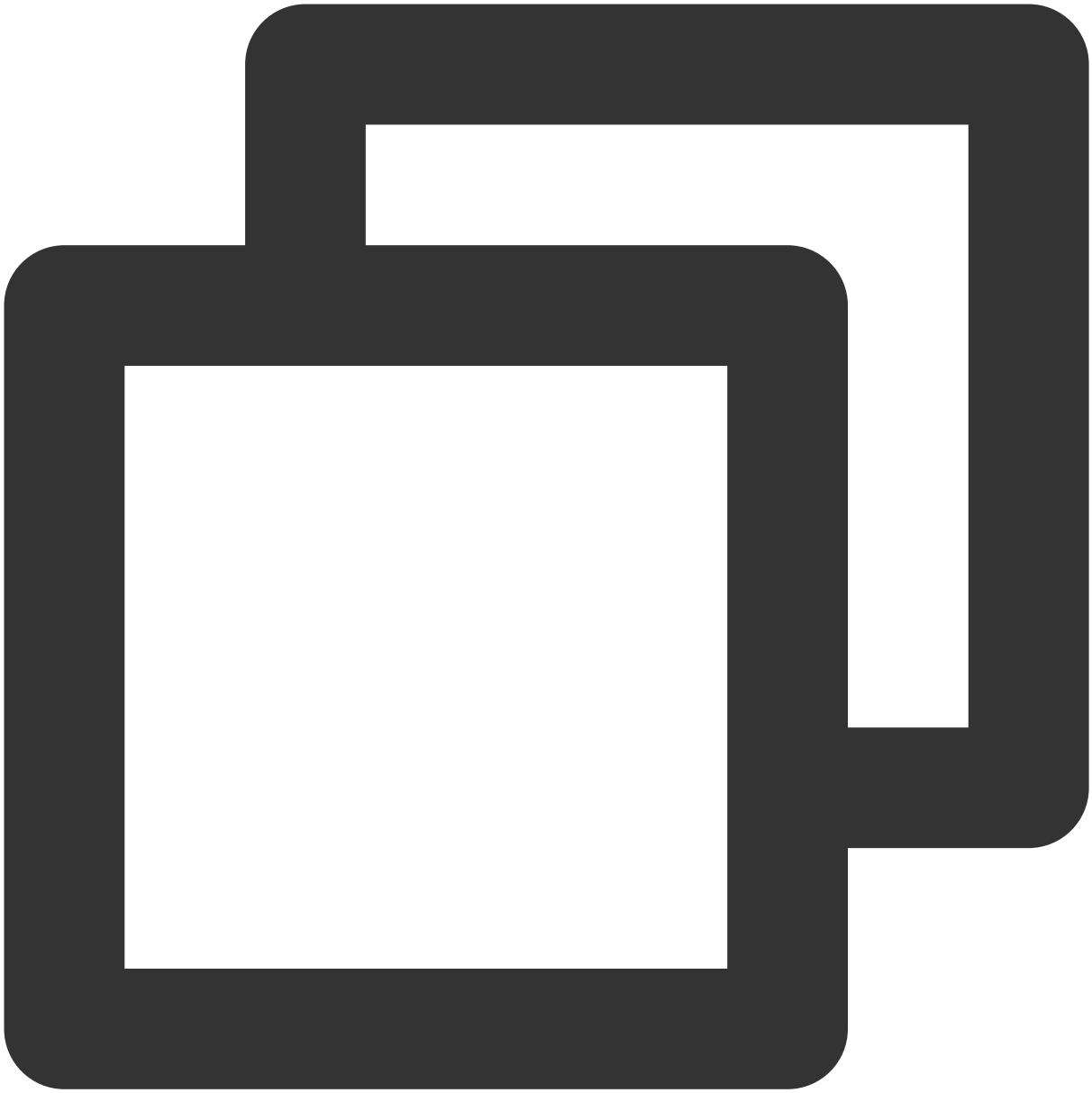
功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:14:31+0800</CreationTime>
        <EndTime>2022-06-30T19:14:36+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jd0c0da74f86511ec8a5a87e016101404</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>2629.749000</Bitrate>
                    <Duration>215.680000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>70898041</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>256.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/48000</CodecTimeBase>
                        <Duration>215.637000</Duration>
                        <Index>1</Index>
                        <Language>eng</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>48000</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/48000</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.000000</AvgFps>
                        <Bitrate>2369.296000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/12288</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>215.583333</Duration>
                        <Fps>24.000000</Fps>
                        <HasBFrame>2</HasBFrame>
```

```
                    <Height>1080</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>40</Level>
                    <NumFrames>5174</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>High</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12288</Timebase>
                    <Width>1920</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>59a4edd95c6ddca43277d82cf33edc0d</Md5>
                    <ObjectName>output/segment-0</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>7fdacd7f2e2dbdc1dc3ccc00f96893b5</Md5>
                    <ObjectName>output/segment-1</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>4c377ddb6f65021f6741dbd45bbff654</Md5>
                    <ObjectName>output/segment-2</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>59317dbc9db6d4d6fb723e3005809c01</Md5>
                    <ObjectName>output/segment-3</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>a81091e57005eece211b5f60bfbddea1</Md5>
                    <ObjectName>output/segment-4</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>7b87e88264a983a06b17558c8a3a2bd8</Md5>
                    <ObjectName>output/segment-5</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>607b2abc29ed11ad1d2f678b594c6ffa</Md5>
                    <ObjectName>output/segment-6</ObjectName>
                </Md5Info>
                <Md5Info>
```

```
                        <Md5>4fe2d33dc9d08a5ea9d5d6db0f724008</Md5>
                        <ObjectName>output/segment-7</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>a84dd4f738ce81f4e3da55b393b2fae4</Md5>
                        <ObjectName>output/segment-8</ObjectName>
                    </Md5Info>
                    <ObjectName>0</ObjectName>
                    <ObjectName>1</ObjectName>
                    <ObjectName>2</ObjectName>
                    <ObjectName>3</ObjectName>
                    <ObjectName>4</ObjectName>
                    <ObjectName>5</ObjectName>
                    <ObjectName>6</ObjectName>
                    <ObjectName>7</ObjectName>
                    <ObjectName>8</ObjectName>
                    <ObjectPrefix>output/segment-</ObjectPrefix>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/segment-${number}</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <Segment>
                <Duration>25</Duration>
                <Format>mp4</Format>
                <HlsEncrypt>
                    <IsHlsEncrypt>false</IsHlsEncrypt>
                    <UriKey/>
                </HlsEncrypt>
            </Segment>
            <UserData>This is my Segment job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:14:32+0800</StartTime>
        <State>Success</State>
        <Tag>Segment</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交转封装任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

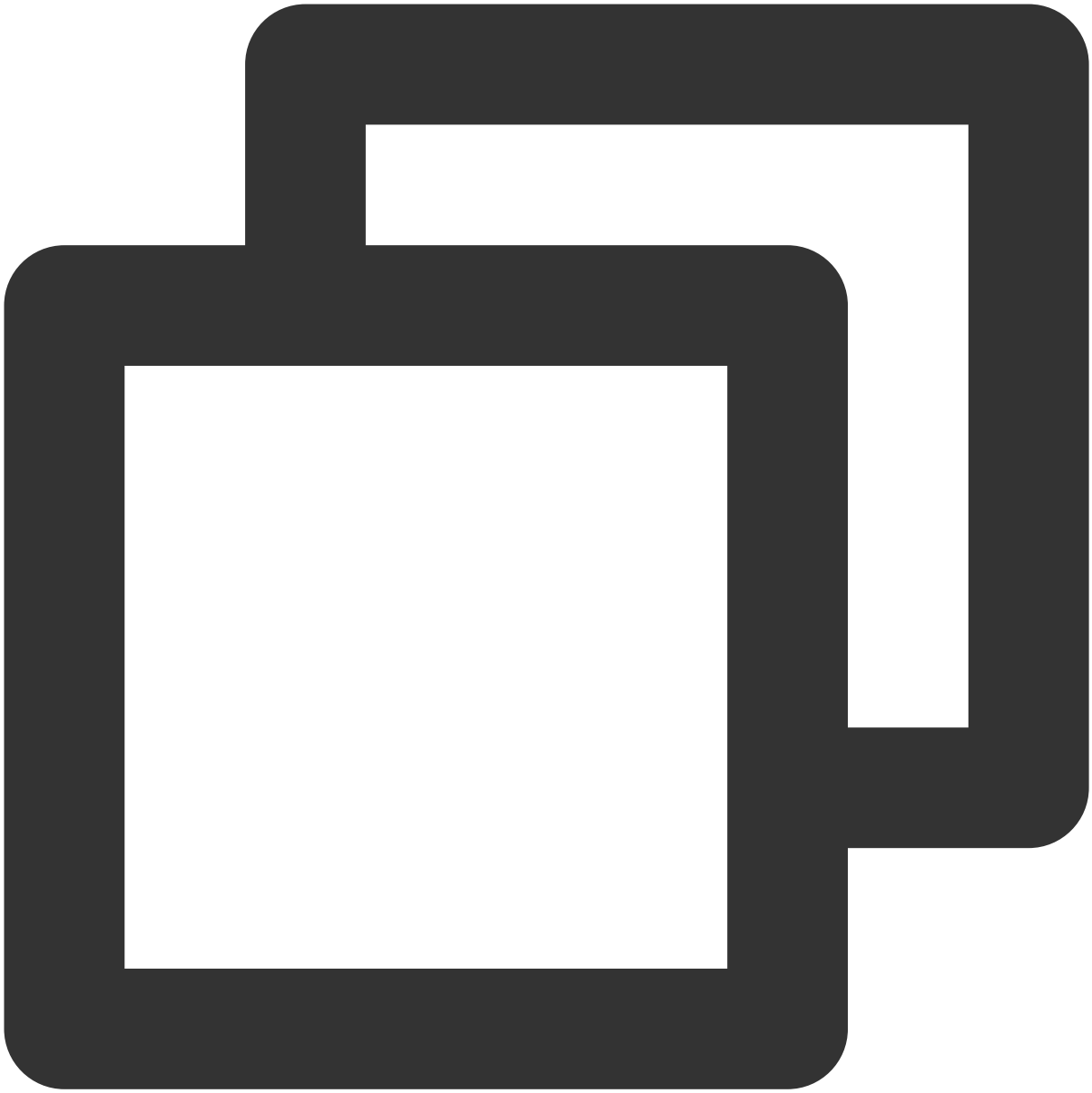
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调



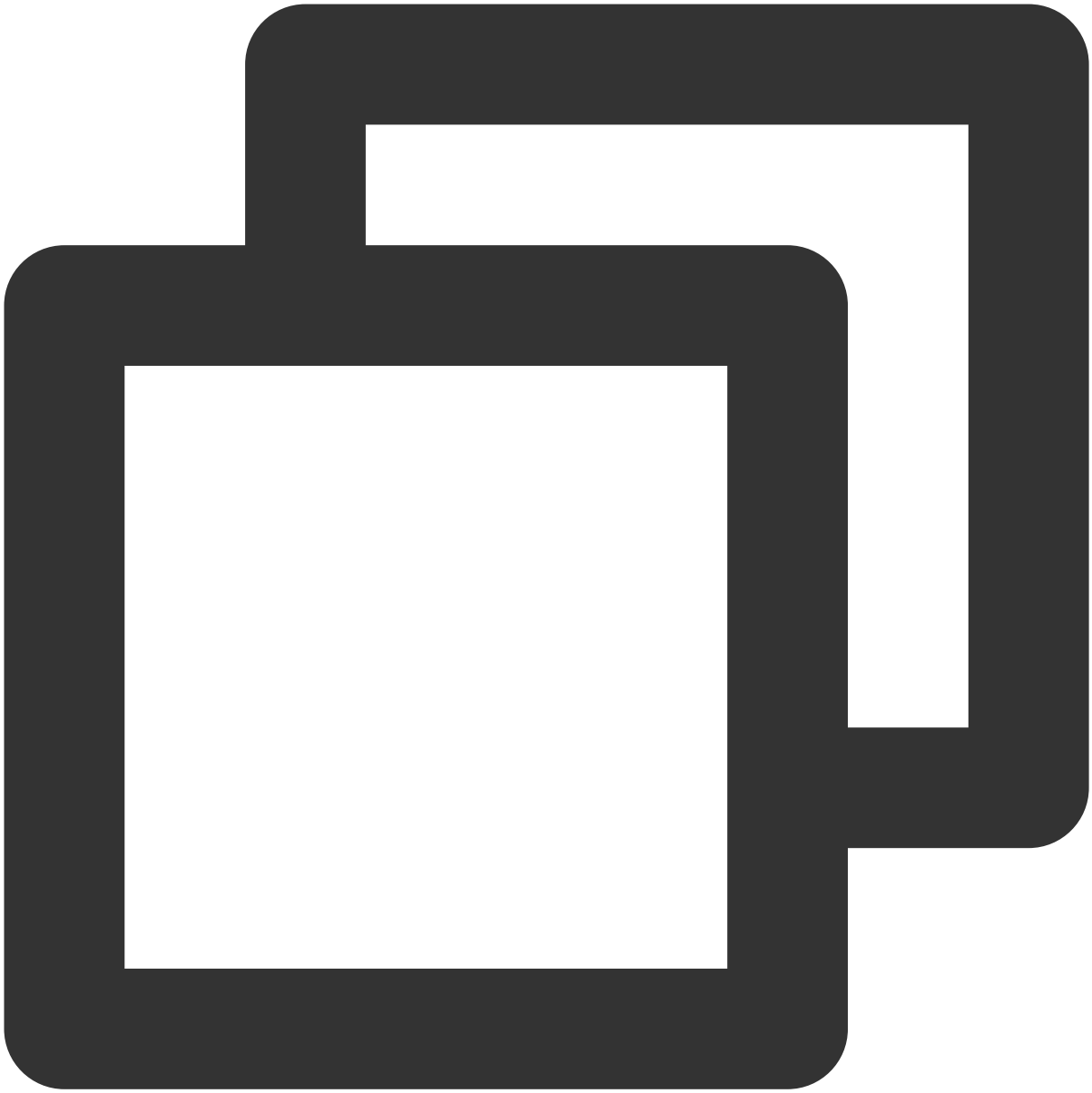
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:14:31+0800</CreationTime>
        <EndTime>2022-06-30T19:14:36+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>jd0c0da74f86511ec8a5a87e016101404</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>2629.749000</Bitrate>
                    <Duration>215.680000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>70898041</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>256.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/48000</CodecTimeBase>
                        <Duration>215.637000</Duration>
                        <Index>1</Index>
                        <Language>eng</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>48000</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/48000</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.000000</AvgFps>
                        <Bitrate>2369.296000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/12288</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>215.583333</Duration>
                        <Fps>24.000000</Fps>
                        <HasBFrame>2</HasBFrame>
```

```
                    <Height>1080</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>40</Level>
                    <NumFrames>5174</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>High</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12288</Timebase>
                    <Width>1920</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>59a4edd95c6ddca43277d82cf33edc0d</Md5>
                    <ObjectName>output/segment-0</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>7fdacd7f2e2dbdc1dc3ccc00f96893b5</Md5>
                    <ObjectName>output/segment-1</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>4c377ddb6f65021f6741dbd45bbff654</Md5>
                    <ObjectName>output/segment-2</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>59317dbc9db6d4d6fb723e3005809c01</Md5>
                    <ObjectName>output/segment-3</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>a81091e57005eece211b5f60bfbddea1</Md5>
                    <ObjectName>output/segment-4</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>7b87e88264a983a06b17558c8a3a2bd8</Md5>
                    <ObjectName>output/segment-5</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>607b2abc29ed11ad1d2f678b594c6ffa</Md5>
                    <ObjectName>output/segment-6</ObjectName>
                </Md5Info>
                <Md5Info>
```

```
                        <Md5>4fe2d33dc9d08a5ea9d5d6db0f724008</Md5>
                        <ObjectName>output/segment-7</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>a84dd4f738ce81f4e3da55b393b2fae4</Md5>
                        <ObjectName>output/segment-8</ObjectName>
                    </Md5Info>
                    <ObjectName>0</ObjectName>
                    <ObjectName>1</ObjectName>
                    <ObjectName>2</ObjectName>
                    <ObjectName>3</ObjectName>
                    <ObjectName>4</ObjectName>
                    <ObjectName>5</ObjectName>
                    <ObjectName>6</ObjectName>
                    <ObjectName>7</ObjectName>
                    <ObjectName>8</ObjectName>
                    <ObjectPrefix>output/segment-</ObjectPrefix>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/segment-${number}</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <Segment>
                <Duration>25</Duration>
                <Format>mp4</Format>
                <HlsEncrypt>
                    <IsHlsEncrypt>false</IsHlsEncrypt>
                    <UriKey/>
                </HlsEncrypt>
            </Segment>
            <UserData>This is my Segment job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:14:32+0800</StartTime>
        <State>Success</State>
        <Tag>Segment</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:14:31+0800</CreationTime>
        <EndTime>2022-06-30T19:14:36+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

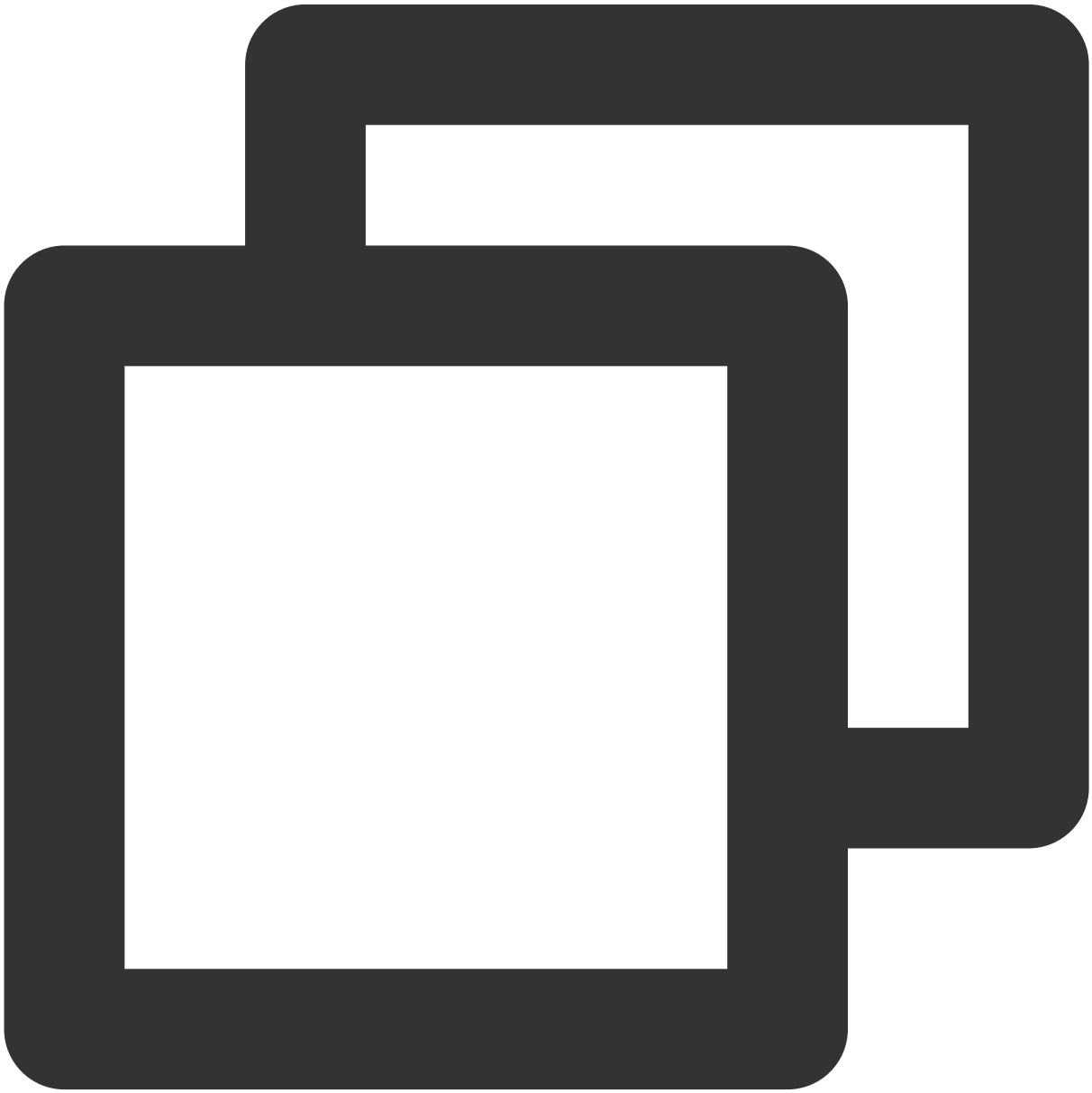
```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>jd0c0da74f86511ec8a5a87e016101404</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>2629.749000</Bitrate>
                    <Duration>215.680000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>70898041</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>256.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/48000</CodecTimeBase>
                        <Duration>215.637000</Duration>
                        <Index>1</Index>
                        <Language>eng</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>48000</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/48000</Timebase>
```

```
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>24.000000</AvgFps>
                    <Bitrate>2369.296000</Bitrate>
                    <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                    <CodecName>h264</CodecName>
                    <CodecTag>0x31637661</CodecTag>
                    <CodecTagString>avc1</CodecTagString>
                    <CodecTimeBase>1/12288</CodecTimeBase>
                    <ColorPrimaries>unknown</ColorPrimaries>
                    <ColorRange>unknown</ColorRange>
                    <ColorTransfer>unknown</ColorTransfer>
                    <Duration>215.583333</Duration>
                    <Fps>24.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>1080</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>40</Level>
                    <NumFrames>5174</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>High</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12288</Timebase>
                    <Width>1920</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>59a4edd95c6ddca43277d82cf33edc0d</Md5>
                    <ObjectName>output/segment-0</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>7fdacd7f2e2dbdc1dc3ccc00f96893b5</Md5>
                    <ObjectName>output/segment-1</ObjectName>
                </Md5Info>
                <Md5Info>
                    <Md5>4c377ddb6f65021f6741dbd45bbff654</Md5>
                    <ObjectName>output/segment-2</ObjectName>
                </Md5Info>
                <Md5Info>
```

```
                <Md5>59317dbc9db6d4d6fb723e3005809c01</Md5>
                <ObjectName>output/segment-3</ObjectName>
            </Md5Info>
            <Md5Info>
                <Md5>a81091e57005eece211b5f60bfbddea1</Md5>
                <ObjectName>output/segment-4</ObjectName>
            </Md5Info>
            <Md5Info>
                <Md5>7b87e88264a983a06b17558c8a3a2bd8</Md5>
                <ObjectName>output/segment-5</ObjectName>
            </Md5Info>
            <Md5Info>
                <Md5>607b2abc29ed11ad1d2f678b594c6ffa</Md5>
                <ObjectName>output/segment-6</ObjectName>
            </Md5Info>
            <Md5Info>
                <Md5>4fe2d33dc9d08a5ea9d5d6db0f724008</Md5>
                <ObjectName>output/segment-7</ObjectName>
            </Md5Info>
            <Md5Info>
                <Md5>a84dd4f738ce81f4e3da55b393b2fae4</Md5>
                <ObjectName>output/segment-8</ObjectName>
            </Md5Info>
            <ObjectName>0</ObjectName>
            <ObjectName>1</ObjectName>
            <ObjectName>2</ObjectName>
            <ObjectName>3</ObjectName>
            <ObjectName>4</ObjectName>
            <ObjectName>5</ObjectName>
            <ObjectName>6</ObjectName>
            <ObjectName>7</ObjectName>
            <ObjectName>8</ObjectName>
            <ObjectPrefix>output/segment-</ObjectPrefix>
            <Region>ap-chongqing</Region>
        </OutputFile>
    </MediaResult>
    <Output>
        <Bucket>test-123456789</Bucket>
        <Object>output/segment-${number}</Object>
        <Region>ap-chongqing</Region>
    </Output>
    <Segment>
        <Duration>25</Duration>
        <Format>mp4</Format>
        <HlsEncrypt>
            <IsHlsEncrypt>false</IsHlsEncrypt>
            <UriKey/>
```

```
                </HlsEncrypt>
            </Segment>
            <UserData>This is my Segment job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:14:32+0800</StartTime>
        <State>Success</State>
        <Tag>Segment</Tag>
        <Workflow>
            <Name>Segment_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:14:31+0800",
        "EndTime": "2022-06-30T19:14:36+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "jd0c0da74f86511ec8a5a87e016101404",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "2629.749000",
                    "Duration": "215.680000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "70898041",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "256.000000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/48000",
                        "Duration": "215.637000",
                        "Index": "1",
                        "Language": "eng",
                        "SampleFmt": "fltp",
                        "SampleRate": "48000",
                        "StartTime": "0.000000",
                        "Timebase": "1/48000"
```

```
                },
                "Video": {
                    "AvgFps": "24.000000",
                    "Bitrate": "2369.296000",
                    "CodecLongName": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10
                    "CodecName": "h264",
                    "CodecTag": "0x31637661",
                    "CodecTagString": "avc1",
                    "CodecTimeBase": "1/12288",
                    "ColorPrimaries": "unknown",
                    "ColorRange": "unknown",
                    "ColorTransfer": "unknown",
                    "Duration": "215.583333",
                    "Fps": "24.000000",
                    "HasBFrame": "2",
                    "Height": "1080",
                    "Index": "0",
                    "Language": "und",
                    "Level": "40",
                    "NumFrames": "5174",
                    "PixFormat": "yuv420p",
                    "Profile": "High",
                    "RefFrames": "1",
                    "Rotation": "0.000000",
                    "StartTime": "0.000000",
                    "Timebase": "1/12288",
                    "Width": "1920"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": [{
                        "Md5": "59a4edd95c6ddca43277d82cf33edc0d",
                        "ObjectName": "output/segment-0"
                    },
                    {
                        "Md5": "7fdacd7f2e2dbdc1dc3ccc00f96893b5",
                        "ObjectName": "output/segment-1"
                    },
                    {
                        "Md5": "4c377ddb6f65021f6741dbd45bbff654",
                        "ObjectName": "output/segment-2"
                    },
                    {
                        "Md5": "59317dbc9db6d4d6fb723e3005809c01",
```

```
                        "ObjectName": "output/segment-3"
                    },
                    {
                        "Md5": "a81091e57005eece211b5f60bfbddea1",
                        "ObjectName": "output/segment-4"
                    },
                    {
                        "Md5": "7b87e88264a983a06b17558c8a3a2bd8",
                        "ObjectName": "output/segment-5"
                    },
                    {
                        "Md5": "607b2abc29ed11ad1d2f678b594c6ffa",
                        "ObjectName": "output/segment-6"
                    },
                    {
                        "Md5": "4fe2d33dc9d08a5ea9d5d6db0f724008",
                        "ObjectName": "output/segment-7"
                    },
                    {
                        "Md5": "a84dd4f738ce81f4e3da55b393b2fae4",
                        "ObjectName": "output/segment-8"
                    }
                ],
                "ObjectName": [
                    "0",
                    "1",
                    "2",
                    "3",
                    "4",
                    "5",
                    "6",
                    "7",
                    "8"
                ],
                "ObjectPrefix": "output/segment-",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/segment-${number}",
            "Region": "ap-chongqing"
        },
        "Segment": {
            "Duration": "25",
            "Format": "mp4",
            "HlsEncrypt": {
```

```
                    "IsHlsEncrypt": "false"
                }
            },
            "UserData": "This is my Segment job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:14:32+0800",
        "State": "Success",
        "Tag": "Segment",
        "Workflow": {
            "Name": "Segment_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

智能封面

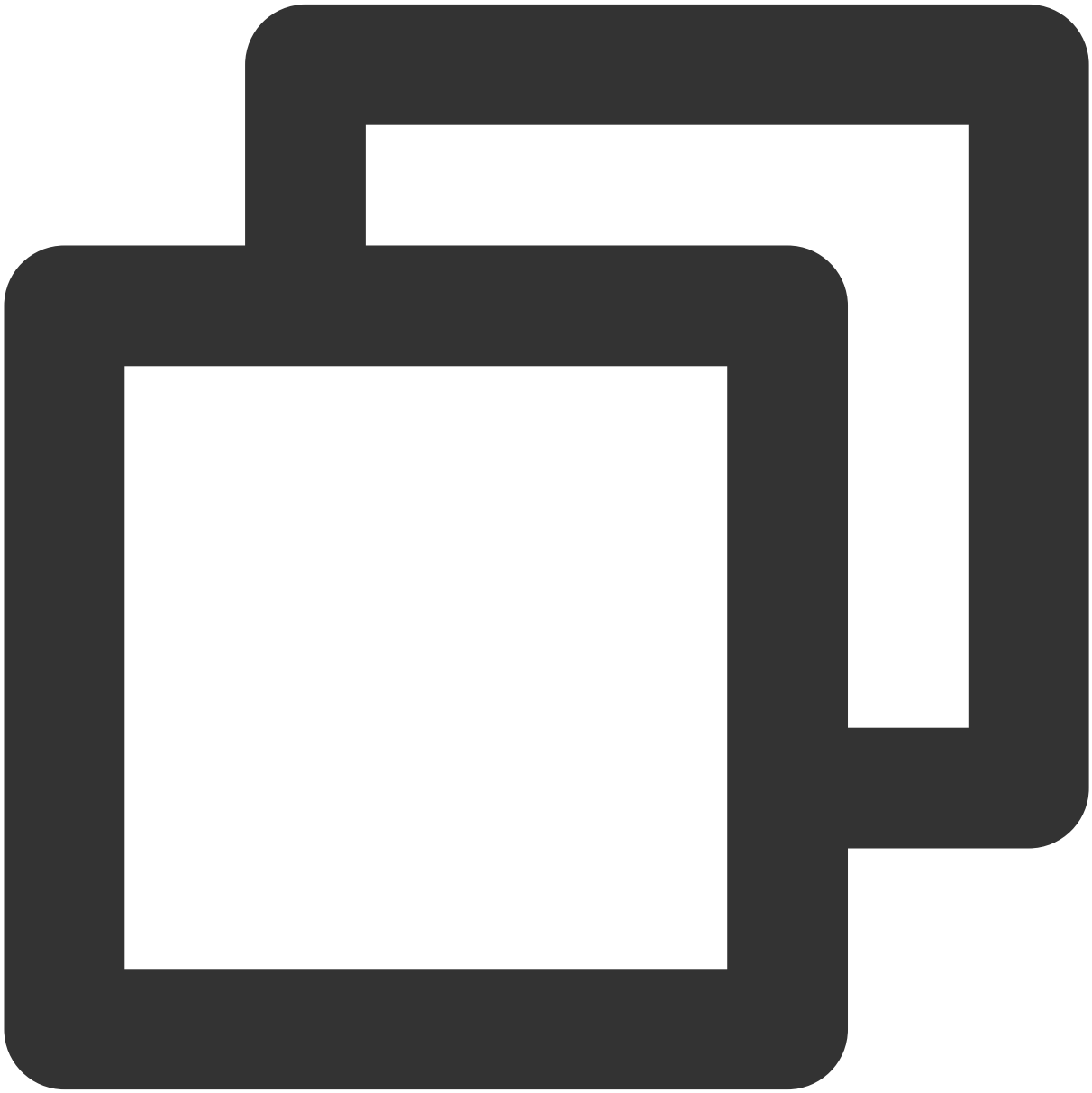
最近更新时间：2023-12-30 16:28:09

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info/>
                    <ObjectName>output/smartcover-0.jpg</ObjectName>
                    <ObjectName>output/smartcover-1.jpg</ObjectName>
                    <ObjectName>output/smartcover-2.jpg</ObjectName>
                    <ObjectName>output/smartcover-3.jpg</ObjectName>
                    <ObjectName>output/smartcover-4.jpg</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SmartCover>
                <Count>5</Count>
                <DeleteDuplicates>false</DeleteDuplicates>
                <Format>jpg</Format>
                <Height>960</Height>
                <Width>1280</Width>
            </SmartCover>
            <UserData>This is my SmartCover job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>SmartCover</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交智能封面任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

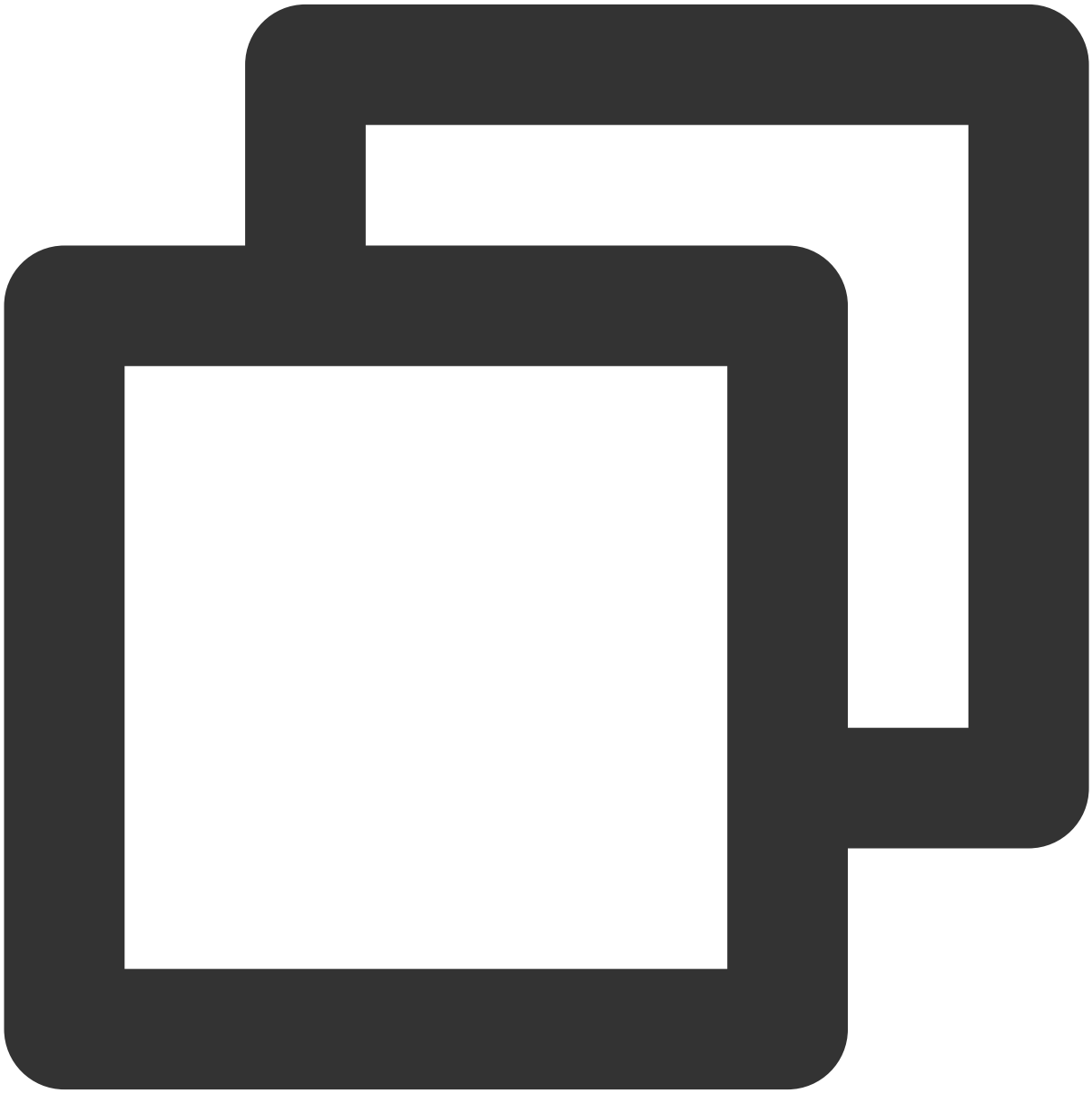
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

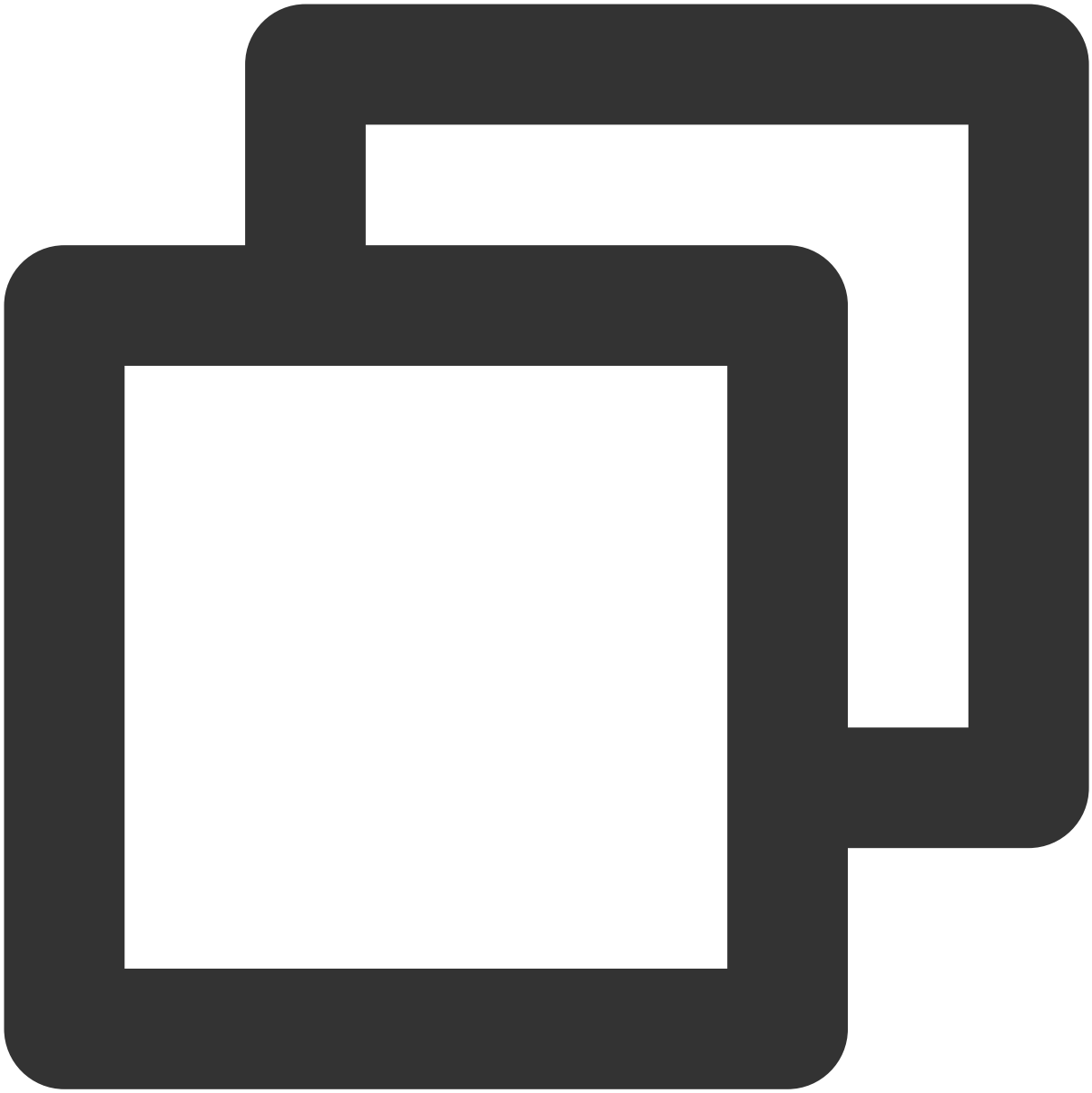
案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info/>
                    <ObjectName>output/smartcover-0.jpg</ObjectName>
                    <ObjectName>output/smartcover-1.jpg</ObjectName>
                    <ObjectName>output/smartcover-2.jpg</ObjectName>
                    <ObjectName>output/smartcover-3.jpg</ObjectName>
                    <ObjectName>output/smartcover-4.jpg</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SmartCover>
                <Count>5</Count>
                <DeleteDuplicates>false</DeleteDuplicates>
                <Format>jpg</Format>
                <Height>960</Height>
                <Width>1280</Width>
            </SmartCover>
            <UserData>This is my SmartCover job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>SmartCover</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调

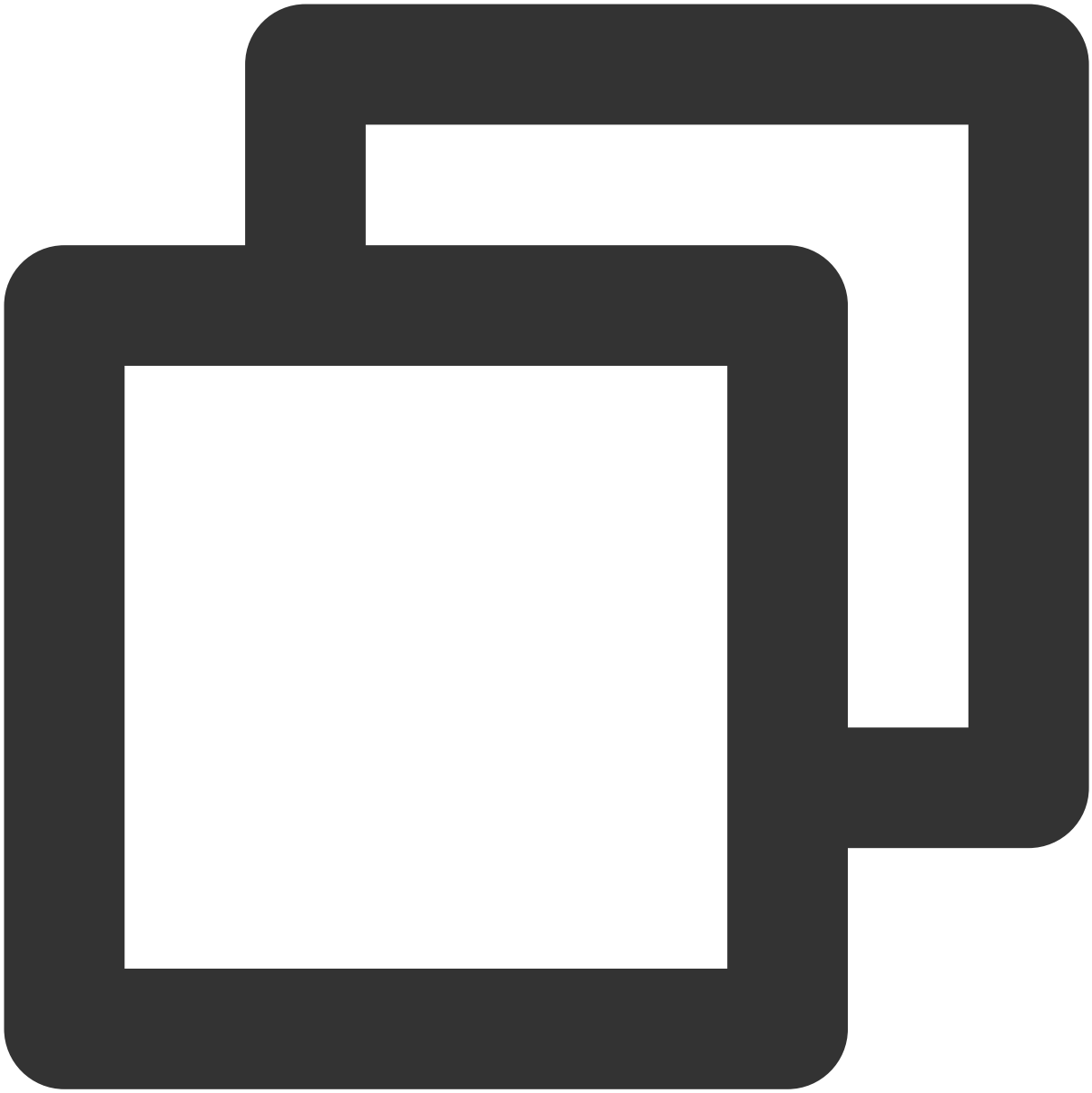


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info/>
                    <ObjectName>output/smartcover-0.jpg</ObjectName>
                    <ObjectName>output/smartcover-1.jpg</ObjectName>
                    <ObjectName>output/smartcover-2.jpg</ObjectName>
                    <ObjectName>output/smartcover-3.jpg</ObjectName>
                    <ObjectName>output/smartcover-4.jpg</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SmartCover>
                <Count>5</Count>
```

```
                <DeleteDuplicates>false</DeleteDuplicates>
                <Format>jpg</Format>
                <Height>960</Height>
                <Width>1280</Width>
            </SmartCover>
            <UserData>This is my SmartCover job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>SmartCover</Tag>
        <Workflow>
            <Name>SmartCover_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T15:29:42+0800",
        "EndTime": "2022-06-30T15:29:48+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j6842765cf84611ecb8546d80f2baf56f",
        "Operation": {
            "MediaInfo": {
                "Stream": {

                }
            },
            "MediaResult": {
                "OutputFile": {
                    "Bucket": "test-123456789",
                    "ObjectName": [
                        "output/smartcover-0.jpg",
                        "output/smartcover-1.jpg",
                        "output/smartcover-2.jpg",
                        "output/smartcover-3.jpg",
                        "output/smartcover-4.jpg"
                    ],
                    "Region": "ap-chongqing"
                }
            },
            "Output": {
                "Bucket": "test-123456789",
                "Object": "output/smartcover-${Number}.jpg",
                "Region": "ap-chongqing"
            },
            "SmartCover": {
                "Count": "5",
                "DeleteDuplicates": "false",
                "Format": "jpg",
                "Height": "960",
```

```
                "Width": "1280"
            },
            "UserData": "This is my SmartCover job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T15:29:43+0800",
        "State": "Success",
        "Tag": "SmartCover",
        "Workflow": {
            "Name": "SmartCover_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

截帧

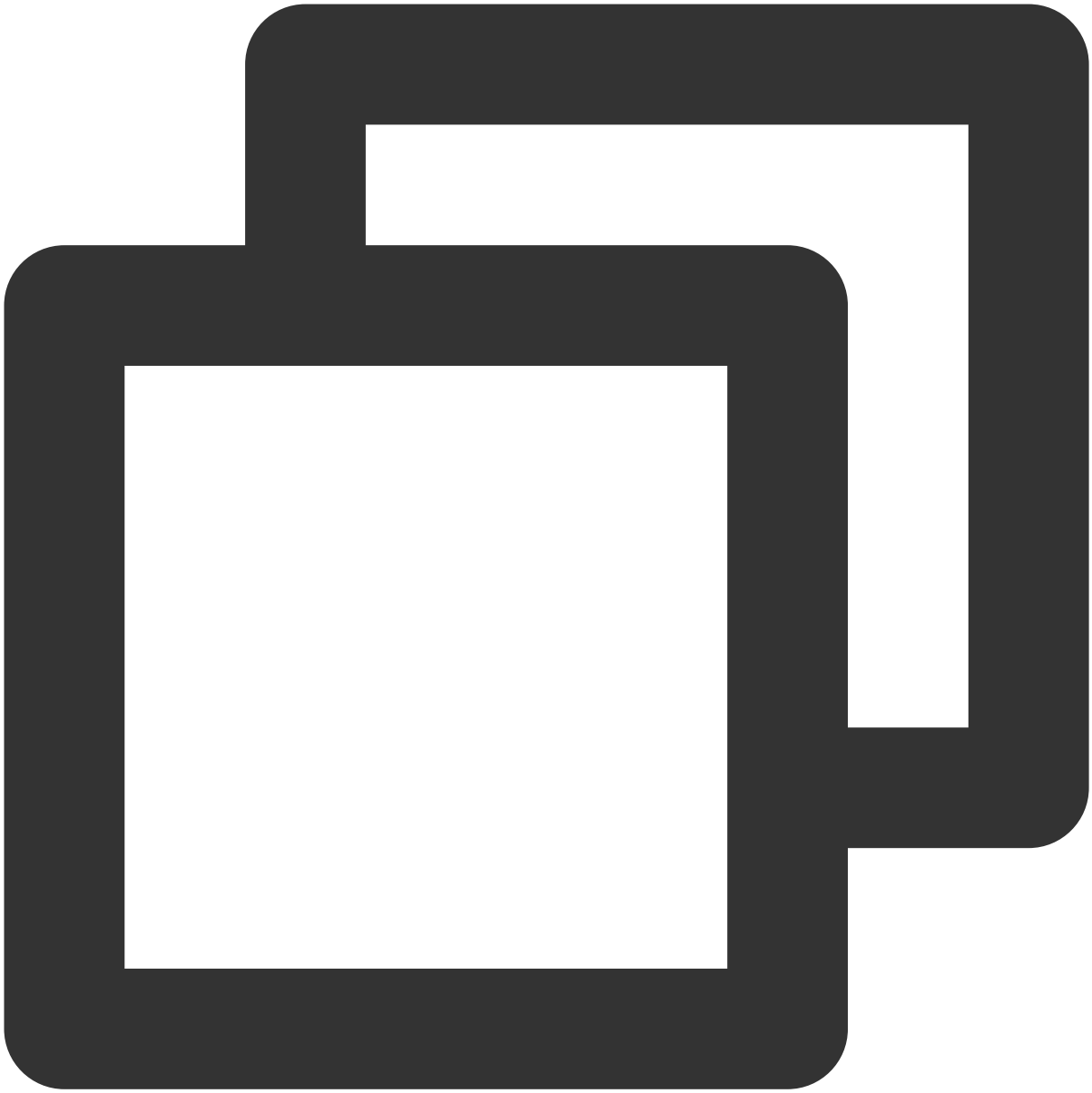
最近更新时间：2023-12-30 16:29:03

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:25:11+0800</CreationTime>
        <EndTime>2022-06-30T19:25:14+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j4e431ba0f86711ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info/>
                    <ObjectName>0.jpg</ObjectName>
                    <ObjectName>1.jpg</ObjectName>
                    <ObjectName>2.jpg</ObjectName>
                    <ObjectName>3.jpg</ObjectName>
                    <ObjectName>4.jpg</ObjectName>
                    <ObjectName>5.jpg</ObjectName>
                    <ObjectPrefix>output/snapshot-</ObjectPrefix>
                    <Region>ap-chongqing</Region>
                    <SpriteOutputFile>
                        <Bucket>test-123456789</Bucket>
                        <Md5Info/>
                        <ObjectName>0.jpg</ObjectName>
                        <ObjectPrefix>output/sprite-</ObjectPrefix>
                        <Region>ap-chongqing</Region>
                    </SpriteOutputFile>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/snapshot-${Number}.jpg</Object>
                <Region>ap-chongqing</Region>
                <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
            </Output>
            <TemplateId>t1f16e1dfbdc9ds4105b31632d45710642a</TemplateId>
            <TemplateName>template_snapshot</TemplateName>
            <UserData>This is my Snapshot job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:25:12+0800</StartTime>
        <State>Success</State>
        <Tag>Snapshot</Tag>
    </JobsDetail>
```

```
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交截图任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

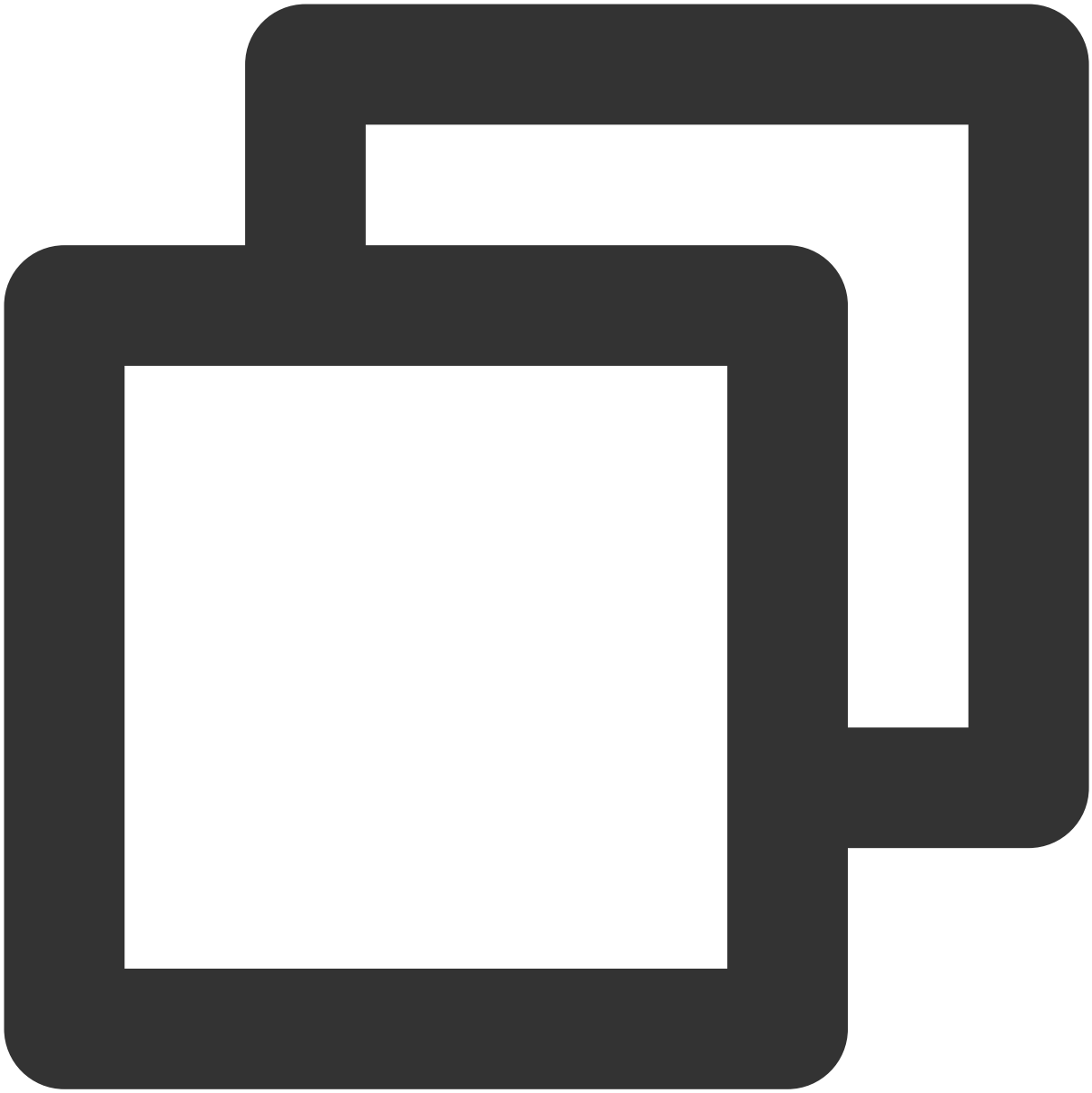
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

版权所有：腾讯云计算（北京）有限责任公司

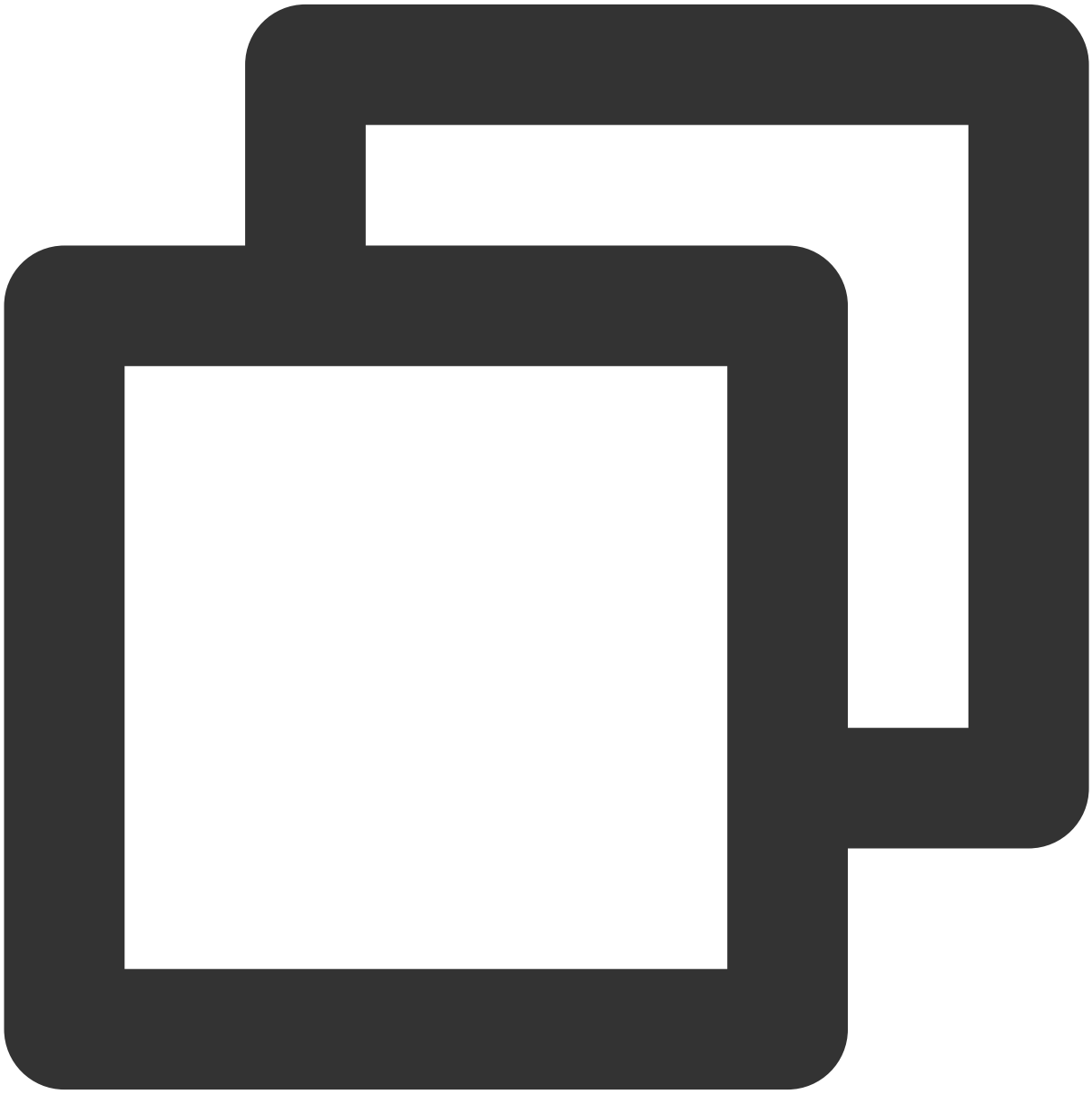


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:25:11+0800</CreationTime>
        <EndTime>2022-06-30T19:25:14+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j4e431ba0f86711ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info/>
                    <ObjectName>0.jpg</ObjectName>
                    <ObjectName>1.jpg</ObjectName>
                    <ObjectName>2.jpg</ObjectName>
                    <ObjectName>3.jpg</ObjectName>
                    <ObjectName>4.jpg</ObjectName>
                    <ObjectName>5.jpg</ObjectName>
                    <ObjectPrefix>output/snapshot-</ObjectPrefix>
                    <Region>ap-chongqing</Region>
                    <SpriteOutputFile>
                        <Bucket>test-123456789</Bucket>
                        <Md5Info/>
                        <ObjectName>0.jpg</ObjectName>
                        <ObjectPrefix>output/sprite-</ObjectPrefix>
                        <Region>ap-chongqing</Region>
                    </SpriteOutputFile>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/snapshot-${Number}.jpg</Object>
                <Region>ap-chongqing</Region>
                <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
            </Output>
            <TemplateId>t1f16e1dfbdc9ds4105b31632d45710642a</TemplateId>
            <TemplateName>template_snapshot</TemplateName>
            <UserData>This is my Snapshot job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:25:12+0800</StartTime>
        <State>Success</State>
        <Tag>Snapshot</Tag>
    </JobsDetail>
```

```
</Response>
```

案例 2：通过工作流触发的任务回调

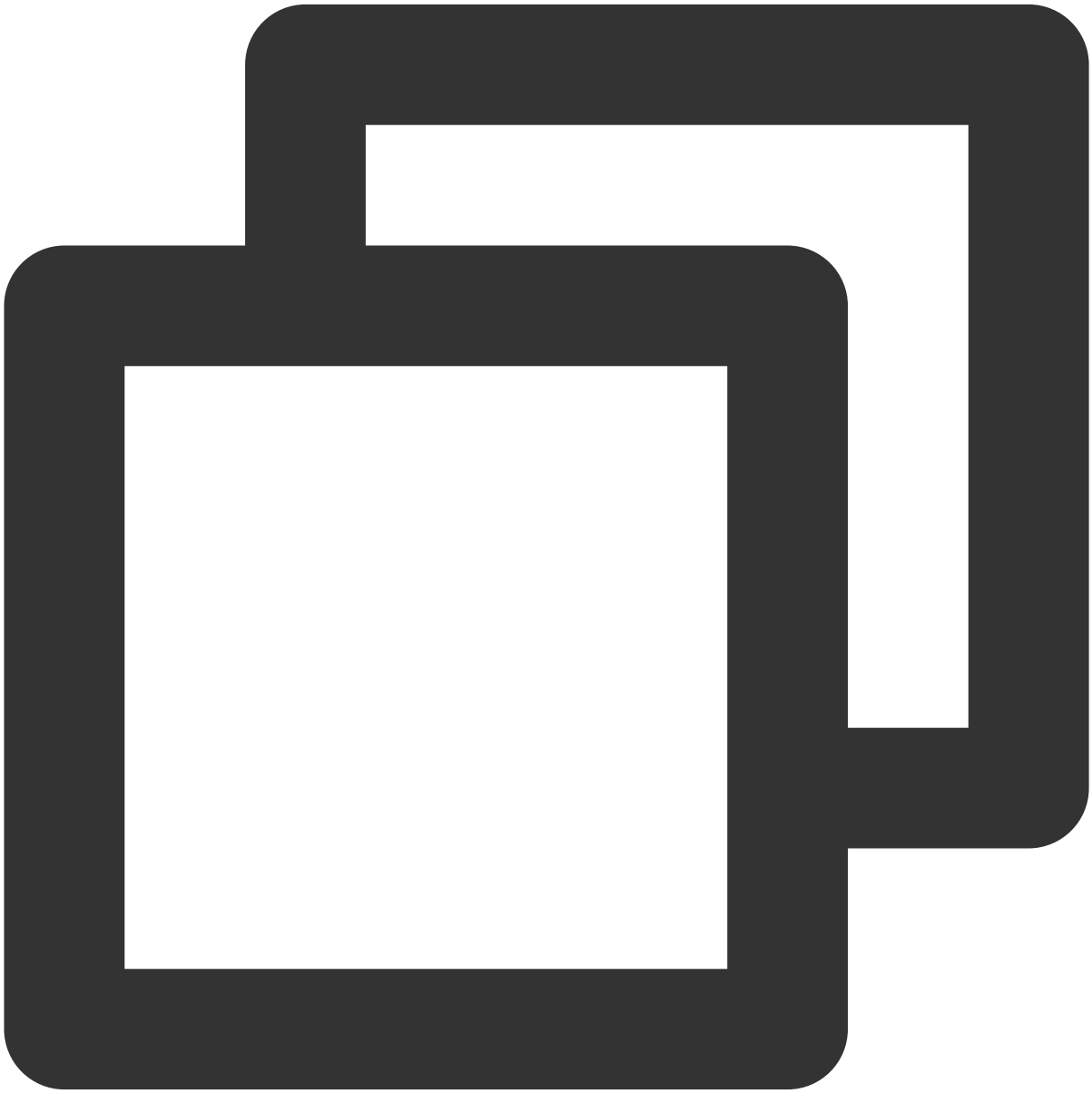


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:25:11+0800</CreationTime>
        <EndTime>2022-06-30T19:25:14+0800</EndTime>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j4e431ba0f86711ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info/>
                    <ObjectName>0.jpg</ObjectName>
                    <ObjectName>1.jpg</ObjectName>
                    <ObjectName>2.jpg</ObjectName>
                    <ObjectName>3.jpg</ObjectName>
                    <ObjectName>4.jpg</ObjectName>
                    <ObjectName>5.jpg</ObjectName>
                    <ObjectPrefix>output/snapshot-</ObjectPrefix>
                    <Region>ap-chongqing</Region>
                    <SpriteOutputFile>
                        <Bucket>test-123456789</Bucket>
                        <Md5Info/>
```

```
                        <ObjectName>0.jpg</ObjectName>
                        <ObjectPrefix>output/sprite-</ObjectPrefix>
                        <Region>ap-chongqing</Region>
                    </SpriteOutputFile>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/snapshot-${Number}.jpg</Object>
                <Region>ap-chongqing</Region>
                <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
            </Output>
            <TemplateId>t1f16e1dfbdc9ds4105b31632d45710642a</TemplateId>
            <TemplateName>template_snapshot</TemplateName>
            <UserData>This is my Snapshot job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:25:12+0800</StartTime>
        <State>Success</State>
        <Tag>Snapshot</Tag>
        <Workflow>
            <Name>Snapshot_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:25:11+0800",
        "EndTime": "2022-06-30T19:25:14+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

版权所有：腾讯云计算（北京）有限责任公司

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j4e431ba0f86711ecb8546d80f2baf56f",
        "Operation": {
            "MediaInfo": {
                "Stream": {

                }
            },
            "MediaResult": {
                "OutputFile": {
                    "Bucket": "test-123456789",
                    "ObjectName": [
                        "0.jpg",
                        "1.jpg",
                        "2.jpg",
                        "3.jpg",
                        "4.jpg",
                        "5.jpg"
                    ],
                    "ObjectPrefix": "output/snapshot-",
                    "Region": "ap-chongqing",
                    "SpriteOutputFile": {
                        "Bucket": "test-123456789",
                        "ObjectName": "0.jpg",
                        "ObjectPrefix": "output/sprite-",
                        "Region": "ap-chongqing"
                    }
                }
            },
            "Output": {
                "Bucket": "test-123456789",
```

```
                "Object": "output/snapshot-${Number}.jpg",
                "Region": "ap-chongqing",
                "SpriteObject": "output/sprite-${Number}.jpg"
            },
            "TemplateId": "t1f16e1dfbdc9ds4105b31632d45710642a",
            "TemplateName": "template_snapshot",
            "UserData": "This is my Snapshot job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:25:12+0800",
        "State": "Success",
        "Tag": "Snapshot",
        "Workflow": {
            "Name": "Snapshot_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

流分离

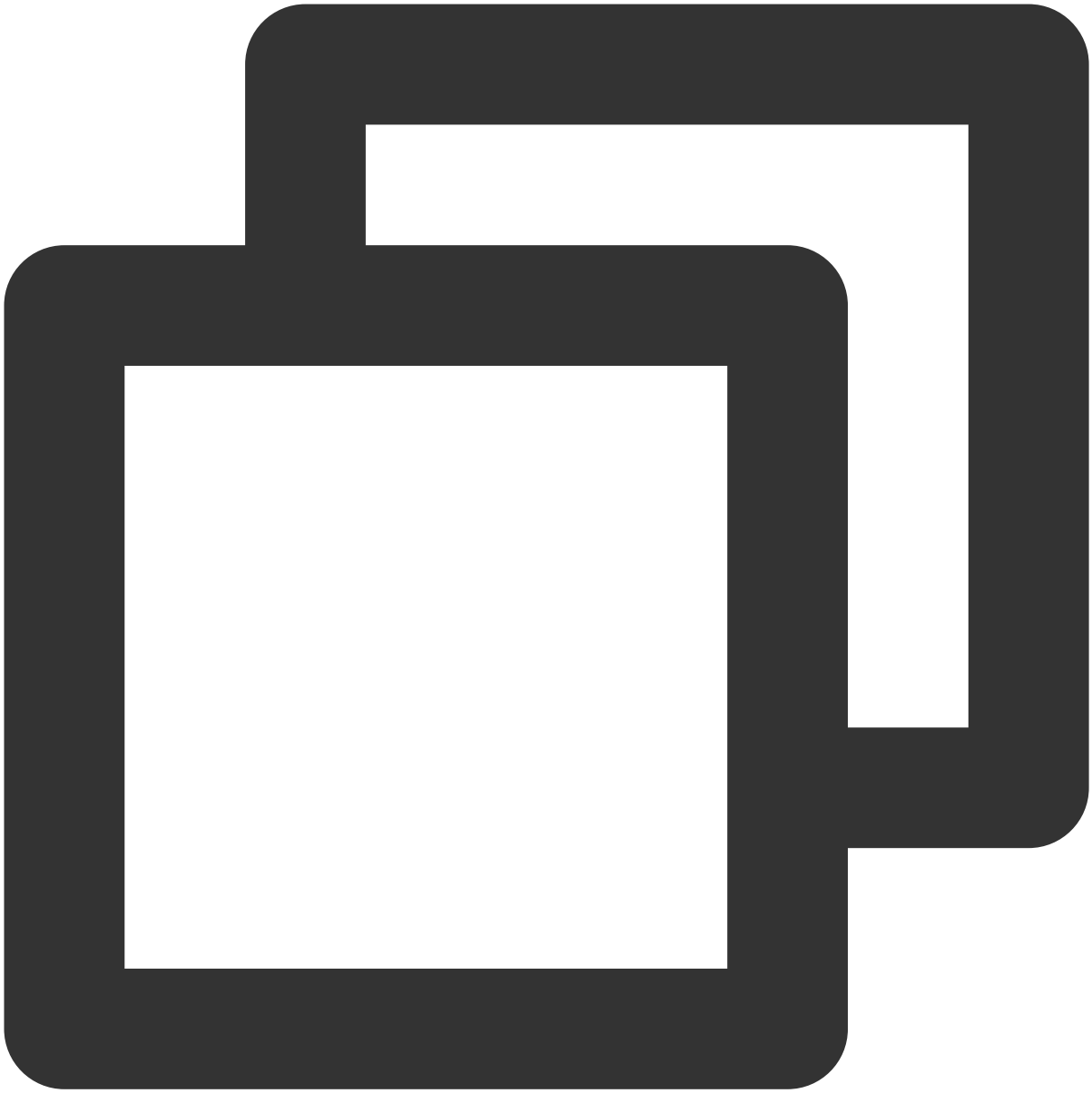
最近更新时间：2023-12-30 16:28:57

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:27:39+0800</CreationTime>
        <EndTime>2022-06-30T19:27:42+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>ja66d6204f86711ec9ca1073b78d316d3</JobId>
        <Message/>
        <Operation>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>e8c802ed78ba7f4c6a2fb5144cba7cbc</Md5>
                        <ObjectName>output/out0.mp4</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>f5883c1380524eecd1a925657ab17b4c</Md5>
                        <ObjectName>output/out1.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/out0.mp4</ObjectName>
                    <ObjectName>output/out1.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Region>ap-chongqing</Region>
                <StreamExtract>
                    <Index>0</Index>
                    <Output>output/out0.mp4</Output>
                </StreamExtract>
                <StreamExtract>
                    <Index>1</Index>
                    <Output>output/out1.mp4</Output>
                </StreamExtract>
                <StreamExtract>
                    <Index>2</Index>
                    <Output>output/out2.mp4</Output>
                </StreamExtract>
            </Output>
            <UserData>This is my StreamExtract job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:27:40+0800</StartTime>
        <State>Success</State>
        <Tag>StreamExtract</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交流分离任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

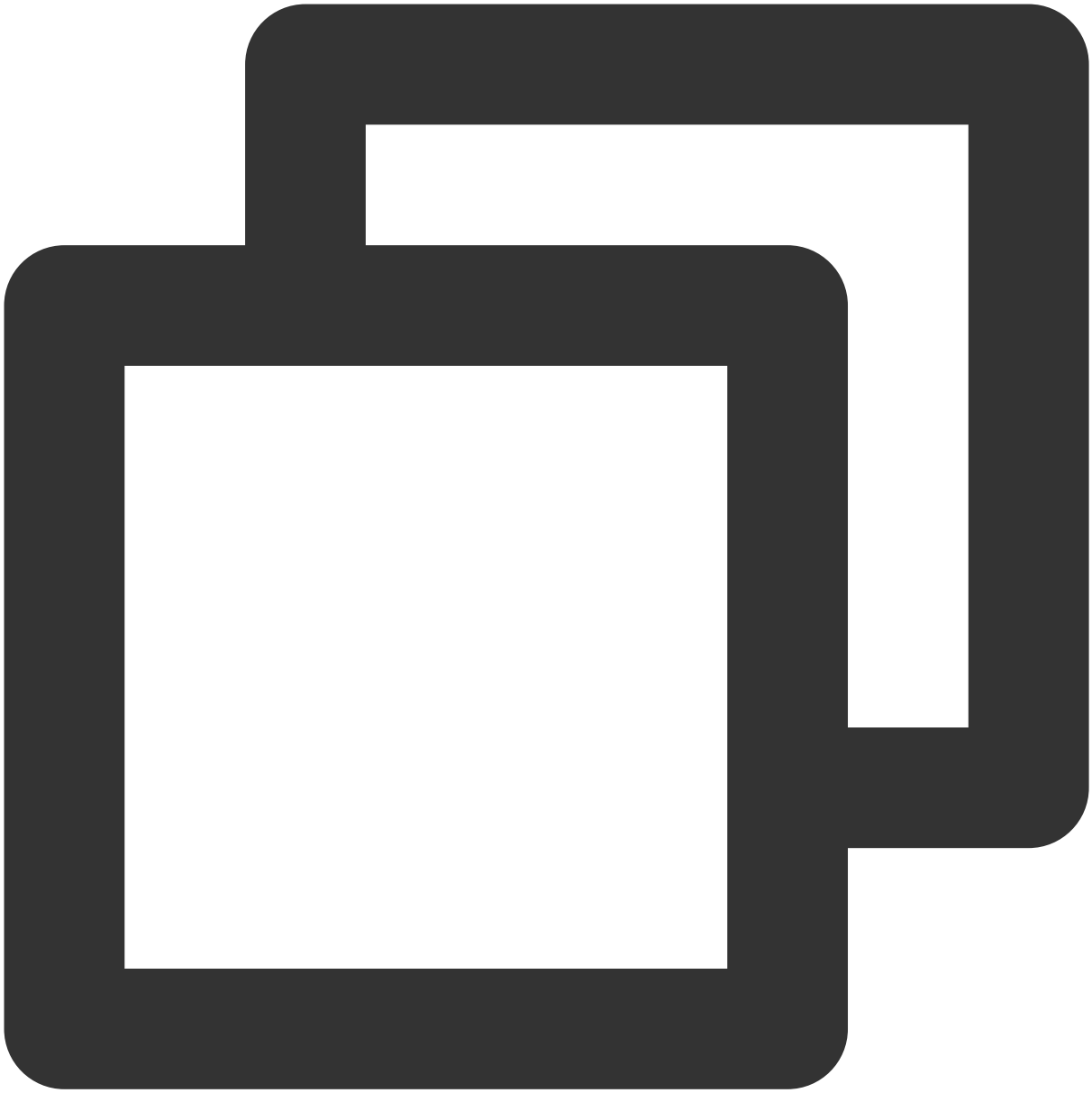
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

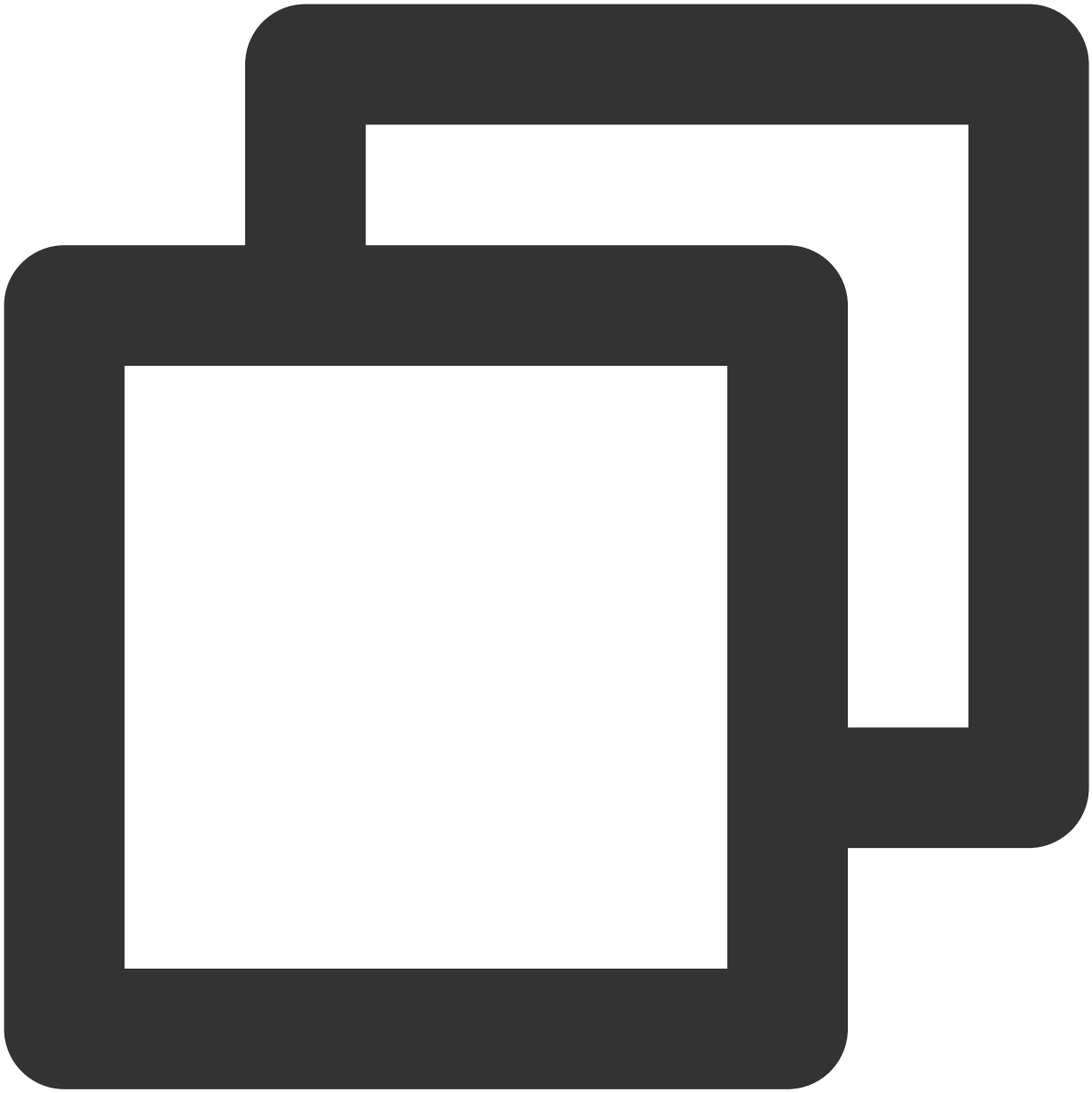
案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:27:39+0800</CreationTime>
        <EndTime>2022-06-30T19:27:42+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>ja66d6204f86711ec9ca1073b78d316d3</JobId>
        <Message/>
        <Operation>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>e8c802ed78ba7f4c6a2fb5144cba7cbc</Md5>
                        <ObjectName>output/out0.mp4</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>f5883c1380524eecd1a925657ab17b4c</Md5>
                        <ObjectName>output/out1.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/out0.mp4</ObjectName>
                    <ObjectName>output/out1.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Region>ap-chongqing</Region>
                <StreamExtract>
                    <Index>0</Index>
                    <Output>output/out0.mp4</Output>
                </StreamExtract>
                <StreamExtract>
                    <Index>1</Index>
                    <Output>output/out1.mp4</Output>
                </StreamExtract>
                <StreamExtract>
                    <Index>2</Index>
                    <Output>output/out2.mp4</Output>
                </StreamExtract>
            </Output>
            <UserData>This is my StreamExtract job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:27:40+0800</StartTime>
        <State>Success</State>
        <Tag>StreamExtract</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:27:39+0800",
        "EndTime": "2022-06-30T19:27:42+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing"
        },
```


版权所有：腾讯云计算（北京）有限责任公司

```
        "JobId": "ja66d6204f86711ec9ca1073b78d316d3",
        "Operation": {
            "MediaResult": {
                "OutputFile": {
                    "Bucket": "test-123456789",
                    "Md5Info": [{
                            "Md5": "e8c802ed78ba7f4c6a2fb5144cba7cbc",
                            "ObjectName": "output/out0.mp4"
                        },
                        {
                            "Md5": "f5883c1380524eecd1a925657ab17b4c",
                            "ObjectName": "output/out1.mp4"
                        }
                    ],
                    "ObjectName": [
                        "output/out0.mp4",
                        "output/out1.mp4"
                    ],
                    "Region": "ap-chongqing"
                }
            },
            "Output": {
                "Bucket": "test-123456789",
                "Region": "ap-chongqing",
                "StreamExtract": [{
                            "Index": "0",
                            "Output": "output/out0.mp4"
                        },
                        {
                            "Index": "1",
                            "Output": "output/out1.mp4"
                        },
                        {
                            "Index": "2",
                            "Output": "output/out2.mp4"
                        }
                    ]
            },
            "UserData": "This is my StreamExtract job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:27:40+0800",
        "State": "Success",
        "Tag": "StreamExtract"
    }
}
```


版权所有：腾讯云计算（北京）有限责任公司

版权所有：腾讯云计算（北京）有限责任公司

超分辨率

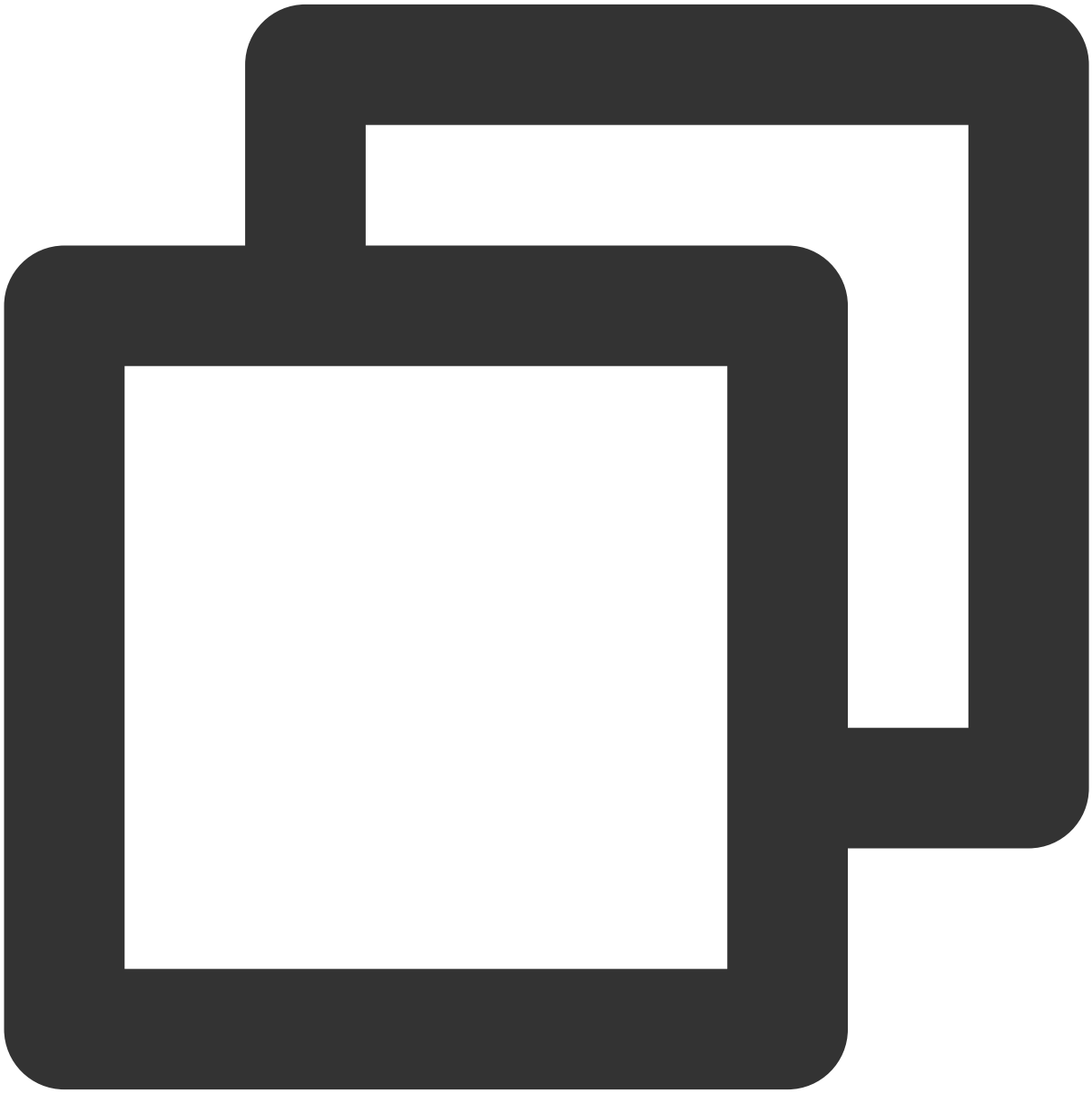
最近更新时间：2023-12-30 16:29:50

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:30:20+0800</CreationTime>
        <EndTime>2022-06-30T19:31:56+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j06668dc0f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <DigitalWatermark>
                <IgnoreError>false</IgnoreError>
                <Message>123456789ab</Message>
                <State>Failed</State>
                <Type>Text</Type>
                <Version>V1</Version>
            </DigitalWatermark>
            <MediaInfo>
                <Format>
                    <Bitrate>8867.172000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>15134046</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000000</AvgFps>
                        <Bitrate>9197.180000</Bitrate>
                        <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                        <CodecName>hevc</CodecName>
                        <CodecTag>0x31766568</CodecTag>
                        <CodecTagString>hev1</CodecTagString>
```

```
                    <CodecTimeBase>1/12800</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>12.960000</Duration>
                    <FieldOrder>progressive</FieldOrder>
                    <Fps>25.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>1920</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>120</Level>
                    <NumFrames>324</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>Main</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>1088</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>852883012a6ba726e6ed8d9b984edfdf</Md5>
                    <ObjectName>output/super_resolution.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/super_resolution.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/super_resolution.mp4</Object>
            <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1f1ae1dfsdc9ds41dsb31632d45710642a</TemplateId>
        <TemplateName>template_superresolution</TemplateName>
        <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
        <UserData>This is my SuperResolution job.</UserData>
        <WatermarkTemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</WatermarkTempl
        <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
    </Operation>
```

```
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:30:21+0800</StartTime>
        <State>Success</State>
        <SubTag>DigitalWatermark</SubTag>
        <Tag>SuperResolution</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交超级分辨率任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

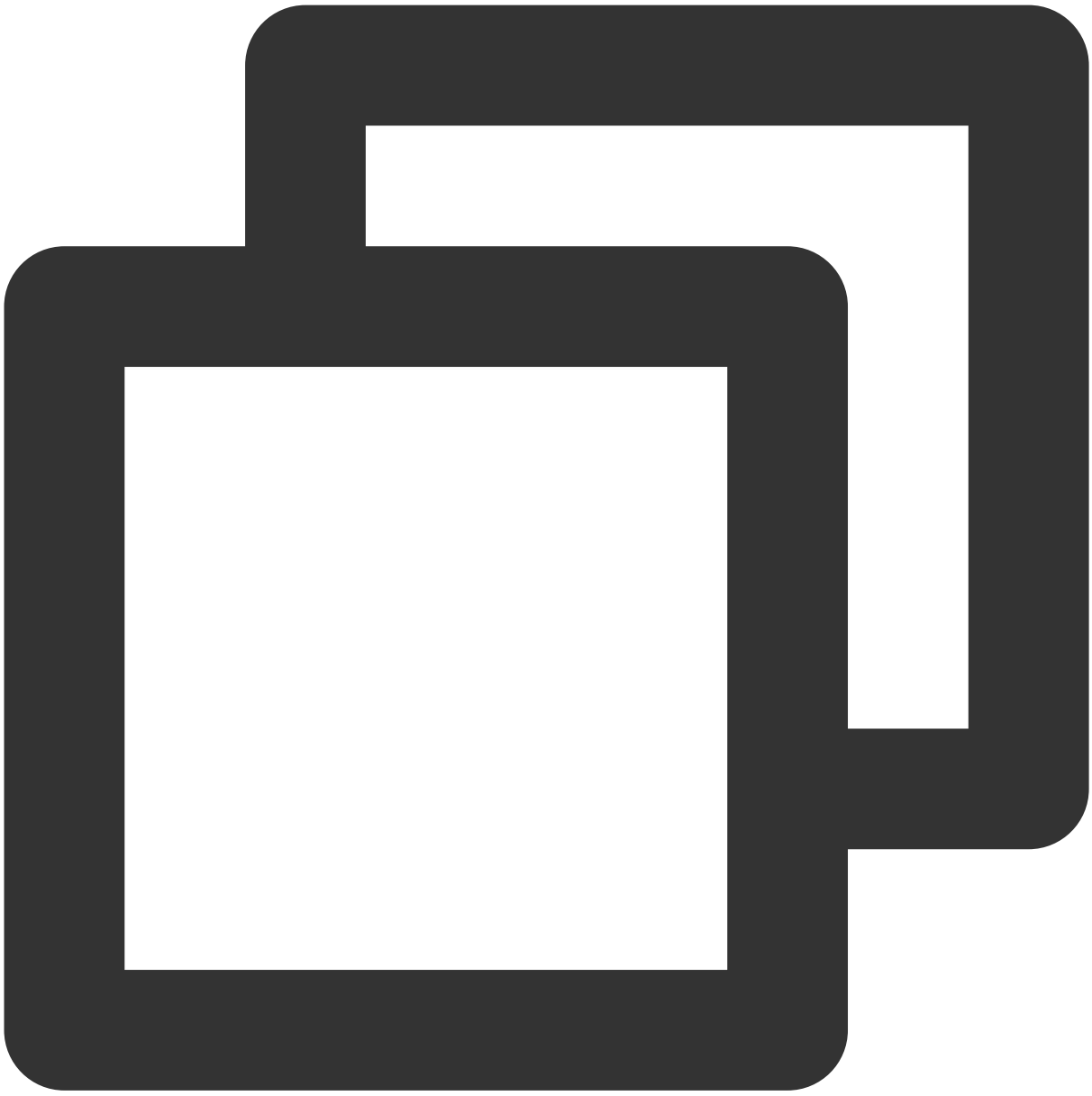
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

版权所有：腾讯云计算（北京）有限责任公司

实际案例

案例 1：通过任务接口触发的任务回调



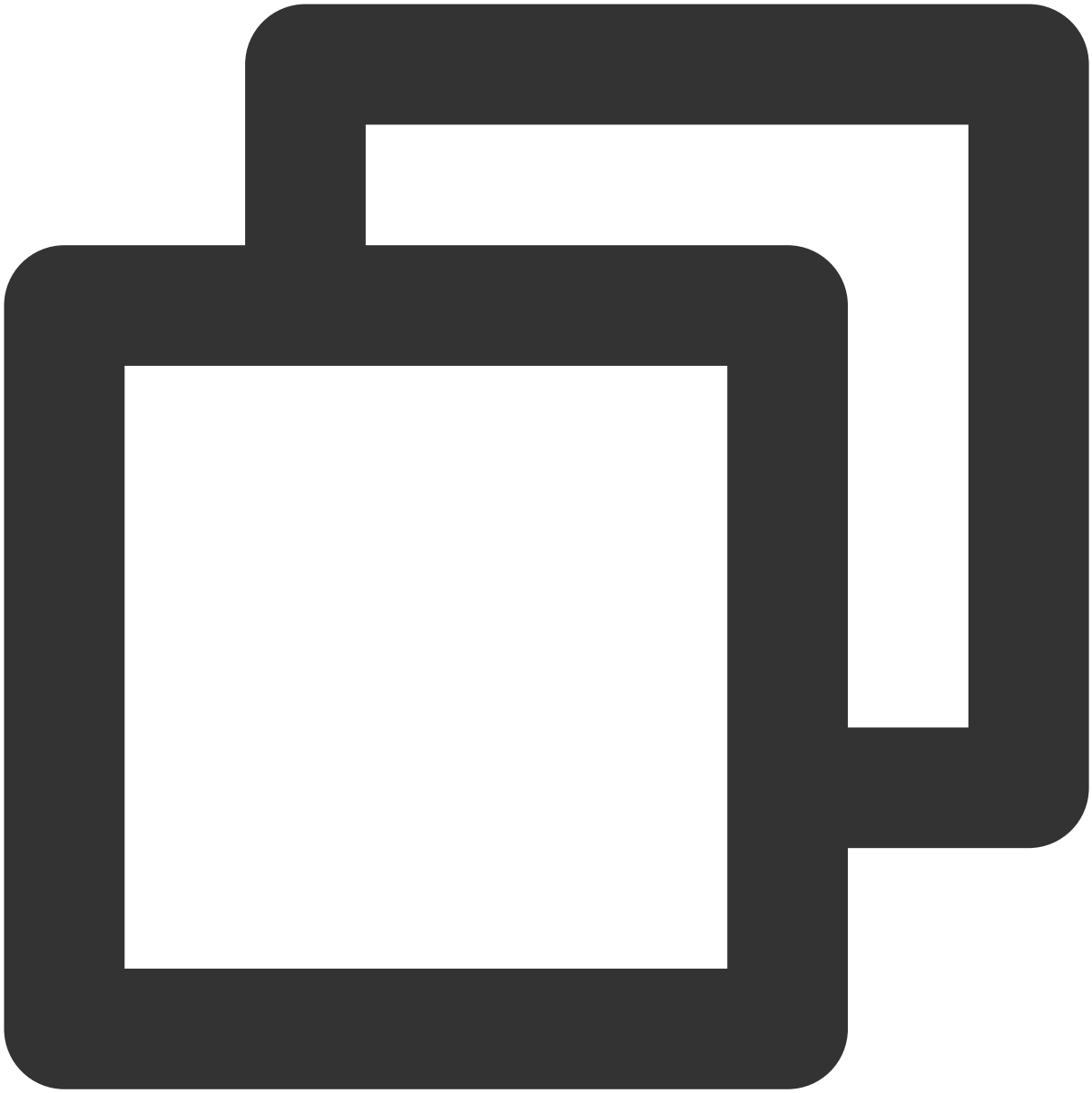
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:30:20+0800</CreationTime>
        <EndTime>2022-06-30T19:31:56+0800</EndTime>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <JobId>j06668dc0f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <DigitalWatermark>
                <IgnoreError>false</IgnoreError>
                <Message>123456789ab</Message>
                <State>Failed</State>
                <Type>Text</Type>
                <Version>V1</Version>
            </DigitalWatermark>
            <MediaInfo>
                <Format>
                    <Bitrate>8867.172000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>15134046</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000000</AvgFps>
```

```
                    <Bitrate>9197.180000</Bitrate>
                    <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                    <CodecName>hevc</CodecName>
                    <CodecTag>0x31766568</CodecTag>
                    <CodecTagString>hev1</CodecTagString>
                    <CodecTimeBase>1/12800</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>12.960000</Duration>
                    <FieldOrder>progressive</FieldOrder>
                    <Fps>25.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>1920</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>120</Level>
                    <NumFrames>324</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>Main</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>1088</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>852883012a6ba726e6ed8d9b984edfdf</Md5>
                    <ObjectName>output/super_resolution.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/super_resolution.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/super_resolution.mp4</Object>
            <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1f1ae1dfsdc9ds41dsb31632d45710642a</TemplateId>
        <TemplateName>template_superresolution</TemplateName>
```

```
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <UserData>This is my SuperResolution job.</UserData>
            <WatermarkTemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</WatermarkTempl
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:30:21+0800</StartTime>
        <State>Success</State>
        <SubTag>DigitalWatermark</SubTag>
        <Tag>SuperResolution</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



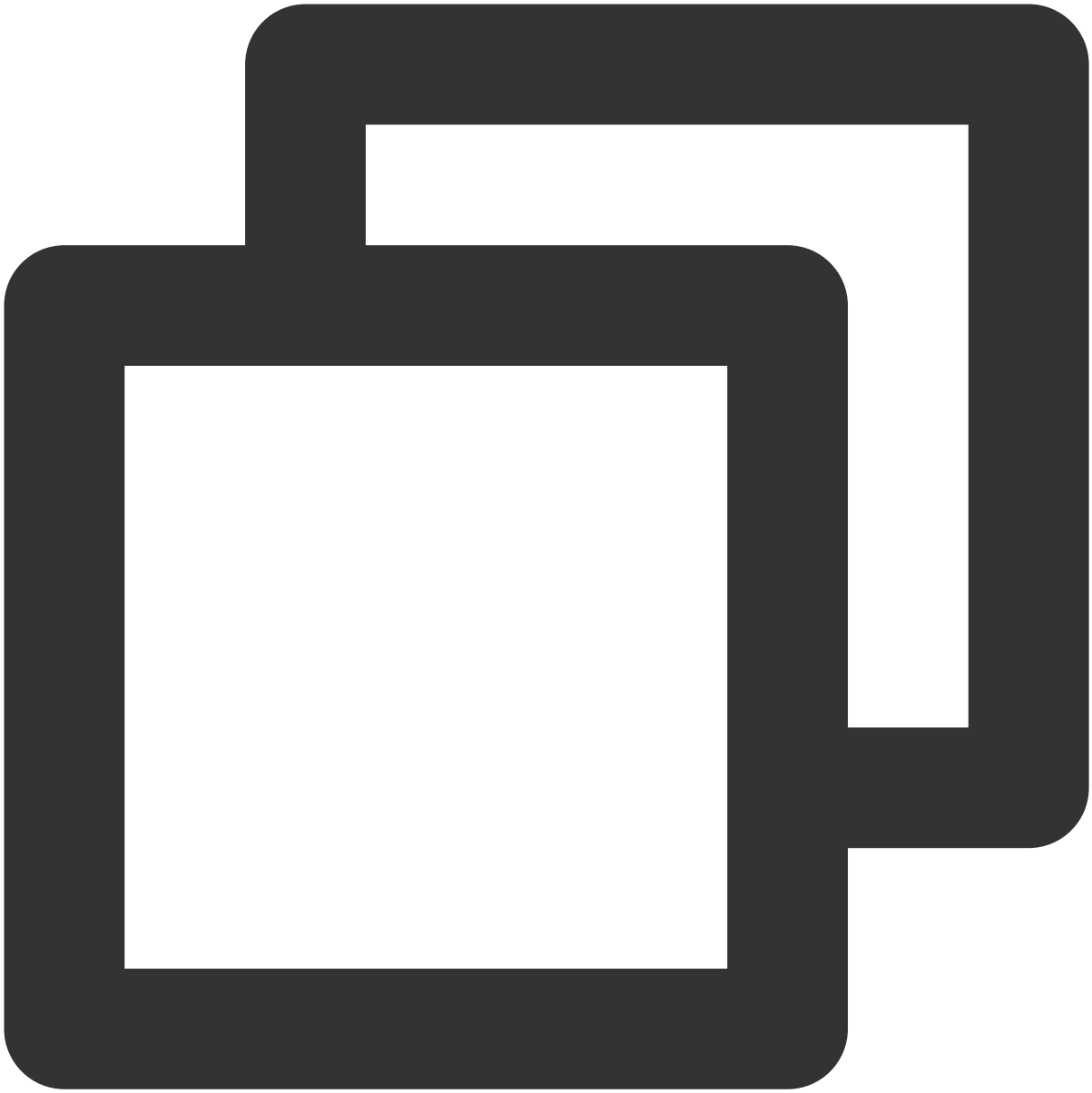
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:30:20+0800</CreationTime>
        <EndTime>2022-06-30T19:31:56+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j06668dc0f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <DigitalWatermark>
                <IgnoreError>false</IgnoreError>
                <Message>123456789ab</Message>
                <State>Failed</State>
                <Type>Text</Type>
                <Version>V1</Version>
            </DigitalWatermark>
            <MediaInfo>
                <Format>
                    <Bitrate>8867.172000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>15134046</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
```

```
                    <Duration>13.652993</Duration>
                    <Index>1</Index>
                    <Language>und</Language>
                    <SampleFmt>fltp</SampleFmt>
                    <SampleRate>44100</SampleRate>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/44100</Timebase>
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>25.000000</AvgFps>
                    <Bitrate>9197.180000</Bitrate>
                    <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                    <CodecName>hevc</CodecName>
                    <CodecTag>0x31766568</CodecTag>
                    <CodecTagString>hev1</CodecTagString>
                    <CodecTimeBase>1/12800</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>12.960000</Duration>
                    <FieldOrder>progressive</FieldOrder>
                    <Fps>25.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>1920</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>120</Level>
                    <NumFrames>324</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>Main</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>1088</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>852883012a6ba726e6ed8d9b984edfdf</Md5>
                    <ObjectName>output/super_resolution.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/super_resolution.mp4</ObjectName>
```

```
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/super_resolution.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1f1ae1dfsdc9ds41dsb31632d45710642a</TemplateId>
            <TemplateName>template_superresolution</TemplateName>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <UserData>This is my SuperResolution job.</UserData>
            <WatermarkTemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</WatermarkTempl
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:30:21+0800</StartTime>
        <State>Success</State>
        <SubTag>DigitalWatermark</SubTag>
        <Tag>SuperResolution</Tag>
        <Workflow>
            <Name>SuperResolution_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:30:20+0800",
        "EndTime": "2022-06-30T19:31:56+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j06668dc0f86811ecb90d0b03267ce0e5",
        "Operation": {
            "DigitalWatermark": {
                "IgnoreError": "false",
                "Message": "123456789ab",
                "State": "Failed",
                "Type": "Text",
                "Version": "V1"
            },
            "MediaInfo": {
                "Format": {
                    "Bitrate": "8867.172000",
                    "Duration": "13.654000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "15134046",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "128.726000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/44100",
```

```
                    "Duration": "13.652993",
                    "Index": "1",
                    "Language": "und",
                    "SampleFmt": "fltp",
                    "SampleRate": "44100",
                    "StartTime": "0.000000",
                    "Timebase": "1/44100"
                },
                "Video": {
                    "AvgFps": "25.000000",
                    "Bitrate": "9197.180000",
                    "CodecLongName": "H.265 / HEVC (High Efficiency Video Codin
                    "CodecName": "hevc",
                    "CodecTag": "0x31766568",
                    "CodecTagString": "hev1",
                    "CodecTimeBase": "1/12800",
                    "ColorPrimaries": "bt470bg",
                    "ColorRange": "tv",
                    "ColorTransfer": "smpte170m",
                    "Duration": "12.960000",
                    "FieldOrder": "progressive",
                    "Fps": "25.000000",
                    "HasBFrame": "2",
                    "Height": "1920",
                    "Index": "0",
                    "Language": "und",
                    "Level": "120",
                    "NumFrames": "324",
                    "PixFormat": "yuv420p",
                    "Profile": "Main",
                    "RefFrames": "1",
                    "Rotation": "0.000000",
                    "StartTime": "0.000000",
                    "Timebase": "1/12800",
                    "Width": "1088"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": {
                    "Md5": "852883012a6ba726e6ed8d9b984edfdf",
                    "ObjectName": "output/super_resolution.mp4"
                },
                "ObjectName": "output/super_resolution.mp4",
                "Region": "ap-chongqing"
```

```
                }
            },
            "Output": {
                "Bucket": "test-123456789",
                "Object": "output/super_resolution.mp4",
                "Region": "ap-chongqing"
            },
            "TemplateId": "t1f1ae1dfsdc9ds41dsb31632d45710642a",
            "TemplateName": "template_superresolution",
            "TranscodeTemplateId": "t156c107210e7243c5817354565d81b578",
            "UserData": "This is my SuperResolution job.",
            "WatermarkTemplateId": [
                "t143ae6e040af6431aa772c9ec3f0a3f36",
                "t12a74d11687d444deba8a6cc52051ac27"
            ]
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:30:21+0800",
        "State": "Success",
        "SubTag": "DigitalWatermark",
        "Tag": "SuperResolution",
        "Workflow": {
            "Name": "SuperResolution_1581665960537_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

音视频转码

最近更新时间：2023-12-30 16:29:36

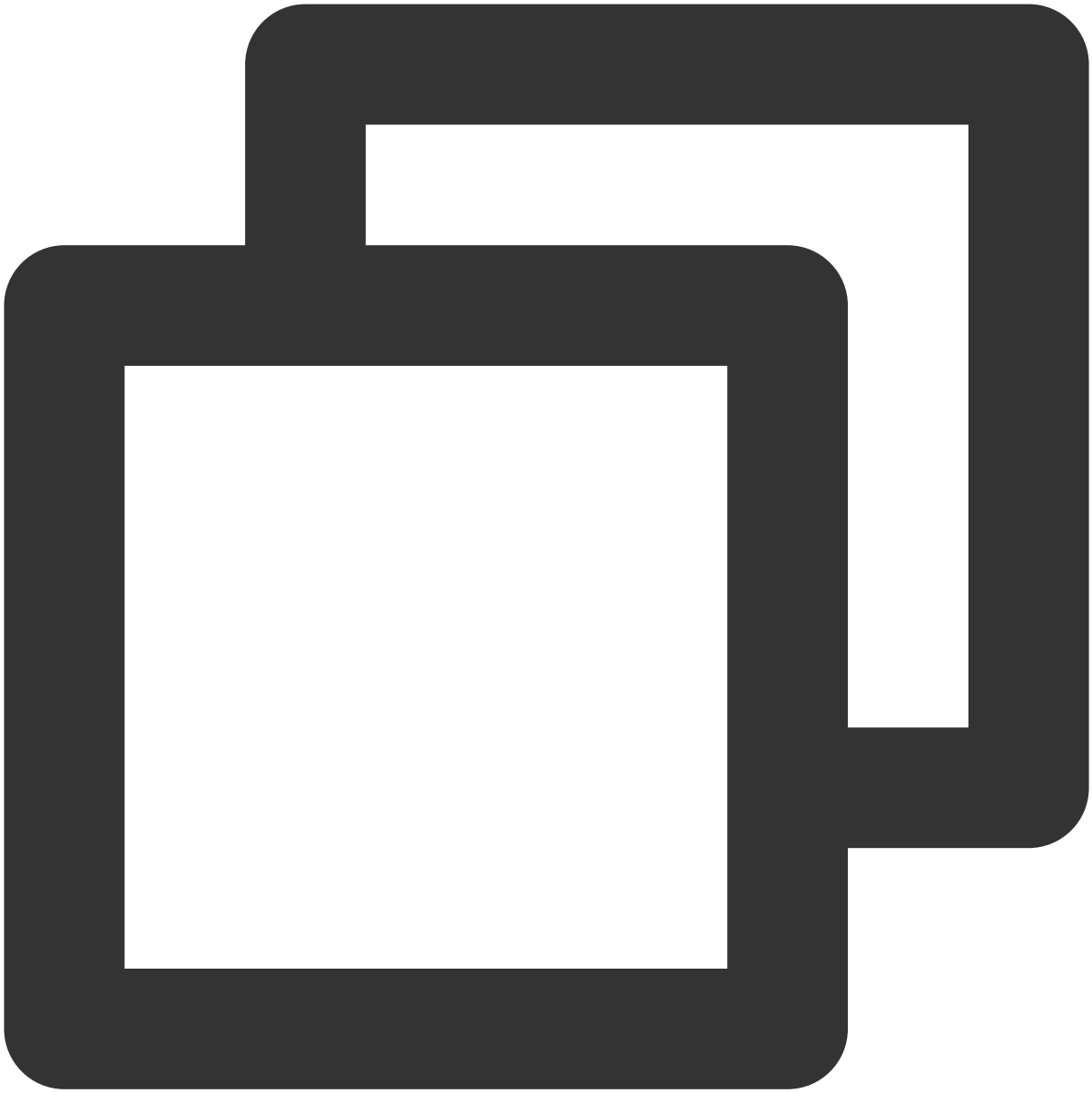
功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:30:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j68427030f84611ecb8546d80f2baf56f</JobId>
        <Message>success</Message>
        <Operation>
            <MediaInfo>
                <Format>
                    <Duration>13</Duration>
                    <NumStream>1</NumStream>
                    <Size>1909389</Size>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.653</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>50.000</AvgFps>
                        <Bitrate>1031968</Bitrate>
                        <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                        <CodecName>hevc</CodecName>
                        <CodecTag>0x31637668</CodecTag>
                        <CodecTagString>hvc1</CodecTagString>
                        <CodecTimeBase>1/50</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960</Duration>
                        <FieldOrder>Unknown</FieldOrder>
                        <Fps>50/1</Fps>
                        <HasBFrame>4</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>4</Level>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                        <NumFrames>648</NumFrames>
                        <PixFormat>hvc1</PixFormat>
                        <Profile>Main</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>540</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>2db78d168edf2f883adce89a1b4ca033</Md5>
                        <ObjectName>output/transcode.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/transcode.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/transcode.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
            <TemplateName>trans_993874</TemplateName>
            <UserData>This is my Transcode job.</UserData>
        </Operation>
        <Progress>100</Progress>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:42+0800</StartTime>
        <State>Success</State>
        <Tag>Transcode</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交转码任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

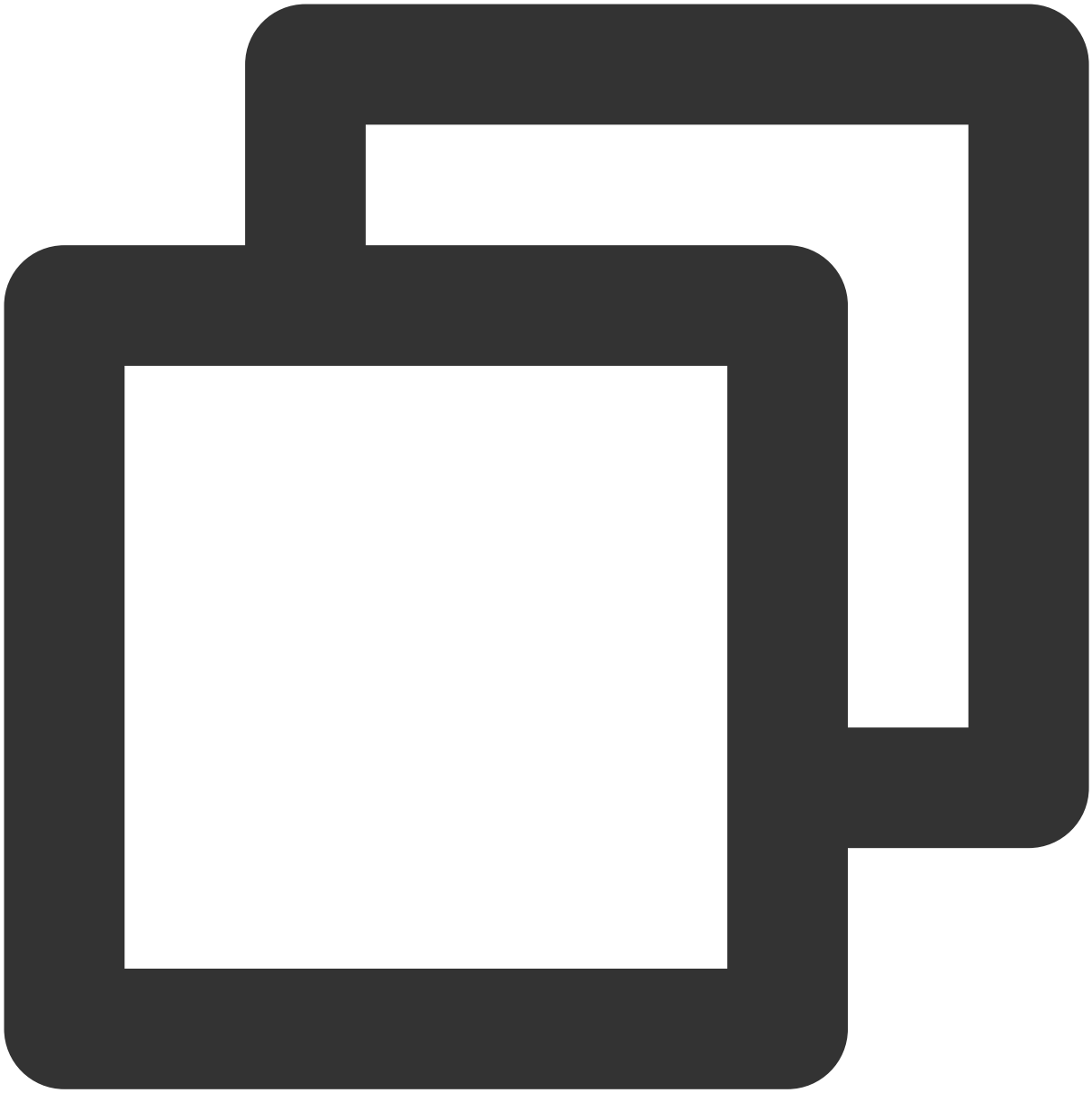
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

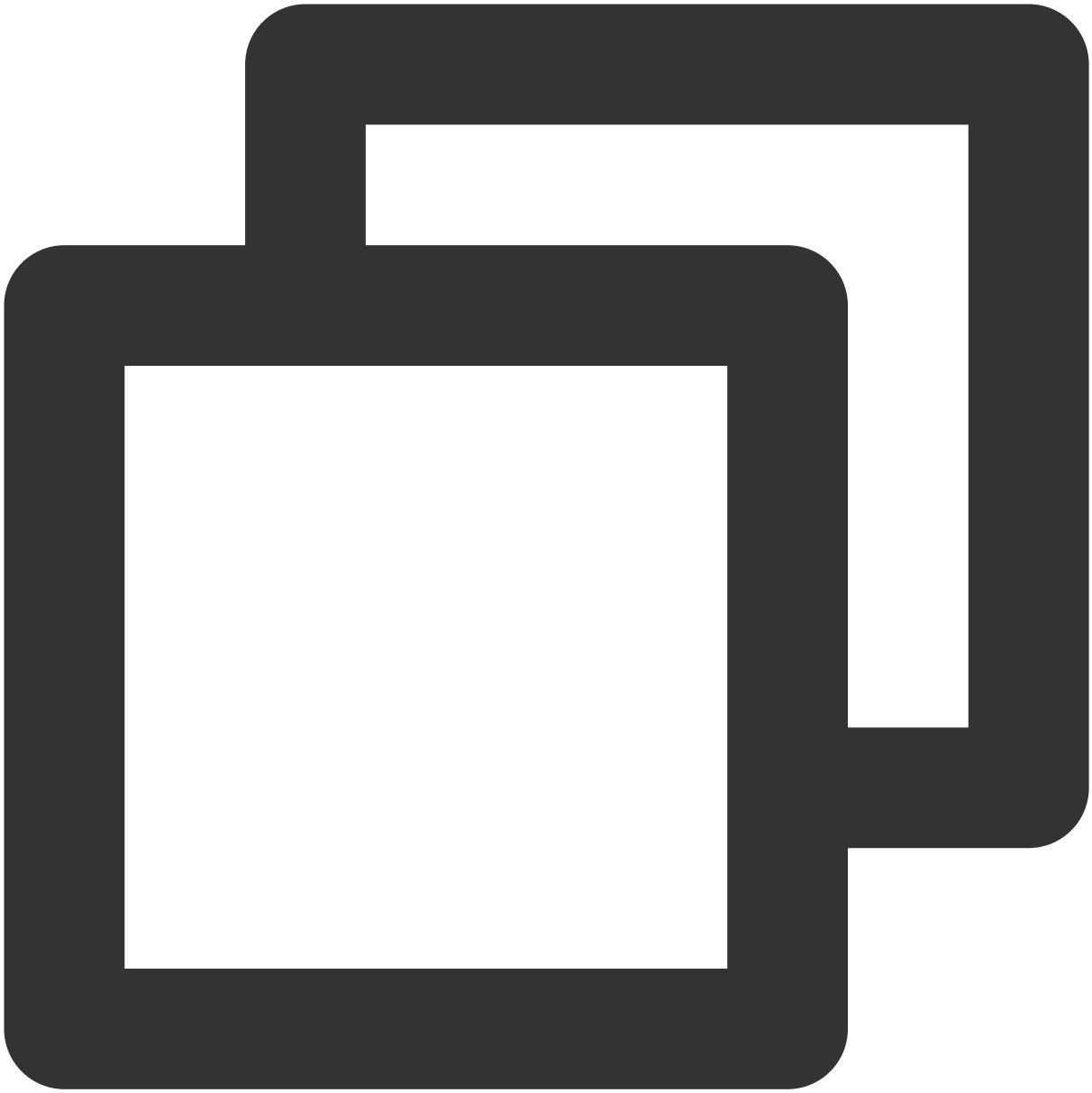


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>97624.424000</Bitrate>
                    <Duration>0.870000</Duration>
                    <FormatLongName>CompuServe Graphics Interchange Format (GIF)</F
                    <FormatName>gif</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>10616656</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio/>
                    <Subtitle/>
                    <Video>
                        <AvgFps>14.666667</AvgFps>
                        <CodecLongName>CompuServe GIF (Graphics Interchange Format)
                        <CodecName>gif</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/100</CodecTimeBase>
                        <ColorPrimaries>unknown</ColorPrimaries>
                        <ColorRange>unknown</ColorRange>
                        <ColorTransfer>unknown</ColorTransfer>
                        <Duration>0.870000</Duration>
                        <Fps>15.083333</Fps>
                        <HasBFrame>0</HasBFrame>
                        <Height>2258</Height>
                        <Index>0</Index>
                        <Level>-99</Level>
                        <NumFrames>7</NumFrames>
                        <PixFormat>bgra</PixFormat>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/100</Timebase>
                        <Width>1280</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
```

```
                        <Md5>3df1f845d2ffd20a525a93ec40014d90</Md5>
                        <ObjectName>output/animation.gif</ObjectName>
                    </Md5Info>
                    <ObjectName>output/animation.gif</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/animation.gif</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>template_name1512456</TemplateName>
            <UserData>This is my Animation job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:43+0800</StartTime>
        <State>Success</State>
        <Tag>Animation</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



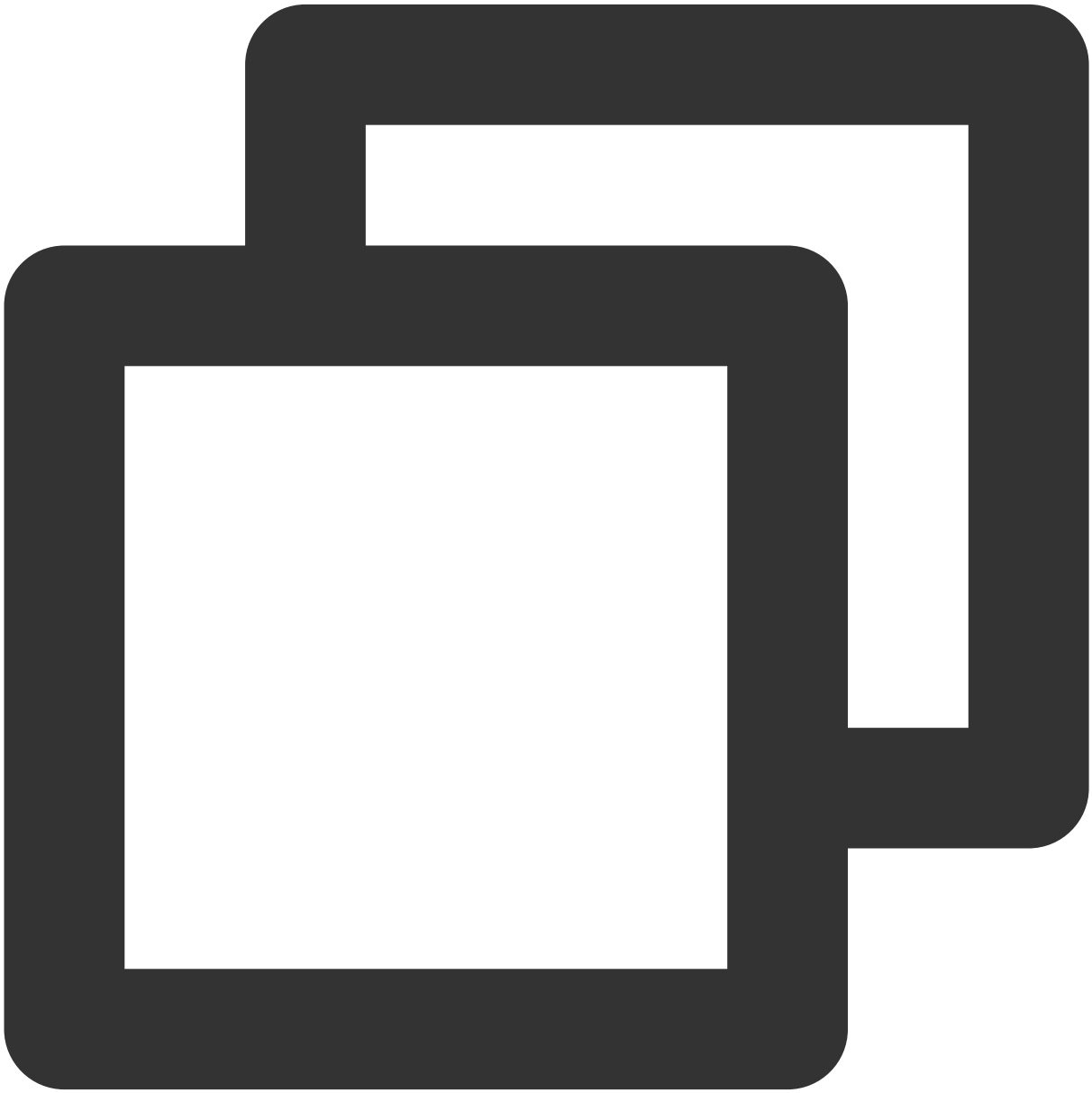
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:30:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j68427030f84611ecb8546d80f2baf56f</JobId>
        <Message>success</Message>
        <Operation>
            <MediaInfo>
                <Format>
                    <Duration>13</Duration>
                    <NumStream>1</NumStream>
                    <Size>1909389</Size>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.653</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>50.000</AvgFps>
                        <Bitrate>1031968</Bitrate>
```

```
                    <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                    <CodecName>hevc</CodecName>
                    <CodecTag>0x31637668</CodecTag>
                    <CodecTagString>hvc1</CodecTagString>
                    <CodecTimeBase>1/50</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>12.960</Duration>
                    <FieldOrder>Unknown</FieldOrder>
                    <Fps>50/1</Fps>
                    <HasBFrame>4</HasBFrame>
                    <Height>960</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>4</Level>
                    <NumFrames>648</NumFrames>
                    <PixFormat>hvc1</PixFormat>
                    <Profile>Main</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>540</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>2db78d168edf2f883adce89a1b4ca033</Md5>
                    <ObjectName>output/transcode.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/transcode.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/transcode.mp4</Object>
            <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
        <TemplateName>trans_993874</TemplateName>
        <UserData>This is my Transcode job.</UserData>
```

```
        </Operation>
        <Progress>100</Progress>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:42+0800</StartTime>
        <State>Success</State>
        <Tag>Transcode</Tag>
        <Workflow>
            <Name>Transcode_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T15:29:42+0800",
        "EndTime": "2022-06-30T15:30:08+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j68427030f84611ecb8546d80f2baf56f",
        "Message": "success",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Duration": "13",
                    "NumStream": "1",
                    "Size": "1909389"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "128000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/44100",
                        "Duration": "13.653",
                        "Index": "1",
                        "Language": "und",
                        "SampleFmt": "fltp",
                        "SampleRate": "44100",
                        "StartTime": "0.000000",
                        "Timebase": "1/44100"
                    },
                    "Video": {
                        "AvgFps": "50.000",
                        "Bitrate": "1031968",
```

```
                    "CodecLongName": "H.265 / HEVC (High Efficiency Video Codin
                    "CodecName": "hevc",
                    "CodecTag": "0x31637668",
                    "CodecTagString": "hvc1",
                    "CodecTimeBase": "1/50",
                    "ColorPrimaries": "bt470bg",
                    "ColorRange": "tv",
                    "ColorTransfer": "smpte170m",
                    "Duration": "12.960",
                    "FieldOrder": "Unknown",
                    "Fps": "50/1",
                    "HasBFrame": "4",
                    "Height": "960",
                    "Index": "0",
                    "Language": "und",
                    "Level": "4",
                    "NumFrames": "648",
                    "PixFormat": "hvc1",
                    "Profile": "Main",
                    "RefFrames": "1",
                    "Rotation": "0",
                    "StartTime": "0.000000",
                    "Timebase": "1/12800",
                    "Width": "540"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": {
                    "Md5": "2db78d168edf2f883adce89a1b4ca033",
                    "ObjectName": "output/transcode.mp4"
                },
                "ObjectName": "output/transcode.mp4",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/transcode.mp4",
            "Region": "ap-chongqing"
        },
        "TemplateId": "t1995d523e42df4c5e858f244b4174360c",
        "TemplateName": "trans_993874",
        "UserData": "This is my Transcode job."
    },
```

```
            "Progress": "100",
            "QueueId": "p2242ab62c7c94486915508540933a2c6",
            "StartTime": "2022-06-30T15:29:42+0800",
            "State": "Success",
            "Tag": "Transcode",
            "Workflow": {
                "Name": "Transcode_1581665960537",
                "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
                "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
                "WorkflowName": "workflow-test"
            }
        }
    }
```

语音合成

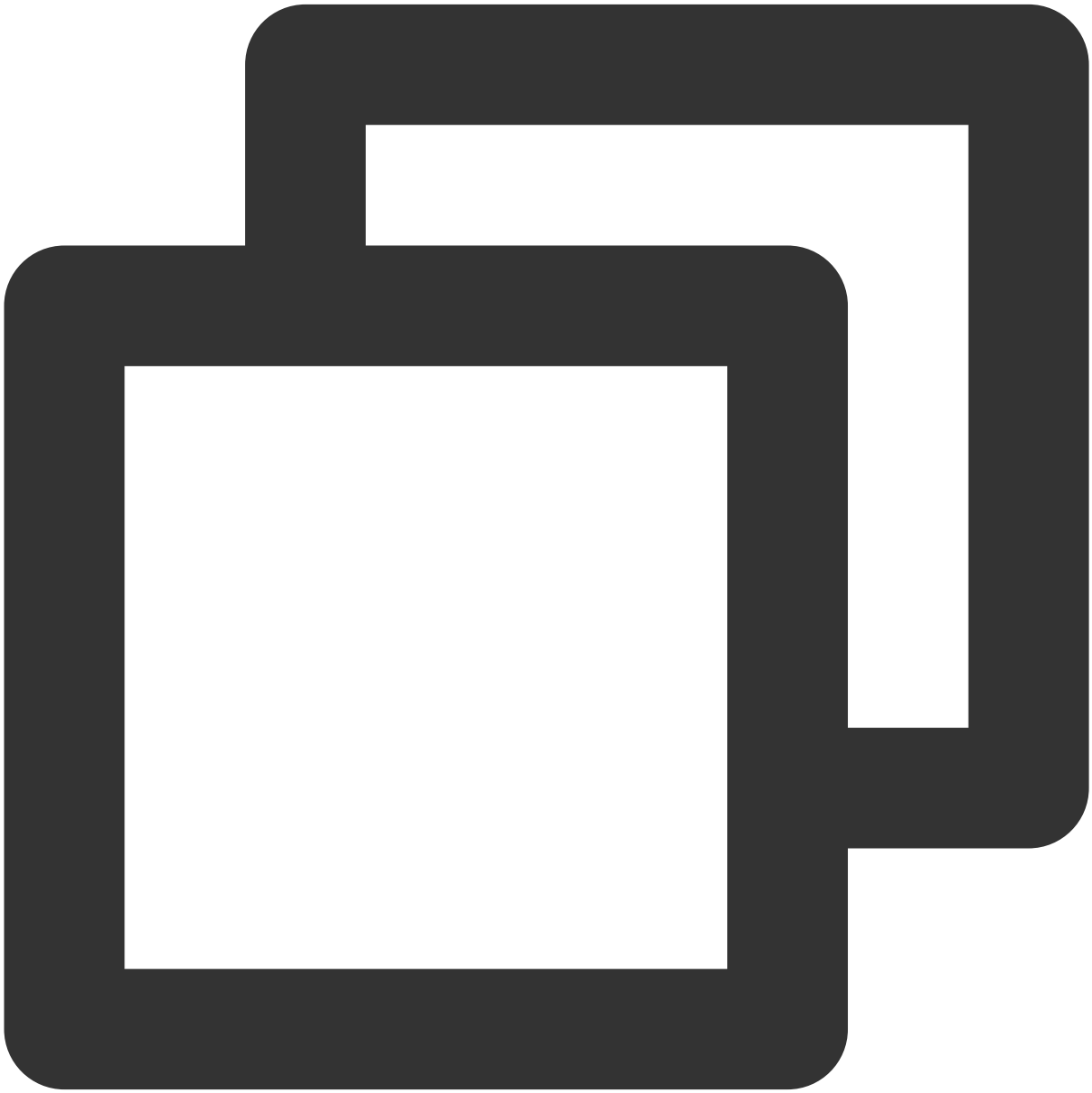
最近更新时间：2023-12-30 16:30:28

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:33:18+0800</CreationTime>
        <EndTime>2022-06-30T19:33:24+0800</EndTime>
        <JobId>j707f4864f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
```

```
                <Stream>
                    <Audio>
                        <CodecName>mp3</CodecName>
                        <SampleRate>16000</SampleRate>
                    </Audio>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>5b2ca33cd7357abb1e0ea29f57544df9</Md5>
                        <ObjectName>/output/aixiaonan.mp3</ObjectName>
                    </Md5Info>
                    <ObjectName>/output/aixiaonan.mp3</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/aixiaonan.mp3</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TtsConfig>
                <Input>你好，我是爱晓楠，很高兴认识你。</Input>
                <InputType>Text</InputType>
            </TtsConfig>
            <TemplateId>t1f1ae1dfsdc9ds41dsb31632d457ds6s2a</TemplateId>
            <TemplateName>template_tts</TemplateName>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:33:19+0800</StartTime>
        <State>Success</State>
        <Tag>Tts</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交语音合成任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

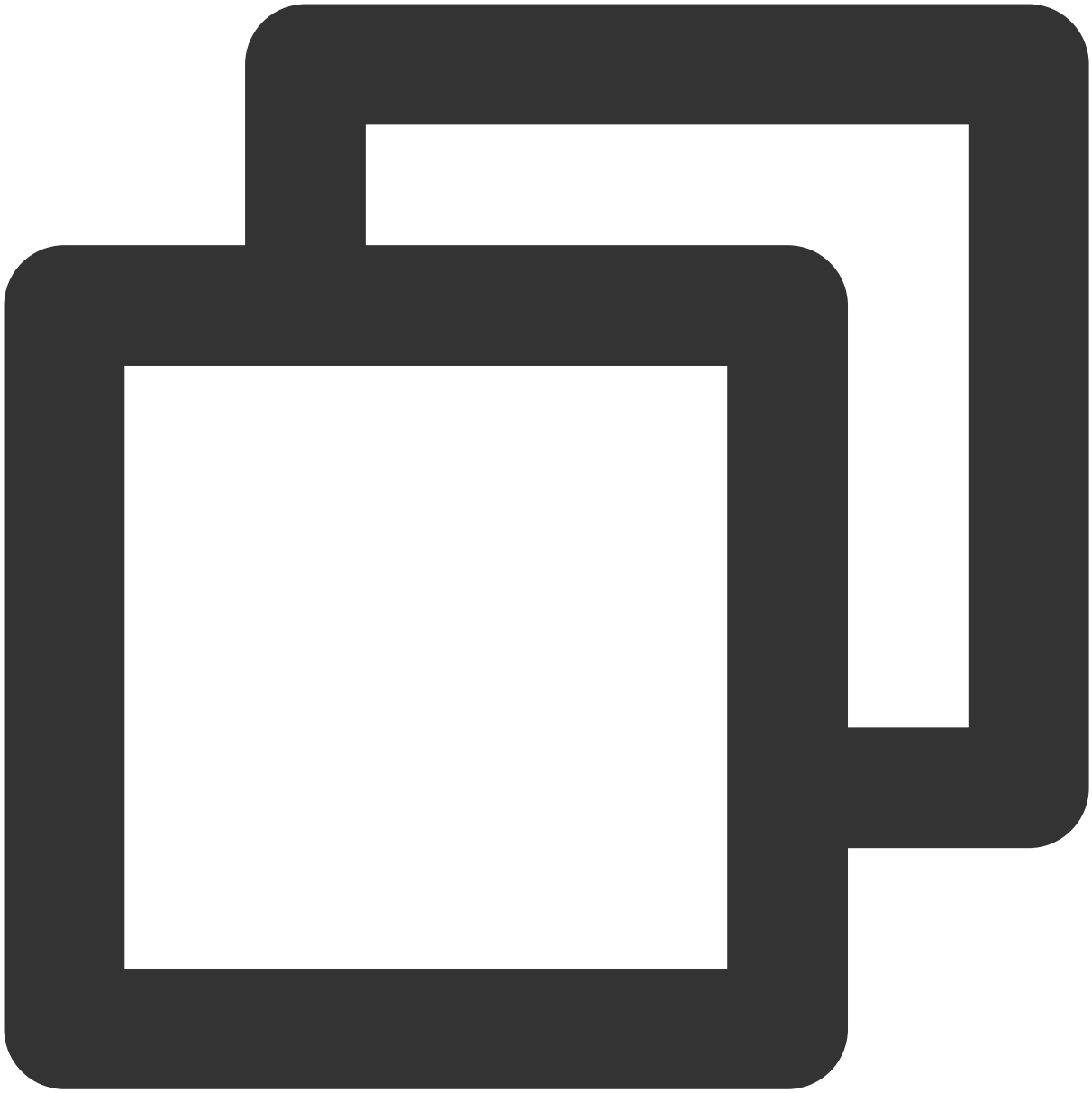
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

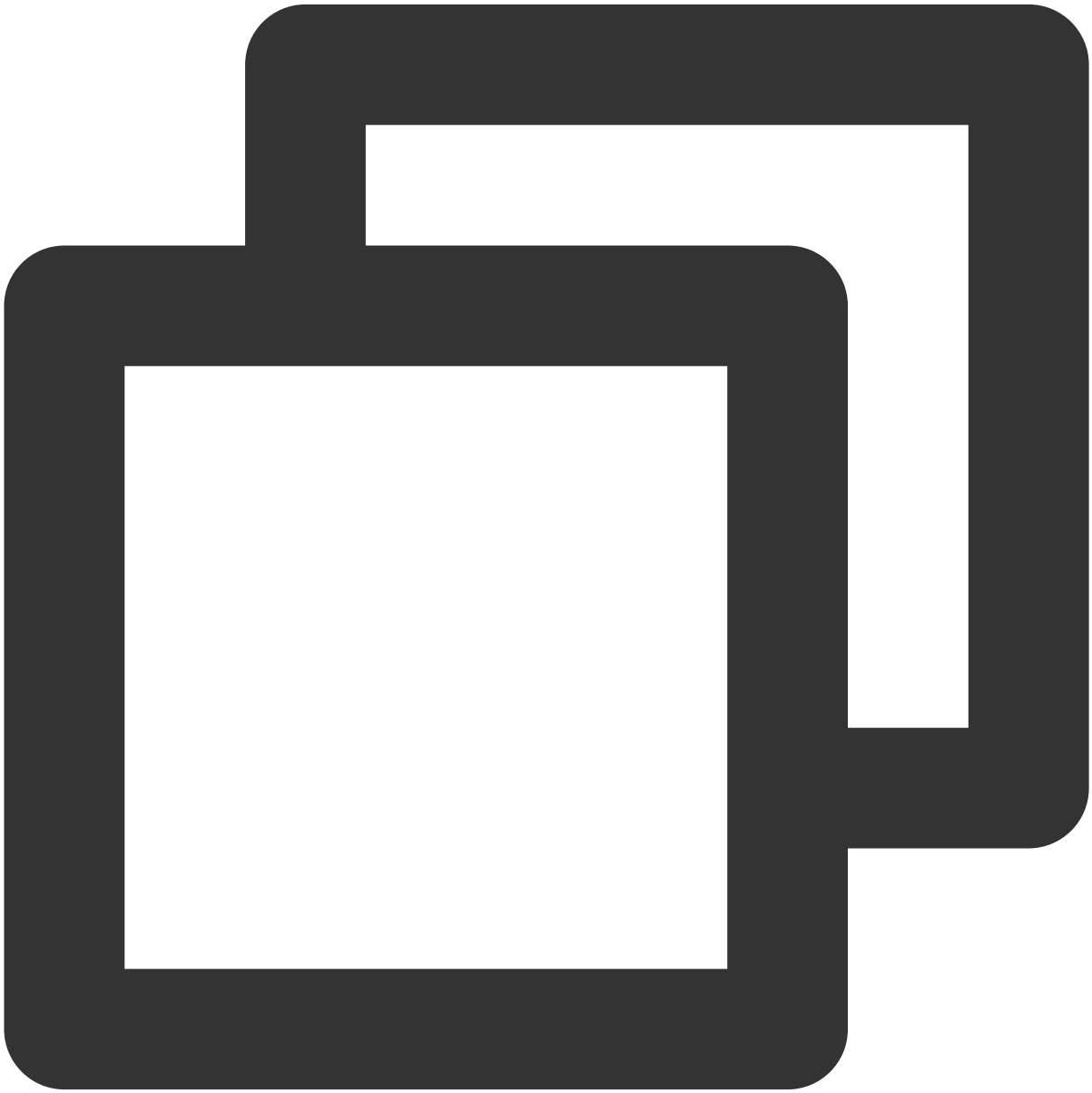
案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:33:18+0800</CreationTime>
        <EndTime>2022-06-30T19:33:24+0800</EndTime>
        <JobId>j707f4864f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format/>
```

```
                <Stream>
                    <Audio>
                        <CodecName>mp3</CodecName>
                        <SampleRate>16000</SampleRate>
                    </Audio>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>5b2ca33cd7357abb1e0ea29f57544df9</Md5>
                        <ObjectName>/output/aixiaonan.mp3</ObjectName>
                    </Md5Info>
                    <ObjectName>/output/aixiaonan.mp3</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/aixiaonan.mp3</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TtsConfig>
                <Input>你好，我是爱晓楠，很高兴认识你。</Input>
                <InputType>Text</InputType>
            </TtsConfig>
            <TemplateId>t1f1ae1dfsdc9ds41dsb31632d457ds6s2a</TemplateId>
            <TemplateName>template_tts</TemplateName>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:33:19+0800</StartTime>
        <State>Success</State>
        <Tag>Tts</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调

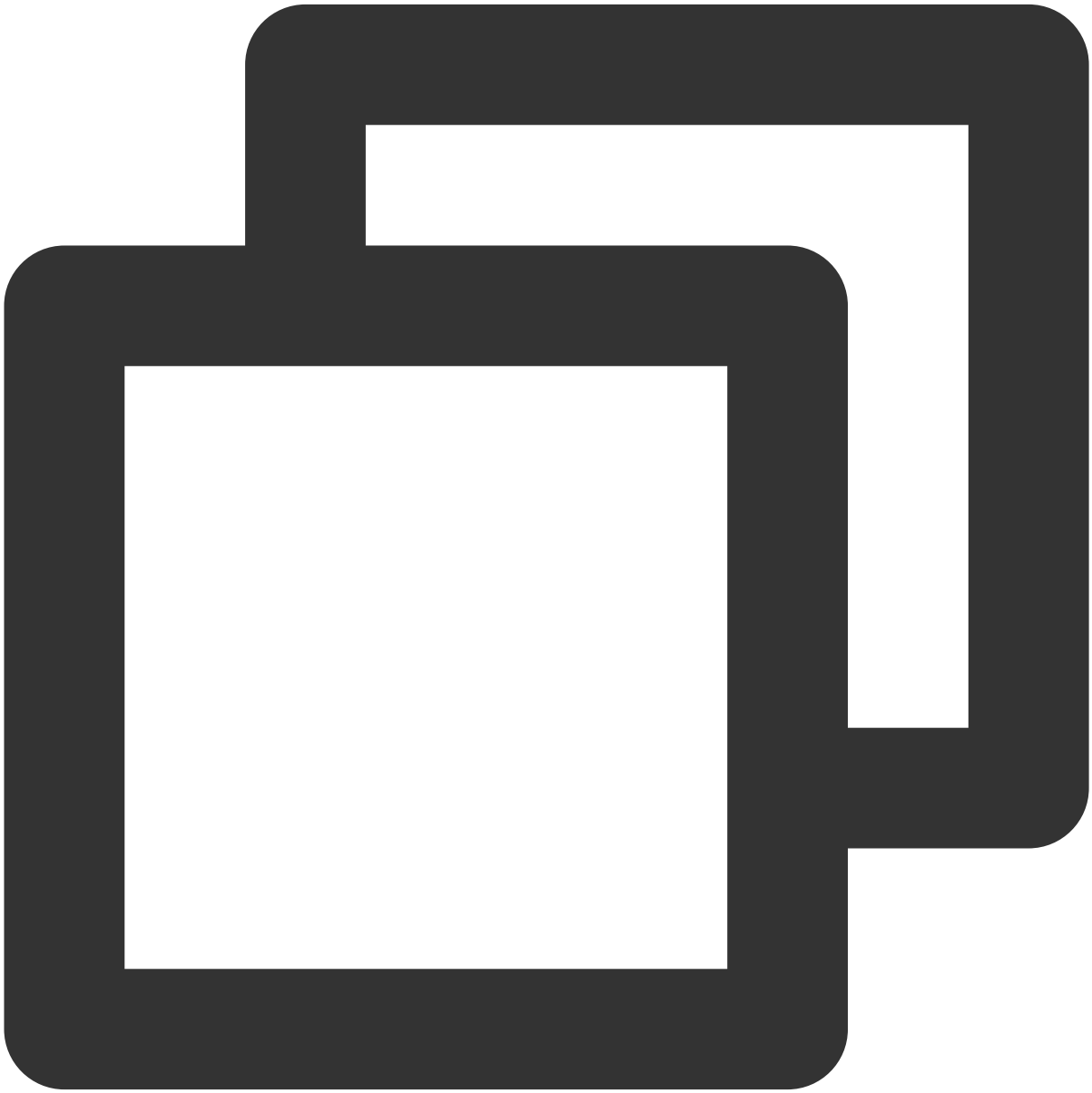


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:33:18+0800</CreationTime>
        <EndTime>2022-06-30T19:33:24+0800</EndTime>
        <JobId>j707f4864f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <Input>
                <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>application/octet-stream</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>265</Value>
            </CosHeaders>
        </Input>
        <MediaInfo>
            <Format/>
            <Stream>
                <Audio>
                    <CodecName>mp3</CodecName>
                    <SampleRate>16000</SampleRate>
                </Audio>
                <Subtitle/>
                <Video/>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>5b2ca33cd7357abb1e0ea29f57544df9</Md5>
                    <ObjectName>/output/aixiaonan.mp3</ObjectName>
                </Md5Info>
                <ObjectName>/output/aixiaonan.mp3</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/aixiaonan.mp3</Object>
            <Region>ap-chongqing</Region>
        </Output>
        <TtsConfig>
            <Input>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/t
            <InputType>Url</InputType>
```

```
            </TtsConfig>
            <TemplateId>t1f1ae1dfsdc9ds41dsb31632d457ds6s2a</TemplateId>
            <TemplateName>template_tts</TemplateName>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:33:19+0800</StartTime>
        <State>Success</State>
        <Tag>Tts</Tag>
        <Workflow>
            <Name>Tts_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:33:18+0800",
        "EndTime": "2022-06-30T19:33:24+0800",
        "JobId": "j707f4864f86811ecb90d0b03267ce0e5",
        "Operation": {
            "Input": {
                "CosHeaders": [{
                        "Key": "Content-Type",
```

```
                        "Value": "application/octet-stream"
                    },
                    {
                        "Key": "x-cos-request-id",
                        "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                    },
                    {
                        "Key": "EventName",
                        "Value": "cos:ObjectCreated:Put"
                    },
                    {
                        "Key": "Size",
                        "Value": "265"
                    }
                ]
            },
            "MediaInfo": {
                "Stream": {
                    "Audio": {
                        "CodecName": "mp3",
                        "SampleRate": "16000"
                    }
                }
            },
            "MediaResult": {
                "OutputFile": {
                    "Bucket": "test-123456789",
                    "Md5Info": {
                        "Md5": "5b2ca33cd7357abb1e0ea29f57544df9",
                        "ObjectName": "/output/aixiaonan.mp3"
                    },
                    "ObjectName": "/output/aixiaonan.mp3",
                    "Region": "ap-chongqing"
                }
            },
            "Output": {
                "Bucket": "test-123456789",
                "Object": "output/aixiaonan.mp3",
                "Region": "ap-chongqing"
            },
            "TtsConfig": {
                "Input": "https://test-123456789.cos.ap-chongqing.myqcloud.com/inpu
                "InputType": "Url"
            },
            "TemplateId": "t1f1ae1dfsdc9ds41dsb31632d457ds6s2a",
            "TemplateName": "template_tts"
        },
```

```
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:33:19+0800",
        "State": "Success",
        "Tag": "Tts",
        "Workflow": {
            "Name": "Tts_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

精彩集锦

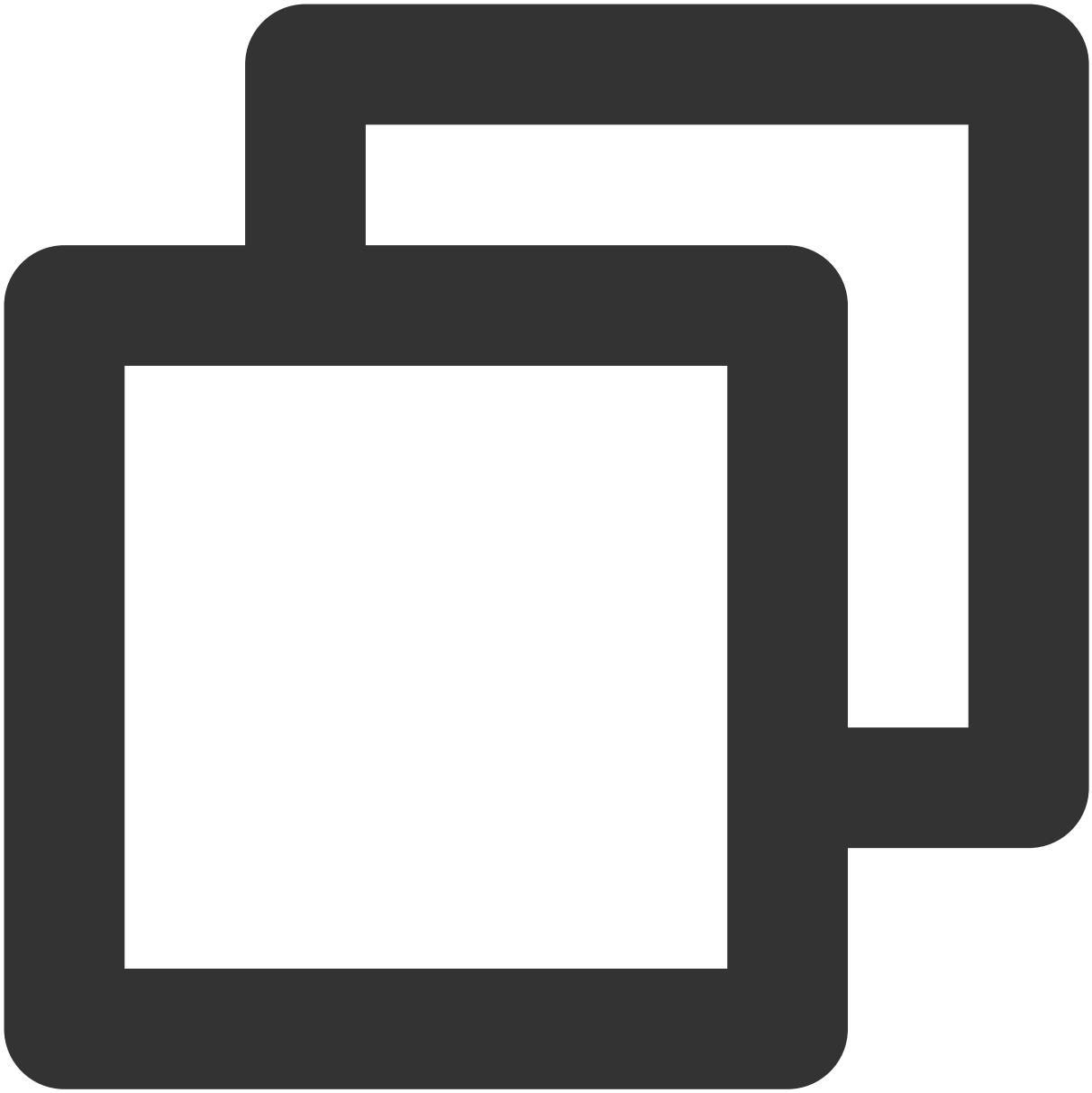
最近更新时间：2023-12-30 16:30:02

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:55:05+0800</CreationTime>
        <EndTime>2022-06-30T19:55:19+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j7b0dfbe2f86b11ec995acf5cffaf6664</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>14044.868000</Bitrate>
                    <Duration>2.067000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>3628843</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>126.400000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>2.066576</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000000</AvgFps>
                        <Bitrate>14664.836000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/90000</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>1.960000</Duration>
                        <Fps>25.000000</Fps>
                        <HasBFrame>2</HasBFrame>
```

```
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>41</Level>
                        <NumFrames>49</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.086000</StartTime>
                        <Timebase>1/90000</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>6080b31ea94defa6c19cc4def539a291</Md5>
                        <ObjectName>output/videoMontage.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/videoMontage.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/videoMontage.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t164ac18da8bdc403fa27ac4b077a962a1</TemplateId>
            <TemplateName>videoMontage</TemplateName>
            <UserData>This is my VideoMontage job.</UserData>
        </Operation>
        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
        <StartTime>2022-06-30T19:55:05+0800</StartTime>
        <State>Success</State>
        <Tag>VideoMontage</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交精彩集锦任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

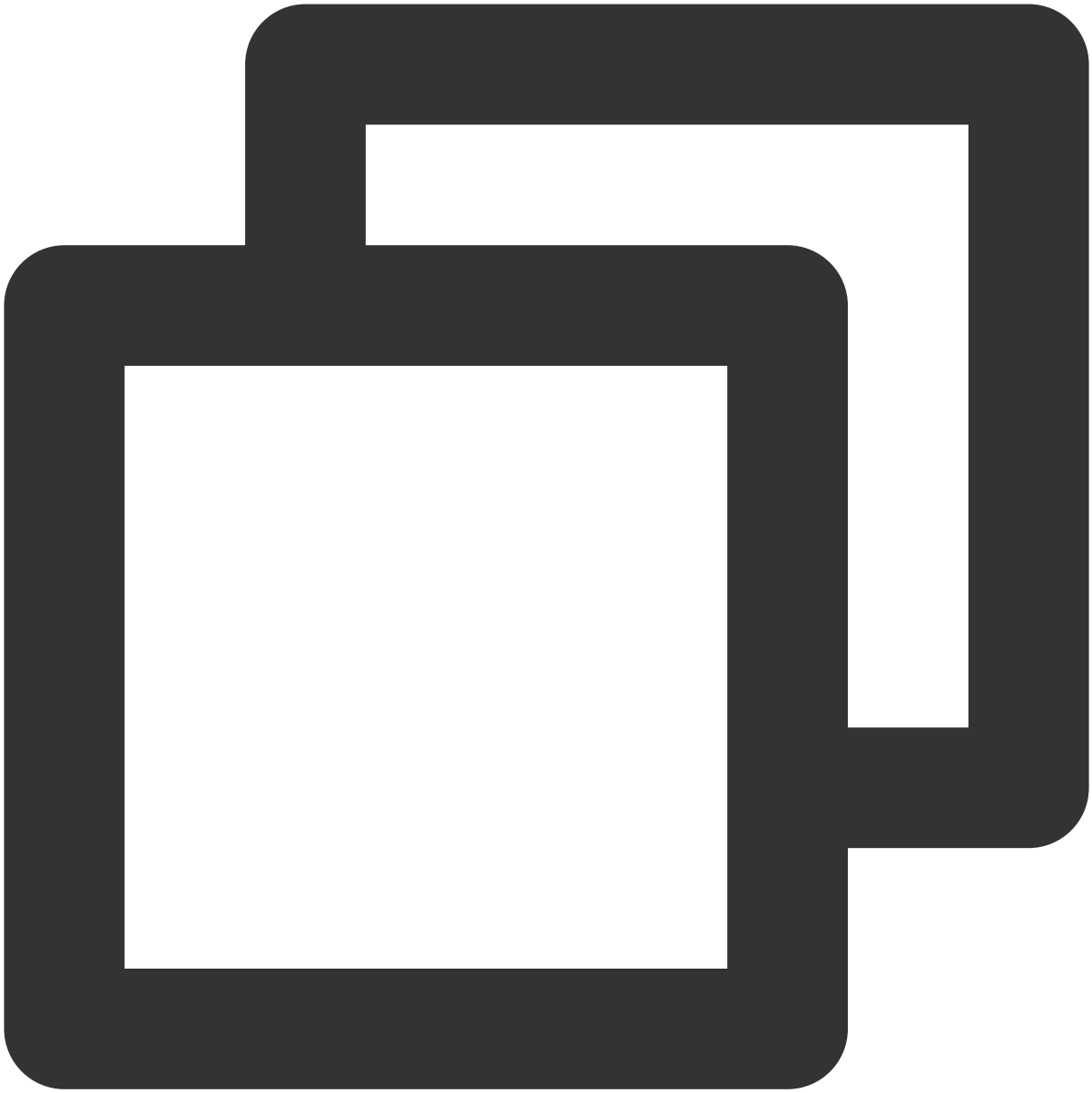
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

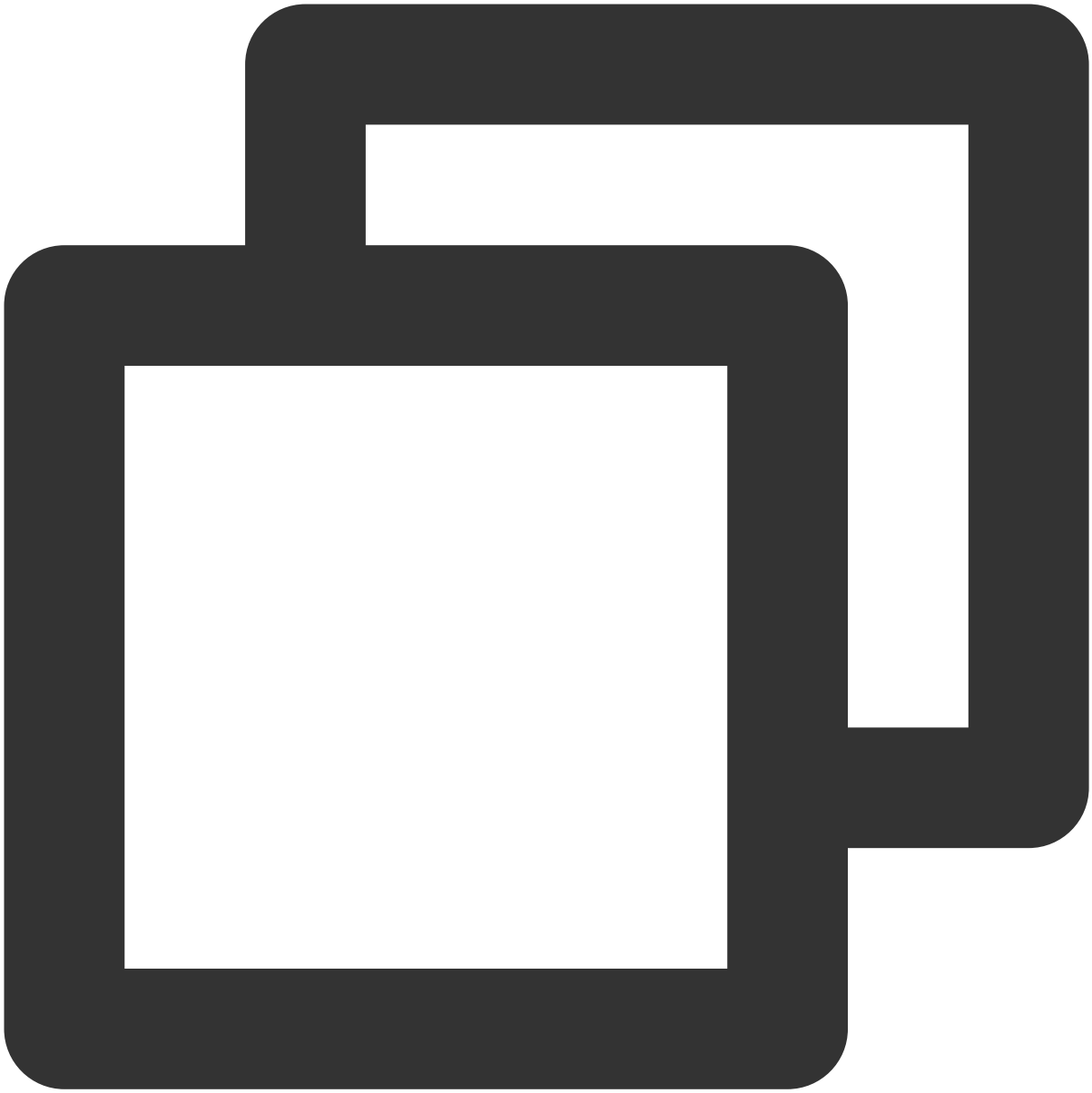


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:55:05+0800</CreationTime>
        <EndTime>2022-06-30T19:55:19+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j7b0dfbe2f86b11ec995acf5cffaf6664</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>14044.868000</Bitrate>
                    <Duration>2.067000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>3628843</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>126.400000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>2.066576</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>25.000000</AvgFps>
                        <Bitrate>14664.836000</Bitrate>
                        <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                        <CodecName>h264</CodecName>
                        <CodecTag>0x31637661</CodecTag>
                        <CodecTagString>avc1</CodecTagString>
                        <CodecTimeBase>1/90000</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>1.960000</Duration>
                        <Fps>25.000000</Fps>
                        <HasBFrame>2</HasBFrame>
```

```
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>41</Level>
                        <NumFrames>49</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>High</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.086000</StartTime>
                        <Timebase>1/90000</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>6080b31ea94defa6c19cc4def539a291</Md5>
                        <ObjectName>output/videoMontage.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/videoMontage.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/videoMontage.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t164ac18da8bdc403fa27ac4b077a962a1</TemplateId>
            <TemplateName>videoMontage</TemplateName>
            <UserData>This is my VideoMontage job.</UserData>
        </Operation>
        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
        <StartTime>2022-06-30T19:55:05+0800</StartTime>
        <State>Success</State>
        <Tag>VideoMontage</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



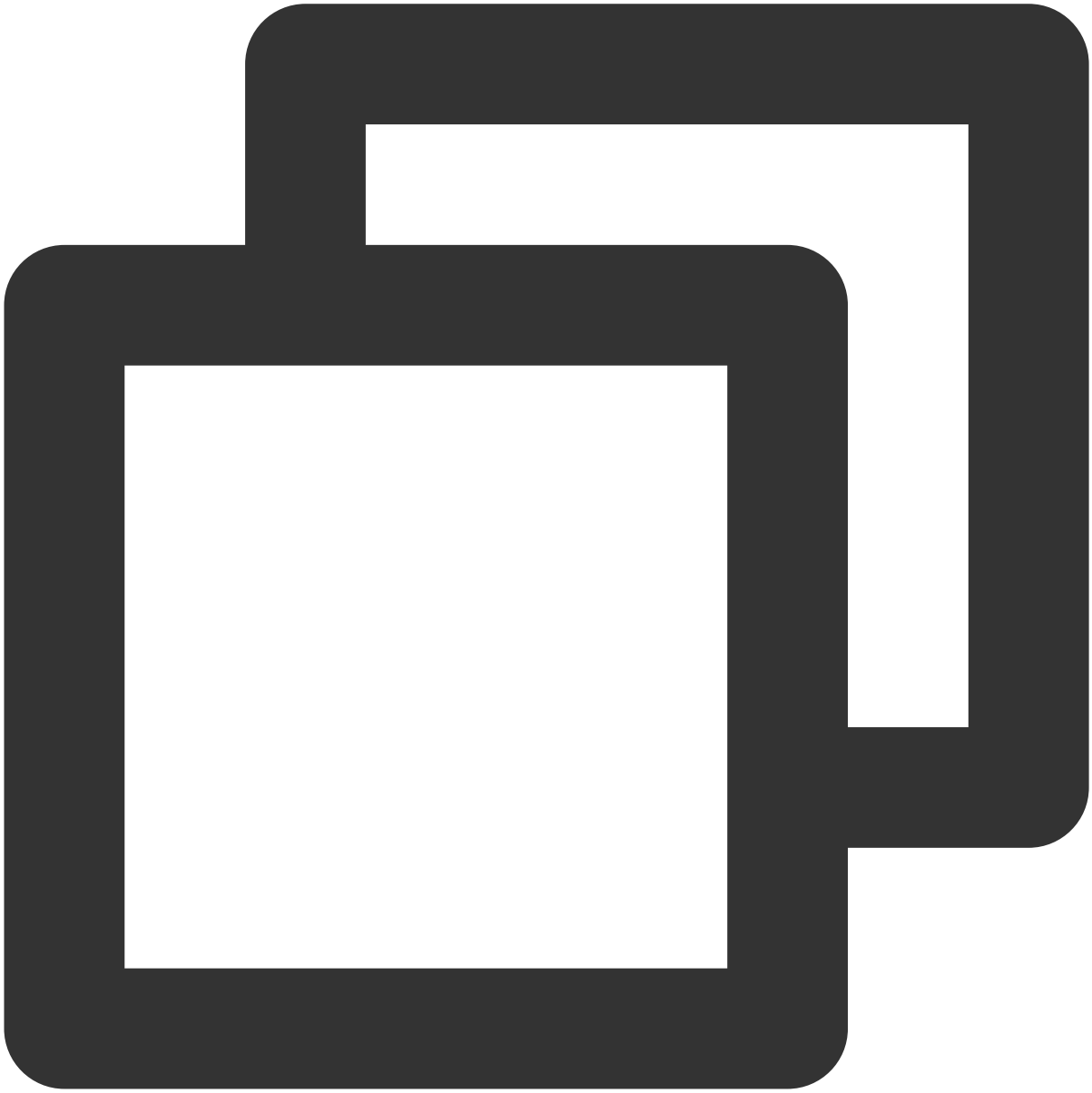
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:55:05+0800</CreationTime>
        <EndTime>2022-06-30T19:55:19+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j7b0dfbe2f86b11ec995acf5cffaf6664</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>14044.868000</Bitrate>
                    <Duration>2.067000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>3628843</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>126.400000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>2.066576</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
```

```
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>25.000000</AvgFps>
                    <Bitrate>14664.836000</Bitrate>
                    <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</C
                    <CodecName>h264</CodecName>
                    <CodecTag>0x31637661</CodecTag>
                    <CodecTagString>avc1</CodecTagString>
                    <CodecTimeBase>1/90000</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>1.960000</Duration>
                    <Fps>25.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>960</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>41</Level>
                    <NumFrames>49</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>High</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.086000</StartTime>
                    <Timebase>1/90000</Timebase>
                    <Width>544</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>6080b31ea94defa6c19cc4def539a291</Md5>
                    <ObjectName>output/videoMontage.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/videoMontage.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/videoMontage.mp4</Object>
            <Region>ap-chongqing</Region>
```

```
            </Output>
            <TemplateId>t164ac18da8bdc403fa27ac4b077a962a1</TemplateId>
            <TemplateName>videoMontage</TemplateName>
            <UserData>This is my VideoMontage job.</UserData>
        </Operation>
        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
        <StartTime>2022-06-30T19:55:05+0800</StartTime>
        <State>Success</State>
        <Tag>VideoMontage</Tag>
        <Workflow>
            <Name>VideoMontage_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T19:55:05+0800",
        "EndTime": "2022-06-30T19:55:19+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                "Key": "Content-Type",
                "Value": "video/mp4"
            },
            {
                "Key": "x-cos-request-id",
                "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
            },
            {
                "Key": "EventName",
                "Value": "cos:ObjectCreated:Put"
            },
            {
                "Key": "Size",
                "Value": "1424687"
            }
        ]
    },
    "JobId": "j7b0dfbe2f86b11ec995acf5cffaf6664",
    "Operation": {
        "MediaInfo": {
            "Format": {
                "Bitrate": "14044.868000",
                "Duration": "2.067000",
                "FormatLongName": "QuickTime / MOV",
                "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                "NumProgram": "0",
                "NumStream": "2",
                "Size": "3628843",
                "StartTime": "0.000000"
            },
            "Stream": {
                "Audio": {
                    "Bitrate": "126.400000",
                    "Channel": "2",
                    "ChannelLayout": "stereo",
                    "CodecLongName": "AAC (Advanced Audio Coding)",
                    "CodecName": "aac",
                    "CodecTag": "0x6134706d",
                    "CodecTagString": "mp4a",
                    "CodecTimeBase": "1/44100",
                    "Duration": "2.066576",
                    "Index": "1",
                    "Language": "und",
                    "SampleFmt": "fltp",
                    "SampleRate": "44100",
                    "StartTime": "0.000000",
                    "Timebase": "1/44100"
```

```
                },
                "Video": {
                    "AvgFps": "25.000000",
                    "Bitrate": "14664.836000",
                    "CodecLongName": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10
                    "CodecName": "h264",
                    "CodecTag": "0x31637661",
                    "CodecTagString": "avc1",
                    "CodecTimeBase": "1/90000",
                    "ColorPrimaries": "bt470bg",
                    "ColorRange": "tv",
                    "ColorTransfer": "smpte170m",
                    "Duration": "1.960000",
                    "Fps": "25.000000",
                    "HasBFrame": "2",
                    "Height": "960",
                    "Index": "0",
                    "Language": "und",
                    "Level": "41",
                    "NumFrames": "49",
                    "PixFormat": "yuv420p",
                    "Profile": "High",
                    "RefFrames": "1",
                    "Rotation": "0.000000",
                    "StartTime": "0.086000",
                    "Timebase": "1/90000",
                    "Width": "544"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": {
                    "Md5": "6080b31ea94defa6c19cc4def539a291",
                    "ObjectName": "output/videoMontage.mp4"
                },
                "ObjectName": "output/videoMontage.mp4",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/videoMontage.mp4",
            "Region": "ap-chongqing"
        },
        "TemplateId": "t164ac18da8bdc403fa27ac4b077a962a1",
```

```
            "TemplateName": "videoMontage",
            "UserData": "This is my VideoMontage job."
        },
        "QueueId": "p09d709939fef48a0a5c247ef39d90cec",
        "StartTime": "2022-06-30T19:55:05+0800",
        "State": "Success",
        "Tag": "VideoMontage",
        "Workflow": {
            "Name": "VideoMontage_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

视频增强

最近更新时间：2023-12-30 16:30:55

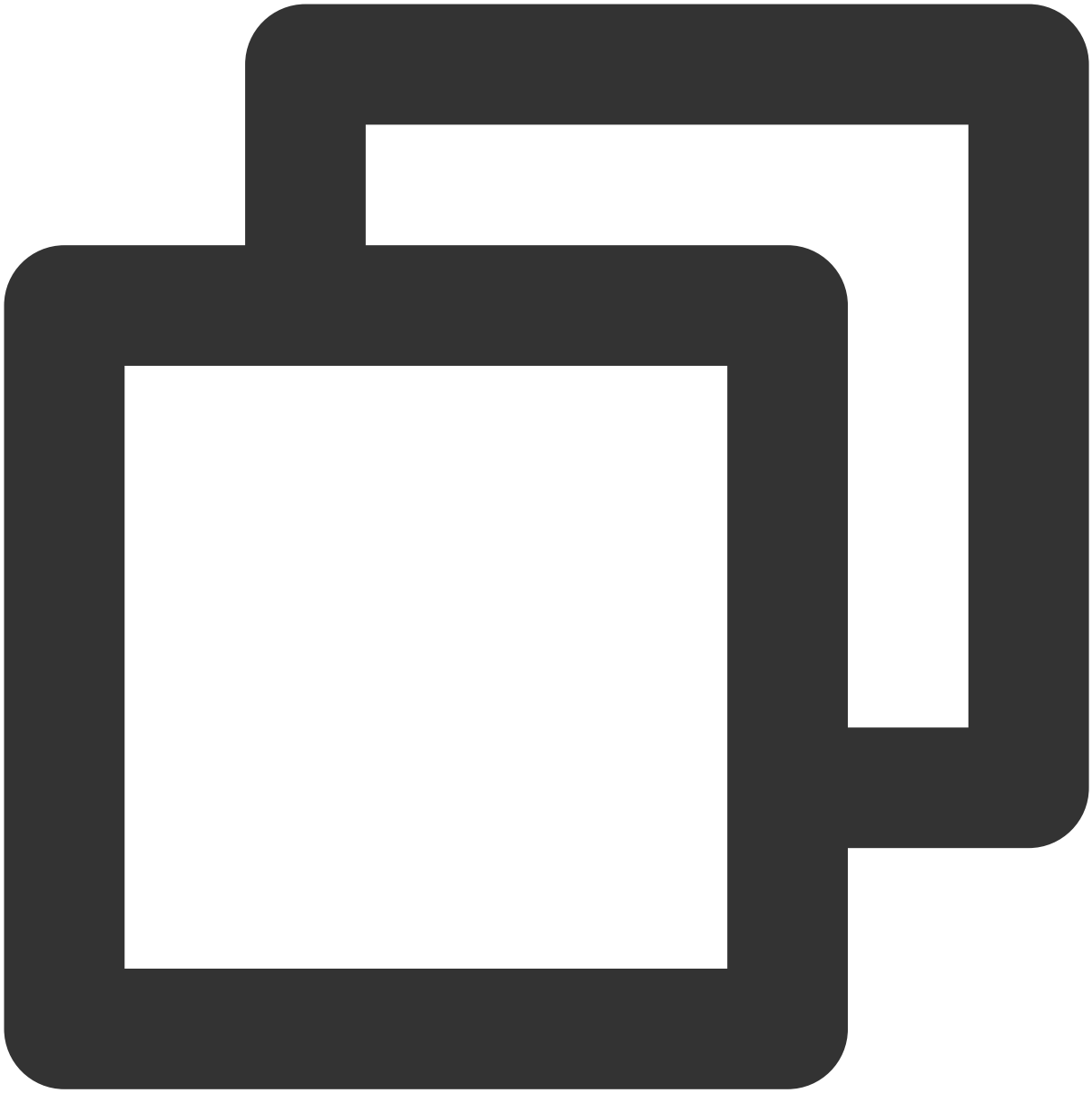
功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T20:02:17+0800</CreationTime>
        <EndTime>2022-06-30T20:02:34+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j7d0eafdaf86c11ecbdf419856486c0ab</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>10269.090000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>17526771</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.922840</AvgFps>
                        <Bitrate>10674.365000</Bitrate>
                        <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                        <CodecName>hevc</CodecName>
                        <CodecTag>0x31766568</CodecTag>
                        <CodecTagString>hev1</CodecTagString>
                        <CodecTimeBase>1/12800</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960000</Duration>
                        <FieldOrder>progressive</FieldOrder>
                        <Fps>25.000000</Fps>
```

```
                        <HasBFrame>2</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>93</Level>
                        <NumFrames>323</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>Main</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>a825aa6c59371509995a47b570c29297</Md5>
                        <ObjectName>output/video-process.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/video-process.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/video-process.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <TemplateId>t1f4ac18da8bdc403fa27ac4bfn77a62a1</TemplateId>
            <TemplateName>videoProcess_12654</TemplateName>
            <UserData>This is my VideoProcess job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T20:02:18+0800</StartTime>
        <State>Success</State>
        <Tag>VideoProcess</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交视频增强任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

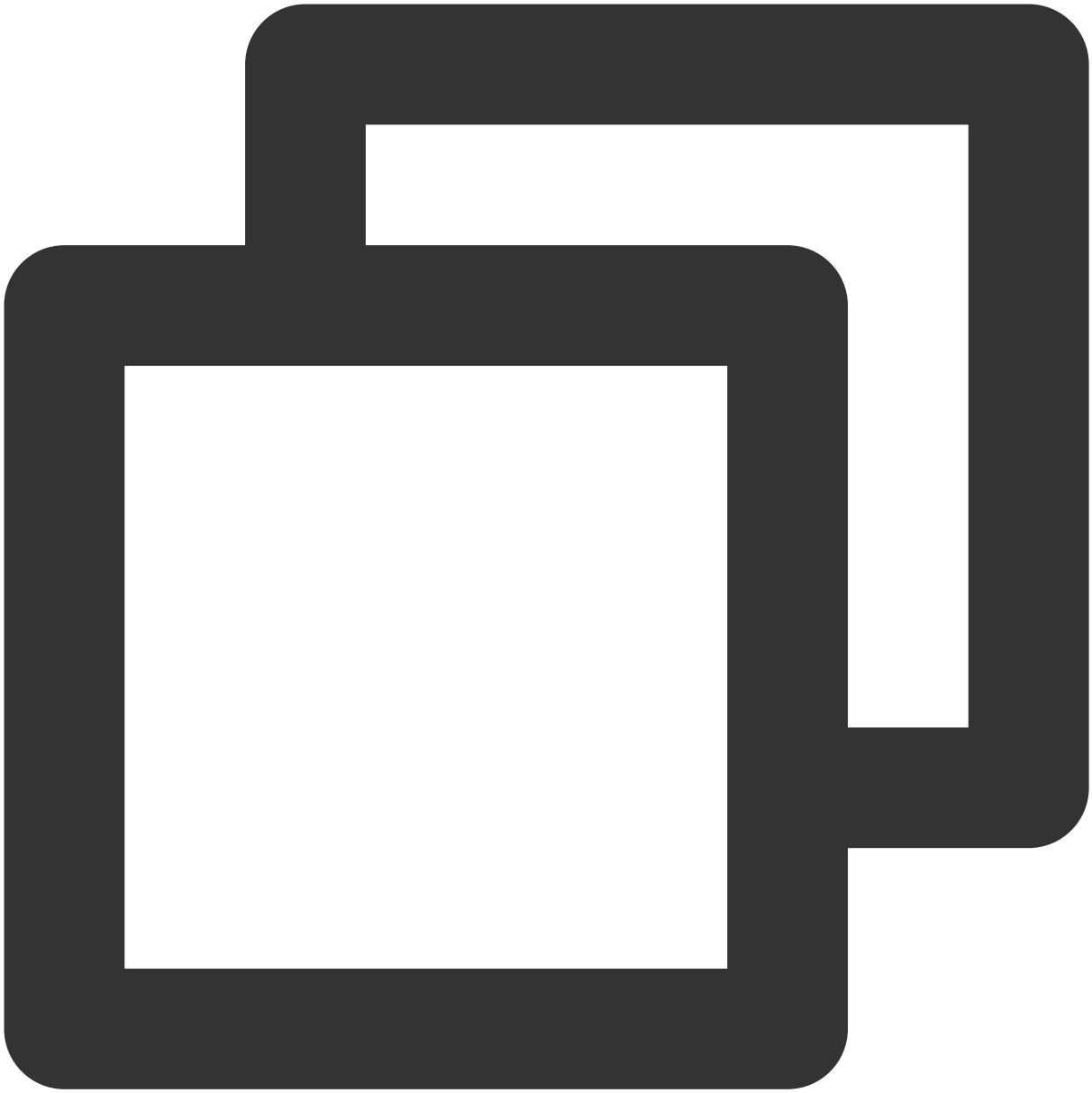
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

版权所有：腾讯云计算（北京）有限责任公司

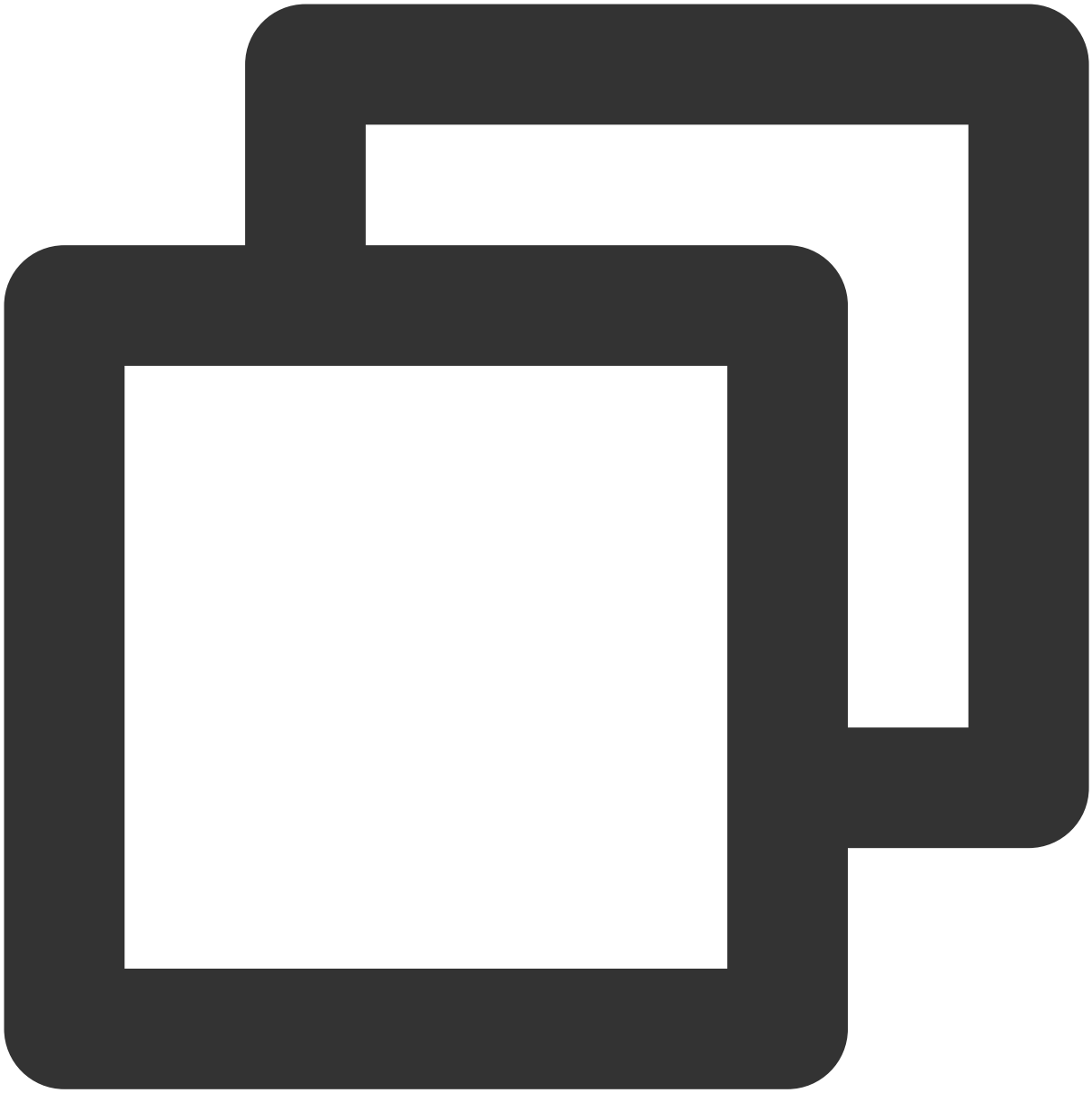


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T20:02:17+0800</CreationTime>
        <EndTime>2022-06-30T20:02:34+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j7d0eafdaf86c11ecbdf419856486c0ab</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>10269.090000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>17526771</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video>
                        <AvgFps>24.922840</AvgFps>
                        <Bitrate>10674.365000</Bitrate>
                        <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                        <CodecName>hevc</CodecName>
                        <CodecTag>0x31766568</CodecTag>
                        <CodecTagString>hev1</CodecTagString>
                        <CodecTimeBase>1/12800</CodecTimeBase>
                        <ColorPrimaries>bt470bg</ColorPrimaries>
                        <ColorRange>tv</ColorRange>
                        <ColorTransfer>smpte170m</ColorTransfer>
                        <Duration>12.960000</Duration>
                        <FieldOrder>progressive</FieldOrder>
                        <Fps>25.000000</Fps>
```

```
                        <HasBFrame>2</HasBFrame>
                        <Height>960</Height>
                        <Index>0</Index>
                        <Language>und</Language>
                        <Level>93</Level>
                        <NumFrames>323</NumFrames>
                        <PixFormat>yuv420p</PixFormat>
                        <Profile>Main</Profile>
                        <RefFrames>1</RefFrames>
                        <Rotation>0.000000</Rotation>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/12800</Timebase>
                        <Width>544</Width>
                    </Video>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>a825aa6c59371509995a47b570c29297</Md5>
                        <ObjectName>output/video-process.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/video-process.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/video-process.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <TemplateId>t1f4ac18da8bdc403fa27ac4bfn77a62a1</TemplateId>
            <TemplateName>videoProcess_12654</TemplateName>
            <UserData>This is my VideoProcess job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T20:02:18+0800</StartTime>
        <State>Success</State>
        <Tag>VideoProcess</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



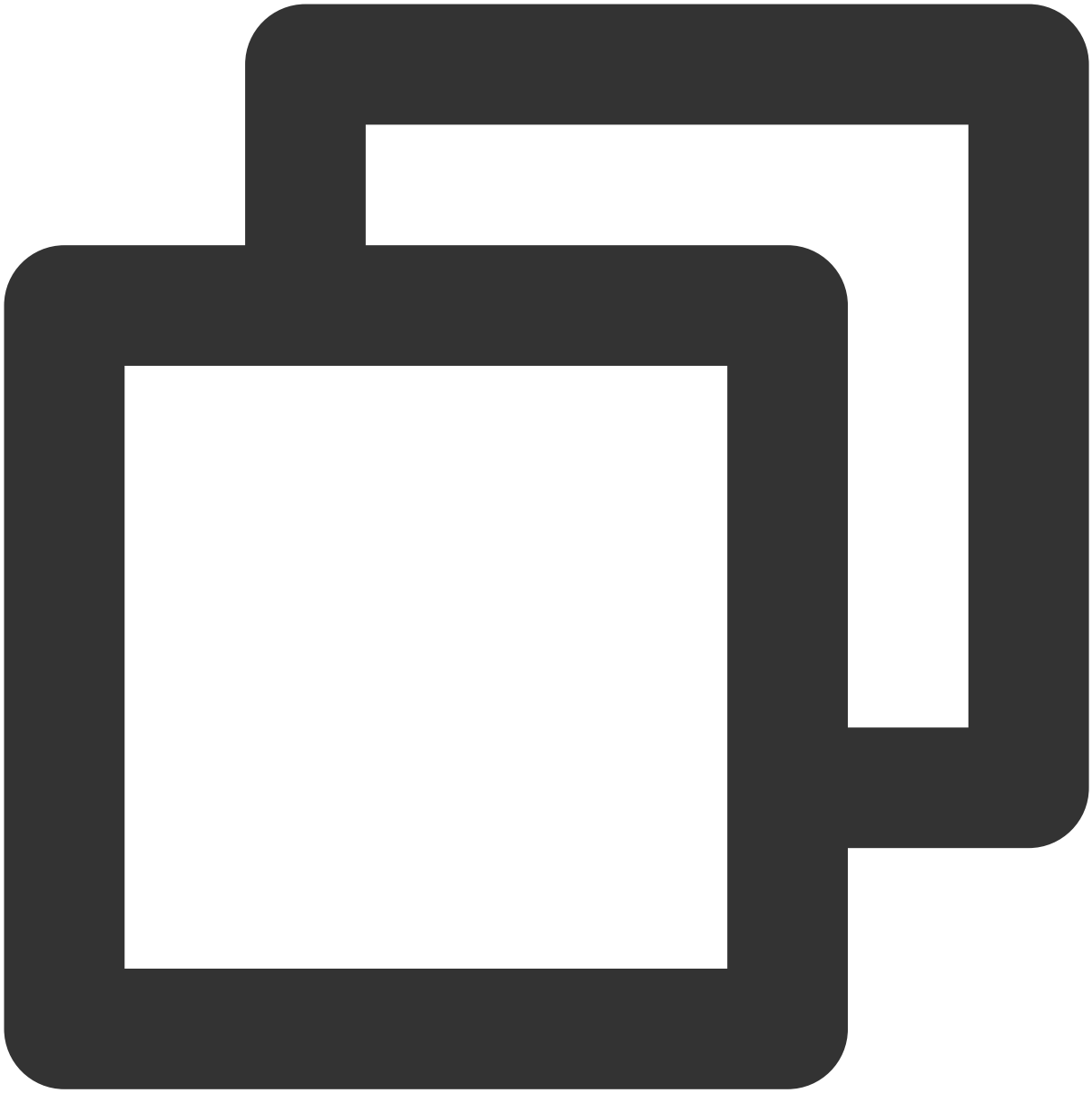
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T20:02:17+0800</CreationTime>
        <EndTime>2022-06-30T20:02:34+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1424687</Value>
            </CosHeaders>
        </Input>
        <JobId>j7d0eafdaf86c11ecbdf419856486c0ab</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>10269.090000</Bitrate>
                    <Duration>13.654000</Duration>
                    <FormatLongName>QuickTime / MOV</FormatLongName>
                    <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>2</NumStream>
                    <Size>17526771</Size>
                    <StartTime>0.000000</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.726000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
                        <CodecName>aac</CodecName>
                        <CodecTag>0x6134706d</CodecTag>
                        <CodecTagString>mp4a</CodecTagString>
                        <CodecTimeBase>1/44100</CodecTimeBase>
                        <Duration>13.652993</Duration>
                        <Index>1</Index>
                        <Language>und</Language>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.000000</StartTime>
                        <Timebase>1/44100</Timebase>
```

```
                </Audio>
                <Subtitle/>
                <Video>
                    <AvgFps>24.922840</AvgFps>
                    <Bitrate>10674.365000</Bitrate>
                    <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)<
                    <CodecName>hevc</CodecName>
                    <CodecTag>0x31766568</CodecTag>
                    <CodecTagString>hev1</CodecTagString>
                    <CodecTimeBase>1/12800</CodecTimeBase>
                    <ColorPrimaries>bt470bg</ColorPrimaries>
                    <ColorRange>tv</ColorRange>
                    <ColorTransfer>smpte170m</ColorTransfer>
                    <Duration>12.960000</Duration>
                    <FieldOrder>progressive</FieldOrder>
                    <Fps>25.000000</Fps>
                    <HasBFrame>2</HasBFrame>
                    <Height>960</Height>
                    <Index>0</Index>
                    <Language>und</Language>
                    <Level>93</Level>
                    <NumFrames>323</NumFrames>
                    <PixFormat>yuv420p</PixFormat>
                    <Profile>Main</Profile>
                    <RefFrames>1</RefFrames>
                    <Rotation>0.000000</Rotation>
                    <StartTime>0.000000</StartTime>
                    <Timebase>1/12800</Timebase>
                    <Width>544</Width>
                </Video>
            </Stream>
        </MediaInfo>
        <MediaResult>
            <OutputFile>
                <Bucket>test-123456789</Bucket>
                <Md5Info>
                    <Md5>a825aa6c59371509995a47b570c29297</Md5>
                    <ObjectName>output/video-process.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/video-process.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/video-process.mp4</Object>
```

```
                <Region>ap-chongqing</Region>
            </Output>
            <TranscodeTemplateId>t156c107210e7243c5817354565d81b578</TranscodeTempl
            <TemplateId>t1f4ac18da8bdc403fa27ac4bfn77a62a1</TemplateId>
            <TemplateName>videoProcess_12654</TemplateName>
            <UserData>This is my VideoProcess job.</UserData>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T20:02:18+0800</StartTime>
        <State>Success</State>
        <Tag>VideoProcess</Tag>
        <Workflow>
            <Name>VideoProcess_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON



```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-06-30T20:02:17+0800",
        "EndTime": "2022-06-30T20:02:34+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp4"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1424687"
                }
            ]
        },
        "JobId": "j7d0eafdaf86c11ecbdf419856486c0ab",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "10269.090000",
                    "Duration": "13.654000",
                    "FormatLongName": "QuickTime / MOV",
                    "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
                    "NumProgram": "0",
                    "NumStream": "2",
                    "Size": "17526771",
                    "StartTime": "0.000000"
                },
                "Stream": {
                    "Audio": {
                        "Bitrate": "128.726000",
                        "Channel": "2",
                        "ChannelLayout": "stereo",
                        "CodecLongName": "AAC (Advanced Audio Coding)",
                        "CodecName": "aac",
                        "CodecTag": "0x6134706d",
                        "CodecTagString": "mp4a",
                        "CodecTimeBase": "1/44100",
                        "Duration": "13.652993",
                        "Index": "1",
                        "Language": "und",
                        "SampleFmt": "fltp",
                        "SampleRate": "44100",
                        "StartTime": "0.000000",
                        "Timebase": "1/44100"
```

```
                },
                "Video": {
                    "AvgFps": "24.922840",
                    "Bitrate": "10674.365000",
                    "CodecLongName": "H.265 / HEVC (High Efficiency Video Codin
                    "CodecName": "hevc",
                    "CodecTag": "0x31766568",
                    "CodecTagString": "hev1",
                    "CodecTimeBase": "1/12800",
                    "ColorPrimaries": "bt470bg",
                    "ColorRange": "tv",
                    "ColorTransfer": "smpte170m",
                    "Duration": "12.960000",
                    "FieldOrder": "progressive",
                    "Fps": "25.000000",
                    "HasBFrame": "2",
                    "Height": "960",
                    "Index": "0",
                    "Language": "und",
                    "Level": "93",
                    "NumFrames": "323",
                    "PixFormat": "yuv420p",
                    "Profile": "Main",
                    "RefFrames": "1",
                    "Rotation": "0.000000",
                    "StartTime": "0.000000",
                    "Timebase": "1/12800",
                    "Width": "544"
                }
            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "Md5Info": {
                    "Md5": "a825aa6c59371509995a47b570c29297",
                    "ObjectName": "output/video-process.mp4"
                },
                "ObjectName": "output/video-process.mp4",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/video-process.mp4",
            "Region": "ap-chongqing"
        },
```

```
            "TranscodeTemplateId": "t156c107210e7243c5817354565d81b578",
            "TemplateId": "t1f4ac18da8bdc403fa27ac4bfn77a62a1",
            "TemplateName": "videoProcess_12654",
            "UserData": "This is my VideoProcess job."
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T20:02:18+0800",
        "State": "Success",
        "Tag": "VideoProcess",
        "Workflow": {
            "Name": "VideoProcess_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

视频标签

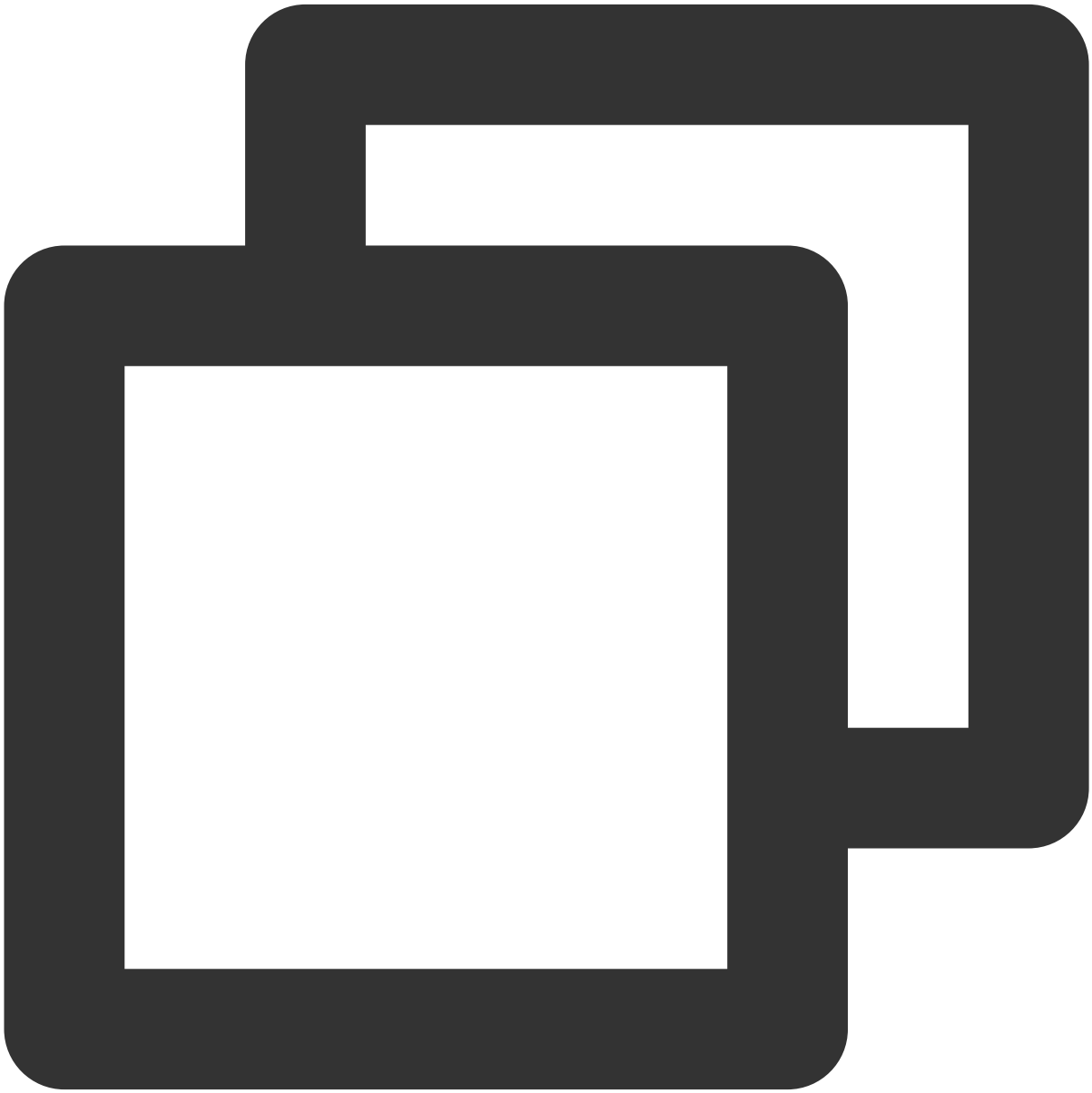
最近更新时间：2023-12-30 16:31:06

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:05:48+0800</CreationTime>
        <EndTime>2022-07-01T10:06:02+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j535b10a4f8e211ec9452c1c27f56a659</JobId>
        <Message>Success</Message>
        <Operation>
            <UserData>This is my VideoTag job.</UserData>
            <VideoTag>
                <Scenario>Stream</Scenario>
            </VideoTag>
            <VideoTagResult>
                <StreamData>
                    <Data>
                        <ActionTags>
                            <EndTime>11</EndTime>
                            <StartTime>0</StartTime>
                            <Tags>
                                <Confidence>0.421531</Confidence>
                                <Tag>说话</Tag>
                                <TagCls>说话</TagCls>
                            </Tags>
                        </ActionTags>
                        <ActionTags>
                            <EndTime>12.88</EndTime>
                            <StartTime>11.04</StartTime>
                            <Tags/>
                        </ActionTags>
                        <ObjectTags>
                            <Objects>
                                <BBox>
                                    <X1>0.10181817412376404</X1>
                                    <X2>0.8069194555282593</X2>
                                    <Y1>0.6880248785018921</Y1>
                                    <Y2>0.8779844045639038</Y2>
                                </BBox>
                                <Confidence>0.808393</Confidence>
                                <Name>电脑键盘</Name>
                            </Objects>
                            <TimeStamp>11</TimeStamp>
                        </ObjectTags>
                        <PersonTags/>
                        <PlaceTags>
                            <ClipFrameResult>办公室:0.55</ClipFrameResult>
                            <ClipFrameResult>办公室:0.55</ClipFrameResult>
                            <ClipFrameResult>办公室:0.59</ClipFrameResult>
                            <ClipFrameResult>办公室:0.54</ClipFrameResult>
                            <ClipFrameResult>办公室:0.42</ClipFrameResult>
                            <ClipFrameResult>办公室:0.43</ClipFrameResult>
                            <ClipFrameResult>办公室:0.33</ClipFrameResult>
                            <ClipFrameResult>办公室:0.29</ClipFrameResult>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                <ClipFrameResult>办公室:0.15</ClipFrameResult>
                <ClipFrameResult>娱乐场:0.42</ClipFrameResult>
                <ClipFrameResult>办公室:0.33</ClipFrameResult>
                <EndIndex>275</EndIndex>
                <EndTime>11</EndTime>
                <StartIndex>0</StartIndex>
                <StartTime>0</StartTime>
                <Tags>
                    <Confidence>0.591724</Confidence>
                    <Tag>办公室</Tag>
                    <TagCls>办公</TagCls>
                </Tags>
                <Tags>
                    <Confidence>0.175342</Confidence>
                    <Tag>超市/便利店/商店</Tag>
                    <TagCls>商业</TagCls>
                </Tags>
            </PlaceTags>
            <PlaceTags>
                <ClipFrameResult>其他:0.09</ClipFrameResult>
                <ClipFrameResult>其他:0.21</ClipFrameResult>
                <ClipFrameResult>其他:0.19</ClipFrameResult>
                <ClipFrameResult>天空:0.28</ClipFrameResult>
                <ClipFrameResult>天空:0.25</ClipFrameResult>
                <ClipFrameResult>其他:0.30</ClipFrameResult>
                <ClipFrameResult>其他:0.28</ClipFrameResult>
                <ClipFrameResult>其他:0.49</ClipFrameResult>
                <ClipFrameResult>其他:0.56</ClipFrameResult>
                <ClipFrameResult>其他:0.58</ClipFrameResult>
                <EndIndex>322</EndIndex>
                <EndTime>12.88</EndTime>
                <StartIndex>276</StartIndex>
                <StartTime>11.04</StartTime>
                <Tags>
                    <Confidence>0.581444</Confidence>
                    <Tag>其他</Tag>
                    <TagCls>其他</TagCls>
                </Tags>
                <Tags>
                    <Confidence>0.283301</Confidence>
                    <Tag>天空</Tag>
                    <TagCls>自然/户外</TagCls>
                </Tags>
            </PlaceTags>
            <Tags>
                <Confidence>0.355083</Confidence>
                <Tag>MOBA类-王者荣耀</Tag>
```

```
                        </Tags>
                        <Tags>
                            <Confidence>0.587103</Confidence>
                            <Tag>游戏</Tag>
                        </Tags>
                        <Tags>
                            <Confidence>0.544651</Confidence>
                            <Tag>办公室</Tag>
                        </Tags>
                        <Tags>
                            <Confidence>0.182886</Confidence>
                            <Tag>超市/便利店/商店</Tag>
                        </Tags>
                    </Data>
                    <SubErrCode>0</SubErrCode>
                    <SubErrMsg>ok</SubErrMsg>
                </StreamData>
            </VideoTagResult>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-07-01T10:05:50+0800</StartTime>
        <State>Success</State>
        <Tag>VideoTag</Tag>
        <VideoTag/>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交视频标签接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

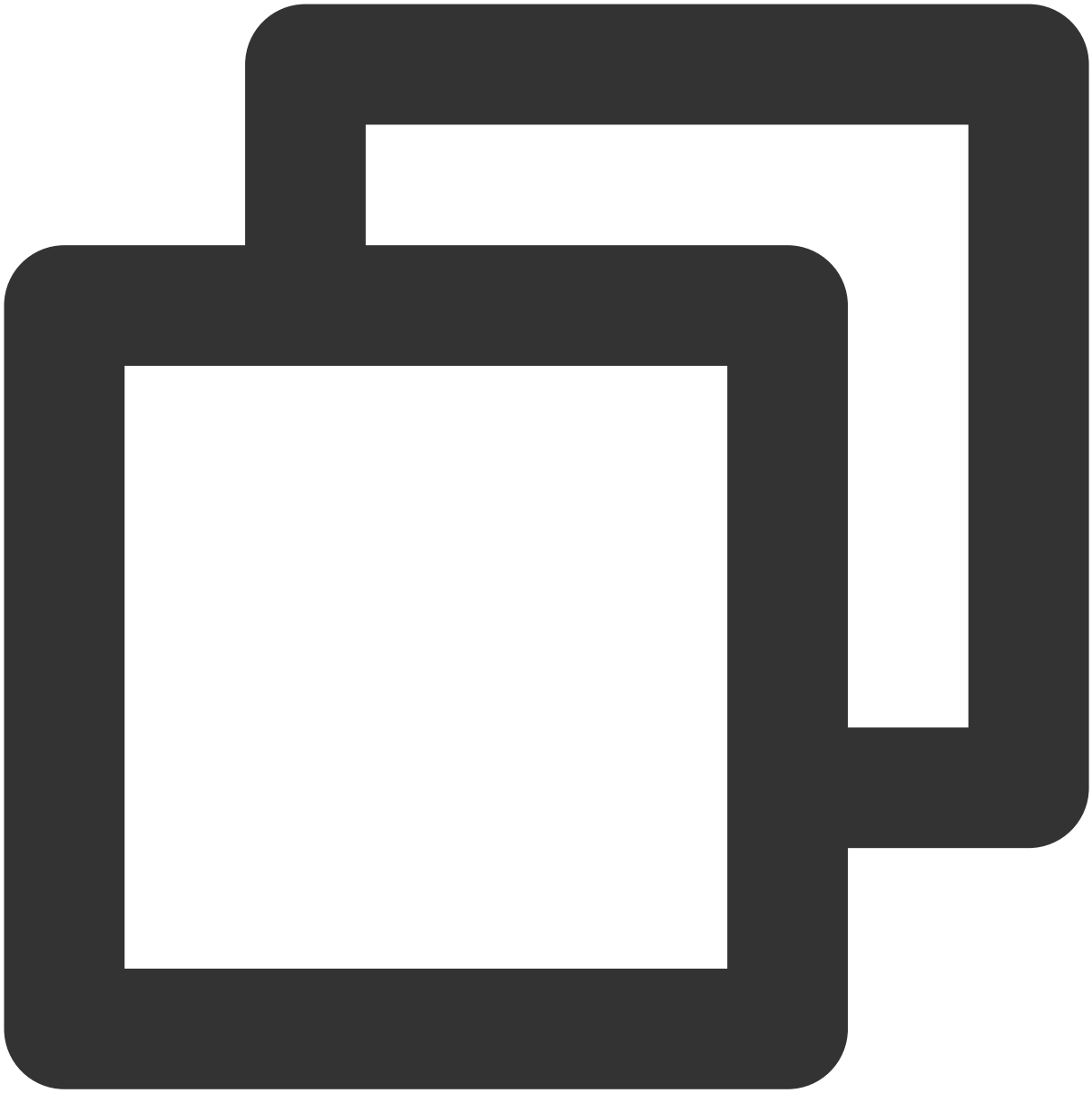
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:05:48+0800</CreationTime>
        <EndTime>2022-07-01T10:06:02+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

版权所有：腾讯云计算（北京）有限责任公司

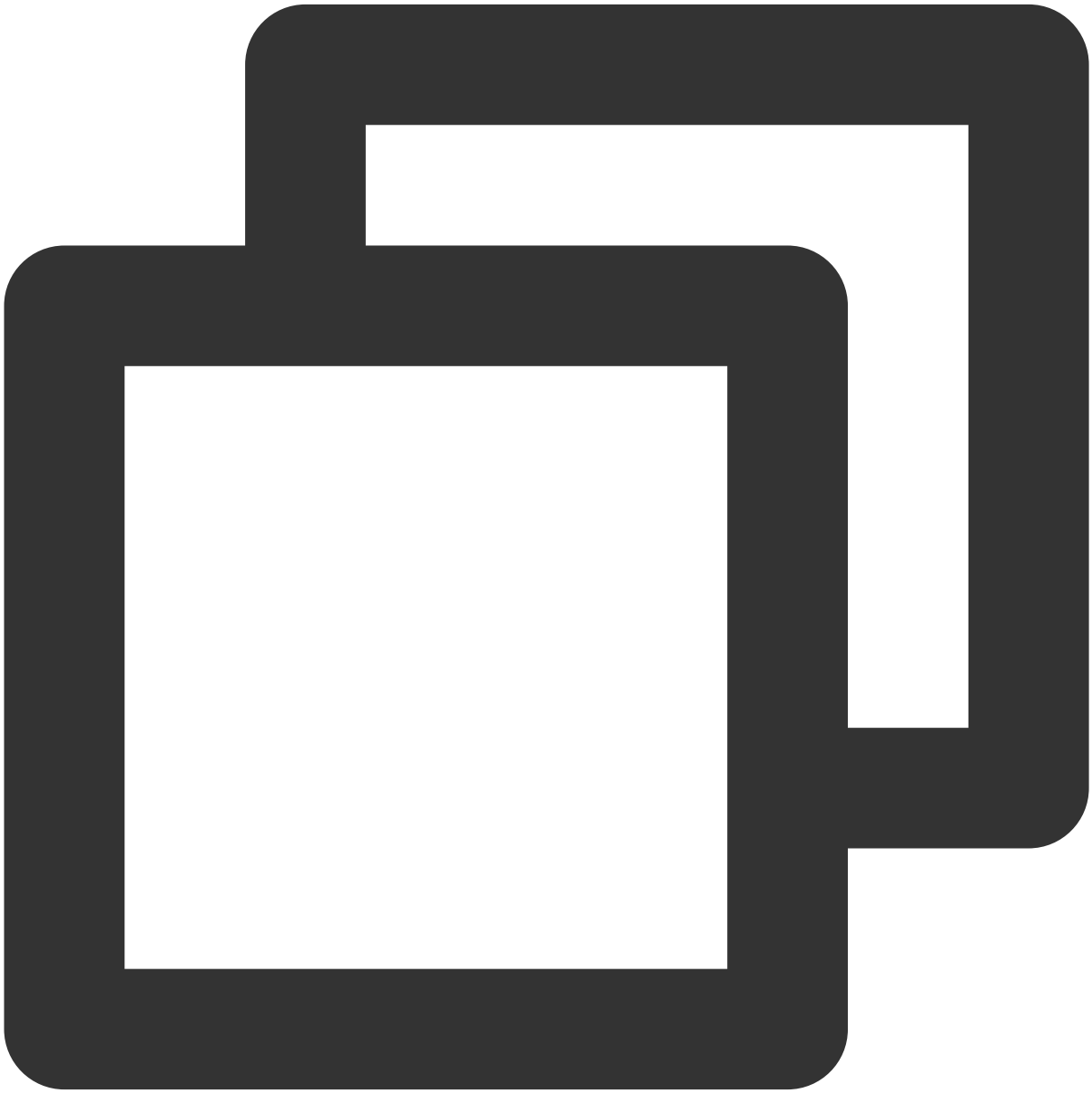
```
        <JobId>j535b10a4f8e211ec9452c1c27f56a659</JobId>
        <Message>Success</Message>
        <Operation>
            <UserData>This is my VideoTag job.</UserData>
            <VideoTag>
                <Scenario>Stream</Scenario>
            </VideoTag>
            <VideoTagResult>
                <StreamData>
                    <Data>
                        <ActionTags>
                            <EndTime>11</EndTime>
                            <StartTime>0</StartTime>
                            <Tags>
                                <Confidence>0.421531</Confidence>
                                <Tag>说话</Tag>
                                <TagCls>说话</TagCls>
                            </Tags>
                        </ActionTags>
                        <ActionTags>
                            <EndTime>12.88</EndTime>
                            <StartTime>11.04</StartTime>
                            <Tags/>
                        </ActionTags>
                        <ObjectTags>
                            <Objects>
                                <BBox>
                                    <X1>0.10181817412376404</X1>
                                    <X2>0.8069194555282593</X2>
                                    <Y1>0.6880248785018921</Y1>
                                    <Y2>0.8779844045639038</Y2>
                                </BBox>
                                <Confidence>0.808393</Confidence>
                                <Name>电脑键盘</Name>
                            </Objects>
                            <TimeStamp>11</TimeStamp>
                        </ObjectTags>
                        <PersonTags/>
                        <PlaceTags>
                            <ClipFrameResult>办公室:0.55</ClipFrameResult>
                            <ClipFrameResult>办公室:0.55</ClipFrameResult>
                            <ClipFrameResult>办公室:0.59</ClipFrameResult>
                            <ClipFrameResult>办公室:0.54</ClipFrameResult>
                            <ClipFrameResult>办公室:0.42</ClipFrameResult>
                            <ClipFrameResult>办公室:0.43</ClipFrameResult>
                            <ClipFrameResult>办公室:0.33</ClipFrameResult>
                            <ClipFrameResult>办公室:0.29</ClipFrameResult>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                    <ClipFrameResult>办公室:0.15</ClipFrameResult>
                    <ClipFrameResult>娱乐场:0.42</ClipFrameResult>
                    <ClipFrameResult>办公室:0.33</ClipFrameResult>
                    <EndIndex>275</EndIndex>
                    <EndTime>11</EndTime>
                    <StartIndex>0</StartIndex>
                    <StartTime>0</StartTime>
                    <Tags>
                        <Confidence>0.591724</Confidence>
                        <Tag>办公室</Tag>
                        <TagCls>办公</TagCls>
                    </Tags>
                    <Tags>
                        <Confidence>0.175342</Confidence>
                        <Tag>超市/便利店/商店</Tag>
                        <TagCls>商业</TagCls>
                    </Tags>
                </PlaceTags>
                <PlaceTags>
                    <ClipFrameResult>其他:0.09</ClipFrameResult>
                    <ClipFrameResult>其他:0.21</ClipFrameResult>
                    <ClipFrameResult>其他:0.19</ClipFrameResult>
                    <ClipFrameResult>天空:0.28</ClipFrameResult>
                    <ClipFrameResult>天空:0.25</ClipFrameResult>
                    <ClipFrameResult>其他:0.30</ClipFrameResult>
                    <ClipFrameResult>其他:0.28</ClipFrameResult>
                    <ClipFrameResult>其他:0.49</ClipFrameResult>
                    <ClipFrameResult>其他:0.56</ClipFrameResult>
                    <ClipFrameResult>其他:0.58</ClipFrameResult>
                    <EndIndex>322</EndIndex>
                    <EndTime>12.88</EndTime>
                    <StartIndex>276</StartIndex>
                    <StartTime>11.04</StartTime>
                    <Tags>
                        <Confidence>0.581444</Confidence>
                        <Tag>其他</Tag>
                        <TagCls>其他</TagCls>
                    </Tags>
                    <Tags>
                        <Confidence>0.283301</Confidence>
                        <Tag>天空</Tag>
                        <TagCls>自然/户外</TagCls>
                    </Tags>
                </PlaceTags>
                <Tags>
                    <Confidence>0.355083</Confidence>
                    <Tag>MOBA类-王者荣耀</Tag>
```

```
                        </Tags>
                        <Tags>
                            <Confidence>0.587103</Confidence>
                            <Tag>游戏</Tag>
                        </Tags>
                        <Tags>
                            <Confidence>0.544651</Confidence>
                            <Tag>办公室</Tag>
                        </Tags>
                        <Tags>
                            <Confidence>0.182886</Confidence>
                            <Tag>超市/便利店/商店</Tag>
                        </Tags>
                    </Data>
                    <SubErrCode>0</SubErrCode>
                    <SubErrMsg>ok</SubErrMsg>
                </StreamData>
            </VideoTagResult>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-07-01T10:05:50+0800</StartTime>
        <State>Success</State>
        <Tag>VideoTag</Tag>
        <VideoTag/>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-07-01T10:05:48+0800",
        "EndTime": "2022-07-01T10:06:02+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing"
        },
```


版权所有：腾讯云计算（北京）有限责任公司

```
        "JobId": "j535b10a4f8e211ec9452c1c27f56a659",
        "Message": "Success",
        "Operation": {
            "UserData": "This is my VideoTag job.",
            "VideoTag": {
                "Scenario": "Stream"
            },
            "VideoTagResult": {
                "StreamData": {
                    "Data": {
                        "ActionTags": [{
                                "EndTime": "11",
                                "StartTime": "0",
                                "Tags": {
                                    "Confidence": "0.421531",
                                    "Tag": "说话",
                                    "TagCls": "说话"
                                }
                            },
                            {
                                "EndTime": "12.88",
                                "StartTime": "11.04"
                            }
                        ],
                        "ObjectTags": {
                            "Objects": {
                                "BBox": {
                                    "X1": "0.10181817412376404",
                                    "X2": "0.8069194555282593",
                                    "Y1": "0.6880248785018921",
                                    "Y2": "0.8779844045639038"
                                },
                                "Confidence": "0.808393",
                                "Name": "电脑键盘"
                            },
                            "TimeStamp": "11"
                        },
                        "PlaceTags": [{
                                "ClipFrameResult": [
                                    "办公室:0.55",
                                    "办公室:0.55",
                                    "办公室:0.59",
                                    "办公室:0.54",
                                    "办公室:0.42",
                                    "办公室:0.43",
                                    "办公室:0.33",
                                    "办公室:0.29",
```

```
                "办公室:0.15",
                "娱乐场:0.42",
                "办公室:0.33"
            ],
            "EndIndex": "275",
            "EndTime": "11",
            "StartIndex": "0",
            "StartTime": "0",
            "Tags": [{
                    "Confidence": "0.591724",
                    "Tag": "办公室",
                    "TagCls": "办公"
                },
                {
                    "Confidence": "0.175342",
                    "Tag": "超市/便利店/商店",
                    "TagCls": "商业"
                }
            ]
        },
        {
            "ClipFrameResult": [
                "其他:0.09",
                "其他:0.21",
                "其他:0.19",
                "天空:0.28",
                "天空:0.25",
                "其他:0.30",
                "其他:0.28",
                "其他:0.49",
                "其他:0.56",
                "其他:0.58"
            ],
            "EndIndex": "322",
            "EndTime": "12.88",
            "StartIndex": "276",
            "StartTime": "11.04",
            "Tags": [{
                    "Confidence": "0.581444",
                    "Tag": "其他",
                    "TagCls": "其他"
                },
                {
                    "Confidence": "0.283301",
                    "Tag": "天空",
                    "TagCls": "自然/户外"
                }
```

