数据万象

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
- 公共请求头部
- 公共响应头部
- 存储桶配置
  - 绑定数据万象服务
  - 查询数据万象服务
  - 解绑数据万象服务
  - 查询防盗链
  - 设置防盗链
  - 开通原图保护
  - 查询原图保护状态
  - 关闭原图保护
- 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
    - 管道操作符
    - 限制图片大小
    - 支持字体列表
  - 图片样式

版权所有：腾讯云计算（北京）有限责任公司

- 增加样式
- 查询样式
- 删除样式
- 图片压缩
  - Guetzli 压缩
    - 开通 Guetzli 压缩
    - 查询 Guetzli 状态
    - 关闭 Guetzli 压缩
  - WebP 压缩
  - HEIF 压缩
  - TPG 压缩
  - AVIF 压缩
  - 高级压缩
  - SVG 压缩
- 图片持久化处理
- 盲水印
- 异常图片检测
- 队列接口
  - 查询队列
  - 更新队列

AI 内容识别

- 人脸识别
  - 人脸特效
- 人脸核身
  - 身份证识别
- 图片文字识别（OCR）
  - 通用文字识别
- 语音识别
  - 任务接口
    - 提交语音识别任务
    - 查询指定的语音识别任务
    - 批量拉取语音识别任务
  - 队列接口
    - 查询语音识别队列
    - 更新语音识别队列
- 以图搜图
  - 开通以图搜图
  - 添加图库图片

- 图片搜索接口
- 删除图库图片
- 队列接口
- 查询队列
- 更新队列
- 服务配置
- 开通 AI 内容识别
- 查询 AI 内容识别

智能语音
- 智能语音接口列表
- 智能语音服务配置接口
- 查询语音识别开通状态

媒体处理
- 媒体处理接口列表
- 媒体处理服务配置接口
- 查询媒体处理服务
- 绑定媒体处理服务
- 解绑媒体处理服务
- 媒体 bucket 接口
- 查询媒体处理开通状态
- 开通媒体处理
- CreateMediaBucket
- 队列接口
- 搜索媒体处理队列
- 更新媒体处理队列
- 同步媒体处理
- 获取媒体截图
- 获取媒体文件截图
- 获取媒体文件截图（万象域名）
- GenerateSnapshot
- 获取媒体信息
- 获取媒体文件信息
- 读取私有 M3U8
- 获取私有 m3u8

内容审核
- 音频审核
- 提交音频审核任务
- 文本审核

- 文本内容同步审核
- 文档预览
  - 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
  - 文档转 HTML
    - 快速入门
    - 自定义配置概述
    - 配置项说明
    - 高级功能
    - 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
    - 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
  - 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

版权所有：腾讯云计算（北京）有限责任公司

- 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
    - 提交媒体处理任务
      - 提交视频转动图任务
      - 提交视频拼接任务
      - 提交添加数字水印任务
      - 提交提取数字水印任务
      - 提交获取媒体信息任务
      - 提交音频降噪任务
      - 提交视频质量评分任务
      - 提交 SDRtoHDR 任务
      - 提交转封装任务
      - 提交智能封面任务
      - 提交视频截帧任务
      - 提交流分离任务
      - 提交超分辨率任务
      - 提交音视频转码任务
      - 提交语音合成任务
      - 提交精彩集锦任务
      - 提交视频增强任务
      - 提交视频标签任务
      - 提交人声分离任务
      - 提交极速高清任务
    - 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

版权所有：腾讯云计算（北京）有限责任公司

SDRtoHDR
转封装（音视频分段）
智能封面
截帧
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
极速高清转码
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
提交语音识别任务
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
批量拉取内容识别任务
内容识别任务回调
翻译任务回调
分词任务回调
模板接口

版权所有：腾讯云计算（北京）有限责任公司

- 媒体处理
  - 创建媒体处理模板
    - 创建动图模板
    - 创建拼接模板
    - 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
  - 媒体处理
    - 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
    - 音视频转码 pro
    - 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

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
批量任务接口
触发任务（独立节点）
触发任务（工作流）
取消任务
查询任务
批量拉取任务
错误码
附录
结构体

API 文档
公共请求头部

最近更新时间：2024-01-18 09:33:22

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-token	使用临时安全凭证时需要传入的安全令牌字段。	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。

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

公共响应头部

最近更新时间：2024-01-18 09:33:22

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

存储桶配置
绑定数据万象服务

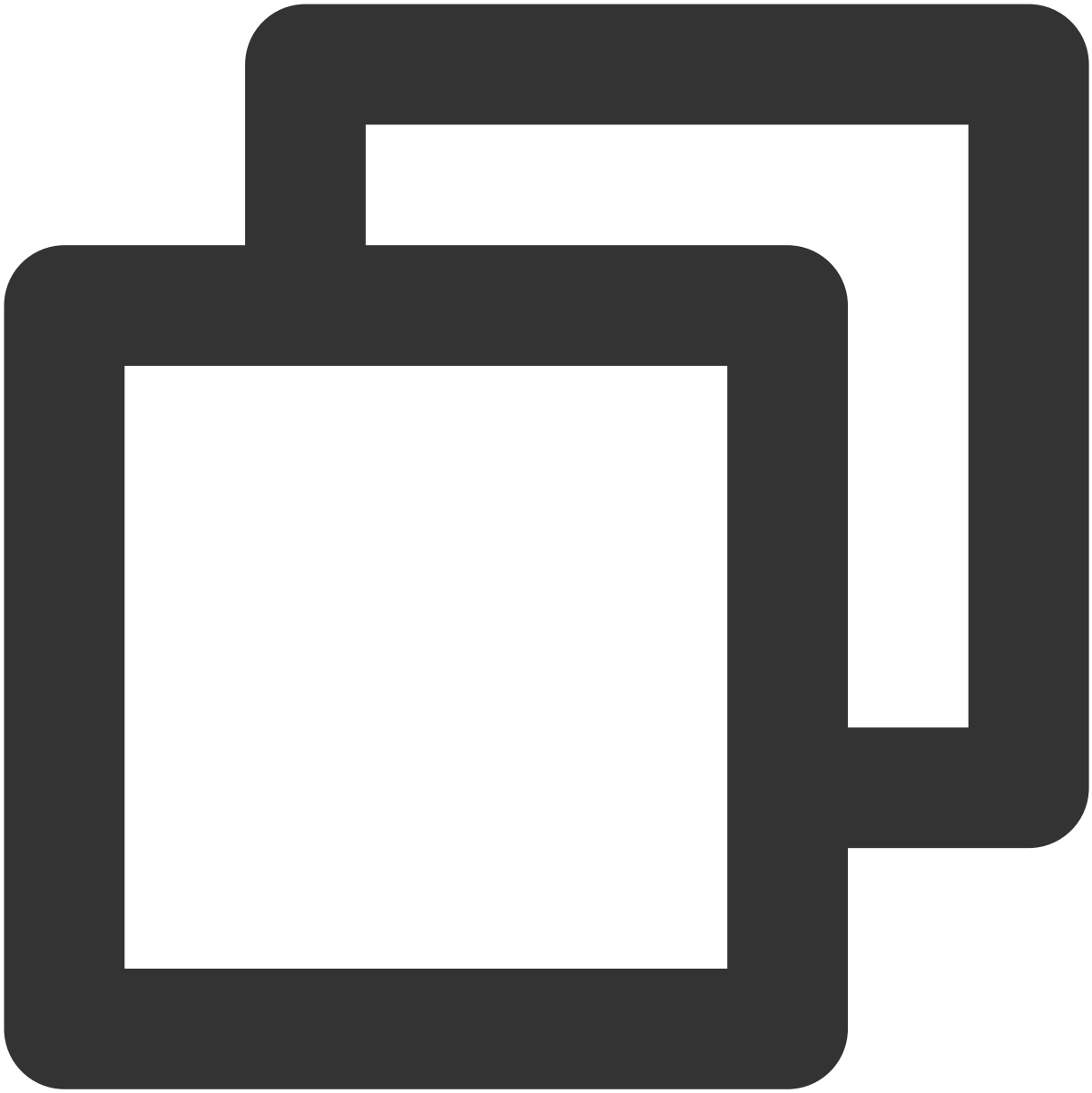
最近更新时间：2024-01-18 09:33:22

功能描述

用于开通数据万象服务，需要绑定对象存储（Cloud Object Storage，COS）上已有的存储桶。存储桶的其他功能如查询、修改等，使用 COS 的 API 接口。

请求

请求示例



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

说明：

Authorization：Auth String （详情参阅 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

该响应的响应体为空。

查询数据万象服务

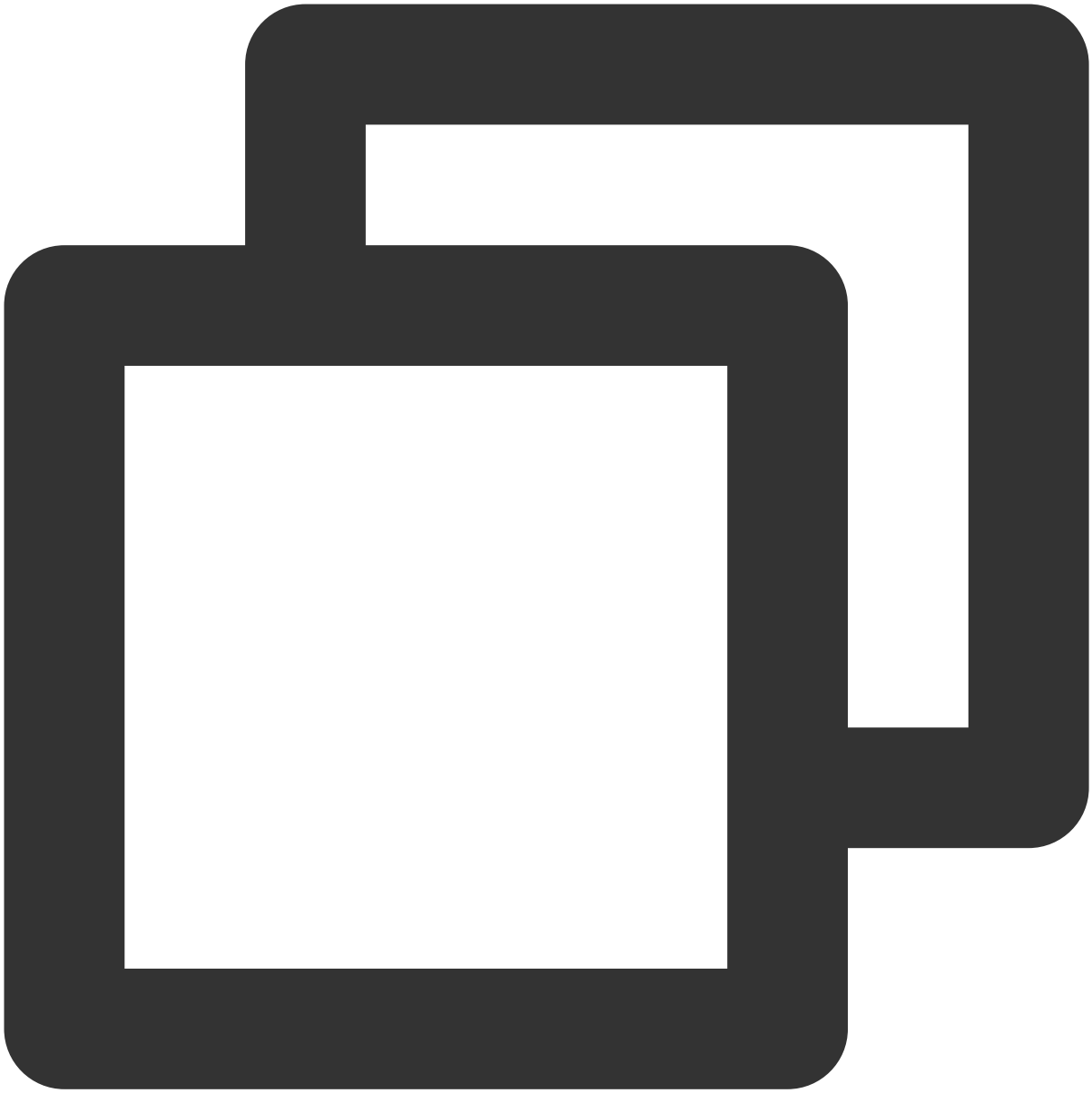
最近更新时间：2024-01-18 09:33:22

功能描述

用于查询该 Bucket 是否已开通数据万象服务。

请求

请求示例



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

说明：

Authorization： Auth String （详情参阅 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