```
                                ]
                            }
                        ],
                        "Tags": [{
                                "Confidence": "0.355083",
                                "Tag": "MOBA类-王者荣耀"
                            },
                            {
                                "Confidence": "0.587103",
                                "Tag": "游戏"
                            },
                            {
                                "Confidence": "0.544651",
                                "Tag": "办公室"
                            },
                            {
                                "Confidence": "0.182886",
                                "Tag": "超市/便利店/商店"
                            }
                        ]
                    },
                    "SubErrCode": "0",
                    "SubErrMsg": "ok"
                }
            }
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-07-01T10:05:50+0800",
        "State": "Success",
        "Tag": "VideoTag"
    }
}
```

人声分离

最近更新时间：2023-12-30 16:32:09

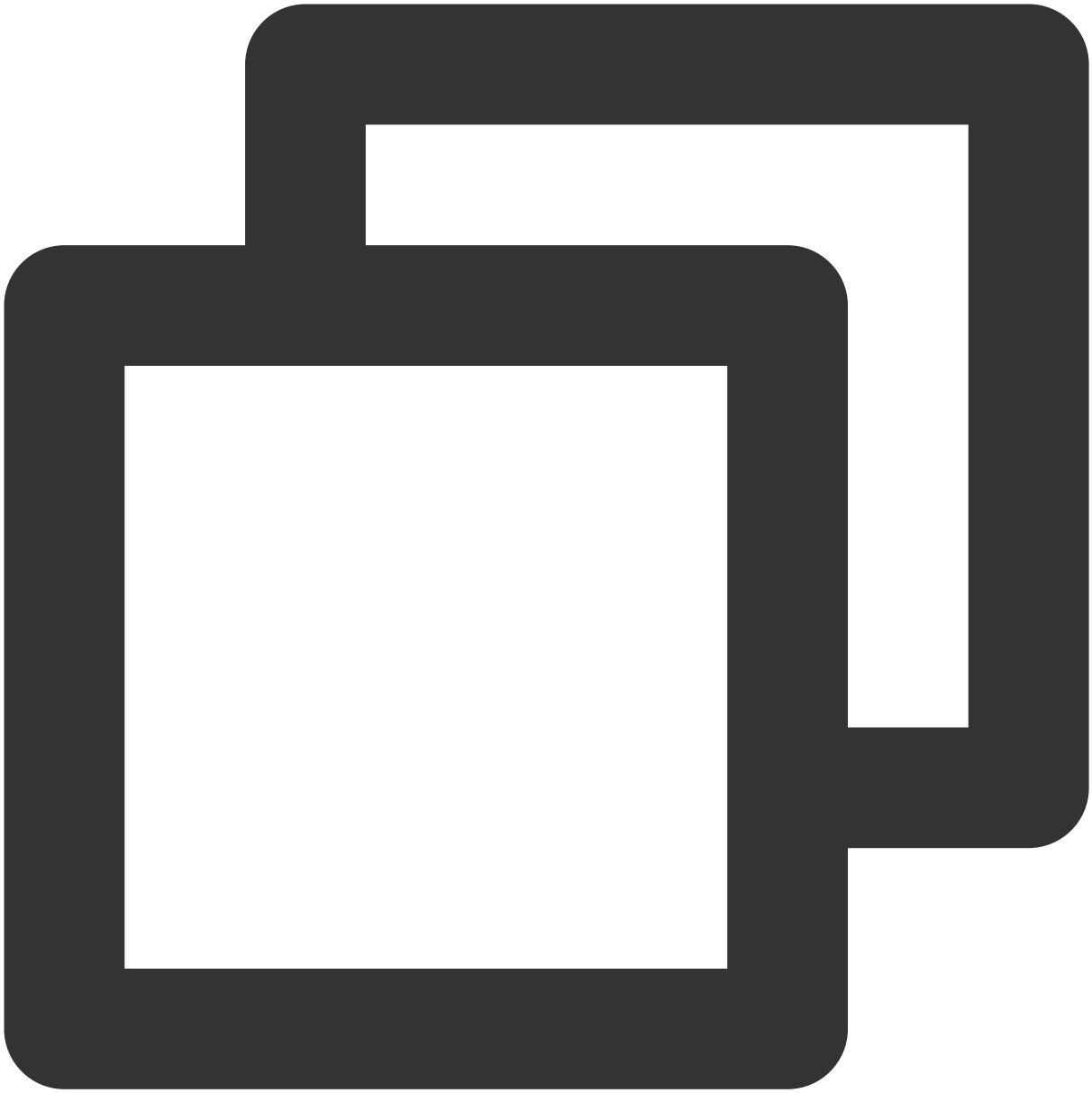
功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:12:26+0800</CreationTime>
        <EndTime>2022-07-01T10:12:32+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j40644e92f8e311ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>128.147000</Bitrate>
                    <Duration>13.688163</Duration>
                    <FormatLongName>MP2/3 (MPEG audio layer 2/3)</FormatLongName>
                    <FormatName>mp3</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>219263</Size>
                    <StartTime>0.025057</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>MP3 (MPEG audio layer 3)</CodecLongName>
                        <CodecName>mp3</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/14112000</CodecTimeBase>
                        <Duration>13.688163</Duration>
                        <Index>0</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.025057</StartTime>
                        <Timebase>1/14112000</Timebase>
                    </Audio>
                    <Audio>
                        <Bitrate>128.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>MP3 (MPEG audio layer 3)</CodecLongName>
                        <CodecName>mp3</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/14112000</CodecTimeBase>
                        <Duration>13.688163</Duration>
                        <Index>0</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.025057</StartTime>
                        <Timebase>1/14112000</Timebase>
                    </Audio>
```

```
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>3b883c0462ba845acdf3eb7e30db1a1c</Md5>
                        <ObjectName>output/audio.mp3</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>62de6bb513d6105eeeb1a6ab0077cb5a</Md5>
                        <ObjectName>output/backgroud.mp3</ObjectName>
                    </Md5Info>
                    <ObjectName>output/audio.mp3</ObjectName>
                    <ObjectName>output/backgroud.mp3</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <AuObject>output/audio.mp3</AuObject>
                <Bucket>test-123456789</Bucket>
                <Object>output/backgroud.mp3</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my VoiceSeparate job.</UserData>
            <TemplateId>t1f6ac18da8bdc403fa27ac4ffnd7a6sa1</TemplateId>
            <TemplateName>videoseparate_12654</TemplateName>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-07-01T10:12:27+0800</StartTime>
        <State>Success</State>
        <Tag>VoiceSeparate</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型

EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交人声分离任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

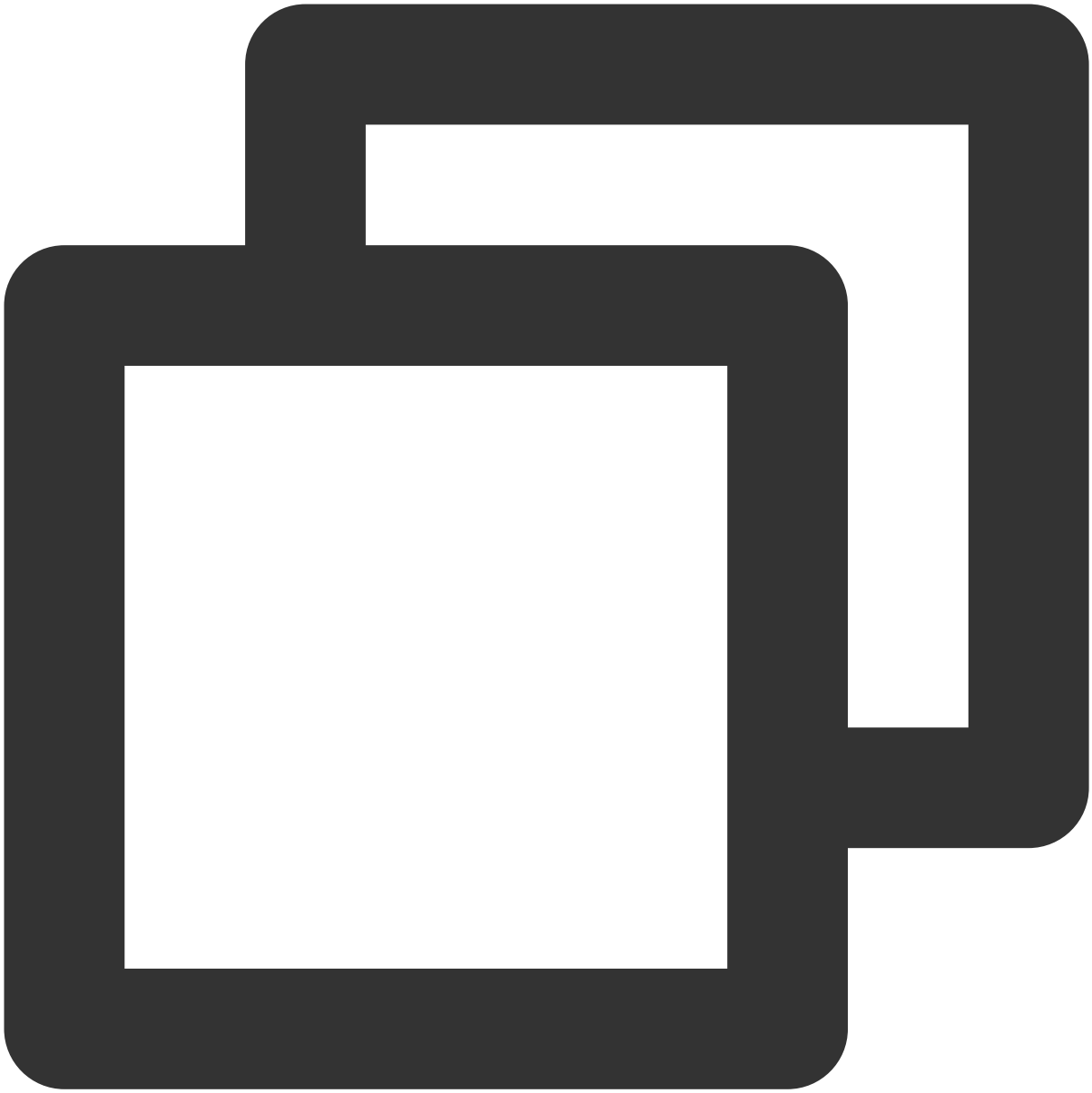
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

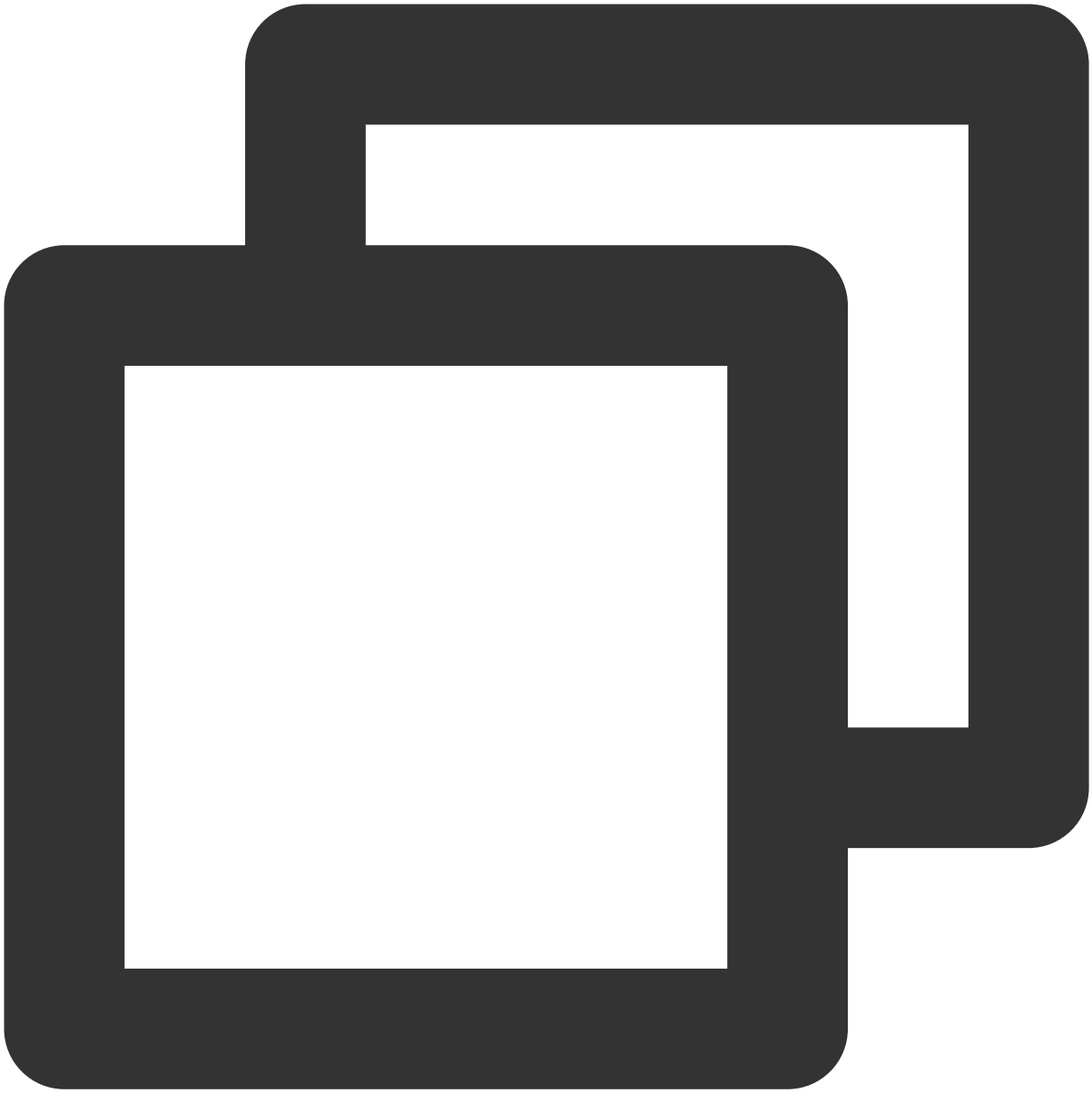


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:12:26+0800</CreationTime>
        <EndTime>2022-07-01T10:12:32+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>j40644e92f8e311ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>128.147000</Bitrate>
                    <Duration>13.688163</Duration>
                    <FormatLongName>MP2/3 (MPEG audio layer 2/3)</FormatLongName>
                    <FormatName>mp3</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>219263</Size>
                    <StartTime>0.025057</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>MP3 (MPEG audio layer 3)</CodecLongName>
                        <CodecName>mp3</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/14112000</CodecTimeBase>
                        <Duration>13.688163</Duration>
                        <Index>0</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.025057</StartTime>
                        <Timebase>1/14112000</Timebase>
                    </Audio>
                    <Audio>
                        <Bitrate>128.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>MP3 (MPEG audio layer 3)</CodecLongName>
                        <CodecName>mp3</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/14112000</CodecTimeBase>
                        <Duration>13.688163</Duration>
                        <Index>0</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.025057</StartTime>
                        <Timebase>1/14112000</Timebase>
                    </Audio>
```

```
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>3b883c0462ba845acdf3eb7e30db1a1c</Md5>
                        <ObjectName>output/audio.mp3</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>62de6bb513d6105eeeb1a6ab0077cb5a</Md5>
                        <ObjectName>output/backgroud.mp3</ObjectName>
                    </Md5Info>
                    <ObjectName>output/audio.mp3</ObjectName>
                    <ObjectName>output/backgroud.mp3</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <AuObject>output/audio.mp3</AuObject>
                <Bucket>test-123456789</Bucket>
                <Object>output/backgroud.mp3</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my VoiceSeparate job.</UserData>
            <TemplateId>t1f6ac18da8bdc403fa27ac4ffnd7a6sa1</TemplateId>
            <TemplateName>videoseparate_12654</TemplateName>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-07-01T10:12:27+0800</StartTime>
        <State>Success</State>
        <Tag>VoiceSeparate</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



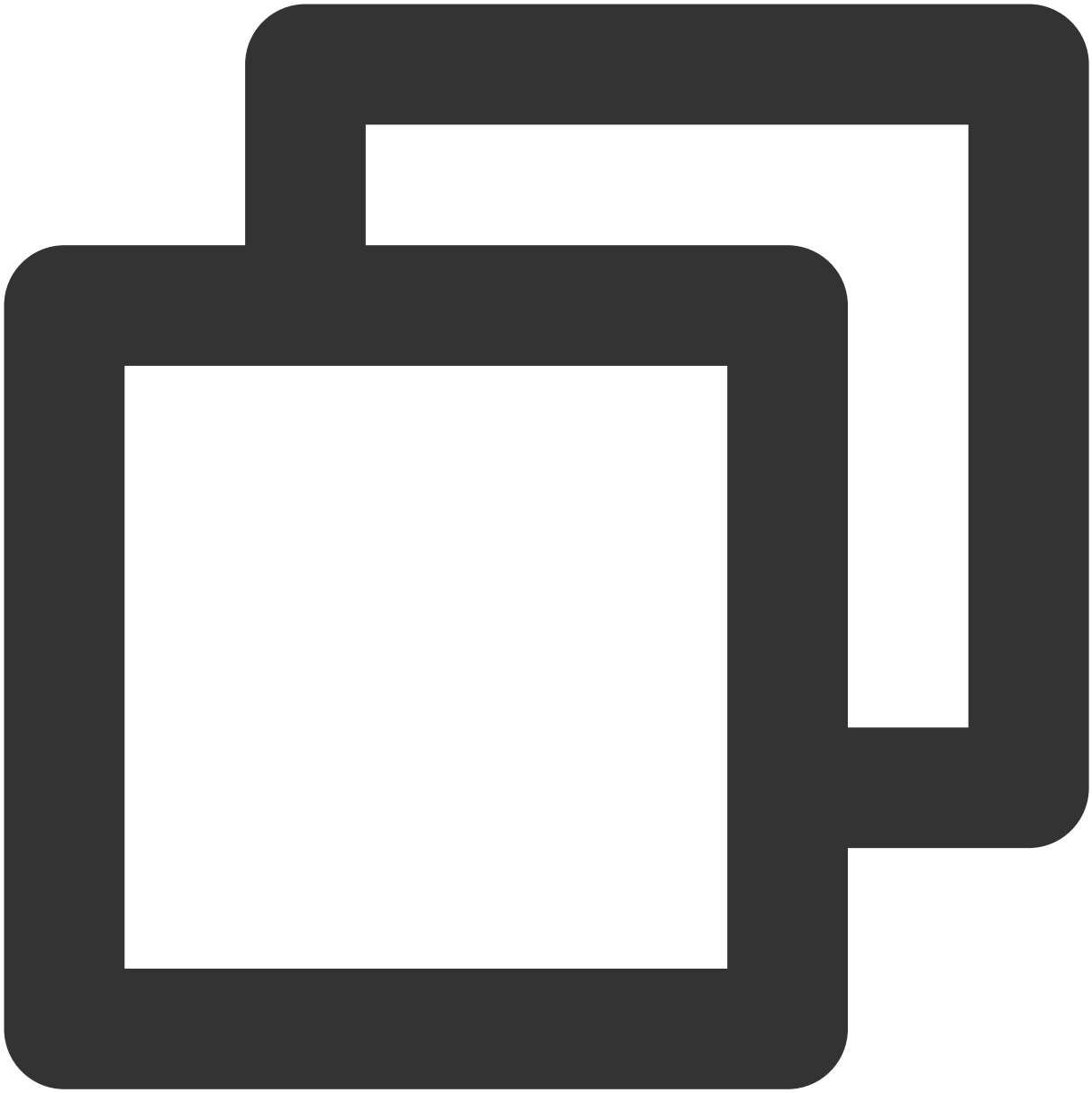
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:12:26+0800</CreationTime>
        <EndTime>2022-07-01T10:12:32+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>video/mp3</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1326515</Value>
            </CosHeaders>
        </Input>
        <JobId>j40644e92f8e311ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <MediaInfo>
                <Format>
                    <Bitrate>128.147000</Bitrate>
                    <Duration>13.688163</Duration>
                    <FormatLongName>MP2/3 (MPEG audio layer 2/3)</FormatLongName>
                    <FormatName>mp3</FormatName>
                    <NumProgram>0</NumProgram>
                    <NumStream>1</NumStream>
                    <Size>219263</Size>
                    <StartTime>0.025057</StartTime>
                </Format>
                <Stream>
                    <Audio>
                        <Bitrate>128.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>MP3 (MPEG audio layer 3)</CodecLongName>
                        <CodecName>mp3</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/14112000</CodecTimeBase>
                        <Duration>13.688163</Duration>
                        <Index>0</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.025057</StartTime>
                        <Timebase>1/14112000</Timebase>
                    </Audio>
```

```
                    <Audio>
                        <Bitrate>128.000000</Bitrate>
                        <Channel>2</Channel>
                        <ChannelLayout>stereo</ChannelLayout>
                        <CodecLongName>MP3 (MPEG audio layer 3)</CodecLongName>
                        <CodecName>mp3</CodecName>
                        <CodecTag>0x0000</CodecTag>
                        <CodecTagString>[0][0][0][0]</CodecTagString>
                        <CodecTimeBase>1/14112000</CodecTimeBase>
                        <Duration>13.688163</Duration>
                        <Index>0</Index>
                        <SampleFmt>fltp</SampleFmt>
                        <SampleRate>44100</SampleRate>
                        <StartTime>0.025057</StartTime>
                        <Timebase>1/14112000</Timebase>
                    </Audio>
                    <Subtitle/>
                    <Video/>
                </Stream>
            </MediaInfo>
            <MediaResult>
                <OutputFile>
                    <Bucket>test-123456789</Bucket>
                    <Md5Info>
                        <Md5>3b883c0462ba845acdf3eb7e30db1a1c</Md5>
                        <ObjectName>output/audio.mp3</ObjectName>
                    </Md5Info>
                    <Md5Info>
                        <Md5>62de6bb513d6105eeeb1a6ab0077cb5a</Md5>
                        <ObjectName>output/backgroud.mp3</ObjectName>
                    </Md5Info>
                    <ObjectName>output/audio.mp3</ObjectName>
                    <ObjectName>output/backgroud.mp3</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <AuObject>output/audio.mp3</AuObject>
                <Bucket>test-123456789</Bucket>
                <Object>output/backgroud.mp3</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <UserData>This is my VoiceSeparate job.</UserData>
            <TemplateId>t1f6ac18da8bdc403fa27ac4ffnd7a6sa1</TemplateId>
            <TemplateName>videoseparate_12654</TemplateName>
        </Operation>
```

```
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-07-01T10:12:27+0800</StartTime>
        <State>Success</State>
        <Tag>VoiceSeparate</Tag>
        <Workflow>
            <Name>VoiceSeparate_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-07-01T10:12:26+0800",
        "EndTime": "2022-07-01T10:12:32+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp3",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "video/mp3"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1326515"
                }
            ]
        },
        "JobId": "j40644e92f8e311ecb90d0b03267ce0e5",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "128.147000",
                    "Duration": "13.688163",
                    "FormatLongName": "MP2/3 (MPEG audio layer 2/3)",
                    "FormatName": "mp3",
                    "NumProgram": "0",
                    "NumStream": "1",
                    "Size": "219263",
                    "StartTime": "0.025057"
                },
                "Stream": {
                    "Audio": [{
                            "Bitrate": "128.000000",
                            "Channel": "2",
                            "ChannelLayout": "stereo",
                            "CodecLongName": "MP3 (MPEG audio layer 3)",
                            "CodecName": "mp3",
                            "CodecTag": "0x0000",
                            "CodecTagString": "[0][0][0][0]",
                            "CodecTimeBase": "1/14112000",
                            "Duration": "13.688163",
                            "Index": "0",
                            "SampleFmt": "fltp",
                            "SampleRate": "44100",
                            "StartTime": "0.025057",
                            "Timebase": "1/14112000"
                        },
```

```
                {
                    "Bitrate": "128.000000",
                    "Channel": "2",
                    "ChannelLayout": "stereo",
                    "CodecLongName": "MP3 (MPEG audio layer 3)",
                    "CodecName": "mp3",
                    "CodecTag": "0x0000",
                    "CodecTagString": "[0][0][0][0]",
                    "CodecTimeBase": "1/14112000",
                    "Duration": "13.688163",
                    "Index": "0",
                    "SampleFmt": "fltp",
                    "SampleRate": "44100",
                    "StartTime": "0.025057",
                    "Timebase": "1/14112000"
                }
            ]
        }
    },
    "MediaResult": {
        "OutputFile": {
            "Bucket": "test-123456789",
            "Md5Info": [{
                    "Md5": "3b883c0462ba845acdf3eb7e30db1a1c",
                    "ObjectName": "output/audio.mp3"
                },
                {
                    "Md5": "62de6bb513d6105eeeb1a6ab0077cb5a",
                    "ObjectName": "output/backgroud.mp3"
                }
            ],
            "ObjectName": [
                "output/audio.mp3",
                "output/backgroud.mp3"
            ],
            "Region": "ap-chongqing"
        }
    },
    "Output": {
        "AuObject": "output/audio.mp3",
        "Bucket": "test-123456789",
        "Object": "output/backgroud.mp3",
        "Region": "ap-chongqing"
    },
    "UserData": "This is my VoiceSeparate job.",
    "TemplateId": "t1f6ac18da8bdc403fa27ac4ffnd7a6sa1",
    "TemplateName": "videoseparate_12654"
```

```
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-07-01T10:12:27+0800",
        "State": "Success",
        "Tag": "VoiceSeparate",
        "Workflow": {
            "Name": "VoiceSeparate_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

图片处理
提交图片处理任务
CreatePicJobs

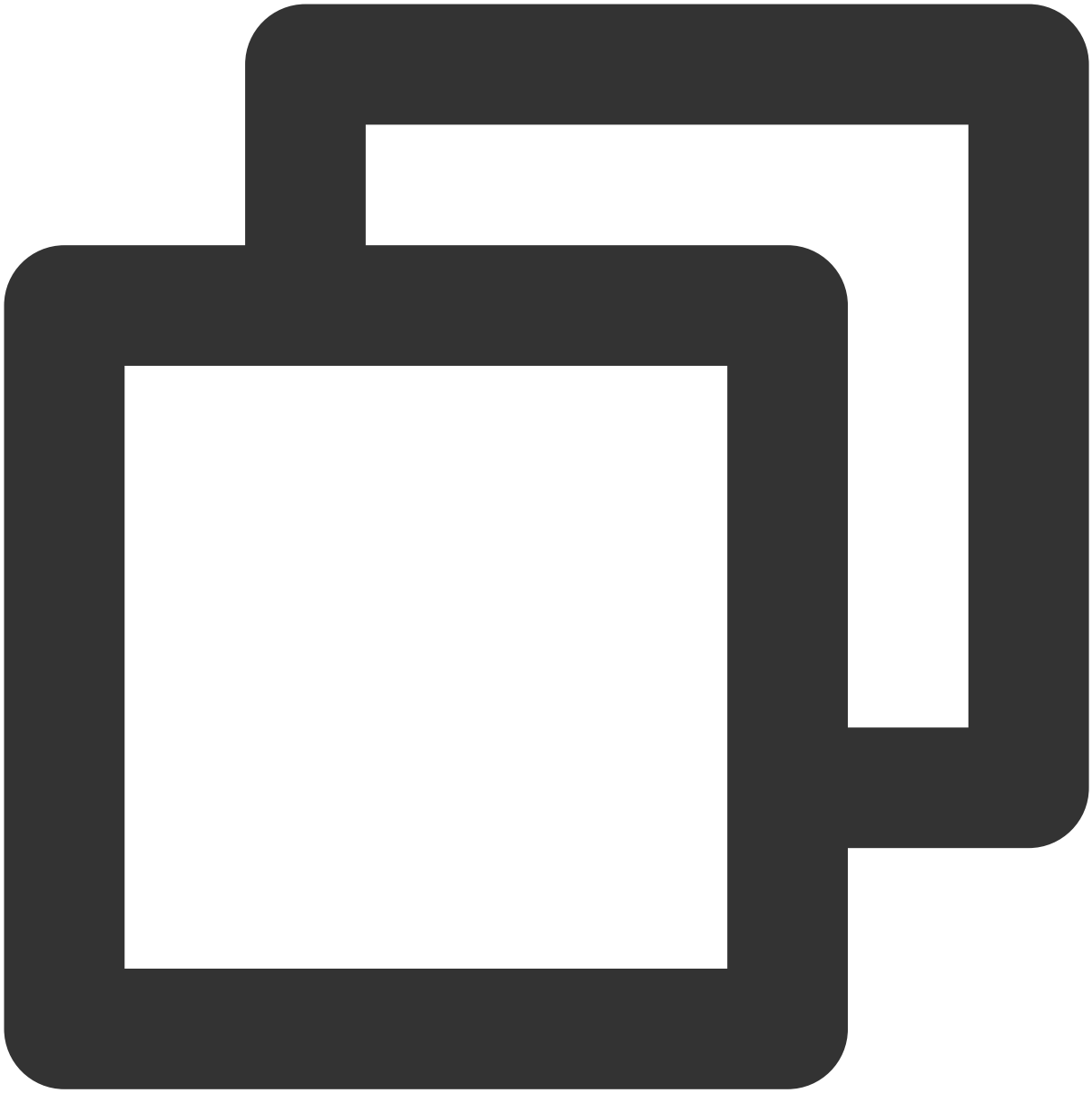
最近更新时间：2023-12-30 16:42:48

功能描述

CreatePicJobs 用于提交一个图片处理任务。

请求

请求示例



```
POST /pic_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

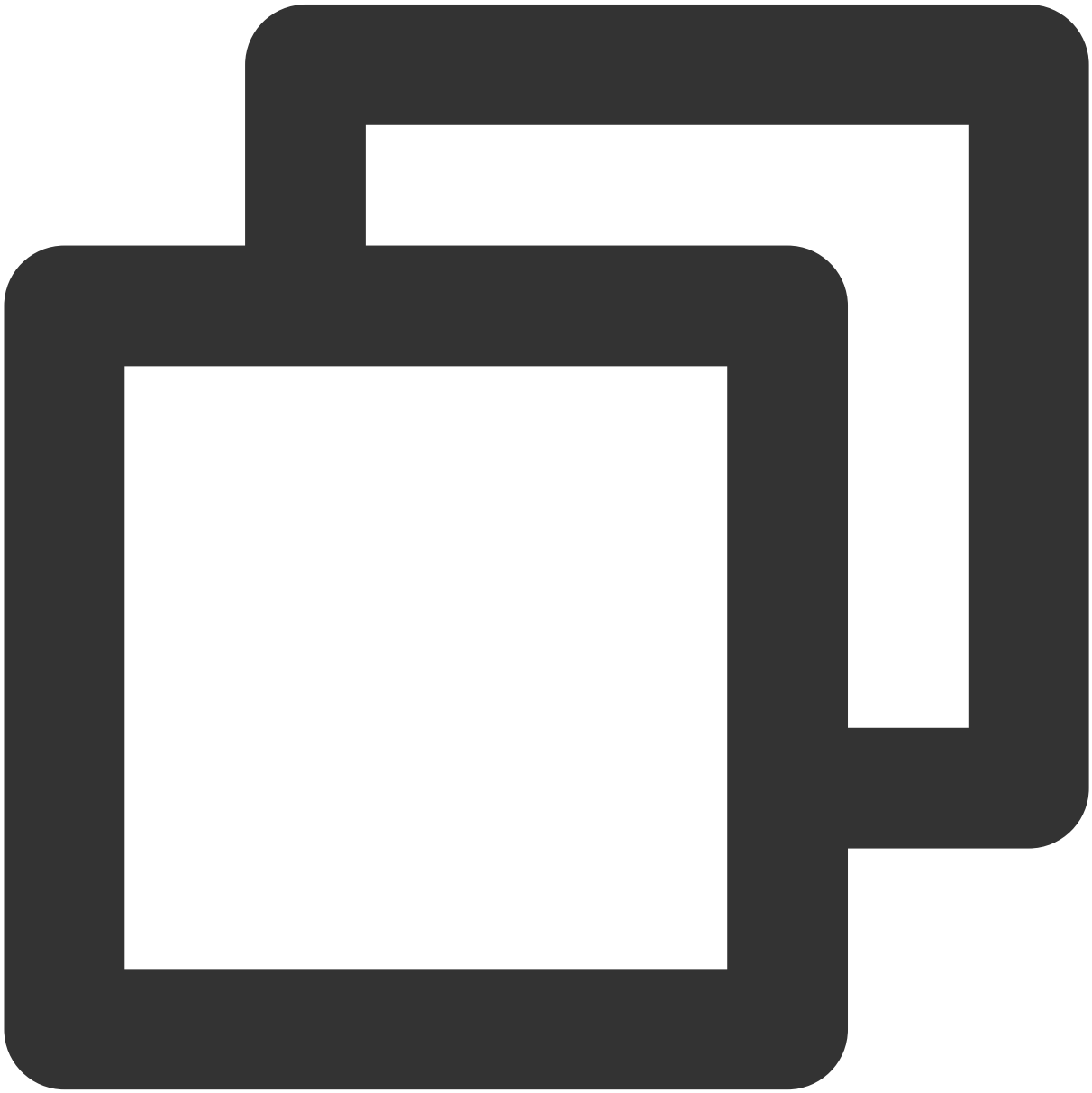
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>PicProcess</Tag>
    <Input>
        <Object>input/deer.jpg</Object>
    <Operation>
        <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.jpg</Object>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	创建任务的 Tag：PicProcess	String	是
Input	Request	待操作的媒体信息	Container	是
Operation	Request	操作规则	Container	是
QueueId	Request	任务所在的队列 ID	String	是
CallBack	Request	任务回调地址，优先级高于队列的回调地址。设置为 no 时，表示队列的回调地址不产生回调	String	否
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	媒体文件名	String	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PicProcess	Request.Operation	指定该任务的参数，同创建图片处理模板 CreateMediaTemplate 接口中的 Request.PicProcess	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
Output	Request.Operation	结果输出地址	Container	是

注意

优先使用 TemplateId，无 TemplateId 时使用对应任务类型的参数。

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Region	Request.Operation.Output	存储桶的地域	String	是
Bucket	Request.Operation.Output	存储结果的存储桶	String	是
Object	Request.Operation.Output	结果文件的名字	String	是

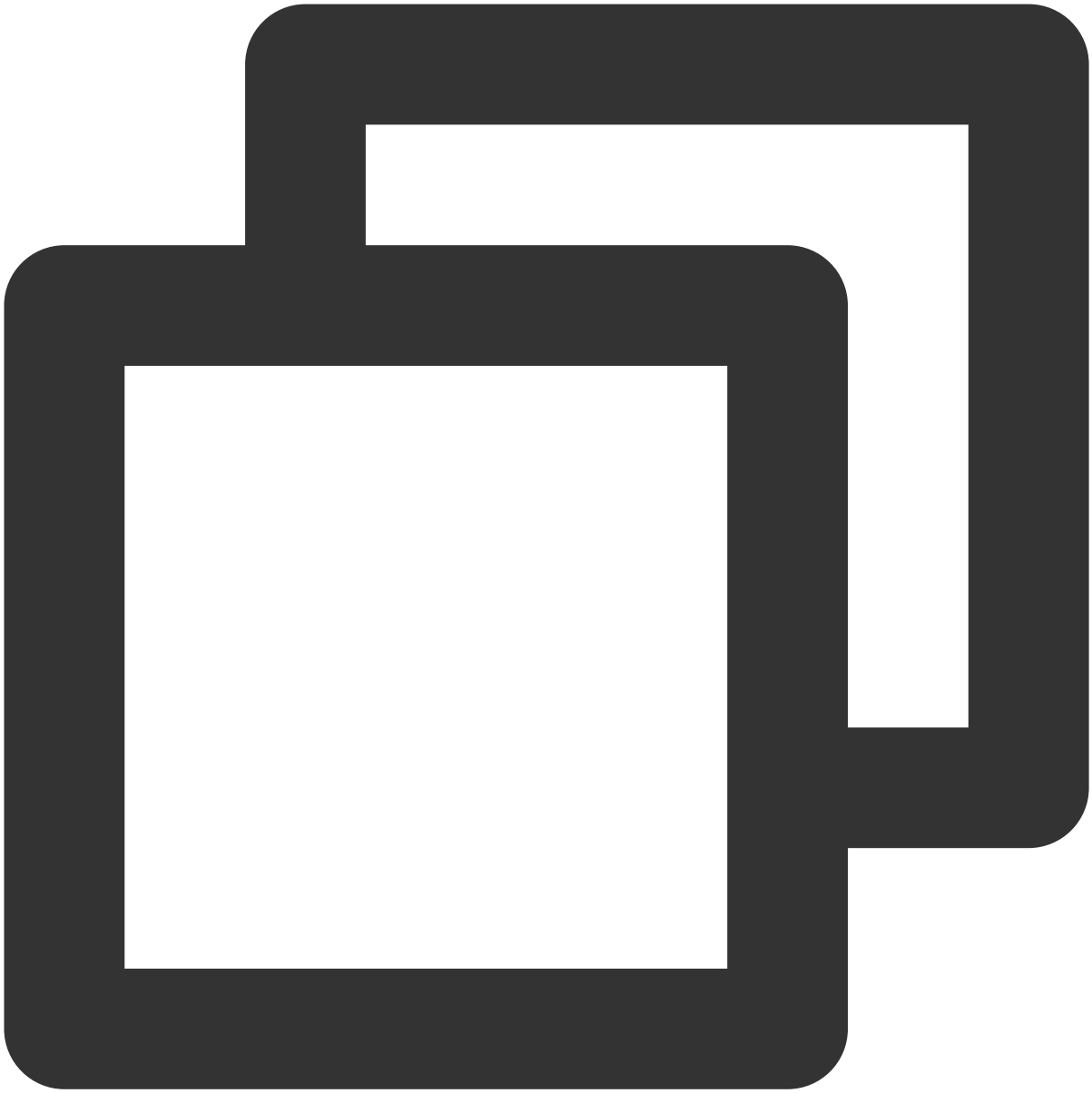
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>c93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>PicProcess</Tag>
```

```
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/out.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
            <TemplateName>test</TemplateName>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
JobId	Response.JobsDetail	新创建任务的 ID	String
Tag	Response.JobsDetail	新创建任务的 Tag：PicProcess	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String

版权所有：腾讯云计算（北京）有限责任公司

StartTime	Response.JobsDetail	任务的开始时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
QueueId	Response.JobsDetail	任务所属的队列 ID	String
Input	Response.JobsDetail	该任务的输入资源地址	Container
Operation	Response.JobsDetail	该任务的规则	Container

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Input	存储桶的地域	String
Bucket	Response.JobsDetail.Input	存储结果的存储桶	String
Object	Response.JobsDetail.Input	输出结果的文件名	String

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.JobsDetail.Operation	任务的模板 ID	String
TemplateName	Response.JobsDetail.Operation	任务的模板名称, 当 TemplateId 存在时返回	String
PicProcess	Response.JobsDetail.Operation	同请求中的 Request.Operation.PicProcess	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
PicProcessResult	Response.JobsDetail.Operation	图片处理后输出图片的信息，没有时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String

Container 节点 PicProcessResult 的内容：

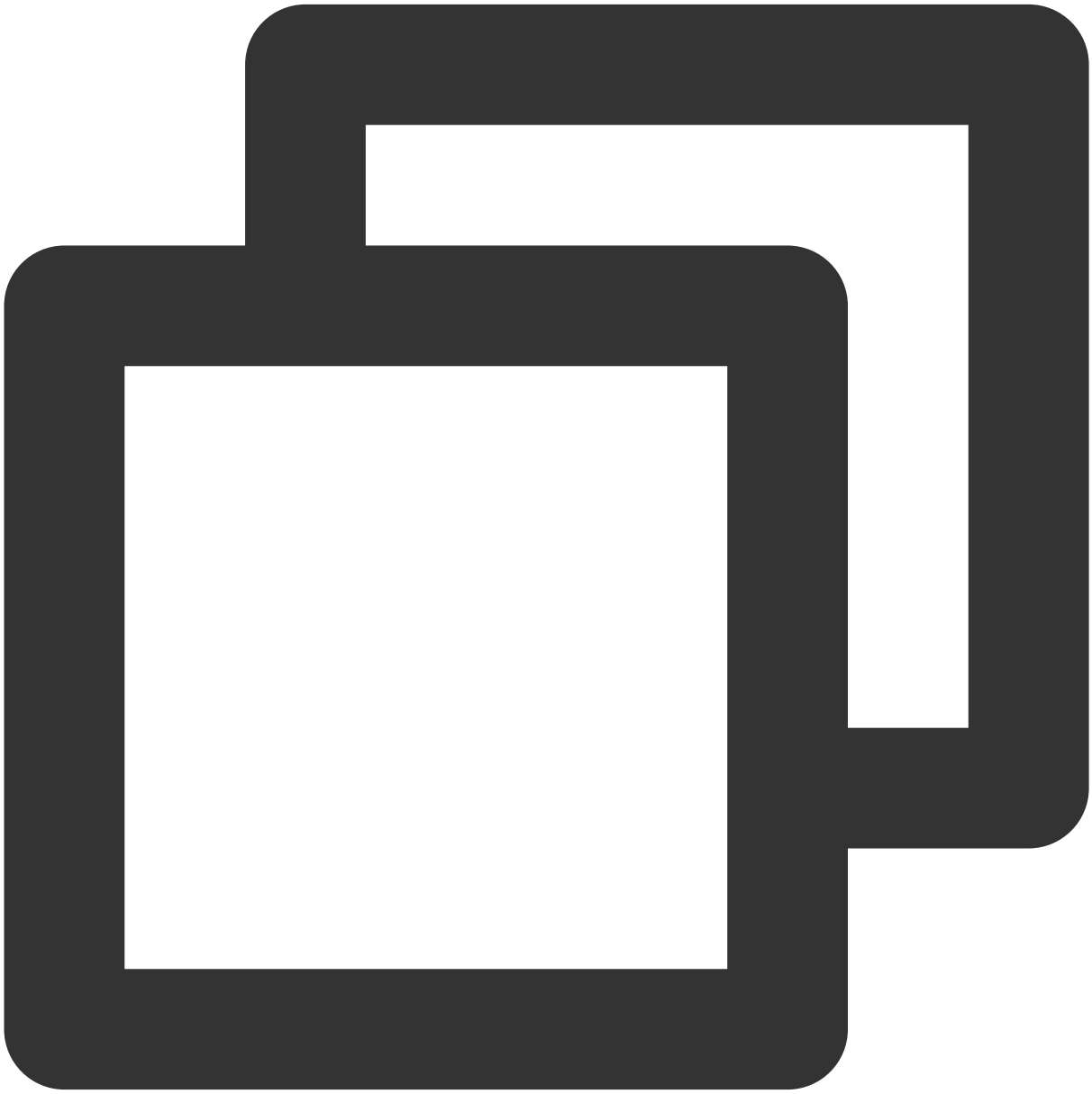
同 图片持久化处理 接口中的 UploadResult 节点。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

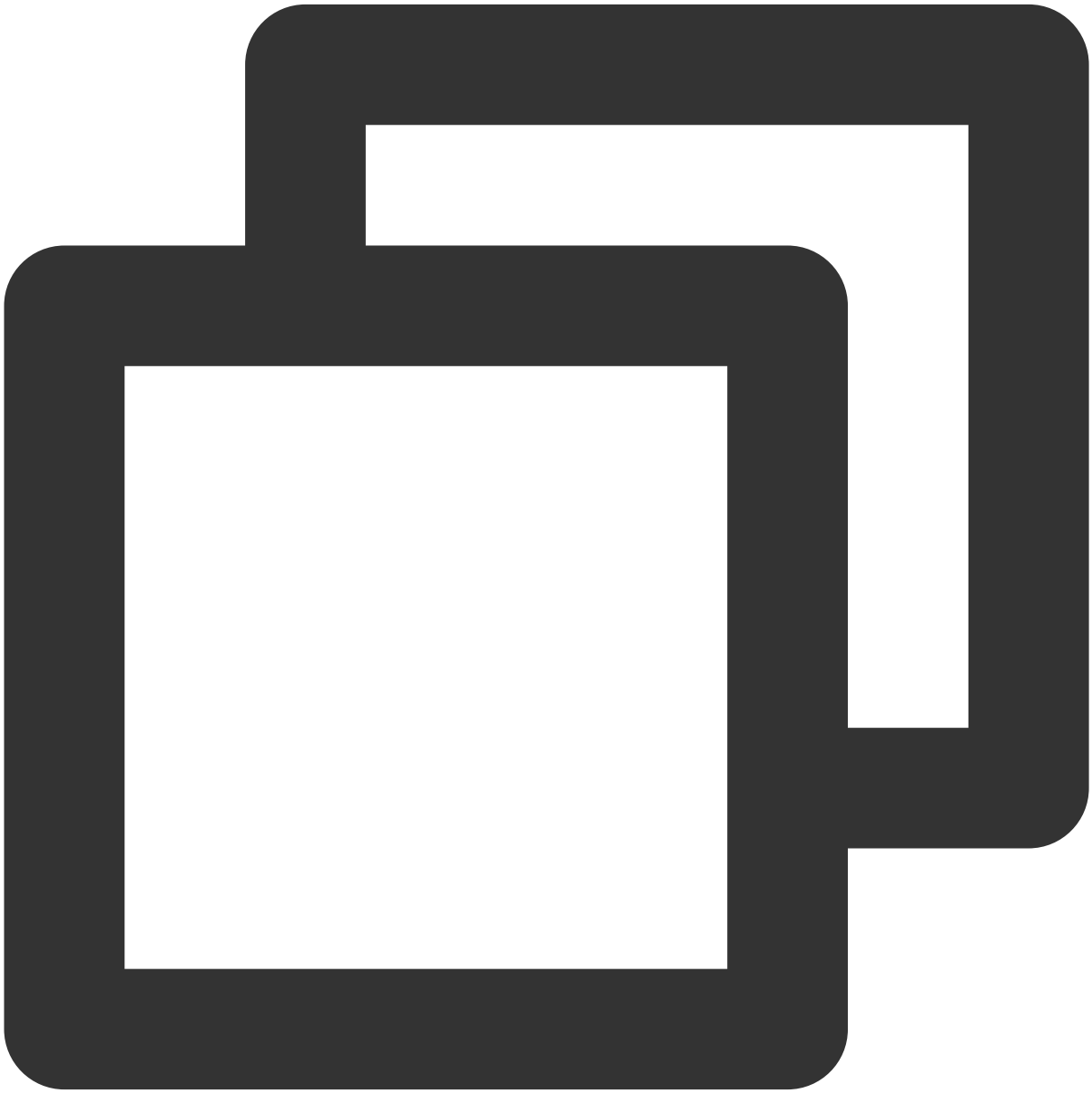
请求1：使用图片处理模板 ID



```
POST /pic_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml
```

```
<Request>
    <Tag>PicProcess</Tag>
    <Input>
        <Object>input/deer.jpg</Object>
    <Operation>
        <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.jpg</Object>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

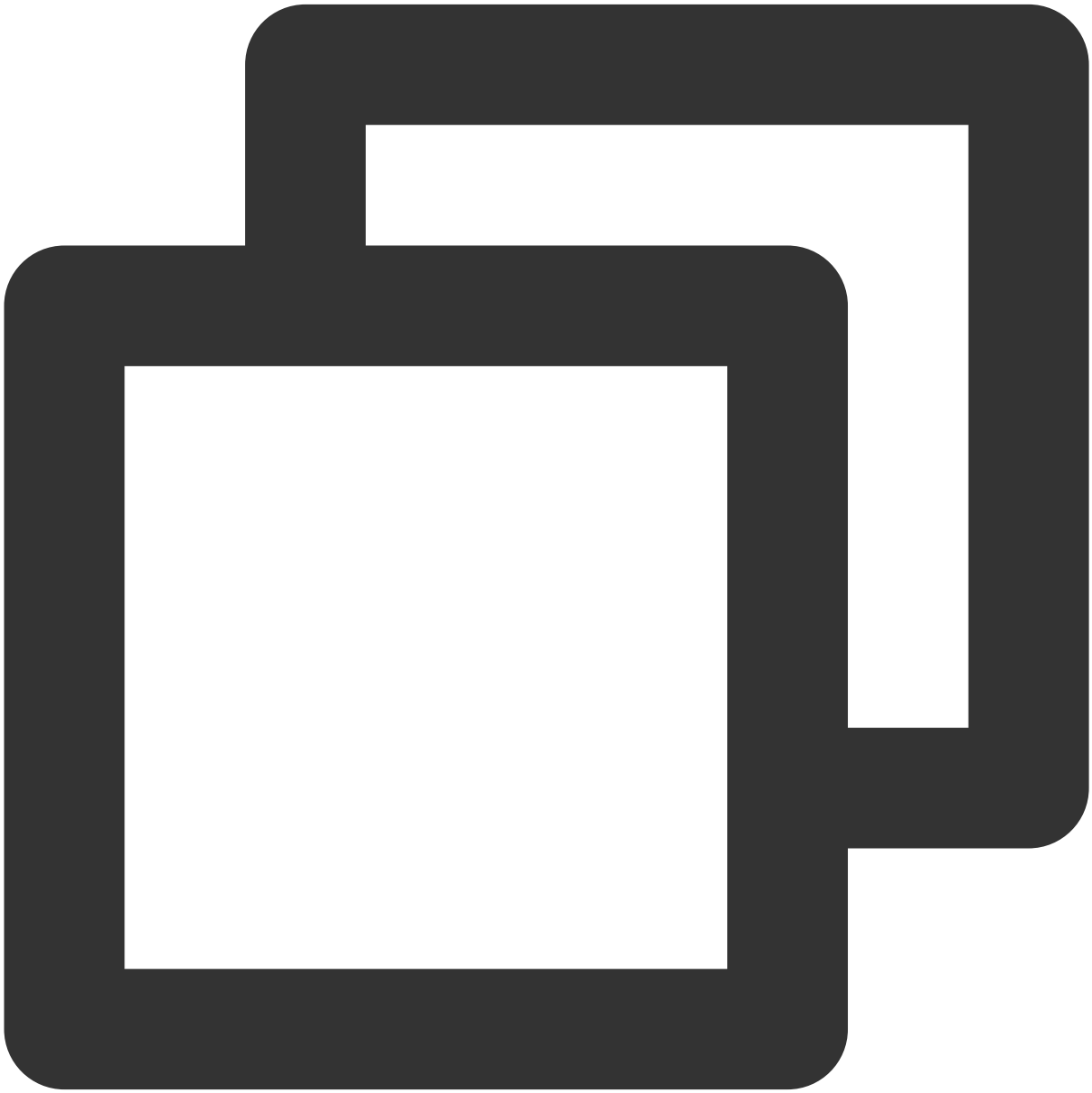
响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>c93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>PicProcess</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/out.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
            <TemplateName>test</TemplateName>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

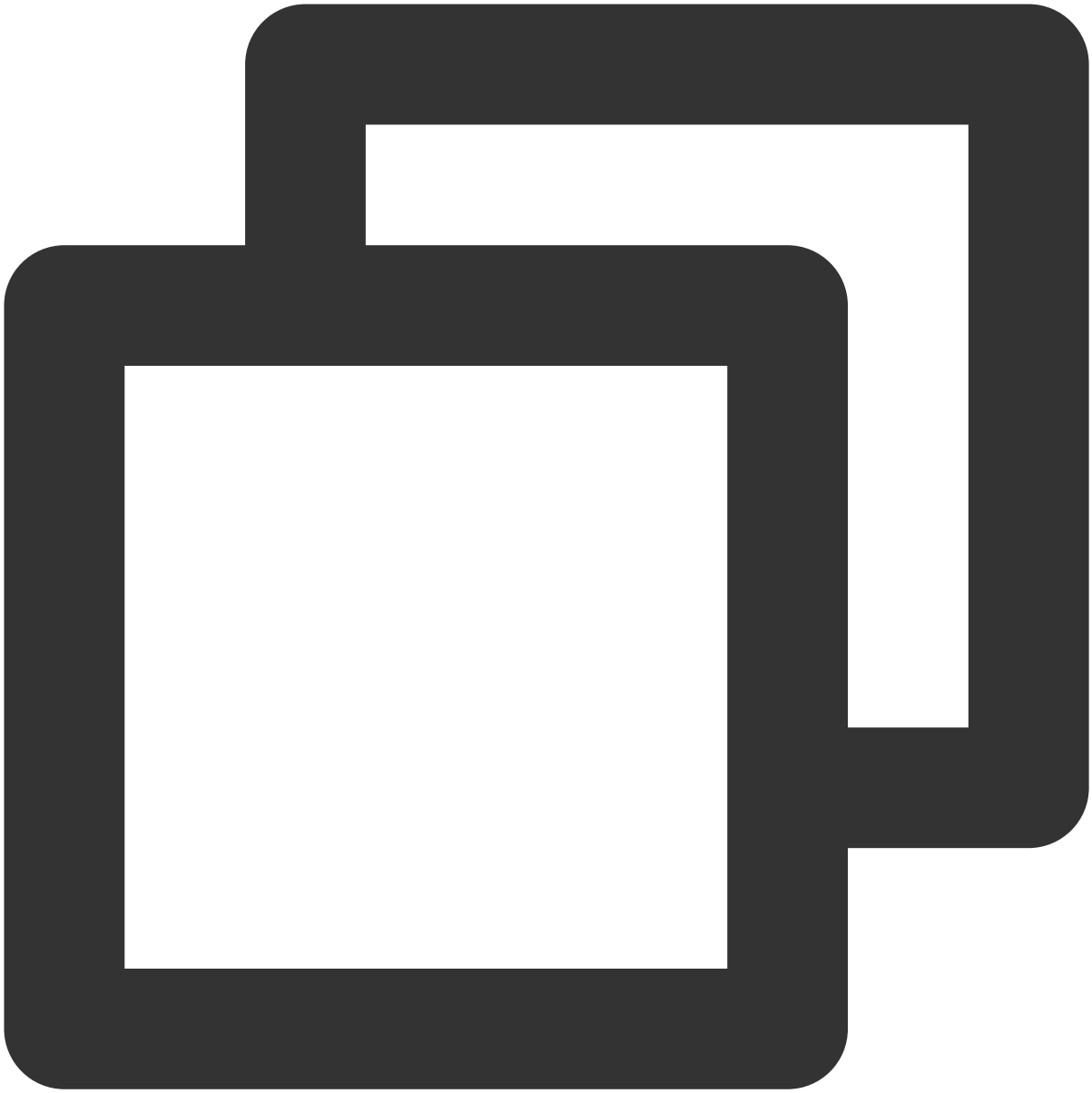
请求2：使用图片处理处理参数



```
POST /pic_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>PicProcess</Tag>
    <Input>
        <Object>input/deer.jpg</Object>
    <Operation>
```

```
        <PicProcess>
            <IsPicInfo>true</IsPicInfo>
            <ProcessRule>imageMogr2/rotate/90</ProcessRule>
        </PicProcess>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.jpg</Object>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>c93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>PicProcess</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/out.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <PicProcess>
                <IsPicInfo>true</IsPicInfo>
                <ProcessRule>imageMogr2/rotate/90</ProcessRule>
            </PicProcess>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

取消图片处理任务
CancelPicJob

最近更新时间：2023-12-30 16:42:41

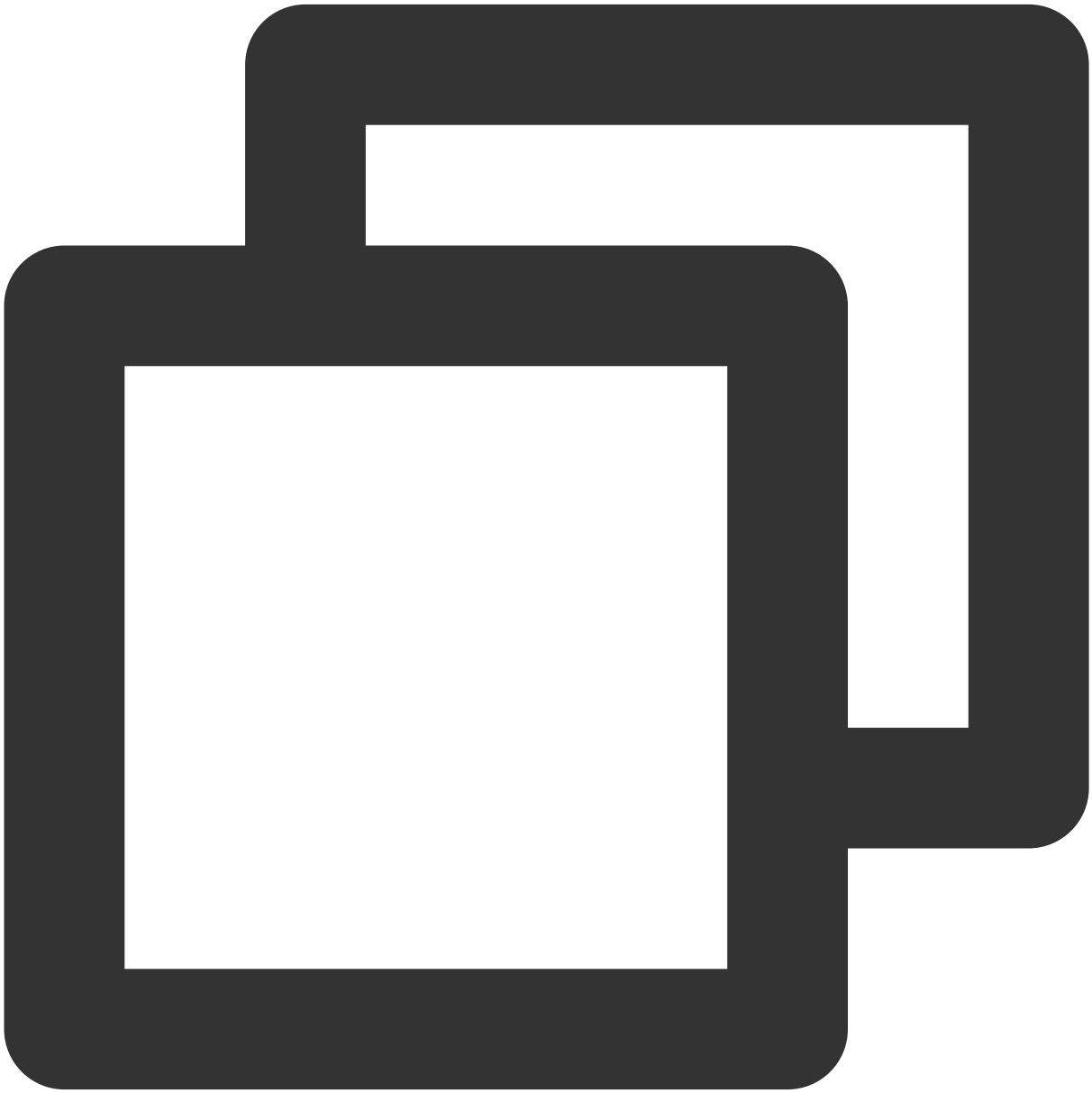
功能描述

CancelPicJob 用于取消一个任务。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
PUT /jobs/<jobId>?cancel HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

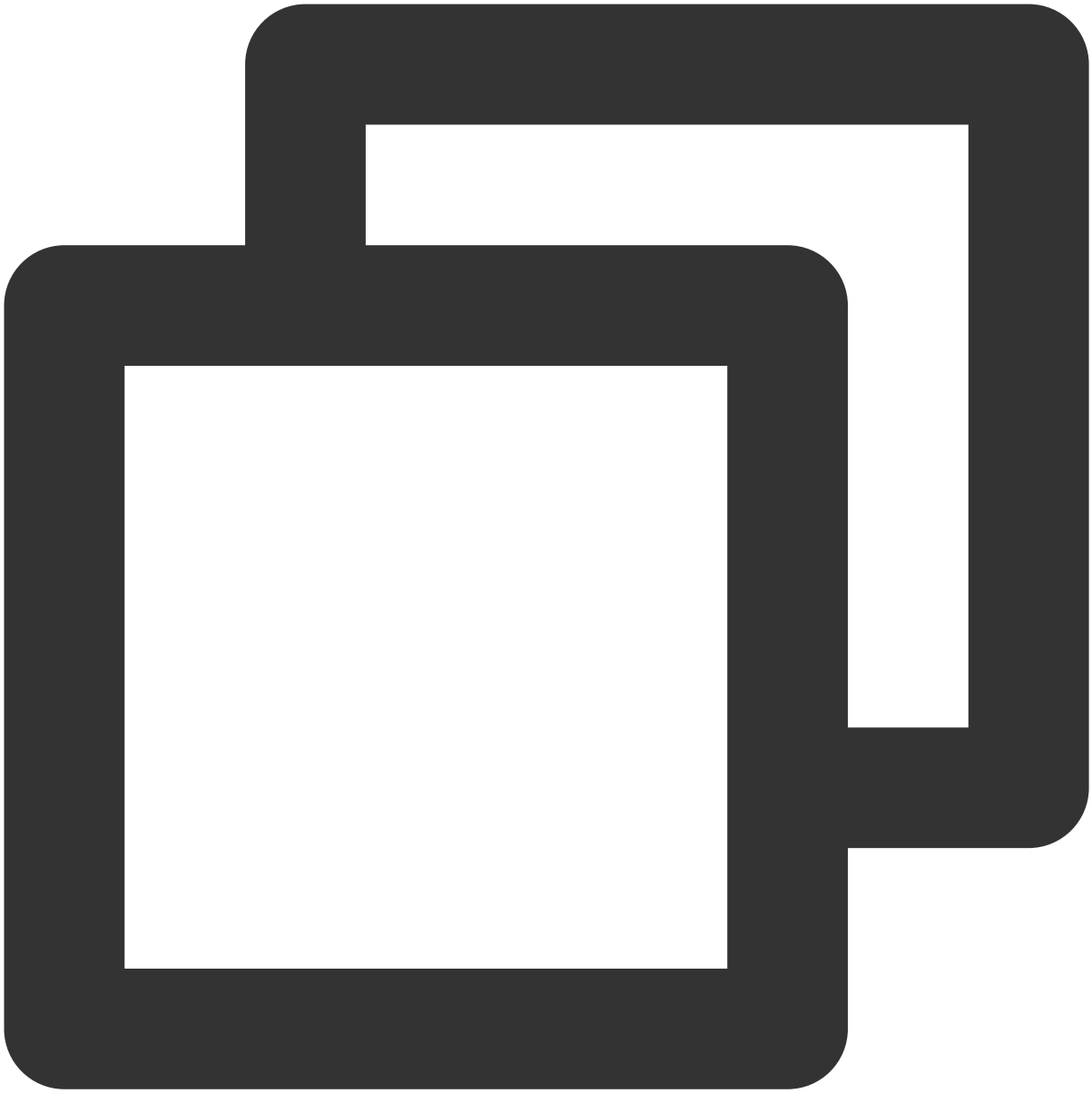
该请求无响应体。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

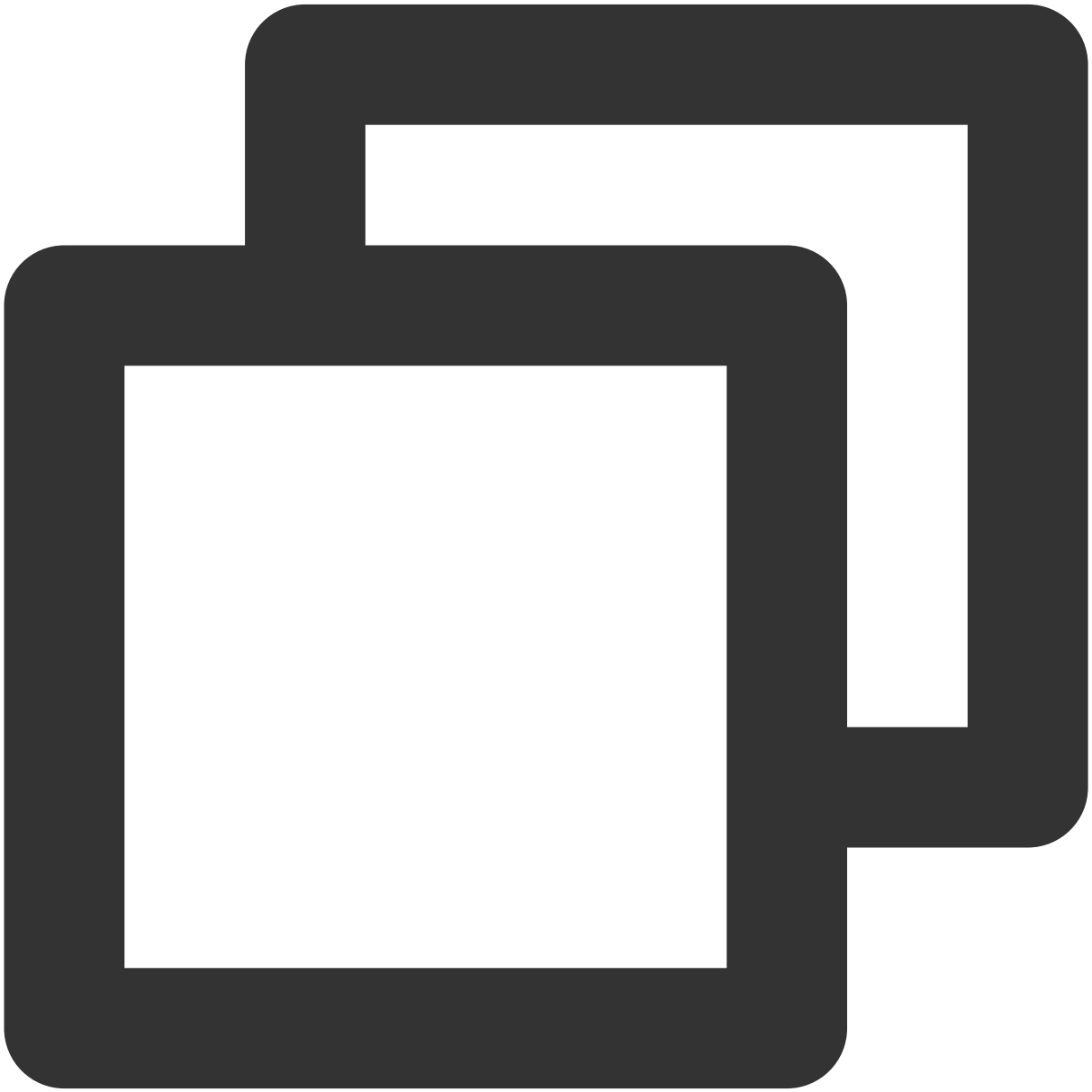
实际案例

请求



```
PUT /jobs/j-xxx-xxx-xxx-xxxx?cancel HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=
```

查询图片处理任务

DescribePicJob

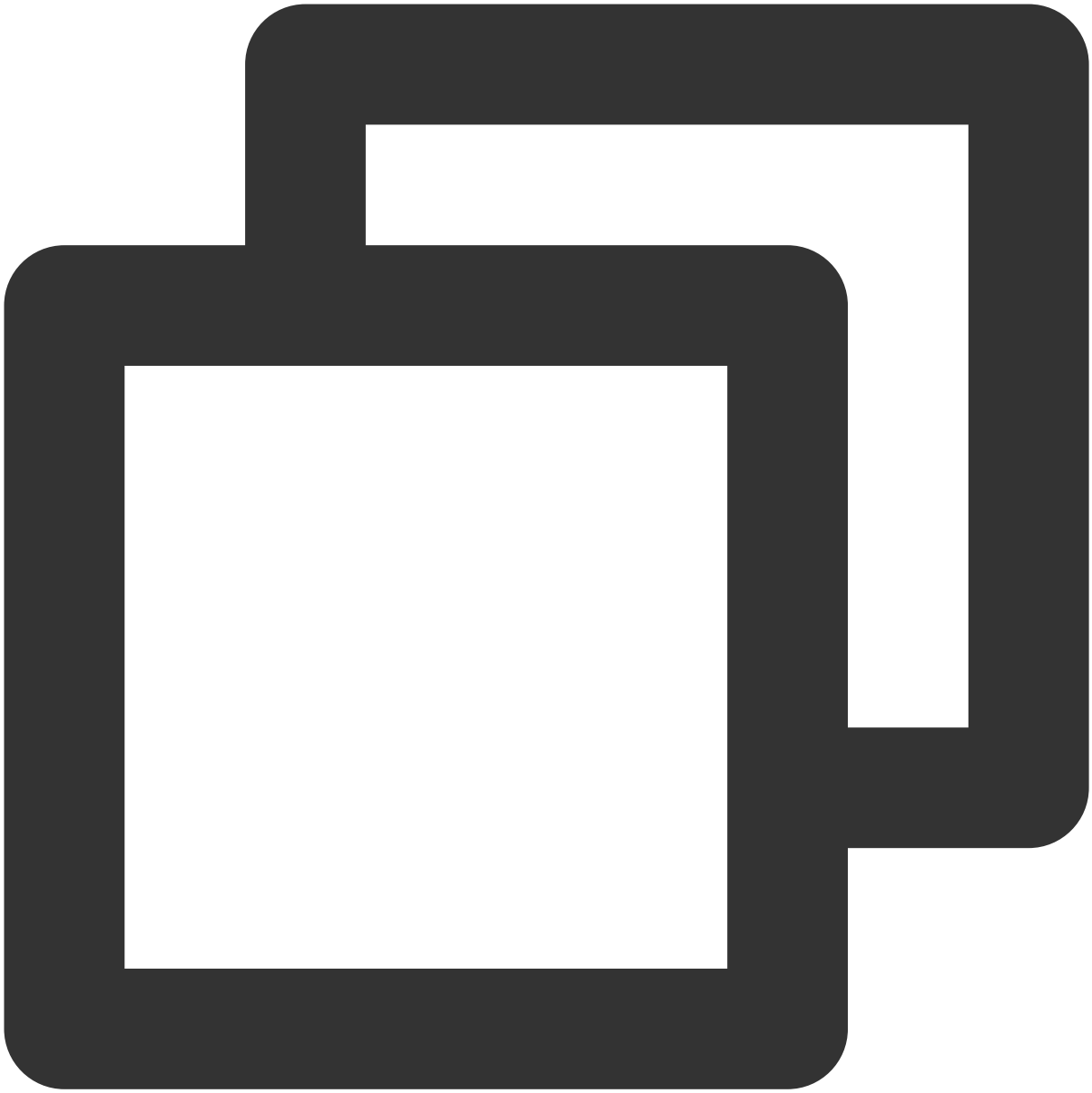
最近更新时间：2023-12-30 16:43:16

功能描述

DescribePicJob 用于查询指定的任务。

请求

请求示例



```
GET /pic_jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

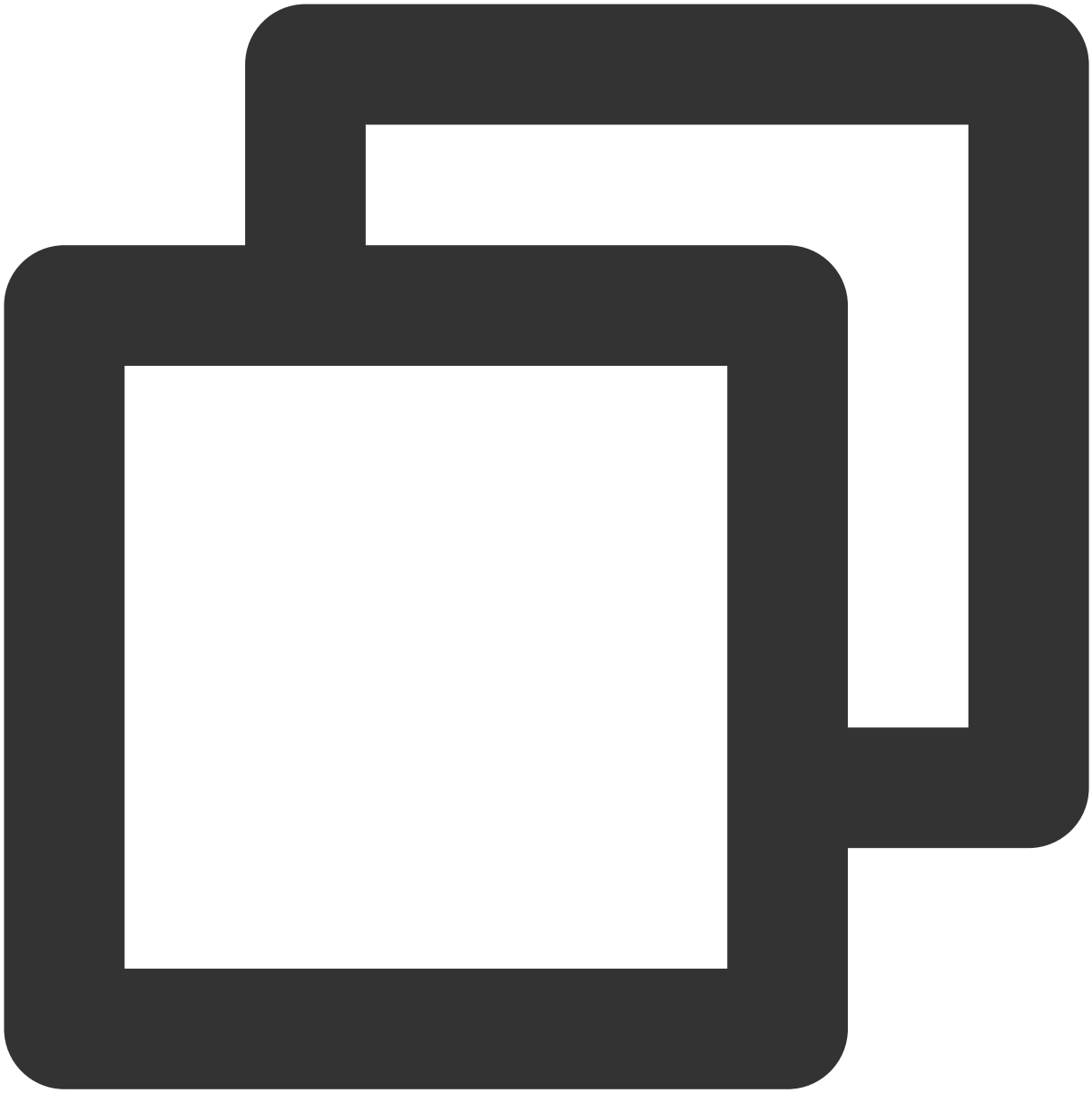
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
  </JobsDetail>
  <NonExistJobIds></NonExistJobIds>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 Response 的内容：

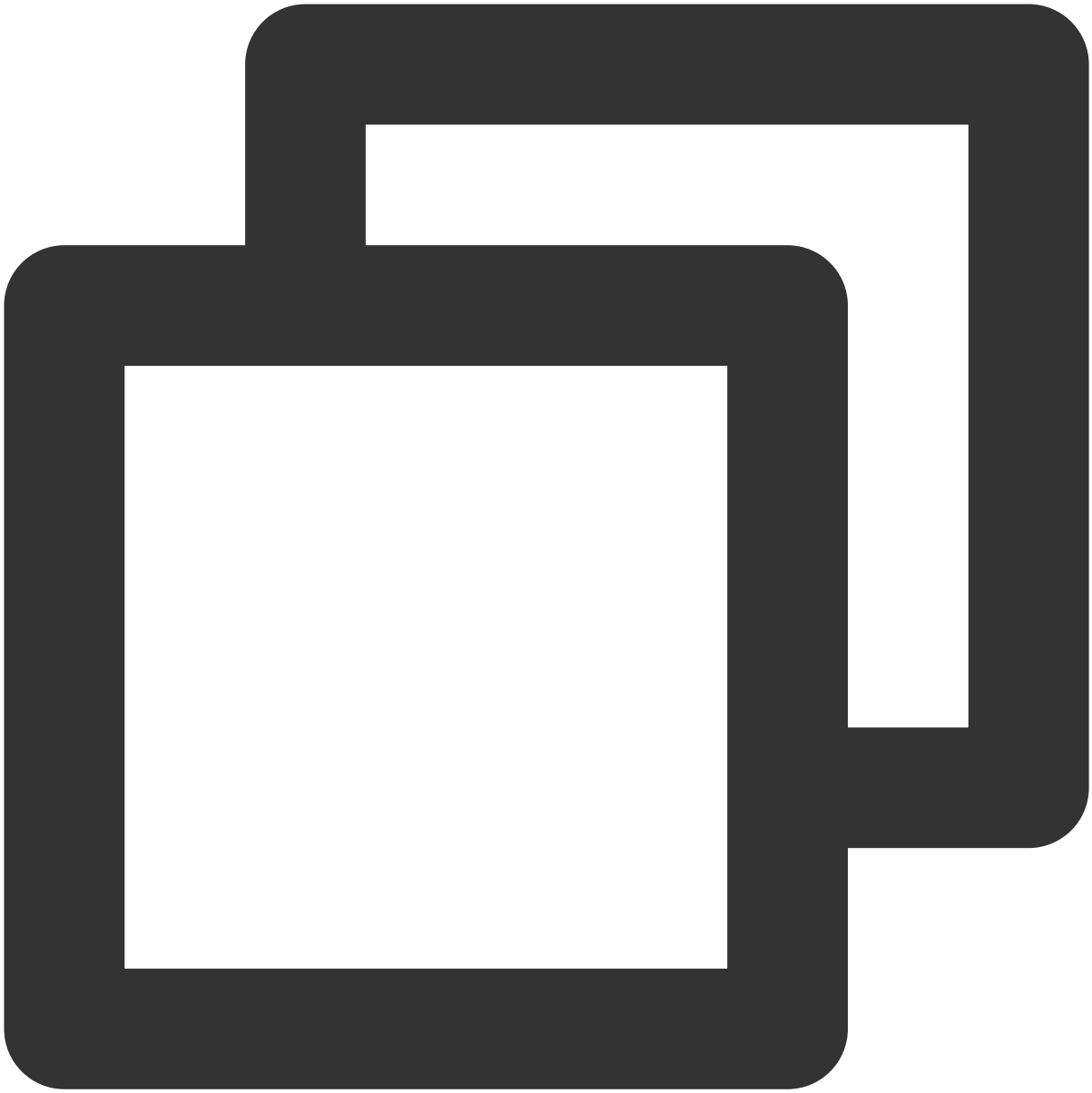
节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息，同 CreateMediaJobs 接口的 Response.JobsDetail 节点	Container
NonExistJobIds	Response	查询的 ID 中不存在的任务，所有任务都存在时不返回	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

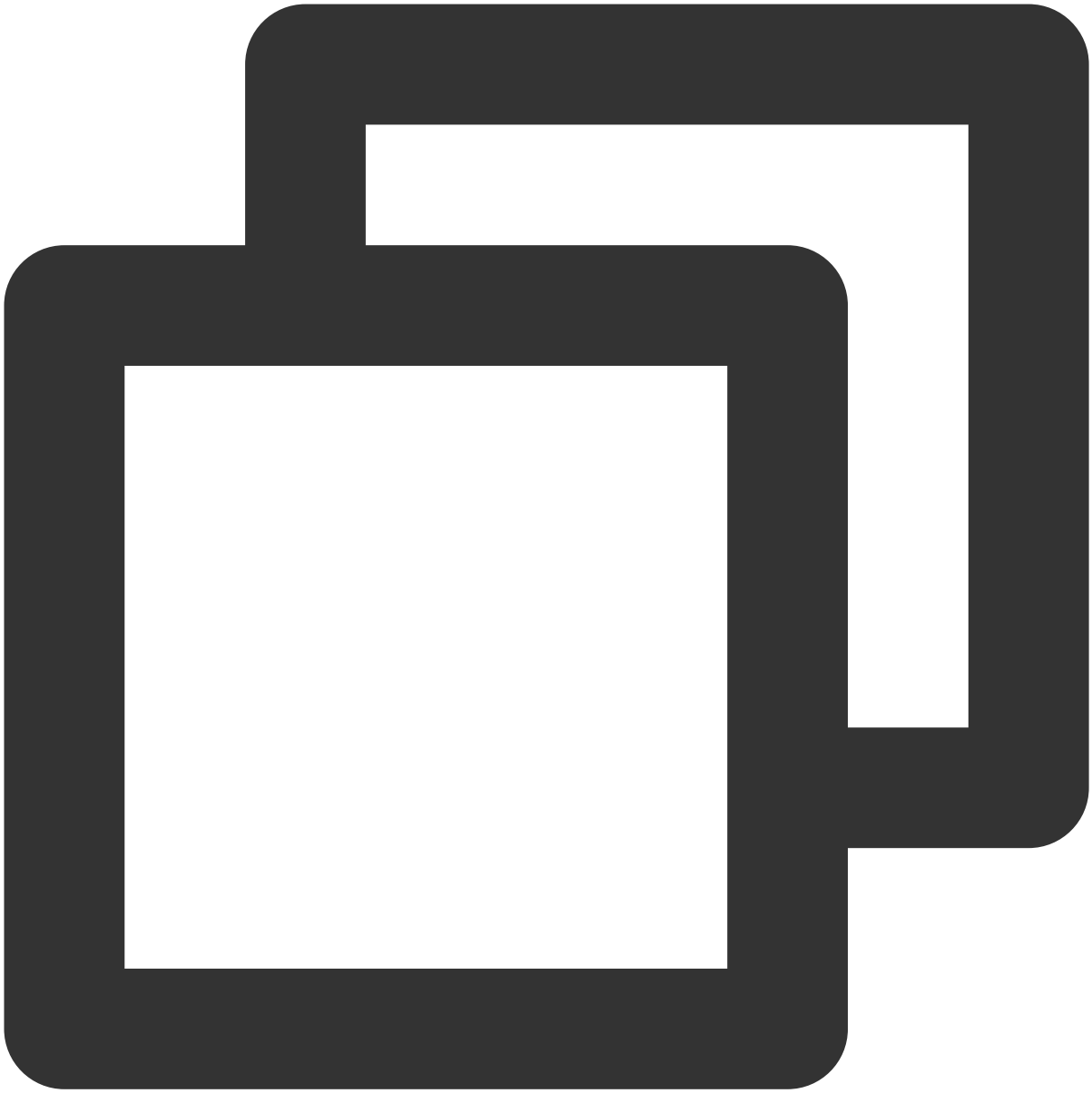
实际案例

请求



```
GET /pic_jobs/jabcsdssfeipplsdfwe HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
  <JobsDetail>
    <Code>Success</Code>
```

```
    <Message>Success</Message>
    <JobId>je8f65004eb8511eaaed4f377124a303c</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <EndTime>2019-07-07T12:12:12+0800</EndTime>
    <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
    <Tag>PicProcess</Tag>
    <Input>
      <Object>test.jpg</Object>
    </Input>
    <Operation>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <Output>
        <Region>ap-beijing</Region>
        <Bucket>abc-1250000000</Bucket>
        <Object>picprocess.jpg</Object>
        <SpriteObject></SpriteObject>
      </Output>
    </Operation>
  </JobsDetail>
</Response>
```

DescribePicJobs

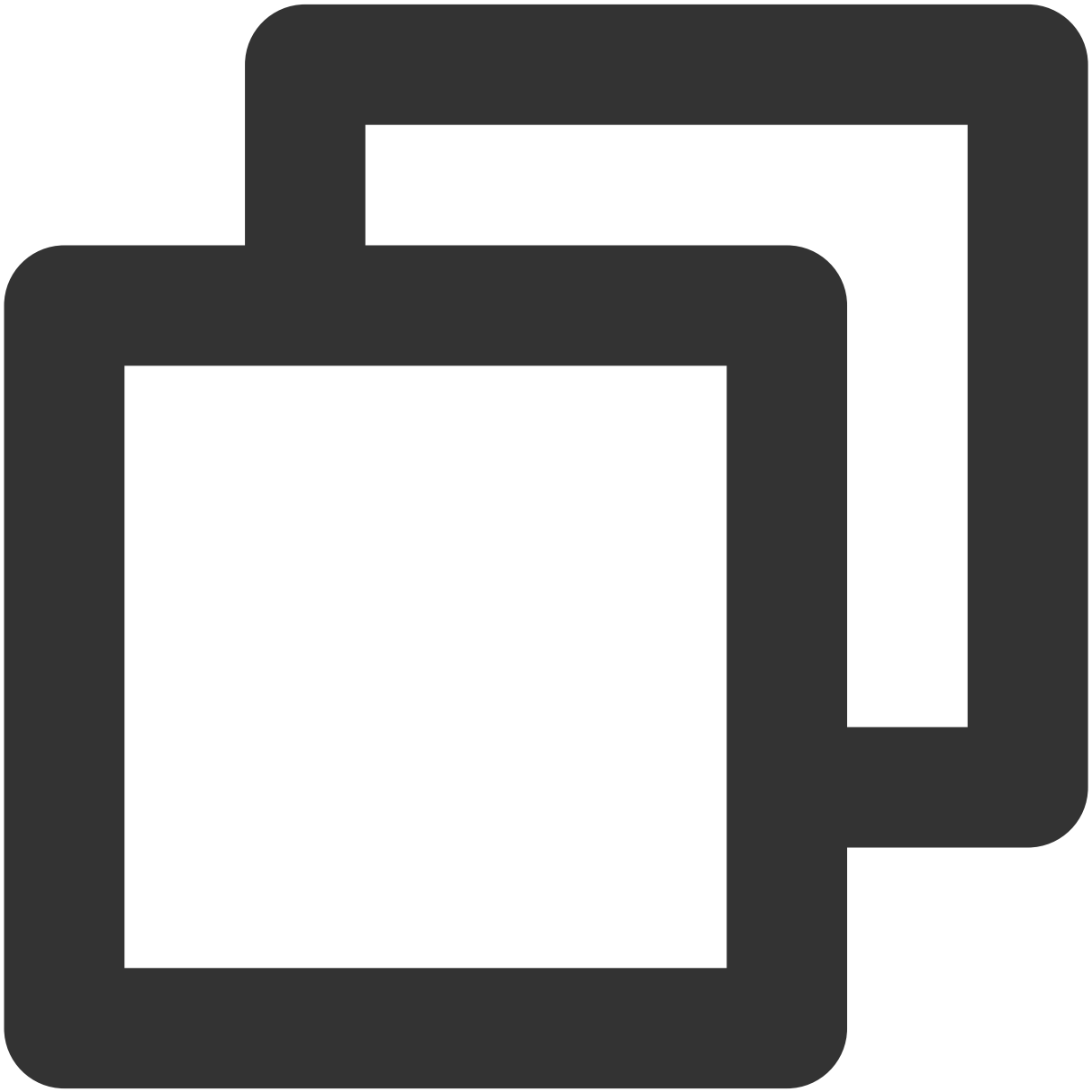
最近更新时间：2023-12-30 16:43:22

功能描述

DescribePicJobs 用于拉取符合条件的任务。

请求

请求示例



```
GET /pic_jobs?size=&states=&queueId=&startCreationTime=&endCreationTime= HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	必选
queueId	无	拉取该队列 ID 下的任务	String	是
tag	无	任务的 Tag：PicProcess	String	是
orderByTime	无	Desc 或者 Asc。默认为 Desc	String	否
nextToken	无	请求的上下文，用于翻页。上次返回的值	String	否
size	无	拉取的最大任务数。默认为10。最大为100	Integer	否
states	无	拉取该状态的任务，以,分割支持多状态 All，Submitted，Running，Success，Failed，Pause，Cancel。默认为 All	String	否
startCreationTime	无	拉取创建时间大于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否
endCreationTime	无	拉取创建时间小于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否

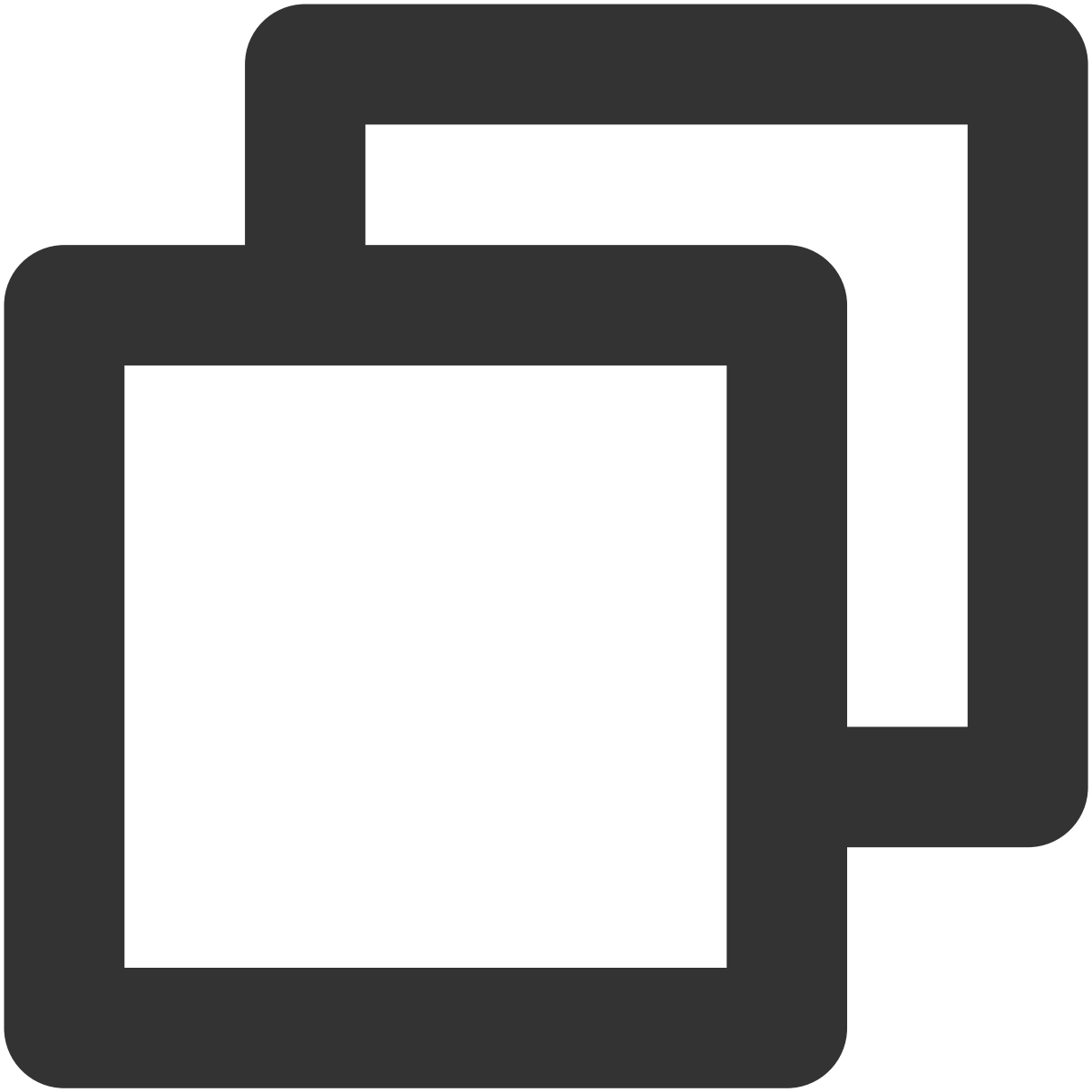
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
  </JobsDetail>
  <NextToken></NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

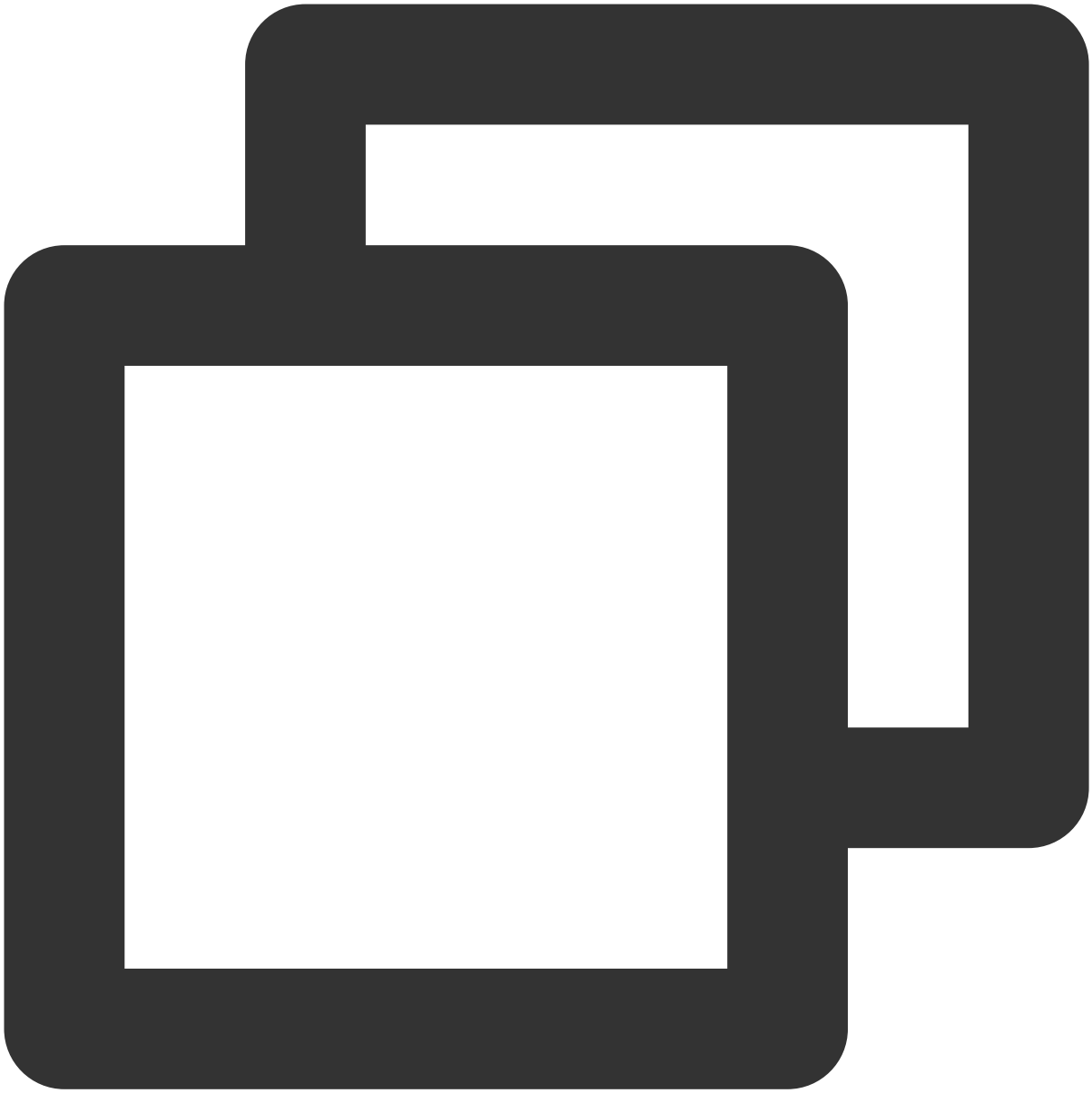
节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息，同 CreateMediaJobs 接口中的 Response.JobsDetail 节点	Container
NextToken	Response	翻页的上下文 Token	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

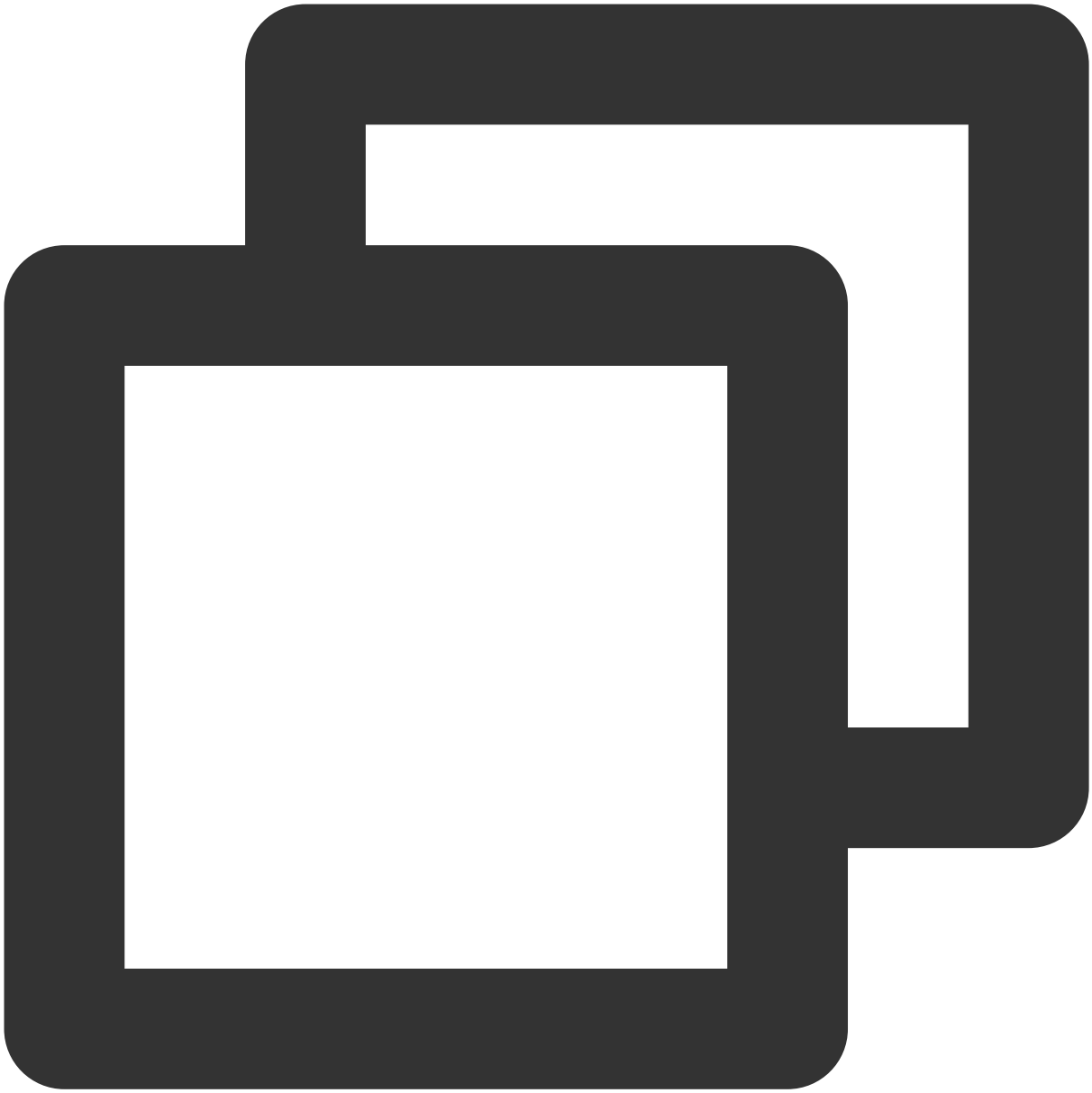
实际案例

请求



```
GET /pic_jobs?queueId=aaaaaaaaaaa&tag=PicProcess HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
  <JobsDetail>
    <Code>Success</Code>
```

```
    <Message>Success</Message>
    <JobId>je8f65004eb8511eaaed4f377124a303c</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <EndTime>2019-07-07T12:12:12+0800</EndTime>
    <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
    <Tag>PicProcess</Tag>
    <Input>
      <Object>test.jpg</Object>
    </Input>
    <Operation>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <Output>
        <Region>ap-beijing</Region>
        <Bucket>abc-1250000000</Bucket>
        <Object>picprocess.jpg</Object>
        <SpriteObject></SpriteObject>
      </Output>
    </Operation>
  </JobsDetail>
  <JobsDetail>
    <Code>Success</Code>
    <Message>Success</Message>
    <JobId>jbbbbccddabcddafesff</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <EndTime>2019-07-07T12:12:12+0800</EndTime>
    <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
    <Tag>PicProcess</Tag>
    <Input>
      <Object>test.jpg</Object>
    </Input>
    <Operation>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <Output>
        <Region>ap-beijing</Region>
        <Bucket>abc-1250000000</Bucket>
        <Object>picprocess.jpg</Object>
        <SpriteObject></SpriteObject>
      </Output>
    </Operation>
  </JobsDetail>
</Response>
```

图片处理回调

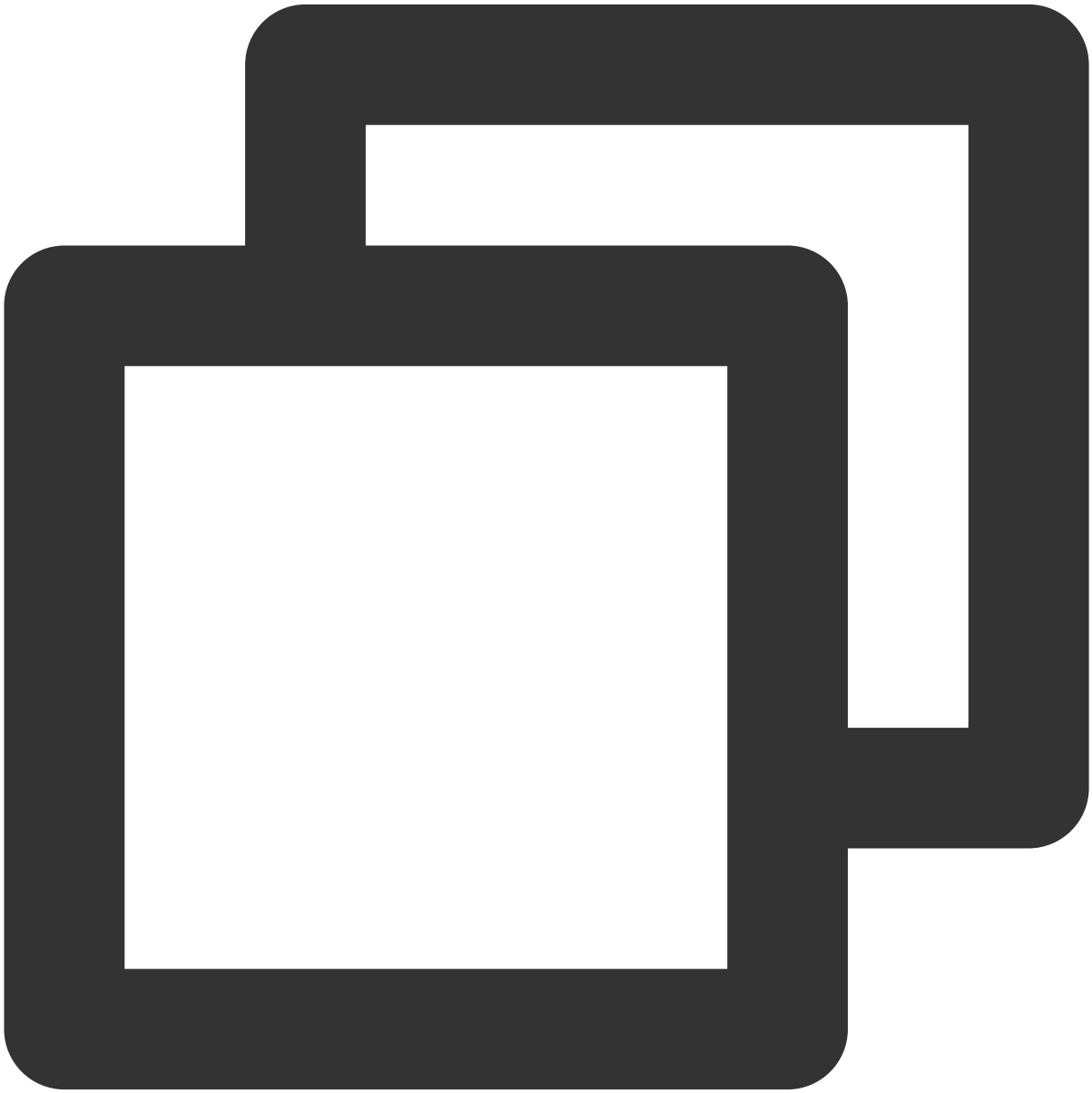
最近更新时间：2023-12-30 16:43:50

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>2022-07-18T15:16:44+0800</EndTime>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>c93984788066911ed89ed352d4d9d2084</JobId>
        <Message/>
        <Operation>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/out.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <PicProcessResult>
                <ObjectName>output/out.jpg</ObjectName>
                <OriginalInfo>
                    <Etag>"d5a491cd3d8d071e3212c3478e8e35a1"</Etag>
                    <ImageInfo>
                        <Ave>0xc64847</Ave>
                        <Format>JPEG</Format>
                        <Height>417</Height>
                        <Orientation>0</Orientation>
                        <Quality>68</Quality>
                        <Width>500</Width>
                    </ImageInfo>
                </OriginalInfo>
                <ProcessResult>
                    <Etag/>
                    <Format>JPEG</Format>
                    <Height>500</Height>
                    <Quality>68</Quality>
                    <Size>24911</Size>
                    <Width>417</Width>
                </ProcessResult>
            </PicProcessResult>
            <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
            <TemplateName>test</TemplateName>
            <UserData>This is my PicProcess job.</UserData>
        </Operation>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <StartTime>2022-07-18T15:16:44+0800</StartTime>
        <State>Success</State>
        <Tag>PicProcess</Tag>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交图片处理任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

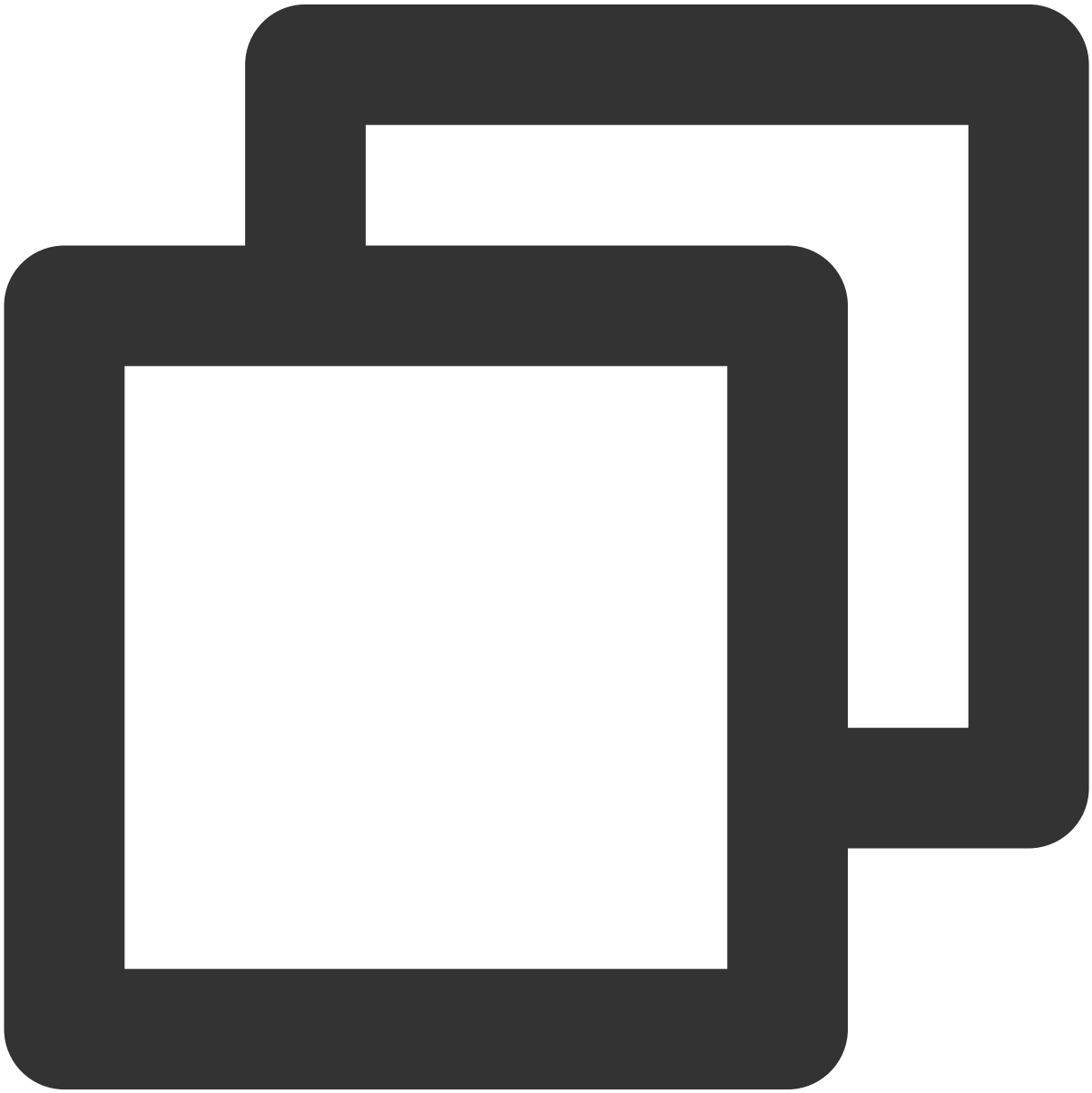
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

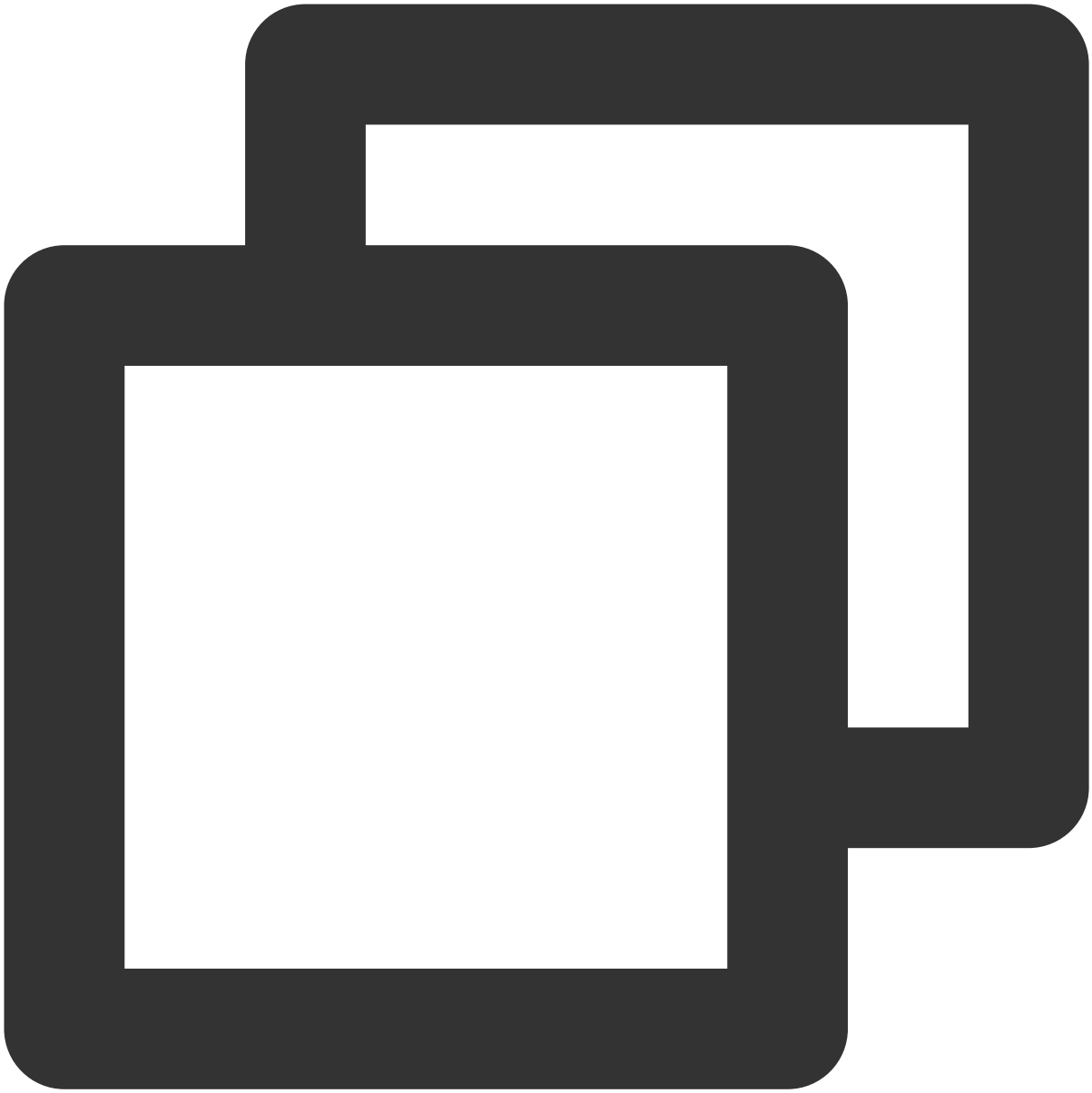
案例 1：通过任务接口触发的任务回调



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>2022-07-18T15:16:44+0800</EndTime>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>c93984788066911ed89ed352d4d9d2084</JobId>
        <Message/>
        <Operation>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/out.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <PicProcessResult>
                <ObjectName>output/out.jpg</ObjectName>
                <OriginalInfo>
                    <Etag>"d5a491cd3d8d071e3212c3478e8e35a1"</Etag>
                    <ImageInfo>
                        <Ave>0xc64847</Ave>
                        <Format>JPEG</Format>
                        <Height>417</Height>
                        <Orientation>0</Orientation>
                        <Quality>68</Quality>
                        <Width>500</Width>
                    </ImageInfo>
                </OriginalInfo>
                <ProcessResult>
                    <Etag/>
                    <Format>JPEG</Format>
                    <Height>500</Height>
                    <Quality>68</Quality>
                    <Size>24911</Size>
                    <Width>417</Width>
                </ProcessResult>
            </PicProcessResult>
            <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
            <TemplateName>test</TemplateName>
            <UserData>This is my PicProcess job.</UserData>
        </Operation>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <StartTime>2022-07-18T15:16:44+0800</StartTime>
        <State>Success</State>
        <Tag>PicProcess</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调

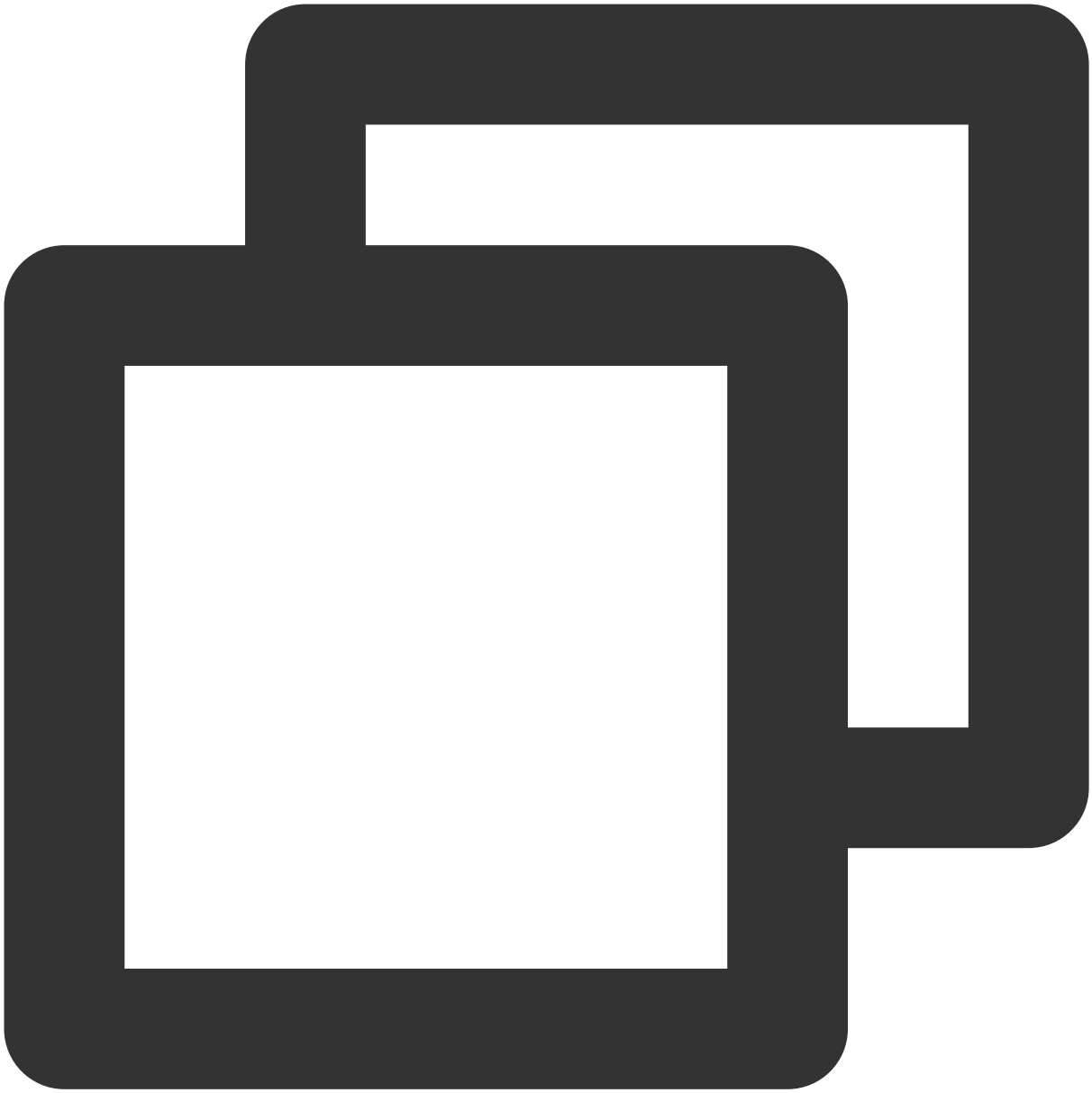


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>2022-07-18T15:16:44+0800</EndTime>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</Object>
            <Region>ap-chongqing</Region>
            <CosHeaders>
```

```
                <Key>Content-Type</Key>
                <Value>image/jpeg</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>EventName</Key>
                <Value>cos:ObjectCreated:Put</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>Size</Key>
                <Value>1024</Value>
            </CosHeaders>
        </Input>
        <JobId>c93984788066911ed89ed352d4d9d2084</JobId>
        <Message/>
        <Operation>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/out.jpg</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <PicProcessResult>
                <ObjectName>output/out.jpg</ObjectName>
                <OriginalInfo>
                    <Etag>"d5a491cd3d8d071e3212c3478e8e35a1"</Etag>
                    <ImageInfo>
                        <Ave>0xc64847</Ave>
                        <Format>JPEG</Format>
                        <Height>417</Height>
                        <Orientation>0</Orientation>
                        <Quality>68</Quality>
                        <Width>500</Width>
                    </ImageInfo>
                </OriginalInfo>
                <ProcessResult>
                    <Etag/>
                    <Format>JPEG</Format>
                    <Height>500</Height>
                    <Quality>68</Quality>
                    <Size>24911</Size>
                    <Width>417</Width>
                </ProcessResult>
            </PicProcessResult>
            <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
```

```
            <TemplateName>test</TemplateName>
            <UserData>This is my PicProcess job.</UserData>
        </Operation>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <StartTime>2022-07-18T15:16:44+0800</StartTime>
        <State>Success</State>
        <Tag>PicProcess</Tag>
        <Workflow>
            <Name>PicProcess_1581665960537</Name>
            <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
            <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
            <WorkflowName>workflow-test</WorkflowName>
        </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调, 格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-07-18T15:16:43+0800",
        "EndTime": "2022-07-18T15:16:44+0800",
        "Input": {
            "BucketId": "test-1234567890",
            "Object": "input/deer.jpg",
            "Region": "ap-chongqing",
            "CosHeaders": [{
```

```
                    "Key": "Content-Type",
                    "Value": "image/jpeg"
                },
                {
                    "Key": "x-cos-request-id",
                    "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
                },
                {
                    "Key": "EventName",
                    "Value": "cos:ObjectCreated:Put"
                },
                {
                    "Key": "Size",
                    "Value": "1024"
                }
            ]
        },
        "JobId": "c93984788066911ed89ed352d4d9d2084",
        "Operation": {
            "Output": {
                "Bucket": "test-1234567890",
                "Object": "output/out.jpg",
                "Region": "ap-chongqing"
            },
            "PicProcessResult": {
                "ObjectName": "output/out.jpg",
                "OriginalInfo": {
                    "Etag": "\\"d5a491cd3d8d071e3212c3478e8e35a1\\"",
                    "ImageInfo": {
                        "Ave": "0xc64847",
                        "Format": "JPEG",
                        "Height": "417",
                        "Orientation": "0",
                        "Quality": "68",
                        "Width": "500"
                    }
                },
                "ProcessResult": {
                    "Format": "JPEG",
                    "Height": "500",
                    "Quality": "68",
                    "Size": "24911",
                    "Width": "417"
                }
            },
            "TemplateId": "t10461fe2bd5a649db9022452ec71e0381",
            "TemplateName": "test",
```

```
            "UserData": "This is my PicProcess job."
        },
        "QueueId": "p2911917386e148639319e13c285cc774",
        "StartTime": "2022-07-18T15:16:44+0800",
        "State": "Success",
        "Tag": "PicProcess",
        "Workflow": {
            "Name": "PicProcess_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }
}
```

多任务处理
提交多任务处理
MultiJobs

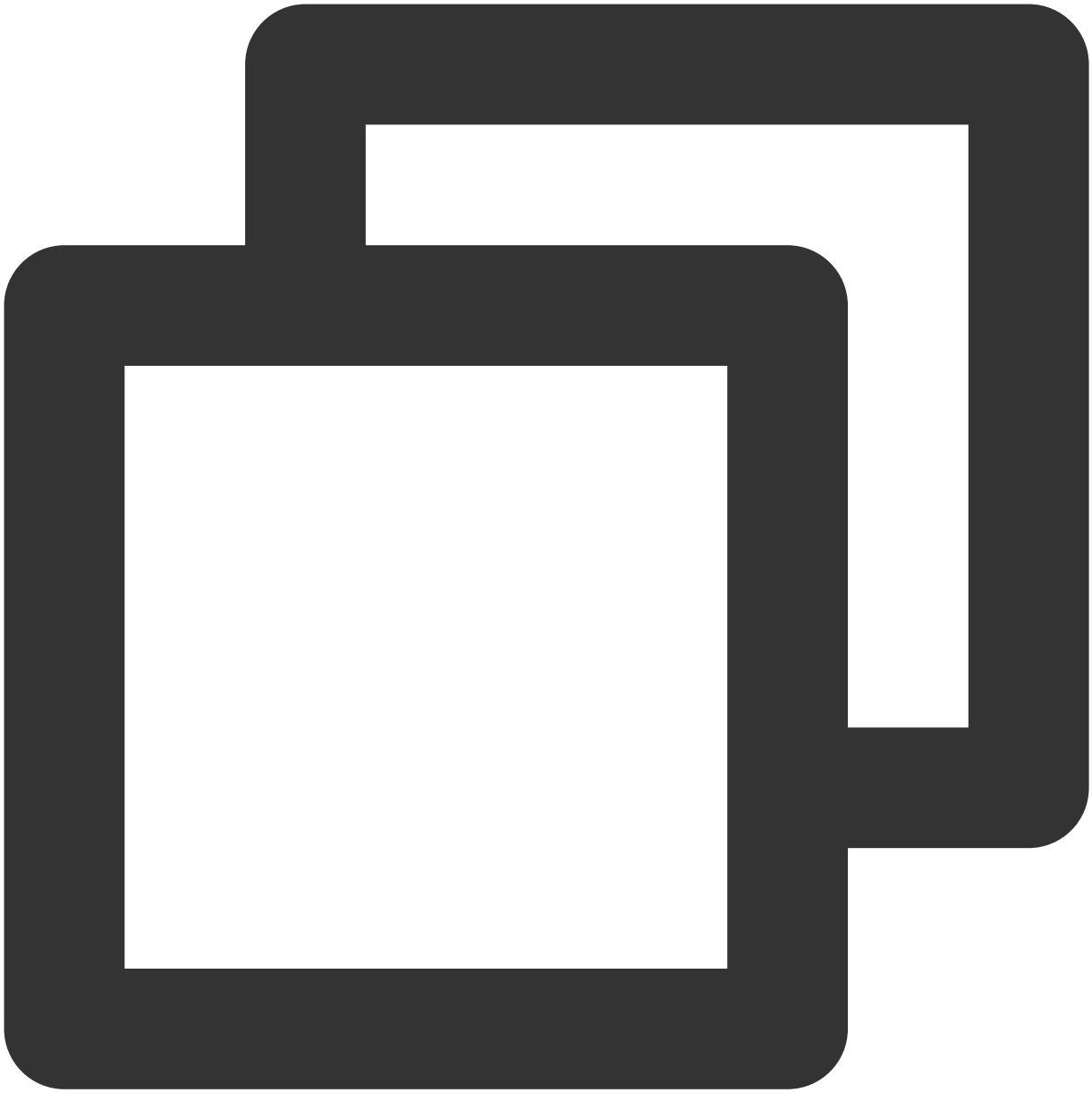
最近更新时间：2023-12-30 16:45:23

功能描述

提交多个任务。

请求

请求示例



```
POST /jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

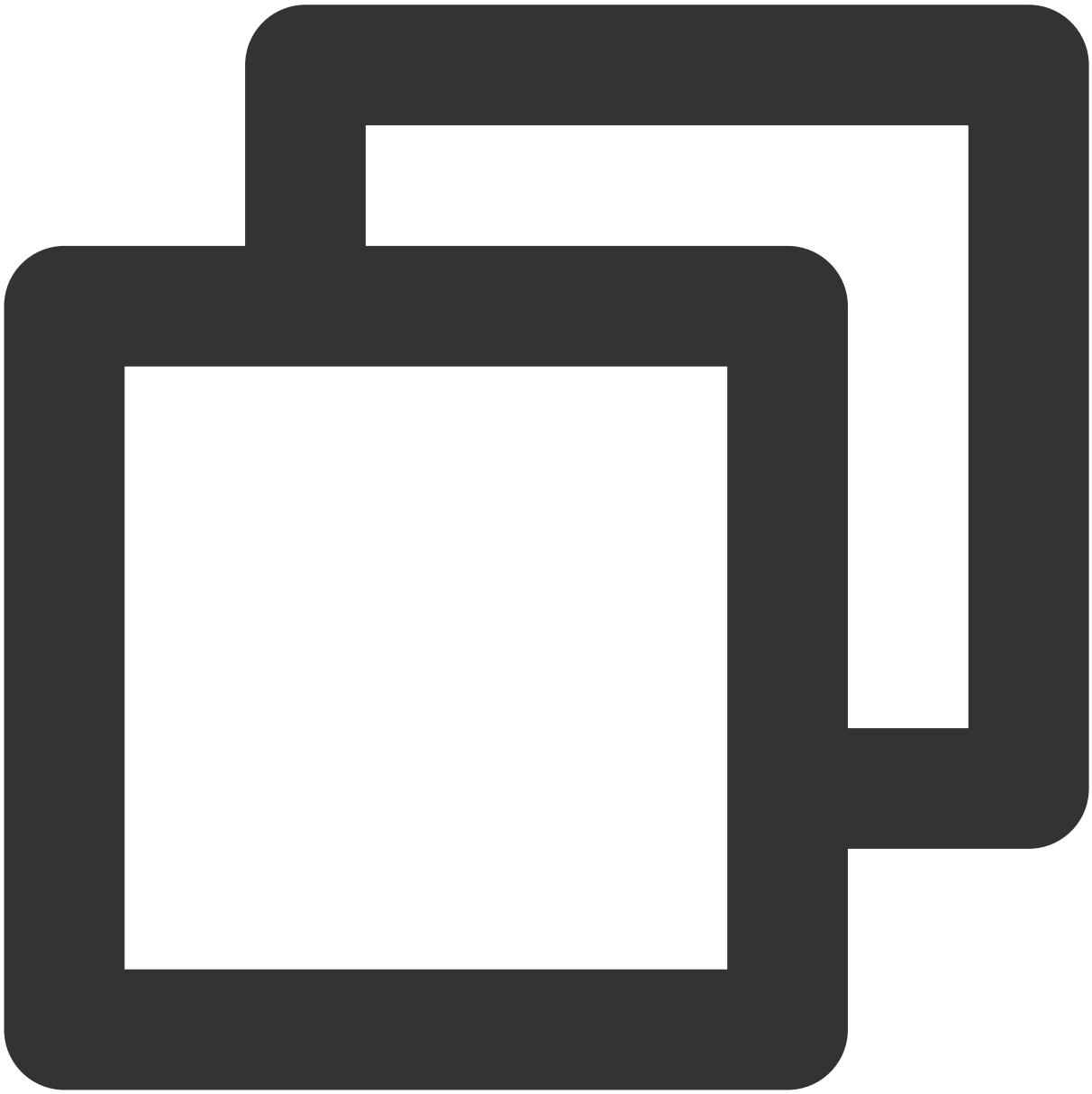
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <Tag>Animation</Tag>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/animation.gif</Object>
        </Output>
        <UserData>This is my Animation job.</UserData>
    </Operation>
    <Operation>
        <Tag>Transcode</Tag>
        <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/transcode.mp4</Object>
        </Output>
        <UserData>This is my Trancode job.</UserData>
    </Operation>
    <Operation>
        <Tag>SmartCover</Tag>
        <SmartCover>
            <Format>jpg</Format>
            <Width>1280</Width>
            <Height>960</Height>
            <Count>5</Count>
            <DeleteDuplicates>true</DeleteDuplicates>
        </SmartCover>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/smartcover-${Number}.jpg</Object>
        </Output>
        <UserData>This is my SmartCover job.</UserData>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键	父节点	描述	类型	是否必选

字）				
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	待操作的媒体信息	Container	是
Operation	Request	操作规则，个数不超过6个	Container	是
QueueId	Request	任务所在的队列 ID	String	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	媒体文件名	String	是

对于不同的任务类型，Operation 的内容不同，请参照以下链接：

音视频转码
视频转动图
视频截帧
智能封面
音视频拼接
人声分离
精彩集锦
SDR to HDR
视频增强
超分辨率
音视频转封装
数字水印
提取数字水印
视频标签
获取媒体信息
音视频流分离

版权所有：腾讯云计算（北京）有限责任公司

视频质量分析
语音合成
音频降噪

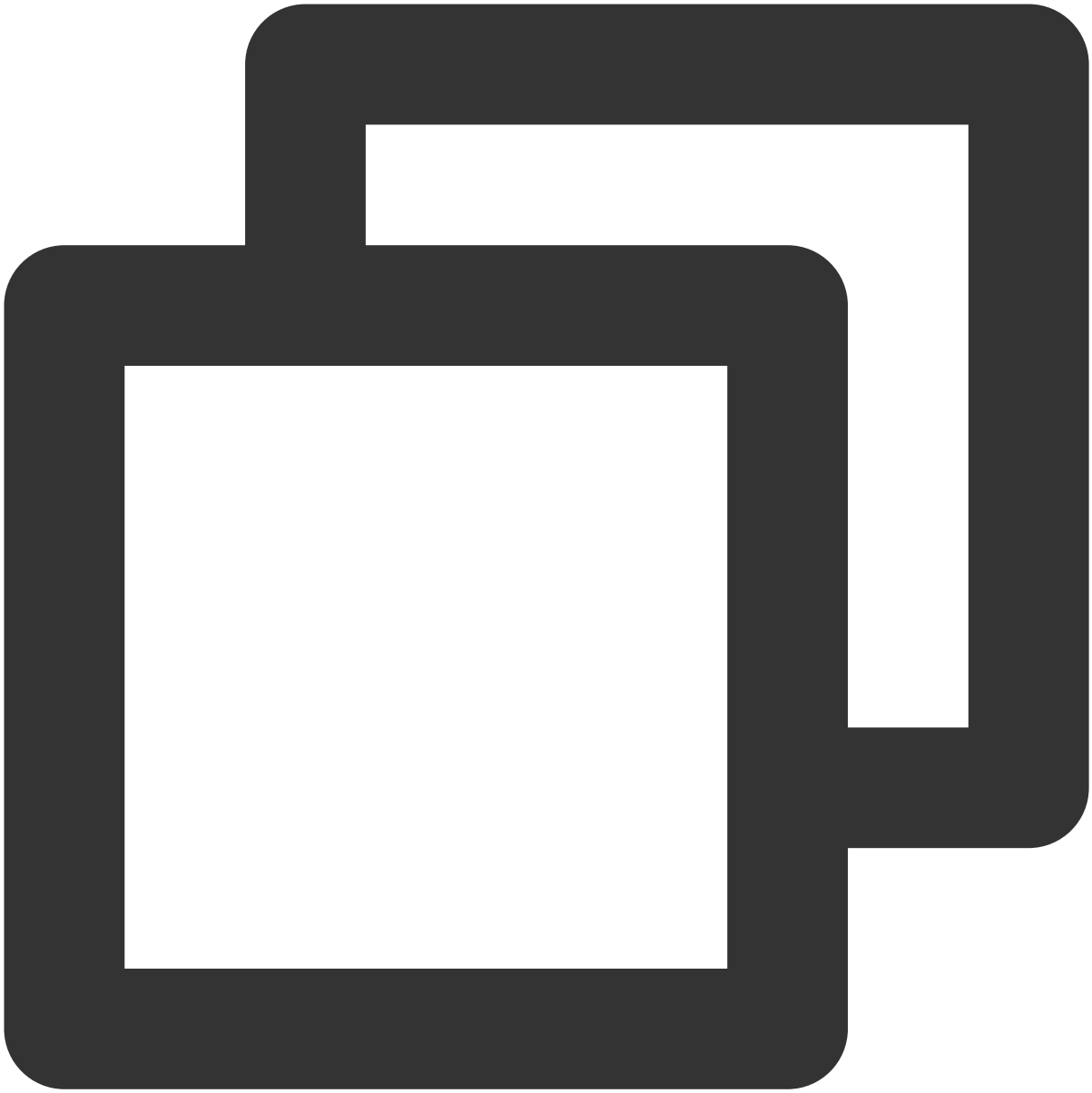
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Input>
```

```
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Tag>Animation</Tag>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>animation_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/animation.gif</Object>
            </Output>
            <UserData>This is my Animation job.</UserData>
        </Operation>
    </JobsDetail>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>j68427030f84611ecb8546d80f2baf56f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Tag>Transcode</Tag>
            <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
            <TemplateName>transcode_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/transcode.mp4</Object>
            </Output>
            <UserData>This is my Trancode job.</UserData>
        </Operation>
    </JobsDetail>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
        <State>Submitted</State>
```

```
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Tag>SmartCover</Tag>
            <SmartCover>
                <Format>jpg</Format>
                <Width>1280</Width>
                <Height>960</Height>
                <Count>5</Count>
                <DeleteDuplicates>true</DeleteDuplicates>
            </SmartCover>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</Object>
            </Output>
            <UserData>This is my SmartCover job.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container 数组

对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：

音视频转码

视频转动图

视频截帧

智能封面

版权所有：腾讯云计算（北京）有限责任公司

音视频拼接
人声分离
精彩集锦
SDR to HDR
视频增强
超分辨率
音视频转封装
数字水印
提取数字水印
视频标签
获取媒体信息
音视频流分离
视频质量分析
语音合成
音频降噪

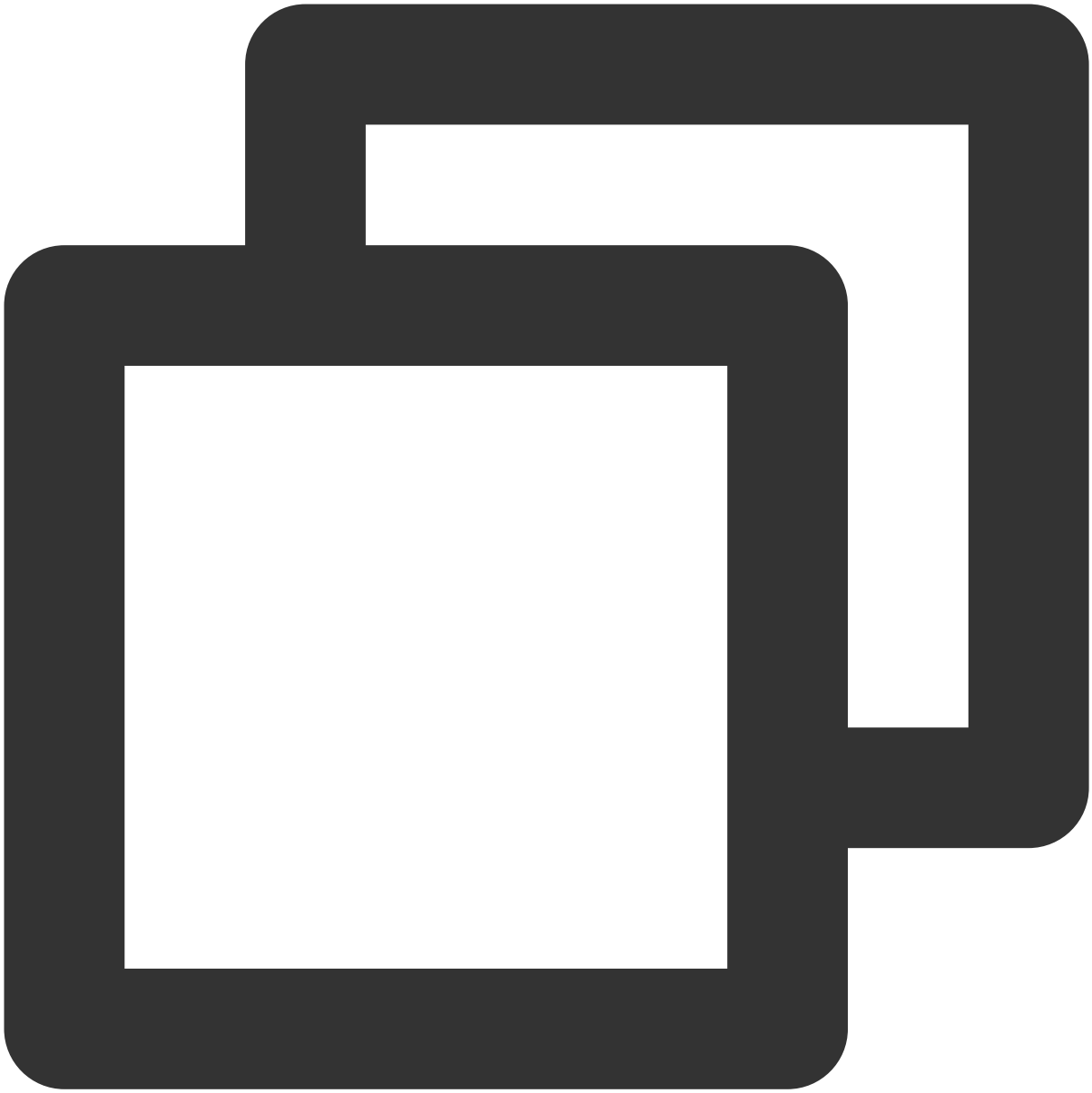
错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

使用多个模板 ID

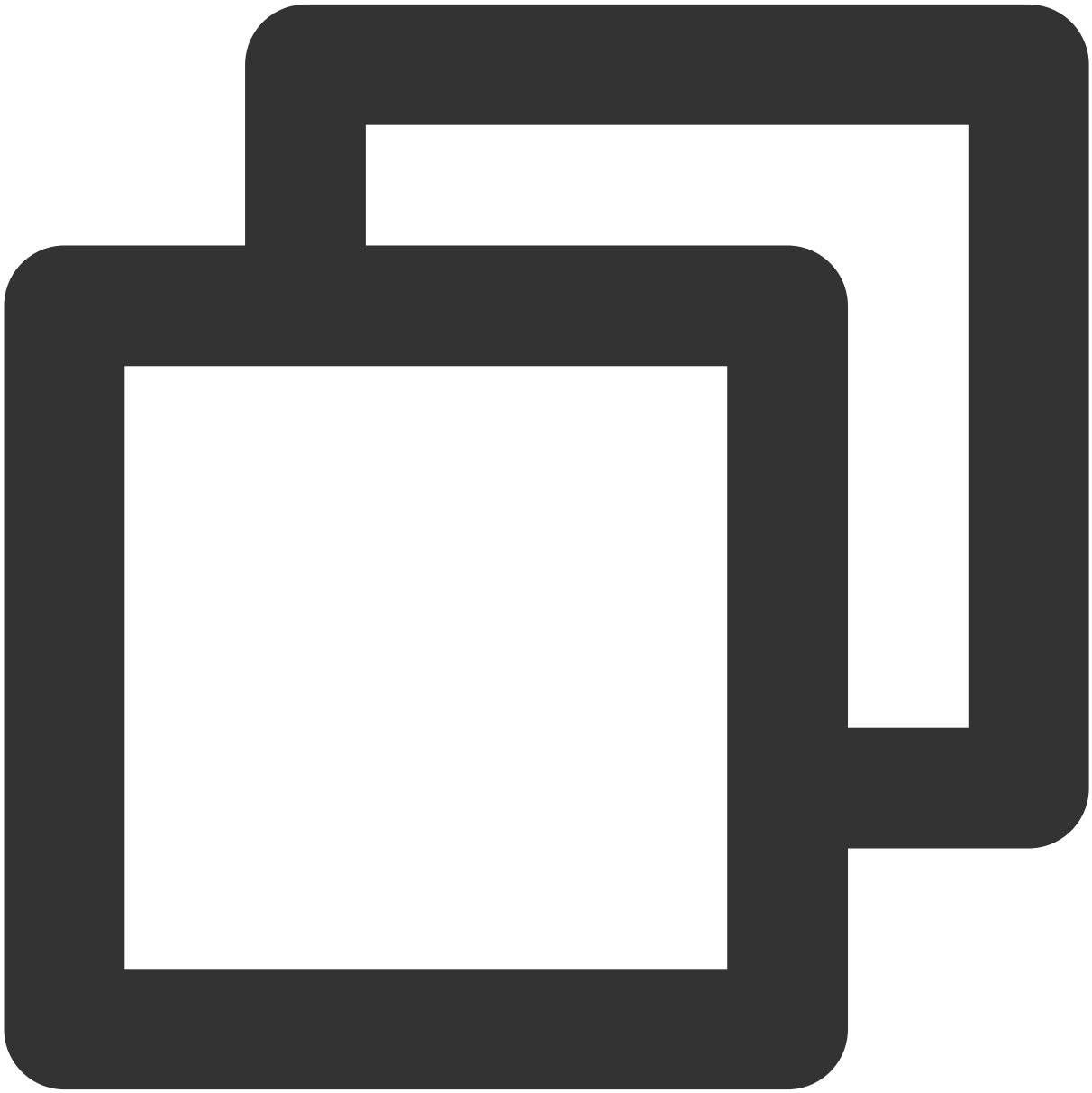
请求



```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
```

```
        <Tag>Animation</Tag>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/animation.gif</Object>
        </Output>
        <UserData>This is my Animation job.</UserData>
    </Operation>
    <Operation>
        <Tag>Transcode</Tag>
        <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/transcode.mp4</Object>
        </Output>
        <UserData>This is my Trancode job.</UserData>
    </Operation>
    <Operation>
        <Tag>SmartCover</Tag>
        <SmartCover>
            <Format>jpg</Format>
            <Width>1280</Width>
            <Height>960</Height>
            <Count>5</Count>
            <DeleteDuplicates>true</DeleteDuplicates>
        </SmartCover>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/smartcover-${Number}.jpg</Object>
        </Output>
        <UserData>This is my SmartCover job.</UserData>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Tag>Animation</Tag>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>animation_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/animation.gif</Object>
            </Output>
            <UserData>This is my Animation job.</UserData>
        </Operation>
    </JobsDetail>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>j68427030f84611ecb8546d80f2baf56f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Tag>Transcode</Tag>
            <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
            <TemplateName>trancode_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/transcode.mp4</Object>
            </Output>
```

```
            <UserData>This is my Trancode job.</UserData>
        </Operation>
    </JobsDetail>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Tag>SmartCover</Tag>
            <SmartCover>
                <Format>jpg</Format>
                <Width>1280</Width>
                <Height>960</Height>
                <Count>5</Count>
                <DeleteDuplicates>true</DeleteDuplicates>
            </SmartCover>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</Object>
            </Output>
            <UserData>This is my SmartCover job.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

AI 内容识别
提交内容识别任务
提交翻译任务

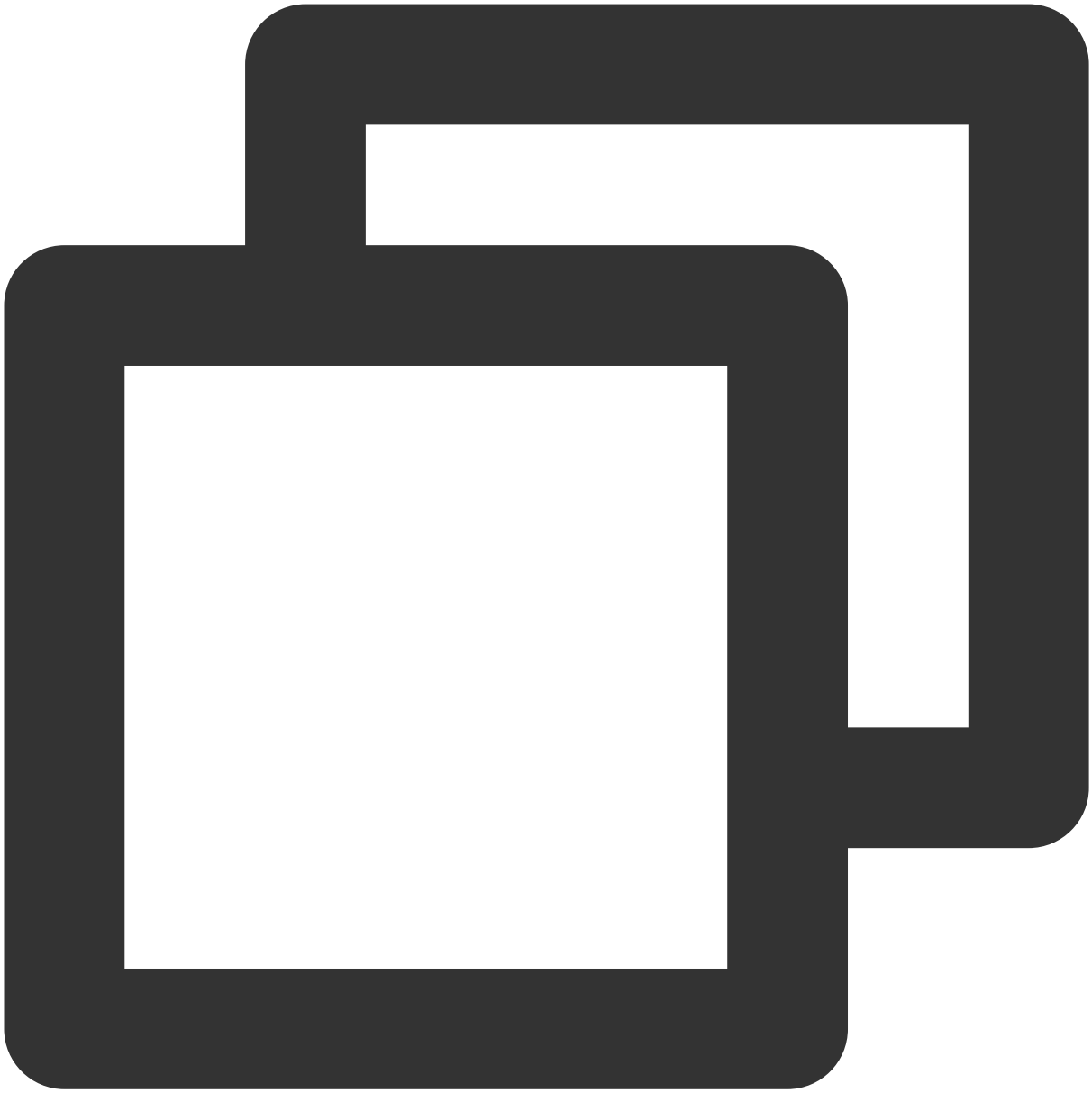
最近更新时间：2023-12-30 16:45:31

功能描述

提交一个翻译任务。

请求

请求示例



```
POST /ai_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

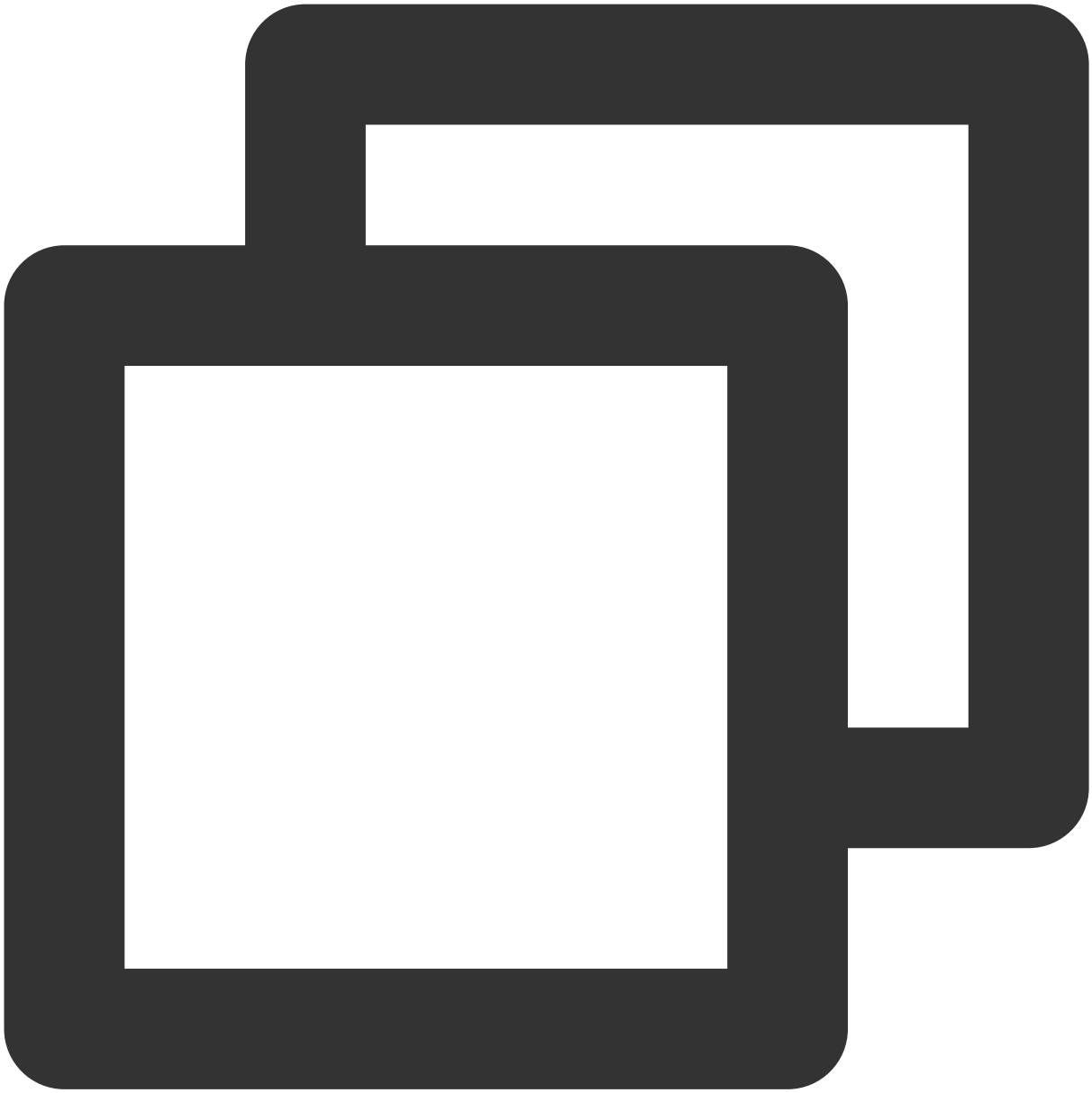
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Translation</Tag>
    <Input>
        <Object>input/en.pdf</Object>
        <Lang>en</Lang>
        <Type>pdf</Type>
        <BasicType>pptx</BasicType>
    </Input>
    <Operation>
        <Translation>
            <Lang>zh</Lang>
            <Type>pdf</Type>
        </Translation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/zh.pdf</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	创建任务的 Tag：Translation	String	是
Input	Request	待操作的对象信息	Container	是
Operation	Request	操作规则	Container	是
QueueId	Request	任务所在的队列 ID	String	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级	String	否

版权所有：腾讯云计算（北京）有限责任公司

		高于队列的回调类型		
CallBack	Request	任务回调地址，优先级高于队列的回调地址。设置为 no 时，表示队列的回调地址不产生回调	String	否
CallBackMqConfig	Request	任务回调 TDMQ 配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig	Container	否

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Object	Request.Input	源文档文件名	String	是	单文件（docx/xlsx/html/markdown/txt）：800万字符 有页数的（pdf/pptx）：300页 文本文件（txt）：10MB 二进制文件（pdf\\docx\\pptx\\xlsx）：60MB 图片文件（jpg\\jpeg\\png）：10MB
Lang	Request.Input	文档语言类型	String	是	zh：简体中文 zh-hk：繁体中文 zh-tw：繁体中文 zh-tr：繁体中文 en：英语 ar：阿拉伯语 de：德语 es：西班牙语 fr：法语 id：印尼语 it：意大利语 ja：日语 pt：葡萄牙语 ru：俄语 ko：韩语 km：高棉语 lo：老挝语
Type	Request.Input	文档类型	String	是	pdf docx pptx xlsx txt

版权所有：腾讯云计算（北京）有限责任公司

					xml html：只能翻译 HTML 里的文本节点，需要通过 JS 动态加载的不进行翻译 markdown jpg jpeg png
BasicType	Request.Input	原始文档类型	String	否	仅在 Type=pdf/jpg/jpeg/png 时使用，当值为pdf时，仅支持 docx、pptx 当值为jpg/jpeg/png/webp时，仅支持 txt，不填写此字段时，直接返回翻译结果

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Translation	Request.Operation	翻译参数	Container	是
Output	Request.Operation	结果输出地址	Container	是（当NoNeedOutput为true时非必选）
UserData	Request.Operation	透传用户信息, 可打印的 ASCII 码, 长度不超过1024	String	否
JobLevel	Request.Operation	任务优先级，级别限制：0 、1 、2 。级别越大任务优先级越高，默认为0	String	否
NoNeedOutput	Request.Operation	仅输出结果，不生成结果文件 true/false。该参数原文档类型为图片时有效	String	否

Container 类型 Translation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Lang	Request.Operation.Translation	目标语言类型	String	是	源语言类型为

版权所有：腾讯云计算（北京）有限责任公司

					zh/zh-hk/zh-tw/zh-tr 时支持：en、ar、de、es、fr、id、it、ja、it、ru、ko、km、lo、pt en 时支持：zh、zh-hk、zh-tw、zh-tr、ar、de、es、fr、id、it、ja、it、ru、ko、km、lo、pt 其他类型时支持：zh、zh-hk、zh-tw、zh-tr、en
Type	Request.Operation.Translation	文档类型	String	是	源文件类型与目标文件类型映射关系如下： docx：docx pptx：pptx xlsx：xlsx txt：txt xml：xml html：html markdown：markdown pdf：pdf docx png：txt jpg：txt jpeg：txt

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Region	Request.Operation.Output	存储桶的地域	String	是
Bucket	Request.Operation.Output	存储结果的存储桶	String	是
Object	Request.Operation.Output	输出结果的文件名	String	是

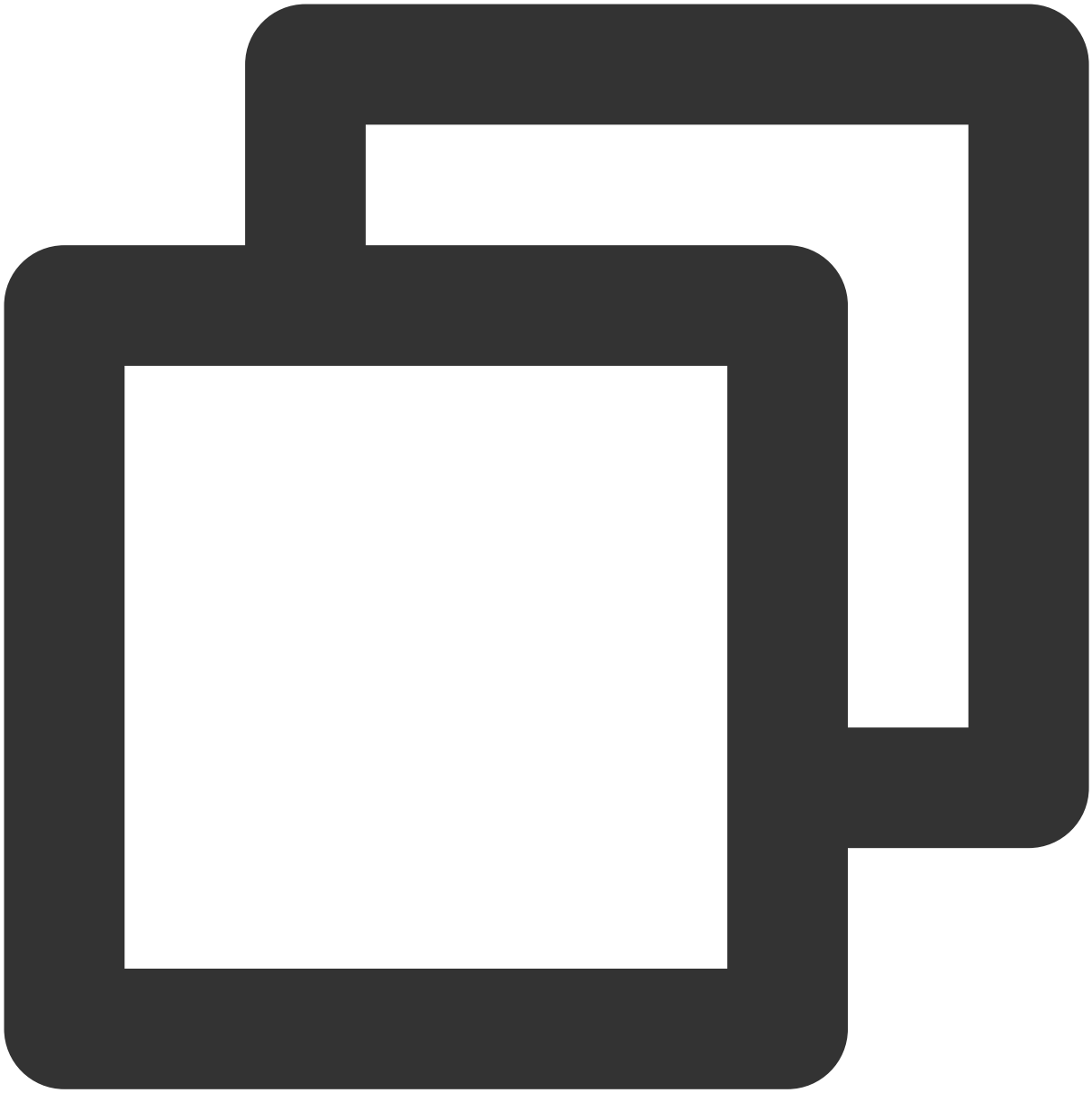
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>ac7c990a00bf211ed946af9e0691f2b7a</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <Tag>Translation</Tag>
```

```
        <Input>
            <Object>input/en.pdf</Object>
            <Lang>en</Lang>
            <Type>pdf</Type>
            <BasicType>pptx</BasicType>
        </Input>
        <Operation>
            <Translation>
                <Lang>zh</Lang>
                <Type>pdf</Type>
            </Translation>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/zh.pdf</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
JobId	Response.JobsDetail	新创建任务的 ID	String
Tag	Response.JobsDetail	新创建任务的 Tag：Translation	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、	String

版权所有：腾讯云计算（北京）有限责任公司

		Failed、Pause、Cancel 其中一个	
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.JobsDetail	任务的开始时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
QueueId	Response.JobsDetail	任务所属的队列 ID	String
Input	Response.JobsDetail	同请求中的 Request.Input 节点。	Container
Operation	Response.JobsDetail	该任务的规则	Container

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
Translation	Response.JobsDetail.Operation	同请求中的 Request.Operation.Translation	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String
AITranslateResult	Response.JobsDetail.Operation	翻译结果详情	Container

Container 节点 AITranslateResult 的内容：

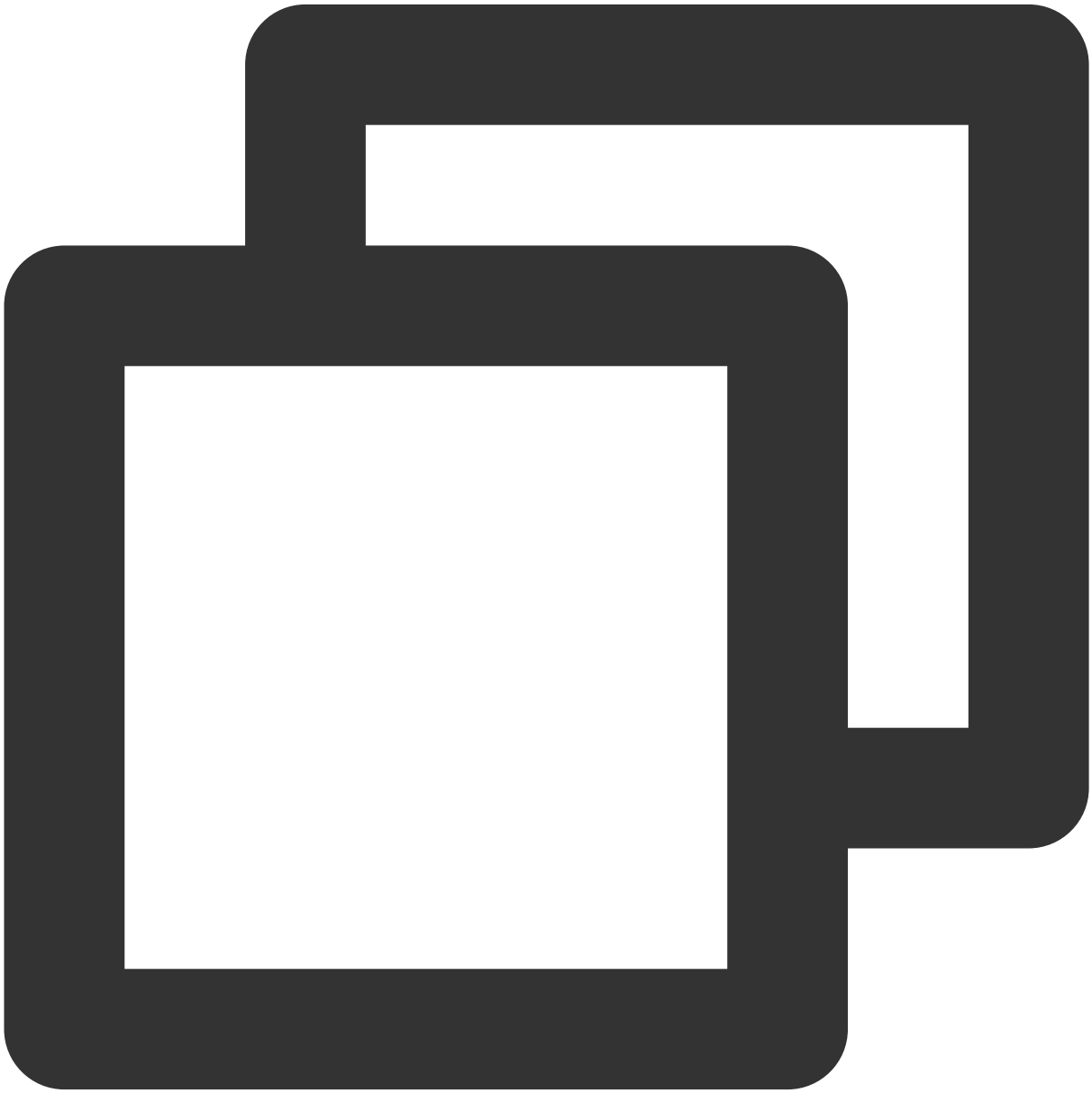
节点名称（关键字）	父节点	描述	类型
Result	Response.JobsDetail.Operation	翻译结果内容	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

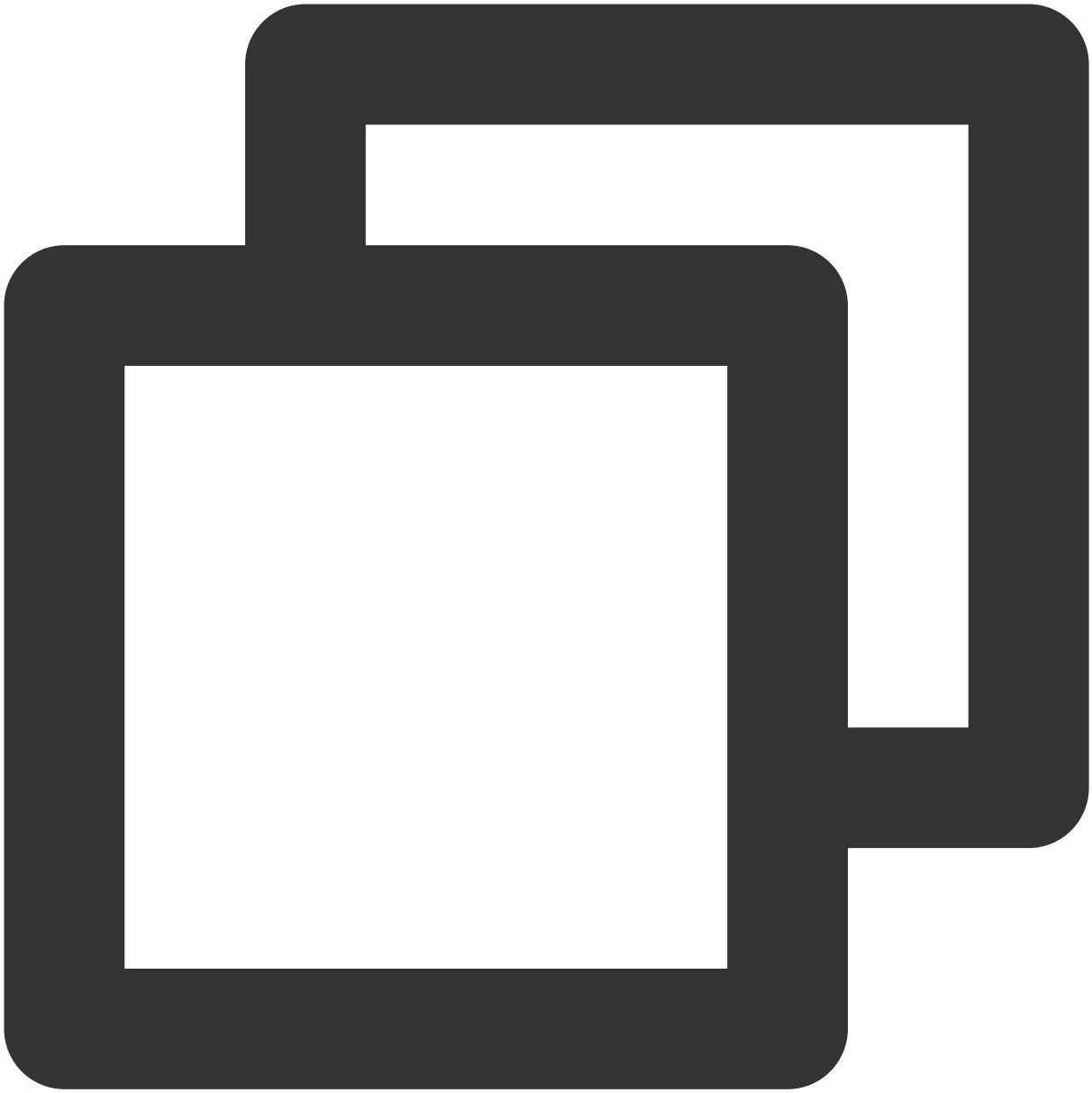
请求



```
POST /ai_jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Translation</Tag>
    <Input>
        <Object>input/en.pdf</Object>
        <Lang>en</Lang>
```

```
        <Type>pdf</Type>
        <BasicType>pptx</BasicType>
    </Input>
    <Operation>
        <Translation>
            <Lang>zh</Lang>
            <Type>pdf</Type>
        </Translation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/zh.pdf</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>ac7c990a00bf211ed946af9e0691f2b7a</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <Tag>Translation</Tag>
        <Input>
            <Object>input/en.pdf</Object>
            <Lang>en</Lang>
            <Type>pdf</Type>
            <BasicType>pptx</BasicType>
        </Input>
        <Operation>
            <Translation>
                <Lang>zh</Lang>
                <Type>pdf</Type>
            </Translation>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/zh.pdf</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

提交分词任务

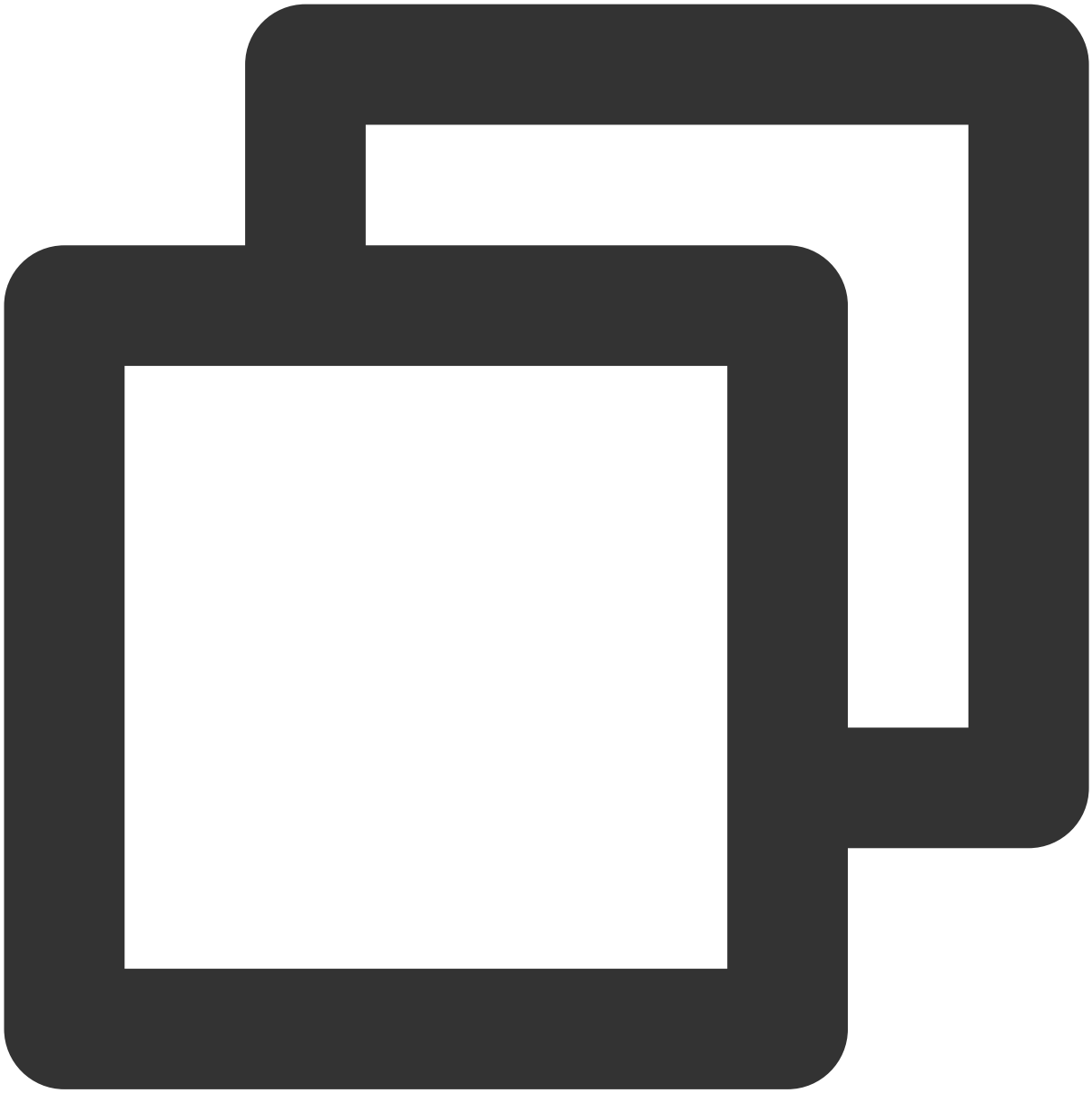
最近更新时间：2023-12-30 16:46:10

功能描述

提交一个分词任务。

请求

请求示例



```
POST /ai_jobs HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

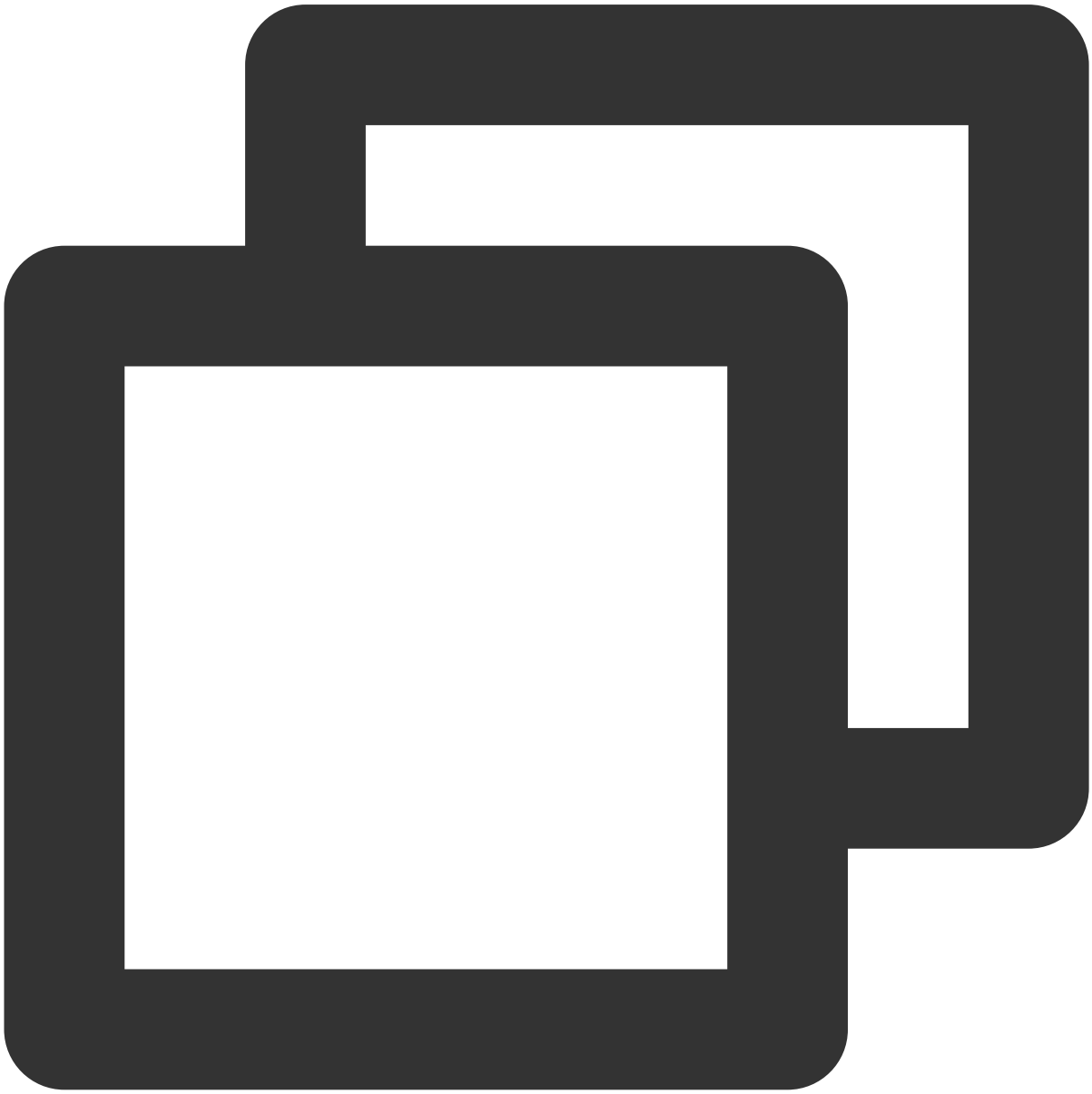
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>WordsGeneralize</Tag>
    <Input>
        <Object>text.txt</Object>
    </Input>
    <Operation>
        <WordsGeneralize>
            <NerMethod>DL</NerMethod>
            <SegMethod>Mix</SegMethod>
        </WordsGeneralize>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	创建任务的 Tag：WordsGeneralize	String	是
Input	Request	待操作的对象信息	Container	是
Operation	Request	操作规则	Container	是
QueueId	Request	任务所在的队列 ID	String	是
CallBack	Request	任务回调地址，优先级高于队列的回调地址。设置为 no 时，表示队列的回调地址不产生回调	String	否
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	媒体文件名	String	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
WordsGeneralize	Request.Operation	指定分词参数	Container	是
UserData	Request.Operation	透传用户信息，可打印的 ASCII 码，长度不超过1024	String	否
JobLevel	Request.Operation	任务优先级，级别限制：0、1、2。级别越大任务优先级越高，默认为0	String	否

Container 类型 WordsGeneralize 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
NerMethod	Request.Operation.WordsGeneralize	ner 方式，默认值 DL	String	否	NerBasic 或 DL
SegMethod	Request.Operation.WordsGeneralize	分词粒度，默认值 MIX	String	否	SegBasic 或 MIX

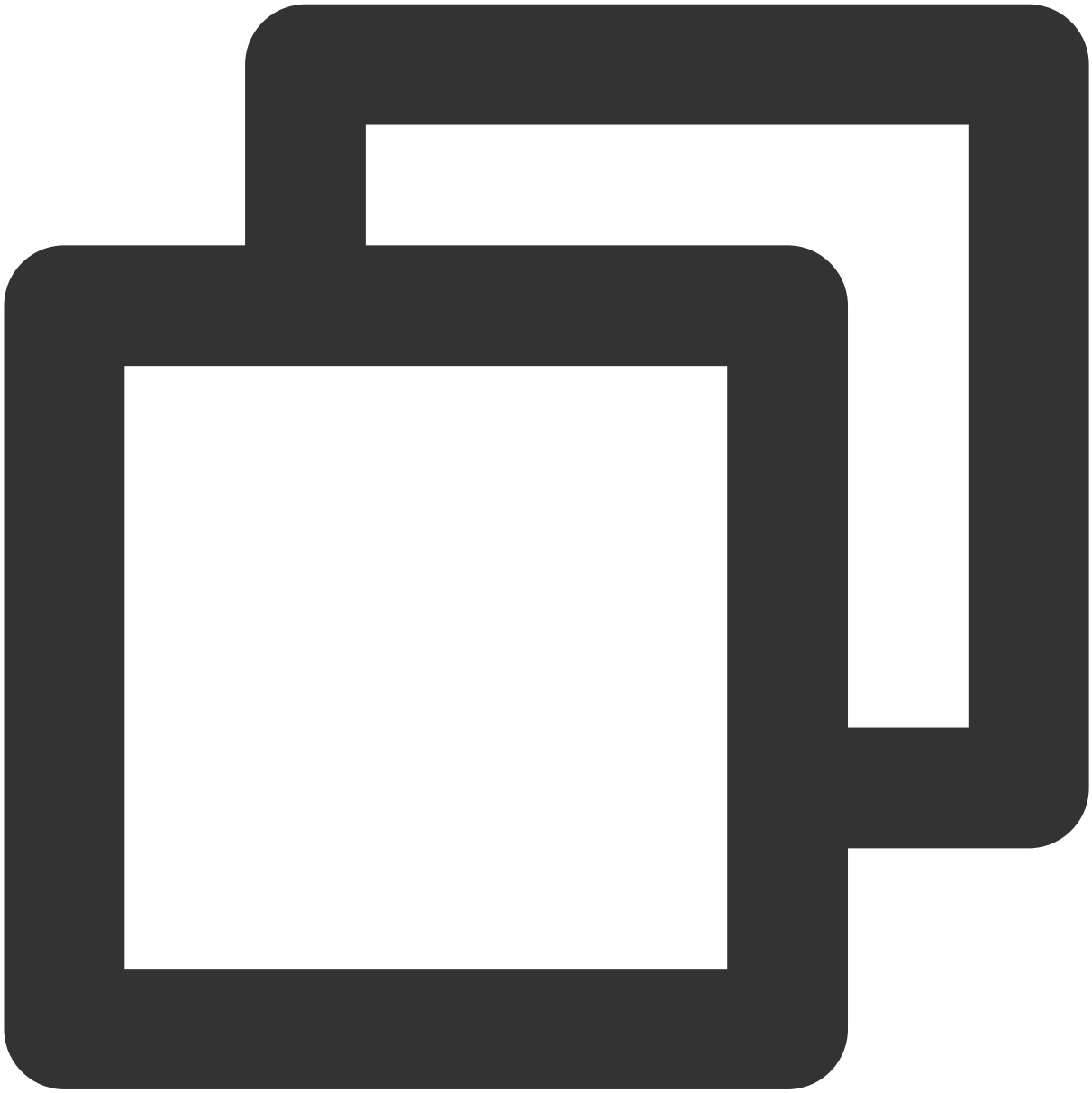
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>ac7c990a00bf211ed946af9e0691f2b7a</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <Tag>WordsGeneralize</Tag>
```

```
        <Input>
            <Object>text.txt</Object>
        </Input>
        <Operation>
            <WordsGeneralize>
                <NerMethod>DL</NerMethod>
                <SegMethod>Mix</SegMethod>
            </WordsGeneralize>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
JobId	Response.JobsDetail	新创建任务的 ID	String
Tag	Response.JobsDetail	新创建任务的 Tag：WordsGeneralize	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.JobsDetail	任务的开始时间	String
EndTime	Response.JobsDetail	任务的结束时间	String

版权所有：腾讯云计算（北京）有限责任公司

QueueId	Response.JobsDetail	任务所属的队列 ID	String
Input	Response.JobsDetail	同请求中的 Request.Input 节点	Container
Operation	Response.JobsDetail	该任务的规则	Container

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
WordsGeneralize	Response.JobsDetail.Operation	同请求中的 Request.Operation.WordsGeneralize	Containe
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String
WordsGeneralizeResult	Response.JobsDetail.Operation	分词结果，任务执行成功时返回	Containe

Container 节点 WordsGeneralizeResult 的内容：

节点名称（关键字）	父节点	描述	类型
WordsGeneralizeLable	Response.JobsDetail.Operation.WordsGeneralizeResult	智能分类结果	Container数组
WordsGeneralizeToken	Response.JobsDetail.Operation.WordsGeneralizeResult	分词详细结果	Container数组

Container 节点 WordsGeneralizeLable 的内容：

节点名称（关键字）	父节点	描述	类型
Category	Response.JobsDetail.Operation.WordsGeneralizeResult.WordsGeneralizeLable	类别	string
Word	Response.JobsDetail.Operation.WordsGeneralizeResult.WordsGeneralizeLable	词汇	string

Container 节点 WordsGeneralizeToken 的内容：

节点名	父节点	描	类型

版权所有：腾讯云计算（北京）有限责任公司

称（关键字）		述	
Word	Response.JobsDetail.Operation.WordsGeneralizeResult.WordsGeneralizeToken	词汇	string
Offset	Response.JobsDetail.Operation.WordsGeneralizeResult.WordsGeneralizeToken	偏移量	string
Length	Response.JobsDetail.Operation.WordsGeneralizeResult.WordsGeneralizeToken	词汇长度	string
Pos	Response.JobsDetail.Operation.WordsGeneralizeResult.WordsGeneralizeToken	词性	string

词性详见下标：

英文缩写	含义
A	形容词
AD	副形词
AN	名形词
B	区别词
C	连词
D	副词
E	叹词
F	方位词
G	语素词
H	前接成分
I	成语
J	简称略语
K	后接成分
L	习用语

版权所有：腾讯云计算（北京）有限责任公司

M	数词
N	名词
NR	人名
NRF	姓
NRG	名
NS	地名
NT	机构团体
NZ	其他专名
NX	非汉字串
O	拟声词
P	介词
Q	量词
R	代词
S	处所词
T	时间词
U	助词
V	动词
VD	副动词
VN	名动词
W	标点符号
X	非语素字
Y	语气词
Z	状态词
AG	形语素
BG	区别语素

版权所有：腾讯云计算（北京）有限责任公司

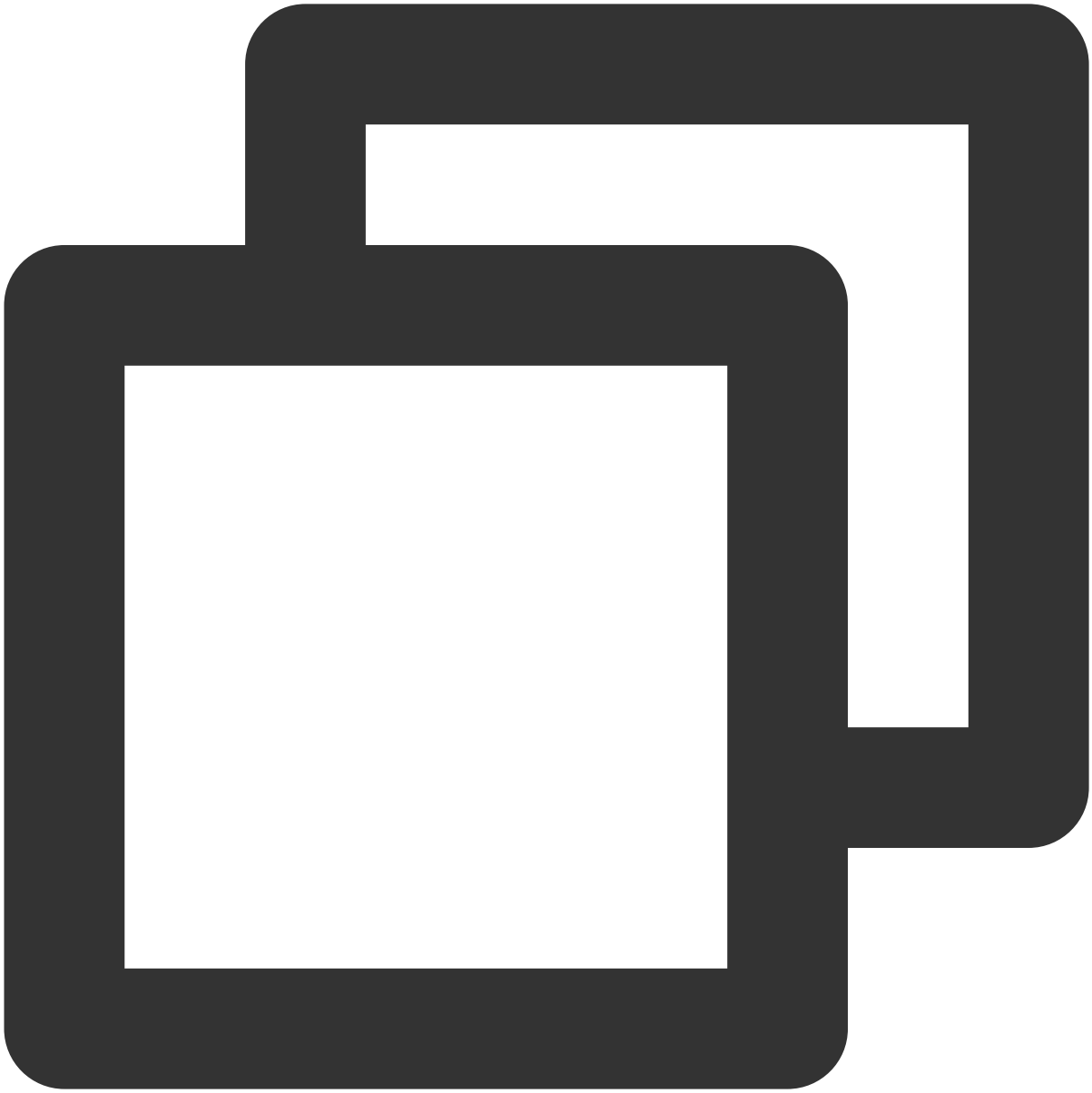
DG	副语素
MG	数词性语素
NG	名语素
QG	量语素
RG	代语素
TG	动语素
VG	量语素
YG	语气词语素
ZG	状态词语素
UNK	未知词性

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

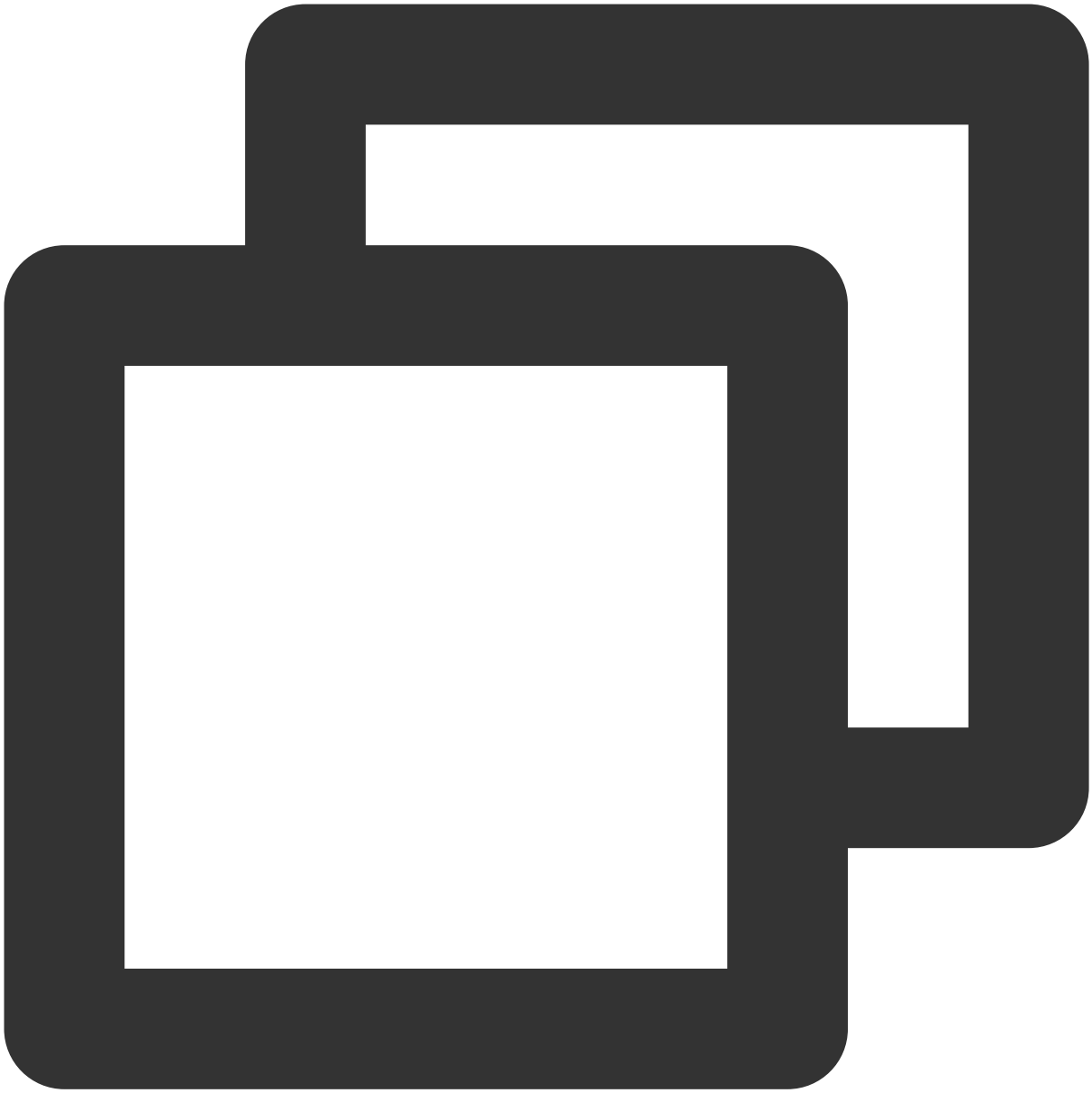
请求



```
POST /ai_jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-123456789.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>WordsGeneralize</Tag>
    <Input>
        <Object>text.txt</Object>
    </Input>
```

```
    <Operation>
        <WordsGeneralize>
            <NerMethod>DL</NerMethod>
            <SegMethod>Mix</SegMethod>
        </WordsGeneralize>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>ac7c990a00bf211ed946af9e0691f2b7a</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <Tag>WordsGeneralize</Tag>
        <Input>
            <Object>text.txt</Object>
        </Input>
        <Operation>
            <WordsGeneralize>
                <NerMethod>DL</NerMethod>
                <SegMethod>Mix</SegMethod>
            </WordsGeneralize>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

取消内容识别任务

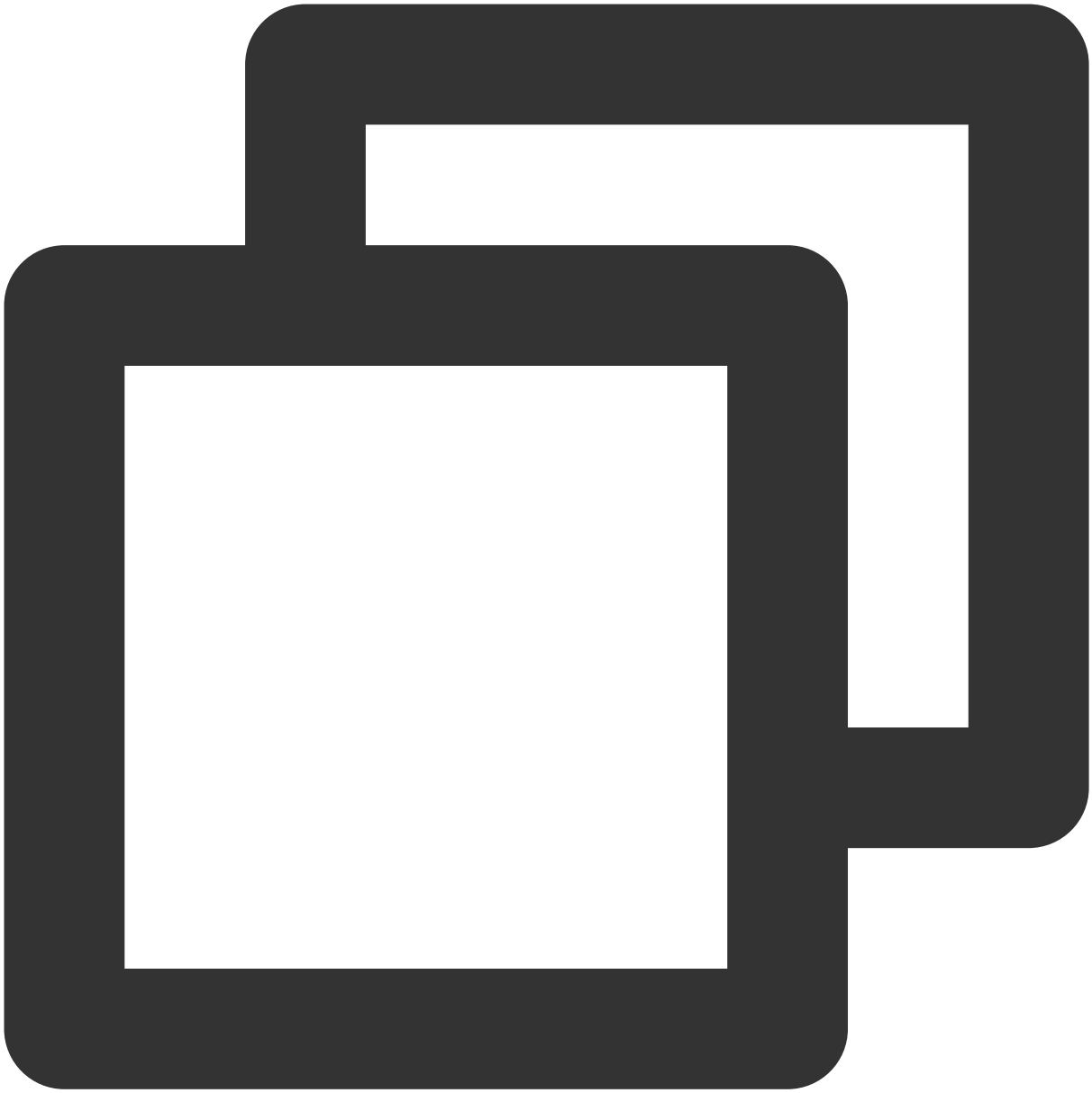
最近更新时间：2023-12-30 16:46:17

功能描述

CancelAIJob 用于取消一个任务，只有未执行的任务才可以被取消。

请求

请求示例



```
PUT /ai_jobs/<jobId>?cancel HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

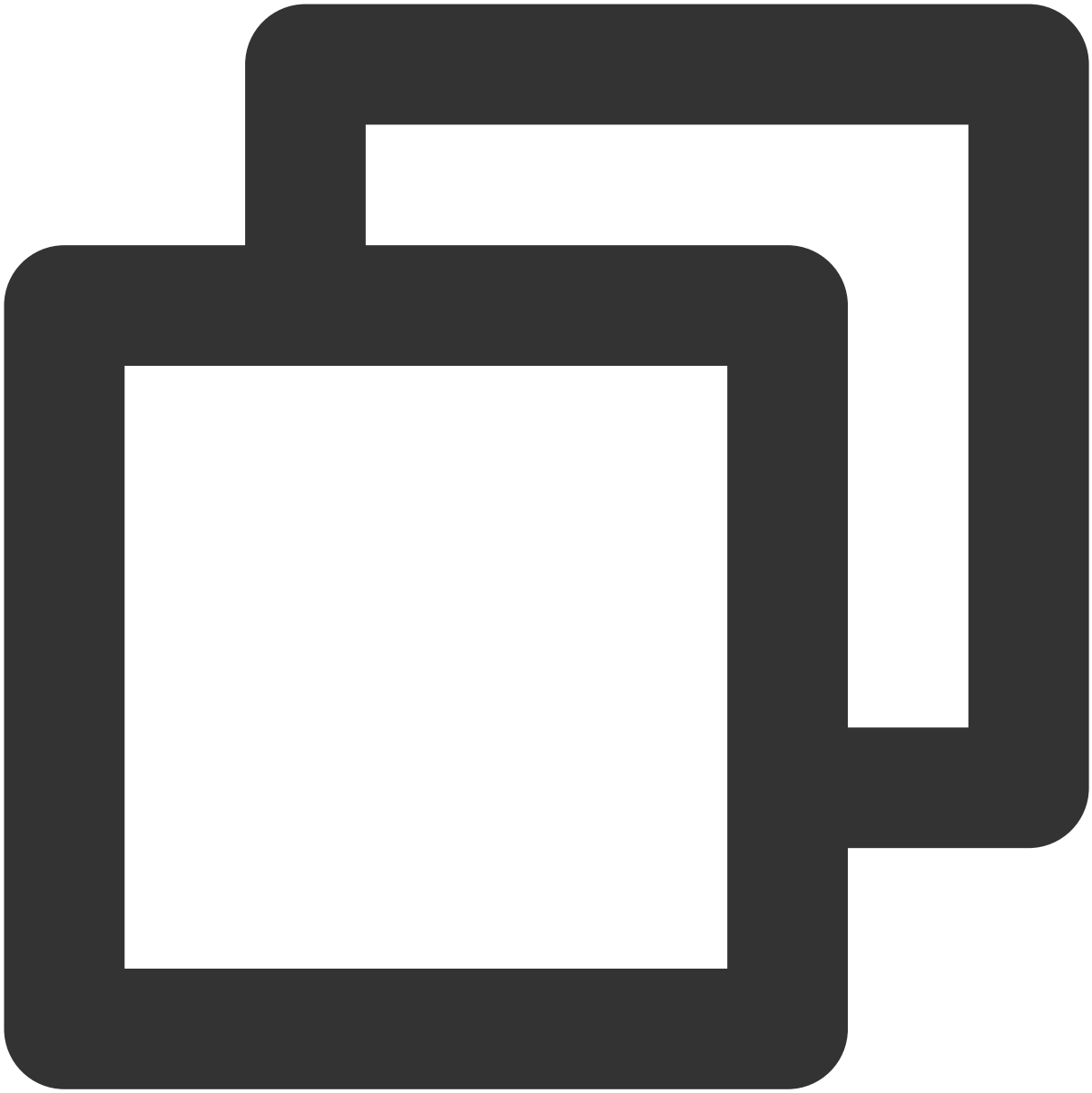
该请求无响应体。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

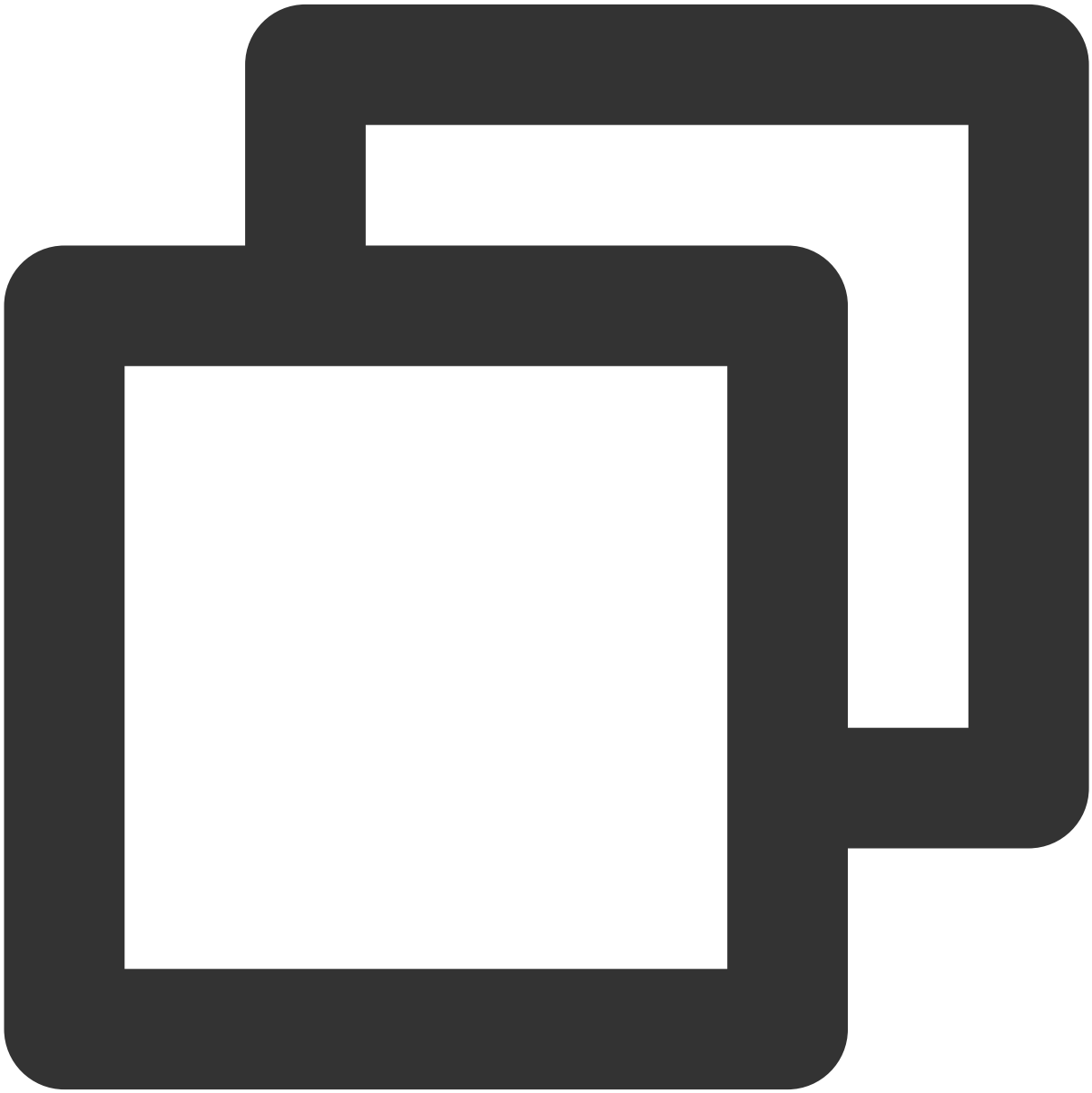
实际案例

请求



```
PUT /ai_jobs/a-xxx-xxx-xxx-xxxx?cancel HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=
```

查询内容识别任务

查询内容识别任务

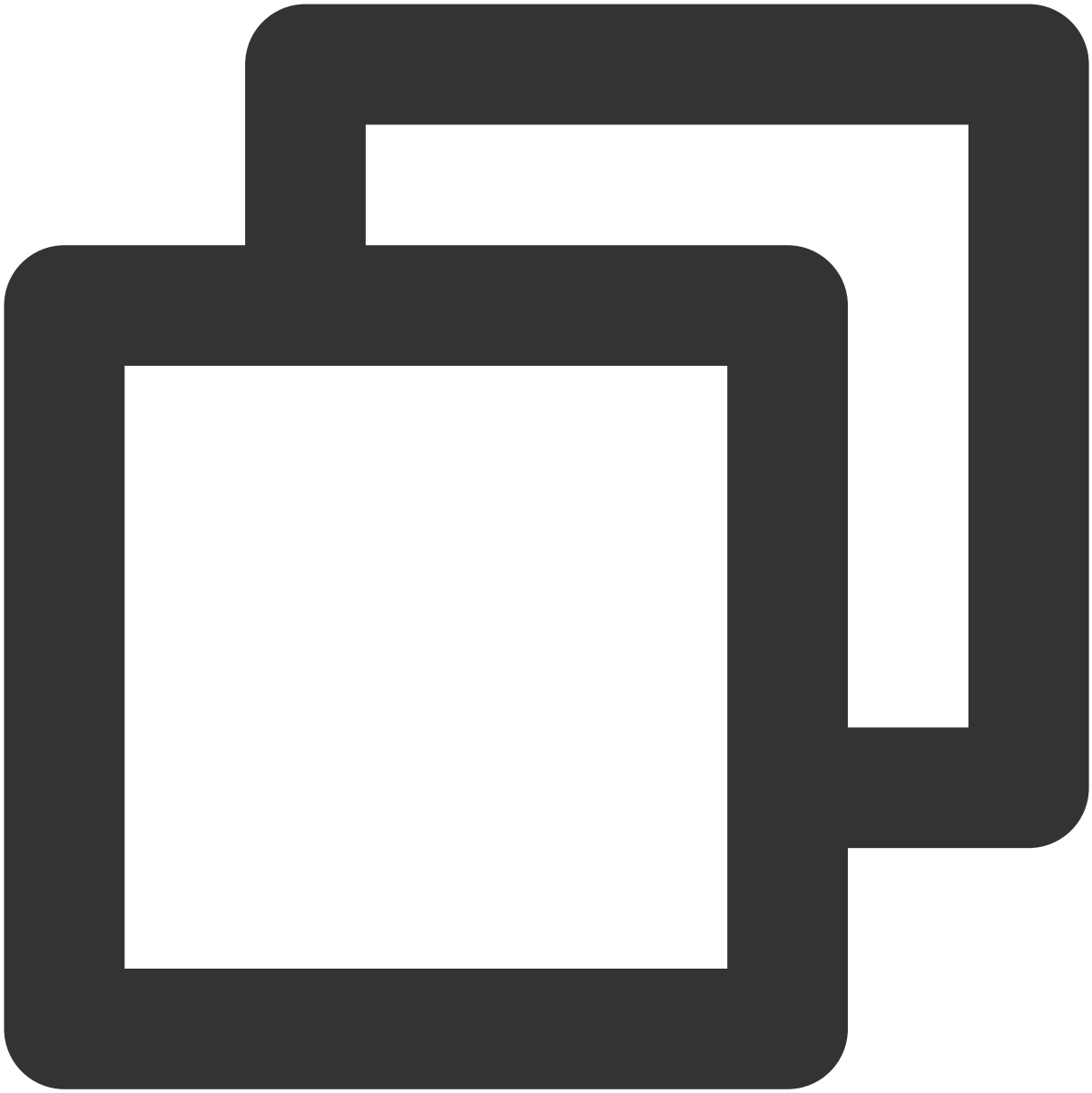
最近更新时间：2023-12-30 16:46:45

功能描述

DescribeAIJob 接口用于查询指定的任务。

请求

请求示例



```
GET /ai_jobs/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

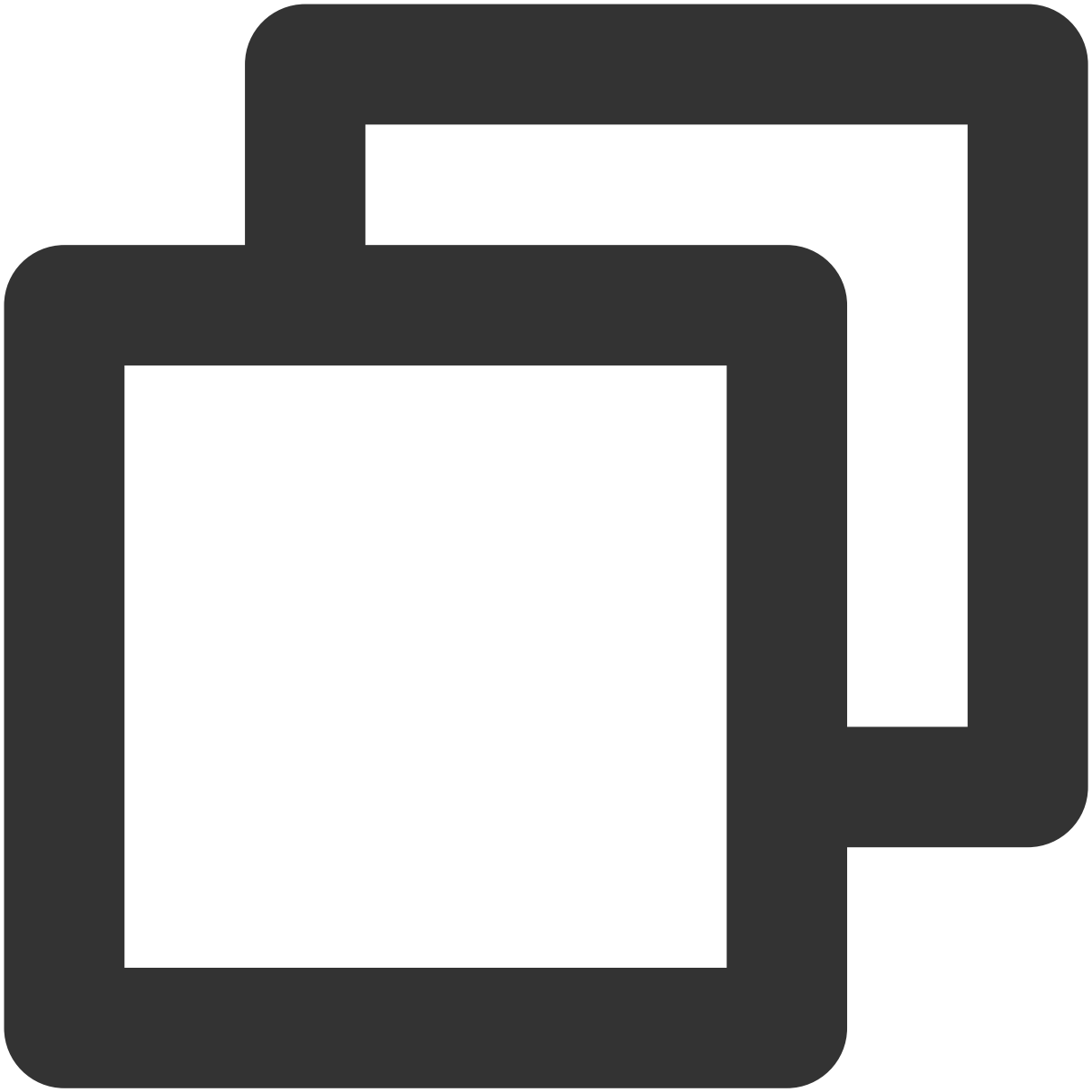
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
    </JobsDetail>
    <NonExistJobIds></NonExistJobIds>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container
NonExistJobIds	Response	查询的 ID 中不存在任务，所有任务都存在时不返回	String

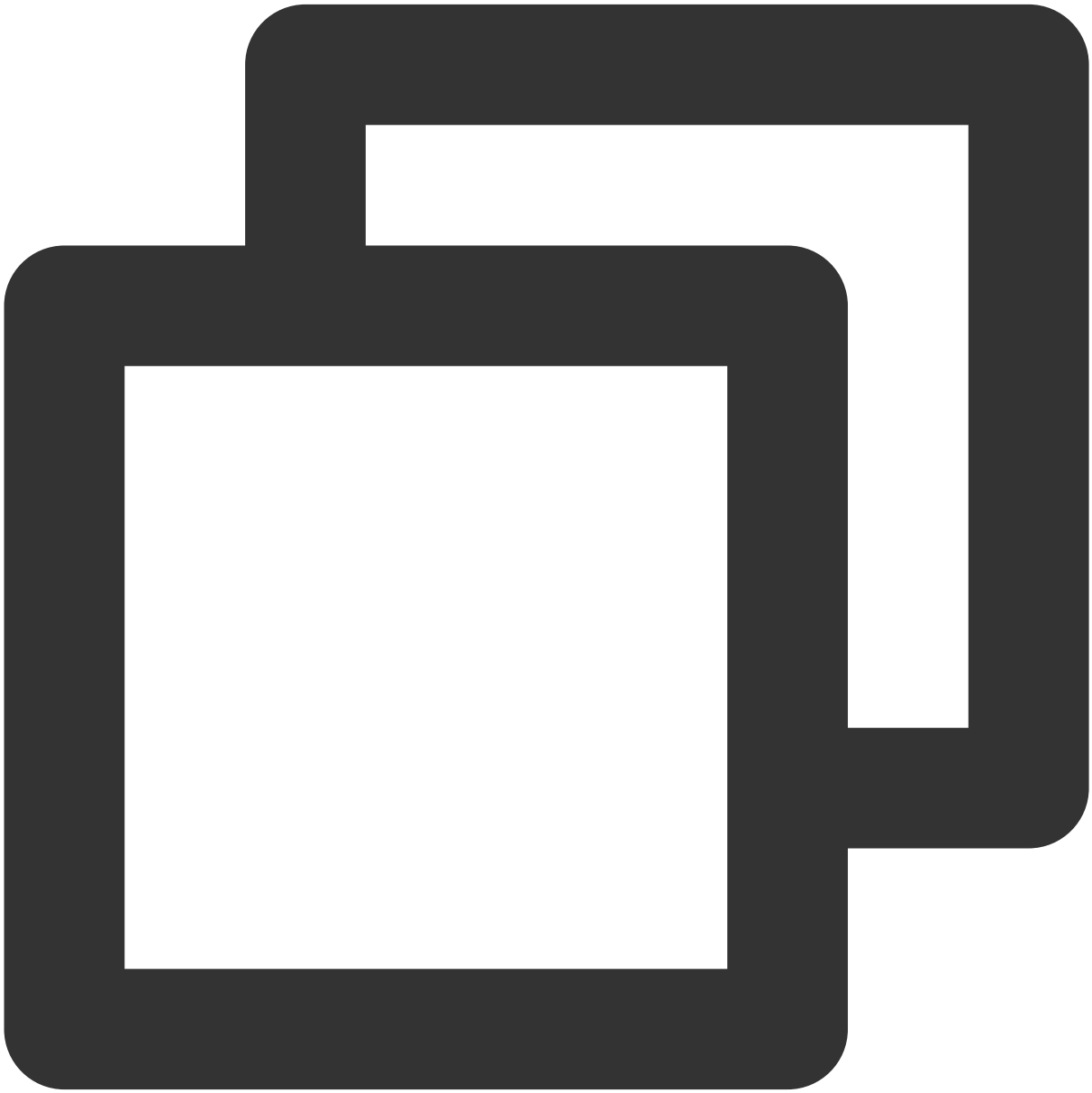
对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：
翻译

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

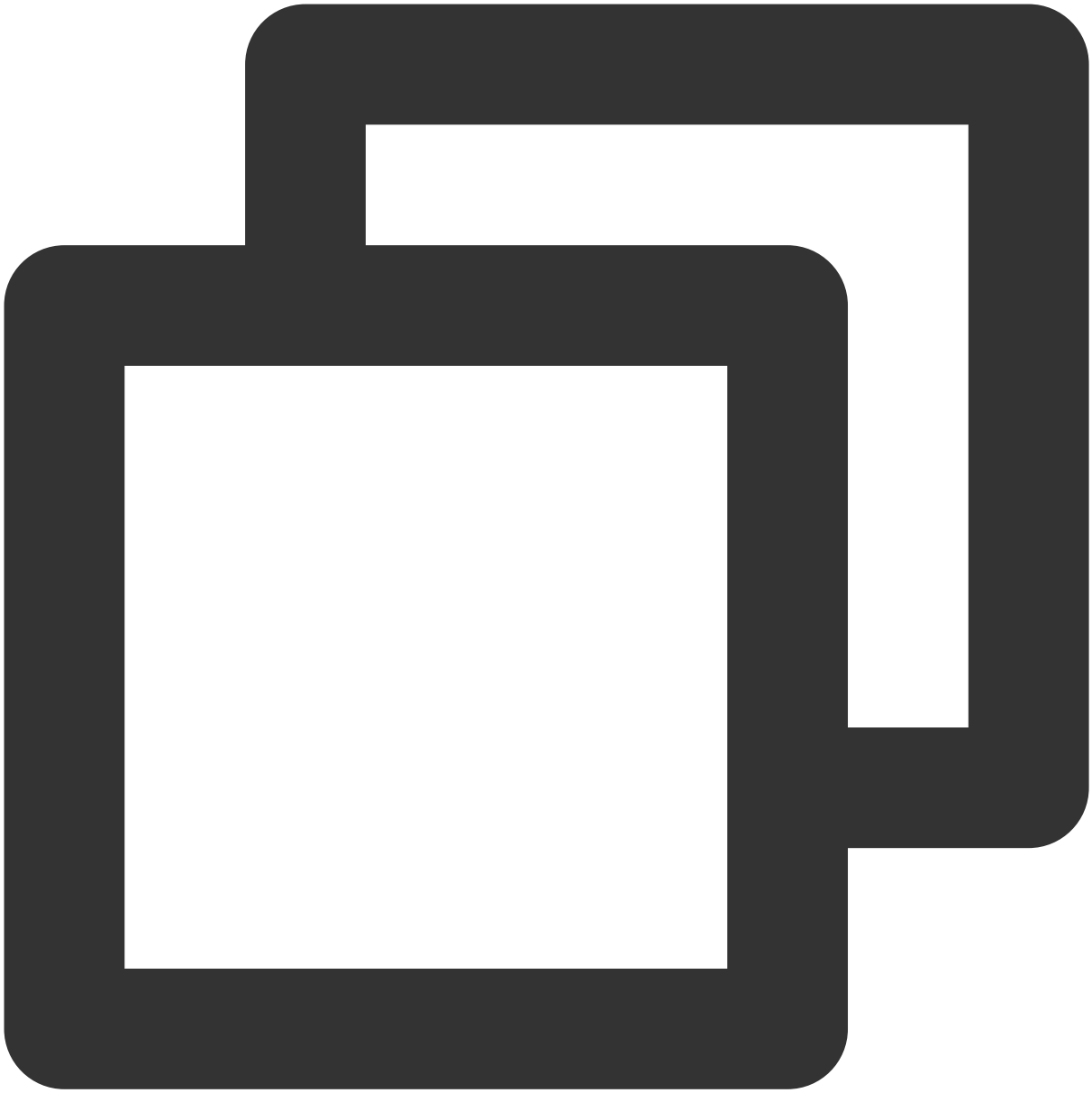
实际案例

请求



```
GET /ai_jobs/a8d121820f5e411ec926ef19d53ba9c6f HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-123456789.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message></Message>
        <JobId>a8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Running</State>
        <CreationTime>2022-07-07T12:12:12+0800</CreationTime>
        <StartTime>2022-07-07T12:12:13+0800</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
        <Tag>Translation<Tag>
        <Input>
            <Object>test.txt</Object>
            <Lang>zh</Lang>
            <Type>txt</Type>
        </Input>
        <Operation>
            <Translation>
                <Lang>en</Lang>
                <Type>txt</Type>
            </Translation>
            <Output>
                <Region>ap-beijing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>test-result.txt</Object>
            </Output>
            <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

内容识别任务回调
翻译任务回调

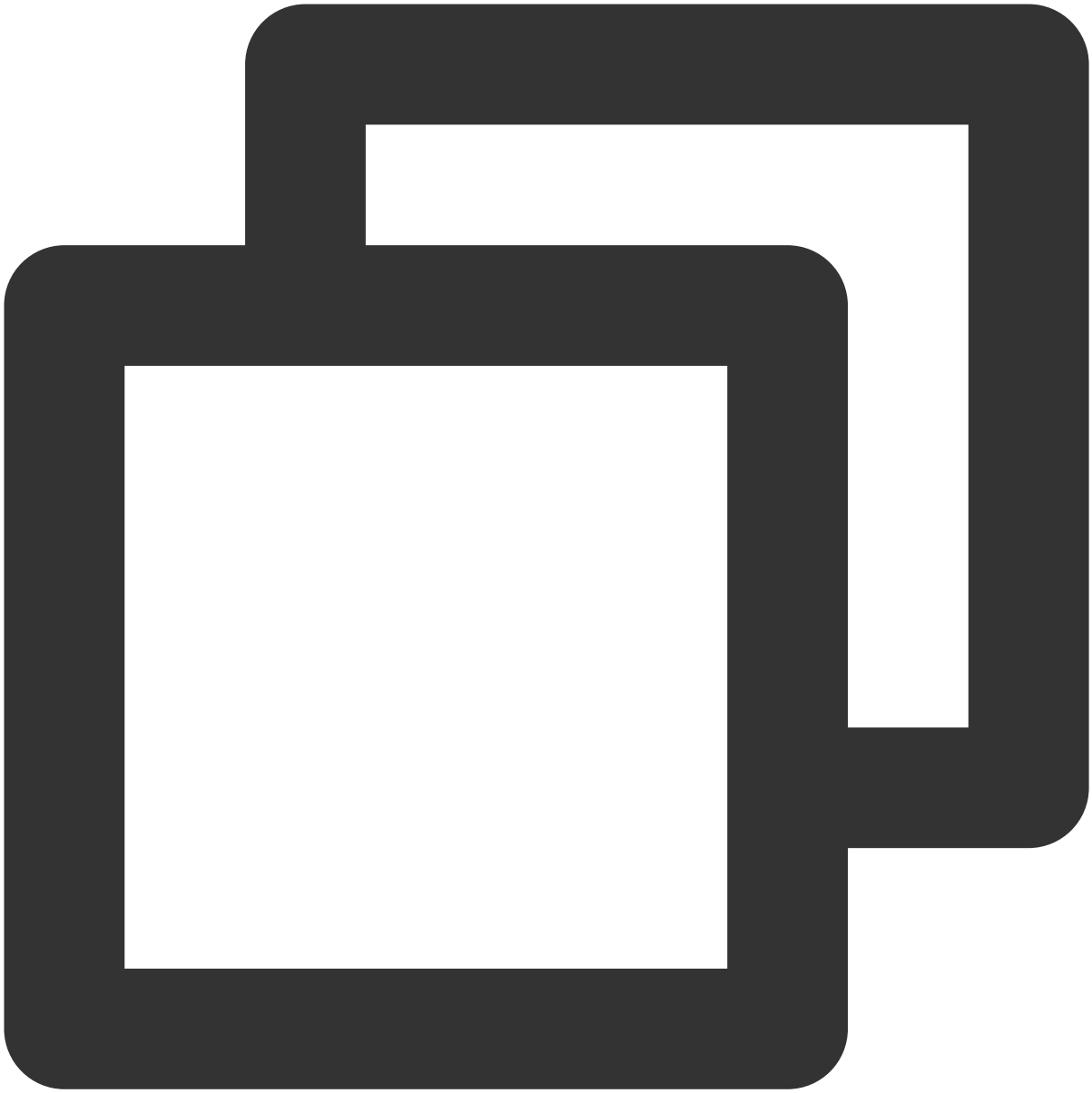
最近更新时间：2023-12-30 16:46:52

功能描述

DescribeMediaJobs 用于拉取符合条件的任务。

请求

请求示例



```
GET /ai_jobs?size=&states=&queueId=&startCreationTime=&endCreationTime= HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	是否必选
queueId	无	拉取该队列 ID 下的任务	String	是
tag	无	任务的 Tag：Translation	String	是
orderByTime	无	Desc 或者 Asc。默认为 Desc	String	否
nextToken	无	请求的上下文，用于翻页。上次返回的值	String	否
size	无	拉取的最大任务数。默认为10。最大为100	Integer	否
states	无	拉取该状态的任务，以 `,` 分割，支持多状态：All、Submitted、Running、Success、Failed、Pause、Cancel。默认为 All	String	否
startCreationTime	无	拉取创建时间大于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`，示例：2001-01-01T00:00:00+0800	String	否
endCreationTime	无	拉取创建时间小于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`，示例：2001-01-01T23:59:59+0800	String	否

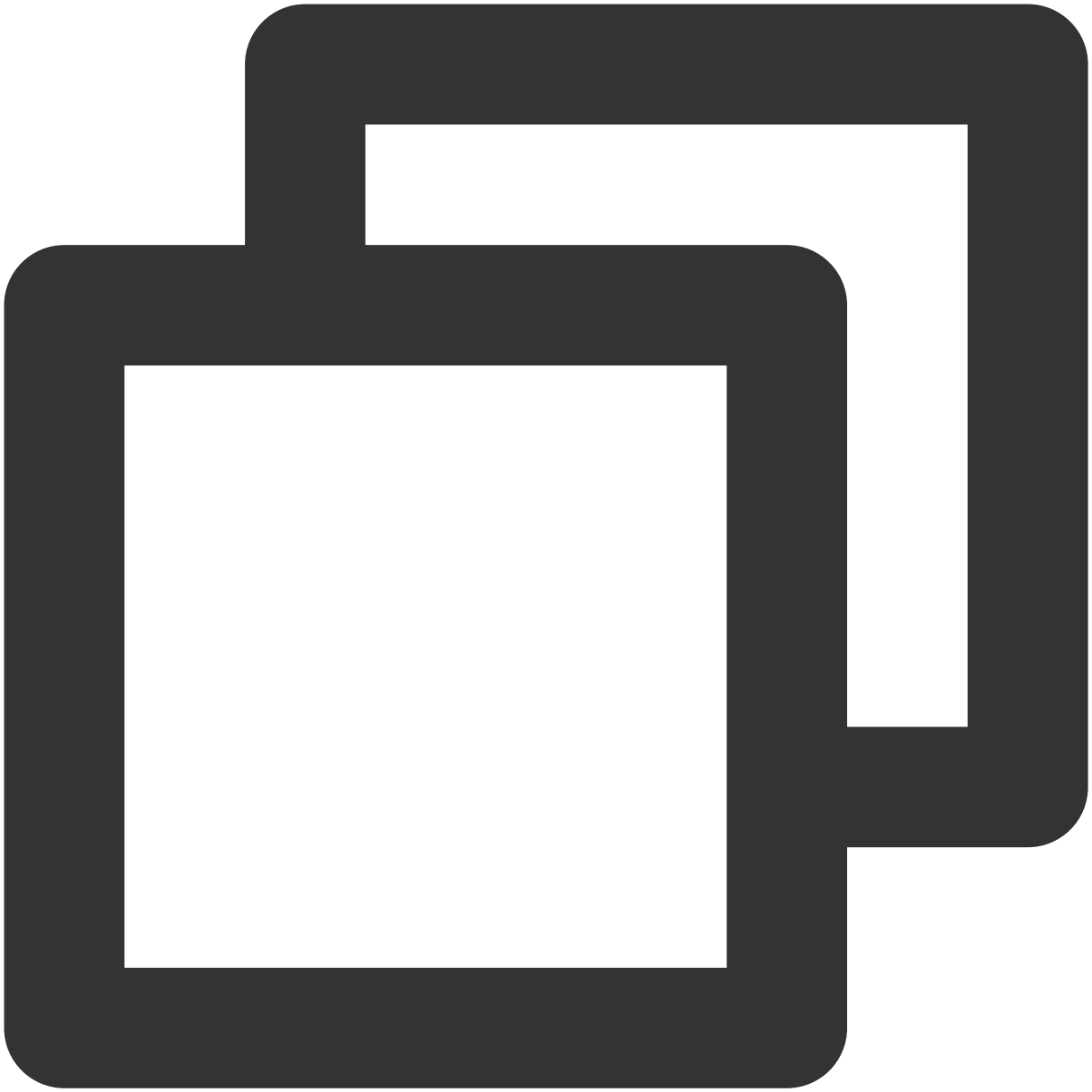
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
  </JobsDetail>
  <NextToken></NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

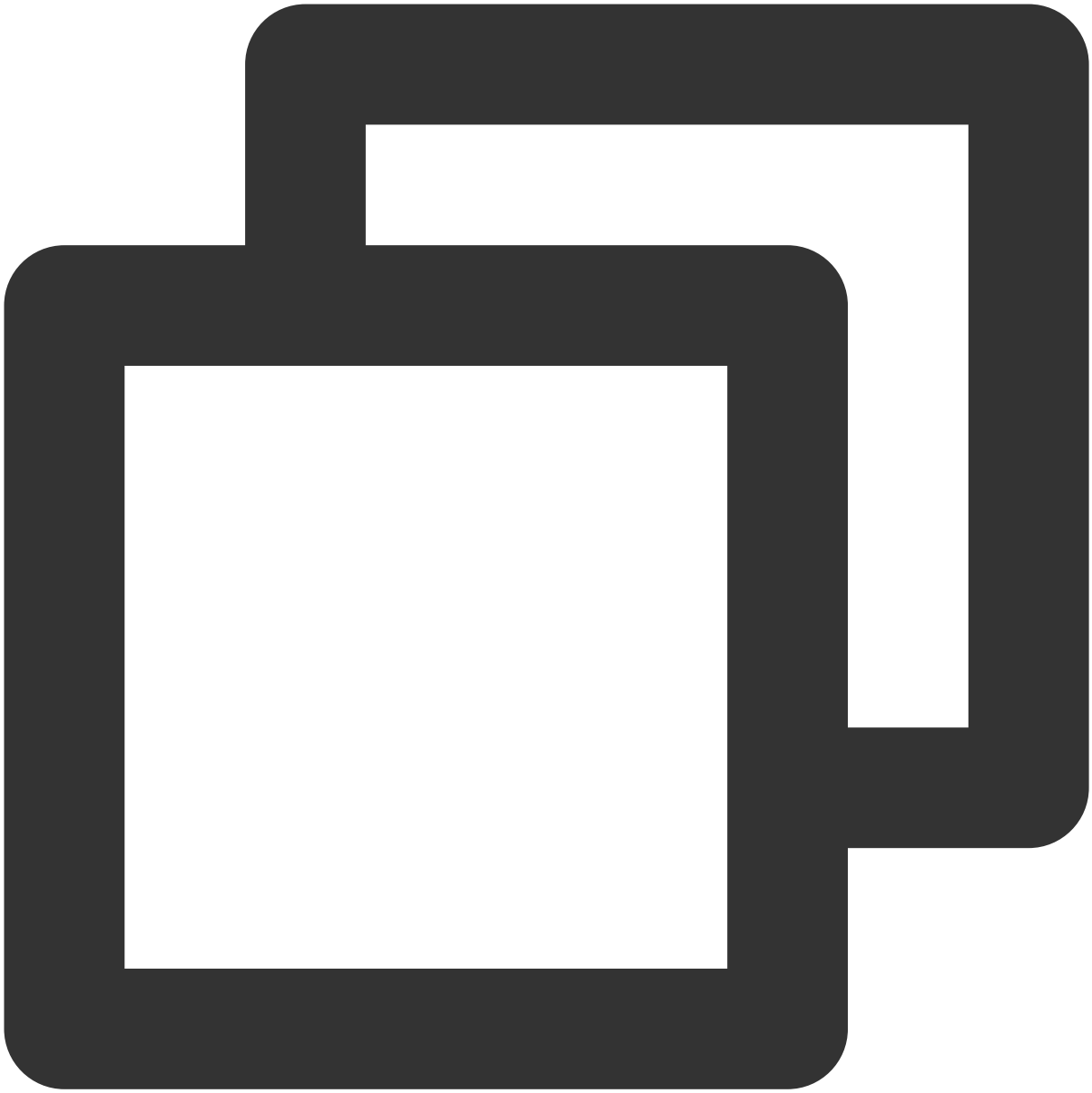
节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息，同 CreateMediaJobs 接口中的 Response.JobsDetail 节点	Container
NextToken	Response	翻页的上下文 Token	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

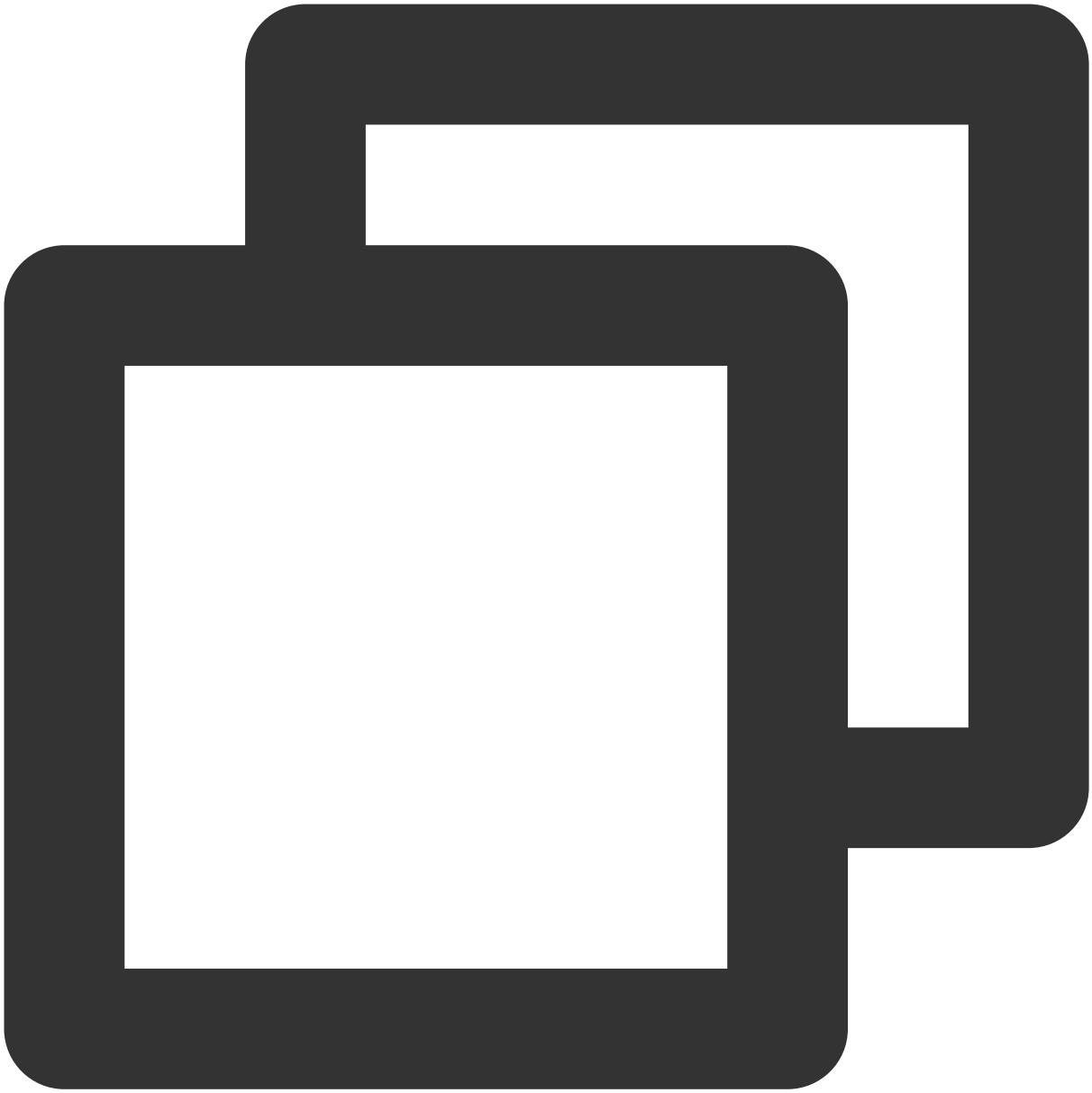
实际案例

请求



```
GET /ai_jobs?queueId=aaaaaaaaaaa&tag=SDRtoHDR HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <Code>Success</Code>
```

```
    <Message>Success</Message>
    <JobId>aabcxxxxfeipplsdfwe</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <StartTime></StartTime>
    <EndTime></EndTime>
    <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
    <Tag>Translation<Tag>
    <Input>
      <Object>test.txt</Object>
      <Lang>zh</Lang>
      <Type>txt</Type>
    </Input>
    <Operation>
      <Translation>
        <Lang>en</Lang>
        <Type>txt</Type>
      </Translation>
      <Output>
        <Region>ap-beijing</Region>
        <Bucket>examplebucket-1250000000</Bucket>
        <Object>test-result.txt</Object>
      </Output>
    </Operation>
  </JobsDetail>
  <JobsDetail>
    <Code>Success</Code>
    <Message>Success</Message>
    <JobId>aabcxxxxfeipplsdfwe</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <StartTime></StartTime>
    <EndTime></EndTime>
    <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
    <Tag>Translation<Tag>
    <Input>
      <Object>test1.txt</Object>
      <Lang>en</Lang>
      <Type>txt</Type>
    </Input>
    <Operation>
      <Translation>
        <Lang>zh</Lang>
        <Type>txt</Type>
      </Translation>
      <Output>
        <Region>ap-beijing</Region>
```

```
        <Bucket>examplebucket-1250000000</Bucket>
        <Object>test-result1.txt</Object>
      </Output>
    </Operation>
  </JobsDetail>
</Response>
```

分词任务回调

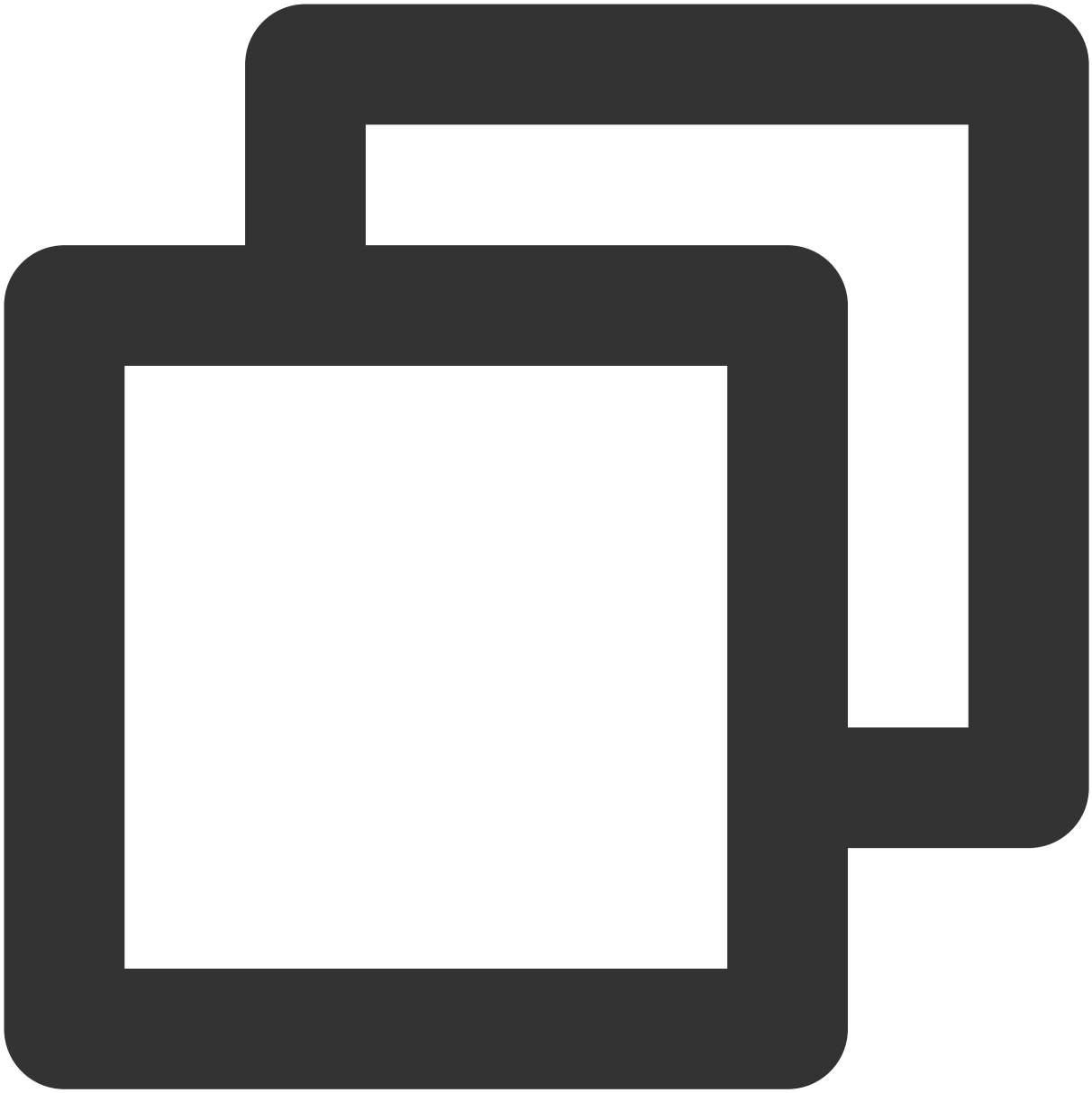
最近更新时间：2023-12-30 16:47:22

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-25T16:35:39+0800</CreationTime>
        <EndTime>2022-07-25T16:35:43+0800</EndTime>
        <Input>
            <Object>text.txt</Object>
            <BucketId>test-123456789</BucketId>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</JobId>
        <Message>Success</Message>
        <Operation>
            <JobLevel>0</JobLevel>
            <WordsGeneralize>
                <NerMethod>DL</NerMethod>
                <SegMethod>MIX</SegMethod>
            </WordsGeneralize>
            <WordsGeneralizeResult>
                <WordsGeneralizeLable>
                    <Category>交通</Category>
                    <Word>打车</Word>
                </WordsGeneralizeLable>
                <WordsGeneralizeToken>
                    <Length>2</Length>
                    <Offset>0</Offset>
                    <Pos>t</Pos>
                    <Word>今天</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>2</Length>
                    <Offset>2</Offset>
                    <Pos>v</Pos>
                    <Word>打车</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>1</Length>
                    <Offset>4</Offset>
                    <Pos>v</Pos>
                    <Word>花</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>2</Length>
                    <Offset>5</Offset>
                    <Pos>nx</Pos>
                    <Word>60</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>1</Length>
                    <Offset>7</Offset>
                    <Pos>q</Pos>
                    <Word>元</Word>
                </WordsGeneralizeToken>
            </WordsGeneralizeResult>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
EventName	Response	固定值，为 TaskFinish	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

同提交分词任务接口中的 Response.JobsDetail

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

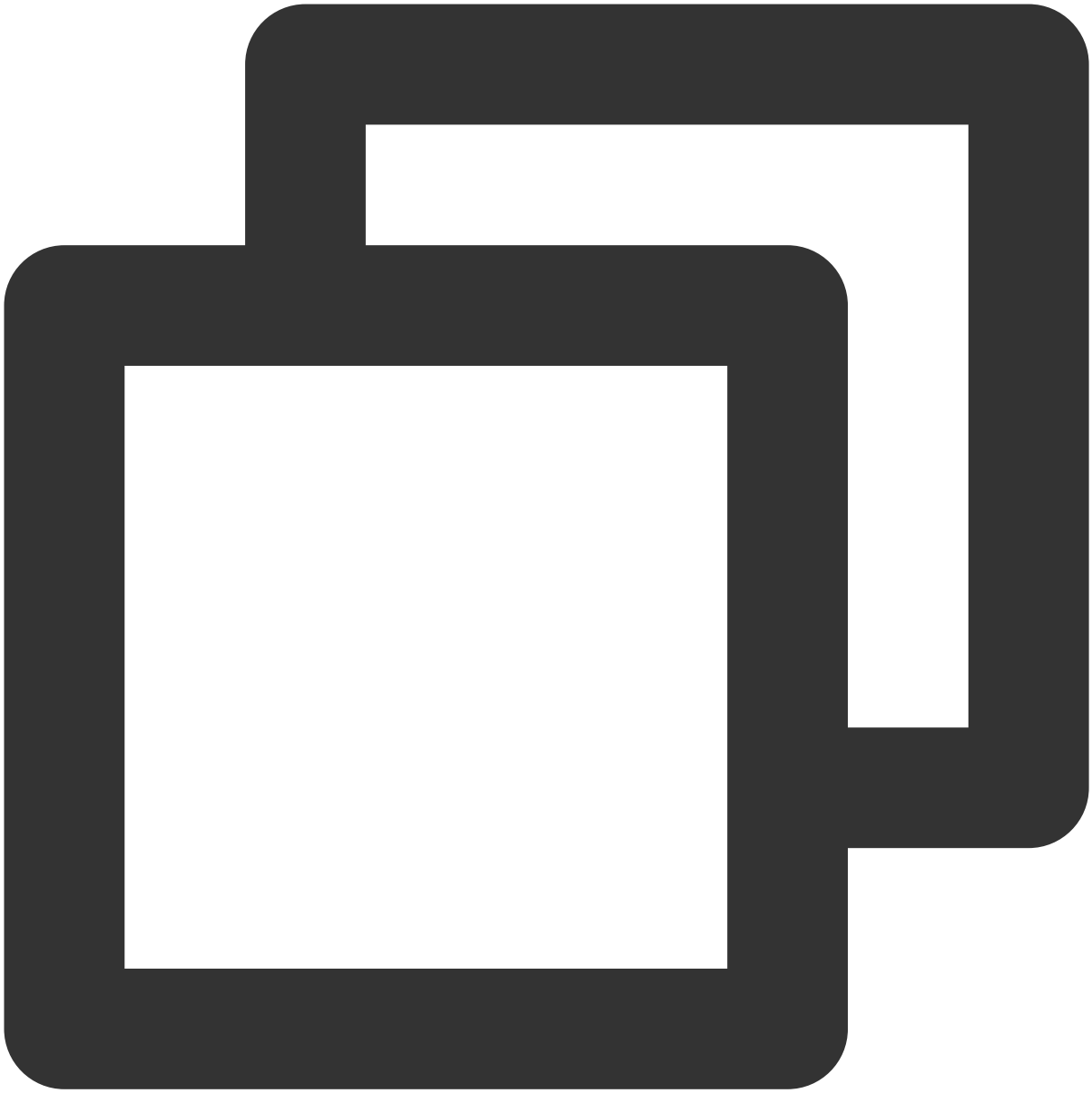
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

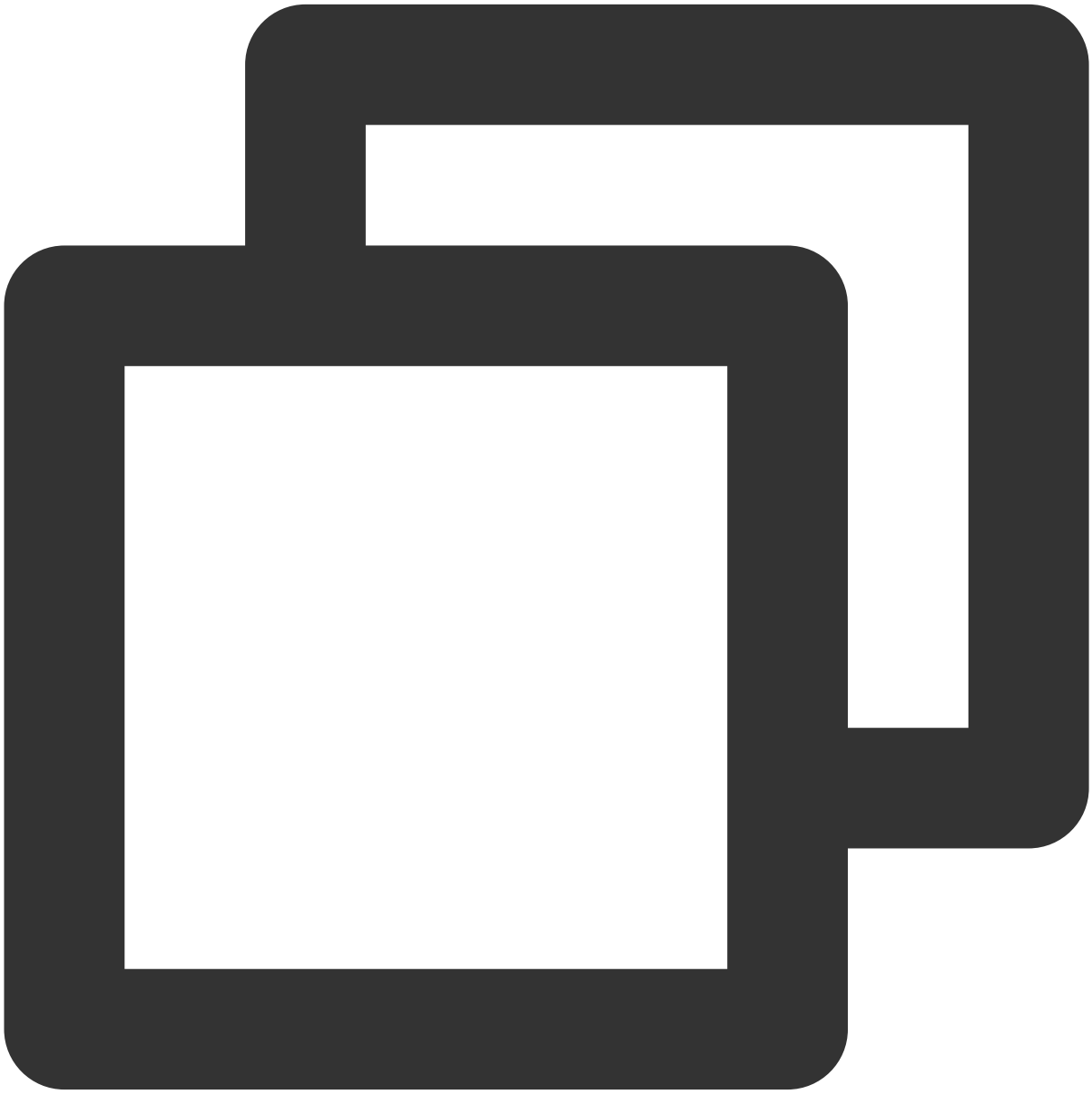
版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-25T16:35:39+0800</CreationTime>
        <EndTime>2022-07-25T16:35:43+0800</EndTime>
        <Input>
            <Object>text.txt</Object>
            <BucketId>test-123456789</BucketId>
            <Region>ap-chongqing</Region>
        </Input>
```

```
        <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</JobId>
        <Message>Success</Message>
        <Operation>
            <JobLevel>0</JobLevel>
            <WordsGeneralize>
                <NerMethod>DL</NerMethod>
                <SegMethod>MIX</SegMethod>
            </WordsGeneralize>
            <WordsGeneralizeResult>
                <WordsGeneralizeLable>
                    <Category>交通</Category>
                    <Word>打车</Word>
                </WordsGeneralizeLable>
                <WordsGeneralizeToken>
                    <Length>2</Length>
                    <Offset>0</Offset>
                    <Pos>t</Pos>
                    <Word>今天</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>2</Length>
                    <Offset>2</Offset>
                    <Pos>v</Pos>
                    <Word>打车</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>1</Length>
                    <Offset>4</Offset>
                    <Pos>v</Pos>
                    <Word>花</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>2</Length>
                    <Offset>5</Offset>
                    <Pos>nx</Pos>
                    <Word>60</Word>
                </WordsGeneralizeToken>
                <WordsGeneralizeToken>
                    <Length>1</Length>
                    <Offset>7</Offset>
                    <Pos>q</Pos>
                    <Word>元</Word>
                </WordsGeneralizeToken>
            </WordsGeneralizeResult>
        </Operation>
    </JobsDetail>
</Response>
```

案例 2：通过任务触发的任务回调，格式为 JSON



```
{
    "Response": {
        "EventName": "TaskFinish",
        "JobsDetail": {
            "Code": "Success",
            "CreationTime": "2022-07-25T16:35:39+0800",
            "EndTime": "2022-07-25T16:35:43+0800",
            "Input": {
```

```
            "Object": "text.txt",
            "BucketId": "test-123456789",
            "Region": "ap-chongqing"
        },
        "JobId": "ac34be3aa0bf411ed9ce39d7cc972e427",
        "Message": "Success",
        "Operation": {
            "JobLevel": "0",
            "WordsGeneralize": {
                "NerMethod": "DL",
                "SegMethod": "MIX"
            },
            "WordsGeneralizeResult": {
                "WordsGeneralizeLable": {
                    "Category": "交通",
                    "Word": "打车"
                },
                "WordsGeneralizeToken": [{
                        "Length": "2",
                        "Offset": "0",
                        "Pos": "t",
                        "Word": "今天"
                    },
                    {
                        "Length": "2",
                        "Offset": "2",
                        "Pos": "v",
                        "Word": "打车"
                    },
                    {
                        "Length": "1",
                        "Offset": "4",
                        "Pos": "v",
                        "Word": "花"
                    },
                    {
                        "Length": "2",
                        "Offset": "5",
                        "Pos": "nx",
                        "Word": "60"
                    },
                    {
                        "Length": "1",
                        "Offset": "7",
                        "Pos": "q",
                        "Word": "元"
                    }
```

```
                        ]
                    }
                }
            }
        }
    }
```

同步媒体处理
媒体截图接口
获取媒体文件截图

最近更新时间：2023-12-30 16:47:39

功能描述

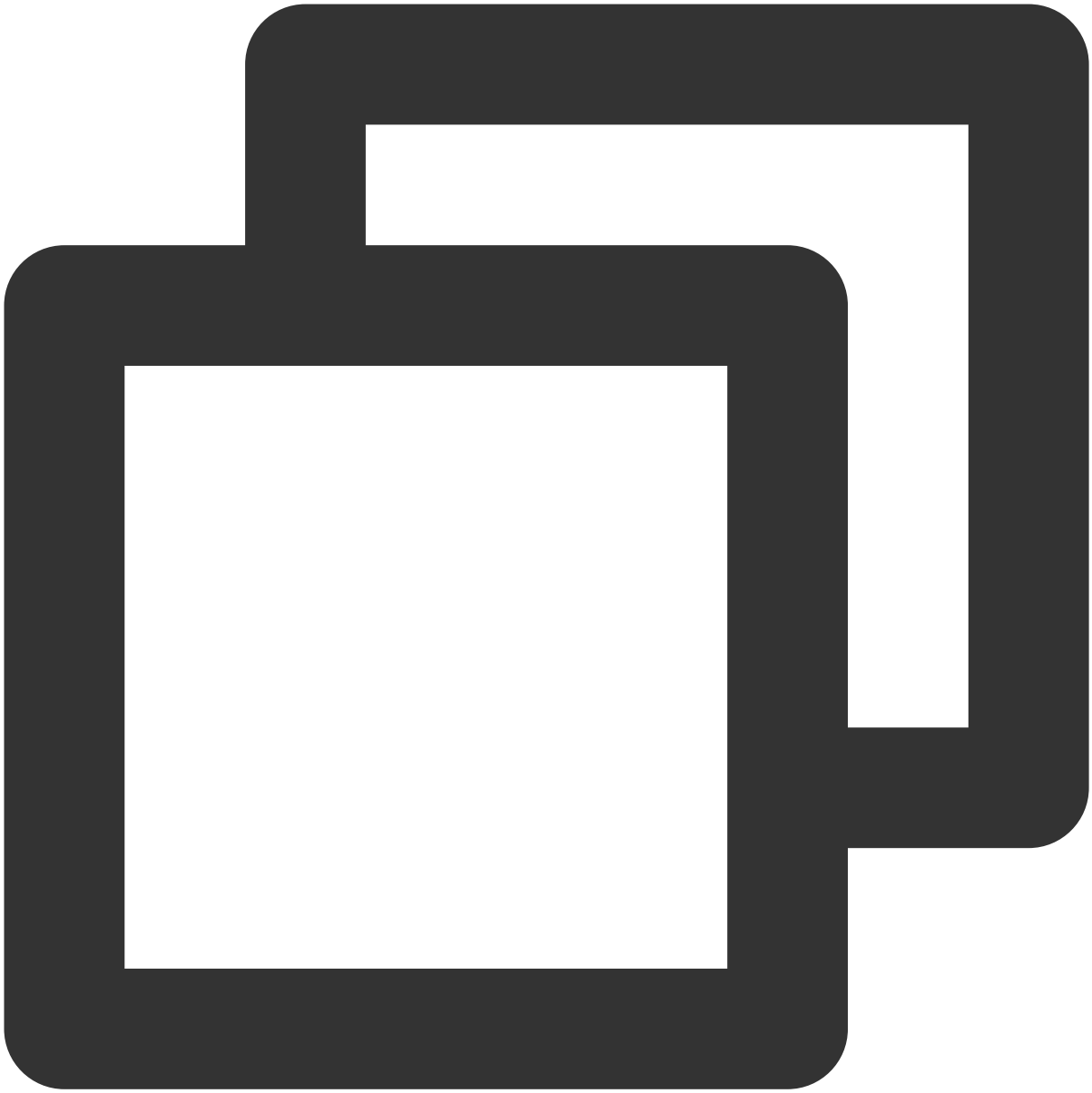
GetSnapshot 接口用于获取媒体文件某个时间的截图。

使用说明

此接口为同步模式，异步模式请参考 任务接口-截图任务。

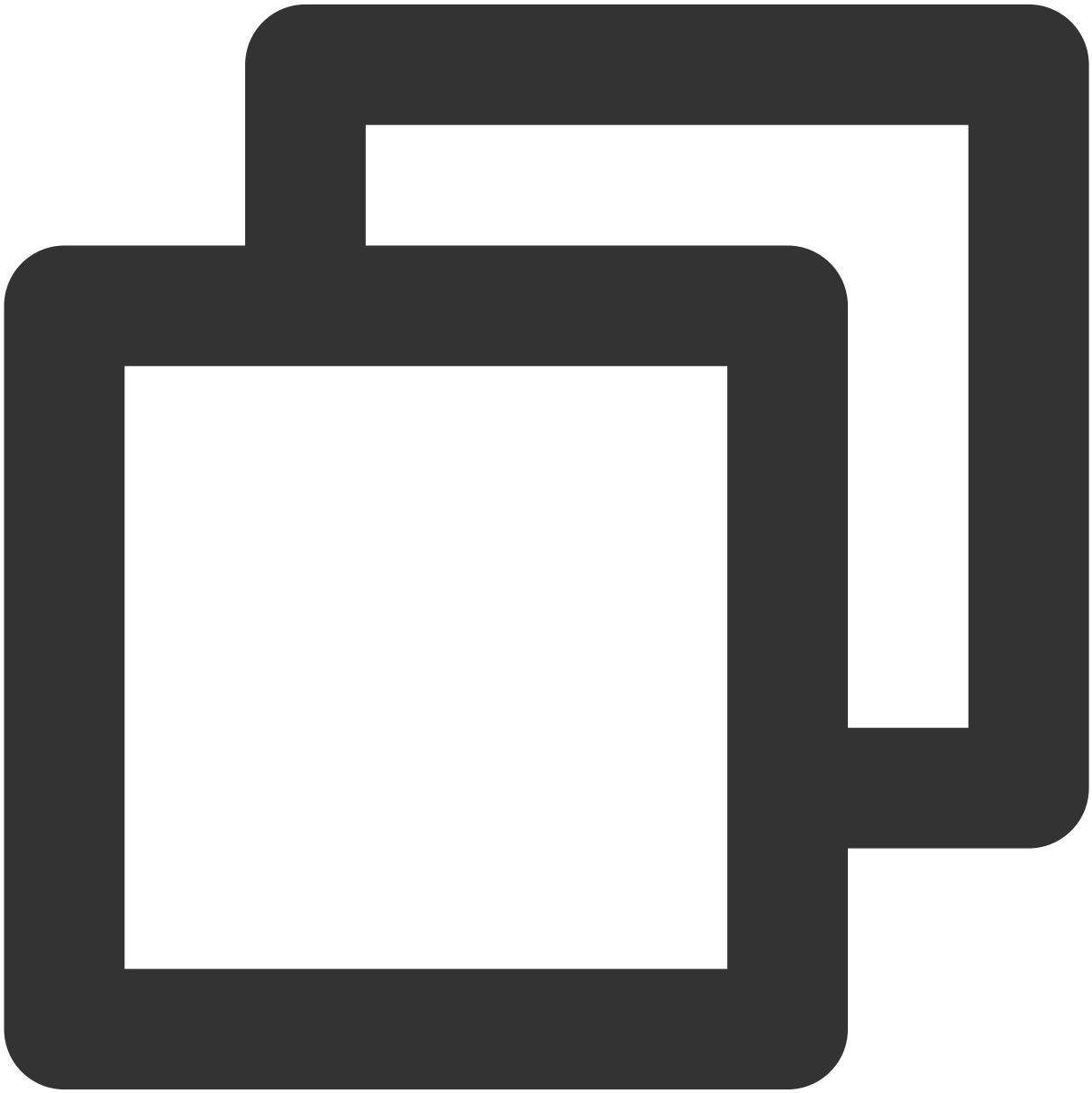
请求示例

1. 通过GET请求方式



```
GET /for-test.mp4?ci-process=snapshot&time=1&format=jpg HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
```

2. 通过URL地址添加参数方式



```
https://<BucketName-APPID>.cos.<Region>.myqcloud.com/for-test.mp4?ci-process=snapsh
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

参数说明如下：

参数名称	描述	类型	是否必选
ci-process	操作类型，固定使用 snapshot	String	是
time	截图的时间点，单位为秒	float	是
width	截图的宽。默认为0	Int	否
height	截图的高。默认为0 当 width 和 height 都为0时，表示使用视频的宽高。如果单个为0，则以另外一个值按视频宽高比例自动适应	Int	否
format	截图的格式，支持 jpg 和 png，默认 jpg	String	否
rotate	图片旋转方式 auto：按视频旋转信息进行自动旋转 off：不旋转 默认值为 auto	String	否
mode	截帧方式 keyframe：截取指定时间点之前的最近的一个关键帧 exactframe：截取指定时间点的帧 默认值为 exactframe	String	否

请求体

该请求无请求体。

响应

响应头

该响应包含公共响应头，详情请参见 公共响应头部 文档。

响应体

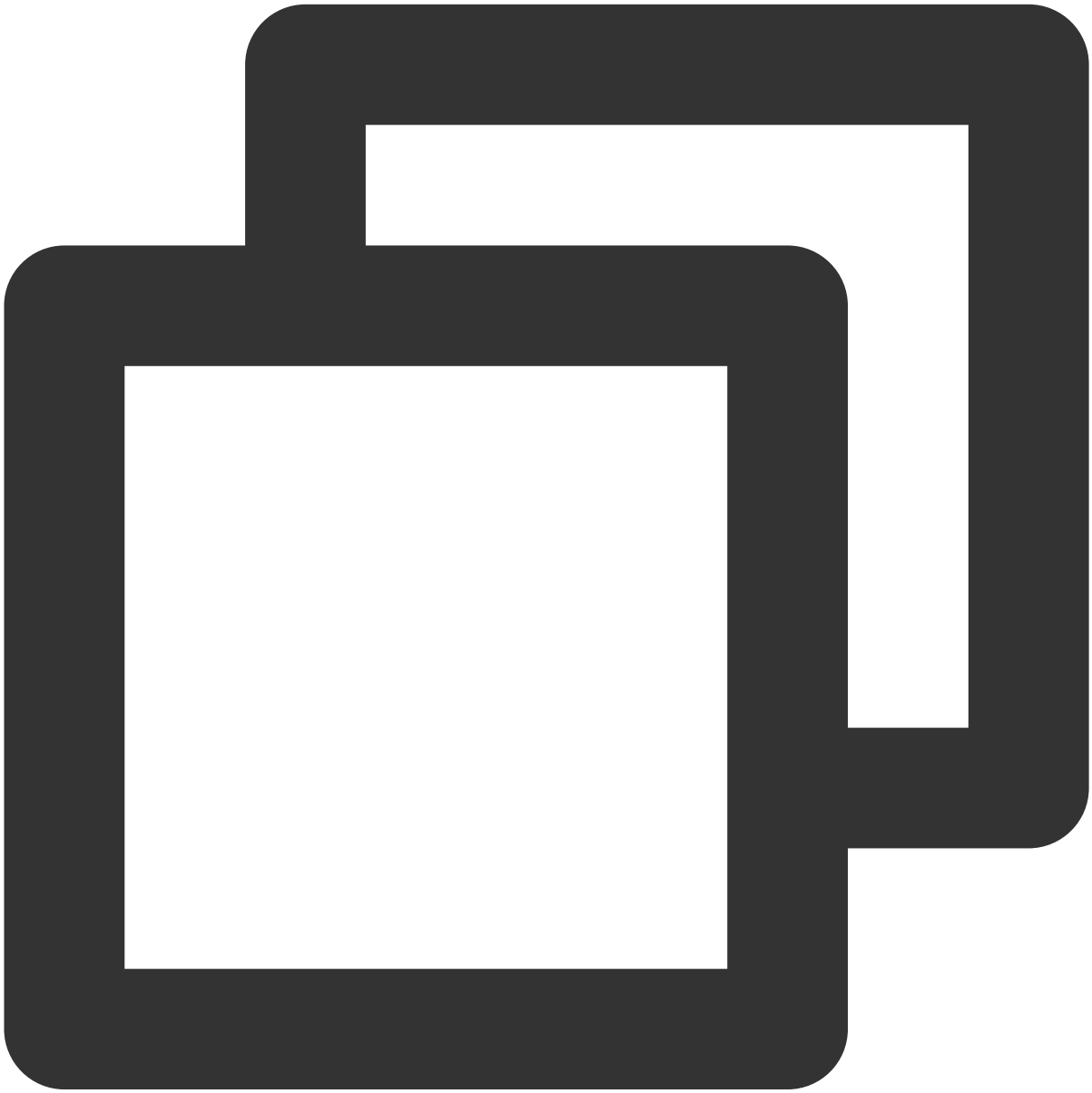
该响应体为截图文件内容。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

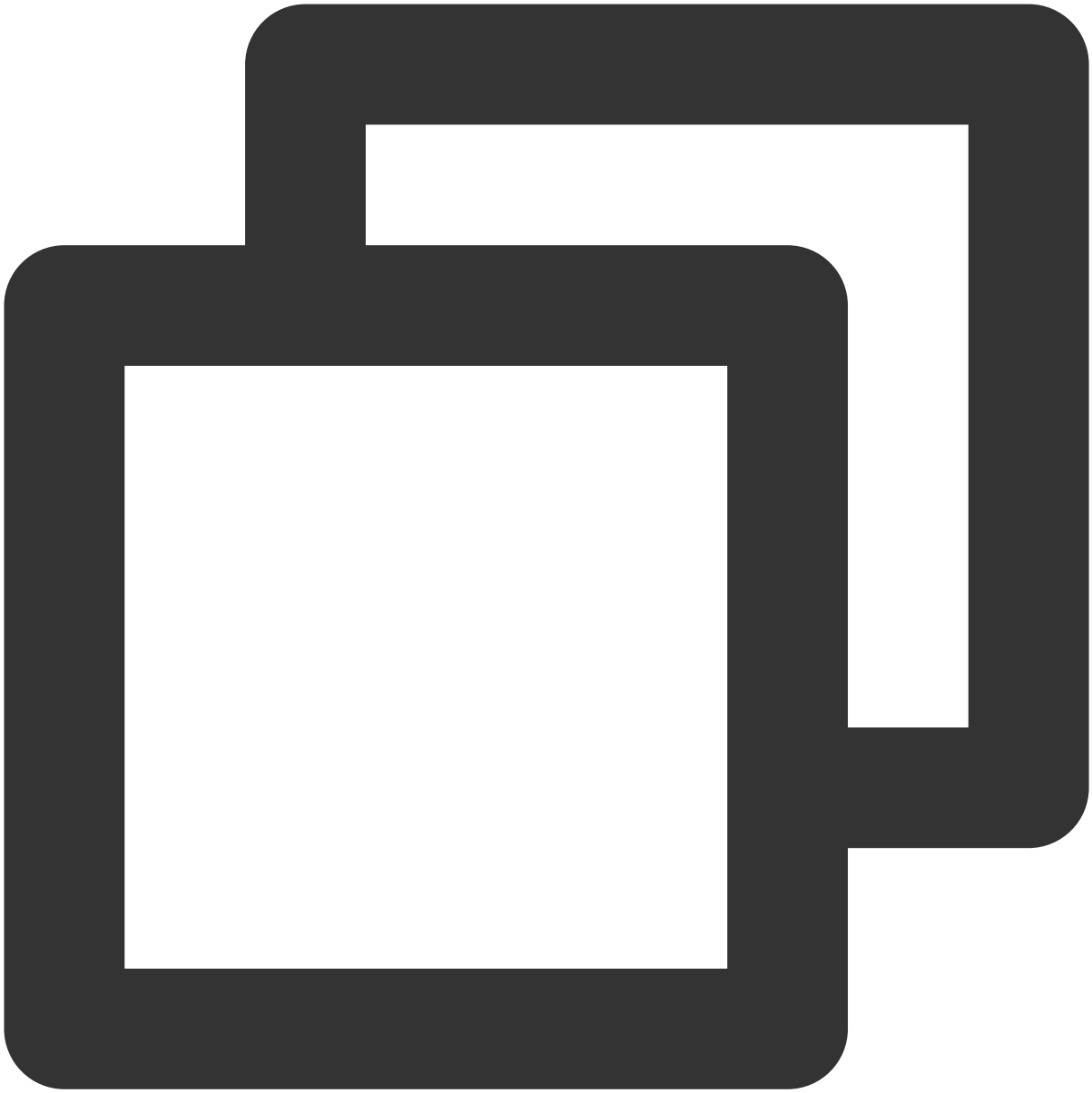
实际案例

请求1 通过GET请求方式



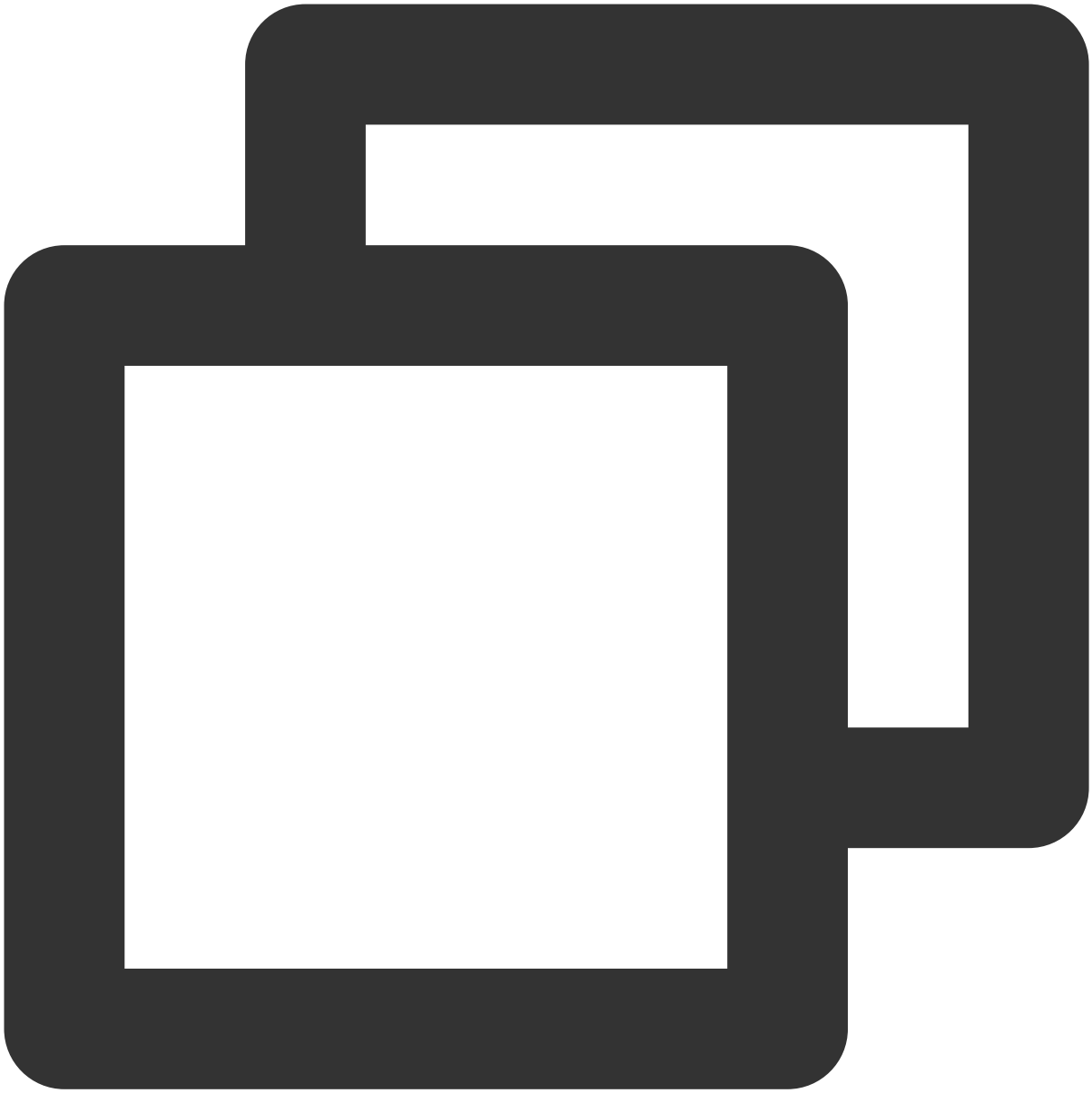
```
GET /for-test.mp4?ci-process=snapshot&time=1.5 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG****-si
Content-Length: 0
```

请求2 通过URL地址添加参数方式



```
https://examplebucket-1250000000.cos.ap-beijing.myqcloud.com/for-test.mp4?ci-proces
```

响应



```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 266005
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==


<图片内容>
```

版权所有：腾讯云计算（北京）有限责任公司

版权所有：腾讯云计算（北京）有限责任公司

模板接口
媒体处理
创建媒体处理模板
创建动图模板

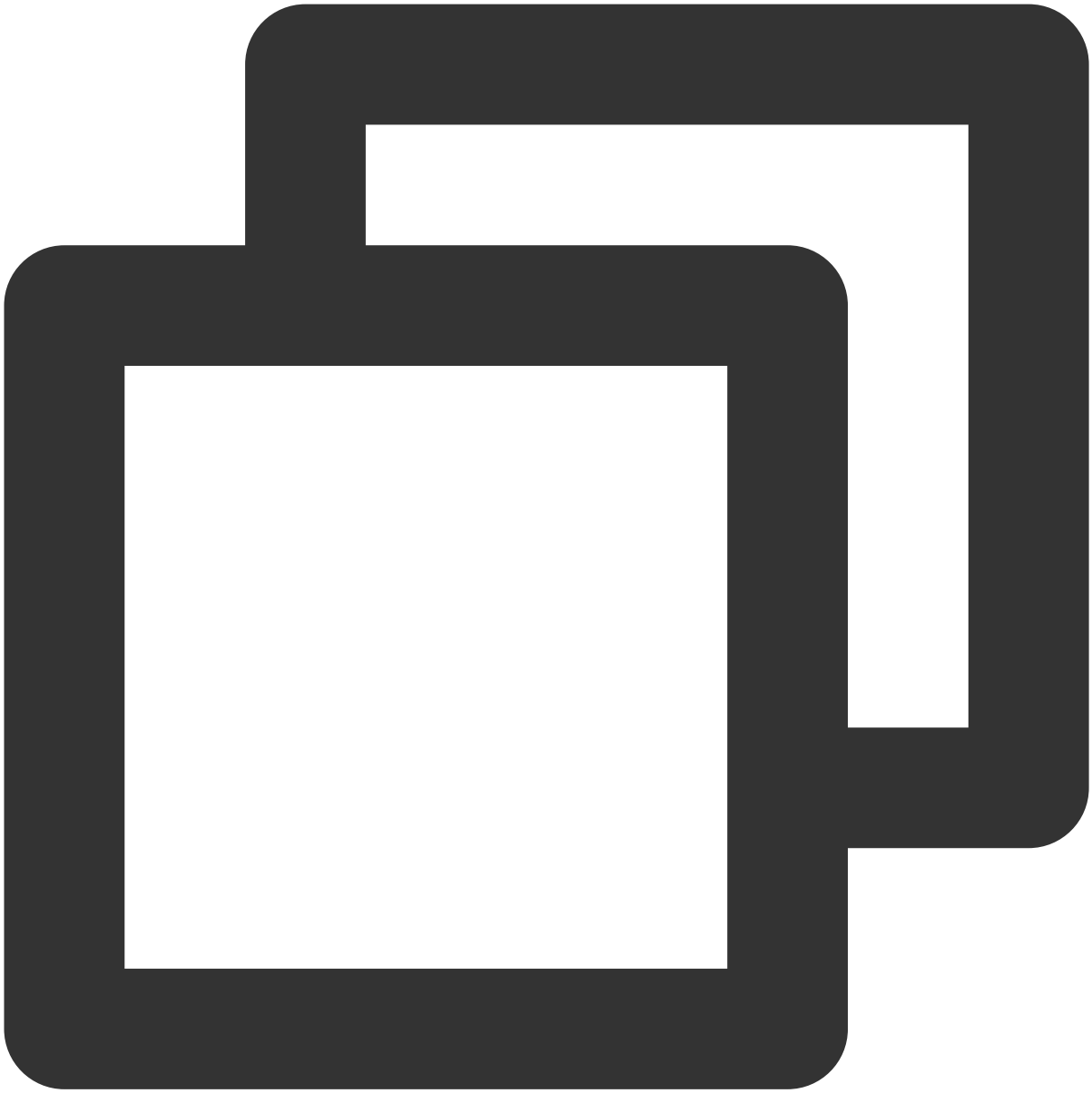
最近更新时间：2024-01-06 10:11:58

功能描述

创建视频转动图模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

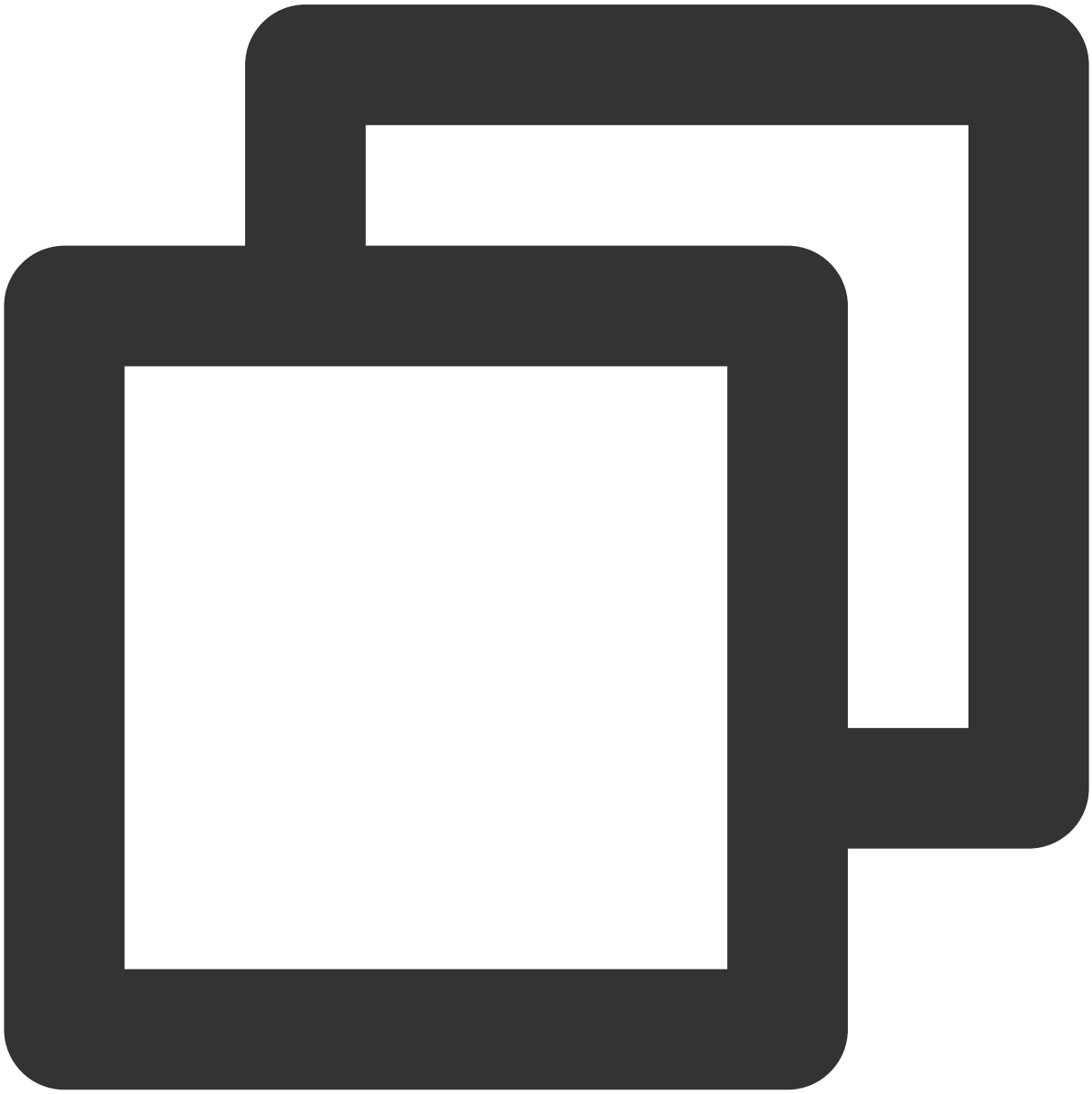
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Animation</Tag>
    <Name>TemplateName</Name>
    <Container>
       <Format>gif</Format>
    </Container>
    <Video>
       <Codec>gif</Codec>
       <Width>1280</Width>
       <Height></Height>
       <Fps>15</Fps>
       <AnimateOnlyKeepKeyFrame>true</AnimateOnlyKeepKeyFrame>
    </Video>
    <TimeInterval>
       <Start>0</Start>
       <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	模板类型：Animation	String	是
Name	Request	模板名称仅支持中文、英文、数字、_、-和*	String	是
Container	Request	容器格式	Container	是
Video	Request	视频信息	Container	否
TimeInterval	Request	时间区间	Container	否

Request 节点 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Format	Request.Container	容器格式：gif，hgif，webp hgif 为高质量	String	是

		gif，即清晰度比较高的 gif 格式图		

Request 节点 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值
Codec	Request.Video	编解码格式	String	是	无
Width	Request.Video	宽	String	否	视频原始宽度
Height	Request.Video	高	String	否	视频原始高度
Fps	Request.Video	帧率	String	否	视频原始帧率
AnimateOnlyKeepKeyFrame	Request.Video	动图只保留关键帧 。若 AnimateOnlyKeepKeyFrame 设置为 true 时，则不考虑 AnimateTimeIntervalOfFrame、AnimateFramesPerSecond；若 AnimateOnlyKeepKeyFrame 设置为 false 时，则必须填写 AnimateTimeIntervalOfFrame 或 AnimateFramesPerSecond	String	否	false
AnimateTimeIntervalOfFrame	Request.Video	动图抽帧间隔时间	String	否	无
AnimateFramesPerSecond	Request.Video	Animation 每秒抽帧帧数	String	否	无

版权所有：腾讯云计算（北京）有限责任公司

Quality	Request.Video	设置相对质量	String	否	无

Request 节点 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.TimeInterval	开始时间	String	否	0	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒
Duration	Request.TimeInterval	持续时间	String	否	视频时长	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒

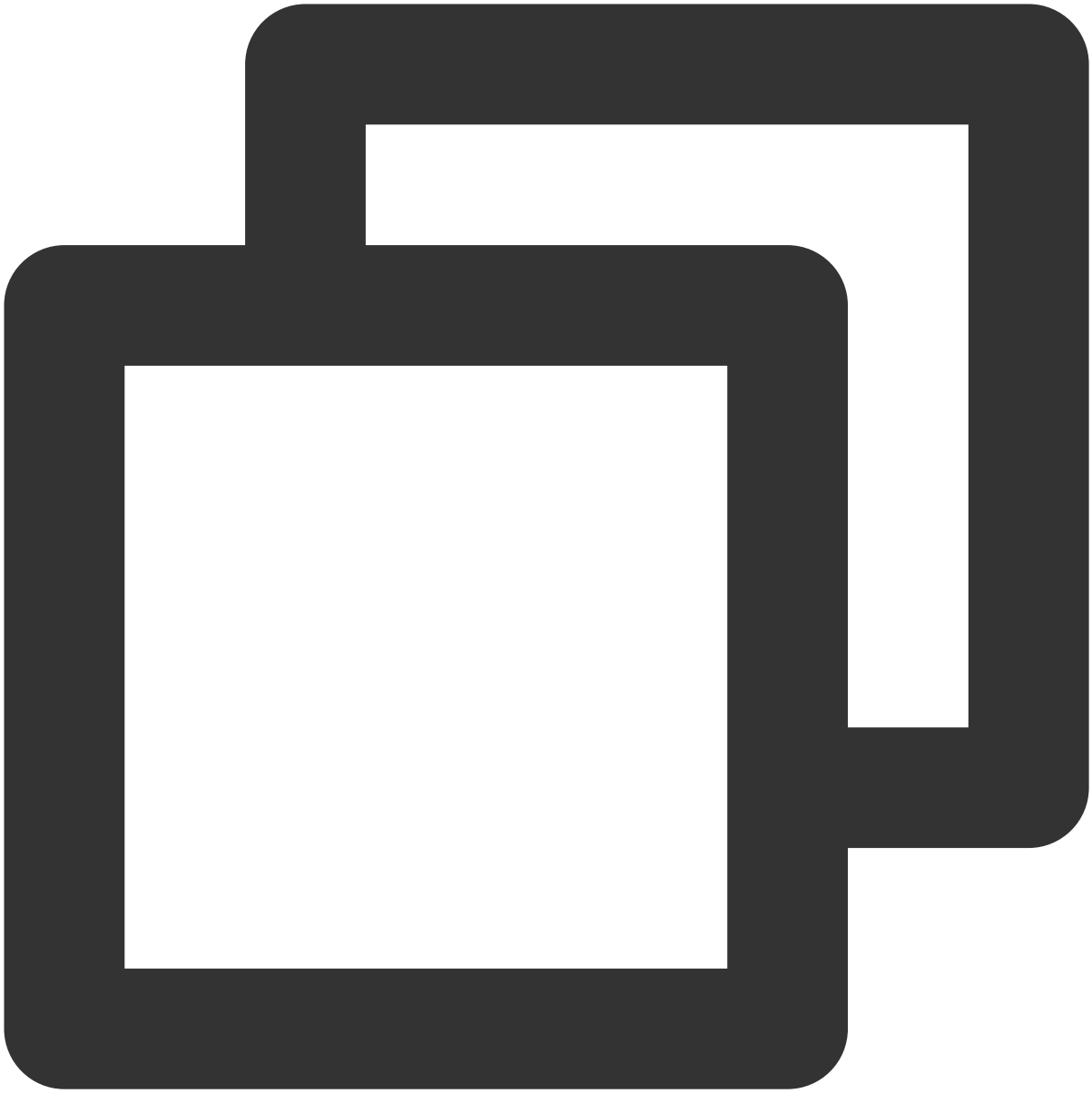
响应

响应头

该响应包含公共响应头，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Tag>Animation</Tag>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TransTpl>
            <Container>
                <Format>gif</Format>
            </Container>
```

```
            <Video>
                <Codec>gif</Codec>
                <Width>1280</Width>
                <Height></Height>
                <Fps>15</Fps>
                <AnimateOnlyKeepKeyFrame>true</AnimateOnlyKeepKeyFrame>
            </Video>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Response 的具体描述：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response	模板 ID	String
Name	Response	模板名字	String
BucketId	Response	模板所属存储桶	String
Category	Response	模板属性，Custom 或者 Official	String
Tag	Response	模板 Tag	String
UpdateTime	Response	更新时间	String
CreateTime	Response	创建时间	String
TransTpl	Response	详细的模板参数	Container

TransTpl 节点 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型

Format	Response.TransTpl.Container	容器格式：gif，hgif，webp hgif 为高质量 gif，即清晰度比较高的 gif 格式图	String

TransTpl 节点 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Codec	Response.TransTpl.Video	编解码格式	String
Width	Response.TransTpl.Video	宽	String
Height	Response.TransTpl.Video	高	String
Fps	Response.TransTpl.Video	帧率	String
AnimateOnlyKeepKeyFrame	Response.TransTpl.Video	动图只保留关键帧	String
AnimateTimeIntervalOfFrame	Response.TransTpl.Video	动图抽帧间隔时间	String
AnimateFramesPerSecond	Response.TransTpl.Video	Animation 每秒抽帧帧数	String
Quality	Response.TransTpl.Video	设置相对质量	String

TransTpl 节点 TimeInterval 的具体数据描述如下：

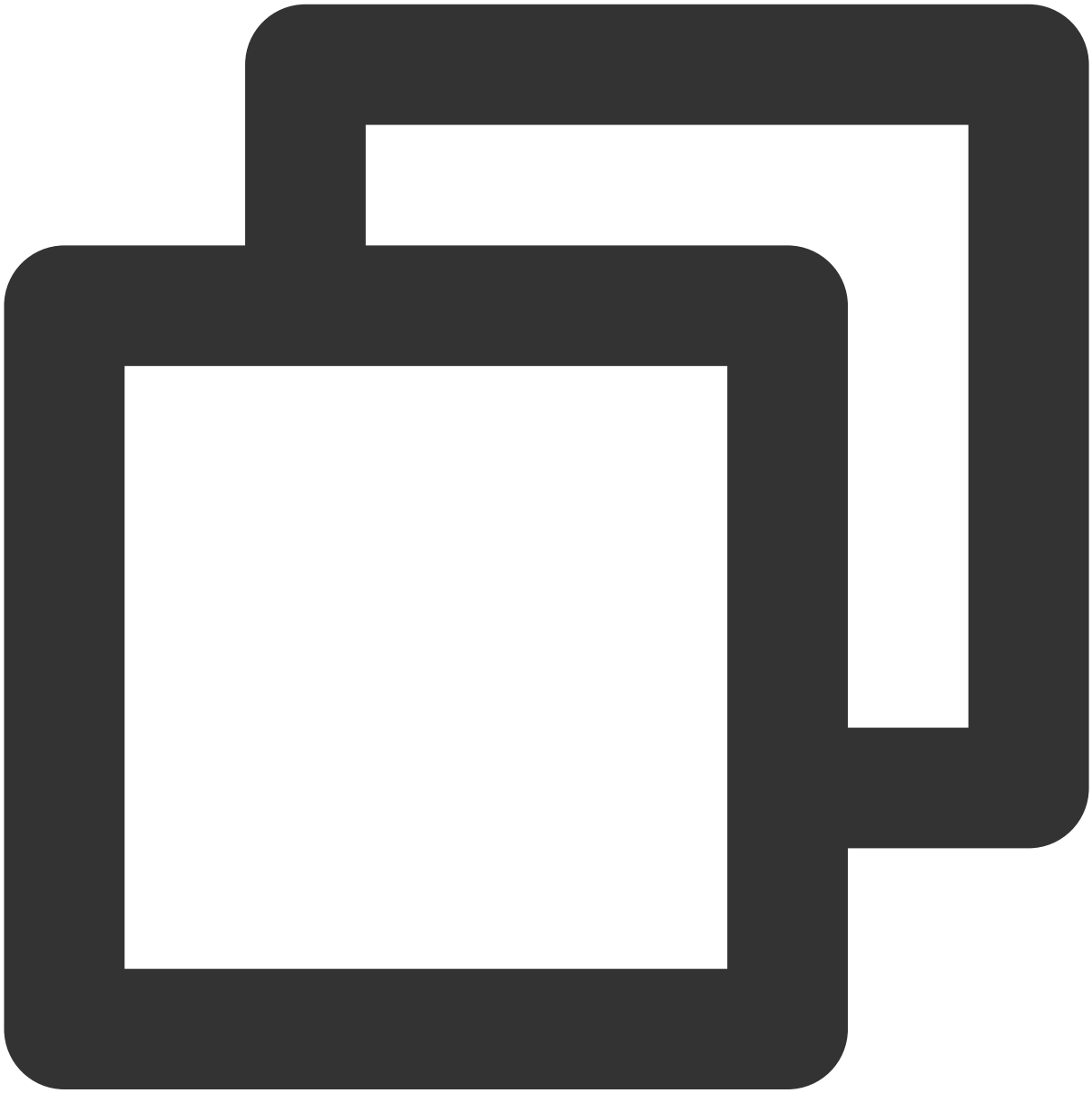
节点名称（关键字）	父节点	描述	类型
Start	Response.TransTpl.TimeInterval	开始时间	String
Duration	Response.TransTpl.TimeInterval	持续时间	String

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

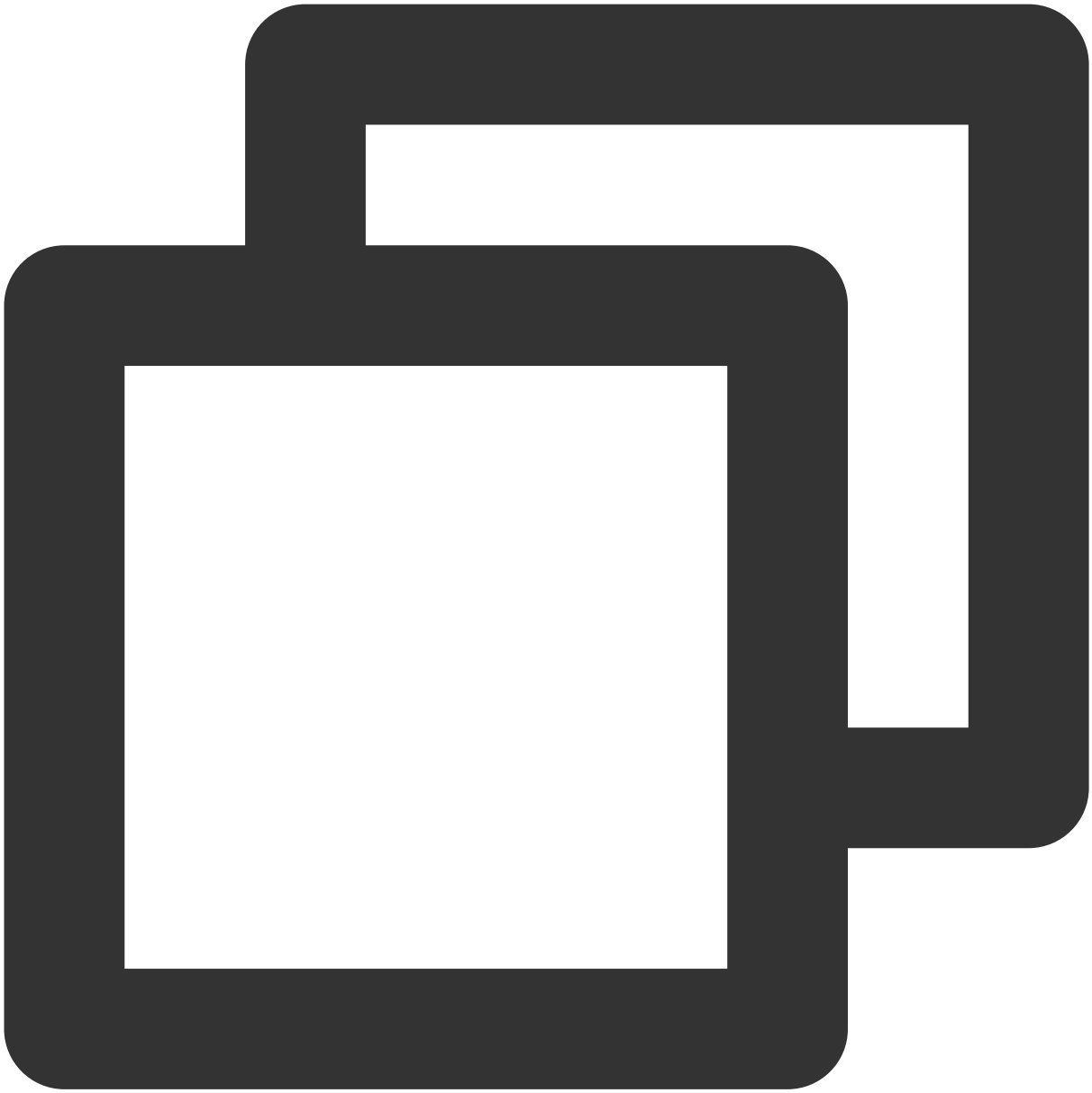
请求



```
POST /template HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Animation</Tag>
    <Name>TemplateName</Name>
    <Container>
       <Format>gif</Format>
```

```
    </Container>
    <Video>
        <Codec>gif</Codec>
        <Width>1280</Width>
        <Height></Height>
        <Fps>15</Fps>
        <AnimateOnlyKeepKeyFrame>true</AnimateOnlyKeepKeyFrame>
    </Video>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYz****=

<Response>
    <Template>
        <Name>TemplateName</Name>
```

```
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Tag>Animation</Tag>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TransTpl>
            <Container>
                <Format>gif</Format>
            </Container>
            <Video>
                <Codec>gif</Codec>
                <Width>1280</Width>
                <Height></Height>
                <Fps>15</Fps>
                <AnimateOnlyKeepKeyFrame>true</AnimateOnlyKeepKeyFrame>
            </Video>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

创建拼接模板

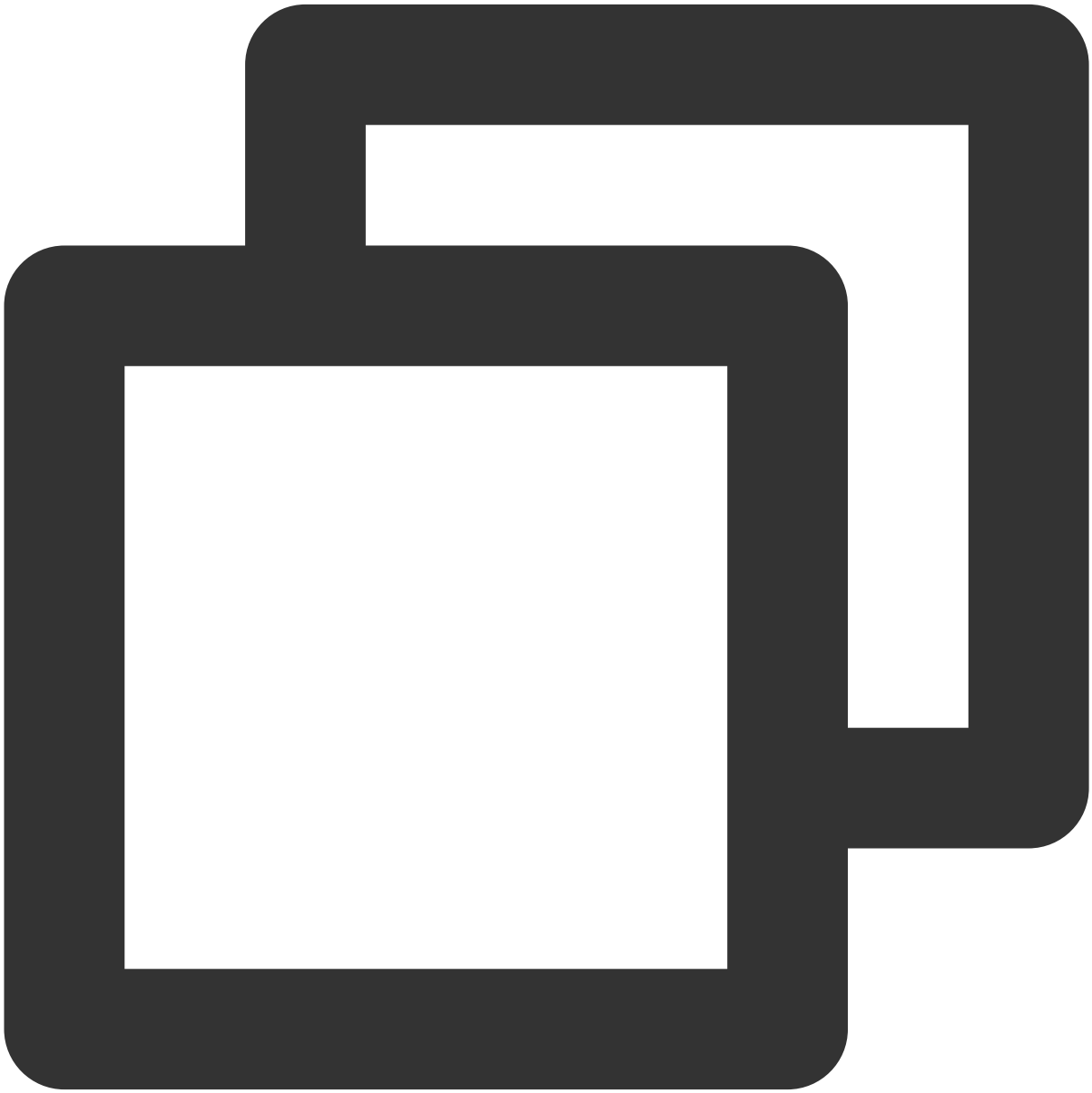
最近更新时间：2024-01-06 10:11:58

功能描述

创建拼接模板。

请求

请求实例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

公共头部

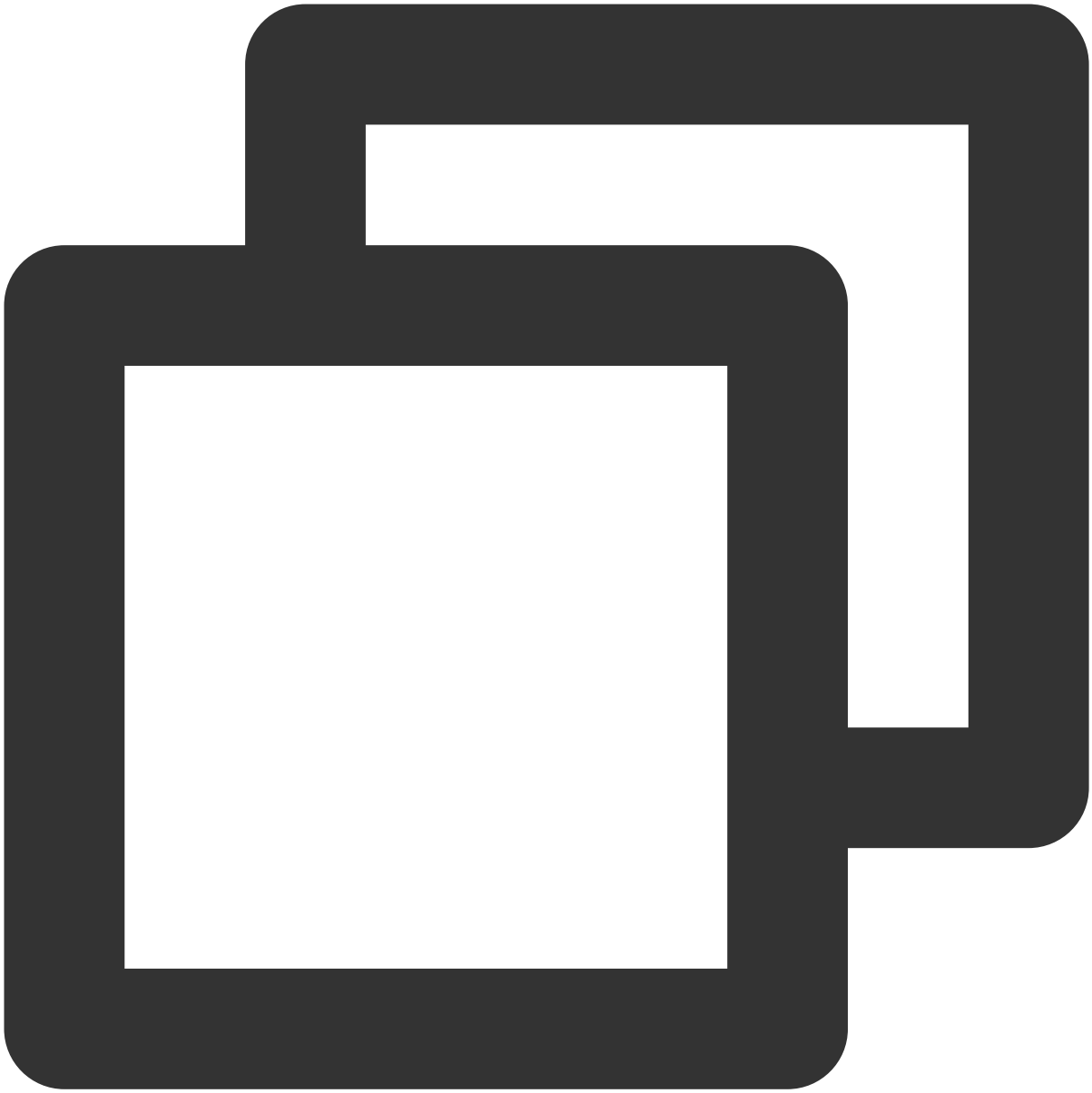
该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

非公共头部

该请求操作无特殊的请求头部信息。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
    <Tag>Concat</Tag>
    <Name>TemplateName</Name>
    <ConcatTemplate>
        <ConcatFragment>
            <Mode>Start</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4</Ur
        </ConcatFragment>
        <ConcatFragment>
            <Mode>End</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</Url>
```

```
        </ConcatFragment>
        <Audio>
            <Codec>mp3</Codec>
            <Samplerate></Samplerate>
            <Bitrate></Bitrate>
            <Channels></Channels>
        </Audio>
        <Video>
            <Codec>H.264</Codec>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Height></Height>
            <Fps>30</Fps>
        </Video>
        <Container>
            <Format>mp4</Format>
        </Container>
        <AudioMix>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</Au
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMix>
    </ConcatTemplate>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Tag	Request	模板类型：Concat	String	是
Name	Request	模板名称仅支持中文、英文、数字、_、-和*	String	是

ConcatTemplate	Request	拼接模板	Container	是

Container 类型 ConcatTemplate 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
ConcatFragment	Request.ConcatTemplate	拼接节点	Container 数组	是	无	无
Audio	Request.ConcatTemplate	音频参数	Container	否	媒体原始值	目标文件不需要 Audio 信息，需要设置 Audio.Remove 为 true
Video	Request.ConcatTemplate	视频参数	Container	否	媒体原始值	目标文件不需要 Video 信息，需要设置 Video.Remove 为 true
Container	Request.ConcatTemplate	封装格式	Container	是	无	无
AudioMix	Request.ConcatTemplate	混音参数, 详情见 AudioMix	Container 数组	否	无	仅在 Audio.Remove 为 false 时生效
DirectConcat	Request.ConcatTemplate	只拼接不转码	String	否	false	true/ false

Container 数组 类型 ConcatFragment 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
Url	Request.ConcatTemplate.ConcatFragment	拼接对象地址	String	是	无	同 bucket 对象文件
Mode	Request.ConcatTemplate.ConcatFragment	节点类型	String	是	无	Start：开头 End：结尾

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Audio 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
Codec	Request.ConcatTemplate.Audio	编解码格式	String	是	文件原编码	取值 aac、mp3
Samplerate	Request.ConcatTemplate.Audio	采样率	String	否	文件原采样率	单位：Hz 可选 11025、22050、32000、44100、48000、96000 不同的封装，mp3 支持不同的采样率，如下表所示
Bitrate	Request.ConcatTemplate.Audio	音频码率	String	否	文件原音频码率	单位：Kbps 值范围：[8, 1000]
Channels	Request.ConcatTemplate.Audio	声道数	String	否	文件原声道数	当 Codec 设置为 aac，支持1、2、4、5、6、8 当 Codec 设置为 mp3，支持1、2

Y 表示支持这种采样率，N 表示不支持。

封装格式/音频采样率	11025	22050	32000	44100	48000	96000
mp3	Y	Y	Y	Y	Y	N

Container 类型 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Format	Request.ConcatTemplate.Container	容器格式：mp4，flv，hls，ts，mp3, aac	String	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
Codec	Request.ConcatTemplate.Video	编解码格式	String	是	H.264	H.264
Width	Request.ConcatTemplate.Video	宽	String	否	视频原始宽度	值范围：[128, 4096] 单位：px 若只设置 Width 时，按照视频原始比例计算 Height
Height	Request.ConcatTemplate.Video	高	String	否	视频原始高度	值范围：[128, 4096] 单位：px 若只设置 Height 时，按照视频原始比例计算 Width
Fps	Request.ConcatTemplate.Video	帧率	String	否	视频原始帧率	值范围：(0，60] 单位：fps
Bitrate	Request.ConcatTemplate.Video	视频输出文件的码率	String	否	视频原始码率	值范围：(0，60] 单位：fps
Crf	Request.ConcatTemplate.Video	码率-质量控制因子	String	否	视频原始码率	值范围：(0, 51] 如果设置了 Crf，则 Bitrate 的设置失效 当 Bitrate 为空时，默认为25
Remove	Request.ConcatTemplate.Video	是否删除视频流	String	否	false	取值 true、false
Rotate	Request.Video	旋转角度	String	否	无	1. 值范围：[0, 360) 2. 单位：度

版权所有：腾讯云计算（北京）有限责任公司

响应

响应头

公共响应头

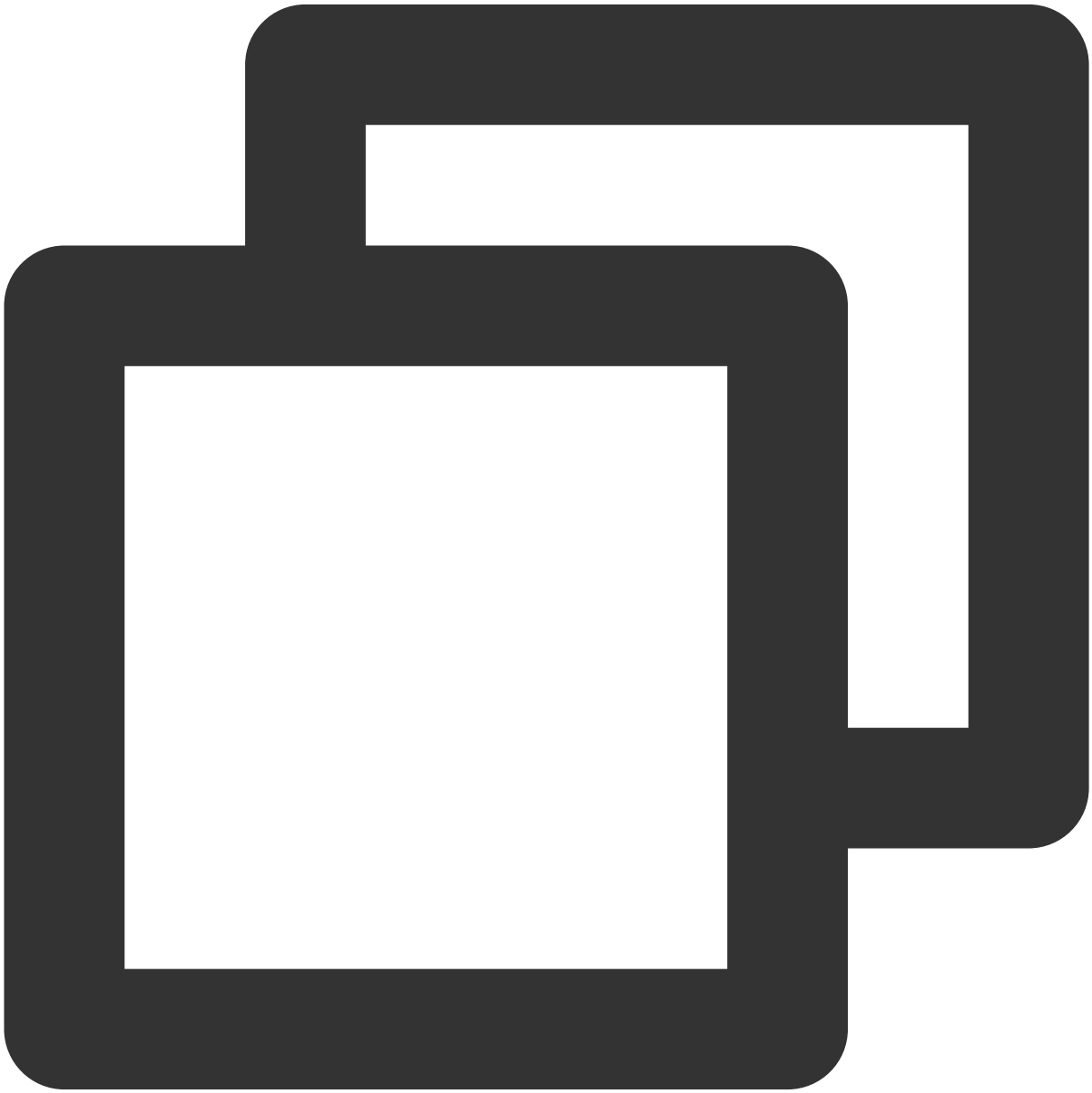
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

特有响应头

该响应无特殊的响应头。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Tag>Concat</Tag>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <ConcatTemplate>
            <ConcatFragment>
                <Mode>Start</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4
```

```
            </ConcatFragment>
            <ConcatFragment>
                <Mode>End</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</
            </ConcatFragment>
            <Audio>
                <Codec>mp3</Codec>
                <Samplerate></Samplerate>
                <Bitrate></Bitrate>
                <Channels></Channels>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Height></Height>
                <Fps>30</Fps>
            </Video>
            <Container>
                <Format>mp4</Format>
            </Container>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
                <EffectConfig>
                    <EnableStartFadein>true</EnableStartFadein>
                    <StartFadeinTime>3</StartFadeinTime>
                    <EnableEndFadeout>false</EnableEndFadeout>
                    <EndFadeoutTime>0</EndFadeoutTime>
                    <EnableBgmFade>true</EnableBgmFade>
                    <BgmFadeTime>1.7</BgmFadeTime>
                </EffectConfig>
            </AudioMix>
        </ConcatTemplate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

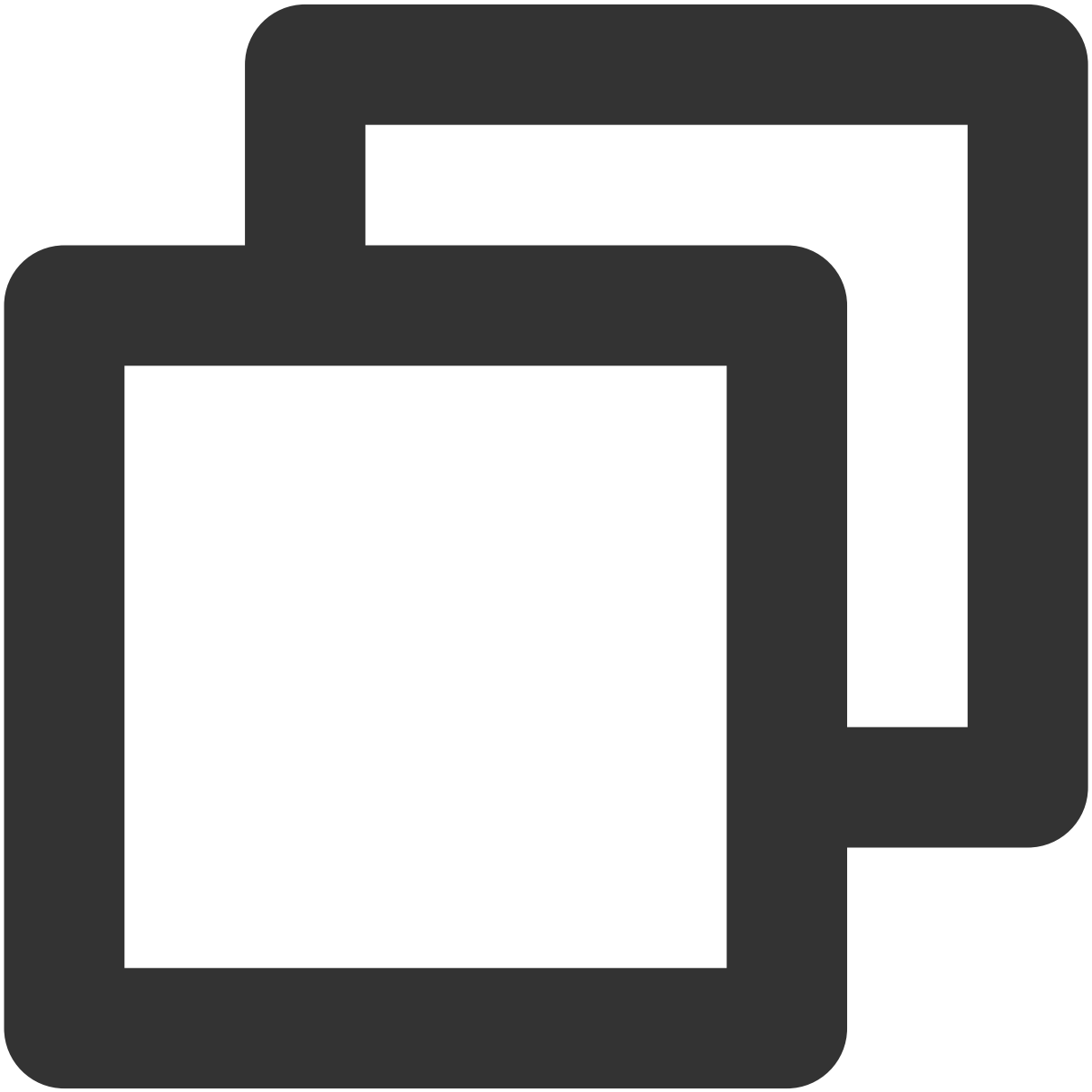
节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，Concat	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
ConcatTemplate	Response.Template	同请求体中的 Request.ConcatTemplate	Container

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

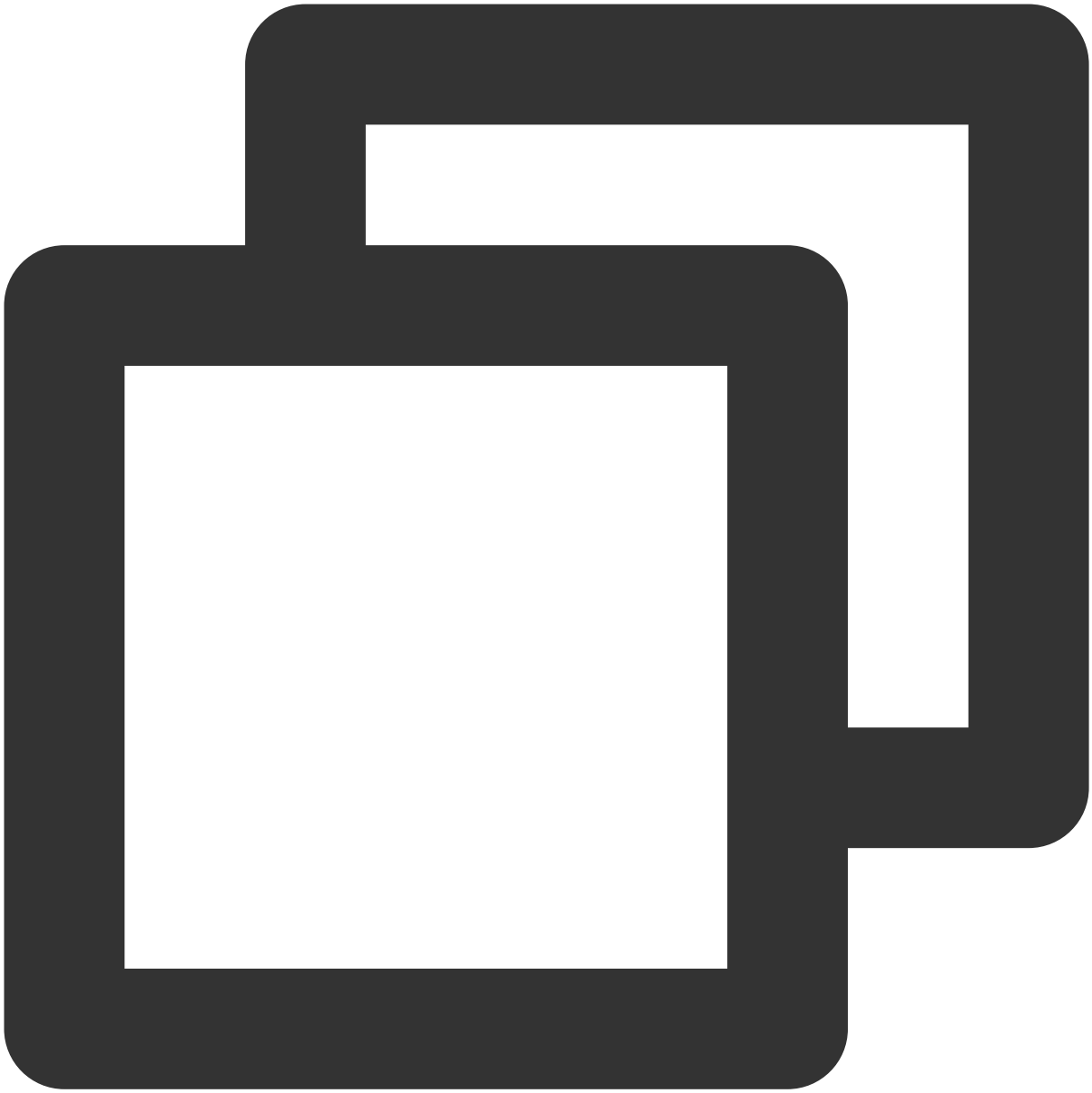
请求



```
POST /template HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
   <Tag>Concat</Tag>
   <Name>TemplateName</Name>
   <ConcatTemplate>
        <ConcatFragment>
```

```
            <Mode>Start</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4</Ur
        </ConcatFragment>
        <ConcatFragment>
            <Mode>End</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</Url>
        </ConcatFragment>
        <Audio>
            <Codec>mp3</Codec>
            <Samplerate></Samplerate>
            <Bitrate></Bitrate>
            <Channels></Channels>
        </Audio>
        <Video>
            <Codec>H.264</Codec>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Height></Height>
            <Fps>30</Fps>
        </Video>
        <Container>
            <Format>mp4</Format>
        </Container>
        <AudioMix>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</Au
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMix>
    </ConcatTemplate>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Tag>Concat</Tag>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <ConcatTemplate>
            <ConcatFragment>
                <Mode>Start</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4
            </ConcatFragment>
            <ConcatFragment>
                <Mode>End</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</
            </ConcatFragment>
            <Audio>
                <Codec>mp3</Codec>
                <Samplerate></Samplerate>
                <Bitrate></Bitrate>
                <Channels></Channels>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Height></Height>
                <Fps>30</Fps>
            </Video>
            <Container>
                <Format>mp4</Format>
            </Container>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
                <EffectConfig>
                    <EnableStartFadein>true</EnableStartFadein>
                    <StartFadeinTime>3</StartFadeinTime>
                    <EnableEndFadeout>false</EnableEndFadeout>
                    <EndFadeoutTime>0</EndFadeoutTime>
                    <EnableBgmFade>true</EnableBgmFade>
                    <BgmFadeTime>1.7</BgmFadeTime>
                </EffectConfig>
            </AudioMix>
        </ConcatTemplate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

创建极速高清转码模板

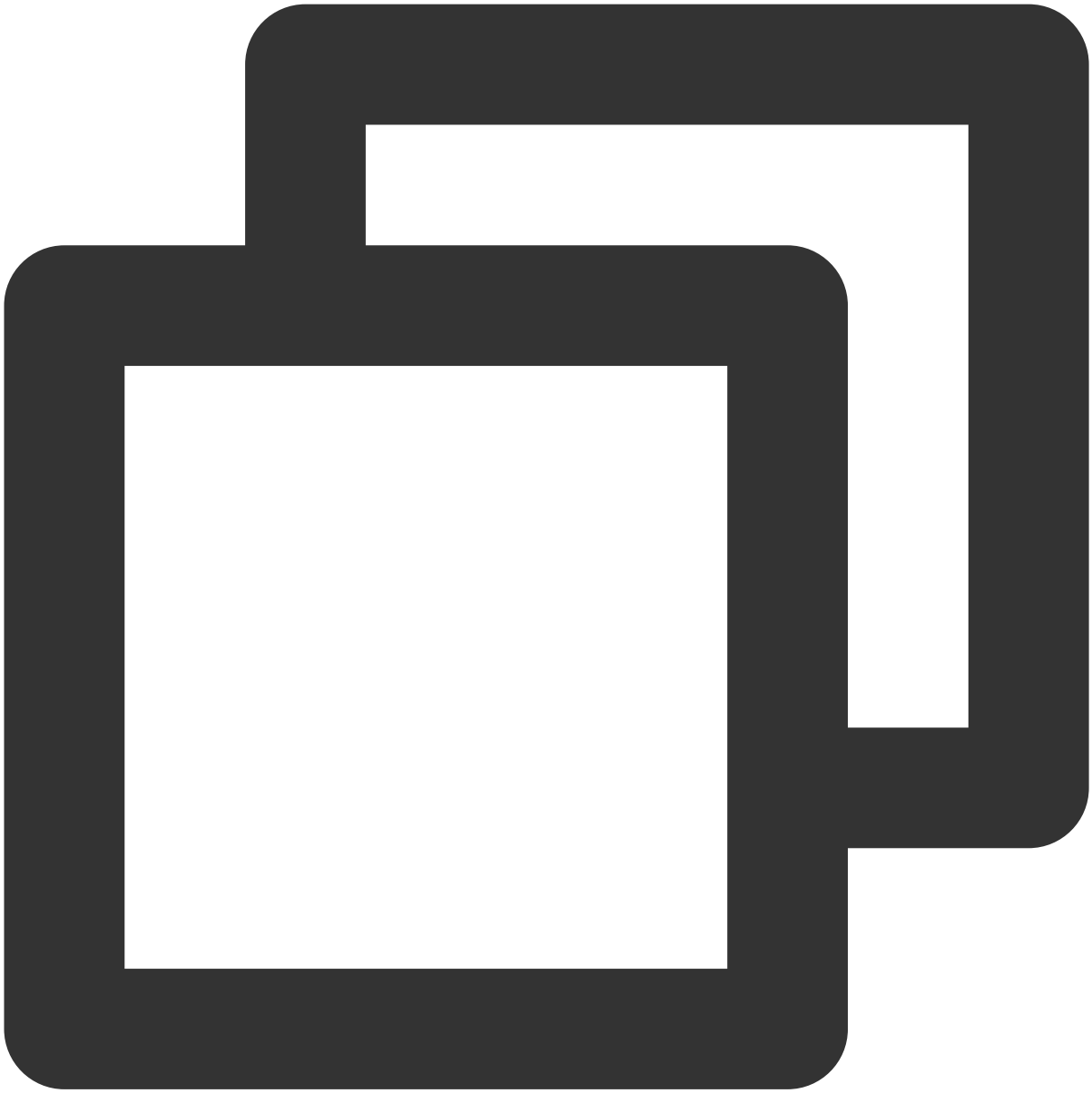
最近更新时间：2024-01-06 10:11:58

功能描述

创建极速高清转码模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

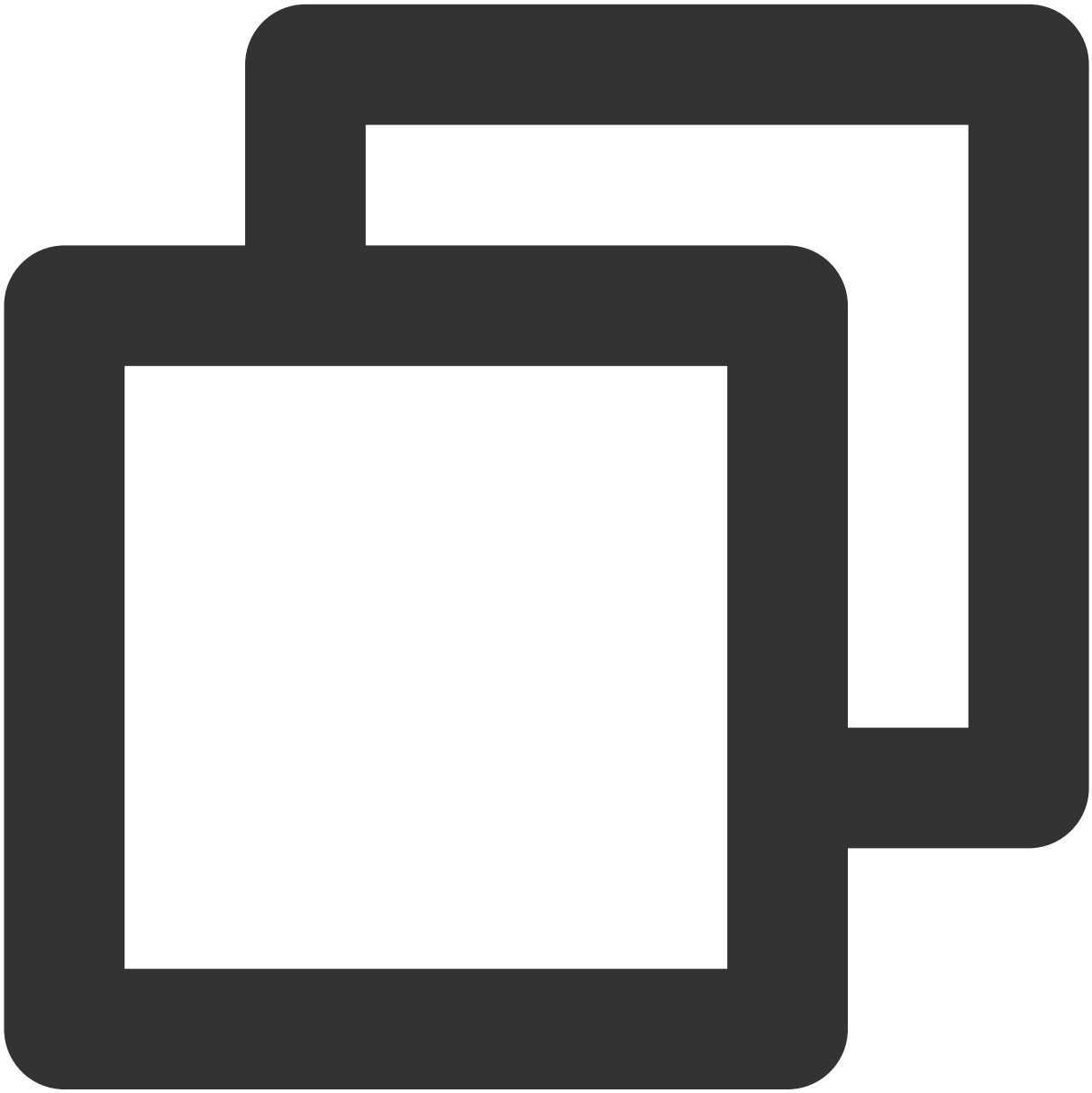
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>HighSpeedHd</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Bufsize>0</Bufsize>
        <Maxrate>0</Maxrate>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型: HighSpeedHd	String	是	无
Name	Request	模板名称。仅支持中	String	是	无

		文、英文、数字、_、-和*			
Container	Request	容器格式	Container	是	无
Video	Request	视频信息	Container	是	无
TimeInterval	Request	时间区间	Container	否	无
Audio	Request	音频信息	Container	否	无
TransConfig	Request	转码配置	Container	否	无

Container 类型 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Format	Request.Container	容器格式: mp4，flv，hls，mkv	String	是
ClipConfig	Request.Container	分片配置, 当Format为hls时有效	Container	否

Container 类型 ClipConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Duration	Request.Container.ClipConfig	分片时长, 默认5s	String	否

设定 container，音频视频支持的格式如下表：

Container	Audio Codecs	Video Codecs
mp4/hls/mkv	AAC、MP3	H.264、H.265
flv	AAC、MP3	H.264

Container 类型 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Video	编解码格式	String	否	H.264	H.264、H.265
Width	Request.Video	宽	String	否	视频原始	1. 值范围：[128，4096]

版权所有：腾讯云计算（北京）有限责任公司

					宽度	2. 单位：px 3. 若只设置 Width 时，按照视频原始比例计算 Height 4. 必须为偶数
Height	Request.Video	高	String	否	视频原始高度	1. 值范围：[128，4096] 2. 单位：px 3. 若只设置 Height 时，按照 视频原始比例计算 Width 4. 必须为偶数
Fps	Request.Video	帧率	String	否	无	1. 值范围：(0，60] 2. 单位：fps
Profile	Request.Video	编码级别	String	否	high	1. 支持 baseline、main、high 2. baseline：适合移动设备； 3. main：适合标准分辨率设备； 4. high：适合高分辨率设备; 仅H.264支持此参数。
Bitrate	Request.Video	视频输出文件的码率	String	否	无	1. 值范围：[10，50000] 2. 单位：Kbps
Crf	Request.Video	码率-质量控制因子	String	否	无	1. 值范围：(0，51] 2. 如果设置了 Crf，则 Bitrate 的设置失效 3. 当 Bitrate 为空时，默认为25
Gop	Request.Video	关键帧间最大帧数	String	否	无	值范围：[0, 100000]
Preset	Request.Video	视频算法器预置	String	否	medium	1. 仅H.264支持该参数 2. 取值 veryfast、fast、medium、slow、slower
Bufsize	Request.Video	缓冲区大小	String	否	无	1. 值范围：[1000，128000] 2. 单位：Kb
Maxrate	Request.Video	视频码率峰值	String	否	无	1. 值范围：[10，50000] 2. 单位：Kbps

版权所有：腾讯云计算（北京）有限责任公司

Pixfmt	Request.Video	视频颜色格式	String	否	无	1. H.264支持：yuv420p、auto 2. H.265支持：yuv420p、yuv420p10le、auto
Rotate	Request.Video	旋转角度	String	否	无	1. 值范围：[0, 360) 2. 单位：度

Container 类型 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.TimeInterval	开始时间	String	否	无	1. [0 视频时长] 2. 单位为秒 3. 支持 float 格式，执行精度精确到毫秒
Duration	Request.TimeInterval	持续时间	String	否	无	1. [0 视频时长] 2. 单位为秒 3. 支持 float 格式，执行精度精确到毫秒

Container 类型 Audio 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Audio	编解码格式	String	否	aac	取值 aac、mp3
Samplerate	Request.Audio	采样率	String	否	44100	1. 单位：Hz 2. 可选 11025、22050、32000、44100、48000、96000 3. 不同的封装，mp3 支持不同的采样率，如下表所示
Bitrate	Request.Audio	原始音频码率	String	否	无	1. 单位：Kbps 2. 值范围：[8，1000]
Channels	Request.Audio	声道数	String	否	无	1. 当 Codec 设置为

版权所有：腾讯云计算（北京）有限责任公司

						aac，支持1、2、4、5、6、8 2. 当 Codec 设置为 mp3，支持1、2
Remove	Request.Audio	是否删除源音频流	String	否	false	取值 true、false

说明

Y 表示支持这种采样率，N 表示不支持。

封装格式/音频采样率	11025	22050	32000	44100	48000	96000
flv	N	Y	N	Y	N	N
mp4	N	Y	Y	Y	Y	N
hls/mkv	Y	Y	Y	Y	Y	N

Container 类型 TransConfig 的具体数据描述如下：

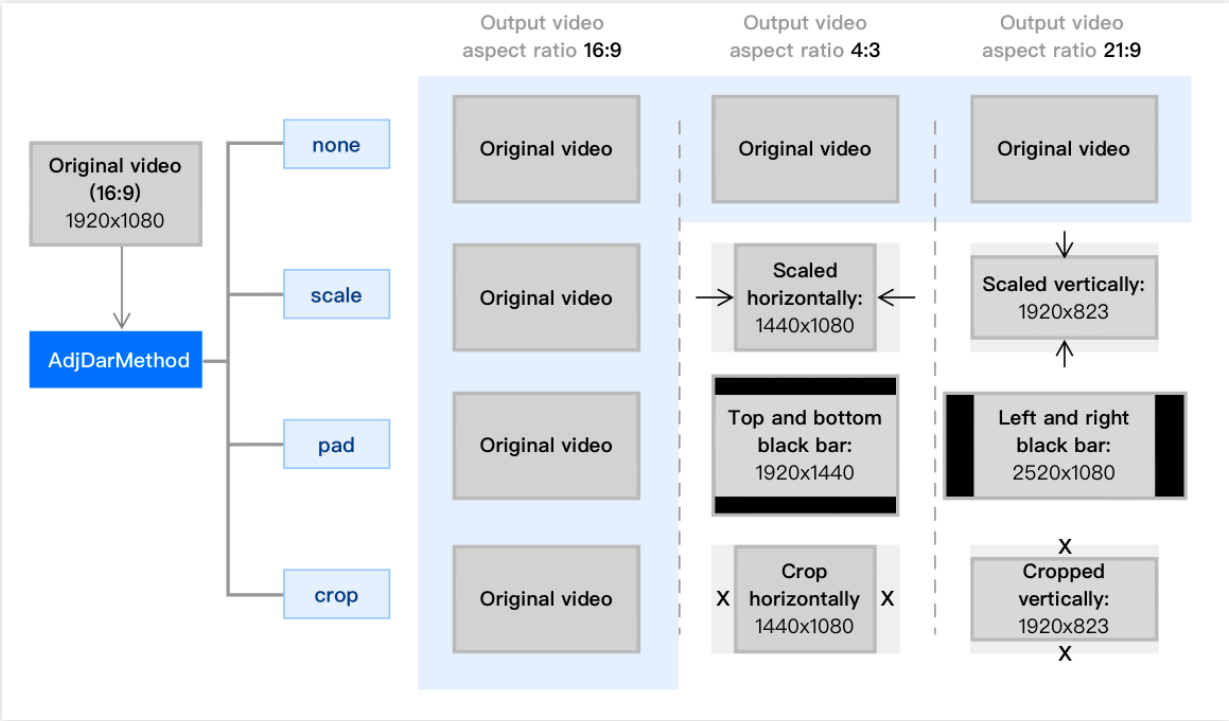
节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
AdjDarMethod	Request.TransConfig	分辨率调整方式	String	否	none	取值 scale、crop pad、none 当输出视频的宽高与原视频不等时，据此参数做分辨率相应调整
IsCheckReso	Request.TransConfig	是否检查分辨率	String	否	false	1. true、false 2. 当为 false 时，照配置参数转码
ResoAdjMethod	Request.TransConfig	分辨率调整方式	String	否	0	1. 取值0、1；0 表使用原视频分辨率1表示返回转码失 2. 当 IsCheckRes为 true 时生效
IsCheckVideoBitrate	Request.TransConfig	是否检查视频码率	String	否	false	true、false

						当为 false 时，按配置参数转码
VideoBitrateAdjMethod	Request.TransConfig	视频码率调整方式	String	否	0	取值0、1；当输视频码率大于原码率时，0表示使原视频码率；1表返回转码失败 当 IsCheckVideoBit为 true 时生效
IsCheckAudioBitrate	Request.TransConfig	是否检查音频码率	String	否	false	true、false 当为 false 时，按配置参数转码
AudioBitrateAdjMethod	Request.TransConfig	音频码率调整方式	String	否	0	取值0、1；当输音频码率大于原码率时，0 表示使原音频码率；1表返回转码失败 当 IsCheckAudioBit为 true 时生效
IsCheckVideoFps	Request.TransConfig	是否检查视频帧率	String	否	false	true、false 当为 false 时，按配置参数转码
VideoFpsAdjMethod	Request.TransConfig	视频帧率调整方式	String	否	0	取值0、1；当输视频帧率大于原帧率时，0表示使原视频帧率；1表返回转码失败 当 IsCheckVideo为 true 时生效
DeleteMetadata	Request.TransConfig	是否删除文件中的 MetaData 信息	String	否	false	1. true、false 2. 当为 false 时，留源文件信息
IsHdr2Sdr	Request.TransConfig	是否开启 HDR 转 SDR	String	否	false	true/false
HlsEncrypt	Request.TransConfig	hls 加密	Container	否	无	无

版权所有：腾讯云计算（北京）有限责任公司

		配置				

AdjDarMethod 参数图示：



Container 类型 HlsEncrypt 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
IsHlsEncrypt	Request.TransConfig.HlsEncrypt	是否开启HLS加密	String	否	false	1. true/false 2. 当Container.Format为hls时支持加密
UriKey	Request.TransConfig.HlsEncrypt	HLS加密的key	String	否	无	当IsHlsEncrypt为true时，该参数才有意义

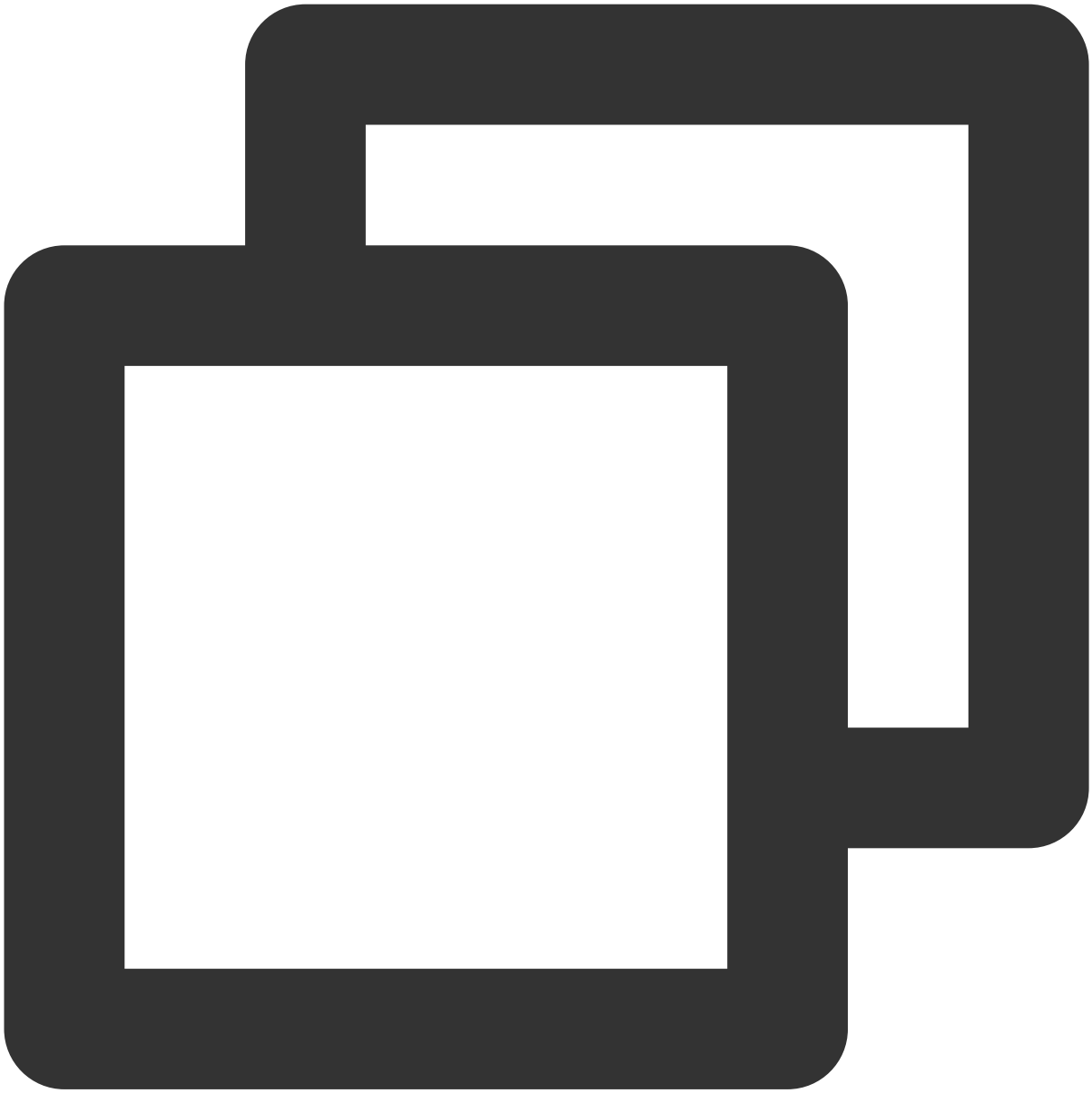
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>HighSpeedHd</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <HighSpeedHd>
```

```
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Bufsize>0</Bufsize>
                <Maxrate>0</Maxrate>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </HighSpeedHd>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String

Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，HighSpeedHd	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
HighSpeedHd	Response.Template	详细的模板参数	Container

Container 节点 HighSpeedHd 的内容：

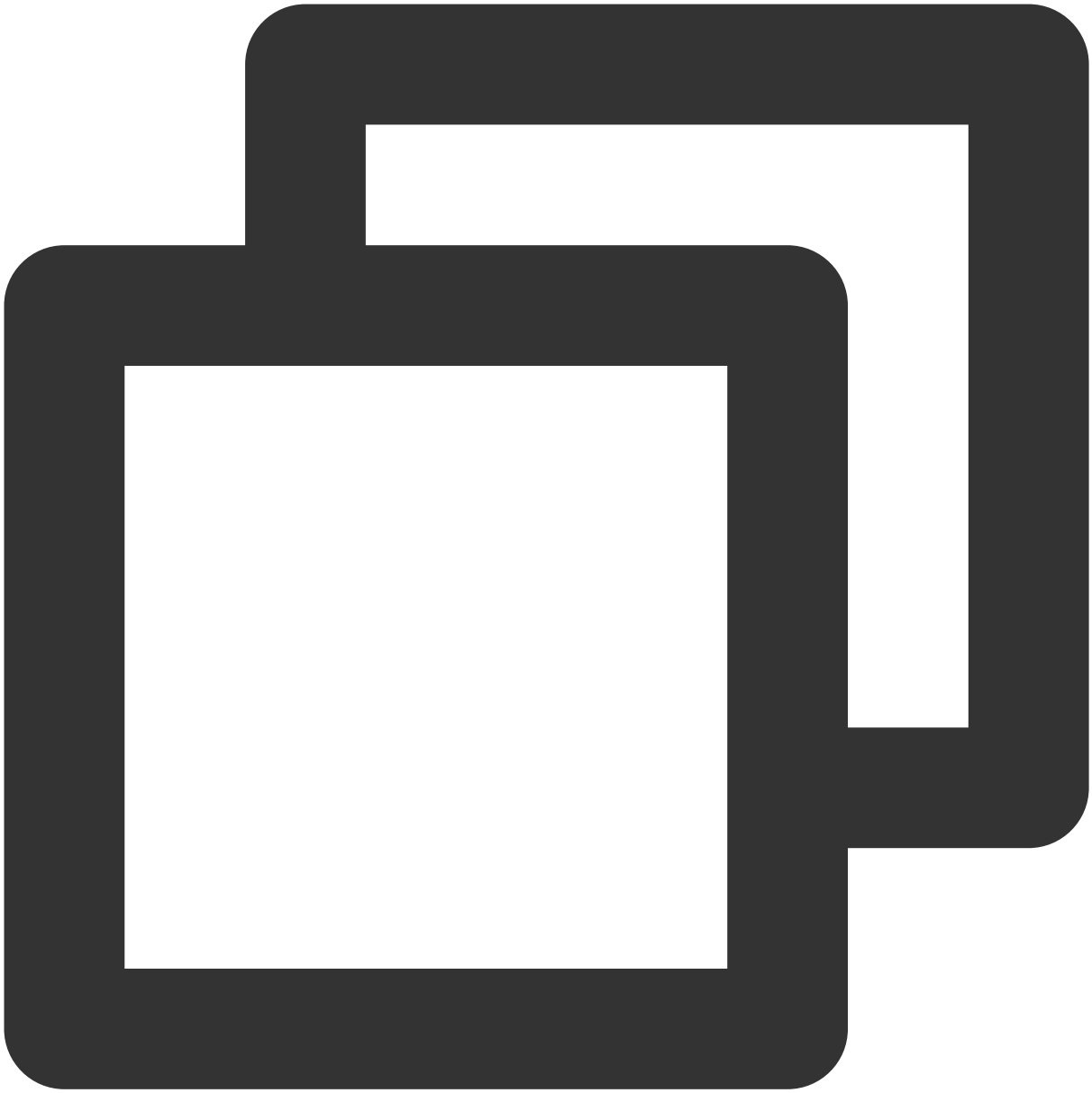
节点名称（关键字）	父节点	描述	类型
TimeInterval	Response.Template.HighSpeedHd	同请求体中的 Request.TimeInterval	Container
Container	Response.Template.HighSpeedHd	同请求体中的 Request.Container	Container
Video	Response.Template.HighSpeedHd	同请求体中的 Request.Video	Container
Audio	Response.Template.HighSpeedHd	同请求体中的 Request.Audio	Container
TransConfig	Response.Template.HighSpeedHd	同请求体中的 Request.TransConfig	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

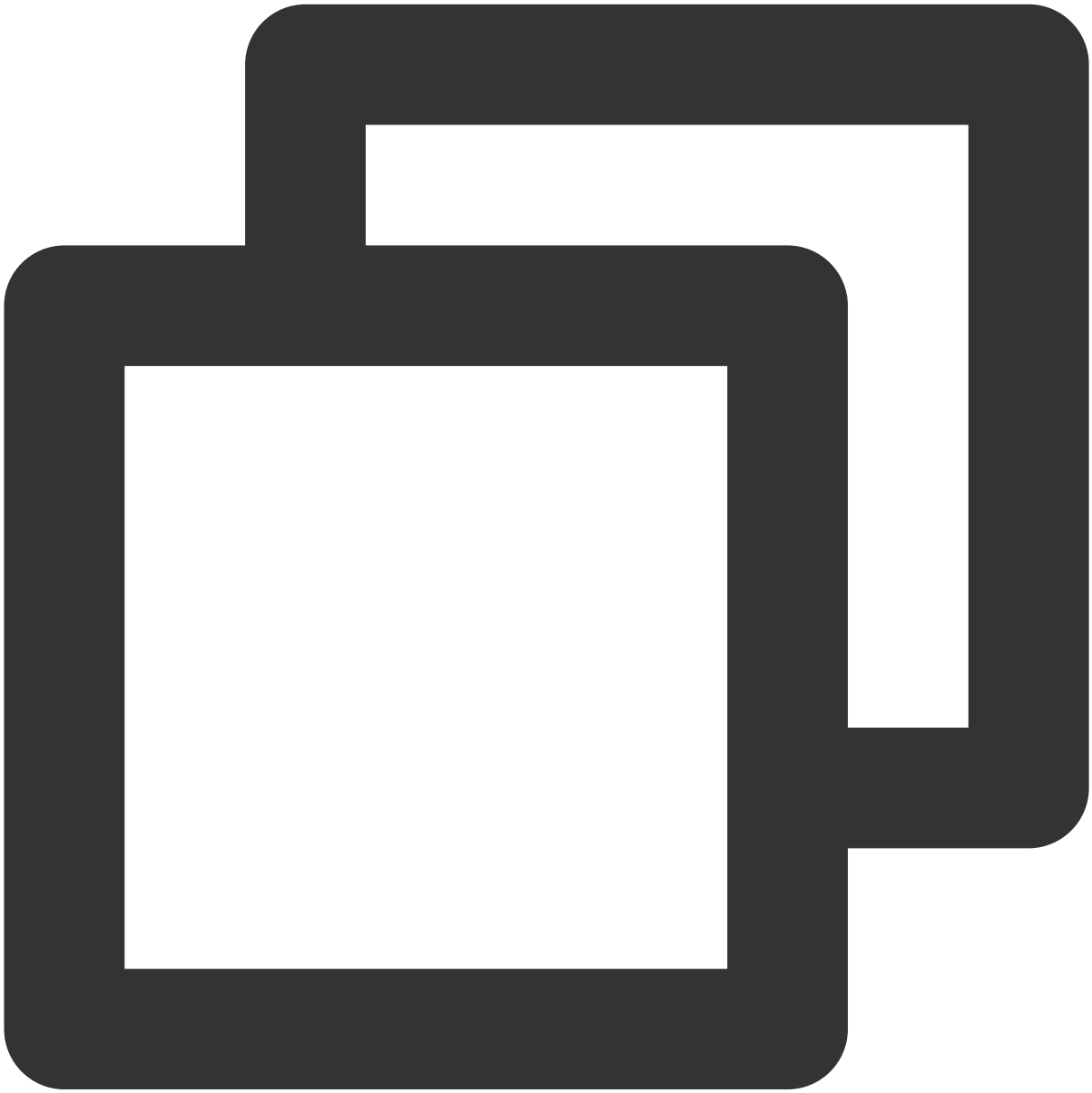
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>HighSpeedHd</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
```

```
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Bufsize>0</Bufsize>
        <Maxrate>0</Maxrate>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>HighSpeedHd</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <HighSpeedHd>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Bufsize>0</Bufsize>
                <Maxrate>0</Maxrate>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </HighSpeedHd>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

创建截图模板

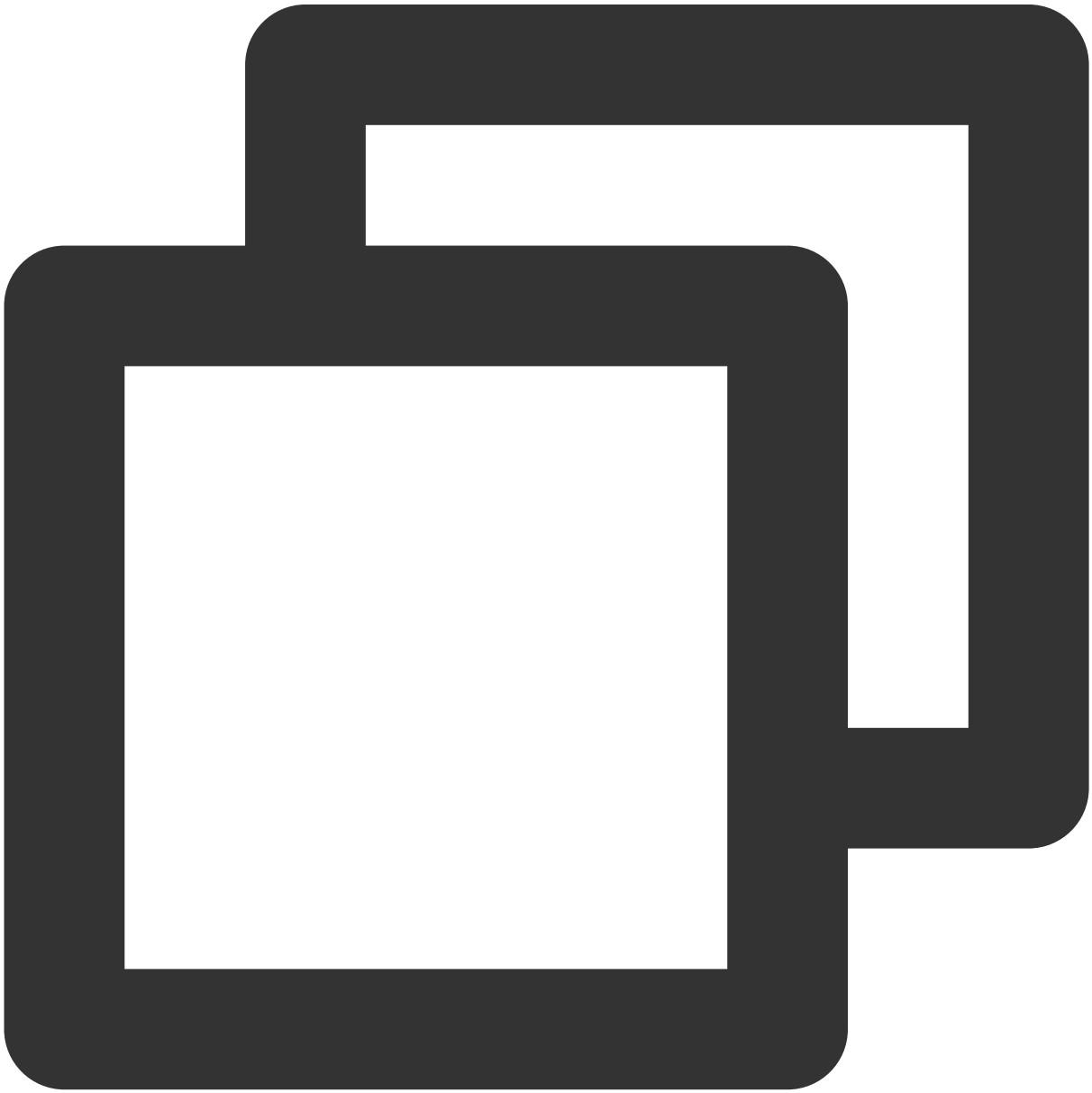
最近更新时间：2024-01-06 10:11:58

功能描述

创建截图模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

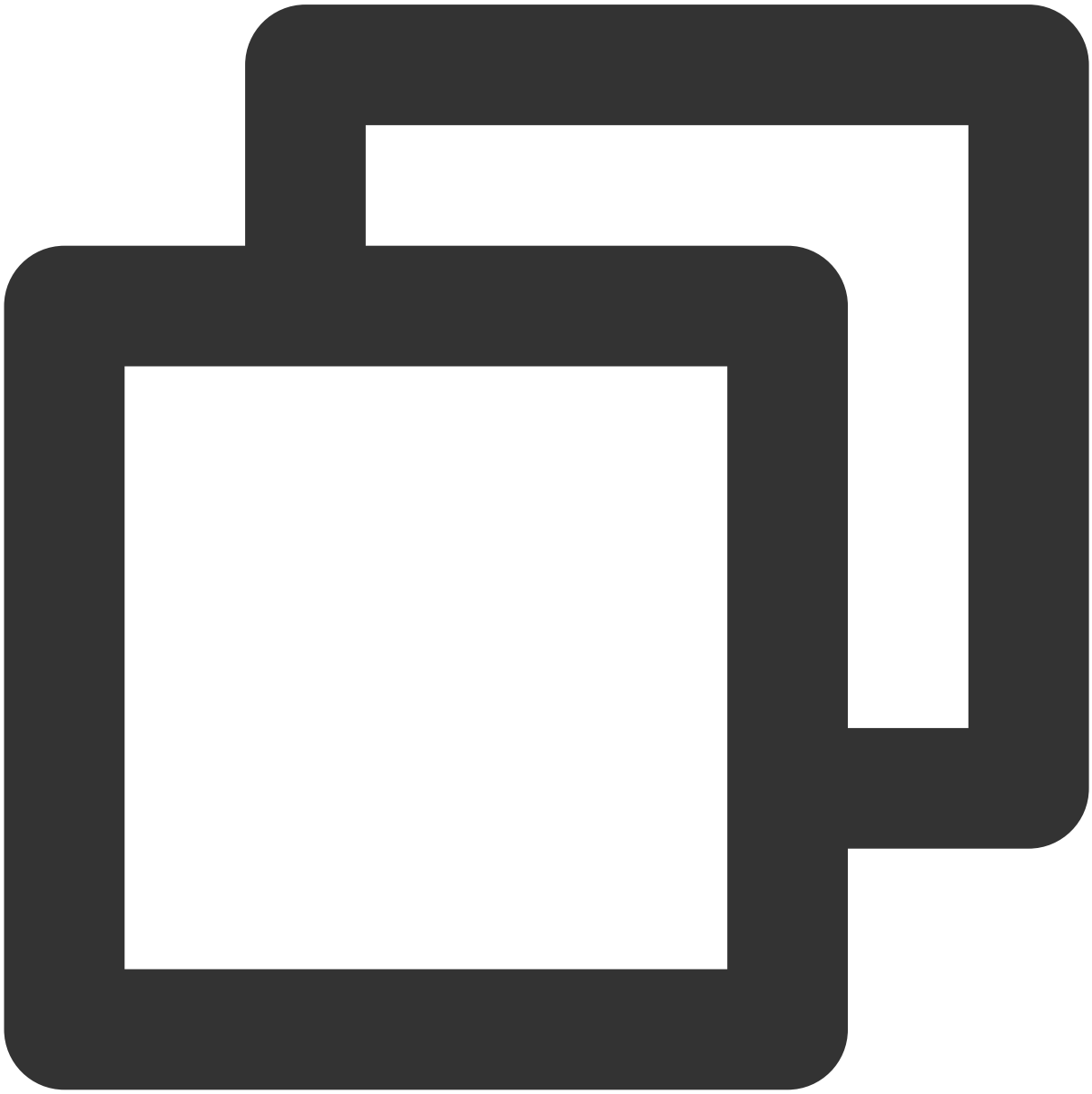
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
        <Width>1280</Width>
        <Height>960</Height>
        <Start>0</Start>
        <TimeInterval>0.5</TimeInterval>
        <Count>10</Count>
        <SnapshotOutMode>OnlySprite</SnapshotOutMode>
        <SpriteSnapshotConfig>
            <Color>Aquamarine</Color>
            <Columns>3</Columns>
            <Lines>3</Lines>
        </SpriteSnapshotConfig>
    </Snapshot>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	模板类型：Snapshot	String	是
Name	Request	模板名称仅支持中文、英文、数字、_、-和*	String	是
Snapshot	Request	截图	Container	是

Container 类型 Snapshot 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Mode	Request.Snapshot	截	String	否	Interval	值范围：{Interval,

		图模式				Average, KeyFram Interval 表示间隔模 Average 表示平均 KeyFrame 表示关键 模式 Interval模式：Start TimeInterval， Count参数生效。当 置Count，未设置 TimeInterval时， 表示截取所有帧， Count张图片 Average模式：Sta Count参数生效。表 从Start开始到视频 束，按平均间隔截 Count张图片
Start	Request.Snapshot	开始时间	String	否	0	[0 视频时长] 单位为秒 支持 float 格式，执 度精确到毫秒
TimeInterval	Request.Snapshot	截图时间间隔	String	否	无	(0 3600] 单位为秒 支持 float 格式，执 度精确到毫秒
Count	Request.Snapshot	截图数量	String	是	无	(0 10000]
Width	Request.Snapshot	宽	String	否	视频原始宽度	值范围：[128，40 单位：px 若只设置 Width 时， 照视频原始比例计 Height
Height	Request.Snapshot	高	String	否	视频原始高度	值范围：[128，40 单位：px 若只设置 Height 时 照视频原始比例计 Width

CIParam	Request.Snapshot	截图图片处理参数	String	否	无	参考 图片处理 例如: imageMogr2/forma
IsCheckCount	Request.Snapshot	是否强制检查截图个数	String	否	false	使用自定义间隔模 图时，视频时长不 取 Count 个截图， 转为平均截图模式 Count 个截图
IsCheckBlack	Request.Snapshot	是否开启黑屏检测	String	否	false	true/false
BlackLevel	Request.Snapshot	截图黑屏检测参数	String	否	空	当 IsCheckBlack=t 时有效 值参考范围[30，1 表示黑色像素的占 值，值越小，黑色 越小 Start>0，参数设置 效，不做过滤黑屏 Start =0 参数有效， 帧的开始时间为第 非黑屏开始
PixelBlackThreshold	Request.Snapshot	截图黑屏检	String	否	空	当IsCheckBlack=tr 有效 判断像素点是否为 点的阈值，取值范 [0，255]

		测参数				
SnapshotOutMode	Request.Snapshot	截图输出模式参数	String	否	OnlySnapshot	值范围：{OnlySnapshot, OnlySprite, SnapshotAndSprite OnlySnapshot 表示出截图模式 OnlySp 表示仅输出雪碧图 SnapshotAndSprite 示输出截图与雪碧 式
SpriteSnapshotConfig	Request.Snapshot	雪碧图输出配置	Container	否	无	无

Container 类型 Snapshot.SpriteSnapshotConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
CellWidth	Request.Snapshot.SpriteSnapshotConfig	单图宽度	String	否	截图宽度	值范围：[8，4096] 单位：px
CellHeight	Request.Snapshot.SpriteSnapshotConfig	单图高度	String	否	截图高度	值范围：[8，4096] 单位：px
Padding	Request.Snapshot.SpriteSnapshotConfig	雪碧图内边距大小	String	否	0	值范围：[0，1024] 单位：px
Margin	Request.Snapshot.SpriteSnapshotConfig	雪碧图外边距大小	String	否	0	值范围：[0，1024] 单位：px

版权所有：腾讯云计算（北京）有限责任公司

Color	Request.Snapshot.SpriteSnapshotConfig	背景颜色	String	是	无	支持颜色详见 FFmpeg
Columns	Request.Snapshot.SpriteSnapshotConfig	雪碧图列数	String	是	0	值范围：[1，10000]
Lines	Request.Snapshot.SpriteSnapshotConfig	雪碧图行数	String	是	0	值范围：[1，10000]

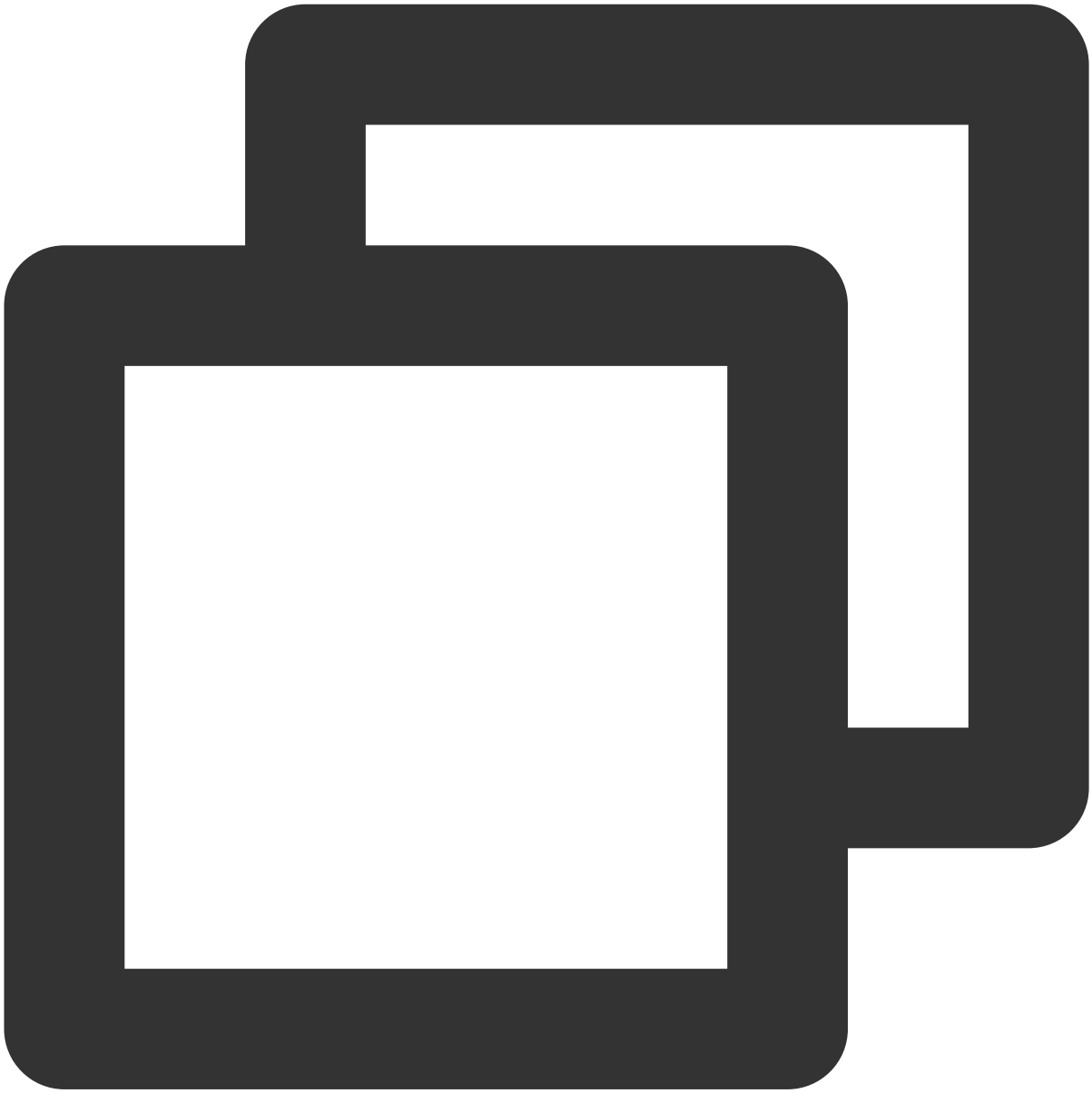
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Snapshot</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Snapshot>
            <Mode>Interval</Mode>
            <Width>1280</Width>
            <Height>960</Height>
```

```
            <Start>0</Start>
            <TimeInterval>0.5</TimeInterval>
            <Count>10</Count>
            <SnapshotOutMode>OnlySprite</SnapshotOutMode>
            <SpriteSnapshotConfig>
                <Color>Aquamarine</Color>
                <Columns>3</Columns>
                <Lines>3</Lines>
            </SpriteSnapshotConfig>
        </Snapshot>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

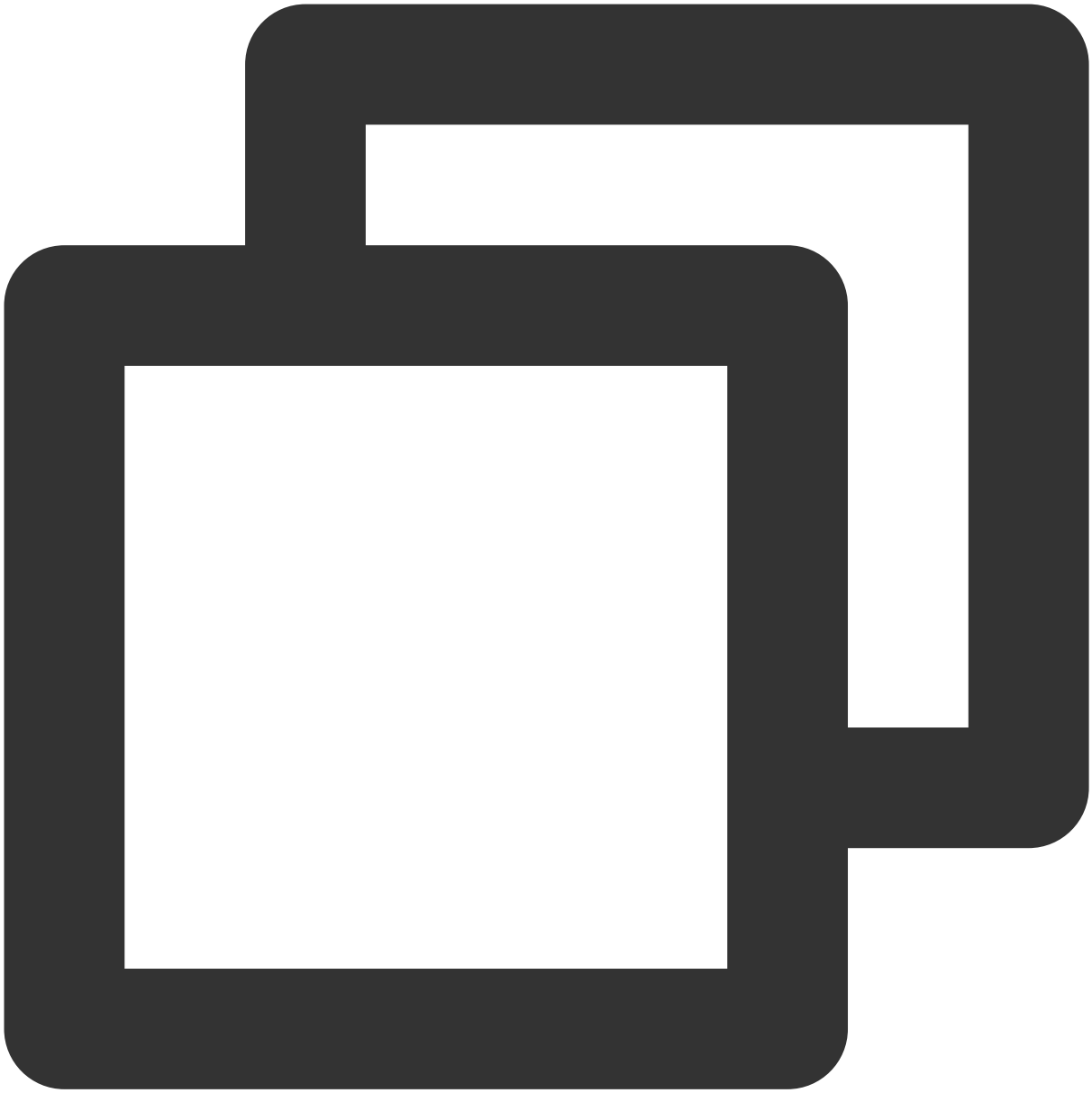
节点名称（关键字）	父节点	描述	类型
Tag	Response	模板类型，Snapshot	String
Name	Response	模板名字	String
TemplateId	Response	模板 ID	String
UpdateTime	Response	更新时间	String
CreateTime	Response	创建时间	String
Snapshot	Response	详细的模板参数，同上述请求体部分 Snapshot 说明	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

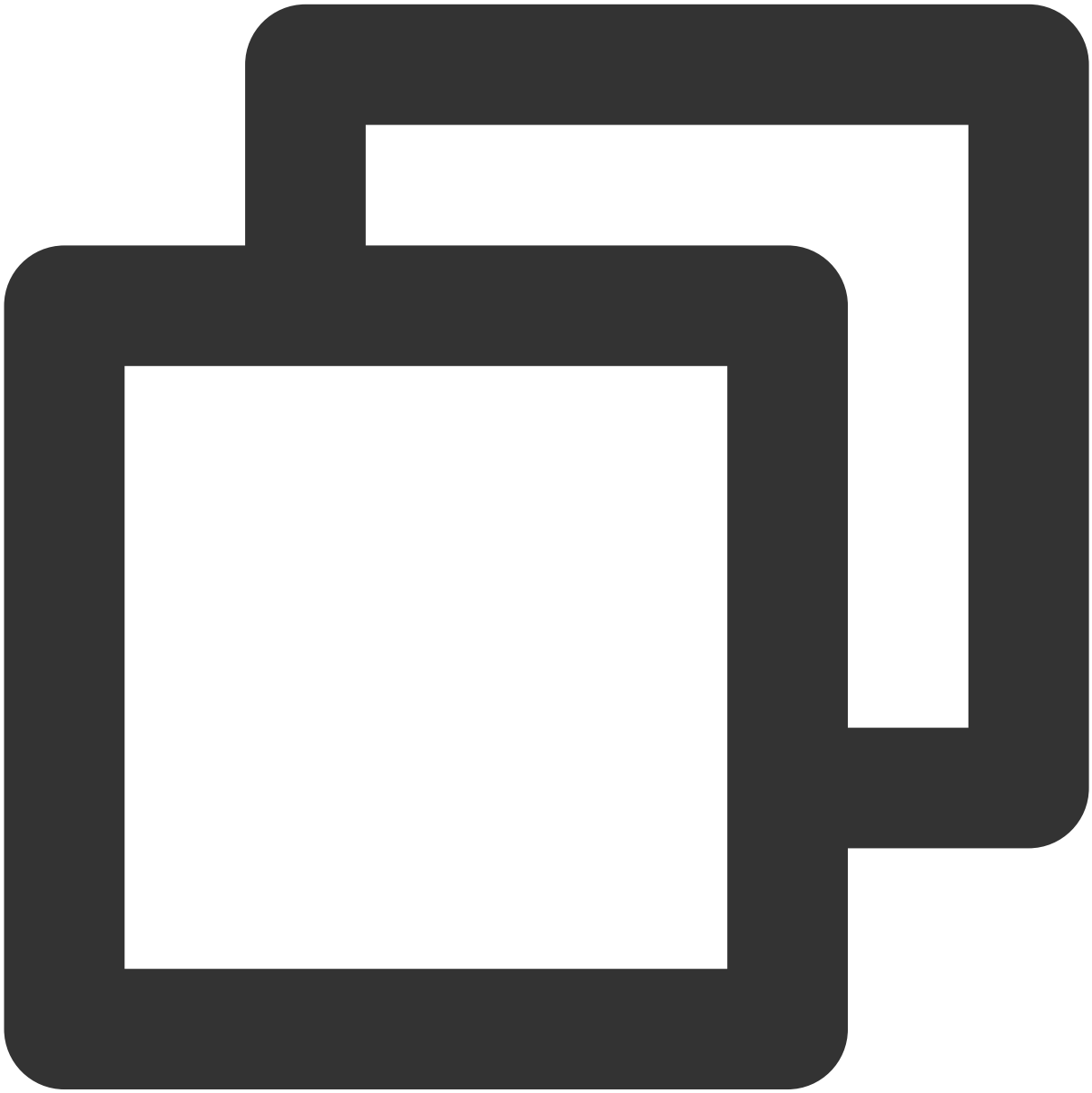
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
```

```
        <Width>1280</Width>
        <Height>960</Height>
        <Start>0</Start>
        <TimeInterval>0.5</TimeInterval>
        <Count>10</Count>
        <SnapshotOutMode>OnlySprite</SnapshotOutMode>
        <SpriteSnapshotConfig>
            <Color>Aquamarine</Color>
            <Columns>3</Columns>
            <Lines>3</Lines>
        </SpriteSnapshotConfig>
    </Snapshot>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <Template>
        <Tag>Snapshot</Tag>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Snapshot>
            <Mode>Interval</Mode>
            <Width>1280</Width>
            <Height>960</Height>
            <Start>0</Start>
            <TimeInterval>0.5</TimeInterval>
            <Count>10</Count>
            <SnapshotOutMode>OnlySprite</SnapshotOutMode>
            <SpriteSnapshotConfig>
                <Color>Aquamarine</Color>
                <Columns>3</Columns>
                <Lines>3</Lines>
            </SpriteSnapshotConfig>
        </Snapshot>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

创建超分辨率模板

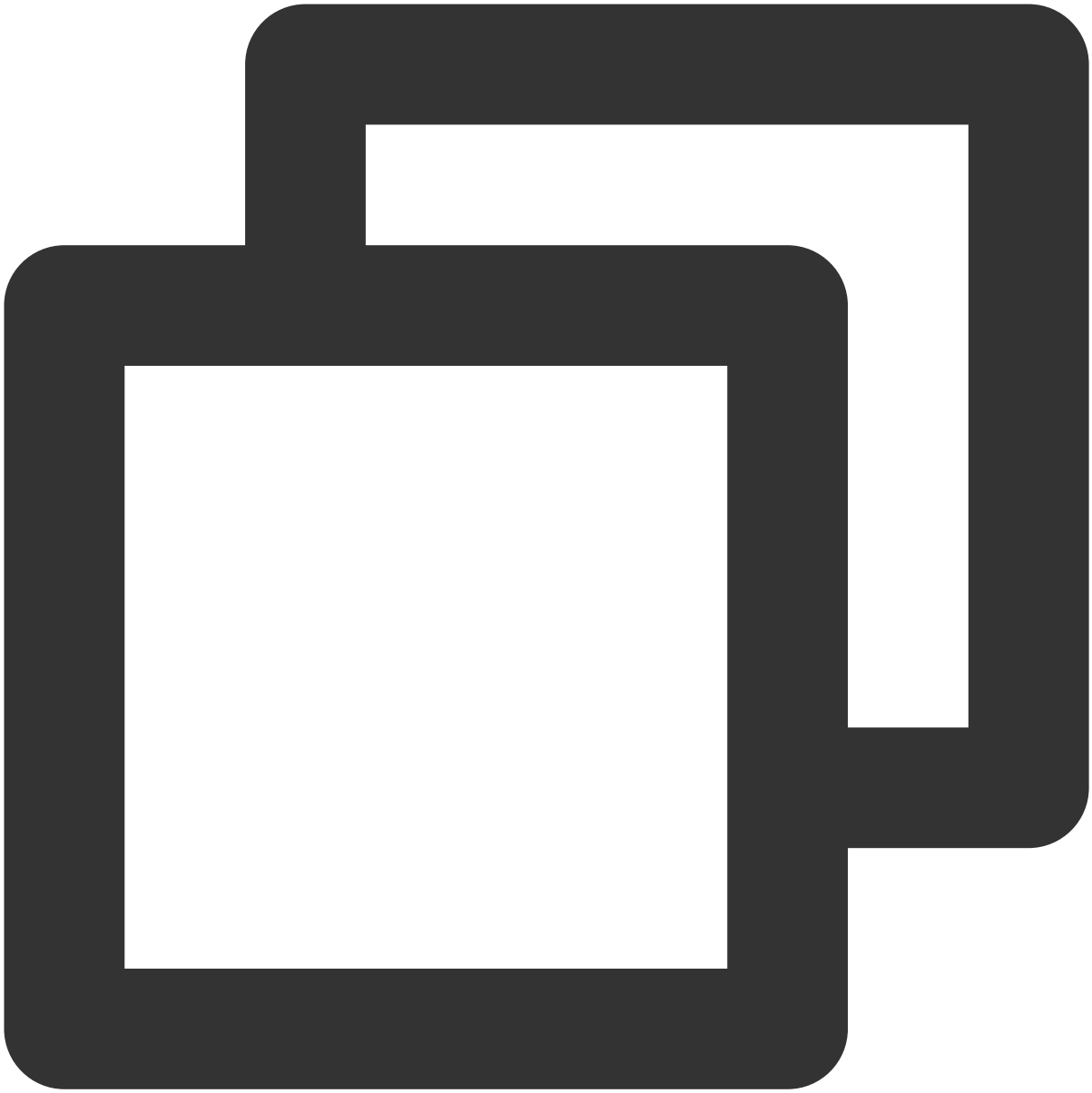
最近更新时间：2024-01-06 10:11:58

功能描述

创建超分辨率模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

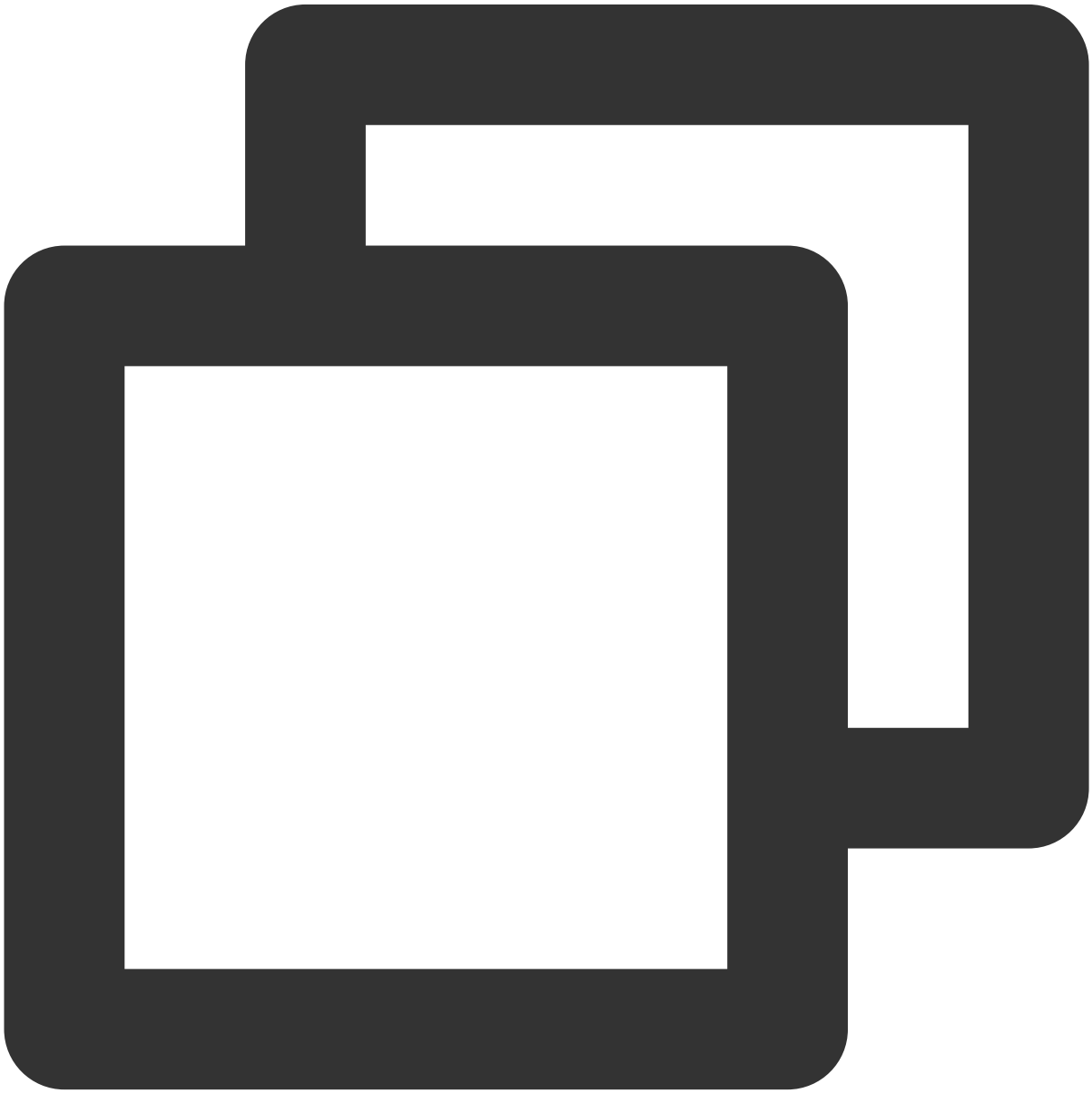
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>SuperResolution</Tag>
    <Name>TemplateName</Name>
    <Resolution>sdtohd</Resolution>
    <EnableScaleUp>true</EnableScaleUp>
    <Version>Enhance</Version>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型：SuperResolution	String	是	无
Name	Request	模板名称：仅支持中文、英文、数字、_、-和*	String	是	无
Resolution	Request	分辨率选项	String	是	1. sdtohd：标清到超清 2. hdto4k：高清到4K
EnableScaleUp	Request	自动缩放开关，默认关闭	String	否	true、false
Version	Request	版本，默认值 Base	String	否	1. Base：基础版 2. Enhance：增强版

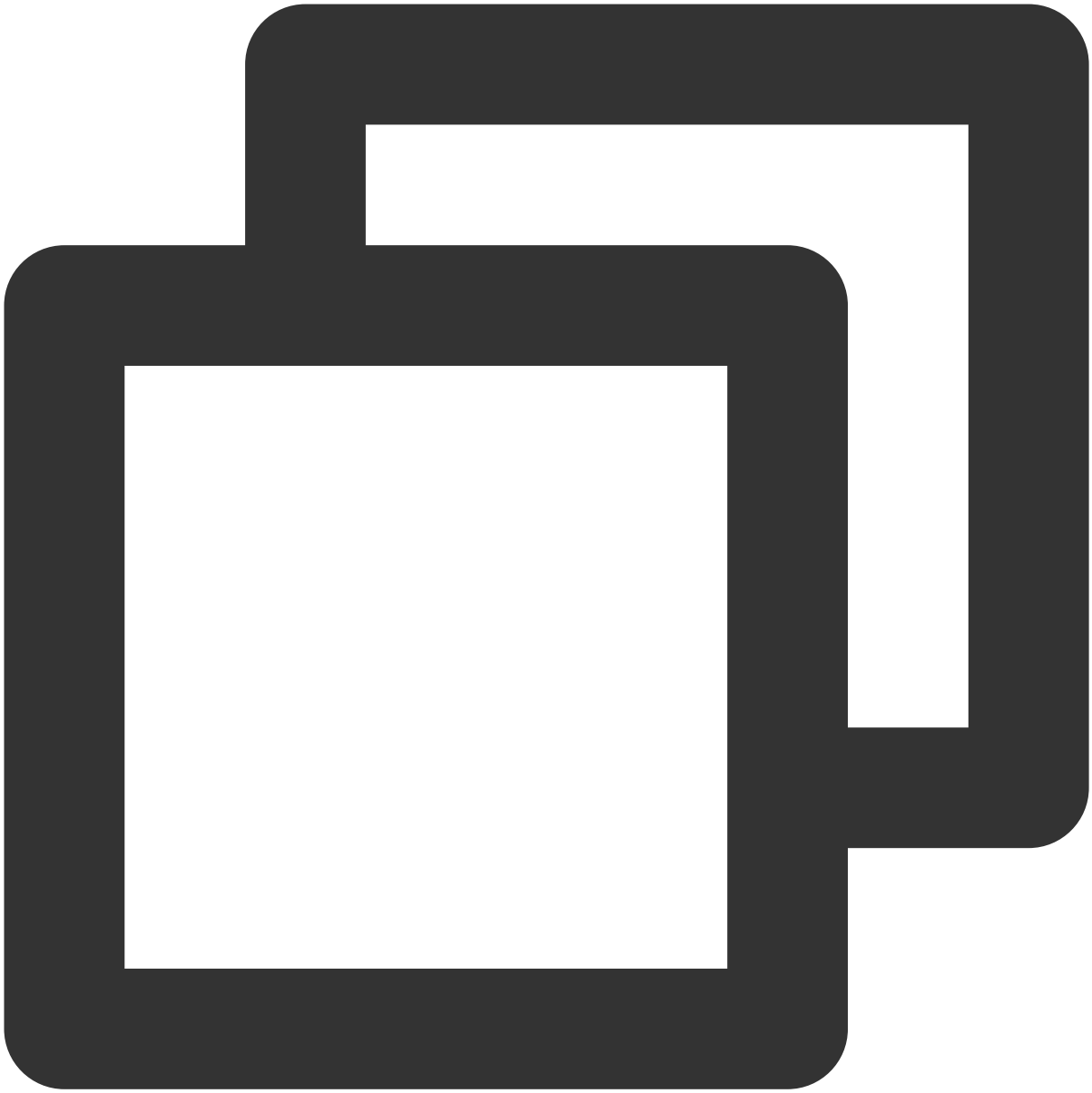
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>SuperResolution</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SuperResolution>
            <Resolution>sdtohd</Resolution>
            <EnableScaleUp>true</EnableScaleUp>
            <Version>Enhance</Version>
```

```
        </SuperResolution>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，SuperResolution	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
SuperResolution	Response.Template	详细的模板参数	Container

Container 节点 SuperResolution 的内容：

节点名称（关键字）	父节点	描述	类型
Resolution	Response.Template.SuperResolution	同请求体中的 Request.Resolution	String
EnableScaleUp	Response.Template.SuperResolution	同请求体中的 Request.EnableScaleUp	String

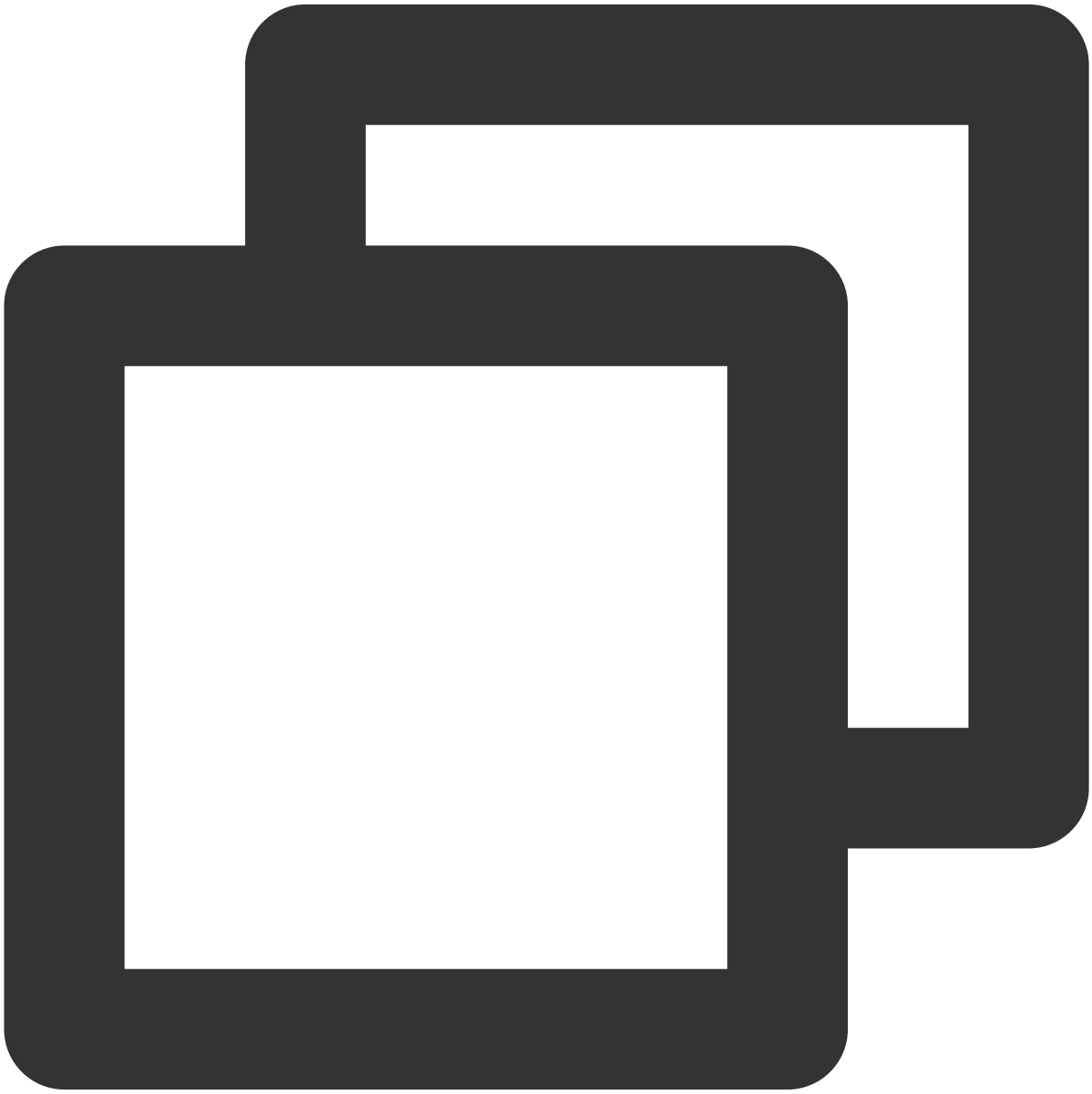
错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

版权所有：腾讯云计算（北京）有限责任公司

实际案例

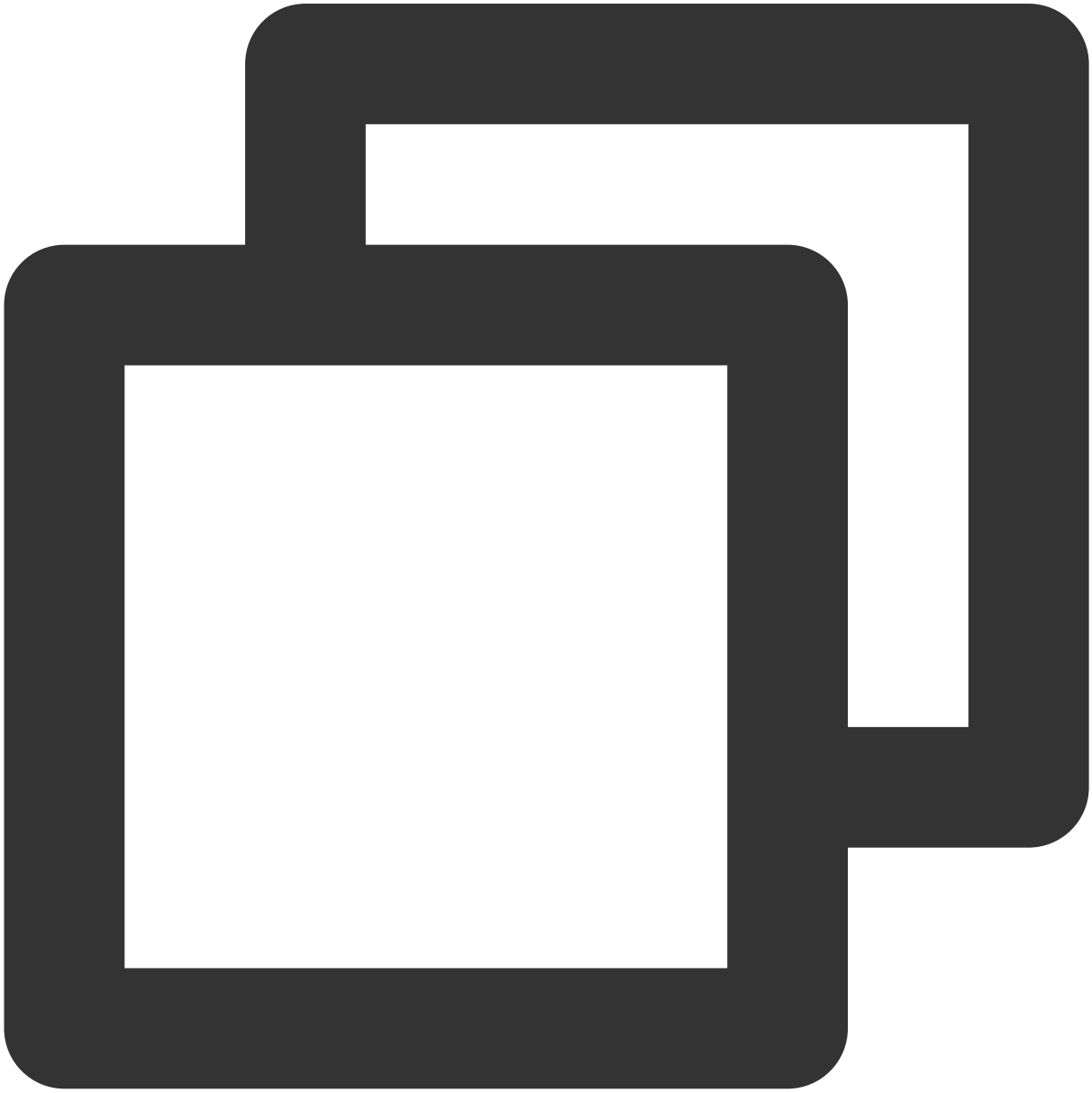
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml
```

```
<Request>
    <Tag>SuperResolution</Tag>
    <Name>TemplateName</Name>
    <Resolution>sdtohd</Resolution>
    <EnableScaleUp>true</EnableScaleUp>
    <Version>Enhance</Version>
</Request>
```

响应



```
HTTP/1.1 200 OK
```

```
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>SuperResolution</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SuperResolution>
            <Resolution>sdtohd</Resolution>
            <EnableScaleUp>true</EnableScaleUp>
            <Version>Enhance</Version>
        </SuperResolution>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

创建音视频转码模板

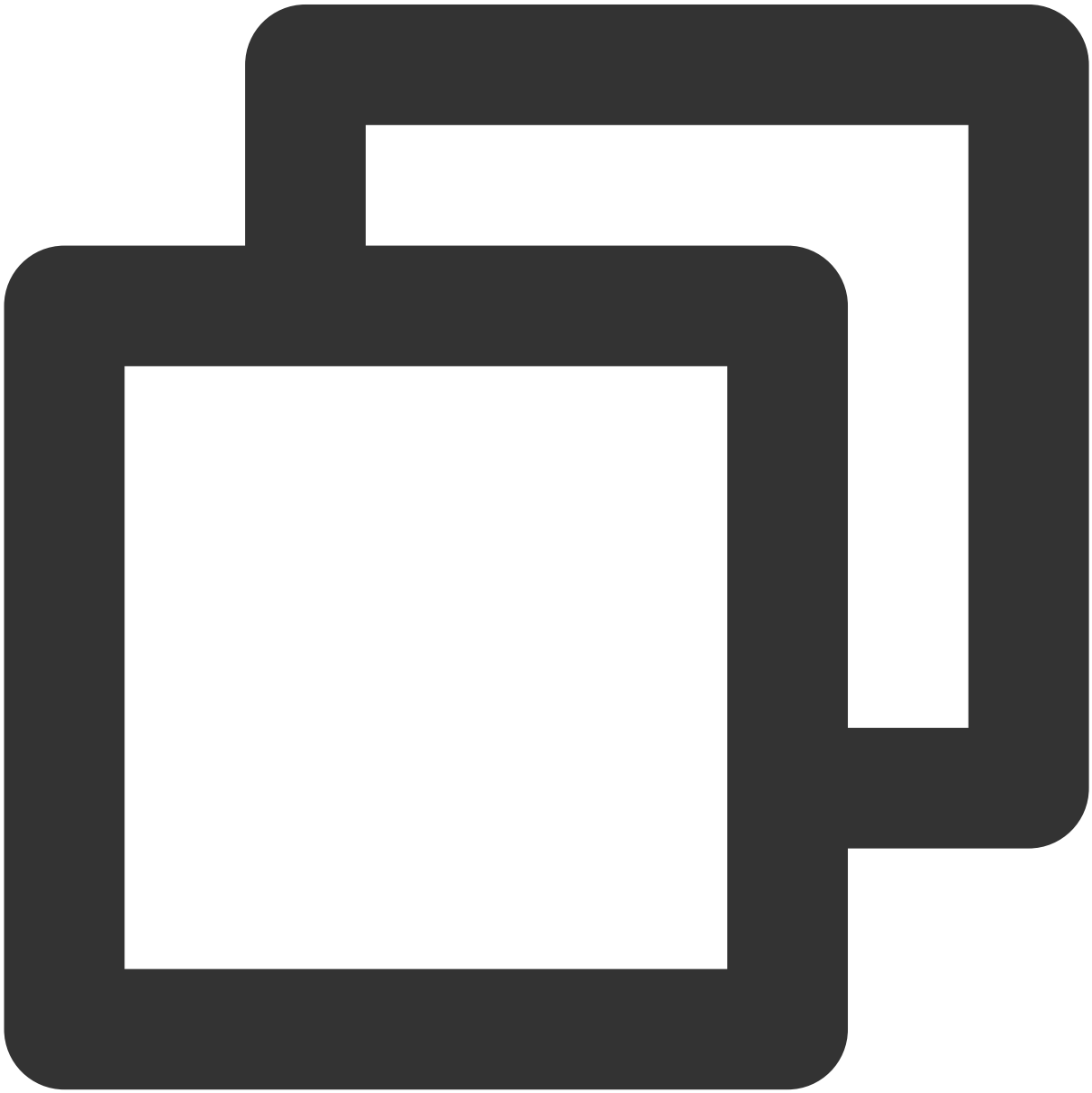
最近更新时间：2024-01-06 10:11:58

功能描述

创建音视频转码模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

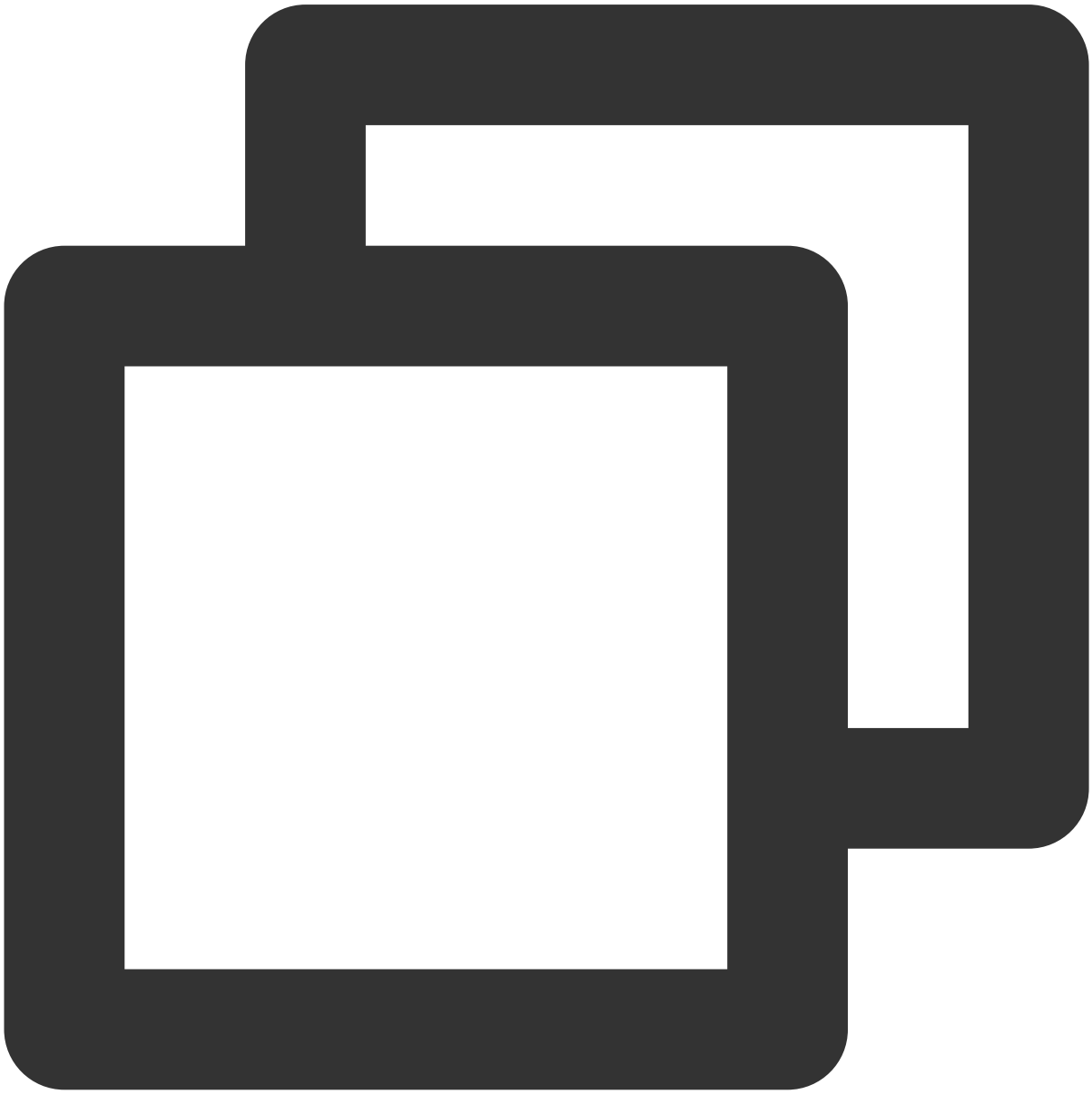
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Transcode</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Profile>high</Profile>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Preset>medium</Preset>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Audio
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
        <EffectConfig>
            <EnableStartFadein>true</EnableStartFadein>
            <StartFadeinTime>3</StartFadeinTime>
            <EnableEndFadeout>false</EnableEndFadeout>
            <EndFadeoutTime>0</EndFadeoutTime>
            <EnableBgmFade>true</EnableBgmFade>
            <BgmFadeTime>1.7</BgmFadeTime>
        </EffectConfig>
    </AudioMixArray>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.mp3</Audio
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
        <EffectConfig>
            <EnableStartFadein>true</EnableStartFadein>
```

```
            <StartFadeinTime>3</StartFadeinTime>
            <EnableEndFadeout>false</EnableEndFadeout>
            <EndFadeoutTime>0</EndFadeoutTime>
            <EnableBgmFade>true</EnableBgmFade>
            <BgmFadeTime>1.7</BgmFadeTime>
        </EffectConfig>
    </AudioMixArray>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型：Transcode	String	是	无
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*	String	是	无
Container	Request	容器格式	Container	是	无
Video	Request	视频信息	Container	否	不传 Video，相当于删除视频信息
TimeInterval	Request	时间区间	Container	否	无
Audio	Request	音频信息	Container	否	不传 Audio，相当于删除音频信息
TransConfig	Request	转码配置	Container	否	无
AudioMix	Request	混音参数，详情见 AudioMix	Container	否	Audio.Remove 为 false 时生效
AudioMixArray	Request	混音参数，最多同时传2个。详情见 AudioMixArray	Container数组	否	Audio.Remove 为 false 时生效

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Format	Request.Container	封装格式：取值见下表	String	是
ClipConfig	Request.Container	分片配置，当 format 为 hls 和 dash 时有效	Container	否

Container 类型 ClipConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Duration	Request.Container.ClipConfig	分片时长，默认5s	String	否

设定 container，音频视频支持的格式如下表所示：

Container	Audio Codecs	Video Codecs
mp4/hls/mkv	aac、mp3	H.264、H.265、AV1
hls/mkv	aac、mp3	H.264、H.265
ts/flv/avi/mov	aac、mp3	H.264
dash	aac	H.264
WebM	Vorbis、opus	VP8、VP9、AV1
aac	aac	不支持
mp3	mp3	不支持
flac	flac	不支持
amr	amr	不支持
adts	aac	不支持
m4a	aac	不支持
wav	pcm_s16le	不支持

Container 类型 Video 的具体数据描述如下：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Video	编解码格式	String	否	H.264 当 format 为 WebM 时，为 VP8	H.264 H.265 VP8 VP9 AV1
Width	Request.Video	宽	String	否	视频原始宽度	值范围：[128, 4096] 单位：px 若只设置 Width 时，按照视频原始比例计算 Height 必须为偶数
Height	Request.Video	高	String	否	视频原始高度	值范围：[128, 4096] 单位：px 若只设置 Height 时，按照视频原始比例计算 Width 必须为偶数
Fps	Request.Video	帧率	String	否	无	值范围：(0, 60] 单位：fps
Remove	Request.Video	是否删除视频流	String	否	false	true、false
Profile	Request.Video	编码级别	String	否	high	支持 baseline、main、high、auto 当 Pixfmt 为 auto 时，该参数仅能设置为 auto，当设置为其他选项时，参数值将被设置为 auto baseline：适合移动设备 main：适合标准分辨率设备 high：适合高分辨率设备 仅H.264支持此参数

Bitrate	Request.Video	视频输出文件的码率	String	否	无	值范围：[10, 50000]，单位：Kbps auto 表示自适应码率
Crf	Request.Video	码率-质量控制因子	String	否	无	值范围：(0, 51] 如果设置了 Crf，则 Bitrate 的设置失效 当 Bitrate 为空时，默认为 25
Gop	Request.Video	关键帧间最大帧数	String	否	无	值范围：[1, 100000]
Preset	Request.Video	视频算法器预置	String	否	当 Codec 为 H.264 时，为 medium 当 Codec 为 VP8 时，为 good 当 Codec 为 AV1 时，为 5	H.264支持该参数，取值 veryfast、fast、medium、slow、slower VP8 支持该参数，取值 good、realtime AV1 支持该参数，取值5（推荐值）、4 H.265 和 VP9 不支持该参数
Bufsize	Request.Video	缓冲区大小	String	否	无	值范围：[1000, 128000] 单位：Kb Codec 为 VP8/VP9时不支持此参数
Maxrate	Request.Video	视频	String	否	无	值范围：[10, 50000] 单位：Kbps

		码率峰值				Codec 为 VP8/VP9时不支持此参数
Pixfmt	Request.Video	视频颜色格式	String	否	无	H.264支持：yuv420p、yuv422p、yuv444p、yuvj420p、yuvj422p、yuvj444p、auto H.265支持：yuv420p、yuv420p10le、auto Codec 为 VP8/VP9/AV1 时不支持此参数
LongShortMode	Request.Video	长短边自适应	String	否	false	true、false Codec 为 VP8/VP9/AV1 时不支持此参数
Rotate	Request.Video	旋转角度	String	否	无	值范围：[0, 360) 单位：度

Container 类型 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.TimeInterval	开始时间	String	否	0	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒
Duration	Request.TimeInterval	持续时间	String	否	视频原始时长	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒

Container 类型 Audio 的具体数据描述如下：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Audio	编解码格式	String	否	aac 当 format 为 WebM 时，为 Vorbis 当 format 为 wav 时，为 pcm_s16le	取值 aac、mp3、flac、amr、Vorbis、opus、pcm_s16le
Samplerate	Request.Audio	采样率	String	否	44100，当 Codec 为 opus 时，默认值为48000	单位：Hz 可选 8000、11025、12000、16000、22050、24000、32000、44100、48000、88200、96000 不同的封装，mp3 支持不同的采样率，如下表所示 当 Codec 设置为 amr 时，只支持8000 当 Codec 设置为 opus 时，支持8000，16000，24000，48000
Bitrate	Request.Audio	原始音频码率	String	否	无	单位：Kbps 值范围：[8，1000]
Channels	Request.Audio	声道数	String	否	无	当 Codec 设置为 aac/flac，支持1、2、4、5、6、8 当 Codec 设置为 mp3/opus 时，支持1、2 当 Codec 设置为 Vorbis 时，只支持2 当 Codec 设置为 amr，只支持1 当 Codec 设置为 pcm_s16le 时，只支持1、

						2 当封装格式为 dash 时，不支持8
Remove	Request.Audio	是否删除源音频流	String	否	false	取值 true、false
KeepTwoTracks	Request.Audio	保持双音轨	String	否	false	取值 true、false。 当 Video.Codec 为 H.265 时，此参数无效
SwitchTrack	Request.Audio	转换轨道	String	否	false	取值 true、false。 当 Video.Codec 为 H.265 时，此参数无效
SampleFormat	Request.Audio	采样位宽	String	否	无	当 Codec 设置为 aac，支持 fltp 当 Codec 设置为 mp3，支持 fltp、s16p、s32p 当 Codec 设置为 flac，支持 s16、s32、s16p、s32p 当 Codec 设置为 amr，支持 s16、s16p 当 Codec 设置为 opus，支持 s16 当 Codec 设置为 pcm_s16le，支持 s16 当 Codec 设置为 Vorbis，支持 fltp 当 Video.Codec 为 H.265 时，此参数无效

说明

Y 表示支持这种采样率，N 表示不支持。

Audio.Codec 采样率支持情况，如下表所示：

Audio.Codec	aac	amr	flac	opus	Vorbis	pcm_s16le	mp3
8000	Y	Y	Y	Y	Y	Y	封装不同，支持不同
11025	Y	N	Y	N	Y	Y	
12000	Y	N	Y	N	Y	Y	
16000	Y	N	Y	Y	Y	Y	
22050	Y	N	Y	N	Y	Y	
24000	Y	N	Y	Y	Y	Y	
32000	Y	N	Y	N	Y	Y	
44100	Y	N	Y	N	Y	Y	
48000	Y	N	Y	Y	Y	Y	
88200	Y	N	Y	N	Y	Y	
96000	Y	N	Y	N	Y	Y	

当 Audio.Codec 为 mp3 时，Container.Format 的采样率兼容情况如下所示：

封装格式/音频采样率	8000	11025	12000	16000	22050	24000	32000	44100	4
flv	N	N	N	N	Y	N	N	Y	N
mp4	N	N	N	Y	Y	Y	Y	Y	Y
hls/ts/mp3/mkv/avi/mov	Y	Y	Y	Y	Y	Y	Y	Y	Y

Container 类型 TransConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
AdjDarMethod	Request.TransConfig	分辨率调整方式	String

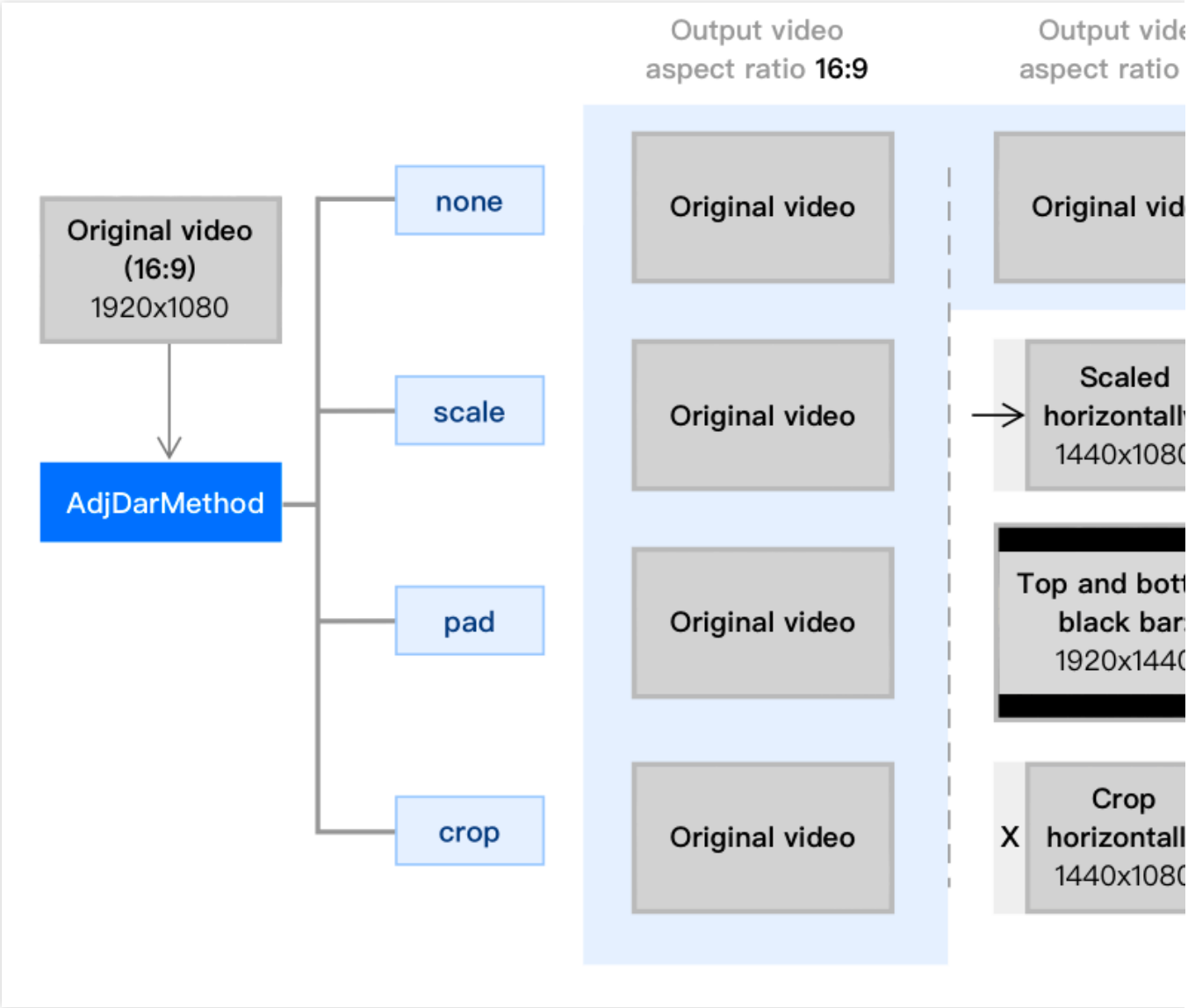
IsCheckReso	Request.TransConfig	是否检查分辨率	String
ResoAdjMethod	Request.TransConfig	分辨率调整方式	String
IsCheckVideoBitrate	Request.TransConfig	是否检查视频码率	String
VideoBitrateAdjMethod	Request.TransConfig	视频码率调整方式	String
IsCheckAudioBitrate	Request.TransConfig	是否检查音频码率	String
AudioBitrateAdjMethod	Request.TransConfig	音频码率调整方式	String
IsCheckVideoFps	Request.TransConfig	是否检查视频帧率	String
VideoFpsAdjMethod	Request.TransConfig	视频帧率调整方式	String

版权所有：腾讯云计算（北京）有限责任公司

DeleteMetadata	Request.TransConfig	是否删除文件中的 MetaData 信息	String
IsHdr2Sdr	Request.TransConfig	是否开启 HDR 转 SDR	String
TranscodeIndex	Request.TransConfig	指定处理的流编号，对应媒体信息中的 Response.MediaInfo.Stream.Video.Index 和 Response.MediaInfo.Stream.Audio.Index，详见 获取媒体信息接口	String
HlsEncrypt	Request.TransConfig	hls 加密配置	Container
DashEncrypt	Request.TransConfig	dash 加密配置	Container

AdjDarMethod 参数图示：

版权所有：腾讯云计算（北京）有限责任公司



Container 类型 HlsEncrypt 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
IsHlsEncrypt	Request.TransConfig.HlsEncrypt	是否开启 HLS 加密	String	否	false	true/false 当 Container.Format 为 hls 时支持加密
UriKey	Request.TransConfig.HlsEncrypt	HLS 加密的 key	String	否	无	当 IsHlsEncrypt 为 true 时，该参数才有意义

Container 类型 DashEncrypt 的具体数据描述如下：

节点名称	父节点	描述	类型	是	默认	限制

（关键字）				否 必选	值	
IsEncrypt	Request.TransConfig.DashEncrypt	是否开启 DASH 加密	String	否	false	true/false 当 Container.Format 为 dash 时支持加密
UriKey	Request.TransConfig.DashEncrypt	DASH 加密的 key	String	否	无	当 IsEncrypt 为 true 时，该参数才有意义

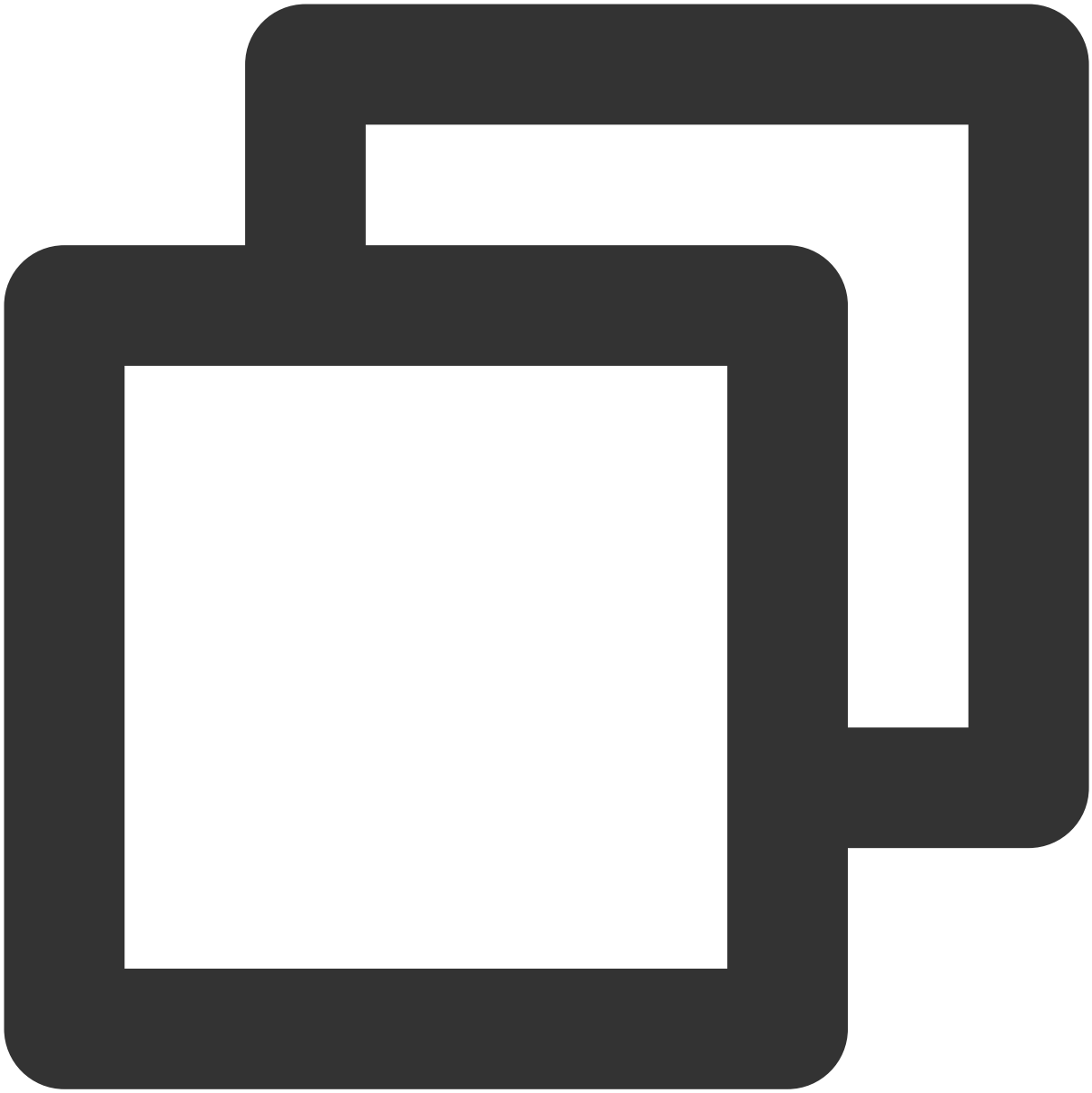
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Transcode</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Container>
            <Format>mp4</Format>
        </Container>
        <Video>
```

```
            <Codec>H.264</Codec>
            <Profile>high</Profile>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Fps>30</Fps>
            <Preset>medium</Preset>
        </Video>
        <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
        </Audio>
        <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
        </TransConfig>
        <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
        </TimeInterval>
        <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</A
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMixArray>
        <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.mp3</A
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
```

```
        </AudioMixArray>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，Transcode	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
TransTpl	Response.Template	详细的模板参数	Container

Container节点TransTpl的内容：

节点名称（关键字）	父节点	描述	类型
TimeInterval	Response.Template.TransTpl	同请求体中的 Request.TimeInterval	Container
Container	Response.Template.TransTpl	同请求体中的 Request.Container	Container
Video	Response.Template.TransTpl	同请求体中的 Request.Video	Container
Audio	Response.Template.TransTpl	同请求体中的 Request.Audio	Container
TransConfig	Response.Template.TransTpl	同请求体中的	Container

版权所有：腾讯云计算（北京）有限责任公司

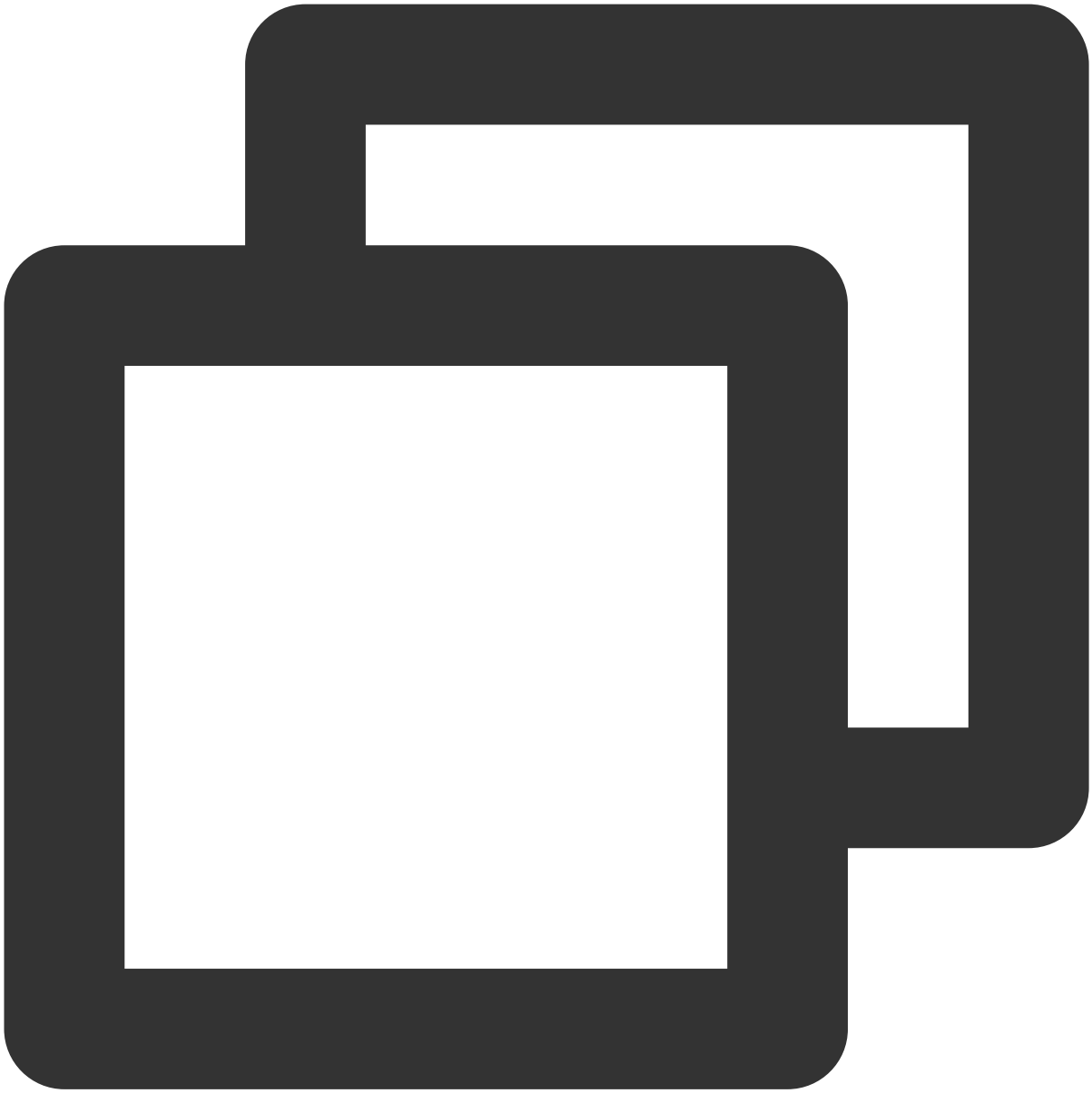
		Request.TransConfig	
AudioMix	Response.Template.TransTpl	同请求体中的 Request.AudioMix	Container
AudioMixArray	Response.Template.TransTpl	同请求体中的 Request.AudioMixArray	Container 数组

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求



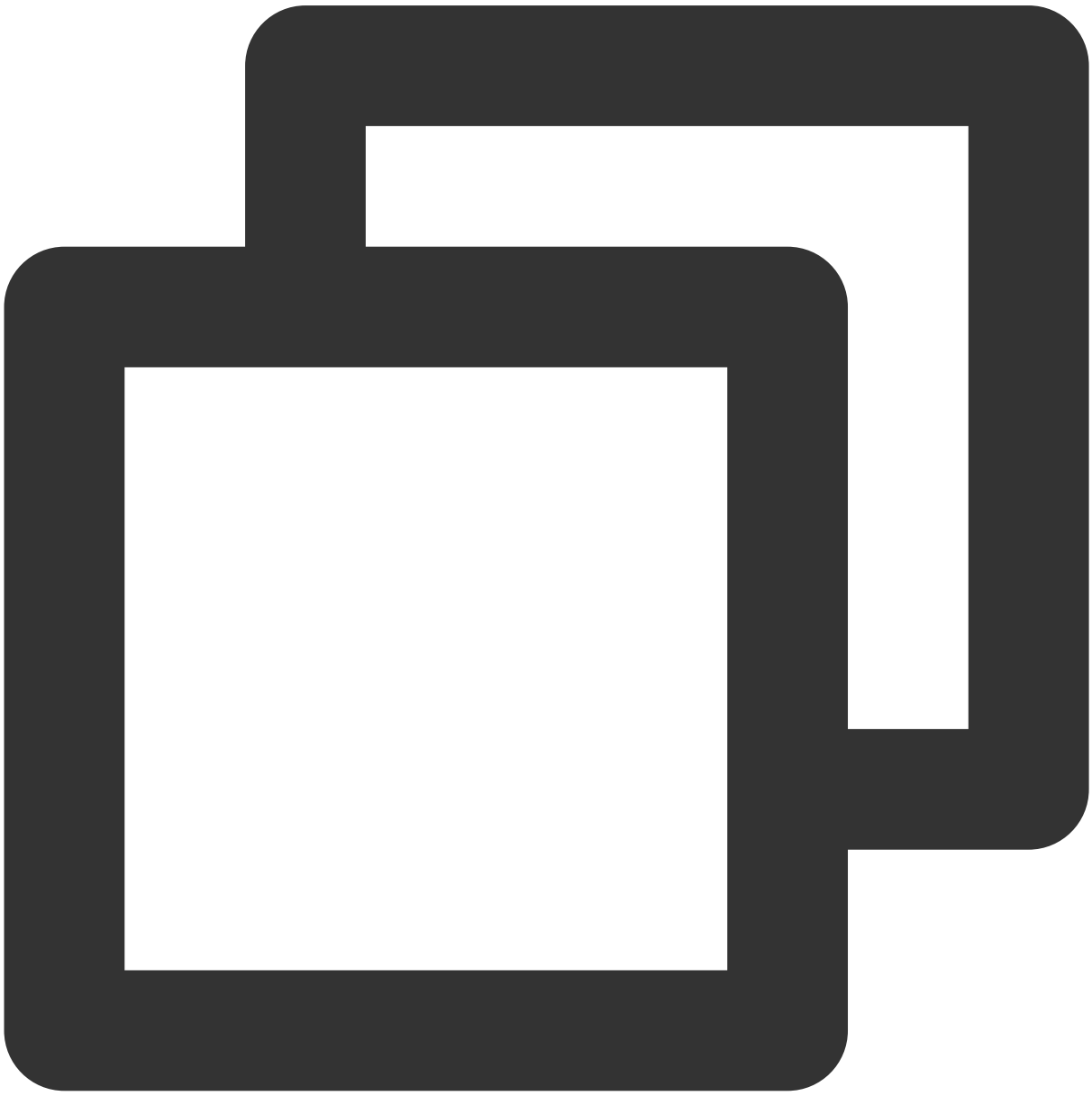
```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Transcode</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Profile>high</Profile>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Preset>medium</Preset>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Audio
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
        <EffectConfig>
            <EnableStartFadein>true</EnableStartFadein>
            <StartFadeinTime>3</StartFadeinTime>
            <EnableEndFadeout>false</EnableEndFadeout>
            <EndFadeoutTime>0</EndFadeoutTime>
            <EnableBgmFade>true</EnableBgmFade>
            <BgmFadeTime>1.7</BgmFadeTime>
        </EffectConfig>
    </AudioMixArray>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.mp3</Audio
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
        <EffectConfig>
            <EnableStartFadein>true</EnableStartFadein>
            <StartFadeinTime>3</StartFadeinTime>
            <EnableEndFadeout>false</EnableEndFadeout>
            <EndFadeoutTime>0</EndFadeoutTime>
            <EnableBgmFade>true</EnableBgmFade>
```