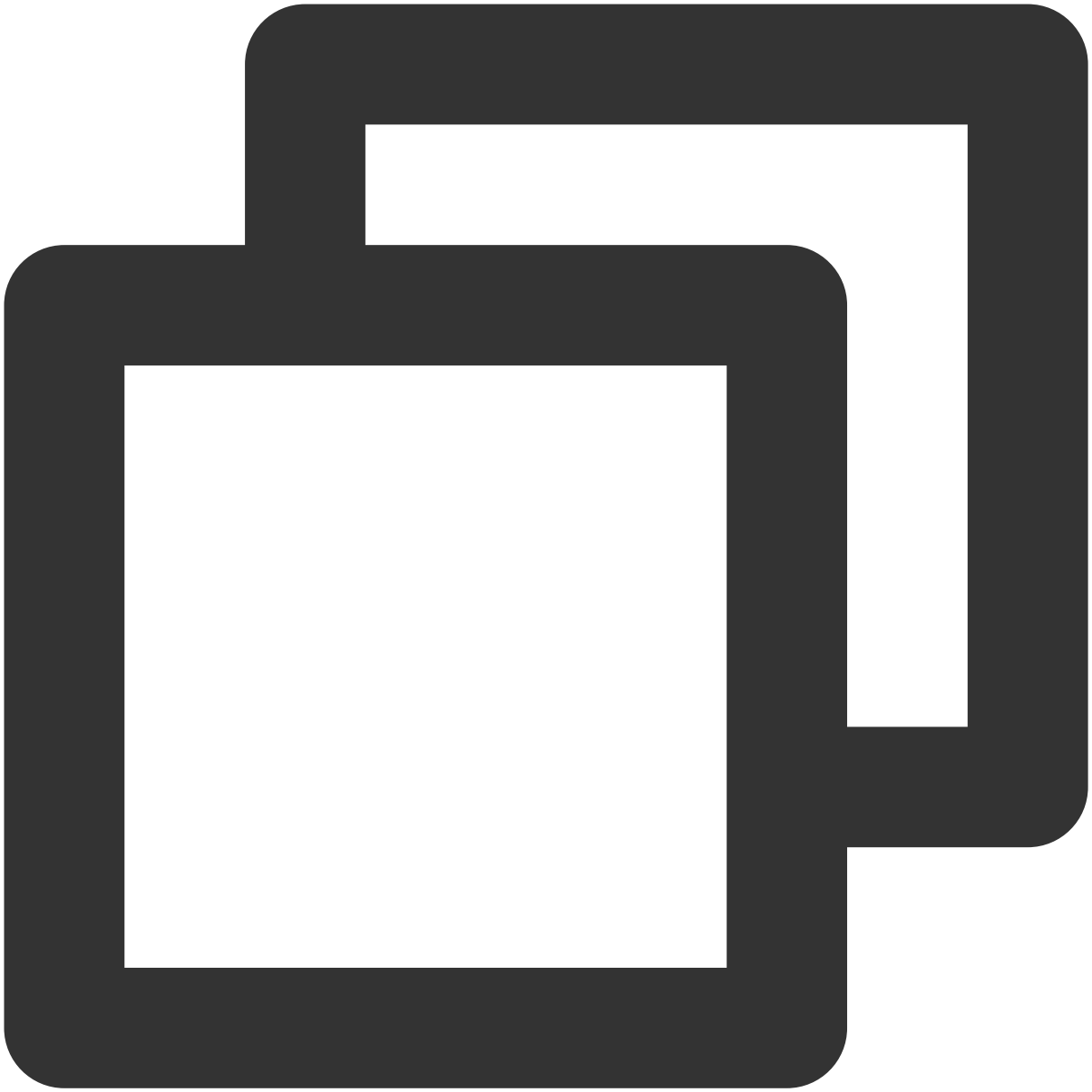
该请求的请求体为空。

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



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

节点信息如下：

节点名称（关键字）	描述
CIStatus	on 与 off 两种状态

解绑数据万象服务

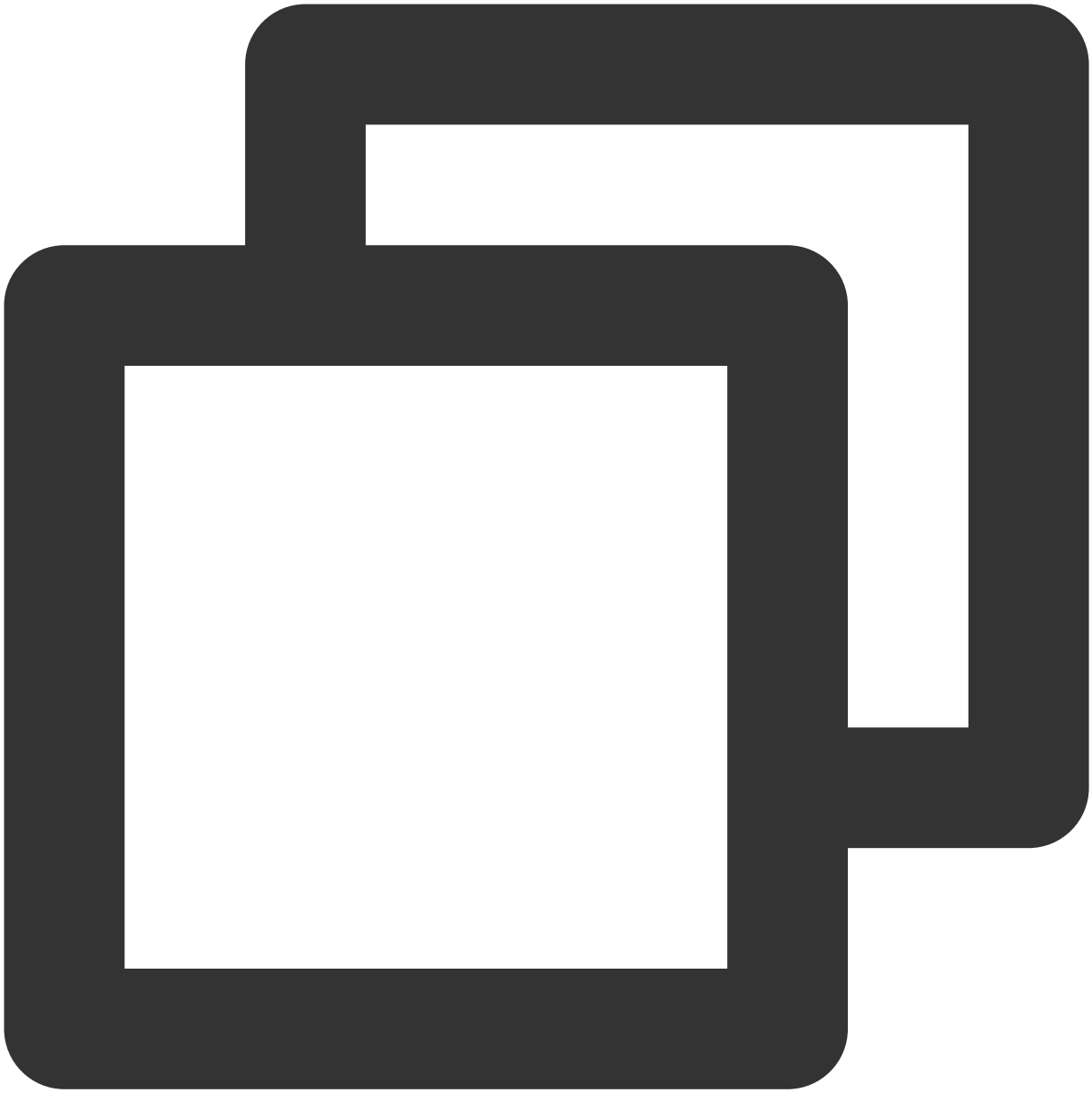
最近更新时间：2024-01-18 09:33:22

功能描述

解绑开通数据万象服务的 Bucket（Bucket 及桶中文件会保留在对象存储（Cloud Object Storage，COS））。

请求

请求示例



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

说明：

Authorization：Auth String （详情参阅 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

该响应的响应体为空。

查询防盗链

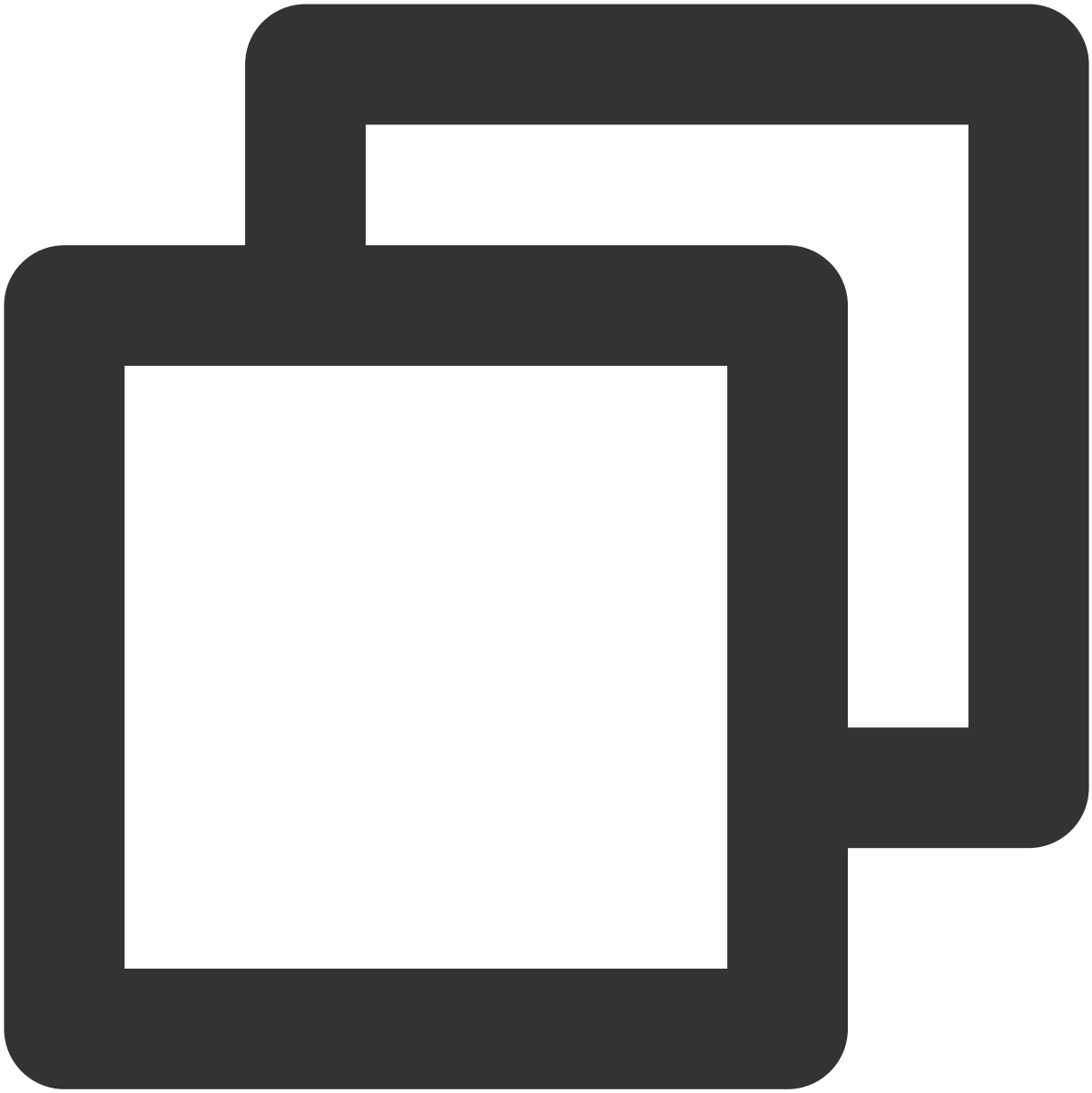
最近更新时间：2024-01-18 09:33:22

功能描述

用于查询 Bucket 中防盗链详情（黑白名单）。

请求

请求示例



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

说明：

Authorization：Auth String （详情参阅 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。

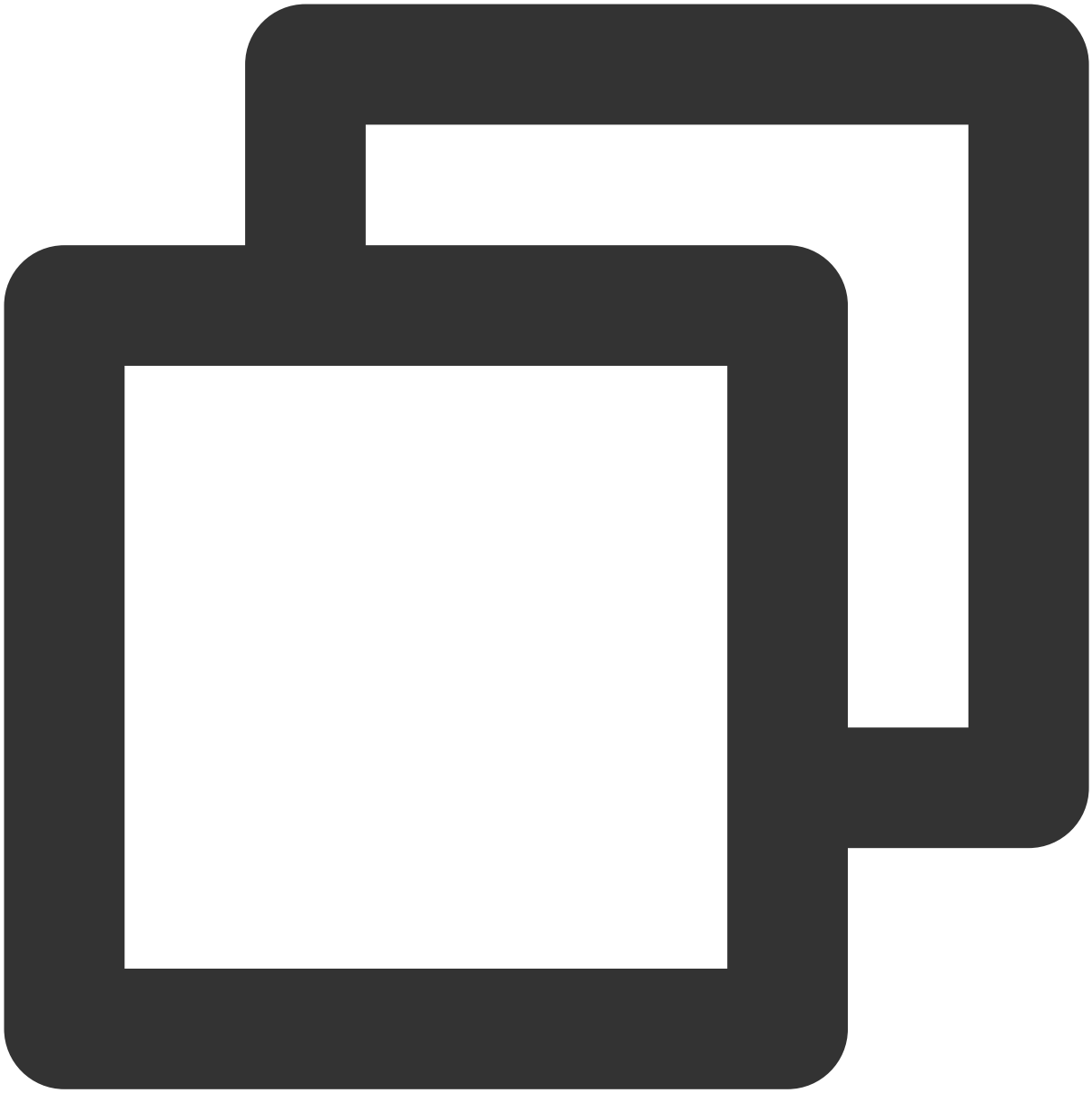
响应

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

具体内容包括防盗链的种类并枚举所有地址。



```
<?xml version="1.0" encoding="UTF-8" ?>
<Hotlink>
    <Status>on</Status>
    <Type>white</Type>
    <Url>xxx</Url>
    <Url>xxx</Url>
</Hotlink>
```

设置防盗链

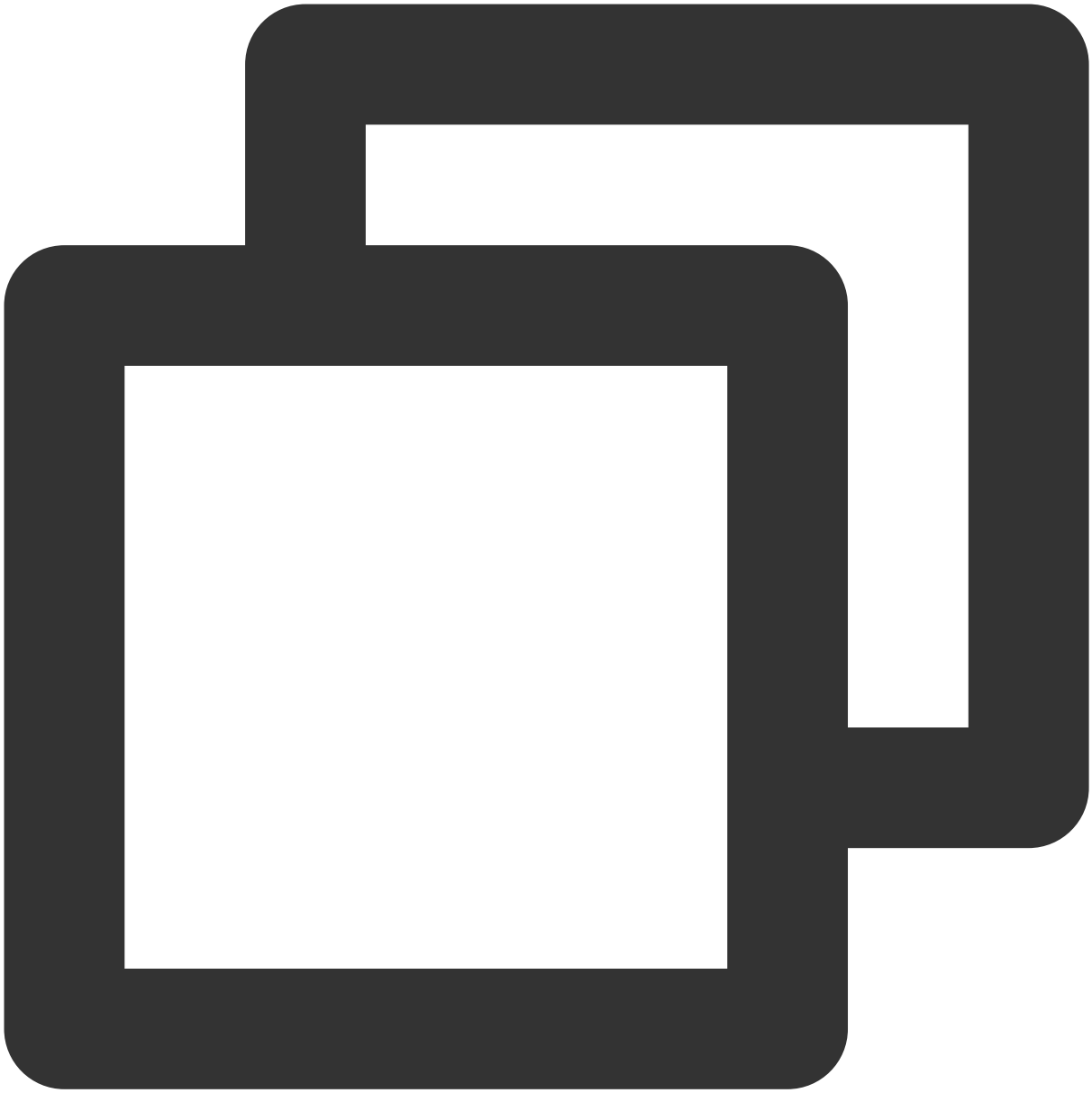
最近更新时间：2024-01-18 09:33:22

功能描述

用于对存储桶添加防盗链（设置黑白名单）。

请求

请求示例



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

说明：

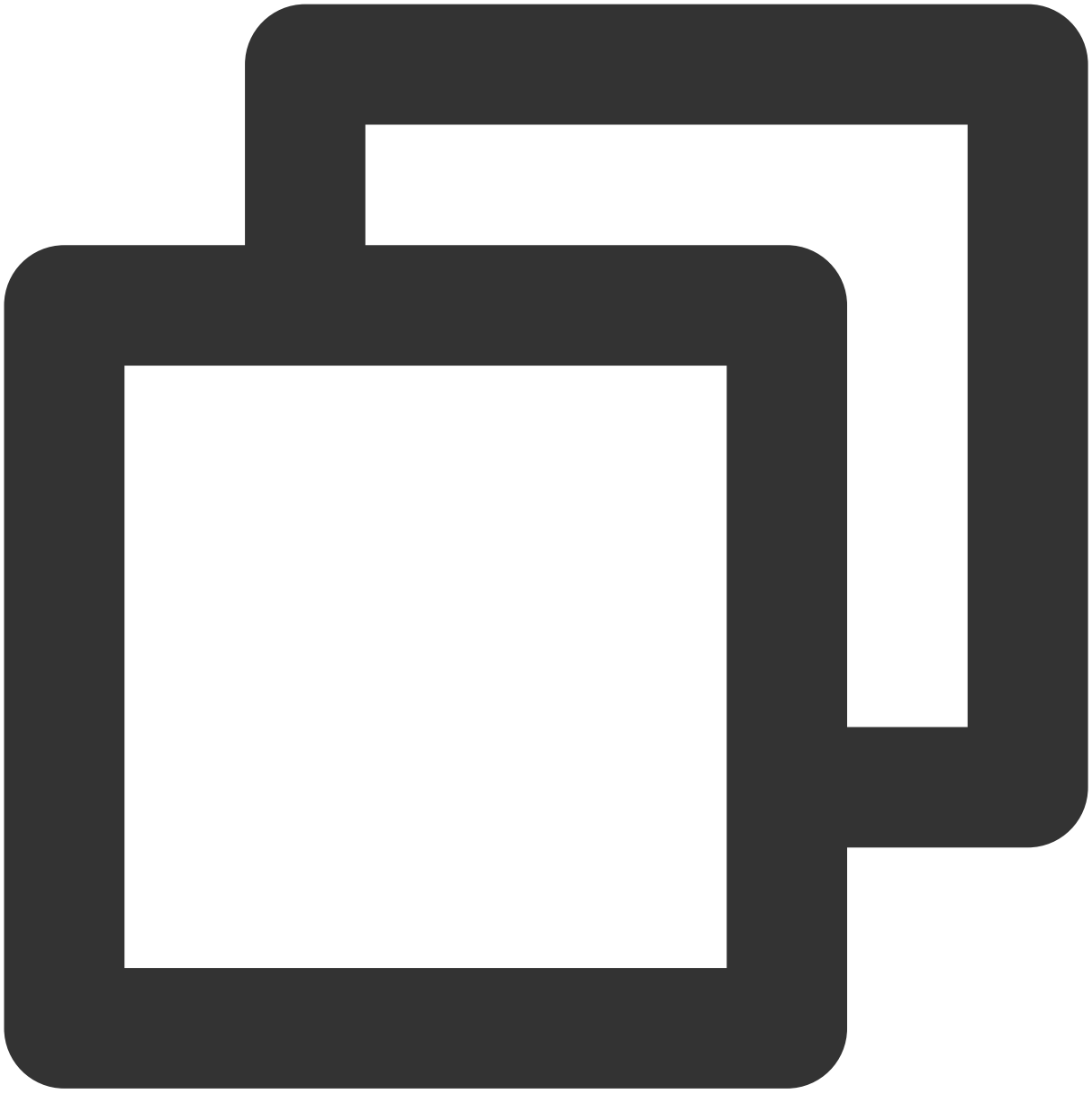
Authorization: Auth String （详情请参见 请求签名 文档）。

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

该请求的请求体具体节点内容为：



```
<Hotlink>
        <Url>xxxxx</Url>
        <Type>black</Type>
</Hotlink>>
```

具体内容描述如下：

名称	父节点	描述	类型	是否必选

Url	Hotlink	域名地址	Container	是
Type	Hotlink	防盗链类型，white 为白名单，black 为黑名单，off 为关闭。	String	是

响应

响应头

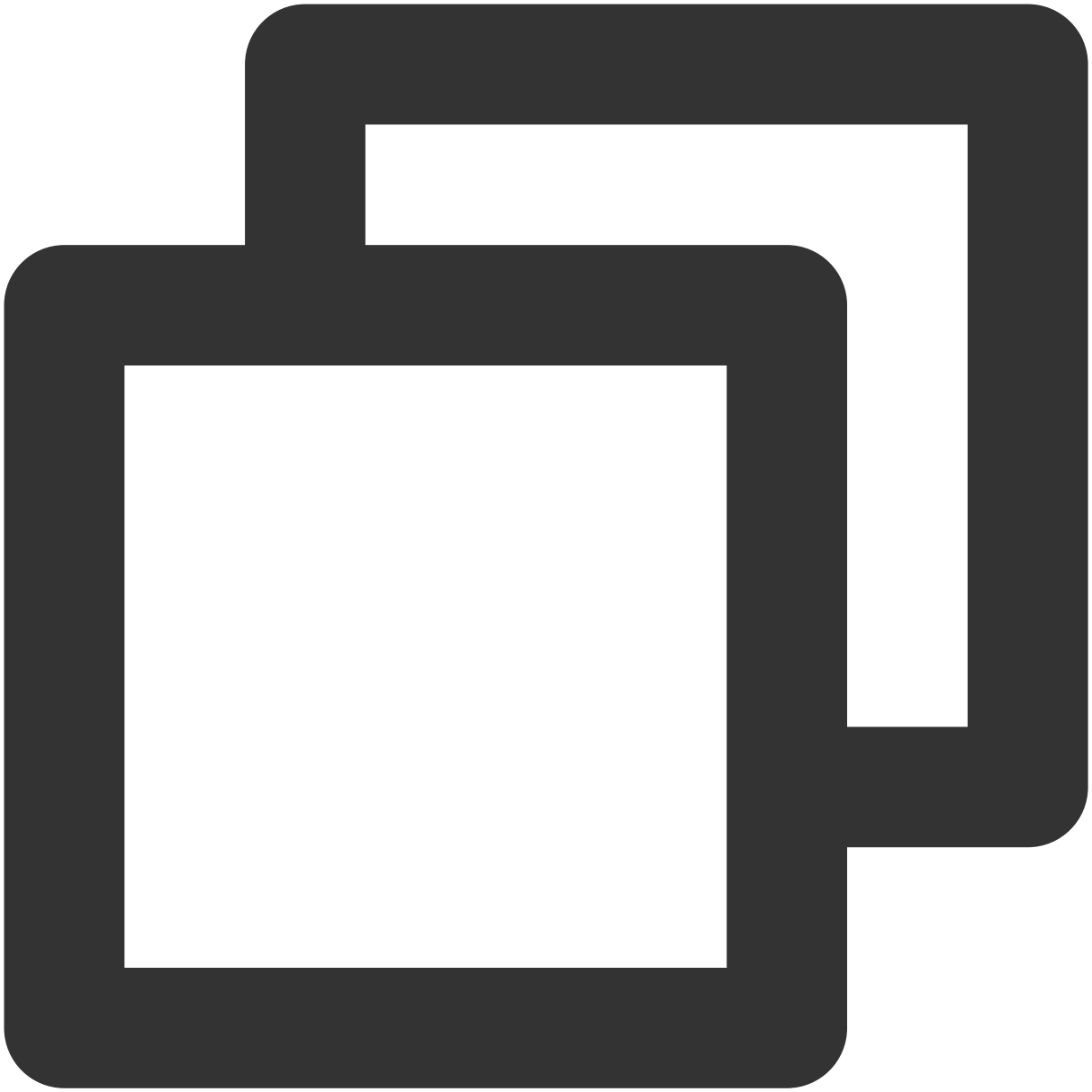
此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回为空。

实际案例

请求

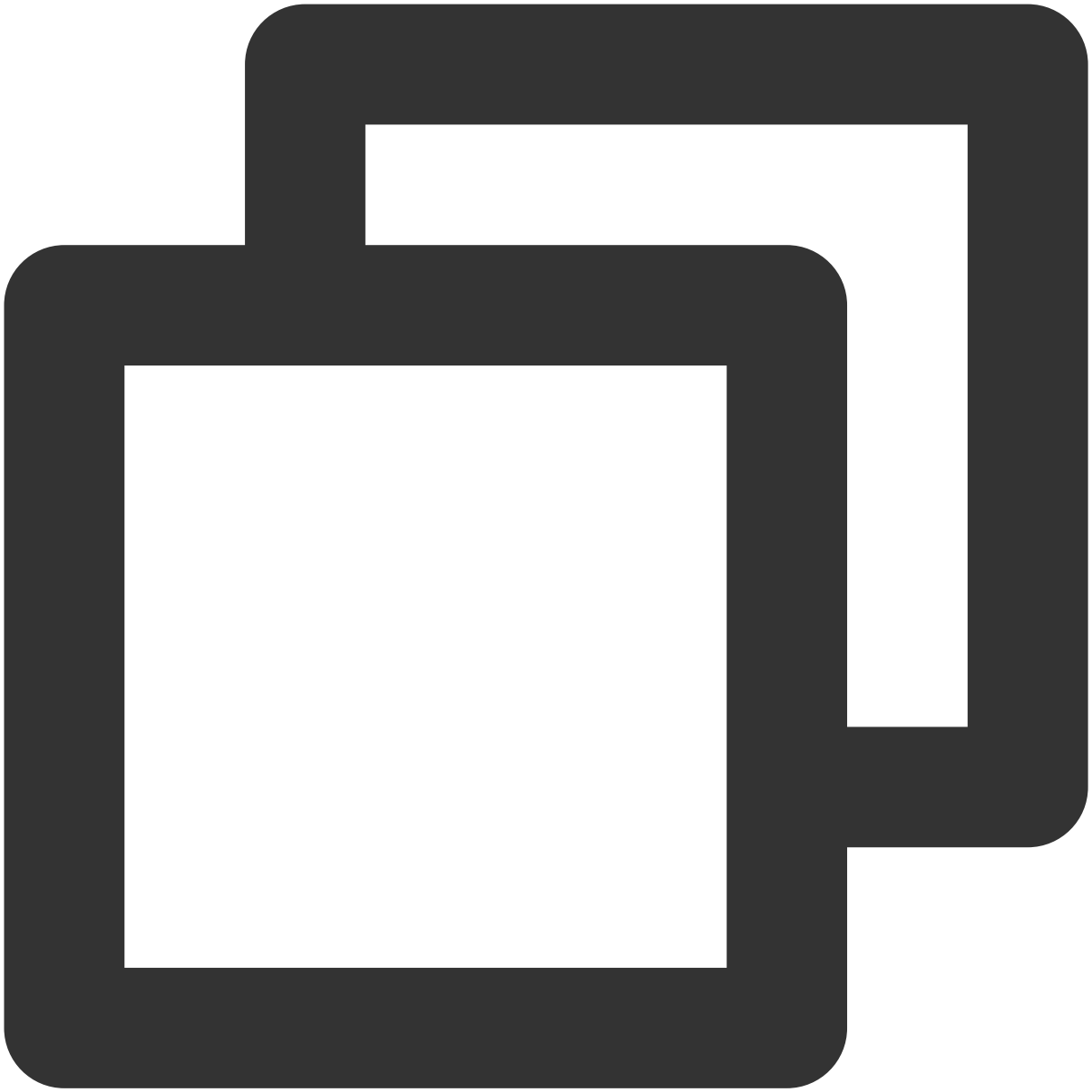


```
PUT /?hotlink HTTP/1.1
Host: examplebucket-1250000000.pic.ap-chengdu.myqcloud.com
Date: Wed, 18 Mar 2000 06:57:54 GMT
Authorization: XXXXXXXXXXXX

<Hotlink>
        <Url>www.example.com</Url>
        <Type>black</Type>
</Hotlink>
```