```
            <BgmFadeTime>1.7</BgmFadeTime>
        </EffectConfig>
    </AudioMixArray>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
```

```
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>Transcode</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Container>
            <Format>mp4</Format>
        </Container>
        <Video>
            <Codec>H.264</Codec>
            <Profile>high</Profile>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Fps>30</Fps>
            <Preset>medium</Preset>
        </Video>
        <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
        </Audio>
        <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
        </TransConfig>
        <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
        </TimeInterval>
        <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</A
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
```

```
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMixArray>
        <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.mp3</A
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMixArray>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

创建转码 pro 模板

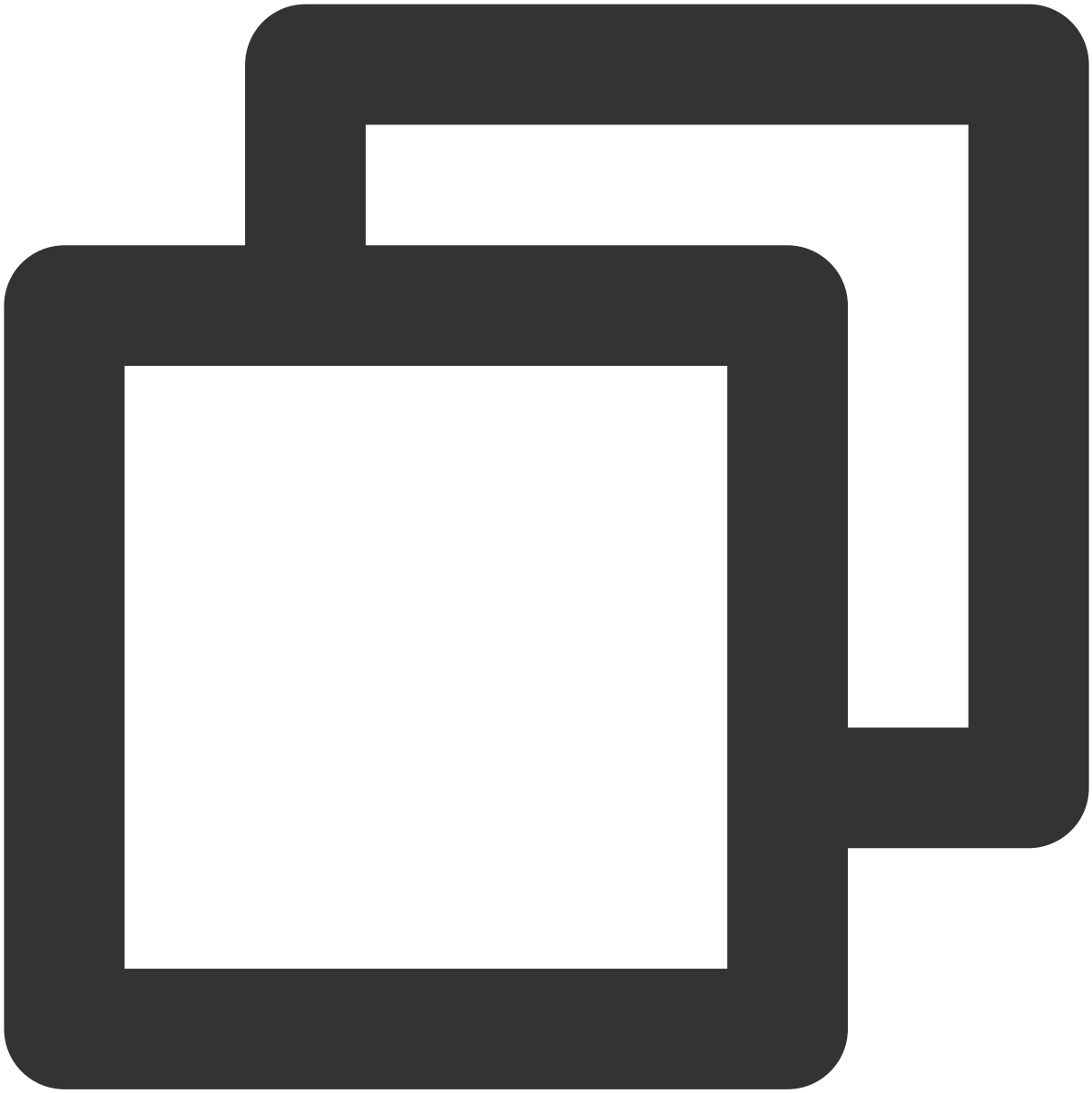
最近更新时间：2024-01-06 10:11:58

功能描述

创建转码 pro 模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

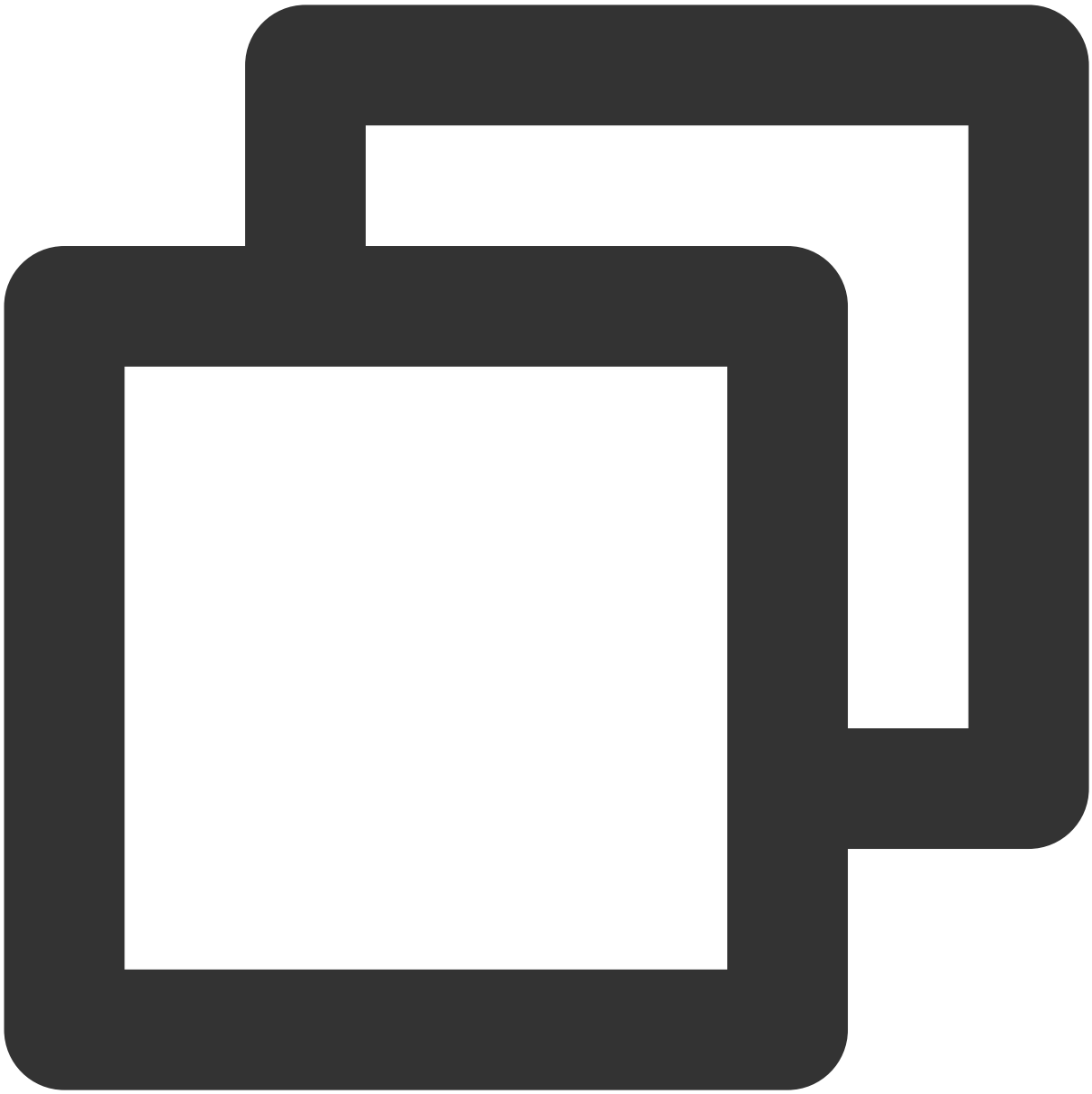
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>TranscodePro</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mxf</Format>
    </Container>
    <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
    </Video>
    <Audio>
        <Codec>pcm_s24le</Codec>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型：TranscodePro	String	是	无
Name	Request	模板名称 仅支持中文、英文、数字、_、-和*	String	是	无

Container	Request	容器格式	Container	是	无
Video	Request	视频信息	Container	是	无
TimeInterval	Request	时间区间	Container	否	无
Audio	Request	音频信息	Container	否	不传 Audio，相当于删除音频信息
TransConfig	Request	转码配置	Container	否	无

Container 类型 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Format	Request.Container	容器格式：mxf、mov、mkv	String	是

设定 container，音频视频支持的格式如下表：

Container	Audio Codecs	Video Codecs
mxf	pcm_s24le	xavc
mov、mkv	aac、mp3	apple_prores

Container 类型 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Video	编解码格式	String	是	无	xavc、apple_prores
Profile	Request.Video	视频算法器预置	String	是	无	取值范围见下表
Width	Request.Video	宽	String	是	无	取值范围见下表
Height	Request.Video	高	String	是	无	取值范围见下表
Interlaced	Request.Video	场模式	String	是	无	取值范围见下表
Fps	Request.Video	帧率	String	是	无	取值范围见下表
Bitrate	Request.Video	视频输出文件的码率	String	否	无	取值范围见下表

版权所有：腾讯云计算（北京）有限责任公司

Rotate	Request.Video	旋转角度	String	否	无	1. 值范围：[0, 360) 2. 单位：度

视频参数约束表:

注：'-' 表示留空。码率可选值与帧率可选值序号对应。

当 Video.Codec 为 xavc 时：

Profile	Width	Height	Interlaced	Fps	Bitrate(单位:kbps)
XAVC-HD_intra_420_10bit_class50	1440	1080	true	1、25 2、30000/1001	-
XAVC-HD_intra_420_10bit_class50	1440	1080	false	1、25 2、24000/1001 3、30000/1001	-
XAVC-HD_intra_422_10bit_class100	1280	720	false	1、50 2、60000/1001	-
XAVC-HD_intra_422_10bit_class100	1920	1080	true	1、25 2、30000/1001	-
XAVC-HD_intra_422_10bit_class100	1920	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-HD_intra_422_10bit_class200	1920	1080	true	1、25 2、30000/1001	-
XAVC-HD_intra_422_10bit_class200	1920	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class100	2048	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class300	3840	2160	false	1、25 2、50 3、24000/1001	-

版权所有：腾讯云计算（北京）有限责任公司

				4、30000/1001 5、60000/1001	
XAVC-4K_intra_422_10bit_class300	4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class480	3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class480	4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit	2048	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	1、93000 2、185000 3、89000 4、111000 5、222000
XAVC-4K_intra_422_10bit	3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	1、250000或400000 2、500000或800000 3、240000或384000 4、300000或480000 5、600000或960000
XAVC-4K_intra_422_10bit	4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	1、250000或400000 2、500000或800000 3、240000或384000 4、300000或480000

版权所有：腾讯云计算（北京）有限责任公司

					5、600000或960000
XAVC-HD_422_10bit	1280	720	false	1、50 2、60000/1001	1、50000 2、50000
XAVC-HD_422_10bit	1920	1080	true	1、25 2、30000/1001	1、25000或35000或50000 2、25000或35000或50000
XAVC-HD_422_10bit	1920	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	1、25000或35000或50000 2、35000或50000 3、25000或35000或50000 4、25000或35000或50000 5、35000或50000
XAVC-4K_422_10bit	3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	1、100000或140000或200000 2、140000或200000 3、100000或140000或200000 4、100000或140000或200000 5、140000或200000
XAVC-4K_420_8bit	3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	1、188000 2、300000 3、188000 4、188000 5、300000

当 Video.Codec 为 apple_prores 时：

Profile
1.ProRes_422_Proxy 2.ProRes_422_LT 3.ProRes_422 4.ProRes_422_HQ 5.ProRes_4444

版权所有：腾讯云计算（北京）有限责任公司

6.ProRes_4444_XQ
7.ProRes_4444_alpha
8.ProRes_4444_XQ_alpha

Width	Height	Interlaced	Fps	Bitrate(单位:kbps)
720	486	false	1、24000/1001 2、30000/1001	取值范围: [2000,3000000]
720	486	true	1、60000/1001	取值范围: [2000,3000000]
720	576	false	1、25	取值范围: [2000,3000000]
720	576	true	1、50	取值范围: [2000,3000000]
960	720	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
1280	720	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
1280	1080	false	1、24000/1001 2、30000/1001	取值范围: [2000,3000000]
1280	1080	true	1、60000/1001	取值范围: [2000,3000000]
1440	1080	false	1、24000/1001 2、25 3、30000/1001	取值范围: [2000,3000000]
1440	1080	true	1、50 2、60000/1001	取值范围: [2000,3000000]
1920	1080	false	1、25 2、50 3、24000/1001	取值范围: [2000,3000000]

版权所有：腾讯云计算（北京）有限责任公司

			4、30000/1001 5、60000/1001	
1920	1080	true	1、50 2、60000/1001	取值范围: [2000,3000000]
2048	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
2048	1556	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
5120	2700	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
6144	3240	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]
8192	4320	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围: [2000,3000000]

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.TimeInterval	开始时间	String	否	0	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒
Duration	Request.TimeInterval	持续时间	String	否	视频原始时长	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒

Container 类型 Audio 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Audio	编解码格式	String	是	无	pcm_s24le、aac、mp3
Remove	Request.Audio	是否删除源音频流	String	否	false	取值 true、false

Container 类型 TransConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
AdjDarMethod	Request.TransConfig	分辨率调整方式	String	否	none	取值 scale、crop、pad、none

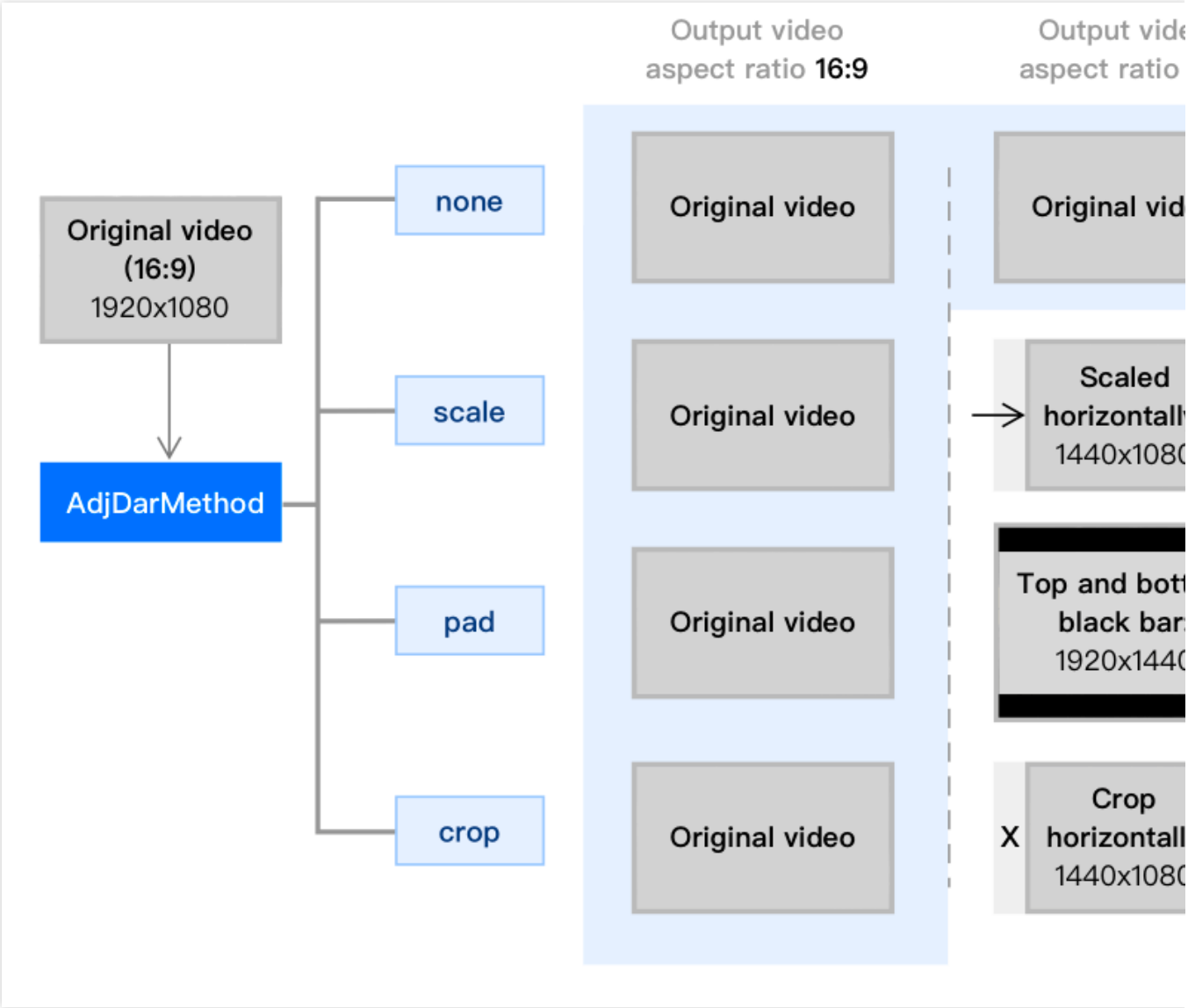
版权所有：腾讯云计算（北京）有限责任公司

						当输出视频的宽高比与原视频不等时，根据此参数做分辨率的相应调整
IsCheckReso	Request.TransConfig	是否检查分辨率	String	否	false	rue、false 当为 false时，按照配置参数转码
ResoAdjMethod	Request.TransConfig	分辨率调整方式	String	否	0	取值0、1；0 表示使用原视频分辨率；1表示返回转码失败 当 IsCheckReso 为 true 时生效
IsCheckVideoBitrate	Request.TransConfig	是否检查视频码率	String	否	false	true、false 当为 false 时，按照配置参数转码
VideoBitrateAdjMethod	Request.TransConfig	视频码率调整方式	String	否	0	取值0、1；当输出视频码率大于原视频码率时，0表示使用原视频码率；1表示返回转码失败 当 IsCheckVideoBitrate 为 true 时生效
IsCheckAudioBitrate	Request.TransConfig	是否检查音频码率	String	否	false	true、false 当为 false 时，按照配置参数转码
AudioBitrateAdjMethod	Request.TransConfig	音频码率调整方式	String	否	0	取值0、1；当输出音频码率大于原音频码率时，0 表示使用原音频码率；1表示返回转码失败 当 IsCheckAudioBitrate 为 true 时生效
IsCheckVideoFps	Request.TransConfig	是否检查视频帧率	String	否	false	true、false 当为 false 时，按照配置参数转码
VideoFpsAdjMethod	Request.TransConfig	视频帧率调整方式	String	否	0	取值0、1；当输出视频帧率大于原视频

版权所有：腾讯云计算（北京）有限责任公司

						帧率时，0表示使用原视频帧率；1表示返回转码失败 当 IsCheckVideoFps 为 true 时生效
DeleteMetadata	Request.TransConfig	是否删除文件中的 MetaData 信息	String	否	false	true、false 当为 false 时，保留源文件信息
IsHdr2Sdr	Request.TransConfig	是否开启 HDR转 SDR	String	否	false	true/false

AdjDarMethod 参数图示：



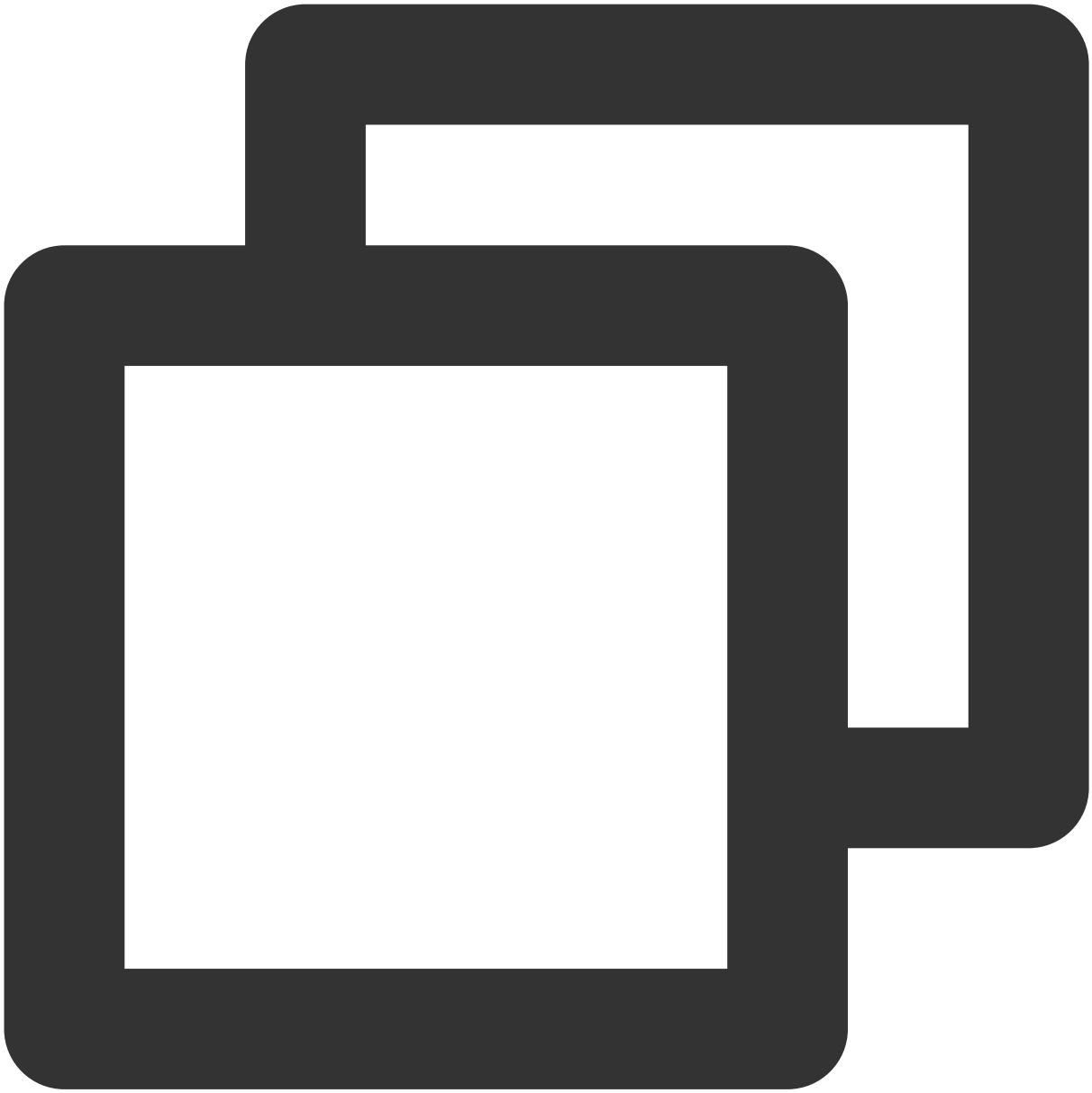
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>TranscodePro</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TransProTpl>
            <Container>
                <Format>mxf</Format>
            </Container>
```

```
            <Video>
                <Codec>xavc</Codec>
                <Profile>XAVC-HD_422_10bit</Profile>
                <Width>1920</Width>
                <Height>1080</Height>
                <Interlaced>true</Interlaced>
                <Fps>30000/1001</Fps>
                <Bitrate>50000</Bitrate>
            </Video>
            <Audio>
                <Codec>pcm_s24le</Codec>
            </Audio>
            <TransConfig>
                <AdjDarMethod>scale</AdjDarMethod>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransProTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，	String

		TranscodePro	
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
TransProTpl	Response.Template	详细的模板参数	Container

Container 节点 TransProTpl 的内容：

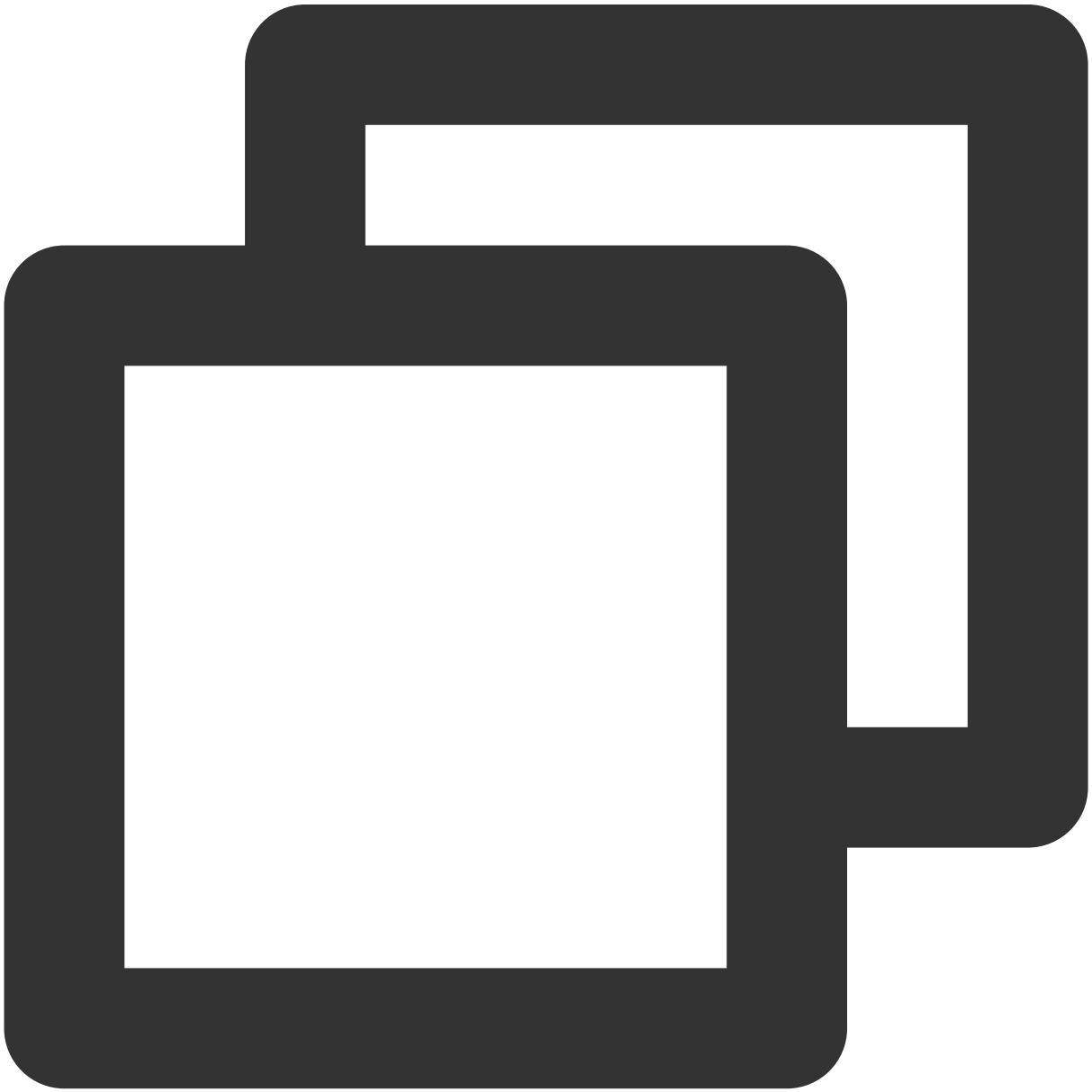
节点名称（关键字）	父节点	描述	类型
TimeInterval	Response.Template.TransProTpl	同请求体中的 Request.TimeInterval	Container
Container	Response.Template.TransProTpl	同请求体中的 Request.Container	Container
Video	Response.Template.TransProTpl	同请求体中的 Request.Video	Container
Audio	Response.Template.TransProTpl	同请求体中的 Request.Audio	Container
TransConfig	Response.Template.TransProTpl	同请求体中的 Request.TransConfig	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

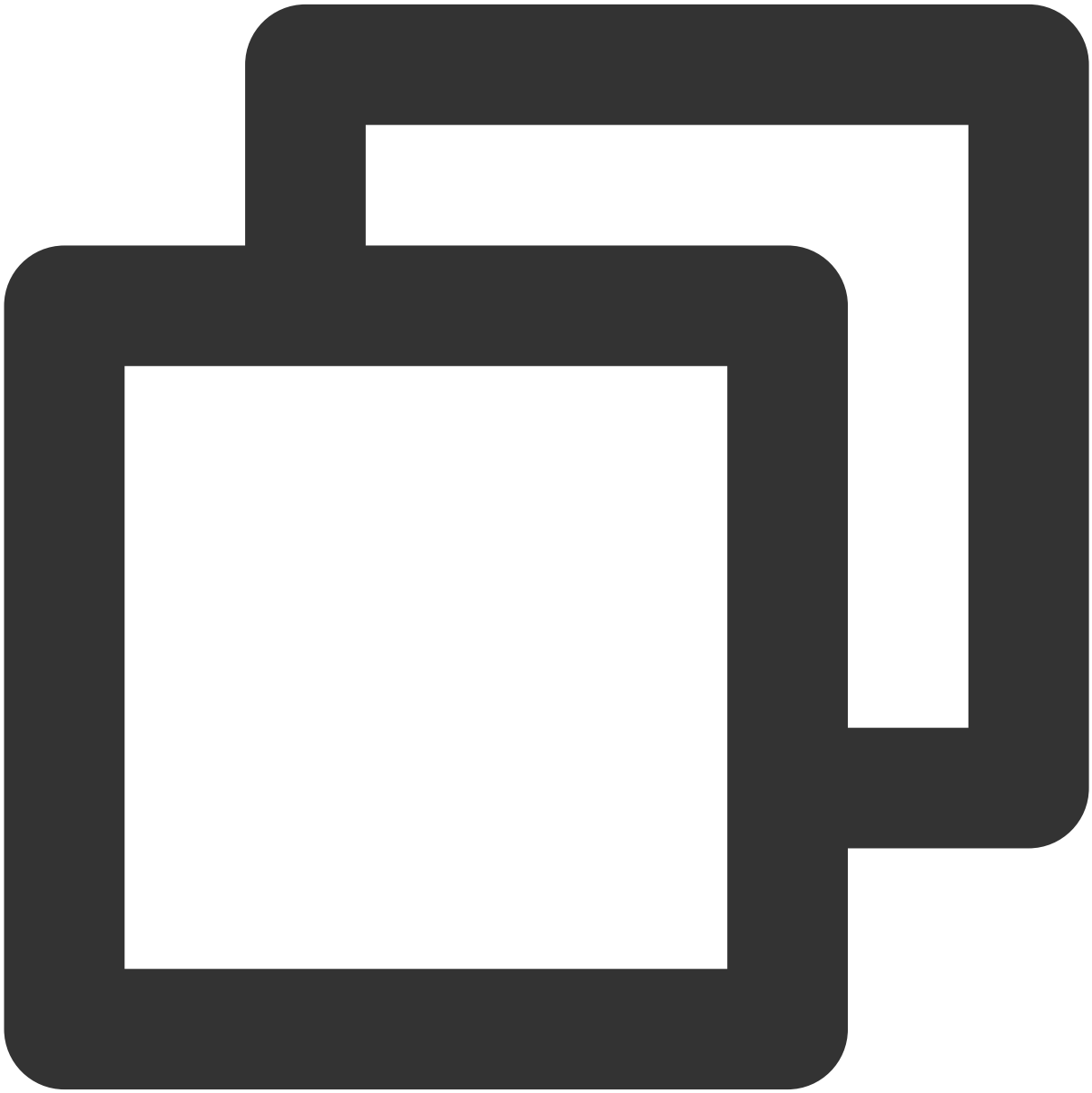
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>TranscodePro</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mxf</Format>
```

```
    </Container>
    <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
    </Video>
    <Audio>
        <Codec>pcm_s24le</Codec>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>TranscodePro</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TransProTpl>
            <Container>
                <Format>mxf</Format>
            </Container>
            <Video>
                <Codec>xavc</Codec>
                <Profile>XAVC-HD_422_10bit</Profile>
                <Width>1920</Width>
                <Height>1080</Height>
                <Interlaced>true</Interlaced>
                <Fps>30000/1001</Fps>
                <Bitrate>50000</Bitrate>
            </Video>
            <Audio>
                <Codec>pcm_s24le</Codec>
            </Audio>
            <TransConfig>
                <AdjDarMethod>scale</AdjDarMethod>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransProTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

创建语音合成模板

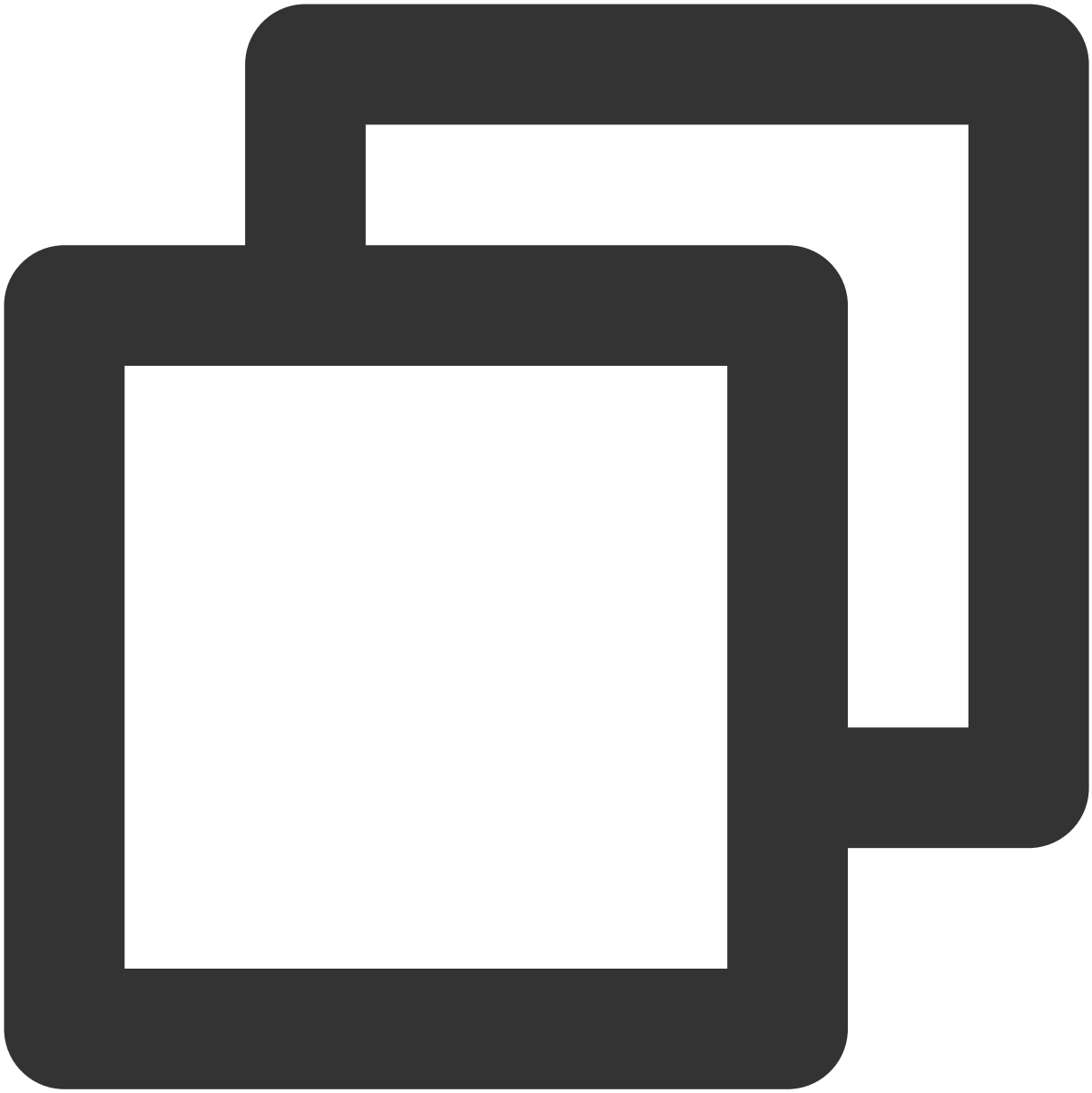
最近更新时间：2024-01-06 10:11:58

功能描述

创建语音合成模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

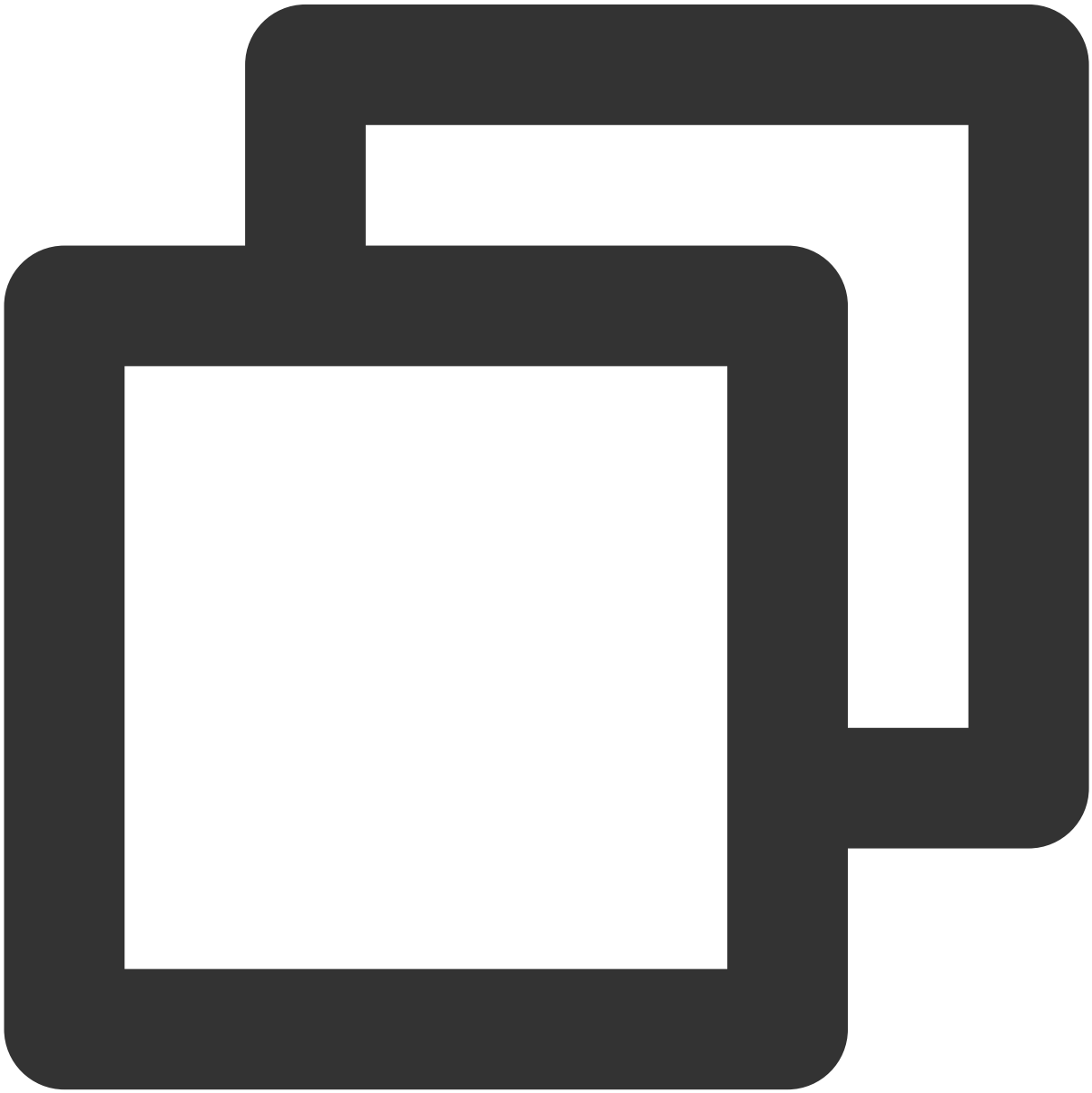
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Tts</Tag>
    <Name>TemplateName</Name>
    <Mode>Sync</Mode>
    <Codec>pcm</Codec>
    <VoiceType>ruxue</VoiceType>
    <Volume>2</Volume>
    <Speed>200</Speed>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型：Tts	String	是	Tts
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*	String	是	无
Mode	Request	处理模式，默认值 Asyc	String	否	Asyc（异步合成）、Sync（同步合成） 当选择 Asyc 时，codec 只支持 pcm
Codec	Request	音频格式，默认值 wav（同步）/pcm（异步）	String	否	wav、mp3、pcm
VoiceType	Request	音色，默认值 ruxue	String	否	ruxue、aixiaonan、aixiaoxing
Volume	Request	音量，默认值0	String	否	[-10,10]
Speed	Request	语速，默认值100	String	否	[50,200]

音色说明

名称	Voice 参数值	类型	适用场景	支持语言	声音品质
茹雪	ruxue	标准女声	通用场景	中文及中英文	标准版

版权所有：腾讯云计算（北京）有限责任公司

				混合	
爱小楠	aixiaonan	甜美女声	通用/社交场景	中文及中英文混合	精品版
爱小星	aixiaoxing	解说男声	通用/解说场景	中文及中英文混合	精品版

多情感音色说明

名称	Voice 参数值	情绪分类
爱小星	aixiaoxing	中性、广播、平静、兴奋

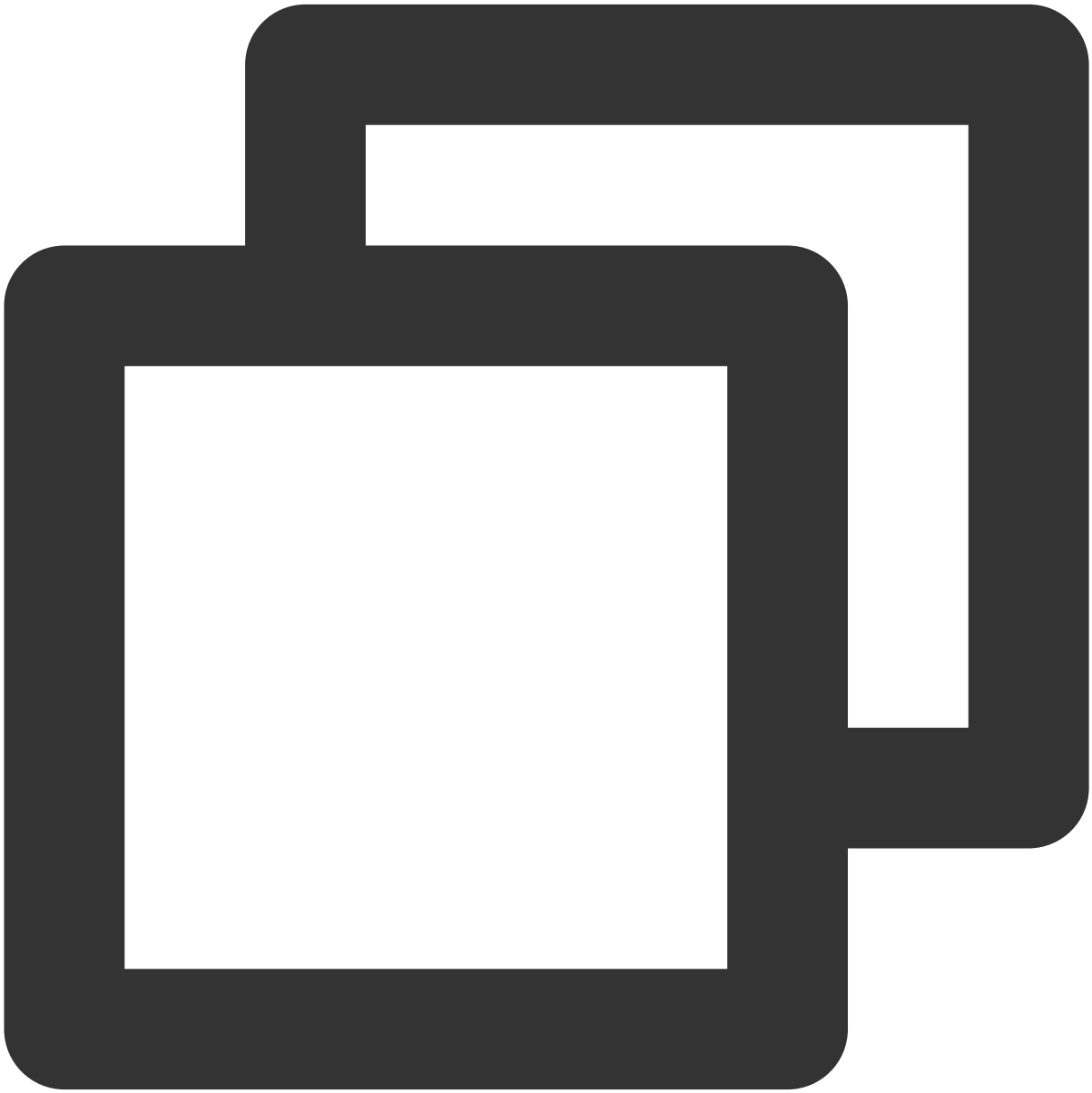
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Tts</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TtsTpl>
            <Mode>Sync</Mode>
            <Codec>pcm</Codec>
            <VoiceType>ruxue</VoiceType>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Volume>2</Volume>
            <Speed>200</Speed>
        </TtsTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，Tts	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
TtsTpl	Response.Template	详细的模板参数	Container

Container节点 TtsTpl 的内容：

节点名称（关键字）	父节点	描述
Mode	Response.Template.TtsTpl	同请求体中的 Request.Mode
Codec	Response.Template.TtsTpl	同请求体中的 Request.Codec
VoiceType	Response.Template.TtsTpl	同请求体中的 Request.VoiceType
Volume	Response.Template.TtsTpl	同请求体中的 Request.Volume
Speed	Response.Template.TtsTpl	同请求体中的 Request.Speed

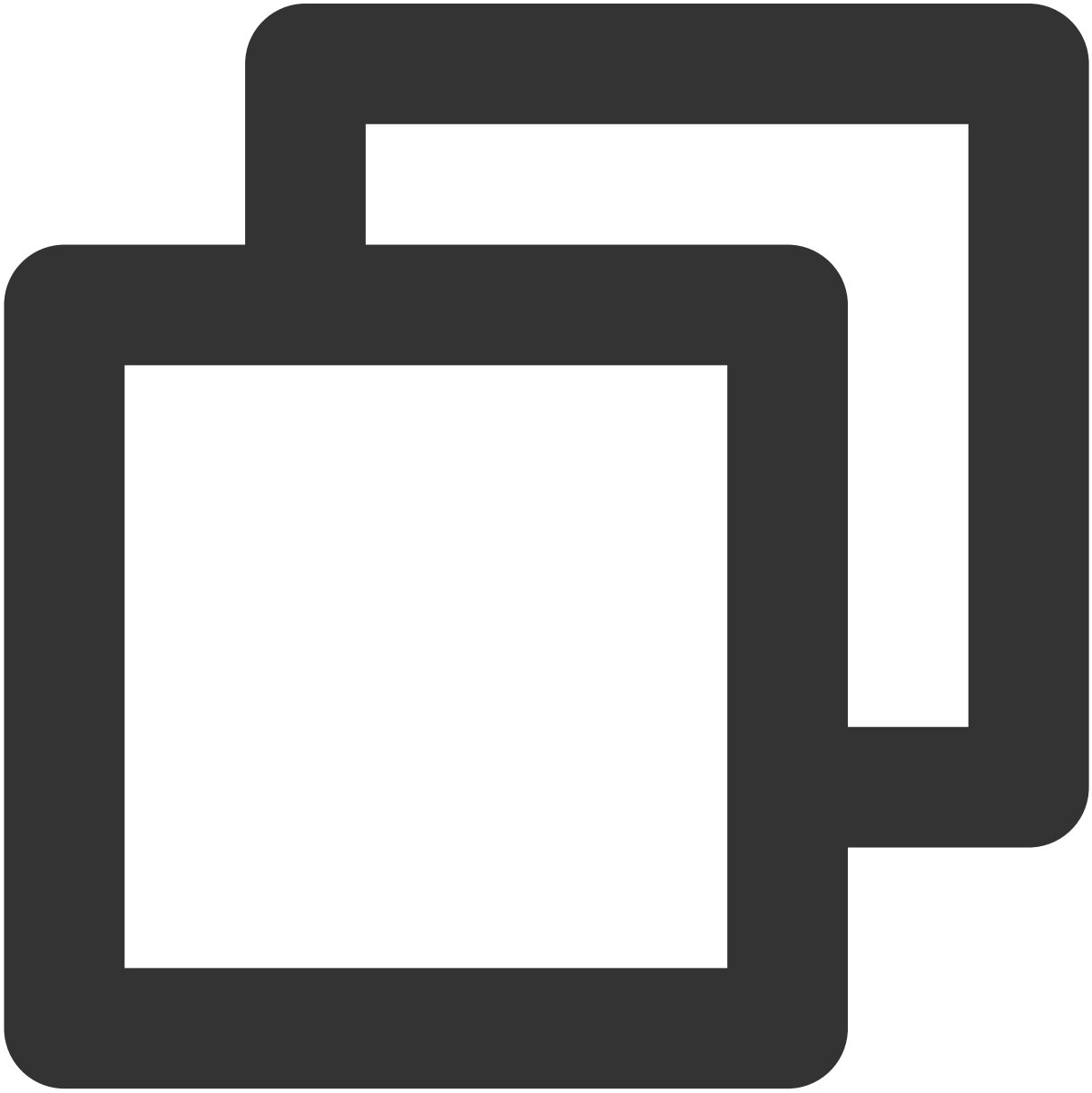
版权所有：腾讯云计算（北京）有限责任公司

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

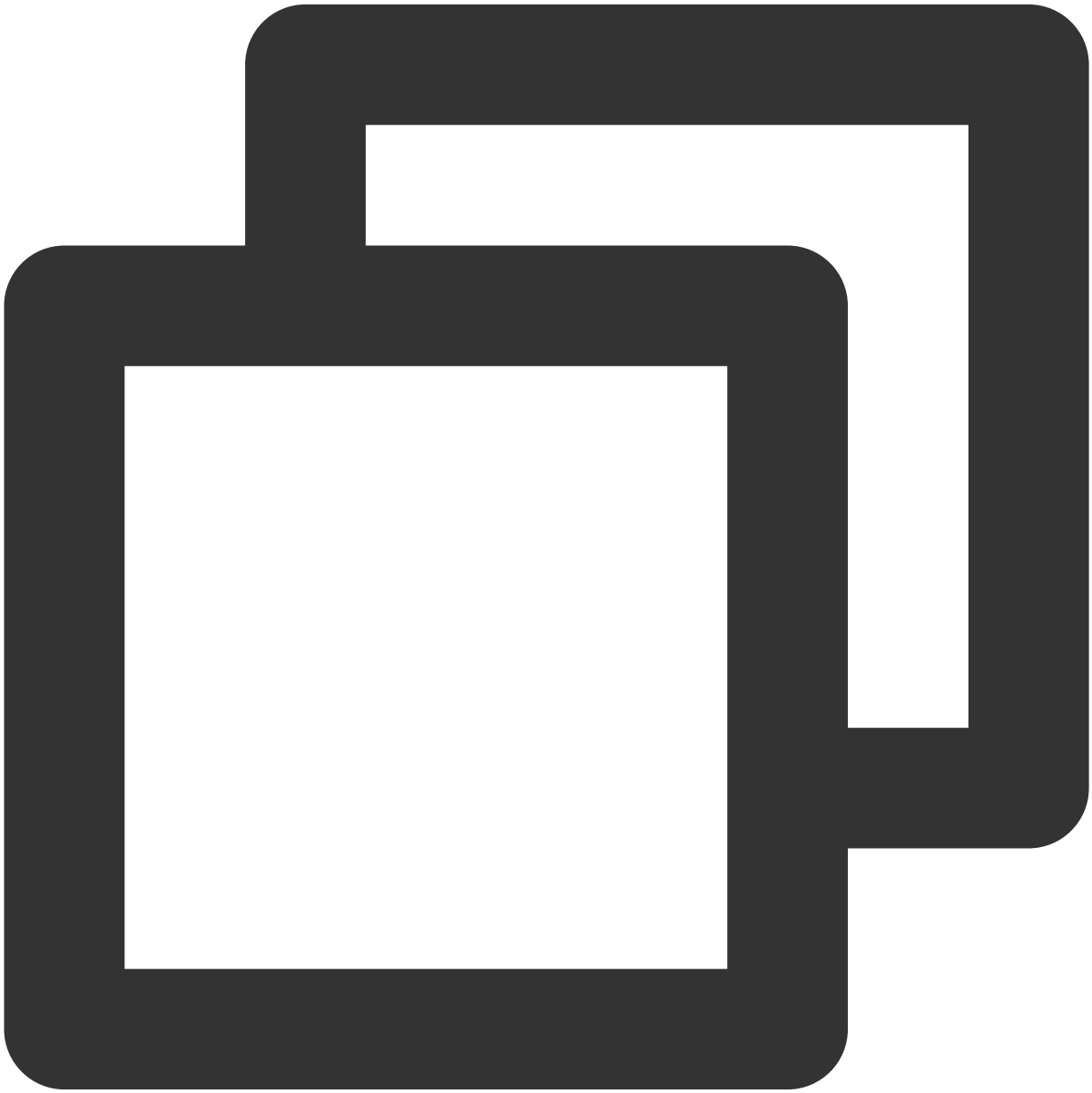
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
```

```
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Tts</Tag>
    <Name>TemplateName</Name>
    <Mode>Sync</Mode>
    <Codec>pcm</Codec>
    <VoiceType>ruxue</VoiceType>
    <Volume>2</Volume>
    <Speed>200</Speed>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>Tts</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TtsTpl>
            <Mode>Sync</Mode>
            <Codec>pcm</Codec>
            <VoiceType>ruxue</VoiceType>
            <Volume>2</Volume>
            <Speed>200</Speed>
        </TtsTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

创建精彩集锦模板

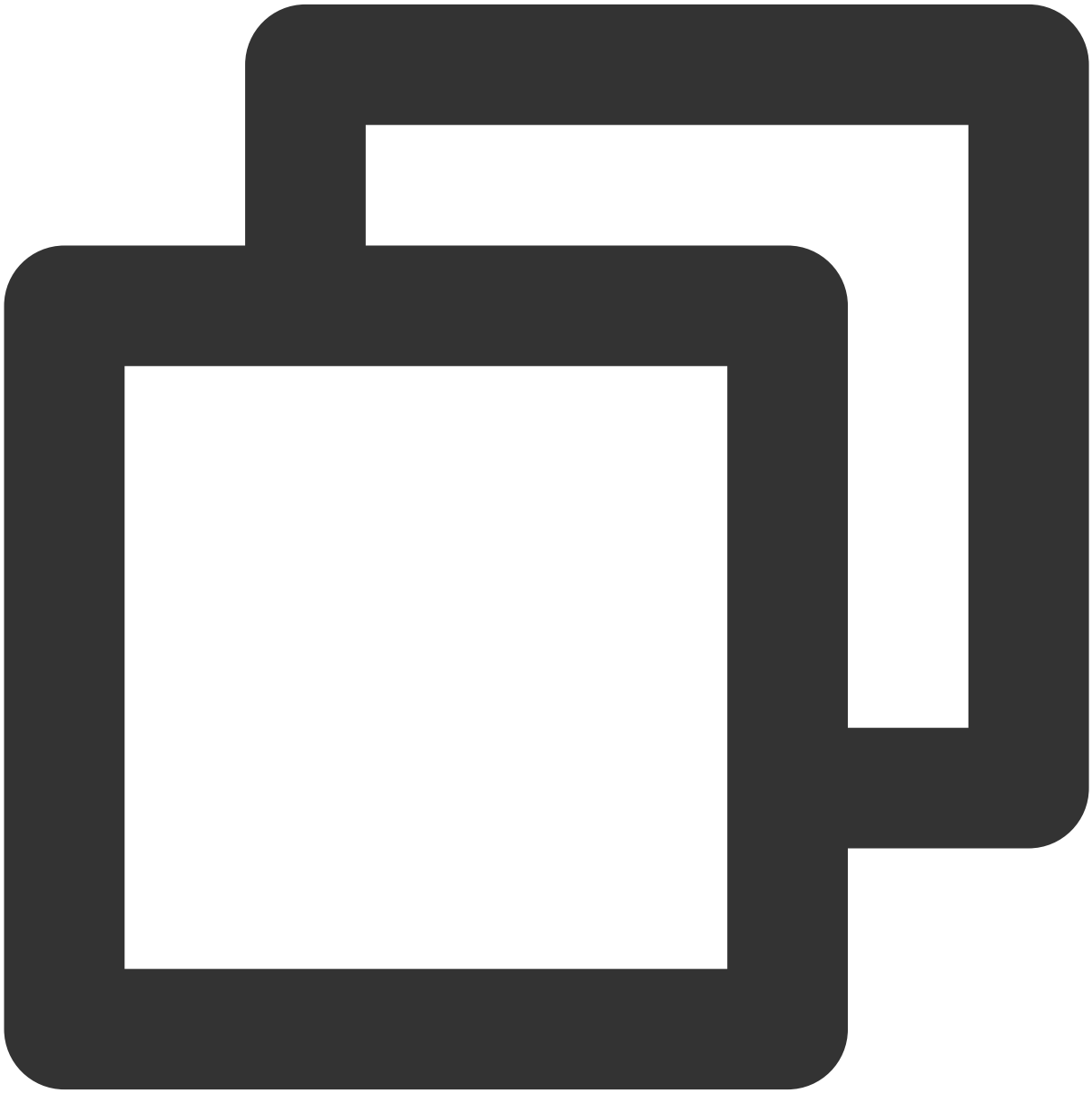
最近更新时间：2024-01-06 10:11:58

功能描述

创建精彩集锦模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

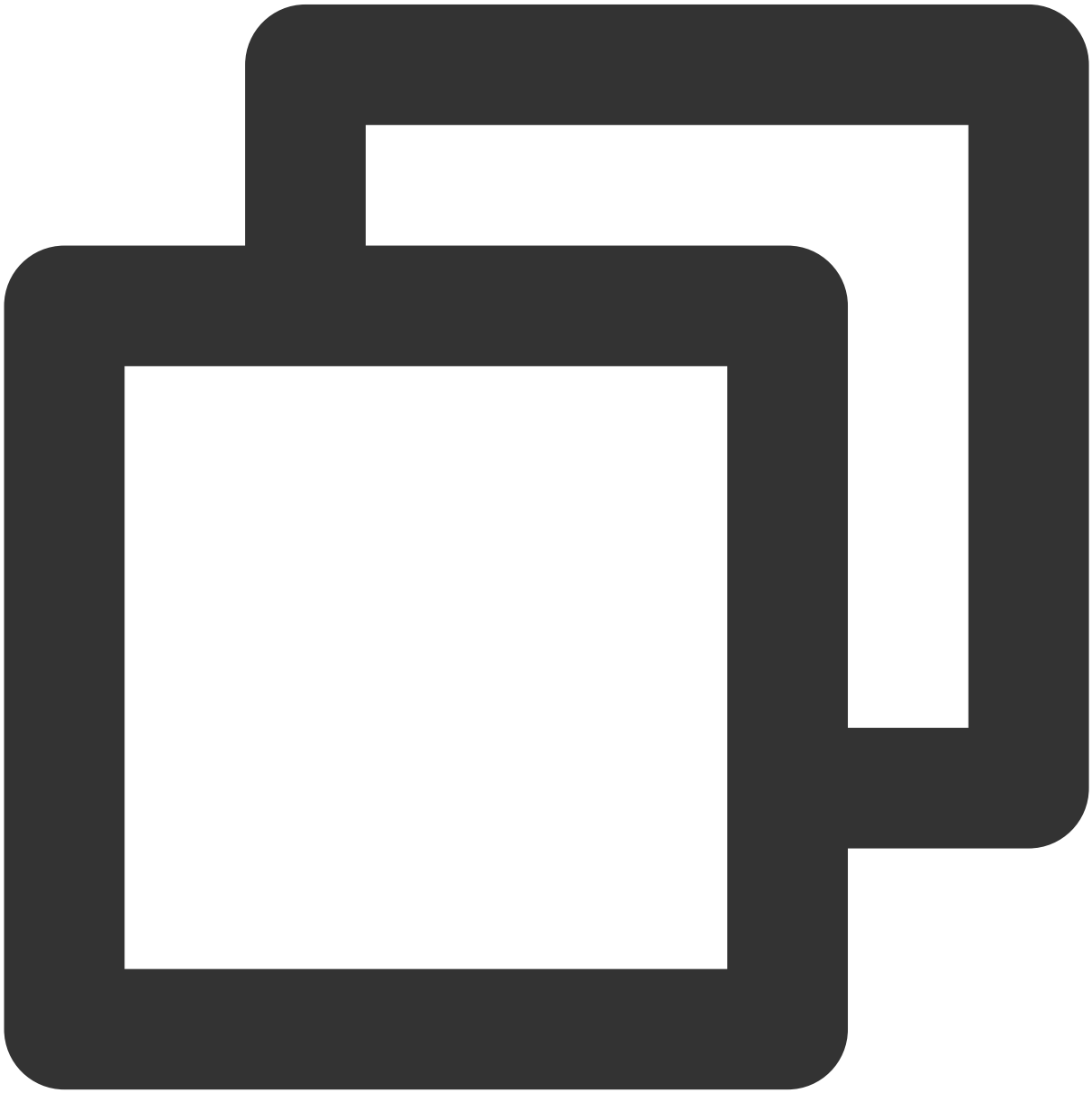
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>VideoMontage</Tag>
    <Name>TemplateName</Name>
    <Duration>10.5</Duration>
    <Container>
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
        <Remove>false</Remove>
    </Audio>
    <AudioMix>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</AudioS
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
    </AudioMix>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型: VideoMontage	String	是	无
Name	Request	模板名称 仅支持中文、英文、数字、_、-和*	String	是	无

Duration	Request	集锦时长	String	否	1. 默认自动分析时长 2. 单位为秒 3. 支持 float 格式，执行精度精确到毫秒
Container	Request	容器格式	Container	是	无
Video	Request	视频信息	Container	是	无
Audio	Request	音频信息	Container	否	无
AudioMix	Request	混音参数，详情见 AudioMix	Container 数组	否	仅在 Audio.Remove 为 false 时生效

Container 类型 Container 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Format	Request.Container	容器格式: mp4、flv、hls、ts、mkv	String	是

设定 container，音频视频支持的格式如下表：

Container	Audio Codecs	Video Codecs
mp4/ts/hls/mkv	AAC、MP3	H.264、H.265
flv	AAC、MP3	H.264

Container 类型 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Video	编解码格式	String	否	H.264	1. H.264 2.H.265
Width	Request.Video	宽	String	否	视频原始	1. 值范围：[128，4096]

版权所有：腾讯云计算（北京）有限责任公司

					宽度	2. 单位：px 3. 若只设置 Width 时，按照视频原始比例计算 Height 4. 必须为偶数
Height	Request.Video	高	String	否	视频原始高度	1. 值范围：[128，4096] 2. 单位：px 3. 若只设置 Height 时，按照视频原始比例计算 Width 4. 必须为偶数
Fps	Request.Video	帧率	String	否	无	1. 值范围：(0，60] 2. 单位：fps
Bitrate	Request.Video	视频输出文件的码率	String	否	无	1. 值范围：[10，50000] 2. 单位：Kbps
Crf	Request.Video	码率-质量控制因子	String	否	无	1. 值范围：(0, 51] 2. 如果设置了 Crf，则 Bitrate 的设置失效 3. 当 Bitrate 为空时，默认为25
Rotate	Request.Video	旋转角度	String	否	无	1. 值范围：[0, 360) 2. 单位：度

Container 类型 Audio 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Codec	Request.Audio	编解码格式	String	否	aac	取值 aac、mp3
Samplerate	Request.Audio	采样率	String	否	44100	1. 单位：Hz 2. 可选 11025、22050、32000、44100、48000、96000 3. 不同的封装，mp3 支持不同的采样率，如下表所示

版权所有：腾讯云计算（北京）有限责任公司

Bitrate	Request.Audio	原始音频码率	String	否	无	1. 单位：Kbps 2. 值范围：[8，1000]
Channels	Request.Audio	声道数	String	否	无	1. 当 Codec 设置为 aac，支持1、2、4、5、6、8 2. 当 Codec 设置为 mp3，支持1、2
Remove	Request.Audio	是否删除音频流	String	否	false	取值 true、false

说明

Y 表示支持这种采样率，N 表示不支持。

封装格式/音频采样率	11025	22050	32000	44100	48000	96000
flv	N	Y	N	Y	N	N
mp4	N	Y	Y	Y	Y	N
hls/ts/mkv	Y	Y	Y	Y	Y	N

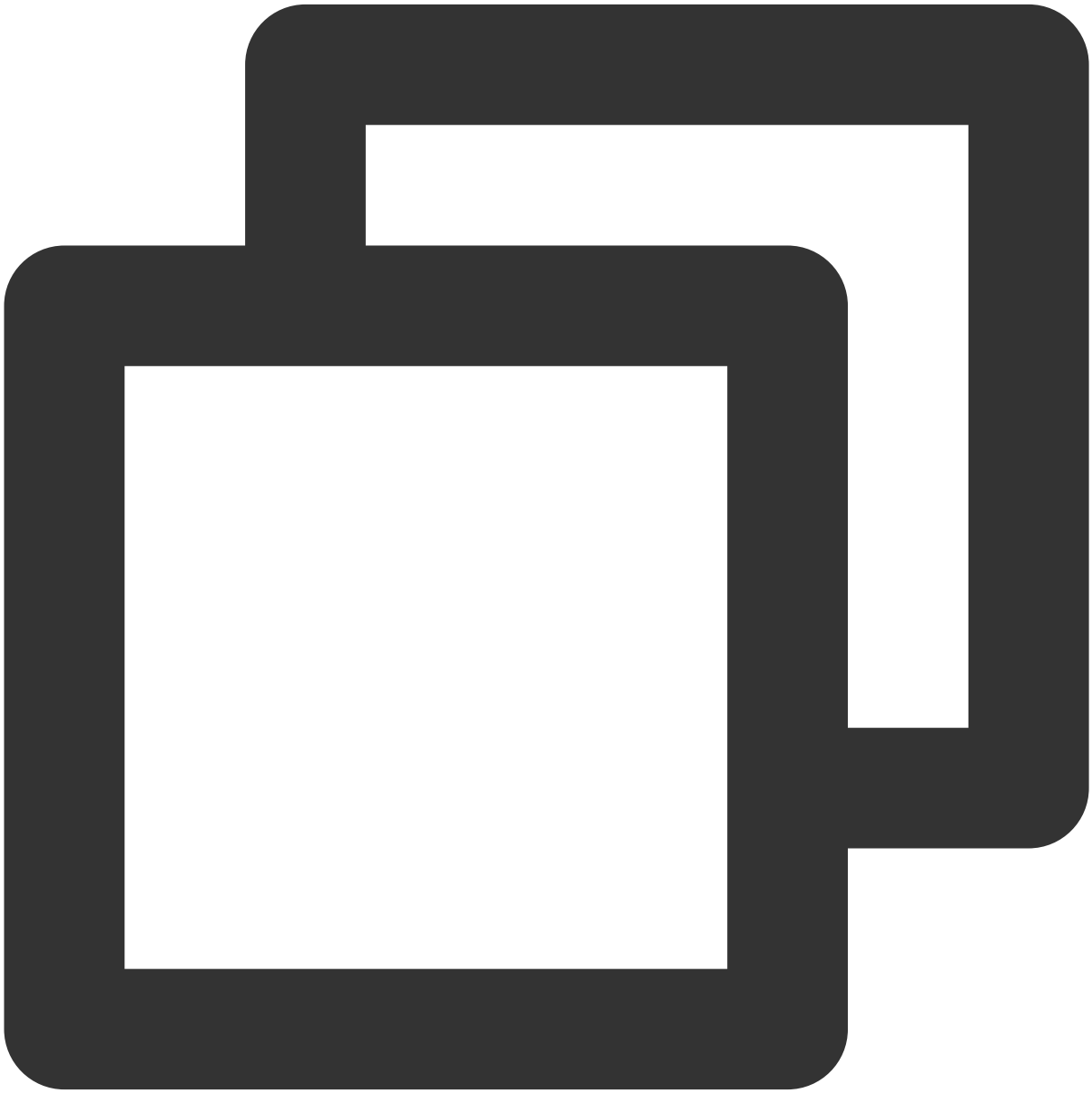
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>VideoMontage</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VideoMontage>
            <Duration>10.5</Duration>
            <Container>
                <Format>mp4</Format>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
                <Remove>false</Remove>
            </Audio>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
            </AudioMix>
        </VideoMontage>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，VideoMontage	String
UpdateTime	Response.Template	更新时间	String

CreateTime	Response.Template	创建时间	String
VideoMontage	Response.Template	详细的模板参数	Container

Container节点VideoMontage的内容：

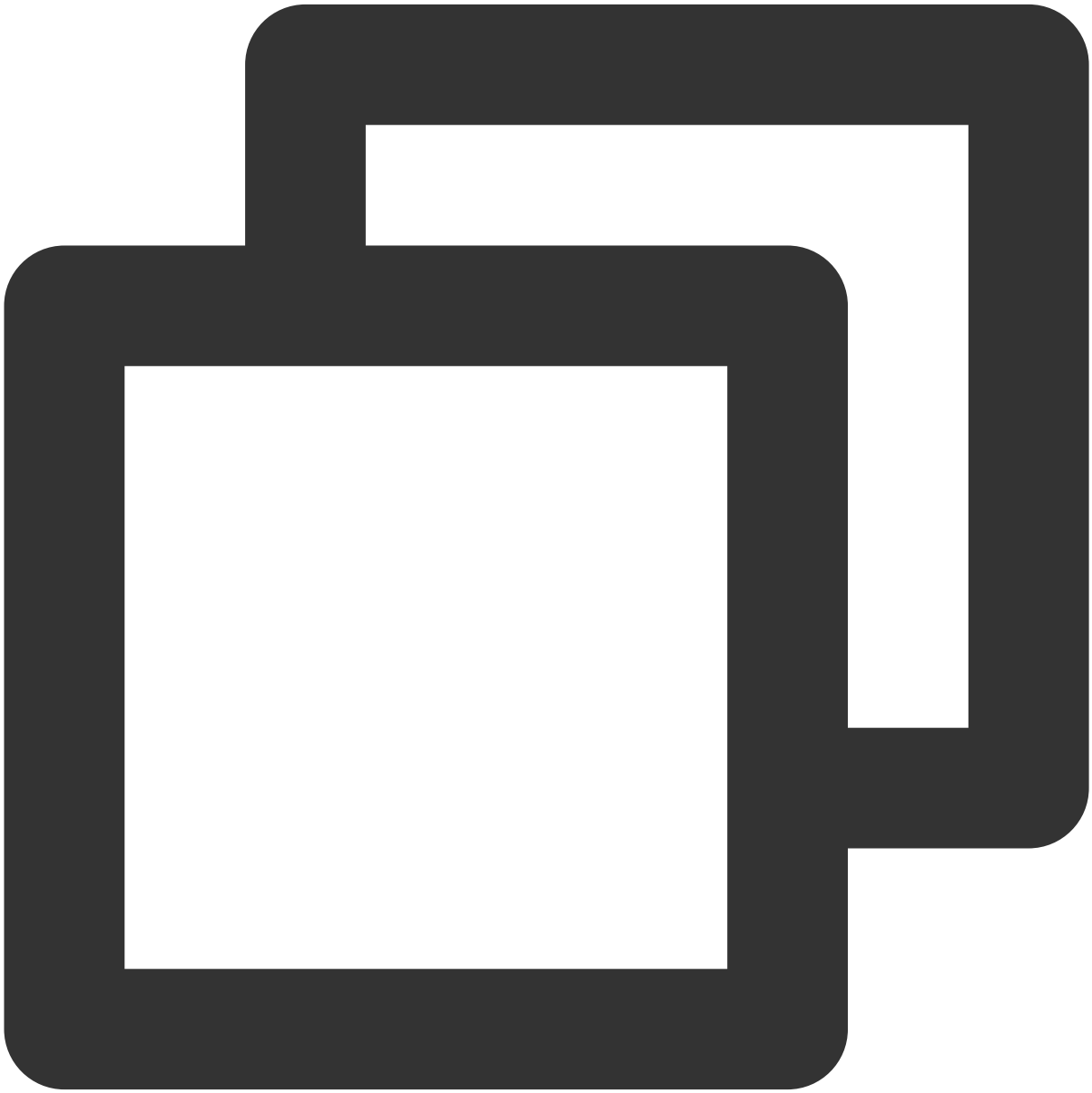
节点名称（关键字）	父节点	描述
Duration	Response.TemplateList.VideoMontage	同请求体中的 Request.Duration
TimeInterval	Response.TemplateList.VideoMontage	同请求体中的 Request.TimeInterval
Container	Response.TemplateList.VideoMontage	同请求体中的 Request.Container
Video	Response.TemplateList.VideoMontage	同请求体中的 Request.Video
Audio	Response.TemplateList.VideoMontage	同请求体中的 Request.Audio
AudioMix	Response.TemplateList.VideoMontage	同请求体中的 Request.AudioMix

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

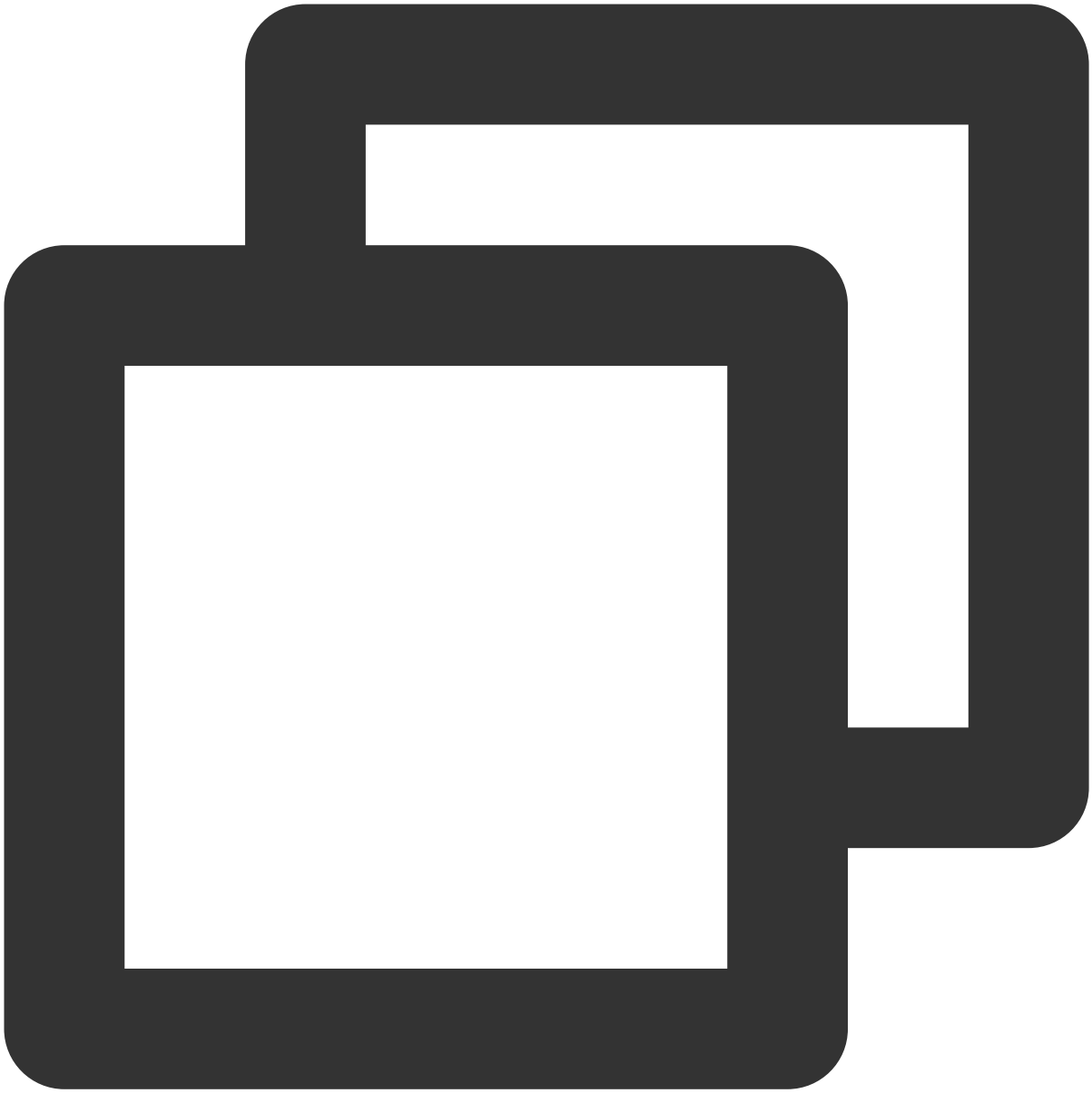
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoMontage</Tag>
    <Name>TemplateName</Name>
    <Duration>10.5</Duration>
    <Container>
```

```
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
        <Remove>false</Remove>
    </Audio>
    <AudioMix>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</AudioS
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
    </AudioMix>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>VideoMontage</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VideoMontage>
            <Duration>10.5</Duration>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
                <Remove>false</Remove>
            </Audio>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
            </AudioMix>
        </VideoMontage>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

创建视频增强模板

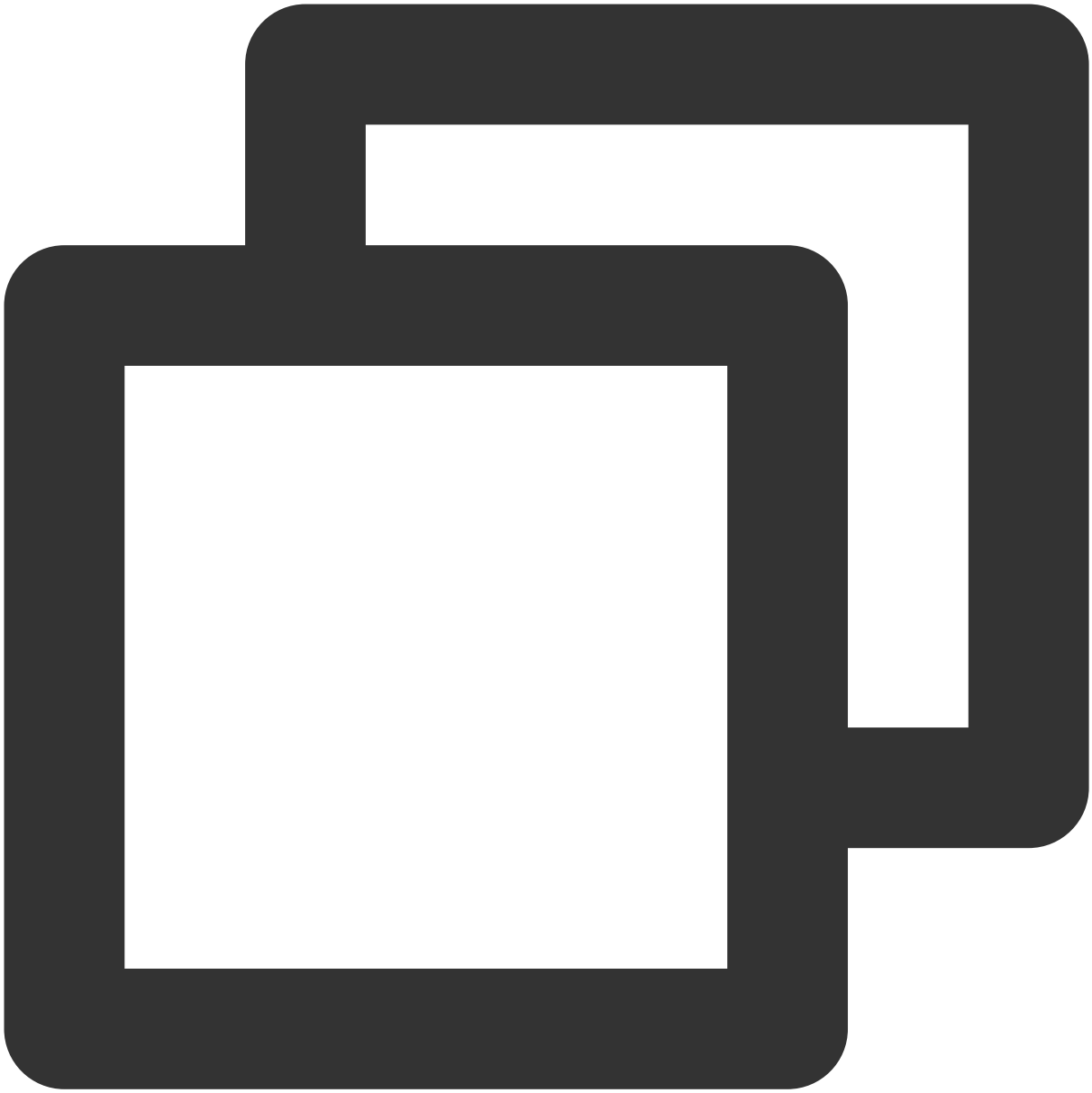
最近更新时间：2024-01-06 10:11:58

功能描述

创建视频增强模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

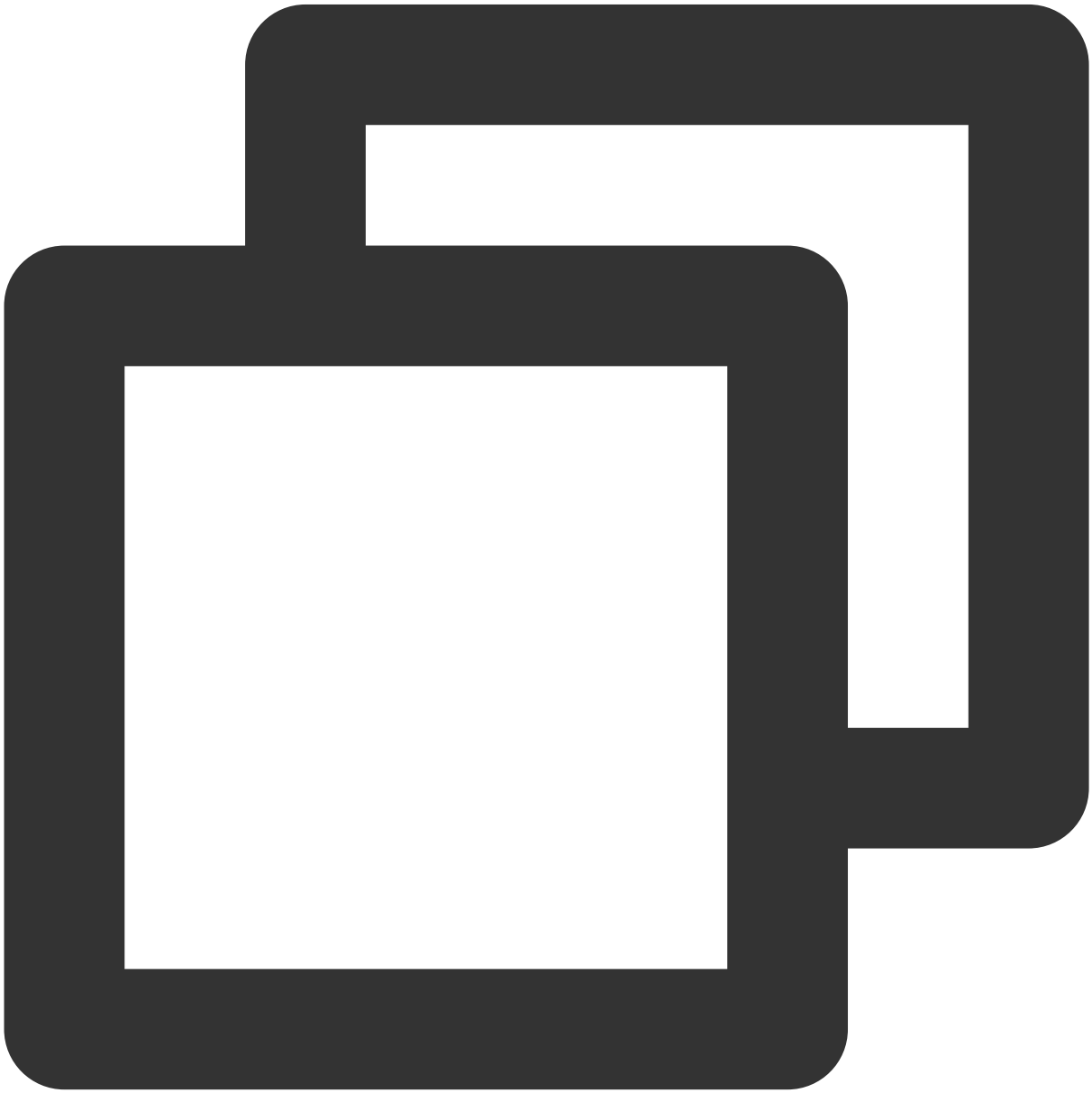
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>VideoProcess</Tag>
    <Name>TemplateName</Name>
    <ColorEnhance>
        <Enable>true</Enable>
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
    </ColorEnhance>
    <MsSharpen>
        <Enable>true</Enable>
        <SharpenLevel></SharpenLevel>
    </MsSharpen>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型：VideoProcess	String	是	无
Name	Request	模板名称 仅支持中文、英文、数字、_、-和*	String	是	无
ColorEnhance	Request	色彩增强	Container	否，ColorEnhance 和 MsSharpen 不能同时为空	无
MsSharpen	Request	细节增强	Container	否，ColorEnhance 和 MsSharpen 不能同时为空	无

Container 类型 ColorEnhance 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Enable	Request.ColorEnhance	色彩增强是否开启	String	否	false	true、false

Contrast	Request.ColorEnhance	对比度	String	否	无	值范围：[0, 100]，空字符串（表示自动分析）
Correction	Request.ColorEnhance	色彩校正	String	否	无	值范围：[0, 1000]，空字符串（表示自动分析）
Saturation	Request.ColorEnhance	饱和度	String	否	无	值范围：[0, 300]，空字符串（表示自动分析）

Container 类型 MsSharpen 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Enable	Request.MsSharpen	细节增强是否开启	String	否	false	true、false
SharpenLevel	Request.MsSharpen	增强等级	String	否	无	值范围：[0, 10]，空字符串（表示自动分析）

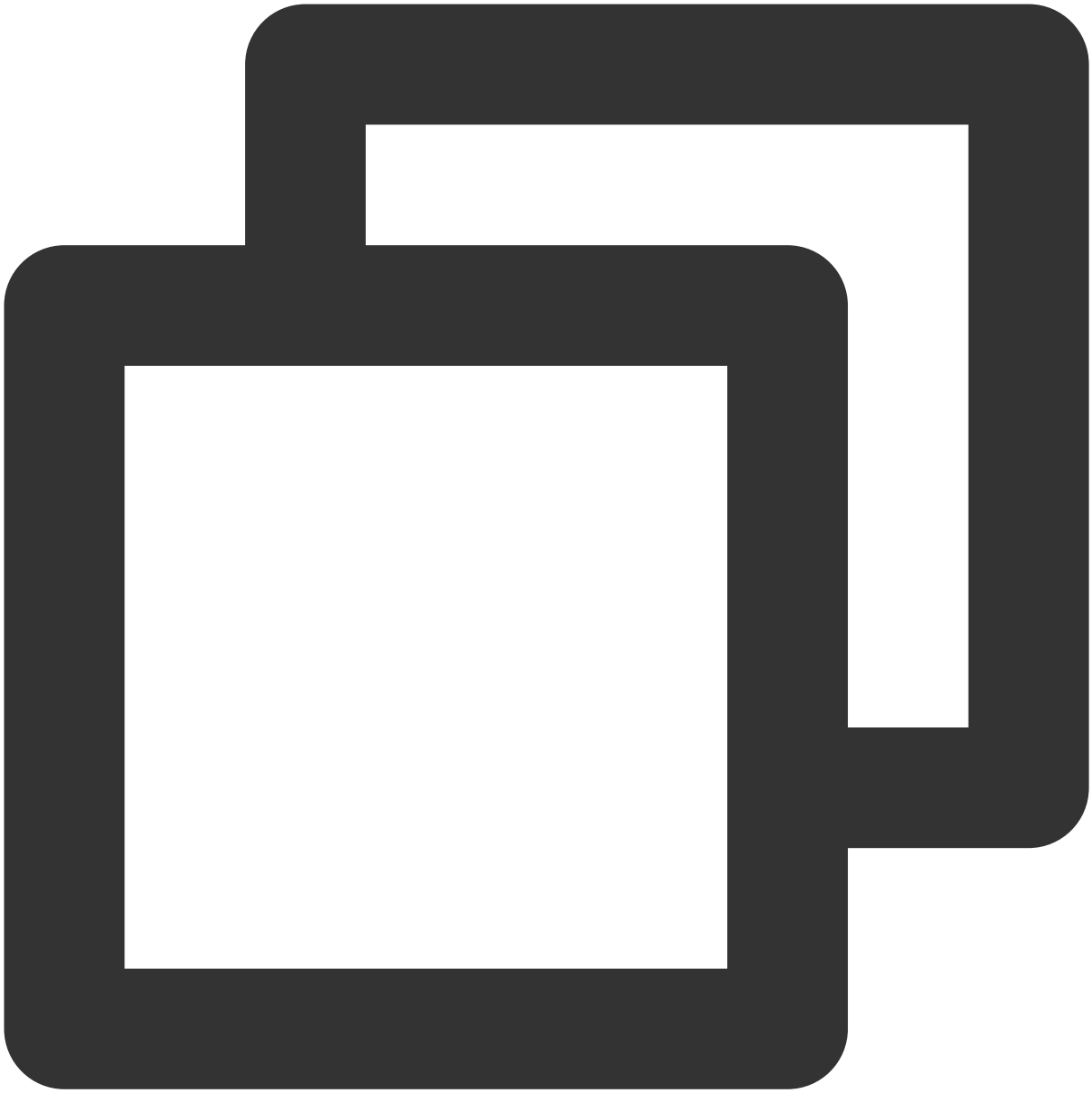
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>VideoProcess</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VideoProcess>
            <ColorEnhance>
                <Enable>true</Enable>
                <Contrast></Contrast>
```

```
                <Correction></Correction>
                <Saturation></Saturation>
            </ColorEnhance>
            <MsSharpen>
                <Enable>true</Enable>
                <SharpenLevel></SharpenLevel>
            </MsSharpen>
        <VideoProcess>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，VideoProcess	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
VideoProcess	Response.Template	详细的模板参数	Container

Container节点 VideoProcess 的内容：

节点名称（关键字）	父节点	描述
ColorEnhance	Response.Template.VideoProcess	同请求体中的 Request.ColorEnhance

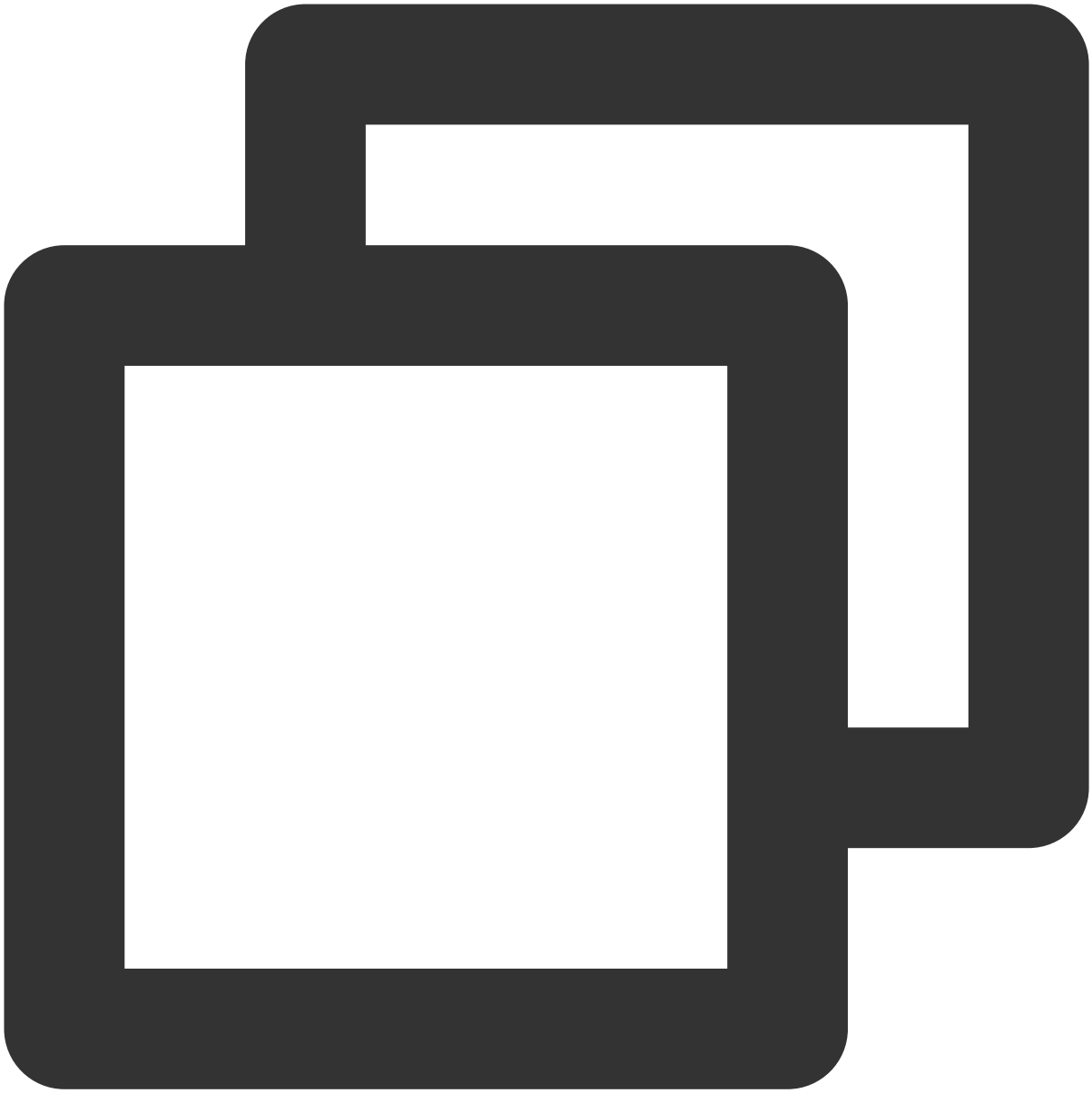
MsSharpen	Response.Template.VideoProcess	同请求体中的 Request.MsSharpen

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

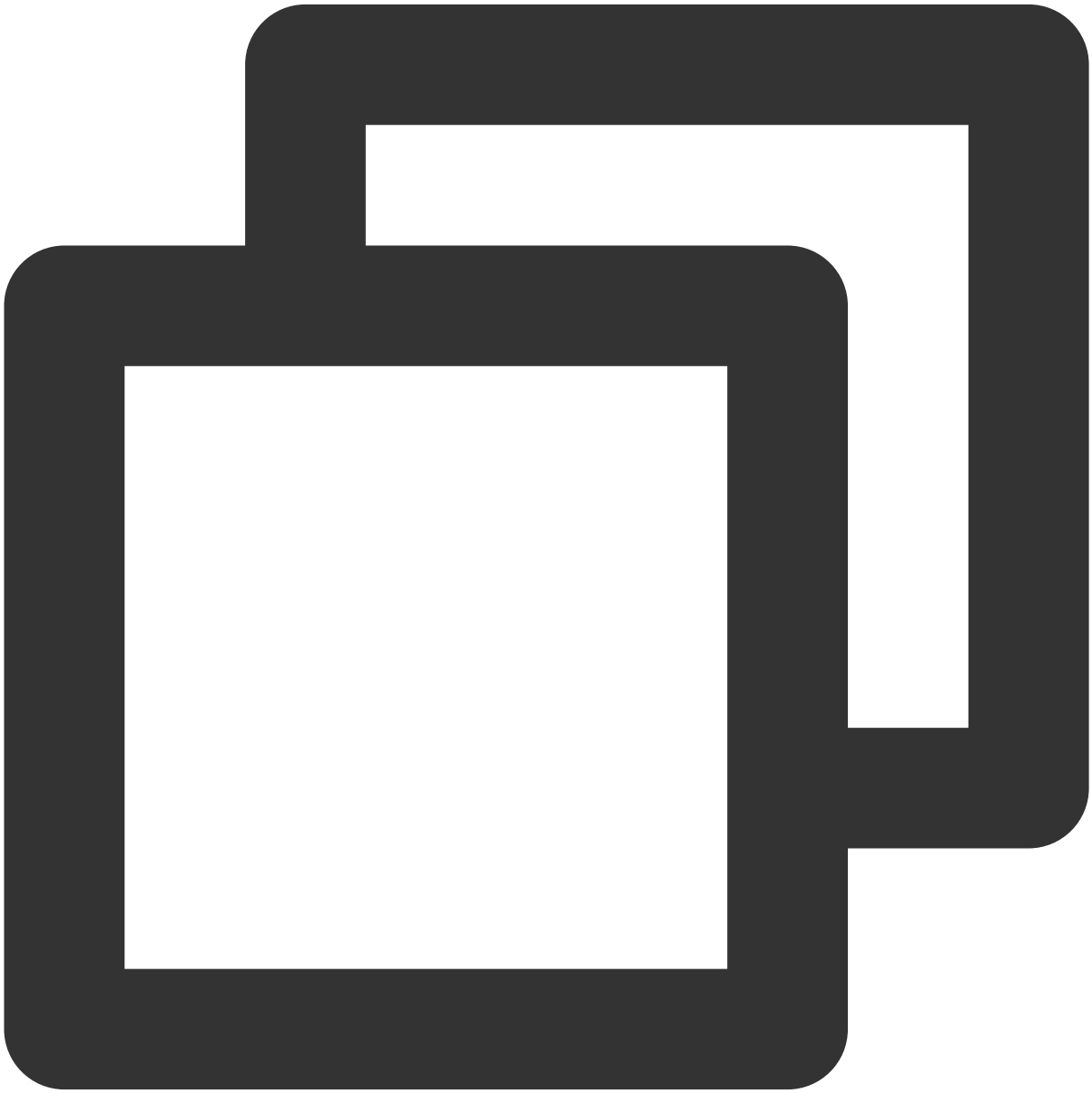
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoProcess</Tag>
    <Name>TemplateName</Name>
    <ColorEnhance>
        <Enable>true</Enable>
```

```
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
    </ColorEnhance>
    <MsSharpen>
        <Enable>true</Enable>
        <SharpenLevel></SharpenLevel>
    </MsSharpen>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>VideoProcess</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <ColorEnhance>
            <Enable>true</Enable>
            <Contrast></Contrast>
            <Correction></Correction>
            <Saturation></Saturation>
        </ColorEnhance>
        <MsSharpen>
            <Enable>true</Enable>
            <SharpenLevel></SharpenLevel>
        </MsSharpen>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

创建人声分离模板

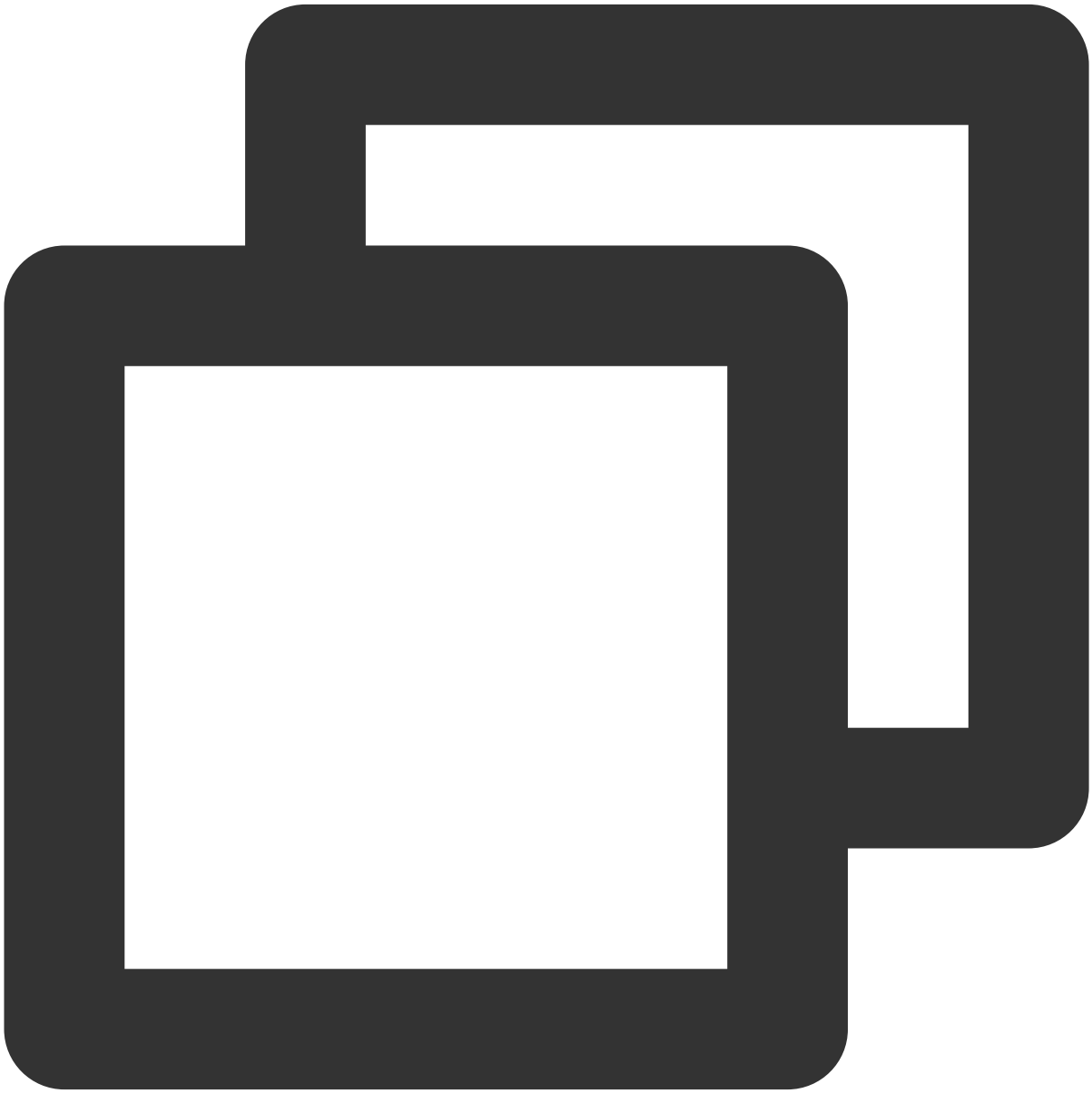
最近更新时间：2024-01-06 10:11:58

功能描述

创建人声分离模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

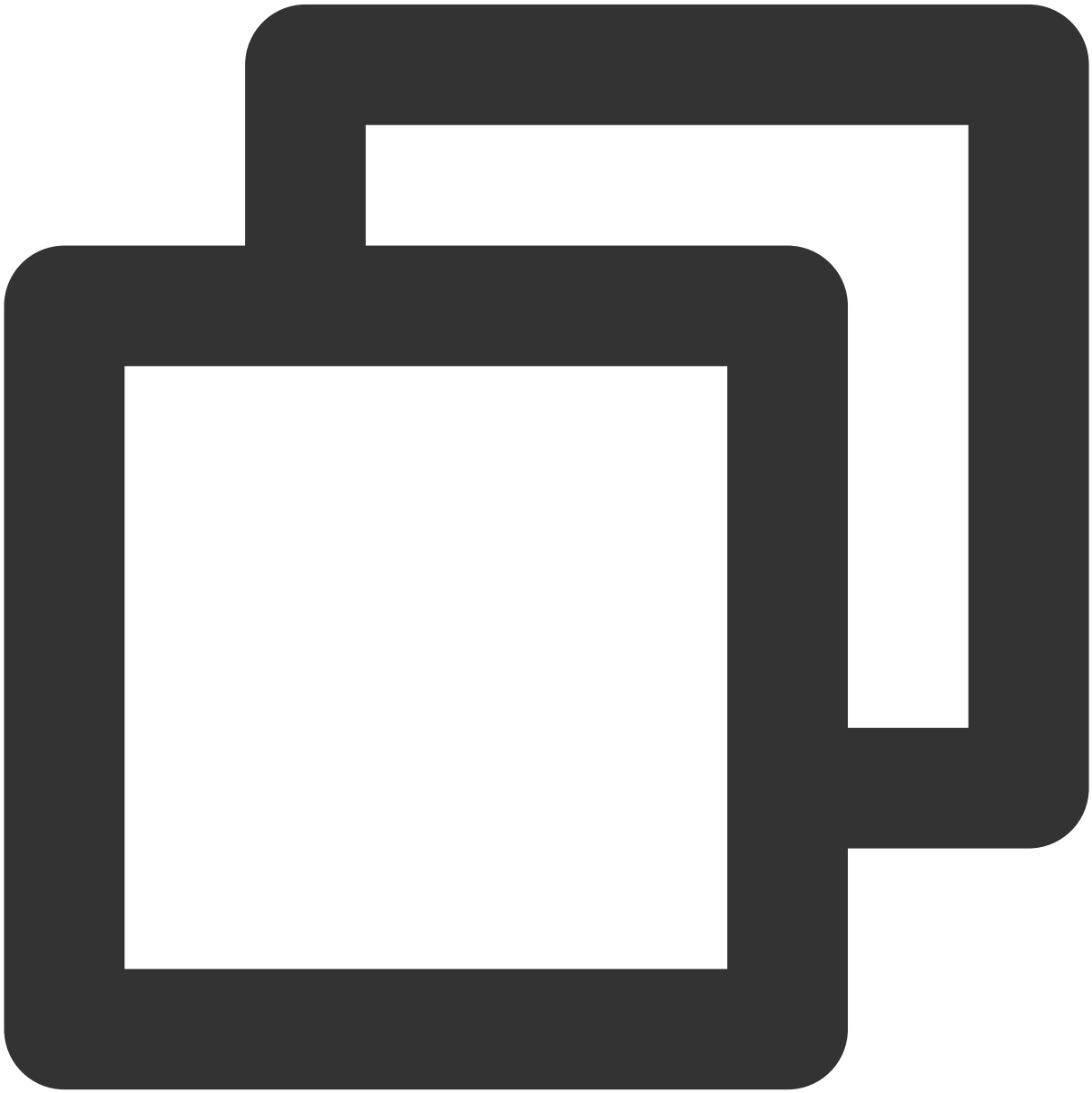
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>VoiceSeparate</Tag>
    <Name>TemplateName</Name>
    <AudioMode>IsAudio</AudioMode>
    <AudioConfig>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </AudioConfig>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型: VoiceSeparate	String	是	无
Name	Request	模板名称 仅支持中文、英文、数字、_、-和*	String	是	无
AudioMode	Request	输出音频	String	是	IsAudio：输出人声 IsBackground：输出背景声 AudioAndBackground：输出人声和背景声
AudioConfig	Request	音频配置	Container	是	无

Container 类型 AudioConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否	默认值	限制

				必选		
Codec	Request.Audio	编解码格式	String	否	aac	取值 aac、mp3、flac、amr
Samplerate	Request.Audio	采样率	String	否	44100	1. 单位：Hz 2. 可选 8000、11025、22050、32000、44100、48000、96000 3. 当 Codec 设置为 aac/flac 时，不支持8000 4. 当 Codec 设置为 mp3 时，不支持8000和96000 5. 当 Codec 设置为 amr 时，只支持8000
Bitrate	Request.Audio	原始音频码率	String	否	无	1. 单位：Kbps 2. 值范围：[8，1000]
Channels	Request.Audio	声道数	String	否	无	1. 当 Codec 设置为 aac/flac，支持1、2、4、5、6、8 2. 当 Codec 设置为 mp3，支持1、2 3. 当 Codec 设置为 amr，只支持1

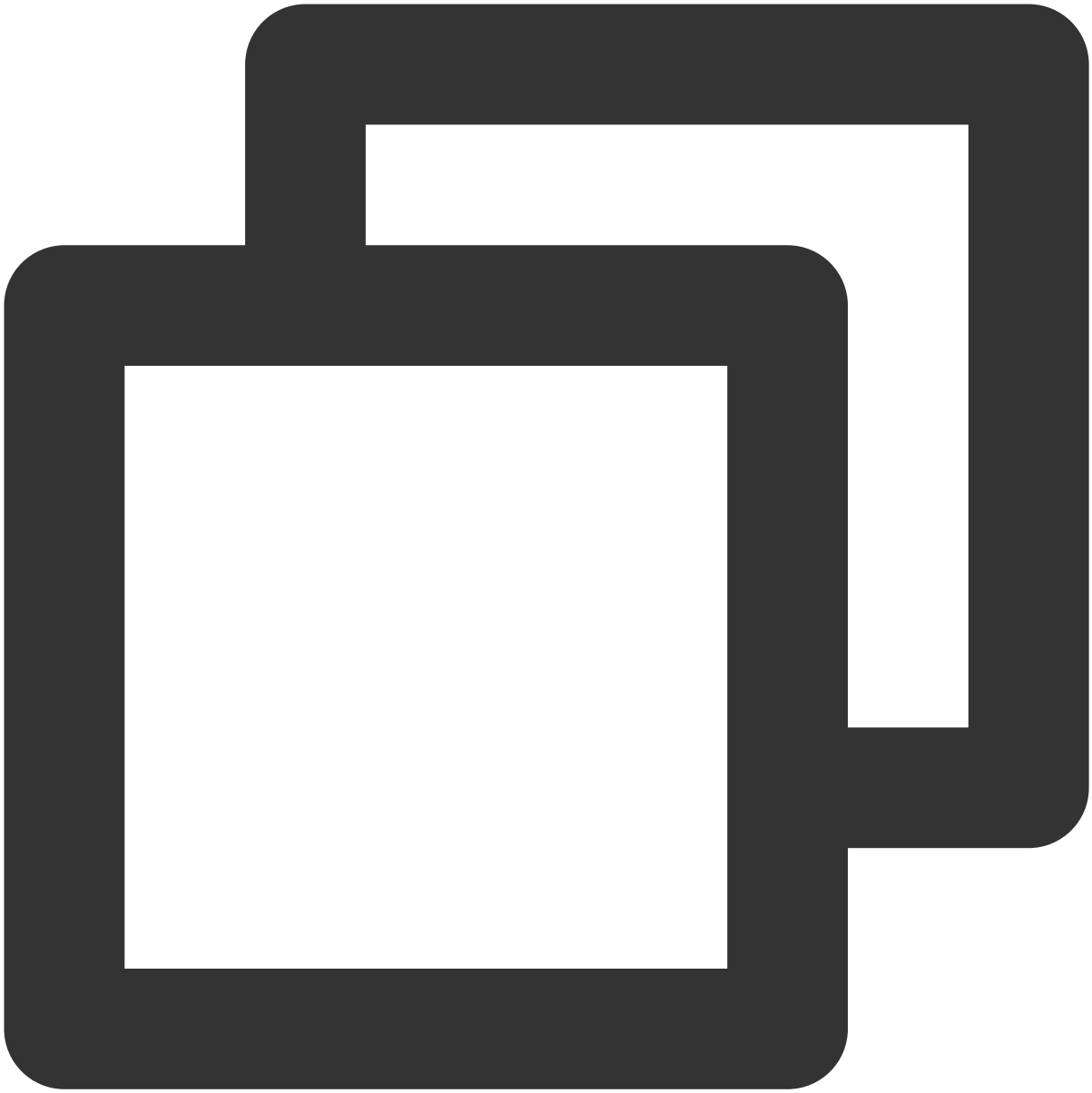
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>VoiceSeparate</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VoiceSeparate>
            <AudioMode>IsAudio</AudioMode>
            <AudioConfig>
                <Codec>mp3</Codec>
```

```
                <Samplerate>44100</Samplerate>
                <Bitrate>12</Bitrate>
                <Channels>2</Channels>
            </AudioConfig>
        </VoiceSeparate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，VoiceSeparate	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
VoiceSeparate	Response.Template	详细的模板参数	Container

Container 节点 VoiceSeparate 的内容：

节点名称（关键字）	父节点	描述	类型
AudioMode	Response.Template.VoiceSeparate	同请求体中的 Request.AudioMode	String
AudioConfig	Response.Template.VoiceSeparate	同请求体中的 Request.AudioConfig	Container

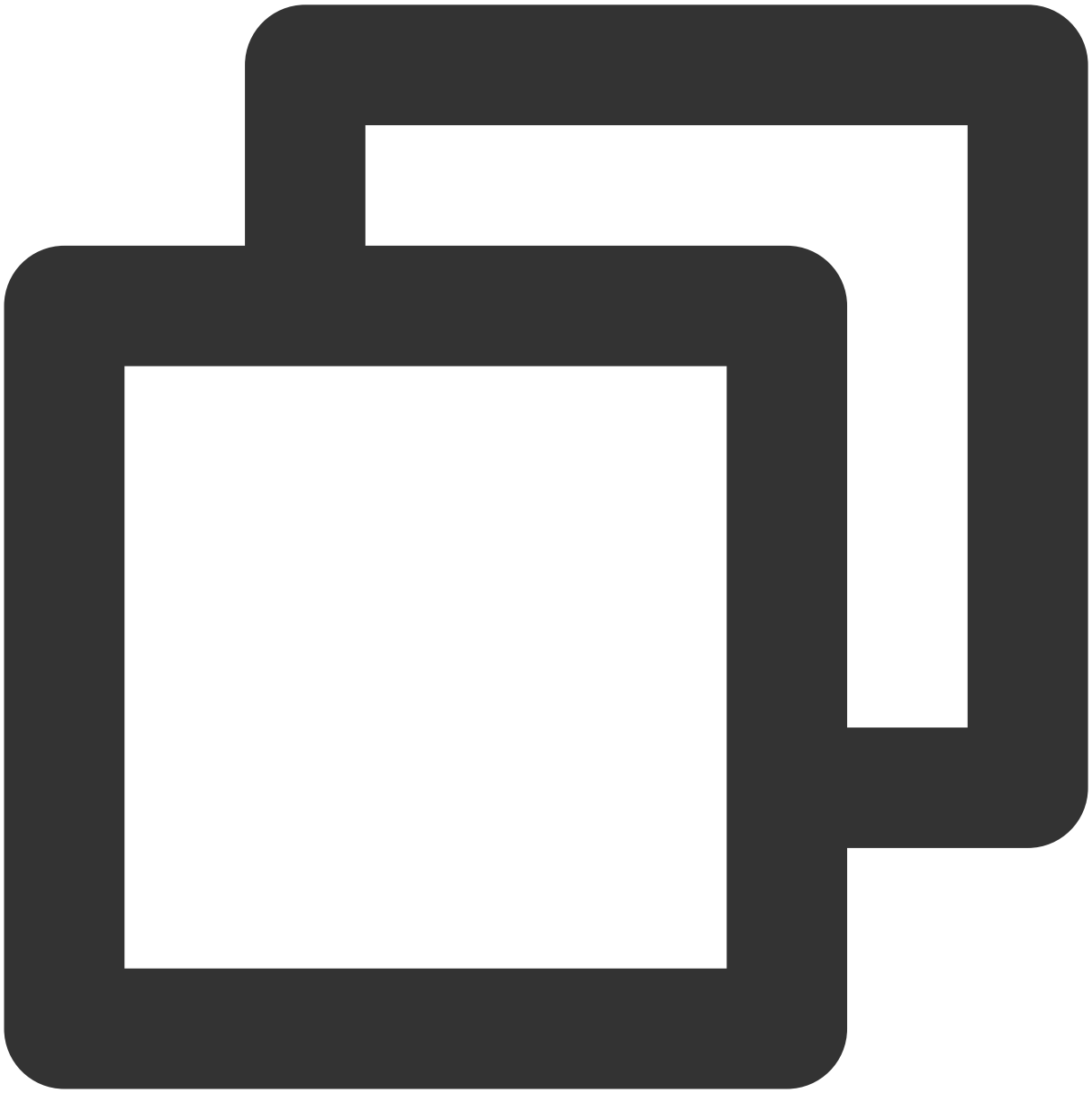
版权所有：腾讯云计算（北京）有限责任公司

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

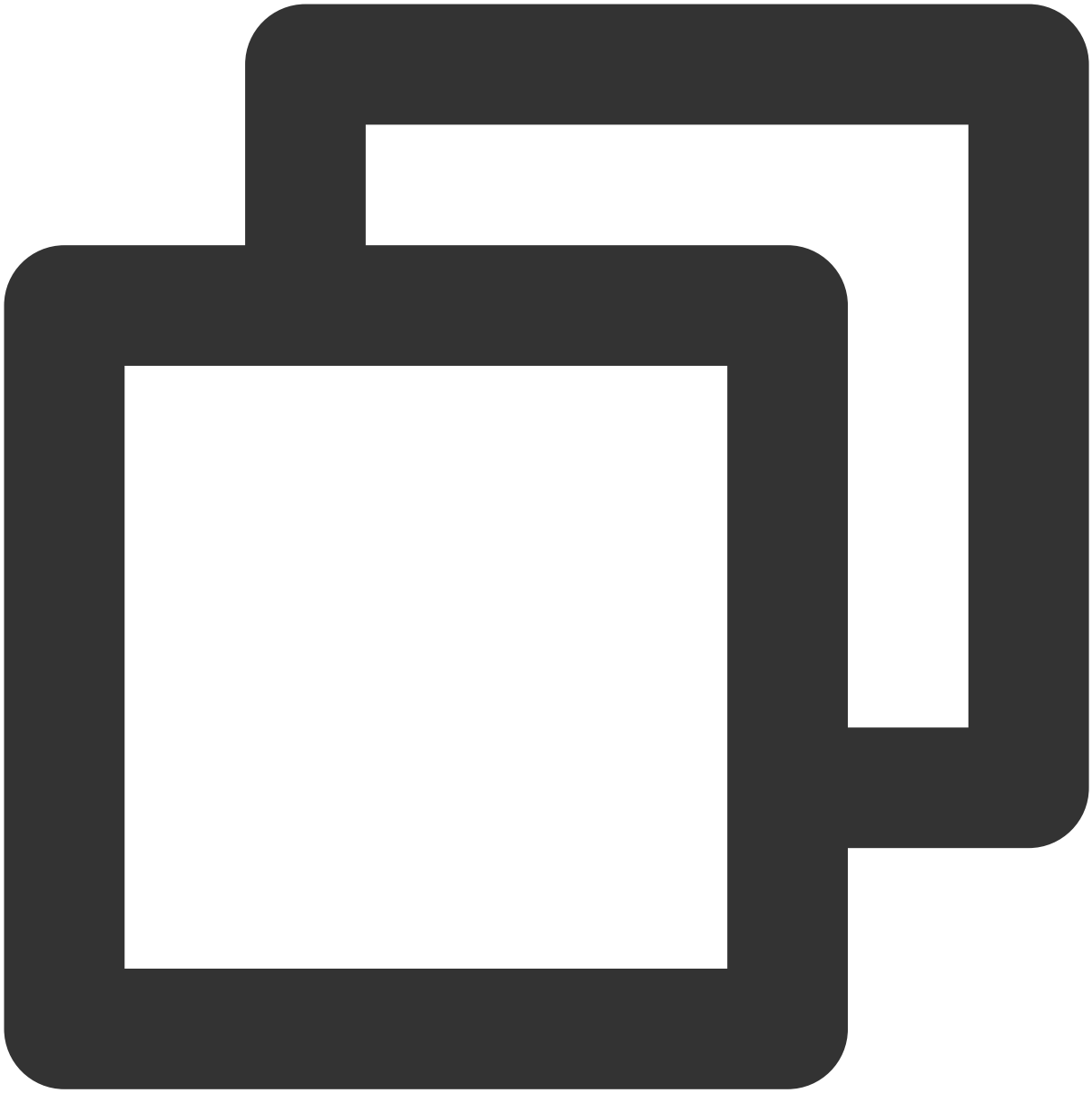
请求



```
POST /template HTTP/1.1
```

```
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VoiceSeparate</Tag>
    <Name>TemplateName</Name>
    <AudioMode>IsAudio</AudioMode>
    <AudioConfig>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </AudioConfig>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>VoiceSeparate</Tag>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VoiceSeparate>
            <AudioMode>IsAudio</AudioMode>
            <AudioConfig>
                <Codec>mp3</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>12</Bitrate>
                <Channels>2</Channels>
            </AudioConfig>
        </VoiceSeparate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

创建水印模板

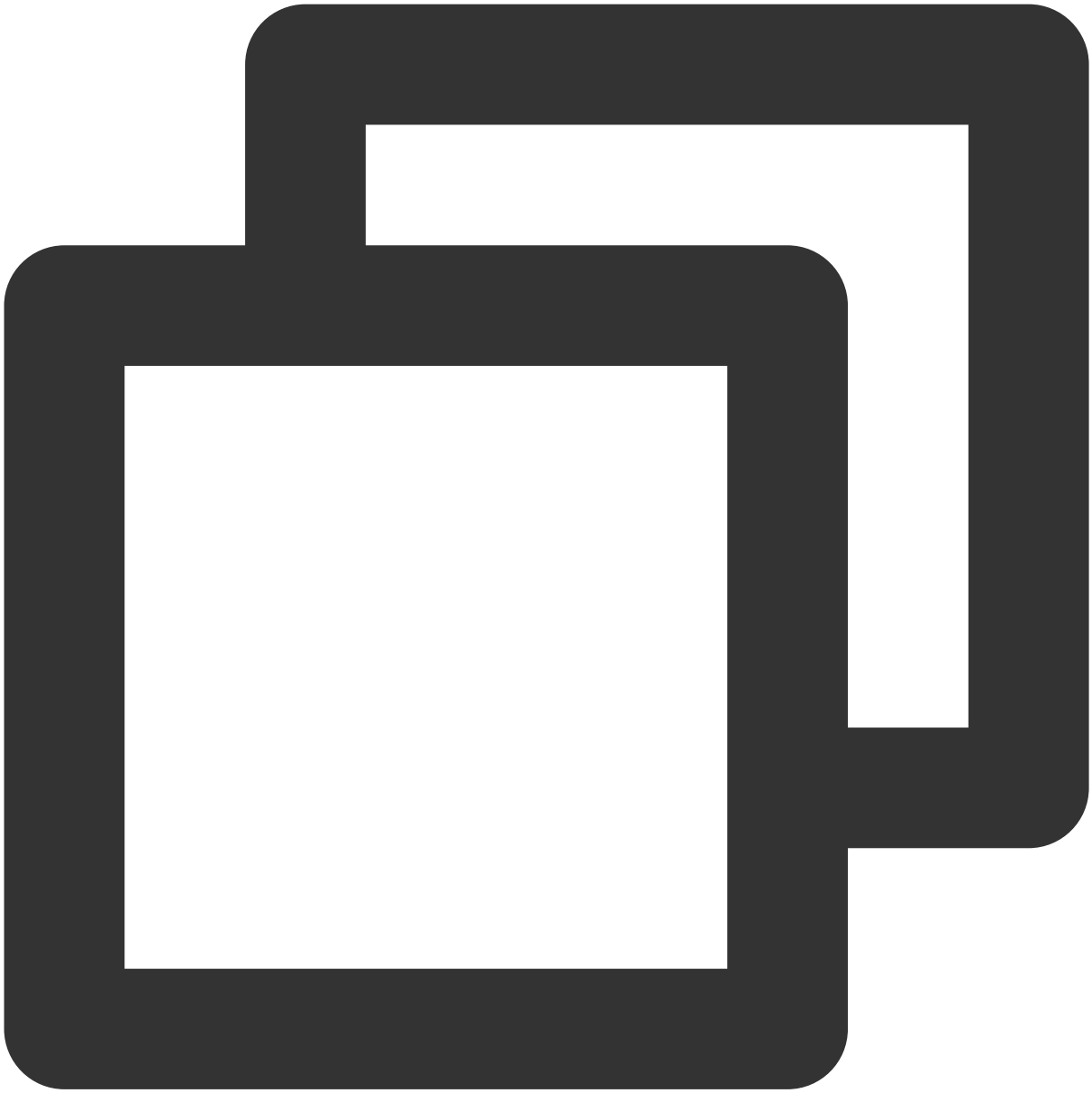
最近更新时间：2024-01-06 10:11:58

功能描述

创建水印模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

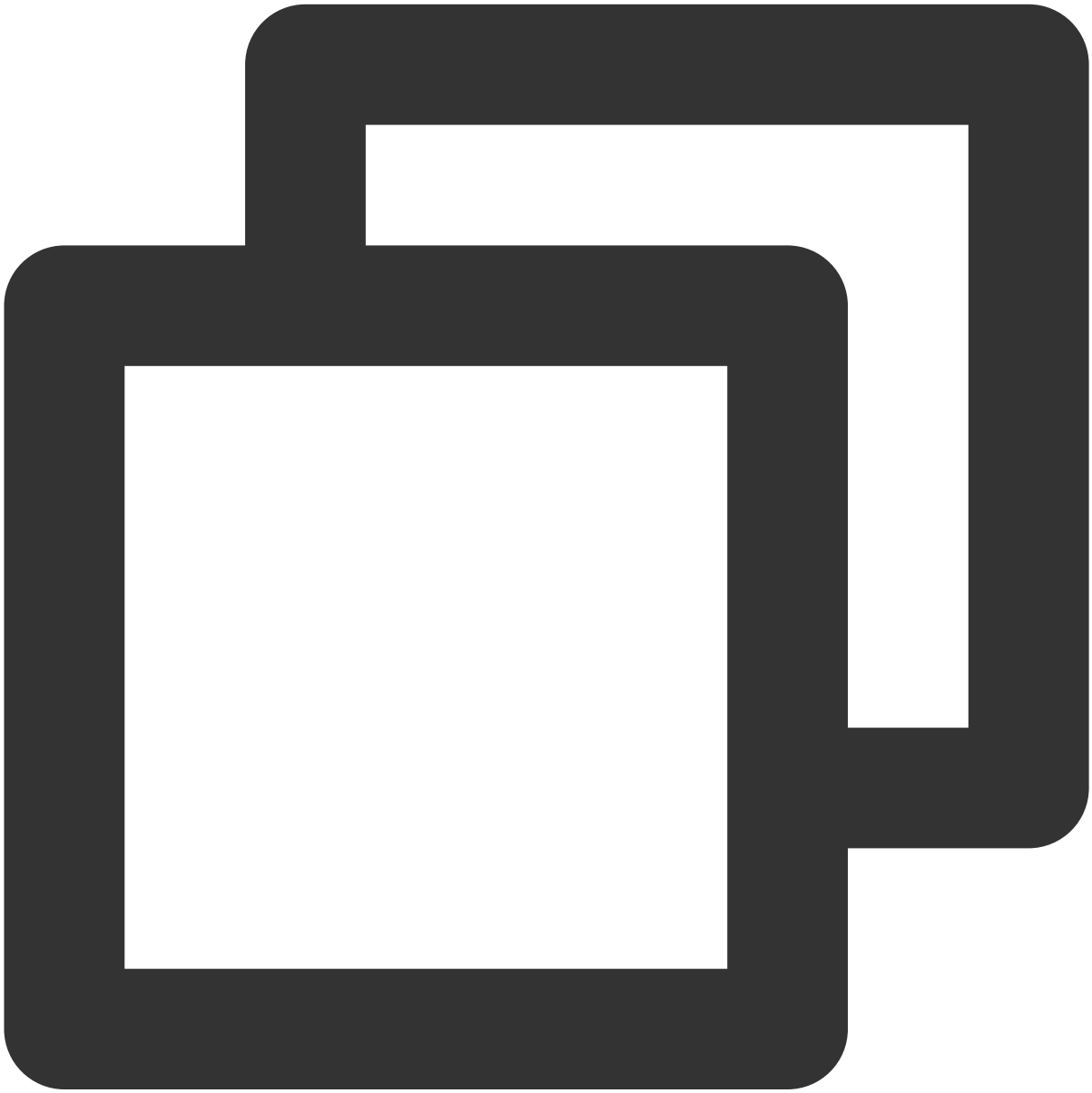
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Watermark</Tag>
    <Name>TemplateName</Name>
    <Watermark>
        <Type>Text</Type>
        <LocMode>Absolute</LocMode>
        <Dx>128</Dx>
        <Dy>128</Dy>
        <Pos>TopRight</Pos>
        <StartTime>0</StartTime>
        <EndTime>100.5</EndTime>
        <Text>
            <Text>水印内容</Text>
            <FontSize>30</FontSize>
            <FontType></FontType>
            <FontColor>0xRRGGBB</FontColor>
            <Transparency>30</Transparency>
        </Text>
    </Watermark>
</Request>

<Request>
    <Tag>Watermark</Tag>
    <Name>TemplateName</Name>
    <Watermark>
        <Type>Image</Type>
        <LocMode>Absolute</LocMode>
        <Dx>128</Dx>
        <Dy>128</Dy>
        <Pos>TopRight</Pos>
        <StartTime>0</StartTime>
        <EndTime>100.5</EndTime>
        <Image>
            <Url>http://test-1234567890.ci.ap-beijing.myqcloud.com/shuiyin_2.png</U
            <Mode>Proportion</Mode>
            <Width>10</Width>
            <Height></Height>
            <Transparency>100</Transparency>
        </Image>
    </Watermark>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Request	无	保存请求的容器	Container

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Tag	Request	模板类型: Watermark	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*	String	是
Watermark	Request	水印信息	Container	是

Container 类型 Watermark 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
Type	Request.Watermark	水印类型	String	是	无	Text：文字水印、Image：图片水印
Pos	Request.Watermark	基准位置	String	是	无	TopRight、TopLeft、BottomRight、BottomLeft、Left、Right、Top、Bottom、Center
LocMode	Request.Watermark	偏移方式	String	是	无	Relativity：按比例，Absolute：固定位置
Dx	Request.Watermark	水平偏移	String	是	无	1. 在图片水印中，如果 Background 为 true，当 locMode 为 Relativity 时，为%，值范围：[-300 0]；当 locMode 为 Absolute 时，为 px，值范围：[-4096 0]。 2. 在图片水印中，如果 Background 为 false，当 locMode 为 Relativity 时，为%，值范围：[0 100]；当 locMode 为 Absolute

						时，为 px，值范围：[0 4096]。 3. 在文字水印中，当 locMode 为 Relativity 时，为%，值范围：[0 100]；当 locMode 为 Absolute 时，为 px，值范围：[0 4096]。 4. 当Pos为Top、Bottom和Center时，该参数无效。
Dy	Request.Watermark	垂直偏移	String	是	无	1. 在图片水印中，如果 Background 为 true，当 locMode 为 Relativity 时，为%，值范围：[-300 0]；当 locMode 为 Absolute 时，为 px，值范围：[-4096 0]。 2. 在图片水印中，如果 Background 为 false，当 locMode 为 Relativity 时，为%，值范围：[0 100]；当 locMode 为 Absolute 时，为 px，值范围：[0 4096]。 3. 在文字水印中，当 locMode 为 Relativity 时，为%，值范围：[0 100]；当 locMode 为 Absolute 时，为 px，值范围：[0 4096]。 4. 当Pos为Left、Right和Center时，该参数无效。
StartTime	Request.Watermark	水印开始时间	String	否	0	1. [0 视频时长] 2. 单位为秒 3. 支持 float 格式，执行精度精确到毫秒
EndTime	Request.Watermark	水印结束时间	String	否	视频结	1. [0 视频时长] 2. 单位为秒

					束时间	3. 支持 float 格式，执行精度精确到毫秒
SlideConfig	Request.Watermark	水印滑动配置，配置该参数后水印位移设置不生效，极速高清/H265转码暂时不支持该参数	Container	否	无	无
Image	Request.Watermark	图片水印节点	Container	否	无	无
Text	Request.Watermark	文本水印节点	Container	否	无	无

Container 类型 SlideConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
SlideMode	Request.Watermark.SlideConfig	滑动模式	String	是	无	Default: 默认不开启、ScrollFromLeft: 从左到右滚动，若设置了ScrollFromLeft模式，则Watermark.Pos参数不生效
XSlideSpeed	Request.Watermark.SlideConfig	横向滑动速度	String	是	无	取值范围：[0,10]内的整数，默认为0
YSlideSpeed	Request.Watermark.SlideConfig	纵向滑动速度	String	是	无	取值范围：[0,10]内的整数，默认为0

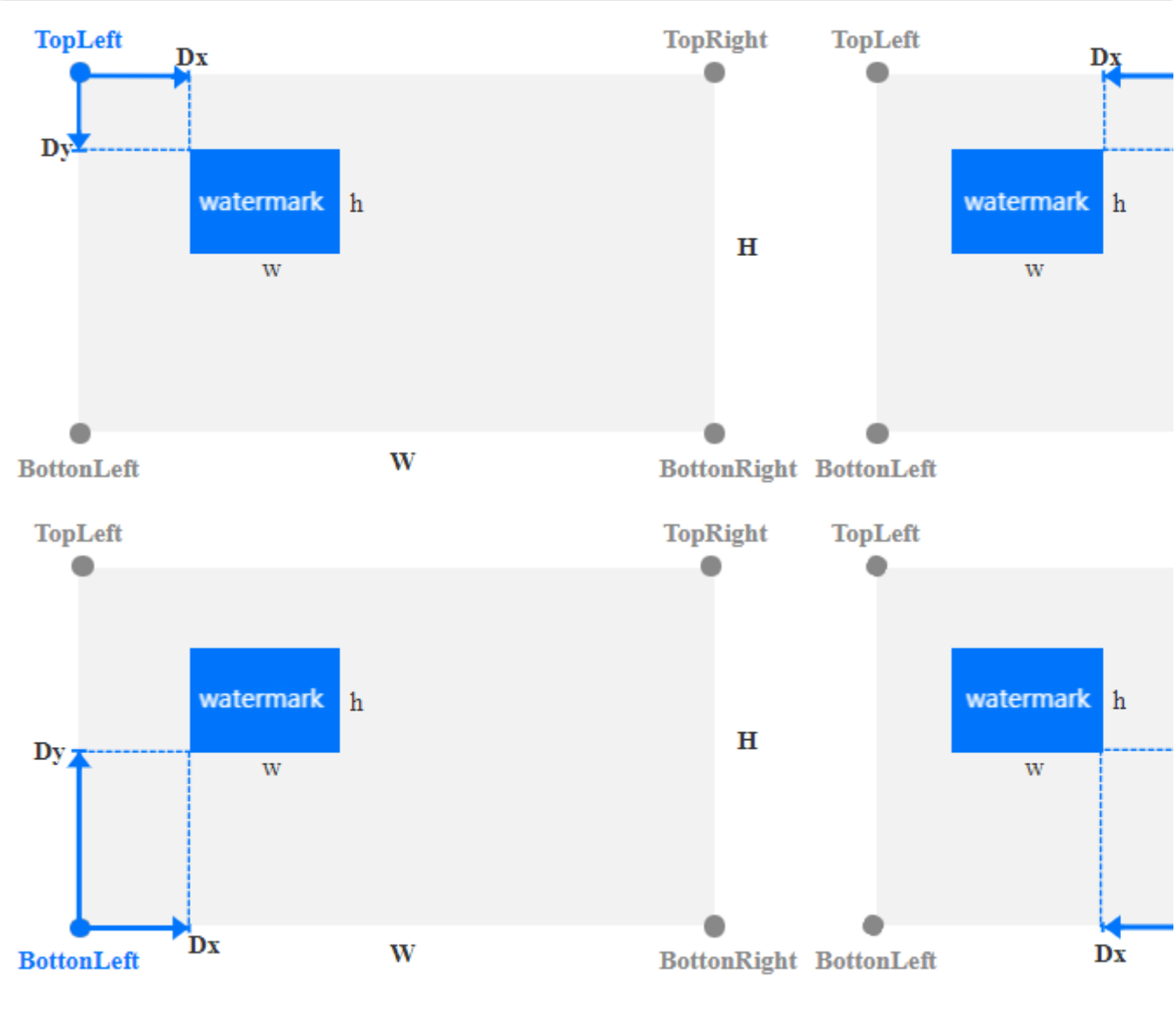
版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Image 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
Url	Request.Watermark.Image	水印图地址(需要 Urlencode 后传入)	String	是	无	同 bucket 的水印图片地址
Mode	Request.Watermark.Image	尺寸模式	String	是	无	1. Original：原有尺寸 2. Proportion：按比例 3. Fixed：固定大小
Width	Request.Watermark.Image	宽	String	否	无	1. 当 Mode 为 Original 时，不支持设置水印图宽度 2. 当 Mode 为 Proportion，单位为%，背景图值范围：[100 300]；前景图值范围：[1 100]，相对于视频宽，最大不超过 4096px 3. 当 Mode 为 Fixed，单位为 px，值范围：[8, 4096] 4. 若只设置 Width 时，按照水印图比例计算 Height
Height	Request.Watermark.Image	高	String	否	无	1. 当 Mode 为 Original 时，不支持设置水印图高度 2. 当 Mode 为 Proportion，单位为%，背景图值范围：[100 300]；前景图值范围：[1 100]，相对于视频

						高，最大不超过4096px 3. 当 Mode 为 Fixed，单位为 px，值范围：[8, 4096] 4. 若只设置 Height 时，按照水印图比例计算 Width
Transparency	Request.Watermark.Image	透明度	String	是	无	值范围：[1 100]，单位%
Background	Request.Watermark.Image	是否背景图	String	否	false	true、false

水印位置说明：



Container 类型 Text 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选	默认值	限制
FontSize	Request.Watermark.Text	字体大小	String	是	无	值范围：[5 100]，单位 px
FontType	Request.Watermark.Text	字体类型	String	是	无	参考下表
FontColor	Request.Watermark.Text	字体颜色	String	是	无	格式：0xRRGGBB
Transparency	Request.Watermark.Text	透明度	String	是	无	值范围：[1 100]，单位%

版权所有：腾讯云计算（北京）有限责任公司

Text	Request.Watermark.Text	水印内容	String	是	无	长度不超过64个字符，仅支持中文、英文、数字、_、-和*

Text 的 FontType 具体数据描述如下：

字体名称	支持的语言	描述
simfang.ttf	中/英	仿宋
simhei.ttf	中/英	黑体
simkai.ttf	中/英	楷体
simsun.ttc	中/英	宋体
STHeiti-Light.ttc	中/英	华文黑体
STHeiti-Medium.ttc	中/英	华文黑体中
youyuan.TTF	中/英	幼圆
ariblk.ttf	英	无
arial.ttf	英	无
ahronbd.ttf	英	无
Helvetica.dfont	英	无
HelveticaNeue.dfont	英	无

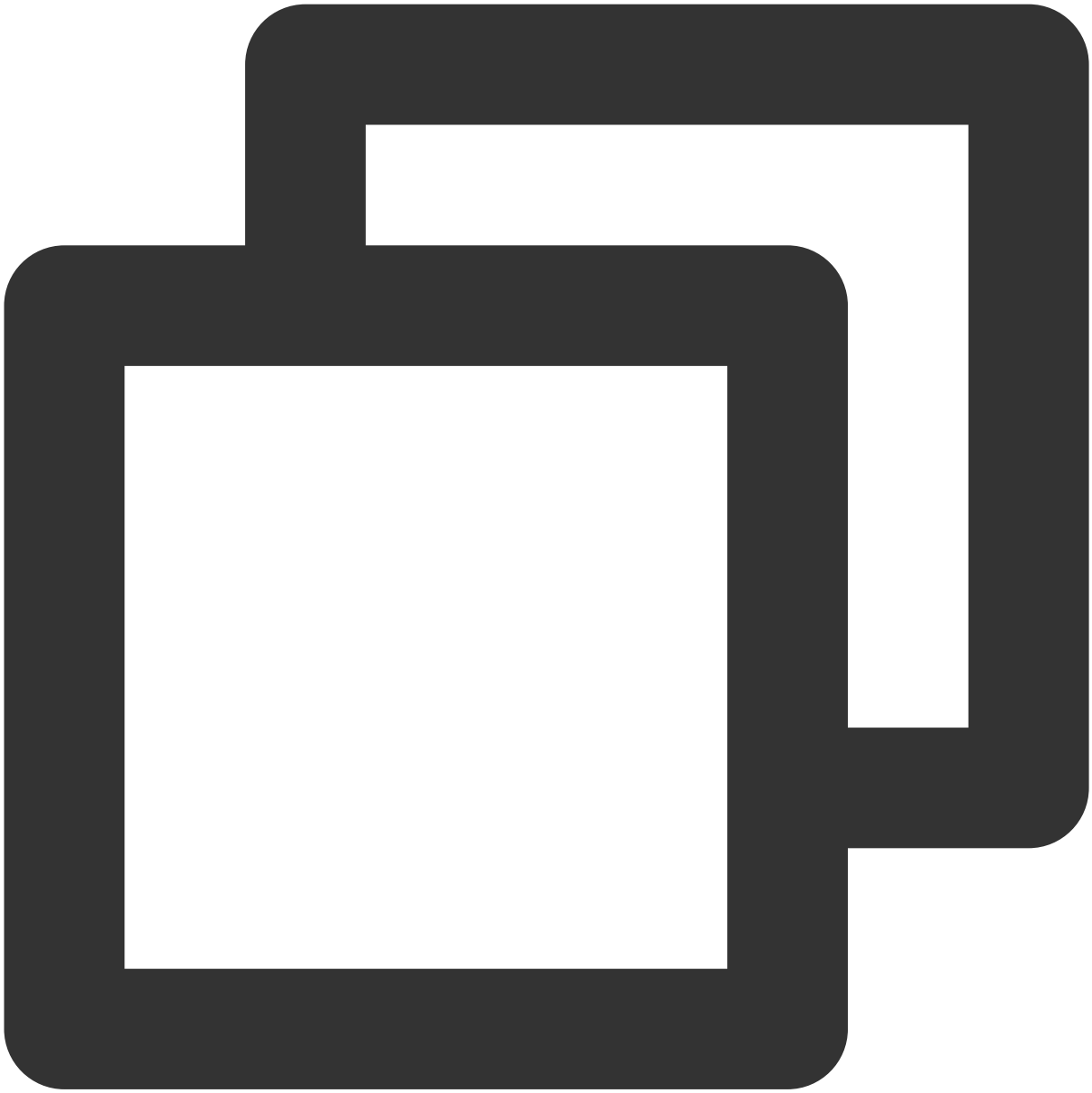
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Tag>Watermark</Tag>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Watermark>
            <Type>Text</Type>
            <LocMode>Absolute</LocMode>
            <Dx>128</Dx>
```

```
            <Dy>128</Dy>
            <Pos>TopRight</Pos>
            <StartTime>0</StartTime>
            <EndTime>100.5</EndTime>
            <Text>
                <Text>水印内容</Text>
                <FontSize>30</FontSize>
                <FontType></FontType>
                <FontColor>0xRRGGBB</FontColor>
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

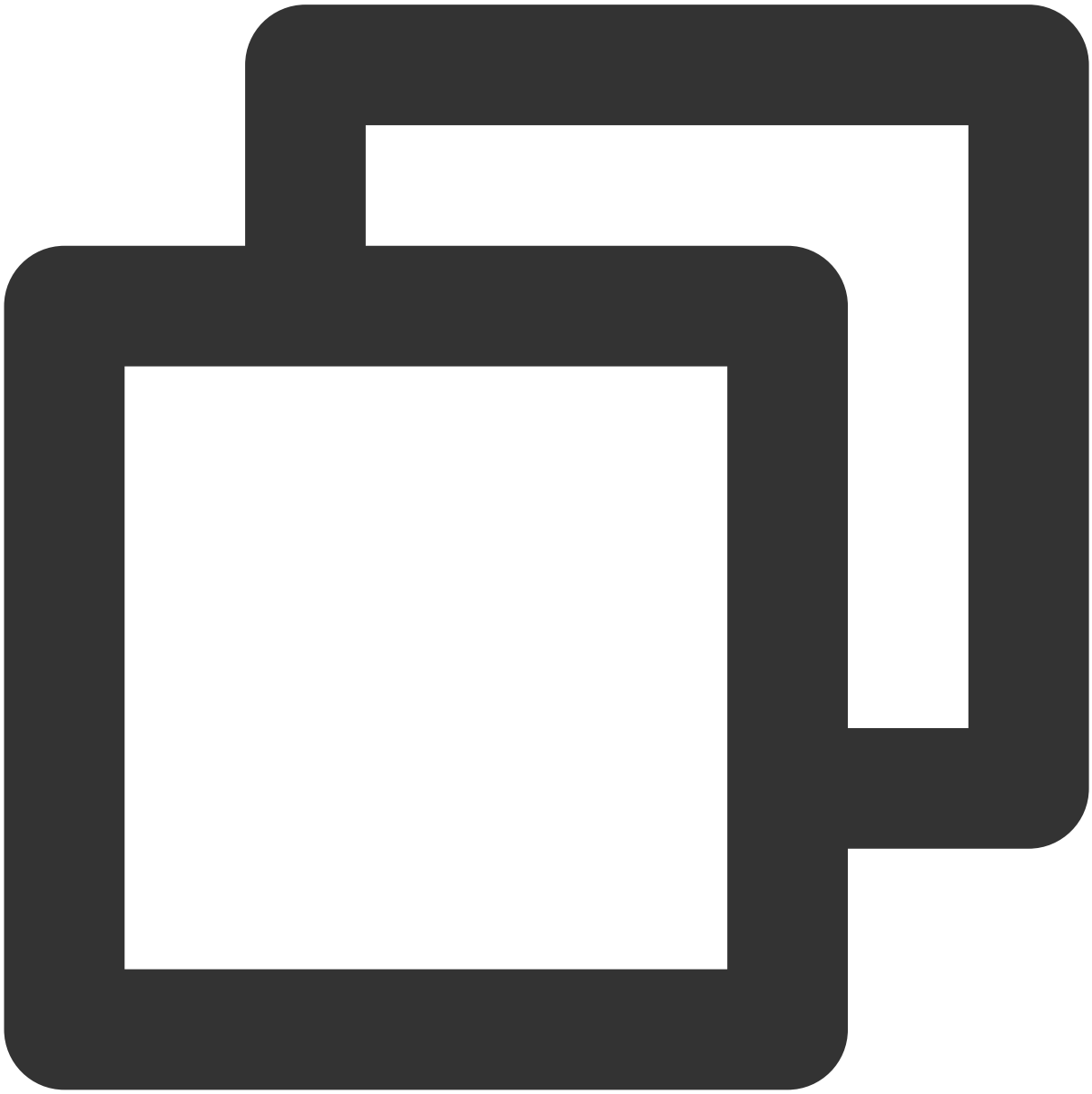
节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，VideoProcess	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
Watermark	Response.Template	同请求体中的 Request.Watermark	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

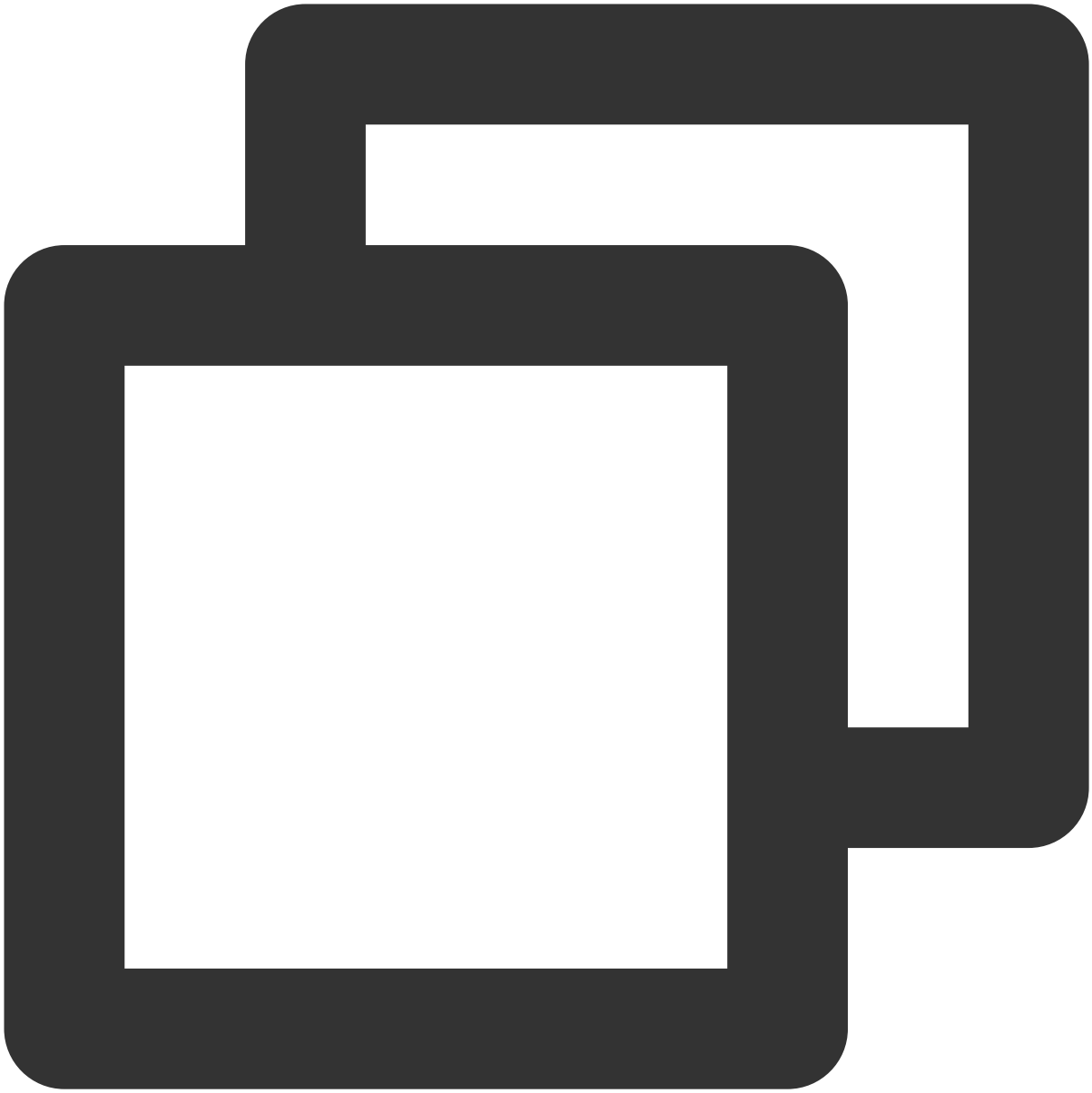
请求1（文本水印）



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0****&q-sign-tim
Host: test-1234567890.ci.ap-beijing.myqcloud.com
```

```
Content-Length: 1666
Content-Type: application/xml

<Request>
   <Tag>Watermark</Tag>
   <Name>TemplateName</Name>
   <Watermark>
      <Type>Text</Type>
      <LocMode>Absolute</LocMode>
      <Dx>128</Dx>
      <Dy>128</Dy>
      <Pos>TopRight</Pos>
      <StartTime>0</StartTime>
      <EndTime>100.5</EndTime>
      <Text>
        <Text>水印内容</Text>
        <FontSize>30</FontSize>
        <FontType></FontType>
        <FontColor>0xRRGGBB</FontColor>
        <Transparency>30</Transparency>
      </Text>
   </Watermark>
</Request>
```

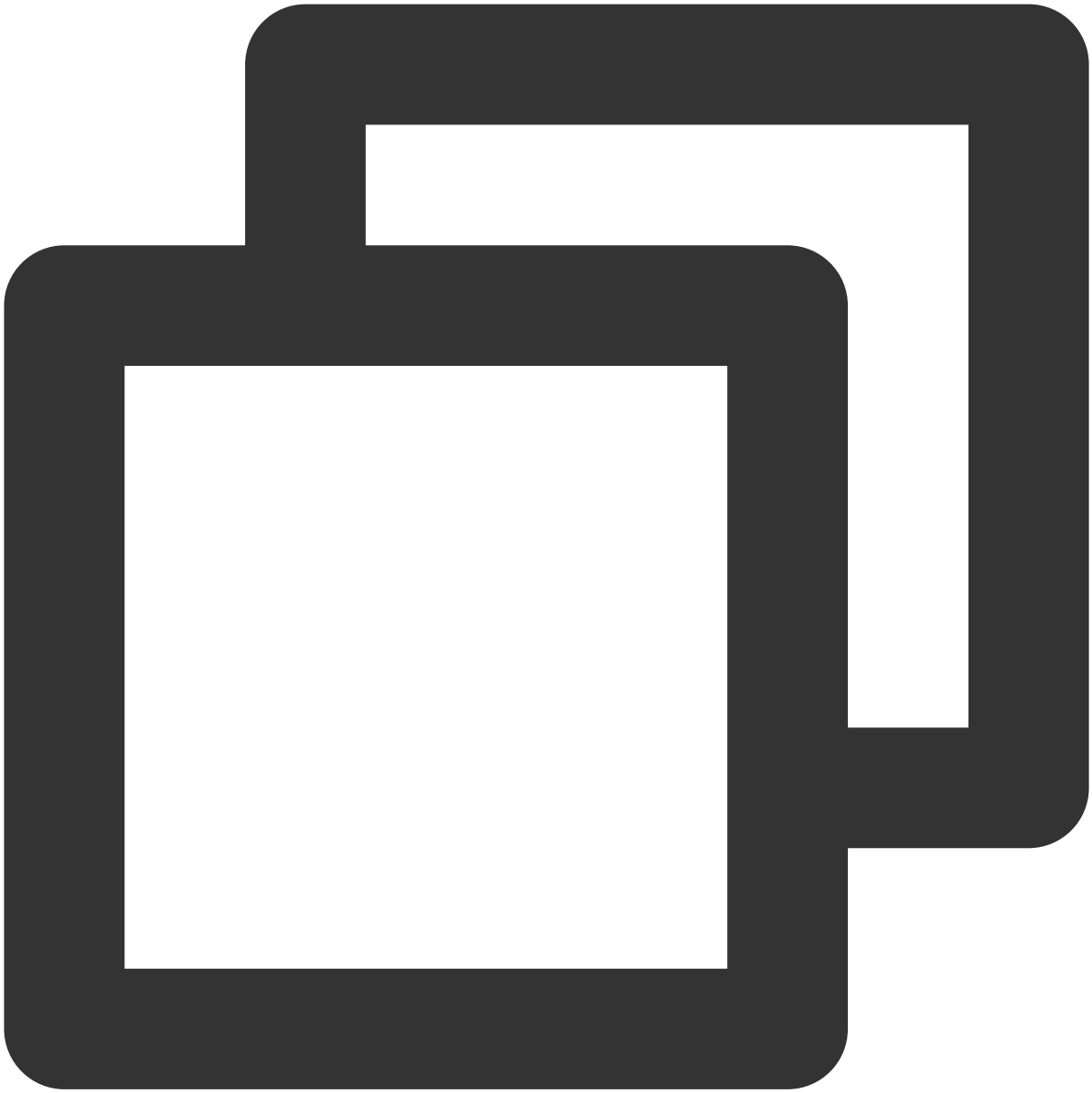
响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Tag>Watermark</Tag>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Watermark>
            <Type>Text</Type>
            <LocMode>Absolute</LocMode>
            <Dx>128</Dx>
            <Dy>128</Dy>
            <Pos>TopRight</Pos>
            <StartTime>0</StartTime>
            <EndTime>100.5</EndTime>
            <Text>
                <Text>水印内容</Text>
                <FontSize>30</FontSize>
                <FontType></FontType>
                <FontColor>0xRRGGBB</FontColor>
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

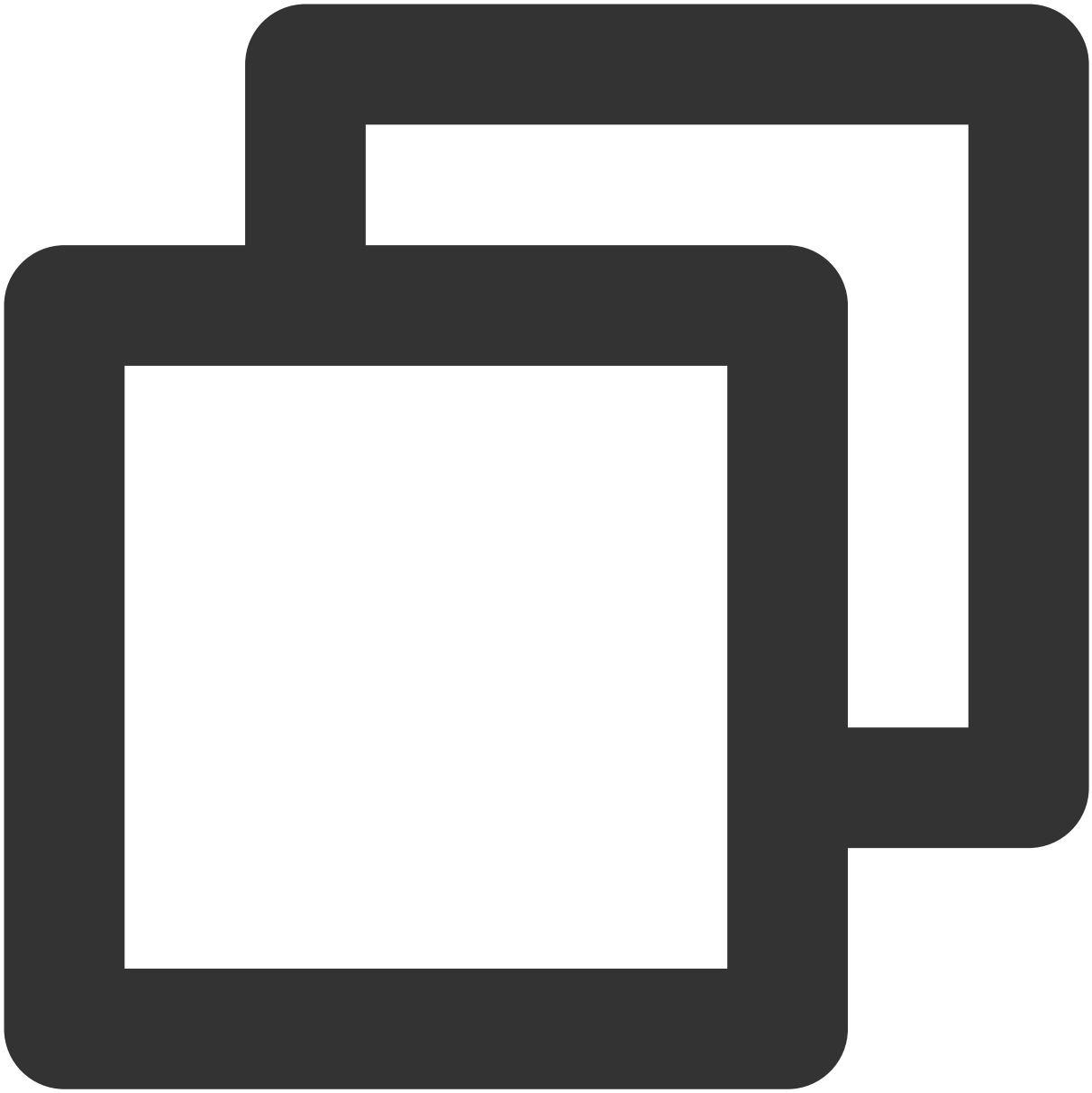
请求2（图片水印）



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0****&q-sign-tim
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
   <Tag>Watermark</Tag>
   <Name>TemplateName</Name>
   <Watermark>
      <Type>Image</Type>
```

```
        <LocMode>Absolute</LocMode>
        <Dx>128</Dx>
        <Dy>128</Dy>
        <Pos>TopRight</Pos>
        <StartTime>0</StartTime>
        <EndTime>100.5</EndTime>
        <Image>
          <Url>http://test-1234567890.ci.ap-beijing.myqcloud.com/shuiyin_2.png</Url>
          <Mode>Proportion</Mode>
          <Width>10</Width>
          <Height>10</Height>
          <Transparency>30</Transparency>
        </Image>
    </Watermark>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Tag>Watermark</Tag>
        <Name>TemplateName</Name>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <Watermark>
            <Type>Image</Type>
            <LocMode>Absolute</LocMode>
            <Dx>128</Dx>
            <Dy>128</Dy>
            <Pos>TopRight</Pos>
            <StartTime>0</StartTime>
            <EndTime>100.5</EndTime>
            <Image>
                <Url>http://test-1234567890.ci.ap-beijing.myqcloud.com/shuiyin_2.pn
                <Mode>Proportion</Mode>
                <Width>10</Width>
                <Height>10</Height>
                <Transparency>30</Transparency>
            </Image>
        </Watermark>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

创建智能封面模板

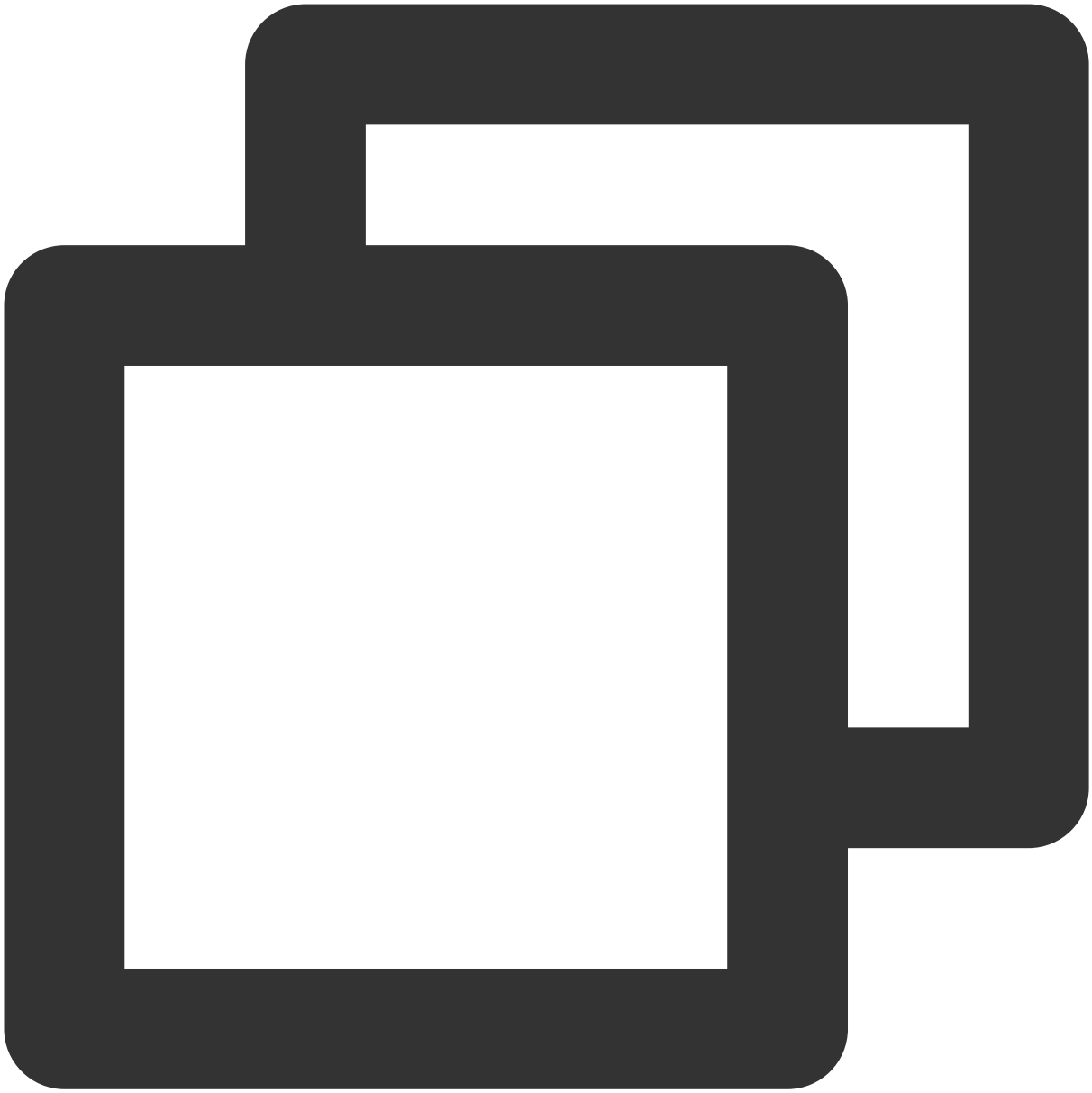
最近更新时间：2024-01-06 10:11:58

功能描述

创建智能封面模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

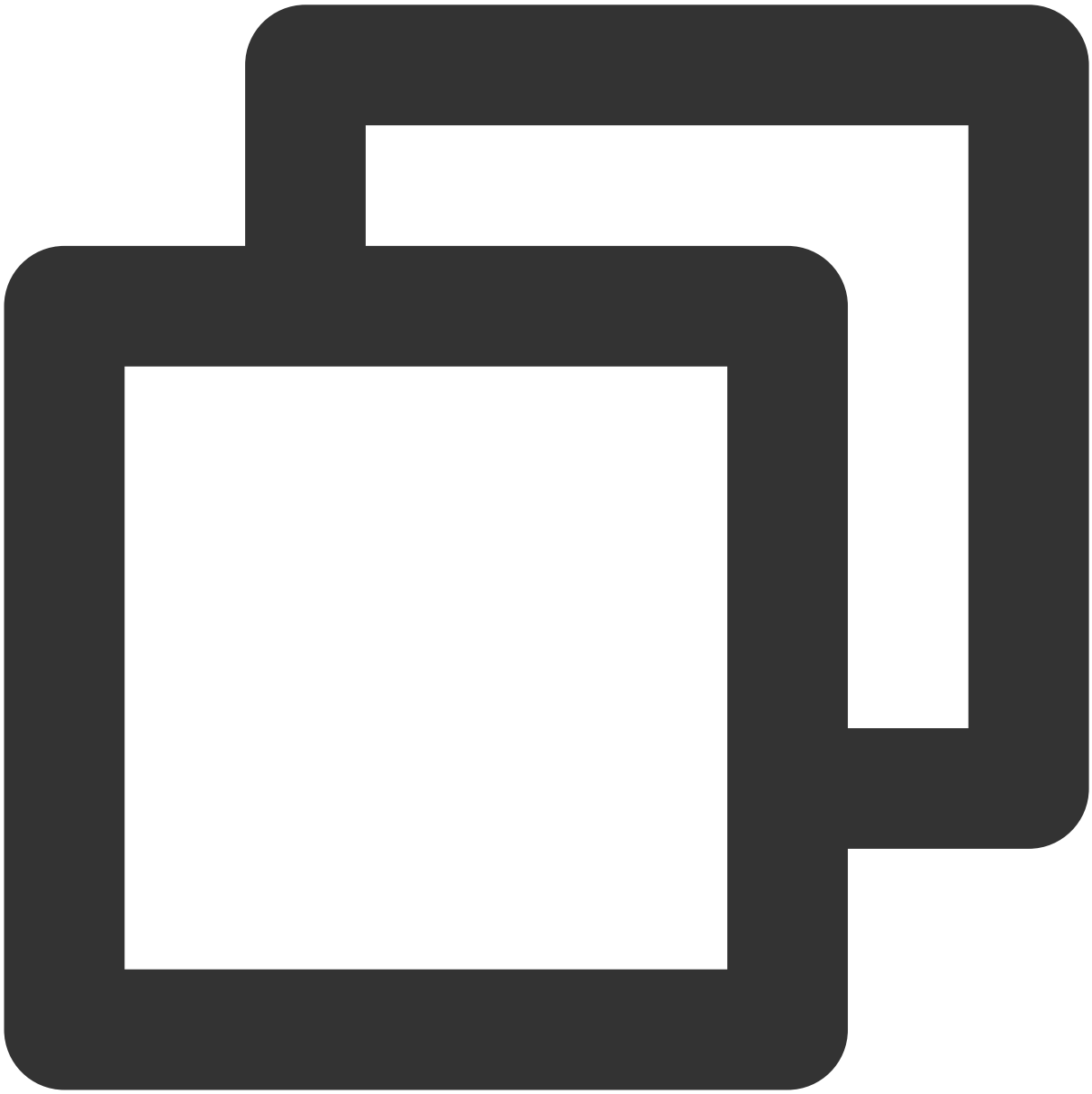
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>SmartCover</Tag>
    <Name>TemplateName</Name>
    <SmartCover>
        <Format>jpg</Format>
        <Width>1280</Width>
        <Height>960</Height>
        <Count>10</Count>
        <DeleteDuplicates>true</DeleteDuplicates>
    </SmartCover>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	模板类型：SmartCover	String	是
Name	Request	模板名称仅支持中文、英文、数字、_、-和*	String	是
SmartCover	Request	智能封面参数	Container	是

ontainer 类型 SmartCover 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Format	Request.SmartCover	图片格式	String	否	jpg	jpg、png 、webp
Width	Request.SmartCover	宽	String	否	视频原始宽度	值范围：[128, 4096] 单位：px 若只设置 Width 时，按照视频原始比例计算 Height
Height	Request.SmartCover	高	String	否	视频原始高度	值范围：[128, 4096] 单位：px

版权所有：腾讯云计算（北京）有限责任公司

						若只设置 Height 时，按照视频原始比例计算 Width
Count	Request.SmartCover	截图数量	String	否	3	[1,10]
DeleteDuplicates	Request.SmartCover	封面去重	String	否	false	true/false

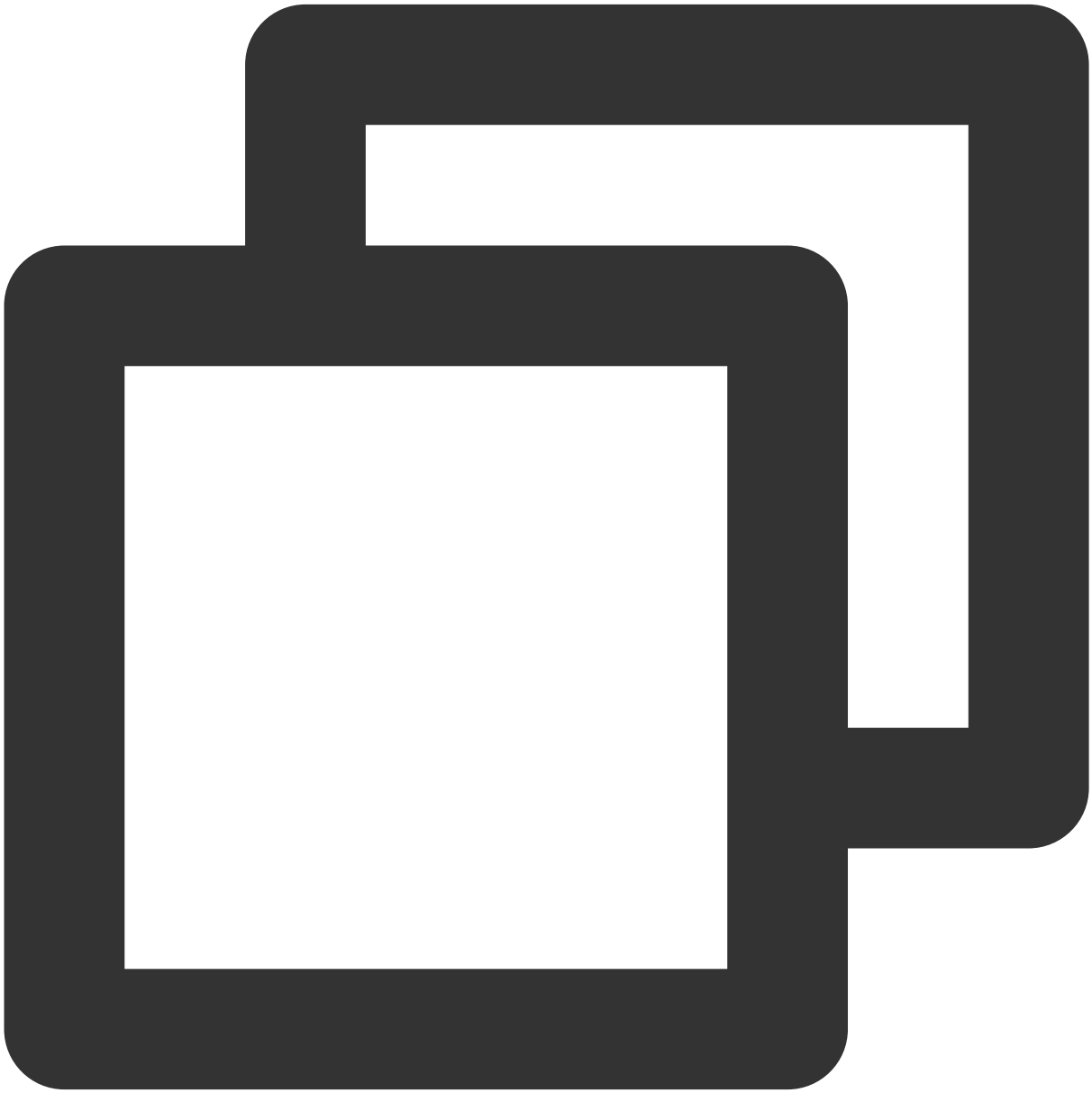
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>SmartCover</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f43110a08702470b8fa75894629501fa</TemplateId>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SmartCover>
            <Width>1280</Width>
            <Height>960</Height>
            <Format>jpg</Format>
```

```
            <Count>5</Count>
            <DeleteDuplicates>false</DeleteDuplicates>
        </SmartCover>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

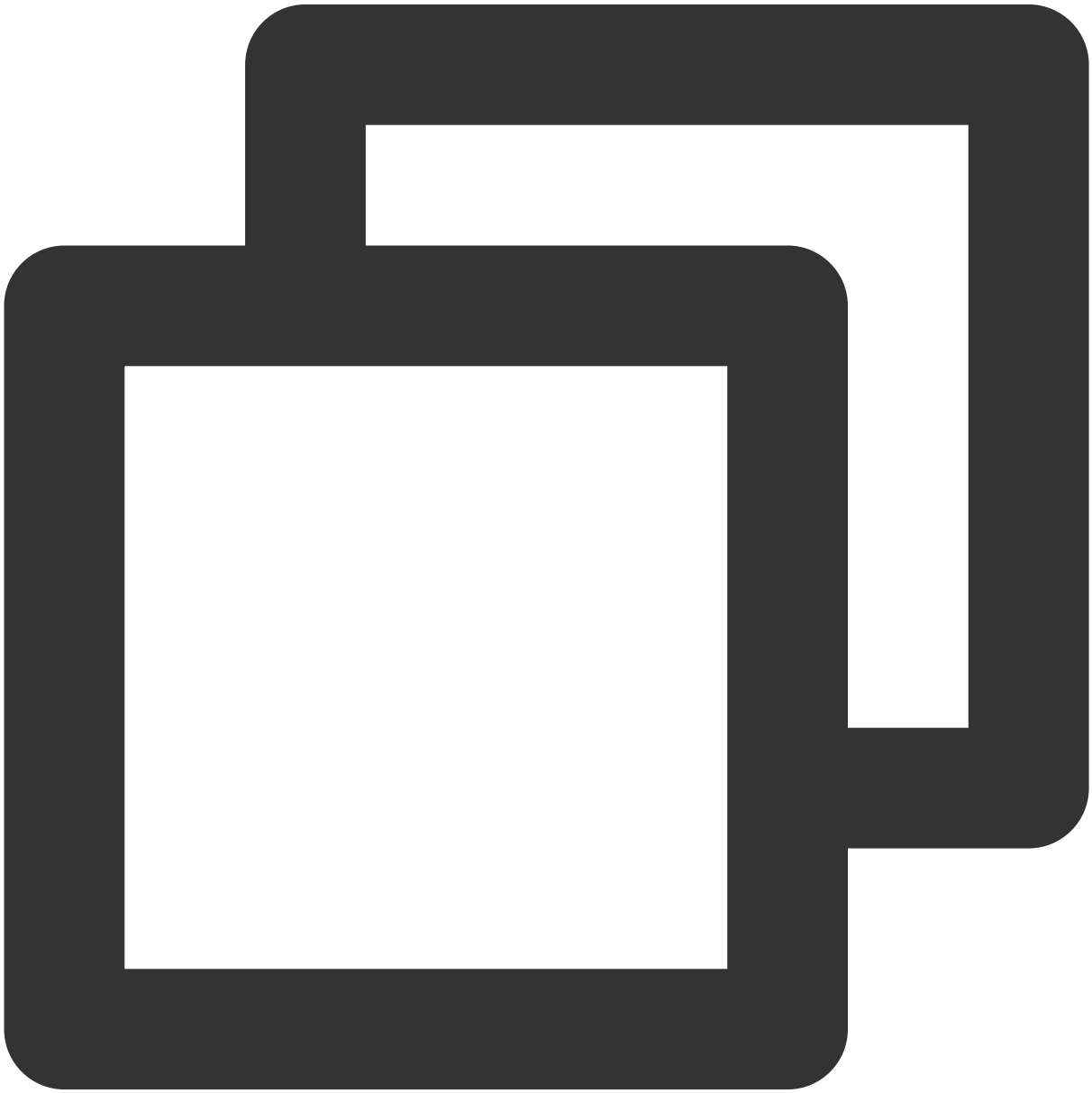
节点名称（关键字）	父节点	描述	类型
Tag	Response.Template	模板类型，SmartCover	String
Name	Response.Template	模板名字	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
TemplateId	Response.Template	模板 ID	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
SmartCover	Response.Template	详细的模板参数，同上述请求体部分 SmartCover 说明	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

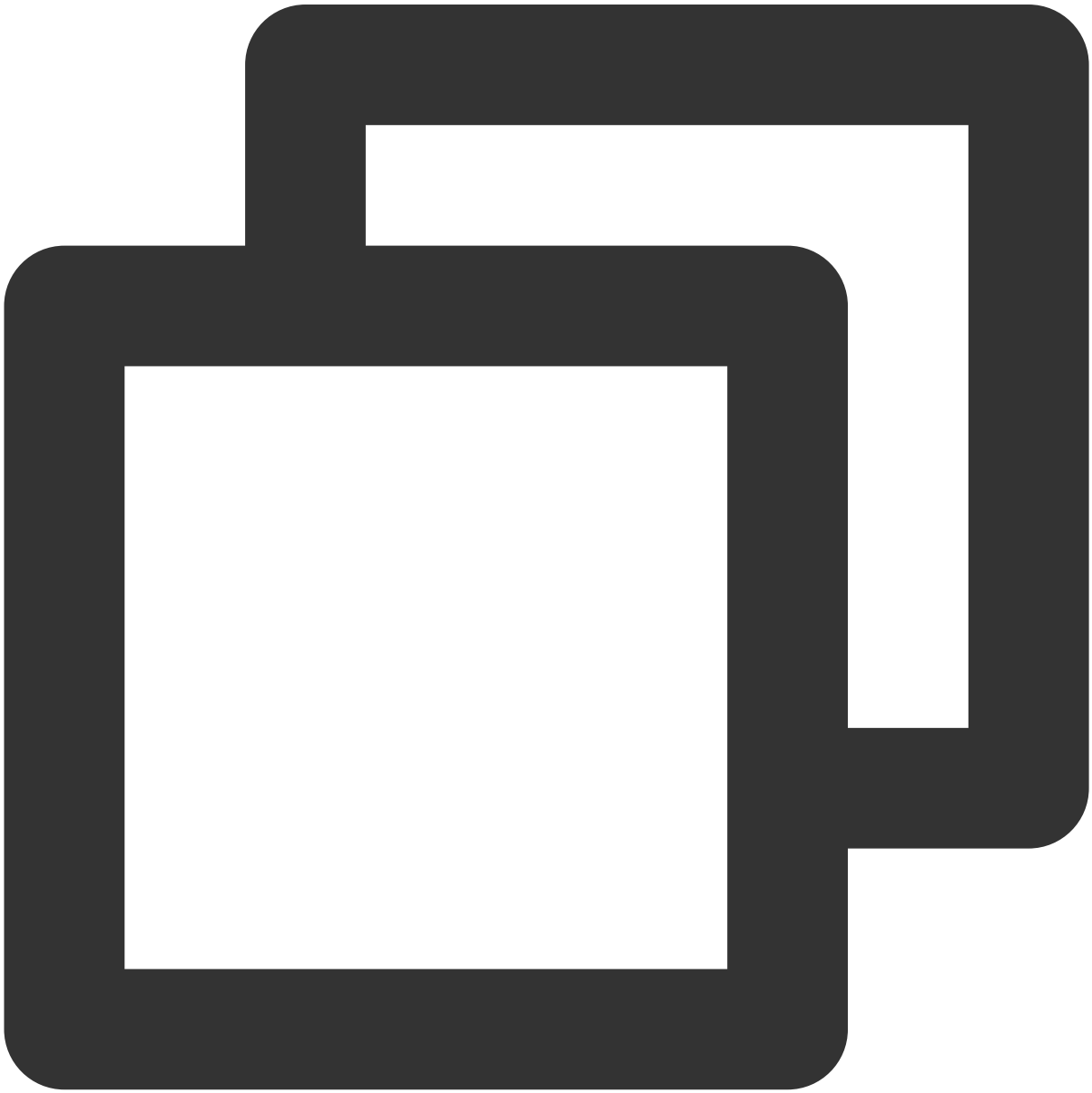
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>SmartCover</Tag>
    <Name>TemplateName</Name>
    <SmartCover>
        <Format>jpg</Format>
```

```
        <Width>1280</Width>
        <Height>960</Height>
        <Count>10</Count>
        <DeleteDuplicates>true</DeleteDuplicates>
    </SmartCover>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <Template>
        <Tag>SmartCover</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SmartCover>
            <Format>jpg</Format>
            <Width>1280</Width>
            <Height>960</Height>
            <Count>10</Count>
            <DeleteDuplicates>true</DeleteDuplicates>
        </SmartCover>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

删除媒体处理模板

DeleteMediaTemplate

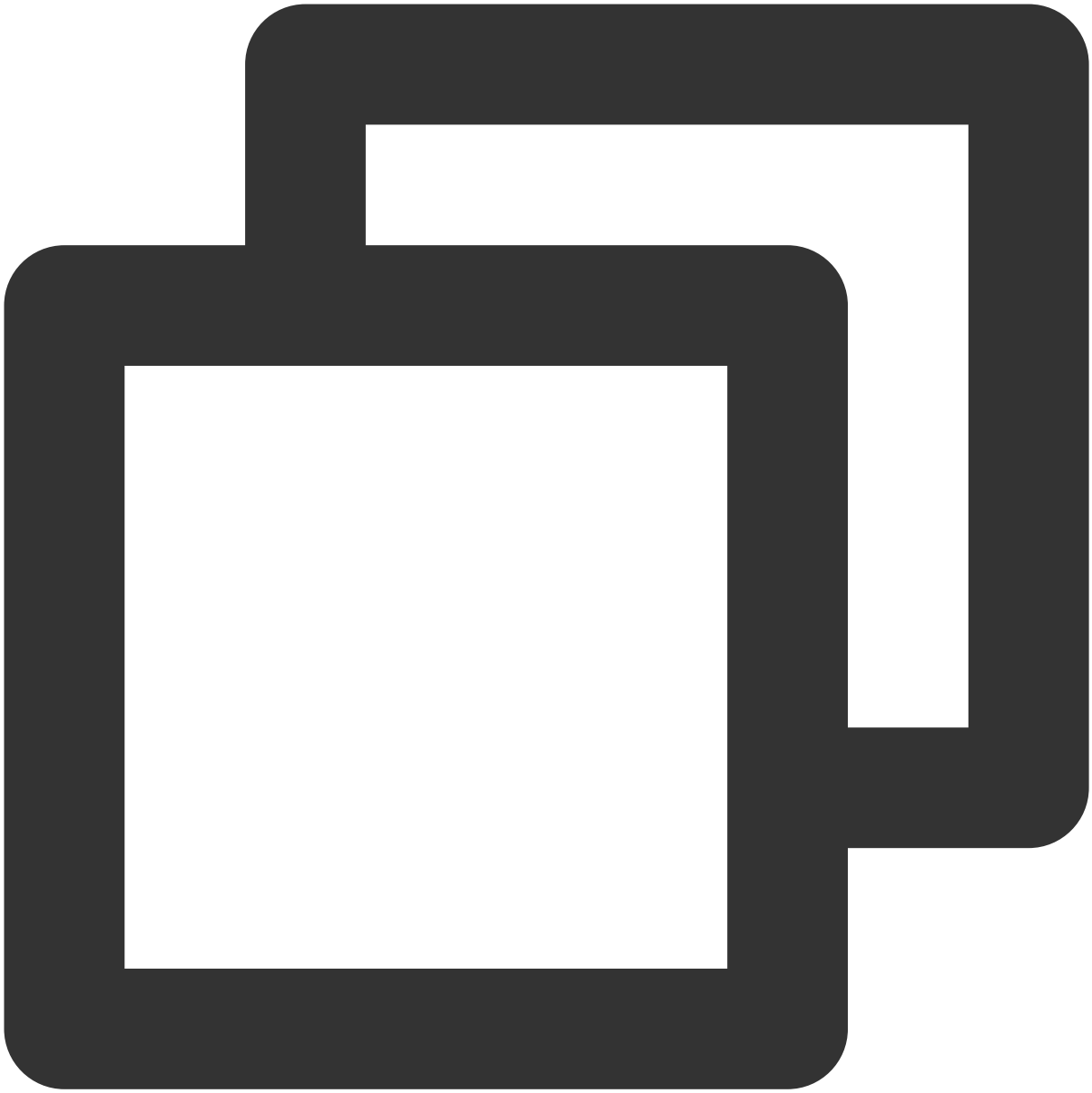
最近更新时间：2023-12-30 16:12:55

功能描述

删除模板。

请求

请求实例



```
DELETE /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

请求头

公共头部

该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

非公共头部

该请求操作无特殊的请求头部信息。

请求体

该请求的请求体为空。

响应

响应头

公共响应头

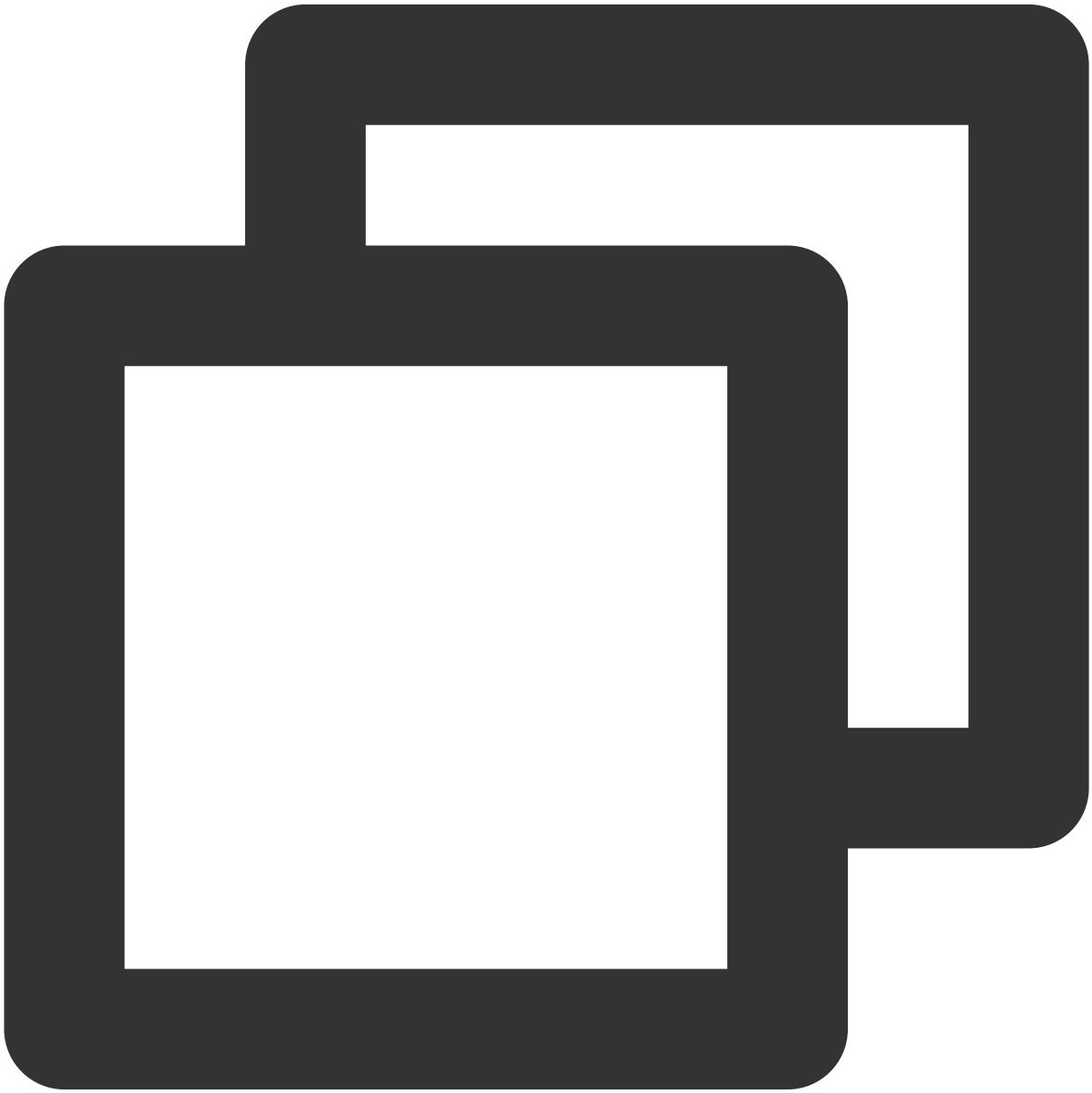
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

特有响应头

该响应无特殊的响应头。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TemplateId>A</TemplateId>
<Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

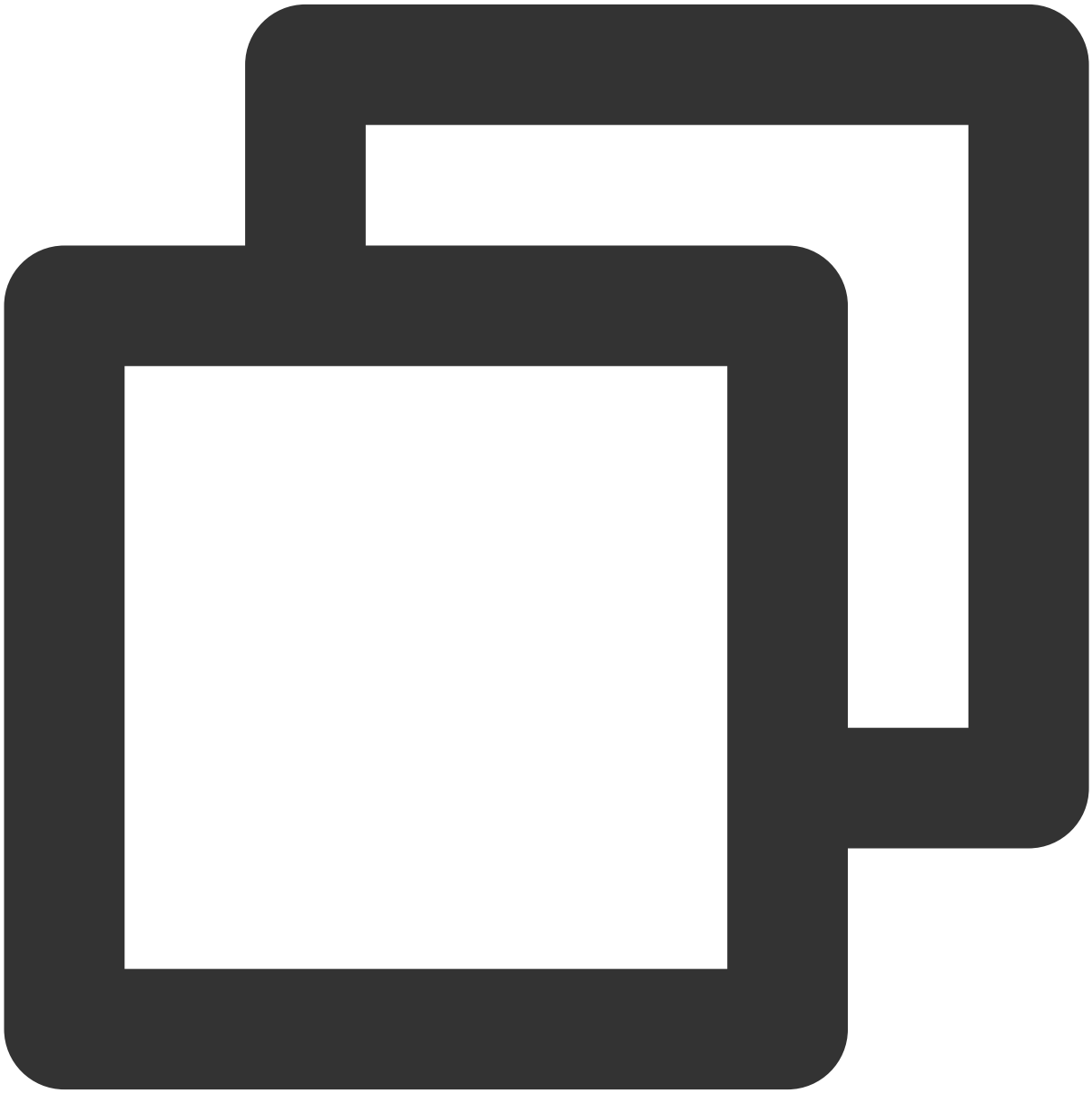
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TemplateId	Response	被删除的模板的 ID	String

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

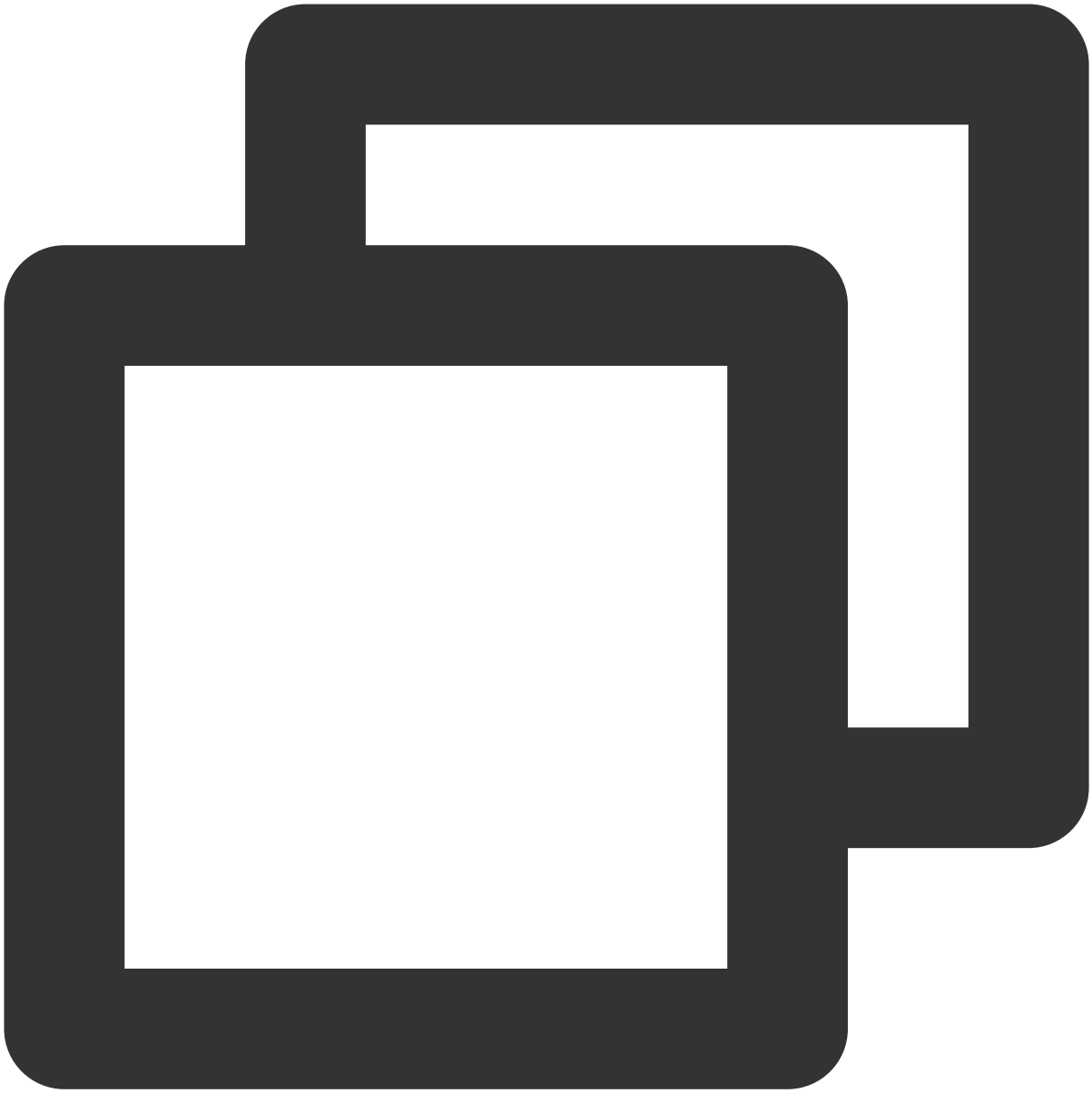
实际案例

请求



```
DELETE /template/t143ae6e040af6431aa772c9ec3f0a3f36 HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJjYmU0YmRfOTBmYTUwNjRfODhjXzE=

<Response>
    <RequestId>NjJjYmU0YmRfOTBmYTUwNjRfODhjXzE=</RequestId>
    <TemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</TemplateId>
```

```
<Response>
```

查询媒体处理模板

DescribeMediaTemplates

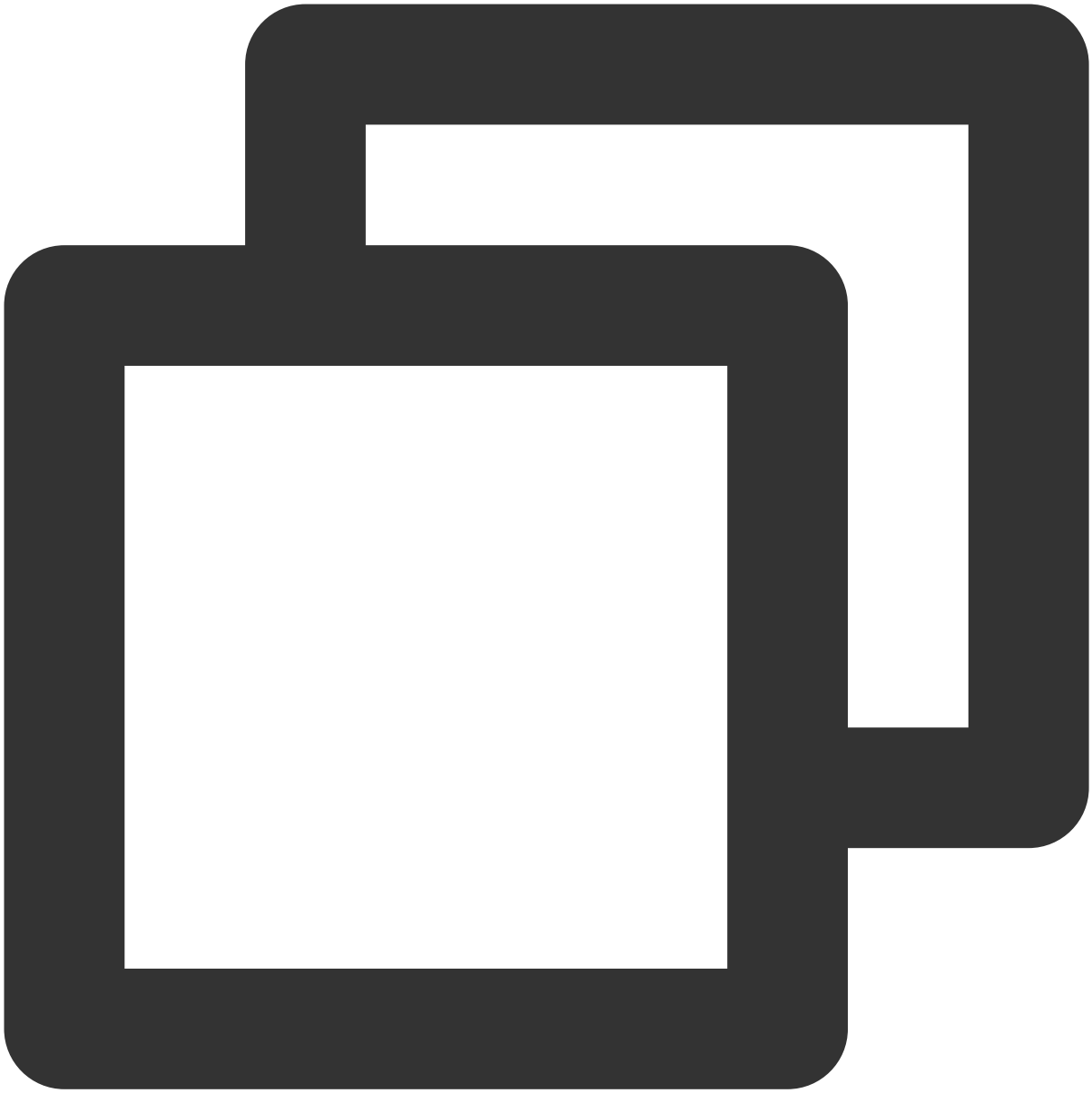
最近更新时间：2023-12-30 16:14:46

功能描述

查询模板。

请求

请求示例



```
GET /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String （详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	必选
tag	无	模板类型，默认值: All	String	否
category	无	Official(系统预设模板)，Custom(自定义模板)，默认值: Custom	String	否
ids	无	模板 ID，以 `,` 符号分割字符串	String	否
name	无	模板名称前缀	String	否
pageNumber	无	第几页，默认值:1	Integer	否
pageSize	无	每页个数，默认值:10	Integer	否

tag 支持以下几种类型:

模板类型	tag
音视频转码	Transcode
视频转动图	Animation
视频截图	Snapshot
音视频拼接	Concat
人声分离	VoiceSeparate
精彩集锦	VideoMontage
视频增强	VideoProcess
超级分辨率	SuperResolution
图片/文字水印	Watermark

版权所有：腾讯云计算（北京）有限责任公司

语音合成	Tts
音视频转码Pro	TranscodePro
极速高清	HighSpeedHd

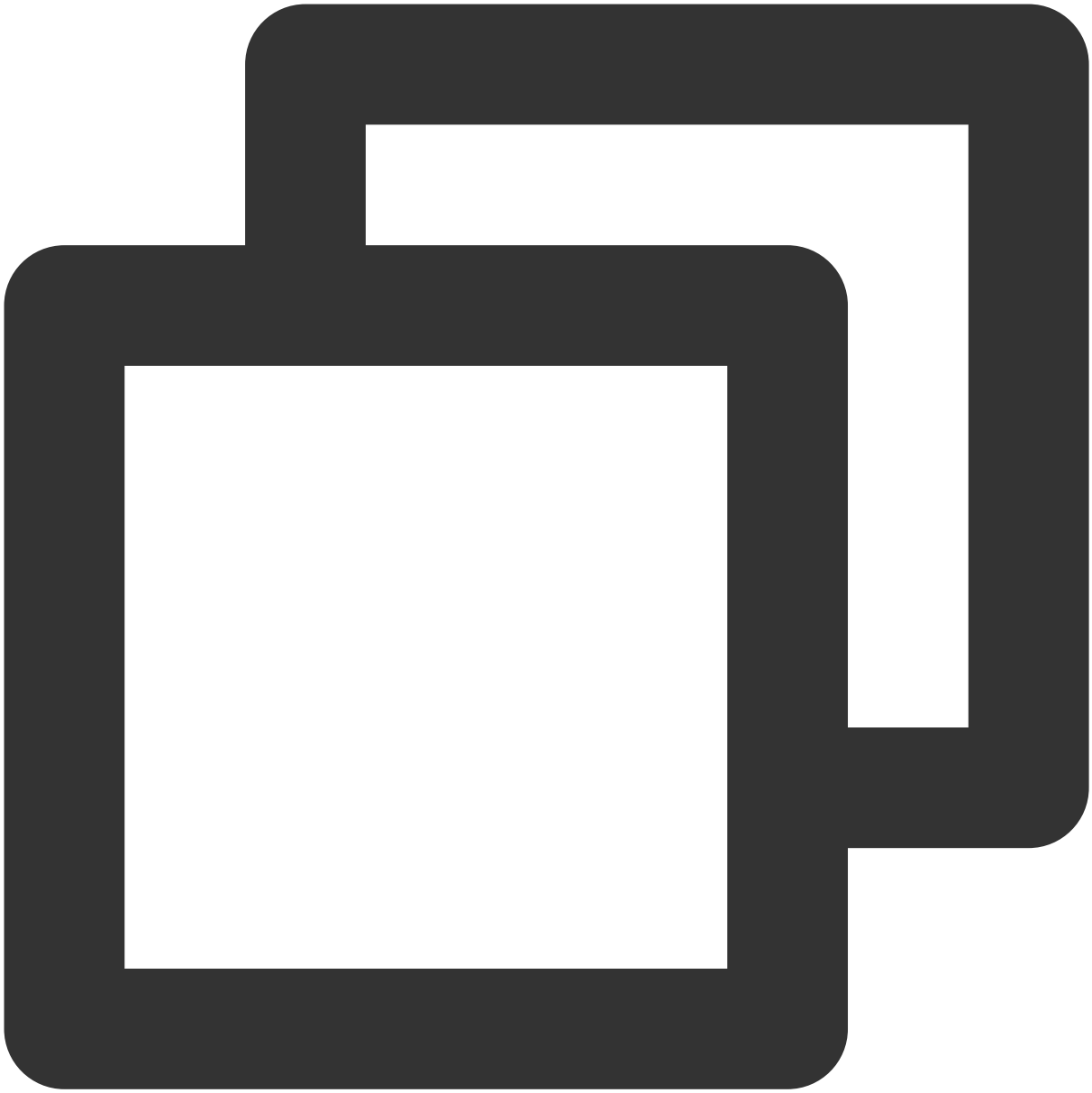
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TotalCount>1</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <TemplateList>
        <TemplateId>A</TemplateId>
        <Name>TemplateName</Name>
        <Tag>Transcode</Tag>
        <TransTpl>
            <Container>
```

```
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Profile>high</Profile>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Preset>medium</Preset>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <AdjDarMethod>scale</AdjDarMethod>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	模板总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int

PageSize	Response	每页个数，同请求中的pageSize	Int
TemplateList	Response	模板数组	Container 数组

对于不同的模板类型，TemplateList 内容同创建模板接口的Response，请参照以下链接：

音视频转码
视频转动图
视频截图
音视频拼接
人声分离
精彩集锦
视频增强
超级分辨率
图片/文字水印
语音合成
音视频转码Pro
极速高清

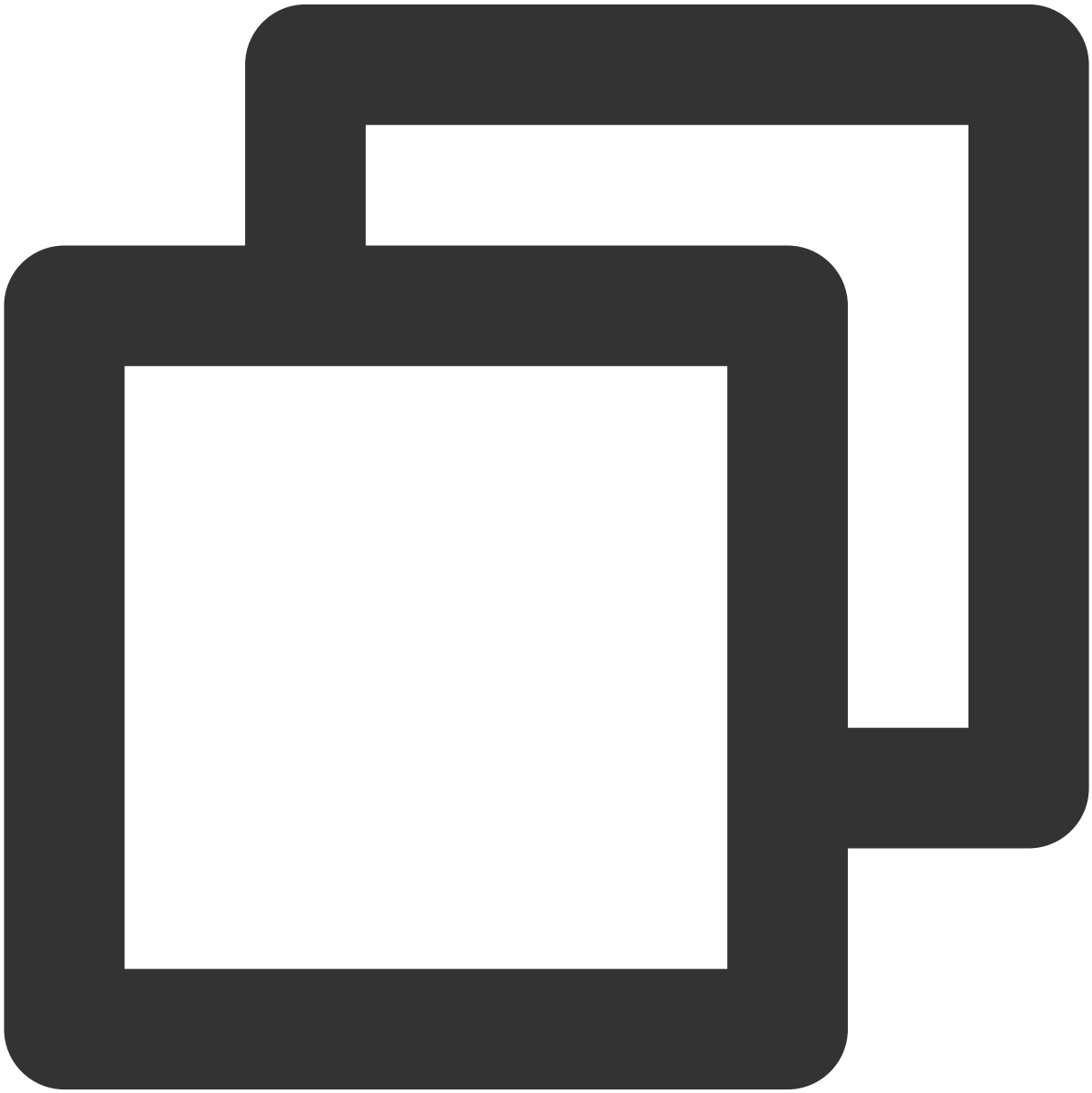
错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