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

开通原图保护

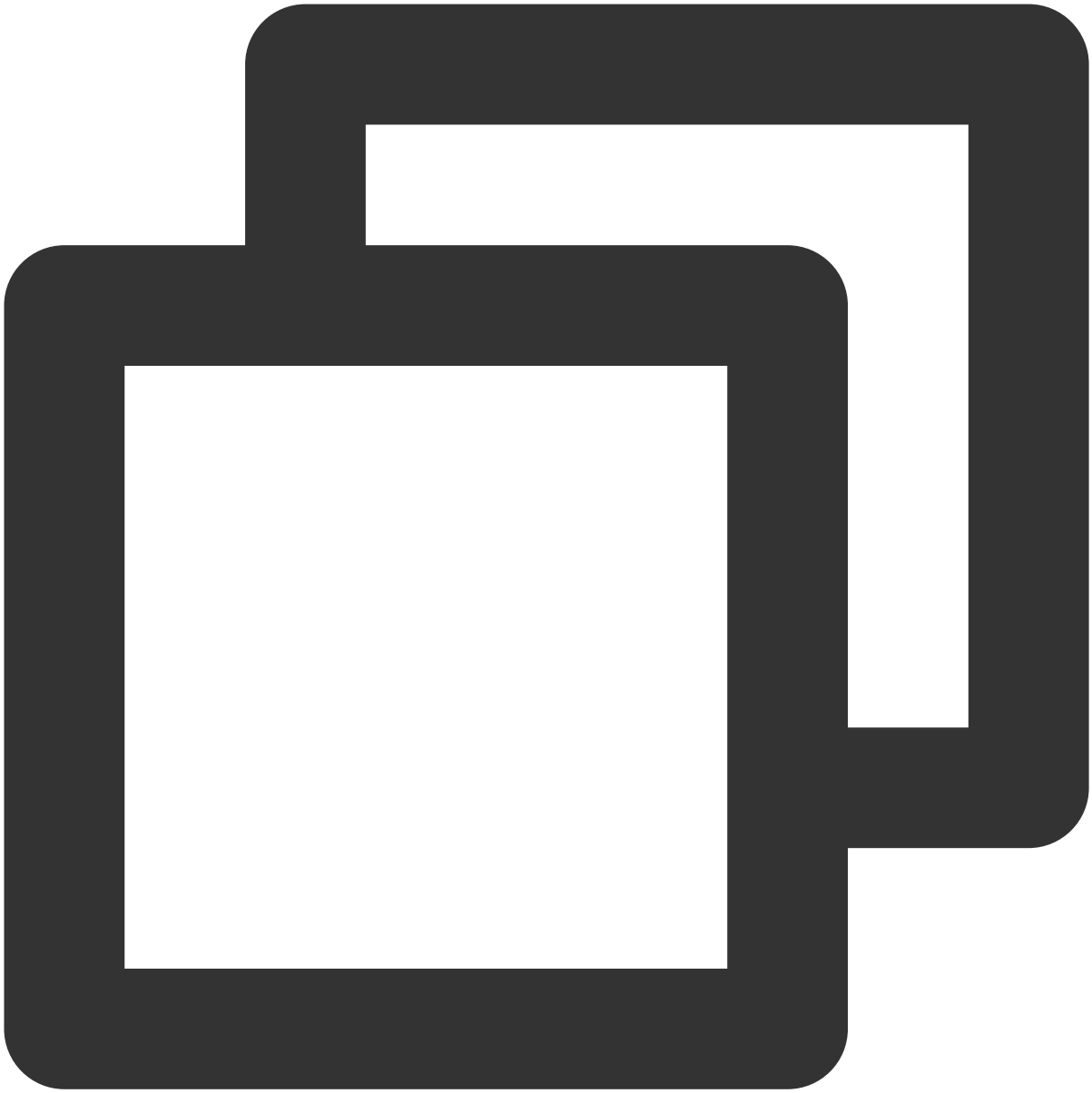
最近更新时间：2024-01-18 09:33:22

功能描述

对 Bucket 开通原图保护功能。

请求

请求示例



```
PUT /?origin-protect HTTP/1.1
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明：

Authorization：Auth String （详情参阅 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

该响应的响应体为空。

查询原图保护状态

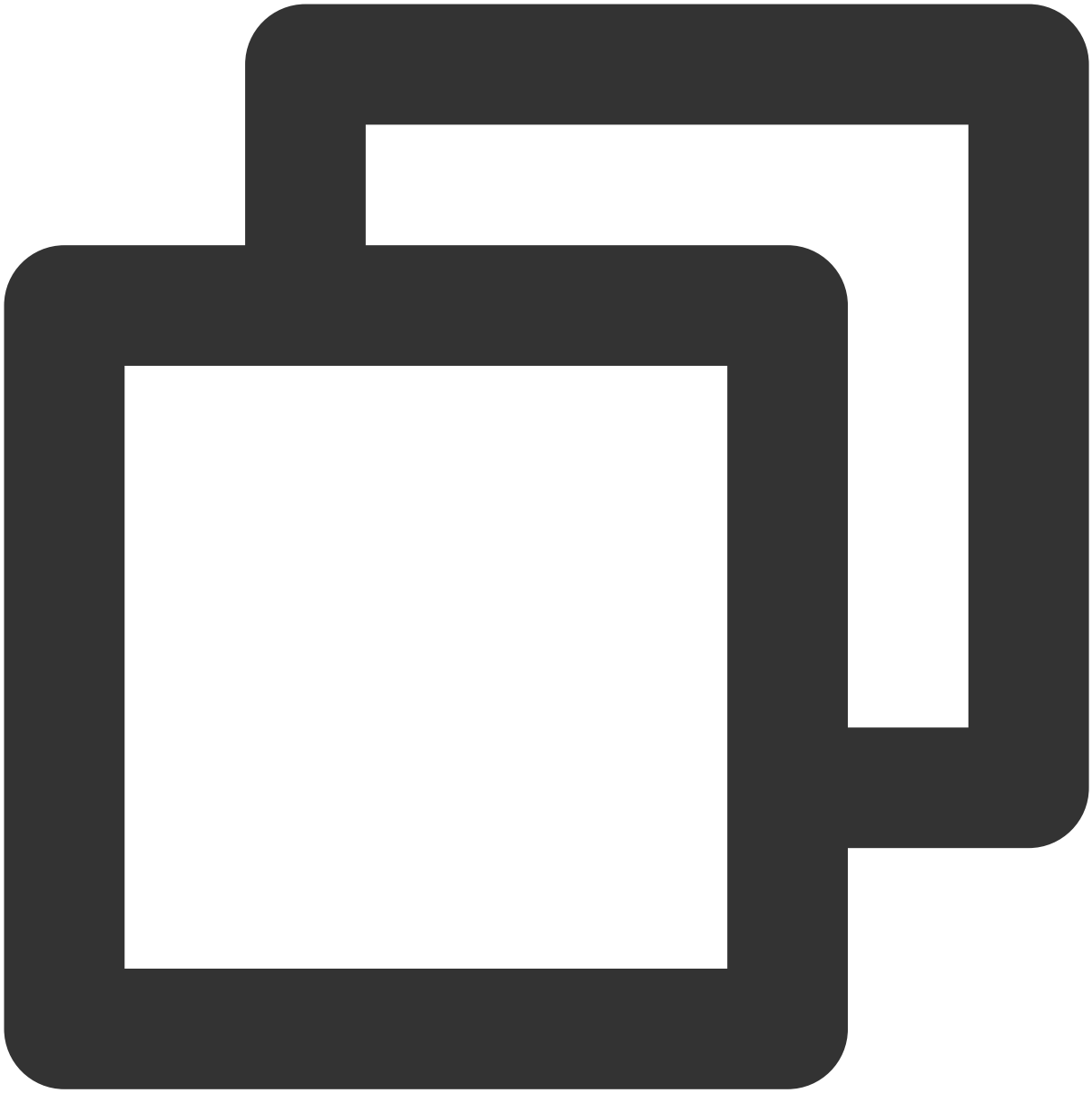
最近更新时间：2024-01-18 09:33:22

功能描述

用于查询原图保护功能是否开启。

请求

请求示例



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

说明：

Authorization：Auth String （详情参阅 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

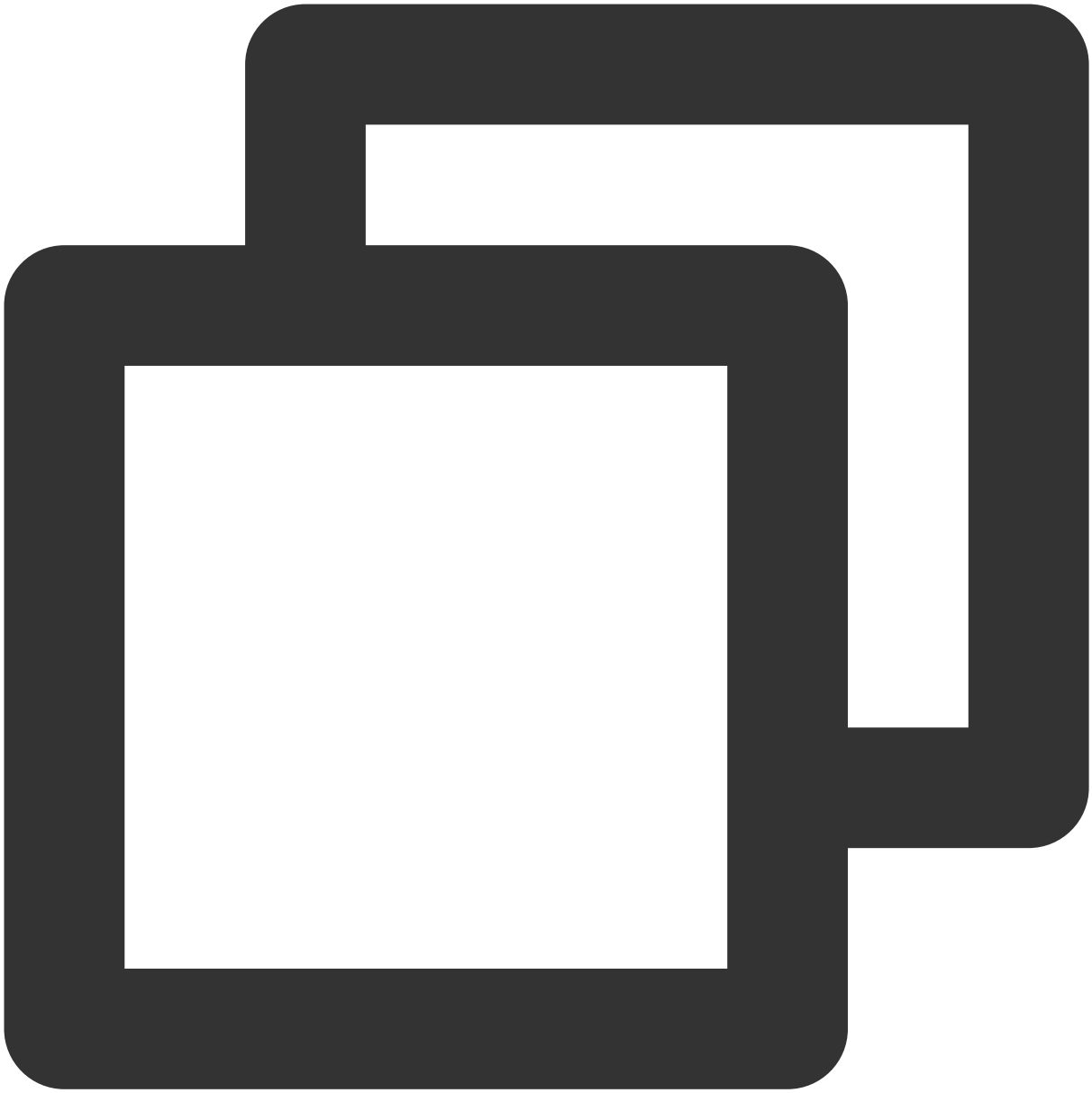
该请求的请求体为空。

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



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

节点名称（关键字）	描述
OriginProtectStatus	on 与 off 两种状态。

关闭原图保护

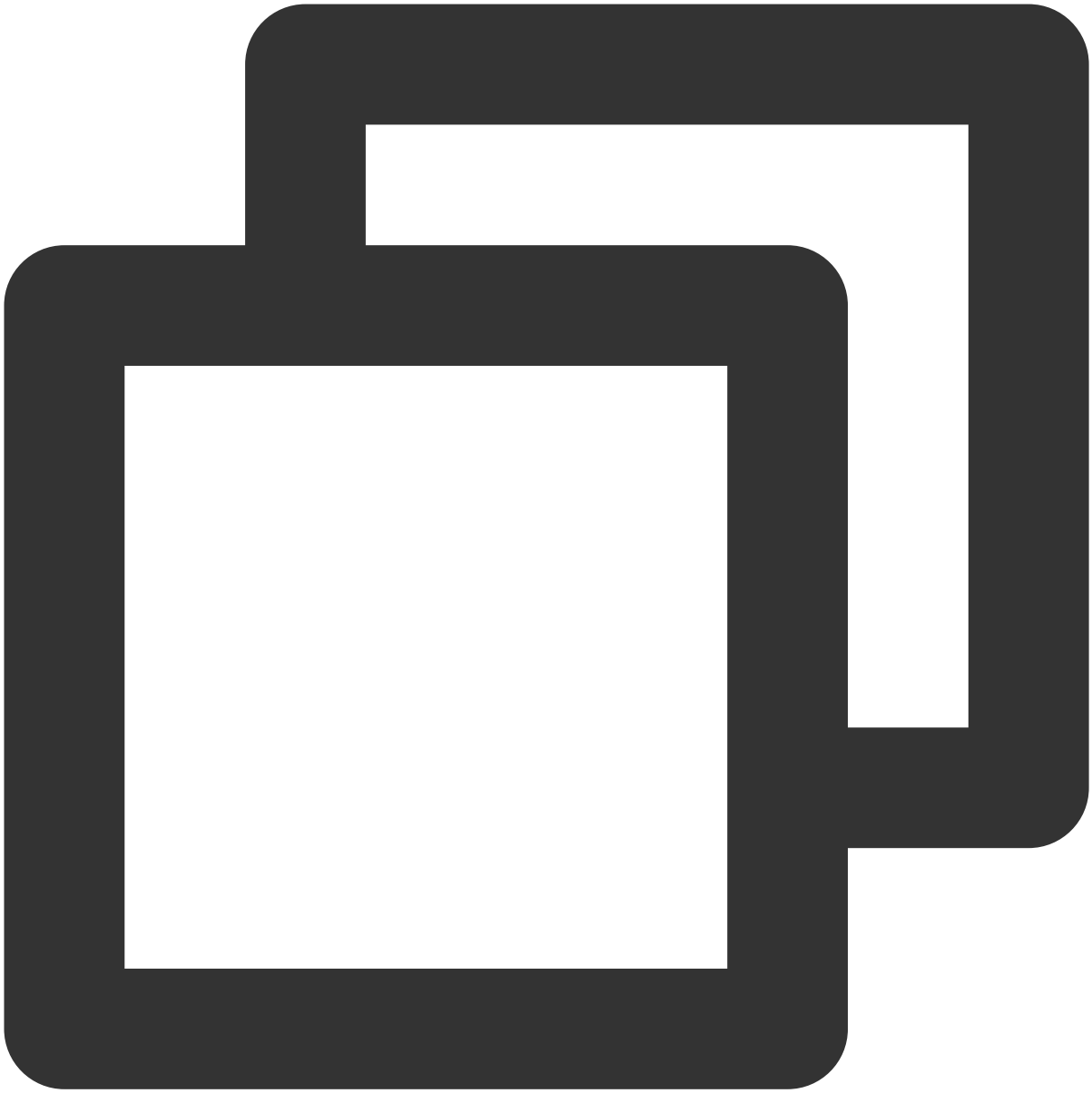
最近更新时间：2024-01-18 09:33:22

功能描述

用于关闭原图保护功能。

请求

请求语法



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

说明：

Authorization：Auth String （详情参阅 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

该响应的响应体为空。

图片处理
基础图片处理
缩放

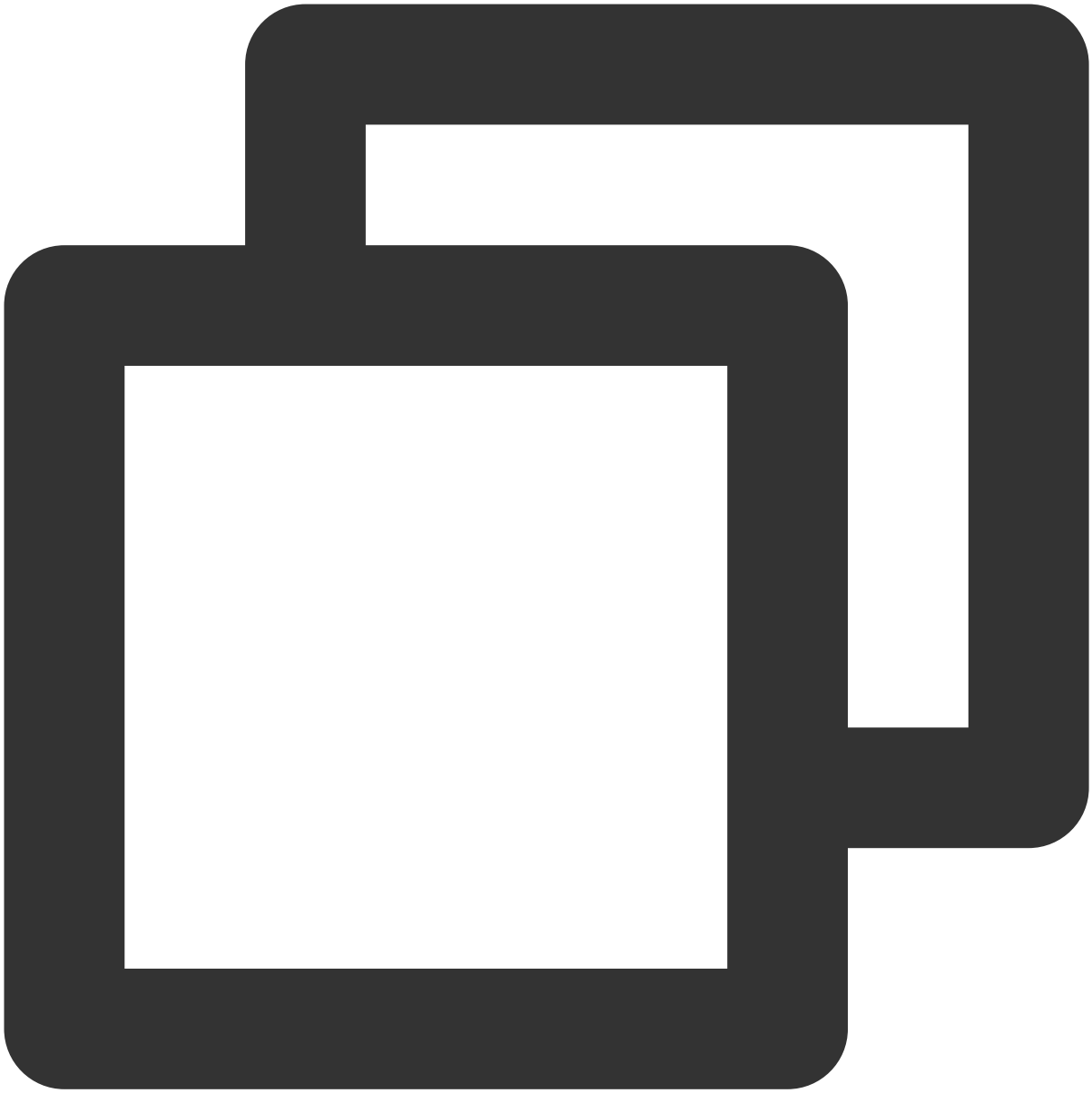
最近更新时间：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **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

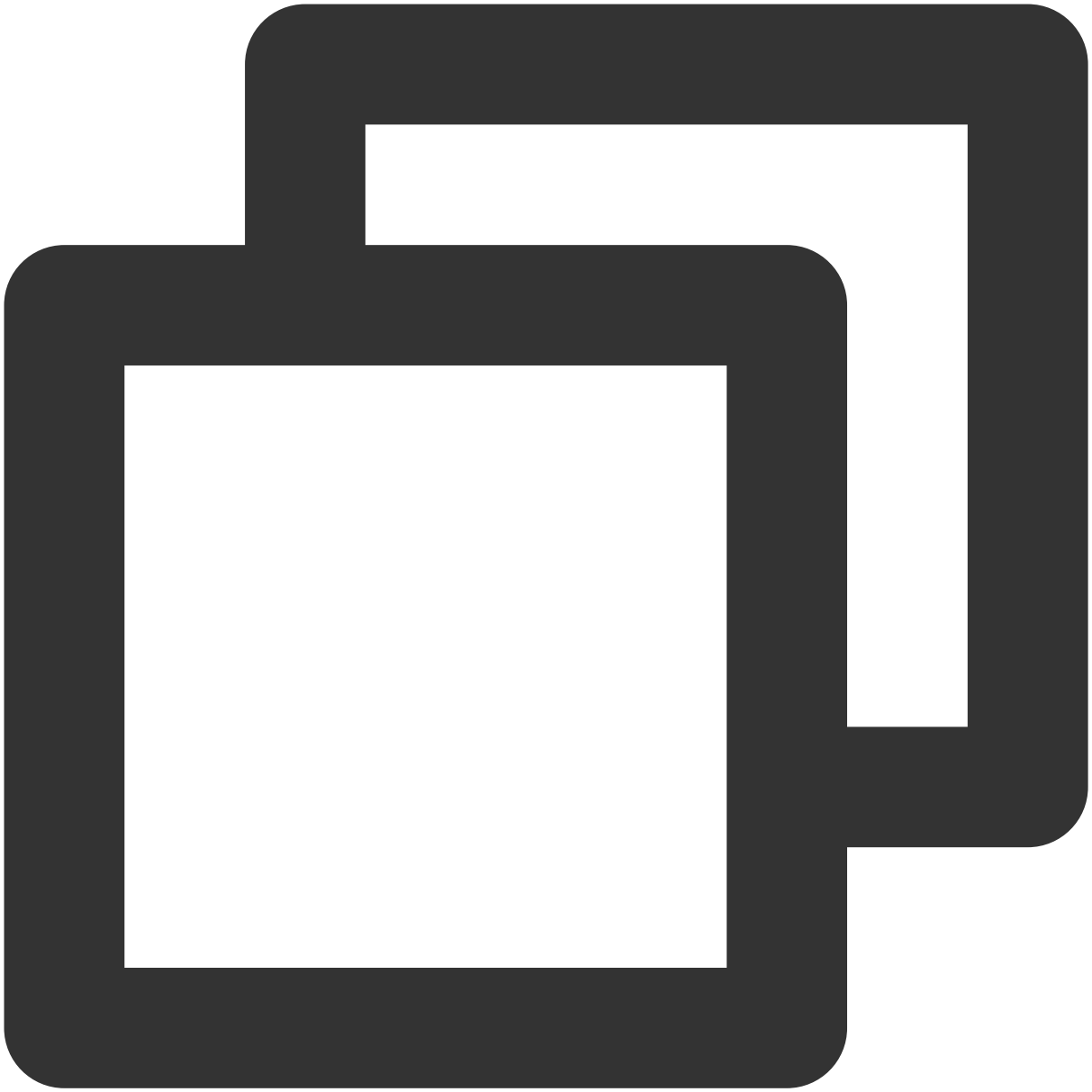
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/thumbnail/<imageSizeAndOffsetGeometry> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



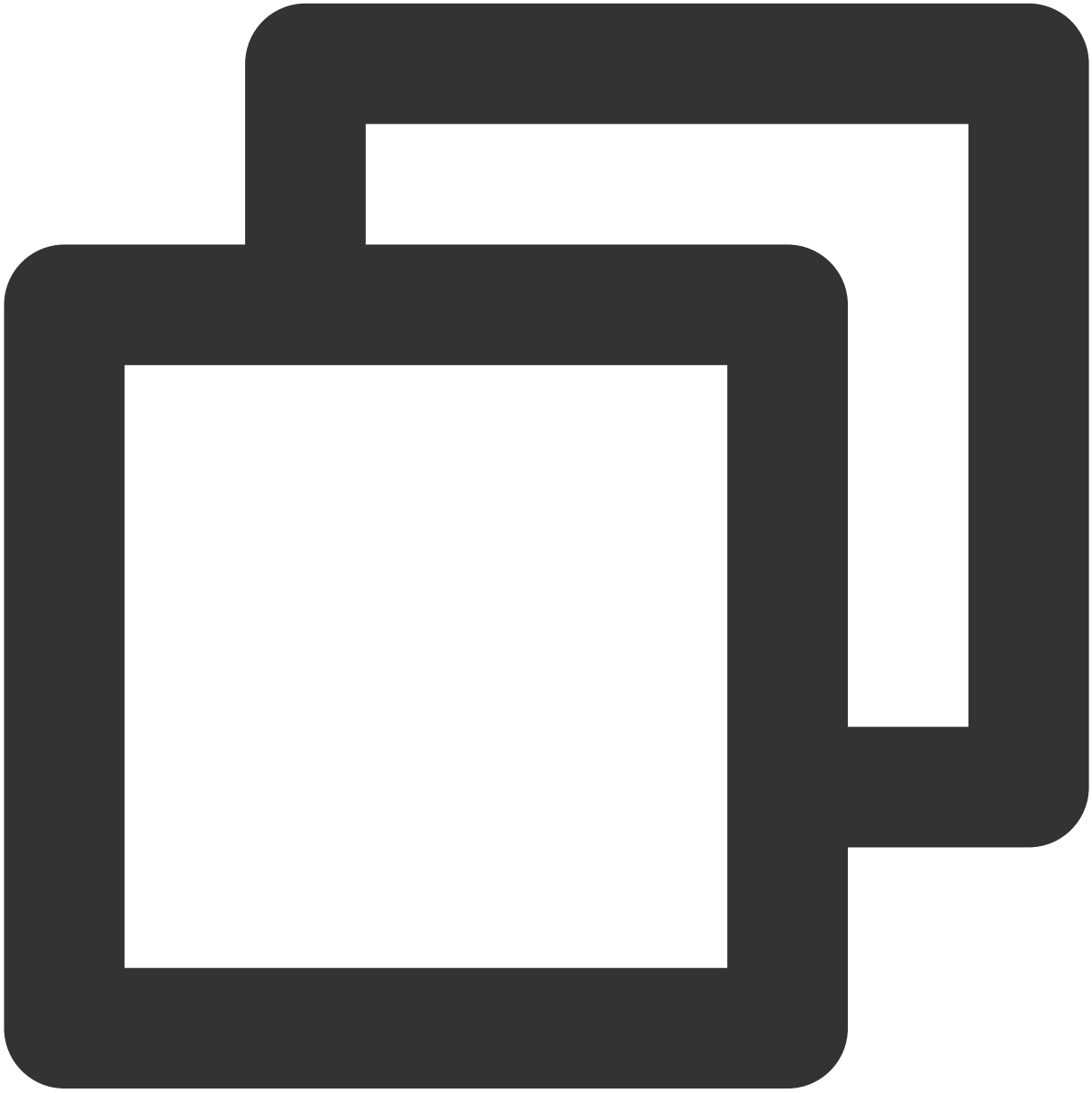
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/thumbnail/<imageSizeAndOffsetGeometry>"
  }]
```

```
}
```

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/thumbnail/<imageSizeAndOffsetGeometry>"
  }]
}
```

处理参数说明

操作名称：thumbnail。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/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

版权所有：腾讯云计算（北京）有限责任公司

/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，缺省为白色，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 #FFFFFF
/ignore-error/1	当处理参数中携带此参数时，针对文件过大、参数超限等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：

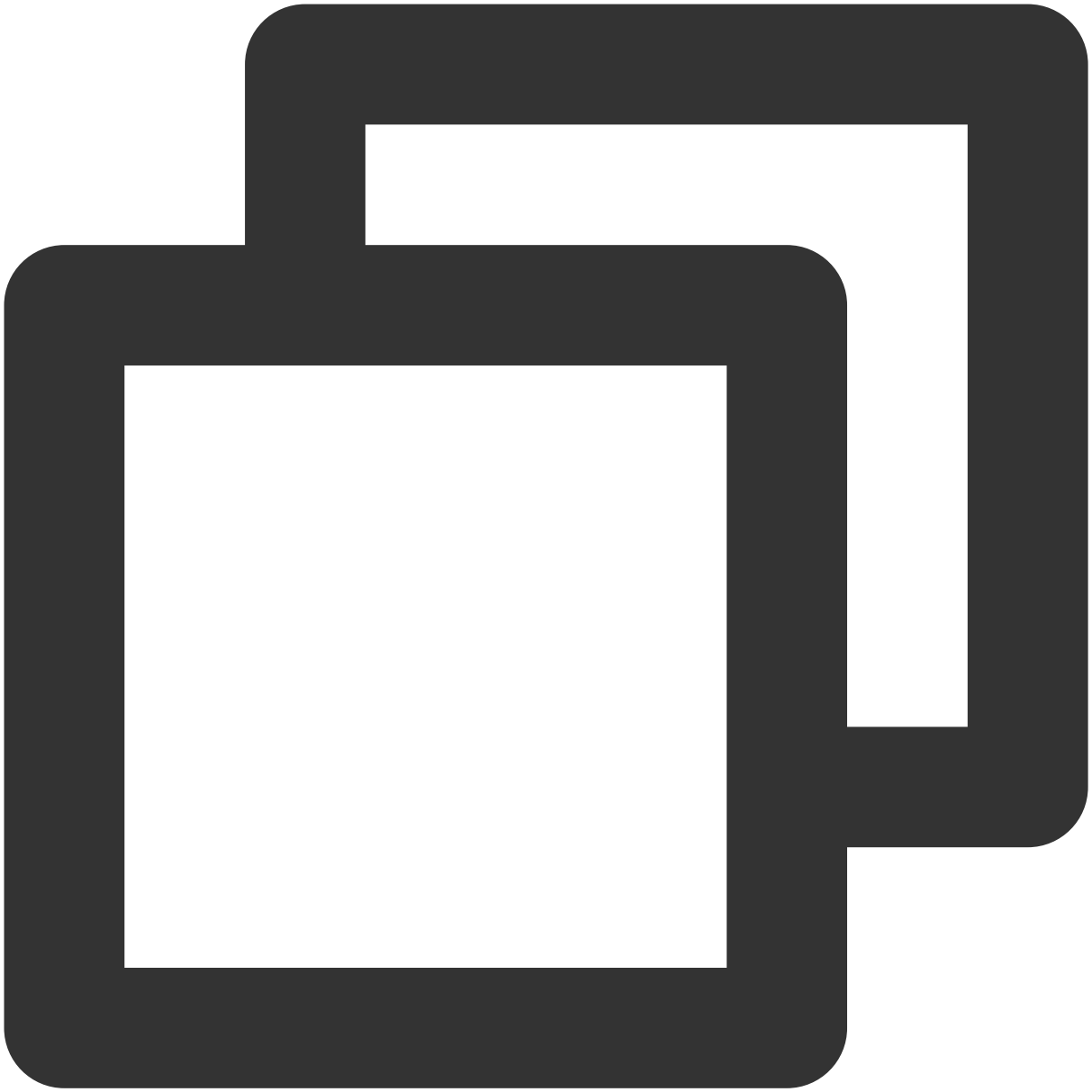
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，您可查阅 图片持久化处理 文档并配置**上传时处理**或**云上数据处理**。

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：



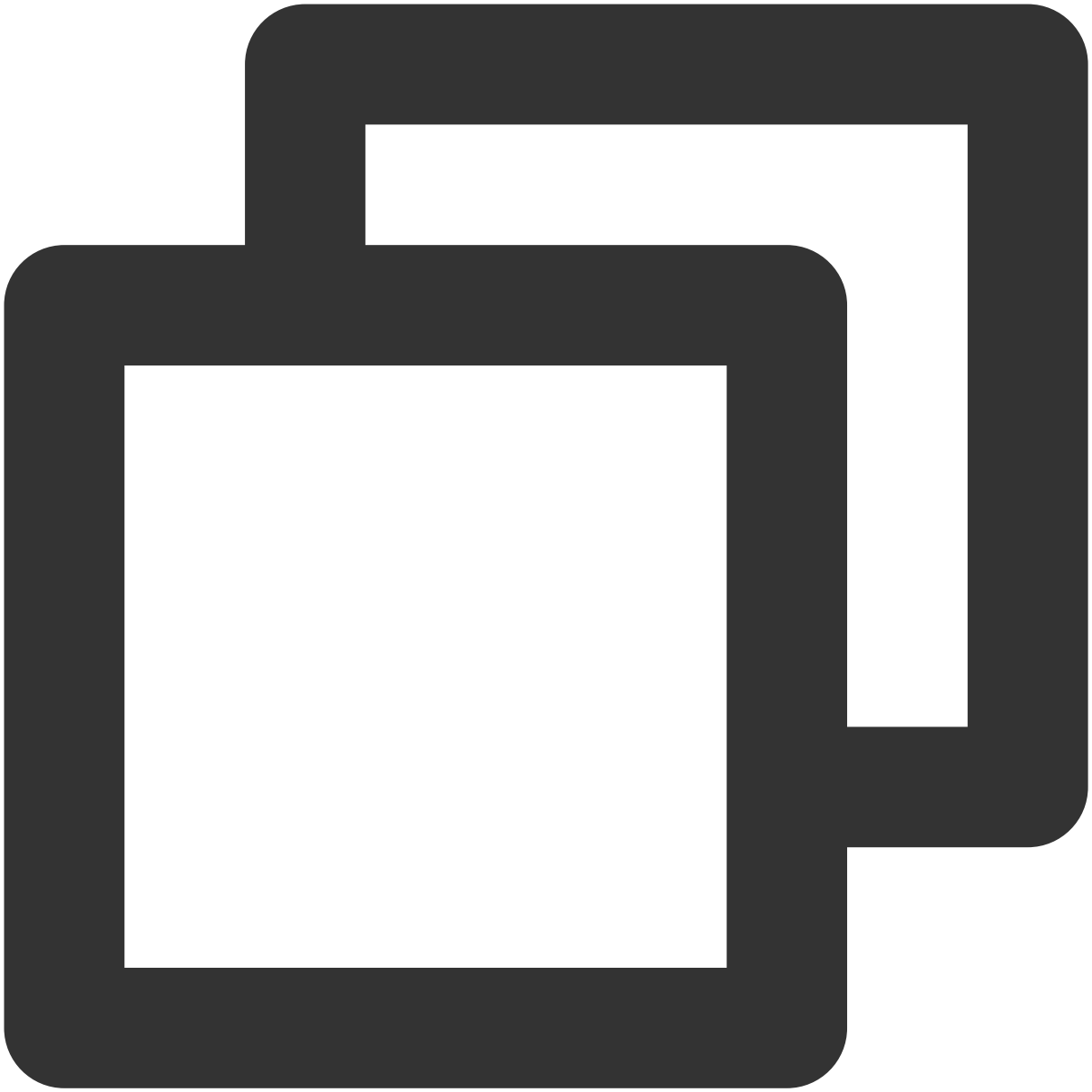
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

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：



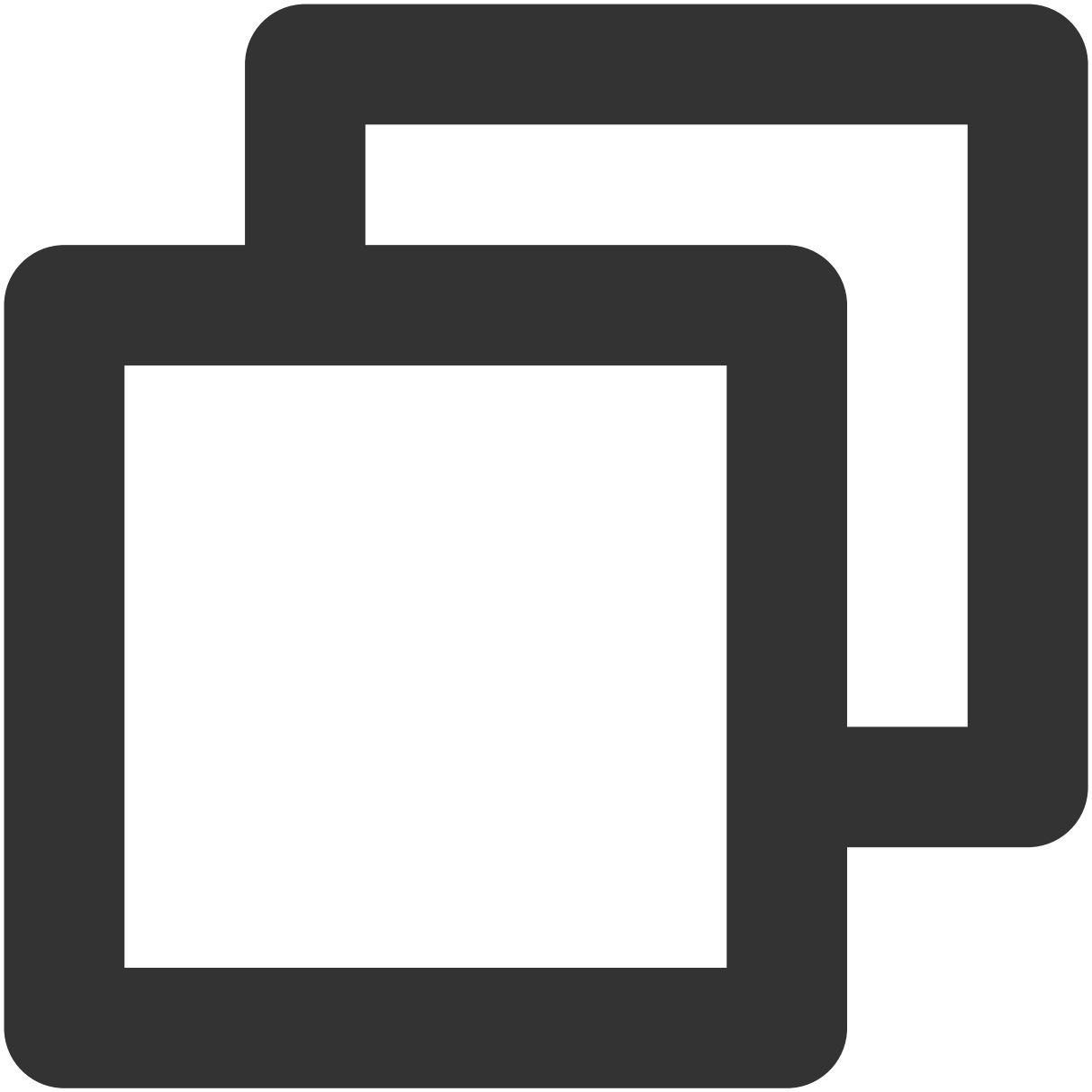
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

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：



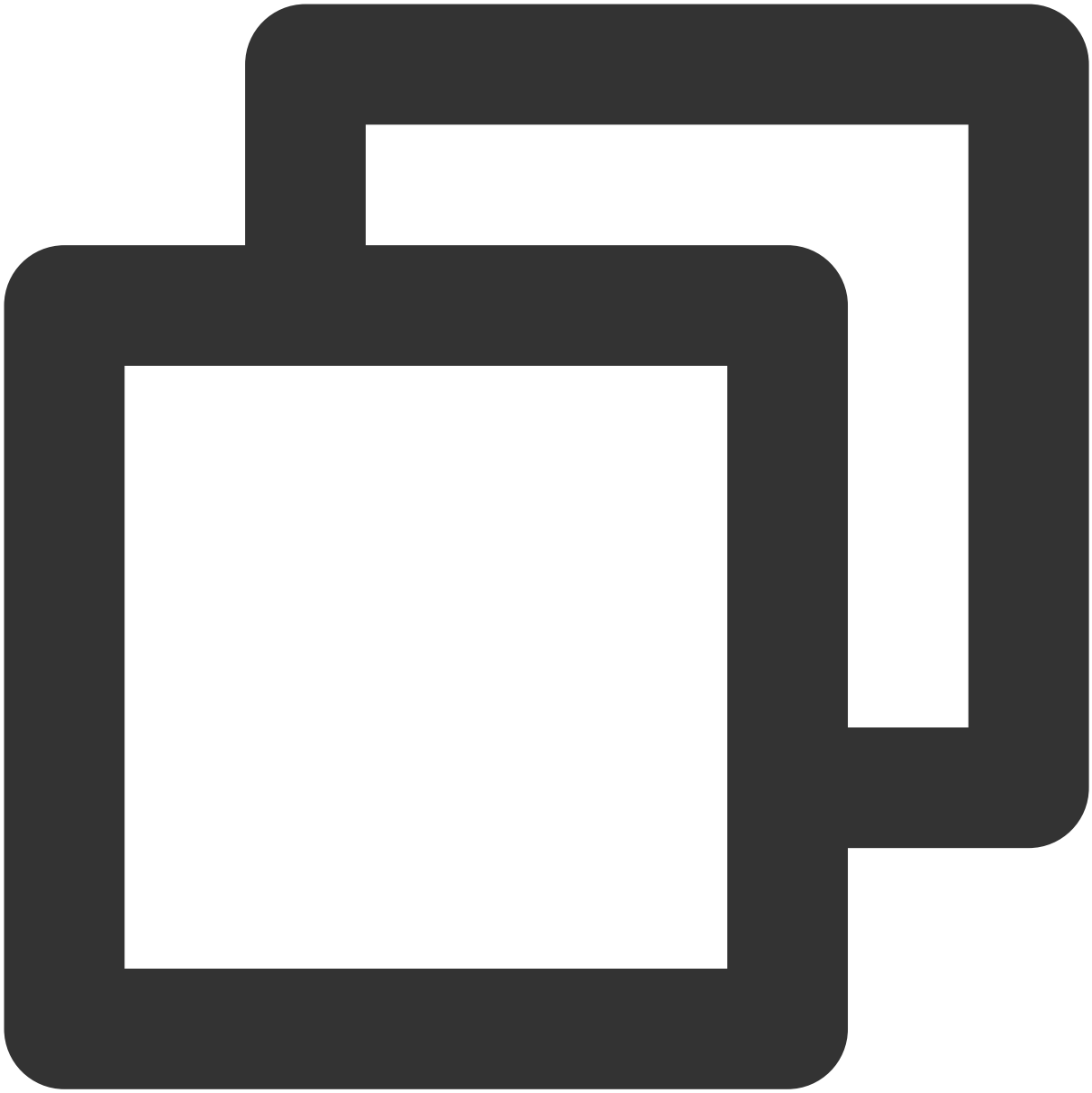
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

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：



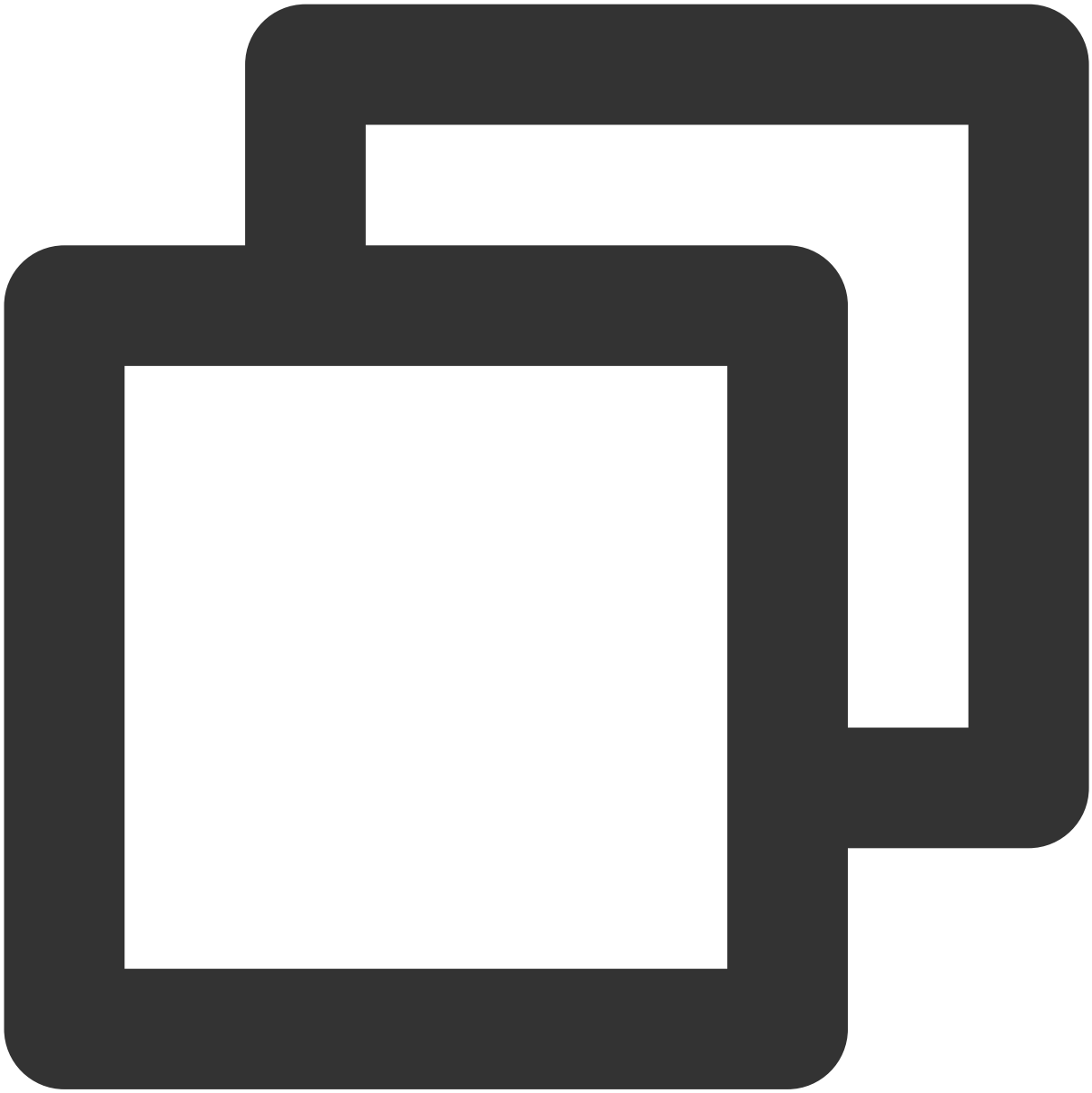
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供裁剪功能，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

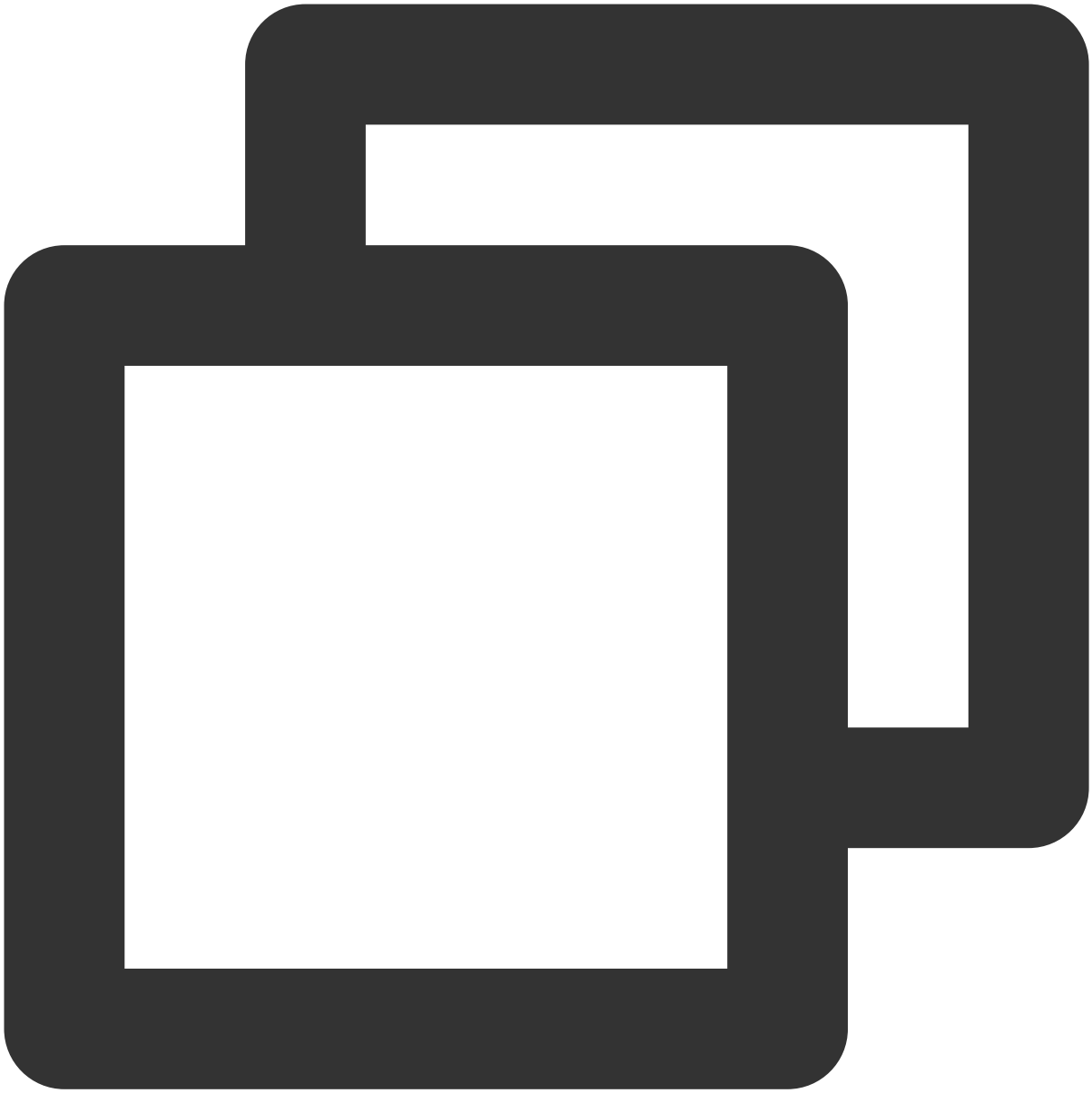
限制说明

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

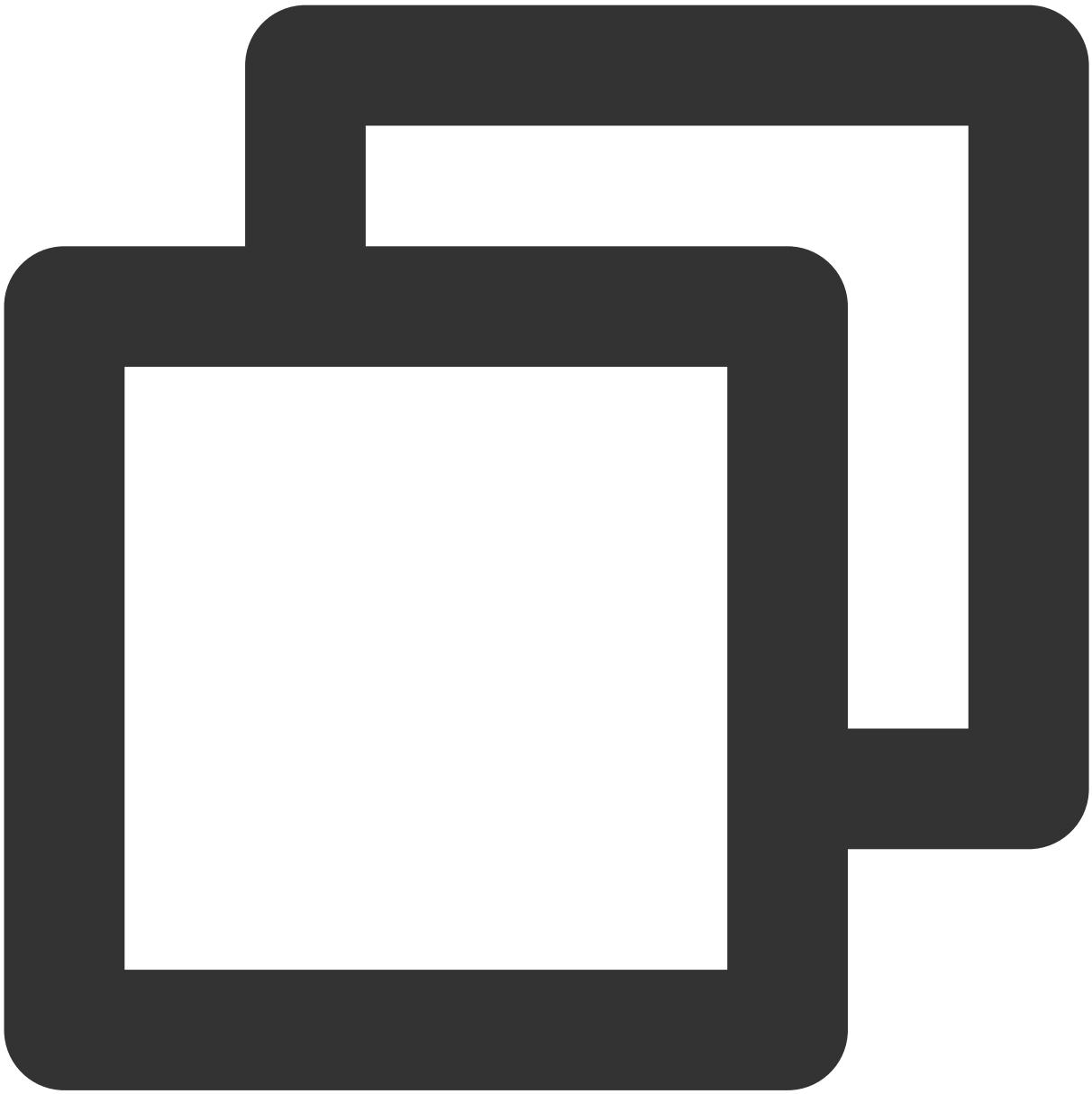