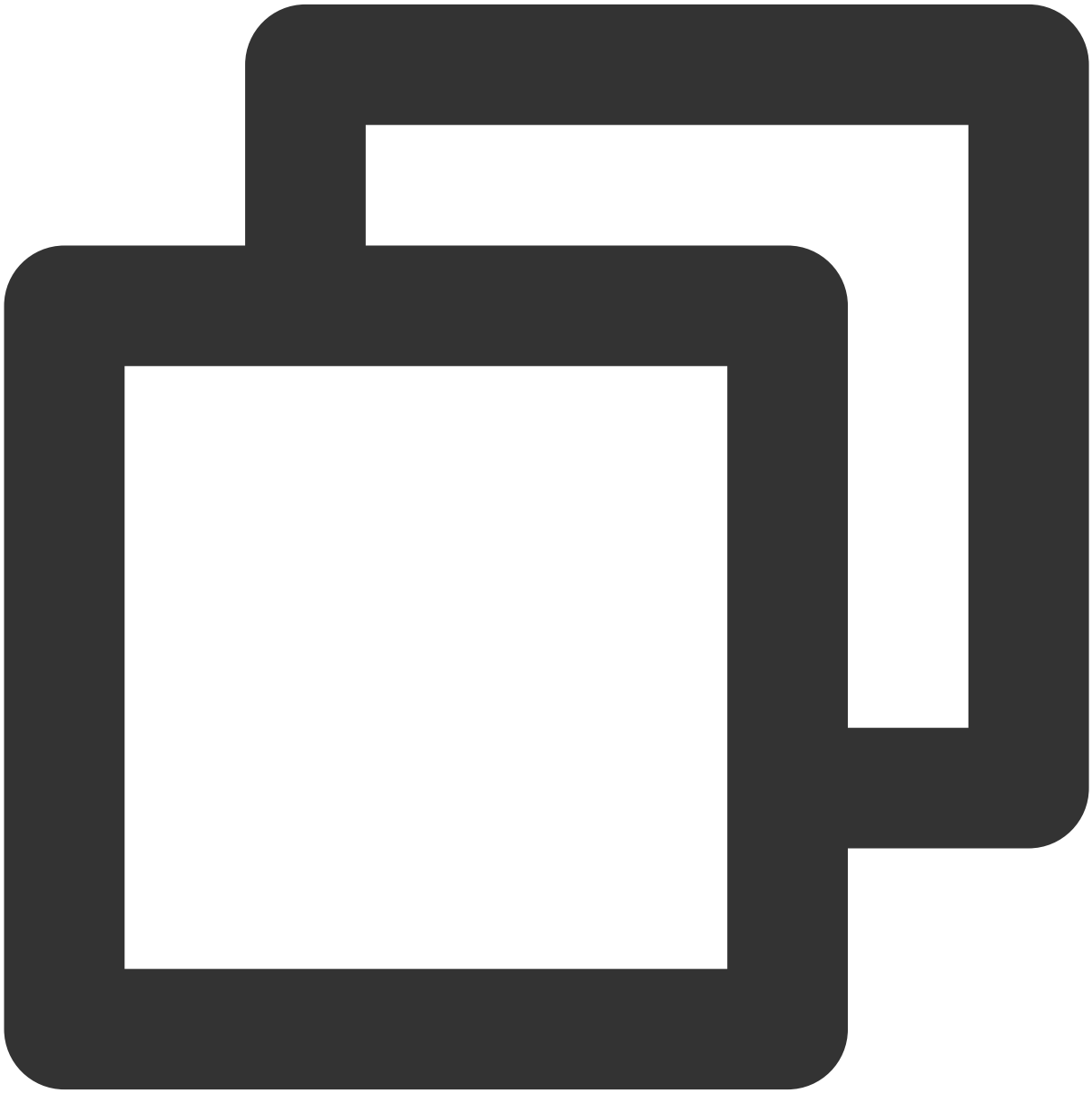
案例一：按照模板 ID 维度查询

请求



```
GET /template?ids=t1847cd4ca57f543e89f551dbe68169eb9,t1a30a323f55434a29b25bf2a16bdf
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



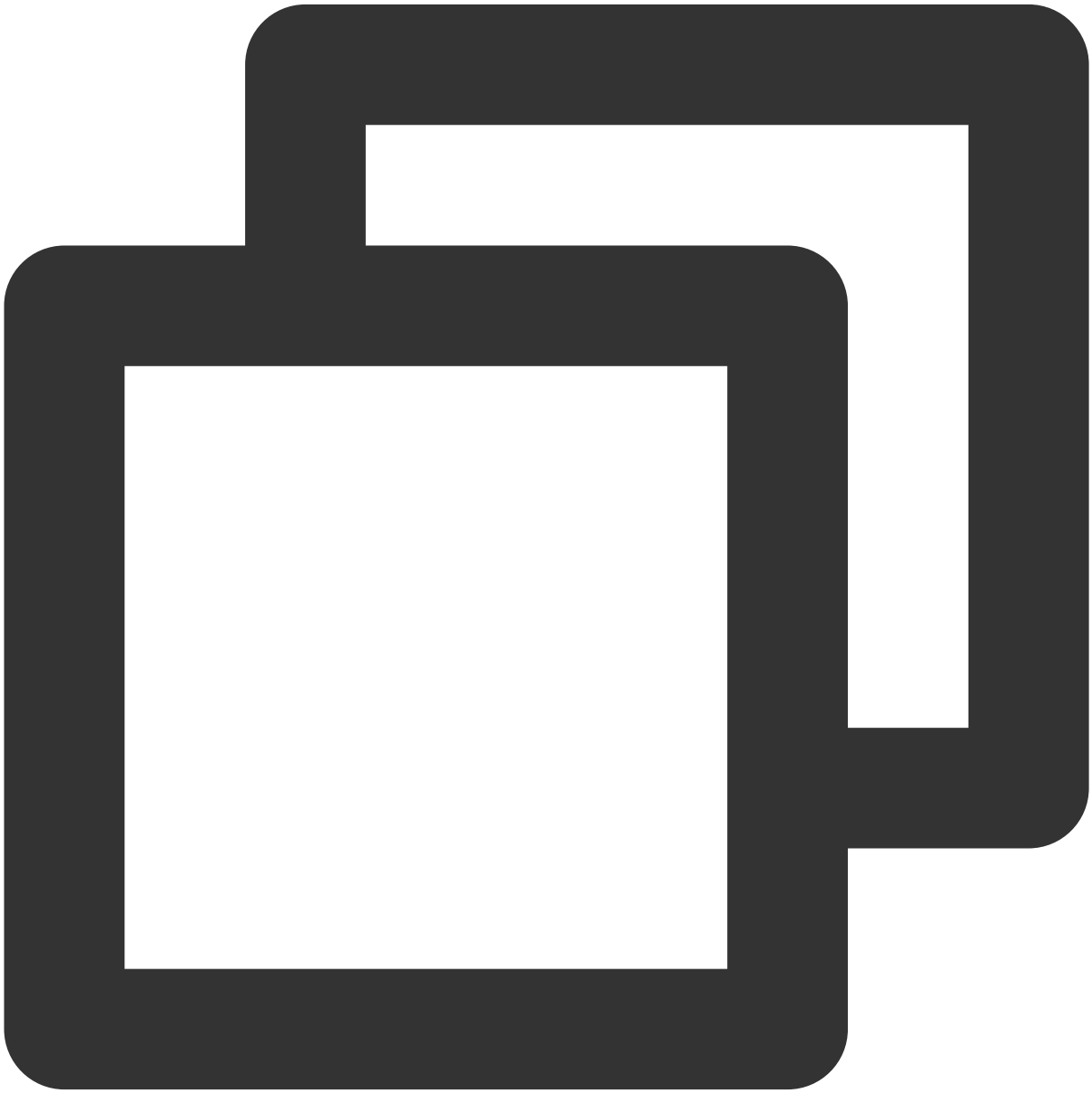
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TemplateList>
```

```
        <TemplateId>t1847cd4ca57f543e89f551dbe68169eb9</TemplateId>
        <Name>TemplateName</Name>
        <Tag>Transcode</Tag>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <TransTpl>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Profile>high</Profile>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Preset>medium</Preset>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <AdjDarMethod>scale</AdjDarMethod>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
    <TemplateList>
        <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
        <Name>template_name3544697</Name>
        <Tag>Concat</Tag>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <ConcatTemplate>
            <ConcatFragment>
                <Mode>Start</Mode>
                <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/input/ad
            </ConcatFragment>
```

```
            <Audio>
                <Codec>aac</Codec>
                <Samplerate/>
                <Bitrate/>
                <Channels/>
                <Remove>false</Remove>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Width>1280</Width>
                <Height>960</Height>
                <Fps>15</Fps>
                <Remove>false</Remove>
                <Crf>25</Crf>
            </Video>
            <Container>
                <Format>mp4</Format>
            </Container>
            <DirectConcat>false</DirectConcat>
        </ConcatTemplate>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
    </TemplateList>
    <NonExistTIDs>
        <TemplateId>C</TemplateId>
    </NonExistTIDs>
</Response>
```

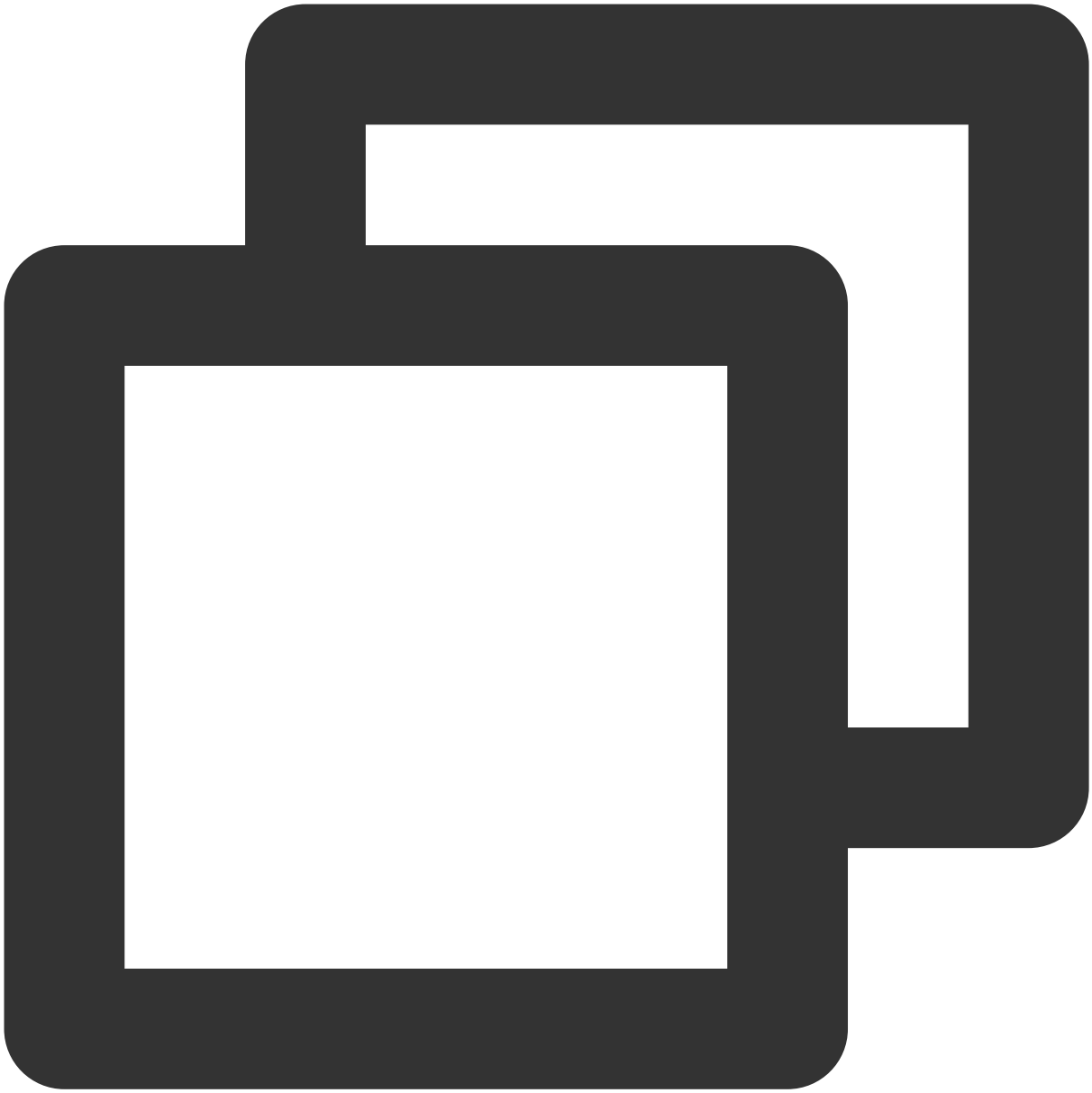
案例二：按照分页列表维度查询

请求



```
GET /template?pageSize=10&pageNumber=1 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TotalCount>2</TotalCount>
```

```
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <TemplateList>
        <TemplateId>t1847cd4ca57f543e89f551dbe68169eb9</TemplateId>
        <Name>TemplateName</Name>
        <Tag>Transcode</Tag>
        <TransTpl>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Profile>high</Profile>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Preset>medium</Preset>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <AdjDarMethod>scale</AdjDarMethod>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
    <TemplateList>
        <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
        <Name>template_name3544697</Name>
        <Tag>Concat</Tag>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <ConcatTemplate>
            <ConcatFragment>
                <Mode>Start</Mode>
                <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/input/ad
```

```
            </ConcatFragment>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate/>
                <Bitrate/>
                <Channels/>
                <Remove>false</Remove>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Width>1280</Width>
                <Height>960</Height>
                <Fps>15</Fps>
                <Remove>false</Remove>
                <Crf>25</Crf>
            </Video>
            <Container>
                <Format>mp4</Format>
            </Container>
            <DirectConcat>false</DirectConcat>
        </ConcatTemplate>
        <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
        <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
    </TemplateList>
</Response>
```

更新媒体处理模板
更新动图模板

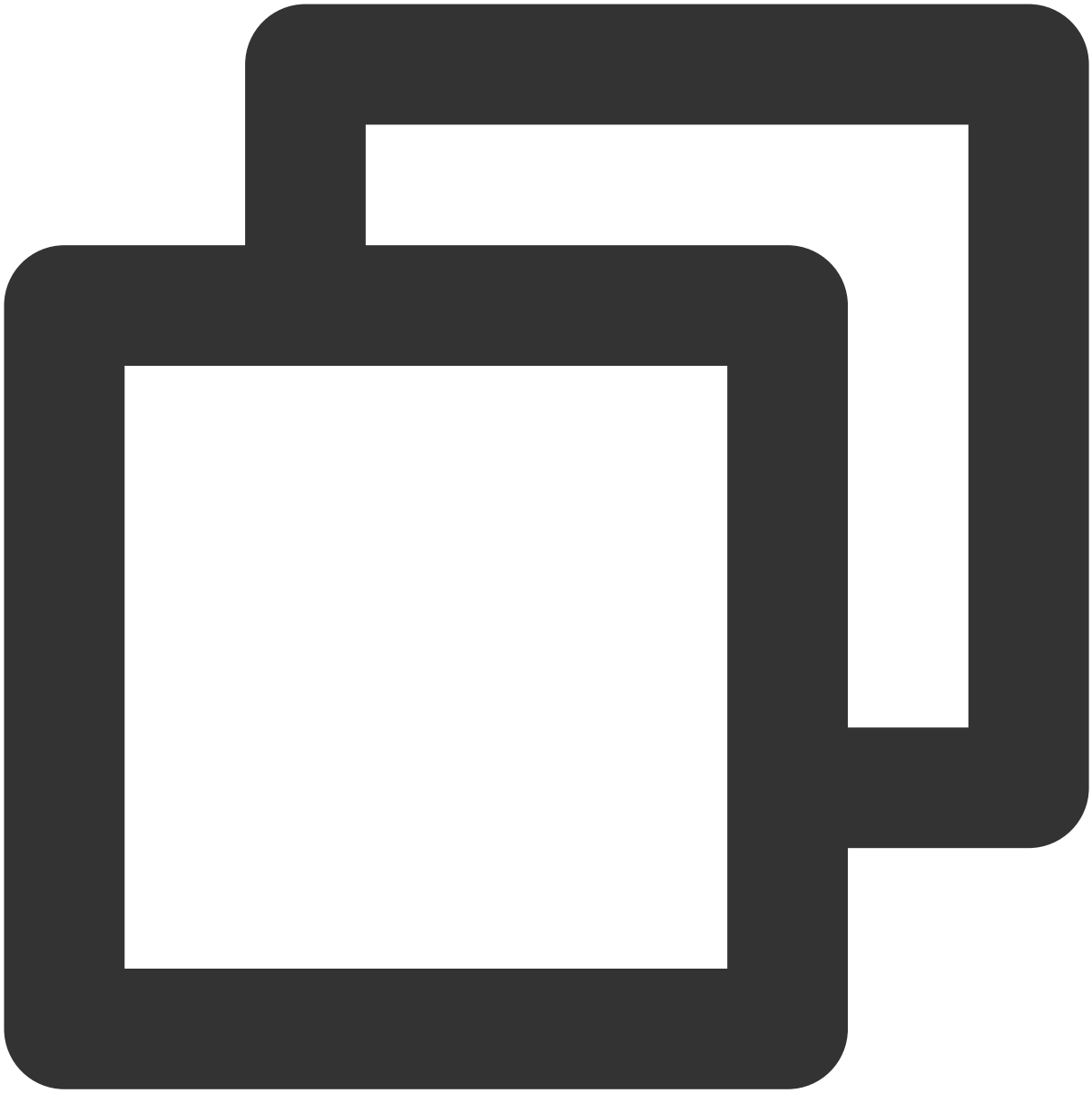
最近更新时间：2023-12-30 16:13:56

功能描述

更新动图模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

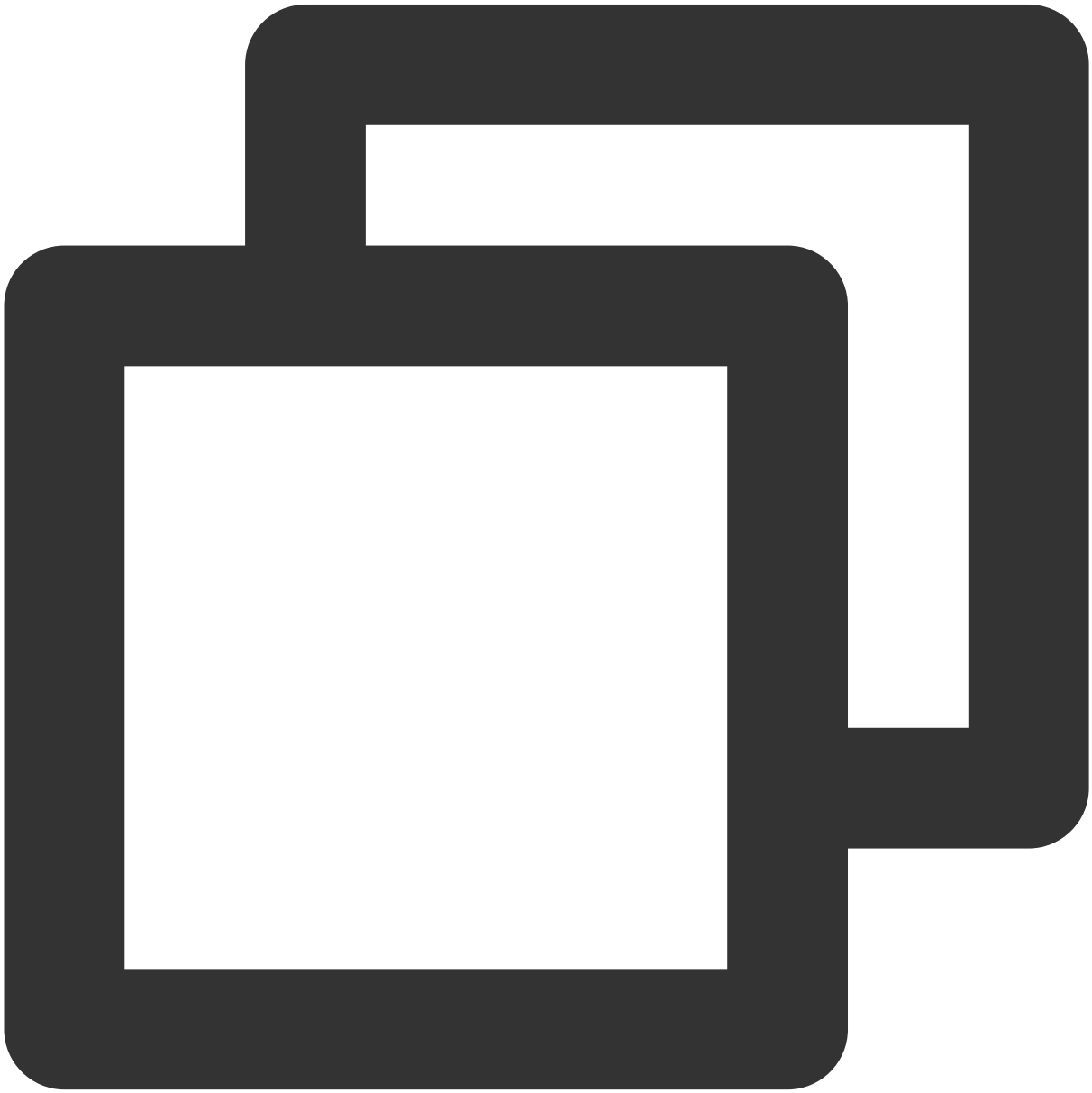
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Animation</Tag>
    <Name>TemplateName</Name>
    <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
    <Container>
        <Format>gif</Format>
    </Container>
    <Video>
        <Codec>gif</Codec>
        <Width>1280</Width>
        <Height></Height>
        <Fps>15</Fps>
        <AnimateOnlyKeepKeyFrame>false</AnimateOnlyKeepKeyFrame>
    </Video>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建动图模板接口的 Request	Container	是

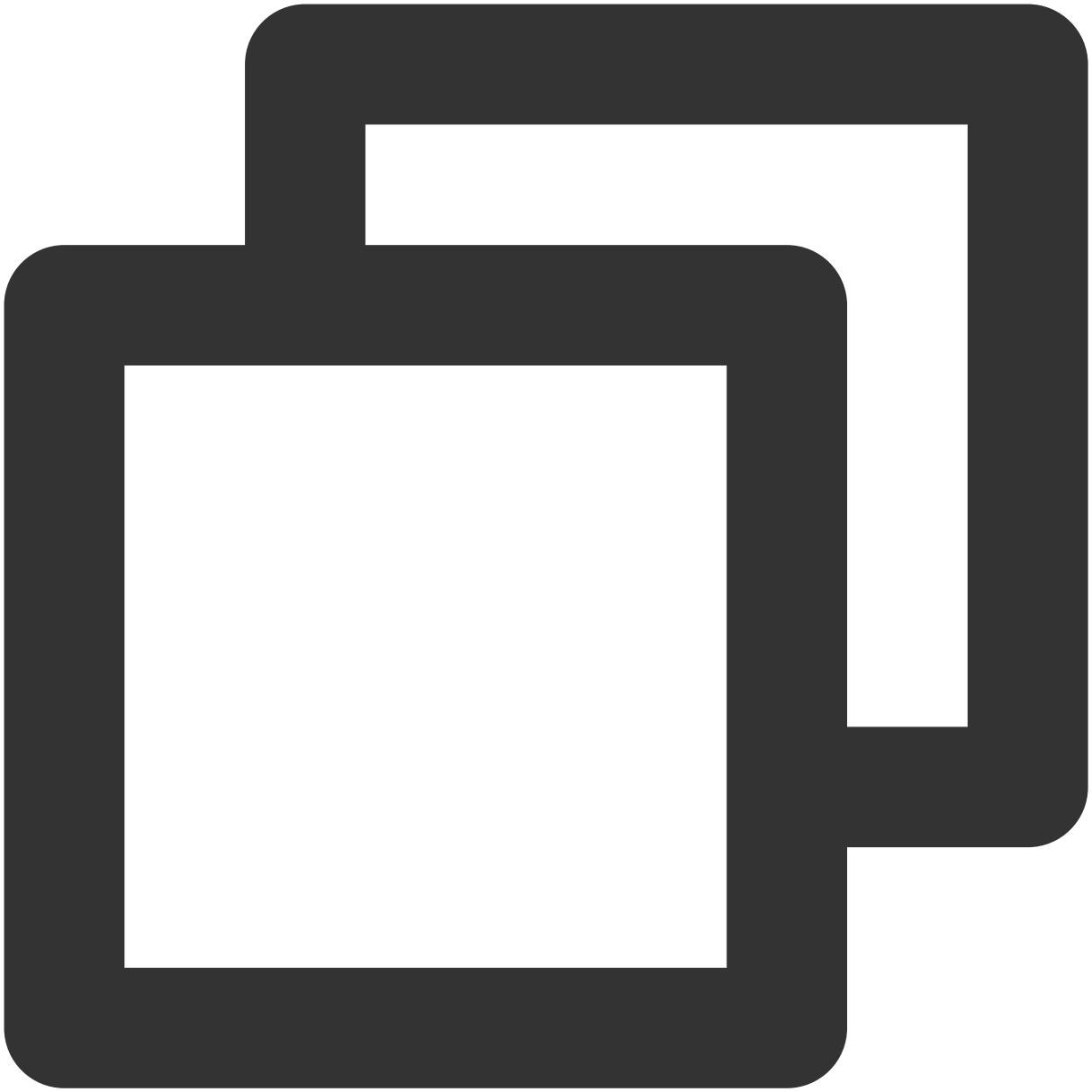
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Animation</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <TransTpl>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>gif</Codec>
```

```
                <Width>1280</Width>
                <Height></Height>
                <Fps>15</Fps>
                <AnimateOnlyKeepKeyFrame>false</AnimateOnlyKeepKeyFrame>
            </Video>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

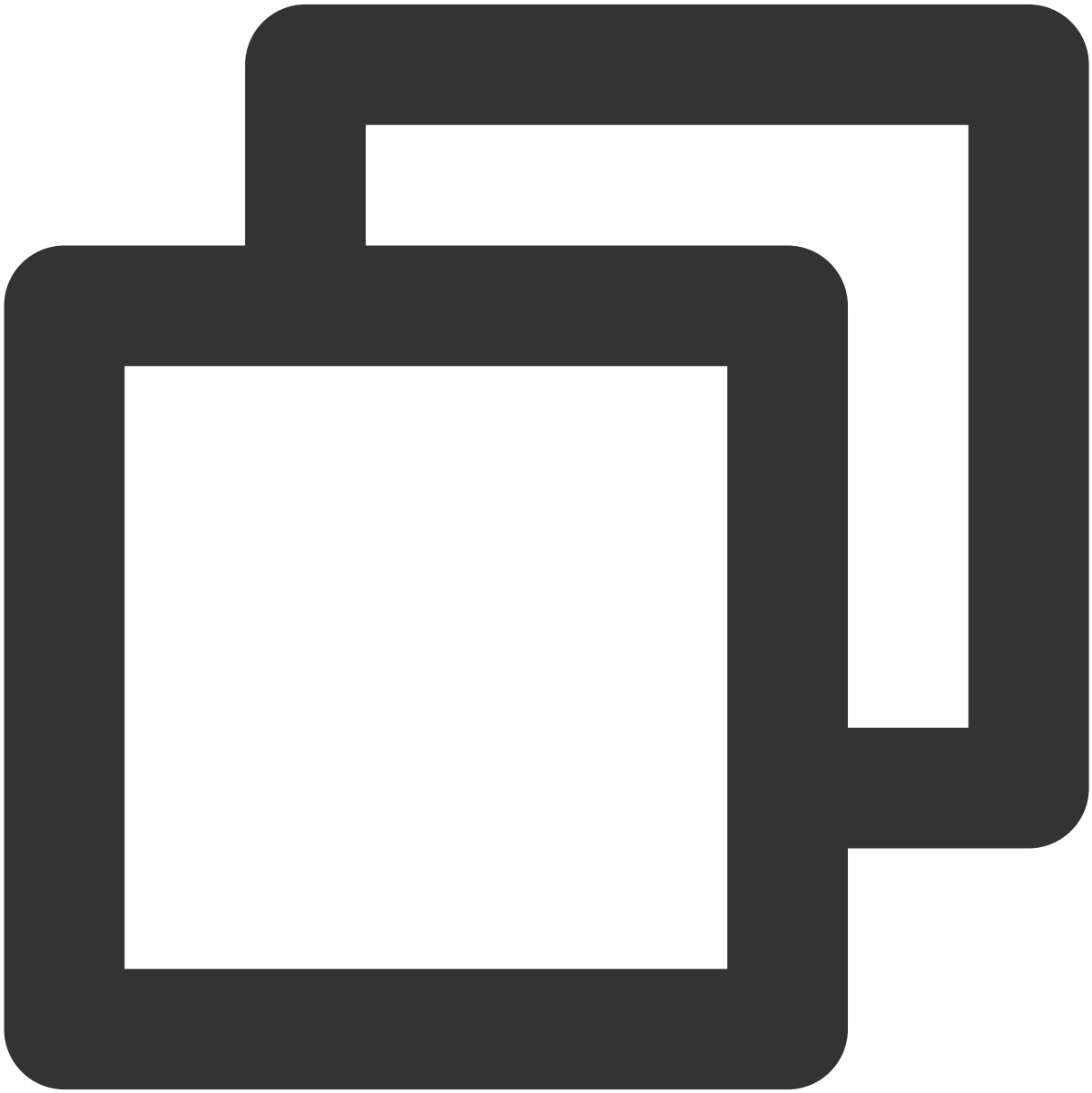
节点名称（关键字）	父节点	描述	类型
Response	无	同创建动图模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

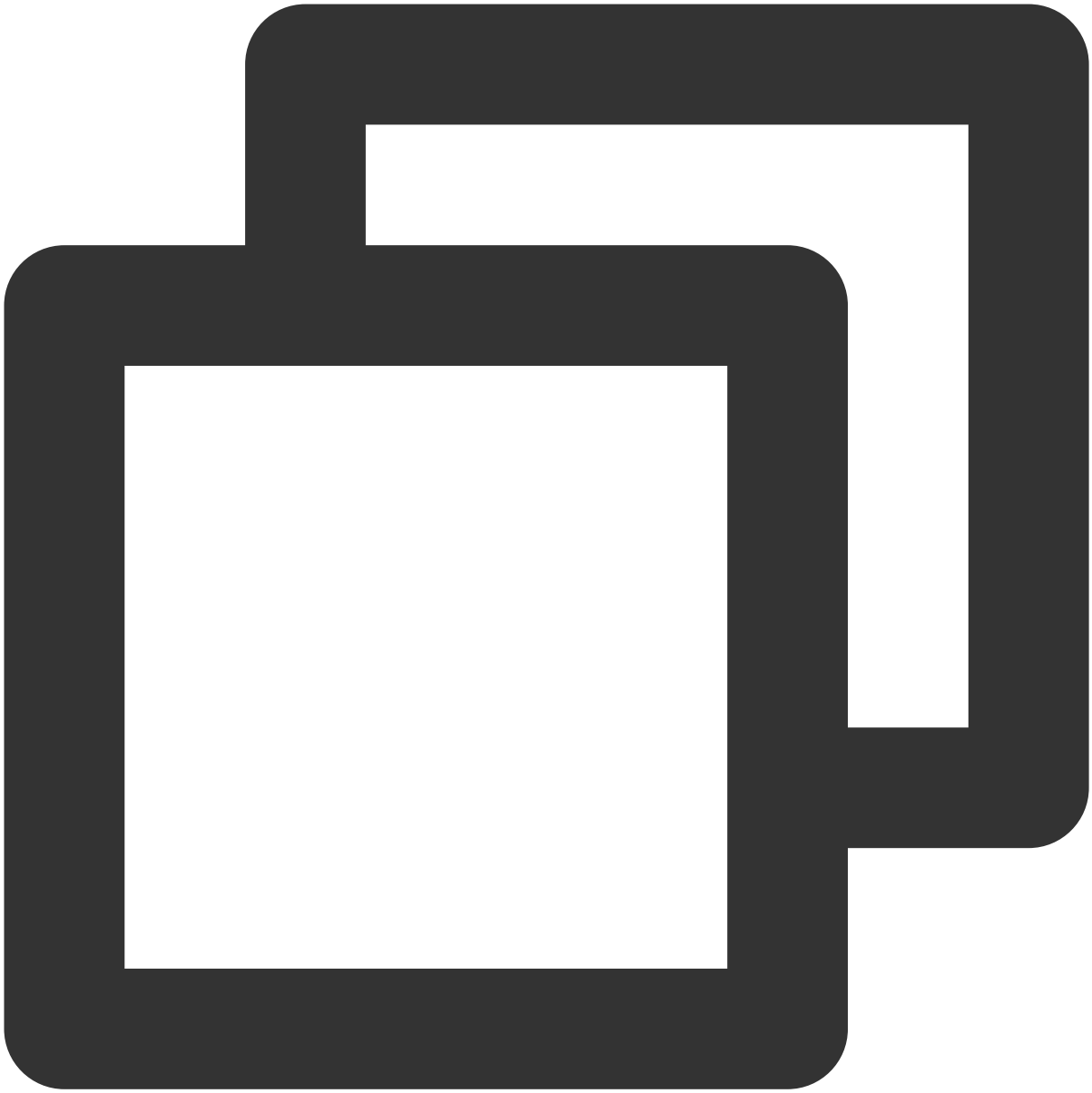
请求



```
PUT /template/t1f16e1dfbdc994105b31292d45710642a HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Animation</Tag>
    <Name>TemplateName</Name>
    <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
    <Container>
```

```
        <Format>gif</Format>
    </Container>
    <Video>
        <Codec>gif</Codec>
        <Width>1280</Width>
        <Height></Height>
        <Fps>15</Fps>
        <AnimateOnlyKeepKeyFrame>false</AnimateOnlyKeepKeyFrame>
    </Video>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <Tag>Animation</Tag>
    <Name>TemplateName</Name>
```

```
    <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
    <TransTpl>
      <Container>
         <Format>mp4</Format>
      </Container>
      <Video>
        <Codec>gif</Codec>
        <Width>1280</Width>
        <Height></Height>
        <Fps>15</Fps>
        <AnimateOnlyKeepKeyFrame>false</AnimateOnlyKeepKeyFrame>
      </Video>
      <TimeInterval>
         <Start>0</Start>
         <Duration>60</Duration>
      </TimeInterval>
   </TransTpl>
   <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
   <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
</Response>
```

更新拼接模板

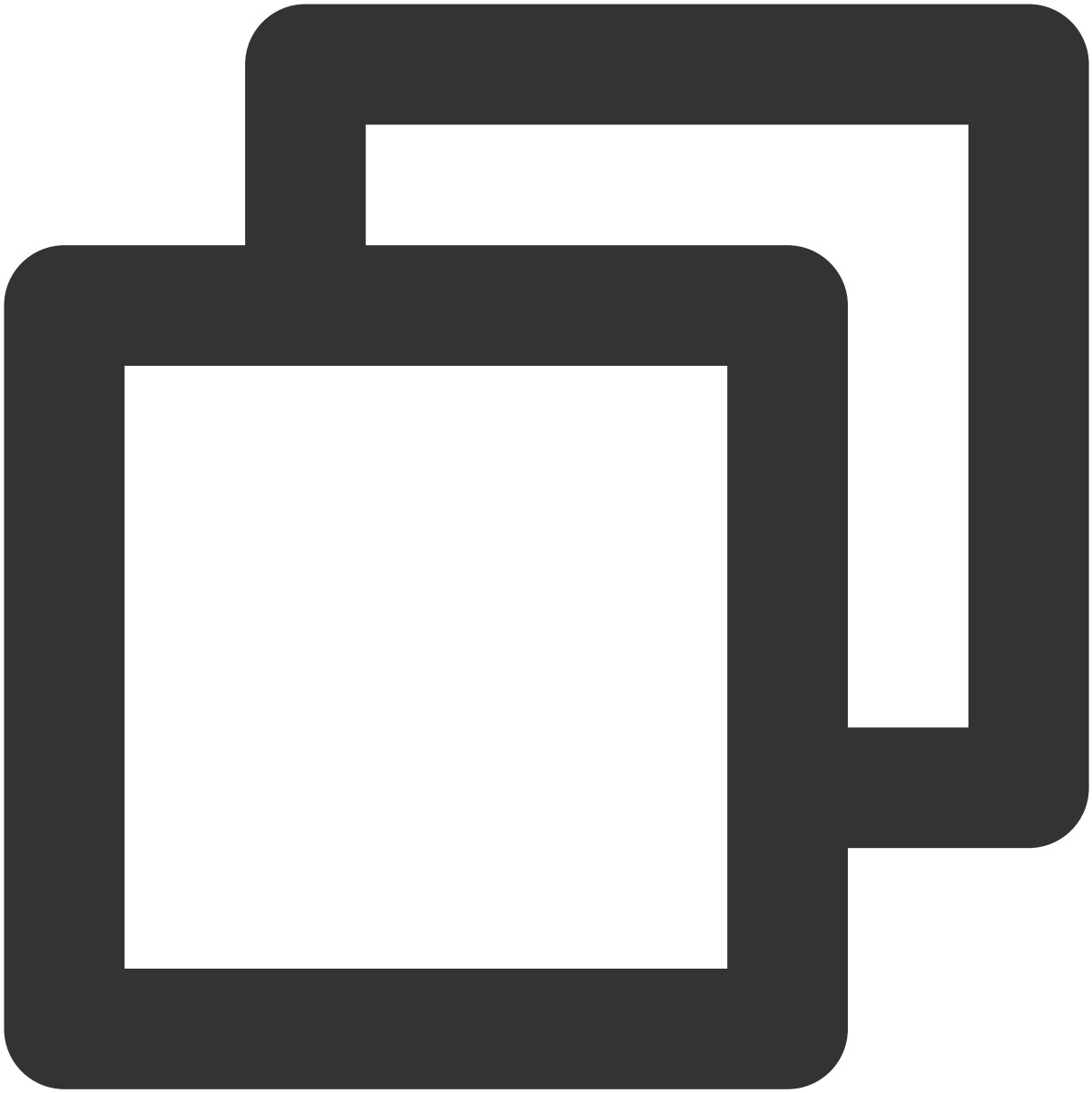
最近更新时间：2023-12-30 16:15:28

功能描述

更新拼接模板。

请求

请求实例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

公共头部

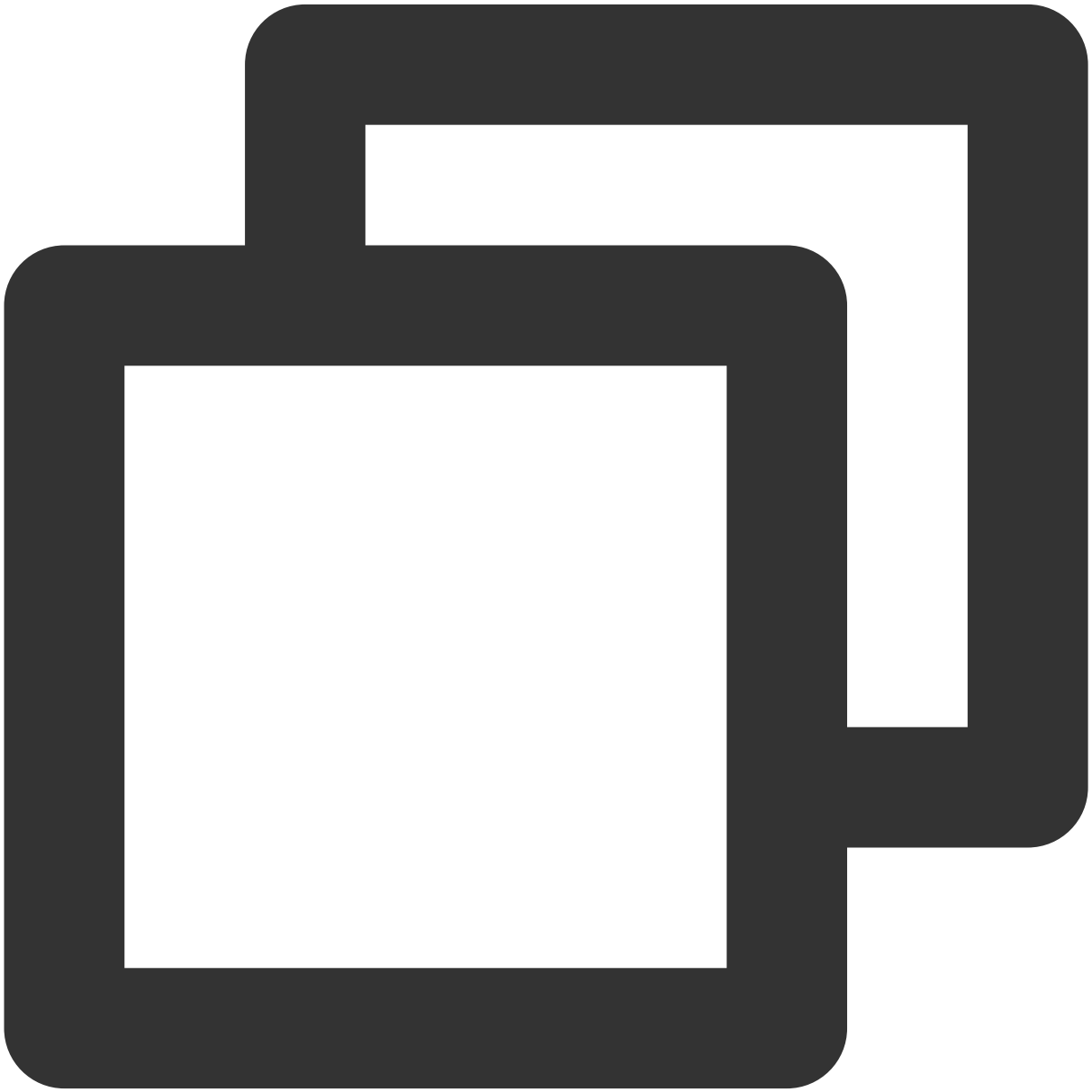
该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

非公共头部

该请求操作无特殊的请求头部信息。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
    <Tag>Concat</Tag>
    <Name>TemplateName</Name>
    <ConcatTemplate>
        <ConcatFragment>
            <Mode>Start</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4</Ur
        </ConcatFragment>
        <ConcatFragment>
            <Mode>End</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</Url>
```

```
        </ConcatFragment>
        <Audio>
            <Codec>mp3</Codec>
            <Samplerate></Samplerate>
            <Bitrate></Bitrate>
            <Channels></Channels>
        </Audio>
        <Video>
            <Codec>H.264</Codec>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Height></Height>
            <Fps>30</Fps>
        </Video>
        <Container>
            <Format>mp4</Format>
        </Container>
        <AudioMix>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</Au
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMix>
    </ConcatTemplate>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Request	无	同创建拼接模板接口的 Request	Container

响应

响应头

公共响应头

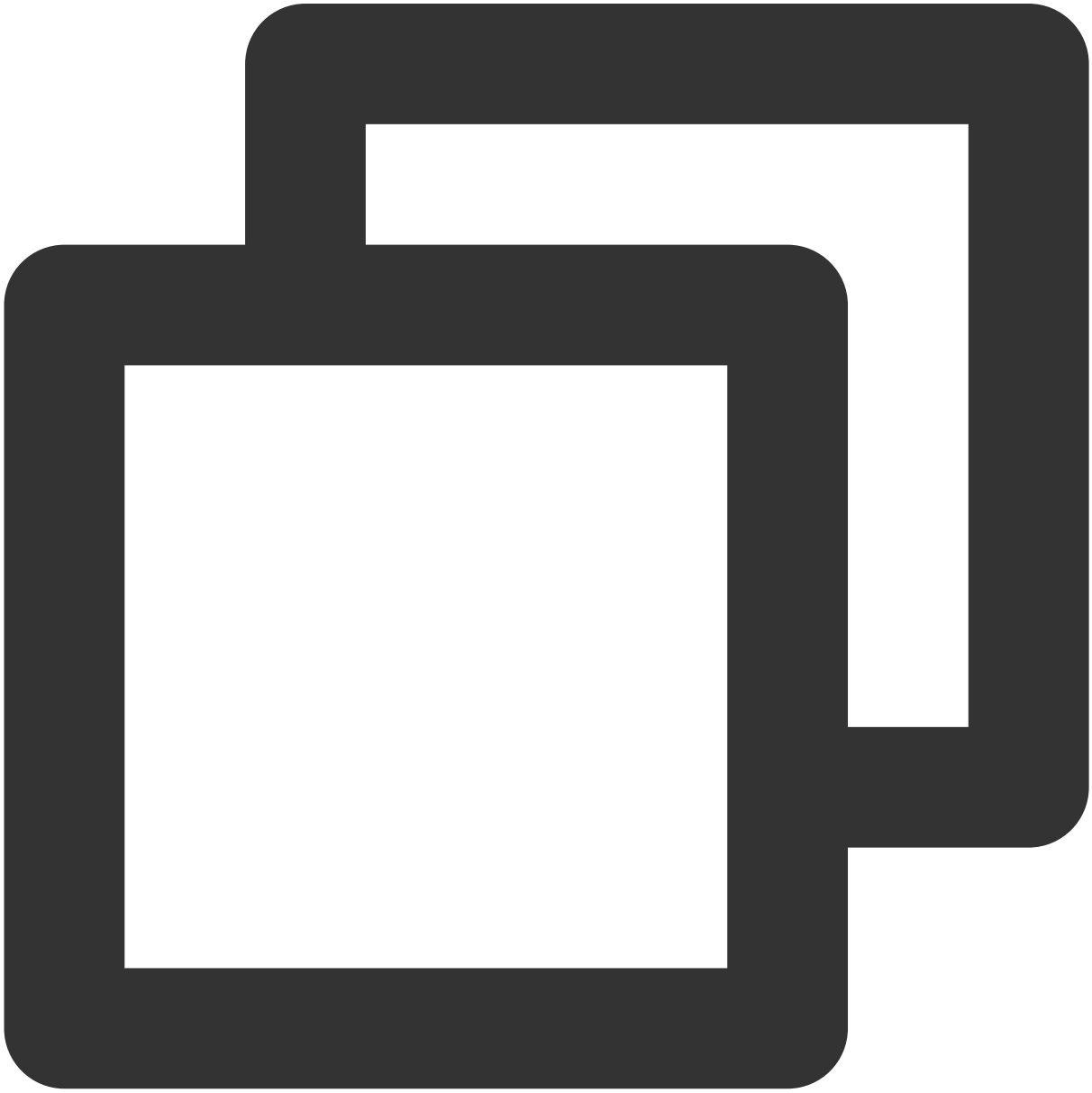
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

特有响应头

该响应无特殊的响应头。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Tag>Concat</Tag>
        <Name>TemplateName</Name>
        <ConcatTemplate>
            <ConcatFragment>
                <Mode>Start</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4
            </ConcatFragment>
            <ConcatFragment>
                <Mode>End</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</
            </ConcatFragment>
            <Audio>
                <Codec>mp3</Codec>
                <Samplerate></Samplerate>
                <Bitrate></Bitrate>
                <Channels></Channels>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Height></Height>
                <Fps>30</Fps>
            </Video>
            <Container>
                <Format>mp4</Format>
            </Container>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
                <EffectConfig>
                    <EnableStartFadein>true</EnableStartFadein>
                    <StartFadeinTime>3</StartFadeinTime>
                    <EnableEndFadeout>false</EnableEndFadeout>
                    <EndFadeoutTime>0</EndFadeoutTime>
                    <EnableBgmFade>true</EnableBgmFade>
                    <BgmFadeTime>1.7</BgmFadeTime>
                </EffectConfig>
            </AudioMix>
        </ConcatTemplate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的数据内容如下：

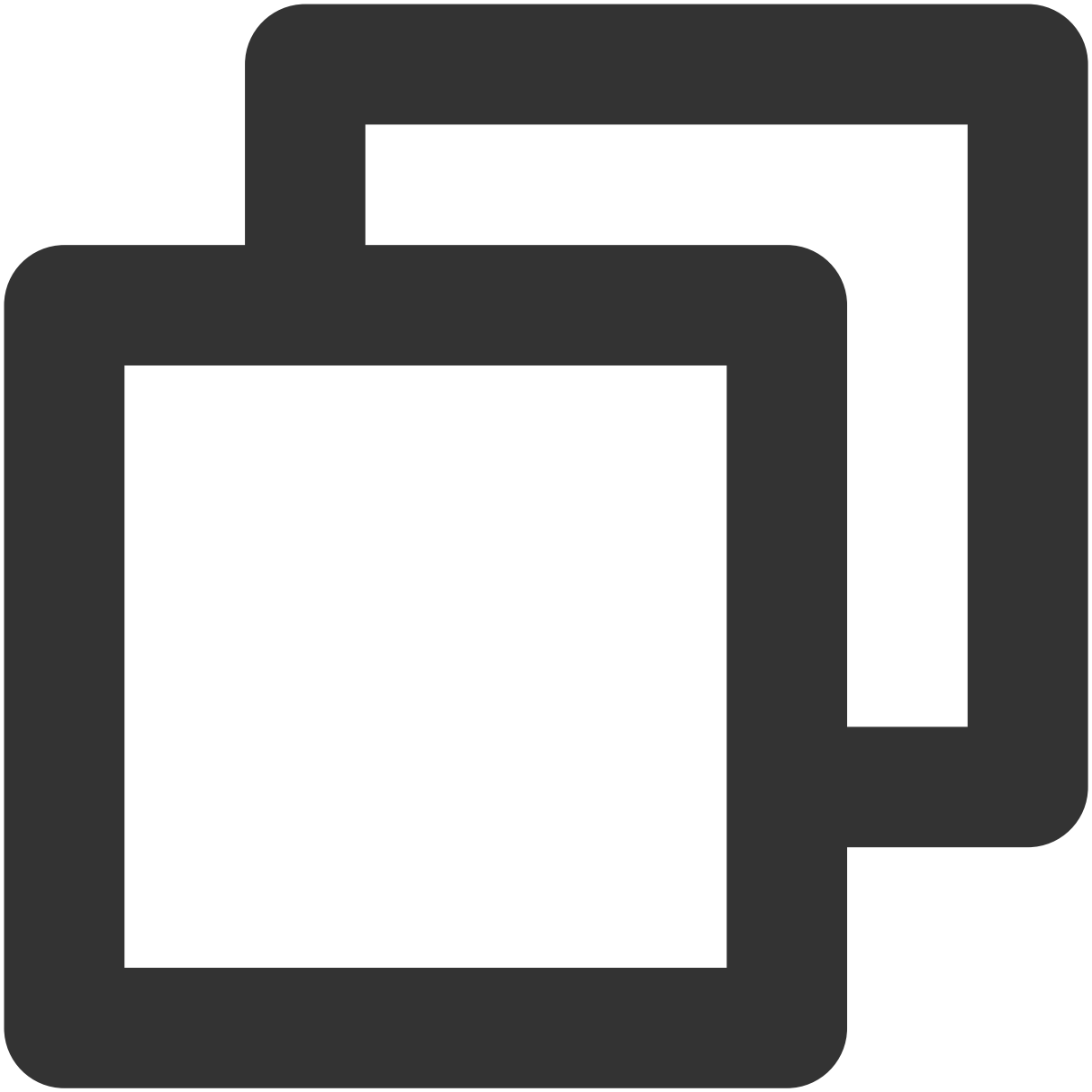
节点名称（关键字）	父节点	描述	类型
Response	无	同创建拼接模板接口的 Response	Container

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

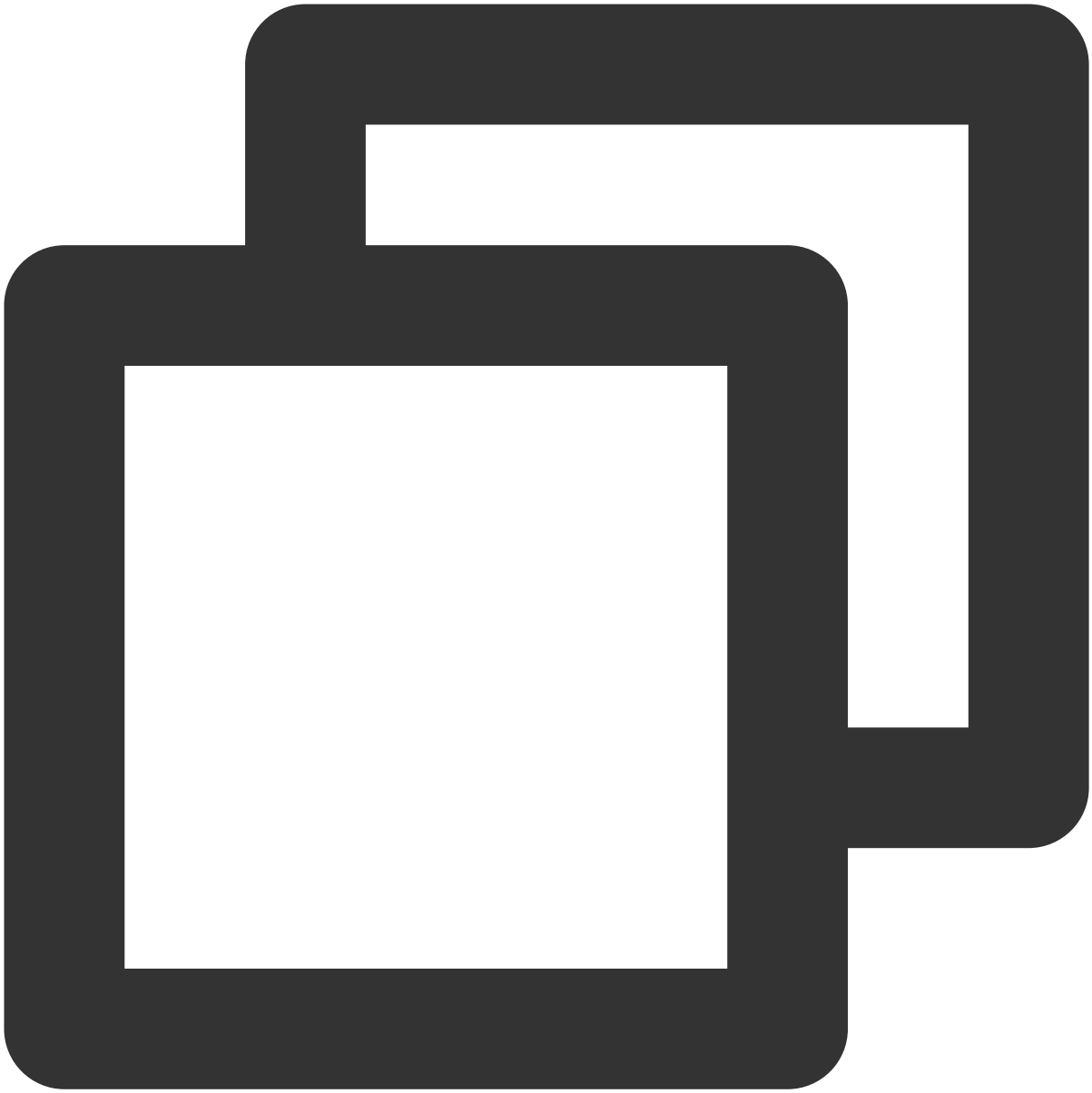
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Concat</Tag>
    <Name>TemplateName</Name>
    <ConcatTemplate>
        <ConcatFragment>
```

```
            <Mode>Start</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4</Ur
        </ConcatFragment>
        <ConcatFragment>
            <Mode>End</Mode>
            <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</Url>
        </ConcatFragment>
        <Audio>
            <Codec>mp3</Codec>
            <Samplerate></Samplerate>
            <Bitrate></Bitrate>
            <Channels></Channels>
        </Audio>
        <Video>
            <Codec>H.264</Codec>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Height></Height>
            <Fps>30</Fps>
        </Video>
        <Container>
            <Format>mp4</Format>
        </Container>
        <AudioMix>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</Au
            <MixMode>Once</MixMode>
            <Replace>true</Replace>
            <EffectConfig>
                <EnableStartFadein>true</EnableStartFadein>
                <StartFadeinTime>3</StartFadeinTime>
                <EnableEndFadeout>false</EnableEndFadeout>
                <EndFadeoutTime>0</EndFadeoutTime>
                <EnableBgmFade>true</EnableBgmFade>
                <BgmFadeTime>1.7</BgmFadeTime>
            </EffectConfig>
        </AudioMix>
    </ConcatTemplate>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Tag>Concat</Tag>
        <Name>TemplateName</Name>
        <ConcatTemplate>
            <ConcatFragment>
                <Mode>Start</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4
            </ConcatFragment>
            <ConcatFragment>
                <Mode>End</Mode>
                <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/end.mp4</
            </ConcatFragment>
            <Audio>
                <Codec>mp3</Codec>
                <Samplerate></Samplerate>
                <Bitrate></Bitrate>
                <Channels></Channels>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Height></Height>
                <Fps>30</Fps>
            </Video>
            <Container>
                <Format>mp4</Format>
            </Container>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
                <EffectConfig>
                    <EnableStartFadein>true</EnableStartFadein>
                    <StartFadeinTime>3</StartFadeinTime>
                    <EnableEndFadeout>false</EnableEndFadeout>
                    <EndFadeoutTime>0</EndFadeoutTime>
                    <EnableBgmFade>true</EnableBgmFade>
                    <BgmFadeTime>1.7</BgmFadeTime>
                </EffectConfig>
            </AudioMix>
        </ConcatTemplate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

更新极速高清转码模板

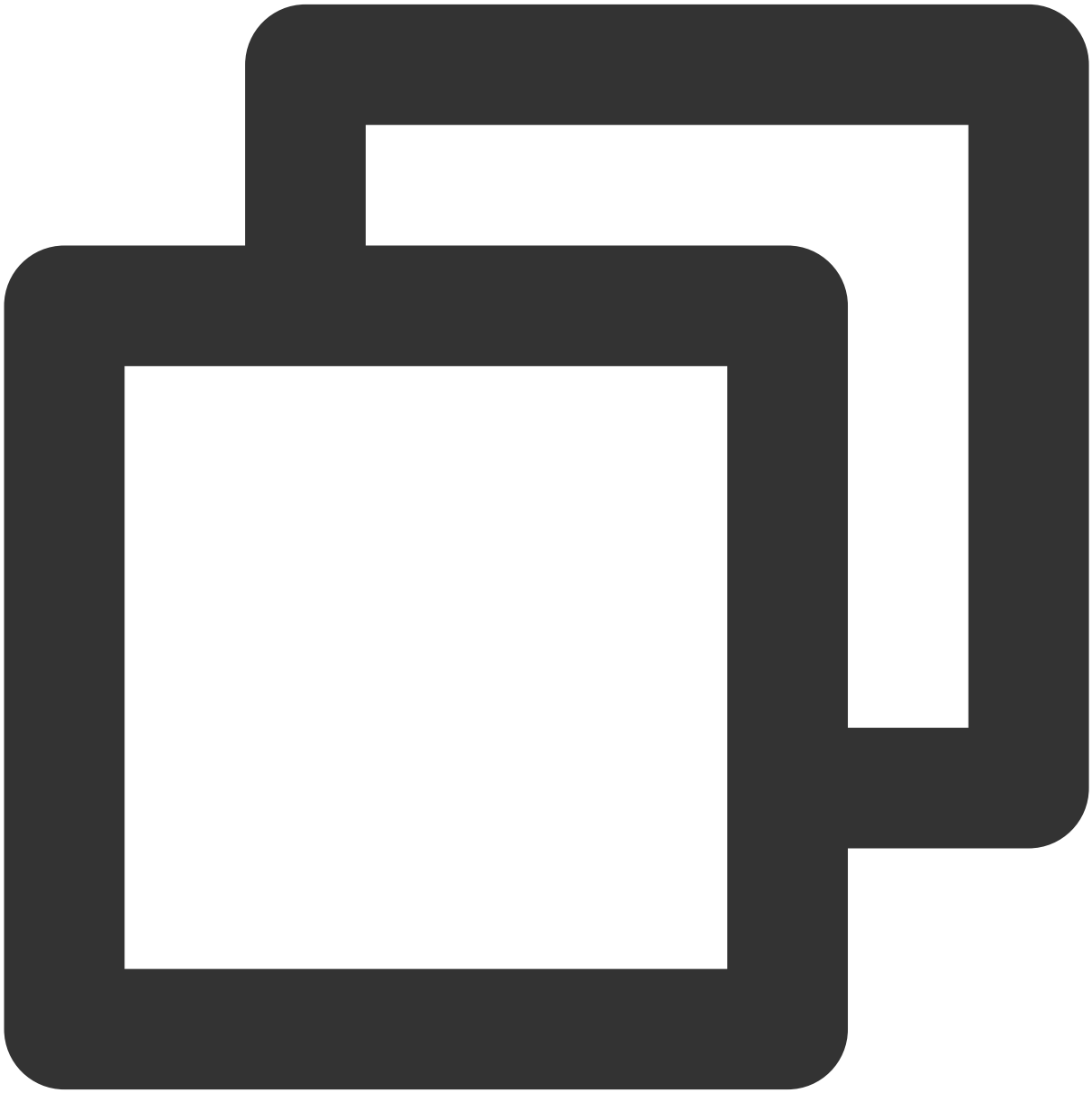
最近更新时间：2023-12-30 16:15:21

功能描述

更新极速高清转码模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

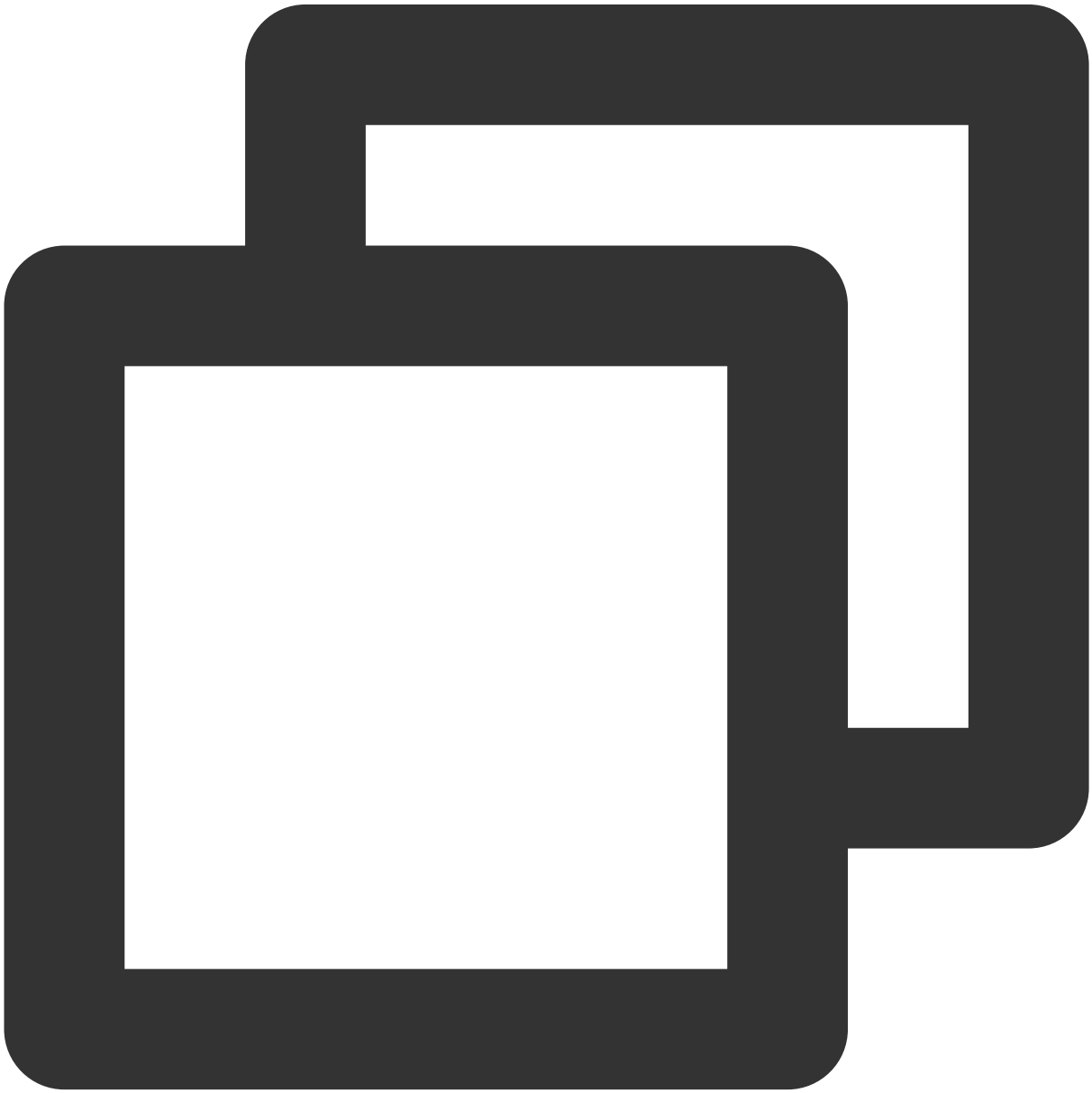
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>HighSpeedHd</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Profile>high</Profile>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Bufsize>0</Bufsize>
        <Maxrate>0</Maxrate>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建极速高清模板接口的 Request	Container	是

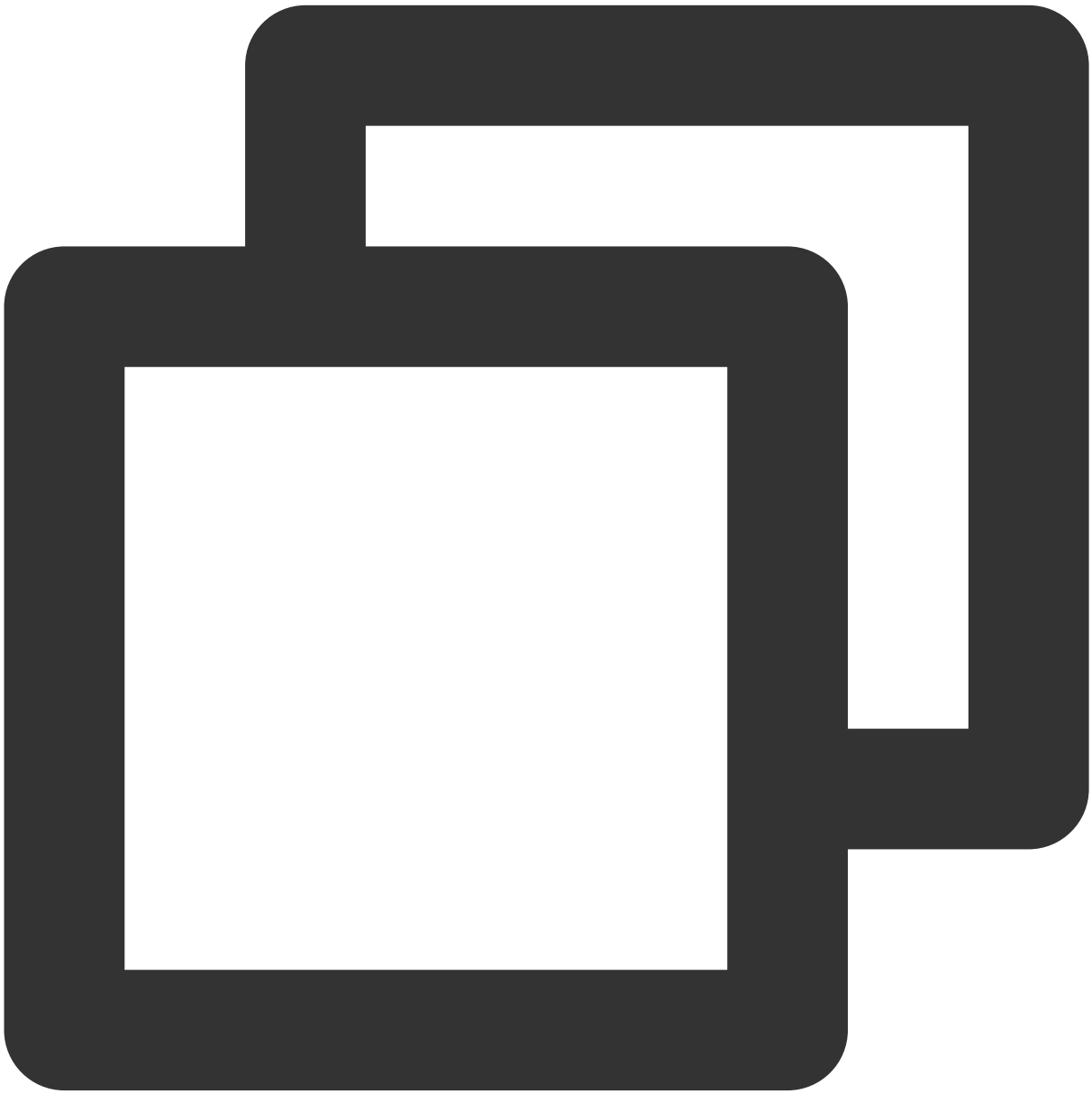
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>HighSpeedHd</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <HighSpeedHd>
            <Container>
                <Format>mp4</Format>
            </Container>
```

```
            <Video>
                <Codec>H.264</Codec>
                <Profile>high</Profile>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Bufsize>0</Bufsize>
                <Maxrate>0</Maxrate>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </HighSpeedHd>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

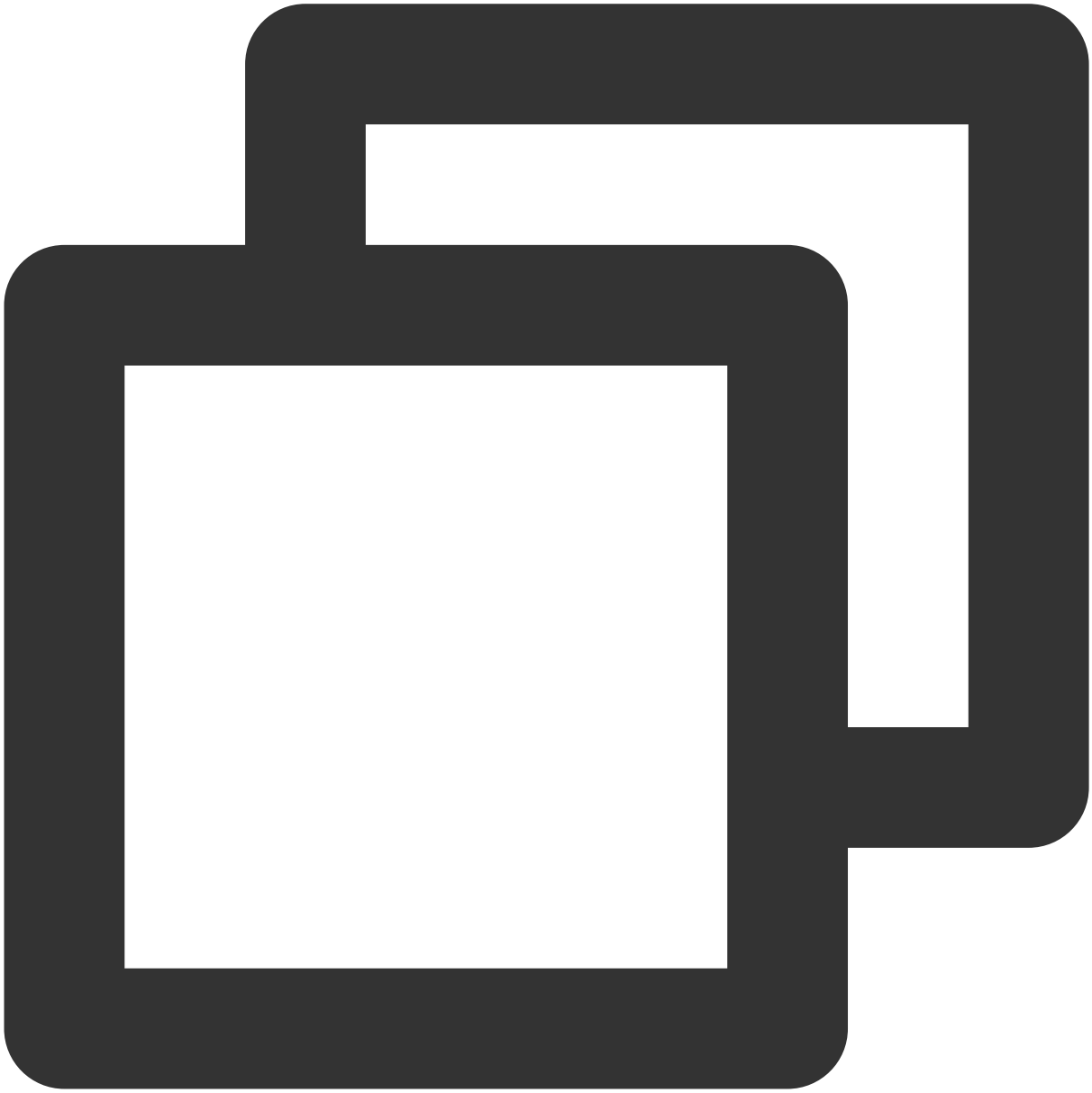
节点名称（关键字）	父节点	描述	类型
Response	无	同创建极速高清模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

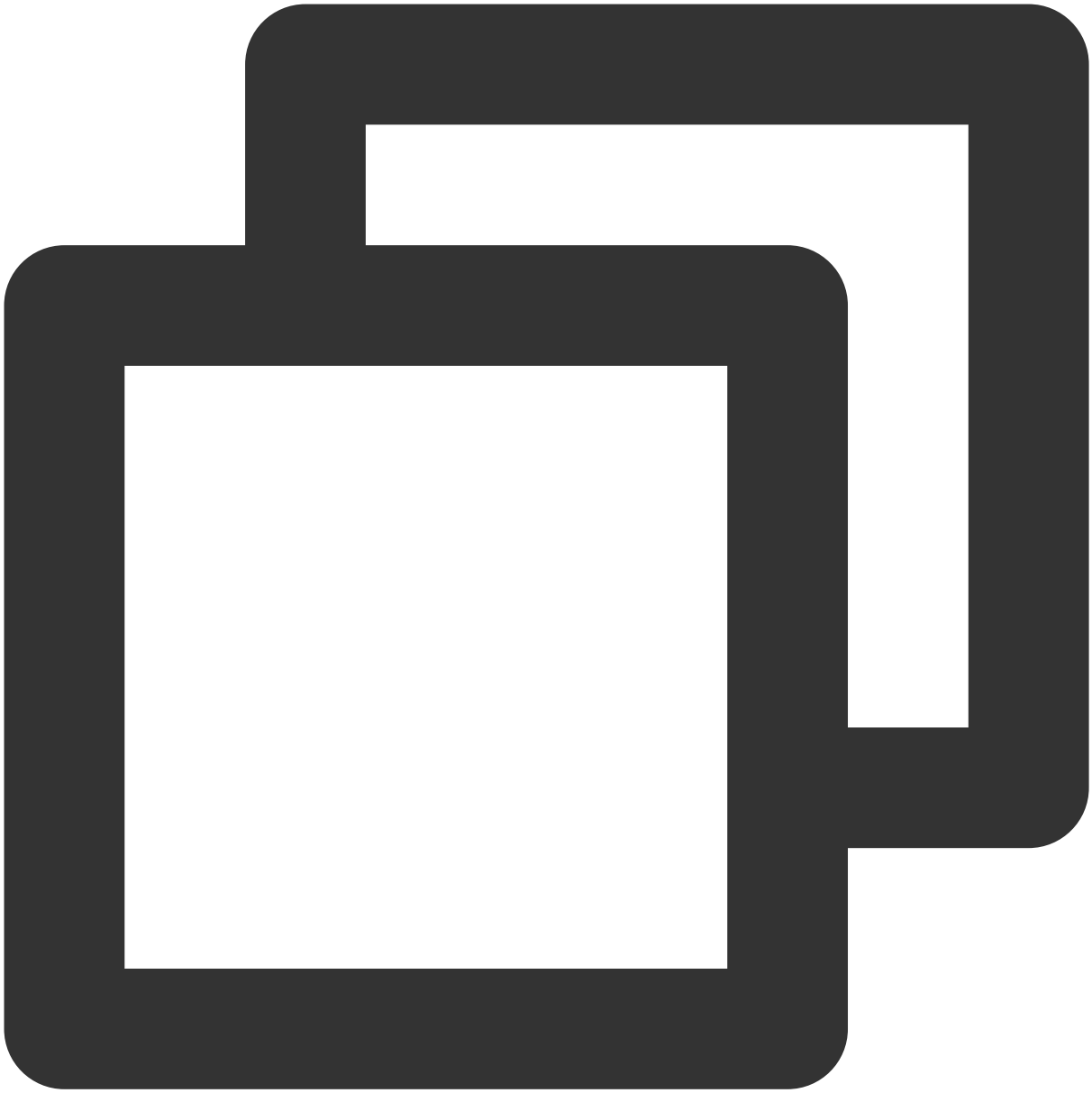
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>HighSpeedHd</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
```

```
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Profile>high</Profile>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Bufsize>0</Bufsize>
        <Maxrate>0</Maxrate>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>HighSpeedHd</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <HighSpeedHd>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Profile>high</Profile>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
                <Bufsize>0</Bufsize>
                <Maxrate>0</Maxrate>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <TransConfig>
                <IsCheckReso>false</IsCheckReso>
                <ResoAdjMethod>1</ResoAdjMethod>
            </TransConfig>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </HighSpeedHd>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

更新截图模板

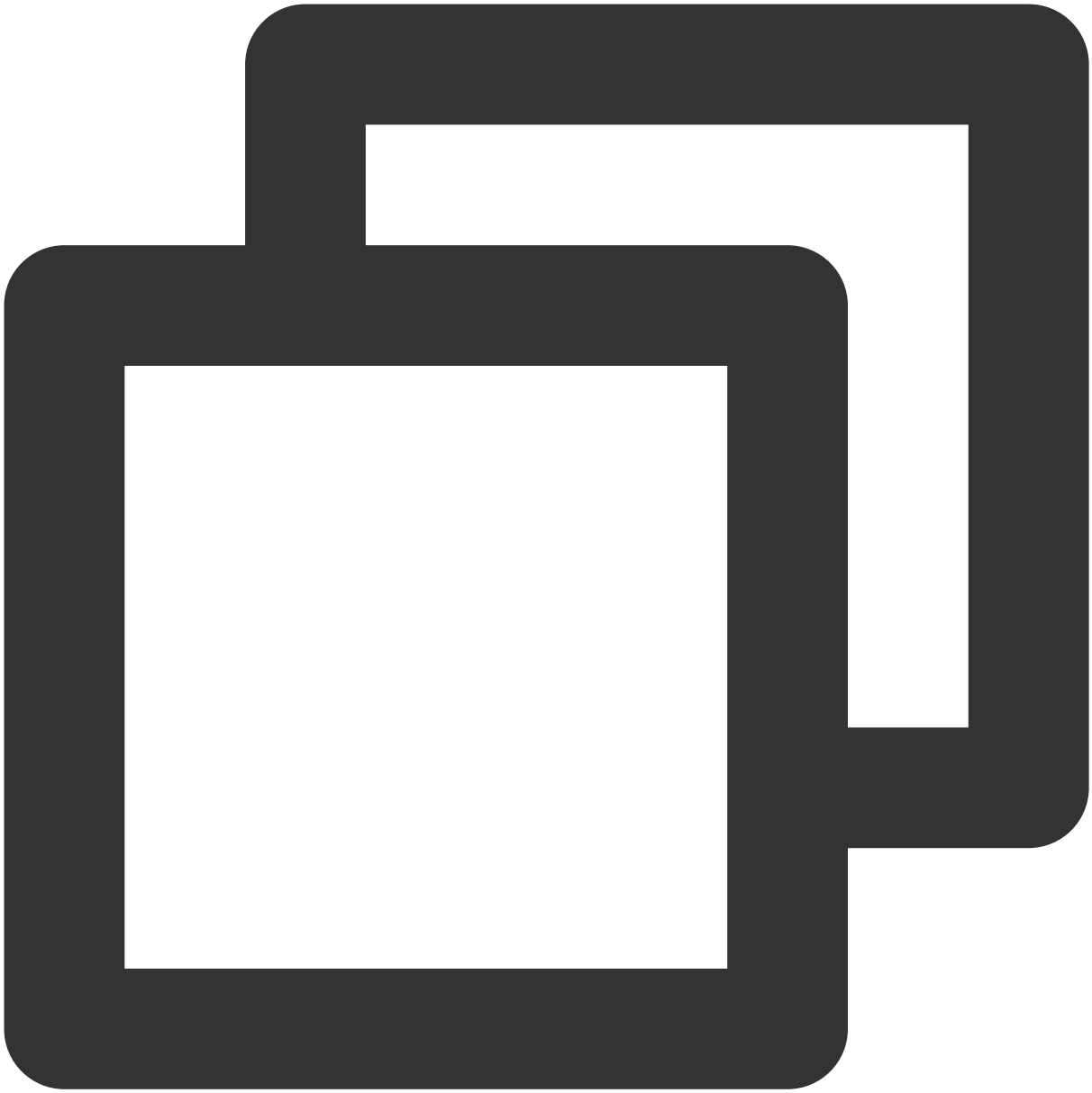
最近更新时间：2023-12-30 16:16:02

功能描述

更新截图模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

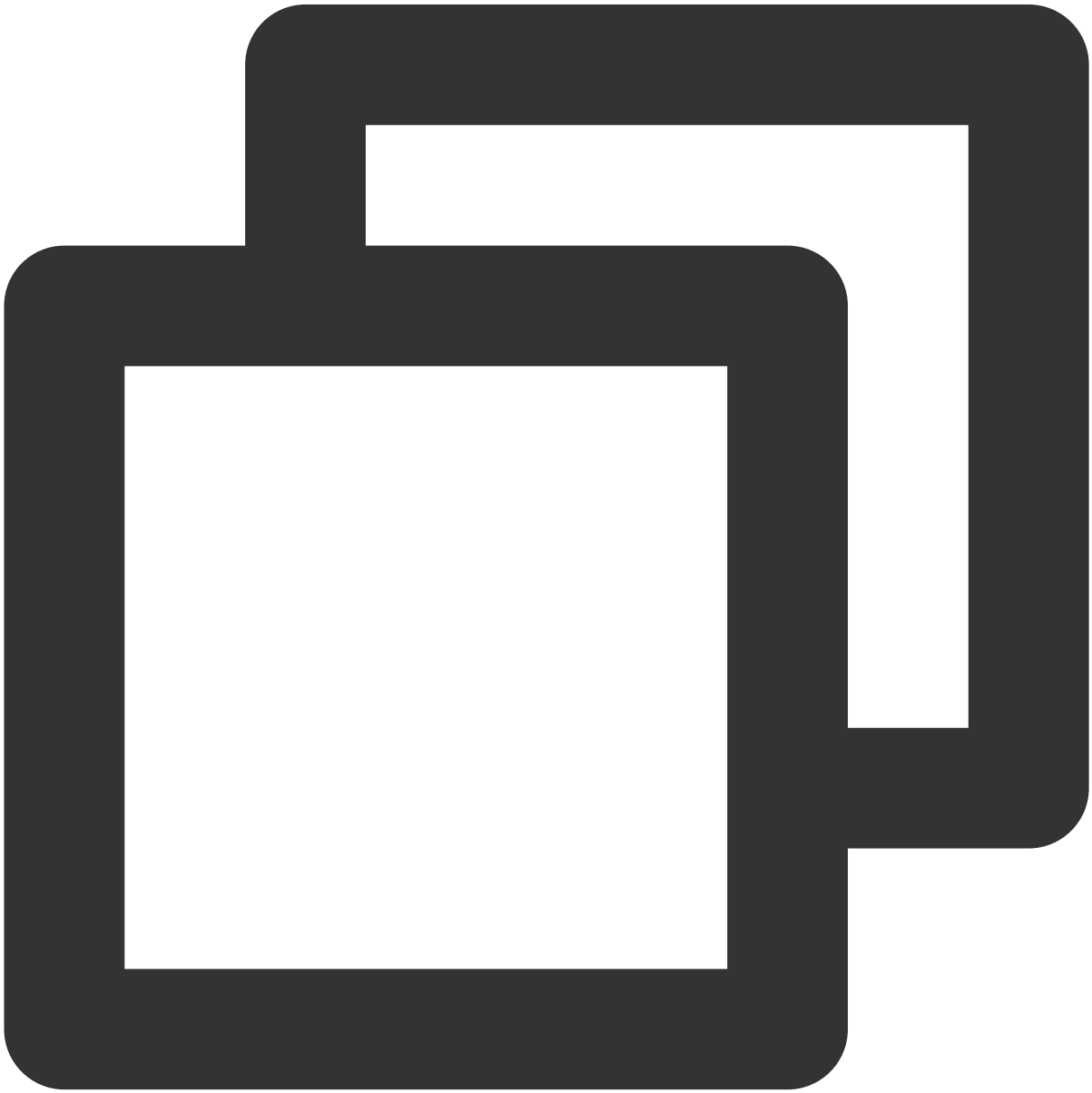
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
        <Width>1280</Width>
        <Height>960</Height>
        <Start>0</Start>
        <TimeInterval>0.5</TimeInterval>
        <Count>10</Count>
        <SnapshotOutMode>OnlySprite</SnapshotOutMode>
        <SpriteSnapshotConfig>
            <Color>Aquamarine</Color>
            <Columns>3</Columns>
            <Lines>3</Lines>
        </SpriteSnapshotConfig>
    </Snapshot>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建截图模板接口的 Request	Container	是

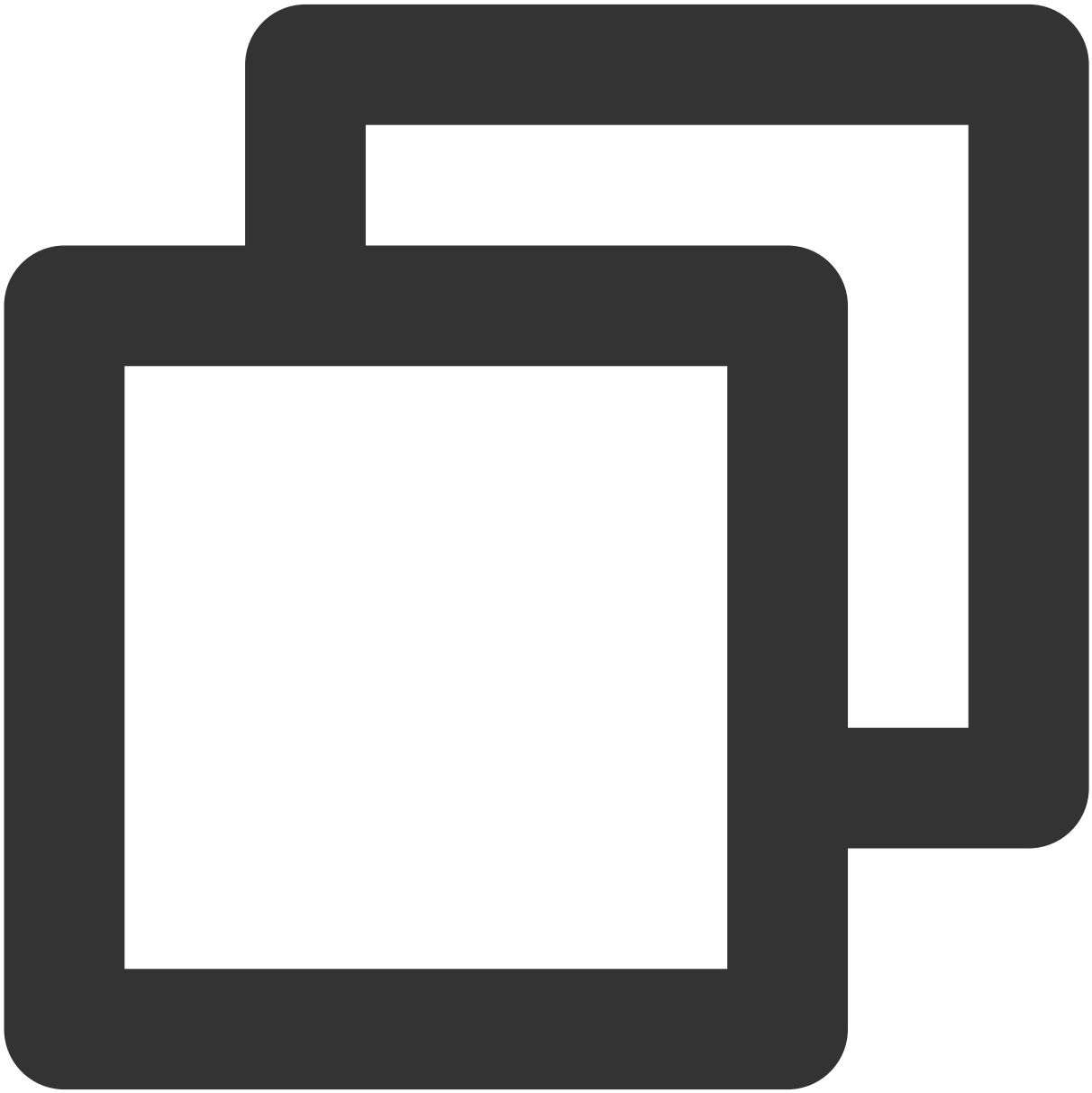
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Snapshot</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Snapshot>
            <Mode>Interval</Mode>
            <Width>1280</Width>
            <Height>960</Height>
            <Start>0</Start>
            <TimeInterval>0.5</TimeInterval>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Count>10</Count>
            <SnapshotOutMode>OnlySprite</SnapshotOutMode>
            <SpriteSnapshotConfig>
                <Color>Aquamarine</Color>
                <Columns>3</Columns>
                <Lines>3</Lines>
            </SpriteSnapshotConfig>
        </Snapshot>>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

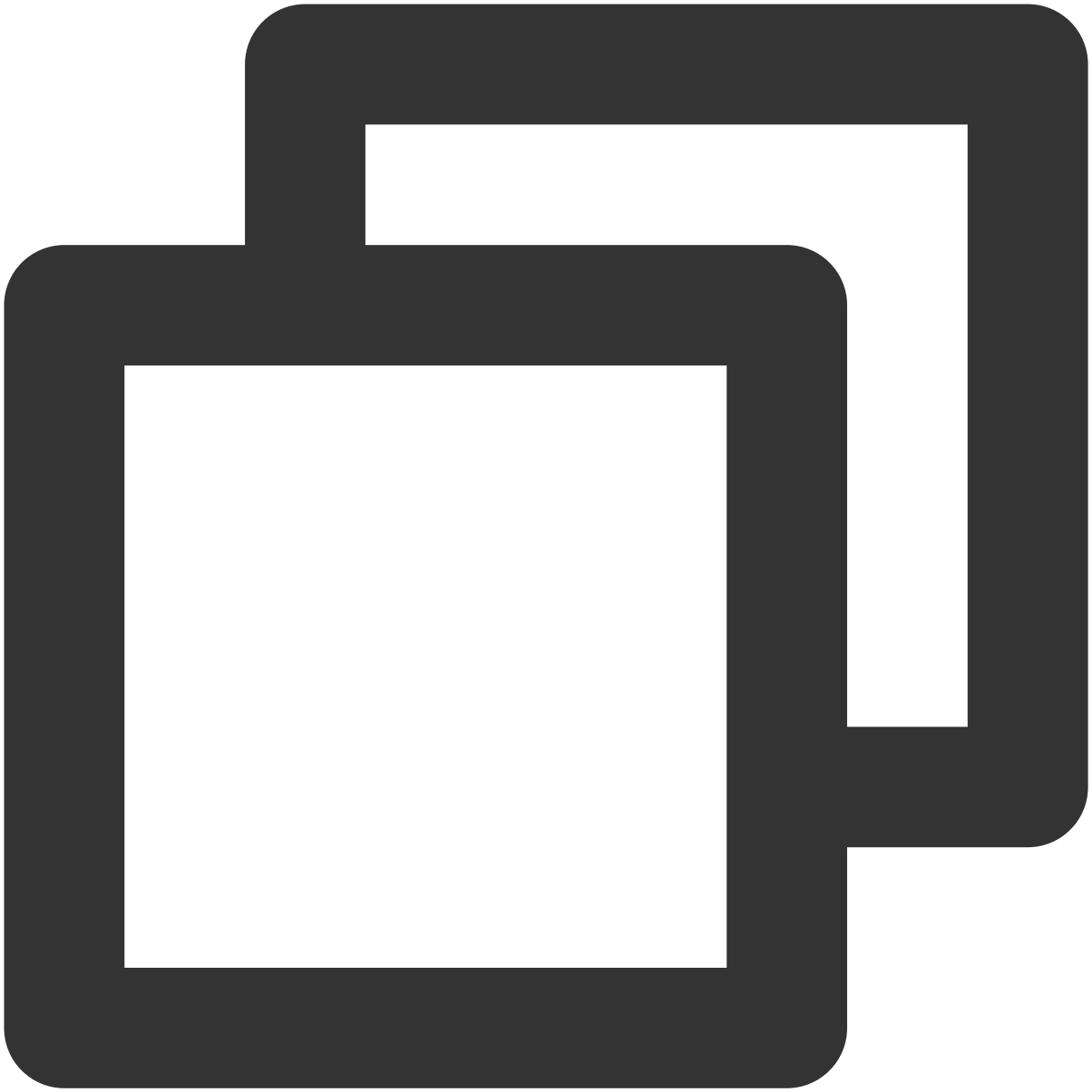
节点名称（关键字）	父节点	描述	类型
Response	无	同创建截图模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

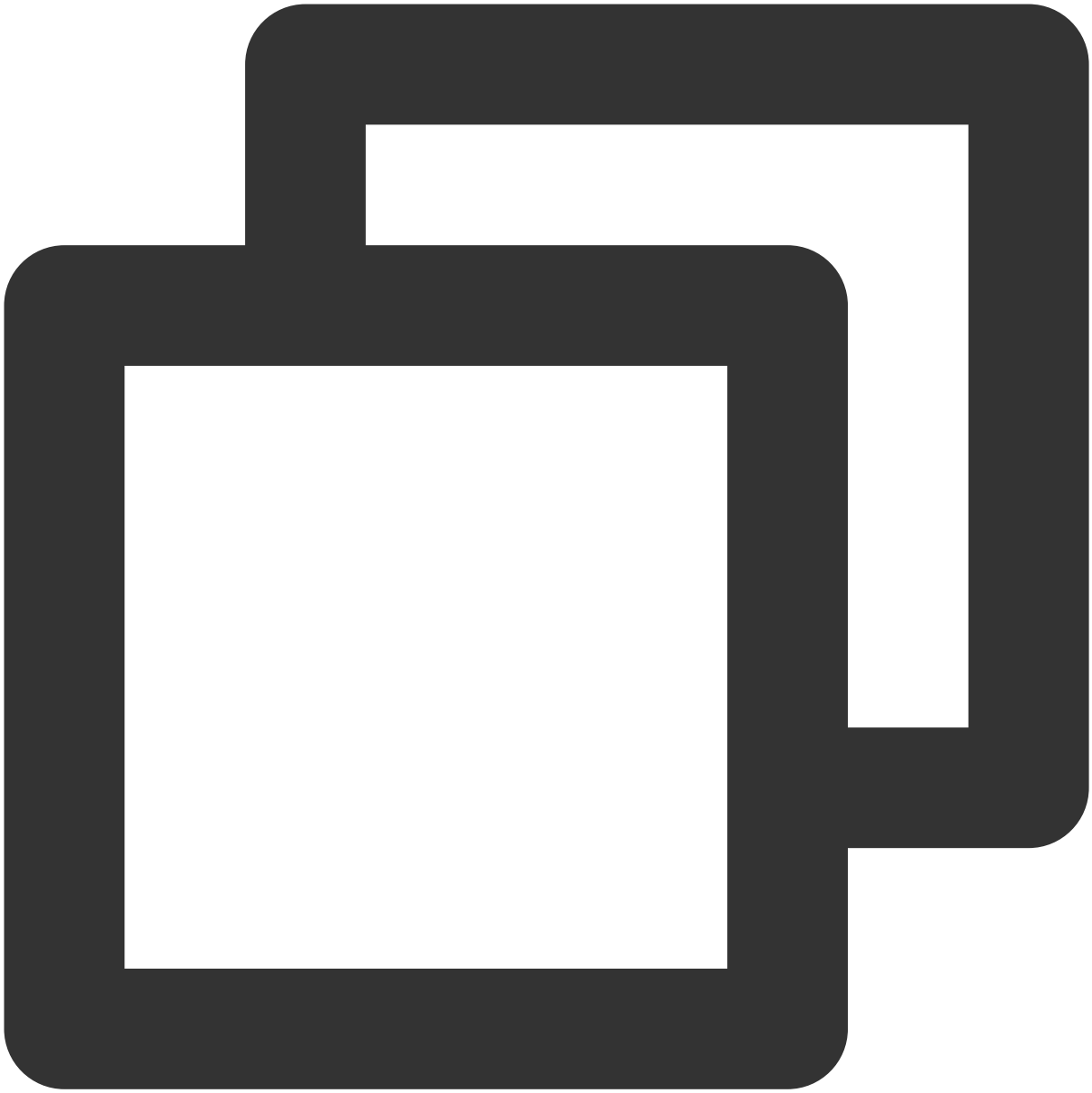
请求



```
PUT /template/t1f16e1dfbdc994105b31292d45710642a HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
```

```
        <Width>1280</Width>
        <Height>960</Height>
        <Start>0</Start>
        <TimeInterval>0.5</TimeInterval>
        <Count>10</Count>
        <SnapshotOutMode>OnlySprite</SnapshotOutMode>
        <SpriteSnapshotConfig>
            <Color>Aquamarine</Color>
            <Columns>3</Columns>
            <Lines>3</Lines>
        </SpriteSnapshotConfig>
    </Snapshot>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <Template>
        <Tag>Snapshot</Tag>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Name>TemplateName</Name>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Snapshot>
            <Mode>Interval</Mode>
            <Width>1280</Width>
            <Height>960</Height>
            <Start>0</Start>
            <TimeInterval>0.5</TimeInterval>
            <Count>10</Count>
            <SnapshotOutMode>OnlySprite</SnapshotOutMode>
            <SpriteSnapshotConfig>
                <Color>Aquamarine</Color>
                <Columns>3</Columns>
                <Lines>3</Lines>
            </SpriteSnapshotConfig>
        </Snapshot>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新超分辨率模板

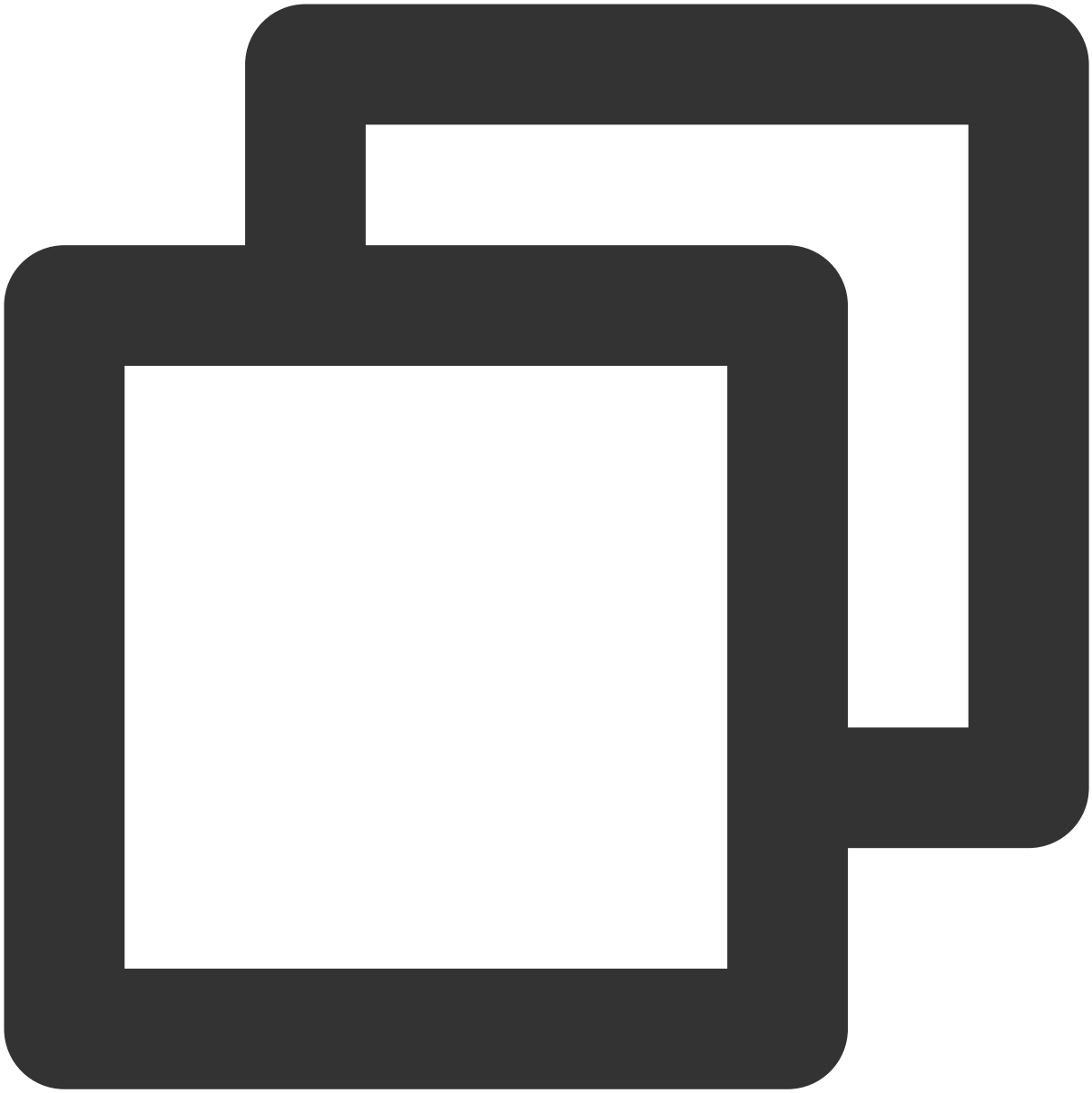
最近更新时间：2023-12-30 16:15:55

功能描述

更新超分辨率模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

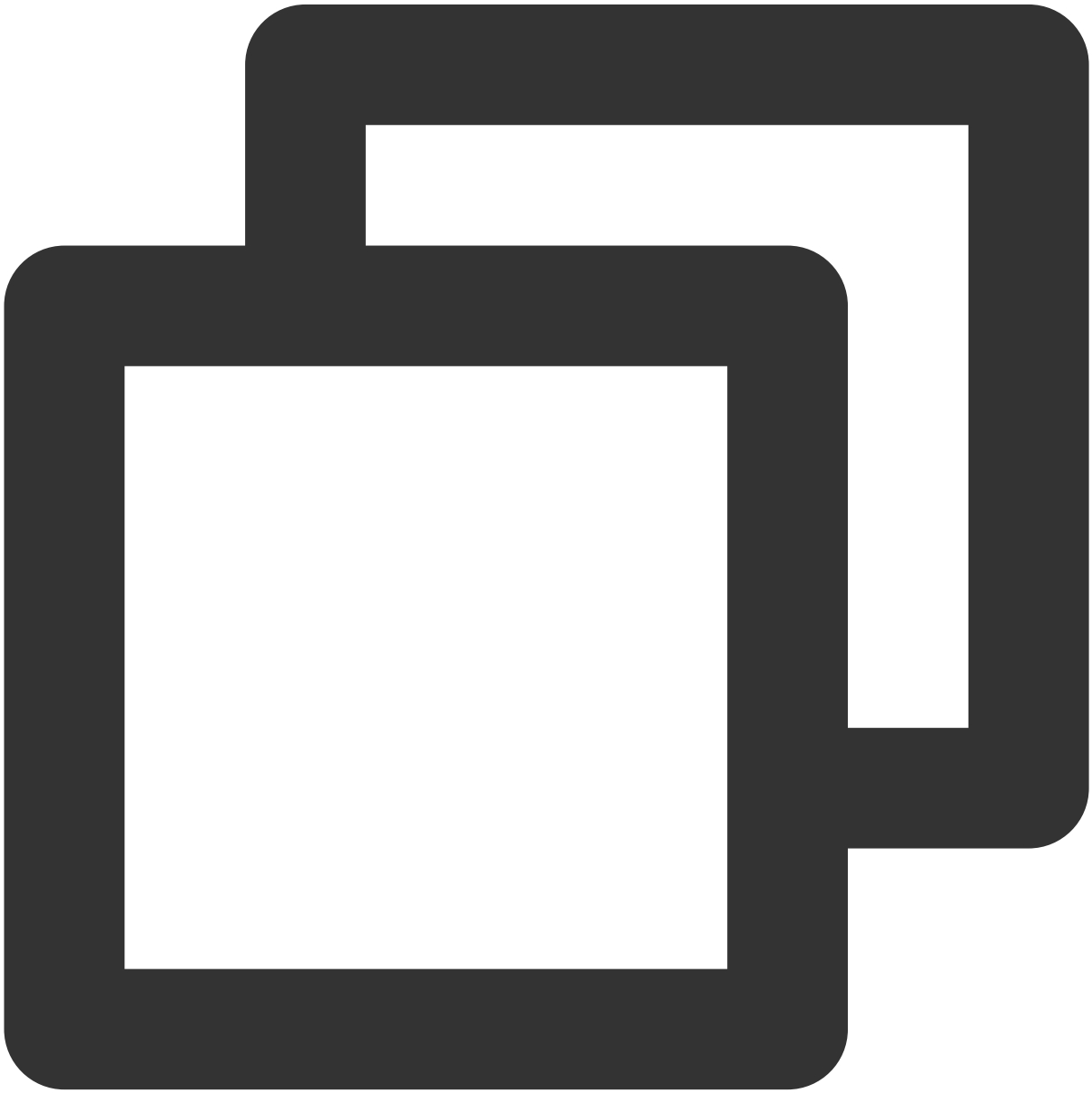
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>SuperResolution</Tag>
    <Name>TemplateName</Name>
    <Resolution>sdtohd</Resolution>
    <EnableScaleUp>true</EnableScaleUp>
    <Version>Enhance</Version>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建超分辨率模板接口的 Request	Container	是

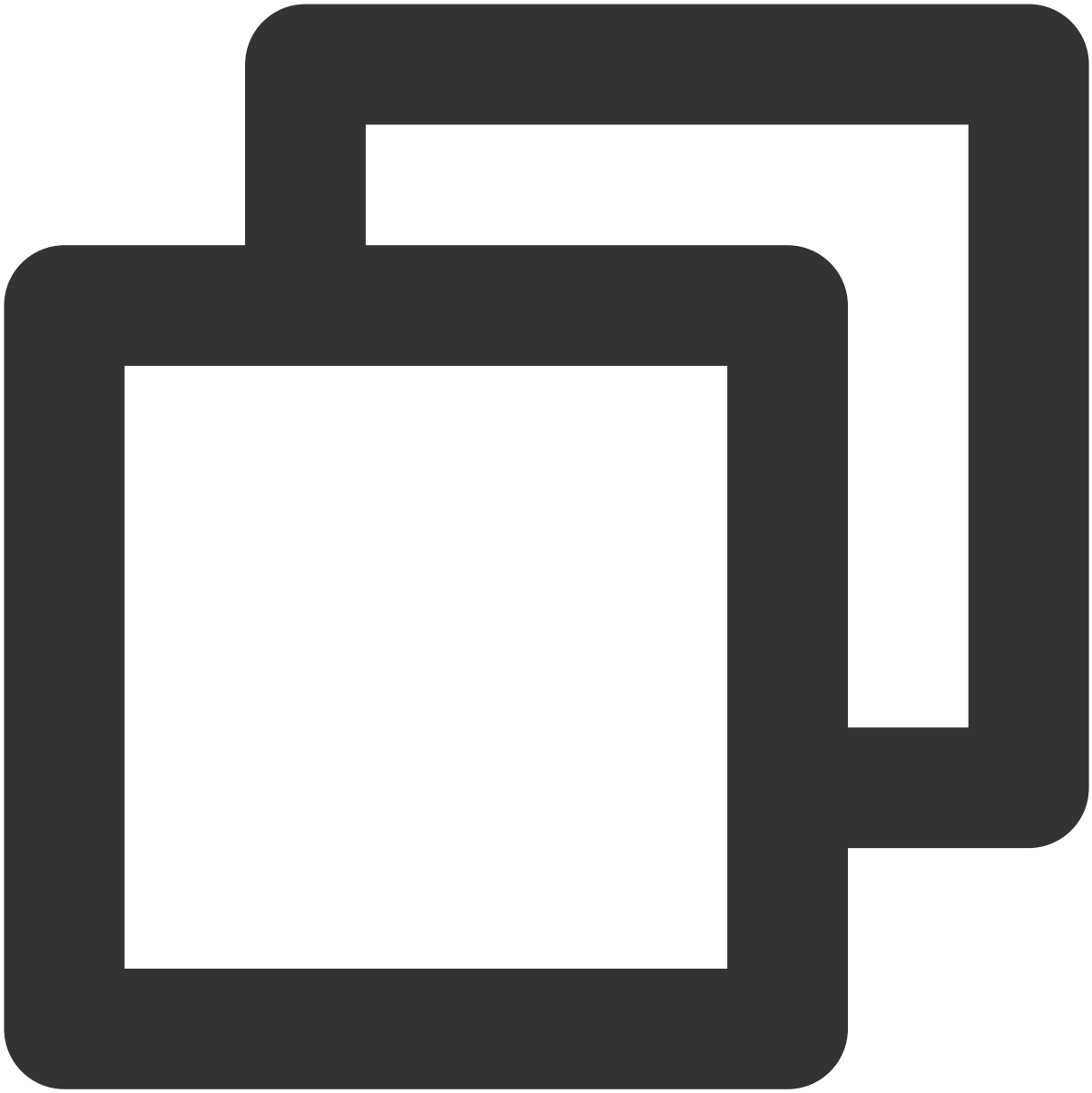
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>SuperResolution</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <SuperResolution>
            <Resolution>sdtohd</Resolution>
            <EnableScaleUp>true</EnableScaleUp>
            <Version>Enhance</Version>
        </SuperResolution>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
```

```
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

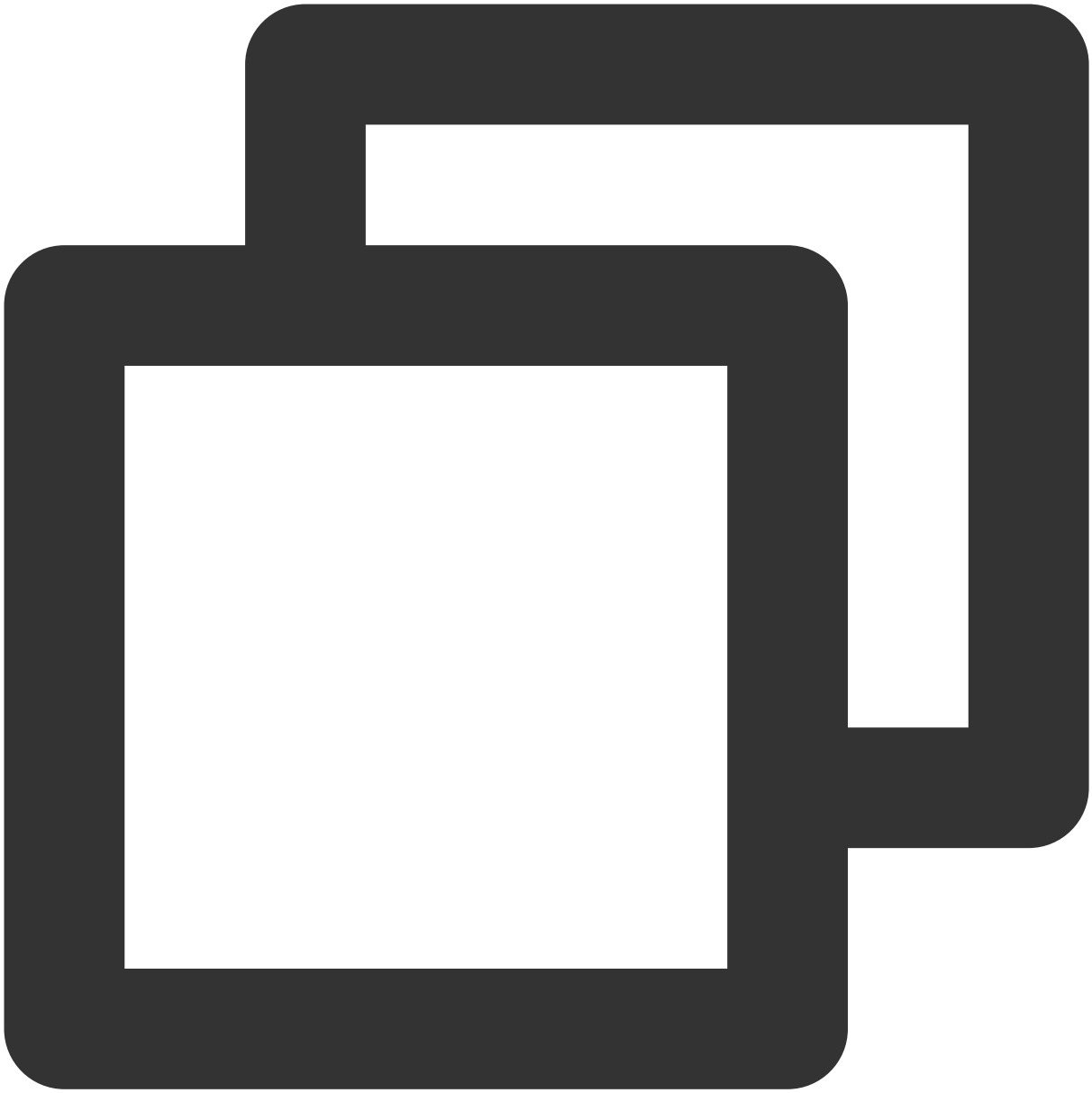
节点名称（关键字）	父节点	描述	类型
Response	无	同创建超分辨率模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

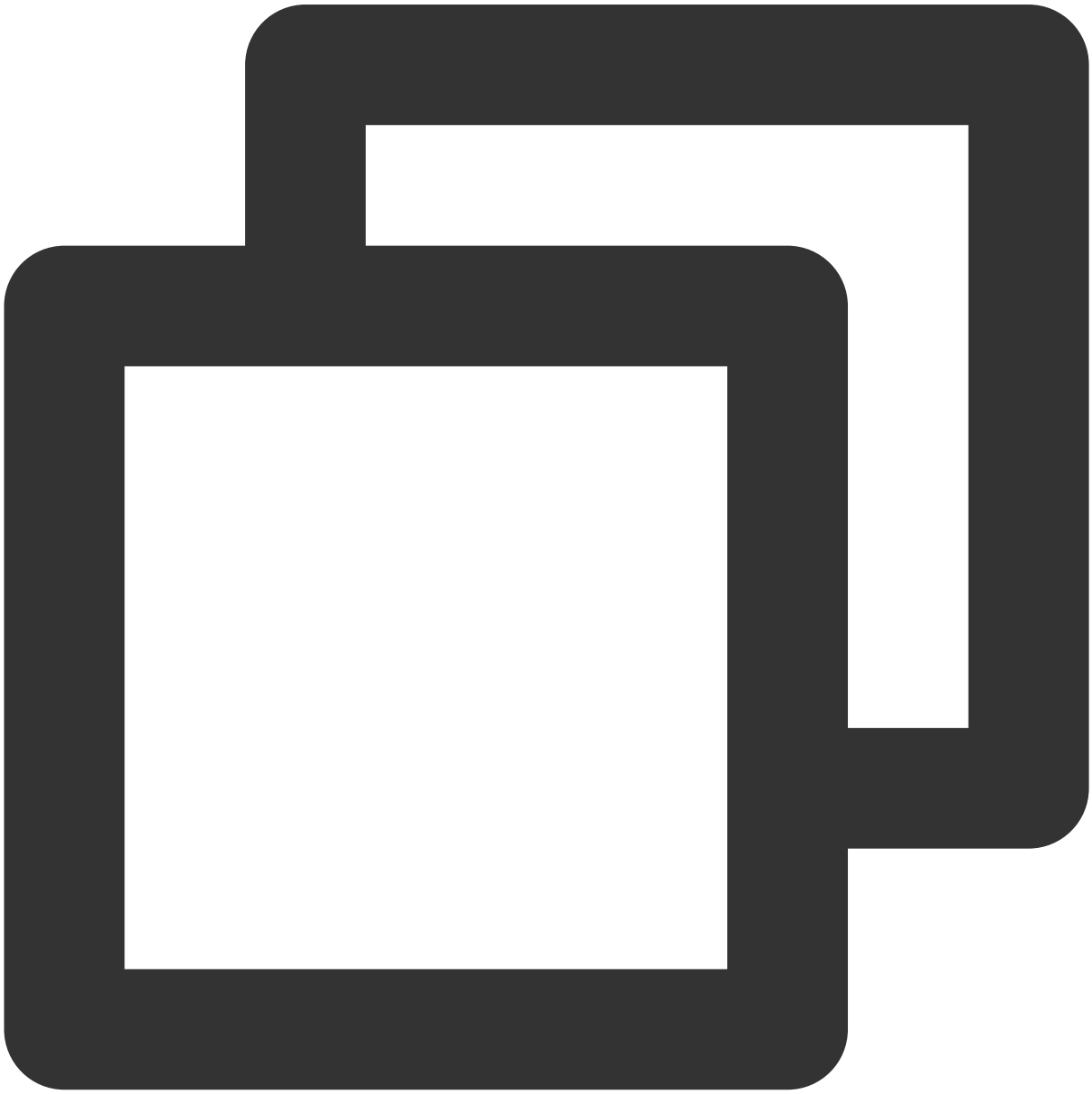
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>SuperResolution</Tag>
    <Name>TemplateName</Name>
    <Resolution>sdtohd</Resolution>
    <EnableScaleUp>true</EnableScaleUp>
```

```
    <Version>Enhance</Version>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
```

```
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>SuperResolution</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <SuperResolution>
            <Resolution>sdtohd</Resolution>
            <EnableScaleUp>true</EnableScaleUp>
            <Version>Enhance</Version>
        </SuperResolution>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新音视频转码模板

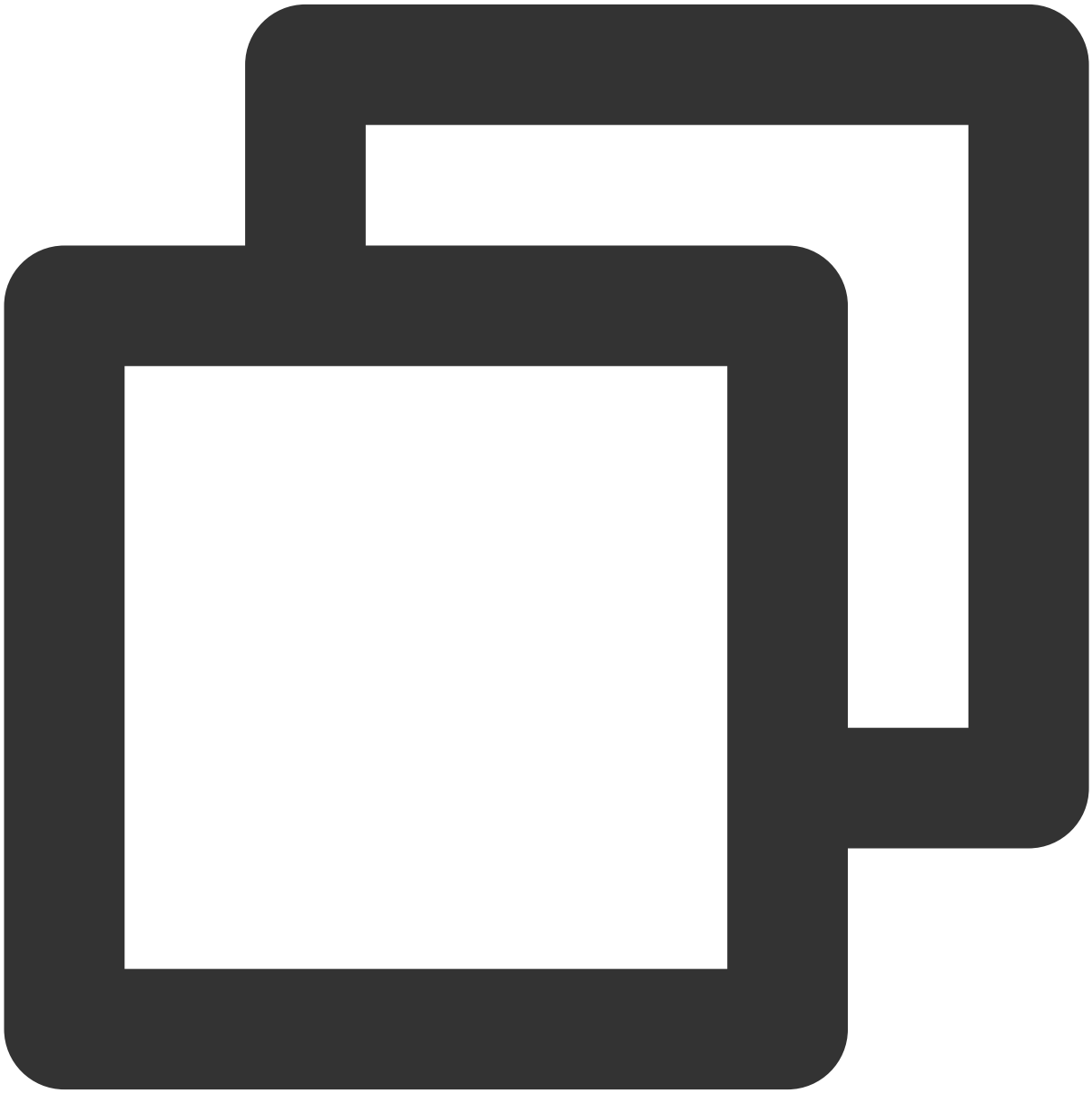
最近更新时间：2023-12-30 16:16:38

功能描述

更新转码模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

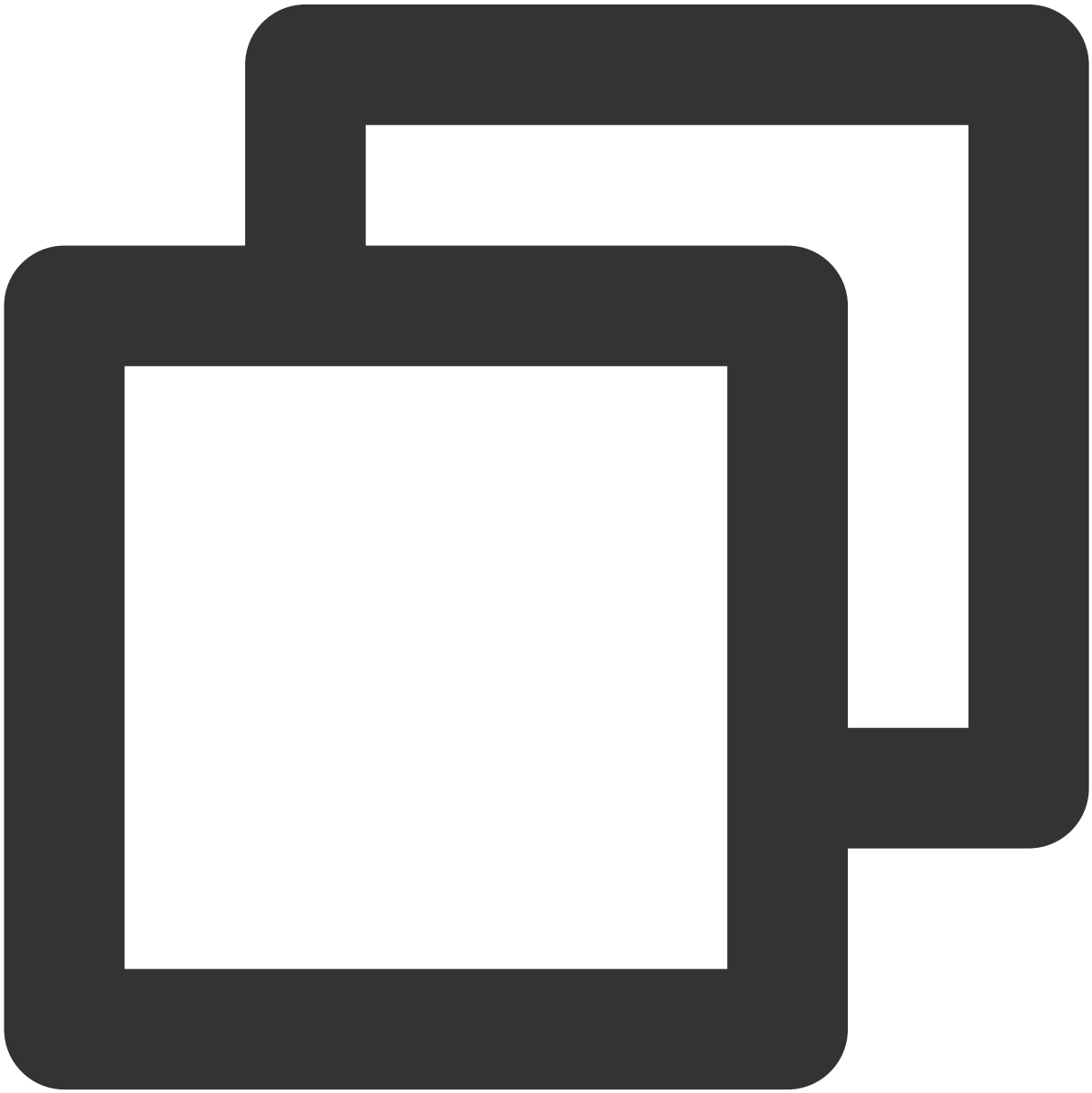
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Transcode</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Profile>high</Profile>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Preset>medium</Preset>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建转码模板接口的 Request	Container	是

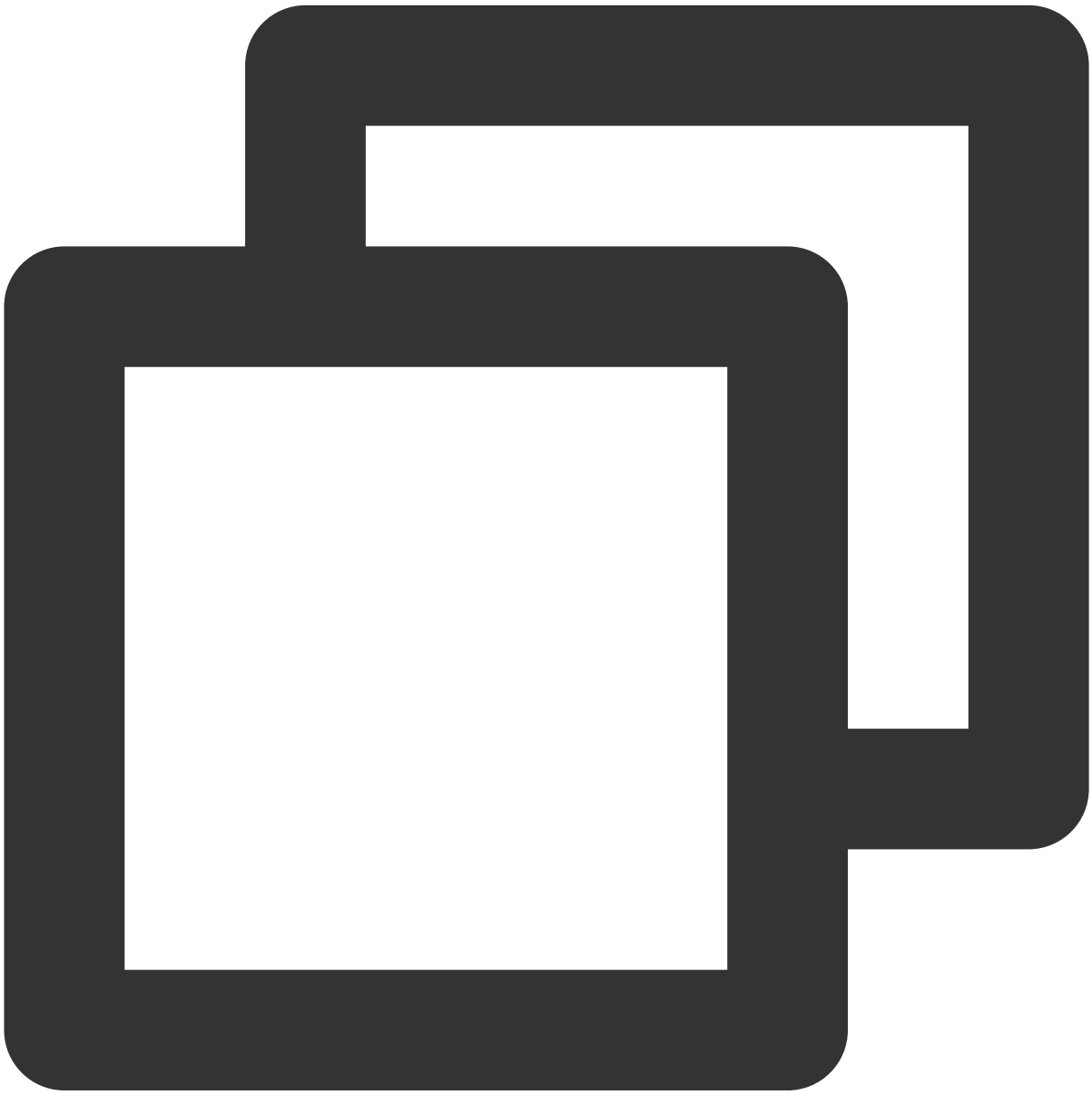
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Transcode</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <Container>
            <Format>mp4</Format>
        </Container>
        <Video>
```

```
            <Codec>H.264</Codec>
            <Profile>high</Profile>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Fps>30</Fps>
            <Preset>medium</Preset>
        </Video>
        <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
        </Audio>
        <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
        </TransConfig>
        <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
        </TimeInterval>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

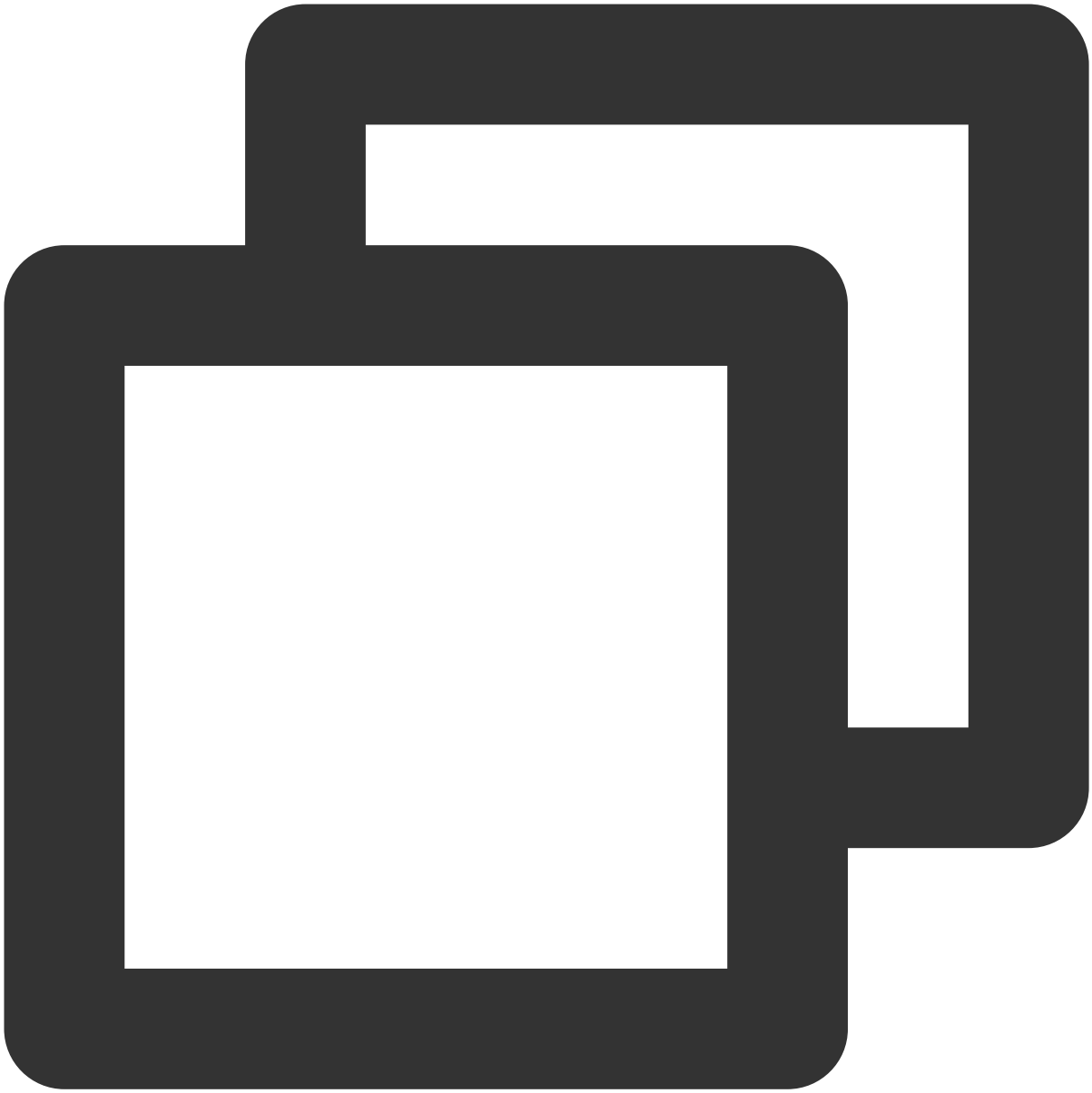
节点名称（关键字）	父节点	描述	类型
Response	无	同创建转码模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

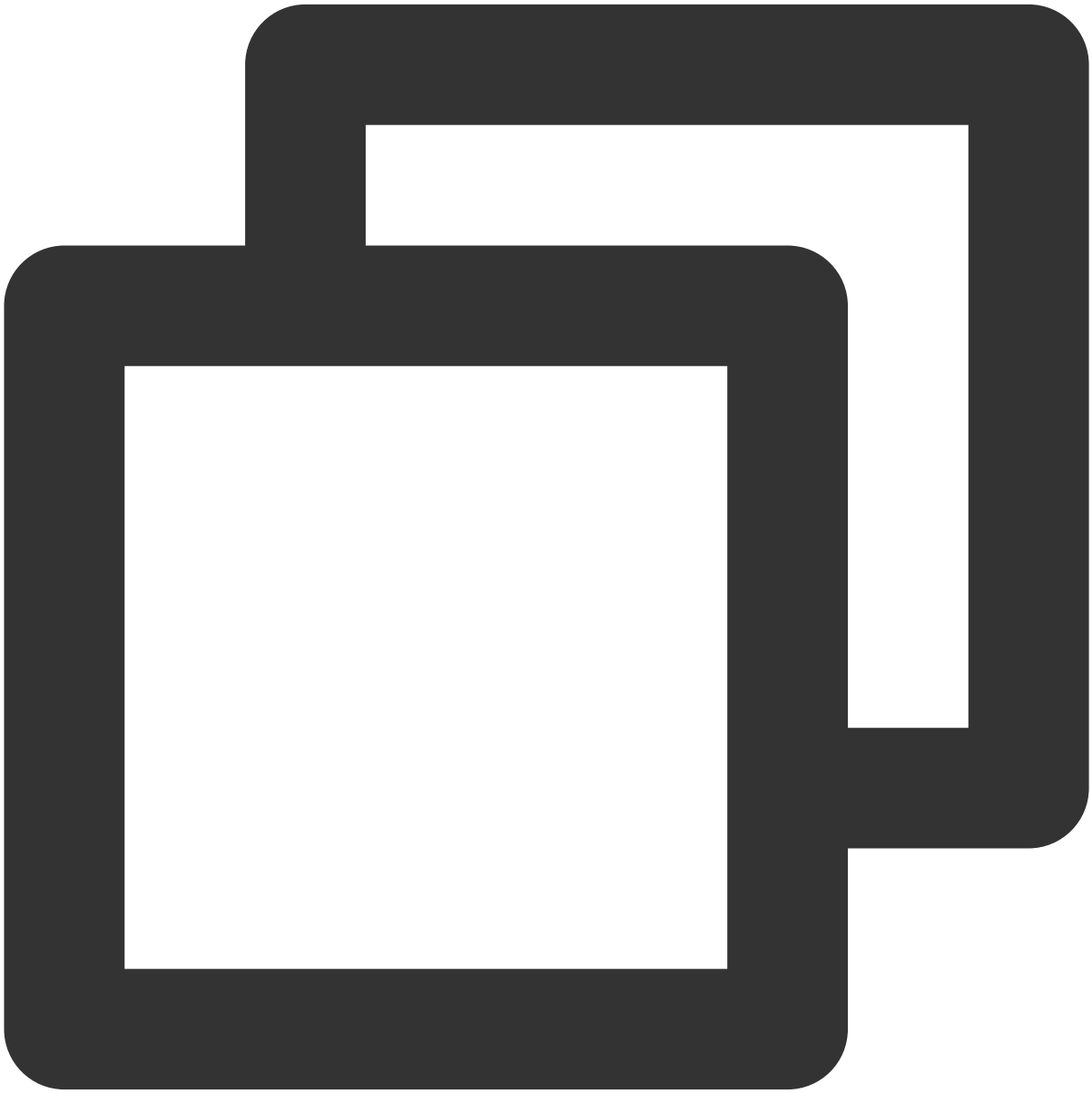
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Transcode</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
```

```
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Profile>high</Profile>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
        <Preset>medium</Preset>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****



<Response>
```

```
    <Template>
        <Tag>Transcode</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <Container>
            <Format>mp4</Format>
        </Container>
        <Video>
            <Codec>H.264</Codec>
            <Profile>high</Profile>
            <Bitrate>1000</Bitrate>
            <Width>1280</Width>
            <Fps>30</Fps>
            <Preset>medium</Preset>
        </Video>
        <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
        </Audio>
        <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
        </TransConfig>
        <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
        </TimeInterval>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新转码 pro 模板

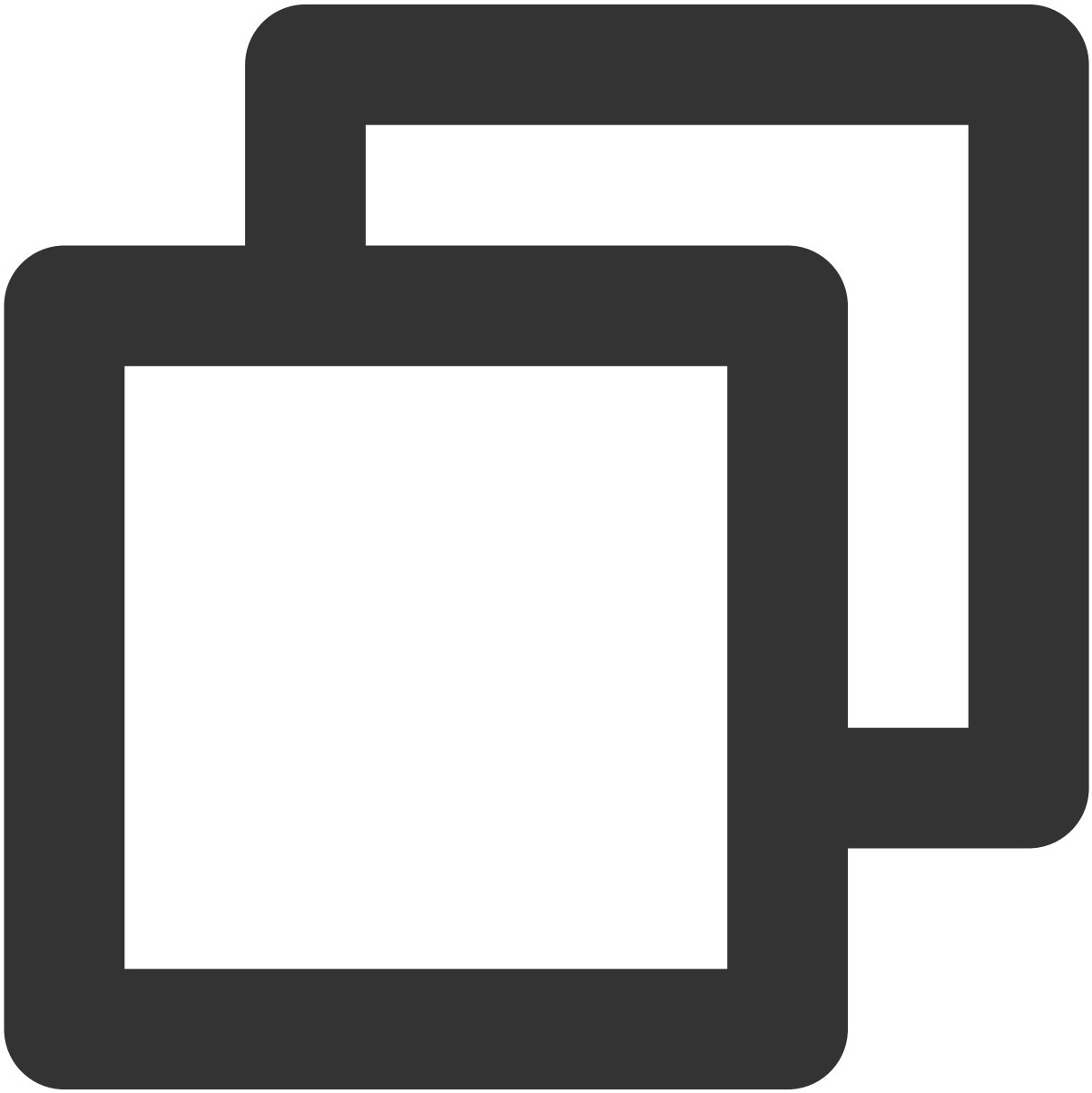
最近更新时间：2023-12-30 16:16:31

功能描述

更新转码 pro 模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

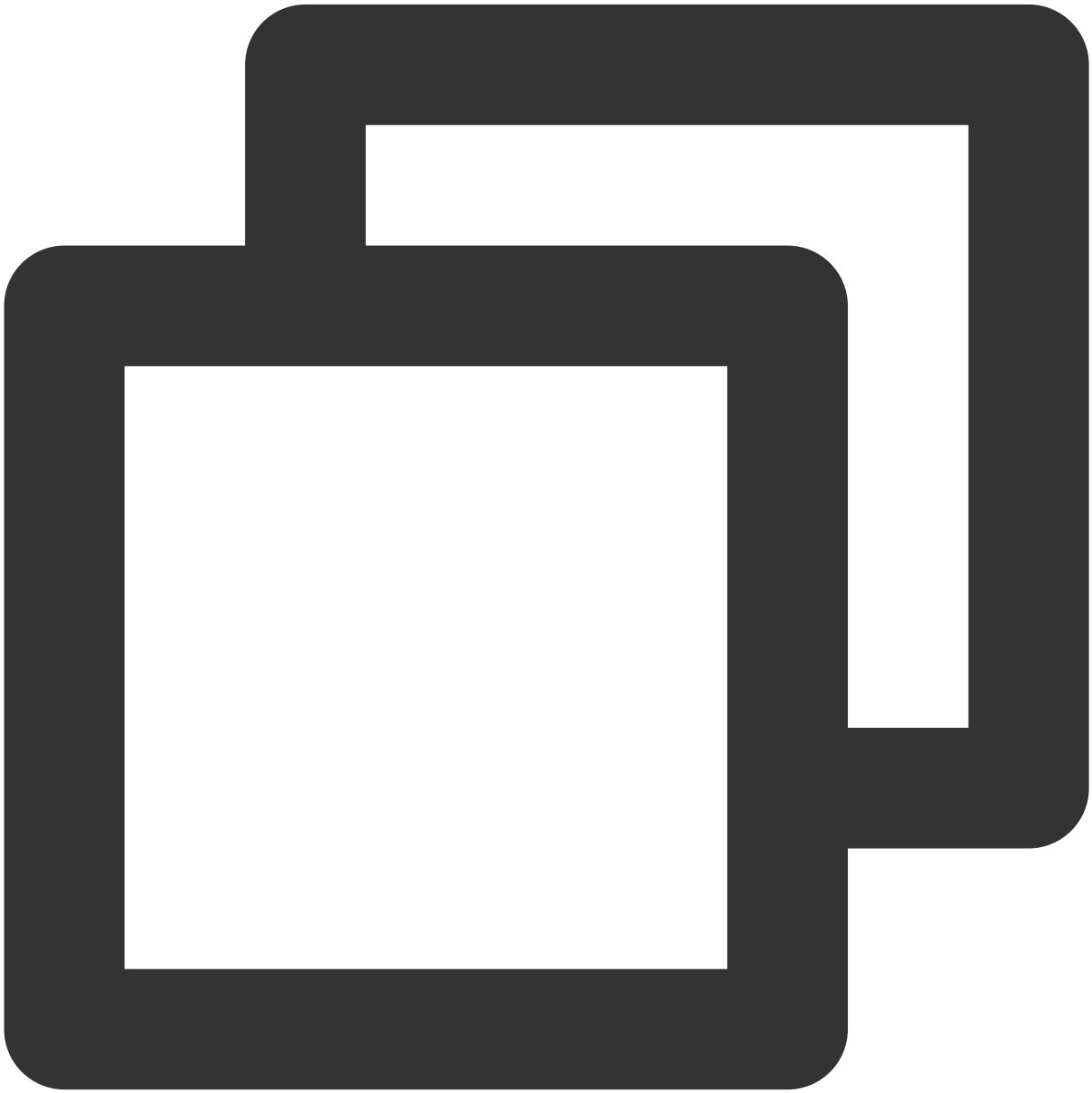
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>TranscodePro</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mxf</Format>
    </Container>
    <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
    </Video>
    <Audio>
        <Codec>pcm_s24le</Codec>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建转码 Pro 模板接口的 Request	Container	是

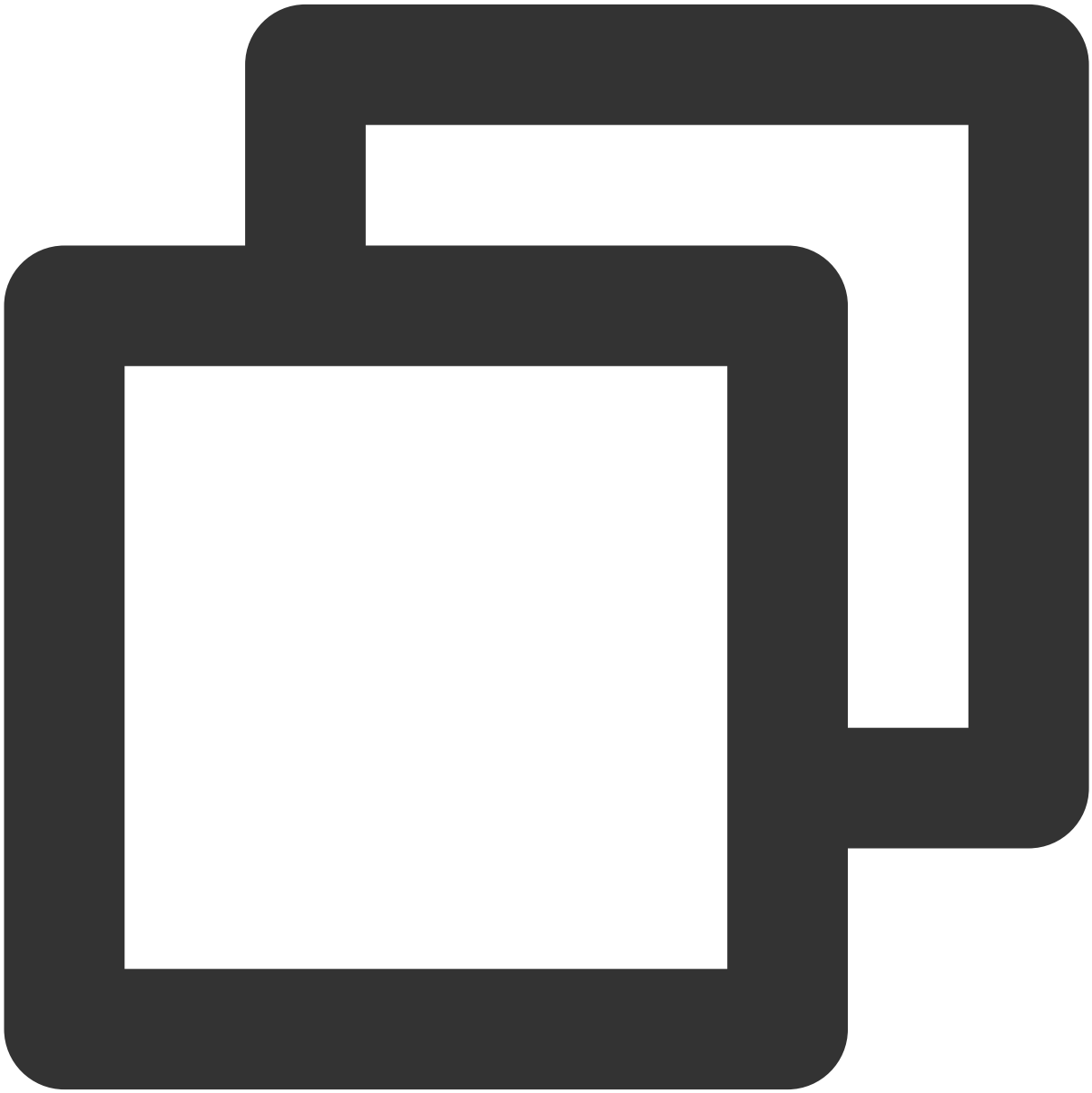
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>TranscodePro</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <Container>
            <Format>mxf</Format>
        </Container>
        <Video>
            <Codec>xavc</Codec>
            <Profile>XAVC-HD_422_10bit</Profile>
```

```
            <Width>1920</Width>
            <Height>1080</Height>
            <Interlaced>true</Interlaced>
            <Fps>30000/1001</Fps>
            <Bitrate>50000</Bitrate>
        </Video>
        <Audio>
            <Codec>pcm_s24le</Codec>
        </Audio>
        <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
        </TransConfig>
        <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
        </TimeInterval>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

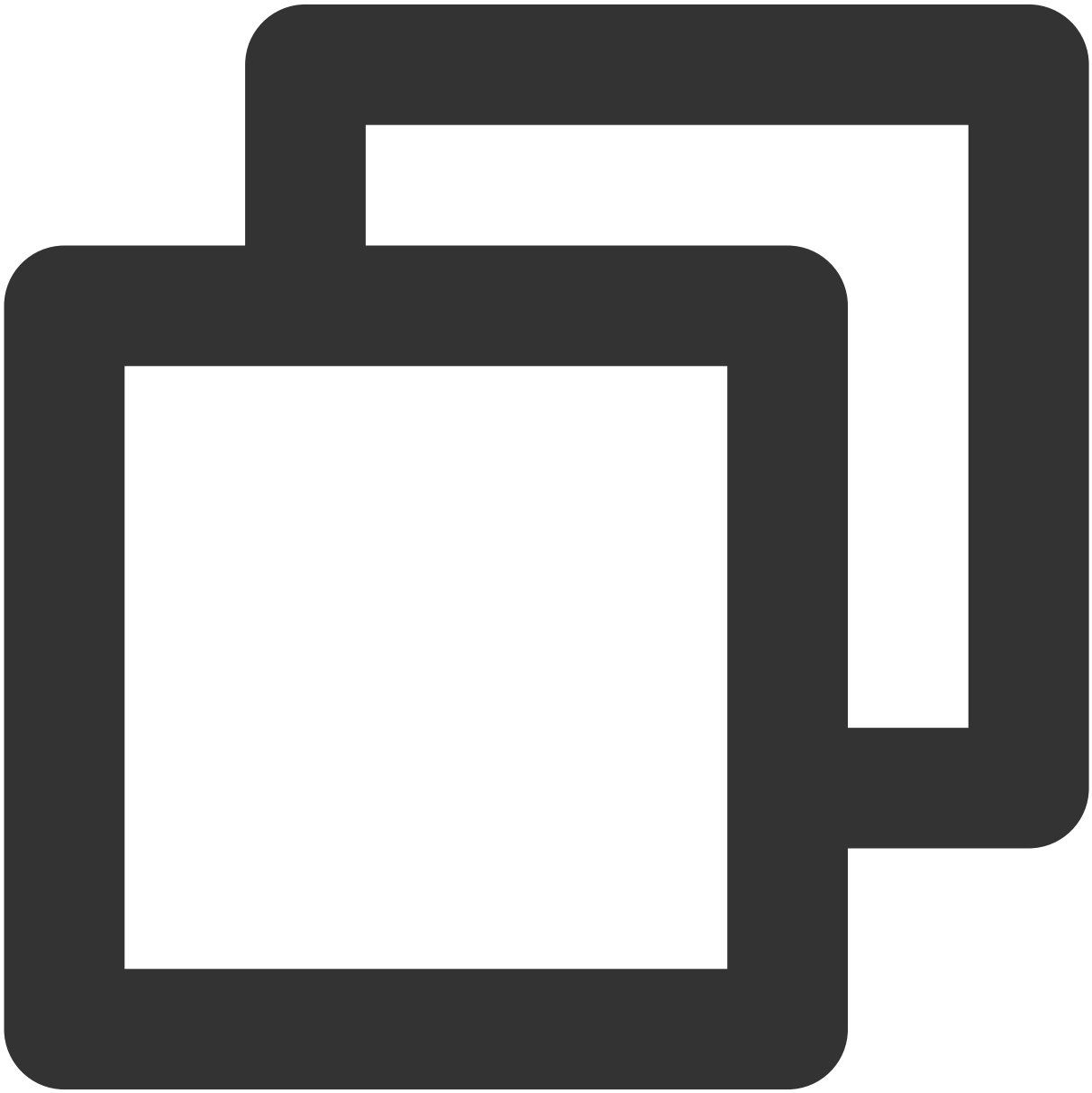
节点名称（关键字）	父节点	描述	类型
Response	无	同创建转码 Pro 模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

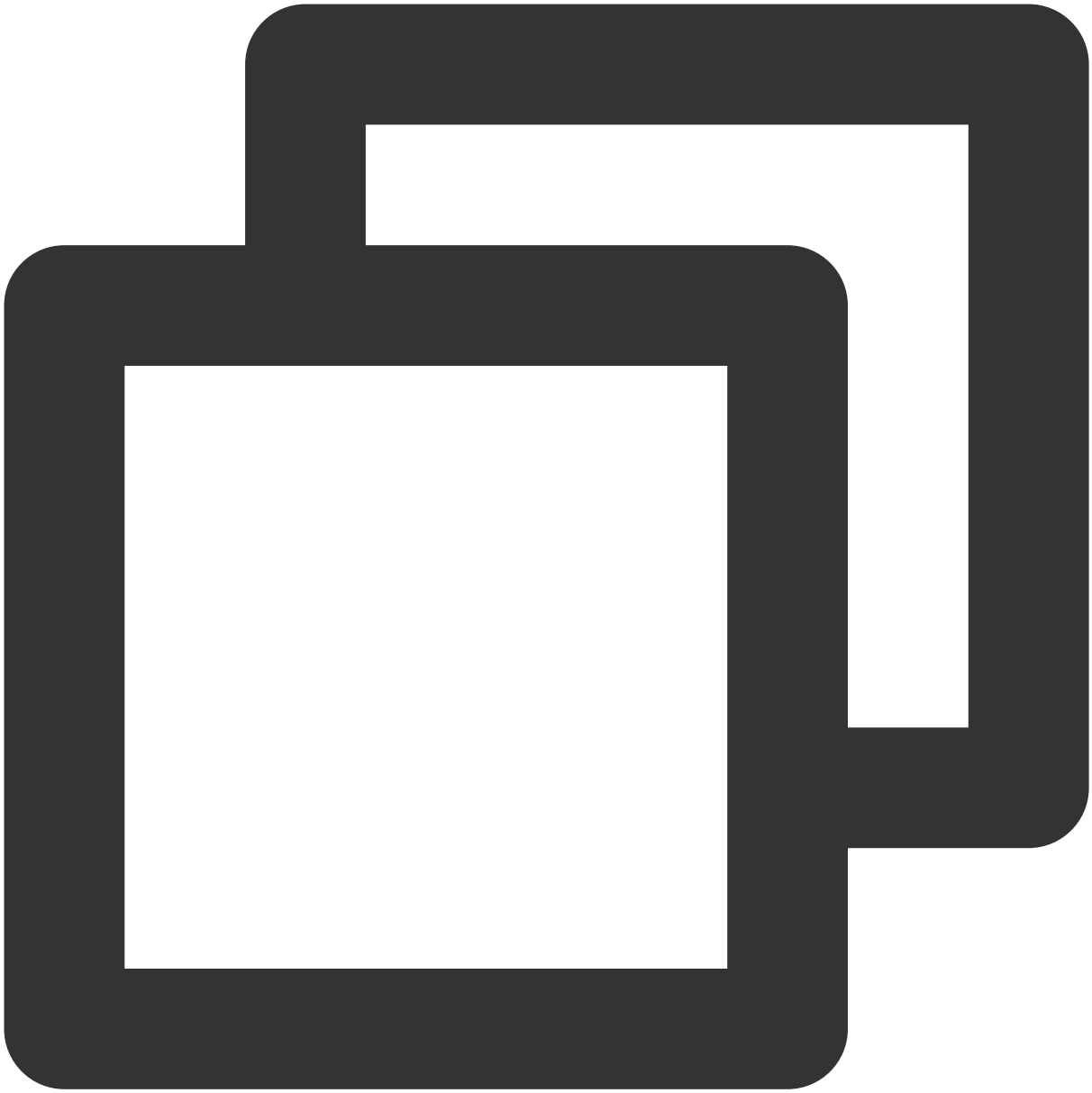
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>TranscodePro</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mxf</Format>
```

```
    </Container>
    <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
    </Video>
    <Audio>
        <Codec>pcm_s24le</Codec>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>TranscodePro</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <Container>
            <Format>mxf</Format>
        </Container>
        <Video>
            <Codec>xavc</Codec>
            <Profile>XAVC-HD_422_10bit</Profile>
            <Width>1920</Width>
            <Height>1080</Height>
            <Interlaced>true</Interlaced>
            <Fps>30000/1001</Fps>
            <Bitrate>50000</Bitrate>
        </Video>
        <Audio>
            <Codec>pcm_s24le</Codec>
        </Audio>
        <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
        </TransConfig>
        <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
        </TimeInterval>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新语音合成模板

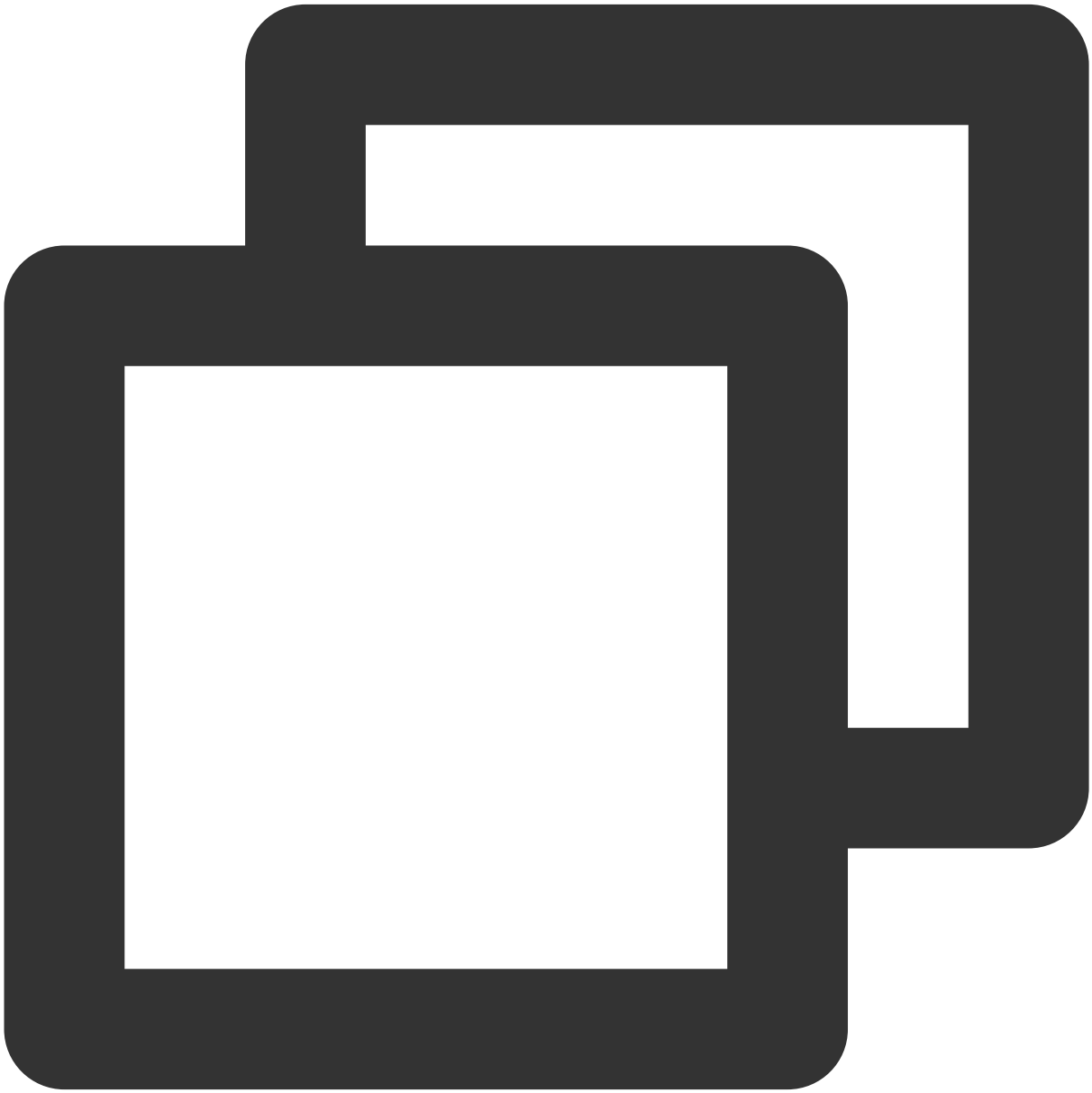
最近更新时间：2023-12-30 16:17:05

功能描述

更新语音合成转码模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

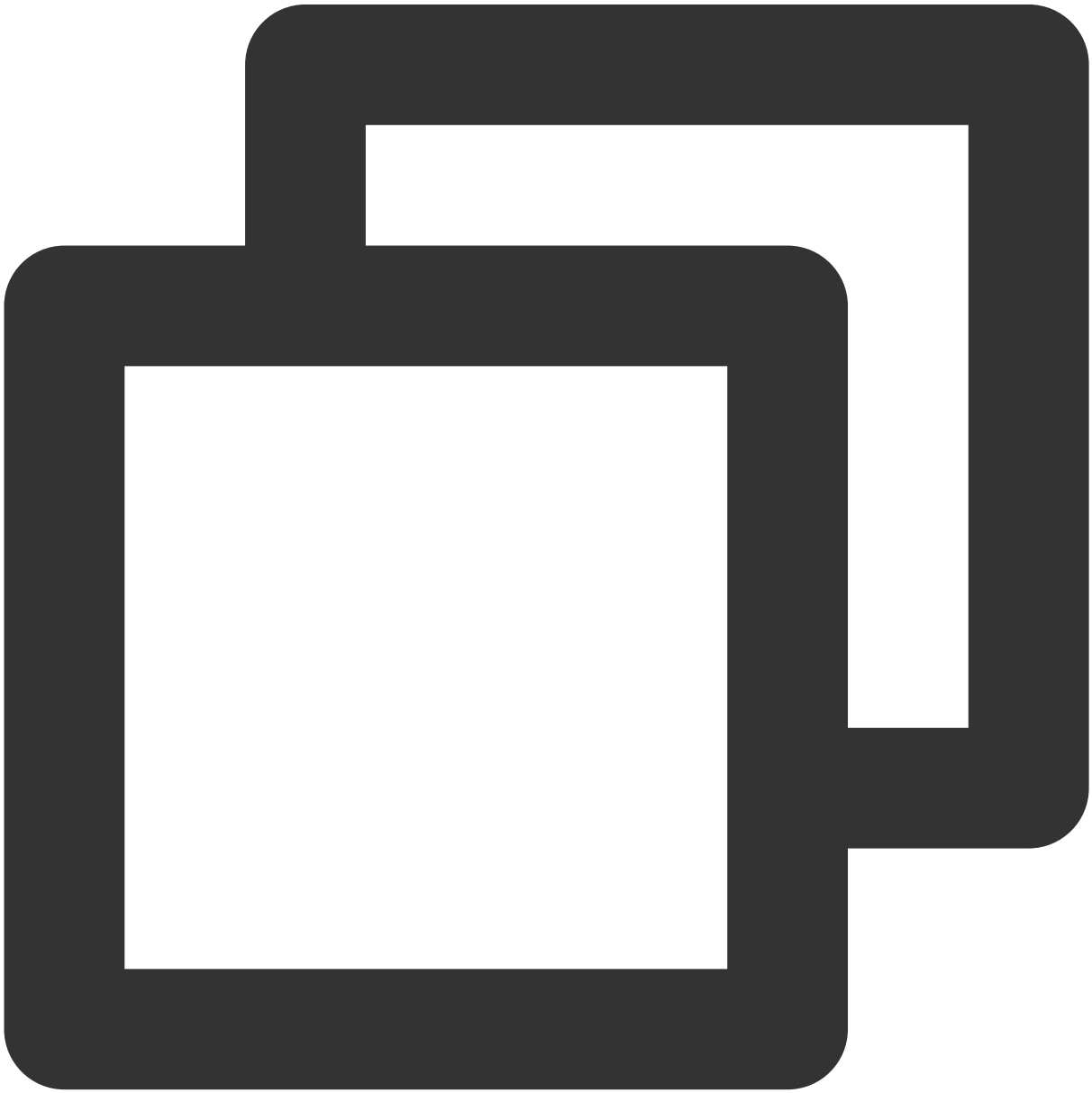
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>Tts</Tag>
    <Name>TemplateName</Name>
    <Mode>Sync</Mode>
    <Codec>pcm</Codec>
    <VoiceType>ruxue</VoiceType>
    <Volume>2</Volume>
    <Speed>200</Speed>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建语音合成模板接口的 Request	Container	是

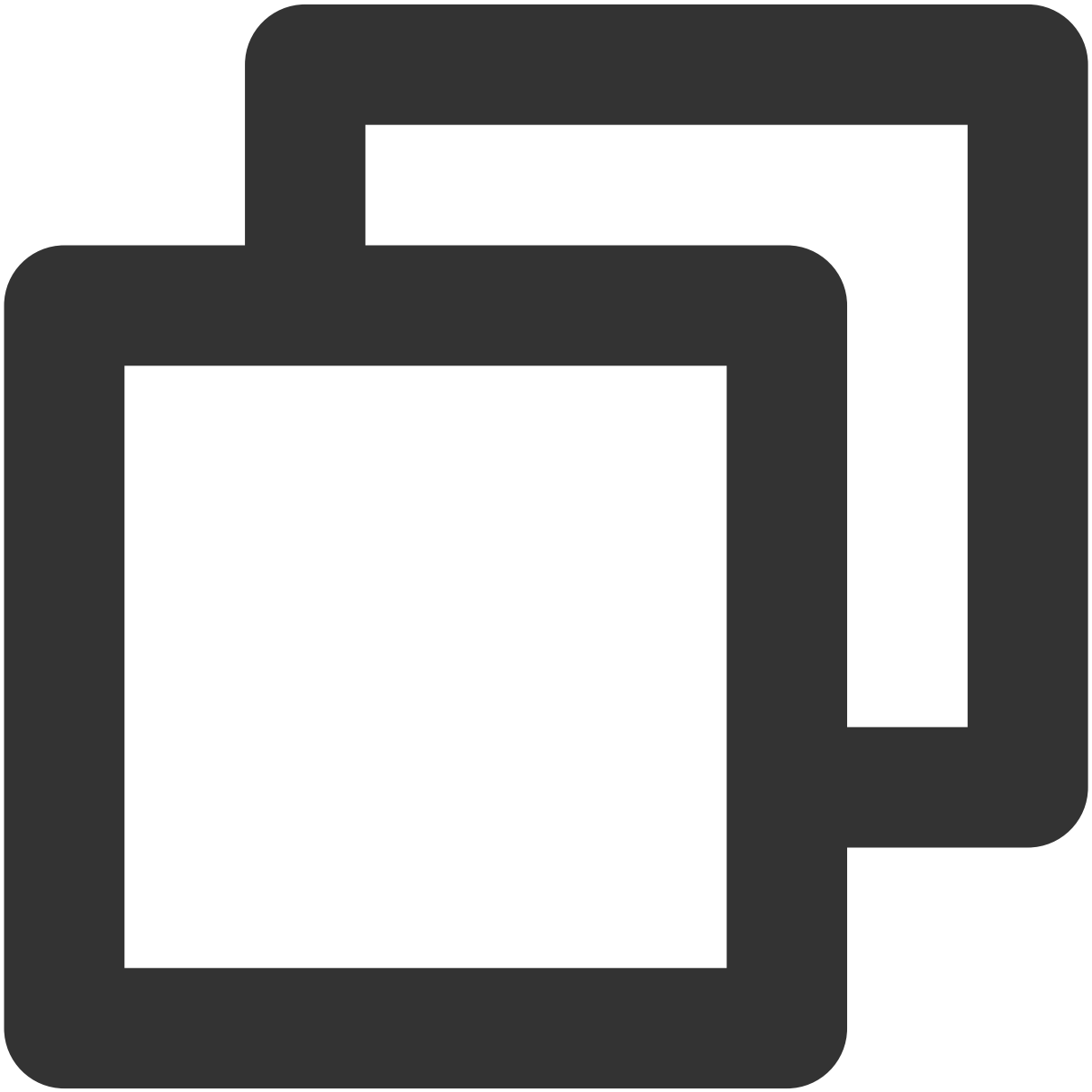
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>Tts</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <TtsTpl>
            <Mode>Sync</Mode>
            <Codec>pcm</Codec>
            <VoiceType>ruxue</VoiceType>
            <Volume>2</Volume>
            <Speed>200</Speed>
```

```
        </TtsTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

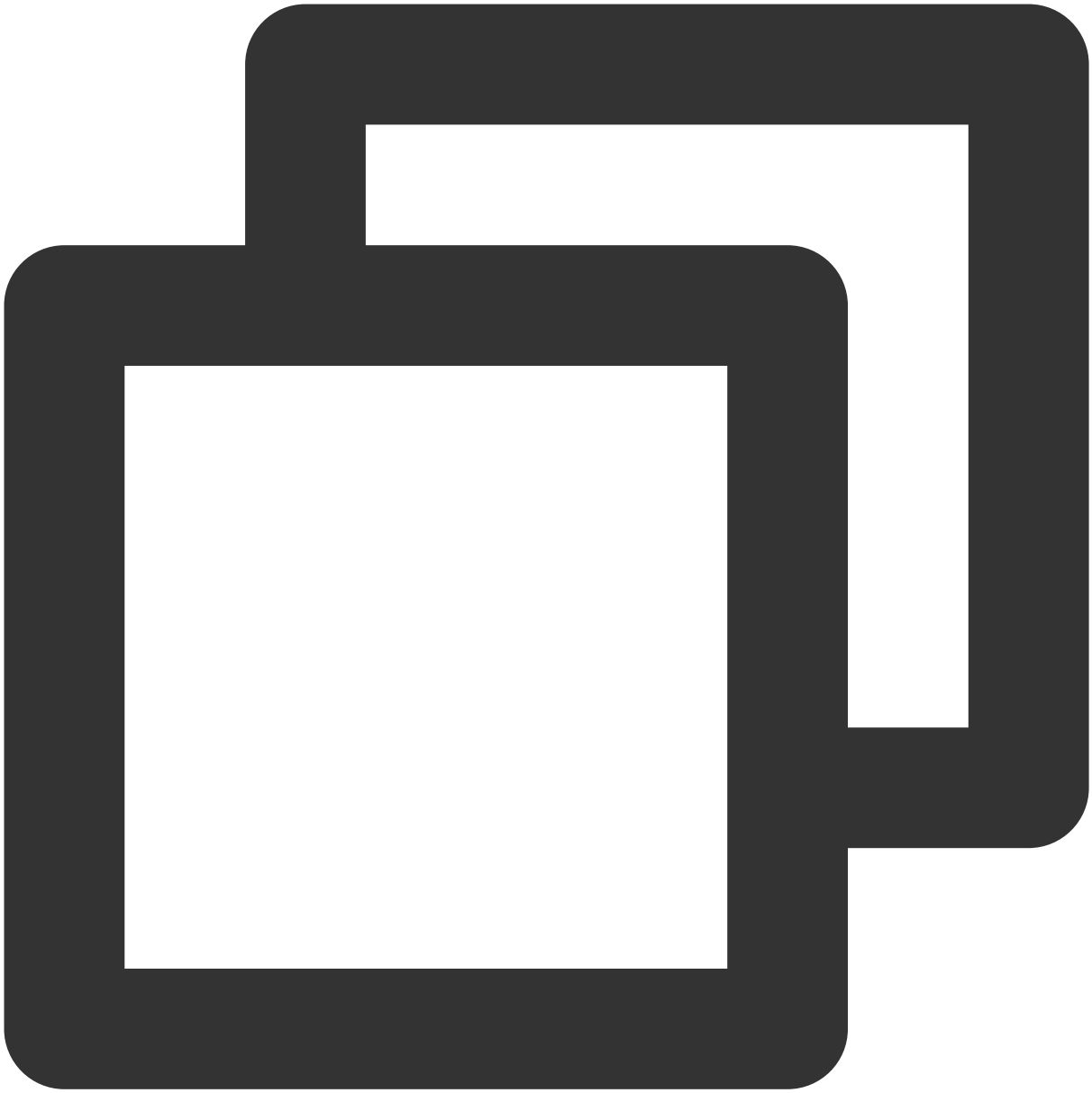
节点名称（关键字）	父节点	描述	类型
Response	无	同创建语音合成模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

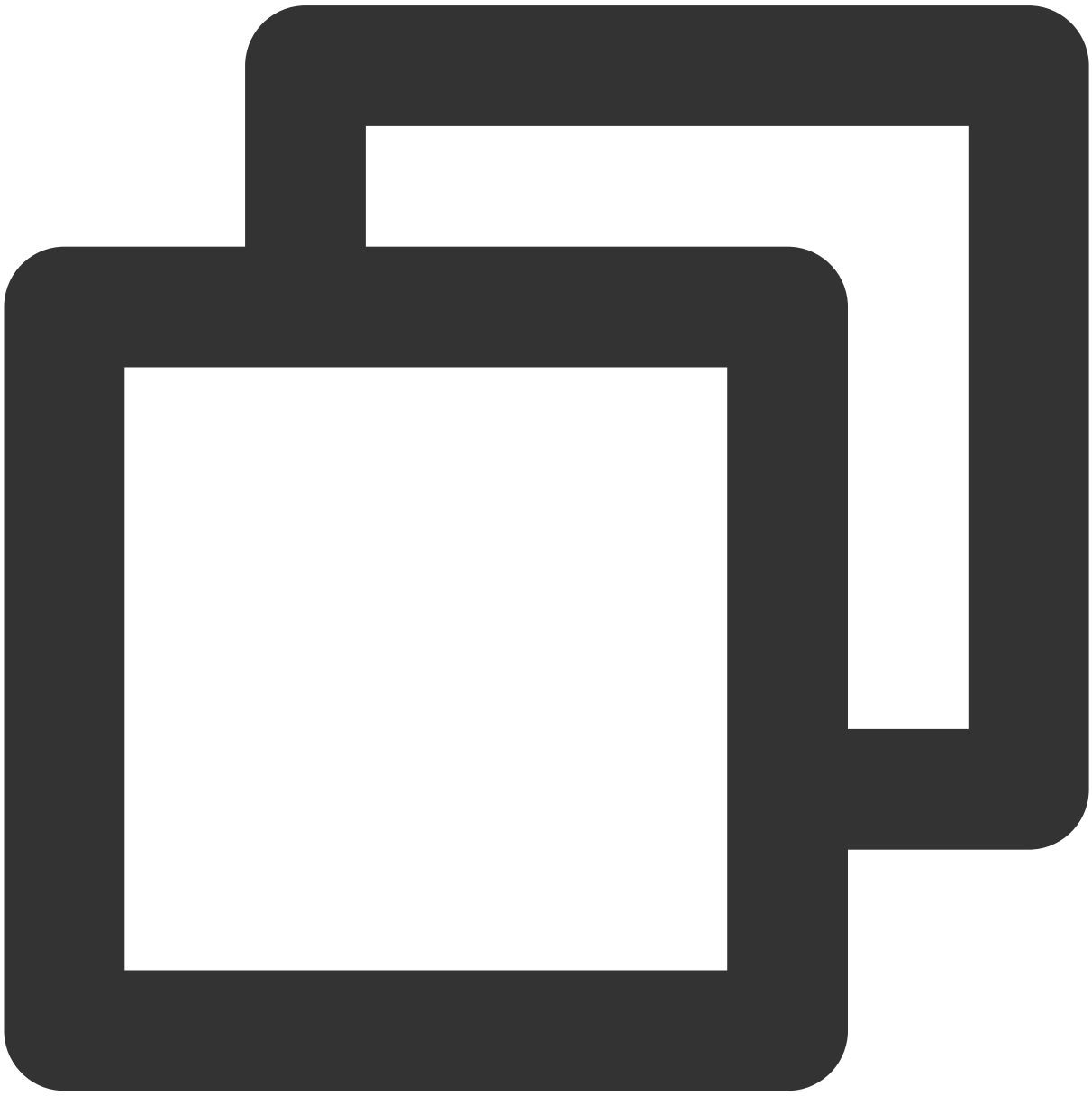
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Tts</Tag>
    <Name>TemplateName</Name>
    <Mode>Sync</Mode>
    <Codec>pcm</Codec>
```

```
    <VoiceType>ruxue</VoiceType>
    <Volume>2</Volume>
    <Speed>200</Speed>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
```

```
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>Tts</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <TtsTpl>
            <Mode>Sync</Mode>
            <Codec>pcm</Codec>
            <VoiceType>ruxue</VoiceType>
            <Volume>2</Volume>
            <Speed>200</Speed>
        </TtsTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新精彩集锦模板

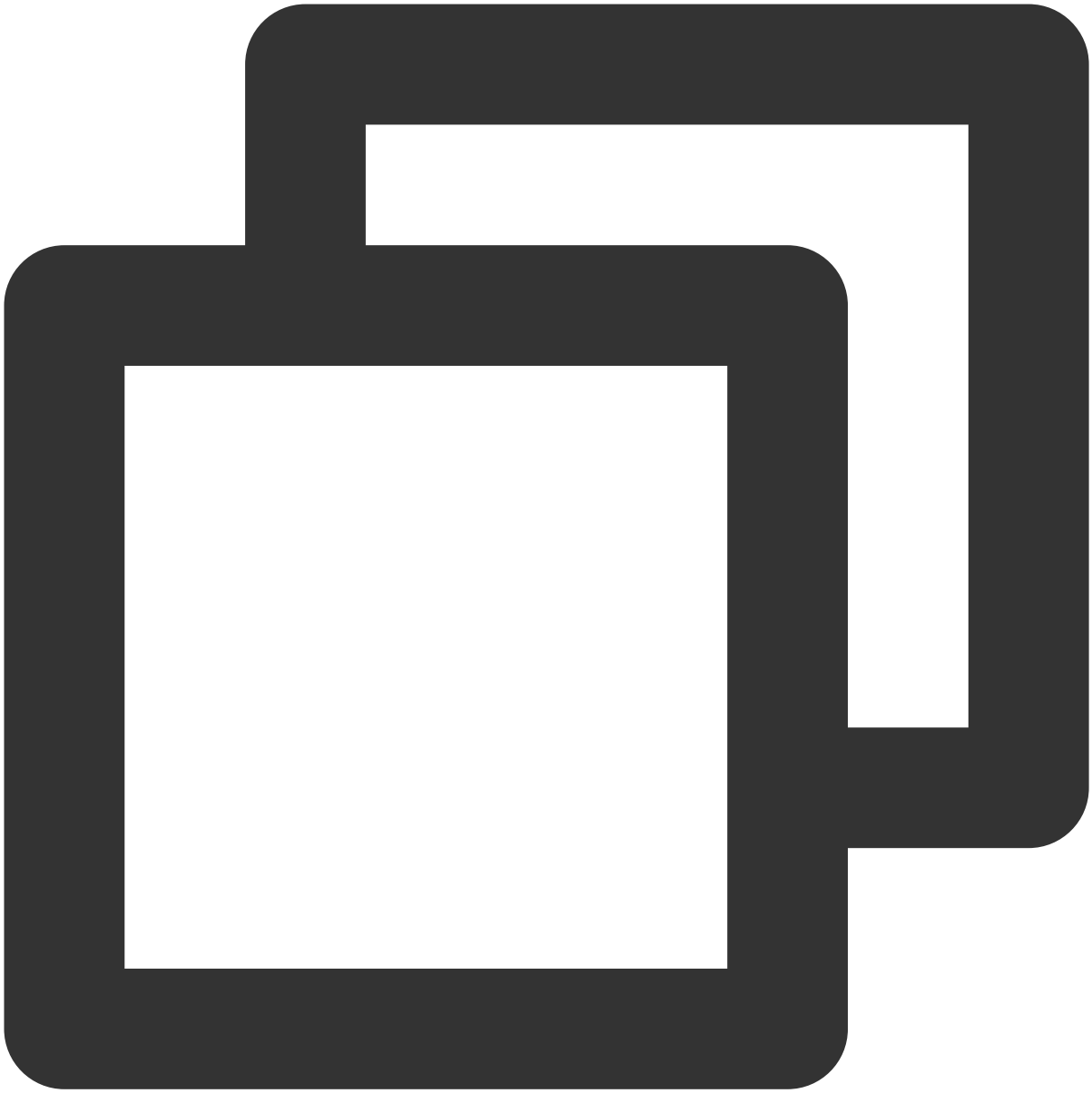
最近更新时间：2023-12-30 16:16:59

功能描述

更新精彩集锦模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

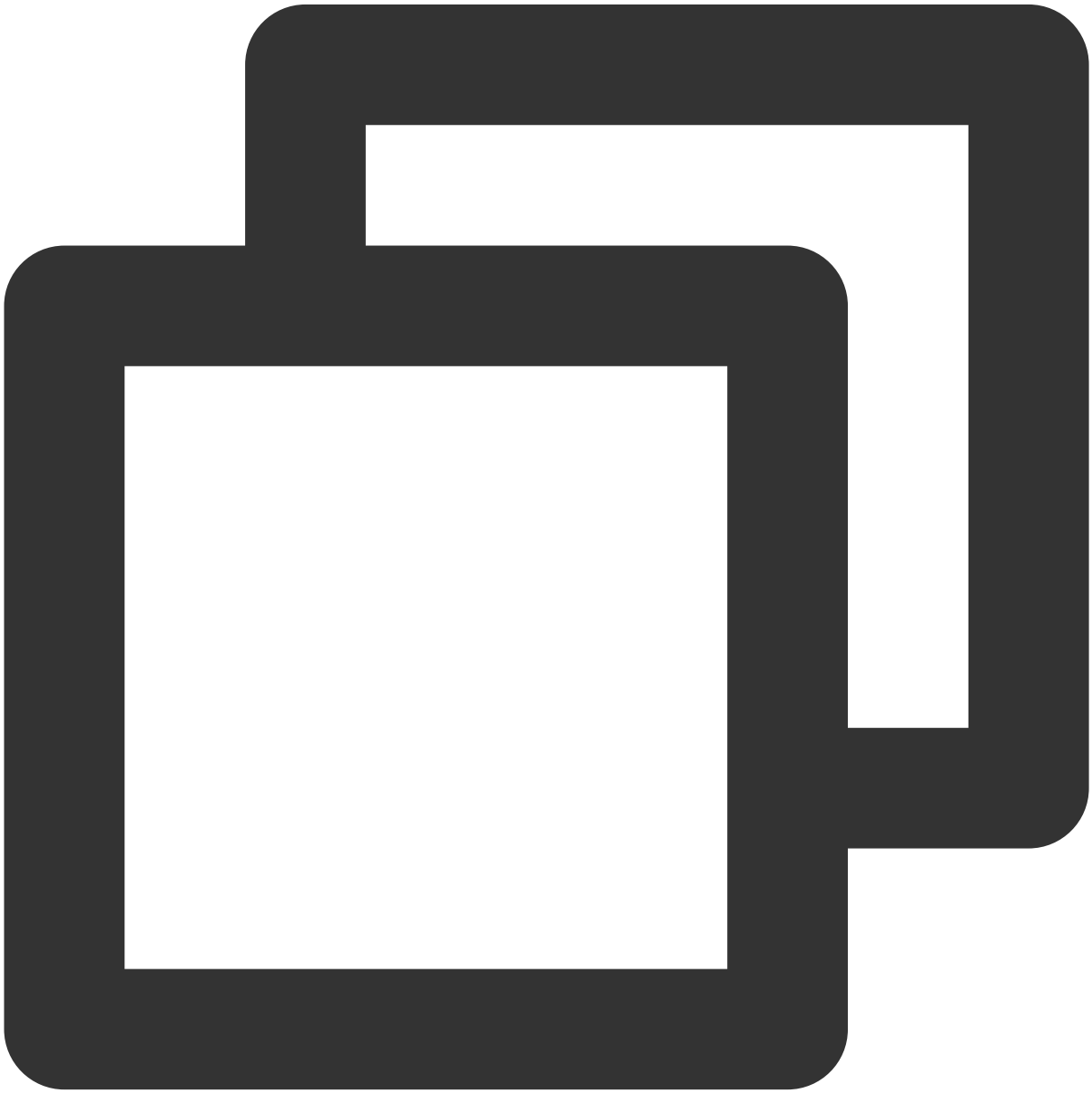
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>VideoMontage</Tag>
    <Name>TemplateName</Name>
    <Duration>10.5</Duration>
    <Container>
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
        <Remove>false</Remove>
    </Audio>
    <AudioMix>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</AudioS
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
    </AudioMix>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建精彩集锦模板接口的 Request	Container	是

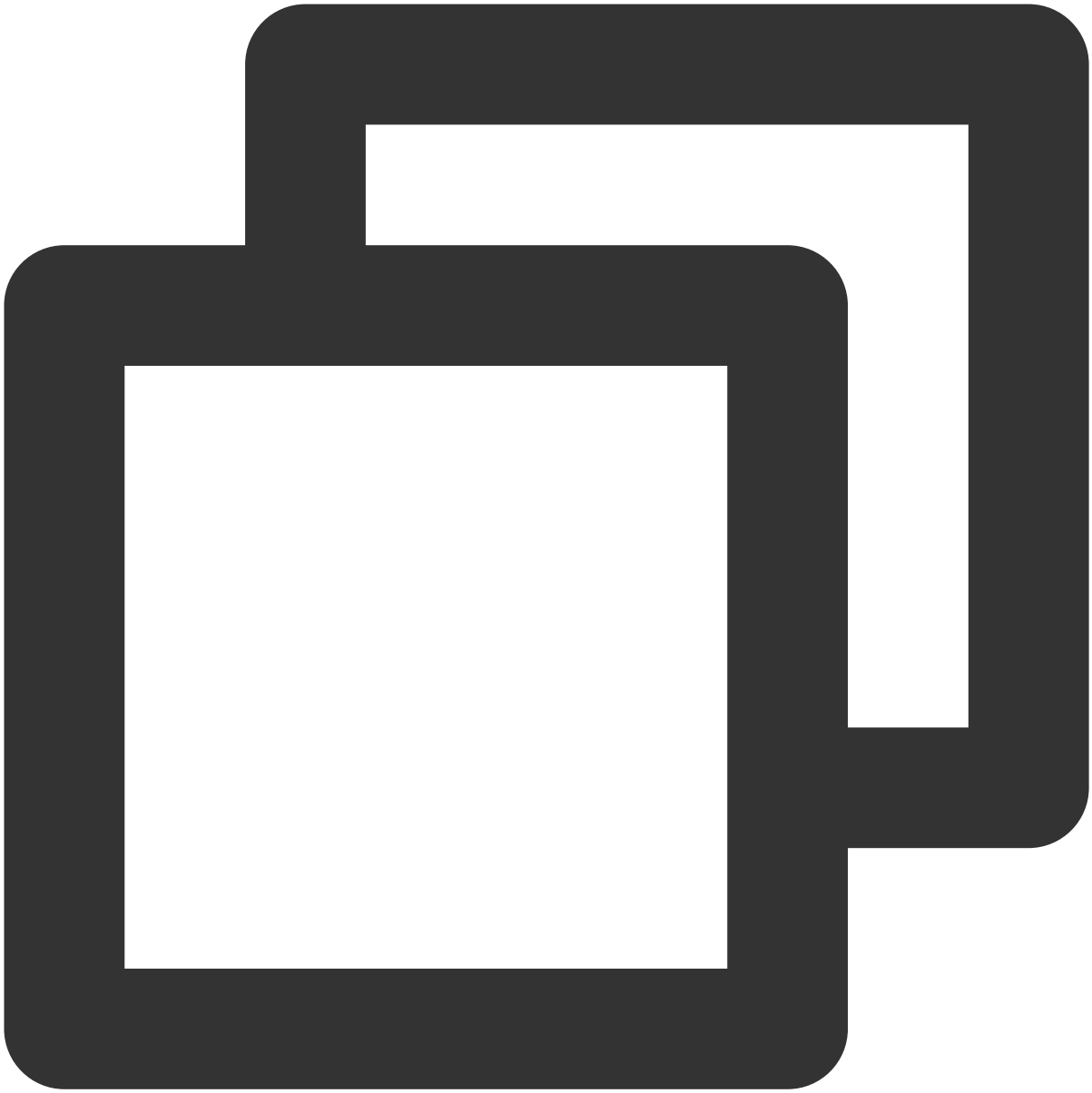
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>VideoMontage</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <VideoMontage>
            <Duration>10.5</Duration>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
```

```
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
                <Remove>false</Remove>
            </Audio>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
            </AudioMix>
        </VideoMontage>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

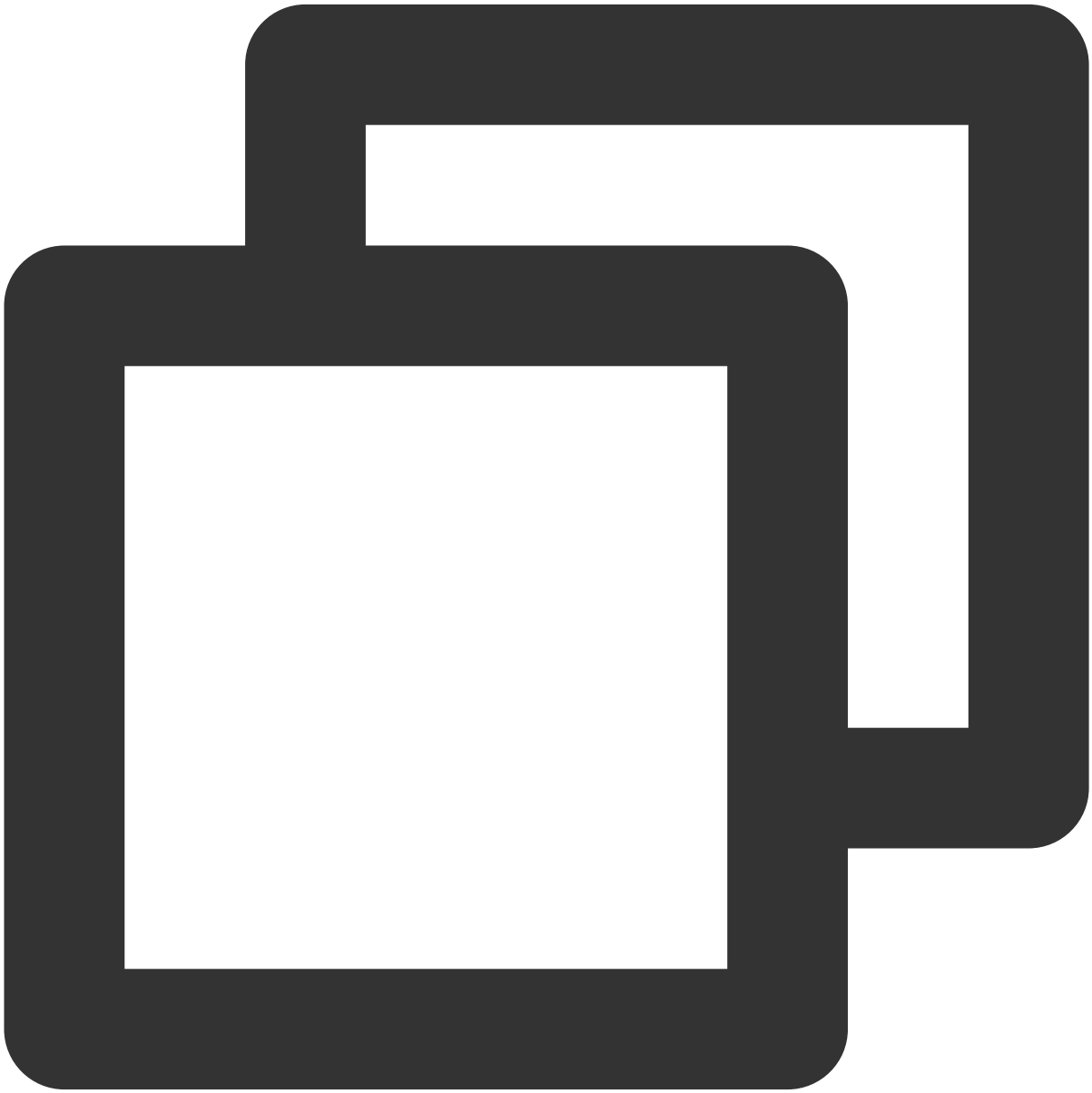
节点名称（关键字）	父节点	描述	类型
Response	无	同创建精彩集锦模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

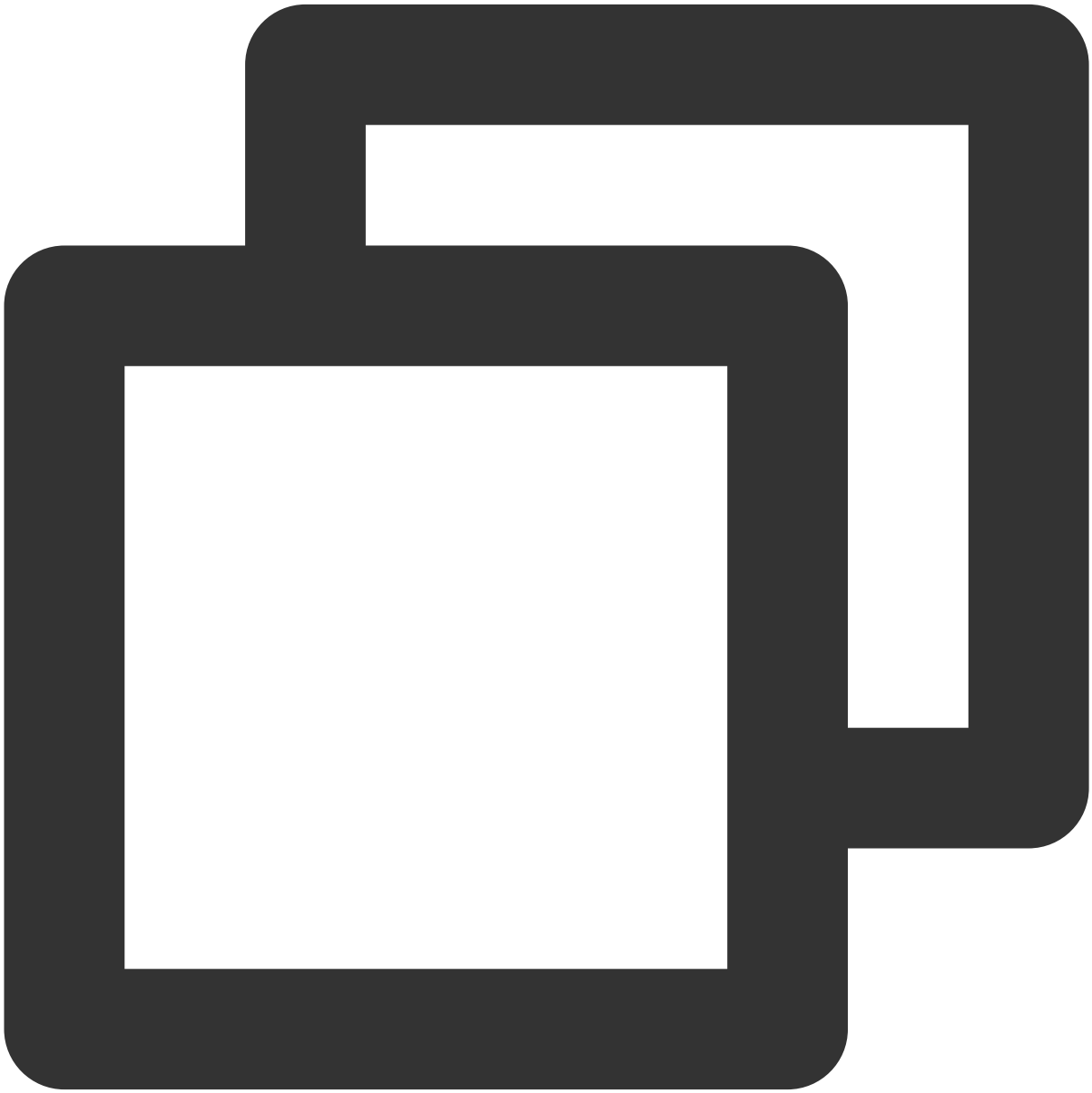
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoMontage</Tag>
    <Name>TemplateName</Name>
    <Duration>10.5</Duration>
    <Container>
```

```
        <Format>mp4</Format>
    </Container>
    <Video>
        <Codec>H.264</Codec>
        <Bitrate>1000</Bitrate>
        <Width>1280</Width>
        <Fps>30</Fps>
    </Video>
    <Audio>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
        <Remove>false</Remove>
    </Audio>
    <AudioMix>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3</AudioS
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
    </AudioMix>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>VideoMontage</Tag>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <VideoMontage>
            <Duration>10.5</Duration>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Fps>30</Fps>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
                <Remove>false</Remove>
            </Audio>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
            </AudioMix>
        </VideoMontage>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新视频增强模板

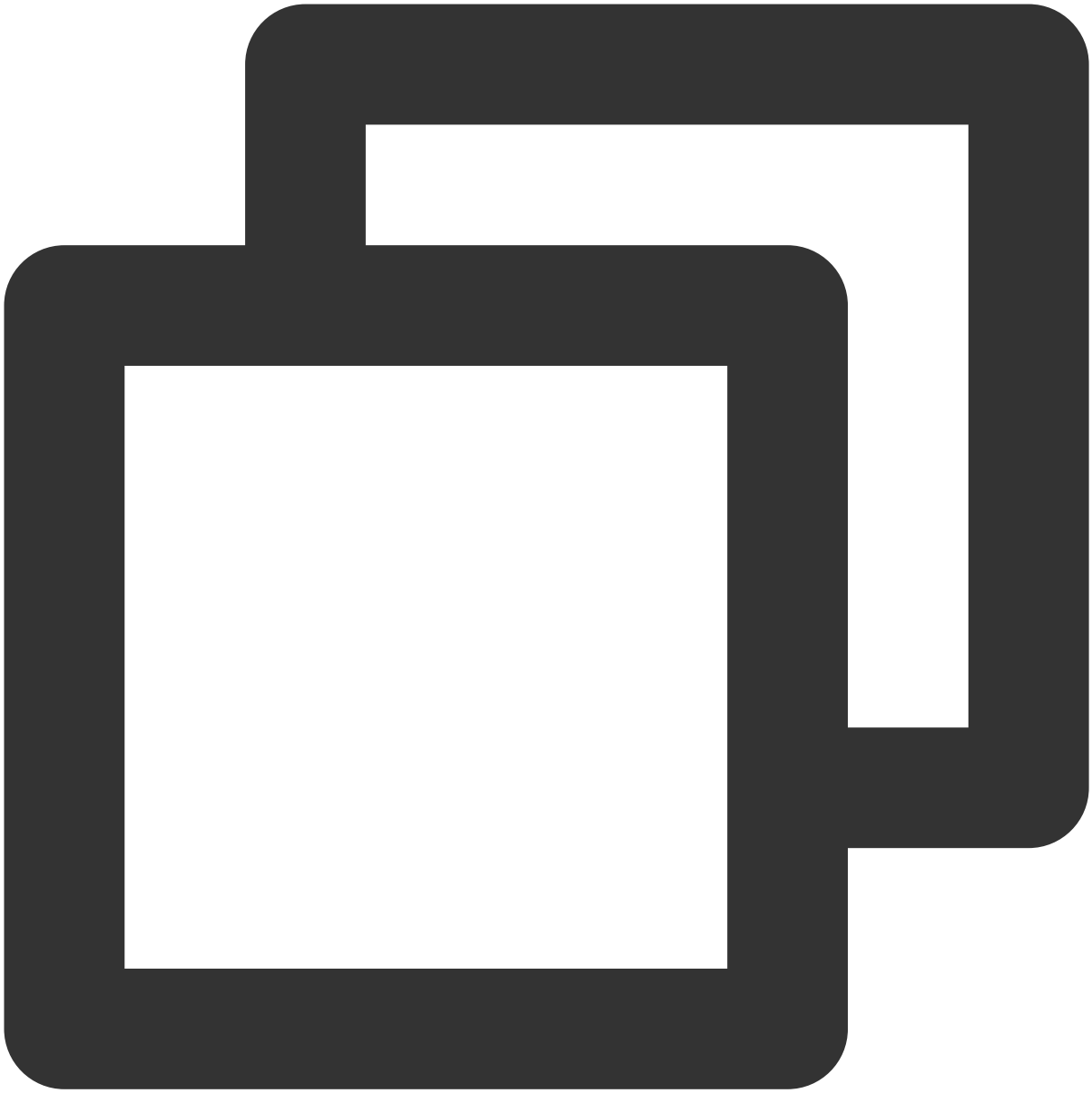
最近更新时间：2023-12-30 16:17:24

功能描述

更新视频增强模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

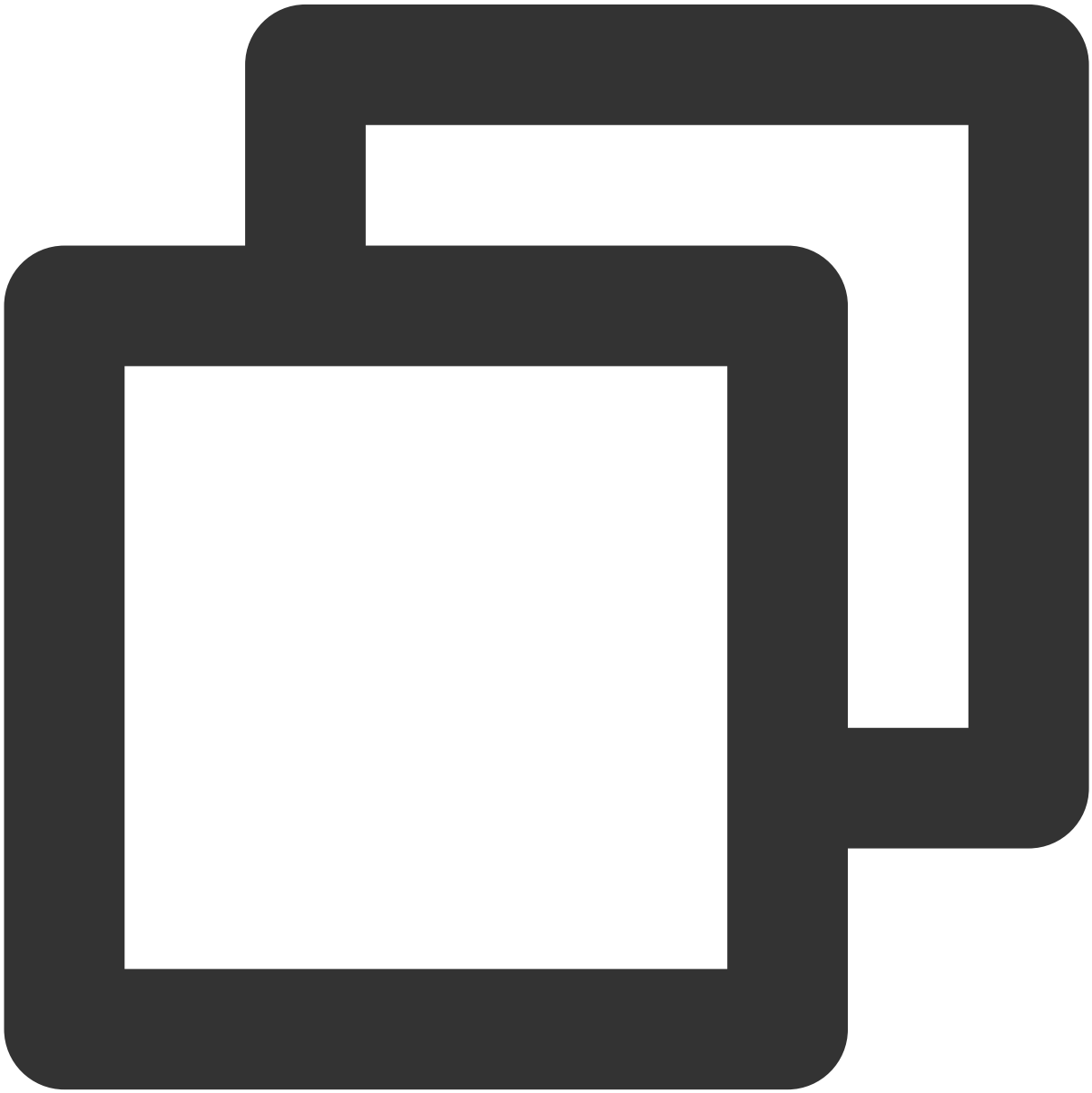
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>VideoProcess</Tag>
    <Name>TemplateName</Name>
    <ColorEnhance>
        <Enable>true</Enable>
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
    </ColorEnhance>
    <MsSharpen>
        <Enable>true</Enable>
        <SharpenLevel></SharpenLevel>
    </MsSharpen>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建视频增强模板接口的 Request	Container	是

响应

响应头

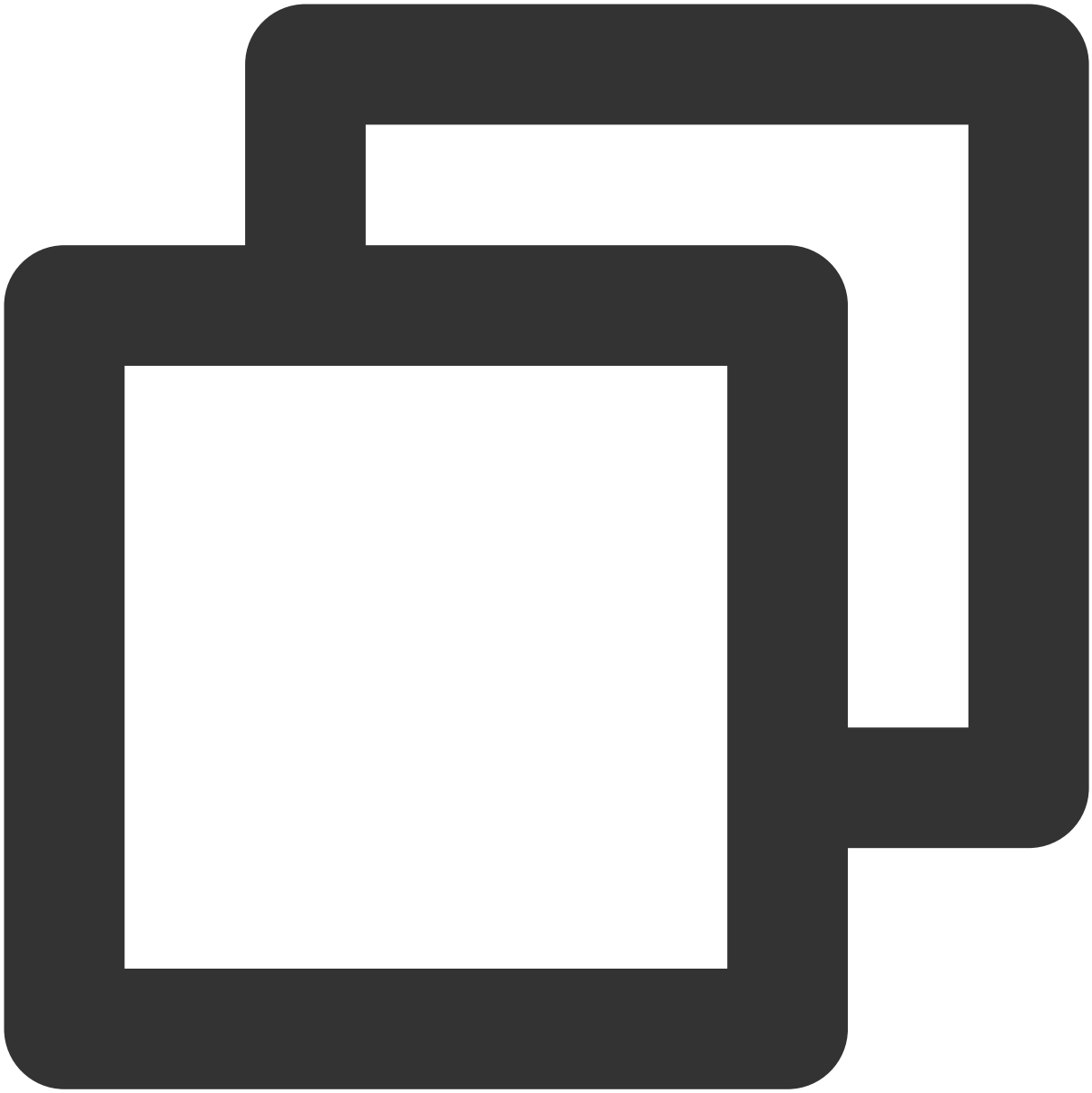
说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>VideoProcess</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <VideoProcess>
            <ColorEnhance>
                <Enable>true</Enable>
                <Contrast></Contrast>
                <Correction></Correction>
                <Saturation></Saturation>
```

```
            </ColorEnhance>
            <MsSharpen>
                <Enable>true</Enable>
                <SharpenLevel></SharpenLevel>
            </MsSharpen>
        </VideoProcess>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

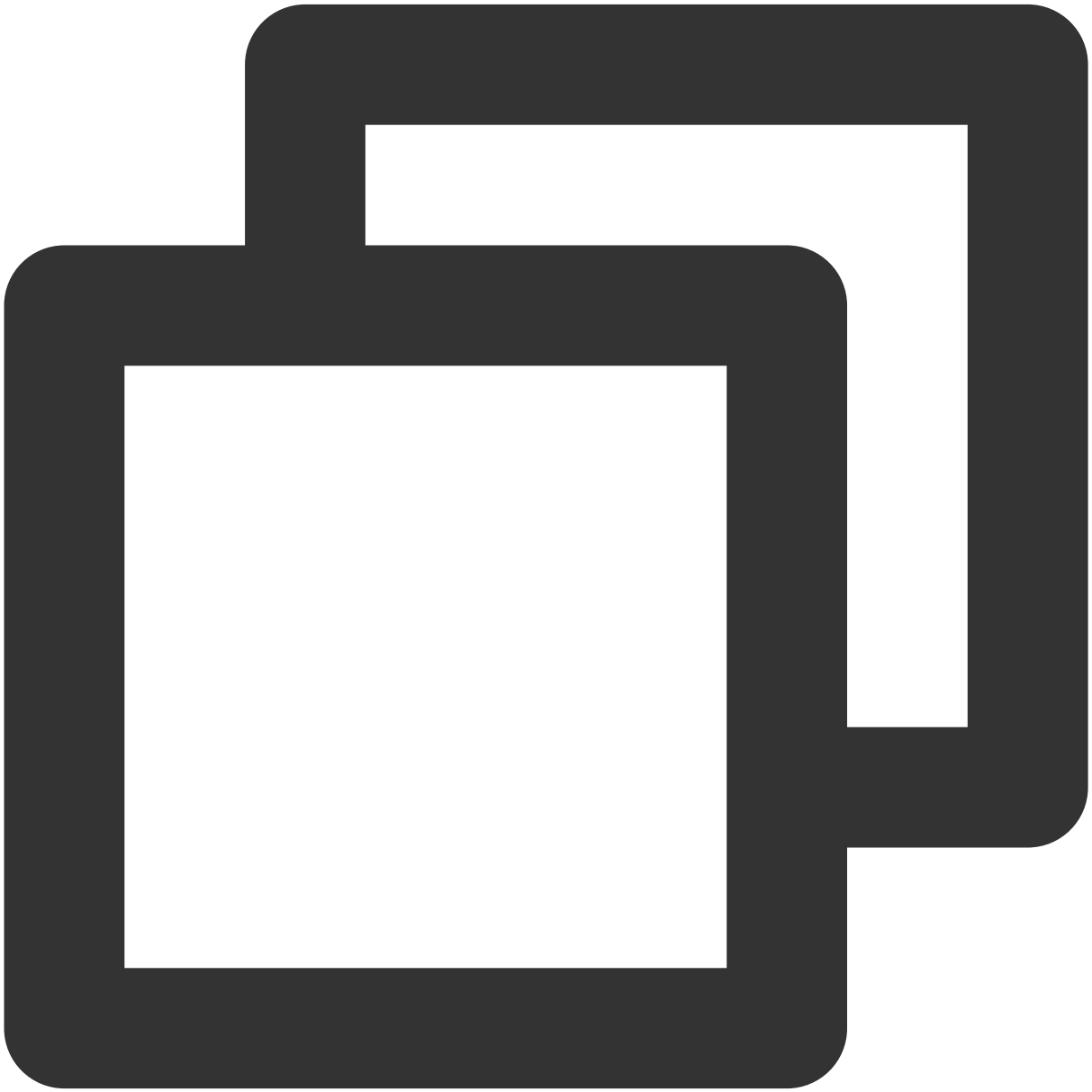
节点名称（关键字）	父节点	描述	类型
Response	无	同创建视频增强模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

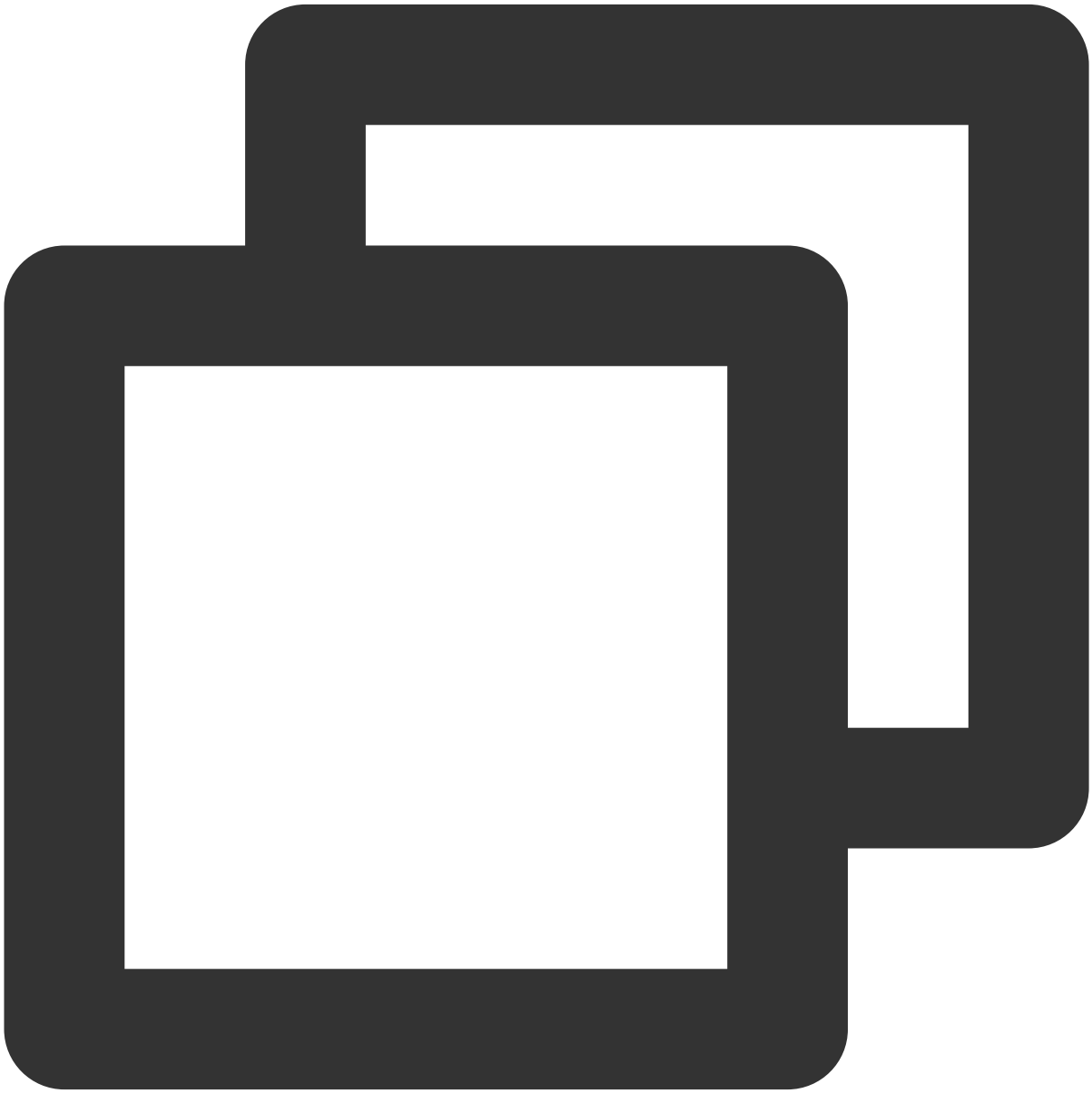
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoProcess</Tag>
    <Name>TemplateName</Name>
    <ColorEnhance>
        <Enable>true</Enable>
```

```
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
    </ColorEnhance>
    <MsSharpen>
        <Enable>true</Enable>
        <SharpenLevel></SharpenLevel>
    </MsSharpen>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>VideoProcess</Tag>
```

```
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <VideoProcess>
            <ColorEnhance>
                <Enable>true</Enable>
                <Contrast></Contrast>
                <Correction></Correction>
                <Saturation></Saturation>
            </ColorEnhance>
            <MsSharpen>
                <Enable>true</Enable>
                <SharpenLevel></SharpenLevel>
            </MsSharpen>
        </VideoProcess>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新人声分离模板

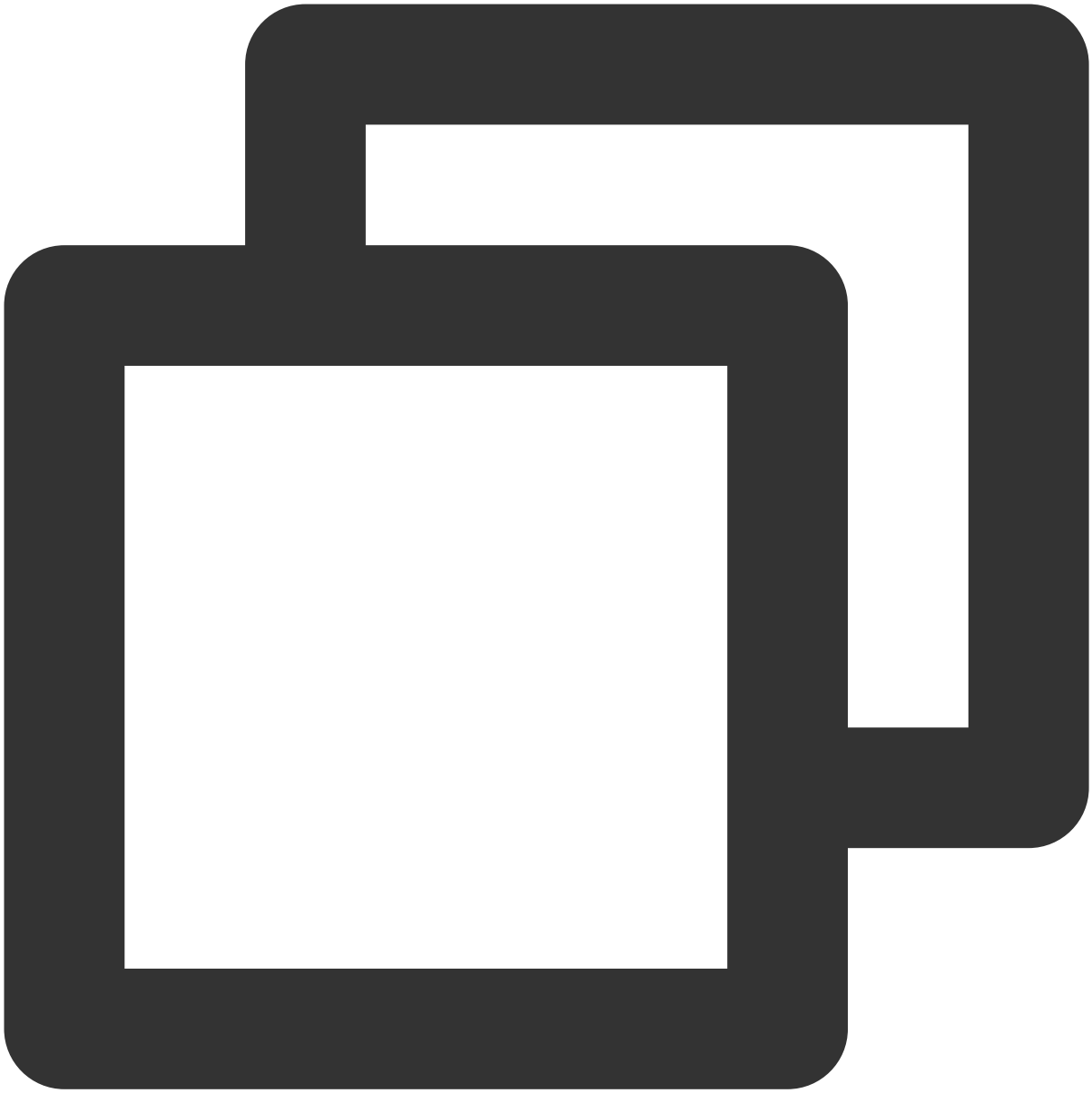
最近更新时间：2023-12-30 16:17:31

功能描述

更新人声分离转码模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

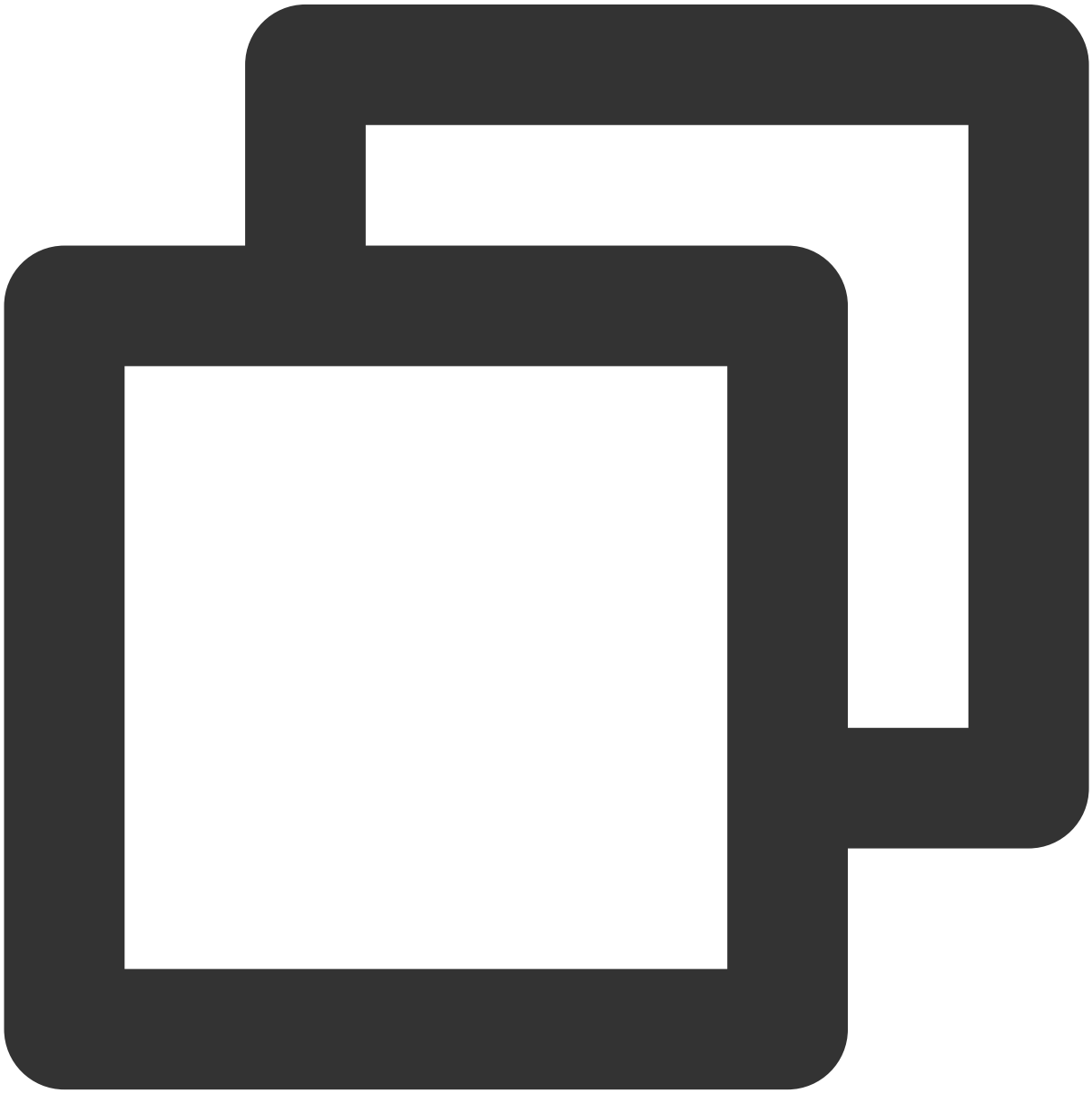
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
```

```
    <Tag>VoiceSeparate</Tag>
    <Name>TemplateName</Name>
    <AudioMode>IsAudio</AudioMode>
    <AudioConfig>
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>2</Channels>
    </AudioConfig>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建人声分离模板接口的 Request	Container	是

响应

响应头

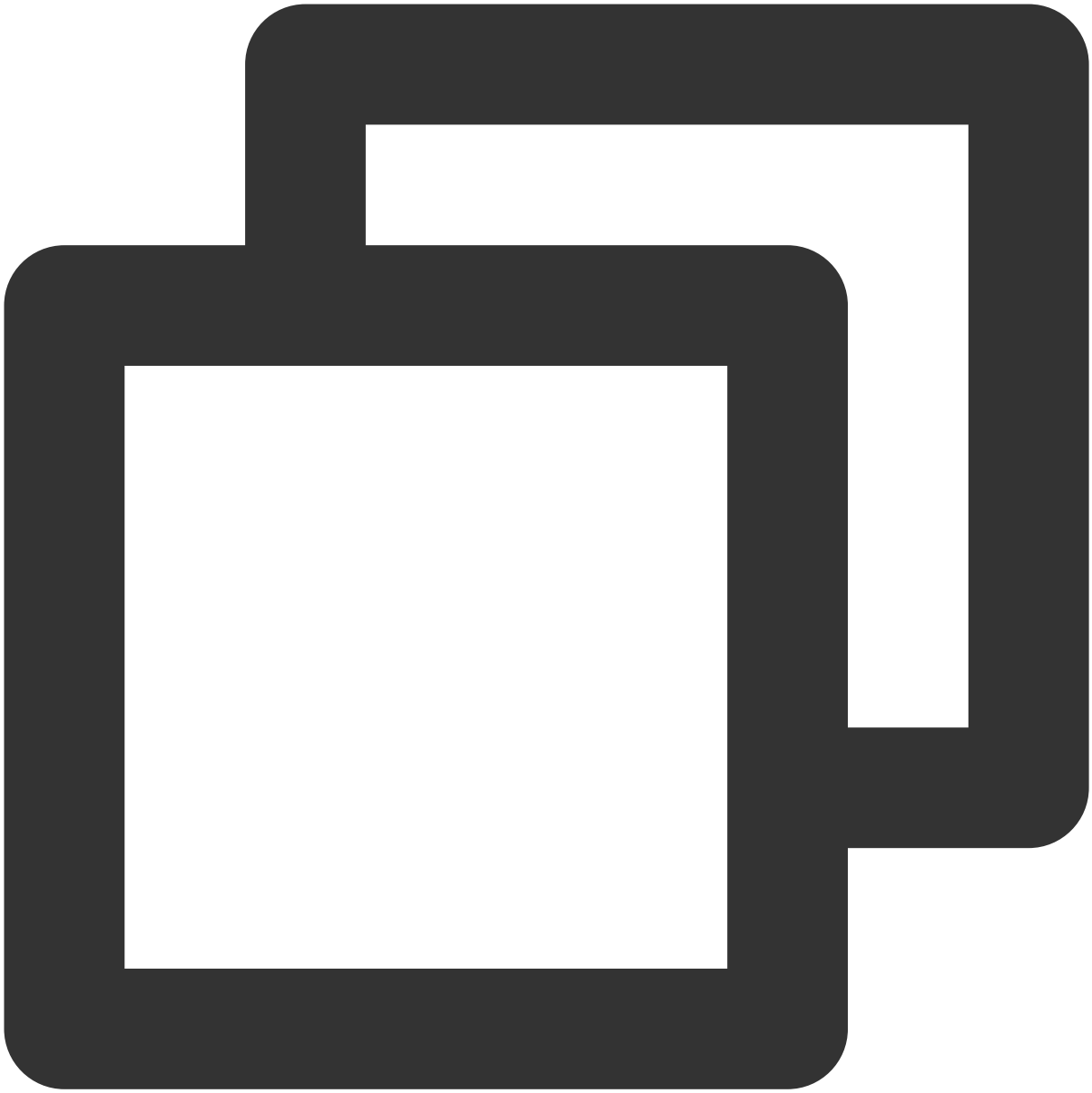
说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>VoiceSeparate</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <VoiceSeparate>
            <AudioMode>IsAudio</AudioMode>
            <AudioConfig>
                <Codec>mp3</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>12</Bitrate>
```

```
                <Channels>2</Channels>
            </AudioConfig>
        </VoiceSeparate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

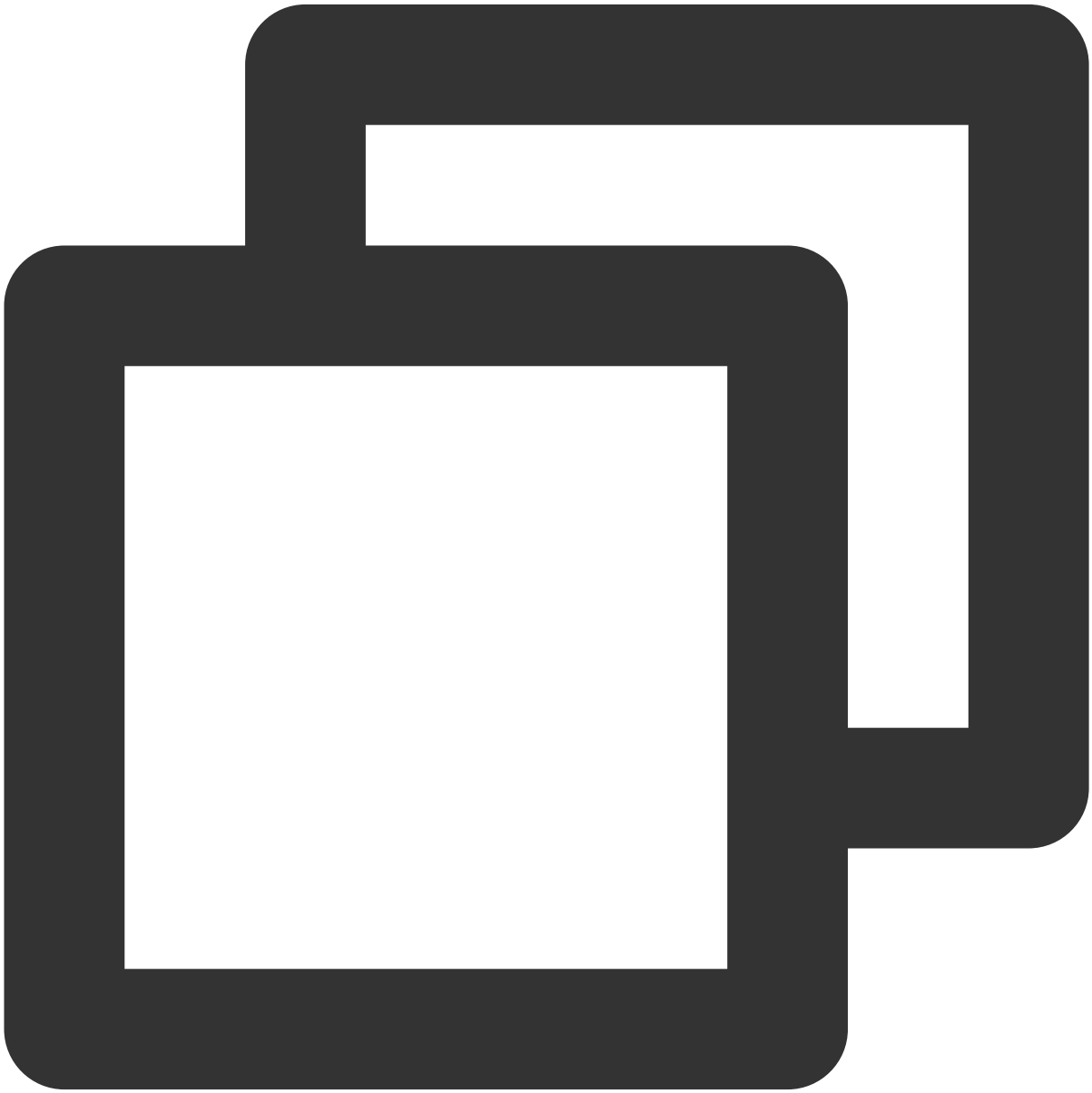
节点名称（关键字）	父节点	描述	类型
Response	无	同创建人声分离模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

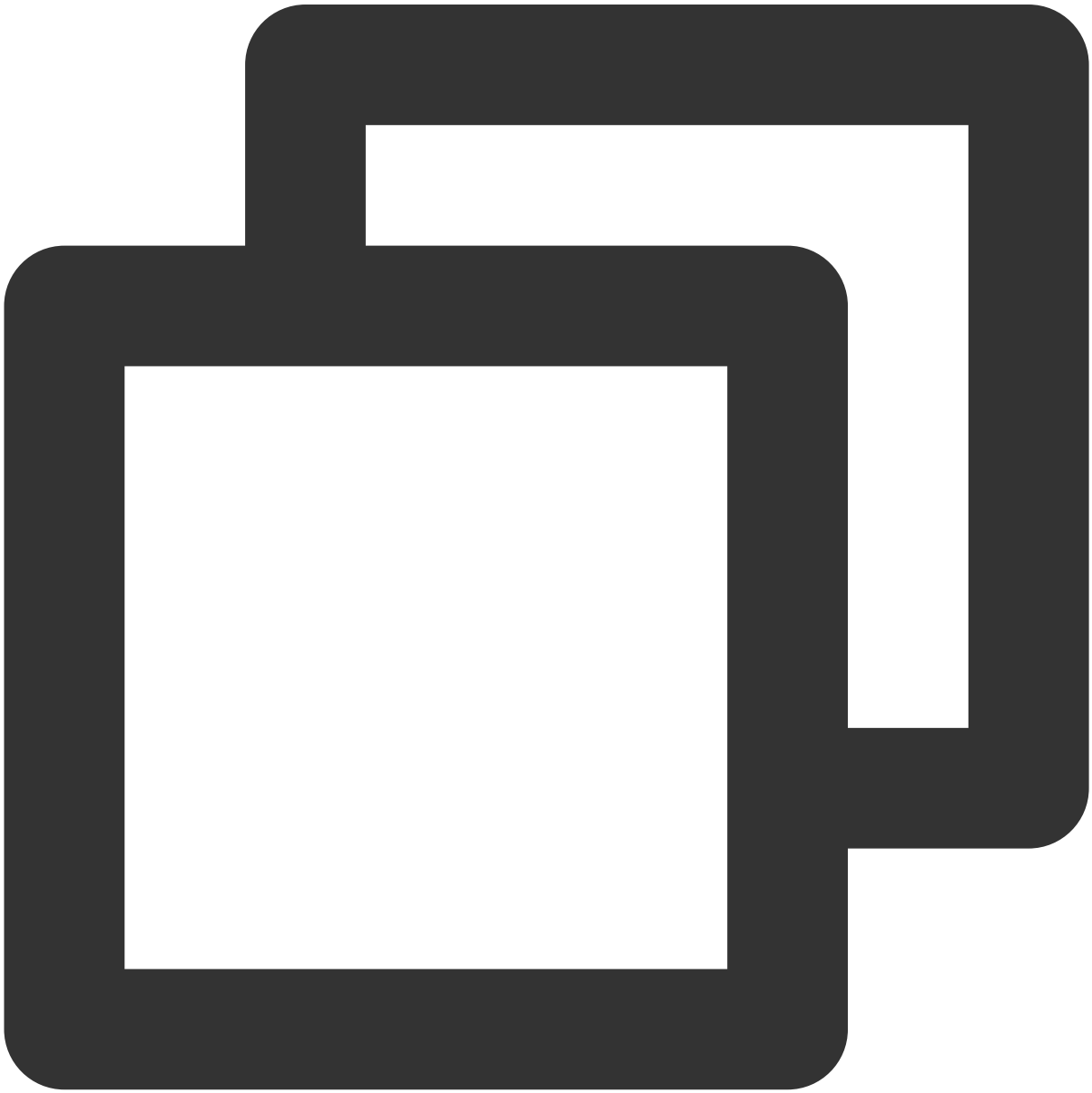
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VoiceSeparate</Tag>
    <Name>TemplateName</Name>
    <AudioMode>IsAudio</AudioMode>
    <AudioConfig>
```

```
        <Codec>aac</Codec>
        <Samplerate>44100</Samplerate>
        <Bitrate>128</Bitrate>
        <Channels>4</Channels>
    </AudioConfig>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <Tag>VoiceSeparate</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <VoiceSeparate>
            <AudioMode>IsAudio</AudioMode>
            <AudioConfig>
                <Codec>mp3</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>12</Bitrate>
                <Channels>2</Channels>
            </AudioConfig>
        </VoiceSeparate>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新水印模板

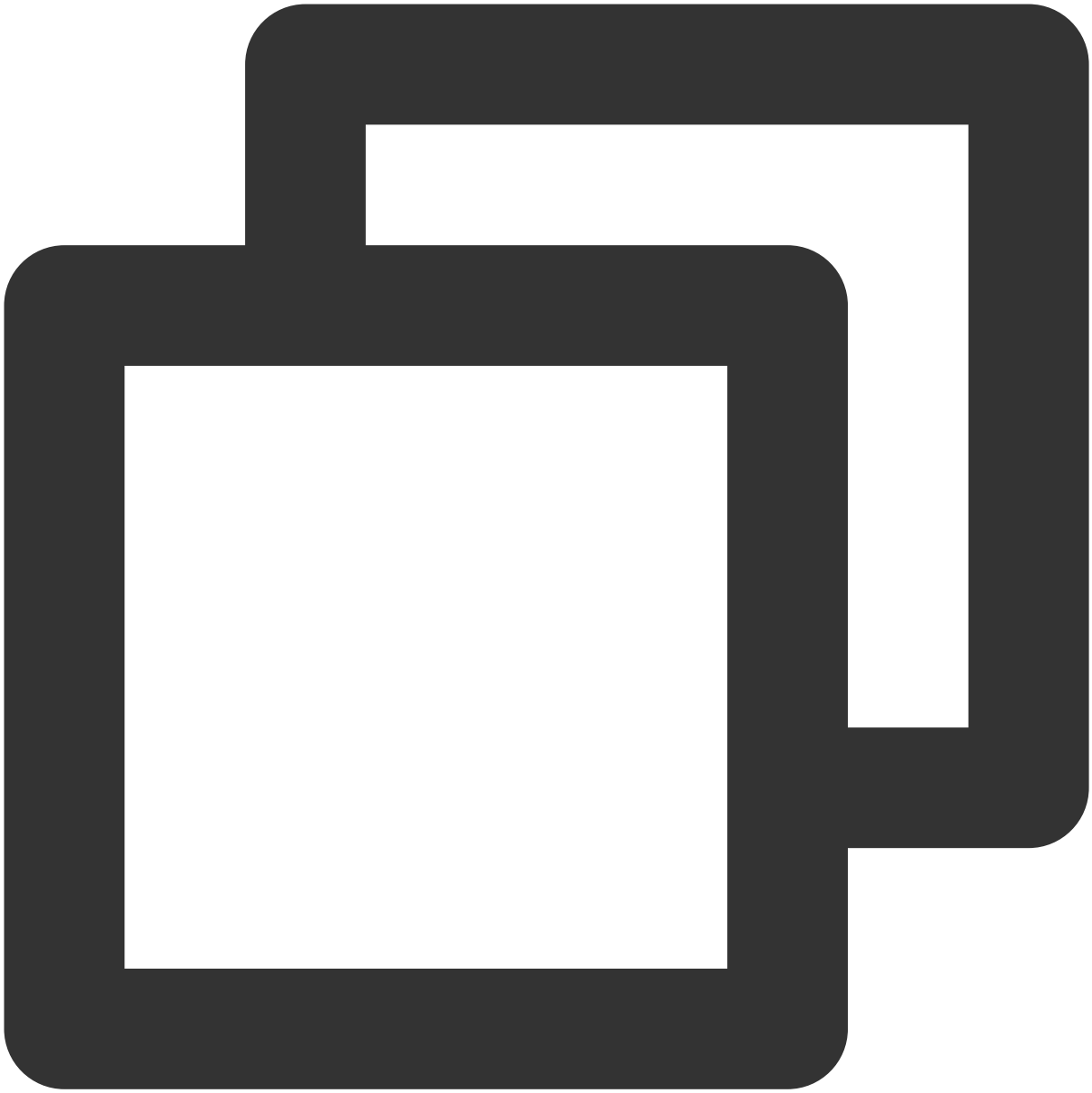
最近更新时间：2023-12-30 16:18:00

功能描述

更新水印模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

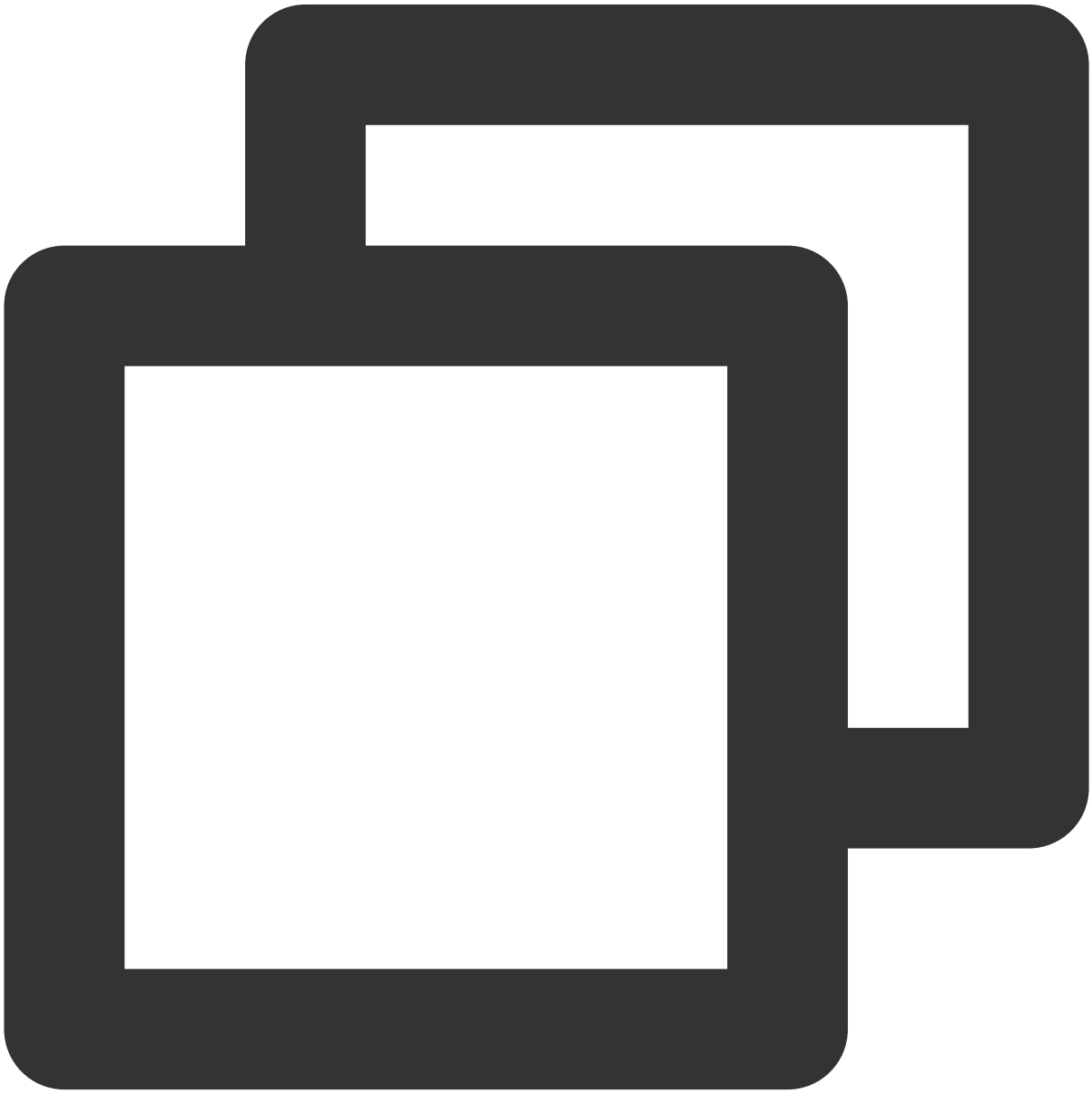
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>Watermark</Tag>
    <Name>TemplateName</Name>
    <Watermark>
        <Type>Text</Type>
        <LocMode>Absolute</LocMode>
        <Dx>128</Dx>
        <Dy>128</Dy>
        <Pos>TopRight</Pos>
        <StartTime>0</StartTime>
        <EndTime>100.5</EndTime>
        <Text>
            <Text>水印内容</Text>
            <FontSize>30</FontSize>
            <FontType></FontType>
            <FontColor>0xRRGGBB</FontColor>
            <Transparency>30</Transparency>
        </Text>
    </Watermark>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建水印模板接口的 Request	Container	是

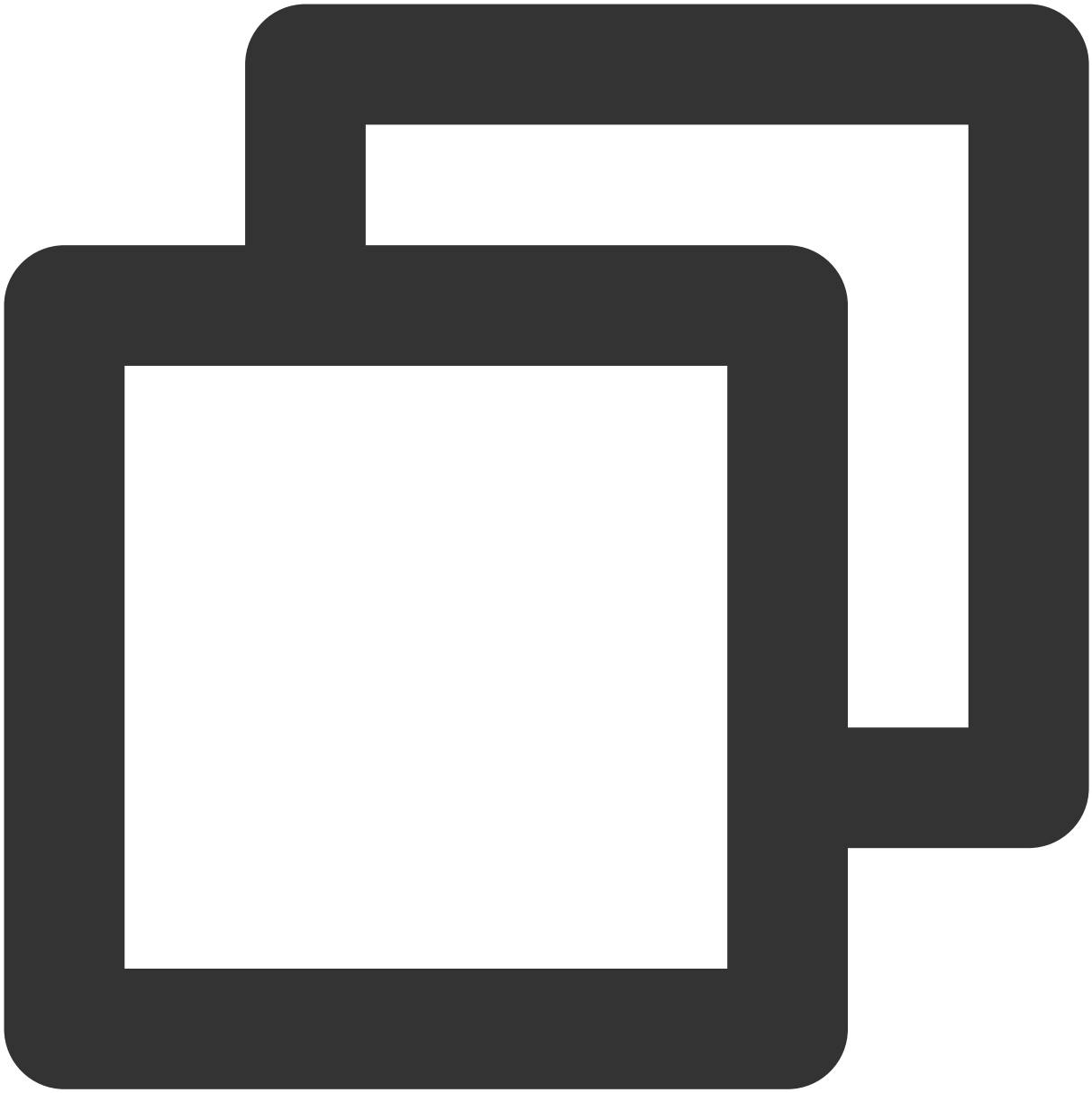
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Tag>Watermark</Tag>
        <Name>TemplateName</Name>
        <Watermark>
            <Type>Text</Type>
            <LocMode>Absolute</LocMode>
            <Dx>128</Dx>
            <Dy>128</Dy>
            <Pos>TopRight</Pos>
```

```
            <StartTime>0</StartTime>
            <EndTime>100.5</EndTime>
            <Text>
                <Text>水印内容</Text>
                <FontSize>30</FontSize>
                <FontType></FontType>
                <FontColor>0xRRGGBB</FontColor>
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

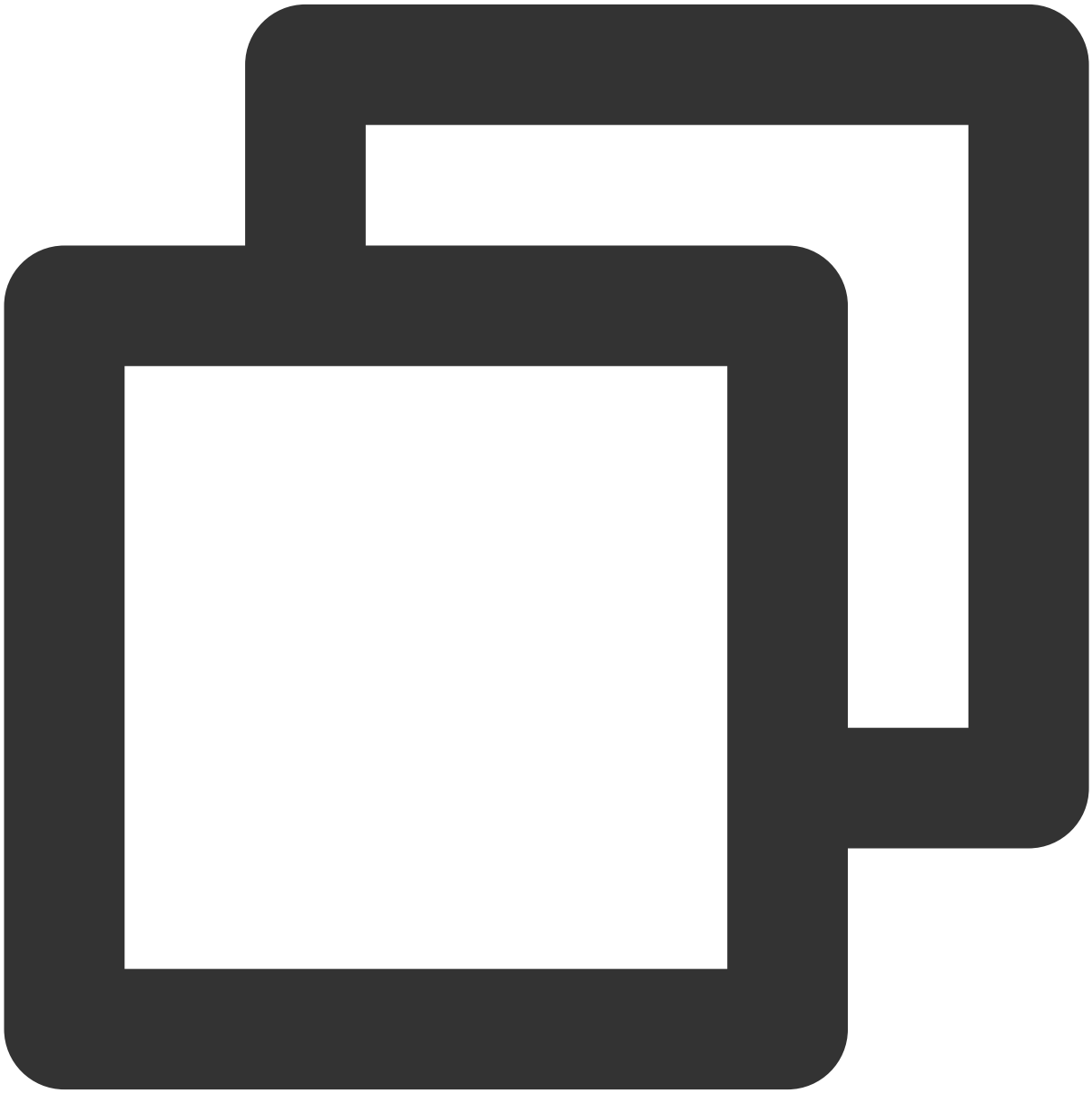
节点名称（关键字）	父节点	描述	类型
Response	无	同创建水印模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

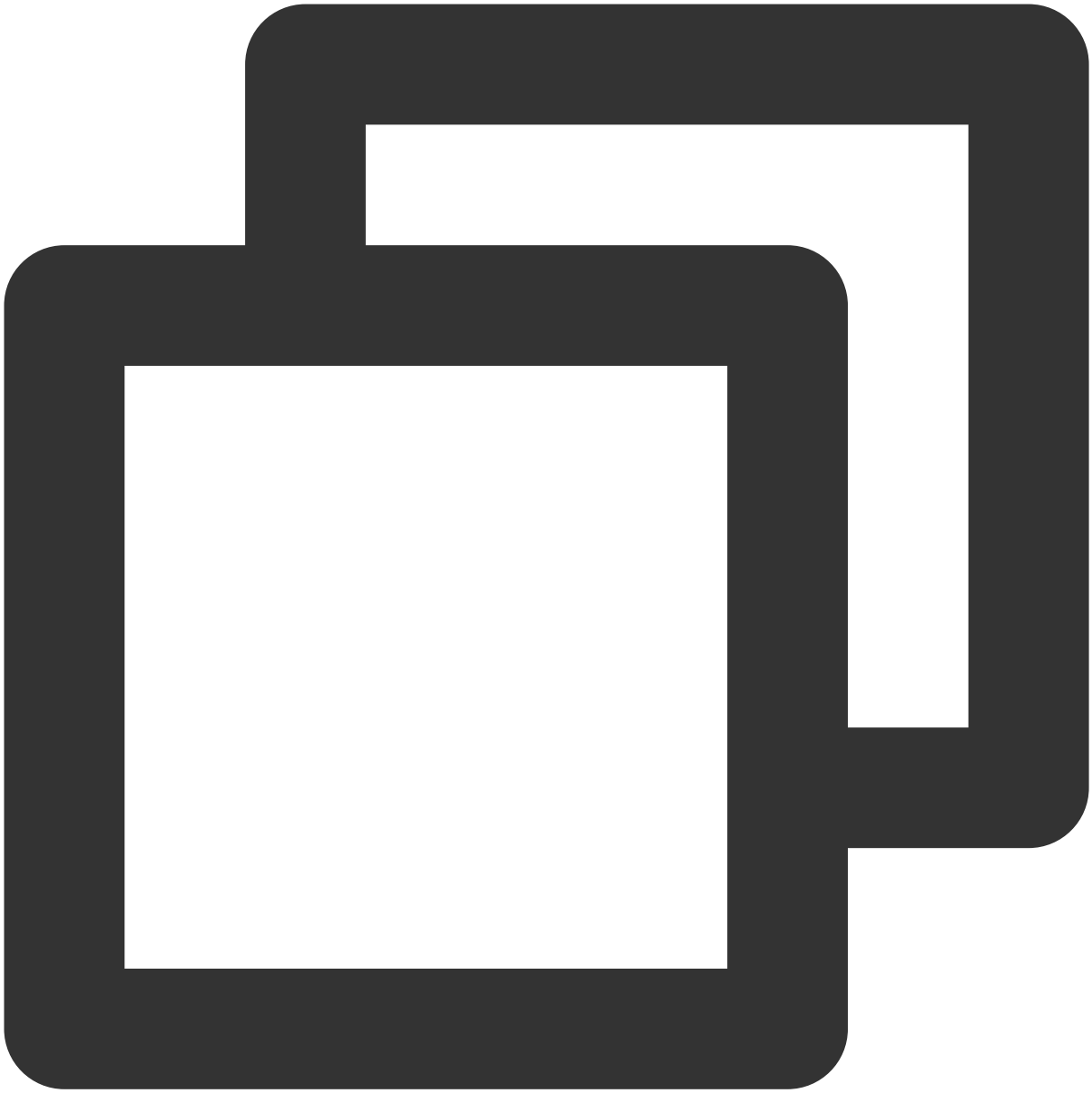


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Watermark</Tag>
    <Name>TemplateName</Name>
    <Watermark>
        <Type>Text</Type>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <LocMode>Absolute</LocMode>
        <Dx>128</Dx>
        <Dy>128</Dy>
        <Pos>TopRight</Pos>
        <StartTime>0</StartTime>
        <EndTime>100.5</EndTime>
        <Text>
            <Text>水印内容</Text>
            <FontSize>30</FontSize>
            <FontType></FontType>
            <FontColor>0xRRGGBB</FontColor>
            <Transparency>30</Transparency>
        </Text>
    </Watermark>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Tag>Watermark</Tag>
        <Name>TemplateName</Name>
        <Watermark>
            <Type>Text</Type>
            <LocMode>Absolute</LocMode>
            <Dx>128</Dx>
            <Dy>128</Dy>
            <Pos>TopRight</Pos>
            <StartTime>0</StartTime>
            <EndTime>100.5</EndTime>
            <Text>
                <Text>水印内容</Text>
                <FontSize>30</FontSize>
                <FontType></FontType>
                <FontColor>0xRRGGBB</FontColor>
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新智能封面模板

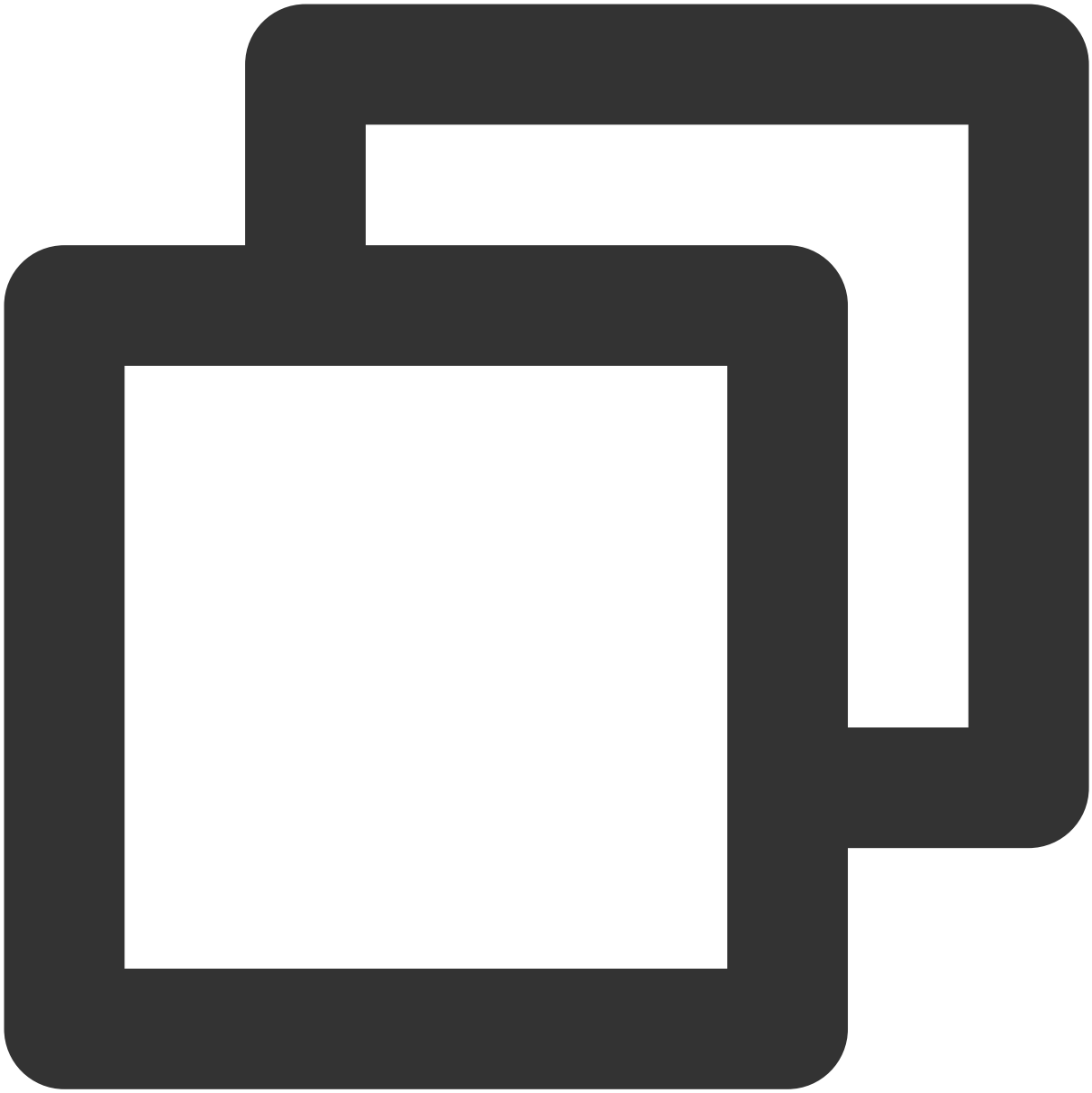
最近更新时间：2023-12-30 16:17:53

功能描述

更新智能封面模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

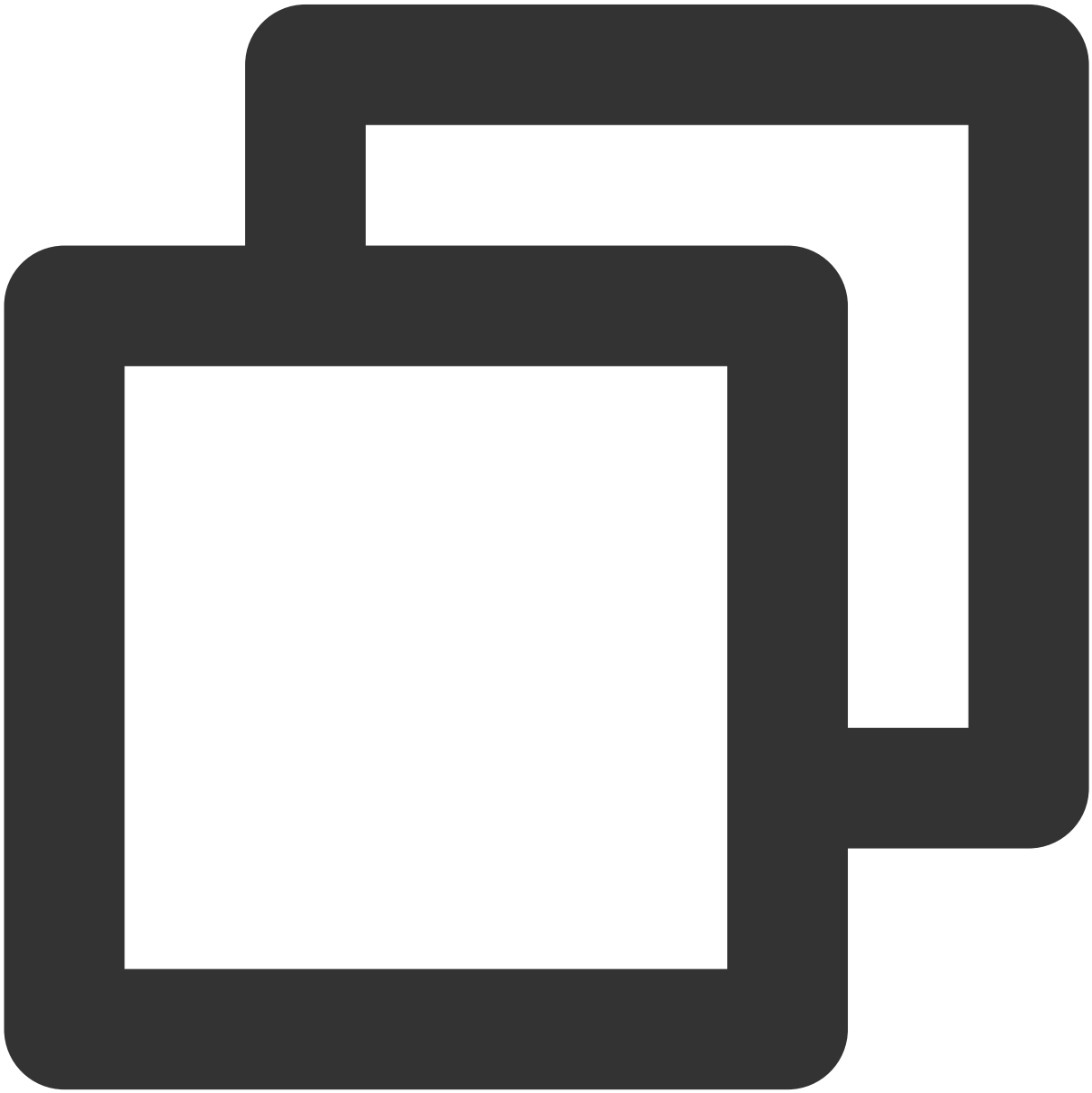
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>SmartCover</Tag>
    <Name>TemplateName</Name>
    <SmartCover>
        <Format>jpg</Format>
        <Width>1280</Width>
        <Height>960</Height>
        <Count>10</Count>
        <DeleteDuplicates>true</DeleteDuplicates>
    </SmartCover>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建智能封面模板接口的 Request	Container	是

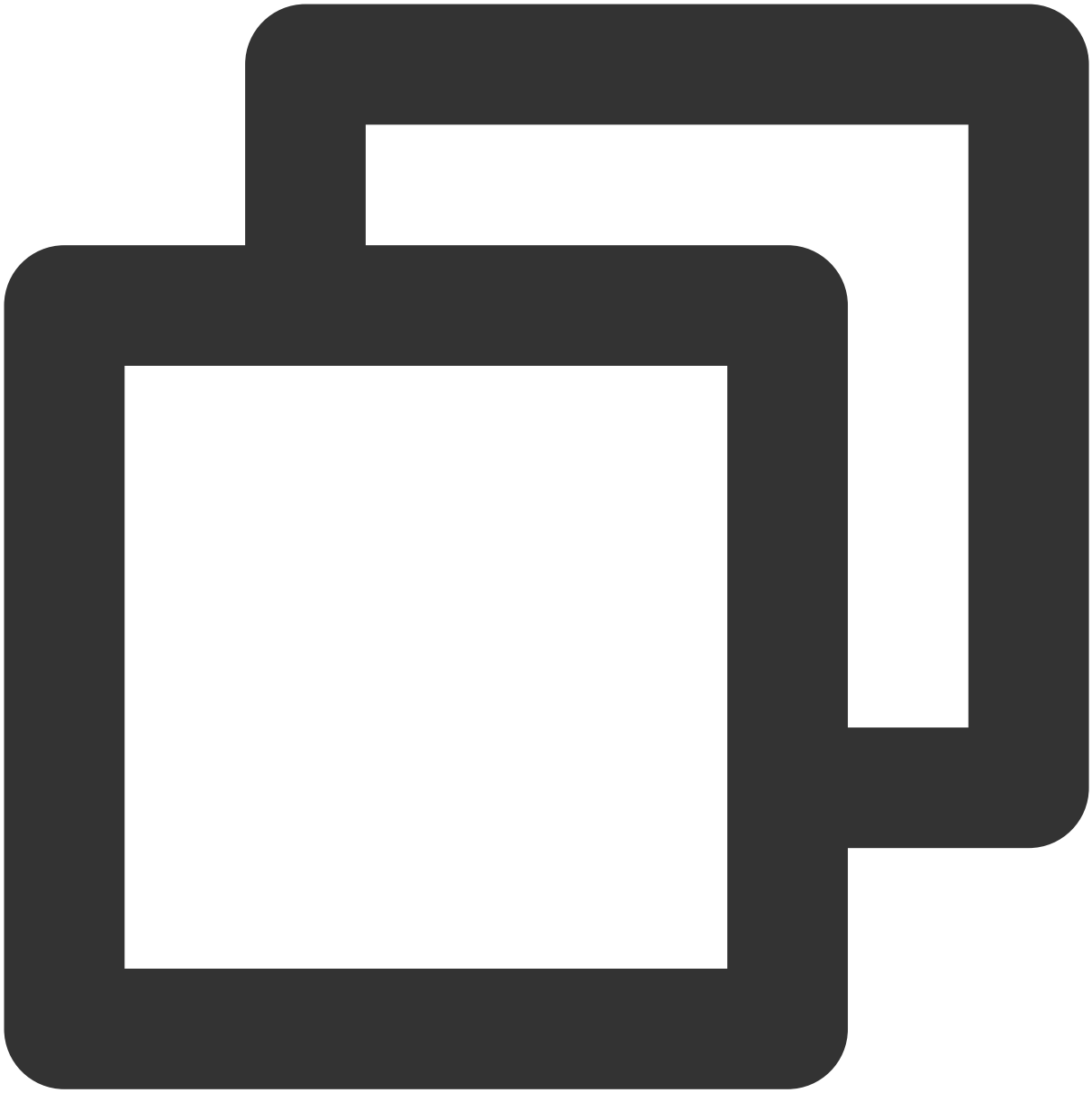
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>SmartCover</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f43110a08702470b8fa75894629501fa</TemplateId>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SmartCover>
            <Width>1280</Width>
            <Height>960</Height>
            <Format>jpg</Format>
```

```
            <Count>5</Count>
            <DeleteDuplicates>false</DeleteDuplicates>
        </SmartCover>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

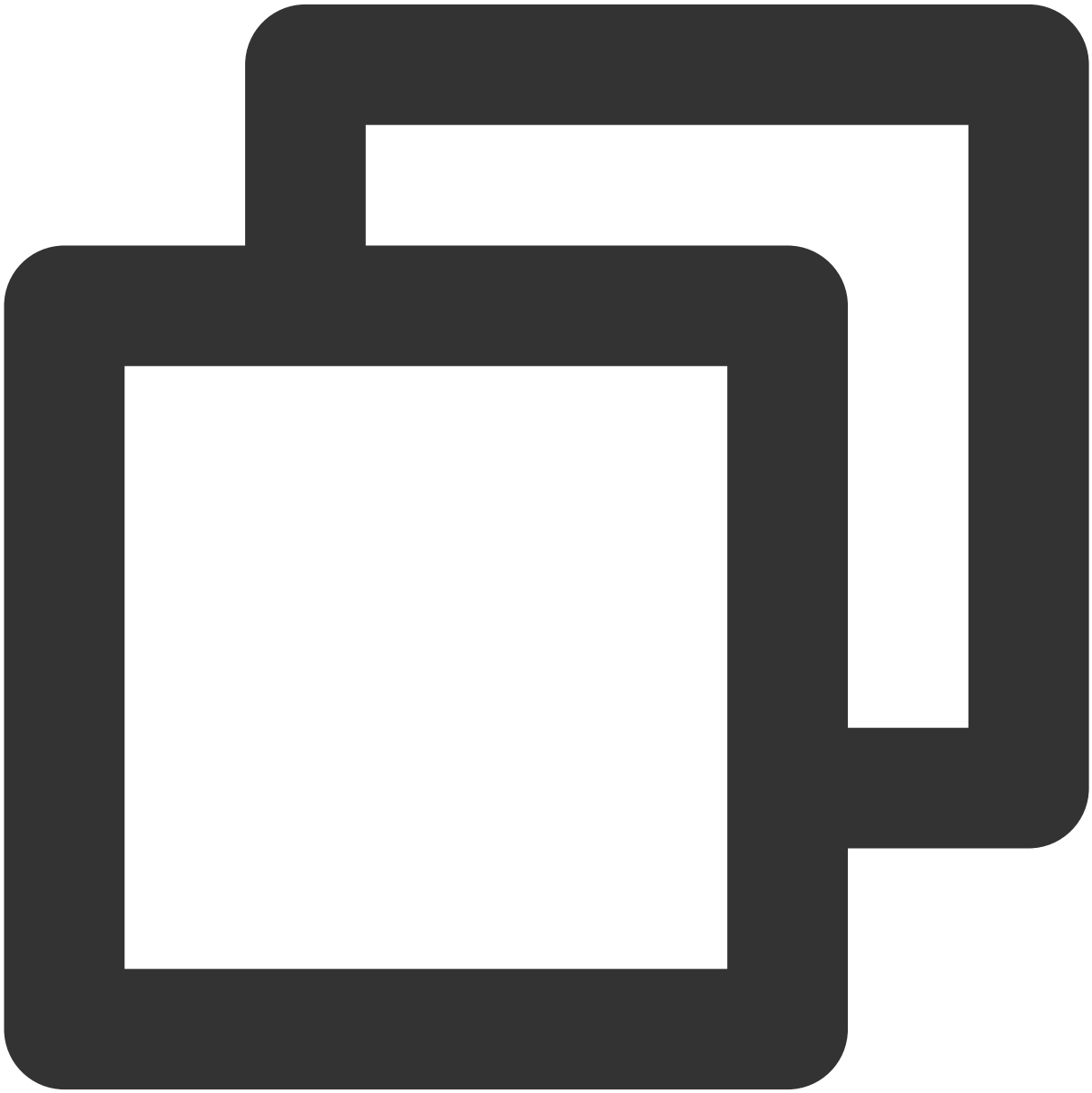
节点名称（关键字）	父节点	描述	类型
Response	无	同创建智能封面模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

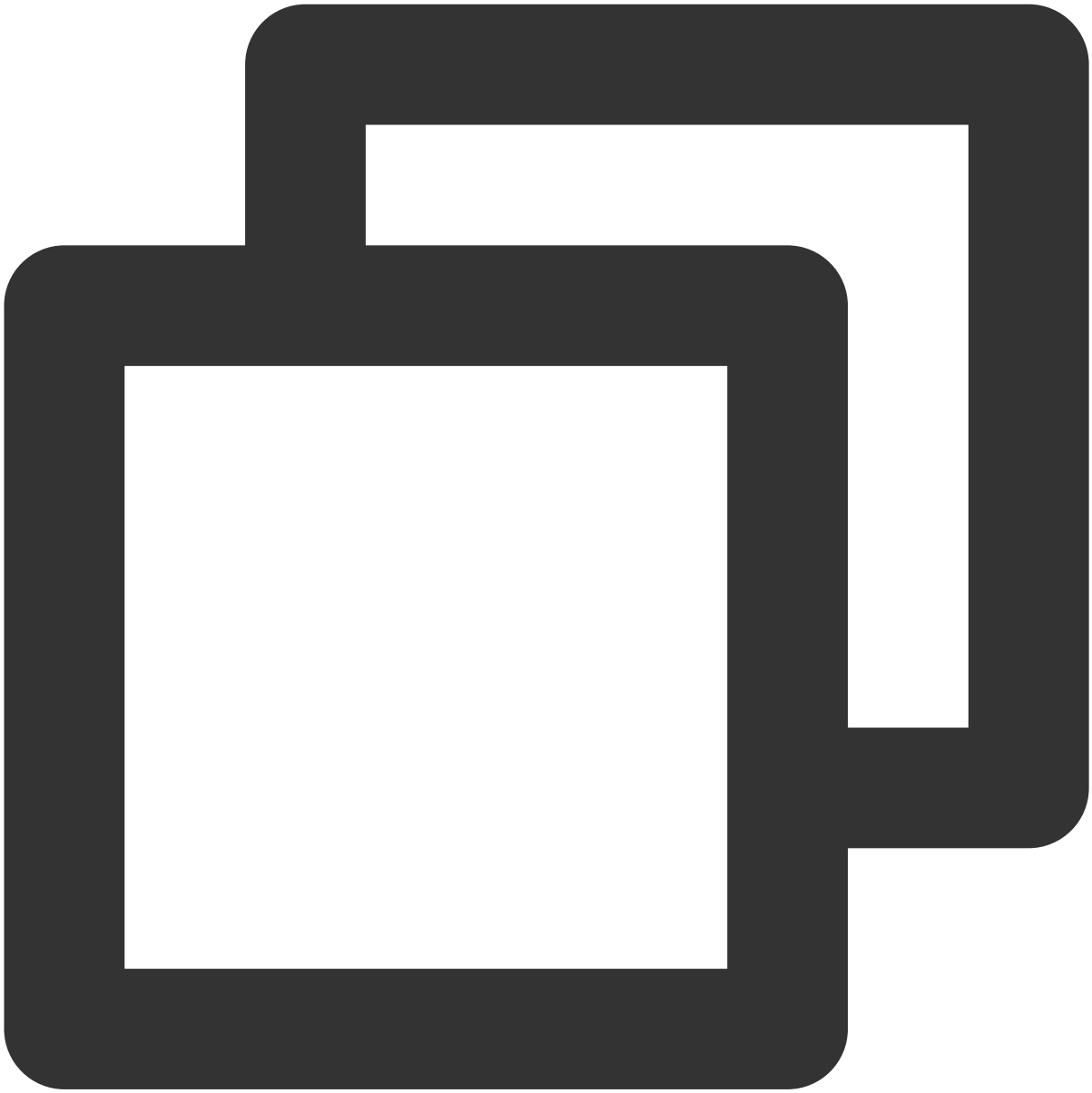
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>SmartCover</Tag>
    <Name>TemplateName</Name>
    <SmartCover>
        <Format>jpg</Format>
```

```
        <Width>1280</Width>
        <Height>960</Height>
        <Count>10</Count>
        <DeleteDuplicates>true</DeleteDuplicates>
    </SmartCover>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****



<Response>
    <Template>
        <Tag>SmartCover</Tag>
        <Name>TemplateName</Name>
        <TemplateId>t1f43110a08702470b8fa75894629501fa</TemplateId>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SmartCover>
            <Width>1280</Width>
            <Height>960</Height>
            <Format>jpg</Format>
            <Count>5</Count>
            <DeleteDuplicates>false</DeleteDuplicates>
        </SmartCover>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

AI 内容识别

创建 AI 内容识别模板

创建语音识别模板

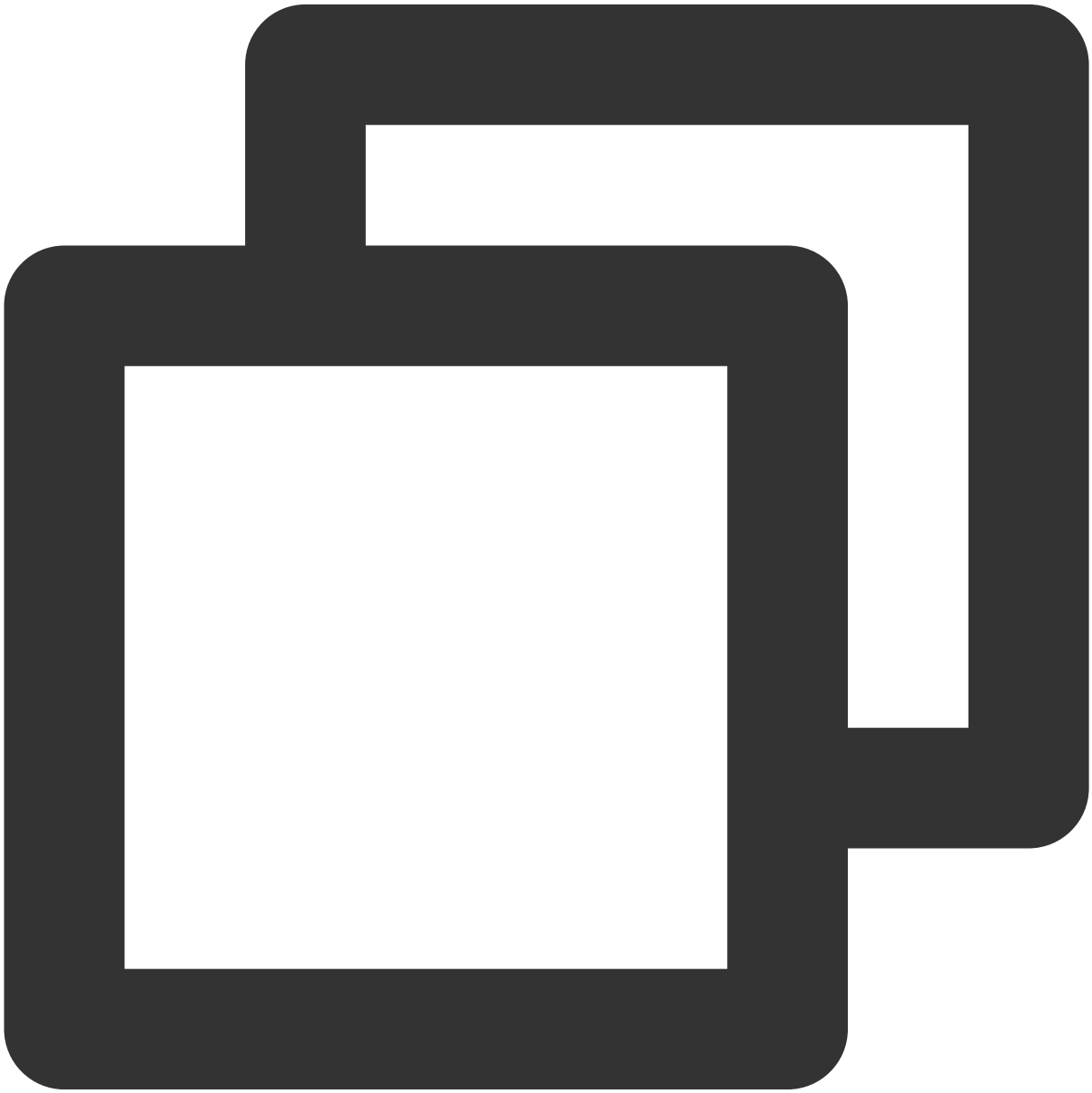
最近更新时间：2023-12-30 16:19:02

功能描述

创建语音识别模板。

请求

请求示例