```
GET /<ObjectKey>?imageMogr2/cut/<width>x<height>x<dx>x<dy>
                           /crop/<imageSizeAndOffsetGeometry>
                           /iradius/<radius>
                           /rradius/<radius>
                           /scrop/<Width>x<Height> HTTP/1.1
Host: <BucketName-APPID>.cos.<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



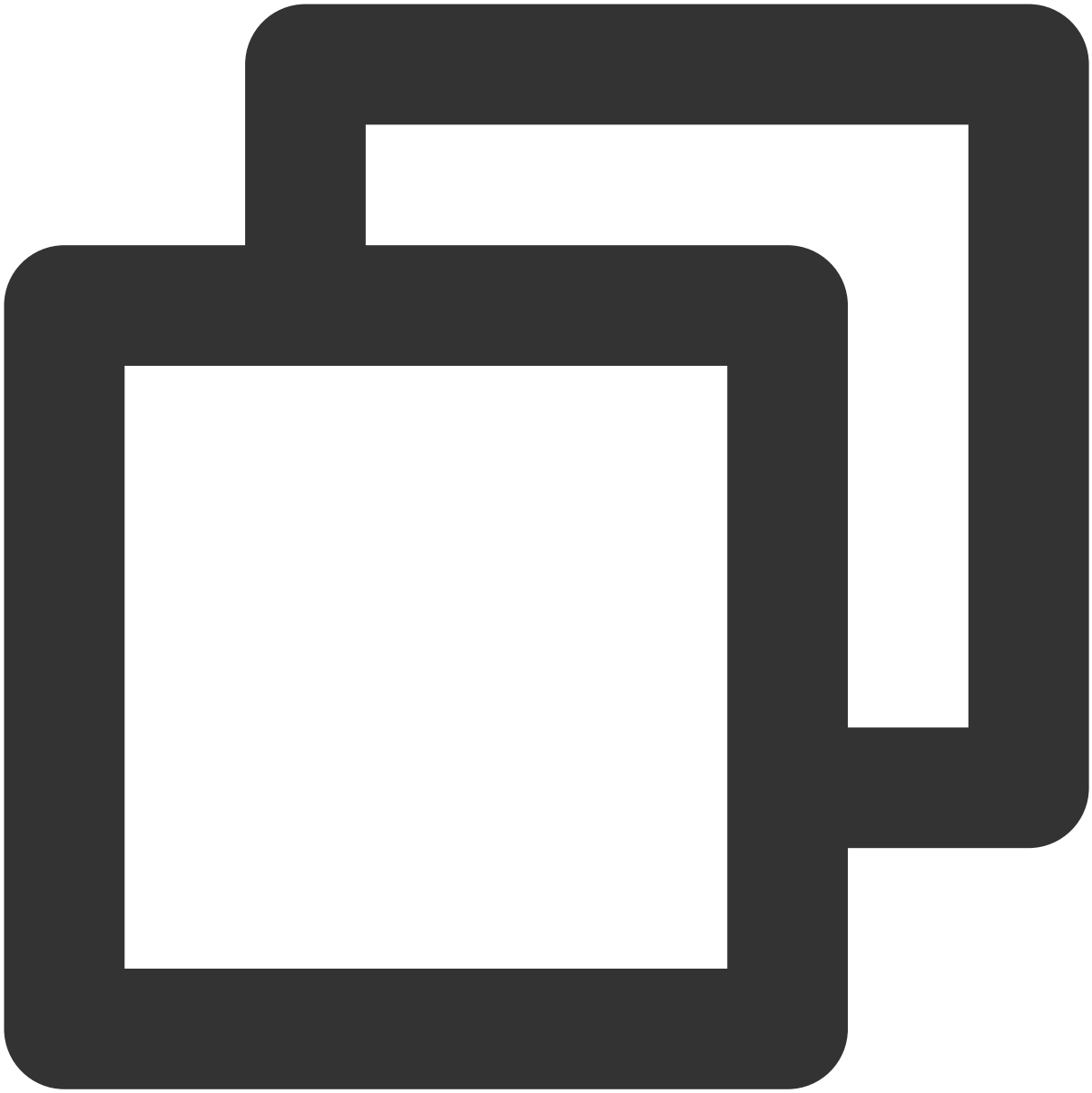
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
```