```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

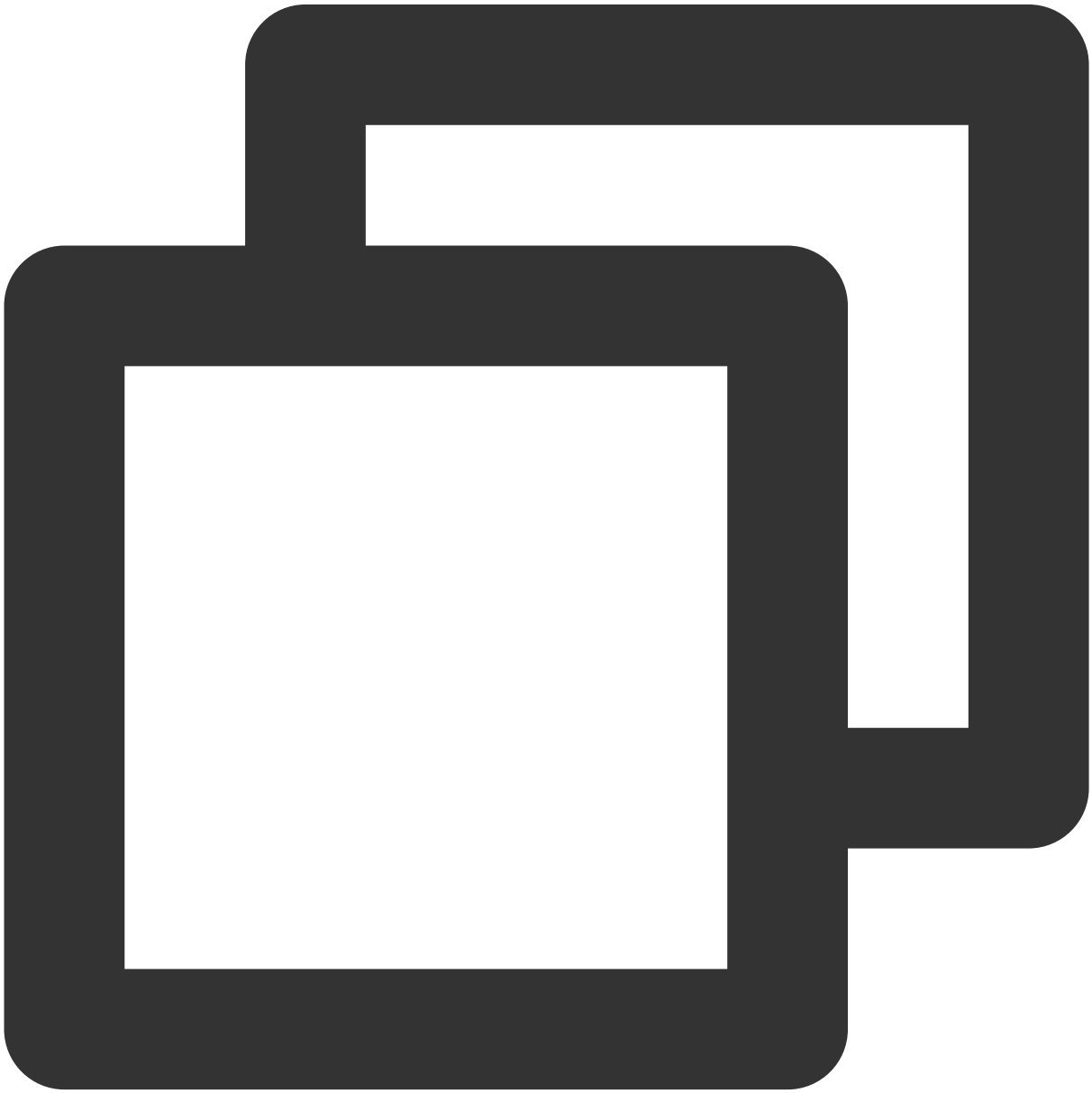
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见公共请求头部文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>SpeechRecognition</Tag>
    <Name>TemplateName</Name>
    <SpeechRecognition>
        <EngineModelType>16k_zh</EngineModelType>
        <ResTextFormat>1</ResTextFormat>
        <FilterDirty>0</FilterDirty>
        <FilterModal>1</FilterModal>
        <ConvertNumMode>0</ConvertNumMode>
        <SpeakerDiarization>1</SpeakerDiarization>
        <SpeakerNumber>0</SpeakerNumber>
        <FilterPunc>0</FilterPunc>
        <OutputFileType>txt</OutputFileType>
    </SpeechRecognition>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Tag	Request	模板类型：SpeechRecognition。	String	是	无
Name	Request	模板名称 仅支持中文、英文、数字、_、-和*。	String	是	无
SpeechRecognition	Request	语音识别参数。	Container	是	无

Container 类型 SpeechRecognition 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
EngineModelType	Request.SpeechRecognition	引擎模型类型，分为电话场景和非电话场景。 电话场景： 8k_zh：电话 8k 中文普通话通用（可用于双声道音频）。	String	是

		8k_zh_s：电话 8k 中文普通话话者分离（仅适用于单声道音频）。 8k_en：电话 8k 英语。 非电话场景： 16k_zh：16k 中文普通话通用。 16k_zh_video：16k 音视频领域。 16k_en：16k 英语。 16k_ca：16k 粤语。 16k_ja：16k 日语。 16k_zh_edu：中文教育。 16k_en_edu：英文教育。 16k_zh_medical：医疗。 16k_th：泰语。 16k_zh_dialect：多方言，支持23种方言。		
ChannelNum	Request.SpeechRecognition	语音声道数： 1表示单声道.EngineModelType为非电话场景仅支持单声道。 2表示双声道（仅支持 8k_zh 引擎模型双声道应分别对应通话双方）。	Integer	是
ResTextFormat	Request.SpeechRecognition	识别结果返回形式： 0表示识别结果文本（含分段时间戳）。 1词级别粒度的详细识别结果，不含标点，含语速值（词时间戳列表，一般用于生成字幕场景）。 2词级别粒度的详细识别结果（包含标点、语速值）。	Integer	是
FilterDirty	Request.SpeechRecognition	是否过滤脏词（目前支持中文普通话引擎）： 0表示不过滤脏词。 1表示过滤脏词。 2表示将脏词替换为*。 默认值为0。	Integer	否
FilterModal	Request.SpeechRecognition	是否过语气词（目前支持中文普通话引擎）： 0表示不过滤语气词。 1表示部分过滤。 2表示严格过滤 。	Integer	否

版权所有：腾讯云计算（北京）有限责任公司

		默认值为0。		
ConvertNumMode	Request.SpeechRecognition	是否进行阿拉伯数字智能转换（目前支持中文普通话引擎）： 0 表示不转换，直接输出中文数字。 1表示根据场景智能转换为阿拉伯数字。 3表示打开数学相关数字转换。 默认值为0。	Integer	否
SpeakerDiarization	Request.SpeechRecognition	是否开启说话人分离： 0表示不开启。 1表示开启(仅支持8k_zh，16k_zh，16k_zh_video，单声道音频)。 默认值为0。 注意：8k电话场景建议使用双声道来区分通话双方，设置ChannelNum=2即可，不用开启说话人分离。	Integer	否
SpeakerNumber	Request.SpeechRecognition	说话人分离人数（需配合开启说话人分离使用），取值范围：0-10。 0代表自动分离（目前仅支持≤6个人），1-10代表指定说话人数分离。 默认值为 0。	Integer	否
FilterPunc	Request.SpeechRecognition	是否过滤标点符号（目前支持中文普通话引擎）： 0 表示不过滤。 1 表示过滤句末标点。 2 表示过滤所有标点。 默认值为 0。	Integer	否
OutputFileType	Request.SpeechRecognition	输出文件类型，可选 txt、srt。默认为 txt。	String	否

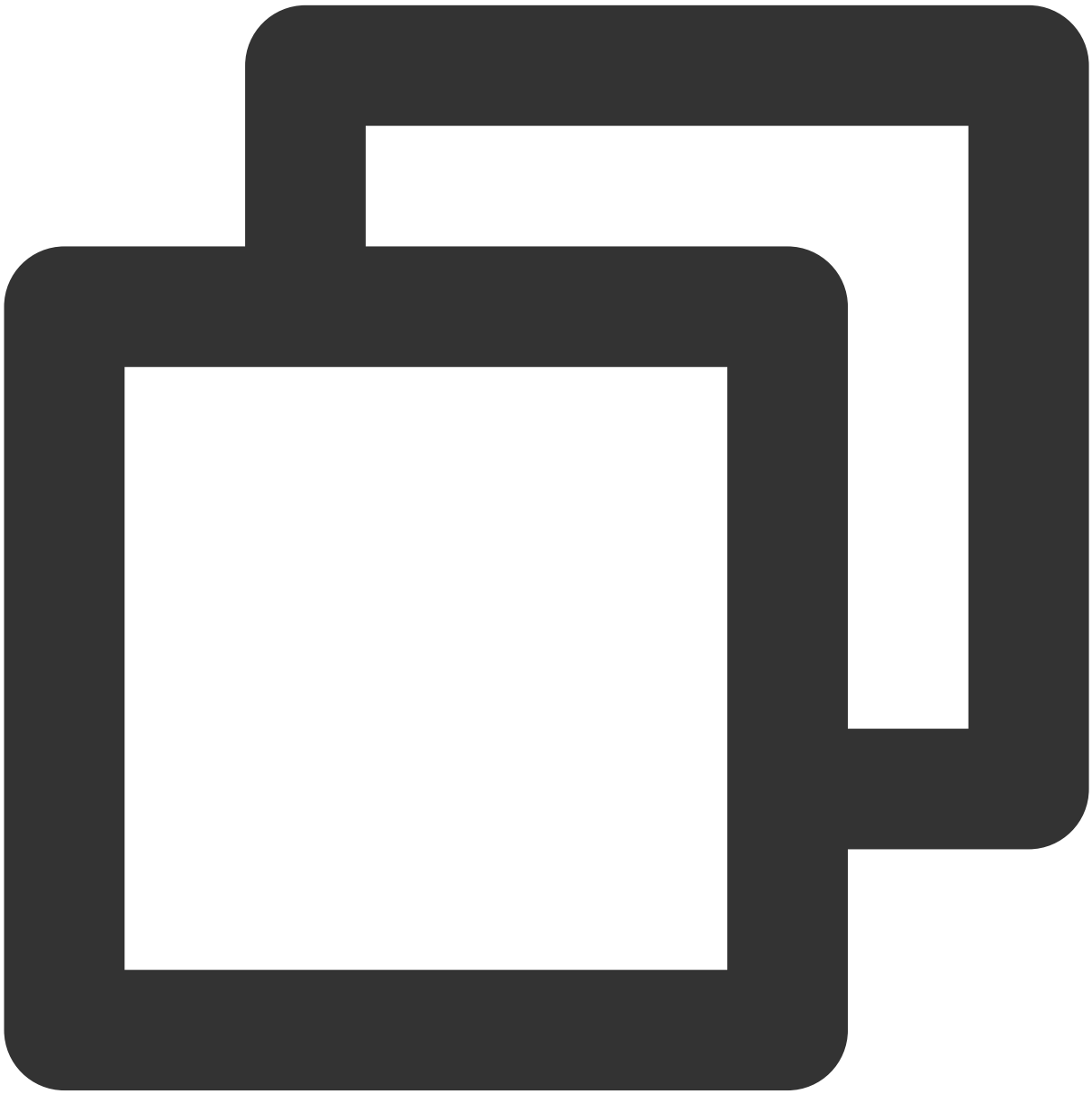
响应

响应头

此接口仅返回公共响应头部，详情请参见公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <Tag>SpeechRecognition</Tag>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>SpeechRecognition</Tag>
        <CreateTime></CreateTime>
        <UpdateTime></UpdateTime>
        <BucketId></BucketId>
```

```
        <Category>Custom</Category>
        <SpeechRecognition>
            <EngineModelType>16k_zh</EngineModelType>
            <ResTextFormat>1</ResTextFormat>
            <FilterDirty>0</FilterDirty>
            <FilterModal>1</FilterModal>
            <ConvertNumMode>0</ConvertNumMode>
            <SpeakerDiarization>1</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
            <OutputFileType>txt</OutputFileType>
        </SpeechRecognition>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID。	String
Name	Response.Template	模板名称。	String
BucketId	Response.Template	模板所属存储桶。	String
Category	Response.Template	模板属性，Custom 或者 Official。	String
Tag	Response.Template	模板类型，SpeechRecognition。	String
UpdateTime	Response.Template	更新时间。	String
CreateTime	Response.Template	创建时间。	String
SpeechRecognition	Response.Template	同请求体中的 Request.SpeechRecognition。	Container

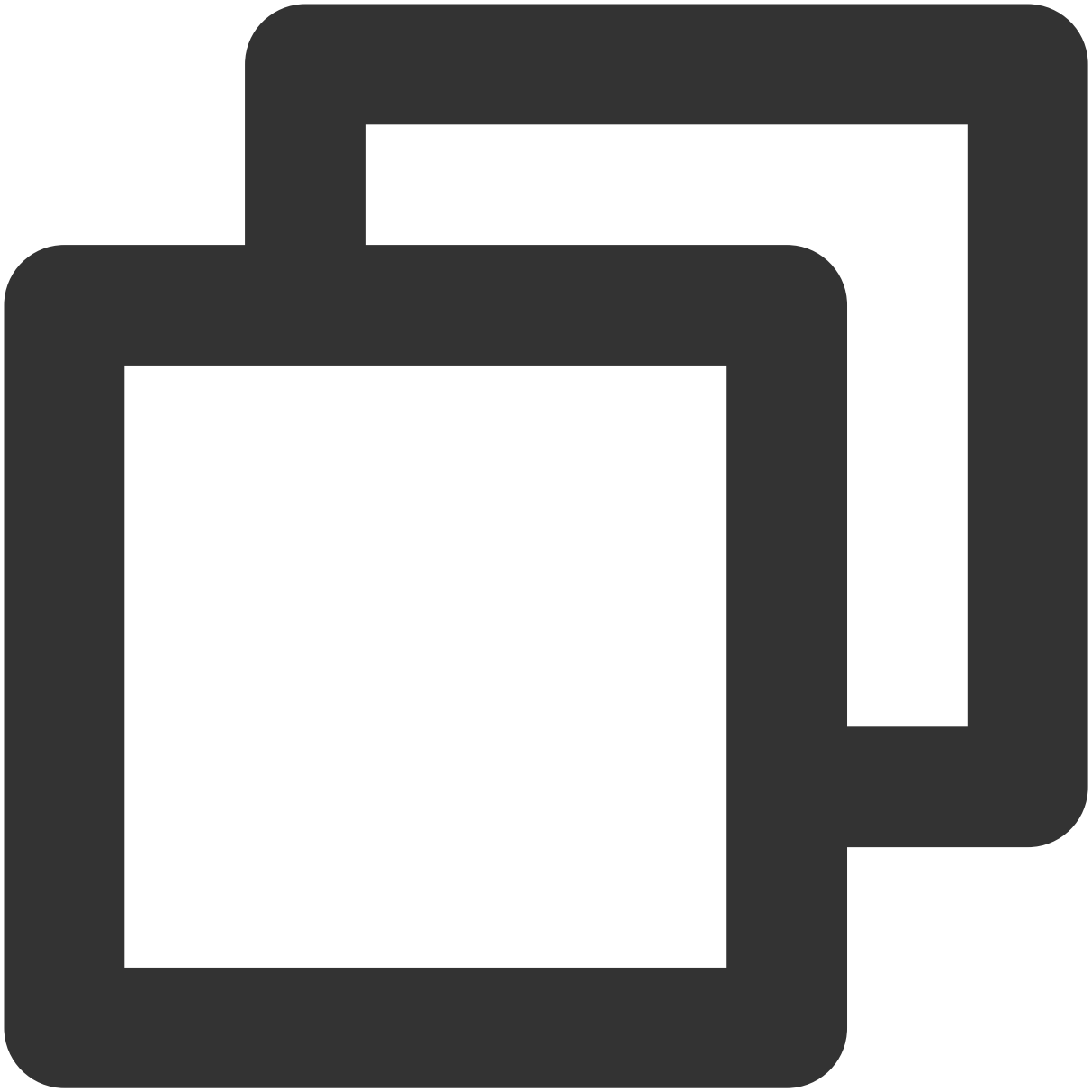
错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

版权所有：腾讯云计算（北京）有限责任公司

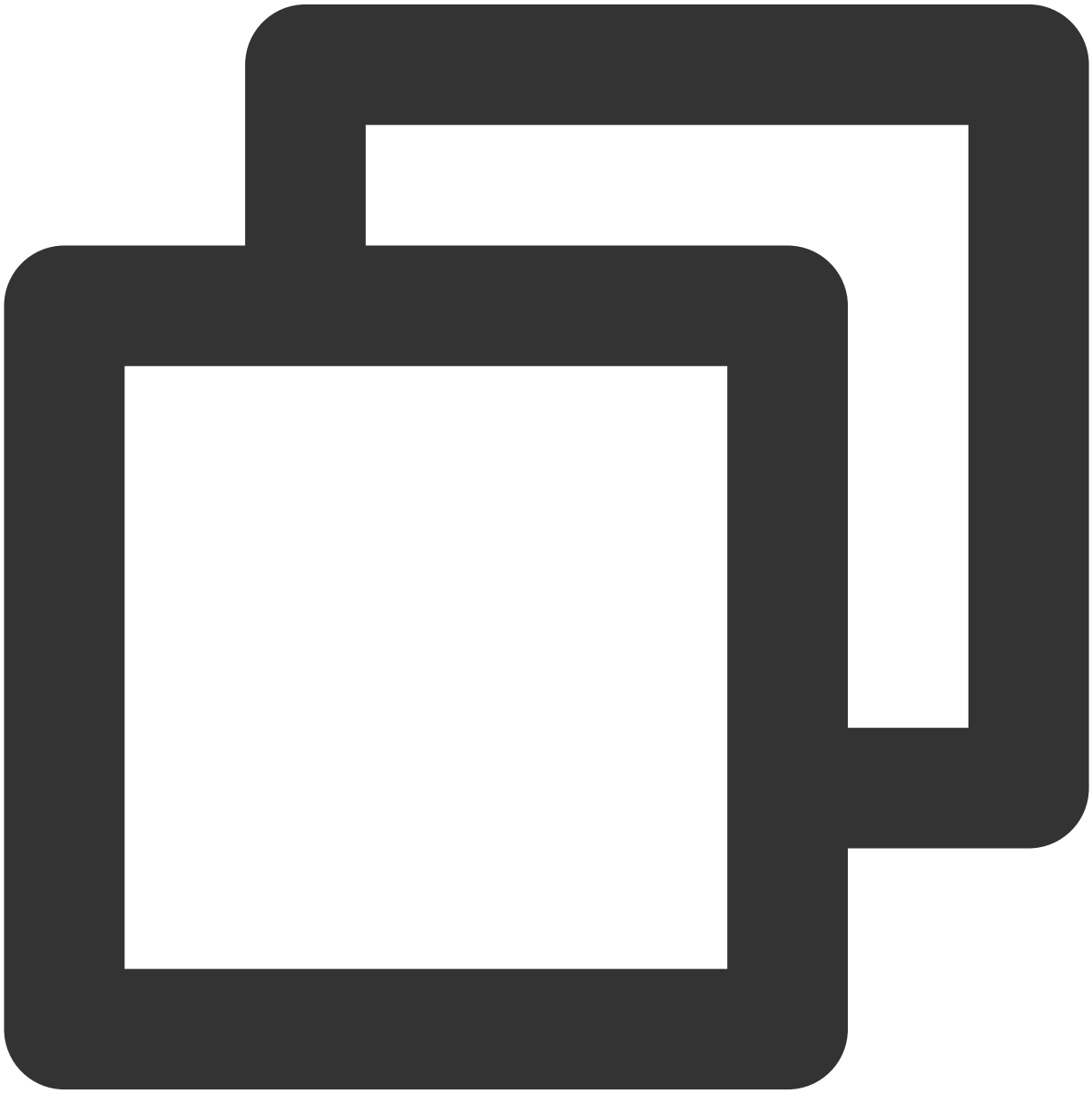
请求



```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>SpeechRecognition</Tag>
    <Name>TemplateName</Name>
```

```
    <SpeechRecognition>
        <EngineModelType>16k_zh</EngineModelType>
        <ResTextFormat>1</ResTextFormat>
        <FilterDirty>0</FilterDirty>
        <FilterModal>1</FilterModal>
        <ConvertNumMode>0</ConvertNumMode>
        <SpeakerDiarization>1</SpeakerDiarization>
        <SpeakerNumber>0</SpeakerNumber>
        <FilterPunc>0</FilterPunc>
        <OutputFileType>txt</OutputFileType>
    </SpeechRecognition>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>SpeechRecognition</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SpeechRecognition>
            <EngineModelType>16k_zh</EngineModelType>
            <ChannelNum>1</ChannelNum>
            <ResTextFormat>0</ResTextFormat>
            <FilterDirty>1</FilterDirty>
            <FilterModal>0</FilterModal>
            <ConvertNumMode>1</ConvertNumMode>
            <SpeakerDiarization>0</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
        </SpeechRecognition>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

查询 AI 内容识别模板
DescribeTemplates

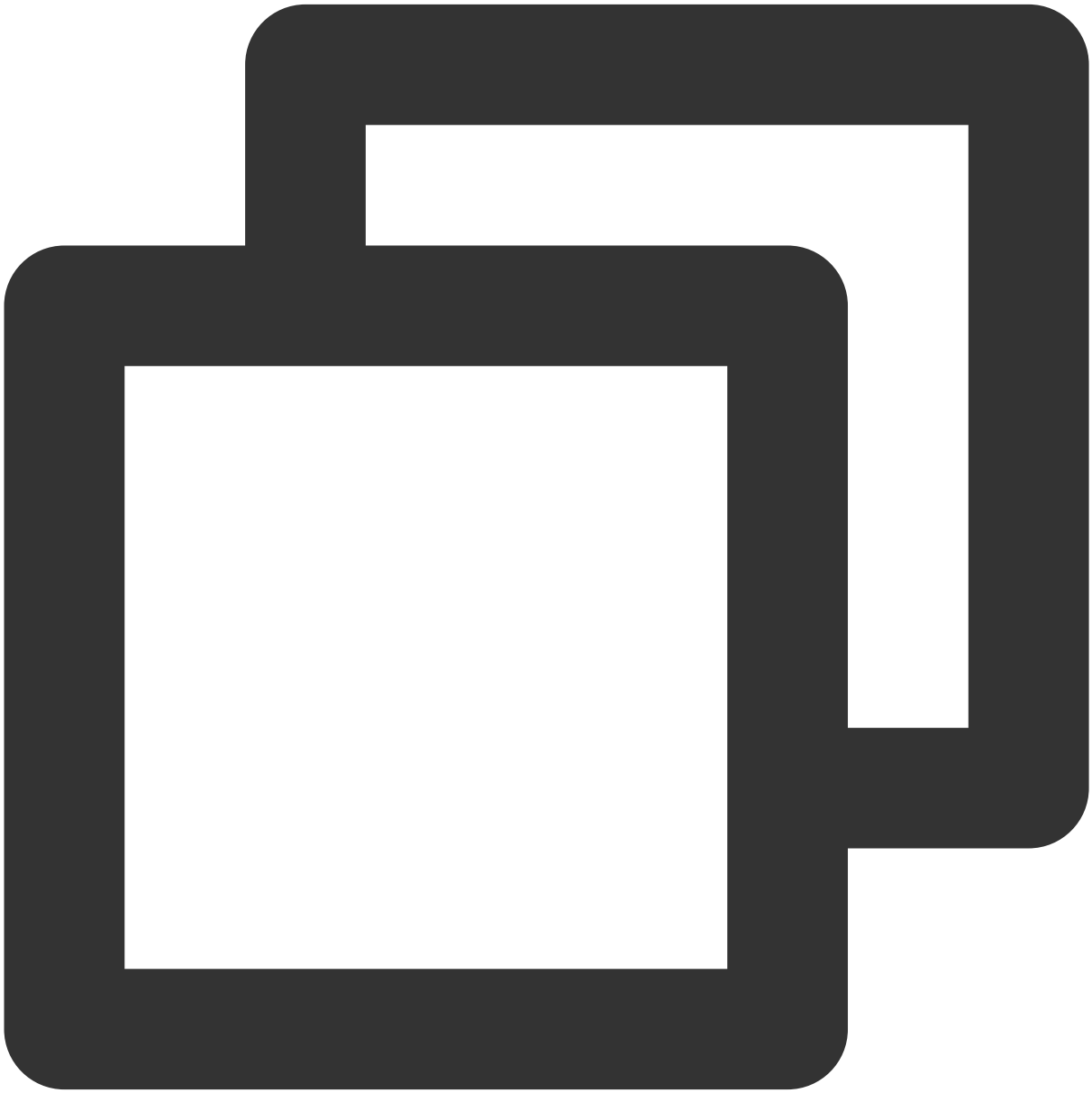
最近更新时间：2023-12-30 16:19:10

功能描述

查询模板。

请求

请求示例



```
GET /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	必选
tag	无	模板类型，默认值：All	String	否
category	无	Official（系统预设模板），Custom（自定义模板），默认值：Custom	String	否
ids	无	模板 ID，以 `,` 符号分割字符串	String	否
name	无	模板名称前缀	String	否
pageNumber	无	第几页，默认值：1	Integer	否
pageSize	无	每页个数，默认值：10	Integer	否

tag 支持以下几种类型:

模板类型	tag
语音识别	SpeechRecognition

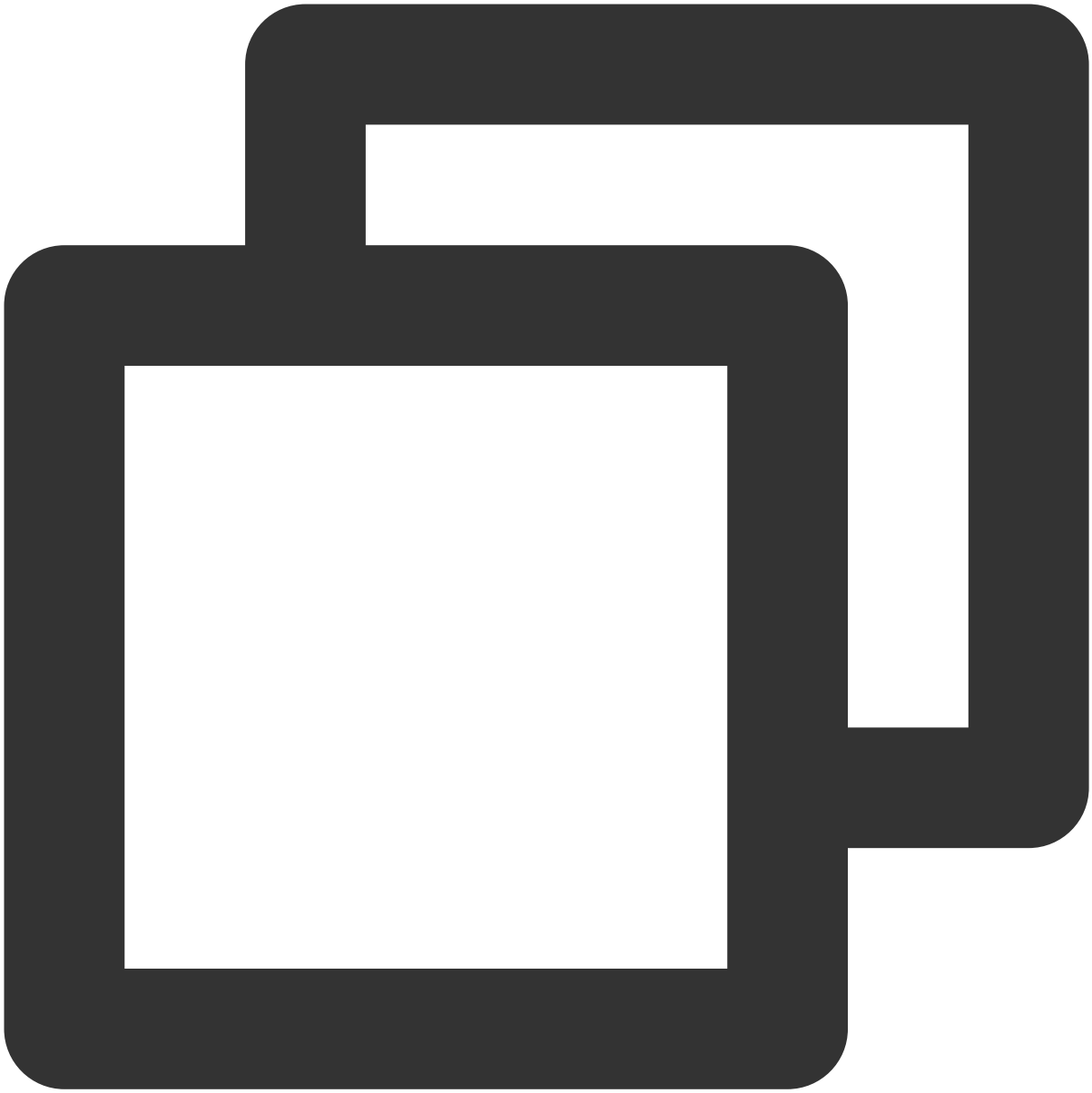
响应

响应头

此接口仅返回公共响应头部，详情请参见公共响应头部文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TotalCount>1</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <TemplateList>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <Tag>SpeechRecognition</Tag>
        <SpeechRecognition>
            <EngineModelType>16k_zh</EngineModelType>
```

```
            <ResTextFormat>1</ResTextFormat>
            <FilterDirty>0</FilterDirty>
            <FilterModal>1</FilterModal>
            <ConvertNumMode>0</ConvertNumMode>
            <SpeakerDiarization>1</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
            <OutputFileType>txt</OutputFileType>
        </SpeechRecognition>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	模板总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
TemplateList	Response	模板数组	Container 数组

对于不同的模板类型，TemplateList 内容同创建模板接口的 Response，请参照以下链接：
语音识别

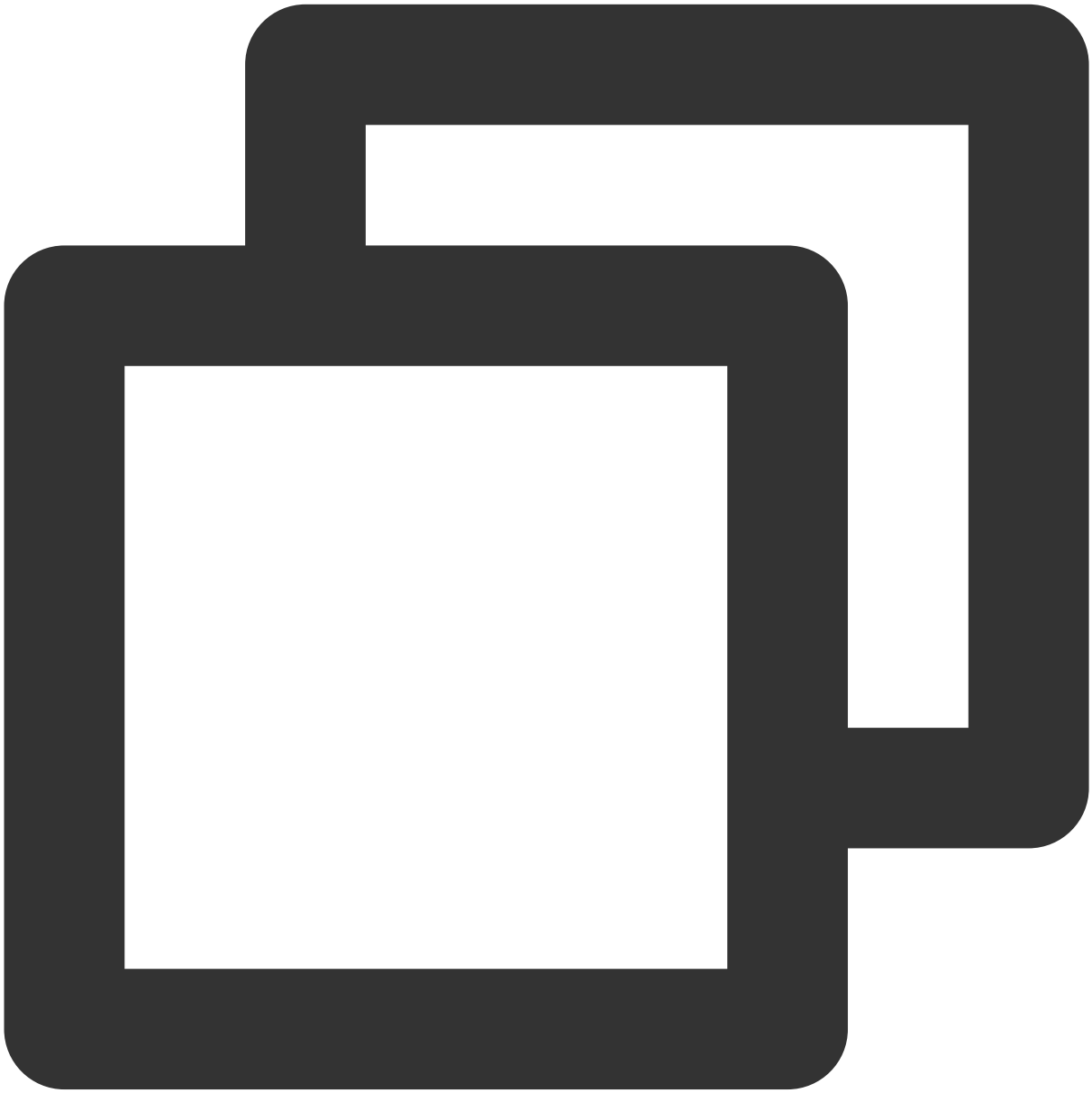
错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

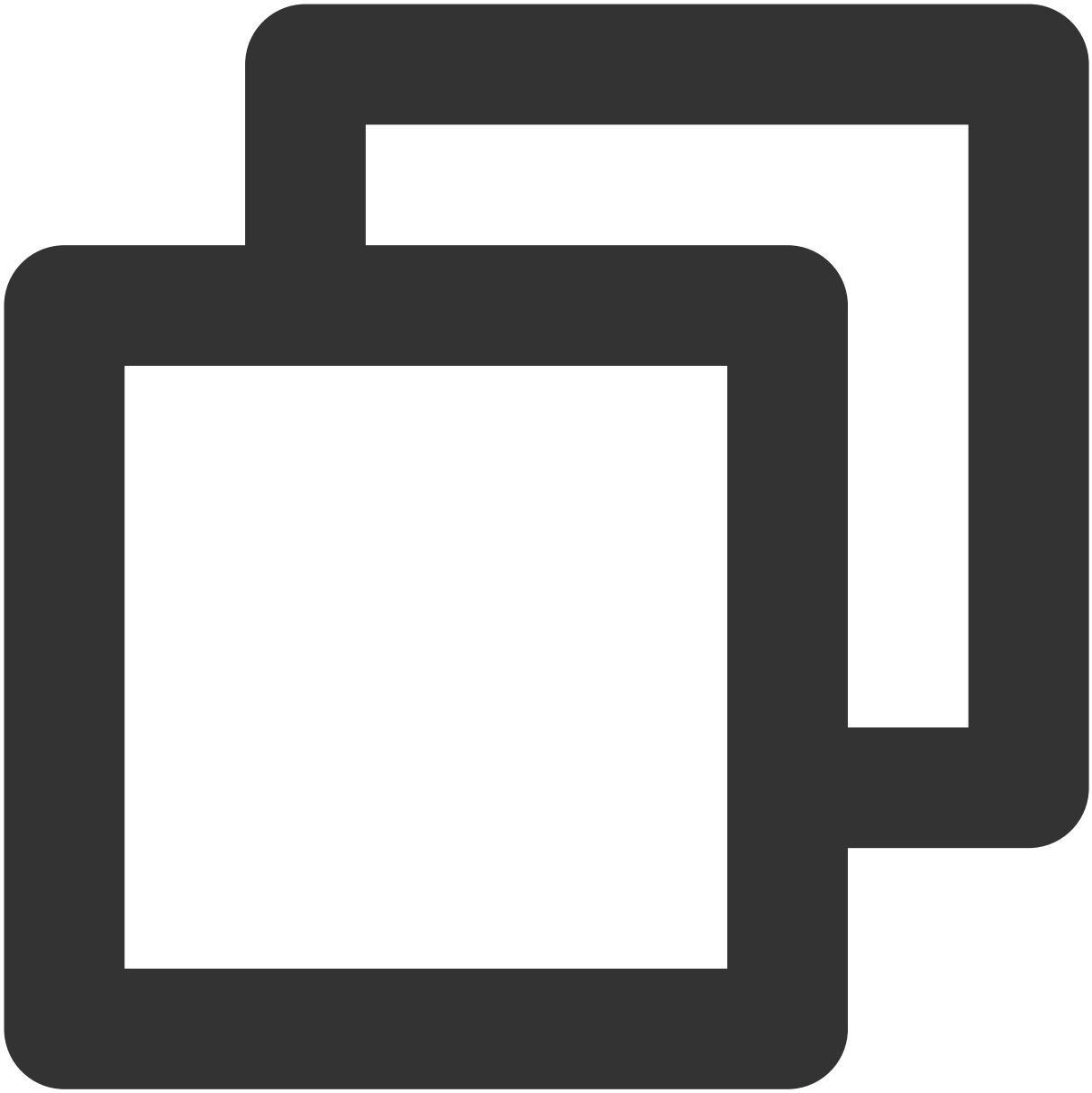
案例一：按照模板 ID 维度查询

请求



```
GET /template?ids=t1847cd4ca57f543e89f551dbe68169eb9,t1a30a323f55434a29b25bf2a16bdf
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应

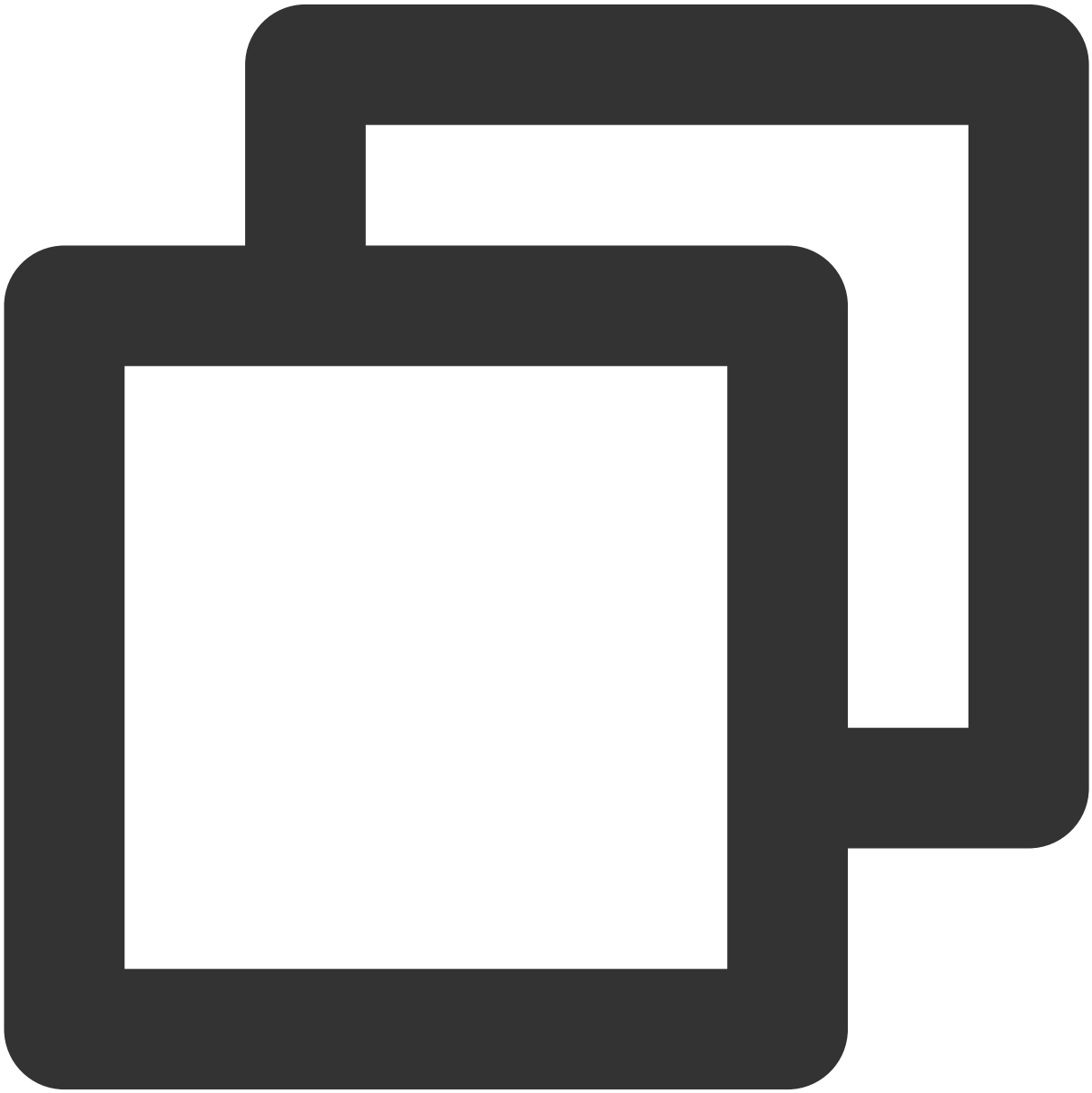


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TemplateList>
```

```
        <TemplateId>t1847cd4ca57f543e89f551dbe68169eb9</TemplateId>
        <Name>TemplateName</Name>
        <Tag>SpeechRecognition</Tag>
        <SpeechRecognition>
            <EngineModelType>16k_zh</EngineModelType>
            <ResTextFormat>1</ResTextFormat>
            <FilterDirty>0</FilterDirty>
            <FilterModal>1</FilterModal>
            <ConvertNumMode>0</ConvertNumMode>
            <SpeakerDiarization>1</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
            <OutputFileType>txt</OutputFileType>
        </SpeechRecognition>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
    <TemplateList>
        <TemplateId>t1a30a323f55434a29b25bf2a16bdf5f59</TemplateId>
        <Name>TemplateName</Name>
        <Tag>SpeechRecognition</Tag>
        <SpeechRecognition>
            <EngineModelType>8k_zh_s</EngineModelType>
            <ResTextFormat>1</ResTextFormat>
            <FilterDirty>0</FilterDirty>
            <FilterModal>1</FilterModal>
            <ConvertNumMode>0</ConvertNumMode>
            <SpeakerDiarization>1</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
            <OutputFileType>txt</OutputFileType>
        </SpeechRecognition>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
    <NonExistTIDs>
        <TemplateId>C</TemplateId>
    </NonExistTIDs>
</Response>
```

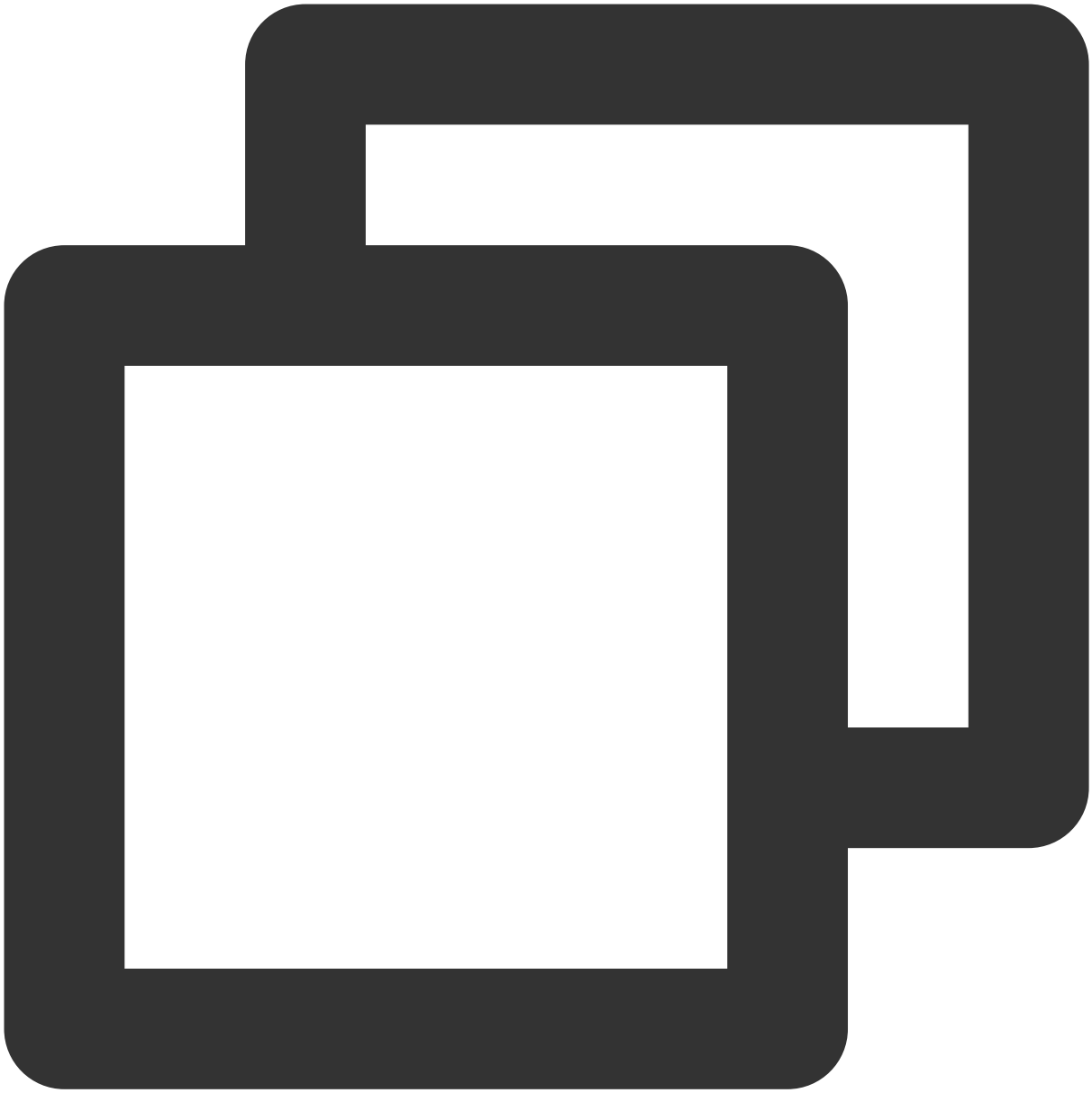
案例二：按照分页列表维度查询

请求



```
GET /template?pageSize=10&pageNumber=1 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TotalCount>2</TotalCount>
```

```
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <TemplateList>
        <TemplateId>t1847cd4ca57f543e89f551dbe68169eb9</TemplateId>
        <Name>TemplateName</Name>
        <Tag>SpeechRecognition</Tag>
        <SpeechRecognition>
            <EngineModelType>16k_zh</EngineModelType>
            <ResTextFormat>1</ResTextFormat>
            <FilterDirty>0</FilterDirty>
            <FilterModal>1</FilterModal>
            <ConvertNumMode>0</ConvertNumMode>
            <SpeakerDiarization>1</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
            <OutputFileType>txt</OutputFileType>
        </SpeechRecognition>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
    <TemplateList>
        <TemplateId>t1a30a323f55434a29b25bf2a16bdf5f59</TemplateId>
        <Name>TemplateName</Name>
        <Tag>SpeechRecognition</Tag>
        <SpeechRecognition>
            <EngineModelType>8k_zh_s</EngineModelType>
            <ResTextFormat>1</ResTextFormat>
            <FilterDirty>0</FilterDirty>
            <FilterModal>1</FilterModal>
            <ConvertNumMode>0</ConvertNumMode>
            <SpeakerDiarization>1</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
            <OutputFileType>txt</OutputFileType>
        </SpeechRecognition>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </TemplateList>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新 AI 内容识别模板
更新语音识别模板

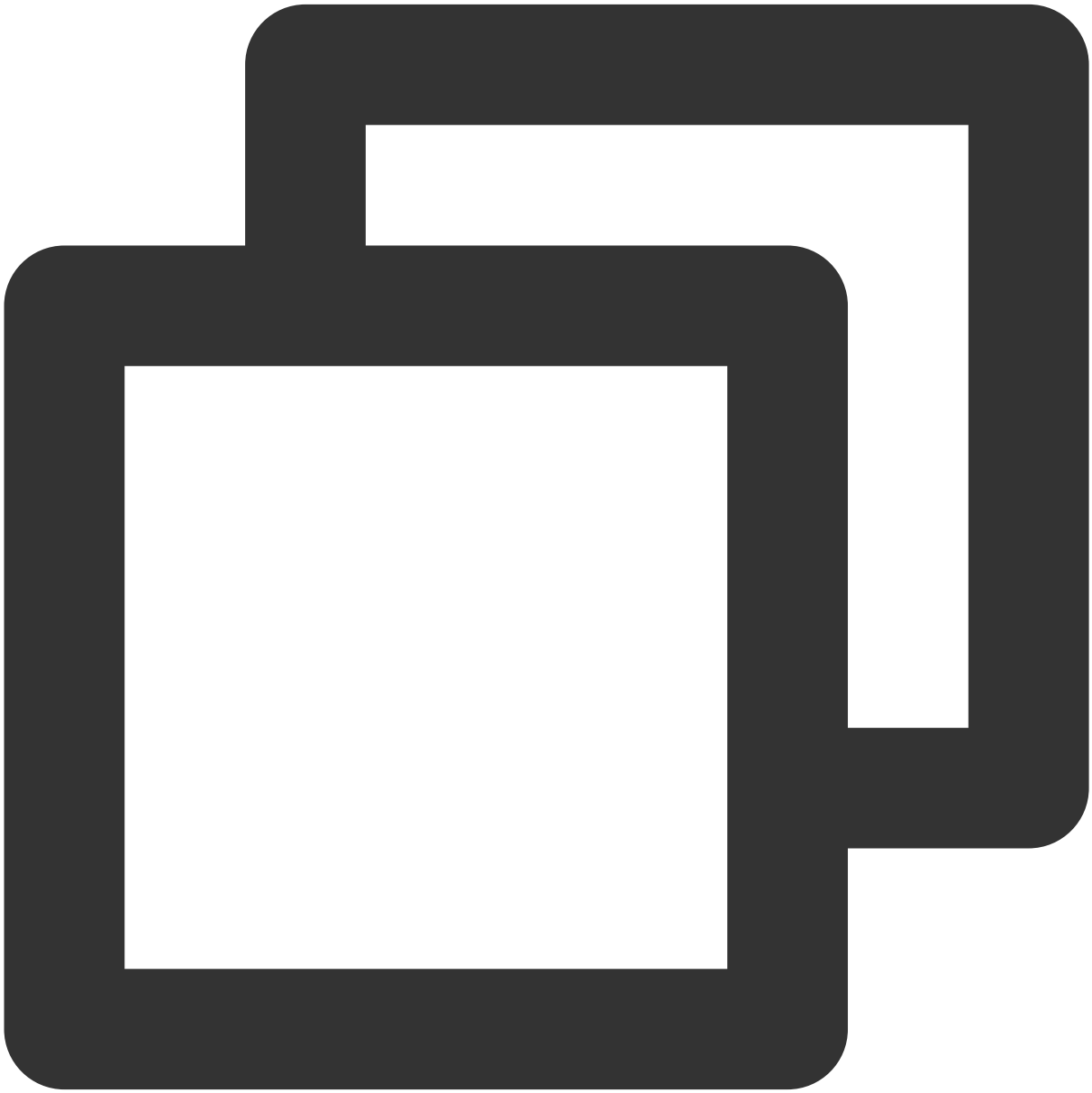
最近更新时间：2023-12-30 16:19:43

功能描述

更新语音识别模板。

请求

请求示例



```
PUT /template/<TemplateId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

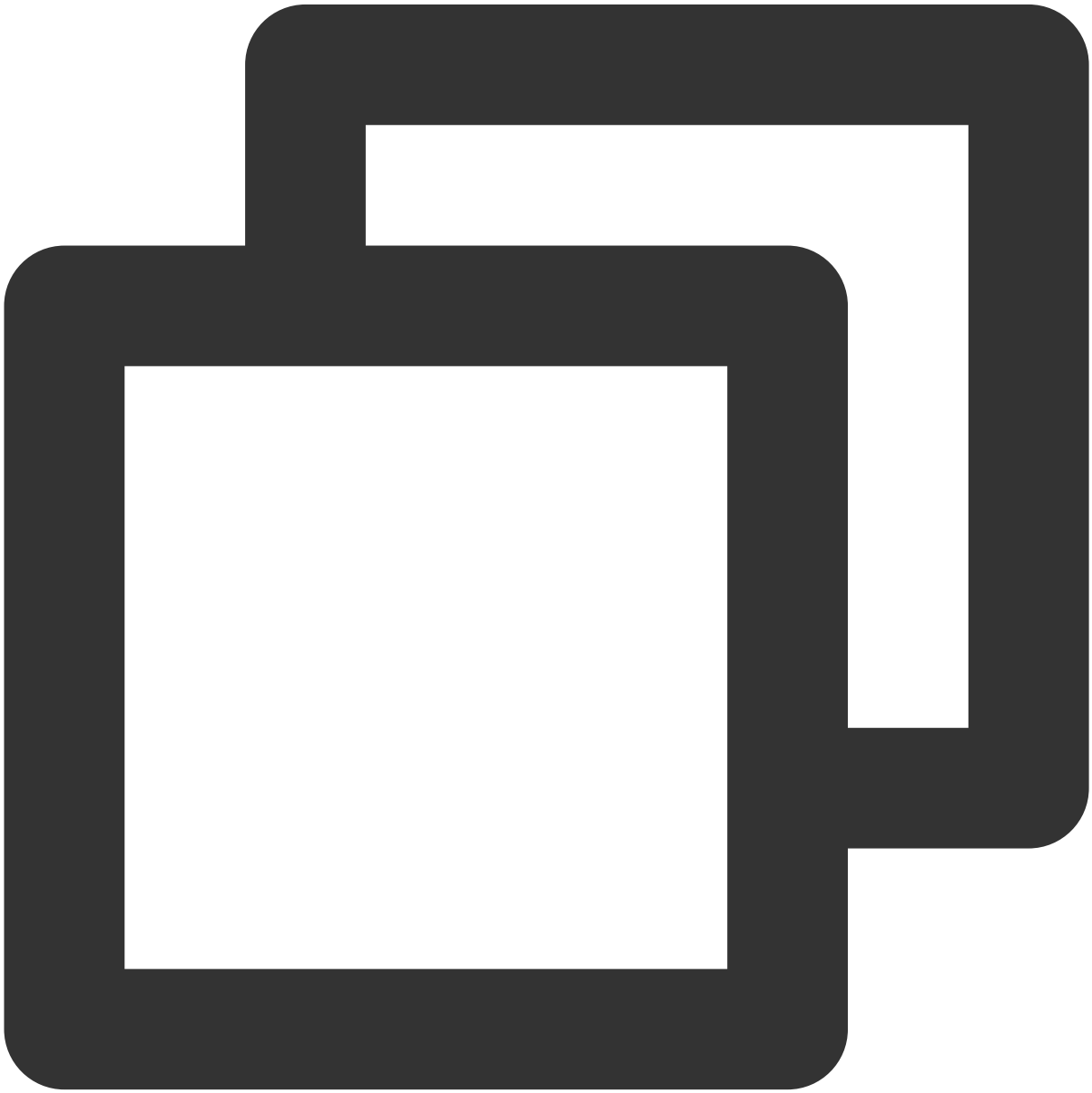
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

请求头

此接口仅使用公共请求头部，详情请参见公共请求头部文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Tag>SpeechRecognition</Tag>
    <Name>TemplateName</Name>
    <SpeechRecognition>
        <EngineModelType>16k_zh</EngineModelType>
        <ResTextFormat>1</ResTextFormat>
        <FilterDirty>0</FilterDirty>
        <FilterModal>1</FilterModal>
        <ConvertNumMode>0</ConvertNumMode>
        <SpeakerDiarization>1</SpeakerDiarization>
        <SpeakerNumber>0</SpeakerNumber>
        <FilterPunc>0</FilterPunc>
        <OutputFileType>txt</OutputFileType>
    </SpeechRecognition>
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Request	无	同创建语音识别模板接口的 Request	Container	是

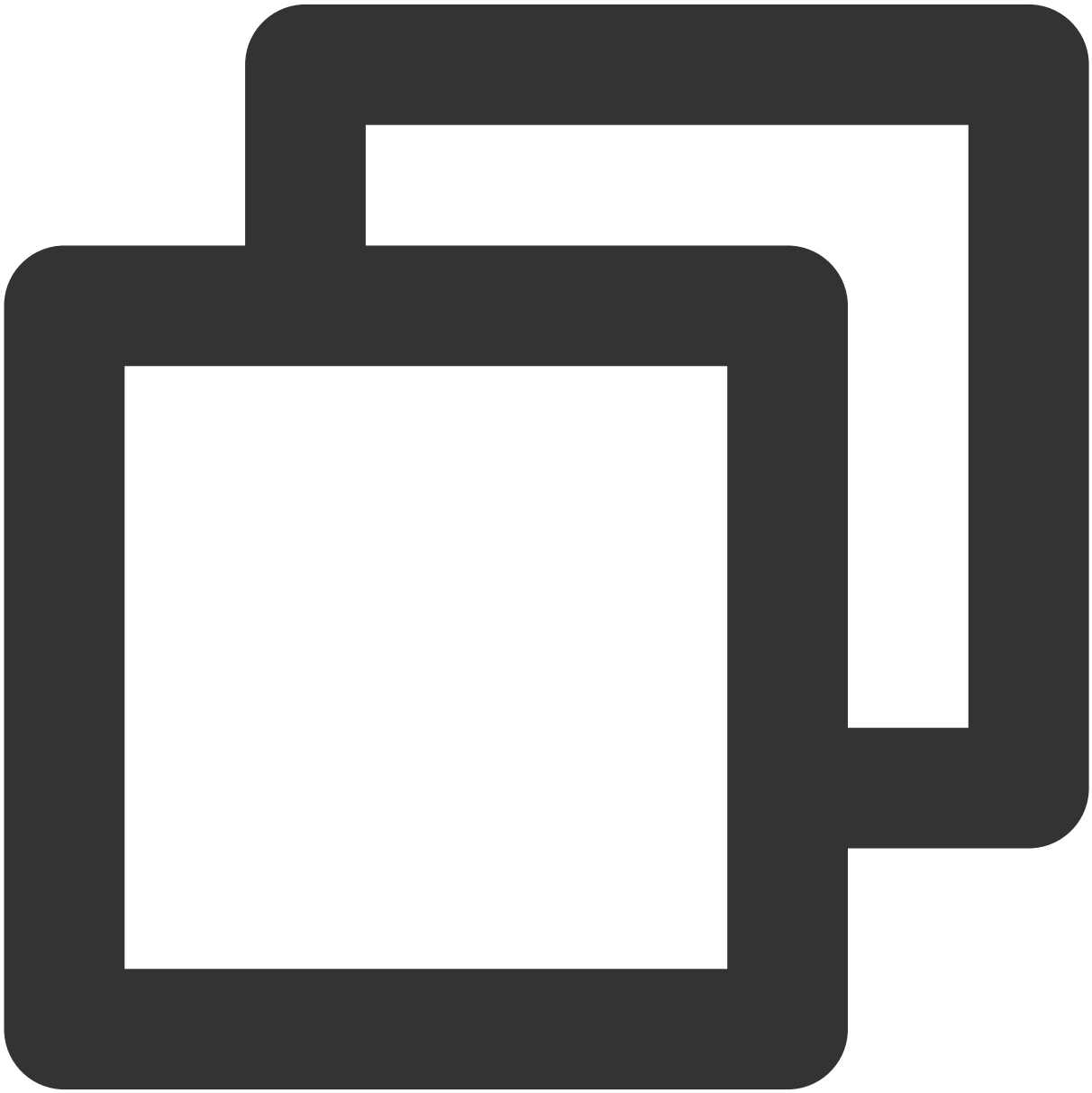
响应

响应头

此接口仅返回公共响应头部，详情请参见公共响应头部文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>SpeechRecognition</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SpeechRecognition>
```

```
            <EngineModelType>16k_zh</EngineModelType>
            <ChannelNum>1</ChannelNum>
            <ResTextFormat>0</ResTextFormat>
            <FilterDirty>1</FilterDirty>
            <FilterModal>0</FilterModal>
            <ConvertNumMode>1</ConvertNumMode>
            <SpeakerDiarization>0</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
        </SpeechRecognition>
    </Template>
</Response>
```

具体的数据内容如下：

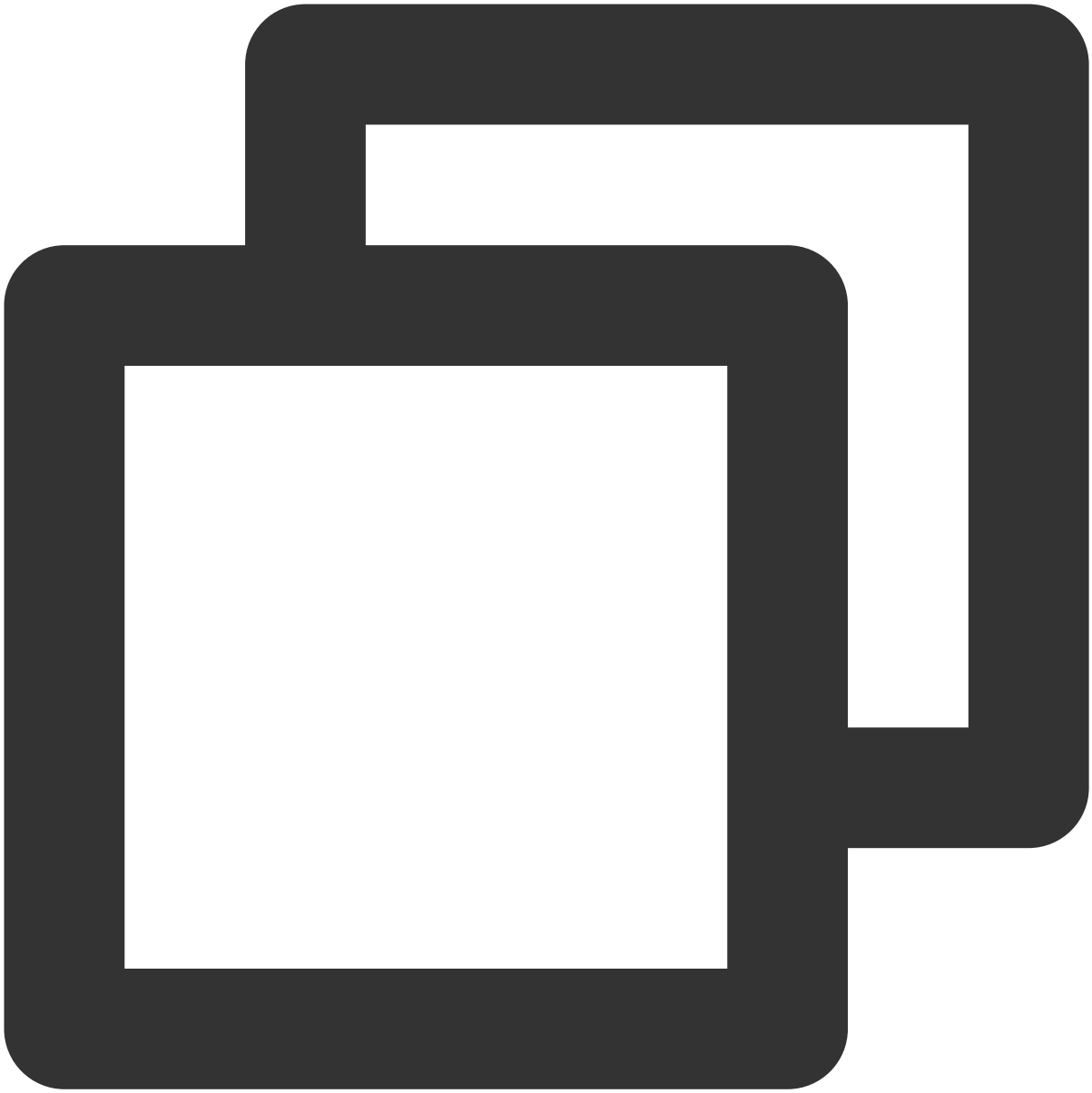
节点名称（关键字）	父节点	描述	类型
Response	无	同创建语音合成模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

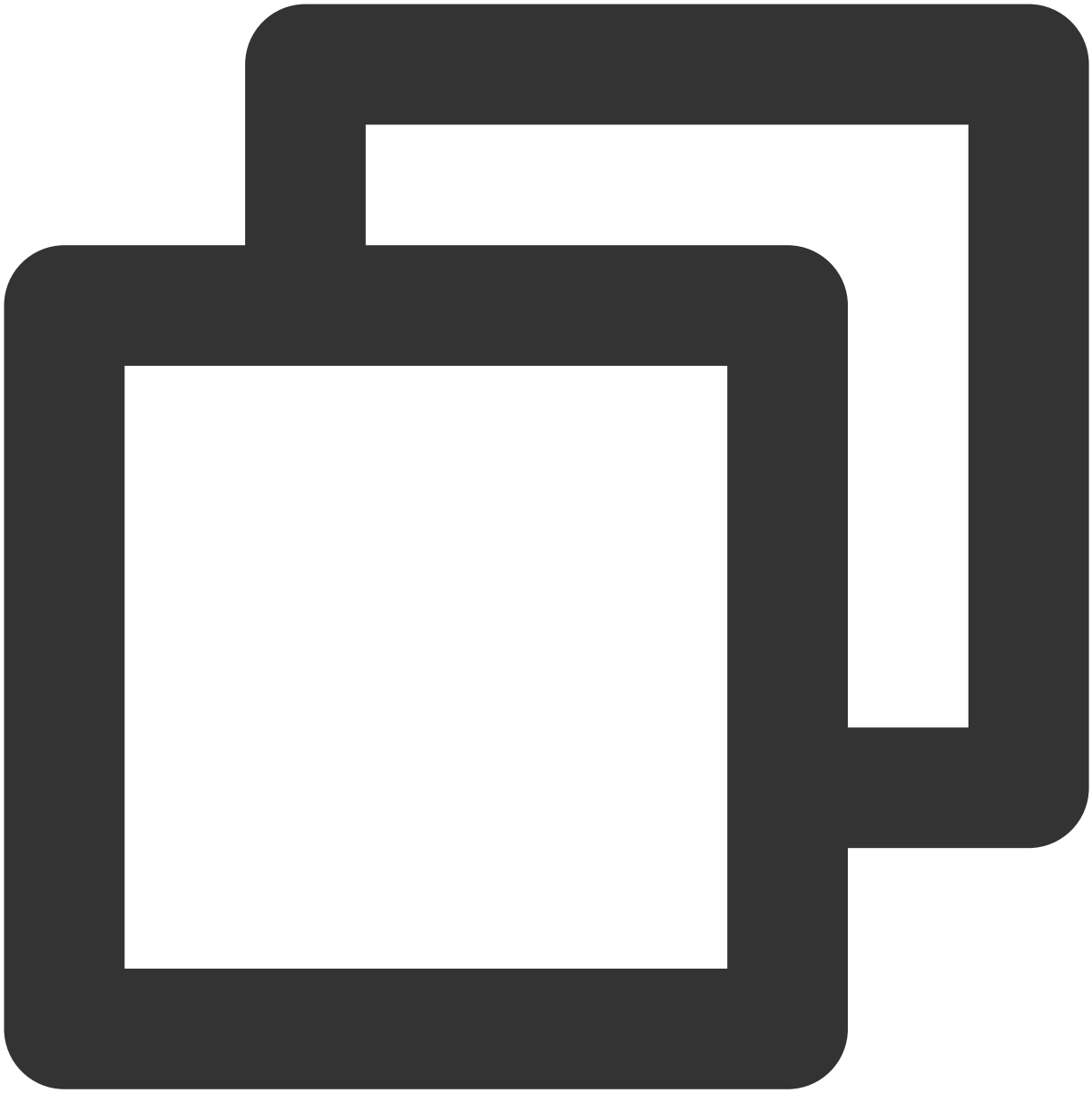
请求



```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>SpeechRecognition</Tag>
    <Name>TemplateName</Name>
    <SpeechRecognition>
        <EngineModelType>16k_zh</EngineModelType>
```

```
        <ResTextFormat>1</ResTextFormat>
        <FilterDirty>0</FilterDirty>
        <FilterModal>1</FilterModal>
        <ConvertNumMode>0</ConvertNumMode>
        <SpeakerDiarization>1</SpeakerDiarization>
        <SpeakerNumber>0</SpeakerNumber>
        <FilterPunc>0</FilterPunc>
        <OutputFileType>txt</OutputFileType>
    </SpeechRecognition>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>SpeechRecognition</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <SpeechRecognition>
            <EngineModelType>16k_zh</EngineModelType>
            <ChannelNum>1</ChannelNum>
            <ResTextFormat>0</ResTextFormat>
            <FilterDirty>1</FilterDirty>
            <FilterModal>0</FilterModal>
            <ConvertNumMode>1</ConvertNumMode>
            <SpeakerDiarization>0</SpeakerDiarization>
            <SpeakerNumber>0</SpeakerNumber>
            <FilterPunc>0</FilterPunc>
        </SpeechRecognition>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

批量任务接口
触发任务（独立节点）

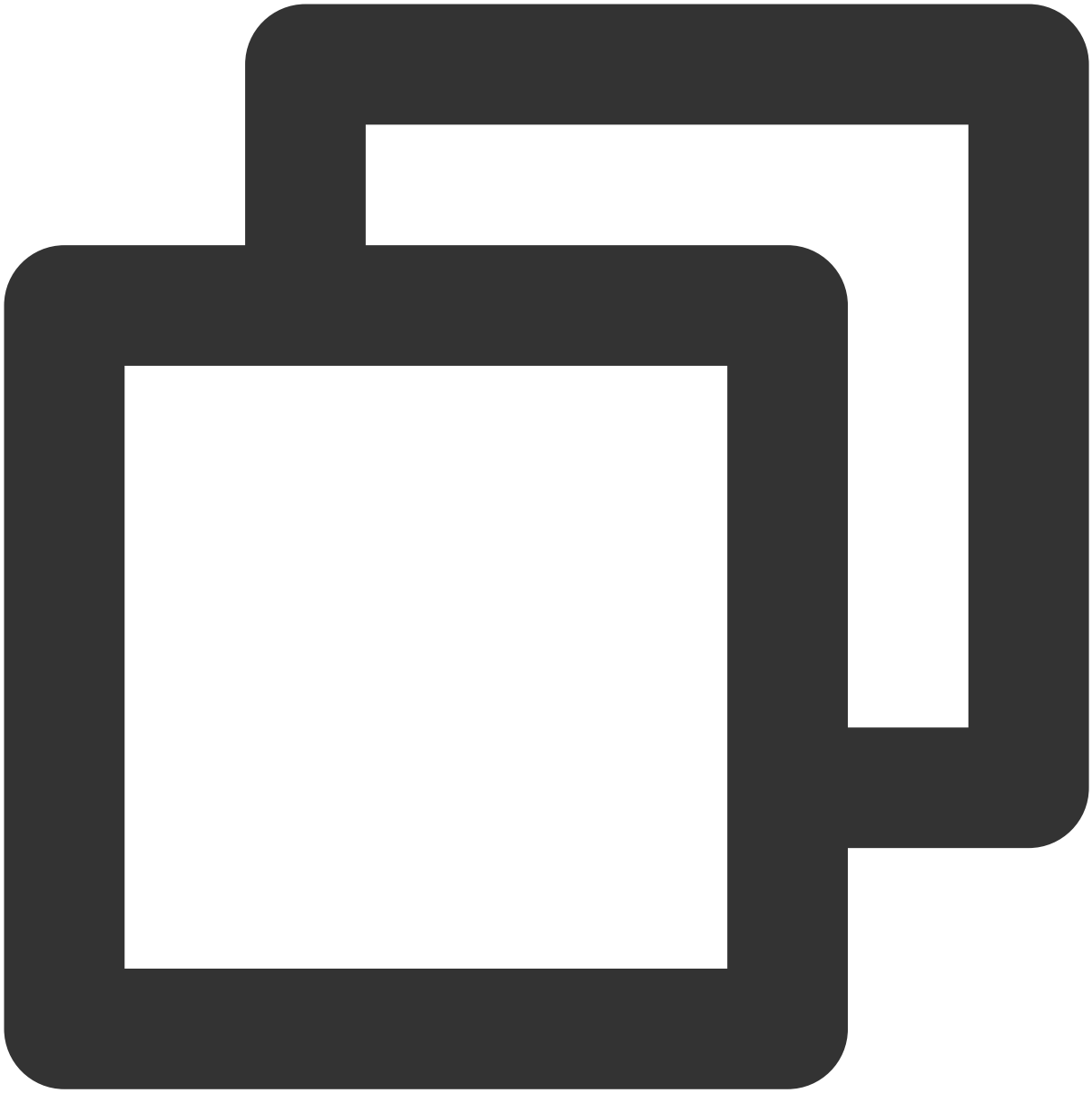
最近更新时间：2023-12-30 16:20:07

功能描述

通过独立任务参数触发批量数据处理任务。

请求

请求示例



```
POST /inventorytriggerjob HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

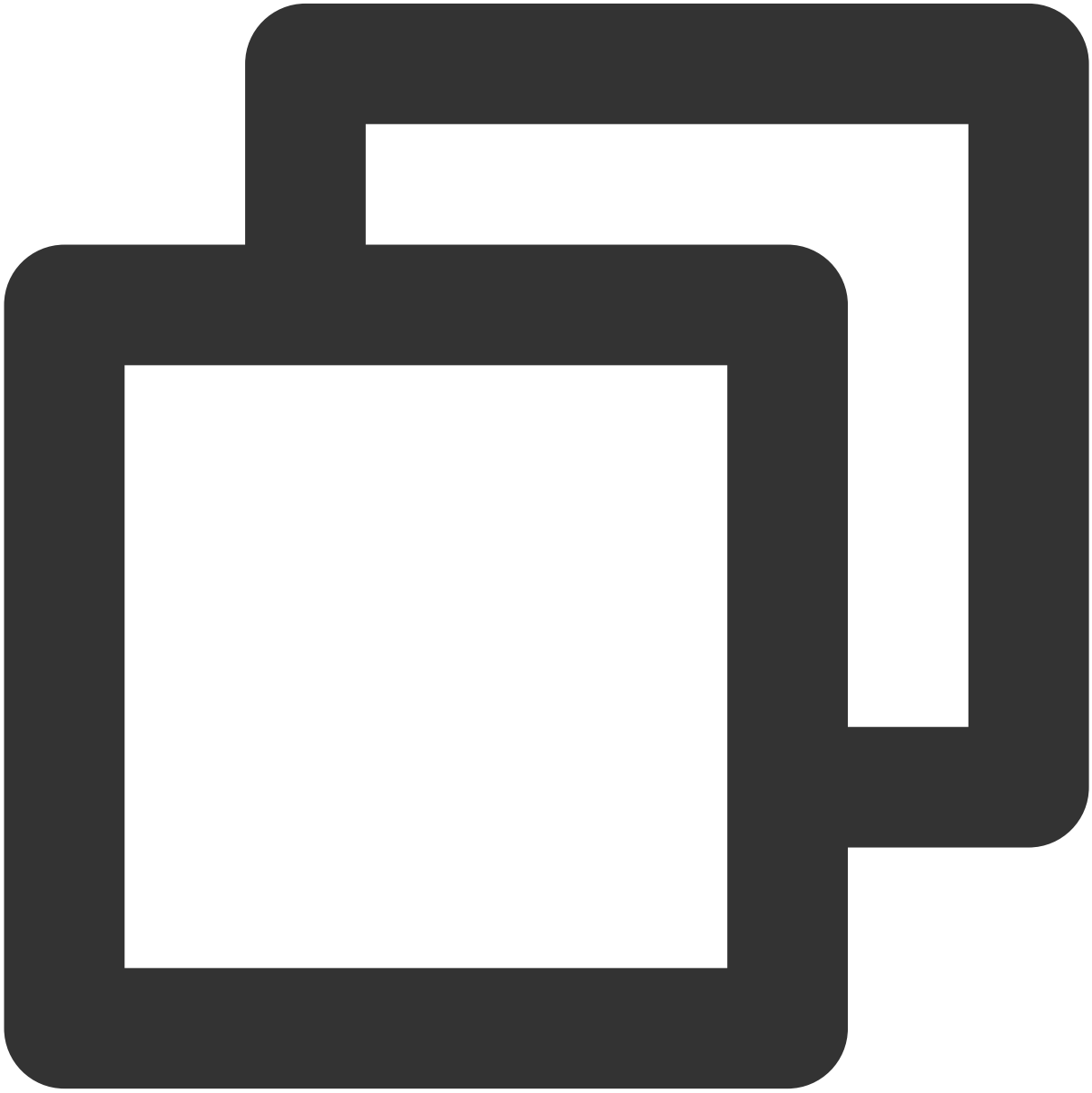
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
    <Name>demo</Name>
```

```
    <Type>Job</Type>
    <Input>
        <Prefix>input</Prefix>
    </Input>
    <Operation>
        <TimeInterval>
            <Start>2022-02-01T12:00:00+0800</Start>
            <End>2022-05-01T12:00:00+0800</End>
        </TimeInterval>
        <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
        <UserData>this is my inventorytriggerjob</UserData>
        <CallBack>https://www.callback.com</CallBack>
        <Tag>Transcode</Tag>
        <JobParam>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        </JobParam>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/${InventoryTriggerJobId}/out.mp4</Object>
        </Output>
        <JobLevel>0</JobLevel>
    </Operation>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Name	Request	存量触发任务名称，支持中文、英文、数字、—和_，长度限制128字符	String	是
Type	Request	批量处理任务类型：Job	String	是
Input	Request	待操作的媒体信息	Container	是
Operation	Request	操作规则	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Manifest	Request.Input	COS 清单列表文件名，COS 清单生成的 manifest.json 文件	String	否
UrlFile	Request.Input	URL 文件名，文件中每行的 URL 为一个 COS 中对象的访问地址	String	否
Prefix	Request.Input	Object 前缀	String	否
Object	Request.Input	媒体文件名	String	否

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
TimeInterval	Request.Operation	按时间过滤的触发范围	Container	否
Tag	Request.Operation	创建的媒体任务类型 Tag	String	是
Output	Request.Operation	输出地址信息 **当媒体任务类型不为 VideoTag 、ExtractDigitalWatermark 、MediaInfo 、 QualityEstimate 时，该参数必选**	Container	否
JobParam	Request.Operation	任务参数	Container	是
QueueId	Request.Operation	队列 ID	String	是
UserData	Request.Operation	用户信息	String	否
JobLevel	Request.Operation	任务优先级，级别限制：0 、1 、2。级别越大任务优先级越高，默认为0	String	否
CallBackFormat	Request.Operation	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request.Operation	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request.Operation	任务回调地址，优先级高于队列的回调地址。设置为 no 时，表示队列的回调地址不产生回调	String	否
CallBackMqConfig	Request.Operation	任务回调 TDMQ 配置，当	Container	否

版权所有：腾讯云计算（北京）有限责任公司

		CallBackType 为 TDMQ 时必填		

Container 类型 CallBackMqConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
MqRegion	Request.Operation.CallBackMqConfig	消息队列所属园区，目前支持园区 sh（上海）、bj（北京）、gz（广州）、cd（成都）、hk（中国香港）	String	是
MqMode	Request.Operation.CallBackMqConfig	消息队列使用模式，默认 Queue ： 主题订阅：Topic 队列服务: Queue	String	是
MqName	Request.Operation.CallBackMqConfig	TDMQ 主题名称	String	是

Operation.Tag 支持的类型如下：

任务类型	Tag
视频转动图	Animatin
智能封面	SmartCover
音视频转码	Transcode
视频截图	Snapshot
语音识别	SpeechRecognition
音视频拼接	Concat
人声分离	VoiceSpearte
精彩集锦	VideoMontage
SDR to HDR	SDRtoHDR
人声分离	VoiceSeparate
视频增强	VideoProcess
超级分辨率	SuperResolution
音视频转封装	Segment

版权所有：腾讯云计算（北京）有限责任公司

视频标签	VideoTag
图片处理	PicProcess
数字水印	DigitalWatermark
提取数字水印	ExtractDigitalWatermark
异步获取媒体信息	MediaInfo
音视频流分离	StreamExtract
语音合成	Tts
音频降噪	NoiseReduction
视频质量分析	QualityEstimate

Container 类型 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.Operation.TimeInterval	开始时间	String	否	0	扫描对象的上传时间 %Y-%m-%dT%H:%m:%S%z
End	Request.Operation.TimeInterval	结束时间	String	否	当前时间	扫描对象的上传时间 %Y-%m-%dT%H:%m:%S%z

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Region	Request.Operation.Output	存储桶的地域	String	是	无
Bucket	Request.Operation.Output	存储桶的名称	String	是	无
Object	Request.Operation.Output	结果文件名称	String	否	**当媒体任务类型不为 Sr 、 StreamExtract 时, 当媒体任务类型为Smar Request.Operation.Jo**

					大于1时，必须包含 ${N 当媒体任务类型为 Snap 大于1时，必须包含 ${N
AuObject	Request.Operation.Output	人声结果文件名	String	否	**当媒体任务类型为 Vide 效，且不能和 Object 同**
SpriteObject	Request.Operation.Output	雪碧图名	String	否	**当媒体任务类型为 Snap 能和 Object 同时为空 当 Snapshot.Count 大 参数** 仅支持 jpg
StreamExtract	Request.Operation.Output	流分离结果文件配置信息，**当媒体任务类型为 StreamExtract 时该参数有效，且必选**	Container	是	无

Container 类型 StreamExtract 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Index	Request.Operation.Output.StreamExtract	流的编号，对应媒体信息中的 Response.MediaInfo.Stream.Video.Index 和 Response.MediaInfo.Stream.Audio.Index，详见 获取媒体信息接口，当输入媒体格式为 mxf 格式时，只支持提取 video 流	String
Object	Request.Operation.Output.StreamExtract	结果输出文件名	String

Container 类型 JobParam 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
TemplateId	Request.Operation.JobParam	模板 ID	String	否

TranscodeTemplateId	Request.Operation.JobParam	转码模板 ID，当媒体任务类型为 VideoProcess 、SuperResolution 、SDR2HDR 时有效	Container	否
Animation	Request.Operation.JobParam	动图任务参数，当媒体任务类型为 Animation 时有效	Container	否
Transcode	Request.Operation.JobParam	转码任务参数，当媒体任务类型为 Transcode 、VideoProcess 、SuperResolution 、SDR2HDR 时该参数有效	String	否
SmartCover	Request.Operation.JobParam	智能封面任务参数，当媒体任务类型为 SmartCover 时有效	Container	否
DigitalWatermark	Request.Operation.JobParam	数字水印参数，当媒体任务类型为 Transcode 、VideoProcess 、SuperResolution 、DigitalWatermark 时该参数有效	Container	否
Watermark	Request.Operation.JobParam	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口的 Request.Watermark，当媒体任务类型为 Transcode 、 SDR2HDR 、VideoProcess 、SuperResolution 时该参数有效	Container	否
RemoveWatermark	Request.Operation.JobParam	指定去除水印参数，当媒体任务类型为 Transcode 时有效	Container	否
WatermarkTemplateId	Request.Operation.JobParam	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口的 Request.Watermark ，当媒体任务类型为 Transcode 、 SDR2HDR 、 VideoProcess 、	Container	否

版权所有：腾讯云计算（北京）有限责任公司

		SuperResolution 时该参数有效		
RemoveWatermark	Request.Operation.JobParam	指定去除水印参数，当媒体任务类型为 Transcode 时有效	Container	否
Snapshot	Request.Operation.JobParam	截图任务参数，当媒体任务类型为 Snapshot 时有效	Container	否
SpeechRecognition	Request.Operation.JobParam	语音识别任务参数，当媒体任务类型为 SpeechRecognition 时有效	Container	否
ConcatTemplate	Request.Operation.JobParam	拼接任务参数，当媒体任务类型为 Concat 时有效	Container	否
VoiceSeparate	Request.Operation.JobParam	人声分离任务参数，当媒体任务类型为 VoiceSeparate 时有效	Container	否
VideoMontage	Request.Operation.JobParam	精彩集锦任务参数，当媒体任务类型为 VideoMontage 时有效	Container	否
SDRtoHDR	Request.Operation.JobParam	SDR2HDR 任务参数，当媒体任务类型为 SDRtoHDR 时有效	Container	否
VideoProcess	Request.Operation.JobParam	视频增强任务参数，当媒体任务类型为 VideoProcess 时有效	Container	否
SuperResolution	Request.Operation.JobParam	超分辨率任务参数，当媒体任务类型为 SuperResolution 时有效	Container	否
Segment	Request.Operation.JobParam	转封装任务参数，当媒体任务类型为 Segment 时有效	Container	否
ExtractDigitalWatermark	Request.Operation.JobParam	提取数字水印任务参数，当媒体任务类型为 ExtractDigitalWatermark 时有效	Container	否
VideoTag	Request.Operation.JobParam	视频标签任务参数，当媒体任务类型为 VideoTag 时有效	Container	否

版权所有：腾讯云计算（北京）有限责任公司

TtsTpl	Request.Operation.JobParam	语音合成模板参数，当媒体任务类型为 Tts 时有效	Container	否

注意

优先使用 TemplateId，无 TemplateId 时使用对应任务类型的参数。

Container 类型 DigitalWatermark 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	必选
Message	Request.Operation.JobParam.DigitalWatermark	数字水印嵌入的字符串信息，长度不超过64个字符，仅支持中文、英文、数字、_、-和*	String	是
Type	Request.Operation.JobParam.DigitalWatermark	水印类型，当前仅可设置为 Text	String	是
Version	Request.Operation.JobParam.DigitalWatermark	水印版本，当前仅可设置为 V1	String	是
IgnoreError	Request.Operation.JobParam.DigitalWatermark	当添加水印失败是否忽略错误继续执行任务，限制为 true/false，默认为false 当 Request.Operation.Tag 为 DigitalWatermark 时，该参数无效，添加水印失败任务失败	String	否

Container 类型 Animation 的内容：

同 提交动图任务 的 Request.Operation.Animation

Container 类型 Transcode 的内容：

同 提交音视频转码任务 的 Request.Operation.Transcode

Container 类型 Watermark 的内容：

同 提交音视频转码任务 的 Request.Operation.Watermark

Container 类型 RemoveWatermark 的内容：

同 提交音视频转码任务 的 Request.Operation.RemoveWatermark

Container 类型 SmartCover 的内容：

同 提交智能封面任务 的 Request.Operation.SmartCover

Container 类型 Snapshot 的内容：

同 提交截图任务 的 Request.Operation.Snapshot

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 SpeechRecognition 的内容：
同 提交语音识别任务 的 Request.Operation.SpeechRecognition
Container 类型 ConcatTemplate 的内容：
同 提交拼接任务 的 Request.Operation.ConcatTemplate
Container 类型 VoiceSeparate 的内容：
同 提交人声分离任务 的 Request.Operation.VoiceSeparate
Container 类型 VideoMontage 的内容：
同 提交精彩锦集任务 的 Request.Operation.VideoMontage
Container 类型 SDRtoHDR 的内容：
同 提交SDRtoHDR任务 的 Request.Operation.SDRtoHDR
Container 类型 VideoProcess 的内容：
同 提交视频增强任务 的 Request.Operation.VideoProcess
Container 类型 SuperResolution 的内容：
同 提交超分辨率任务 的 Request.Operation.SuperResolution
Container 类型 Segment 的内容：
同 提交流分离任务 的 Request.Operation.Segment
Container 类型 ExtractDigitalWatermark 的内容：
同 提交提取数字水印任务 的 Request.Operation.ExtractDigitalWatermark
Container 类型 VideoTag 的内容：
同 提交智能视频标签任务 的 Request.Operation.VideoTag
Container 类型 TtsTpl 的内容：
同 提交语音合成任务 的 Request.Operation.TtsTpl

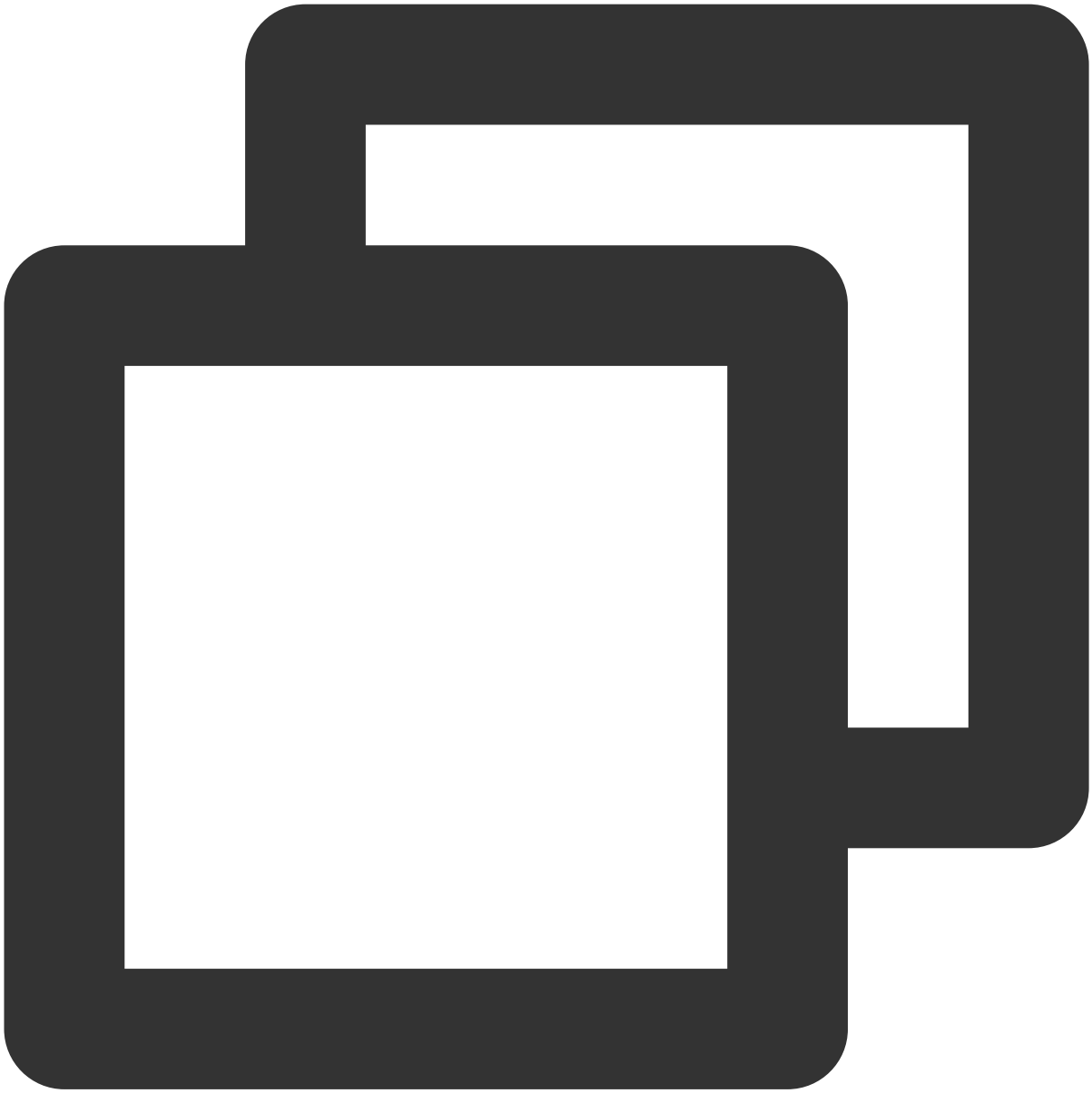
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId>NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18xYQ==<RequestId>
    <JobsDetail>
        <Code>Success</Code>
        <Type>Job</Type>
        <Message/>
        <Name>demo</Name>
        <JobId>be8f65004eb8511eaaed4f377124a303c</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>2022-06-27T15:23:11+0800</StartTime>
```

```
        <EndTime>2022-06-27T15:25:10+0800</EndTime>
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
            <UserData>this is my inventorytriggerjob</UserData>
            <CallBack>https://www.callback.com</CallBack>
            <Tag>Transcode</Tag>
            <JobParam>
                <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            </JobParam>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
                <Object>output/${InventoryTriggerJobId}/out.mp4</Object>
            </Output>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State为 Failed 时有意义	String

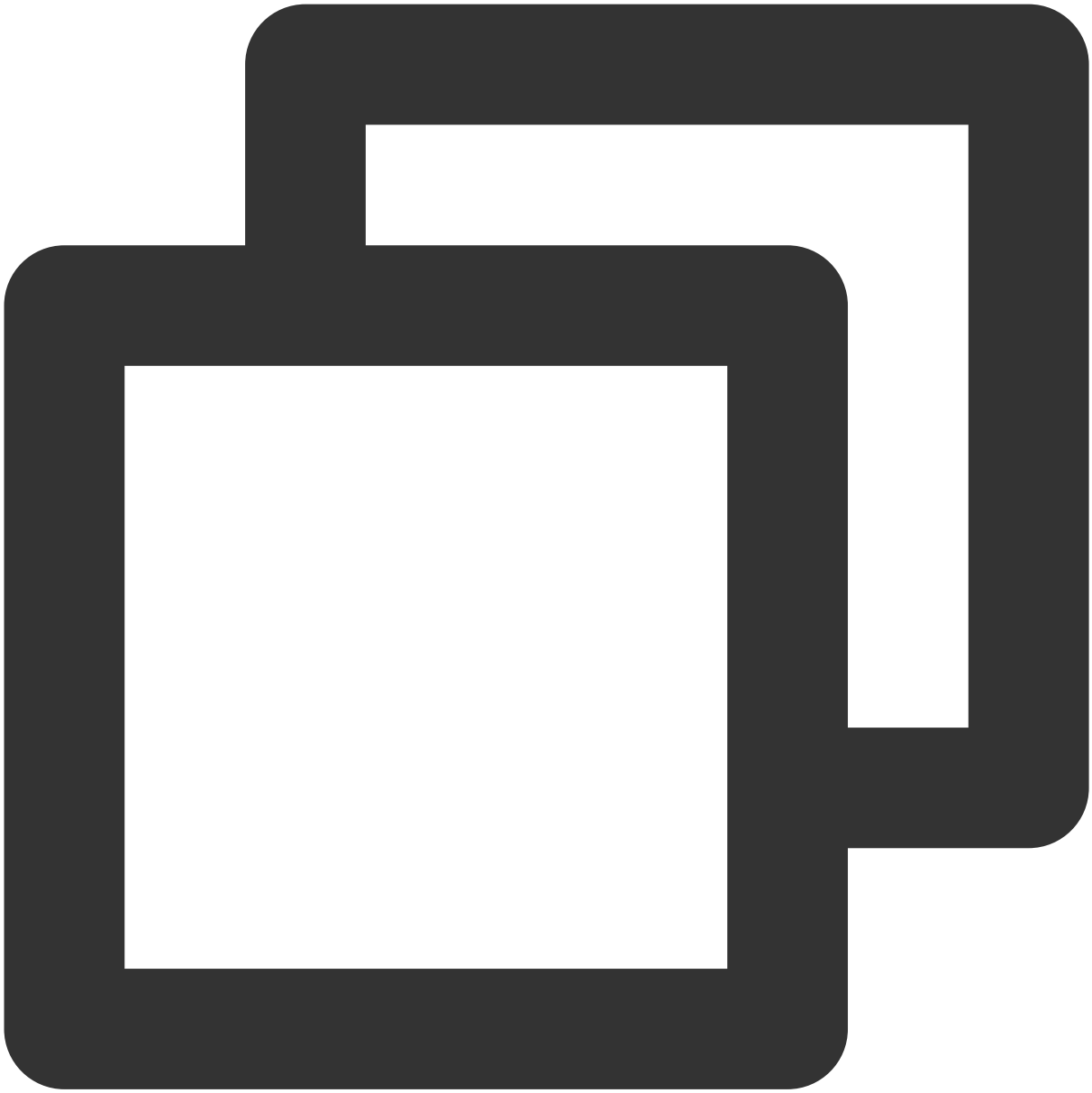
Message	Response.JobsDetail	错误描述，只有 State为 Failed 时有意义	String
Type	Response.JobsDetail	批量处理任务类型，Job	String
Name	Response.JobsDetail	新创建任务的名称	String
JobId	Response.JobsDetail	新创建任务的 ID	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.JobsDetail	任务的开始时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
Input	Response.JobsDetail	该任务的输入资源地址，同请求中的 Request.Input 节点	Container
Operation	Response.JobsDetail	该任务的操作规则，同请求中的 Request.Input Operation	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

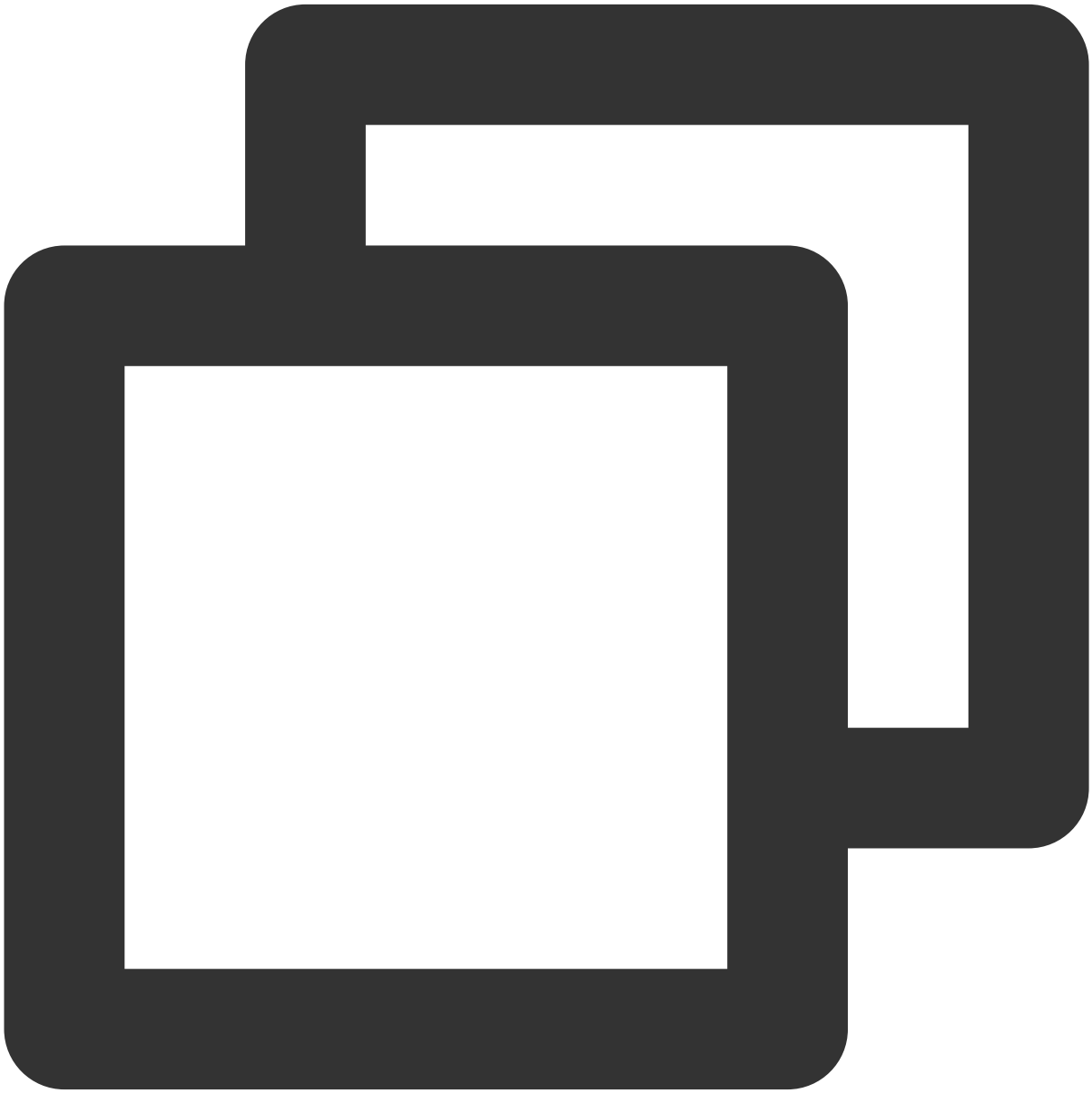
请求



```
POST /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Name>demo</Name>
    <Type>Job</Type>
    <Input>
        <Prefix>input</Prefix>
```

```
    </Input>
    <Operation>
        <TimeInterval>
            <Start>2022-02-01T12:00:00+0800</Start>
            <End>2022-05-01T12:00:00+0800</End>
        </TimeInterval>
        <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
        <UserData>this is my inventorytriggerjob</UserData>
        <CallBack>https://www.callback.com</CallBack>
        <Tag>Transcode</Tag>
        <JobParam>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        </JobParam>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/${InventoryTriggerJobId}/out.mp4</Object>
        </Output>
        <JobLevel>0</JobLevel>
    </Operation>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18xYQ==

<Response>
    <RequestId>NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18xYQ==<RequestId>
    <JobsDetail>
```

```
        <Code>Success</Code>
        <Type>Job</Type>
        <Message/>
        <Name>demo</Name>
        <JobId>be8f65004eb8511eaaed4f377124a303c</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>2022-06-27T15:23:11+0800</StartTime>
        <EndTime>2022-06-27T15:25:10+0800</EndTime>
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
            <UserData>this is my inventorytriggerjob</UserData>
            <CallBack>https://www.callback.com</CallBack>
            <Tag>Transcode</Tag>
            <JobParam>
                <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            </JobParam>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
                <Object>output/${InventoryTriggerJobId}/out.mp4</Object>
            </Output>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

触发任务（工作流）

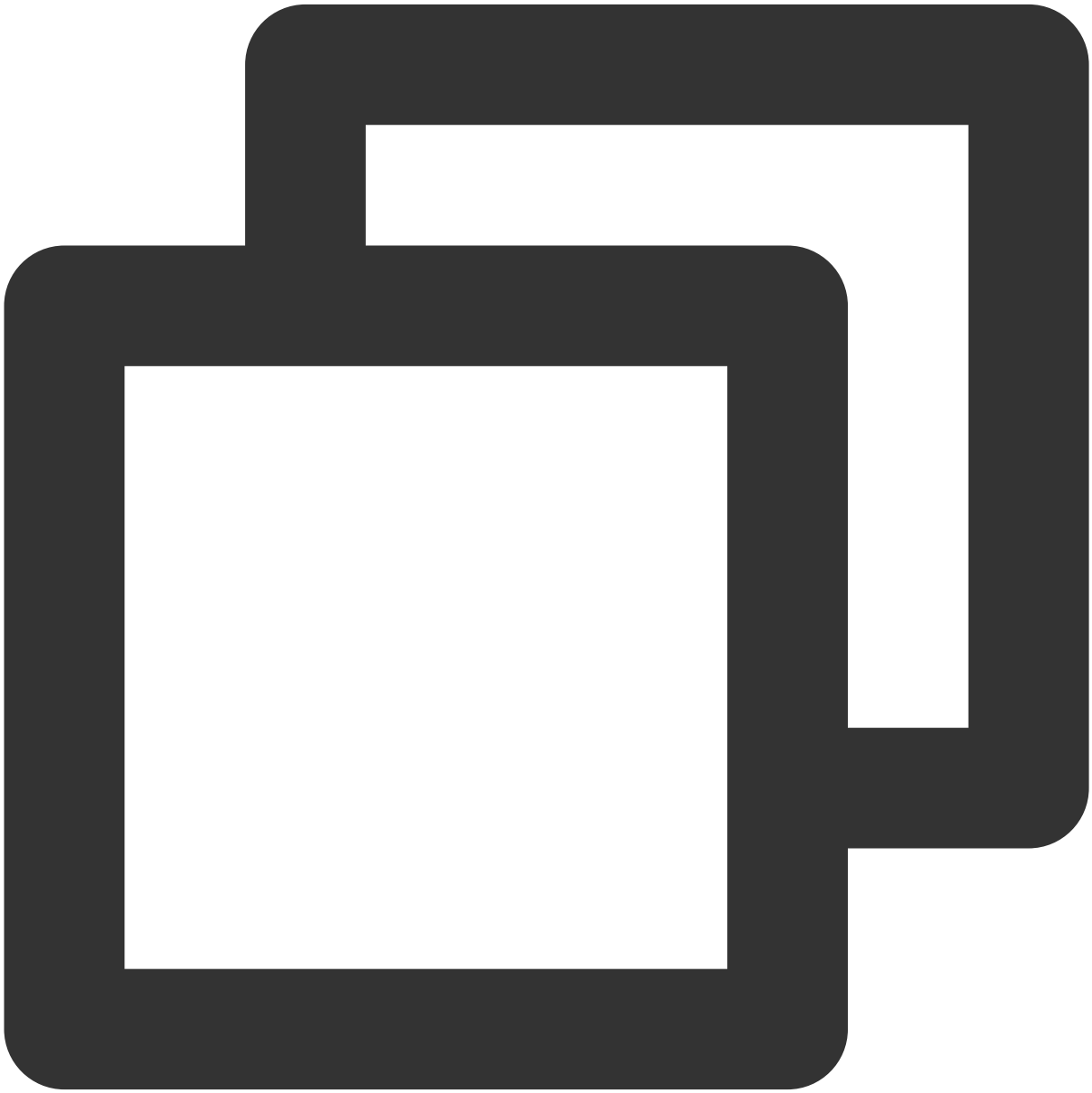
最近更新时间：2023-12-30 16:20:14

功能描述

通过工作流触发批量数据处理任务。

请求

请求示例



```
POST /inventorytriggerjob HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

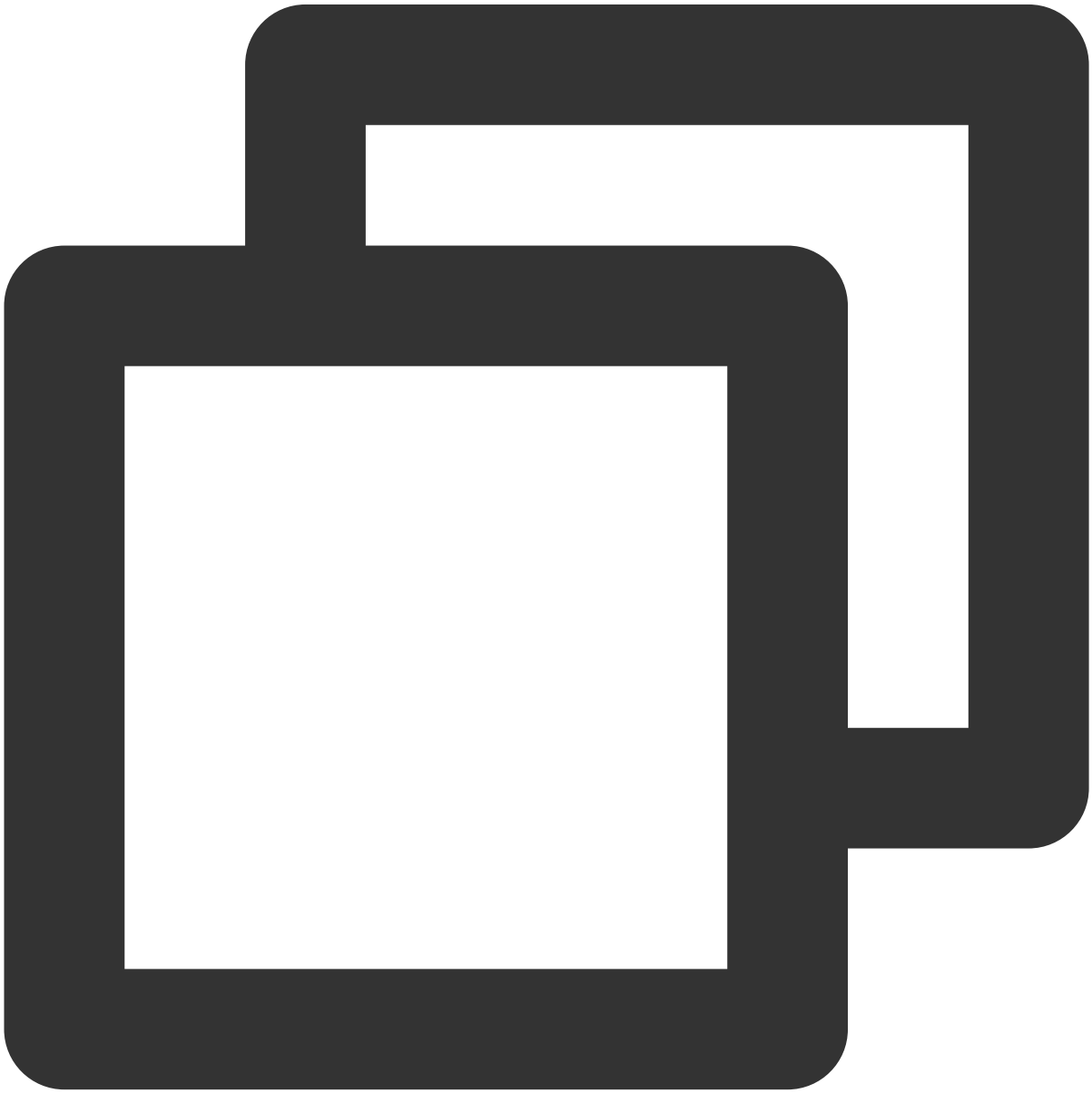
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Name>demo</Name>
    <Type>Workflow</Type>
    <Input>
        <Prefix>input</Prefix>
    </Input>
    <Operation>
        <TimeInterval>
            <Start>2022-02-01T12:00:00+0800</Start>
            <End>2022-05-01T12:00:00+0800</End>
        </TimeInterval>
        <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
    </Operation>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Name	Request	批量触发任务名称，支持中文、英文、数字、—和_，长度限制128字符	String	是
Type	Request	批量处理任务类型: Workflow	String	否
Input	Request	待操作的媒体信息	Container	是
Operation	Request	操作规则	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Manifest	Request.Input	COS 清单列表文件名，COS 清单生成的 manifest.json 文件	String	否
UrlFile	Request.Input	URL 文件名，文件中每行的 URL 为一个 COS 中对象的访问地址	String	否
Prefix	Request.Input	Object 前缀	String	否
Object	Request.Input	媒体文件名	String	否

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
WorkflowIds	Request.Operation	触发的工作流 ID	String	是
TimeInterval	Request.Operation	按时间过滤的触发范围	Container	否

Container 类型 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.TimeInterval	开始时间	String	否	0	扫描对象的上传时间 %Y-%m-%dT%H:%m:%S%z
End	Request.TimeInterval	结束时间	String	否	当前时间	扫描对象的上传时间 %Y-%m-%dT%H:%m:%S%z

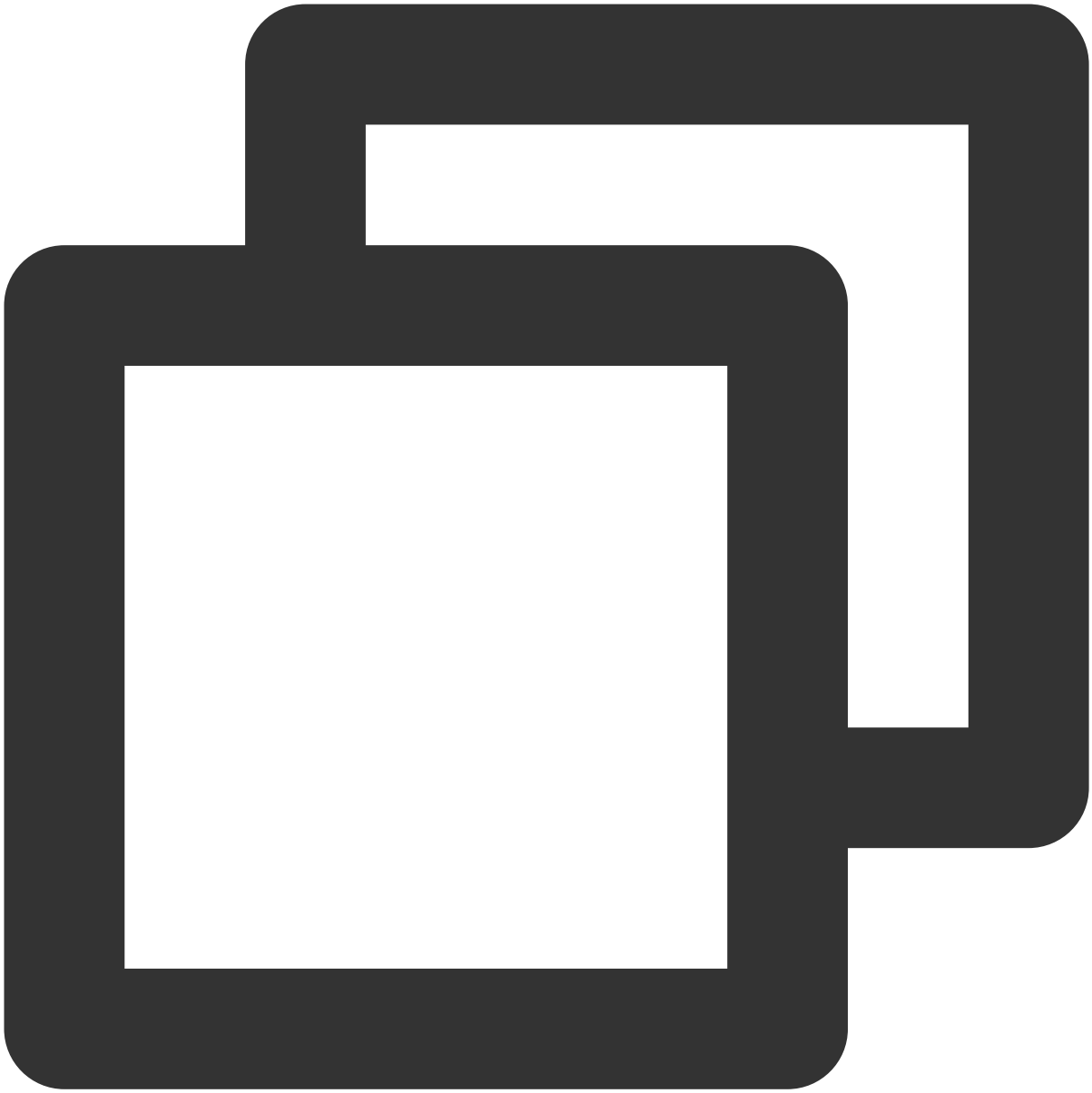
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <Type>Workflow</Type>
        <Name>demo</Name>
        <JobId>b3deffea2f84911ec9cb15254008618d9</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
        </Operation>
    </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
Type	Response.JobsDetail	批量处理任务类型	String
Name	Response.JobsDetail	新创建任务的名称	String
JobId	Response.JobsDetail	新创建任务的 ID	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.JobsDetail	任务的开始时间	String

版权所有：腾讯云计算（北京）有限责任公司

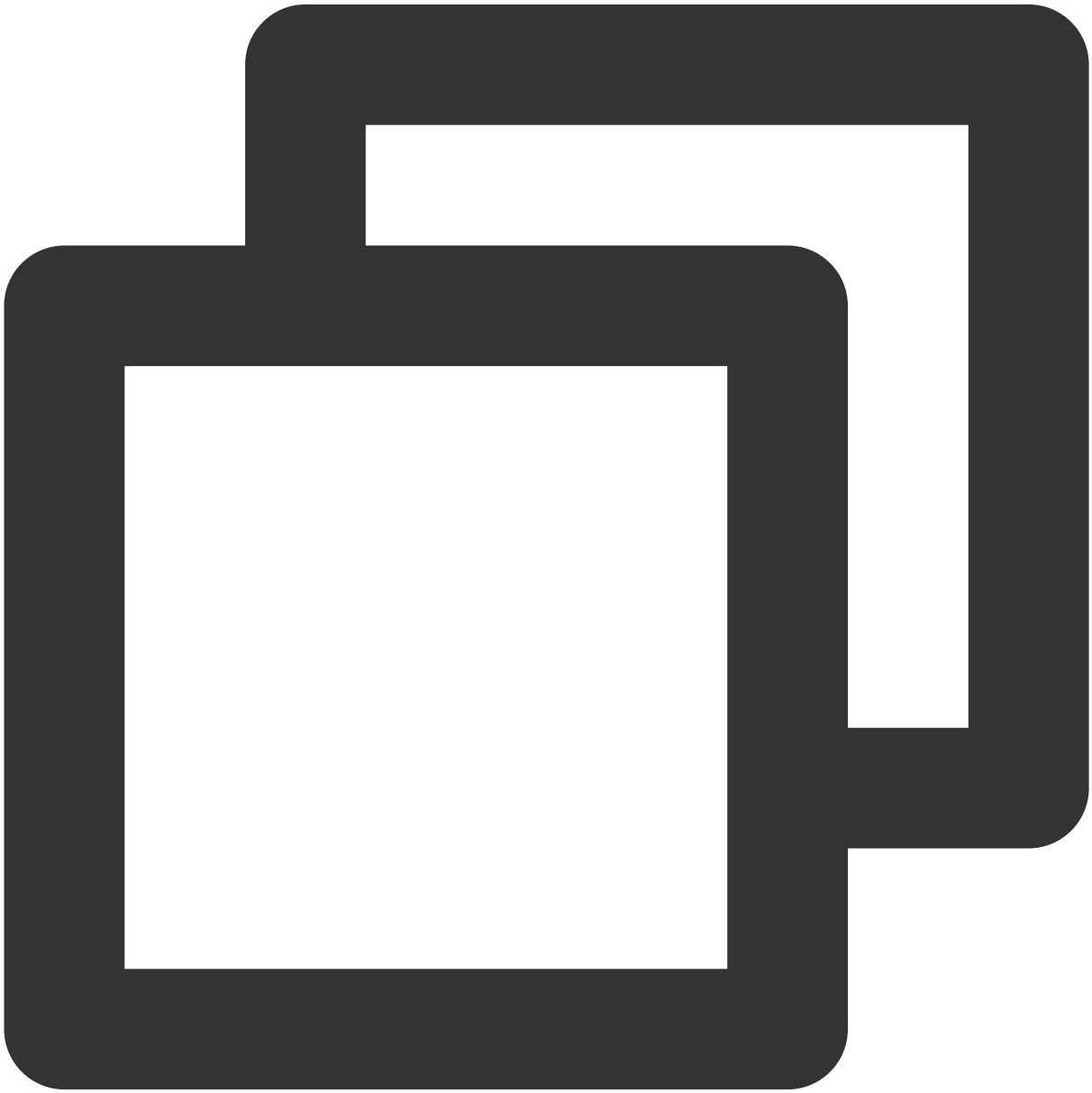
EndTime	Response.JobsDetail	任务的结束时间	String
Input	Response.JobsDetail	同请求中的 Request.Input	Container
Operation	Response.JobsDetail	同请求中的 Request.Operation	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

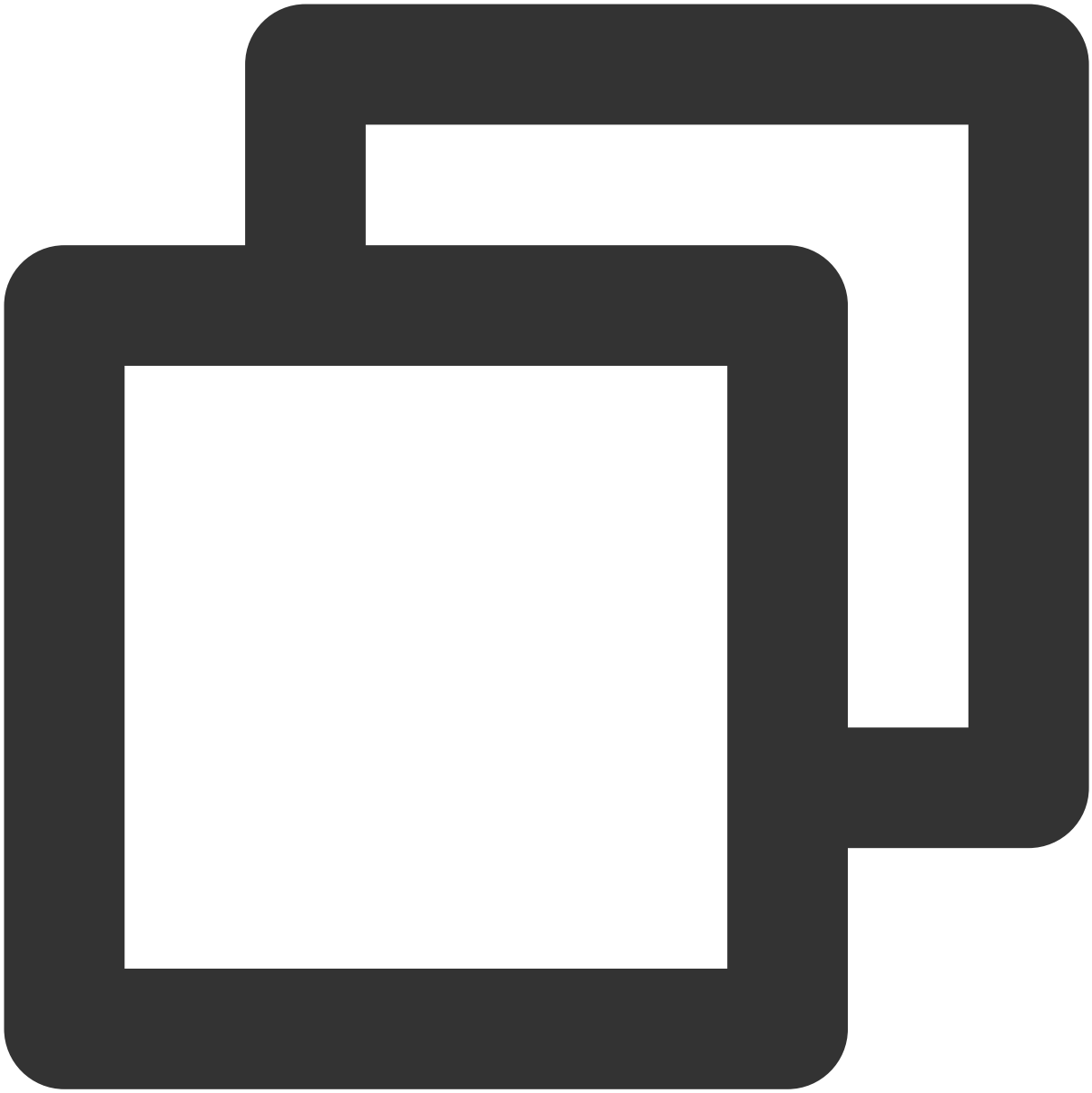
请求



```
POST /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Name>demo</Name>
    <Type>Workflow</Name>
    <Input>
        <Prefix>input</Prefix>
```

```
    </Input>
    <Operation>
        <TimeInterval>
            <Start>2022-02-01T12:00:00+0800</Start>
            <End>2022-05-01T12:00:00+0800</End>
        </TimeInterval>
        <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
    </Operation>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Type>Workflow</Type>
        <Message/>
        <Name>demo</Name>
        <JobId>b3deffea2f84911ec9cb15254008618d9</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
        </Operation>
    </JobsDetail>
</Response>
```

取消任务

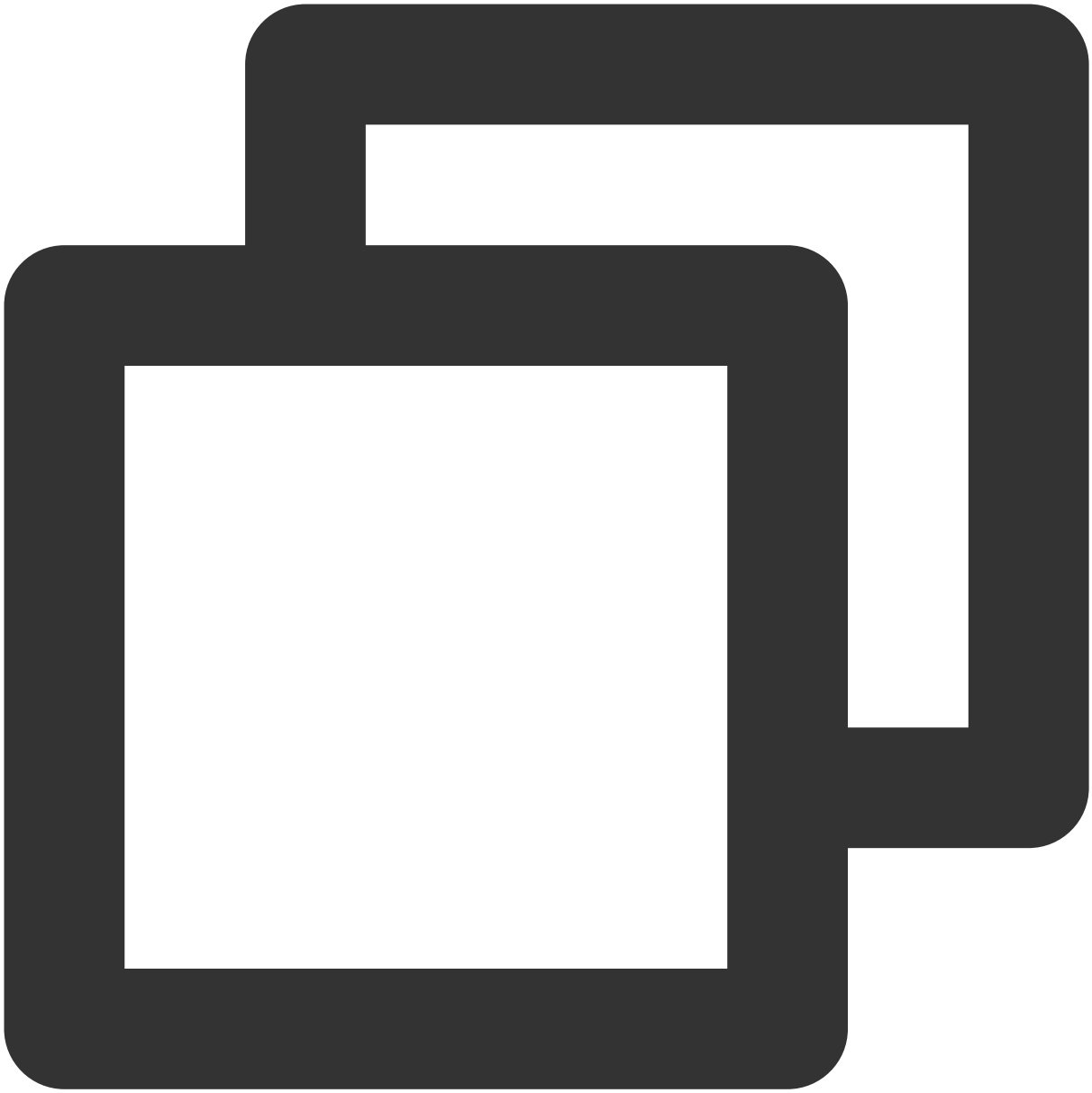
最近更新时间：2023-12-30 16:20:35

功能描述

取消批量数据处理任务。

请求

请求示例



```
PUT /inventorytriggerjob/<jobId>?cancel HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

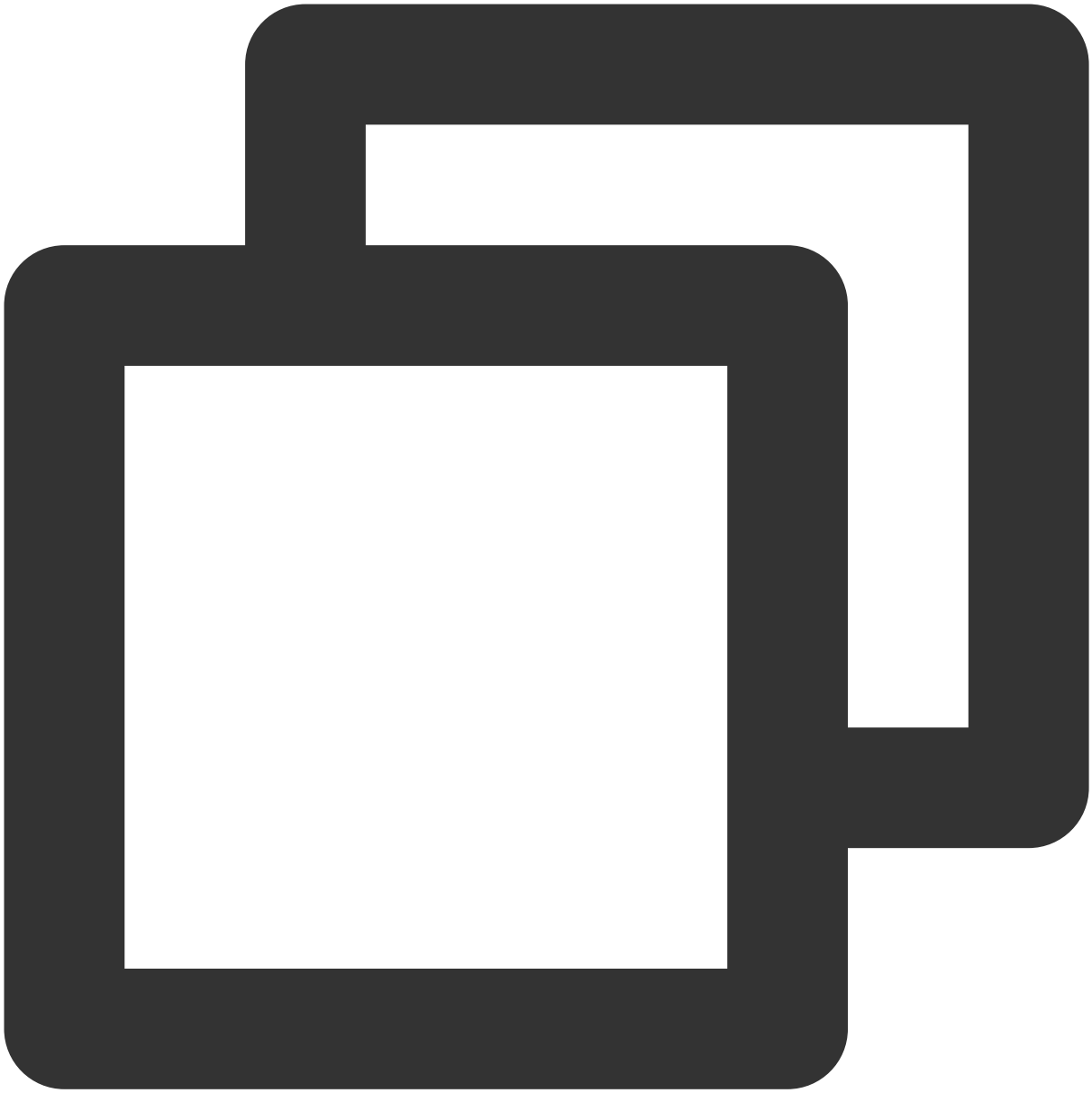
该请求无响应体。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

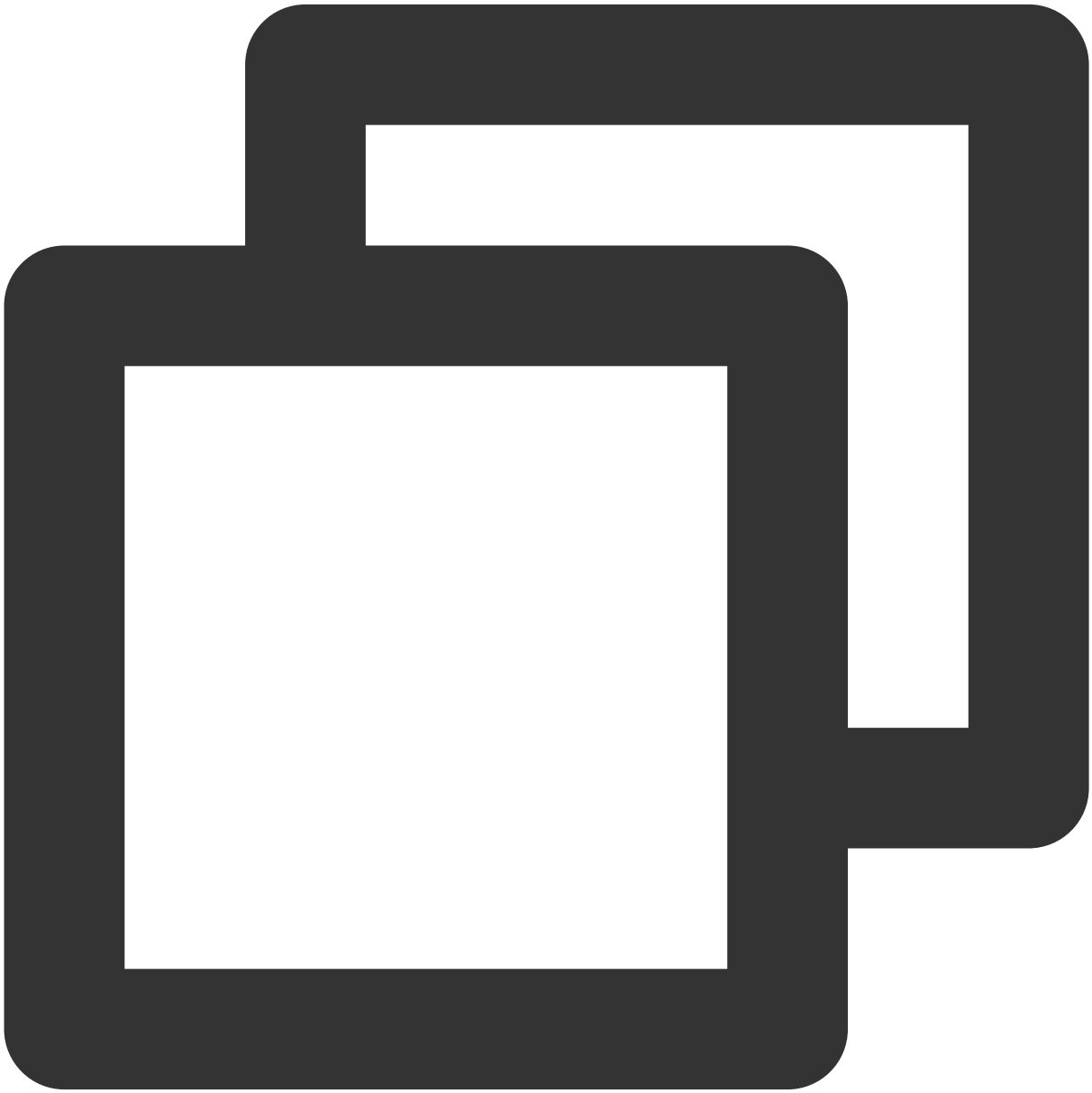
实际案例

请求



```
PUT /inventorytriggerjob/b-xxx-xxx-xxx-xxxx?cancel HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=
```