```
   "rules": [{
       "fileid": "exampleobject",
       "rule": "imageMogr2/cut/<width>x<height>x<dx>x<dy>
                        /crop/<imageSizeAndOffsetGeometry>
                        /iradius/<radius>
                        /rradius/<radius>
                        /scrop/<Width>x<Height>"
   }]
}
```

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/cut/<width>x<height>x<dx>x<dy>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                        /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）。

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。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大、参数超限等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。

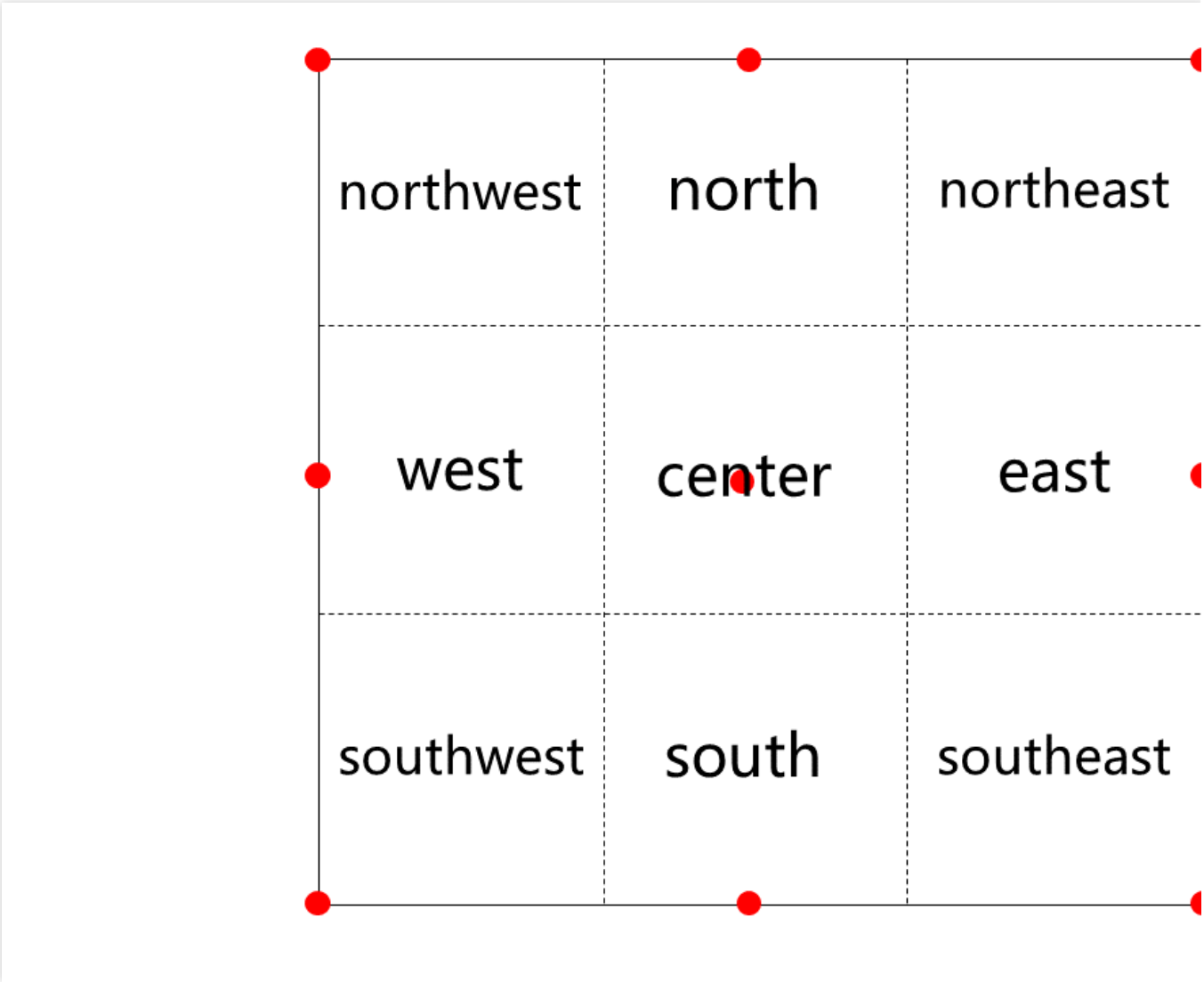
人脸智能裁剪参数说明

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

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）。

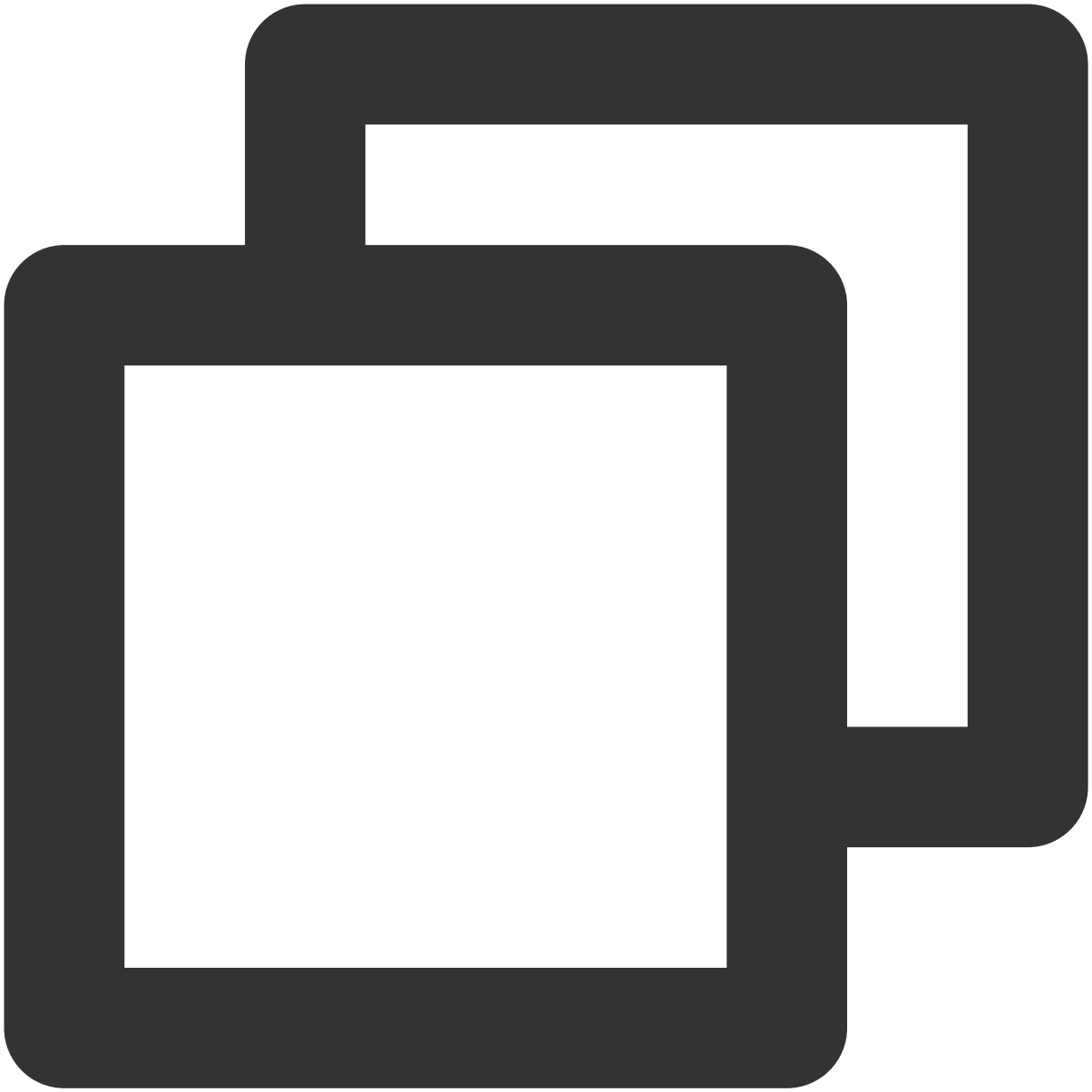
实际案例

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：



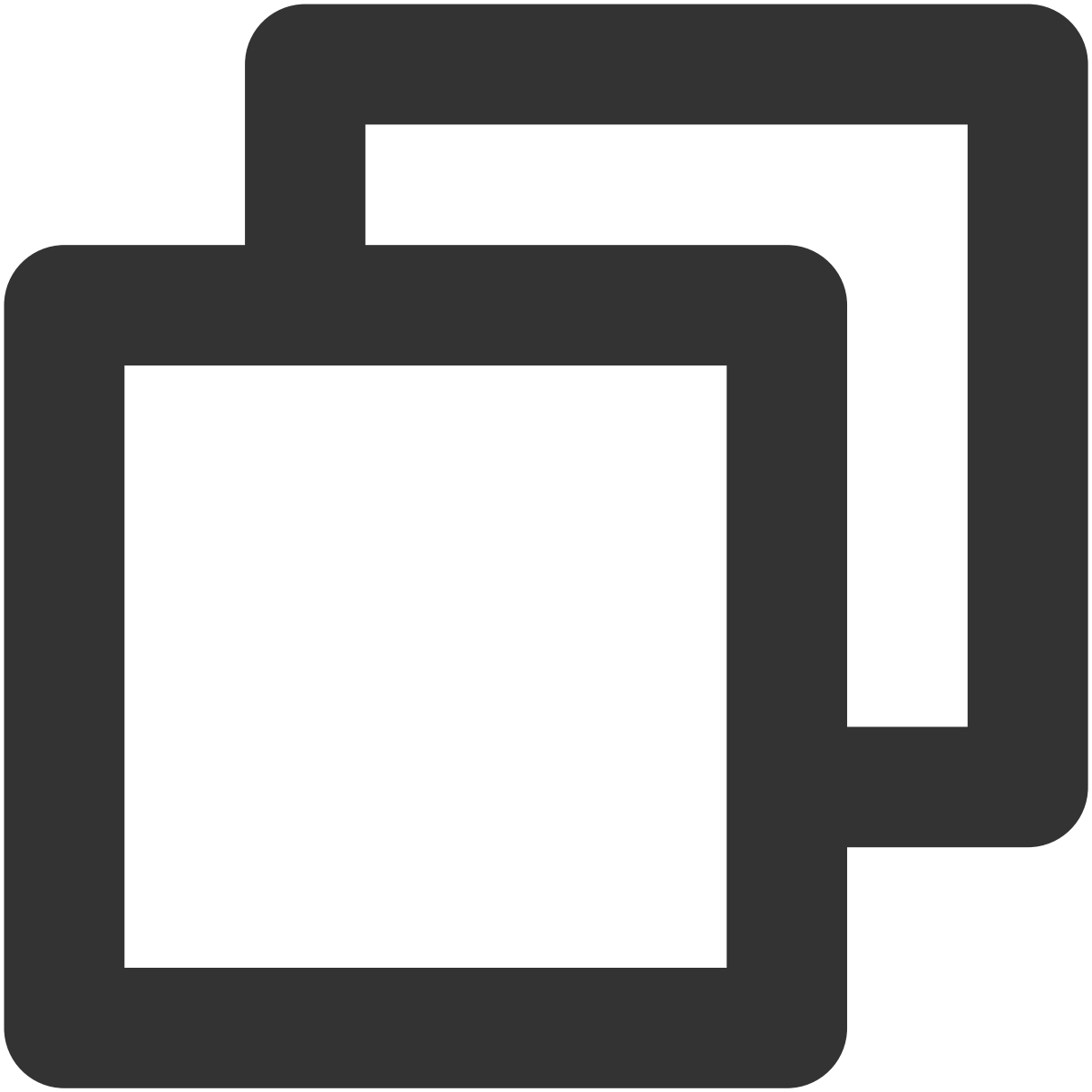
最终效果如下：

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：



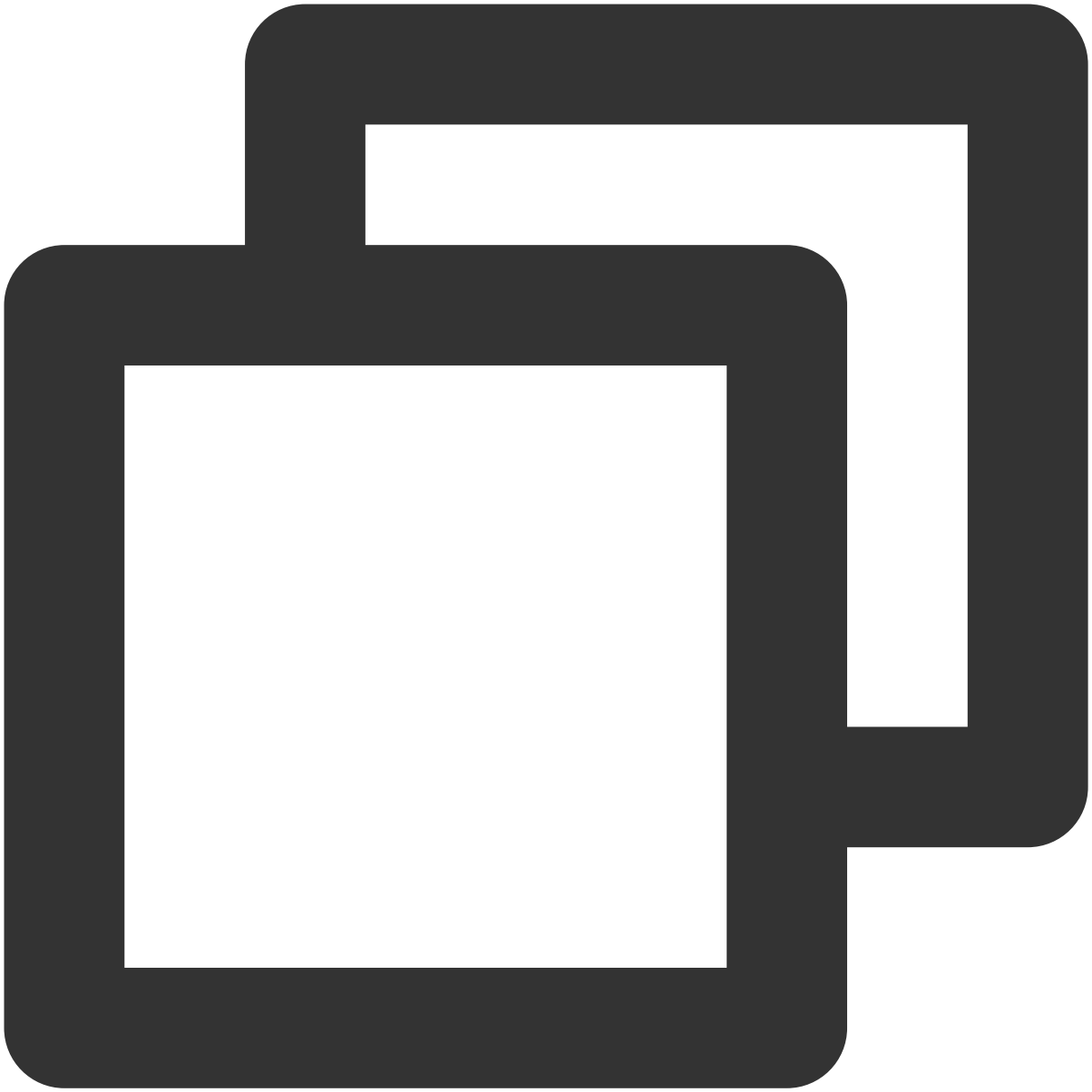
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/crop
```

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：



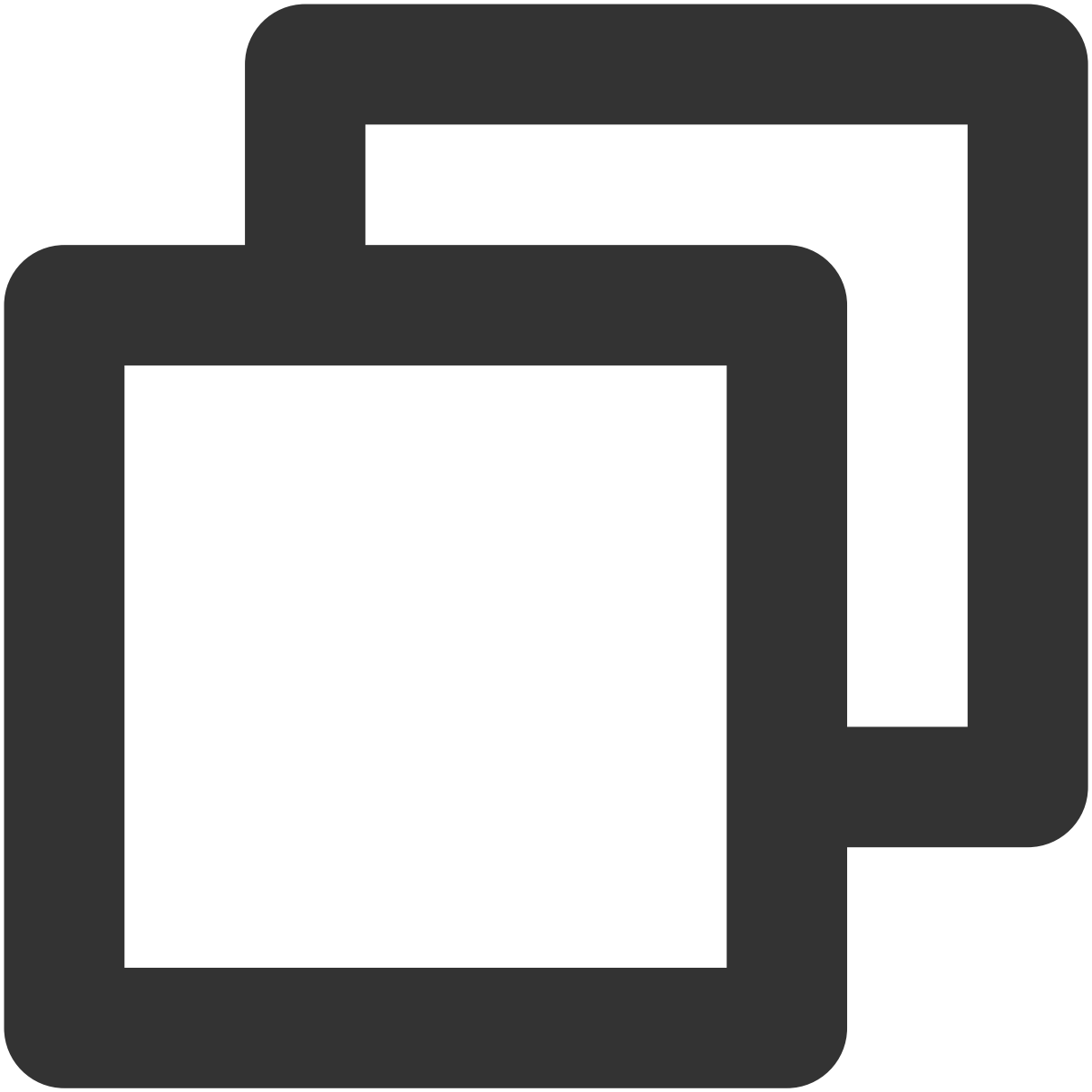
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/irad
```

最终效果如下：



案例四：圆角裁剪

取半径为100，进行圆角裁剪：



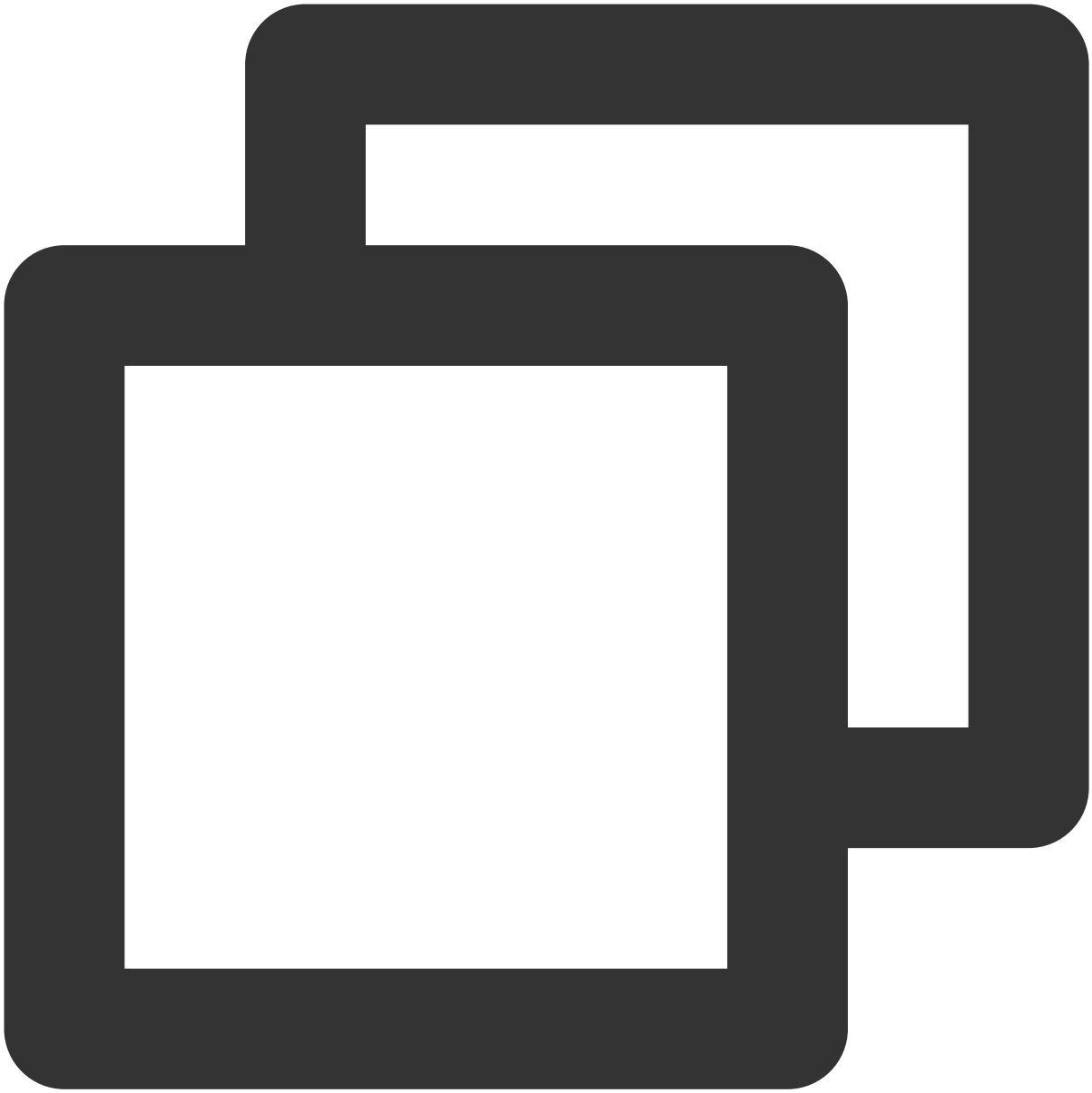
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/rrad
```