查询任务

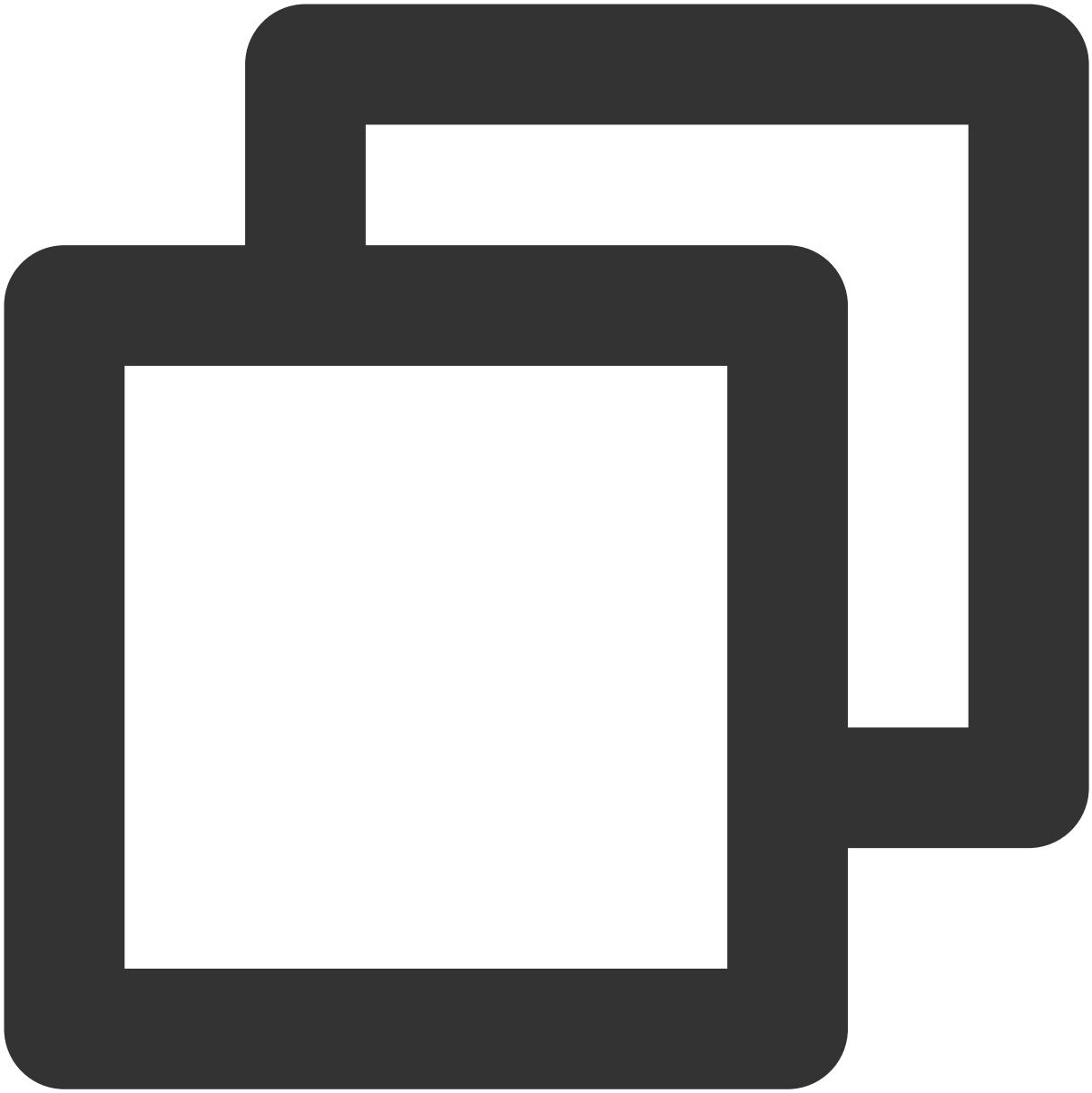
最近更新时间：2023-12-30 16:21:51

功能描述

查询批量数据处理任务。

请求

请求示例



```
GET /inventorytriggerjob/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

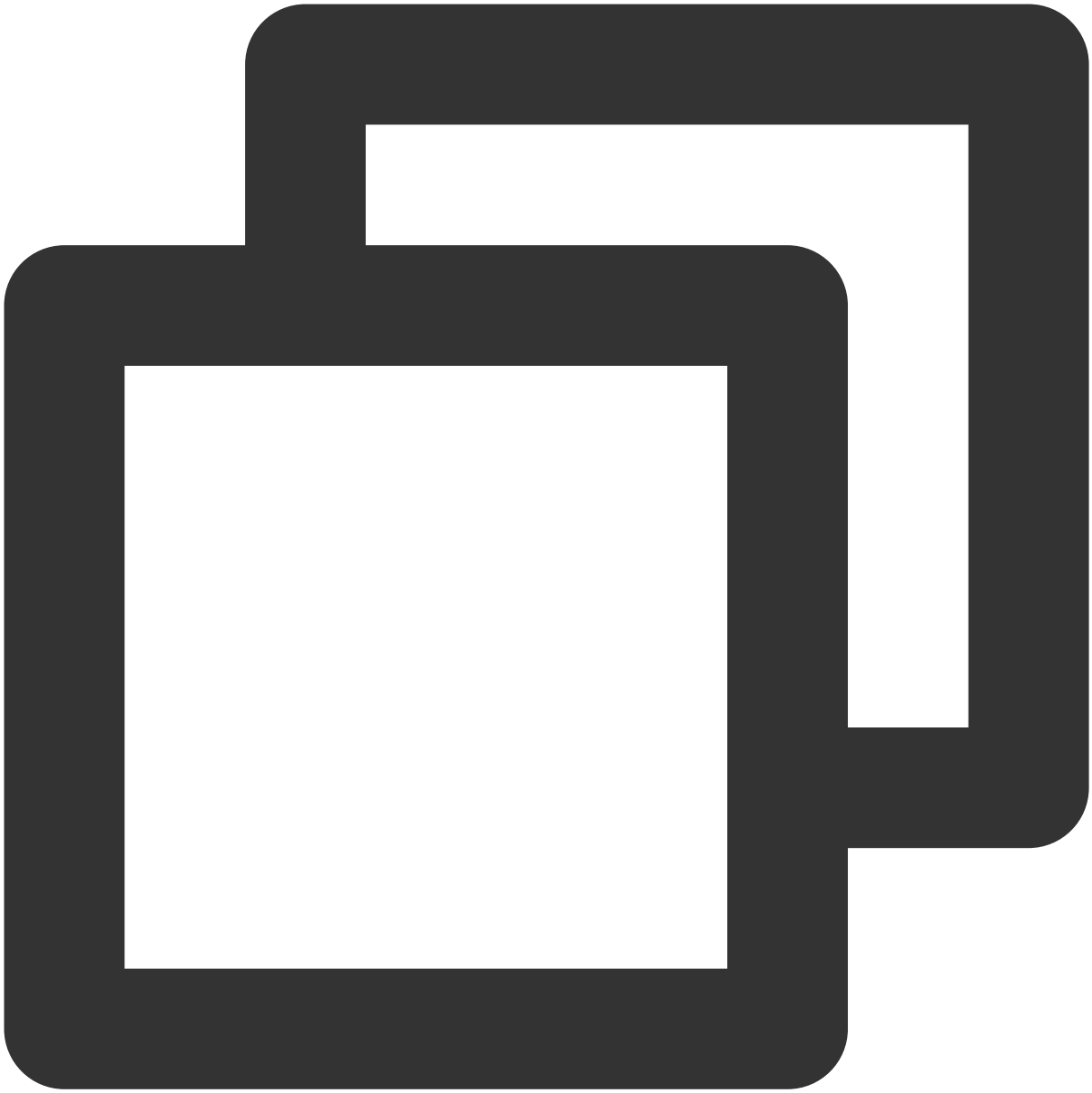
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId><RequestId>
    <JobsDetail>
    ...
    </JobsDetail>
    <NonExistJobIds></NonExistJobIds>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
JobsDetail	Response	任务的详细信息	Container
NonExistJobIds	Response	查询的 ID 中不存在的任务，所有任务都存在时不返回	String

对于不同的触发方式，JobsDetail 的内容不同，请参照以下链接：

工作流

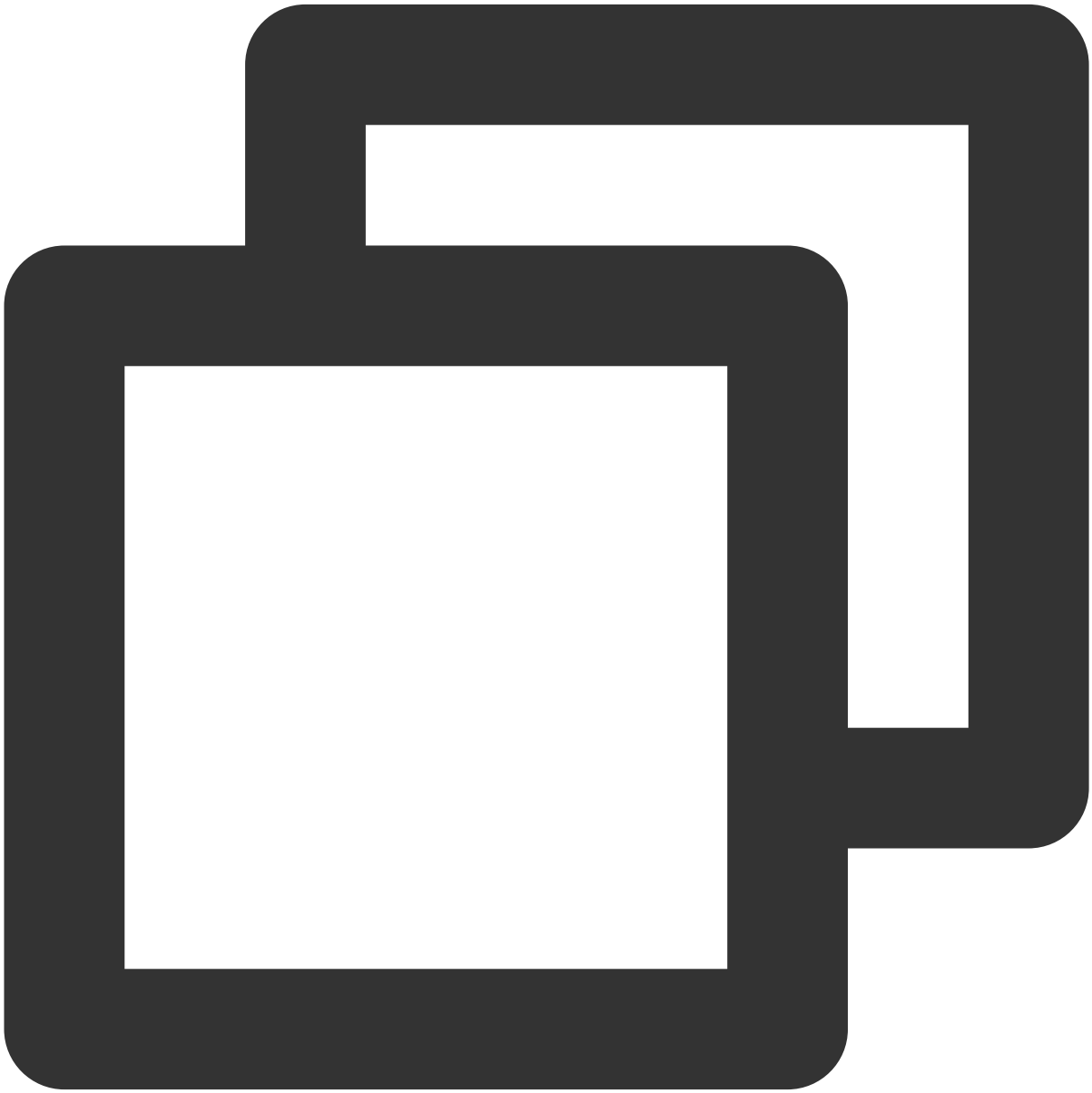
独立任务

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

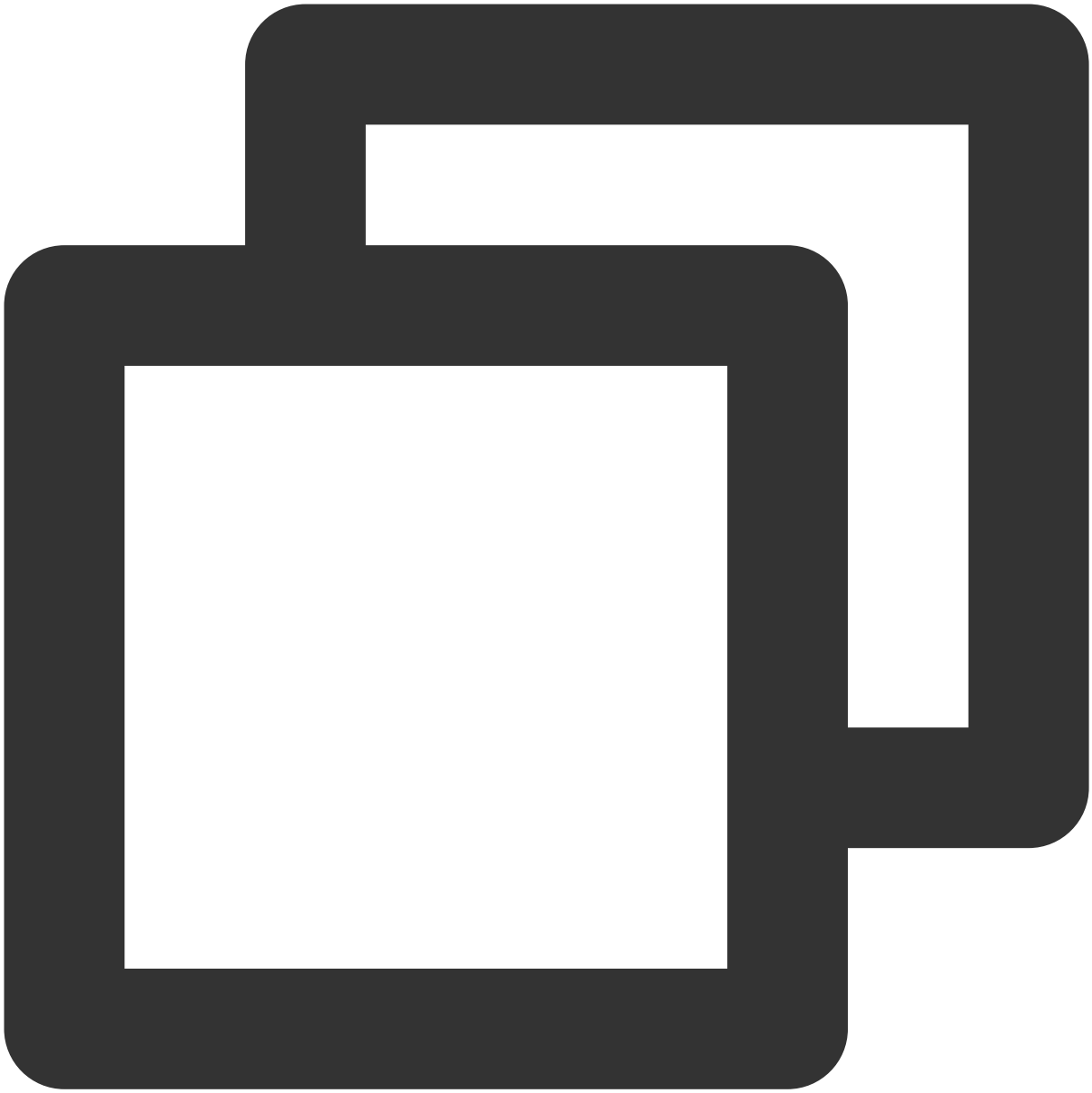
实际案例

请求



```
GET /inventorytriggerjob/b3deffea2f84911ec9cb15254008618d9 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18xYQ==

<Response>
    <RequestId>NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18xYQ==</RequestId>
    <JobsDetail>
```

```
        <Code>Success</Code>
        <Type>Workflow</Type>
        <Message/>
        <Name>demo</Name>
        <JobId>b3deffea2f84911ec9cb15254008618d9</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
        </Operation>
    </JobsDetail>
</Response>
```

批量拉取任务

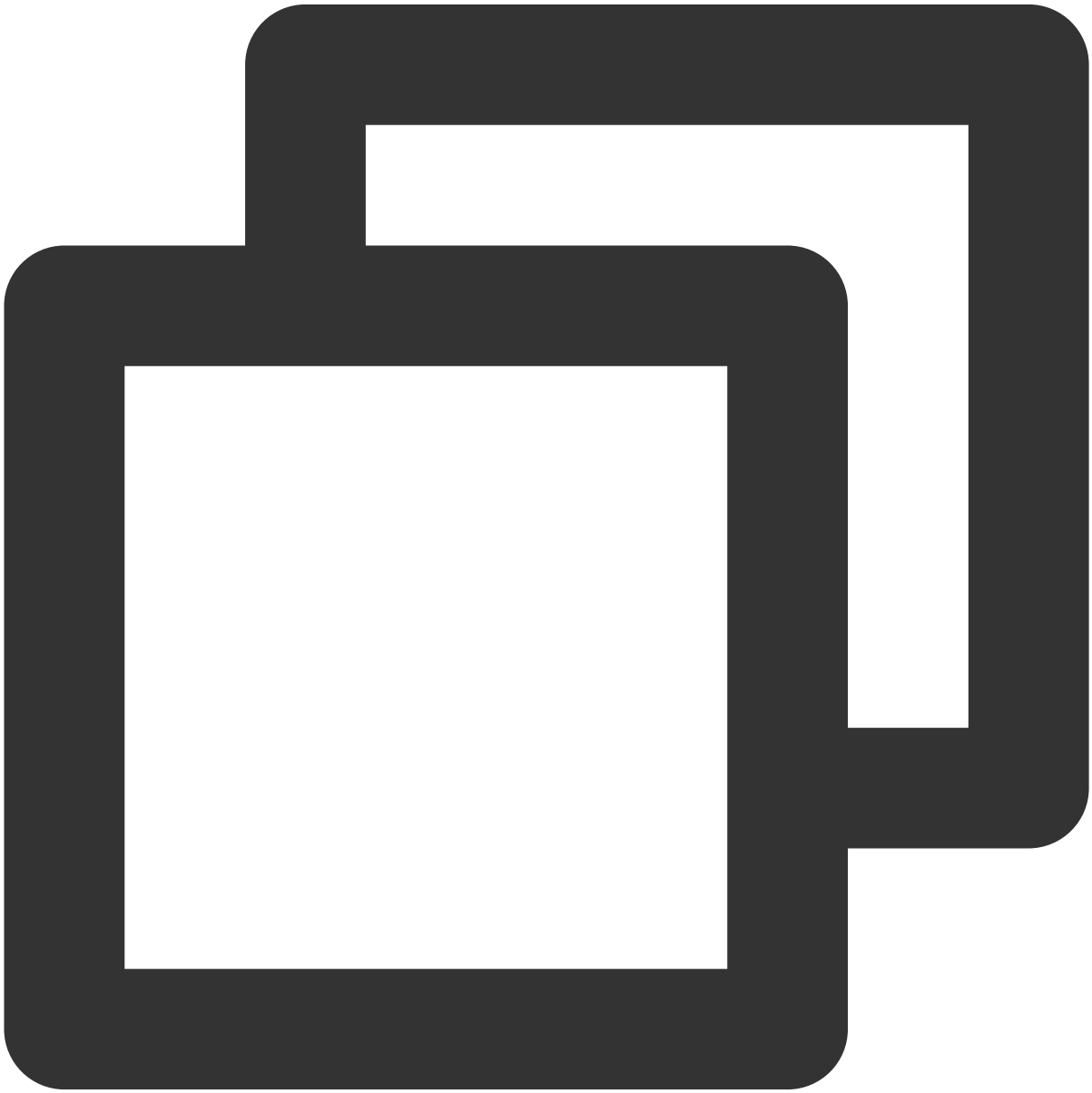
最近更新时间：2023-12-30 16:20:58

功能描述

拉取批量数据处理任务列表。

请求

请求示例



```
GET /inventorytriggerjob?size=&states=&startCreationTime=&endCreationTime= HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明：

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	必选
nextToken	无	请求的上下文，用于翻页。上次返回的值	String	否
size	无	拉取的最大任务数。默认为10。最大为100	Integer	否
type	无	拉取批量任务类型，工作流类型 Workflow 、任务类型 Job	String	否
orderByTime	无	Desc 或者 Asc。默认为 Desc	String	否
states	无	拉取该状态的任务，以,分割支持多状态 All，Submitted，Running，Success，Failed，Pause，Cancel。默认为 All	String	否
startCreationTime	无	拉取创建时间大于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否
endCreationTime	无	拉取创建时间小于等于该时间的任务。格式为：`%Y-%m-%dT%H:%m:%S%z`	String	否
workflowId	无	工作流 ID	String	否
jobId	无	批量触发任务 ID	String	否
name	无	批量触发任务名称	String	否

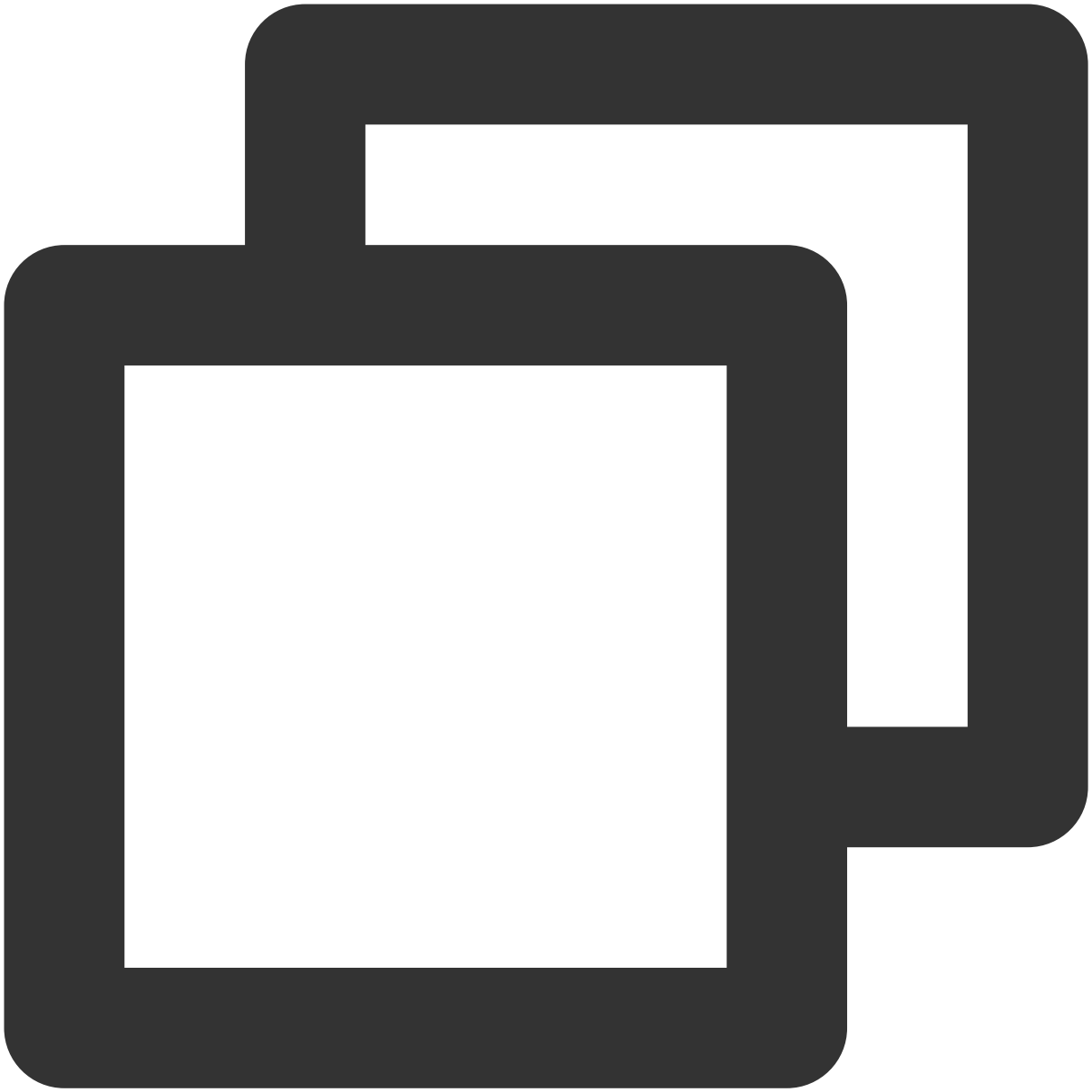
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId></RequestId>
    <JobsDetail>
    ...
    </JobsDetail>
    <NextToken></NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
JobsDetail	Response	任务的详细信息	Container 数组
NextToken	Response	翻页的上下文 Token	String

对于不同的触发方式，JobsDetail 的内容不同，请参照以下链接：

工作流

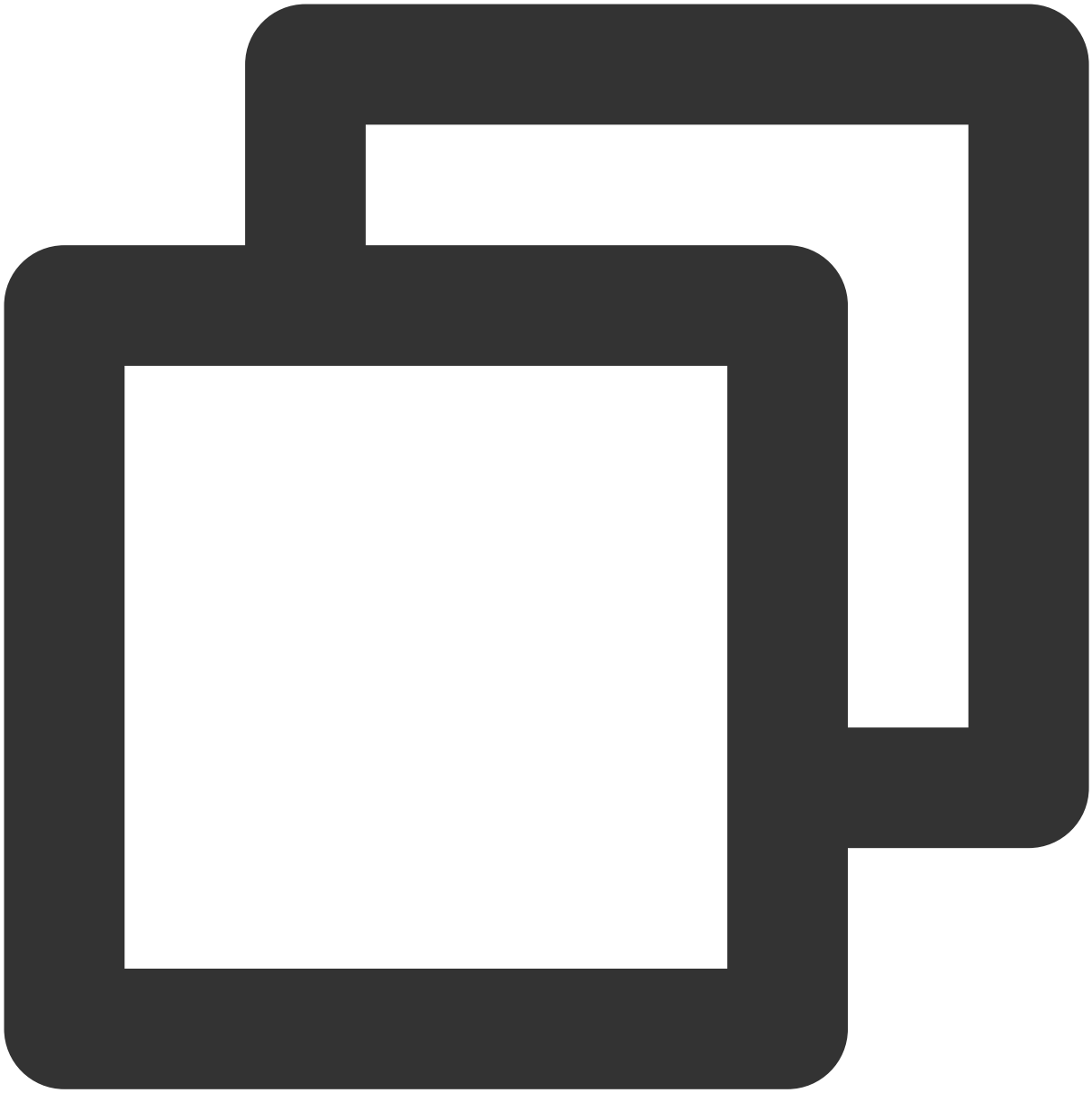
独立任务

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

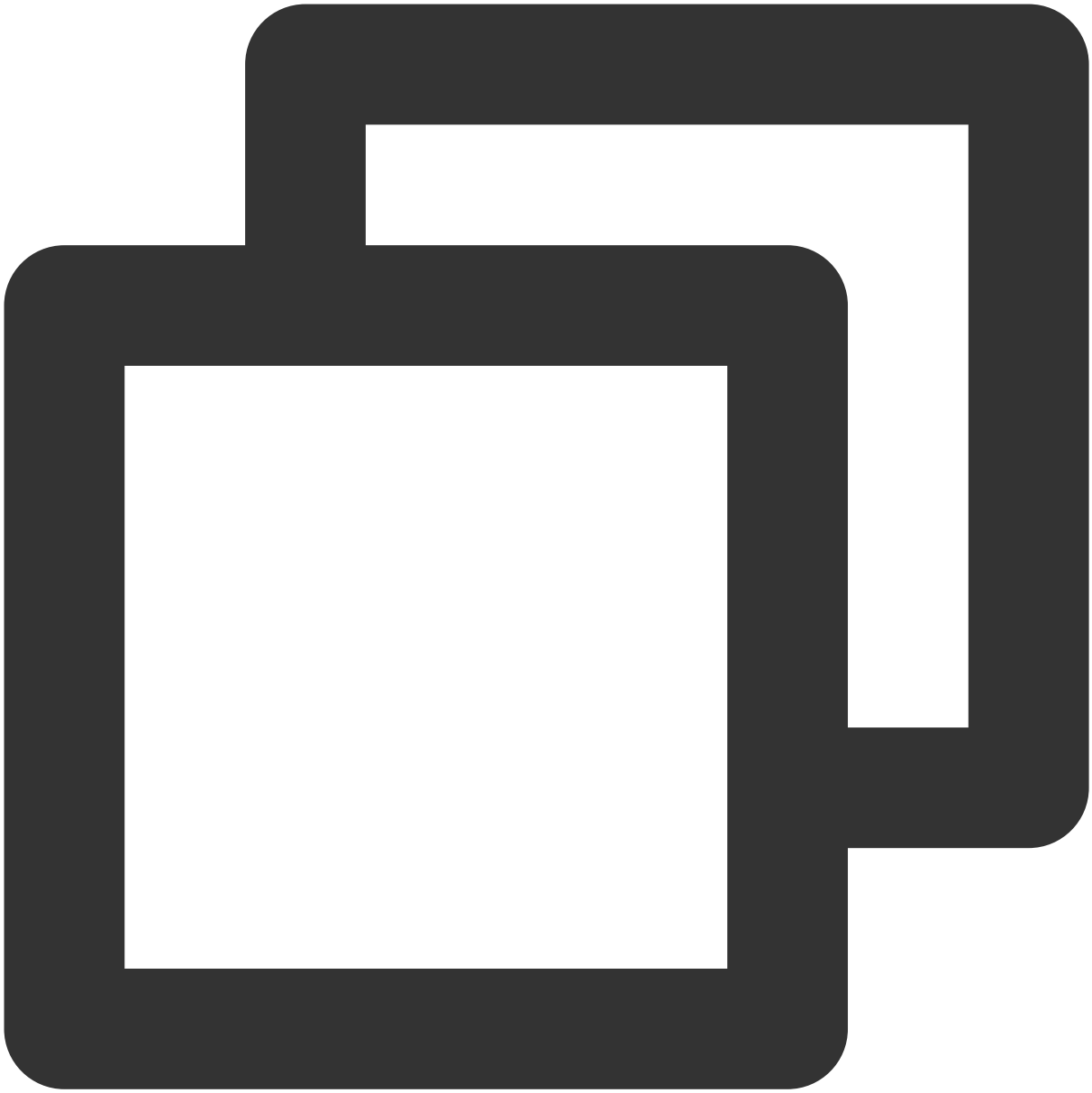
实际案例

请求



```
GET /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <RequestId>NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18xYQ==</RequestId>
    <JobsDetail>
```

```
        <Code>Success</Code>
        <Type>Workflow</Type>
        <Message/>
        <Name>demo</Name>
        <JobId>b3deffea2f84911ec9cb15254008618d9</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
        </Operation>
    </JobsDetail>
    <JobsDetail>
        <Code>Success</Code>
        <Type>Job</Type>
        <Message/>
        <Name>demo</Name>
        <JobId>be8f65004eb8511eaaed4f377124a303c</JobId>
        <State>Running</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>2022-06-27T15:23:11+0800</StartTime>
        <EndTime>2022-06-27T15:25:10+0800</EndTime>
        <Input>
            <Prefix>input</Prefix>
        </Input>
        <Operation>
            <TimeInterval>
                <Start>2022-02-01T12:00:00+0800</Start>
                <End>2022-05-01T12:00:00+0800</End>
            </TimeInterval>
            <QueueId>p893bcda225bf4945a378da6662e81a89</QueueId>
            <UserData>this is my inventorytriggerjob</UserData>
            <CallBack>https://www.callback.com</CallBack>
            <JobParam>
                <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            </JobParam>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-1234567890</Bucket>
```

```
                <Object>output/${InventoryTriggerJobId}/out.mp4</Object>
            </Output>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
    <NextToken>25</NextToken>
</Response>
```

回调内容
工作流回调内容

最近更新时间：2023-12-30 16:22:21

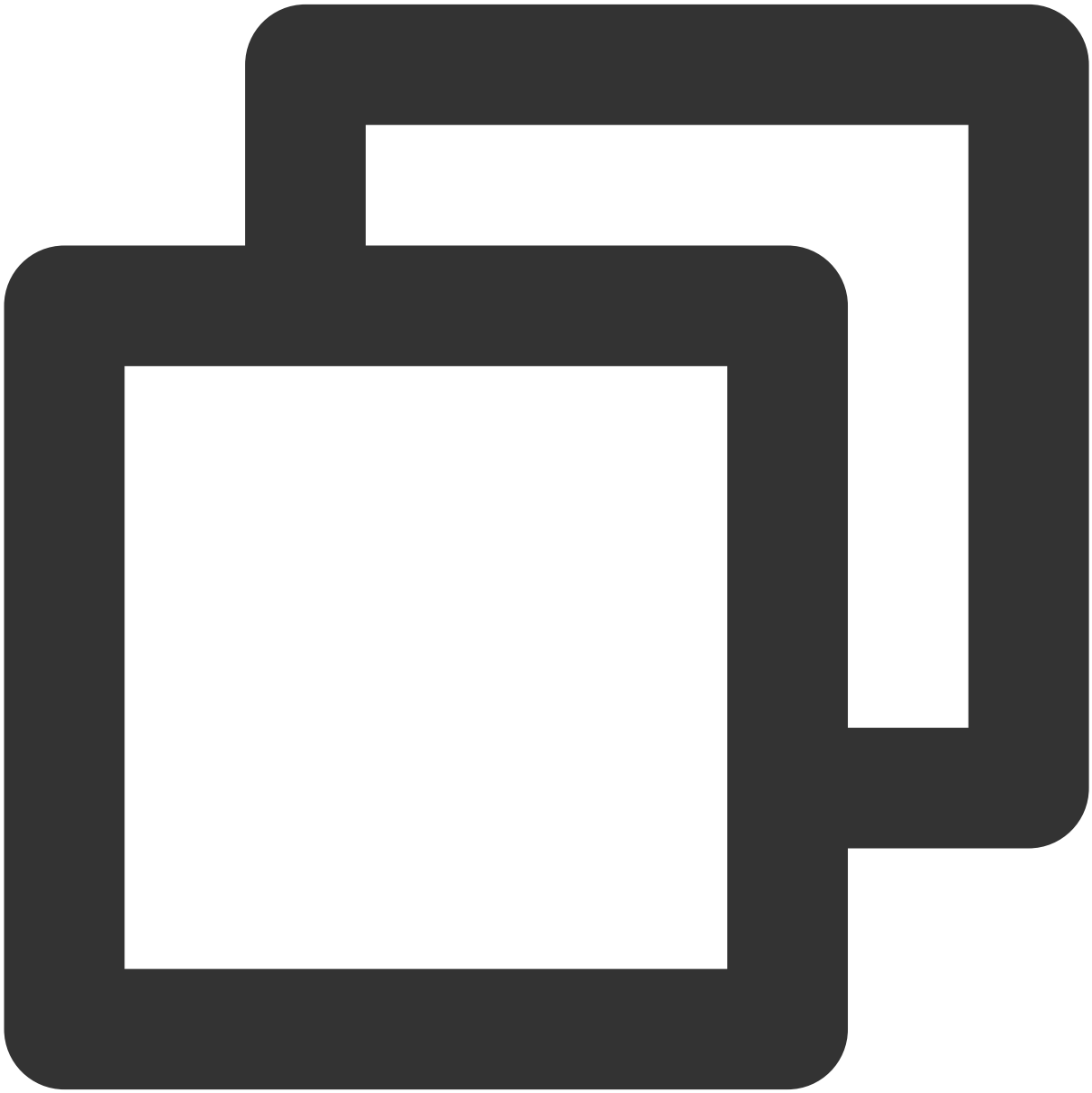
功能说明

数据万象支持自定义设置回调 URL，在工作流开启状态下，每执行一个实例后，系统会向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解工作流实例处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
        <WorkflowExecution>
            <RunId></RunId>
            <BucketId></BucketId>
            <Object></Object>
            <CosHeaders>
                <Key></Key>
                <Value></Value>
            </CosHeaders>
            <WorkflowId></WorkflowId>
            <WorkflowName></WorkflowName>
```

```
            <CreateTime></CreateTime>
            <State></State>
            <Tasks>
                    <Type></Type>
                    <CreateTime></CreateTime>
                    <EndTime></EndTime>
                    <State></State>
                    <JobId></JobId>
                    <Name></Name>
                    <TemplateId></TemplateId>
                    <TemplateName></TemplateName>
            </Tasks>
        </WorkflowExecution>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
WorkflowExecution	Response	工作流的详细信息	Container

Container 节点 WorkflowExecution 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.WorkflowExecution	工作流实例的 ID	String
BucketId	Response.WorkflowExecution	被处理文件所属的存储桶	String
Object	Response.WorkflowExecution	该实例处理的文件名	String
CosHeaders	Response.WorkflowExecution	该 Object 上传时设定的自定义 Header 列表，没有设置时无此内容	Container
State	Response.WorkflowExecution	实例的状态，值为 Success、Failed 其中一个。Tasks 存在一个 Failed，则此实例视为 Failed	String
CreateTime	Response.WorkflowExecution	实例的创建时间	String
WorkflowId	Response.WorkflowExecution	工作流 ID	String

WorkflowName	Response.WorkflowExecution	工作流的名称	String
Tasks	Response.WorkflowExecution	任务信息列表	Container

Container 节点 CosHeaders 的内容：

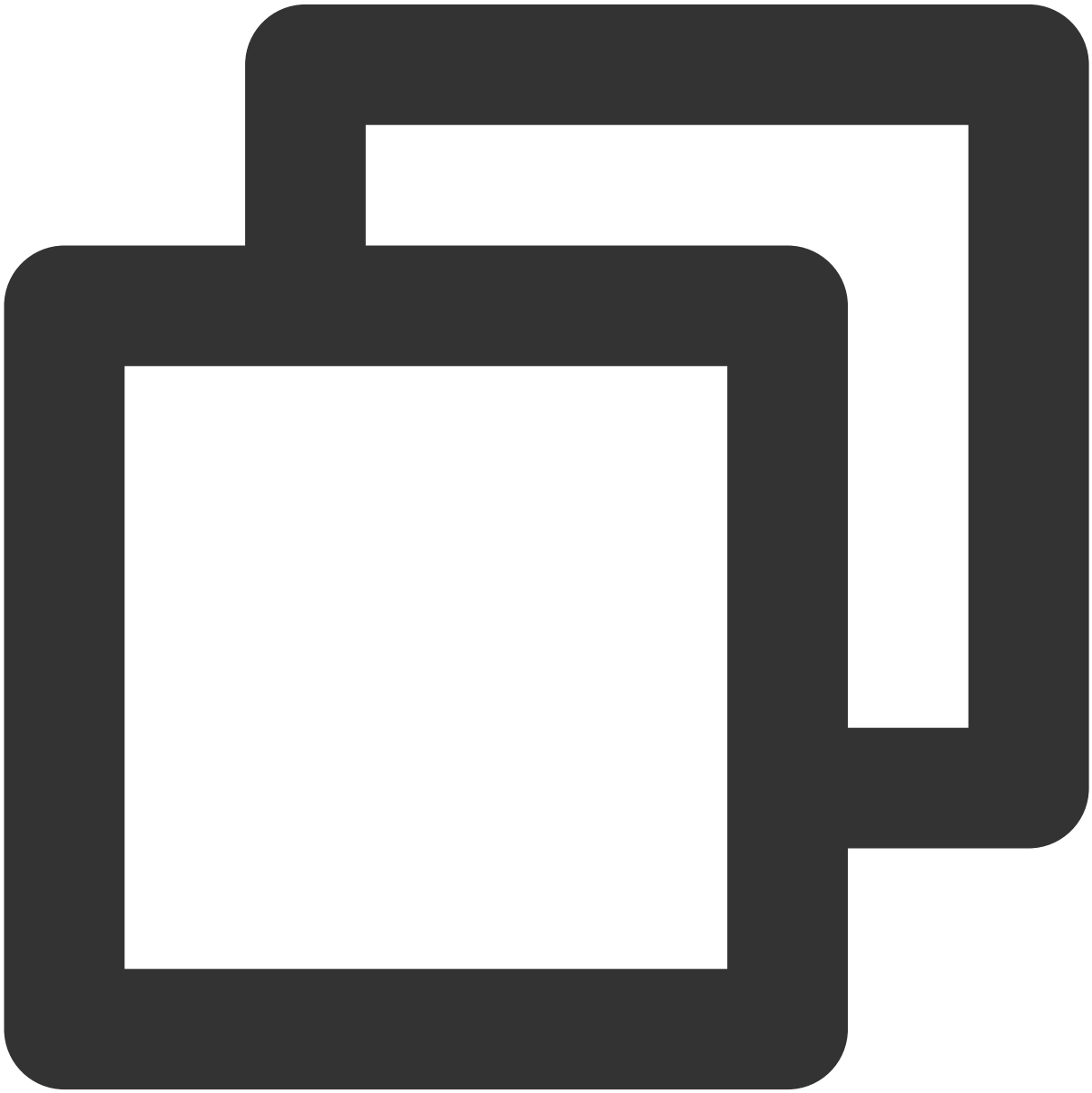
节点名称（关键字）	父节点	描述	类型
Key	Response.WorkflowExecution.CosHeaders	自定义 Header 的名称	String
Value	Response.WorkflowExecution.CosHeaders	自定义 Header 的值	String

Container 节点 Tasks 的内容：

节点名称（关键字）	父节点	描述	类型
Type	Response.WorkflowExecution.Tasks	任务类型，值为 Snapshot、Transcode、SmartCover、Concat 等	String
Createtime	Response.WorkflowExecution.Tasks	任务的创建时间	String
EndTime	Response.WorkflowExecution.Tasks	任务的完成时间	String
State	Response.WorkflowExecution.Tasks	任务的状态，值为 Success、Failed 等	String
JobId	Response.WorkflowExecution.Tasks	任务的 ID	String
Name	Response.WorkflowExecution.Tasks	任务在工作流中的节点名称	String
TemplateId	Response.WorkflowExecution.Tasks	任务使用的模板 ID	String
TemplateName	Response.WorkflowExecution.Tasks	任务使用的模板名称	String

实际案例

版权所有：腾讯云计算（北京）有限责任公司



```
<Response>
    <WorkflowExecution>
        <RunId>ie4d9187d35dc11eb9cec525400860000</RunId>
        <BucketId>examplebucket-1250000000</BucketId>
        <Object>0000.mp4</Object>
        <CosHeaders>
            <Key>x-cos-meta-id</Key>
            <Value>testid</Value>
        </CosHeaders>
        <WorkflowId>w2b22307fb0e44107bc8ffe71e1d20000</WorkflowId>
        <WorkflowName>example</WorkflowName>
```

```
        <CreateTime>2020-12-01 11:00:41+0800</CreateTime>
        <State>Success</State>
        <Tasks>
            <Type>Transcode</Type>
            <CreateTime>2020-12-01 11:00:42+0800</CreateTime>
            <EndTime>2020-12-01 11:00:45+0800</EndTime>
            <State>Success</State>
            <JobId>je5195f3e35dc11ebbe6fed17a7c10000</JobId>
            <Name>Transcode_1600413767353</Name>
            <TemplateId>t182c0ca7d91ca40969a3fc97c5559091a</TemplateId>
            <TemplateName>example</TemplateName>
        </Tasks>
        <Tasks>
            <Type>Transcode</Type>
            <CreateTime>2020-12-01 11:00:42+0800</CreateTime>
            <EndTime>2020-12-01 11:00:45+0800</EndTime>
            <State>Success</State>
            <JobId>je55e922035dc11ebbe6fed17a7c10000</JobId>
            <Name>Transcode_1600413767444</Name>
            <TemplateId>t182c0ca7d91ca40969a3fc97c5559091b</TemplateId>
            <TemplateName>example</TemplateName>
        </Tasks>
    </WorkflowExecution>
</Response>
```

任务回调内容

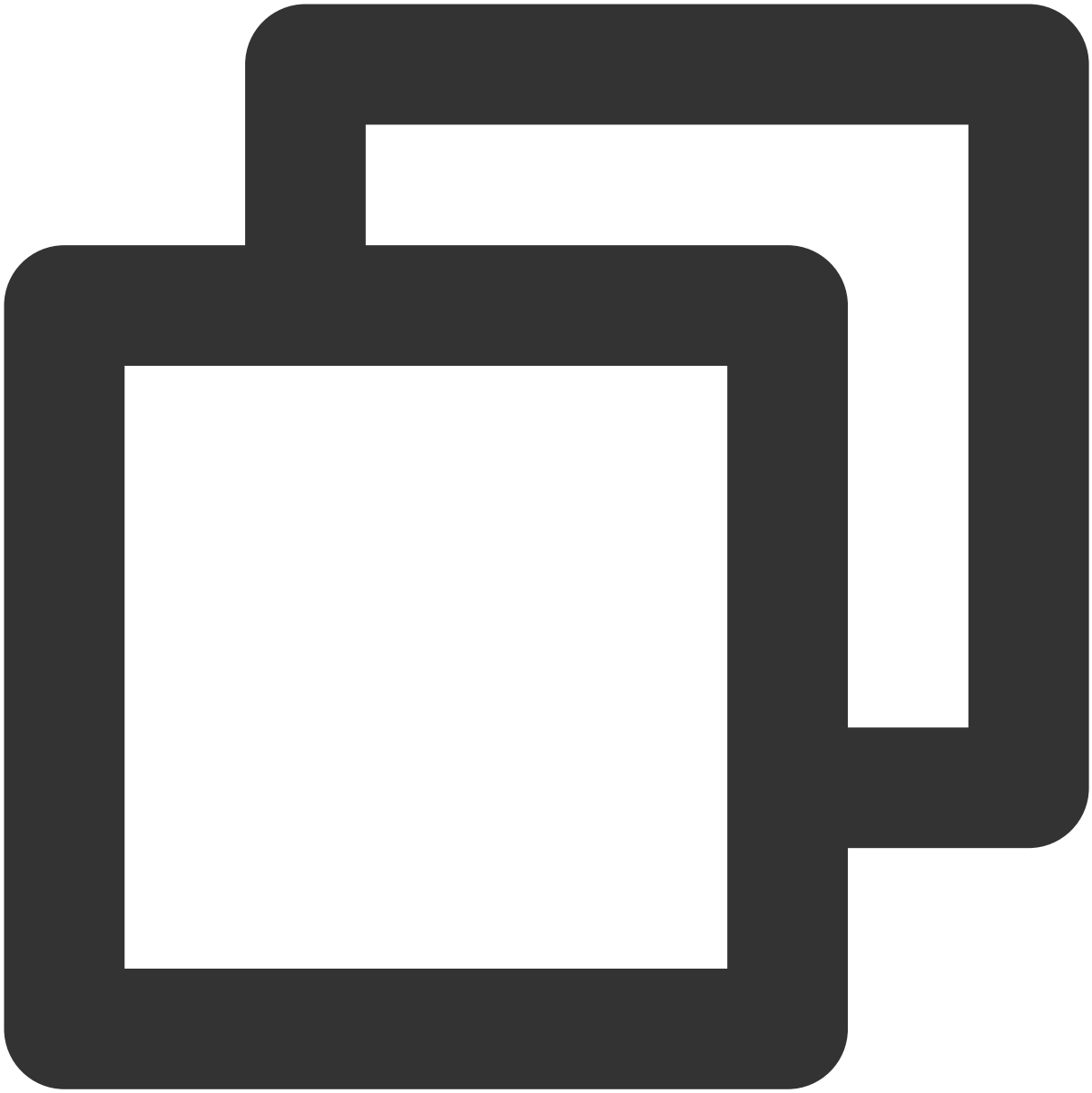
最近更新时间：2023-12-30 16:22:28

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
        <JobsDetail>
            <Code></Code>
            <Message></Message>
            <JobId></JobId>
            <State></State>
            <CreationTime></CreationTime>
            <EndTime></EndTime>
            <QueueId></QueueId>
            <Tag></Tag>
            <Input>
```

```
                <CosHeaders></CosHeaders>
                <Region></Region>
                <BucketId></BucketId>
                <Object></Object>
            </Input>
            <Operation>
                <TemplateId></TemplateId>
                <TemplateName></TemplateName>
                <MediaResult></MediaResult>
                <Output>
                   <Region></Region>
                   <Bucket></Bucket>
                   <Object></Object>
                </Output>
                <MediaInfo>
                </MeidaInfo>
            </Operation>
            <Workflow>
                <RunId></RunId>
                <WorkflowId></WorkflowId>
                <WorkflowName></WorkflowName>
                <Name></Name>
            </Workflow>
        </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String

版权所有：腾讯云计算（北京）有限责任公司

JobId	Response.JobsDetail	任务的 ID	String
Tag	Response.JobsDetail	任务的 Tag	String
State	Response.JobsDetail	任务的状态，值为 Success、Failed 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
QueueId	Response.JobsDetail	任务所属的队列 ID	String
Input	Response.JobsDetail	该任务的输入资源地址	Container
Operation	Response.JobsDetail	该任务的规则	Container
Workflow	Response.JobsDetail	该任务的所属工作流信息	Container

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型
CosHeaders	Response.JobsDetail.Input	输入文件的自定义 Header 信息，数组类型，会有多个，只有工作流触发的任务才会包含此内容。	Container
Object	Response.JobsDetail.Input	输入文件名，在 COS 中称为对象键（ObjectKey）	String
Region	Response.JobsDetail.Input	输入存储桶地域	String
BucketId	Response.JobsDetail.Input	输入存储桶 ID	String

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.JobsDetail.Operation	任务的模板 ID	String
TemplateName	Response.JobsDetail.Operation	任务的模板名称	String
Output	Response.JobsDetail.Operation	设置的文件输出地址	Container
MediaResult	Response.JobsDetail.Operation	任务的输出信息	Container
MediaInfo	Response.JobsDetail.Operation	转码输出视频的信息，没有时不返回	Container

Container 节点 Output 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.Operation.Output	存储桶的地域	String
Bucket	Response.Operation.Output	存储结果的存储桶	String
Object	Response.Operation.Output	结果文件的名称	String

Container 节点 MediaResult 的内容：

节点名称（关键字）	父节点	描述	类型
OutputFile	Response.Operation.MediaResult	实际的输出文件信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.Operation.MediaResult.OutputFile	存储桶的地域	String
Bucket	Response.Operation.MediaResult.OutputFile	存储结果的存储桶	String
ObjectPrefix	Response.Operation.MediaResult.OutputFile	输出文件名前缀	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名列表，数组类型，可能会有多个	String

注意

实际输出的文件为 Region 地域的 Bucket 下的 StringConcat(ObjectPrefix,ObjectName) 文件。

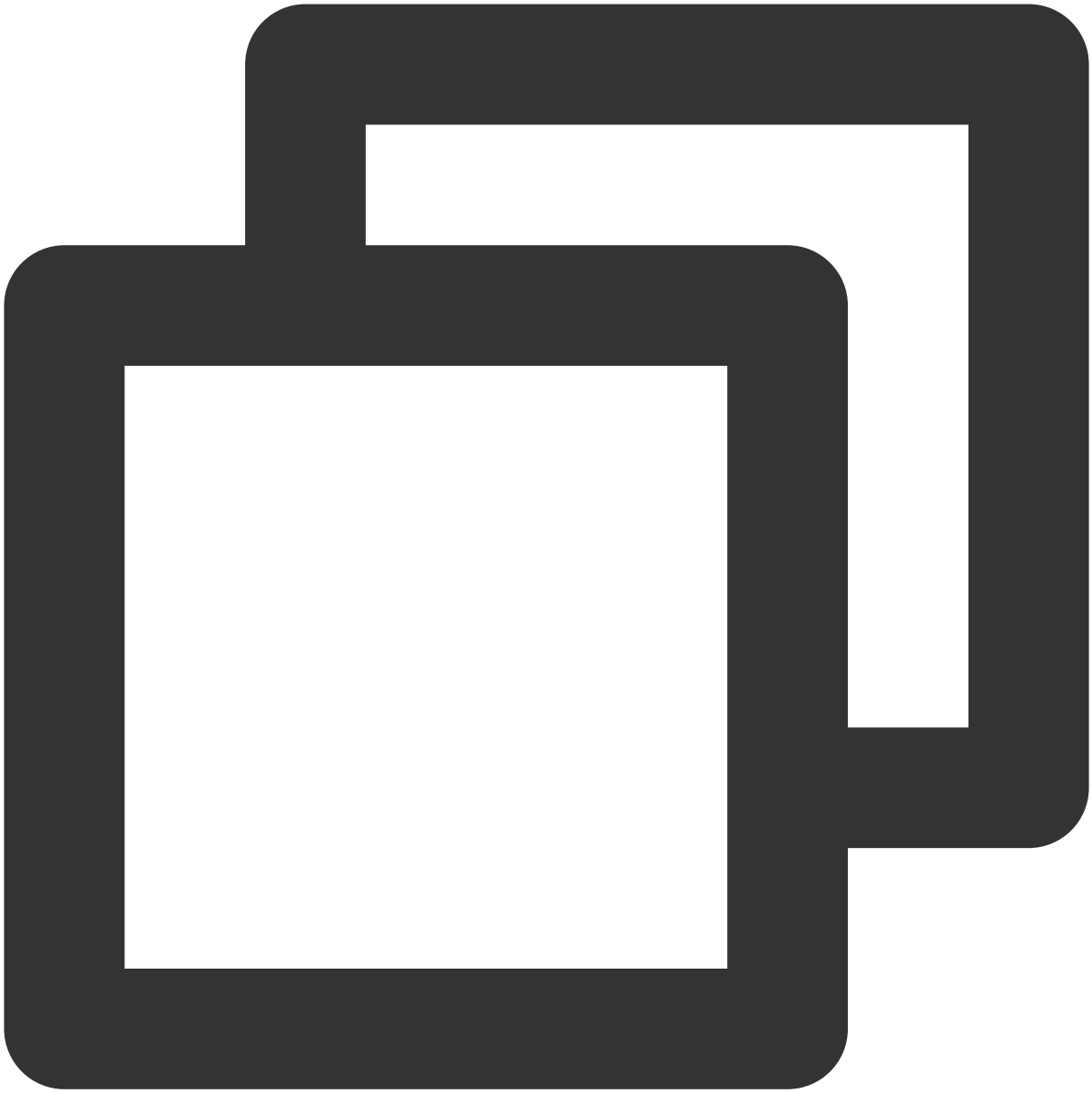
版权所有：腾讯云计算（北京）有限责任公司

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

Container 节点 MediaInfo 的内容：
同GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

实际案例



```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2020-11-16T16:43:29+0800</CreationTime>
        <EndTime>2020-11-16T16:43:33+0800</EndTime>
        <Input>
            <CosHeaders>
                <Key>x-cos-meta-test</Key>
                <Value>testvalue</Value>
            </CosHeaders>
            <CosHeaders>
```

```
                <Key>x-cos-meta-name</Key>
                <Value>xxxxxx</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-meta-age</Key>
                <Value>10</Value>
            </CosHeaders>
            <Object>1/2/3/4/5/xxx.mp4</Object>
            <Region>ap-beijing</Region>
            <BucketId>examplebucket-1250000000</BucketId>
        </Input>
        <JobId>jccddc41c27e711ebbff5874bc5b36868</JobId>
        <Message/>
        <Operation>
            <MediaResult>
                <OutputFile>
                    <Bucket>examplebucket-1250000000</Bucket>
                    <Md5Info>
                        <Md5>39ad161a7807240f4d6f94767e8e2f89</Md5>
                        <ObjectName>1/2/转mp4格式-xxx_iccba81fa27e711eb989d525400276
                    </Md5Info>
                    <ObjectName>1/2/转mp4格式-xxx_iccba81fa27e711eb989d525400276c76.
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>examplebucket-1250000000</Bucket>
                <Object>1/2/转mp4格式-xxx_iccba81fa27e711eb989d525400276c76.${ext}</
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t182c0ca7d91ca40969a3fc97c5559091a</TemplateId>
            <TemplateName>example</TemplateName>
        </Operation>
        <Workflow>
            <RunId>rccddc41c27e711ebbff5874bc5b36868</RunId>
            <WorkflowId>wccddc41c27e711ebbff5874bc5b36868</WorkflowId>
            <WorkflowName>workflow</WorkflowName>
            <Name>Transcode_1600413767444</Name>
        </Workflow>
        <QueueId>p791b0bca54ee44289f0d4b1d90796c4f</QueueId>
        <State>Success</State>
        <Tag>Transcode</Tag>
    </JobsDetail>
</Response>
```

附录
结构体

最近更新时间：2023-12-30 16:22:56

MediaWorkflow_Topology_Nodes_Start_Input

节点名称（关键字）	描述	类型	是否必选	限制
ObjectPrefix	Object 前缀	String	是	无
QueueId	队列 ID	String	是	无
PicProcessQueueId	图片处理队列 ID	String	当存在图片处理节点时必选	无
SpeechRecognitionQueueId	语音识别队列 ID	String	当存在语音识别节点时必选	无
AIProcessQueueId	AI 处理队列 ID	String	当存在 AI 识别节点时必选	无
NotifyConfig	回调信息，如果不设置，则使用队列的回调信息	Container	否	无
ExtFilter	文件后缀过滤器	Container	否	无

Container 类型 NotifyConfig 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	默认值	限制
Url	回调地址	String	是	无	不能为内网地址
Type	回调类型	String	是	无	Url：url 回调 TDMQ：tdmq 消息回调
Event	回调信息	String	是	无	TaskFinish：任务完成 WorkflowFinish：工作流完成 支持多种事件，以逗号分隔
ResultFormat	回调格式	String	否	XML	XML：xml 格式 JSON：json 格式

版权所有：腾讯云计算（北京）有限责任公司

MqRegion	TDMQ 所属园区	String	否	无	当回调类型为 TDMQ 时必填，支持园区详见 TDMQ请求地域说明文档
MqMode	TDMQ 使用模式	String	否	无	当回调类型为 TDMQ 时必填。 Topic：主题订阅 Queue：队列服务
MqName	TDMQ 主题名称	String	否	无	当回调类型为 TDMQ 时必填。

Container 类型 ExtFilter 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	默认值	限制
State	开关	String	否	Off	On/Off
Video	打开视频后缀限制	String	否	false	false/true
Audio	打开音频后缀限制	String	否	false	false/true
Image	打开图片后缀限制	String	否	false	false/true
ContentType	打开 ContentType 限制	String	否	false	false/true
Custom	打开自定义后缀限制	String	否	false	false/true
CustomExts	自定义后缀	String	否	无	多种文件后缀以/分隔，后缀个数不超过10个 当 Custom 为 true 时，该参数必填
AllFile	所有文件	String	否	false	false/true

MediaWorkflow_Topology_Nodes_***_Operation_Output

节点名称（关键字）	描述	类型	是否必选	限制
Region	存储桶的地域	String	是	无
Bucket	存储桶的名称	String	是	无

版权所有：腾讯云计算（北京）有限责任公司

Object	结果文件名称	String	是	当工作流节点类型为 Snapshot 或 SmartCover，且结果文件不止一个时，必须包含 ${Number} $ 当工作流节点类型为 Segment，设置了 Duration，且 Format 不为 HLS 或 m3u8 时，必须包含 ${Number}
SpriteObject	雪碧图的名称	String	否	当工作流节点类型为 Snapshot，且打开了雪碧图时，该字段必填
AuObject	人声结果文件名称	String	否	当工作流节点类型为 VoiceSeparate，且有人声输出时，该字段必填

MediaWorkflow_Topology_Nodes_Snapshot_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	截图模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_Animation_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	视频转动图模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_SmartCover_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	智能封面模板 ID	String	否	无
SmartCover	智能封面参数	Container，详情请参见 SmartCover	否	无
Output	输出地址	Container，详情请参见Output	是	无

版权所有：腾讯云计算（北京）有限责任公司

说明

TemplateId 和 SmartCover 都可以不设置，默认以原视频大小生成3张封面。如果同时设置 TemplateId 和 SmartCover，TemplateId 优先生效。

MediaWorkflow_Topology_Nodes_Transcode_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	音视频转码模板 ID	String	否	无
WatermarkTemplateId	水印模板 ID	String 数组	否	可以使用多个水印模板，不超过3个
RemoveWatermark	去除水印参数	Container，详情请参见 RemoveWatermark	否	无
DigitalWatermark	数字水印参数	Container，详情请参见 DigitalWatermark	否	无
Output	输出地址	Container，详情请参见 Output	是	无
Input	输入配置	Container	否	无

Container 类型 MediaWorkflow_Topology_Nodes_Transcode_***_Operation.Input 的具体数据描述如下：

节点名称（关键字）	描述	类型	必选	限制
SpeedTranscodingQueueId	倍速转码队列	String	否	无

MediaWorkflow_Topology_Nodes_Concat_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	拼接模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

版权所有：腾讯云计算（北京）有限责任公司

MediaWorkflow_Topology_Nodes_VoiceSeparate_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	人声分离模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_VideoMontage_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	精彩集锦模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_SDRtoHDR_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
SDRtoHDR	SDRtoHDR 配置	Container	是	无
TranscodeTemplateId	音视频转码模板 ID	String	是	无
WatermarkTemplateId	水印模板 ID	String 数组	否	可以使用多个水印模板，不超过3个
Output	输出地址	Container，详情请参见 Output	是	无

Container 类型 SDRtoHDR 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
HdrMode	HDR 标准	String	是	HLG HDR10

版权所有：腾讯云计算（北京）有限责任公司

MediaWorkflow_Topology_Nodes_VideoProcess_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	视频增强模板 ID	String	是	无
TranscodeTemplateId	音视频转码模板 ID	String	是	无
WatermarkTemplateId	水印模板 ID	String 数组	否	可以使用多个水印模板，不超过3个
DigitalWatermark	数字水印参数	Container，详情请参见 DigitalWatermark	否	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_SCF_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
SCF	SCF 函数信息	Container	是	无

Container 类型 SCF 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
Region	函数地域	String	是	无
FunctionName	函数名称	String	是	无
Namespace	命名空间	String	否	无
Alias	函数别名	String	否	无

MediaWorkflow_Topology_Nodes_SuperResolution_***_Operation

节点名称（关键字）	描述	类型	是否必	限制

版权所有：腾讯云计算（北京）有限责任公司

			选	
TemplateId	超级分辨率模板 ID	String	是	无
TranscodeTemplateId	音视频转码模板 ID	String	是	无
WatermarkTemplateId	水印模板 ID	String 数组	否	可以使用多个水印模板，不超过3个
DigitalWatermark	数字水印参数	Container，详情请参见 DigitalWatermark	否	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_Segment_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
Segment	音视频转封装参数	Container	是	无
Output	输出地址	Container，详情请参见 Output	是	无

Container 类型 Segment 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
Format	封装格式	String	是	aac、mp3、flac、mp4、ts、mkv、avi、hls、m3u8
Duration	转封装时长,单位:秒	String	否	不小于5的整数
HlsEncrypt	hls 加密配置	Container	否	只有当封装格式为 hls 时生效，详情请参见 HlsEncrypt

MediaWorkflow_Topology_Nodes_PicProcess_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制

TemplateId	图片处理模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_Tts_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	语音合成模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_SpeechRecognition_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
Output	输出地址	Container，详情请参见 Output	是	无
StreamPackConfig	打包配置	Container	是	无

MediaWorkflow_Topology_Nodes_StreamPackConfig_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
StreamPackConfig	打包配置	Container	否	无

Container 类型 StreamPackConfig 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
PackType	打包类型。默认值：HLS	String	否	HLS/DASH
IgnoreFailedStream	忽略转码失败的子流，继续打包。默认值：true	String	否	true/false

版权所有：腾讯云计算（北京）有限责任公司

MediaWorkflow_Topology_Nodes_VideoStream_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	音视频转码模板 ID	String	否	无
WatermarkTemplateId	水印模板 ID	String 数组	否	可以使用多个水印模板，不超过3个
RemoveWatermark	去除水印参数	Container，详情请参见 RemoveWatermark	否	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_StreamPack_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
StreamPackInfo	打包规则	Container	否	无

Container 类型 StreamPackInfo 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
VideoStreamConfig	视频子流配置	Container	否	无

Container 类型 VideoStreamConfig 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
VideoStreamName	视频子流名称	String	是	必须和存在的视频节点对应
BandWidth	视频子流带宽限制，单位b/s，范围[0, 2000000000]，0表示不限制	String	否	大于等于0，默认值是0

AudioMix

节点名称（关键字）	描述	类型	必选	默认值	限制
AudioSource	需要被混音的音轨媒体地址	String	是	无	需与Input媒体文件存储于同一bucket
MixMode	混音模式	String	否	Repeat	Repeat:背景音循环 Once:背景音一次播放
Replace	是否用混音音轨媒体替换Input媒体文件的原音频	String	否	false	true/false
EffectConfig	混音淡入淡出配置	Container	否	false	无

Container 类型 EffectConfig 的具体数据描述如下：

节点名称（关键字）	描述	类型	必选	默认值	限制
EnableStartFadein	开启淡入	String	否	false	true/false
StartFadeinTime	淡入时长	String	否	无	大于0, 支持浮点数
EnableEndFadeout	开启淡出	String	否	false	true/false
EndFadeoutTime	淡出时长	String	否	无	大于0, 支持浮点数
EnableBgmFade	开启bgm转换淡入	String	否	false	true/false
BgmFadeTime	bgm转换淡入时长	String	否	无	大于0, 支持浮点数

CallBackMqConfig

节点名称（关键字）	描述	类型	是否必选
MqRegion	消息队列所属园区，目前支持园区 sh（上海）、bj（北京）、gz（广州）、cd（成都）、hk（中国香港）	String	是
MqMode	消息队列使用模式，默认 Queue ： 主题订阅：Topic 队列服务: Queue	String	是
MqName	TDMQ 主题名称	String	是

DigitalWatermark

节点名称（关键字）	描述	类型	是否必选	限制
Message	嵌入数字水印的水印信息	String	是	长度不超过64个字符，仅支持中文、英文、数字、_、-和*
Type	数字水印类型	String	是	当前仅可设置为 Text
Version	数字水印版本	String	是	当前仅可设置为 V1
IgnoreError	当添加水印失败是否忽略错误继续执行任务	String	是	限制为 true/false
State	添加水印是否成功，执行中为 Running，成功为 Success，失败为 Failed	String	否	该字段不能主动设置，当任务提交成功时，会返回该字段

DashEncrypt

节点名称（关键字）	描述	类型	是否必选	默认值	限制
IsEncrypt	是否开启 DASH 加密	String	否	false	true/false
UriKey	DASH 加密的 key	String	否	无	当 IsEncrypt 为 true 时，该参数才有意义

HlsEncrypt

节点名称（关键字）	描述	类型	是否必选	默认值	限制
IsEncrypt	是否开启 HLS 加密	String	否	false	true/false
UriKey	HLS 加密的 key	String	否	无	当 IsEncrypt 为 true 时，该参数才有

版权所有：腾讯云计算（北京）有限责任公司

					意义

RemoveWatermark

节点名称（关键字）	描述	类型	是否必选	限制
Dx	距离左上角原点 x 偏移	String	是	值范围：[0, 4096] 单位：px
Dy	距离左上角原点 y 偏移	String	是	值范围：[0, 4096] 单位：px
Width	水印的宽度	String	是	值范围：(0, 4096] 单位：px
Height	水印的高度	String	是	值范围：(0, 4096] 单位：px

SmartCover

节点名称（关键字）	描述	类型	是否必选	默认值	限制
Format	图片格式	String	否	jpg	jpg、png 、webp
Width	宽	String	否	视频原始宽度	值范围：[128，4096] 单位：px 若只设置 Width 时，按照视频原始比例计算 Height
Height	高	String	否	视频原始高度	值范围：[128，4096] 单位：px 若只设置 Height 时，按照视频原始比例计算 Width
Count	截图数量	String	否	3	[1,10]
DeleteDuplicates	封面去重	String	否	false	true/false

版权所有：腾讯云计算（北京）有限责任公司

内容审核接口
图片审核
图片单次审核

最近更新时间：2023-12-30 10:29:33

功能描述

图片审核功能支持同步、异步请求方式，您可以通过本接口对图片文件进行内容审核。该接口属于 GET 请求。
该接口支持情况如下：

注意：

支持审核腾讯云 COS 上的图片文件。
支持审核第三方云存储厂商的图片 URL。
支持审核本地图片文件，需要先进行 base64编码。审核本地图片文件请调用 图片批量审核接口。
支持对 COS 上的图片文件或链接进行检测，通过深度学习技术，识别可能令人反感、不安全或不适宜的违规图片内容。
支持对 GIF 图进行截帧审核。
支持识别多种违规场景，包括：低俗、违法违规、色情、广告等场景。
支持多种物体检测（实体、广告台标、二维码等）及图片中文本的 OCR 文本识别。
支持根据不同的业务场景 配置自定义的审核策略。
支持用户通过配置自定义风险库，打击自定义的违规内容。

费用说明

每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1张**图片，**计2次**审核费用。
调用接口会产生图片审核费用和 COS 读请求费用。
如果图片为 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用。
不支持审核 COS 上的归档存储类型和深度归档存储类型的文件，如果需要审核此类型文件，请先 恢复归档文件。
审核第三方云存储厂商的图片 URL 时，将产生所在云厂商的下行流量费用。

限制说明

图片文件大小支持：支持审核32MB以下的图片。对于大小超过5MB的图片，您需要在调用请求时，使用 large-image-detect 参数。

版权所有：腾讯云计算（北京）有限责任公司

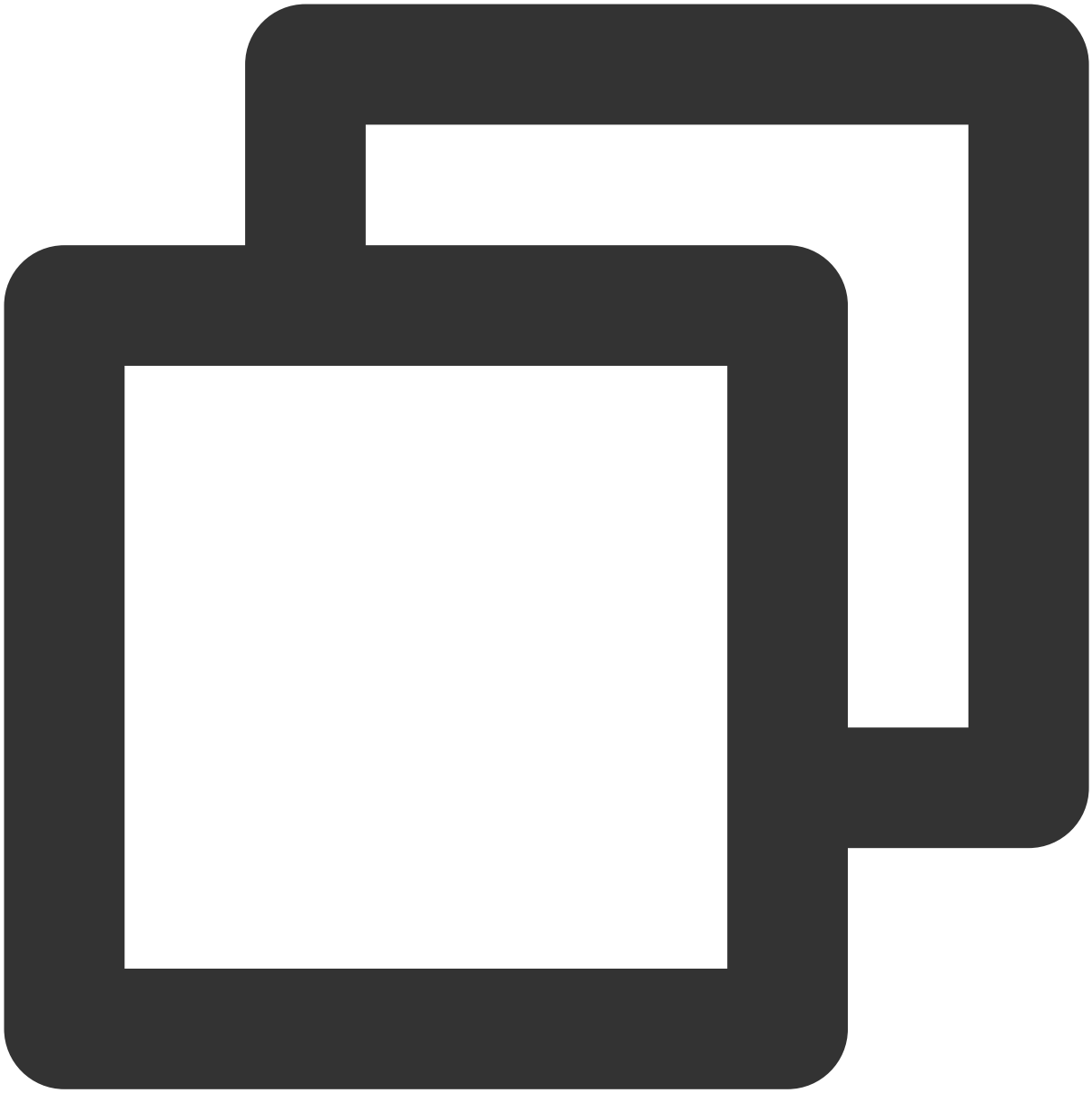
图片文件分辨率支持：建议**分辨率大于256x256**，否则可能会影响识别效果。
图片文件支持格式：PNG、JPG、JPEG、BMP、GIF、WEBP 格式。
图片文件链接支持的传输协议：HTTP、HTTPS。
调用接口需携带签名，具体规则请参见 请求签名 文档。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
GET /<ObjectKey>?ci-process=sensitive-content-recognition&detect-url=<detect-url>&i
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明：

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

具体参数如下：

参数名称	描述	类型	是否必选
ObjectKey	COS 存储桶中的图片文件名称，COS 存储桶由 `Host` 指定，例如在北京的 examplebucket-1250000000存储桶中的目录 test 下的文件 img.jpg，则 `Host` 填写 `examplebucket-1250000000.cos.ap-beijing.myqcloud.com`，`ObjectKey` 填写 `test/img.jpg`。	String	否
ci-process	标识数据处理功能的字段，内容审核的值为：sensitive-content-recognition。	String	是
biz-type	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如涉黄、广告、违法违规等，配置指引：设置公共审核策略。您可以在控制台上获取到 biz-type。biz-type 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。biz-type 不填写时，将自动使用默认的审核策略。	String	否
detect-url	您可以通过填写 `detect-url` 审核任意公网可访问的图片链接不填写 `detect-url` 时，后台会默认审核 `ObjectKey` 填写了 `detect-url` 时，后台会审核 `detect-url` 链接，无需再填写 `ObjectKey``detect-url` 示例：`http://www.example.com/abc.jpg`。	String	否
interval	审核 GIF 动图时，可使用该参数进行截帧配置，代表截帧的间隔。例如值设为5，则表示从第1帧开始截取，每隔5帧截取一帧，默认值5。	Int	否
max-frames	针对 GIF 动图审核的最大截帧数量，需大于0。例如值设为5，则表示最大截取5帧，默认值为5。	Int	否
large-image-detect	对于超过大小限制的图片是否进行压缩后再审核，取值为：0（不压缩），1（压缩）。默认为0。注：压缩最大支持32MB的图片，且会收取图片压缩费用。对于 GIF 等动态图过大时，压缩时间较长，可能会导致审核超时失败。	Int	否
dataid	图片标识，该字段在结果中返回原始内容，长度限制为512字节。	String	否
async	是否异步进行审核，取值 0：同步返回结果，1：异步进行审核，默认为0。	Int	否

版权所有：腾讯云计算（北京）有限责任公司

callback	审核结果（Detail版本）以回调形式发送至您的回调地址，异步审核时生效，支持以 `http://` 或者 `https://` 开头的地址，例如： `http://www.callback.com` 。	String	否

注意：

通过 ObjectKey 进行审核为内网操作，不会产生额外的外网流量。

通过 detect-url 进行审核，会产生图片所在源站对应的外网流量。

请求体

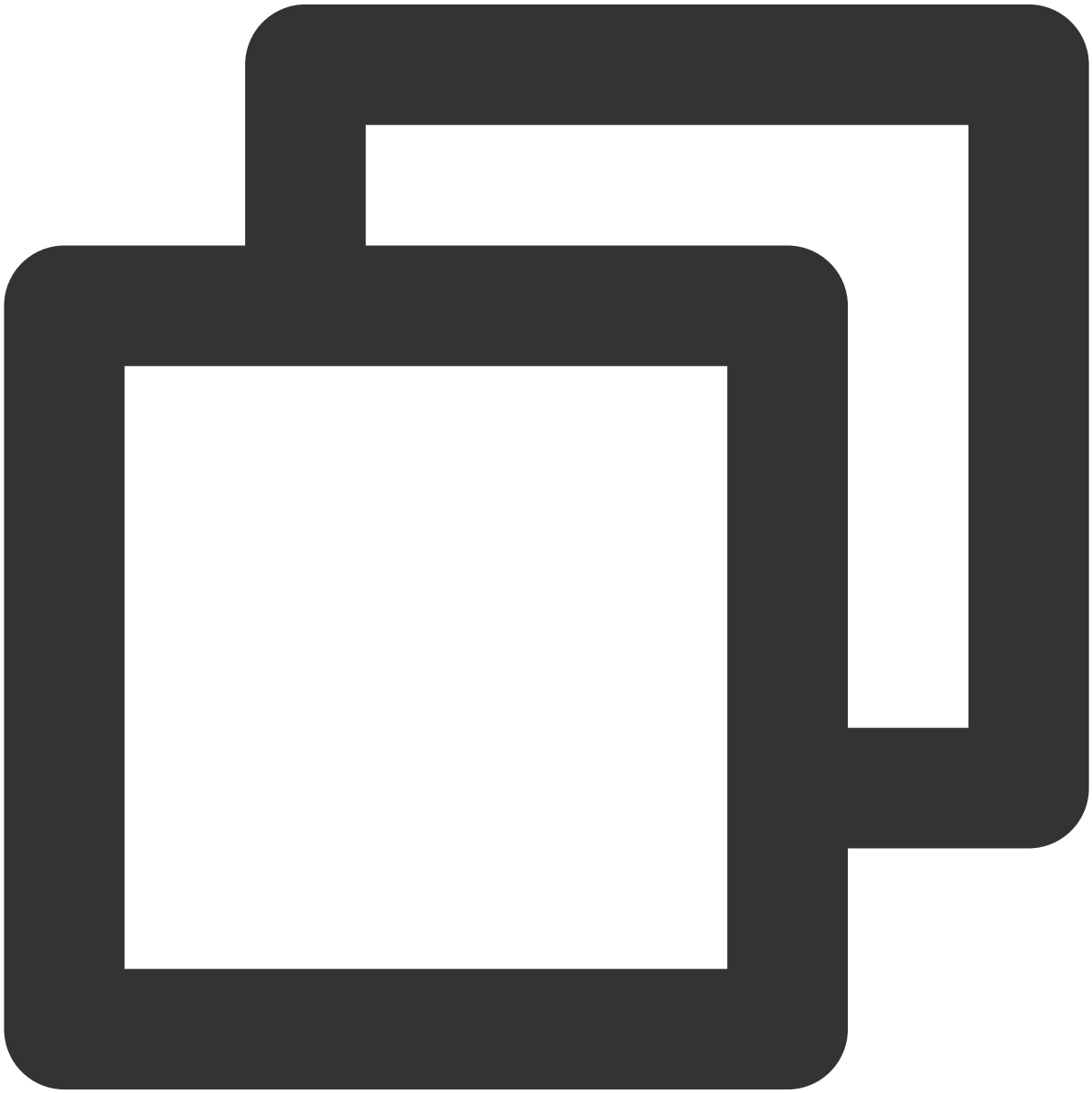
该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体



```
<RecognitionResult>
      <JobId>xxxxxxxxxxxxxxx</JobId>
      <Result>1</Result>
      <Label>Porn</Label>
      <SubLabel>SexBehavior</SubLabel>
      <Score>90</Score>
      <PornInfo>
          <Code>0</Code>
          <Msg>OK</Msg>
          <HitFlag>1</HitFlag>
          <Label>xxx</Label>
```

```
            <SubLabel>SexBehavior</SubLabel>
            <Score>100</Score>
            <OcrResults>
              <Text></Text>
              <Keywords></Keywords>
              <Location>
                <X></X>
                <Y></Y>
                <Width></Width>
                <Height></Height>
                <Rotate></Rotate>
              </Location>
            </OcrResults>
        </PornInfo>
</RecognitionResult>
```

响应包体具体数据内容如下：

参数名称	类型	描述
RecognitionResult	Container	图片审核结果。

异步审核返回的RecognitionResult 节点内容：

参数名称	类型	描述
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	String	图片审核任务的 ID。
State	String	审核任务的状态，值为Submitted（已提交审核）。
Object	String	存储在 COS 桶中的图片名称，创建任务使用 ObjectKey 时返回。
Url	String	图片文件的链接地址，创建任务使用 detect-url 时返回。

同步审核返回的RecognitionResult 节点内容：

参数名称	类型	描述
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	String	图片审核任务的 ID。
State	String	审核任务的状态，值为 Success（审核成功）。
Object	String	存储在 COS 桶中的图片名称，创建任务使用 ObjectKey 时返回。

版权所有：腾讯云计算（北京）有限责任公司

Url	String	图片文件的链接地址，创建任务使用 detect-url 时返回。
CompressionResult	Int	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。
Result	Int	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。
Label	String	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal** 表示正常，**Porn** 表示色情，**Ads** 表示广告，以及其他不安全或不适宜的类型。
Category	String	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。
SubLabel	String	该图命中的二级标签结果。
Score	Int	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。
Text	String	该图里的文字内容（OCR），当审核策略开启文本内容检测时返回。
PornInfo	Container	审核场景为**涉黄**的审核结果信息。
AdsInfo	Container	审核场景为**广告引导**的审核结果信息。

审核信息（PornInfo、AdsInfo）中包含如下内容：

参数名称	类型	描述
Code	Int	错误码，0为正确，其他数字对应相应错误。详情请参见 错误码。
Msg	String	具体错误信息，如正常则为 OK。
HitFlag	Int	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。
Score	Int	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。其中0 - 60分表示图片正常，61 - 90分表示图片疑似敏感，91 - 100分表示图片确定敏感。例如：色情 99，表明该内容非常有可能属于色情内容。
Label	String	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。

Category	String	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。
SubLabel	String	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。
OcrResults	Container Array	该字段表示 OCR 文本识别的详细检测结果，包括文本识别结果、命中的关键词等信息，有相关违规内容时返回。
LibResults	Container Array	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。

Container 节点 LibResults 的内容：

参数名称	类型	描述
ImageId	String	该字段表示命中的风险库中的图片样本 ID。
Score	Integer	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。

Container 节点 OcrResults 的内容：

参数名称	类型	描述
Text	String	图片 OCR 文本识别出的具体文本内容。
Keywords	String Array	在当前审核场景下命中的关键词。
Location	Container	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。

Container 节点 Location 的内容：

名称	类型	描述
X	Float	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Y	Float	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Width	Float	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。

版权所有：腾讯云计算（北京）有限责任公司

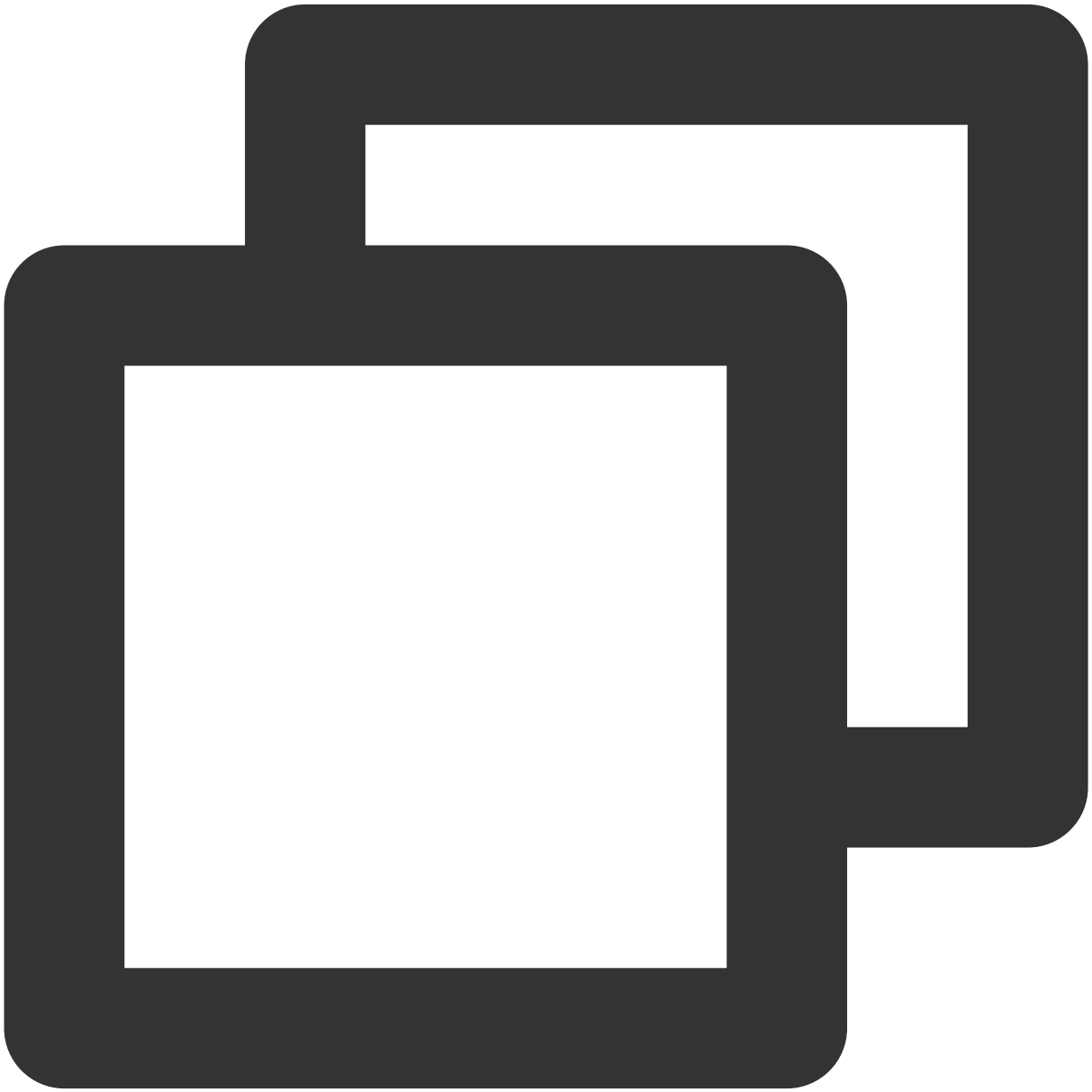
Height	Float	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Rotate	Float	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360**（**角度制**），方向为**逆时针旋转**。

Container 节点 ObjectResults 的内容：

名称	类型	描述
Name	String	该标签用于返回所识别出的实体名称，例如人名。
Location	Container	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。

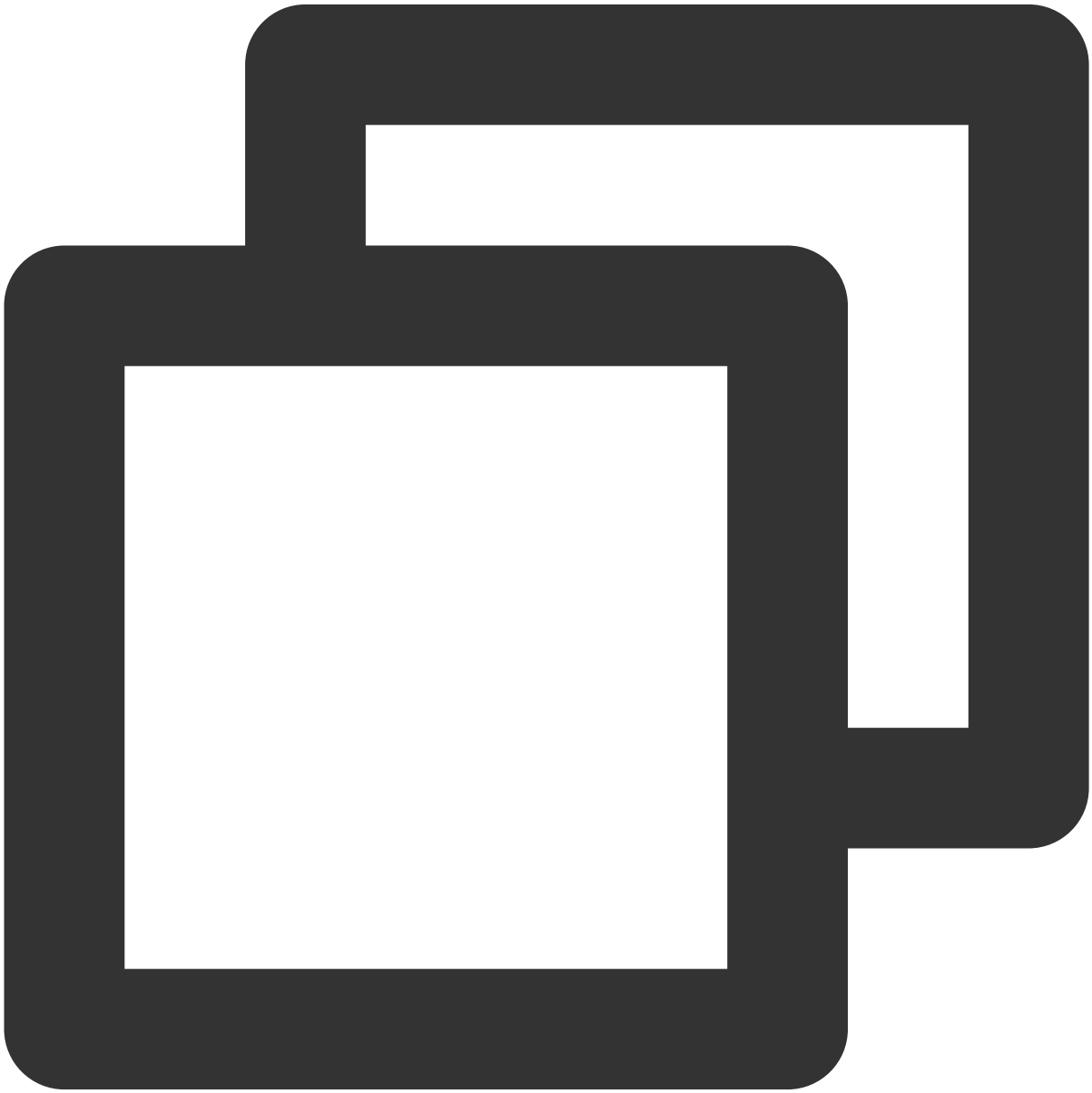
实际案例

请求1：同步图片审核



```
GET /picture.jpg?ci-process=sensitive-content-recognition&interval=0&max-frames=1&b
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX
```

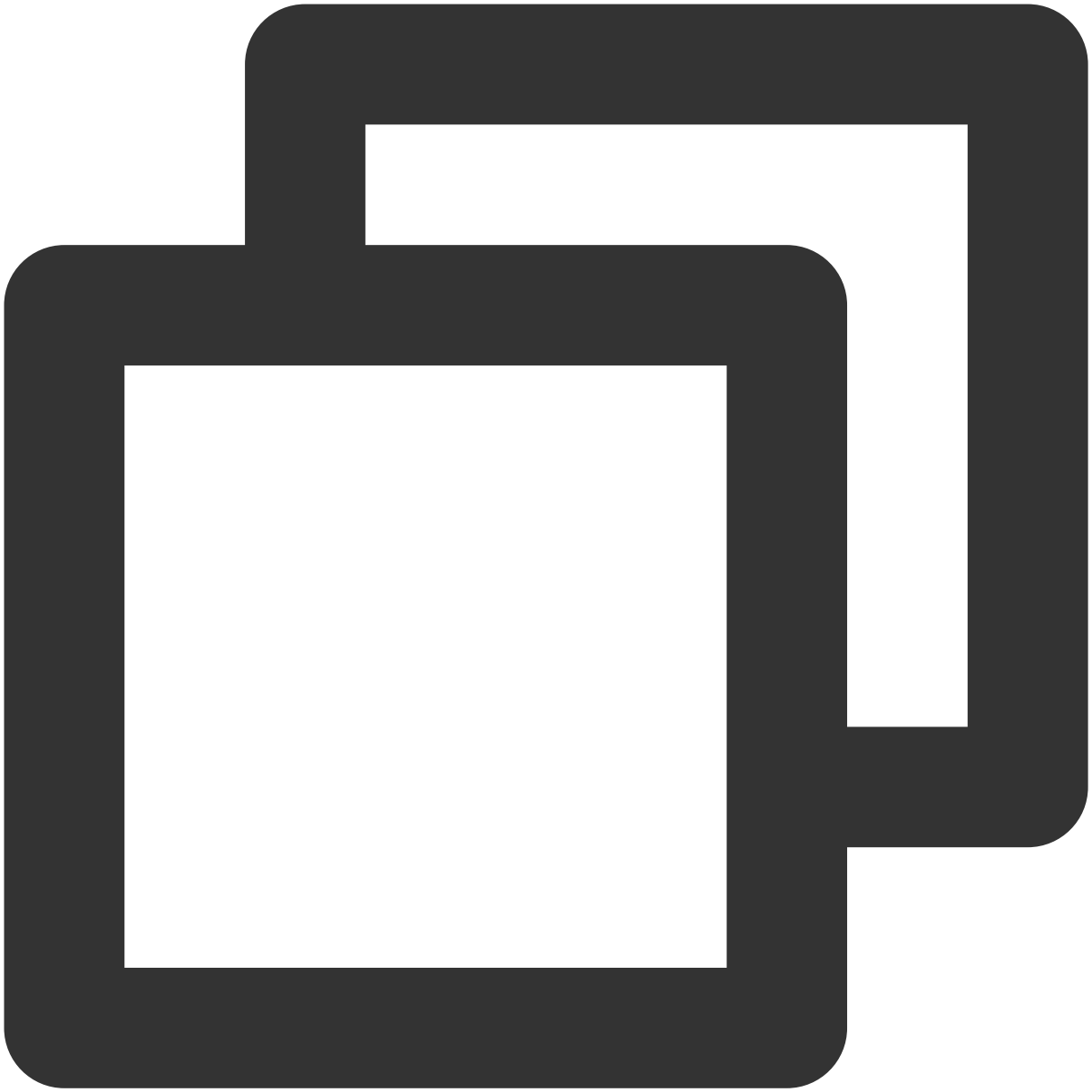
响应1



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<RecognitionResult>
      <JobId>xxxxxxxxxxxxxxx</JobId>
      <CompressionResult>0</CompressionResult>
      <Result>1</Result>
```

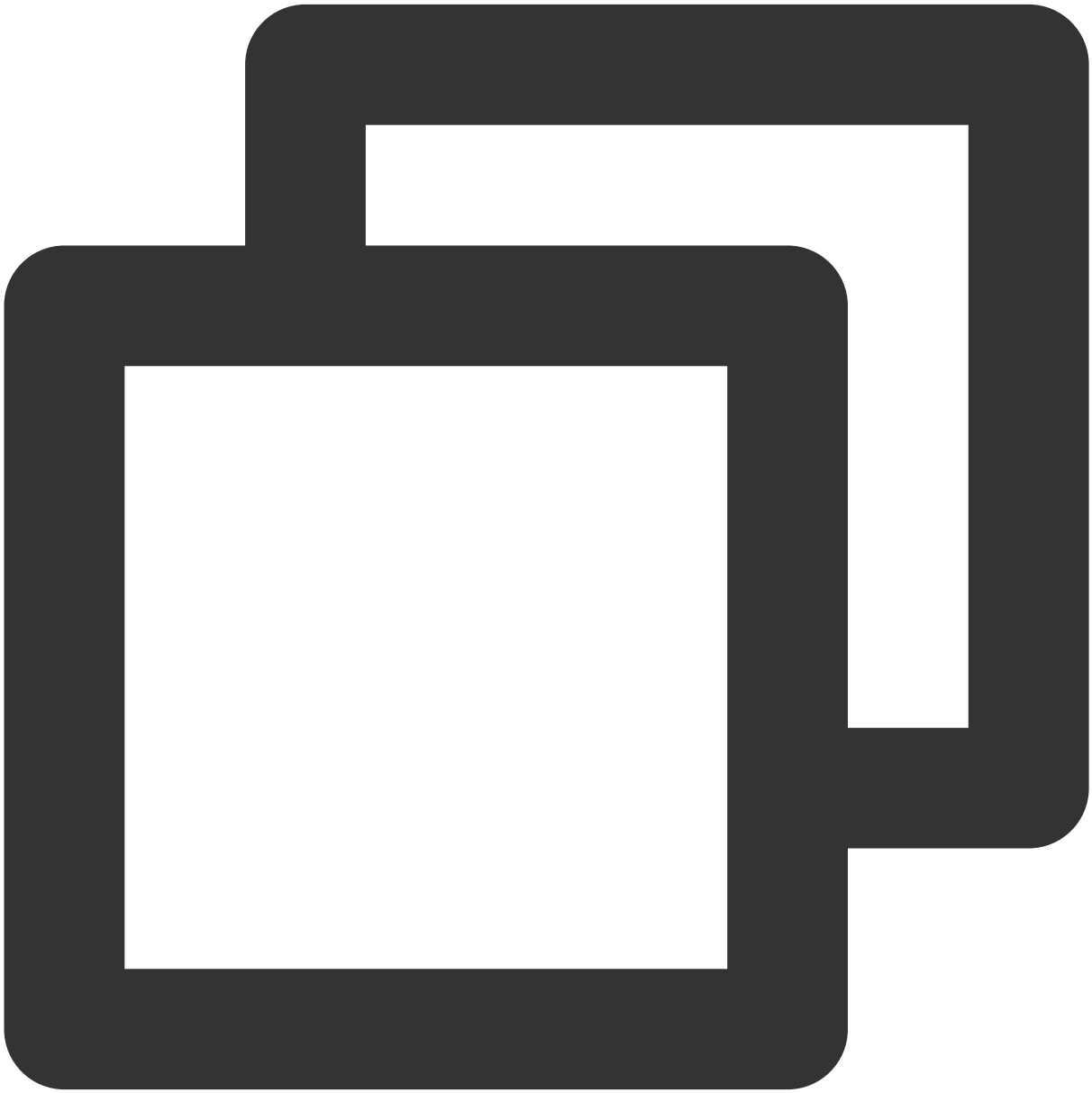
```
        <Label>Porn</Label>
        <SubLabel>SexBehavior</SubLabel>
        <Score>95</Score>
        <PornInfo>
            <Code>0</Code>
            <Msg>OK</Msg>
            <HitFlag>1</HitFlag>
            <Label>xxx</Label>
            <SubLabel>SexBehavior</SubLabel>
            <Score>95</Score>
        </PornInfo>
</RecognitionResult>
```

请求2：异步图片审核



```
GET /picture.jpg?ci-process=sensitive-content-recognition&biz-type=***&async=1&call
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX
```

响应2



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<RecognitionResult>
      <JobId>xxxxxxxxxxxxxxx</JobId>
</RecognitionResult>
```

版权所有：腾讯云计算（北京）有限责任公司

图片批量审核

最近更新时间：2023-12-30 10:30:11

功能描述

图片批量审核接口支持同步、异步请求方式，您可以通过本接口对多个图片文件进行内容审核。该接口属于 POST 请求。

说明：

支持审核腾讯云 COS 上的图片文件。

支持审核第三方云存储厂商的图片 URL。

支持审核加密数据，如有需要，请 联系服务人员 咨询。

支持根据不同的业务场景 配置自定义的审核策略。

限制说明

图片文件大小支持：支持审核32MB以下的图片。对于大小超过5MB的图片，您需要在调用请求时，使用 large-image-detect 参数。

单次请求图片数量不可超过**100**张。

图片文件分辨率支持：建议**分辨率大于256x256**，否则可能会影响识别效果。

图片文件支持格式：PNG、JPG/JPEG、BMP、GIF、WEBP、HEIF/HEIC 格式。

图片文件链接支持的传输协议：HTTP、HTTPS。

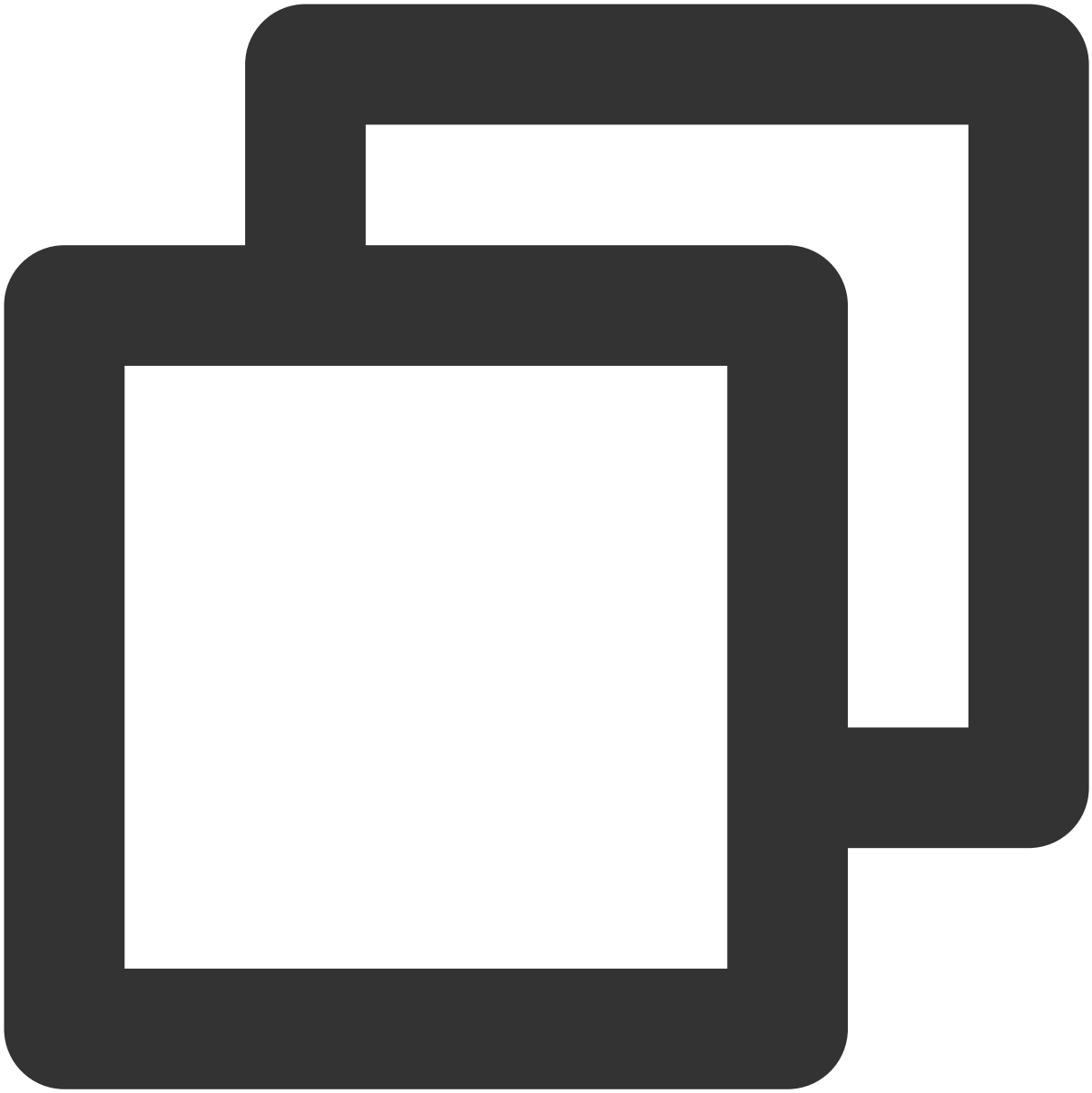
调用接口需携带签名，具体规则请参见 请求签名 文档。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
POST /image/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明：

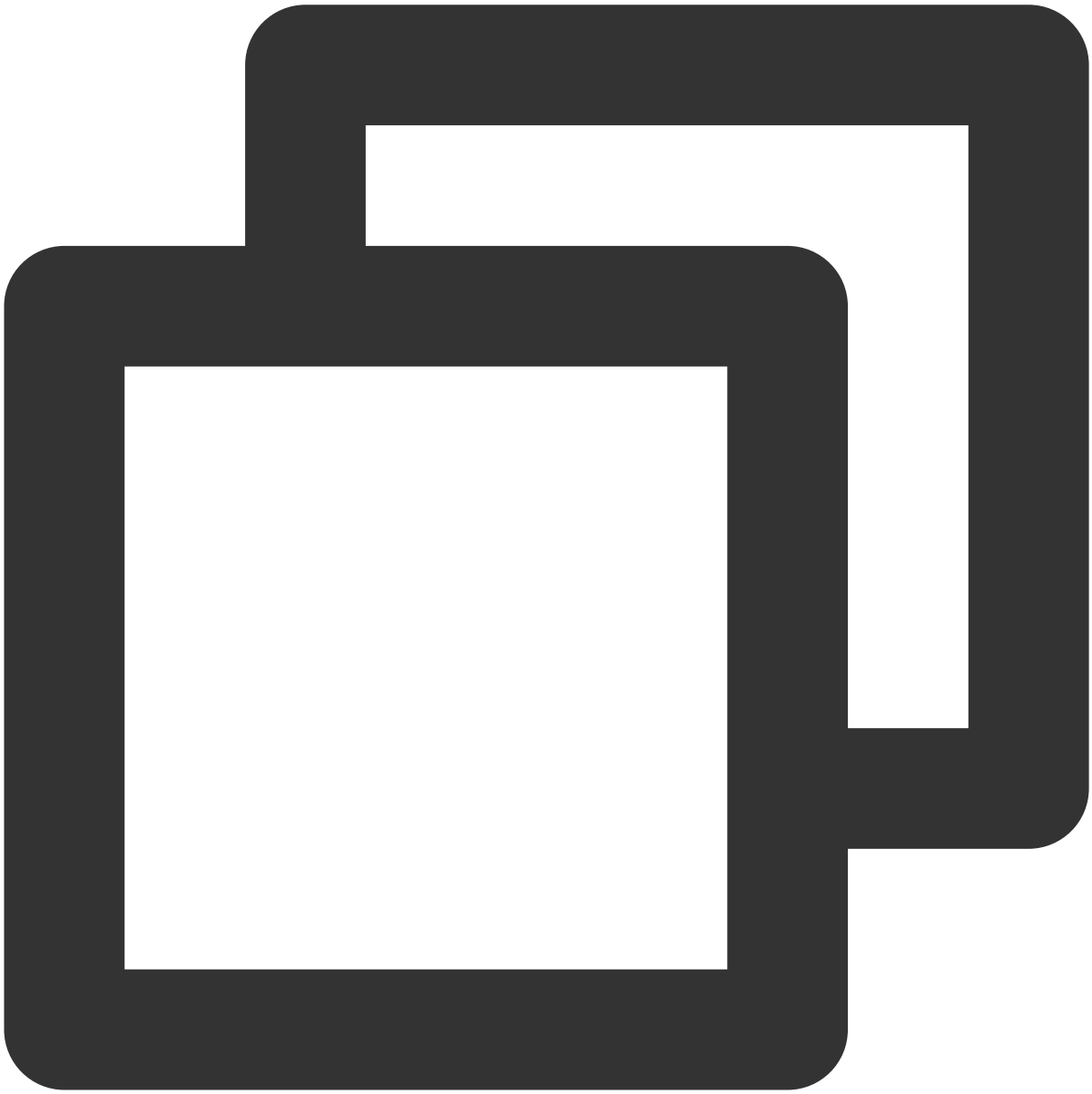
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
    <Input>
      <DataId></DataId>
      <Url></Url>
      <Object></Object>
      <Content></Content>
      <Interval></Interval>
      <MaxFrames></MaxFrames>
      <LargeImageDetect></LargeImageDetect>
      <UserInfo>
        <TokenId></TokenId>
      </UserInfo>
    </Input>
    <Input>
      <DataId></DataId>
      <Url></Url>
      <Object></Object>
      <Content></Content>
      <Interval></Interval>
      <MaxFrames></MaxFrames>
    </Input>
    <Conf>
      <BizType>b81d45f94b91a683255e9a9506f45a11</BizType>
      <Async></Async>
      <Callback></Callback>
      <Freeze>
        <PornScore></PornScore>
      </Freeze>
    </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	图片批量审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容，如有多个图片，请传入多个 Input 结构。	Container Array	是
Conf	Request	审核规则配置。	Container	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Input 的具体数据描述如下，使用其中一种：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的图片文件名称，例如在目录 test 中的文件 image.jpg，则文件名称为 test/image.jpg。Content，Object 和 Url 只能选择其中一种，传入多个时仅一个生效，按 Content，Object， Url 顺序。	String	否
Url	Request.Input	图片文件的链接地址，例如 `http://a-1250000.cos.ap-shanghai.myqcloud.com/image.jpg` 。Content，Object 和 Url 只能选择其中一种，传入多个时仅一个生效，按 Content，Object，Url 顺序。	String	否
Content	Request.Input	图片文件的内容，需要先经过 base64 编码。Content，Object 和 Url 只能选择其中一种，传入多个时仅一个生效，按 Content，Object，Url 顺序。	String	否
Interval	Request.Input	截帧频率，GIF 图检测专用，默认值为5，表示从第一帧（包含）开始每隔5帧截取一帧。	Int	否
MaxFrames	Request.Input	最大截帧数量，GIF 图检测专用，默认值为5，表示只截取 GIF 的5帧图片进行审核，必须大于0。	Int	否
DataId	Request.Input	图片标识，该字段在结果中返回原始内容，长度限制为512字节。	String	否
LargeImageDetect	Request.Input	对于超过大小限制的图片是否进行压缩后再审核，取值为：0（不压缩），1（压缩）。默认为0。注：压缩最大支持32M的图片，且会收取压缩费用。	Int	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如涉黄、广告、违法违规等，配置指引：设置公共审核策略。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Async	Request.Conf	是否异步进行审核，0：同步返回结果，1：异步进行审核。默认值为 0。	Integer	否
Callback	Request.Conf	审核结果（Detail版本）以回调形式发送至您的回调地址，异步审核时生效，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。	String	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对图片进行自动冻结，仅当 `input` 中审核的图片为 `object` 时有效。	Container	否

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请联系服务人员咨询。

注意：

通过 Object 进行审核为内网操作，不会产生额外的外网流量。

通过 Url 进行审核，会产生图片所在源站对应的外网流量。

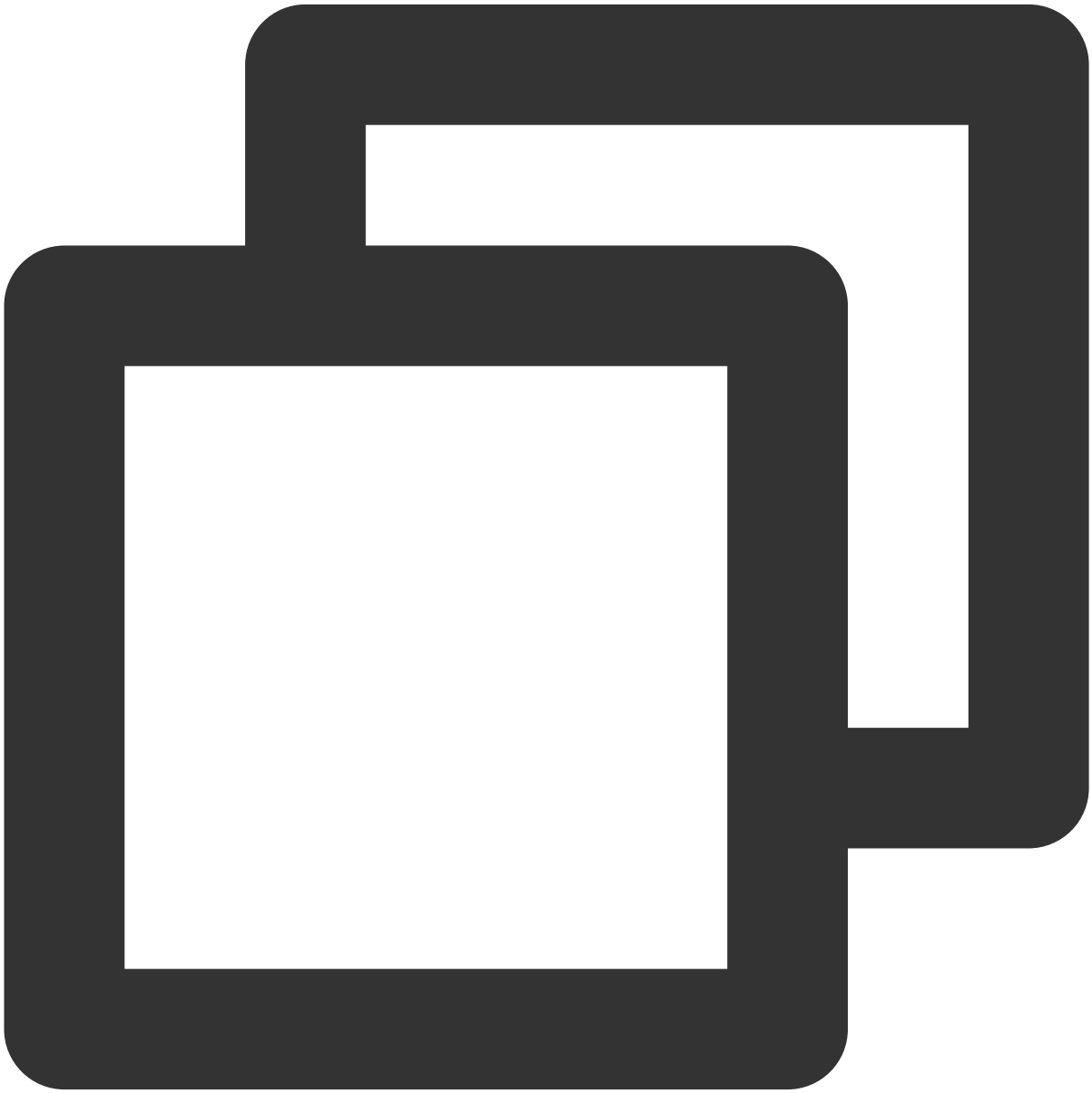
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
    <DataId></DataId>
    <Url></Url>
    <PornInfo>
      <HitFlag>1</HitFlag>
      <Score>100</Score>
      <Category>xxx</Category>
      <SubLabel>xxx</SubLabel>
    </PornInfo>
  </JobsDetail>
```

版权所有：腾讯云计算（北京）有限责任公司

```
  <JobsDetail>
    <Code>InvalidArgument</Code>
    <Message>Param MaxFrames is illegal</Message>
    <DataId></DataId>
  </JobsDetail>
</Response>
```

响应包体具体数据内容如下：

节点名称（关键字）	父节点	描述
Response	Container	图片批量审核返回的具体响应内容。

Container 节点 Response 的内容：

参数名称	类型	描述
JobsDetail	Container Array	图片审核的结果
RequestId	String	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。

异步审核结果的 JobsDetail 节点内容如下：

节点名称（关键字）	类型	描述
Code	String	错误码，只有失败时返回。详情请查看 错误码列表。
Message	String	错误描述，只有失败时返回。
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	String	本次审核任务的 ID。
State	String	审核任务的状态，值为 Submitted（已提交审核）、Failed（审核失败）其中一个。
Object	String	存储在 COS 存储桶中的图片文件名称，创建任务使用 Object 时返回。
Url	String	图片文件的链接地址，创建任务使用Url时返回。

同步审核结果的 JobsDetail 节点内容如下：

参数名称	类型	描述

Code	String	错误码，只有失败时返回。详情请查看 错误码列表。
Message	String	错误描述，只有失败时返回。
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	String	本次审核任务的 ID。
State	String	审核任务的状态，值为 Success（审核成功）、Failed（审核失败）其中一个。
Object	String	存储在 COS 存储桶中的图片文件名称，创建任务使用Object时返回。
Url	String	图片文件的链接地址，创建任务使用Url时返回。
CompressionResult	Integer	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。
Label	String	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告等。
Result	Integer	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。
Score	Integer	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。
Category	String	该图命中的审核类别结果。
SubLabel	String	该图命中的二级标签结果。
Text	String	该图里的文字内容（OCR），当审核策略开启文本内容检测时返回。
PornInfo	Container	审核场景为**涉黄**的审核结果信息。
AdsInfo	Container	审核场景为**广告引导**的审核结果信息。
UserInfo	Container	请求中设置的 UserInfo 原样返回。
ListInfo	Container	账号黑白名单结果。
ForbidState	Integer	若您在请求时设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结。

审核信息（PornInfo、AdsInfo等）中的内容如下：

参数名称	类型	描述
Code	Integer	单个审核场景的错误码，0为成功，其他为失败。详情请参见 错误码。
Msg	String	具体错误信息，如正常则为 OK。
HitFlag	Integer	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。
Score	Integer	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。
Label	String	该图的结果标签（为综合标签，可能为 SubLabel，可能为人物名字等）。
Category	String	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。
SubLabel	String	该图的二级标签结果。
OcrResults	Container Array	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。
LibResults	Container Array	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。

Container 节点 LibResults 的内容：

节点名称（关键字）	类型	描述
ImageId	String	该字段表示命中的风险库中的图片样本 ID。
Score	Integer	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，则表明该数据非常有可能命中库中的色情样本。

OcrResults 的内容如下：

节点名称（关键字）	类型	描述
Text	String	有敏感信息的 Ocr 文本内容。
Keywords	String Array	该段内容中的敏感文字。
Location	Container	该段文字在图片中的位置信息。

版权所有：腾讯云计算（北京）有限责任公司

ObjectResults 的内容如下：

节点名称（关键字）	类型	描述
Name	String	敏感内容的字称。
Location	Container	该内容在图片中的位置信息。

Location 的内容如下：

节点名称（关键字）	类型	描述
X	Float	相对于原点（图片左上角）的 X 坐标。
Y	Float	相对于原点（图片左上角）的 Y 坐标。
Height	Float	敏感区域的高度。
Width	Float	敏感区域的宽度。
Rotate	Float	敏感区域的旋转角度。

UserInfo 的内容（与请求中的UserInfo一致）：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 ListInfo 的内容：

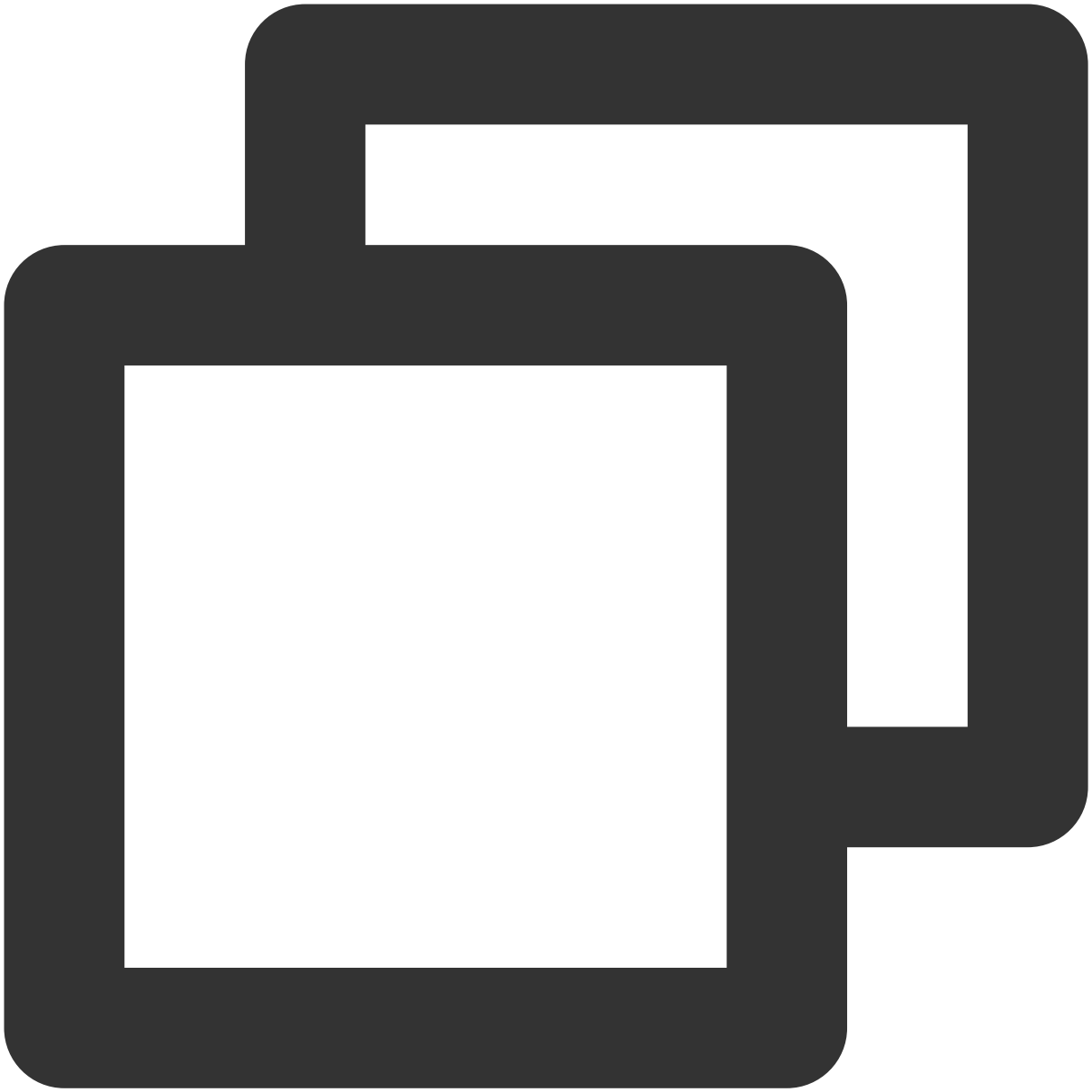
节点名称（关键字）	类型	描述
ListResults	Container Array	命中的所有名单结果。

Container 节点 ListResults 的内容：

节点名称（关键字）	类型	描述
ListType	Integer	命中的名单类型，取值为0（白名单）和1（黑名单）。
ListName	String	命中的名单名称。
Entity	String	命中了名单中的哪条内容。

实际案例

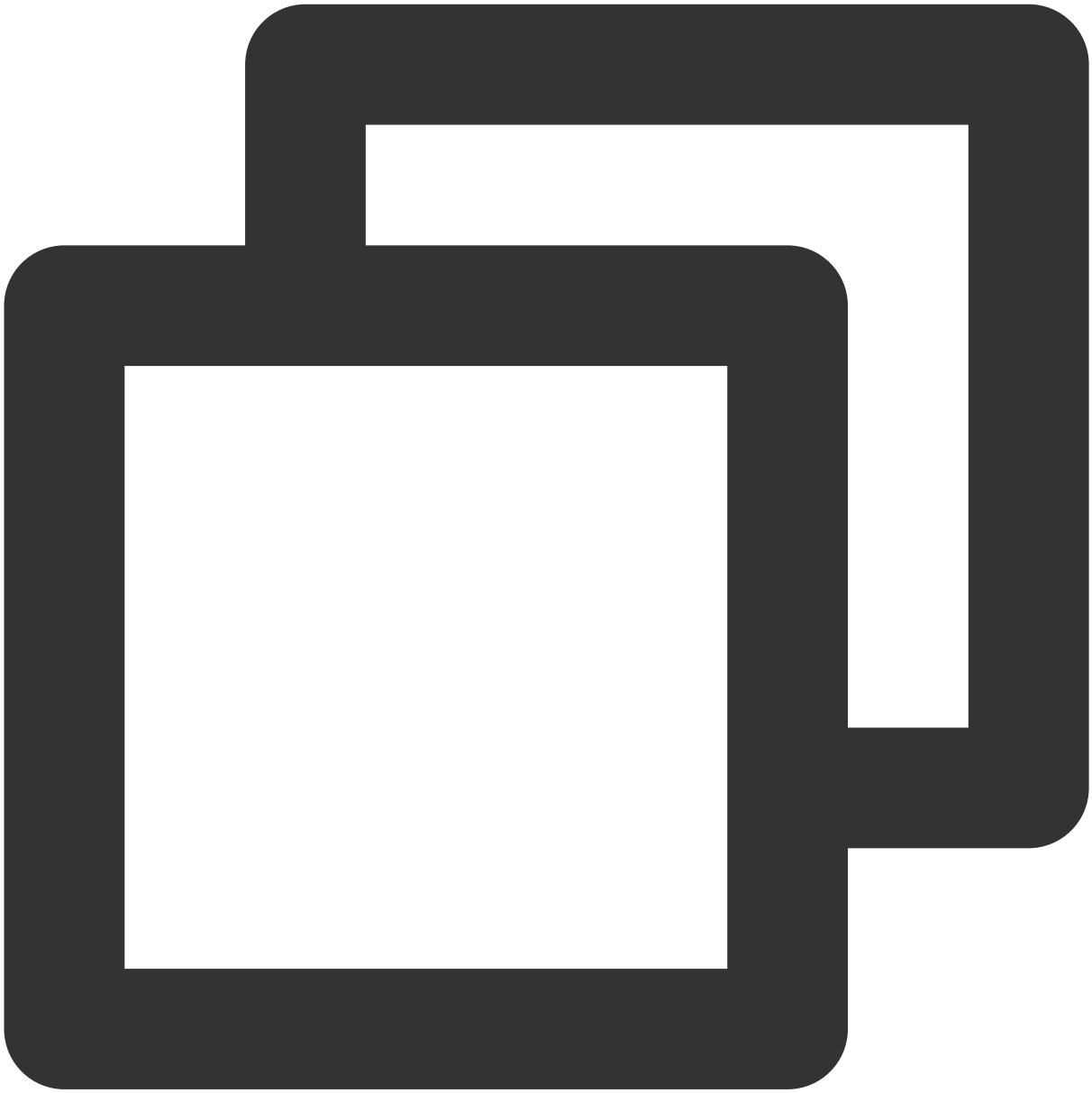
请求



```
POST /image/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Input>
    <Object>a.jpg</Object>
  </Input>
  <Input>
```

```
    <Url>http://exsample.com/b.jpg</Url>
  </Input>
  <Conf>
    <BizType>b81d45f94xxxxxxxxa9506f45a11</BizType>
  </Conf>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<Response>
    <JobsDetail>
        <JobId>xxxxxxxxxx</JobId>
        <CompressionResult>0</CompressionResult>
        <Object>a.jpg</Object>
        <Label>Porn</Label>
        <Result>1</Result>
        <Score>100</Score>
        <SubLabel>xxx</SubLabel>
        <PornInfo>
            <Code>0</Code>
            <Msg>OK</Msg>
            <HitFlag>1</HitFlag>
            <Score>100</Score>
            <Label>xxx</Label>
            <SubLabel>xxx</SubLabel>
        </PornInfo>
    </JobsDetail>
    <JobsDetail>
        <JobId>yyyyyyyyyy</JobId>
        <CompressionResult>0</CompressionResult>
        <Url>http://exsample.com/b.jpg</Url>
        <Label>Porn</Label>
        <Result>1</Result>
        <Score>100</Score>
        <SubLabel>xxx</SubLabel>
        <PornInfo>
            <Code>0</Code>
            <Msg>OK</Msg>
            <HitFlag>1</HitFlag>
            <Score>100</Score>
            <Label>xxx</Label>
            <SubLabel>xxx</SubLabel>
        </PornInfo>
    </JobsDetail>
    <RequestId>xxxxxxxxxxx</RequestId>
</Response>
```

查询图片审核任务结果

最近更新时间：2023-12-30 10:30:32

功能描述

本接口用于主动查询指定的图片审核任务结果。

说明：

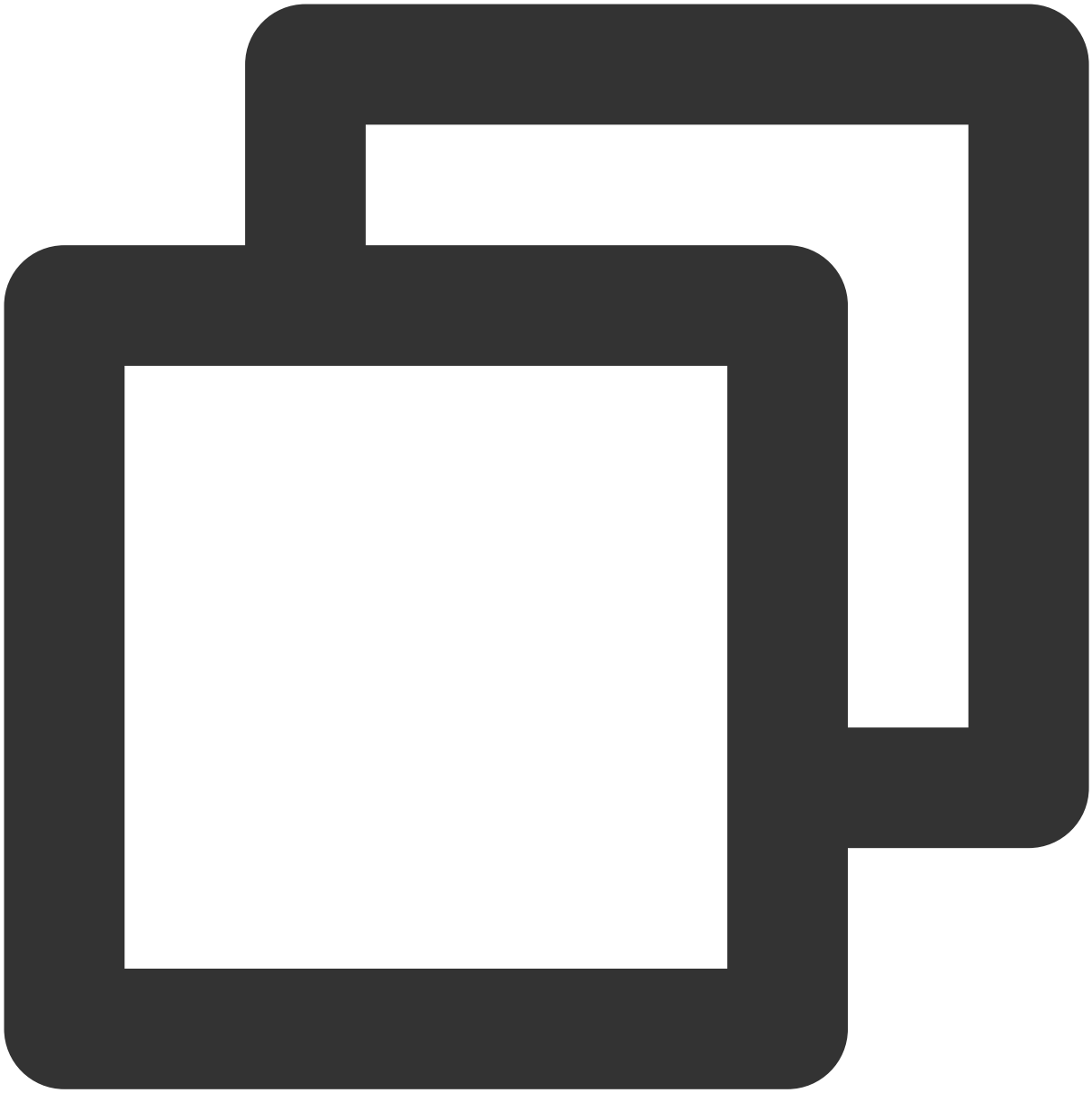
图片审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

数据万象 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
GET /image/auditing/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明：

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

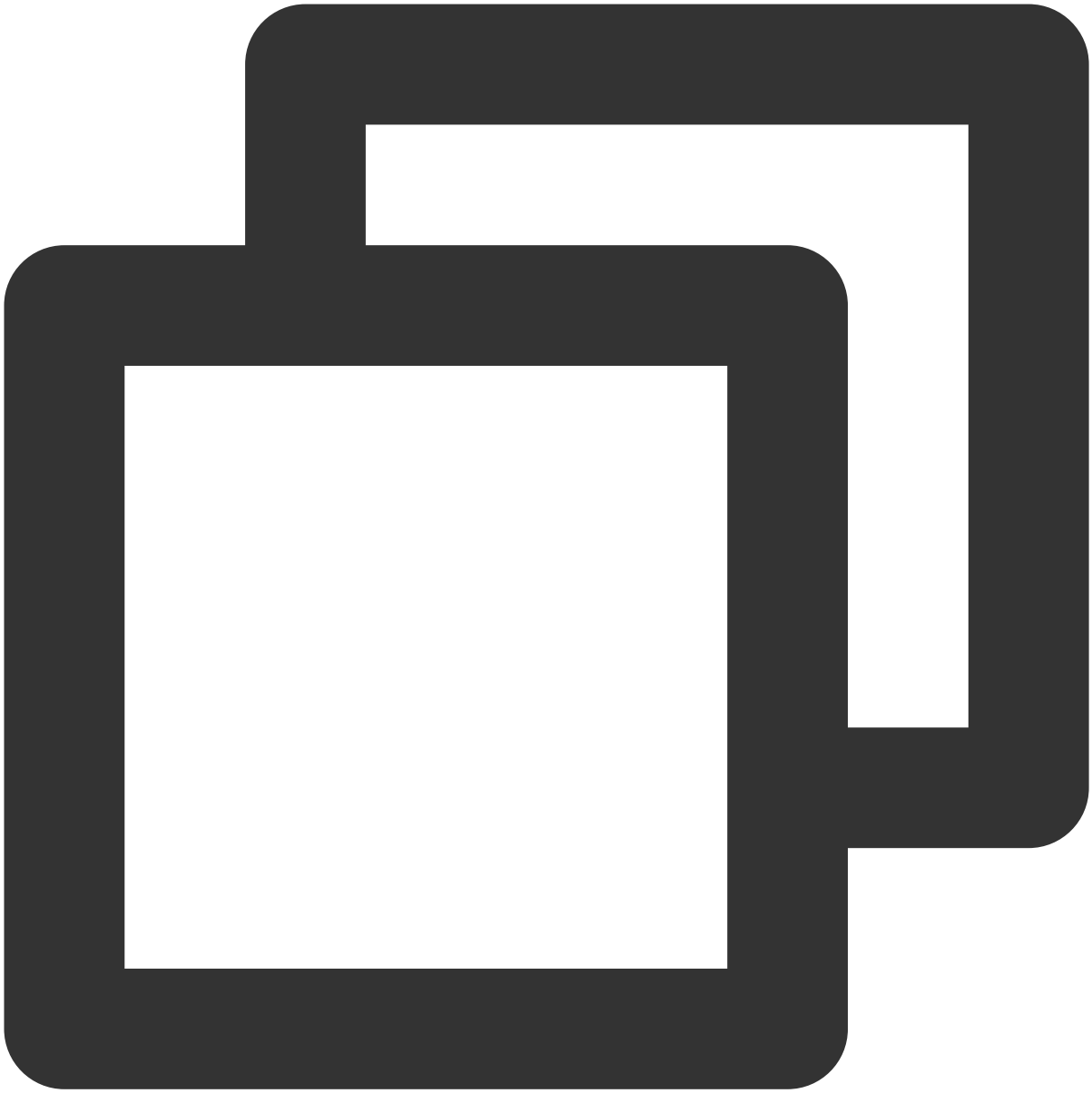
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <Label></Label>
    <Result>1</Result>
    <PornInfo>
        <HitFlag>1</HitFlag>
        <Label>xxx</Label>
```

```
        <SubLabel>SexBehavior</SubLabel>
        <Score>100</Score>
        <OcrResults>
            <Text></Text>
            <Keywords></Keywords>
            <Location>
                <X></X>
                <Y></Y>
                <Width></Width>
                <Height></Height>
                <Rotate></Rotate>
            </Location>
        </OcrResults>
    </PornInfo>
    <AdsInfo>
        <HitFlag>1</HitFlag>
        <Label>xxx</Label>
        <SubLabel>SexBehavior</SubLabel>
        <Score>100</Score>
        <OcrResults>
            <Text></Text>
            <Keywords></Keywords>
            <Location>
                <X></X>
                <Y></Y>
                <Width></Width>
                <Height></Height>
                <Rotate></Rotate>
            </Location>
        </OcrResults>
    </AdsInfo>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

参数名称	类型	描述
Response	Container	审核任务结果。

Container 节点 Response 的内容：

参数名称	类型	描述
JobsDetail	Container	任务的详细信息。

RequestId	String	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。

Container 节点 JobsDetail 的内容：

参数名称	类型	描述
Code	String	错误码，只有 State 为 Failed 时返回。详情请参见 错误码列表。
Message	String	错误描述，只有 State 为 Failed 时返回。
DataId	String	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。
JobId	String	本次审核任务的 ID
State	String	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。
CreationTime	String	审核任务的创建时间。
Object	String	本次审核的文件名称，创建任务使用 Object 时返回。
Url	String	本次审核的文件链接，创建任务使用 Url 时返回。
CompressionResult	Integer	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。
Text	String	该图里的文字内容（OCR），当审核策略开启文本内容检测时返回。
Label	String	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告等。
Result	Integer	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。
Category	String	该图命中的审核类别结果。
SubLabel	String	该图命中的二级标签结果。
PornInfo	Container	审核场景为**涉黄**的审核结果信息。
AdsInfo	Container	审核场景为**广告引导**的审核结果信息。
UserInfo	Container	用户业务字段。创建任务时未设置 UserInfo 则无此字段。
ListInfo	Container	账号黑白名单结果。

ForbidState	Integer	若您在请求时设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。

JobsDetail中Container节点PornInfo，AdsInfo的内容：

参数名称	类型	描述
HitFlag	Integer	用于返回该审核场景的审核结果，返回值：0：正常1：确认为当前场景的违规内容2：疑似为当前场景的违规内容。
Score	Integer	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。其中0 - 60分表示图片正常，61 - 90分表示图片疑似敏感，91 - 100分表示图片确定敏感。例如：色情 99，表明该内容非常有可能属于色情内容。
Label	String	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。
Category	String	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。
SubLabel	String	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。
OcrResults	Container Array	该字段表示 OCR 文本识别的详细检测结果，包括文本识别结果、命中的关键词等信息，有相关违规内容时返回。
LibResults	Container Array	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。

Container 节点 LibResults 的内容：

参数名称	类型	描述
ImageId	String	该字段表示命中的风险库中的图片样本 ID。
Score	Integer	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。

Container 节点 OcrResults 的内容：

参数名称	类型	描述
Text	String	图片 OCR 文本识别出的具体文本内容。
Keywords	String Array	在当前审核场景下命中的关键词。

版权所有：腾讯云计算（北京）有限责任公司

Location	Container	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。

Container 节点 Location 的内容：

名称	类型	描述
X	Float	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Y	Float	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Width	Float	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Height	Float	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Rotate	Float	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360**（**角度制**），方向为**逆时针旋转**。

Container 节点 ObjectResults 的内容：

名称	类型	描述
Name	String	该标签用于返回所识别出的实体名称，例如人名。
Location	Container	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

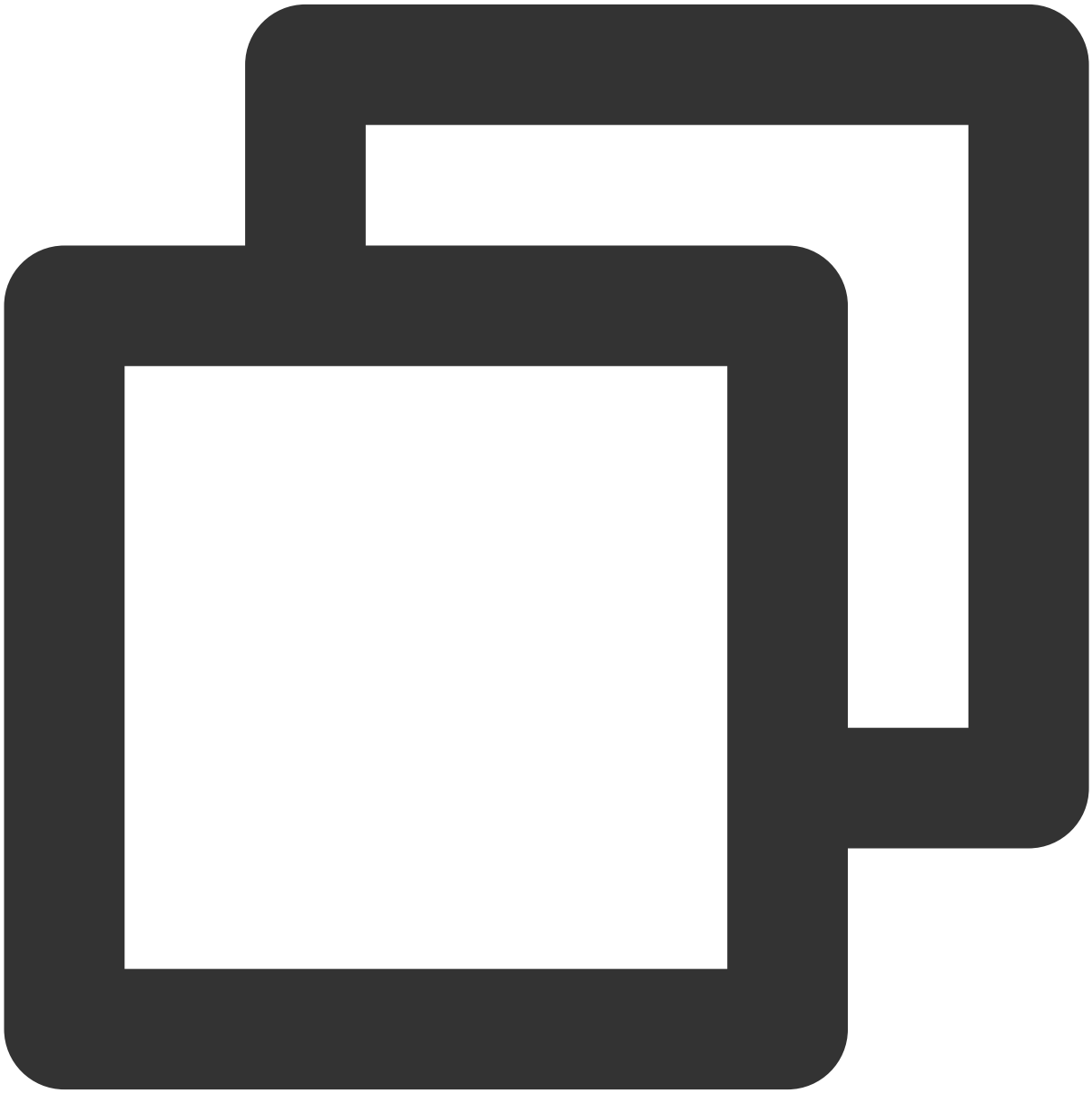
节点名称（关键字）	类型	描述
ListResults	Container Array	命中的所有名单结果。

Container 节点 ListResults 的内容：

节点名称（关键字）	类型	描述
ListType	Integer	命中的名单类型，取值为0（白名单）和1（黑名单）。
ListName	String	命中的名单名称。
Entity	String	命中了名单中的哪条内容。

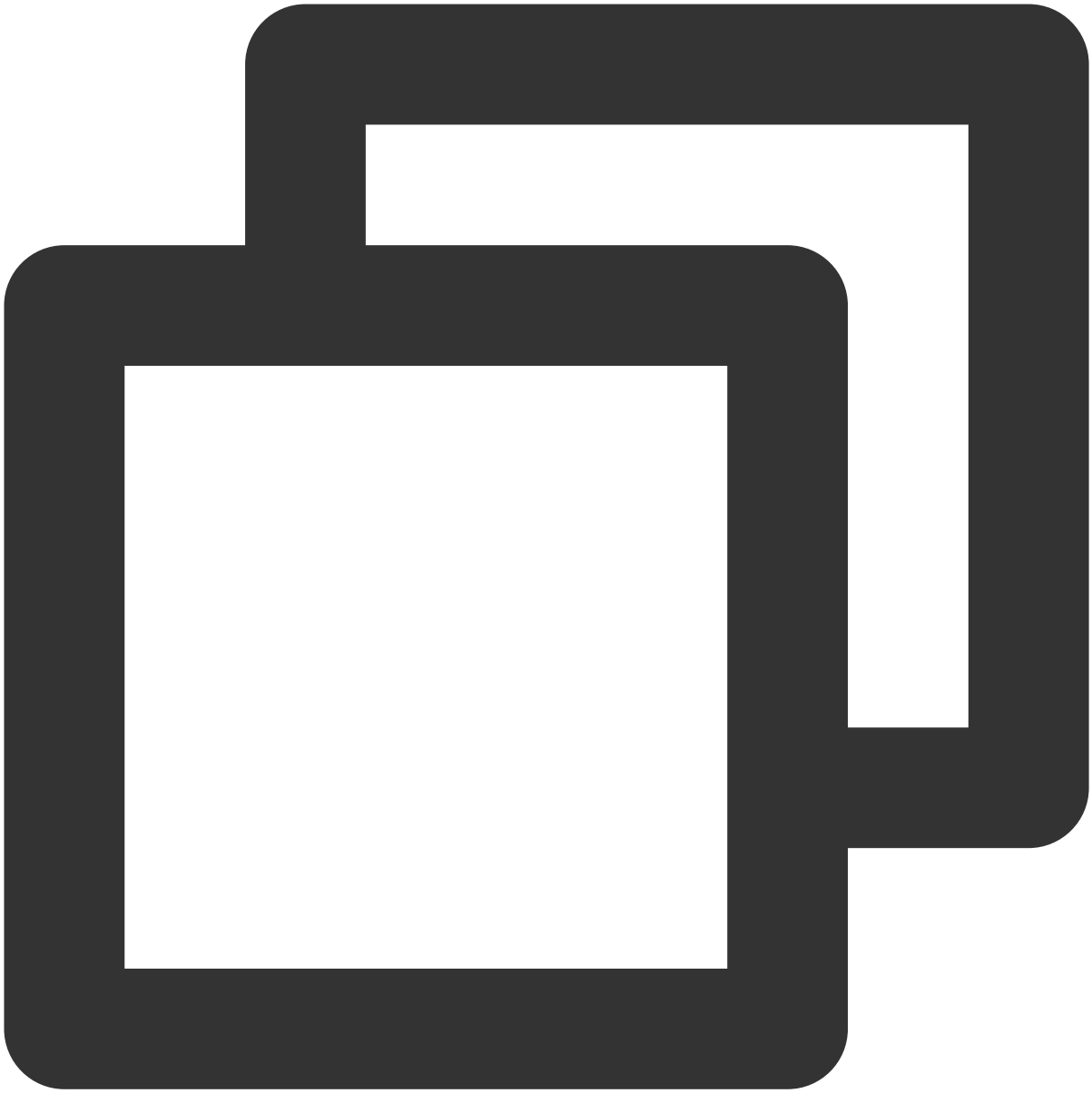
实际案例

请求



```
GET /image/auditing/iab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<Response>
    <JobsDetail>
        <JobId>iab1ca9fc8a3ed11ea834c525400863904</JobId>
```

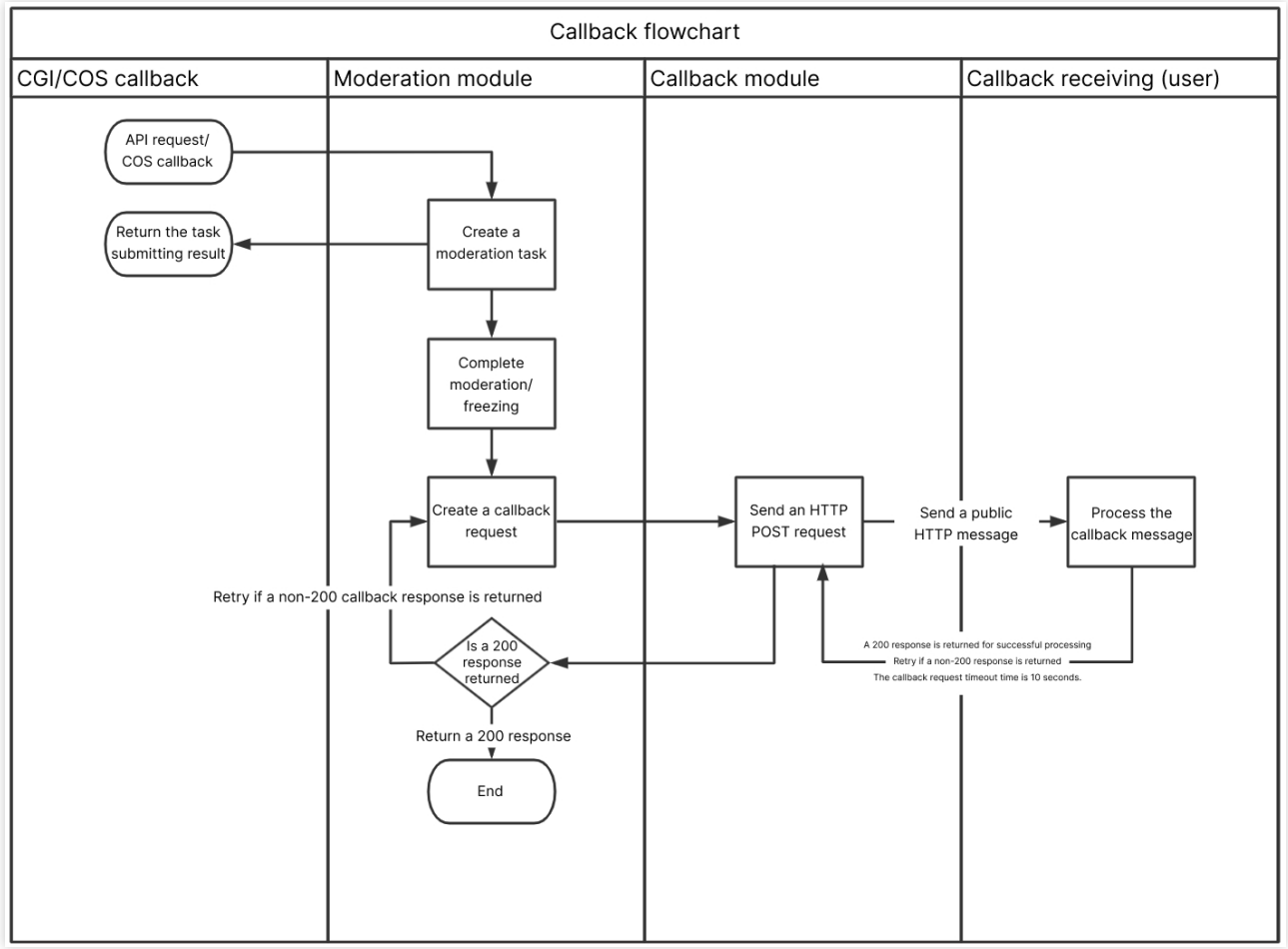
```
        <State>Success</State>
        <CreationTime>2021-05-21T10:44:32+08:00</CreationTime>
        <Object>example.jpg</Object>
        <Label>Ads</Label>
        <Result>1</Result>
        <AdsInfo>
            <HitFlag>1</HitFlag>
            <Label>xxx</Label>
            <SubLabel>SexBehavior</SubLabel>
            <Score>100</Score>
            <OcrResults>
                <Text></Text>
                <Keywords></Keywords>
                <Location>
                    <X></X>
                    <Y></Y>
                    <Width></Width>
                    <Height></Height>
                    <Rotate></Rotate>
                </Location>
            </OcrResults>
        </AdsInfo>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

图片审核回调内容

最近更新时间：2023-12-30 10:30:46

功能描述

若您配置了审核回调地址，在图片审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。

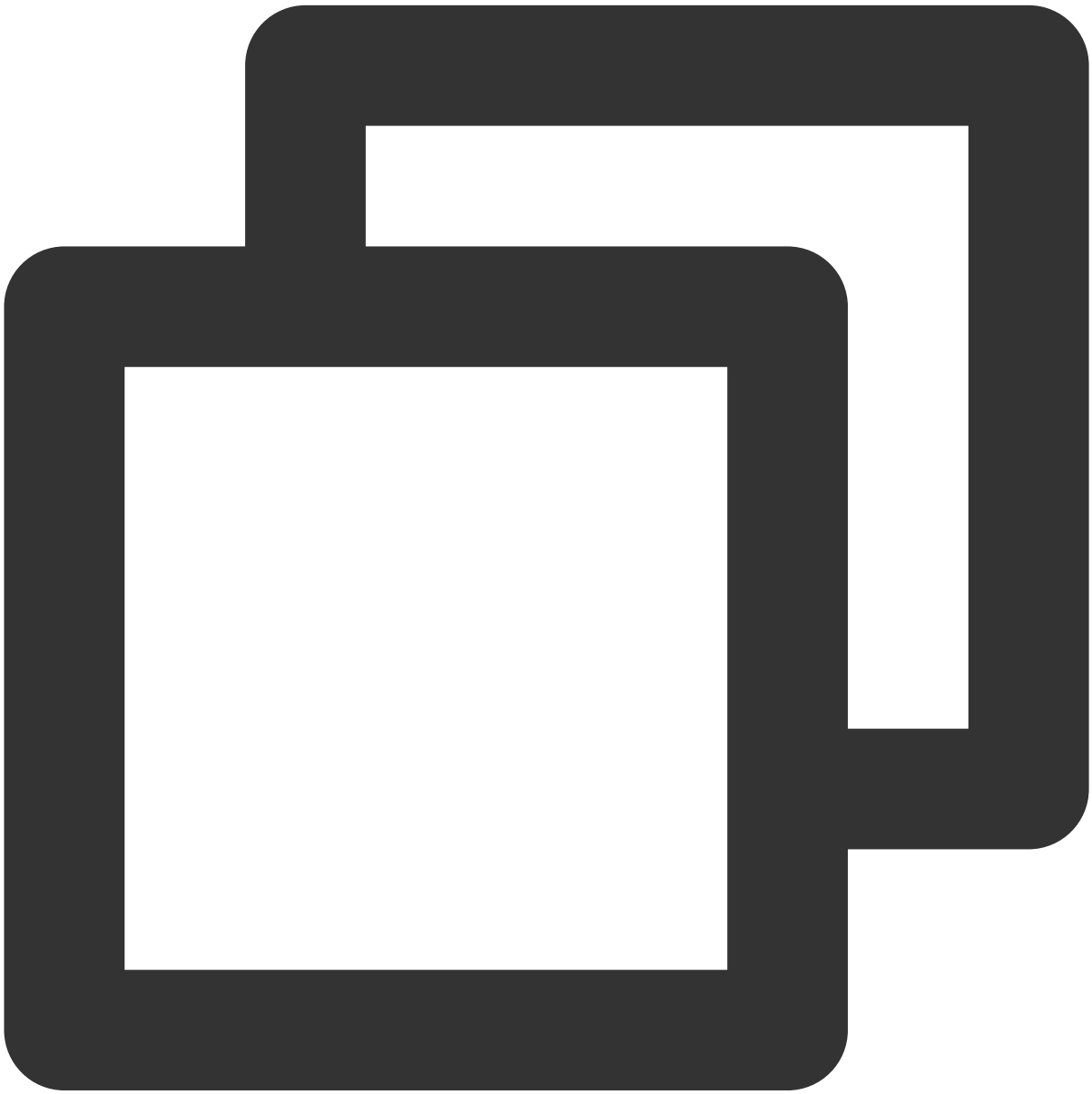


回调内容说明

简洁回调（Simple）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Simple` 的头部。

该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：




```
{
    "code": 0,
    "data": {
        "forbidden_status": 0,
        "event": "ReviewImage",
        "porn_info": {
            "hit_flag": 0,
            "label": "",
            "score": 9
```

```
        },
        "result": 0,
        "trace_id": "test_trace_id",
        "url": "https://examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test.jp
        "cos_headers": {
            "x-cos-meta-xx": "xx"
        }
    },
    "message": "Test request when setting callback url"
}
```

具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码。	Integer	是
message	错误描述。	String	是
data	图片审核结果的详细信息。	Object	是

data 的具体内容如下：

参数名称	描述	类型	是否必选
trace_id	唯一 ID 标识。	String	是
url	被审核图片的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewImage。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer	是
cos_headers	资源上传时设置的自定义 header 内容，若未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否

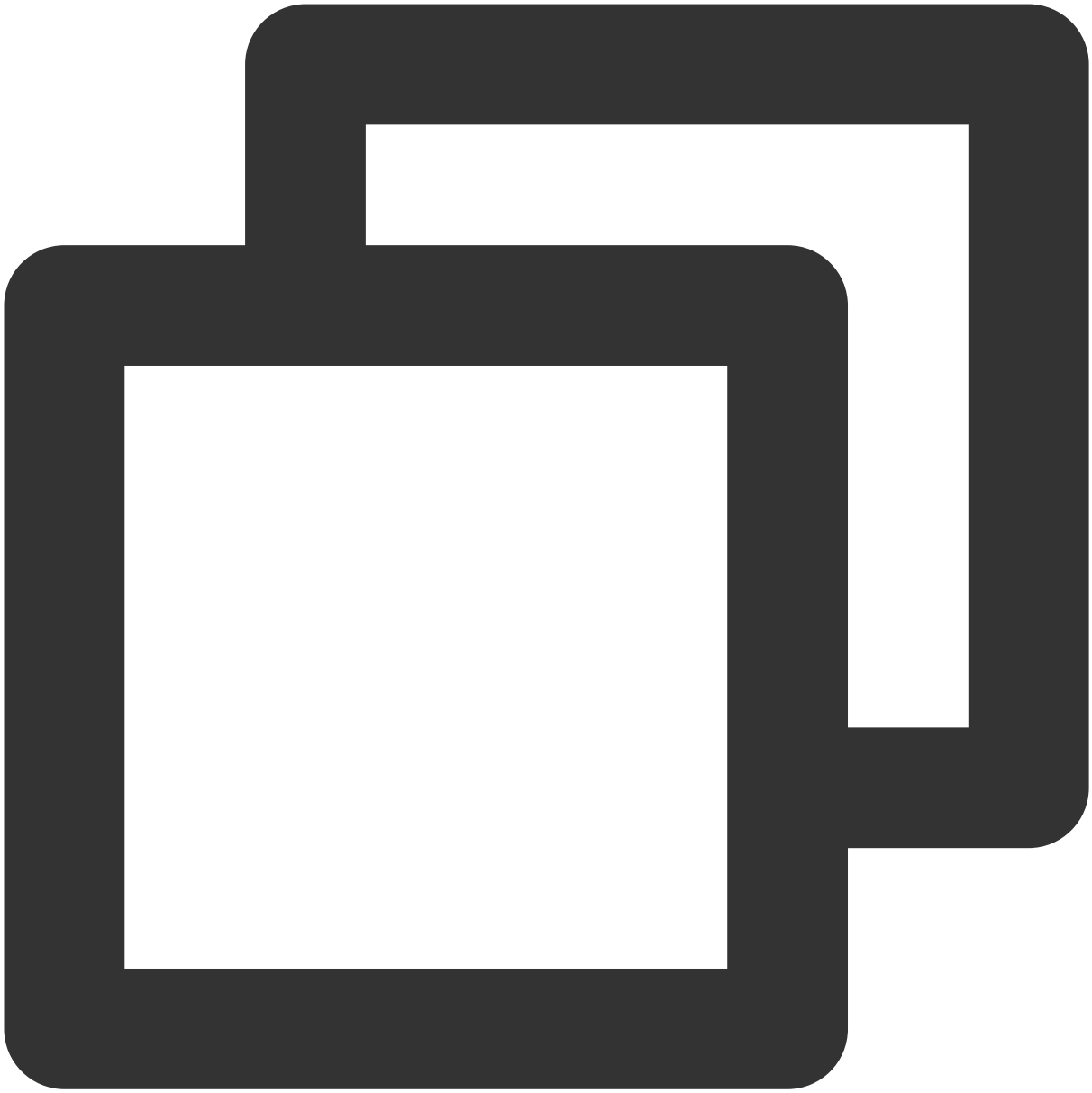
版权所有：腾讯云计算（北京）有限责任公司

xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer	是
score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer	是
label	命中的标签名称。	String	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：




```
{
  "EventName": "ReviewImage",
  "JobsDetail": {
    "JobId": "xxxx",
    "State": "Success",
    "CreationTime": "",
    "Object": "",
    "Url": "https://examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test.jpg",
    "Label": "Normal",
    "Result": 0,
    "Score": 0,
```

```
      "Category": "",
      "SubLabel": "",
      "Text": "",
      "PornInfo": {
        "HitFlag": 0,
        "Score": 0,
        "Label": "",
        "Category": "",
        "SubLabel": ""
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0,
        "Label": "",
        "Category": "",
        "SubLabel": ""
      },
      "BucketId": "examplebucket-1250000000",
      "Region": "ap-chongqing",
      "ForbidState": 0,
      "CosHeaders": {
          "x-cos-meta-id": "xxxx"
      }
    }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息	Object
EventName	任务的类型，为 `ReviewImage`	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
State	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String

CreationTime	审核任务的创建时间。	String
CompressionResult	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。	Integer
Object	被审核的图片文件名称。创建任务使用 Object 时返回。	String
Url	被审核的图片文件链接。创建任务使用 Url 时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Category	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。</br>注意：该字段可能返回空，表示未命中具体的子标签	String
Text	图片 OCR 文本识别出的具体文本内容，当审核策略开启 OCR 内容检测时返回。	String
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
BucketId	审核操作所属的存储桶名称。	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object

版权所有：腾讯云计算（北京）有限责任公司

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer
Label	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本识别结果、命中的关键词等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String

Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

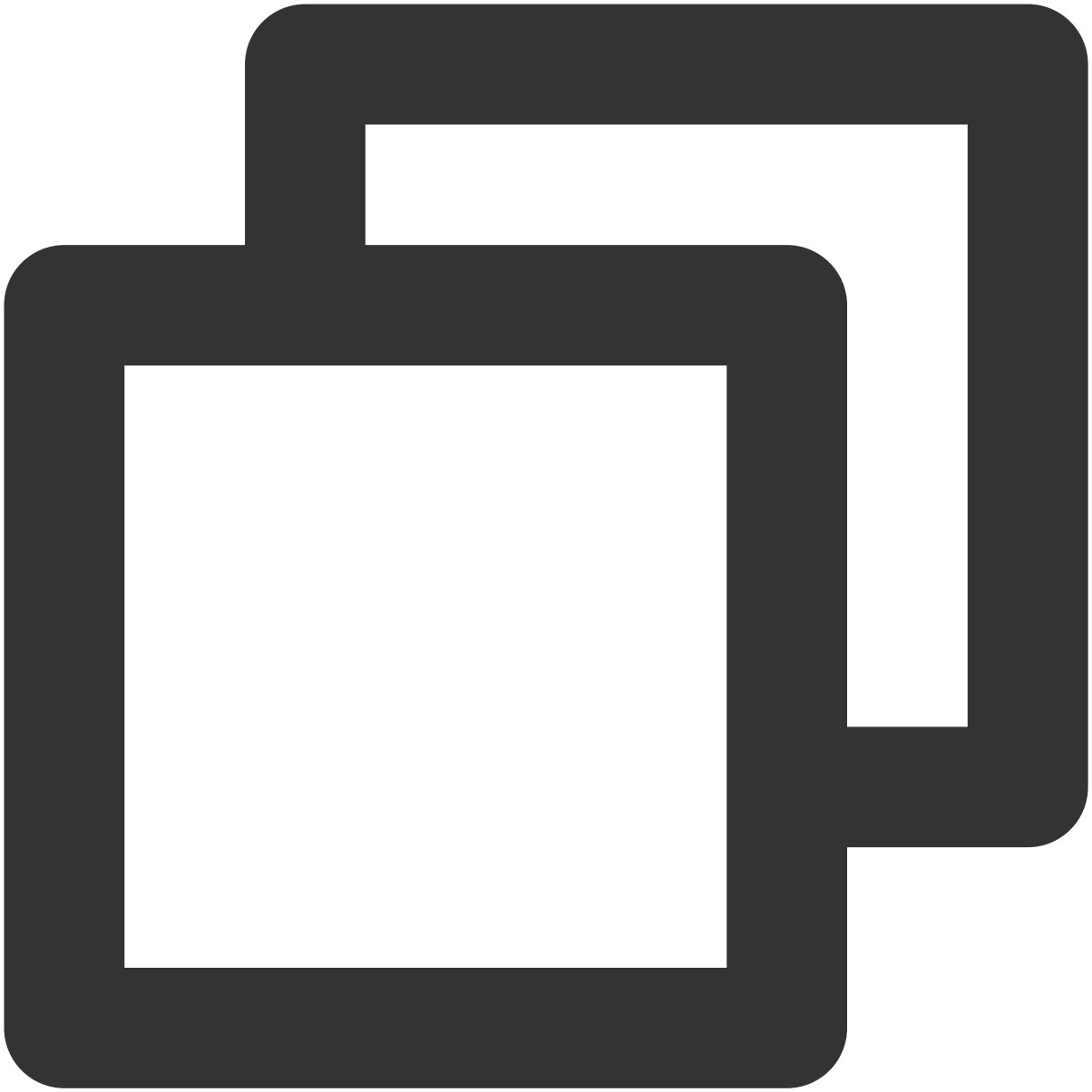
节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，例如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0 - 360**（**角度制**），方向为**逆时针旋转**。	Float

实际案例

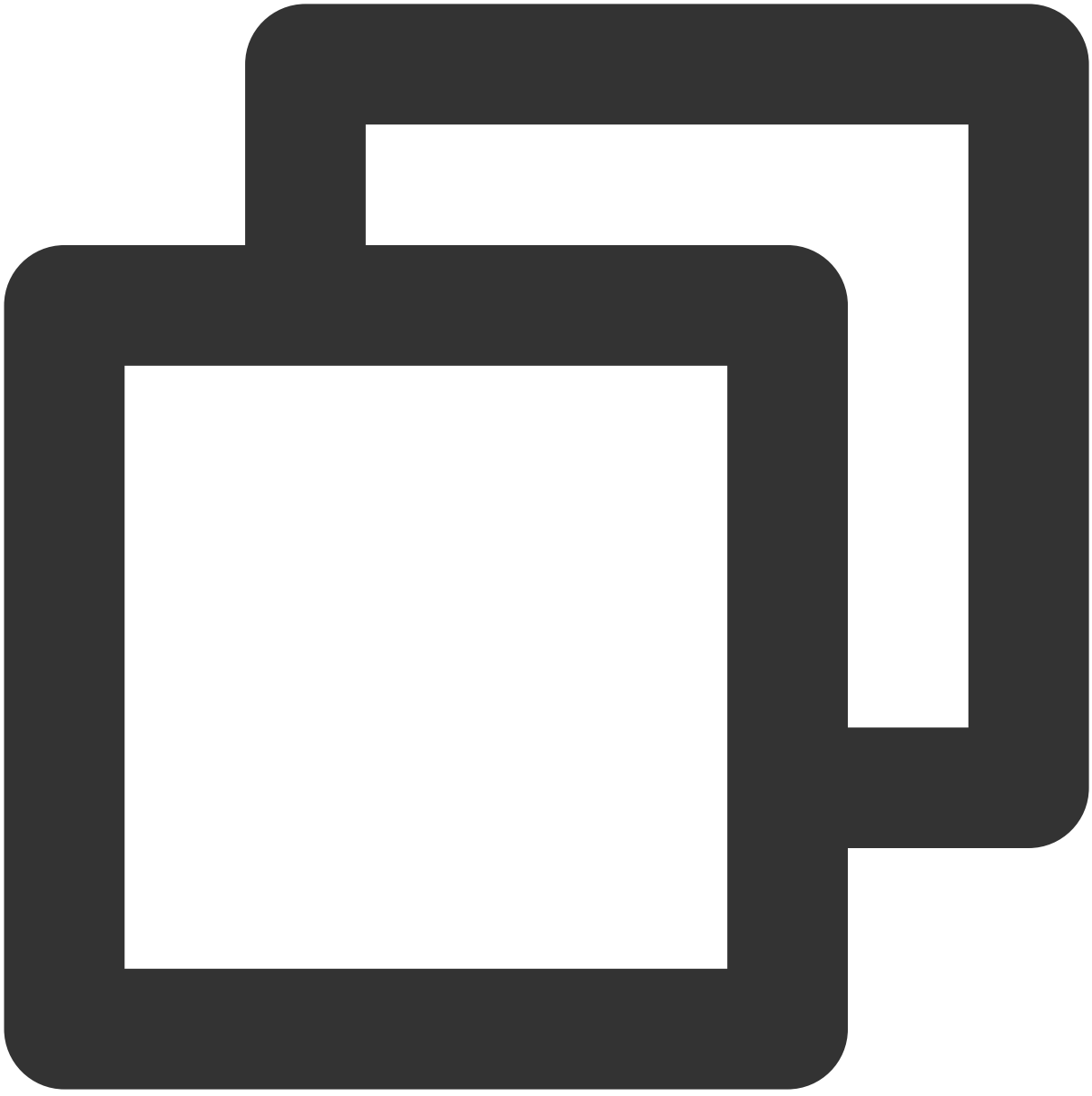
案例一：简洁回调（Simple）




```
{
   "code": 0,
   "data": {
       "event":"ReviewImage",
       "forbidden_status": 0,
       "porn_info": {
           "hit_flag": 0,
           "label": "",
           "score": 9
       },
       "result": 0,
```

```
        "trace_id": "ixzt90jl2dfscxxxxxxxxxxxxxxxx",
        "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/test.j
        "cos_headers": {
            "x-cos-meta-id": "666666"
        }
    },
    "message": "success"
}
```

案例二：详细回调（Detail）



```
{
  "EventName": "ReviewImage",
  "JobsDetail": {
    "JobId": "xxxx",
    "State": "Success",
    "CreationTime": "2021-08-10T21:01:10+08:00",
    "CompressionResult": 0,
    "Object": "1.jpg",
    "Label": "Normal",
    "Result": 0,
    "Score": 0,
    "Category": "",
    "SubLabel": "",
    "Text": "",
    "PornInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": "",
      "Category": "",
      "SubLabel": ""
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": "",
      "Category": "",
      "SubLabel": ""
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
        "x-cos-meta-id": "xxxx"
    }
  }
}
```


版权所有：腾讯云计算（北京）有限责任公司

图片审核结果反馈

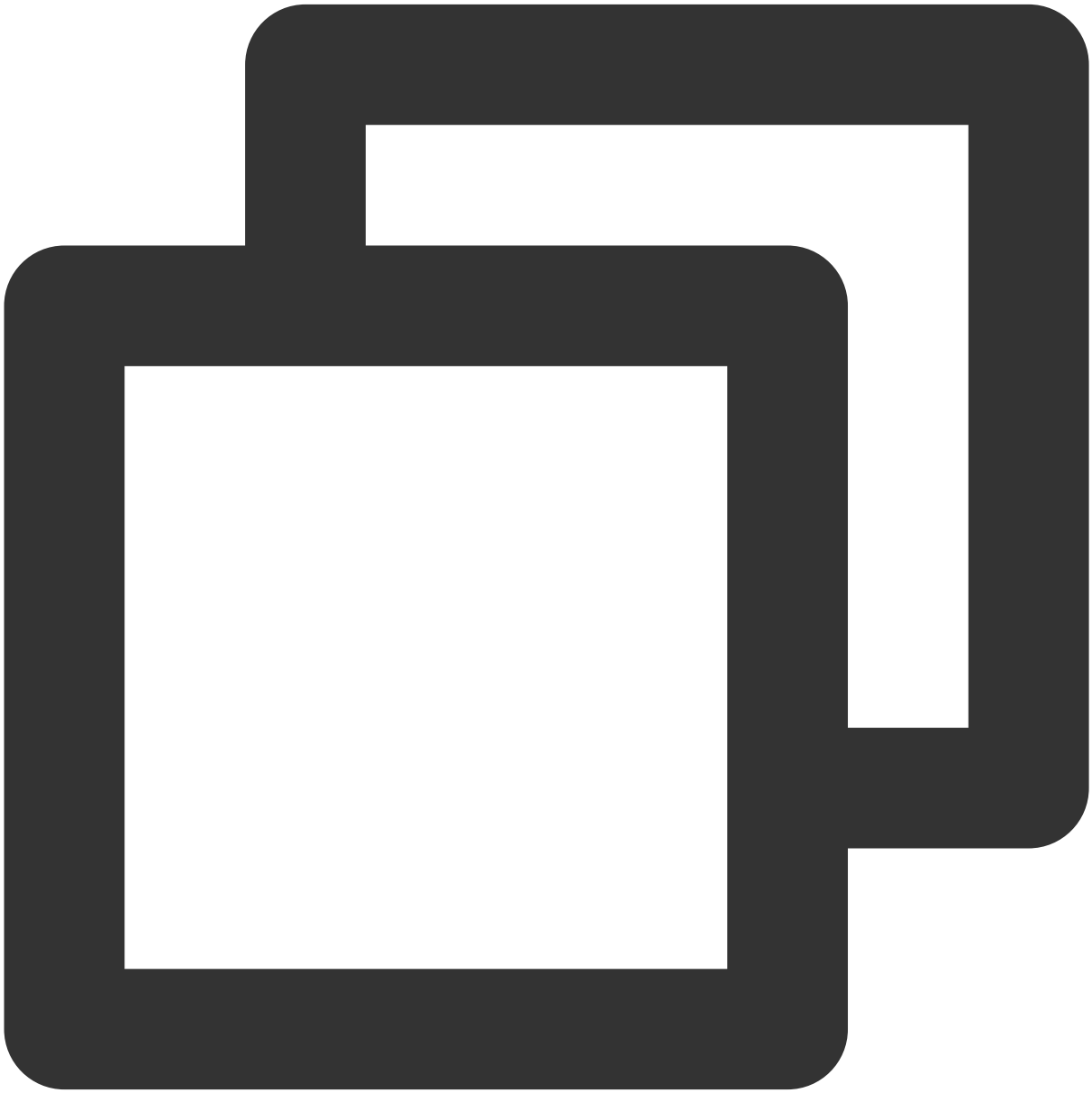
最近更新时间：2023-12-30 10:31:08

功能描述

您可通过本接口反馈与预期不符的审核结果，例如色情图片被审核判定为正常或正常图片被判定为色情时可通过该接口直接反馈。
本接口不会直接修改审核结果，您反馈的错误审核结果将在后台进行确认，并在后续的审核任务中生效。

请求

请求示例



```
POST /report/badcase HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明：

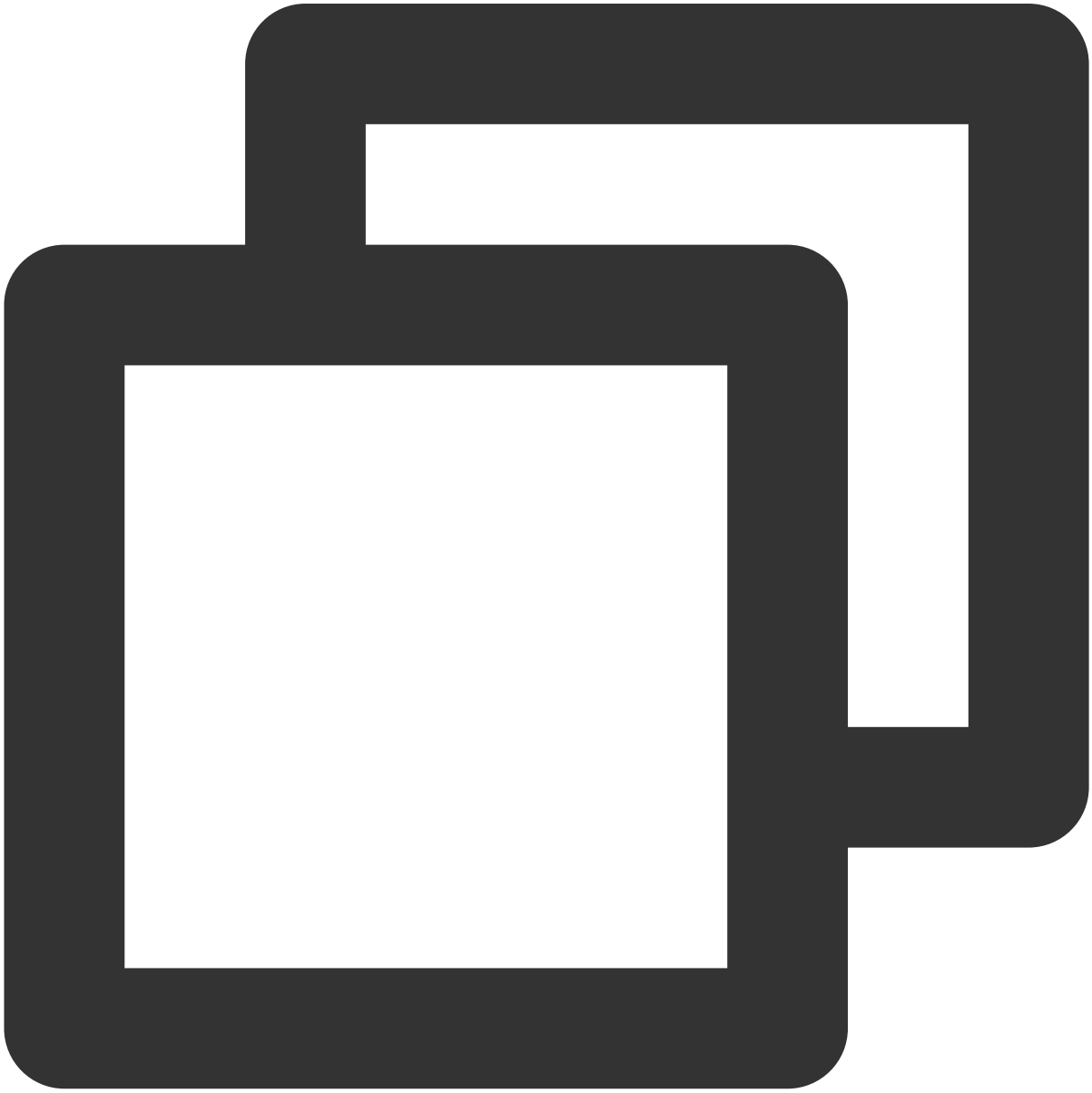
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
    <ContentType></ContentType>
```

```
    <JobId></JobId>
    <ModerationTime></ModerationTime>
    <Url></Url>
    <SuggestedLabel></SuggestedLabel>
    <Label></Label>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	反馈审核错误的具体参数	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
ContentType	Request	需要反馈的数据类型，反馈图片错误样本，取值为2。	Integer	是
Url	Request	图片类型的样本，需要填写图片的 url 链接，ContentType 为2时必填。	String	否
Label	Request	数据万象审核判定的审核结果标签，例如 Porn。	String	是
SuggestedLabel	Request	您自己期望的正确审核结果的标签，例如期望是正常，则填 Normal。	String	是
JobId	Request	该数据样本对应的审核任务 ID，有助于定位审核记录。	String	否
ModerationTime	Request	该数据样本之前审核的时间，有助于定位审核记录。 格式为 2021-08-07T12:12:12+08:00	String	否

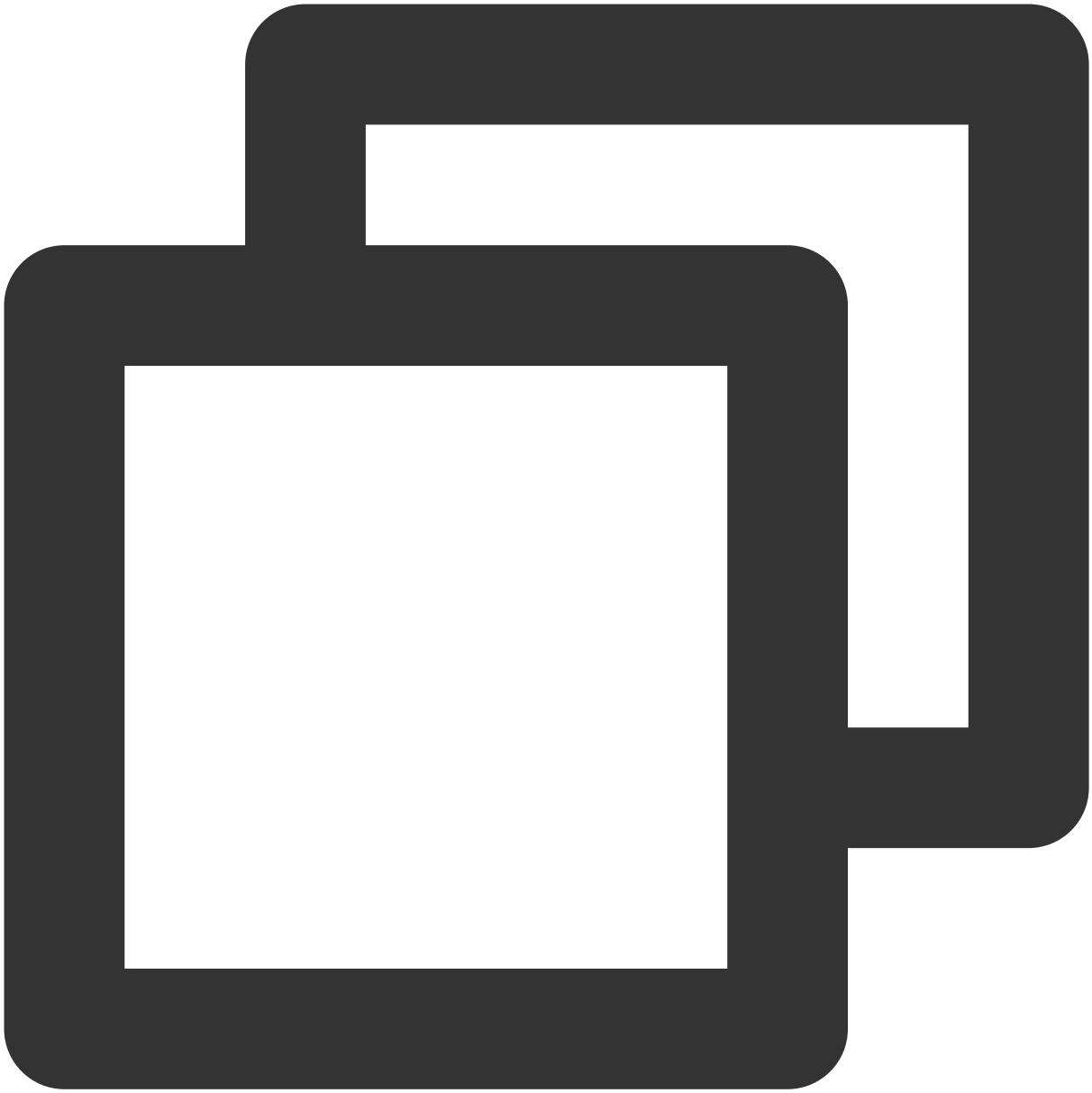
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	返回的具体响应内容	Container

Container 节点 Response 的内容：

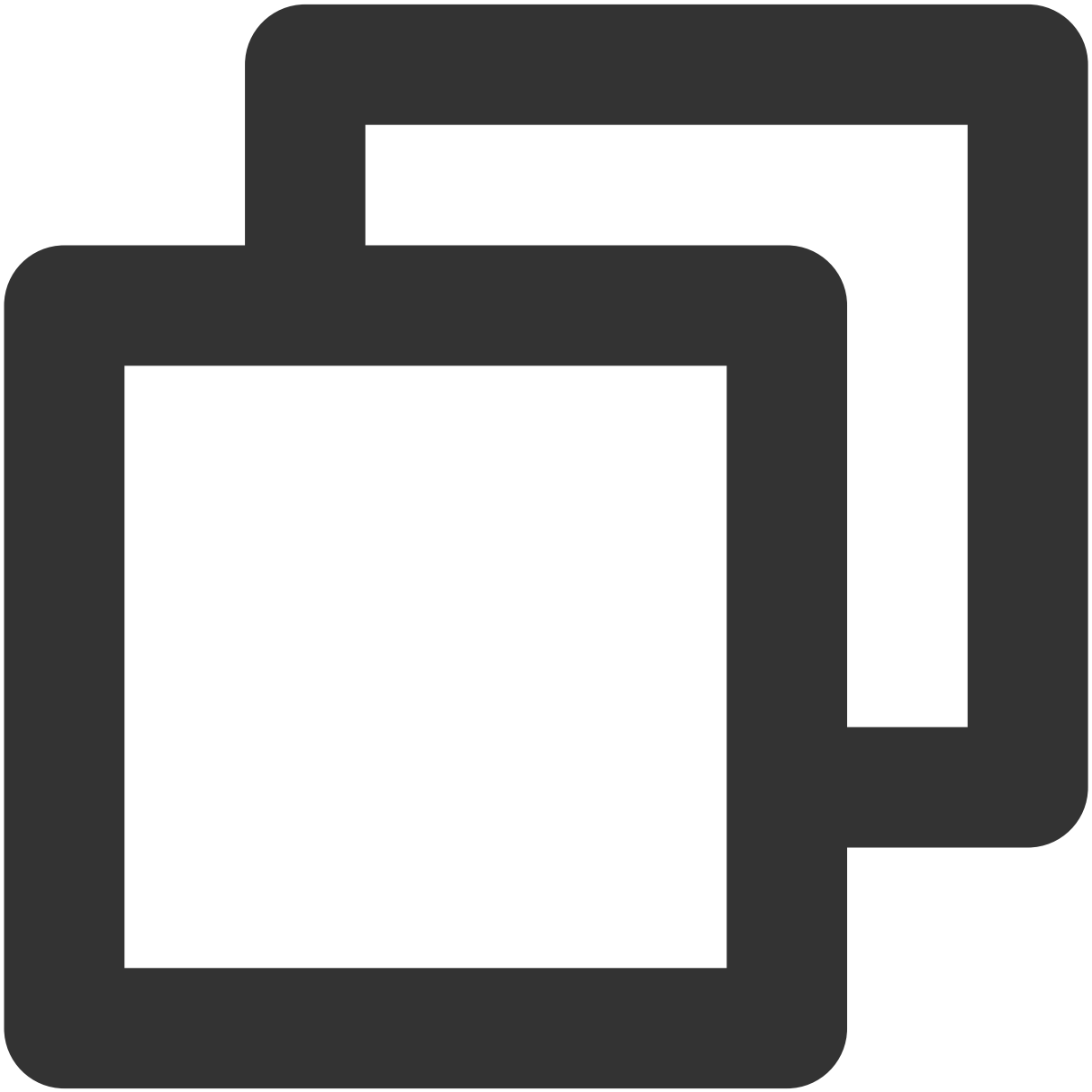
节点名称（关键字）	父节点	描述	类型
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

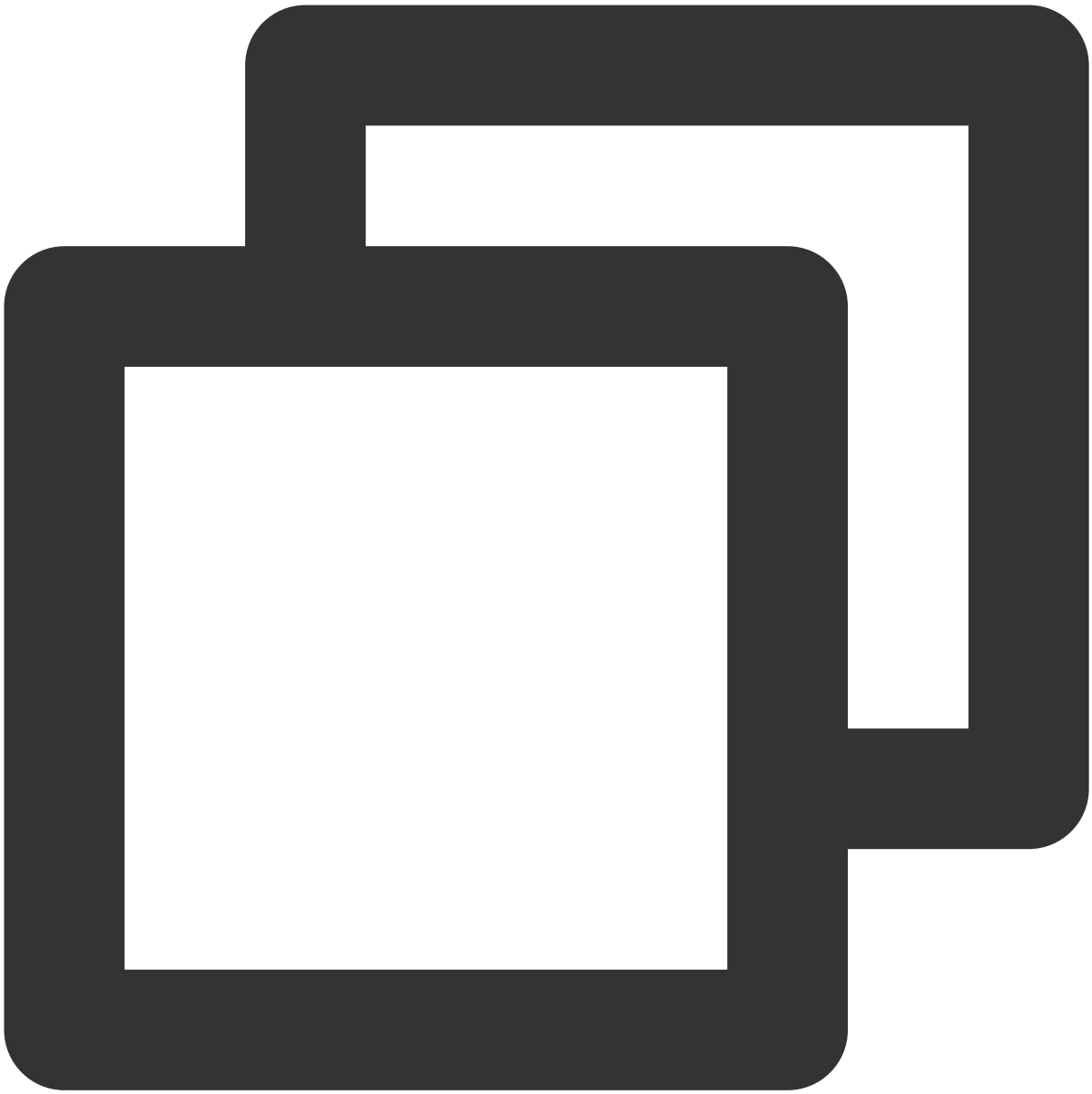
请求



```
POST /report/badcase HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1********&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 66
Content-Type: application/xml
<Request>
    <ContentType>1</ContentType>
    <Url>http://www.example.com/abc.jpg</Url>
    <Label>Porn</Label>
    <SuggestedLabel>Normal</SuggestedLabel>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 May 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
```

```
<Response>
    <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

视频审核
提交视频审核任务

最近更新时间：2023-12-30 09:33:47

功能描述

本接口用于提交一个视频审核任务。视频审核功能为异步任务方式，您可以通过提交视频审核任务审核您的视频文件，然后通过 查询视频审核任务接口 或 视频审核回调 查询审核结果。
该接口支持情况如下：
说明：
支持审核腾讯云 COS 上的视频文件。
支持审核第三方云存储厂商的视频 URL。
支持审核加密数据，如有需要，请 联系服务人员 咨询。
支持对视频文件进行自动检测，从 OCR 文本识别、物体检测（实体、广告台标、二维码等）、图像识别及音频审核四个维度，通过深度学习技术，识别视频中的违规内容。
支持设置回调地址 Callback 获取检测结果，或通过 查询视频审核任务结果接口 主动轮询获取审核结果详情。
支持识别多种违规场景，包括：色情、违法、广告等场景。
支持根据不同的业务场景 配置自定义的审核策略。
支持用户 自定义配置黑白图片库，打击自定义违规内容。

费用说明

视频的审核分为**视频画面审核**、**视频截帧**、**视频声音审核**，其中：
视频画面审核：基于视频截帧能力，通过视频截帧将视频截取为多张图片进行审核，审核费用与图片审核一致。
视频截帧：会产生相应的 视频截帧费用。
视频声音审核：将视频声音分离出来进行音频审核，审核费用与音频审核一致。
每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1个视频**，**计2次**审核费用。
调用接口会产生图片审核费用、音频审核费用和 COS 读请求费用。
如果音频文件属于 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用。
不支持审核 COS 上的归档存储类型和深度归档存储类型的音频，如果需要审核此类型音频，请先 恢复归档文件。

限制说明

视频文件大小支持：**文件 < 5GB**。

版权所有：腾讯云计算（北京）有限责任公司

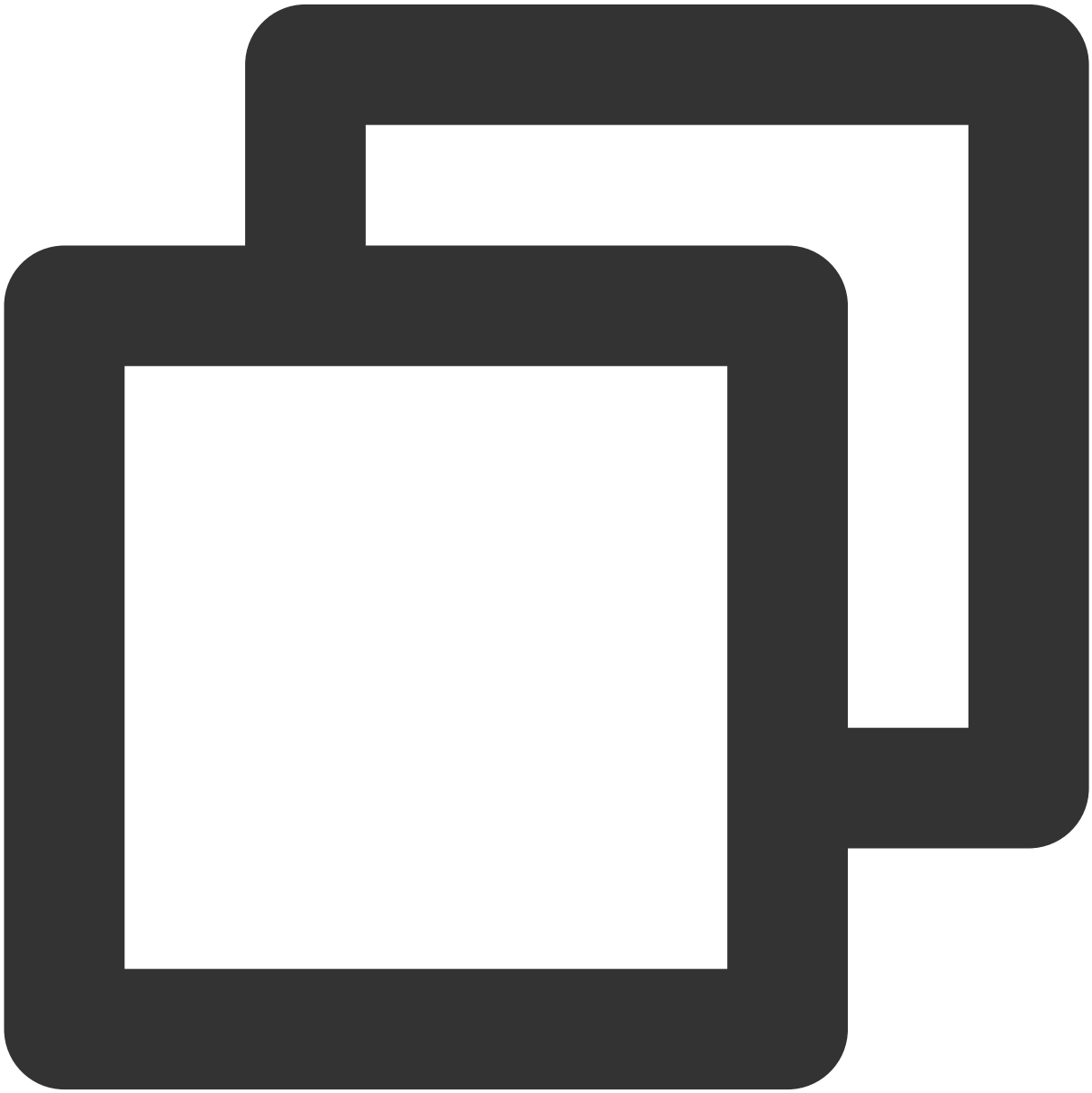
视频文件支持格式：flv、mkv、mp4 、rmvb、avi、wmv、3gp、mov、m3u8、m4v 等。
支持用户配置审核视频画面或视频声音。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
POST /video/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明：

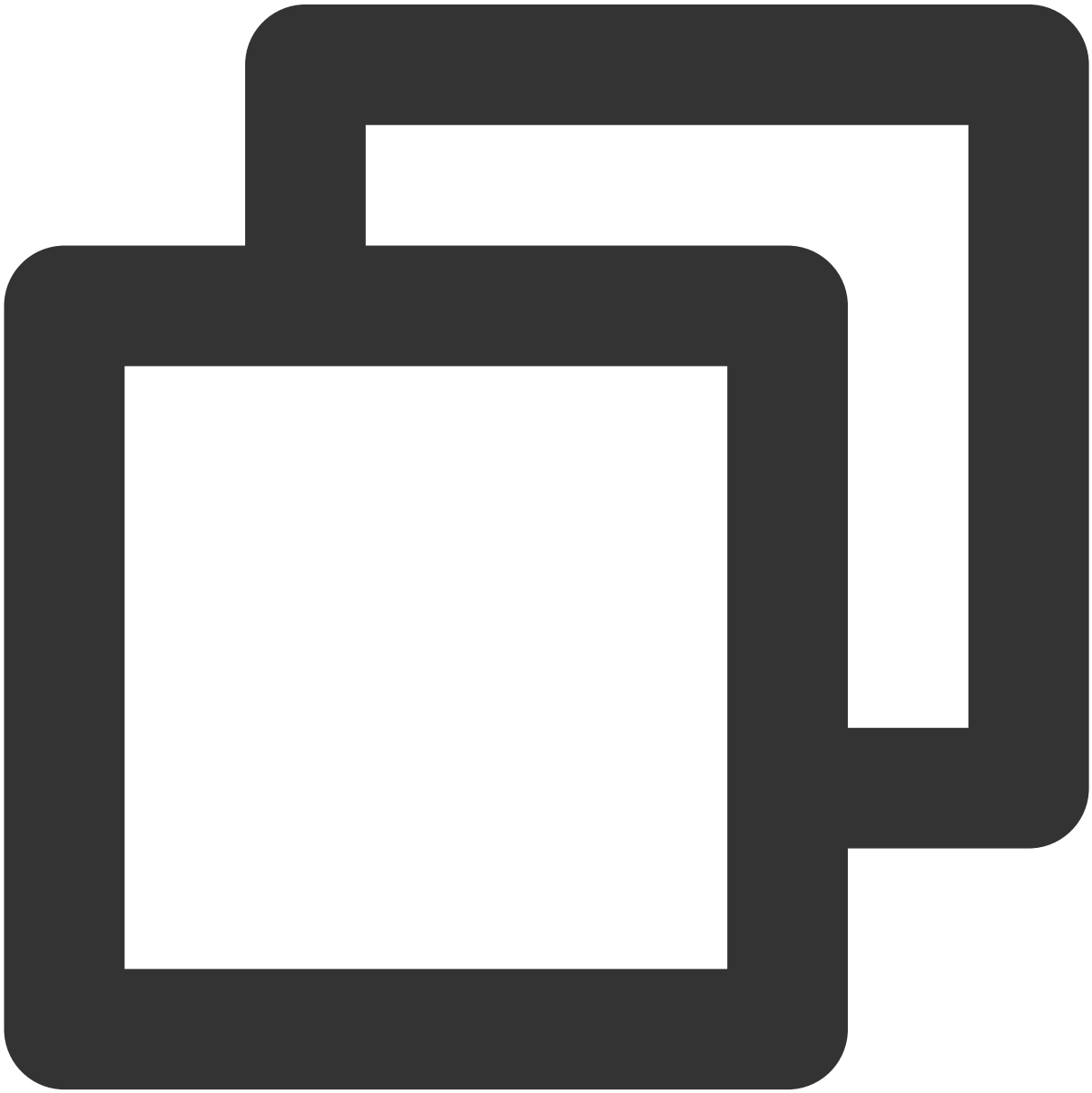
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
```

```
  <Input>
    <Object></Object>
    <DataId></DataId>
  </Input>
  <Conf>
    <Snapshot>
        <Mode>Interval</Mode>
        <TimeInterval></TimeInterval>
        <Count></Count>
    </Snapshot>
    <Callback></Callback>
    <BizType></BizType>
    <DetectContent></DetectContent>
  </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	视频审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的视频。	Container	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	当前 COS 存储桶中的视频文件名称，例如在目录 test 中的文件 video.mp4，则文件名称为 test/video.mp4。	String	否
Url	Request.Input	视频文件的链接地址，例如 `http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/test.mp4` 。Object 和 Url 只能选择其中一种。	String	否

DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	用户业务 TokenId，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	用户业务 Nickname，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	用户业务 DeviceId，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	用户业务 AppId，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	用户业务 Room，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	用户业务 IP，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	用户业务 Type，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	用户业务 ReceiveTokenId，长度不超过128字节。	String	否
Gender	Request.Input.UserInfo	用户业务 Gender，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	用户业务 Level，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	用户业务 Role，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台	String	否

版权所有：腾讯云计算（北京）有限责任公司

		上的审核策略页面，配置您希望审核的场景，如色情、广告、违法违规等，配置指引：设置公共审核策略。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。		
Snapshot	Request.Conf	视频画面的审核通过视频截帧能力截取出一定量的截图，通过对截图逐一审核而实现的，该参数用于指定视频截帧的配置。	Container	是
Callback	Request.Conf	回调地址，以 `http://` 或者 `https://` 开头的地址。	String	否
CallbackVersion	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	String	否
DetectContent	Request.Conf	用于指定是否审核视频声音，当值为0时：表示只审核视频画面截图；值为1时：表示同时审核视频画面截图和视频声音。默认值为0。	Integer	否
CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部截帧和音频片段）、2（回调违规截帧和音频片段）。默认为 1。	Integer	否
Freeze	Request.Conf	自动冻结配置项，可配置指定审核分数的结果进行自动冻结。	Container	否

Container 类型 Snapshot 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Mode	Request.Conf.Snapshot	表示视频的截帧模式，有效值：Interval、Average、Fps。 Interval 表示间隔模式：每间隔 TimeInterval 秒截取一帧；当设置 Count，未设置 TimeInterval 时，表示截取所有帧，共 Count 帧。 Average 表示平均模式：选择该模式时 TimeInterval 参数不生效，需要结合 Count 参数使用，表示整个视频，按平均间隔截取共 Count 张图片。	String	否

		Fps 表示固定帧率模式：选择该模式时，TimeInterval 参数表示每秒截取多少帧，未设置 TimeInterval 时，表示截取所有帧，Count 表示共截取多少帧。		
Count	Request.Conf.Snapshot	视频截帧数量，范围为(0, 10000]。	Integer	是
TimeInterval	Request.Conf.Snapshot	视频截帧频率，范围为(0, 60]，单位为秒，支持 float 格式，执行精度精确到毫秒。	Float	否

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请 联系服务人员咨询。

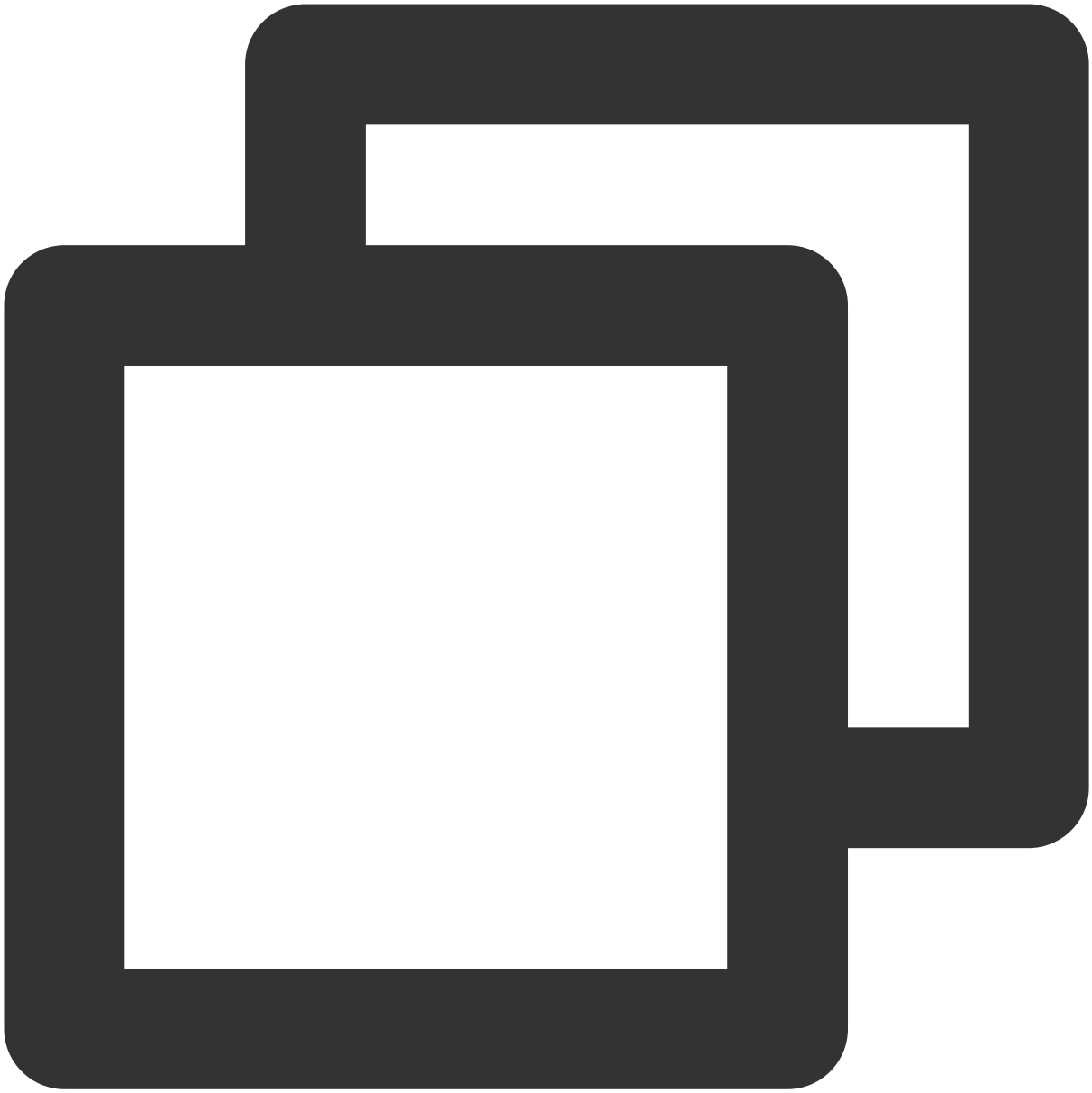
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
      <DataId></DataId>
      <JobId></JobId>
      <State></State>
      <CreationTime></CreationTime>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	视频审核返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	视频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

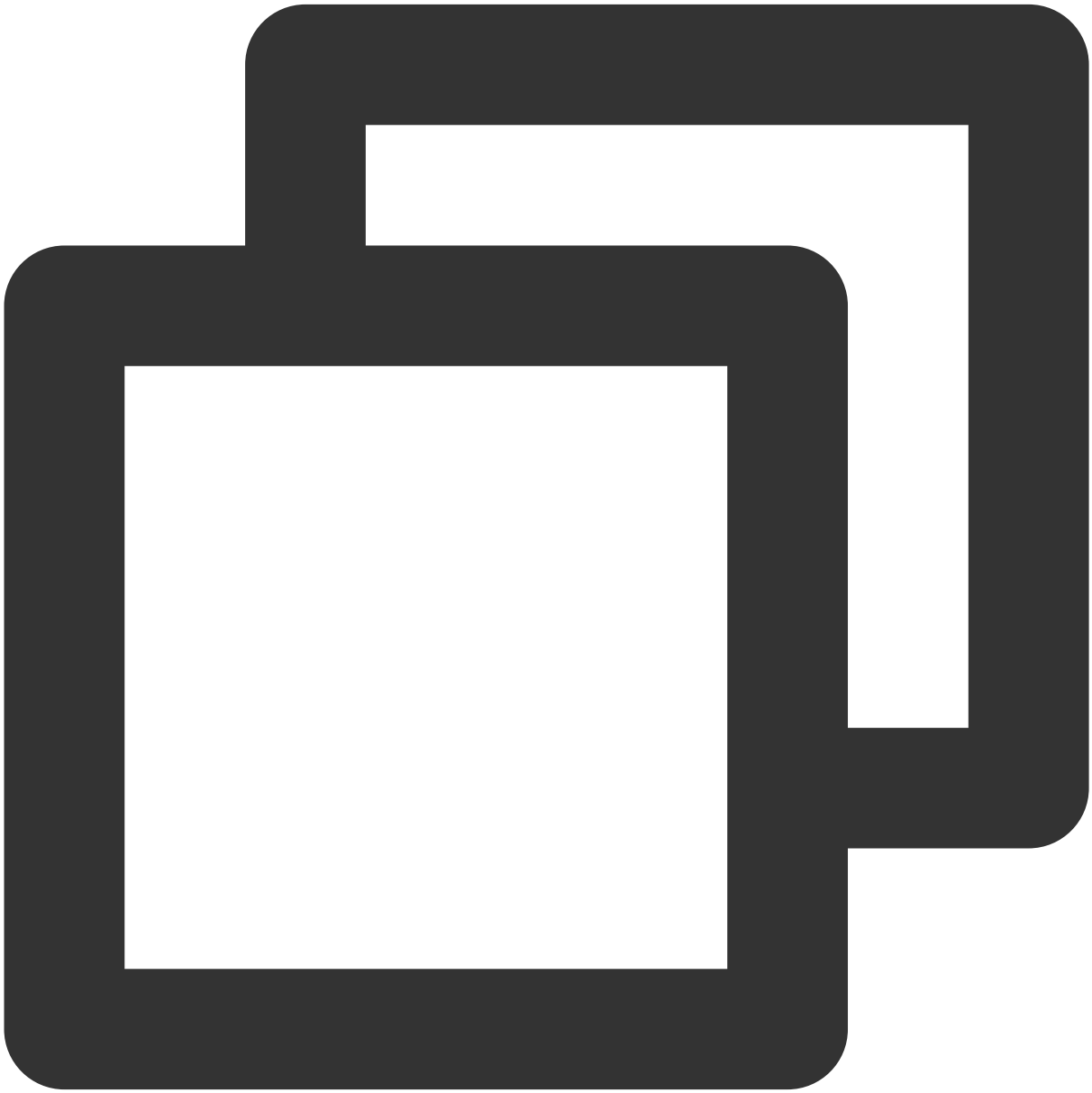
节点名称（关键字）	父节点	描述	类型
DataId	Response.JobsDetail	请求中添加的唯一业务标识。	String
JobId	Response.JobsDetail	本次视频审核任务的 ID。	String
State	Response.JobsDetail	视频审核任务的状态，值为 Submitted（已提交审核）、Snapshoting（视频截帧中）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	视频审核任务的创建时间。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

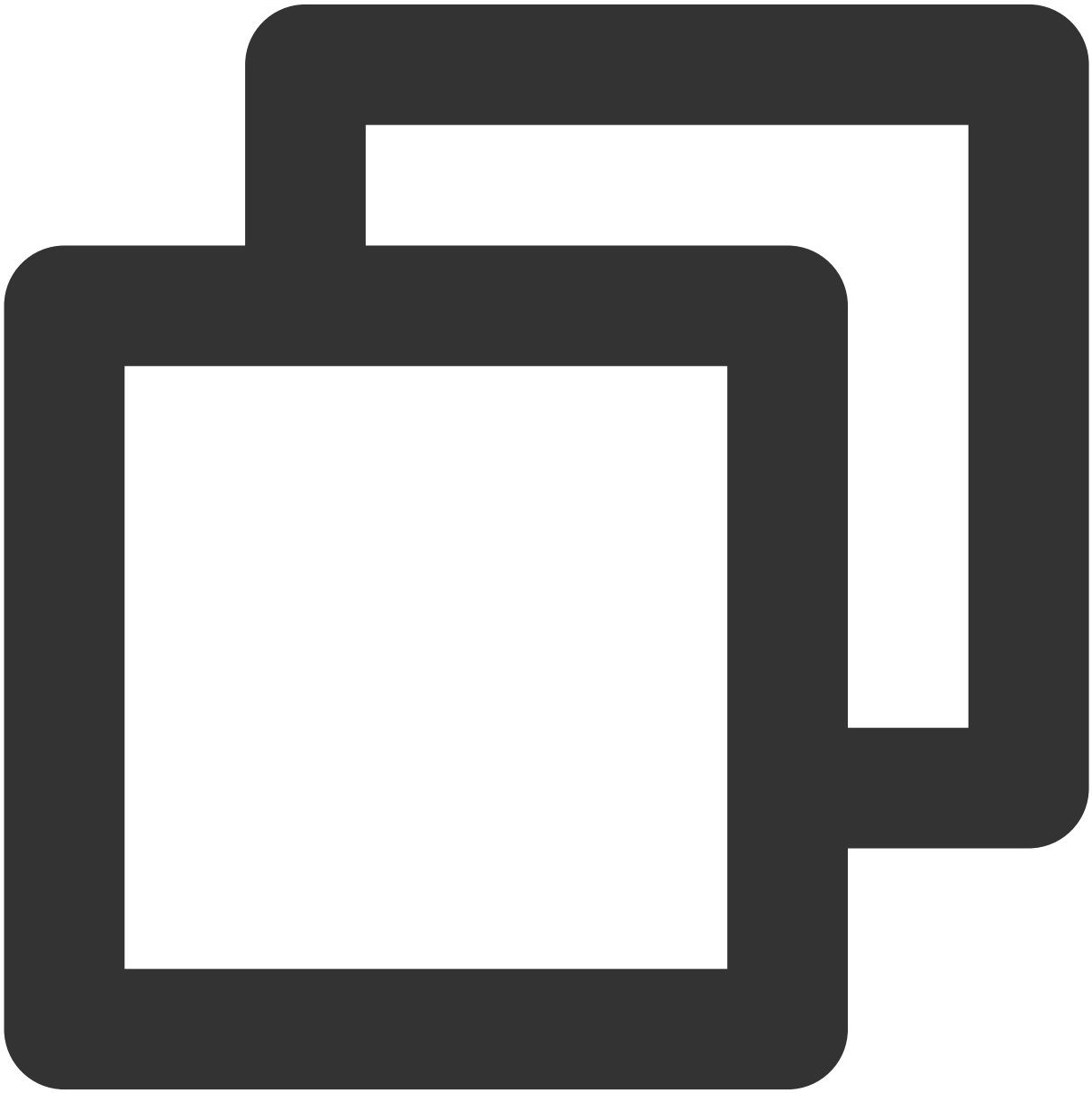
请求



```
POST /video/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Input>
    <Object>a.mp4</Object>
    <DataId>123-fdrsg-123</DataID>
  </Input>
```

```
    <Conf>
      <Snapshot>
          <Mode>Interval</Mode>
          <TimeInterval>50</TimeInterval>
          <Count>100</Count>
      </Snapshot>
      <Callback>http://callback.com/</Callback>
      <BizType>b81d45f94b91a683255e9a9506f45a11</BizType>
      <DetectContent>1</DetectContent>
    </Conf>
  </Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <DataId>123-fdrsg-123</DataID>
```

```
    <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Submitted</State>
    <CreationTime>2021-08-07T12:12:12+0800</CreationTime>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

查询视频审核任务结果

最近更新时间：2023-12-30 09:33:54

功能描述

本接口用于主动查询指定的视频审核任务结果。视频审核功能为异步任务方式，您可以通过提交视频审核任务审核您的视频文件，然后通过查询视频审核任务接口查询审核结果。

说明：

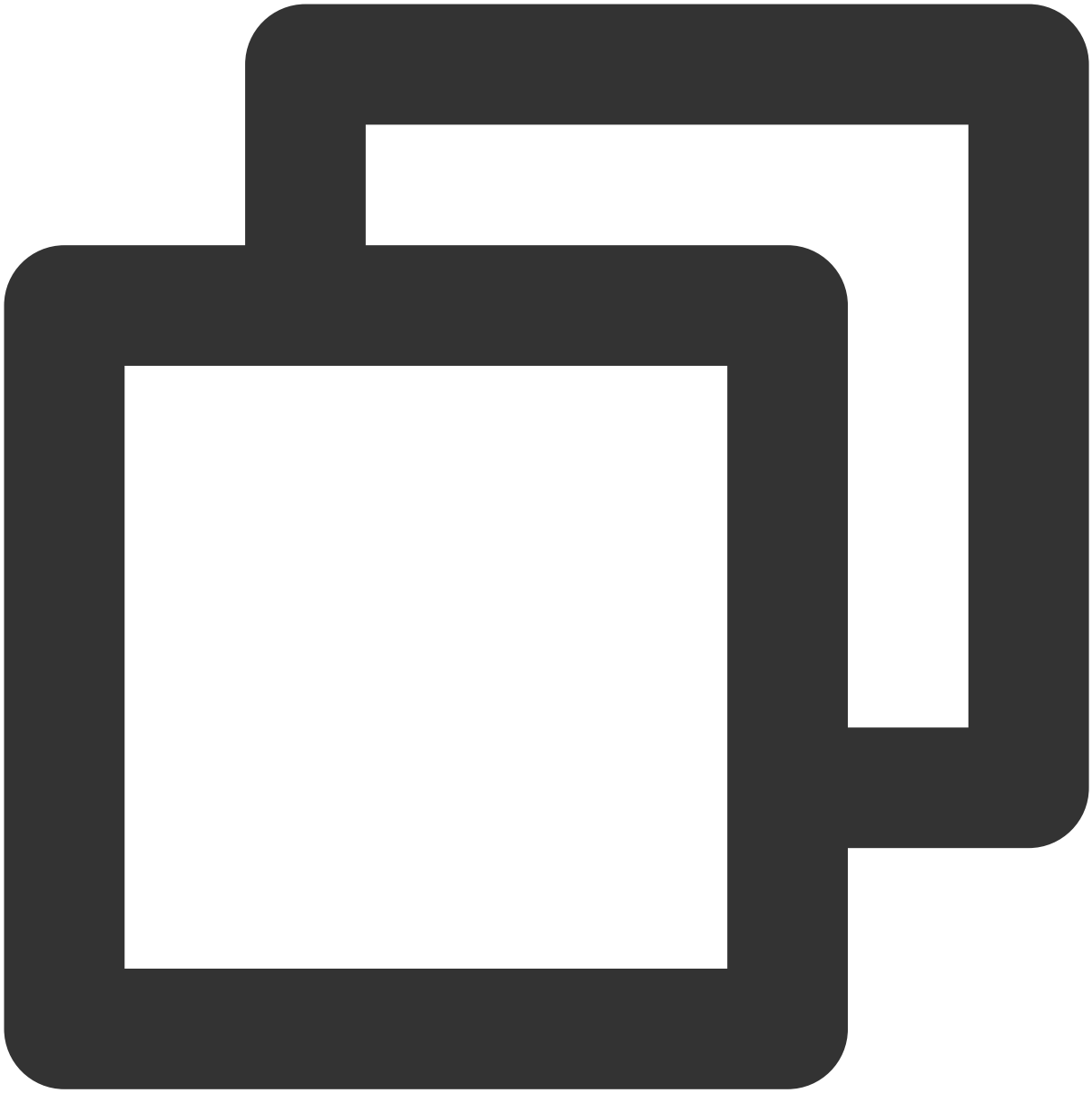
视频审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
GET /video/auditing/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明：

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

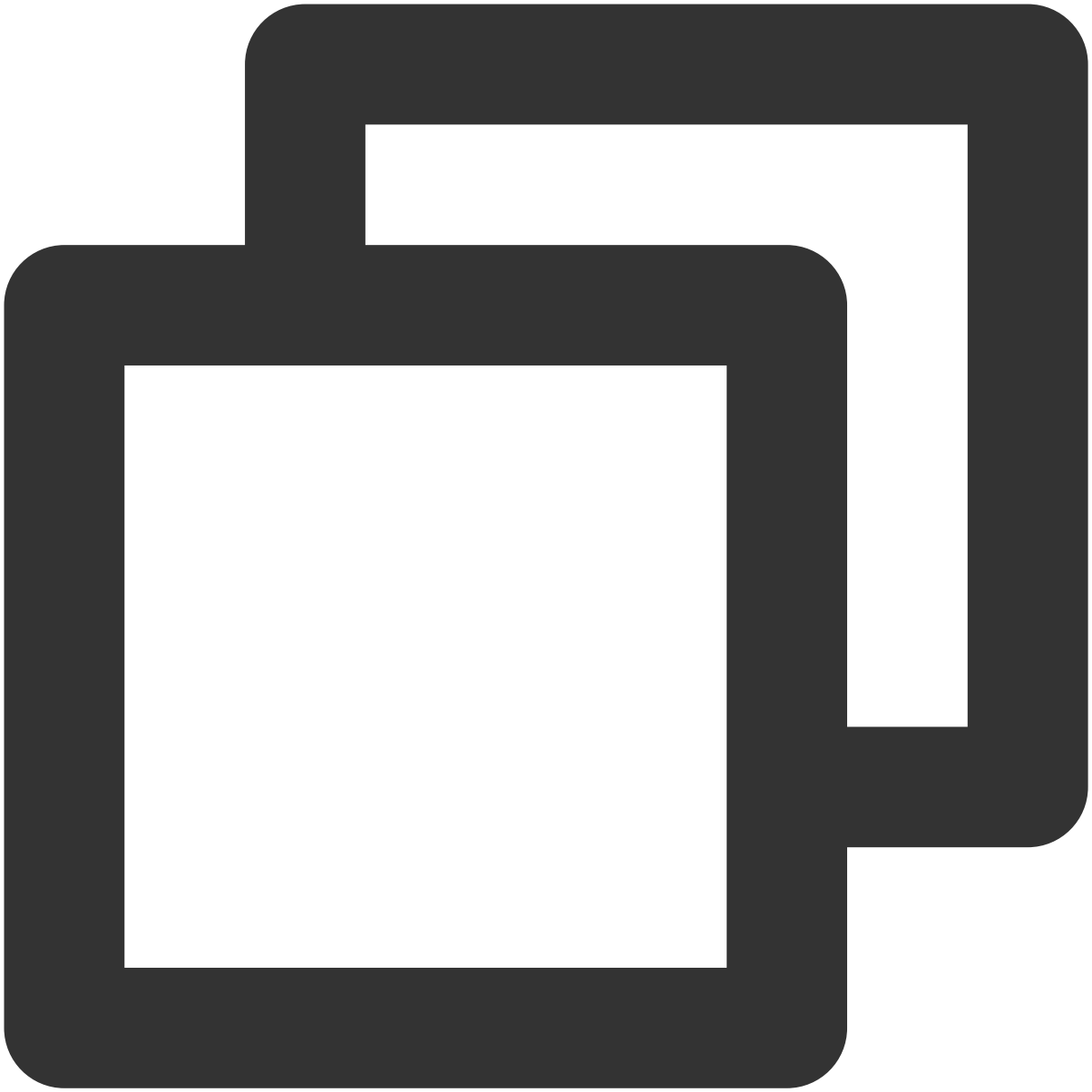
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
    <Code>Success</Code>
    <Message>Success</Message>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <SnapshotCount></SnapshotCount>
    <Result>1</Result>
```

```
    <PornInfo>
        <HitFlag></HitFlag>
        <Count></Count>
    </PornInfo>
    <AdsInfo>
        <HitFlag></HitFlag>
        <Count></Count>
    </AdsInfo>
    <Snapshot>
        <Url></Url>
        <Text></Text>
        <SnapshotTime></SnapshotTime>
        <PornInfo>
            <HitFlag></HitFlag>
            <Score></Score>
            <Label></Label>
            <SubLabel></SubLabel>
        </PornInfo>
        <AdsInfo>
            <HitFlag></HitFlag>
            <Score></Score>
            <Label></Label>
            <SubLabel></SubLabel>
        </AdsInfo>
    </Snapshot>
    <AudioSection>
        <Url></Url>
          <OffsetTime>0</OffsetTime>
          <Duration>3000</Duration>
        <Text>哈哈哈</Text>
        <PornInfo>
            <HitFlag></HitFlag>
            <Score></Score>
            <Keywords></Keywords>
        </PornInfo>
        <AdsInfo>
            <HitFlag></HitFlag>
            <Score></Score>
            <Keywords></Keywords>
        </AdsInfo>
    </AudioSection>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	视频审核的全部结果信息。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	视频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回。详情请查看错误码列表。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	视频审核任务的 ID。	String
State	Response.JobsDetail	视频审核任务的状态，值为 Submitted（已提交审核）、Snapshoting（视频截帧中）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	视频审核任务的创建时间。	String
Object	Response.JobsDetail	本次审核的文件名称，创建任务使用 Object 时返回。	String
Url	Response.JobsDetail	本次审核的文件链接，创建任务使用 Url 时返回。	String
SnapshotCount	Response.JobsDetail	视频截图的总数量。	String
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照	String

		业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	
Result	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail	审核场景为**广告**的审核结果信息。	Container
Snapshot	Response.JobsDetail	该字段用于返回视频中视频画面截图审核的结果。	Container Array
AudioSection	Response.JobsDetail	该字段用于返回视频中视频声音审核的结果。如果未检测音频，则不返回。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。创建任务未设置 UserInfo 时无此字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container
ForbidState	Response.JobsDetail	若您设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Count	Response.JobsDetail.*Info	命中该审核分类的截图张数。	Integer

Container 节点 Snapshot 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.Snapshot	视频截图的访问地址，您可以通过该地址查看该截图内容，地址格式为标准 URL 格式。注意：每次查看数据的有	String

版权所有：腾讯云计算（北京）有限责任公司

		效期为2小时，2小时后如还需查看，请重新发起查询请求。	
SnapshotTime	Response.JobsDetail.Snapshot	该字段用于返回当前截图位于视频中的时间，单位为毫秒。例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前截图的时间戳，单位为毫秒。例如1649387157000。	Integer
Text	Response.JobsDetail.Snapshot	该字段用于返回当前截图的图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	Response.JobsDetail.AudioSection	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
SubLabel	Response.JobsDetail.AudioSection	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Result	Response.JobsDetail.AudioSection	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.Snapshot	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.Snapshot	审核场景为**广告**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Snapshot.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.Snapshot.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最**	Integer

版权所有：腾讯云计算（北京）有限责任公司

		低）-100（**置信度最高**），越高代表该内容越有可能属于当前返回的审核信息。例如：色情 99，则表明该内容非常有可能属于色情内容。	
Label	Response.JobsDetail.Snapshot.*Info	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String
Category	Response.JobsDetail.Snapshot.*Info	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	Response.JobsDetail.Snapshot.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	Response.JobsDetail.Snapshot.*Info	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Container Array
LibResults	Response.JobsDetail.Snapshot.*Info	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
ImageId	Response.JobsDetail.Snapshot.*Info.LibResults	该字段表示命中的风险库中的图片样本 ID。	String
Score	Response.JobsDetail.Snapshot.*Info.LibResults	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本；如：色情 99，则表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称	父节点	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

（关键字）			
Text	Response.JobsDetail.Snapshot.*Info.OcrResults	图片 OCR 文本识别出的具体文本内容。	String
Keywords	Response.JobsDetail.Snapshot.*Info.OcrResults	在当前审核场景下命中的关键词。	String Array
Location	Response.JobsDetail.Snapshot.*Info.OcrResults	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Container

ObjectResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Snapshot.*Info.ObjectResults	该标签用于返回所识别出的实体名称，例如人名。	String
Location	Response.JobsDetail.Snapshot.*Info.ObjectResults	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Container

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯	Float

版权所有：腾讯云计算（北京）有限责任公司

	一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	

Container 节点 AudioSection 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.AudioSection	视频声音片段的访问地址，您可以通过该地址获取该声音片段的内容，地址格式为标准 URL 格式。注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	Response.JobsDetail.AudioSection	该字段用于返回当前视频声音的 ASR 文本识别的检测结果。	String
OffsetTime	Response.JobsDetail.AudioSection	该字段用于返回当前声音片段位于视频中的时间，单位为毫秒，例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前声音片段的时间戳，单位为毫秒，例如1649387157000。	Integer
Duration	Response.JobsDetail.AudioSection	当前视频声音片段的时长，单位毫秒。	Integer
Label	Response.JobsDetail.Snapshot	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail.Snapshot	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.AudioSection	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.AudioSection	审核场景为**广告**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.AudioSection.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	String
Score	Response.JobsDetail.AudioSection.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。	String
Category	Response.JobsDetail.AudioSection.*Info	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	Response.JobsDetail.AudioSection.*Info	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Keywords	Response.JobsDetail.AudioSection.*Info	在当前审核场景下命中的关键词，没有时不返回	String Array
LibResults	Response.JobsDetail.AudioSection.*Info	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
LibType	Response.JobsDetail.AudioSection.*Info.LibResults	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	Response.JobsDetail.AudioSection.*Info.LibResults	命中的风险库名称。	String
Keywords	Response.JobsDetail.AudioSection.*Info.LibResults	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ListResults	Response.JobsDetail.ListInfo	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

节点名称（关键字）	父节点	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

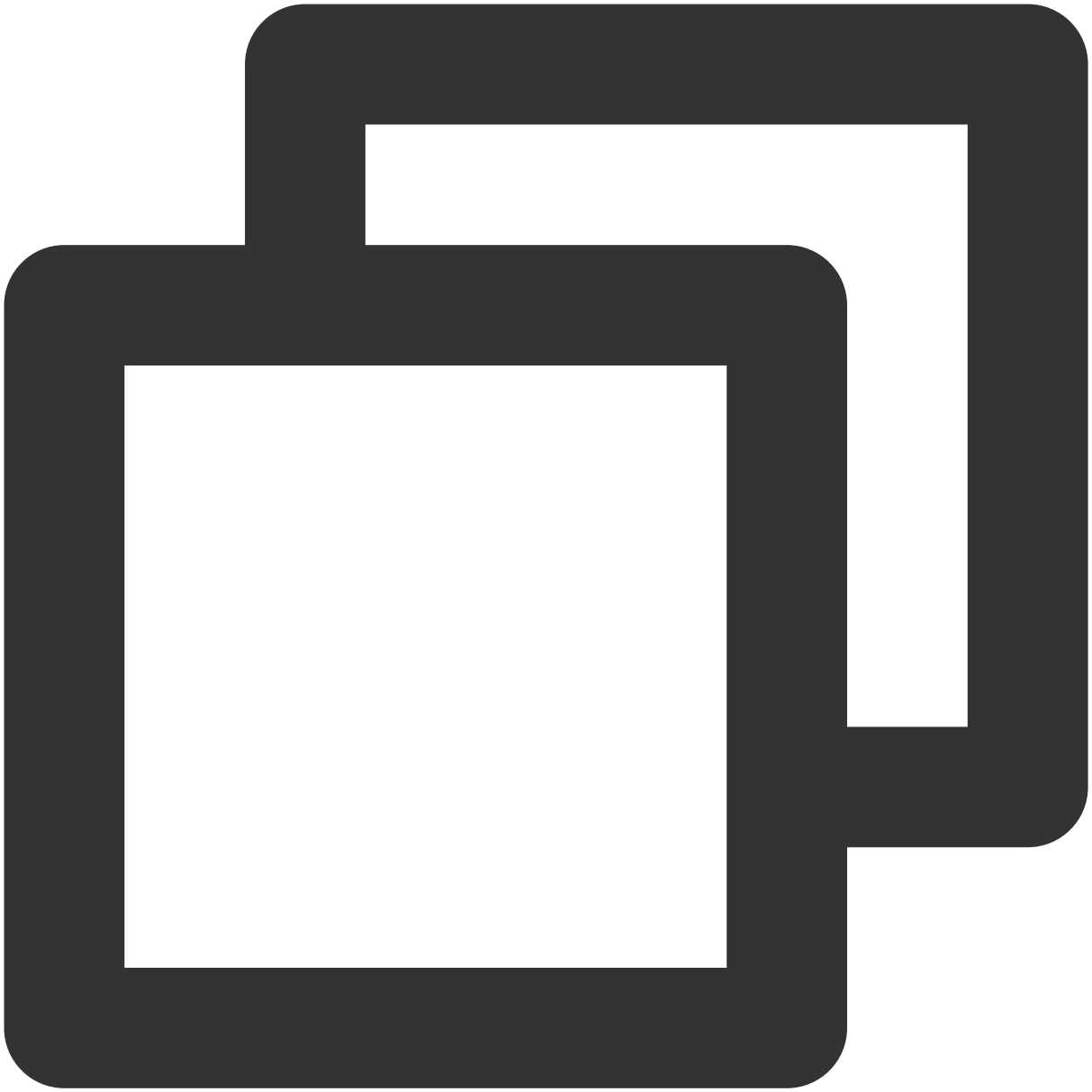
ListType	Response.JobsDetail.ListInfo.ListResults	命中的名单类型，取值为0（白名单）和1（黑名单）。	Integer
ListName	Response.JobsDetail.ListInfo.ListResults	命中的名单名称。	String
Entity	Response.JobsDetail.ListInfo.ListResults	命中了名单中的哪条内容。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

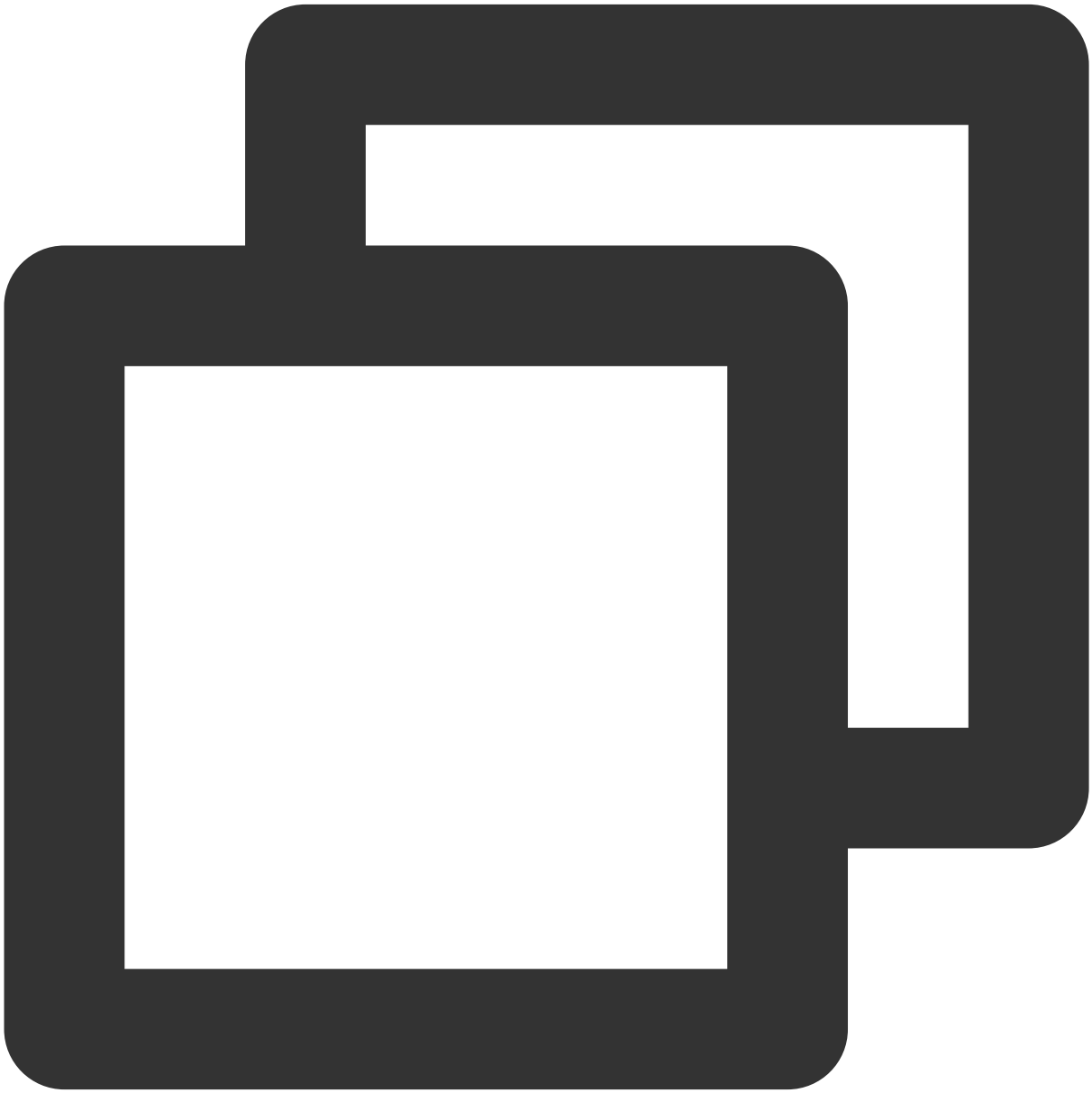
实际案例

请求



```
GET /video/auditing/vab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
```

```
    <State>Success</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Object>a.mp4</Object>
    <SnapshotCount>1</SnapshotCount>
    <Label>Normal</Label>
    <Result>0</Result>
    <PornInfo>
        <HitFlag>0</HitFlag>
        <Count>0</Count>
    </PornInfo>
    <AdsInfo>
        <HitFlag>0</HitFlag>
        <Count>0</Count>
    </AdsInfo>
    <Snapshot>
        <Url>http://examplebucket-1250000000.cos.ap-chongqing.myqcloud.com/0.jpg</U
        <SnapshotTime>0</SnapshotTime>
        <Text>哈哈</Text>
        <Label>Normal</Label>
        <Result>0</Result>
        <PornInfo>
            <HitFlag>0</HitFlag>
            <Score>0</Score>
            <Label></Label>
            <SubLabel></SubLabel>
        </PornInfo>
        <AdsInfo>
            <HitFlag>0</HitFlag>
            <Score>0</Score>
            <Label></Label>
            <SubLabel></SubLabel>
        </AdsInfo>
    </Snapshot>
    <AudioSection>
        <AdsInfo>
            <HitFlag>0</HitFlag>
            <Score>0</Score>
        </AdsInfo>
        <Duration>30000</Duration>
        <OffsetTime>0</OffsetTime>
        <Label>Normal</Label>
        <Result>0</Result>
        <PornInfo>
            <HitFlag>0</HitFlag>
            <Score>0</Score>
        </PornInfo>
        <Text>wc xx</Text>
```

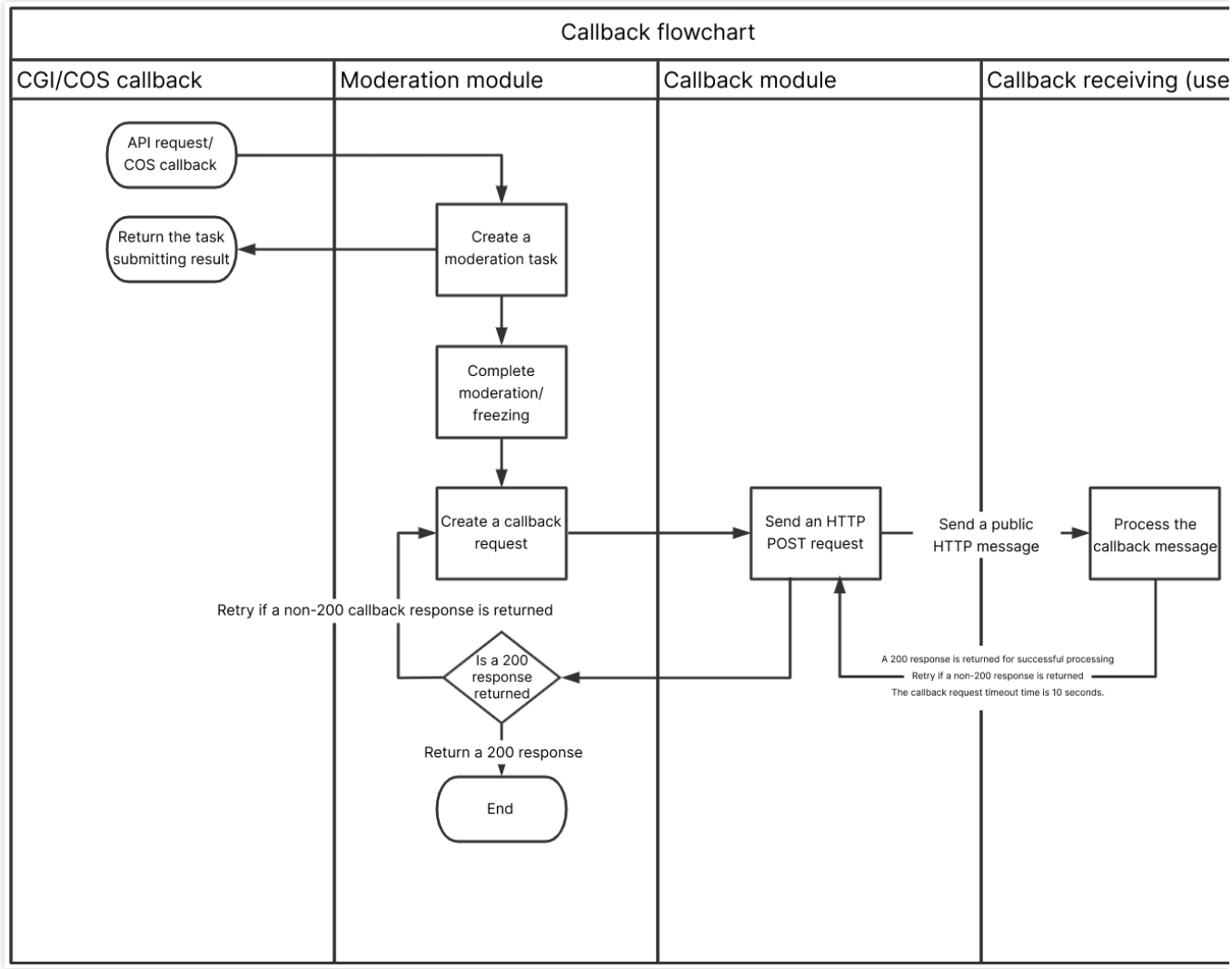
```
        <Url>https://auditing-1250000000.cos.ap-guangzhou.myqcloud.com/q9MhdztF/0.m
    </AudioSection>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

视频审核回调内容

最近更新时间：2023-12-30 09:34:18

功能描述

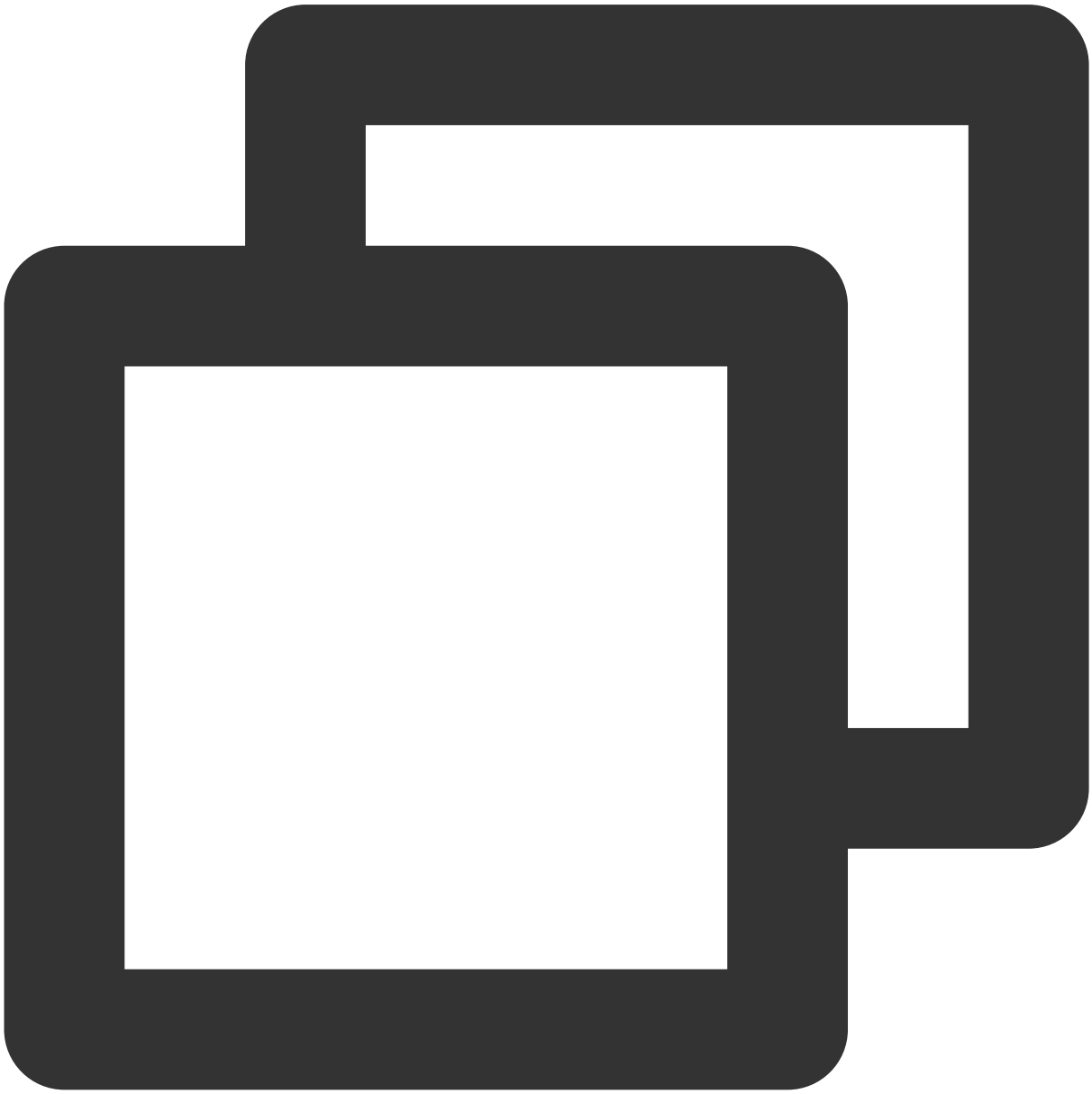
若您配置了审核回调地址，在视频审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。



回调内容说明

简洁回调（Simple）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Simple` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：



```
{
    "code": 0,
    "data": {
        "event":"ReviewVideo",
        "forbidden_status": 0,
        "porn_info": {
```

```
            "hit_flag": 0,
            "label": "",
            "count": 0
        },
        "result": 0,
        "trace_id": "test_trace_id",
        "url": "test_url",
        "cos_headers": {
            "x-cos-meta-xx": "xx"
        }
    },
    "message": "Test request when setting callback url"
}
```


具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码。	Integer	是
message	错误描述。	String	是
data	视频审核结果的详细信息。	Object	是

data 的具体内容如下：

参数名称	描述	类型	是否必选
trace_id	请求的 ID，如果是接口创建的异步任务，则为接口返回的 JobId。	String	是
url	被审核视频的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewVideo。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示视频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer	是
cos_headers	资源上传时设置的自定义 header 内容，若未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否

版权所有：腾讯云计算（北京）有限责任公司

ads_info	审核场景为**广告引导**的审核结果信息。	Object	否
data_id	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String	否

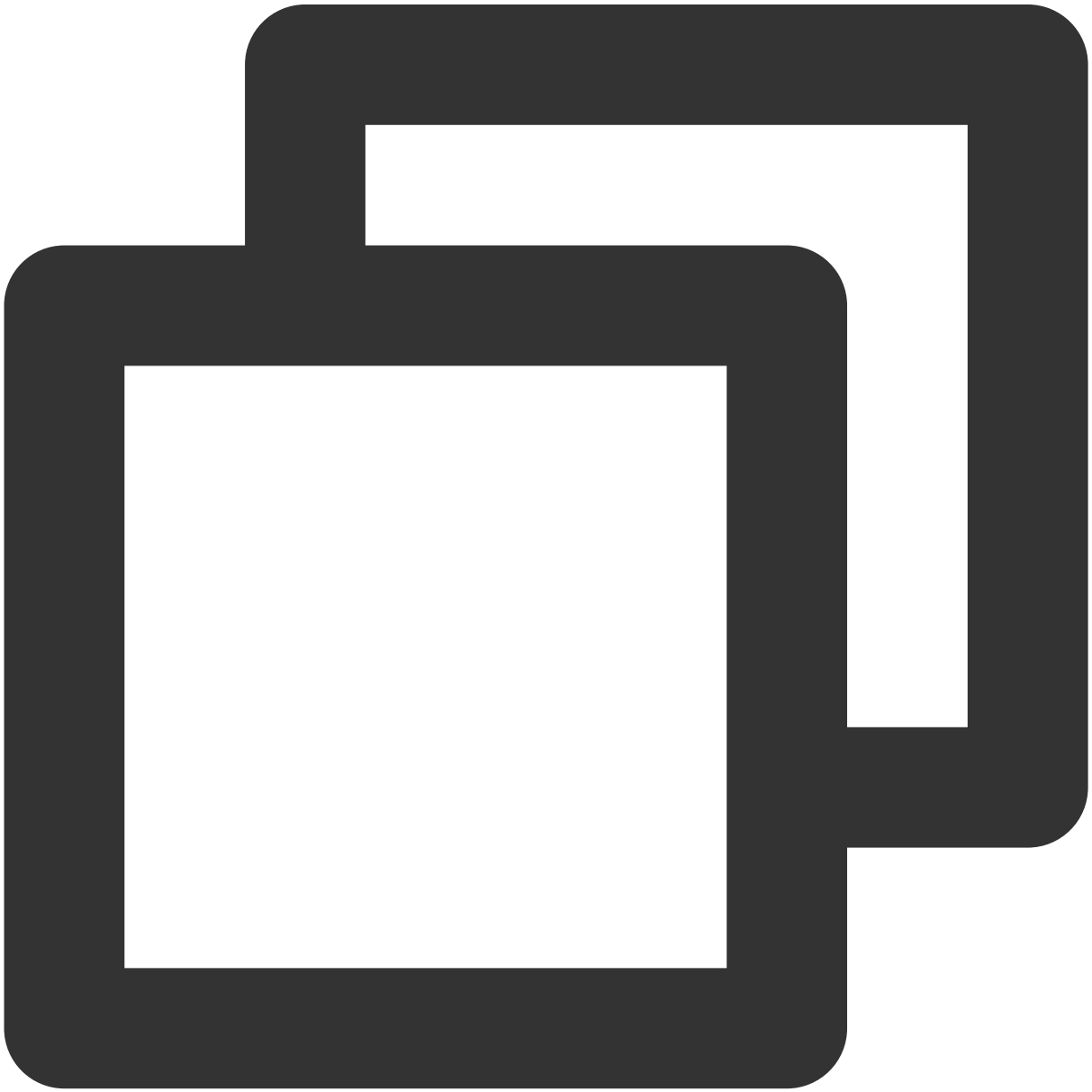
xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer	是
label	命中的标签名称，目前为空。	String	是
count	命中该审核分类的敏感截图张数。	Integer	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。

该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：




```
{
  "EventName": "ReviewVideo",
  "JobsDetail": {
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "",
    "Object": "",
    "Label": "Normal",
    "Result": 0,
    "SnapshotCount": 1,
    "PornInfo": {
```


版权所有：腾讯云计算（北京）有限责任公司

```
      "HitFlag": 0,
      "Count": 0
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Count": 0
    },
    "Snapshot": [
      {
        "Url": "",
        "Text": "",
        "SnapshotTime": 0,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0,
          "Label": "",
          "SubLabel": ""
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0,
          "Label": "",
          "SubLabel": ""
        }
      }
    ],
    "AudioSection": [
      {
        "Url": "",
        "Text": "",
        "OffsetTime": 0,
        "Duration": 30000,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      }
    ],
    "BucketId": "examplebucket-1250000000",
```

```
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
        "x-cos-meta-id": "xxx"
    }
  }
}
```

具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object
EventName	任务的类型， 为 `ReviewVideo` 。	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
DataId	提交任务时设置了DataId参数时返回，返回原始内容，长度限制为512字节。	String
State	视频审核任务的状态，值为 Submitted（已提交审核）、Snapshoting（截图中）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	视频审核任务的创建时间。	String
Object	被审核的视频文件的名称，创建任务使用 Object 时返回。	String
Url	本次审核的文件链接，创建任务使用 Url 时返回。	String
SnapshotCount	视频截图的总数量。直播流状态为 Auditing 的回调中该字段为本次回调包含的截图数量。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议	Integer

	您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
Snapshot	该字段用于返回视频中视频画面截图审核的结果。	Array
AudioSection	该字段用于返回视频中视频声音审核的结果。如果未检测音频，则不返回。	Array
BucketId	创建的审核任务所属的存储桶名称。	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示视频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段。创建任务时未设置 UserInfo 则无此字段。	Object
Type	审核类型，直播流审核返回 live_video。	String
ListInfo	账号黑白名单结果。	Object

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Count	命中该审核分类的截图张数。	Integer

Snapshot 的具体内容如下：

节点名称（关键字）	描述	类型
Url	视频截图的访问地址，您可以通过该地址查看该截图内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
SnapshotTime	该字段用于返回当前截图位于视频中的时间，单位为毫秒。例如5000（视频开	Integer

版权所有：腾讯云计算（北京）有限责任公司

	始后5000毫秒）。审核直播流时，该字段返回当前截图的时间戳，单位为毫秒。例如1649387157000。	
Text	该字段用于返回当前截图的图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Snapshot 中 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Label	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Object Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型

ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

版权所有：腾讯云计算（北京）有限责任公司

AudioSection 的具体内容如下：

节点名称（关键字）	描述	类型
Url	视频声音片段的访问地址，您可以通过该地址获取该声音片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
Text	该字段用于返回当前视频声音的 ASR 文本识别的检测结果。	String
OffsetTime	该字段用于返回当前声音片段位于视频中的时间，单位为毫秒，例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前声音片段的时间戳，单位为毫秒，例如1649387157000。	Integer
Duration	当前视频声音片段的时长，单位毫秒。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	供参考的识别结果，0（正常），1（敏感），2（疑似敏感）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

AudioSection 中 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情99，表明该内容非常有可能属于色情内容。	Integer
Category	该字段表示命中的审核类别。 注意：该字段可能返回空。	String
Keywords	在当前审核场景下命中的关键词，没有时不返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
LibType	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	命中的风险库名称。	String
Keywords	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

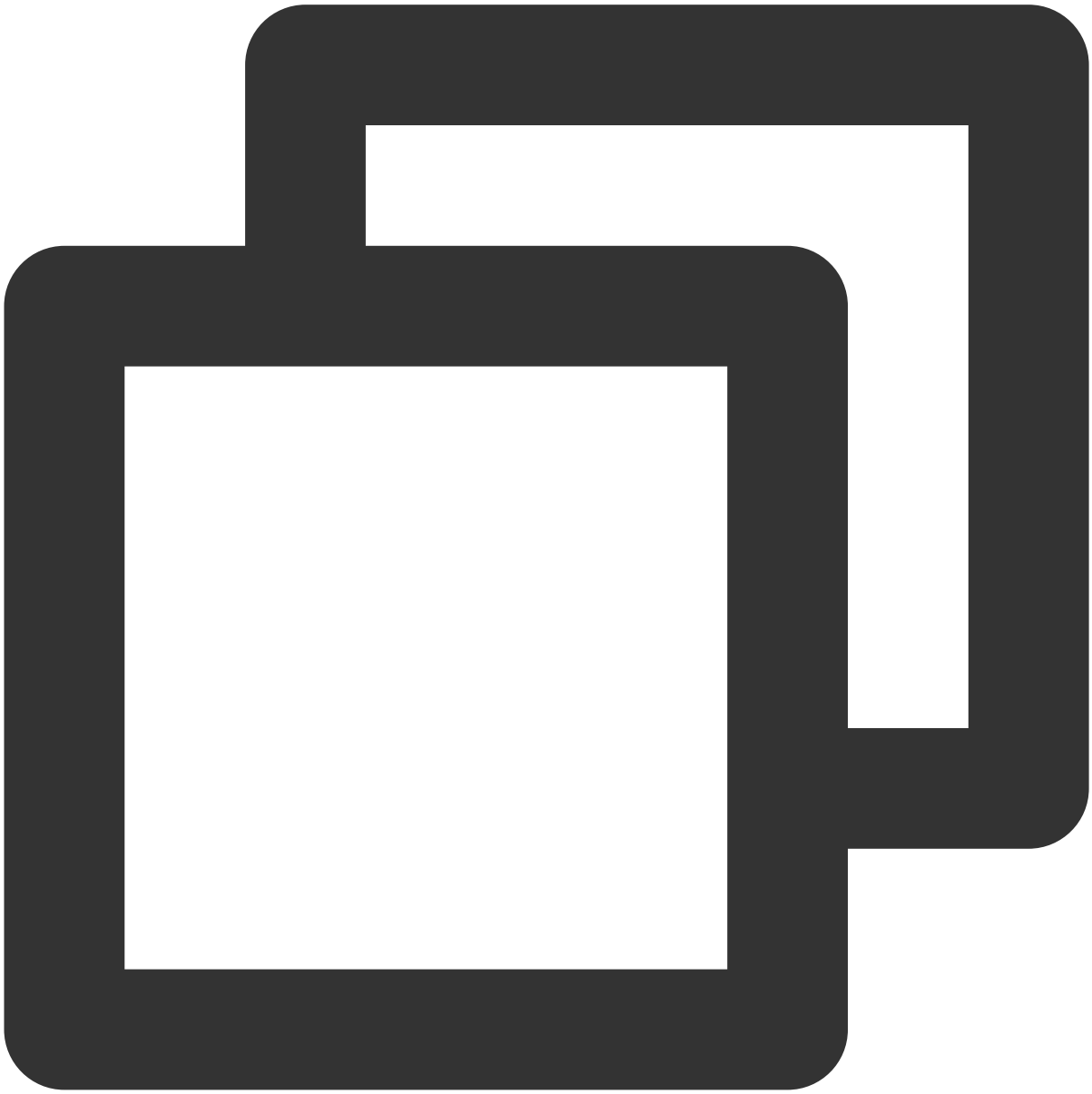
节点名称（关键字）	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

ListType	命中的名单类型，取值为0（白名单）和1（黑名单）。	Integer
ListName	命中的名单名称。	String
Entity	命中了名单中的哪条内容。	String

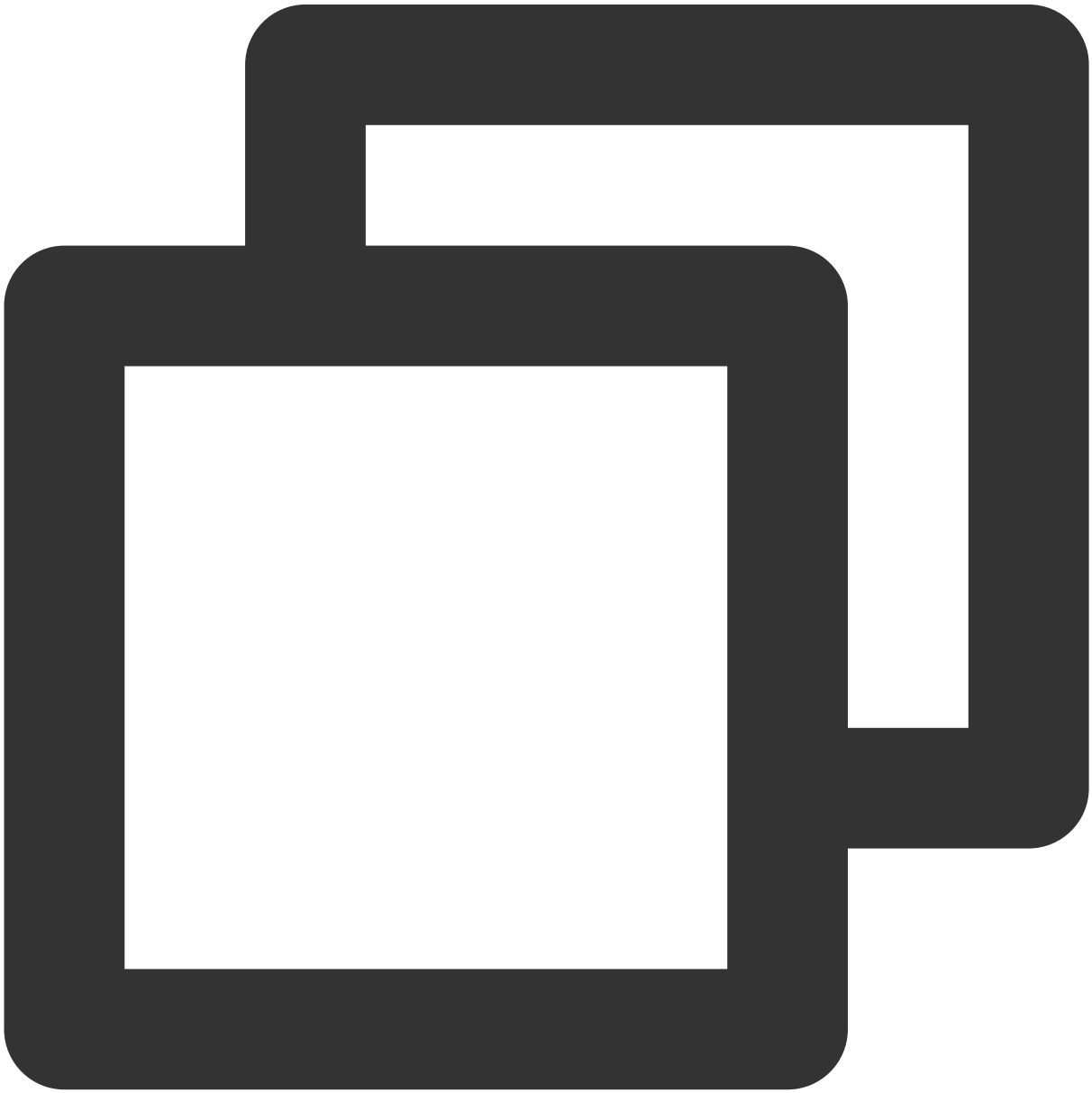
实际案例

案例一：简洁回调（Simple）



```
{
    "code": 0,
    "data": {
        "event":"ReviewVideo",
        "forbidden_status": 0,
        "porn_info": {
            "hit_flag": 0,
            "label": "",
            "count": 0
        },
        "result": 0,
        "trace_id": "vxzt90jl2dfscxxxxxxxxxxxxxxxxx",
        "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/video.
        "cos_headers": {
            "x-cos-meta-id": "666666"
        }
    },
    "message": "success"
}
```


案例二：详细回调（Detail）




```
{
  "EventName": "ReviewVideo",
  "JobsDetail": {
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "2021-08-10T21:01:10+08:00",
    "Object": "1.mp4",
    "Label": "Normal",
    "Result": 0,
    "SnapshotCount": 1,
    "PornInfo": {
```


版权所有：腾讯云计算（北京）有限责任公司

```
      "HitFlag": 0,
      "Count": 0
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Count": 0
    },
    "Snapshot": [
      {
        "Url": "https://video-1250000000.cos.ap-chongqing.myqcloud.com/test/0.jpg",
        "Text": "",
        "SnapshotTime": 41,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0,
          "Label": "",
          "SubLabel": ""
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0,
          "Label": "",
          "SubLabel": ""
        }
      }
    ],
    "AudioSection": [
      {
        "Url": "https://audio-1250000000.cos.ap-guangzhou.myqcloud.com/0.mp3",
        "Text": "",
        "OffsetTime": 0,
        "Duration": 30000,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      }
    ],
    "BucketId": "examplebucket-1250000000",
```

```
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
        "x-cos-meta-id": "xxxx"
    }
  }
}
```

音频审核
提交音频审核任务

最近更新时间：2023-12-30 10:32:07

功能描述

本接口用于提交一个音频审核任务。音频审核功能为异步任务方式，您可以通过提交音频审核任务审核您的音频文件，然后通过 查询音频审核任务接口 或 音频审核回调 查询审核结果。
该接口支持情况如下：
说明：
支持审核腾讯云 COS 上的音频文件。
支持审核第三方云存储厂商的音频 URL。
支持对音频文件进行自动检测，从 ASR 语音文本识别、色情呻吟两个维度，通过深度学习技术识别音频中的违规内容。
支持设置回调地址 Callback 获取检测结果，或通过 查询音频审核任务结果接口 主动轮询获取审核结果详情。
支持识别多种违规场景，包括：色情、广告等场景。
支持根据不同的业务场景 配置自定义的审核策略。

费用说明

每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1个音频**，**计2次**审核费用。
调用接口会产生音频审核费用和 COS 读请求费用。
如果音频属于 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用。
不支持审核 COS 上的归档存储类型和深度归档存储类型的音频，如果需要审核此类型音频，请先 恢复归档文件。

限制说明

音频文件大小支持：**文件 < 600MB**。
音频文件时长支持：**< 3小时**。
音频码率类型支持：128Kbps - 256Kbps。
音频文件支持格式：mp3、wav、aac、flac、amr、3gp、m4a、wma、ogg、ape。
音频文件内容语言支持：普通话、英语、粤语。
当输入为视频文件时，支持分离视频文件音轨，并对音频内容进行独立审核。

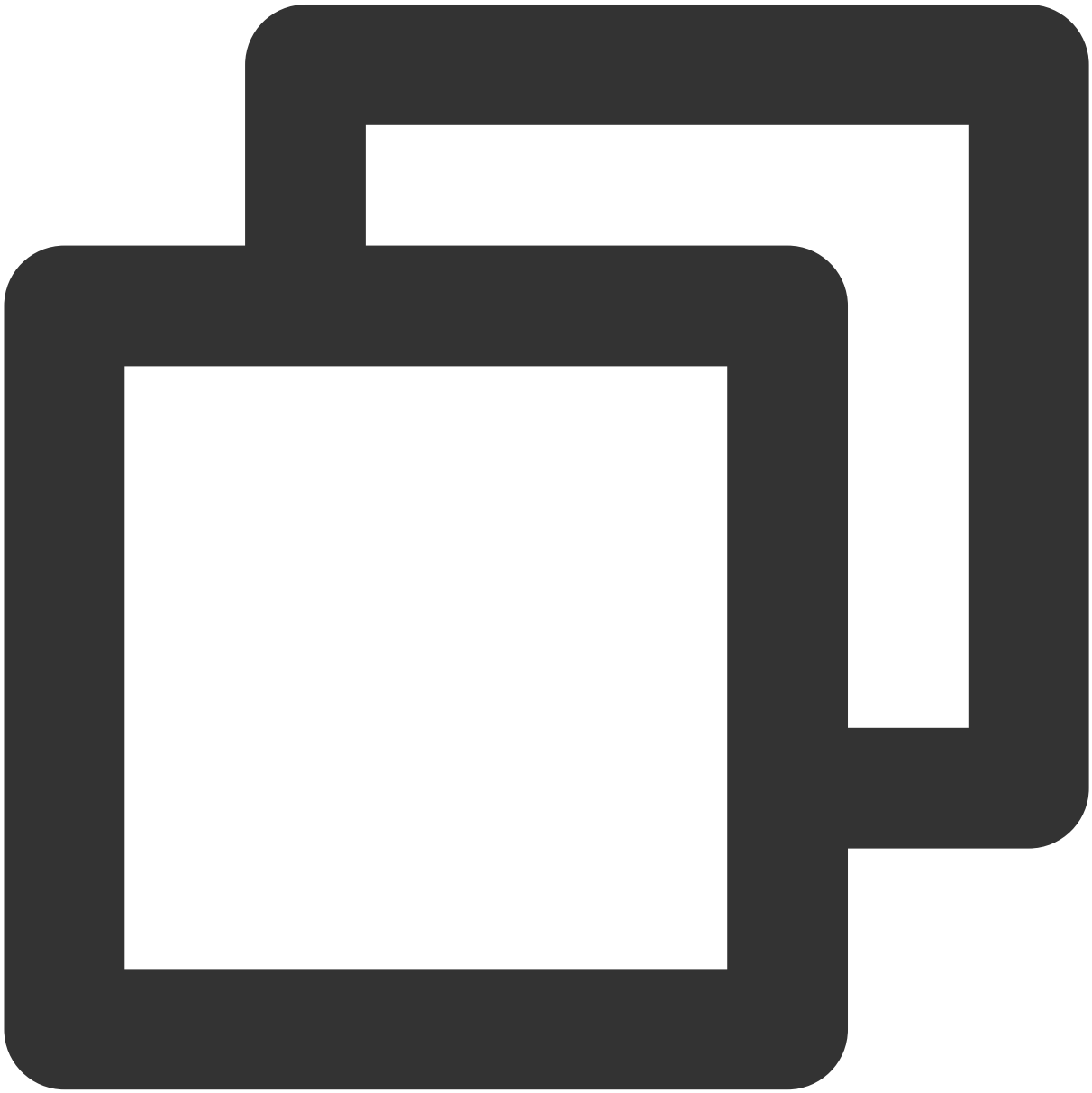
版权所有：腾讯云计算（北京）有限责任公司

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
POST /audio/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
<body>
```

说明：

Authorization: Auth String（详情请参见 请求签名 文档）。

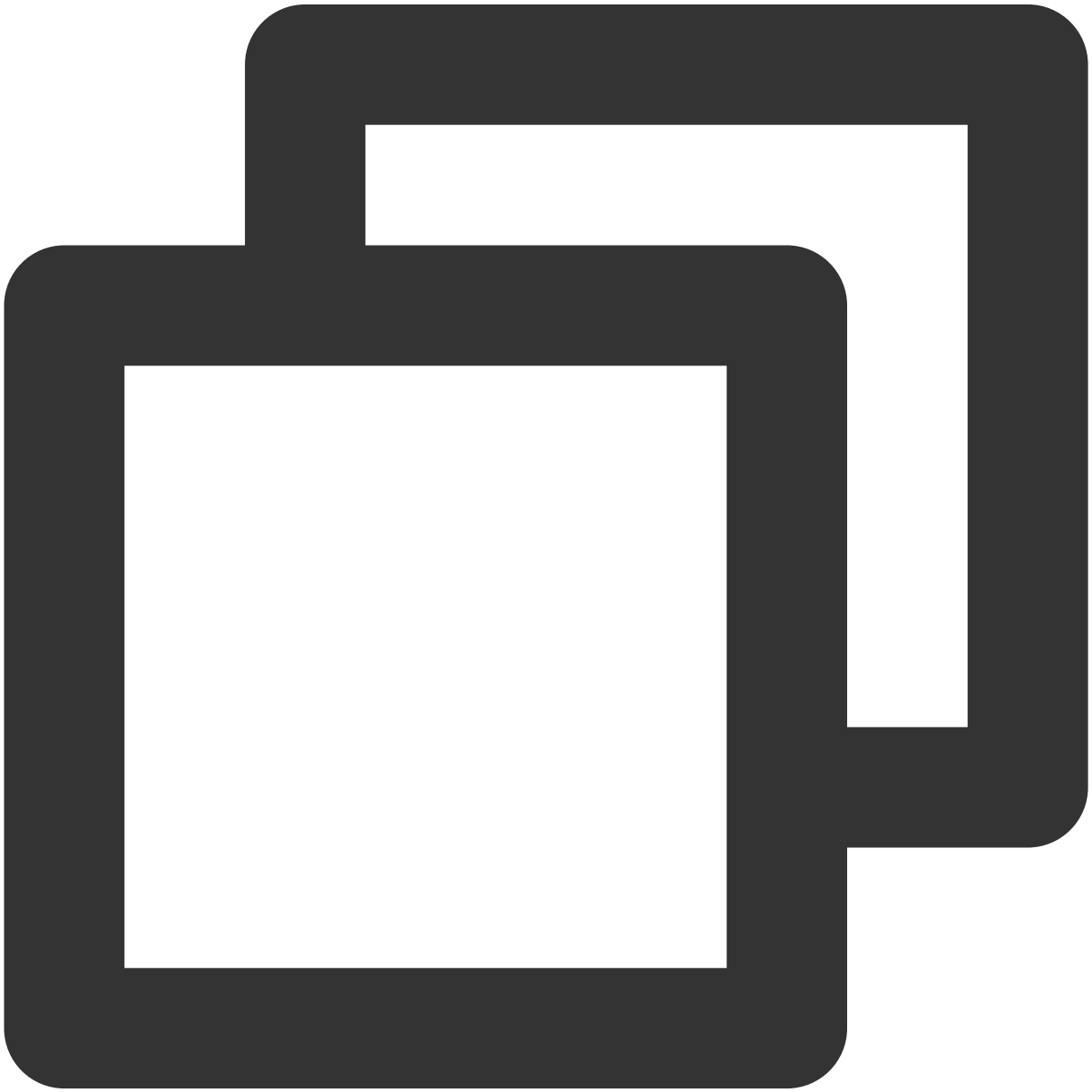
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
    <Input>
        <Object></Object>
        <Url></Url>
        <DataId></DataId>
    </Input>
    <Conf>
        <Callback></Callback>
        <BizType></BizType>
    </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	音频审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容。	Container	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的音频文件名称，例如在目录 test 中的文件 audio.mp3，则文件名称为 test/audio.mp3。Object 和 Url 只能选择其中一种。	String	否
Url	Request.Input	音频文件的链接地址，例如 `http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/audio.mp3` Object 和 Url 只能选择其中一种。	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	用户业务 TokenId，长度不超过128字节。	String	否

Nickname	Request.Input.UserInfo	用户业务 Nickname，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	用户业务 DeviceId，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	用户业务 AppId，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	用户业务 Room，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	用户业务 IP，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	用户业务 Type，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	用户业务 ReceiveTokenId，长度不超过128字节。	String	否
Gender	Request.Input.UserInfo	用户业务 Gender，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	用户业务 Level，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	用户业务 Role，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如色情、广告等，配置指引：设置公共审核策略。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Callback	Request.Conf	审核结果以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。	String	否
CallbackVersion	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	string	否

CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部音频片段）、2（回调违规音频片段）。默认为 1。	Integer	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对音频文件进行自动冻结，仅当 `input` 中审核的音频为 `object` 时有效。	Container	否

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请 联系服务人员咨询。

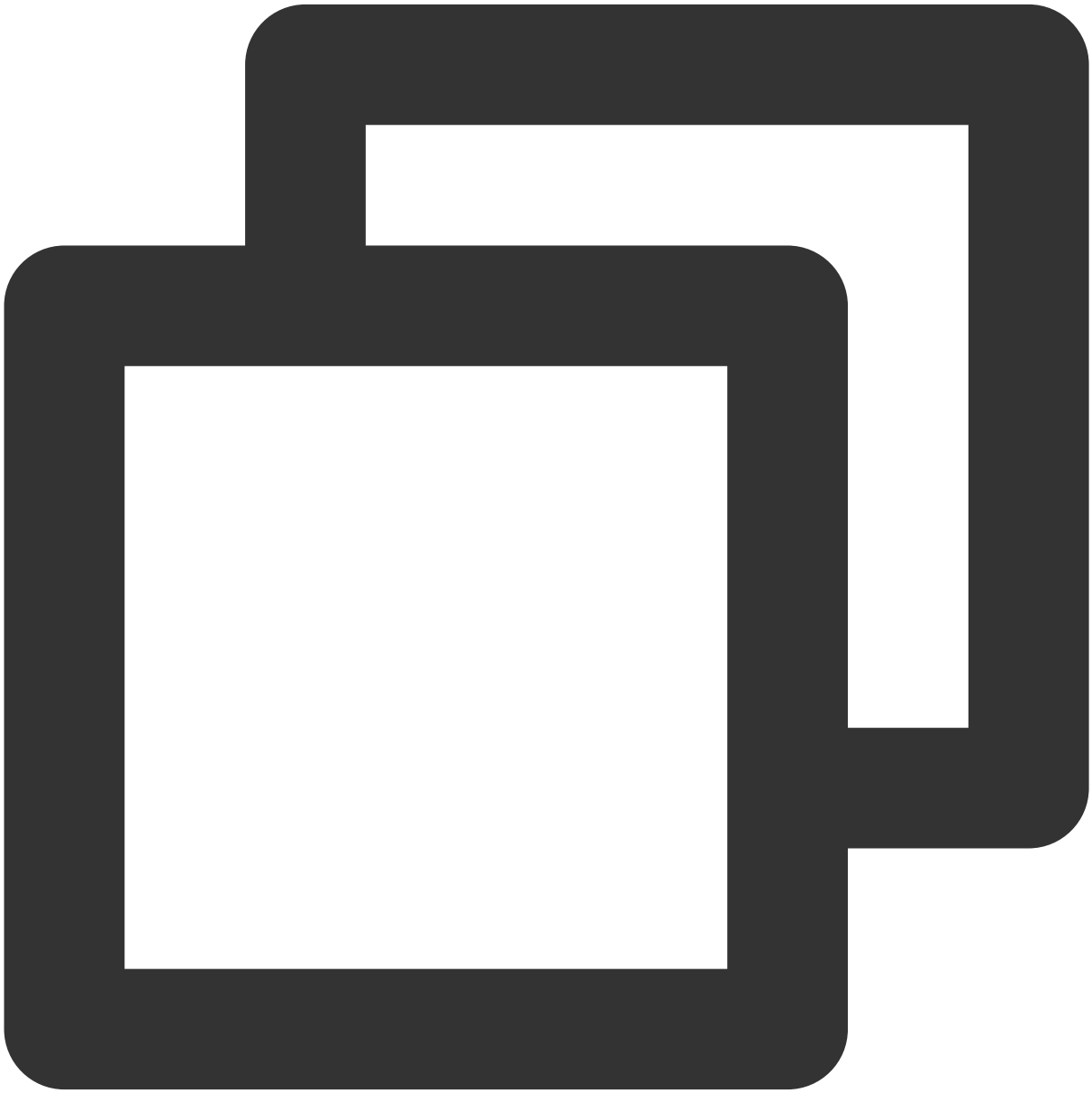
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
        <DataId></DataId>
        <JobId></JobId>
        <State></State>
        <CreationTime></CreationTime>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	音频审核返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	音频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

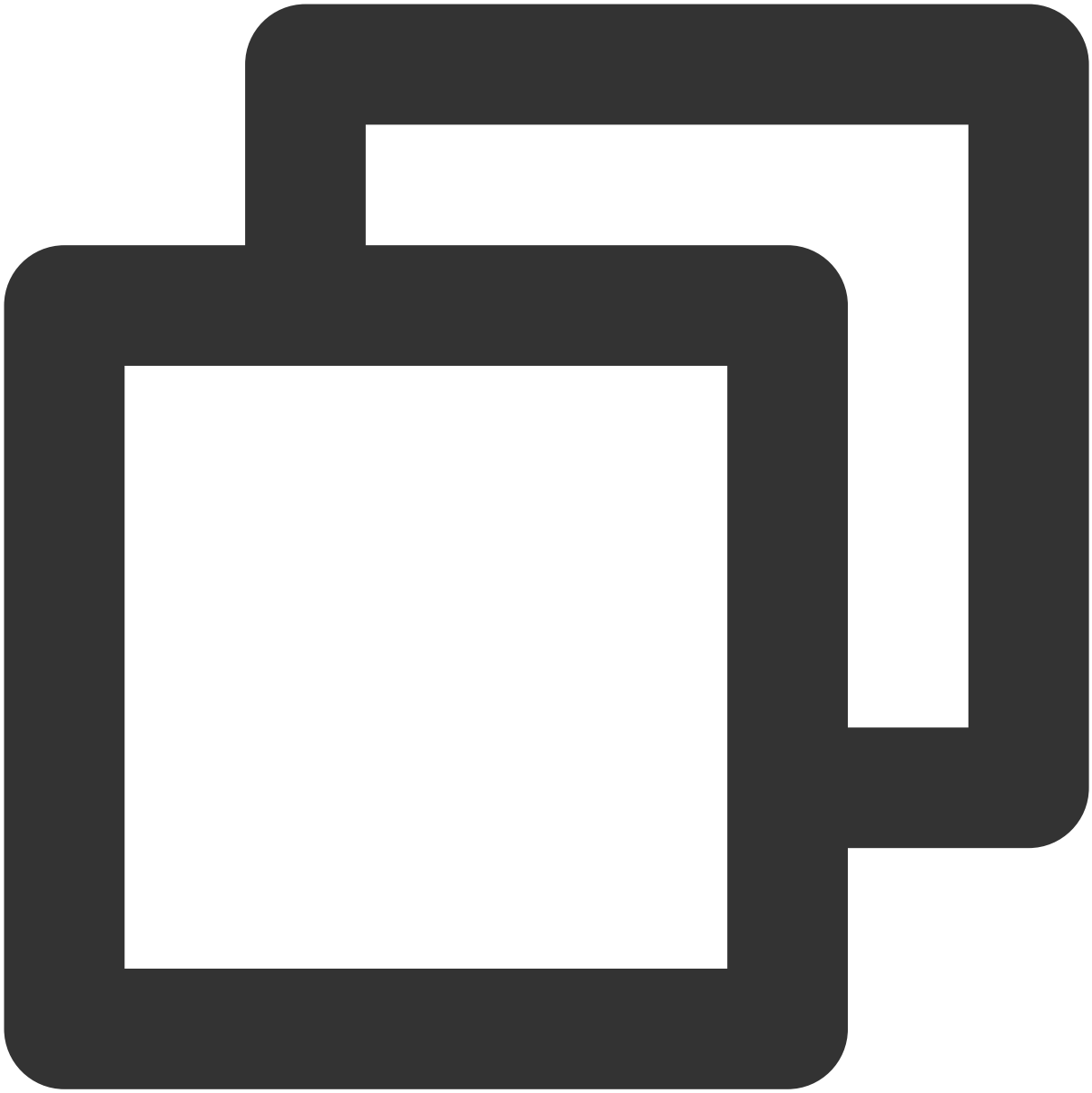
节点名称（关键字）	父节点	描述	类型
DataId	Response.JobsDetail	请求中添加的唯一业务标识。	String
JobId	Response.JobsDetail	本次音频审核任务的 ID。	String
State	Response.JobsDetail	音频审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	音频审核任务的创建时间。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

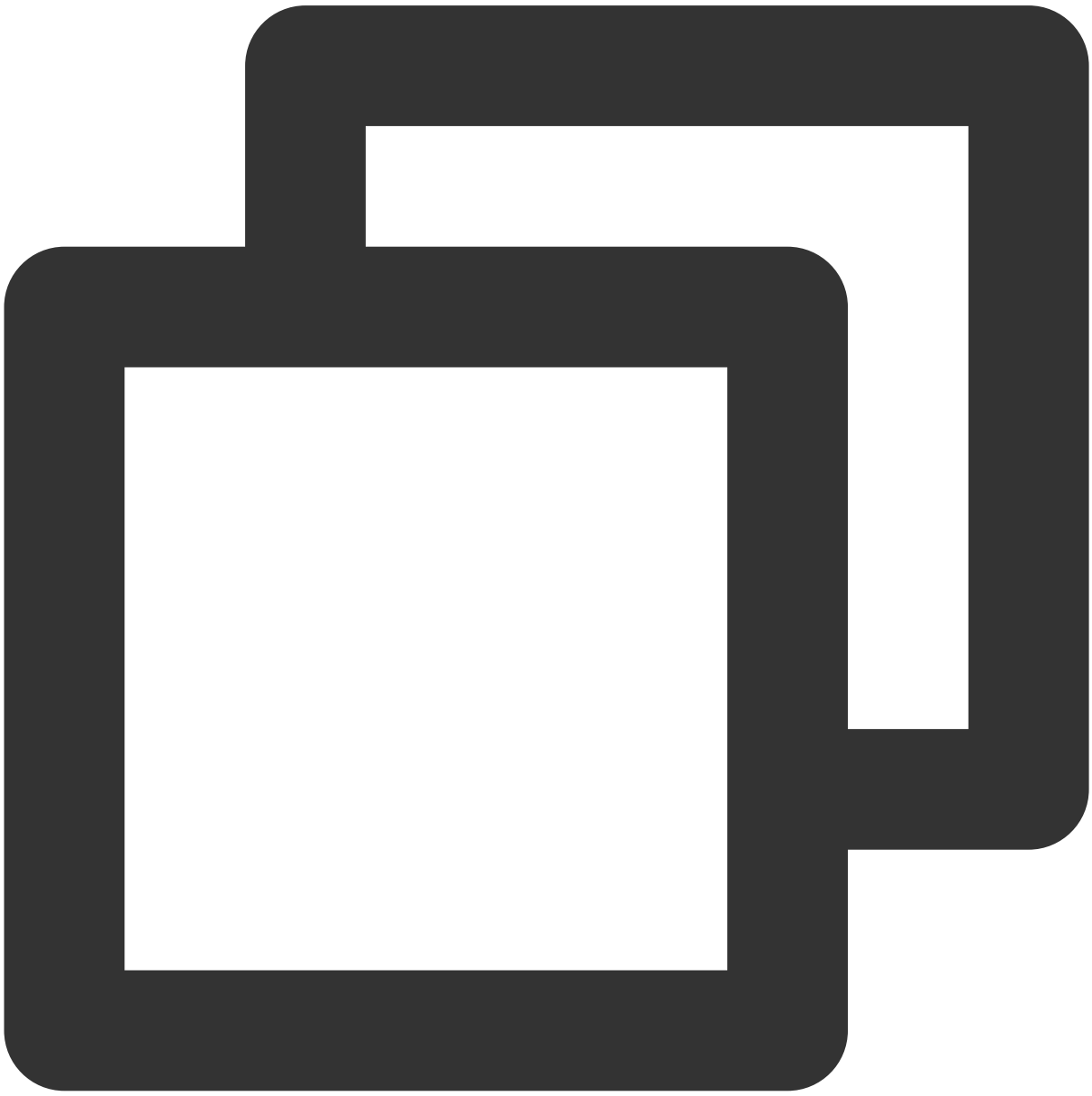
请求



```
POST /audio/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Input>
        <Object>a.mp3</Object>
        <DataId>123-fdrsg-123</DataID>
    </Input>
```

```
    <Conf>
        <BizType>b81d45f94xxxxxxxxa9506f45a11</BizType>
        <Callback>http://callback.com/</Callback>
    </Conf>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
```

```
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <DataId>123-fdrsg-123</DataID>
        <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
        <State>Submitted</State>
        <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    </JobsDetail>
    <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

查询音频审核任务结果

最近更新时间：2023-12-30 10:32:13

功能描述

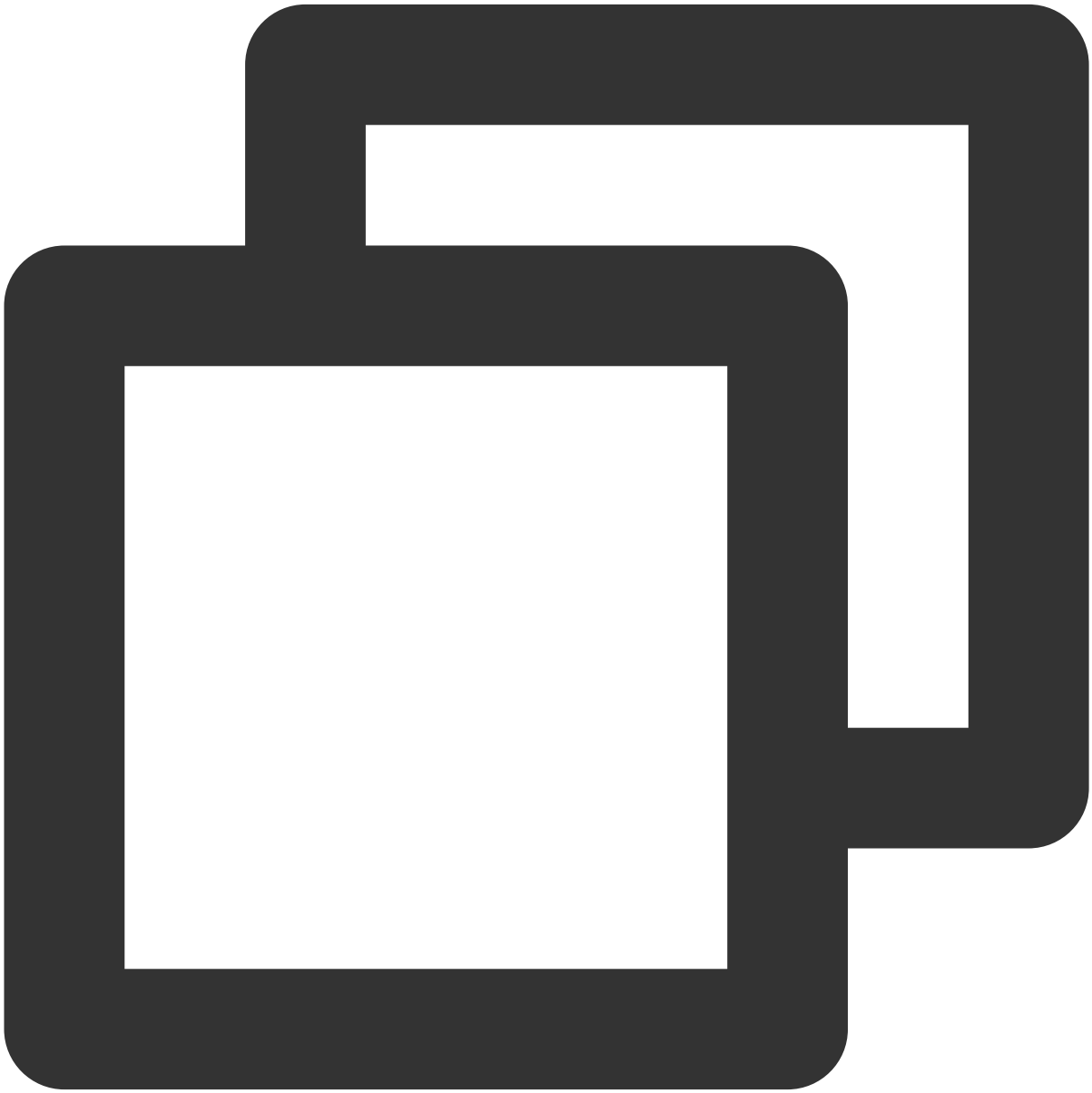
本接口用于查询一个音频审核任务的状态或结果。

说明：

音频审核任务结果保存时效为3个月，您可通过本接口查询近3个月的审核结果。

请求

请求示例



```
GET /audio/auditing/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明：

Authorization: Auth String （详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

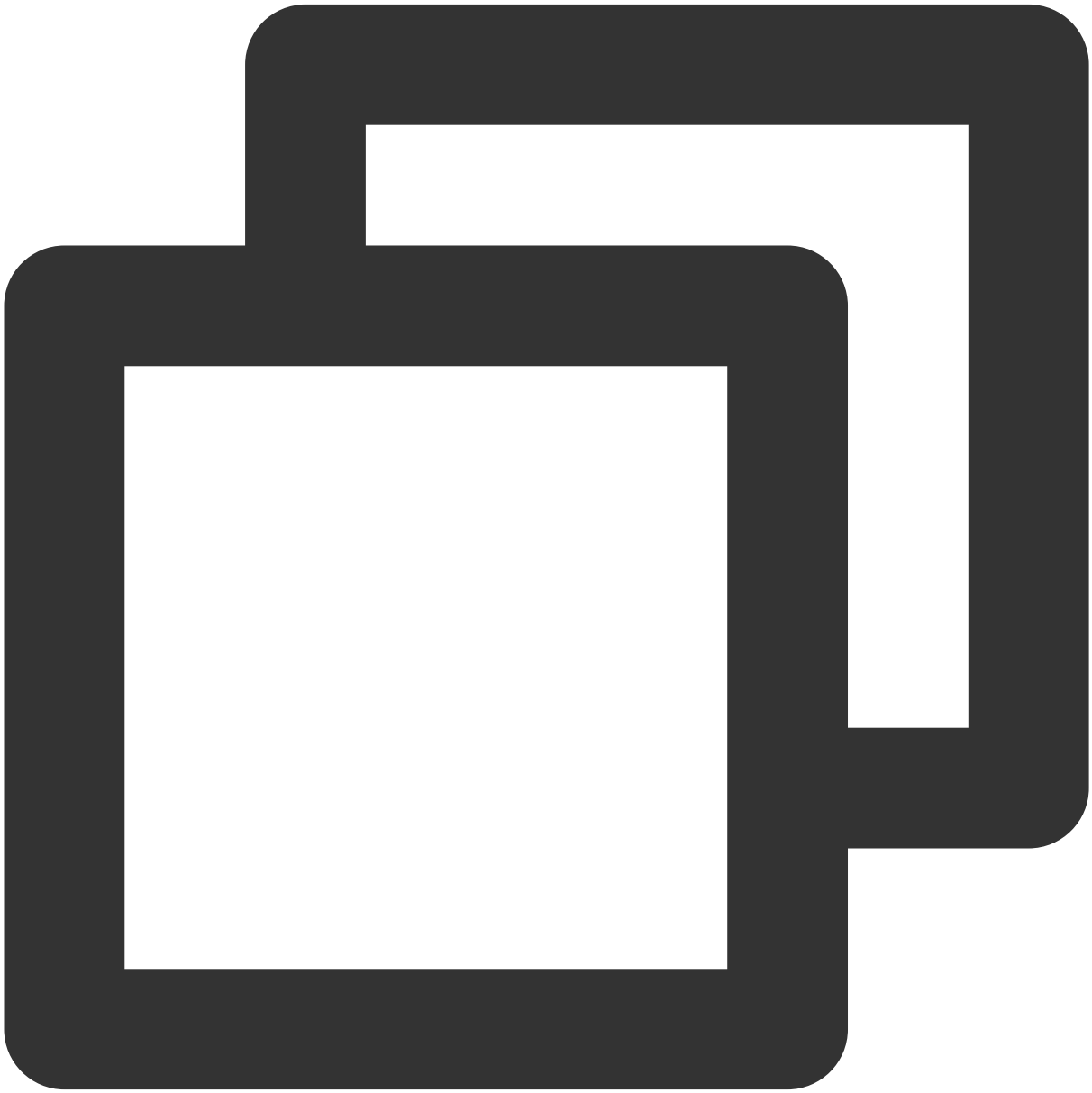
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
    <Code>Success</Code>
    <Message>Success</Message>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <Result>1</Result>
    <AudioText></AudioText>
```

```
    <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Label></Label>
    </PornInfo>
    <Section>
        <Url>https://cos.ap-guangzhou.myqcloud.com/0.mp3</Url>
          <OffsetTime>0</OffsetTime>
          <Duration>3000</Duration>
        <Text>哈哈哈</Text>
        <PornInfo>
            <HitFlag></HitFlag>
            <Score></Score>
            <Keywords></Keywords>
        </PornInfo>
    </Section>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	音频审核任务结果。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	音频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回。详情请查看错误码列表。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	该字段在审核结果中会返回原始内容，长度限制为	String

版权所有：腾讯云计算（北京）有限责任公司

		512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	
JobId	Response.JobsDetail	本次音频审核任务的 ID。	String
State	Response.JobsDetail	音频审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	音频审核任务的创建时间。	String
Object	Response.JobsDetail	审核的音频文件为存储在 COS 中的文件时，该字段表示本次审核的音频文件名称。	String
Url	Response.JobsDetail	审核的音频文件为一条文件链接时，该字段表示本次审核的音频文件链接。	String
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
AudioText	Response.JobsDetail	该字段用于返回音频文件中已识别的对应文本内容。	String
PornInfo	Response.JobsDetail	审核场景为**涉黄**的审核结果信息。	Container
AdsInfo	Response.JobsDetail	审核场景为**广告引导**的审核结果信息。	Container
Section	Response.JobsDetail	当音频过长时，会对音频进行分段，该字段用于返回音频片段的审核结果，主要包括开始时间和音频审核的相应结果。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。创建任务未设置 UserInfo 时无此字段。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型

HitFlag	Response.JobsDetail.*Info	用于返回该审核场景的审核结果，返回值： 0：正常。 1：确认为当前场景的违规内容。 2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。 例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Label	Response.JobsDetail.*Info	本次审核的结果标签，如果命中了敏感的关键词，该字段返回对应的关键词。	String

Container 节点 Section 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.Section	当前音频片段的访问地址，您可以通过该地址获取该音频片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	Response.JobsDetail.Section	该字段用于返回当前音频片段的 ASR 文本识别的检测结果。	String
OffsetTime	Response.JobsDetail.Section	该字段用于返回当前音频片段位于总音频中的时间，单位为毫秒，例如5000（音频开始后5000毫秒）。	Integer
Duration	Response.JobsDetail.Section	当前音频片段的时长，单位毫秒。	Integer
Label	Response.JobsDetail.Section	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail.Section	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer

PornInfo	Response.JobsDetail.Section	审核场景为**涉黄**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.Section	审核场景为**广告引导**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Section.*Info	用于返回该审核场景的审核结果，返回值： 0：正常。 1：确认为当前场景的违规内容。 2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.Section.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。 例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Category	Response.JobsDetail.Section.*Info	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
Keywords	Response.JobsDetail.Section.*Info	本次审核命中的敏感关键词，没有时不返回。	String Array

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型
TokenId	Response.JobsDetail.UserInfo	用户业务 TokenId，长度不超过128字节。	String
Nickname	Response.JobsDetail.UserInfo	用户业务 Nickname，长度不超过128字节。	String
DeviceId	Response.JobsDetail.UserInfo	用户业务 DeviceId，长度不超过128字节。	String
AppId	Response.JobsDetail.UserInfo	用户业务 AppId，长度不超过128字节。	String
Room	Response.JobsDetail.UserInfo	用户业务 Room，长度不超过128字节。	String

版权所有：腾讯云计算（北京）有限责任公司

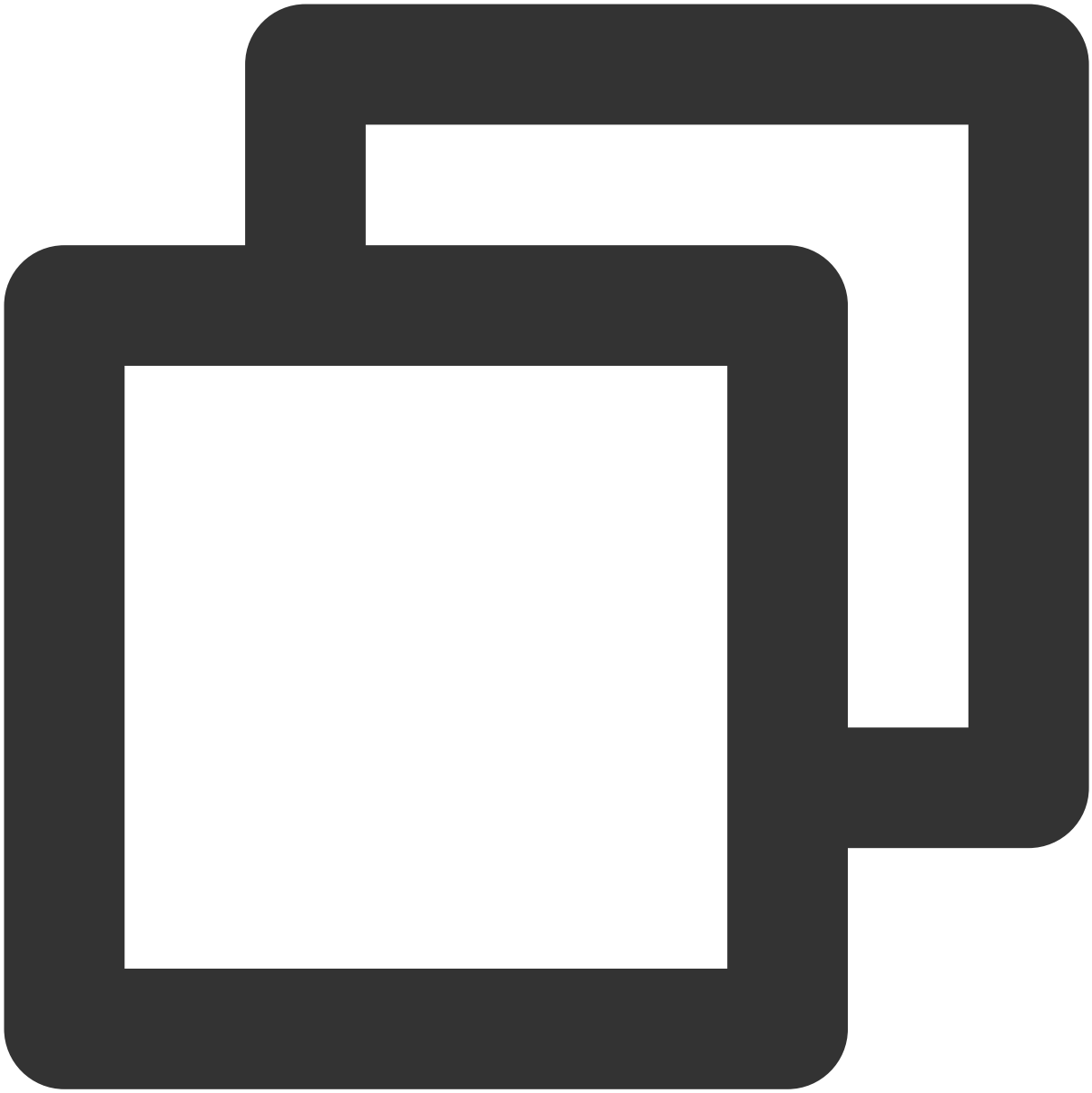
IP	Response.JobsDetail.UserInfo	用户业务 IP，长度不超过128字节。	String
Type	Response.JobsDetail.UserInfo	用户业务 Type，长度不超过128字节。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

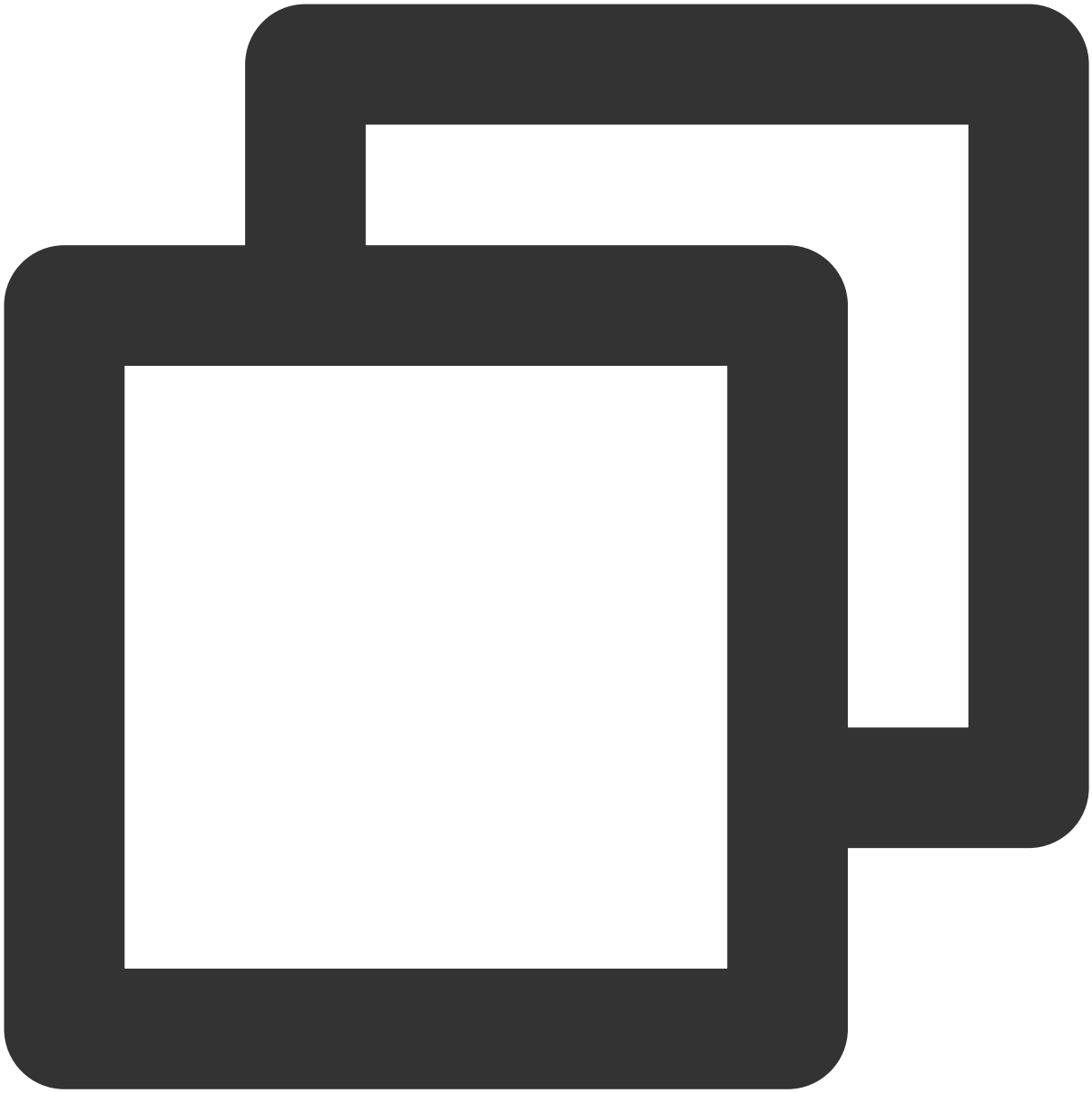
实际案例

请求



```
GET /audio/auditing/vab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>aab1ca9fc8a3ed11ea834c525400863904</JobId>
```

```
    <State>Success</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Object>a.mp3</Object>
    <Result>1</Result>
    <AudioText>wc xx</AudioText>
    <PornInfo>
        <HitFlag>1</HitFlag>
        <Score>91</Score>
        <Label>xx</Label>
    </PornInfo>
    <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Label></Label>
    </AdsInfo>
    <Section>
        <AdsInfo>
            <HitFlag>0</HitFlag>
            <Score>0</Score>
        </AdsInfo>
        <Duration>30000</Duration>
        <OffsetTime>0</OffsetTime>
        <PornInfo>
            <HitFlag>1</HitFlag>
            <Score>91</Score>
            <Keywords>xx</Keywords>
            <Keywords>wc</Keywords>
        </PornInfo>
        <Text>wc xx</Text>
        <Url>https://auditing-1250000000.cos.ap-guangzhou.myqcloud.com/q9MhdztF/0.m
    </Section>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

音频审核回调内容

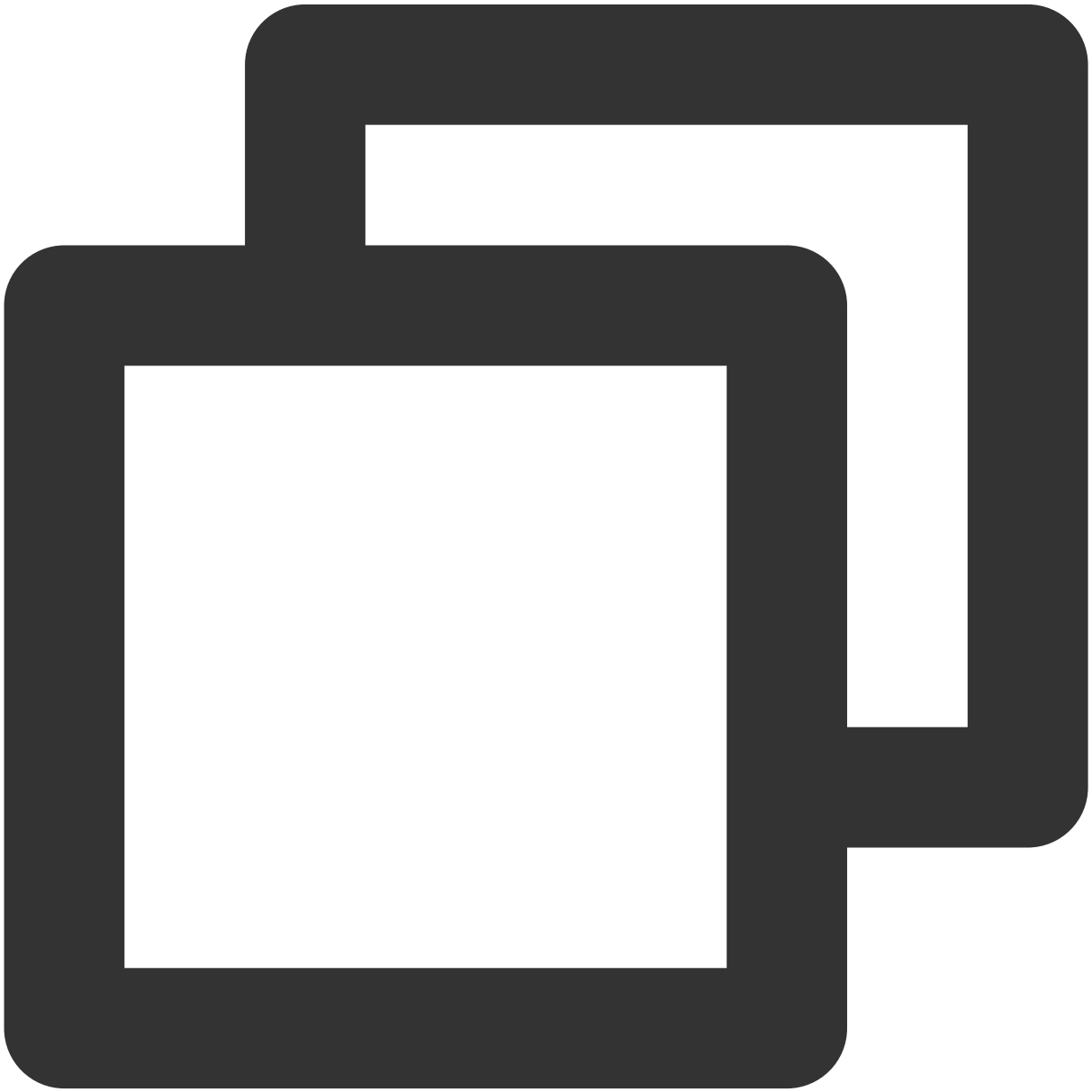
最近更新时间：2023-12-30 10:32:32

功能描述

若您配置了审核回调地址，在音频审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。

回调内容说明

简洁回调（Simple）




```
{
    "code": 0,
    "data": {
        "forbidden_status": 0,
        "porn_info": {
            "hit_flag": 0,
            "label": "",
            "score": 9
        },
        "result": 0,
        "trace_id": "test_trace_id",
```

```
        "url": "test_audio",
        "cos_headers": {
            "x-cos-meta-xx": "xx"
        }
    },
    "message": "Test request when setting callback url"
}
```

具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码。	Integer	是
message	错误描述。	String	是
data	音频审核结果的详细信息。	Object	是

data的具体内容如下：

参数名称	描述	类型	是否必选
trace_id	请求的 ID，如果是接口创建的异步任务，则为接口返回的 JobId。	String	是
url	被审核音频的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewAudio。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示音频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer	是
cos_headers	资源上传时设置的自定义header内容，如果未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否
data_id	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

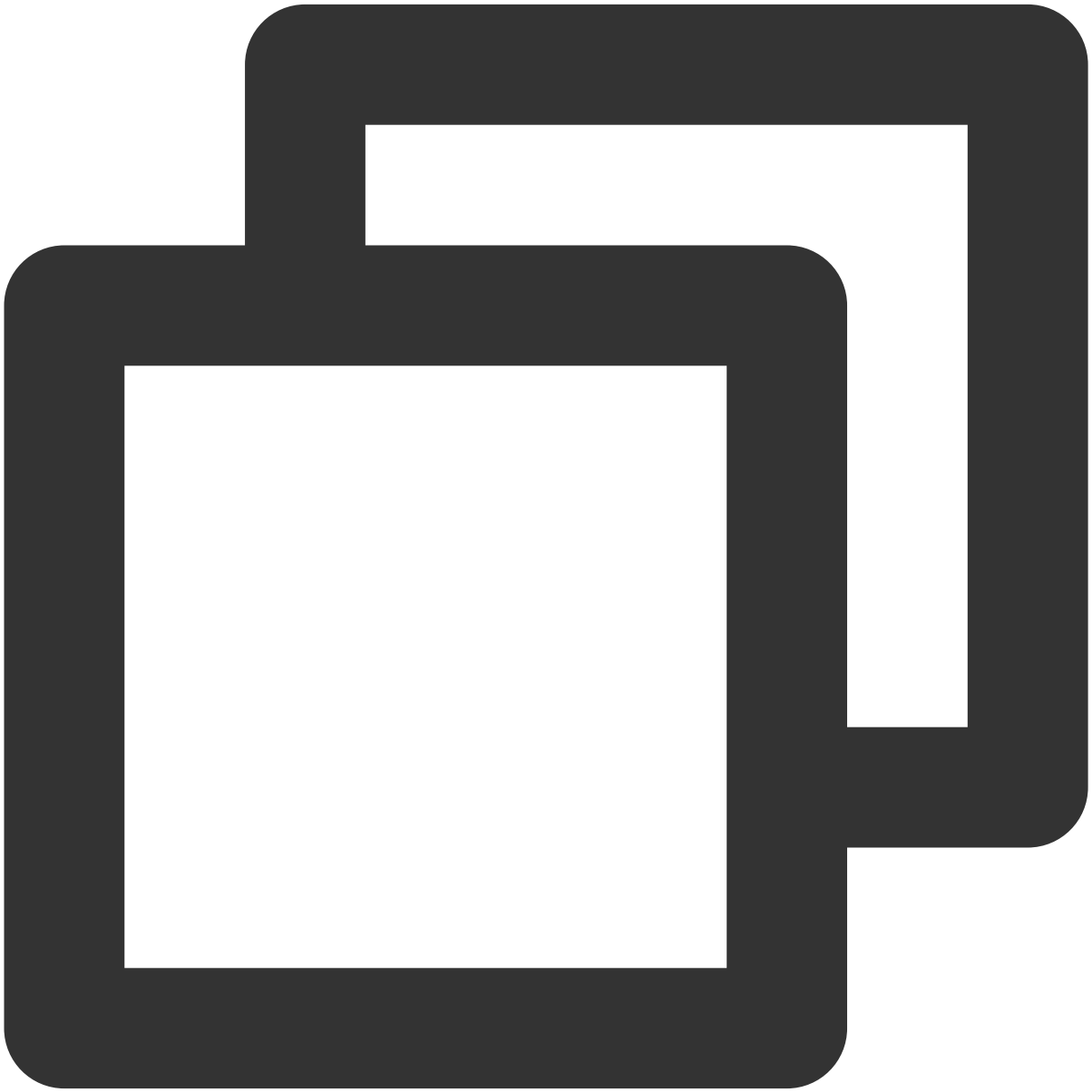
xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值： 0：正常。 1：确认为当前场景的违规内容。 2：疑似为当前场景的违规内容。	Integer	是
score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。 例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer	是
label	本次审核的结果标签，如果命中了敏感的关键词，该字段返回对应的关键词。	String	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。

该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：




```
{
  "EventName": "ReviewAudio",
  "JobsDetail": {
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "",
    "Object": "",
    "Label": "Normal",
    "Result": 0,
    "AudioText": "        ",
    "PornInfo": {
```

```
        "HitFlag": 0,
        "Score": 0,
        "Label": ""
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0,
        "Label": ""
      },
      "Section": [
        {
          "Url": "",
          "Text": "",
          "OffsetTime": 0,
          "Duration": 30000,
          "Label": "Normal",
          "Result": 0,
          "PornInfo": {
            "HitFlag": 0,
            "Score": 0
          },
          "AdsInfo": {
            "HitFlag": 0,
            "Score": 0
          }
        }
      ],
      "BucketId": "",
      "Region": "ap-chongqing",
      "ForbidState": 0,
      "CosHeaders": {
          "x-cos-meta-id": "xxxx"
      }
    }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息	Object
EventName	任务的类型，为 `ReviewAudio`	String

JobsDetail 的具体内容如下：

节点名称（关	描述	类型

键字）		
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	本次音频审核任务的 ID。	String
DataId	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String
State	音频审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	音频审核任务的创建时间。	String
Object	本次审核的音频文件名称，创建任务使用 Object 时返回。	String
Url	本次审核的音频文件链接，创建任务使用 Url 时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
AudioText	该字段用于返回音频文件识别出的对应文本内容。	String
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
Section	当音频过长时，会对音频进行分段，该字段用于返回音频片段的审核结果，主要包括开始时间和音频审核的相应结果。	Array
BucketId	创建的音频审核任务所属的存储桶名称	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示音频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段，创建任务未设置 UserInfo 时无此字段。	Object

版权所有：腾讯云计算（北京）有限责任公司

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值： 0：正常。 1：确认为当前场景的违规内容。 2：疑似为当前场景的违规内容。	Integer
Score	本次审核的结果分数，分数越高表示越敏感。	Integer
Label	本次审核的结果标签，如果命中了敏感的关键词，该字段返回对应的关键词。	String

Section 的具体内容如下：

节点名称（关键字）	描述	类型
Url	当前音频片段的访问地址，您可以通过该地址获取该音频片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
Text	该字段用于返回当前音频片段的 ASR 文本识别的检测结果。	String
OffsetTime	该字段用于返回当前音频片段位于总音频中的时间，单位为毫秒，例如5000（音频开始后5000毫秒）。	Integer
Duration	当前音频片段的时长，单位毫秒。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Section 中的 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键	描述	类型

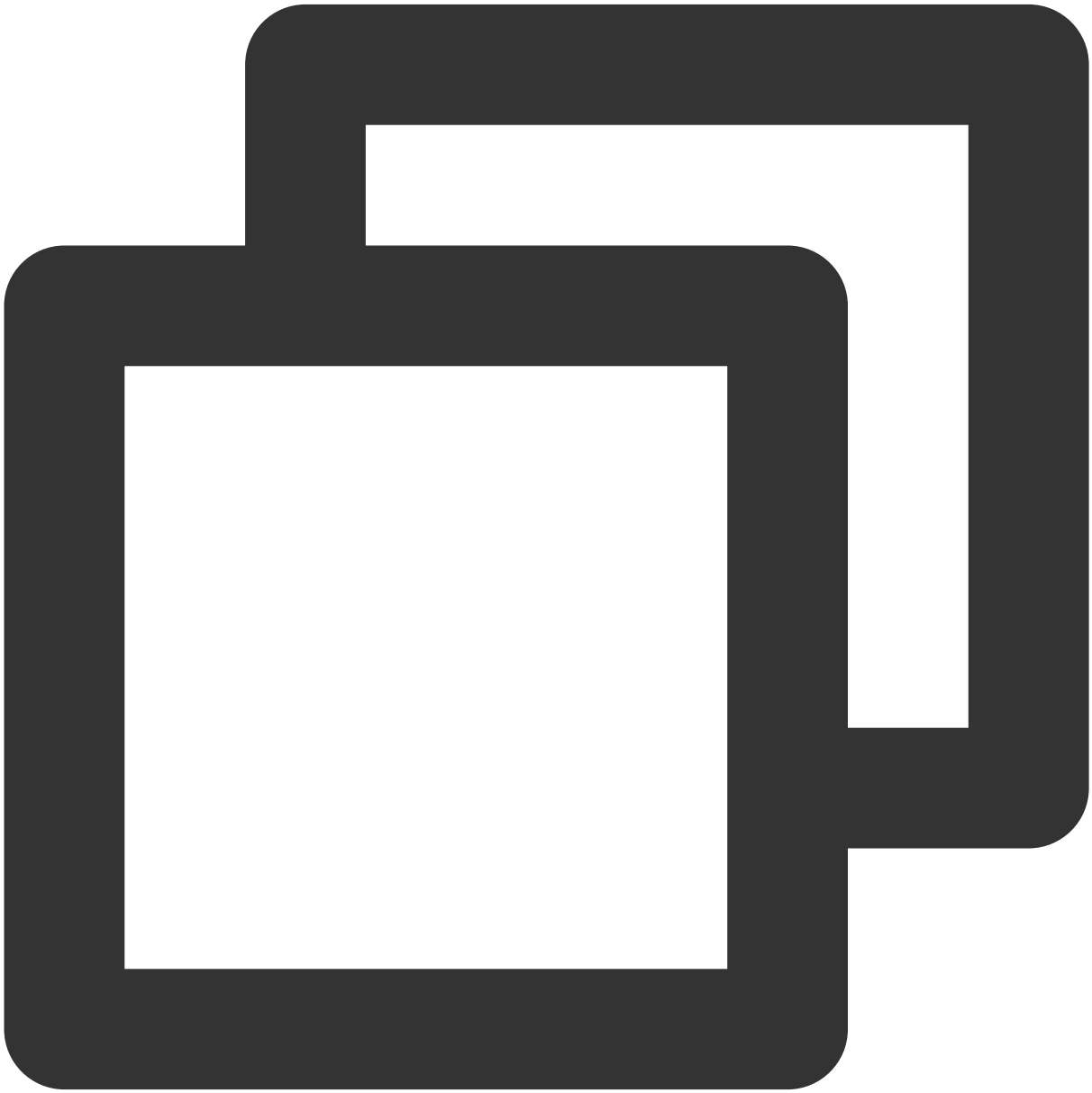
字）		
HitFlag	用于返回该审核场景的审核结果，返回值： 0：正常。 1：确认为当前场景的违规内容。 2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息 例如：色情99，则表明该内容非常有可能属于色情内容。	Integer
Category	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
Keywords	本次审核命中的敏感关键词，没有时不返回。	Array

UserInfo 的内容：

节点名称（关键字）	描述	类型
TokenId	用户业务 TokenId，长度不超过128字节。	String
Nickname	用户业务 Nickname，长度不超过128字节。	String
DeviceId	用户业务 DeviceId，长度不超过128字节。	String
AppId	用户业务 AppId，长度不超过128字节。	String
Room	用户业务 Room，长度不超过128字节。	String
IP	用户业务 IP，长度不超过128字节。	String
Type	用户业务 Type，长度不超过128字节。	String

实际案例

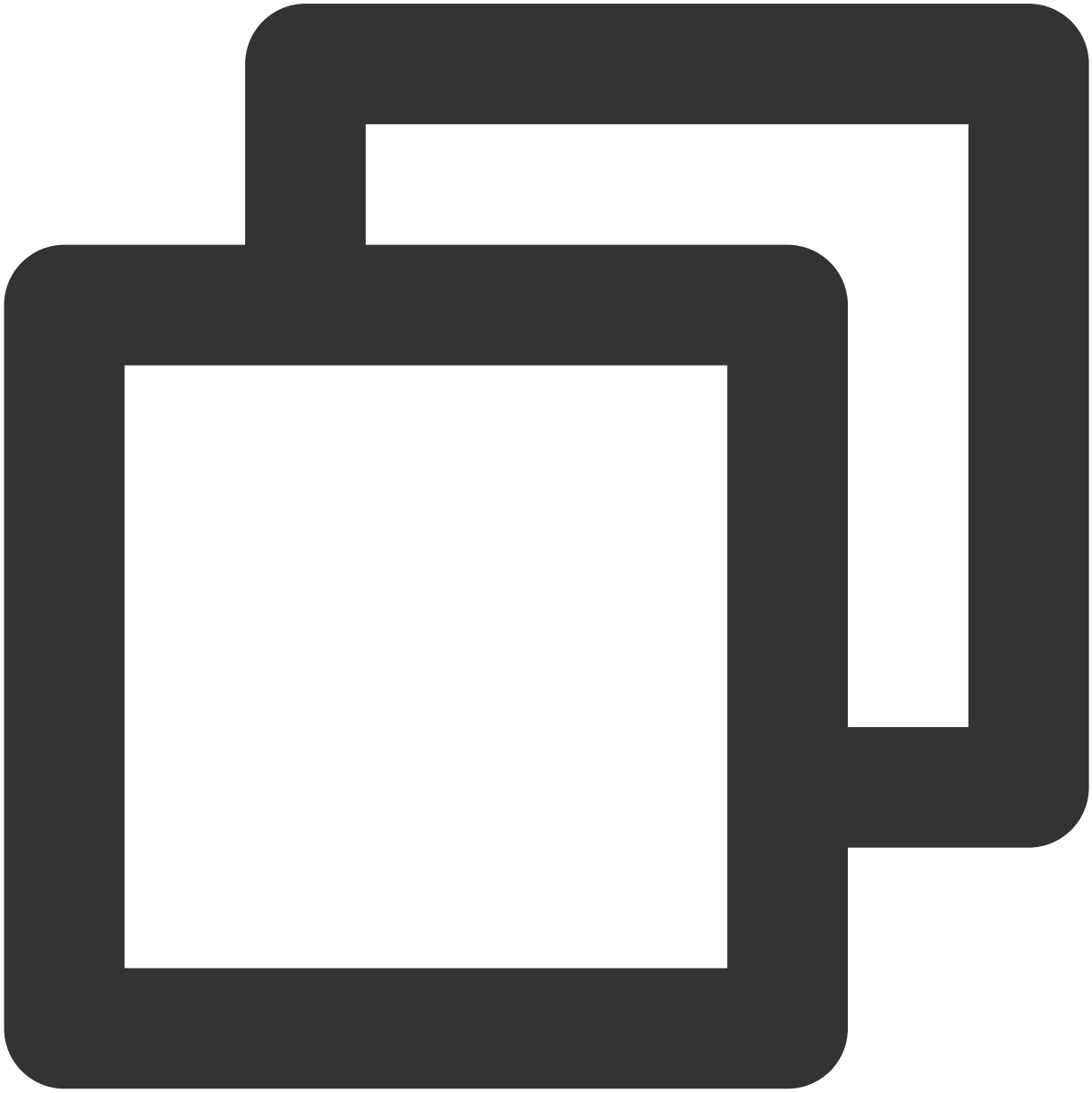
案例一：简洁回调（Simple）




```
{
    "code": 0,
    "data": {
        "event":"ReviewAudio",
        "forbidden_status": 0,
        "porn_info": {
            "hit_flag": 0,
            "label": "",
            "score": 9
        },
        "result": 0,
```

```
        "trace_id": "ixzt90jl2dfscxxxxxxxxxxxxxxxx",
        "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/music.
        "cos_headers": {
            "x-cos-meta-id": "666666"
        }
    },
    "message": "success"
}
```

案例二：详细回调（Detail）



```
{
  "EventName": "ReviewAudio",
  "JobsDetail": {
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "2021-08-10T21:01:08+08:00",
    "Object": "1.mp3",
    "Result": 0,
    "AudioText": "        ",
    "PornInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": ""
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": ""
    },
    "Section": [
      {
        "Url": "https://audio-1250000000.cos.ap-guangzhou.myqcloud.com/0.mp3",
        "Text": "",
        "OffsetTime": 0,
        "Duration": 30000,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      }
    ],
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
        "x-cos-meta-id": "xxxx"
    }
  }
}
```

文本审核
文本内容同步审核

最近更新时间：2023-12-28 15:10:16

功能描述

本接口用于进行文本内容的直接审核，请求方式为同步请求。
该接口的支持情况如下：
说明
直接审核纯文本信息，需要先经过 base64 编码（仅支持 UTF8 编码）。
支持识别多种违规场景，包括：色情、违法、广告等场景。
支持根据不同的业务场景 配置自定义的审核策略。

授权说明

授权策略中 action 设置为 `ci:CreateAuditingTextJob` 。查看所有 action。

服务开通

使用该功能需提前开通数据万象，并绑定存储桶。详情请参见 绑定存储桶。

使用限制

使用该接口时，请先确认相关限制条件。详情请参见 使用限制。

费用说明

每个审核场景单独计费，例如您选择审核色情、广告两种场景，则审核**1条文本**，**计2次**审核费用。
长度限制：当审核的内容为纯文本信息，需要先经过 base64 编码，文本编码前的原文长度不能超过10000个 UTF8 编码字符。
文本审核语言支持：目前支持中文、中英混合、阿拉伯数字的检测。

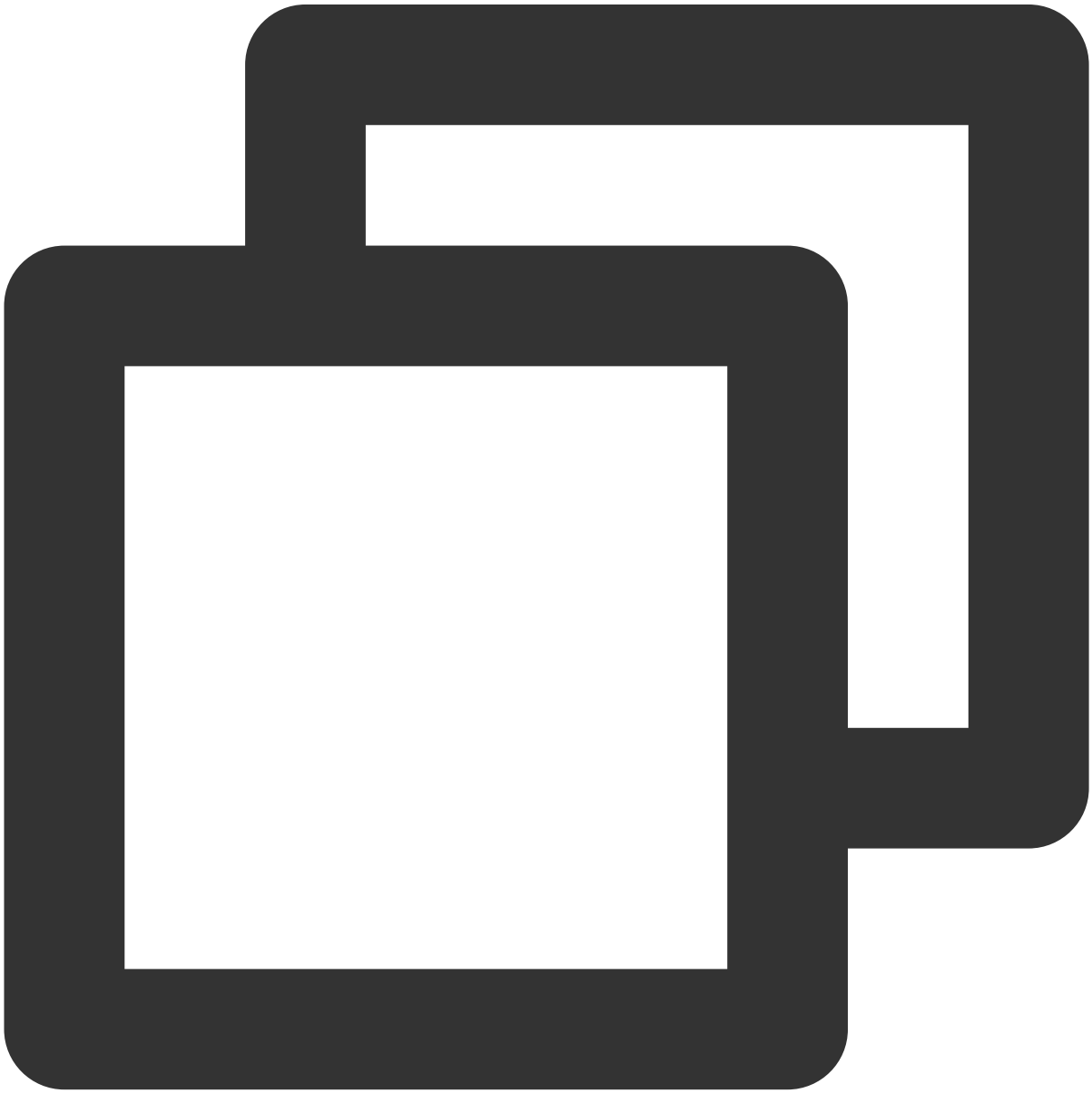
版权所有：腾讯云计算（北京）有限责任公司

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
POST /text/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

Authorization: Auth String（详情请参见 请求签名 文档）。

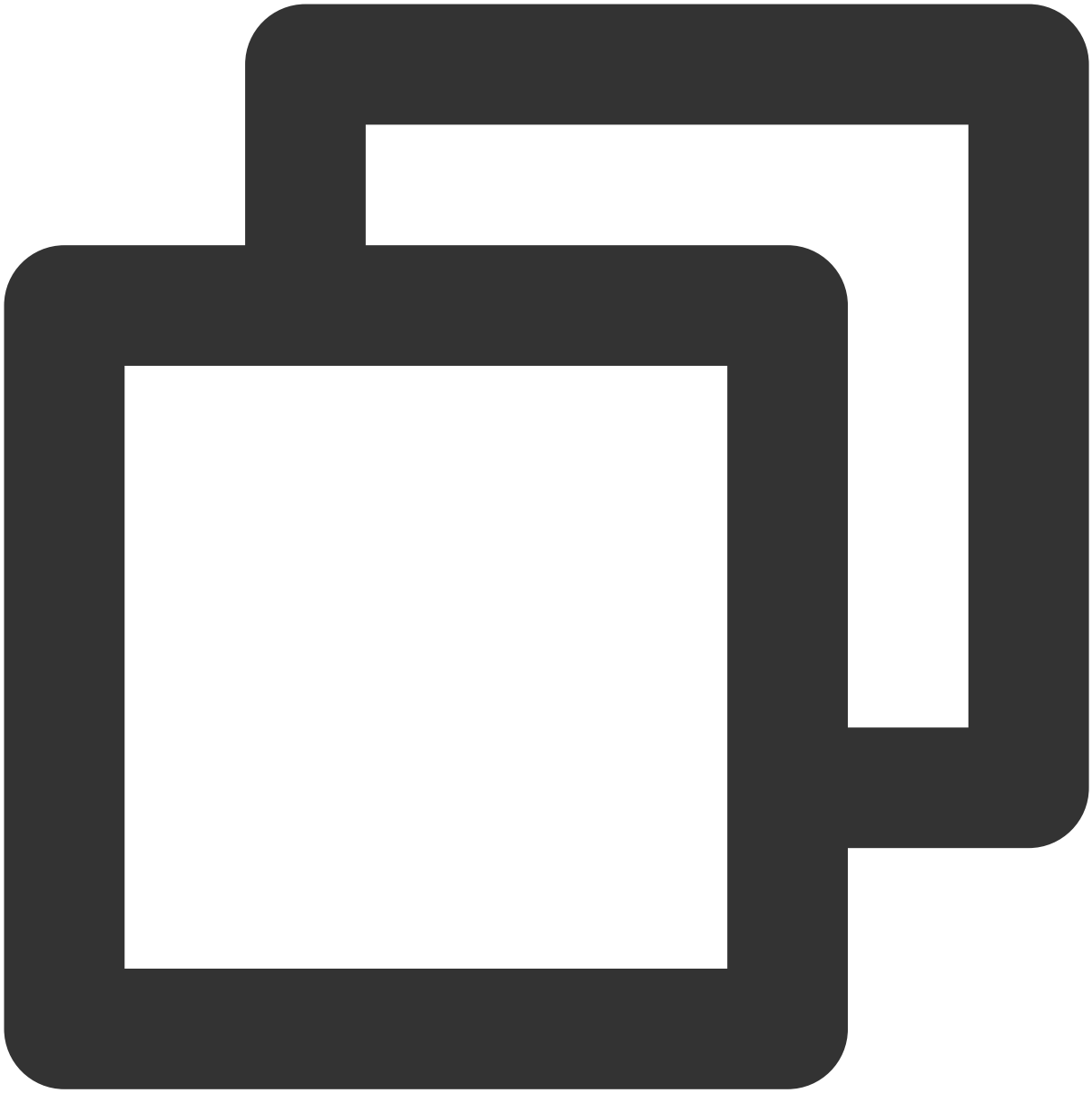
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
  <Input>
    <Content></Content>
    <DataId></DataId>
  </Input>
  <Conf>
    <Callback></Callback>
    <CallbackVersion></CallbackVersion>
    <CallbackType></CallbackType>
    <BizType></BizType>
    <Freeze>
```

```
      <PornScore></PornScore>
    </Freeze>
  </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	文本审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容。	Container	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Content	Request.Input	当传入的内容为纯文本信息，需要先经过 base64 编码，文本编码前的原文长度不能超过 10000 个 UTF8 编码字符。若超出长度限制，接口将会报错。 注：目前支持中文、中英混合、阿拉伯数字的检测与审核。如有其他语言需要，请 联系我们。	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如色情、广告、违法等，配置指引：设置公共审核策略。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Callback	Request.Conf	审核结果可以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。当 Input 使用 Content 时，该参数不生效，结果会直接返回。	String	否
CallbackVersion	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	String	否
CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部文本片段）、2（回调违规文本片段）。默认为 1。	Integer	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对文本文件进行自动冻结，仅当 `input` 中审核的文本为 `object` 时有效。	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
IllegalScore	Request.Conf.Freeze	取值为[0,100]，表示当违法审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AbuseScore	Request.Conf.Freeze	取值为[0,100]，表示当谩骂审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请 联系服务人员 咨询。

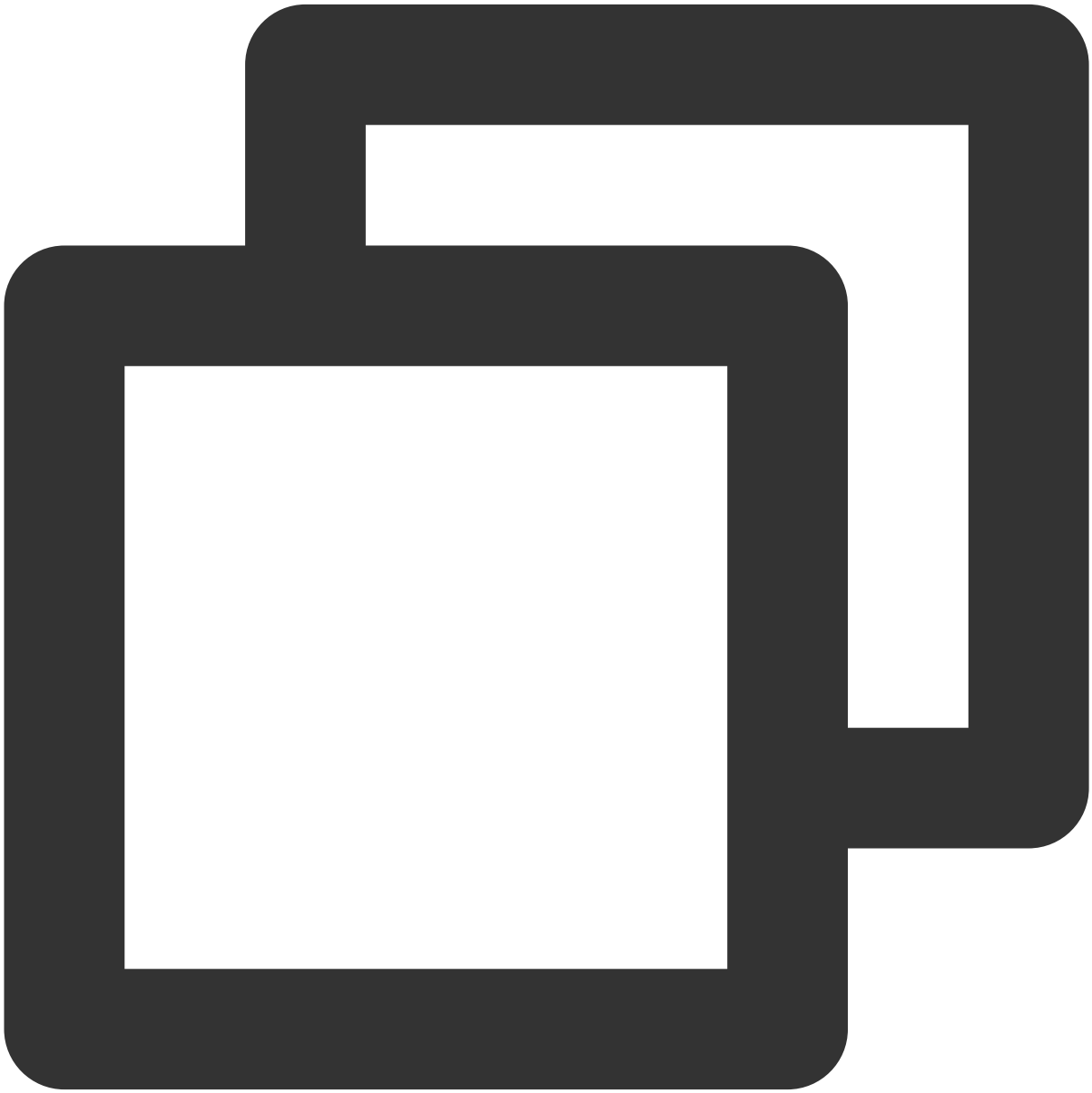
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Code>Success</Code>
    <Message>Success</Message>
    <SectionCount></SectionCount>
    <Result>1</Result>
    <PornInfo>
```

```
        <HitFlag></HitFlag>
        <Count></Count>
      </PornInfo>
      <Section>
        <StartByte></StartByte>
        <PornInfo>
          <HitFlag></HitFlag>
          <Score></Score>
          <Keywords></Keywords>
        </PornInfo>
      </Section>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	文本审核返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	文本审核的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回，详情请查看 错误码列表。	String
DataId	Response.JobsDetail	请求中添加的唯一业务标识。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
JobId	Response.JobsDetail	本次文本审核的任务ID。	String
CreationTime	Response.JobsDetail	文本审核的创建时间。	String
State	Response.JobsDetail	文本审核的状态，值为 Success（审核成功）或者	String

		Failed（审核失败）。	
Content	Response.JobsDetail	文本内容的 Base64 编码。	String
ContextText	Response.JobsDetail	当启用了文本上下文关联审核能力时，会返回该字段，同时返回当前审核的文本内容及关联的文本内容，以原文形式返回。 注意：使用上下文关联能力，在发起文本审核时需要结合UserInfo字段来使用，表示关联的上下文仅属于某个用户ID，不同用户ID上传的文本内容不会互相关联。 如需开启上下文关联能力，请 联系服务人员 开通。	String
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
SectionCount	Response.JobsDetail	审核的文本内容分片数，固定为1。	Integer
PornInfo	Response.JobsDetail	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail	审核场景为**广告**的审核结果信息。	Container
IllegalInfo	Response.JobsDetail	审核场景为**违法**的审核结果信息。	Container
AbuseInfo	Response.JobsDetail	审核场景为**谩骂**的审核结果信息。	Container
Section	Response.JobsDetail	文本审核的具体结果信息。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container

Container 节点 PornInfo，AdsInfo，IllegalInfo，AbuseInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.*Info	用于返回该审核场景的审核结果，返回值：0：正	Integer

		常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	
Count	Response.JobsDetail.*Info	命中该审核分类的分片数。	Integer
Score	Response.JobsDetail.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer

Container 节点 Section 的内容：

节点名称（关键字）	父节点	描述	类型
StartByte	Response.JobsDetail.Section	该分片位于文本中的起始位置信息（即10代表第11个utf8文字）。从0开始。	Integer
Label	Response.JobsDetail.Section	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail.Section	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.Section	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.Section	审核场景为**广告**的审核结果信息。	Container
IllegalInfo	Response.JobsDetail.Section	审核场景为**违法**的审核结果信息。	Container
AbuseInfo	Response.JobsDetail.Section	审核场景为**谩骂**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo，IllegalInfo，AbuseInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Section.*Info	用于返回该审核场景的审核结果，返回	Integer

版权所有：腾讯云计算（北京）有限责任公司

		值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	
Score	Response.JobsDetail.Section.*Info	该分片中审核的结果分数，分数越高表示越敏感。	Integer
Keywords	Response.JobsDetail.Section.*Info	在当前审核场景下命中的关键词，多关键词以 `,` 分隔。	String
LibResults	Response.JobsDetail.Section.*Info	该字段用于返回基于文本风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array
SubLabel	Response.JobsDetail.Section.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签	String

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
LibType	Response.JobsDetail.Section.*Info.LibResults	命中的文本风险库类型，取值为1（预设文本库）和2（自定义文本风险库）。	Integer
LibName	Response.JobsDetail.Section.*Info.LibResults	命中的风险库名称。	String
Keywords	Response.JobsDetail.Section.*Info.LibResults	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ListResults	Response.JobsDetail.ListInfo	命中的所有业务字段风险库结果。	Container Array

Container 节点 ListResults 的内容：

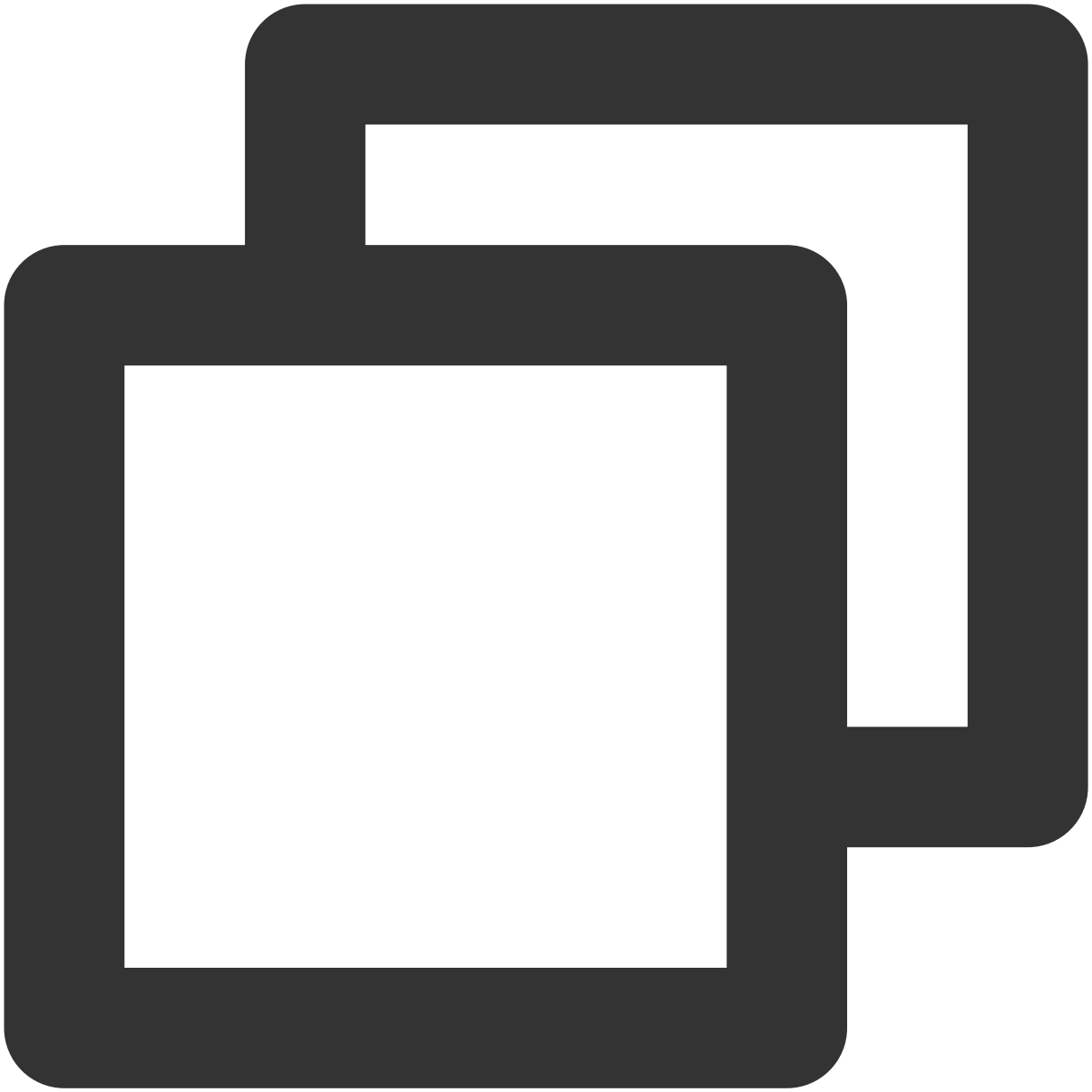
节点名称（关键字）	父节点	描述	类型
ListType	Response.JobsDetail.ListInfo.ListResults	命中的业务字段风险库类型，取值为0（白库）和1（黑库）。	Integer
ListName	Response.JobsDetail.ListInfo.ListResults	命中的业务字段风险库名称。	String
Entity	Response.JobsDetail.ListInfo.ListResults	命中了业务字段风险库中的哪条内容。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

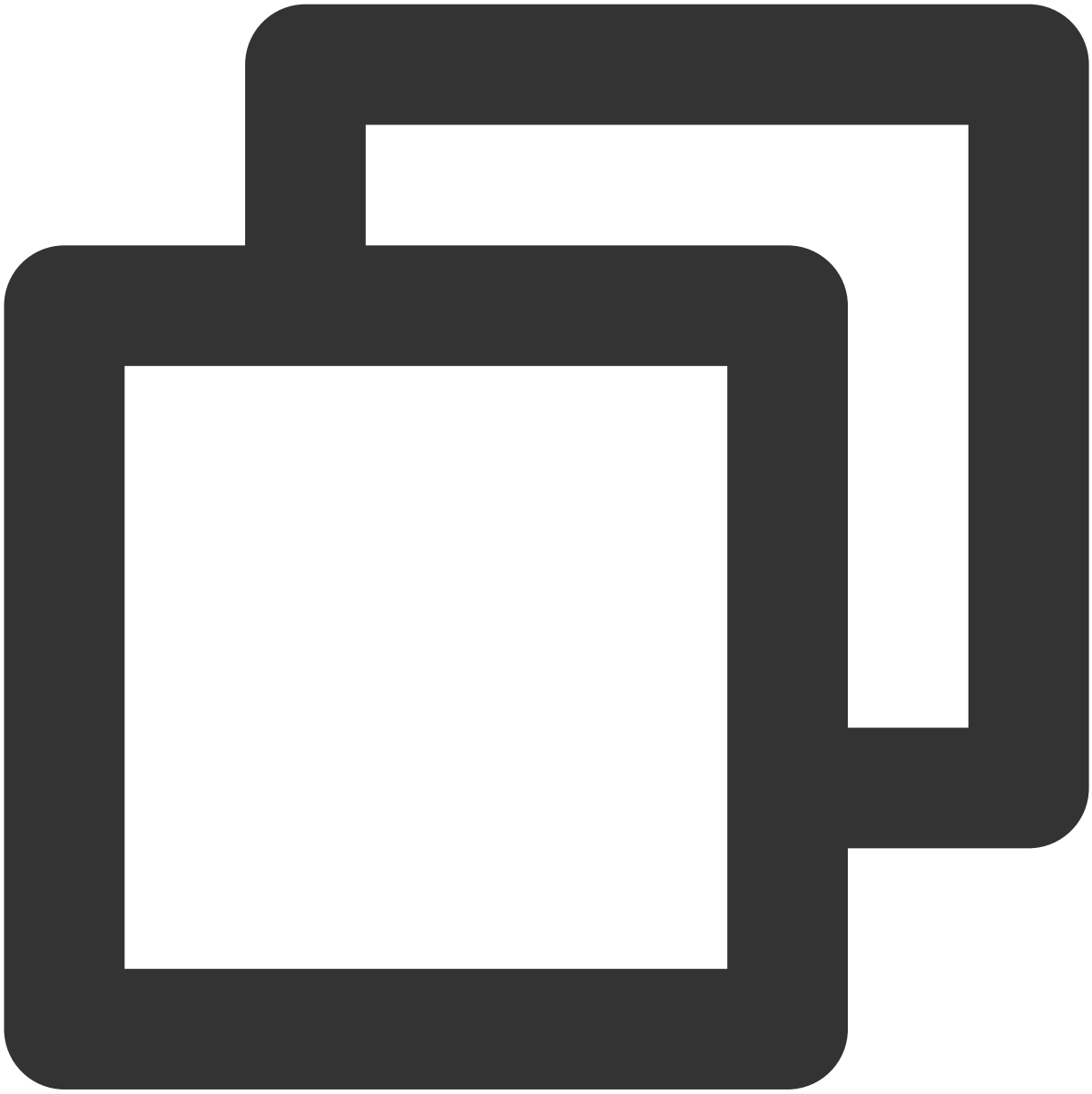
请求



```
POST /text/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml
<Request>
  <Input>
    <Content>54uZ5Ye75omL</Content>
  </Input>
  <Conf>
    <BizType>b81d45f94b91a683255e9a9506f45a11</BizType>
```

```
  </Conf>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
```

```
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
<Response>
  <JobsDetail>
    <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
    <Content>54uZ5Ye75omL</Content>
    <State>Success</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <SectionCount>1</SectionCount>
    <Label>Illegal</Label>
    <Result>2</Result>
    <PornInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </PornInfo>
    <Section>
      <StartByte>0</StartByte>
      <Label>Illegal</Label>
      <Result>2</Result>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Keywords/>
      </PornInfo>
    </Section>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

文档审核
提交文档审核任务

最近更新时间：2023-12-30 10:34:09

功能描述

本接口用于提交一个文档审核任务。文档审核功能为异步任务方式，您可以通过提交文档审核任务审核您的文档文件，然后通过 查询文档审核任务接口 或 文档审核回调 查询审核结果。

说明：

支持审核腾讯云 COS 上的文档文件。

支持审核第三方云存储厂商的文档 URL。

目前支持的输入文件类型包含如下格式：

演示文件：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。

文字文件：doc、dot、wps、wpt、docx、dotx、docm、dotm。

表格文件：xls、xlt、et、ett、xlsx、xltx、csv、xlsb、xlsm、xltm、ets。

其他格式文件：pdf。

表格文件，一张表可能分割为多页转换，生成多张图片。

输入文件大小限制在200MB之内。

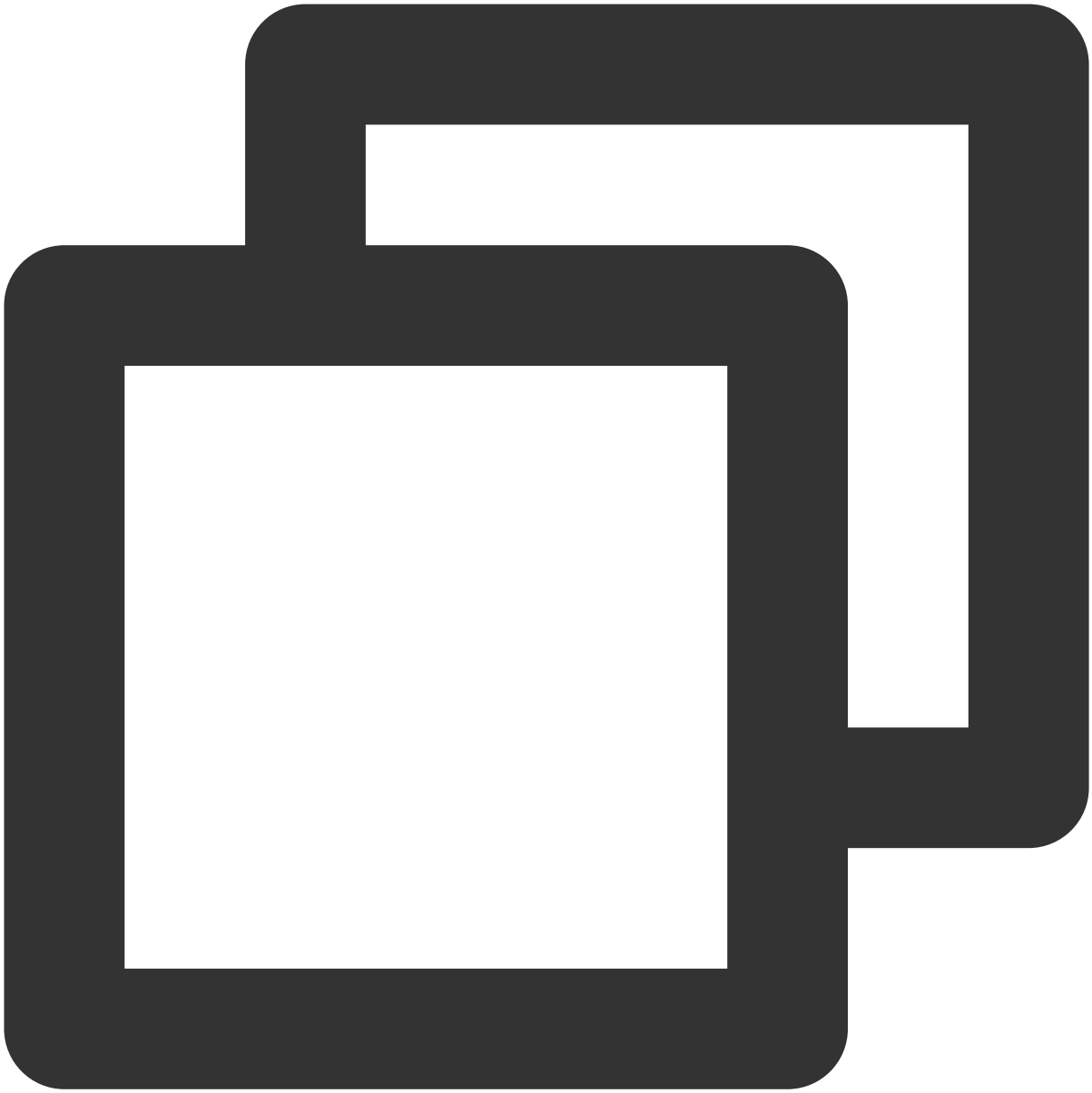
输入文件页数限制在5000页之内。

支持根据不同的业务场景配置自定义的审核策略。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
POST /document/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明：

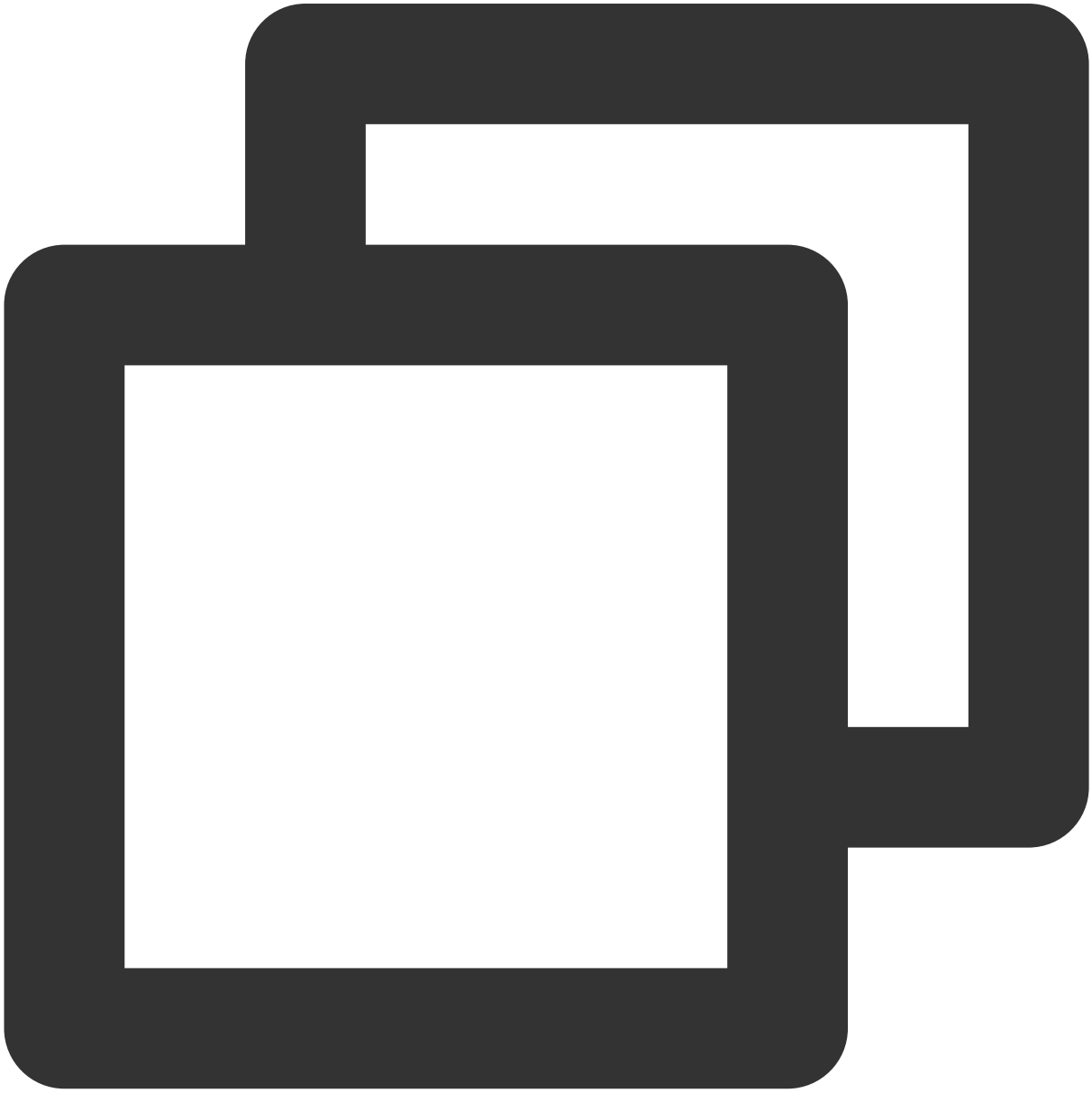
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。



```
<Request>
      <Input>
```

```
            <Url></Url>
            <Type></Type>
            <DataId></DataId>
        </Input>
        <Conf>
            <DetectType></DetectType>
            <Callback></Callback>
            <BizType></BizType>
        </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	文档审核的具体配置项	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容	Container	是
Conf	Request	审核规则配置	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的文件名称，例如在目录 test 中的文件test.doc，则文件名称为 test/test.doc。 Object 和 Url 只能选择其中一种。	String	否
Url	Request.Input	文档文件的链接地址，例如 http://www.example.com/doctest.doc，Object 和 Url 只能选择其中一种。	String	否
Type	Request.Input	指定文档文件的类型，如未指定则默认以文件的后缀为类型。 如果文件没有后缀，该字段必须指定，否则会审核失败。例如：doc、docx、ppt、pptx 等	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为	String	否

版权所有：腾讯云计算（北京）有限责任公司

		512字节。您可以使用该字段对待审核的数据进行唯一业务标识。		
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	用户业务 TokenId，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	用户业务 Nickname，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	用户业务 DeviceId，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	用户业务 AppId，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	用户业务 Room，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	用户业务 IP，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	用户业务 Type，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如涉黄、广告、违法违规等，配置指引：设置公共审核策略。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。 BizType 不填写时，将自动使用默认的审核策略。	String	否
DetectType	Request.Conf	（后续不再维护）审核的场景类型，有效值：Porn（涉黄）、Ads（广告）等，可以传入多种类型，不同类型以逗号分隔，例如：Porn,Ads。如您有更多场景的审核需要，请使用 BizType 参数。	String	否

Callback	Request.Conf	审核结果可以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com`	String	否

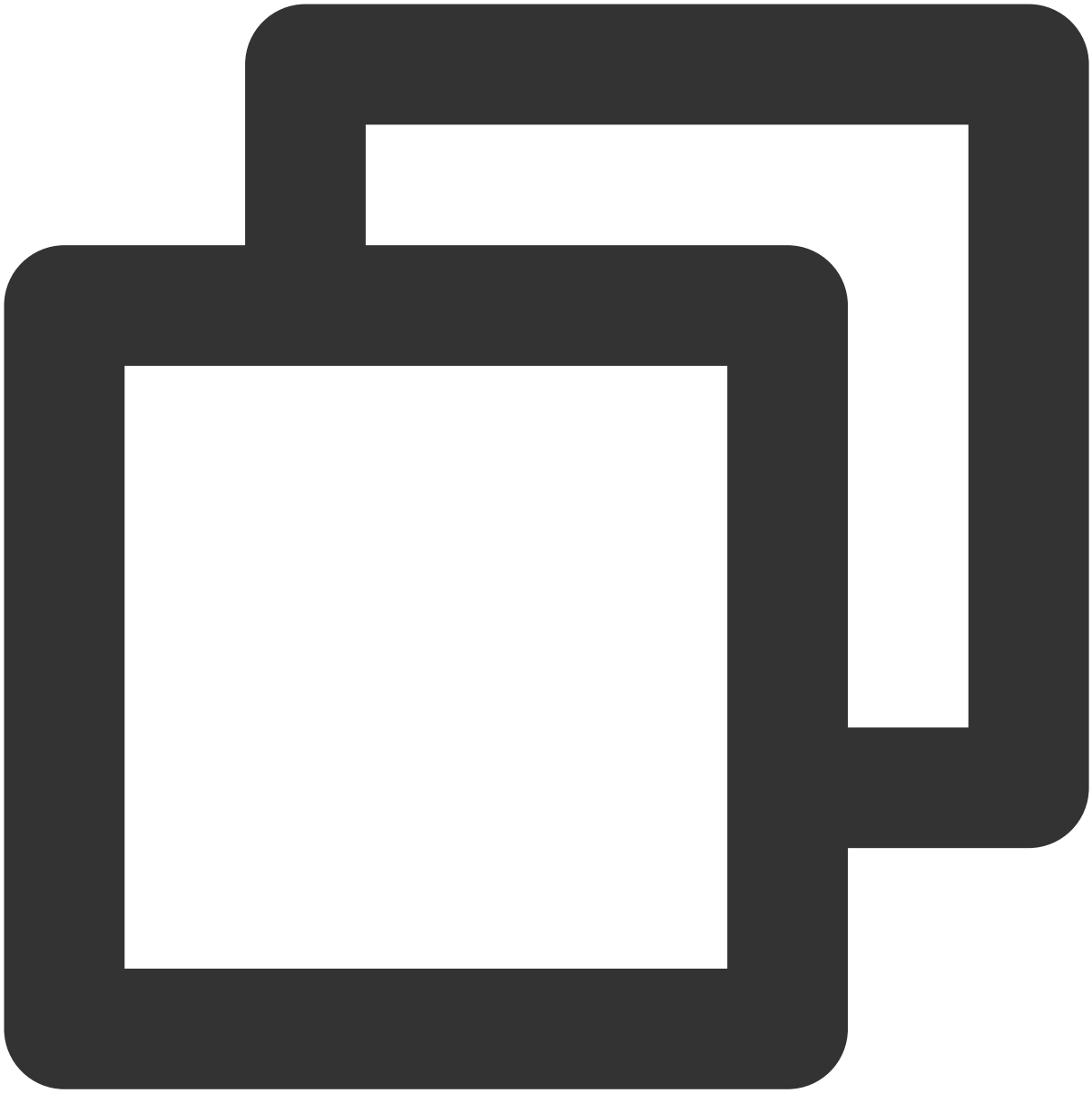
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Request>
    <JobsDetail>
       <DataId></DataId>
       <JobId></JobId>
       <State></State>
       <CreationTime></CreationTime>
    </JobsDetail>
    <RequestId></RequestId>
</Request>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	提交文档审核任务返回的具体响应内容	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	文档审核任务的详细信息	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题	String

Container 节点 JobsDetail 的内容：

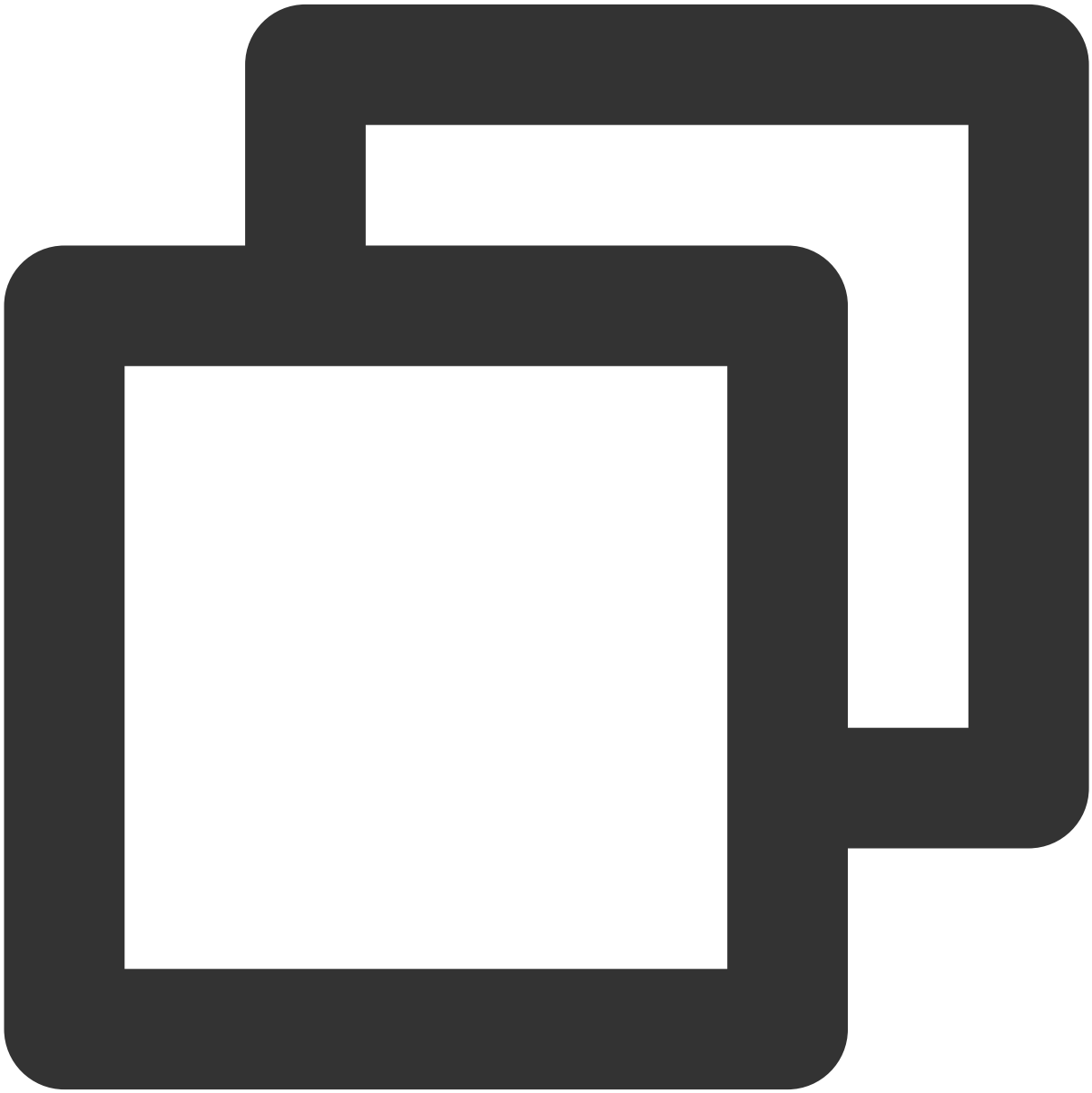
节点名称（关键字）	父节点	描述	类型
DataId	Response.JobsDetail	请求中添加的唯一业务标识	String
JobId	Response.JobsDetail	本次文档审核任务的 ID	String
State	Response.JobsDetail	文档审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个	String
CreationTime	Response.JobsDetail	文档审核任务的创建时间	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

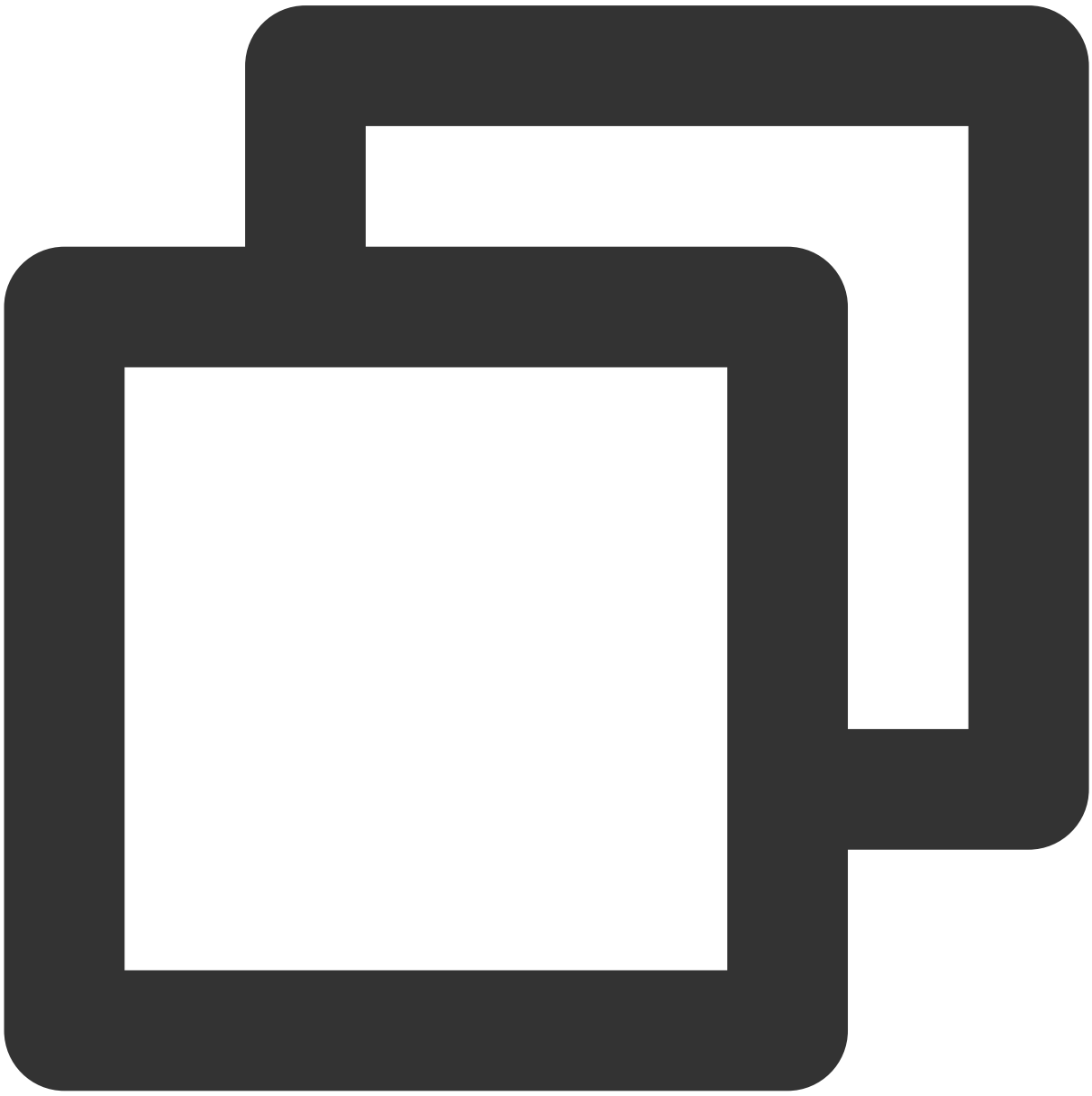
请求



```
POST /document/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml
<Request>
      <Input>
         <Url>http://www.example.com/doctest.doc</Url>
         <DataId>123-fdrsg-123</DataID>
      </Input>
      <Conf>
```

```
        <DetectType>Porn,Ads</DetectType>
        <Callback></Callback>
        <BizType>b81d45f94b91a683255e9a9506f4****</BizType>
    </Conf>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
```

```
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <DataId>123-fdrsg-123</DataID>
        <JobId>vab1ca9fc8a3ed11ea834c52540086****</JobId>
        <State>Submitted</State>
        <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    </JobsDetail>
    <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

查询文档审核任务结果

最近更新时间：2023-12-30 10:34:14

功能描述

本接口用于查询指定的文档审核任务结果。

说明：

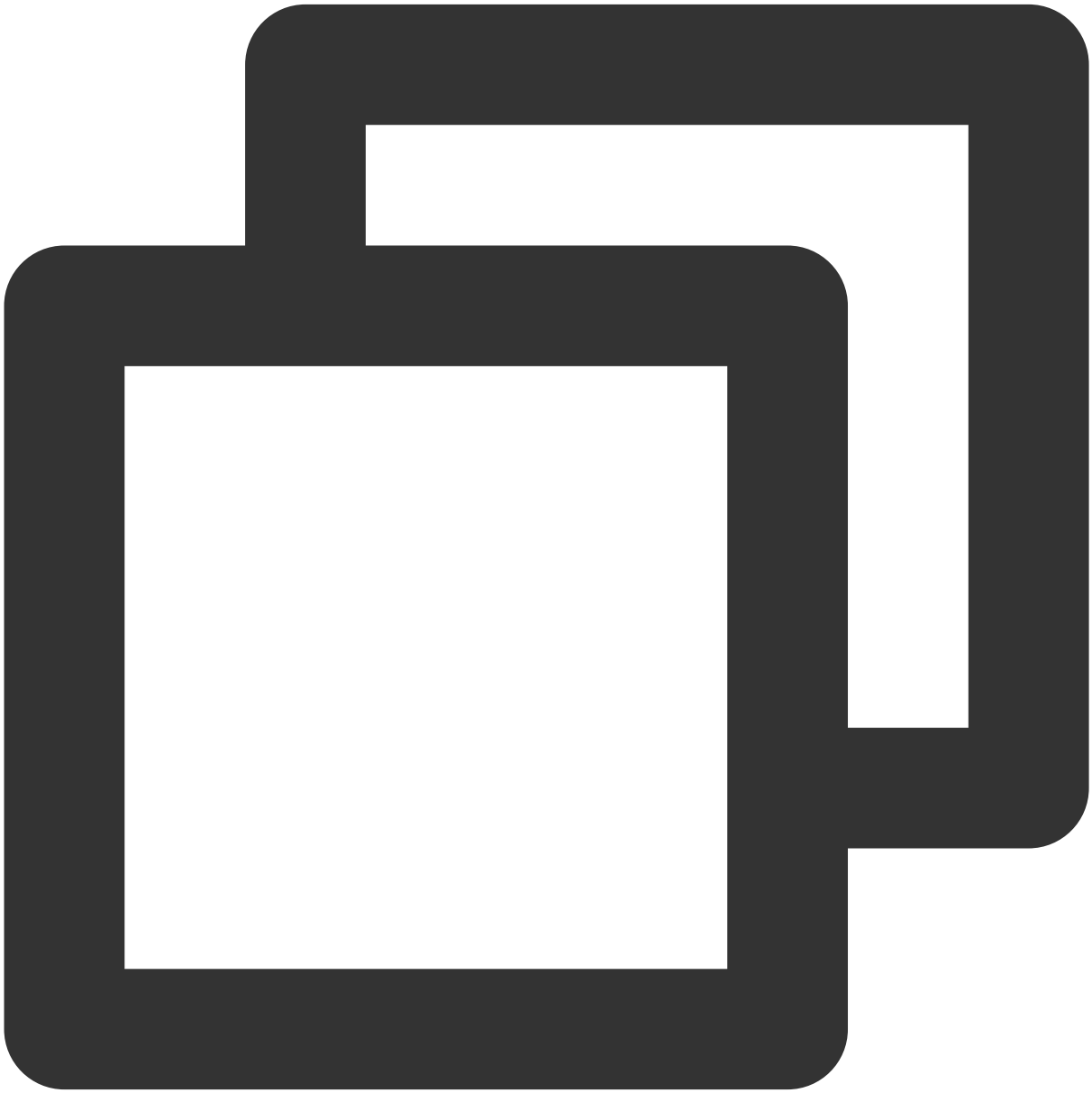
文档审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例



```
GET /document/auditing/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明：

Authorization: Auth String（详情请参见 请求签名 文档）。

通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

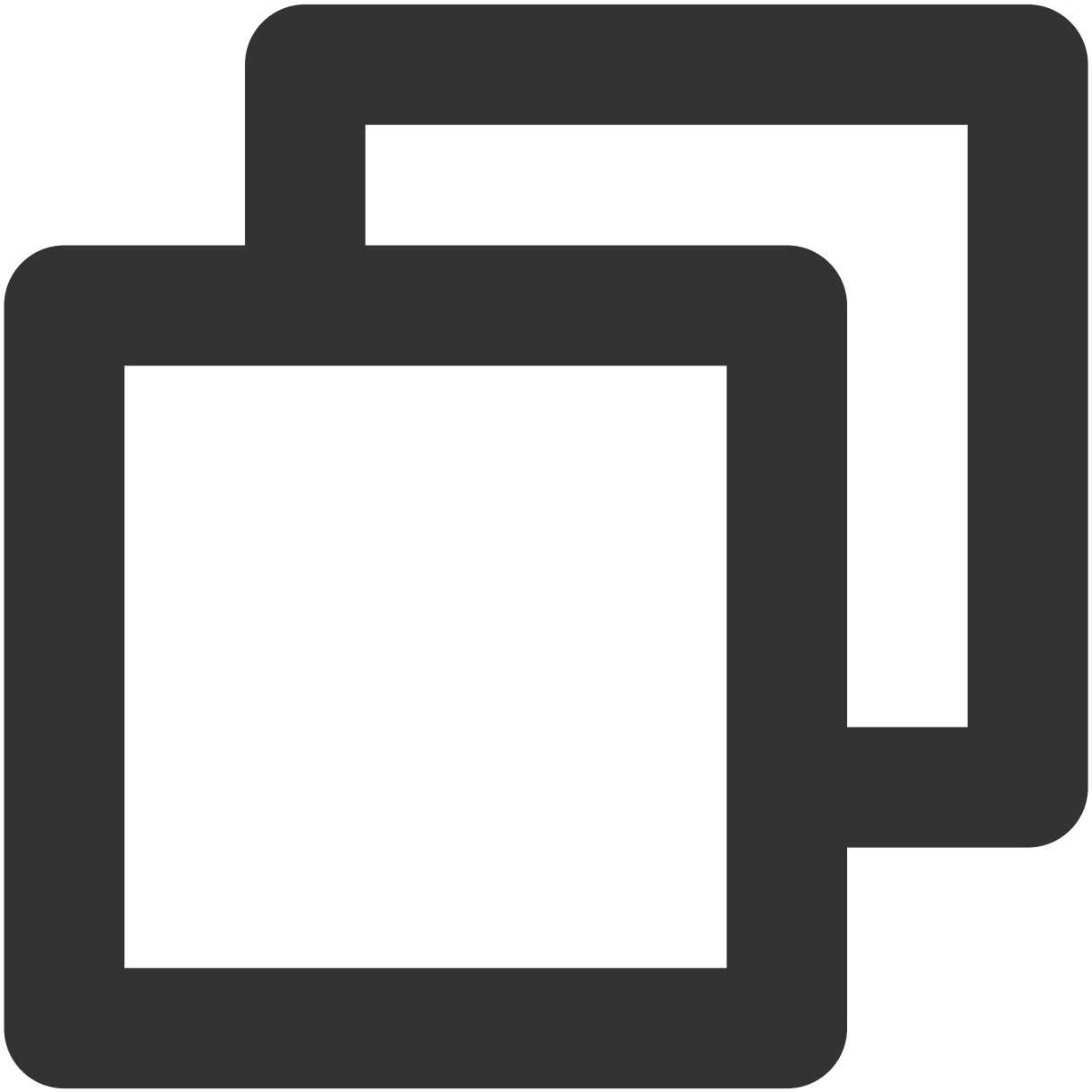
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
    <JobsDetail>
      <JobId>aab1ca9fc8a3ed11ea834c525400863904</JobId>
      <State>Success</State>
      <Code>Success</Code>
      <Message>Success</Message>
      <DataId></DataId>
      <Suggestion>1</Suggestion>
      <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
      <Url>http://www.test.com/123</Url>
      <PageCount>3</PageCount>
```

```
      <Labels>
        <PornInfo>
          <HitFlag>1</HitFlag>
          <Score>96</Score>
        </PornInfo>
      </Labels>
      <PageSegment>
        <Results>
          <Url></Url>
          <Text></Text>
          <PageNumber>1</PageNumber>
          <SheetNumber>1</SheetNumber>
          <PornInfo>
            <HitFlag></HitFlag>
            <SubLabel></SubLabel>
            <Score></Score>
            <OcrResults>
              <Text></Text>
              <Keywords></Keywords>
            </OcrResults>
          </PornInfo>
        </Results>
      </PageSegment>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	文档审核的全部信息	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	文档审核任务的详细信息	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型

Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回，详情请查看 错误码列表	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回	String
DataId	Response.JobsDetail	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	文档审核任务的 ID	String
State	Response.JobsDetail	任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个	String
Suggestion	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）	Integer
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
CreationTime	Response.JobsDetail	任务的创建时间	String
Object	Response.JobsDetail	审核的文件为存储在 COS 中的文件时，该字段表示本次审核的文件名称	String
Url	Response.JobsDetail	审核的文件链接，创建任务使用 Url 时返回	String
PageCount	Response.JobsDetail	文档审核会将文档转换为图片进行审核，该字段表示转换的图片总数量	Integer
Labels	Response.JobsDetail	该字段用于返回命中的审核场景及对应的结果	Container
PageSegment	Response.JobsDetail	文档转换为图片后，具体每张图片的审核结果信息	Container
UserInfo	Response.JobsDetail	用户业务字段。创建任务时未设置 UserInfo 则无此字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container
ForbidState	Response.JobsDetail	若您设置了自动冻结，该字段表示文档的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 Labels 的内容：

节点名称（关键字）	父节点	描述	类型
PornInfo	Response.JobsDetail.Labels	审核场景为**色情**的审核结果信息	Container
AdsInfo	Response.JobsDetail.Labels	审核场景为**广告**的审核结果信息	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Labels.*Info	用于返回该审核场景的审核结果，返回值： 0：正常 1：确认为当前场景的违规内容 2：疑似为当前场景的违规内容	Integer
Score	Response.JobsDetail.Labels.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息 例如：色情 99，则表明该内容非常有可能属于色情内容	Integer

Container 节点 PageSegment 的内容：

节点名称（关键字）	父节点	描述	类型
Results	Response.JobsDetail.PageSegment	文档转换为图片后，每张图片的详细审核结果信息	Container Array

Container 节点 Results 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.PageSegment.Results	文档转换成图片后，您可以通过该地址查看该图片内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String

版权所有：腾讯云计算（北京）有限责任公司

Text	Response.JobsDetail.PageSegment.Results	该字段用于返回当前图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
PageNumber	Response.JobsDetail.PageSegment.Results	该图片的页码，通常是文档的页码	Integer
SheetNumber	Response.JobsDetail.PageSegment.Results	如果审核的为表格文件，该字段表示表格内的 Sheet 页码	Integer
Label	Response.JobsDetail.PageSegment.Results	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Suggestion	Response.JobsDetail.PageSegment.Results	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）	Integer
PornInfo	Response.JobsDetail.PageSegment.Results	审核场景为**涉黄**的审核结果信息	Container
AdsInfo	Response.JobsDetail.PageSegment.Results	审核场景为**广告引导**的审核结果信息	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.PageSegment.Results.*Info	用于返回该审核场景的审核结果，返回值： 0：正常。	Integer

		1：确认为当前场景的违规内容。 2：疑似为当前场景的违规内容。	
Category	Response.JobsDetail.PageSegment.Results.*Info	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	Response.JobsDetail.PageSegment.Results.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签 注意：该字段可能返回空，表示未命中具体的子标签	String
Score	Response.JobsDetail.PageSegment.Results.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息 例如：色情 99，则表明该内容非常有可能属于色情内容	Integer
OcrResults	Response.JobsDetail.PageSegment.Results.*Info	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Container Array

Container 节点 OcrResults 的内容：

节点名称（关键字）	父节点	描述	类型
Text	Response.JobsDetail.PageSegment.Results.*Info.OcrResults	图片 OCR 文本识别出的具体文本内容。	String
Keywords	Response.JobsDetail.PageSegment.Results.*Info.OcrResults	在当前审核场景	String

版权所有：腾讯云计算（北京）有限责任公司

		下命中的关键词。	Array
Location	Response.JobsDetail.PageSegment.Results.*Info.OcrResults	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Container

ObjectResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.PageSegment.Results.*Info.ObjectResults	该标签用于返回所识别出的实体名称，例如人名。	String
Location	Response.JobsDetail.PageSegment.Results.*Info.ObjectResults	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Container

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float

版权所有：腾讯云计算（北京）有限责任公司

Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360**（**角度制**），方向为**逆时针旋转**。	Float

Container 节点 UserInfo 的内容：

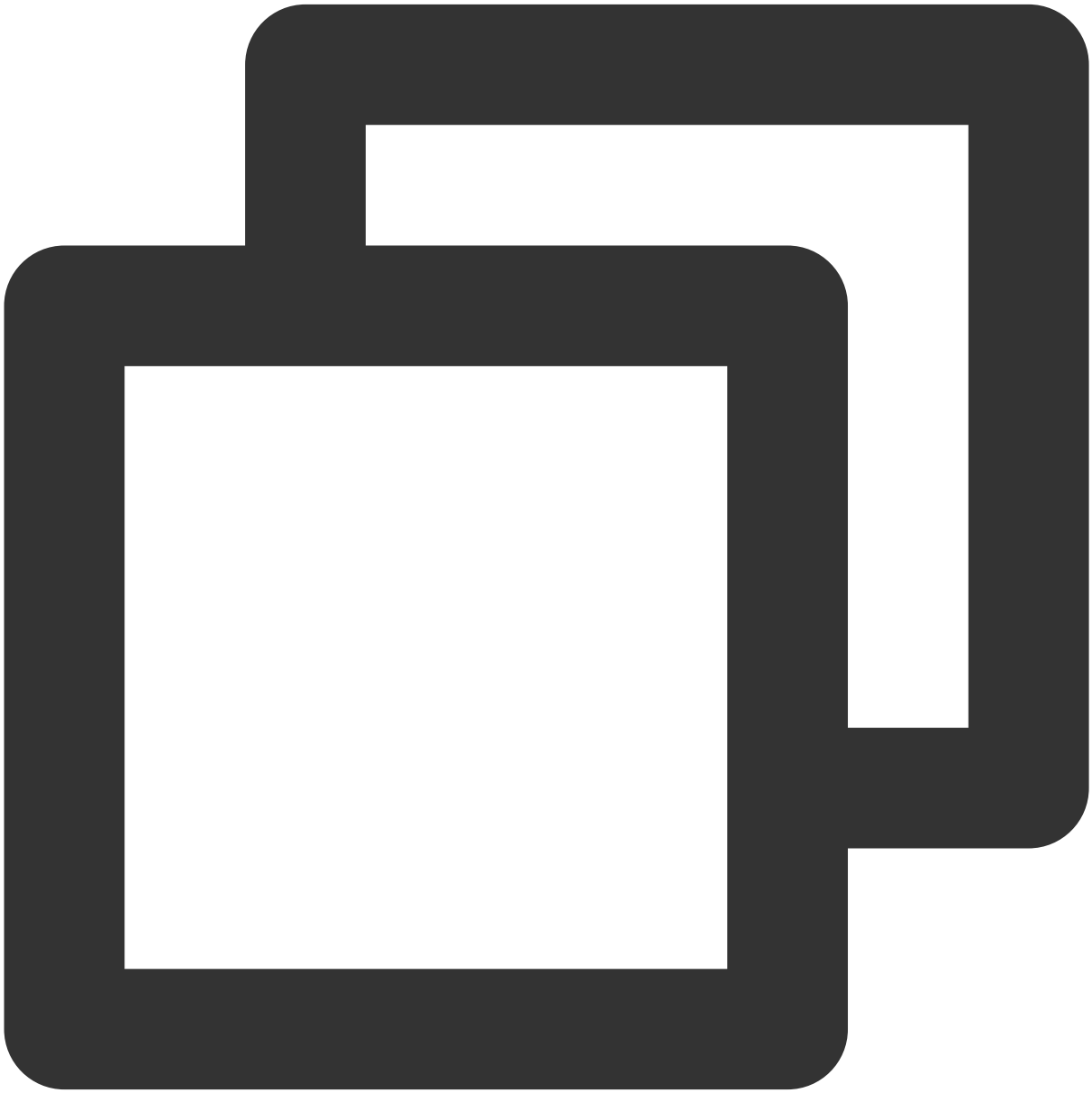
节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示IP地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

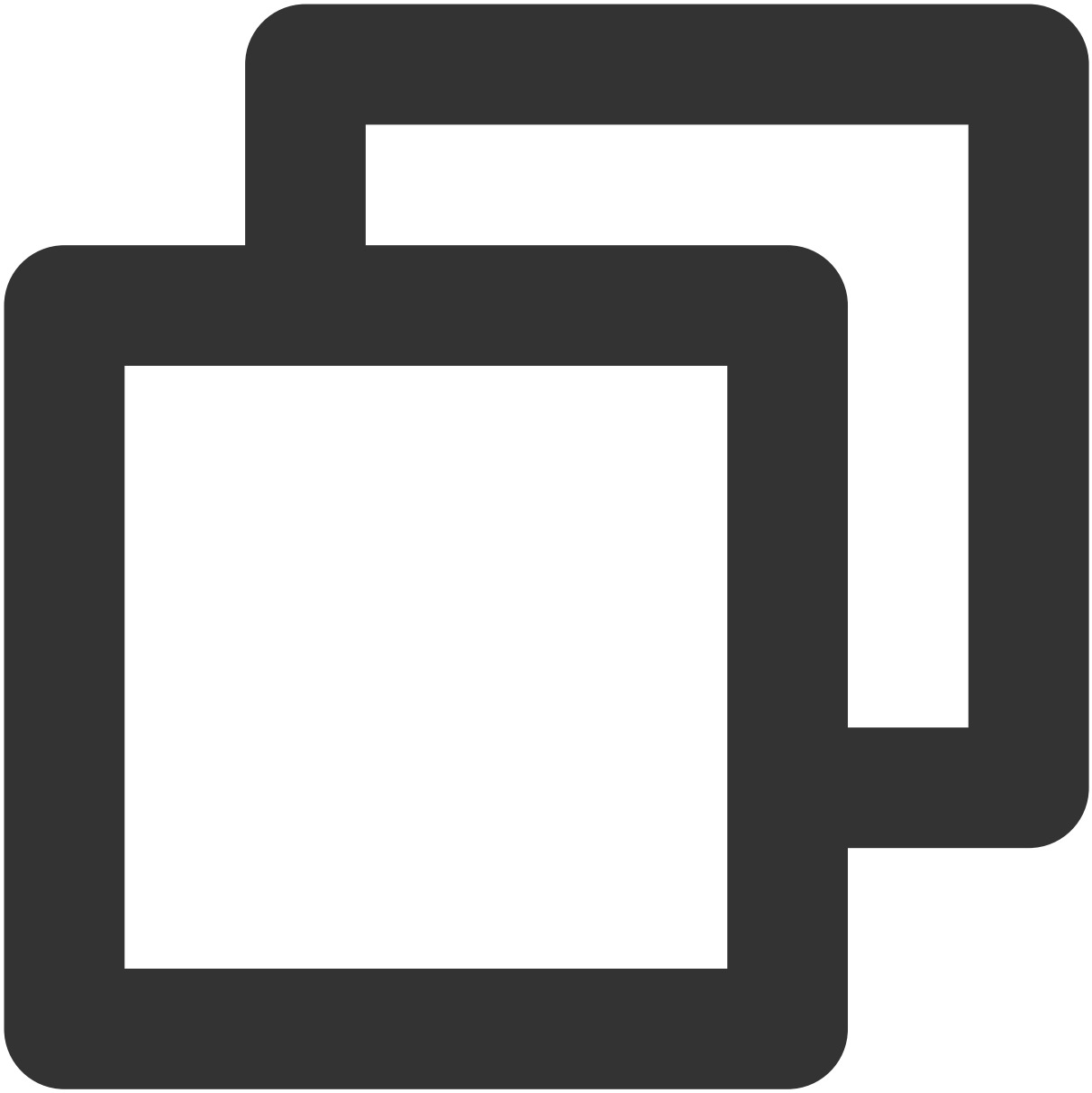
实际案例

请求



```
GET /document/auditing/stb1ca9fc8a34c525400863904**** HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
<Response>
    <JobsDetail>
      <JobId>stb1ca9fc8a34c525400863904****</JobId>
      <State>Success</State>
```

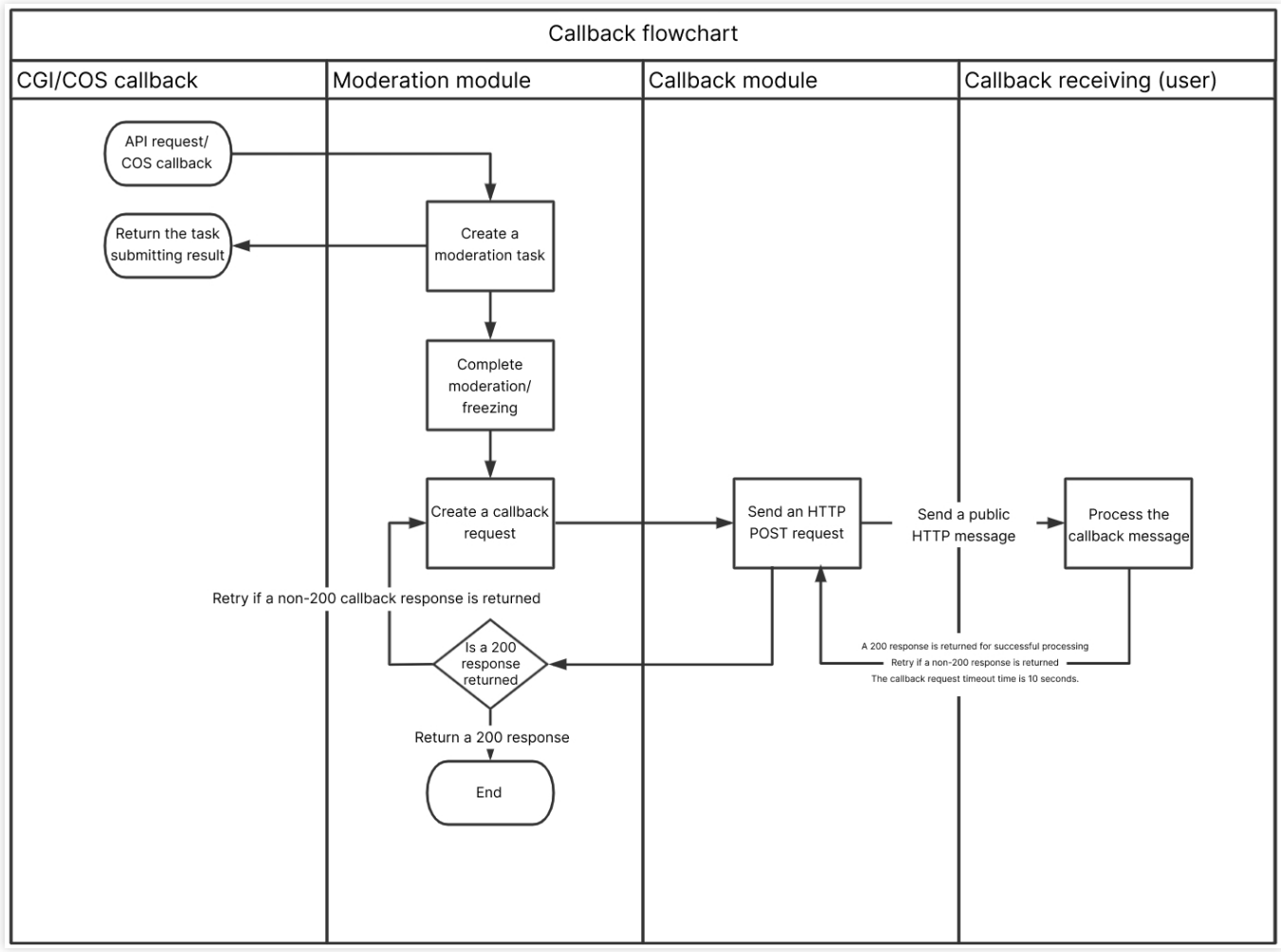
```
        <Suggestion>1</Suggestion>
        <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
        <Url>http://www.test.com/123.doc</Url>
        <PageCount>3</PageCount>
        <Labels>
          <PornInfo>
            <HitFlag>1</HitFlag>
            <Score>96</Score>
          </PornInfo>
        </Labels>
        <PageSegment>
          <Results>
            <Url>https://examplebucket-1250000000.cos.ap-nanjing.myqcloud.com/audit-1
            <Text>哈哈哈，您好</Text>
            <PageNumber>1</PageNumber>
            <PornInfo>
              <HitFlag>0</HitFlag>
              <SubLabel>sex</SubLabel>
              <Score>0</Score>
              <OcrResults>
                <Text>哈哈哈，您好</Text>
                <Keywords>您好</Keywords>
              </OcrResults>
            </PornInfo>
          </Results>
          <Results>
            <Url>https://examplebucket-1250000000.cos.ap-nanjing.myqcloud.com/audit-2
            <Text>哈哈哈，您好</Text>
            <PageNumber>1</PageNumber>
            <PornInfo>
              <HitFlag>0</HitFlag>
              <SubLabel>sex</SubLabel>
              <Score>0</Score>
              <OcrResults>
                <Text>哈哈哈，您好</Text>
                <Keywords>您好</Keywords>
              </OcrResults>
            </PornInfo>
          </Results>
        </PageSegment>
      </JobsDetail>
      <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

文档审核回调内容

最近更新时间：2023-12-30 10:34:29

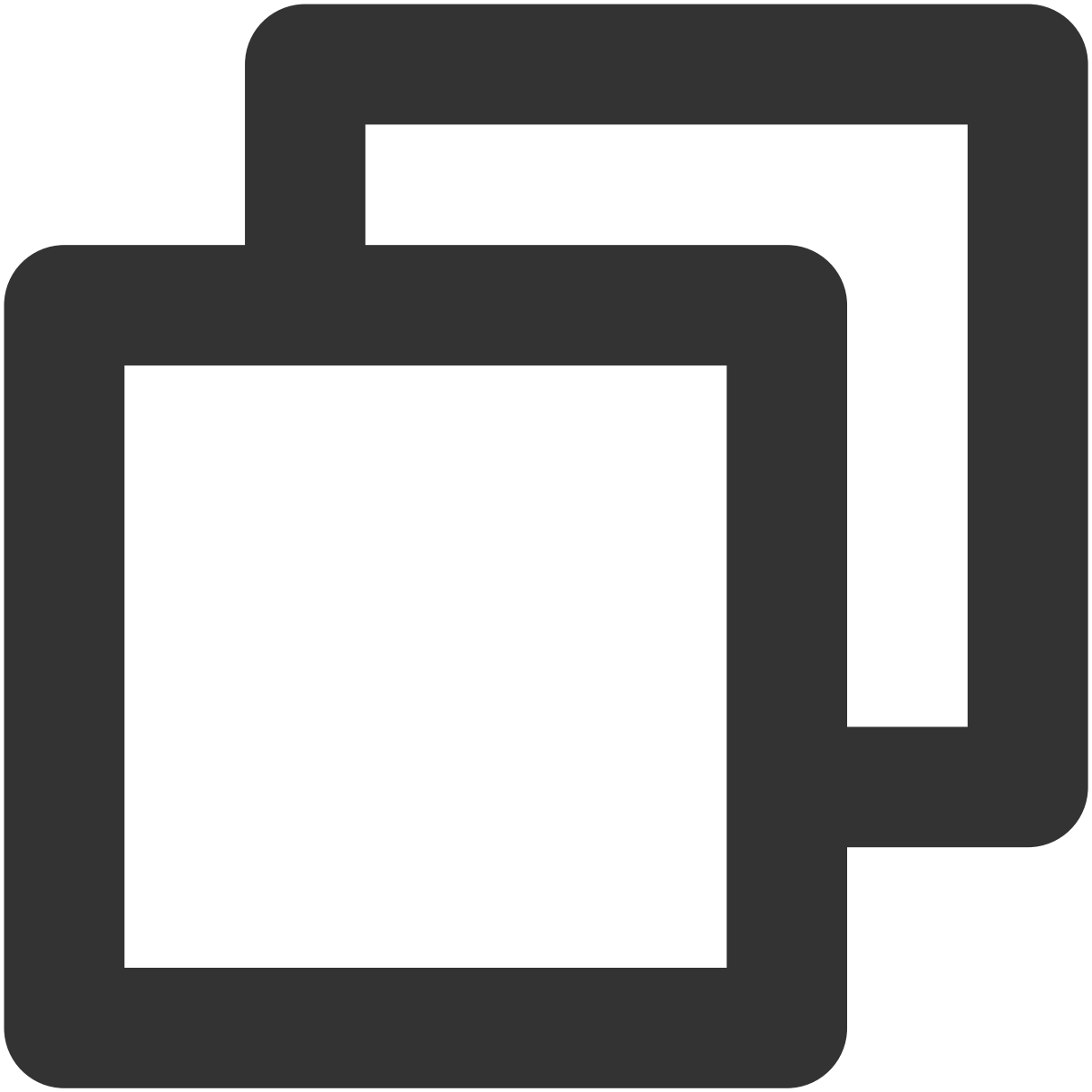
功能描述

若您配置了审核回调地址，在文档审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。



回调内容说明

该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：




```
{
  "EventName": "ReviewDocument",
  "JobsDetail": {
    "JobId": "6666666666666666666666666666666666",
    "State": "Success",
    "CreationTime": "",
    "Url": "",
    "Label": "Normal",
    "Suggestion": 0,
    "PageCount": 1,
    "Labels": {
```

```
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      },
      "PageSegment": {
        "Results": [
        {
          "Url": "",
          "Text": "",
          "PageNumber": 0,
          "SheetNumber": 0,
          "Label": "Normal",
          "Suggestion": 0,
          "PornInfo": {
            "HitFlag": 0,
            "Score": 0,
            "SubLabel": "",
            "OcrResults": [{
              "Text": "",
              "Keywords":["",""],
              "Location": {
                "X": 0,
                "Y": 0,
                "Width": 0,
                "Height": 0,
                "Rotate": 0
              }
            }]
          },
          "AdsInfo": {
            "HitFlag": 0,
            "Score": 0,
            "SubLabel": ""
          }
        }
        ]
      },
      "BucketId": "examplebucket-1250000000",
      "Region": "ap-chongqing",
      "ForbidState": 0,
      "CosHeaders": {
          "x-cos-meta-id": "666666"
```

```
      }
    }
  }
```

具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object
EventName	任务的类型， 为 `ReviewDocument` 。	String

JobsDetail 的内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
DataId	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String
State	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
CreationTime	审核任务的创建时间。	String
Object	审核的文件为存储在 COS 中的文件时，该字段表示本次审核的文件名称。	String
Url	审核的文件链接，创建任务使用Url时返回。	String
PageCount	文档审核会将文档转换为图片进行审核，该字段表示转换的图片总数量。	Integer
Labels	该字段用于返回命中的审核场景及对应的结果。	Object
PageSegment	文档转换为图片后，具体每张图片的审核结果信息。	Object

版权所有：腾讯云计算（北京）有限责任公司

ForbidState	若您设置了自动冻结，该字段表示文档的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段。创建任务时未设置UserInfo则无此字段。	Object
ListInfo	账号黑白名单结果。	Container

Labels 的内容如下：

节点名称（关键字）	描述	类型
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Labels 中 PornInfo，AdsInfo 的内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer

PageSegment 的内容如下：

节点名称（关键字）	描述	类型
Results	文档转换为图片后，每张图片的详细审核结果信息。	Array

Results 的内容如下：

节点名称（关键字）	描述	类型
Url	文档转换成图片后，您可以通过该地址查看该图片内容，地址格式为标准 URL 格式。注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String

版权所有：腾讯云计算（北京）有限责任公司

Text	该字段用于返回当前图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
PageNumber	该图片的页码，通常是文档的页码。	Integer
SheetNumber	如果审核的为表格文件，该字段表示表格内的 Sheet 页码。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Results 中 PornInfo，AdsInfo 的内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，例如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参	Float

版权所有：腾讯云计算（北京）有限责任公司

	数可唯一确定检测框的大小和位置。	
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

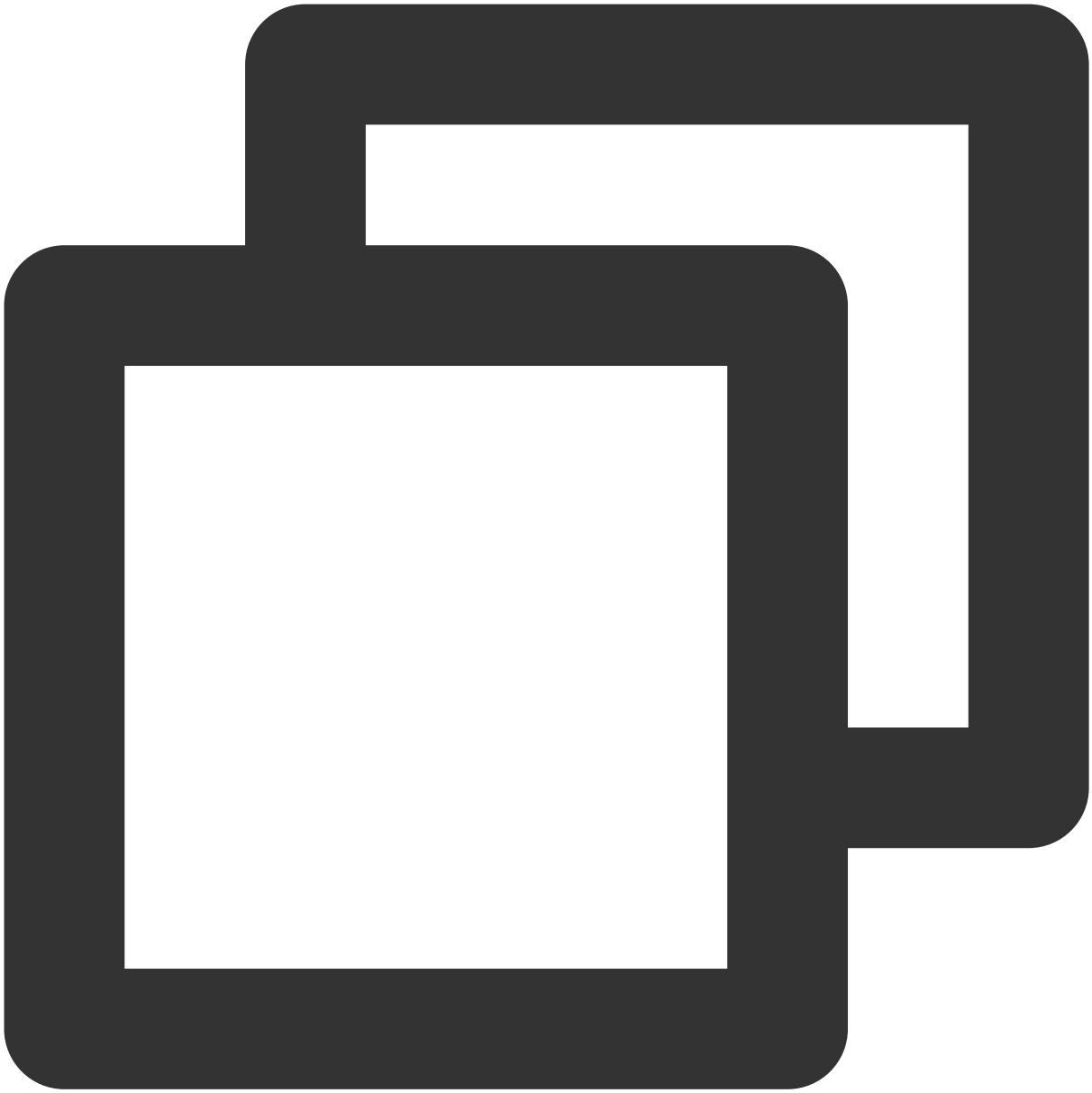
节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

节点名称（关键字）	描述	类型
ListType	命中的名单类型，取值为0（白名单）和1（黑名单）。	Integer
ListName	命中的名单名称。	String
Entity	命中了名单中的哪条内容。	String

版权所有：腾讯云计算（北京）有限责任公司

实际案例




```
{
  "EventName": "ReviewDocument",
  "JobsDetail": {
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "2021-08-10T21:01:10+08:00",
    "Url": "http://test.com/test.doc",
    "Label": "Normal",
```

```
      "Suggestion": 0,
      "PageCount": 1,
      "Labels": {
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      },
      "PageSegment": {
        "Results": [
        {
          "Url": "http://audit-125000000.cos.ap-chongqing.myqcloud.com/1.jpg",
          "Text": "",
          "PageNumber": 1,
          "SheetNumber": 0,
          "Label": "Normal",
          "Suggestion": 0,
          "PornInfo": {
            "HitFlag": 0,
            "Score": 0,
            "SubLabel": ""
          },
          "AdsInfo": {
            "HitFlag": 0,
            "Score": 0,
            "SubLabel": ""
          }
        }
        ]
      },
      "BucketId": "examplebucket-1250000000",
      "Region": "ap-chongqing",
      "ForbidState": 0,
      "CosHeaders": {
          "x-cos-meta-id": "xxxx"
      }
    }
}
```

云查毒接口
提交病毒检测任务

最近更新时间：2023-12-30 09:21:08

功能描述

云查毒功能为异步请求方式，您可以通过本接口对云上的文件进行文件病毒（例如木马病毒、蠕虫病毒等）检测，然后通过查询病毒检测结果接口，查询病毒检测的结果。

费用说明

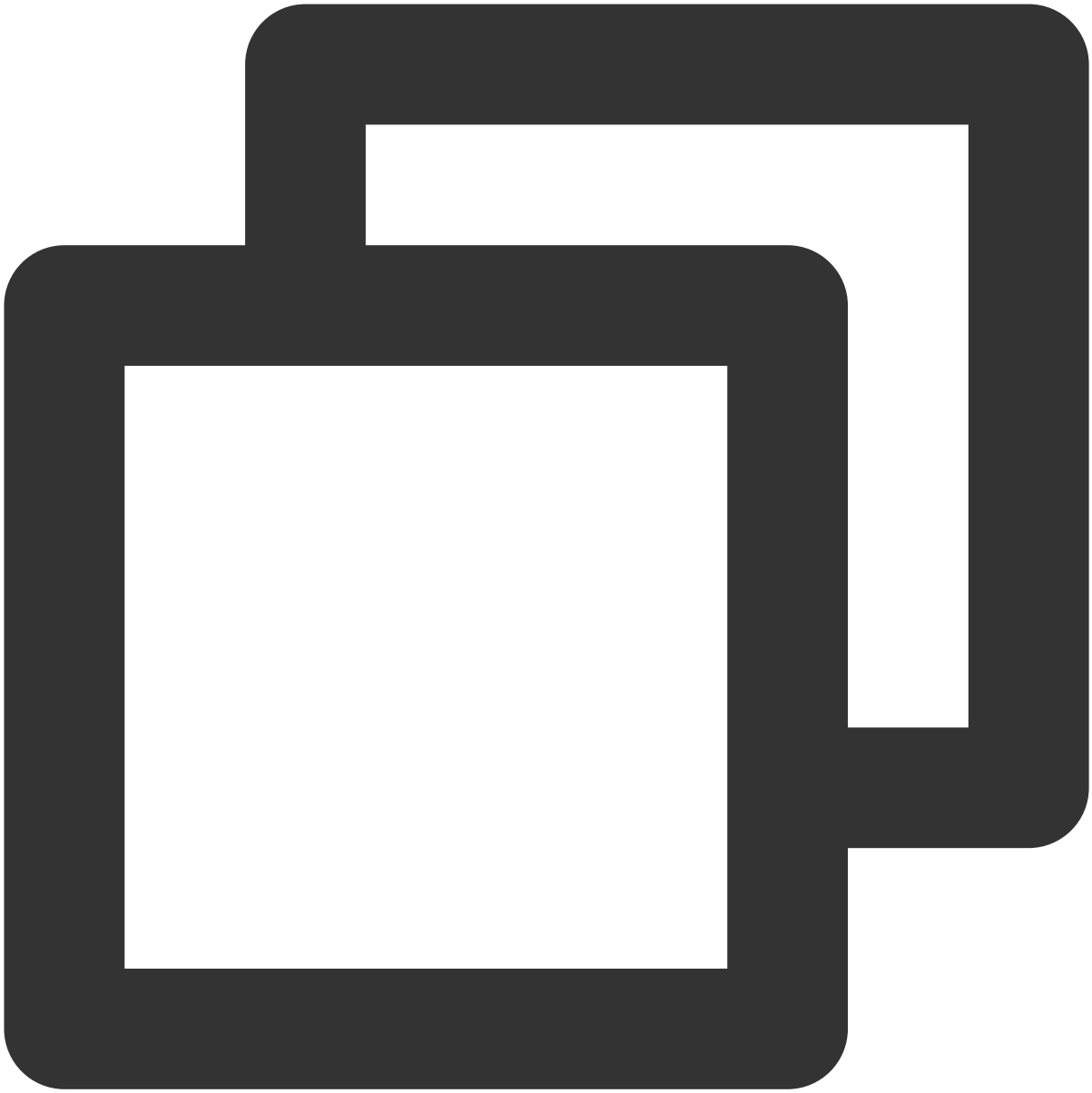
如果文件属于 COS 上的低频存储类型，调用检测会产生 COS 低频数据取回费用。
不支持检测 COS 上的归档存储类型和深度归档存储类型的文件，此类型文件请先 恢复归档文件。

限制说明

文件大小支持：**文件 < 1GB**。
文件支持格式：EXE、DLL、SCR、SYS、MSI、SWF、JAR、APPLETJAR、JS、VBS、HTML、BAT、WSF、PS1、HTM、MHT、HTA、CHM、LNK、DOC、DOCX、DOCX、DOTX、DOCM、DOTM、XLS、XLSX、XLSM、XLSB、XLTM、 XLTX、XLAM、PPT、PPTX、POTX、PPSX、PPTM、POTM、PPSM、XML、RTF、PDF、ELF、 XML、MAC、VBE、3DMAX、ISO、CPIO、RPM、XZ、DEB。
支持检测压缩文件，例如：zip、7z、rar4、rar5、inno、cab、autoit、nsis、gz、tar、iso、mime、chm、jar、dmg、apk 等。

请求

请求示例



```
POST /virus/detect HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

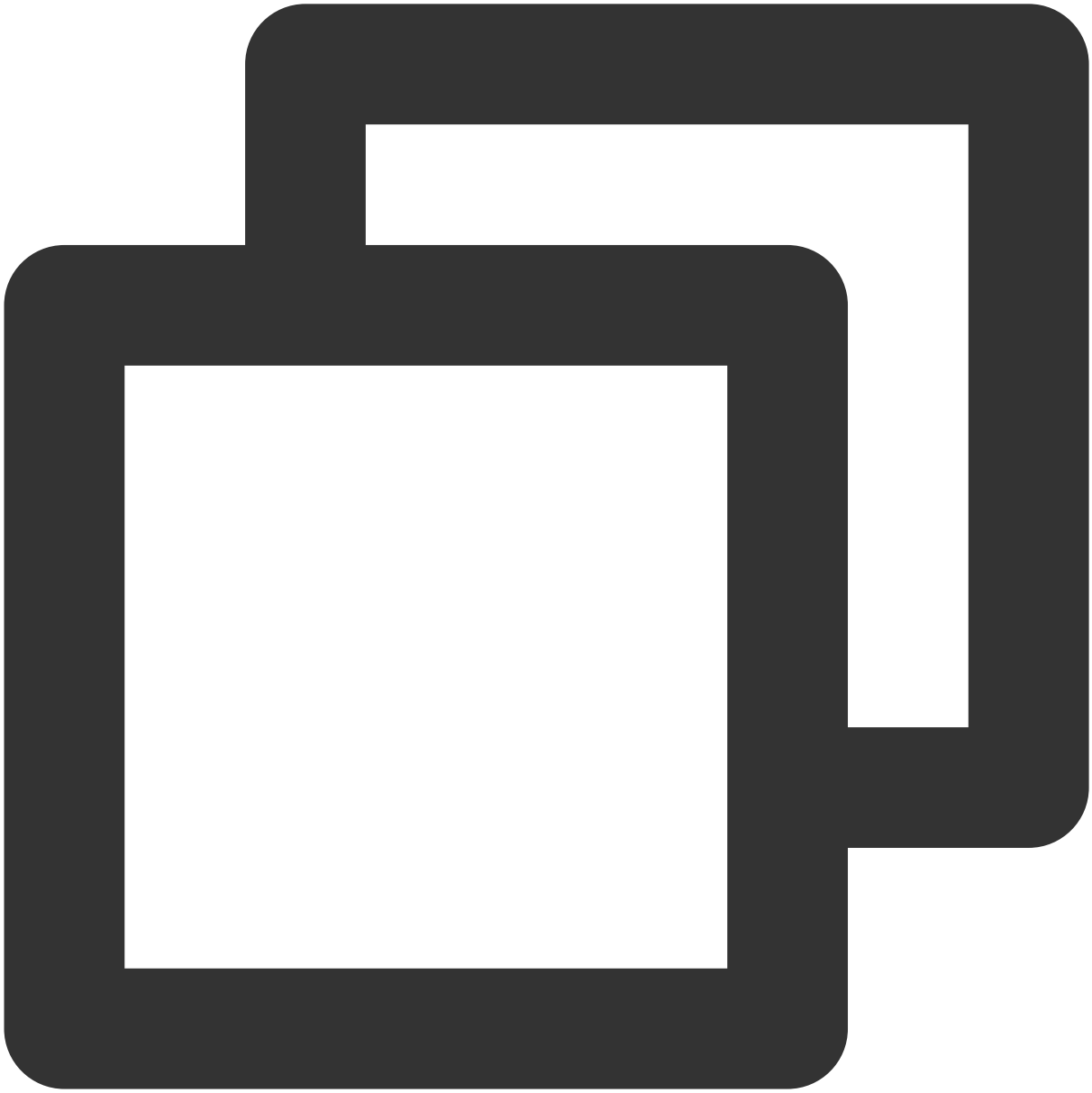
Authorization: Auth String（详情请参见 请求签名 文档）。

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：



```
<Request>
  <Input>
```

```
    <Object></Object>
    <Url></Url>
  </Input>
  <Conf>
    <DetectType>Virus</DetectType>
    <Callback></Callback>
  </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	病毒检测的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要检测的文件信息。	Container	是
Conf	Request	检测规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的病毒文件名称，例如在目录 test 中的文件 virus.doc，则文件名称为 `test/virus.doc` 。 注意：Object 和 Url 只能选择其中一种。	String	是
Url	Request.Input	病毒文件的链接地址，例如 `http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/virus.doc` 注意：Object 和 Url 只能选择其中一种。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
DetectType	Request.Conf	检测的病毒类型，当前固定为：Virus	String	是
Callback	Request.Conf	检测结果回调通知到您设置的地址，支持以	String	否

版权所有：腾讯云计算（北京）有限责任公司

		`http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com`		

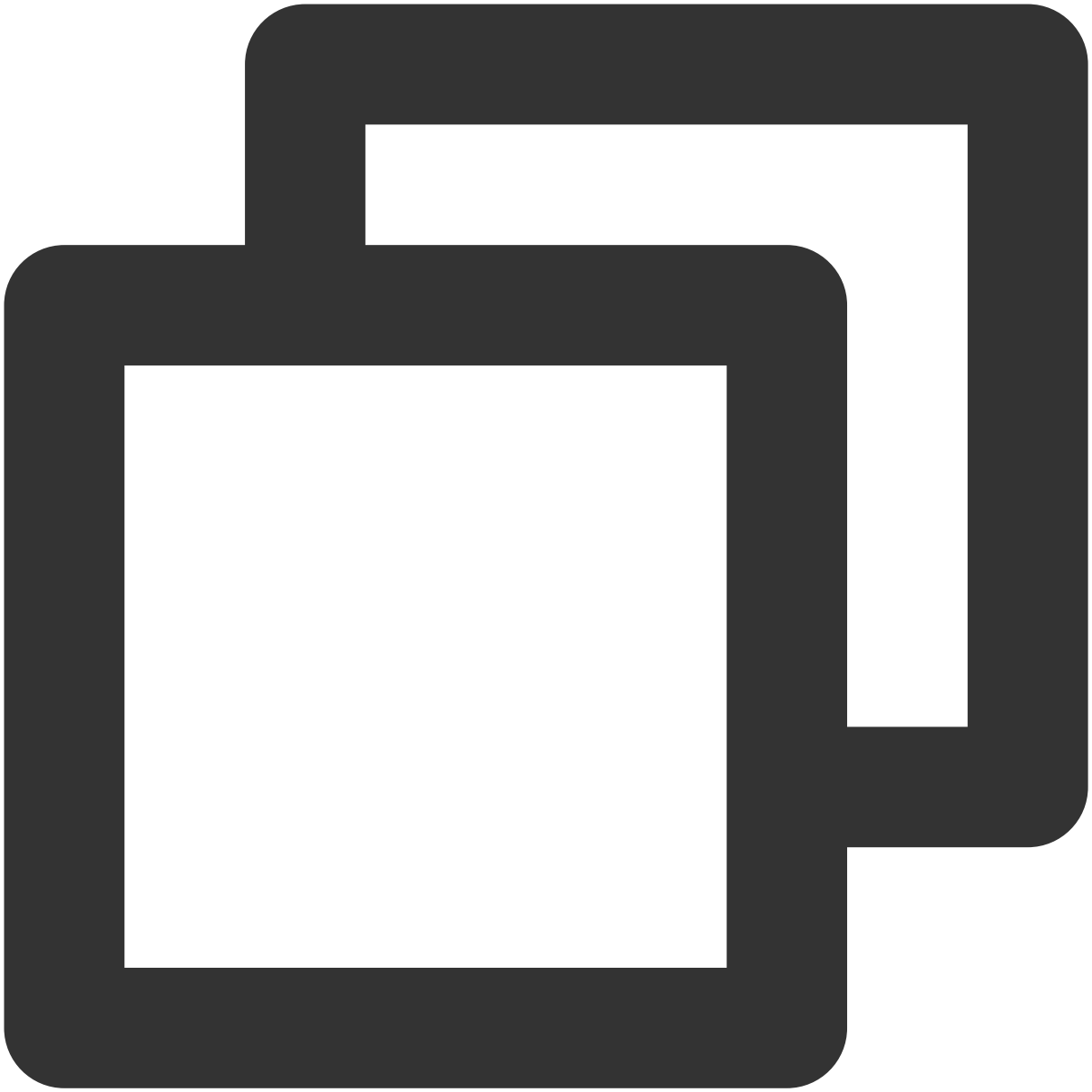
响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为 **application/xml** 数据，包含完整节点数据的内容展示如下：



```
<Response>
  <JobsDetail>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
  </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	病毒检测返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	病毒检测任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

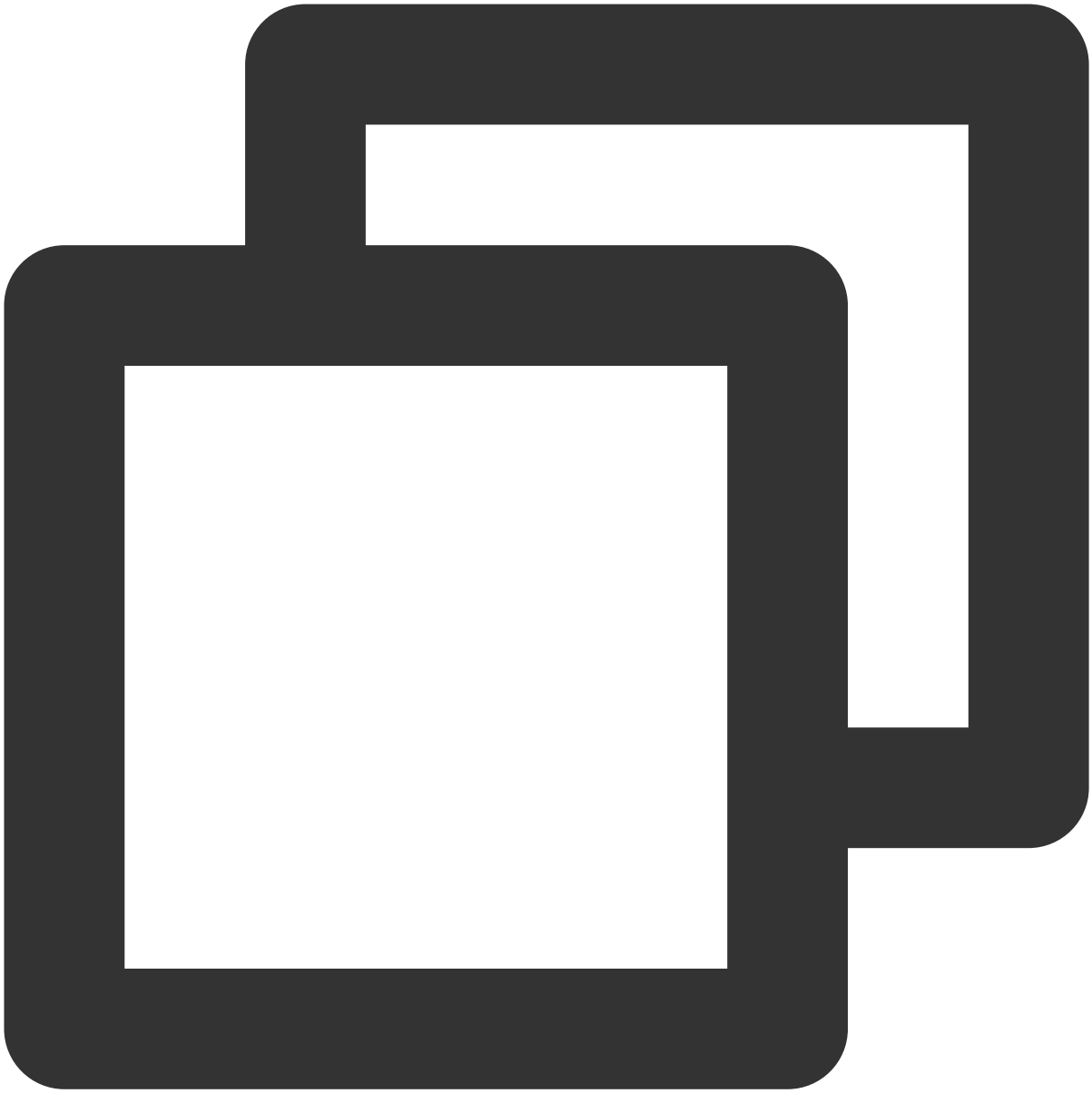
节点名称（关键字）	父节点	描述	类型
JobId	Response.JobsDetail	本次病毒检测任务的 ID。	String
State	Response.JobsDetail	病毒检测任务的状态，值为 Submitted（已提交检测）、Success（检测成功）、Failed（检测失败）、Auditing（检测中）其中一个	String
CreationTime	Response.JobsDetail	病毒检测任务的创建时间。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

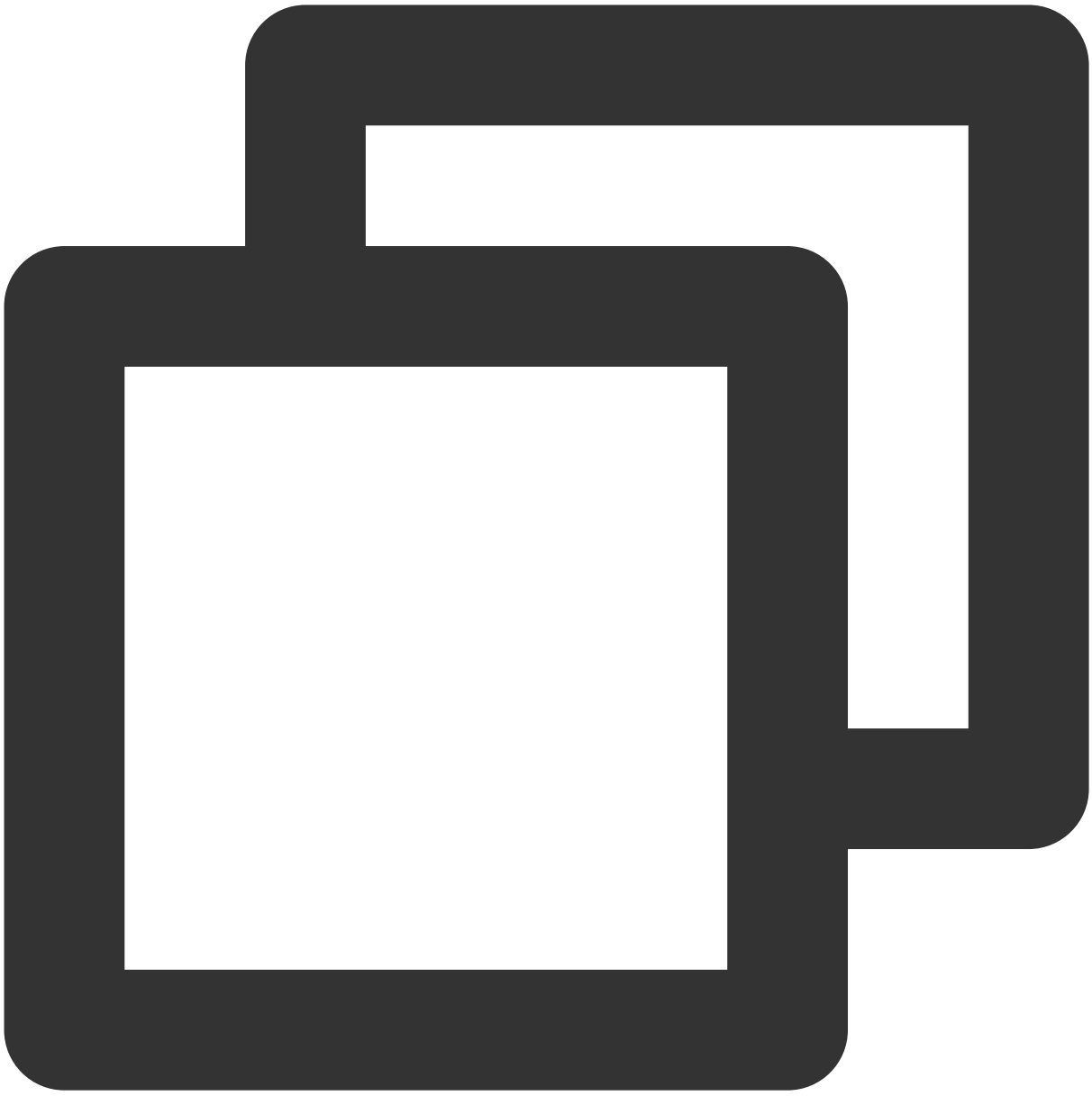
请求



```
POST /virus/detect HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Input>
    <Object>a.doc</Object>
  </Input>
  <Conf>
```

```
    <DetectType>Virus</DetectType>
    <Callback>http://callback.com/</Callback>
  </Conf>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
```

```
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>ssb1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Submitted</State>
    <CreationTime>2021-07-07T12:12:12+0800</CreationTime>
  </JobsDetail>
</Response>
```