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：



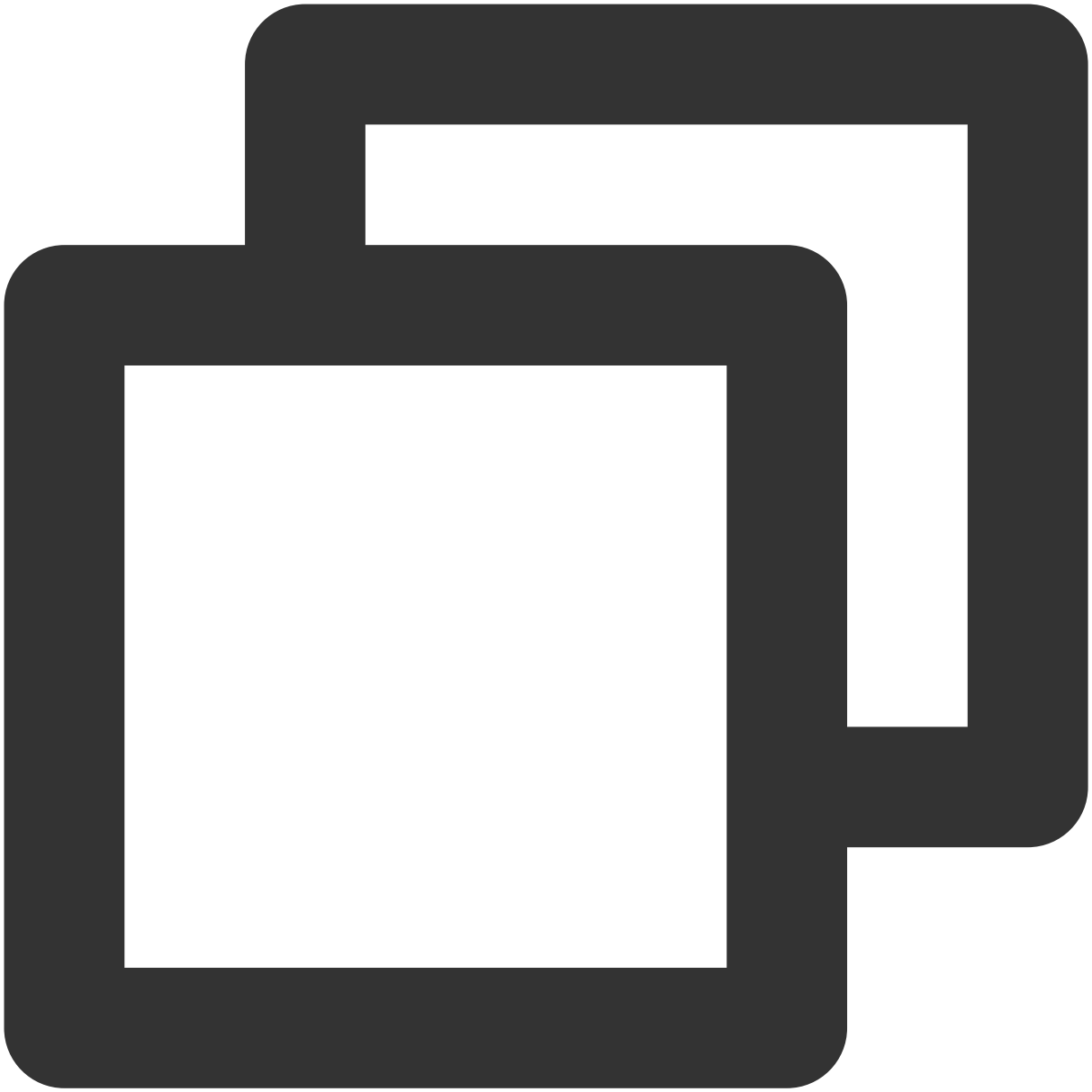
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/scro
```

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：



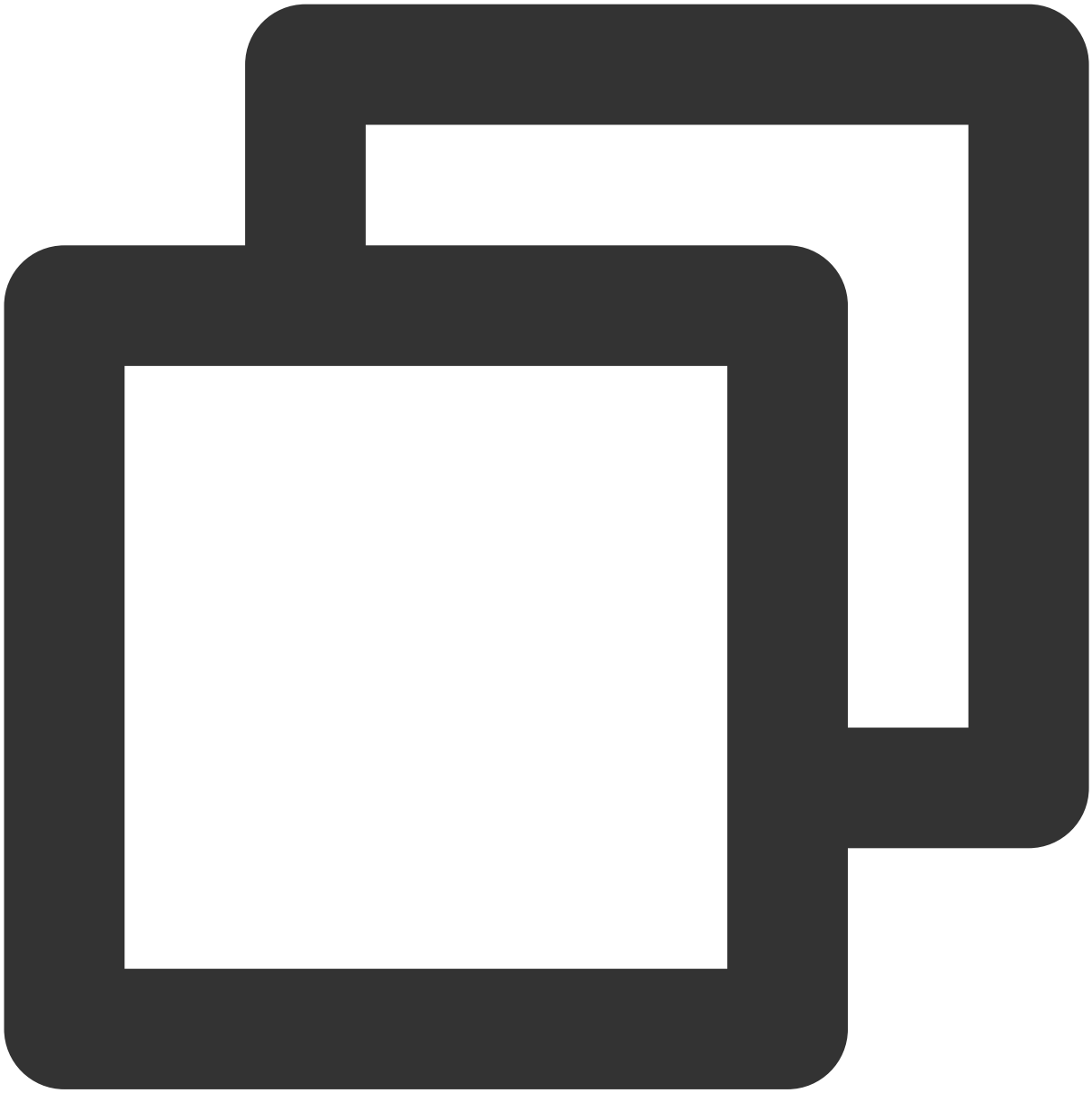
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **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

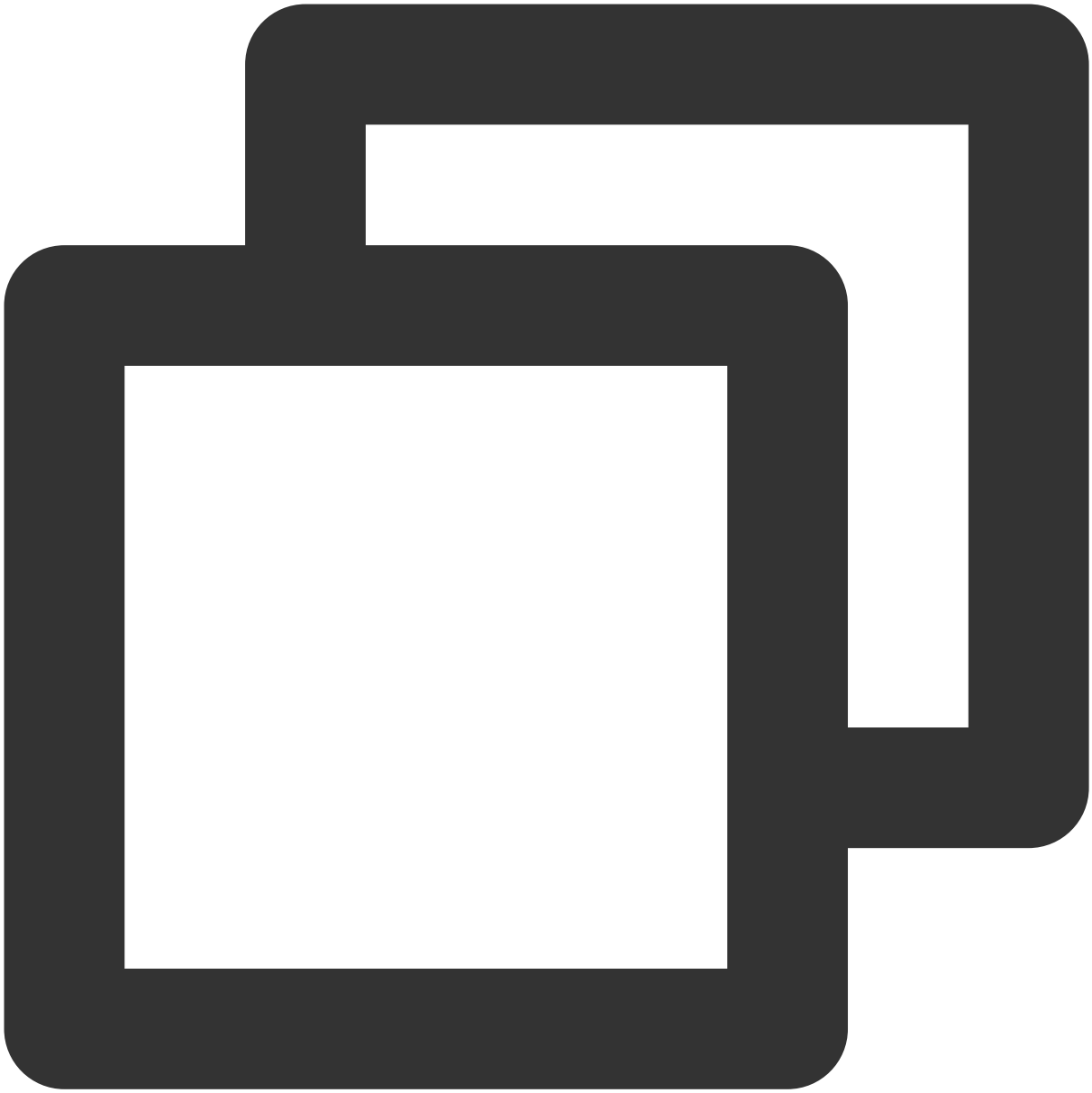
限制说明

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

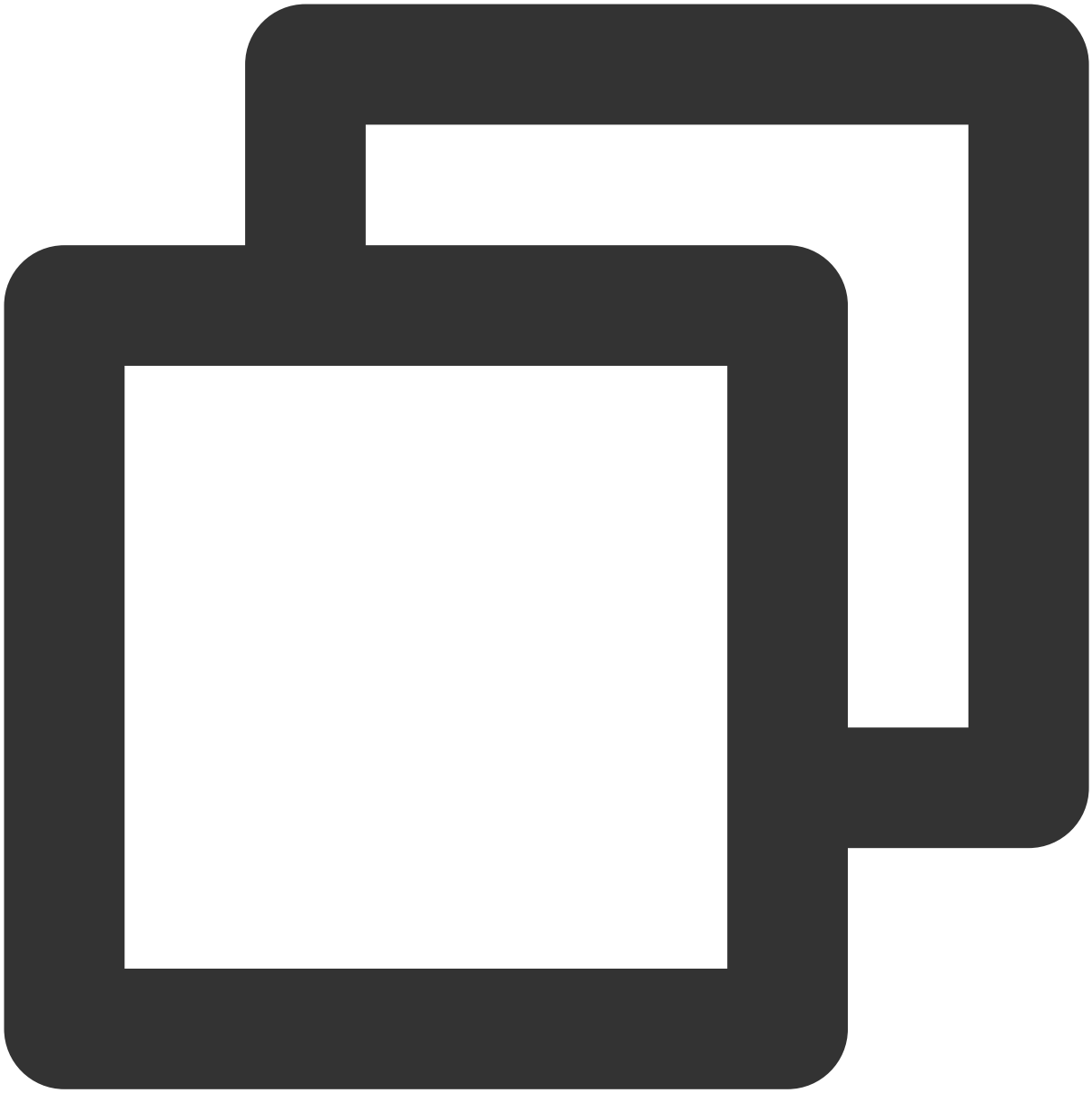


```
GET /<ObjectKey>?imageMogr2/rotate/<rotateDegree>
                /auto-orient HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

注意：

请忽略上面的空格与换行符。

2. 上传时处理



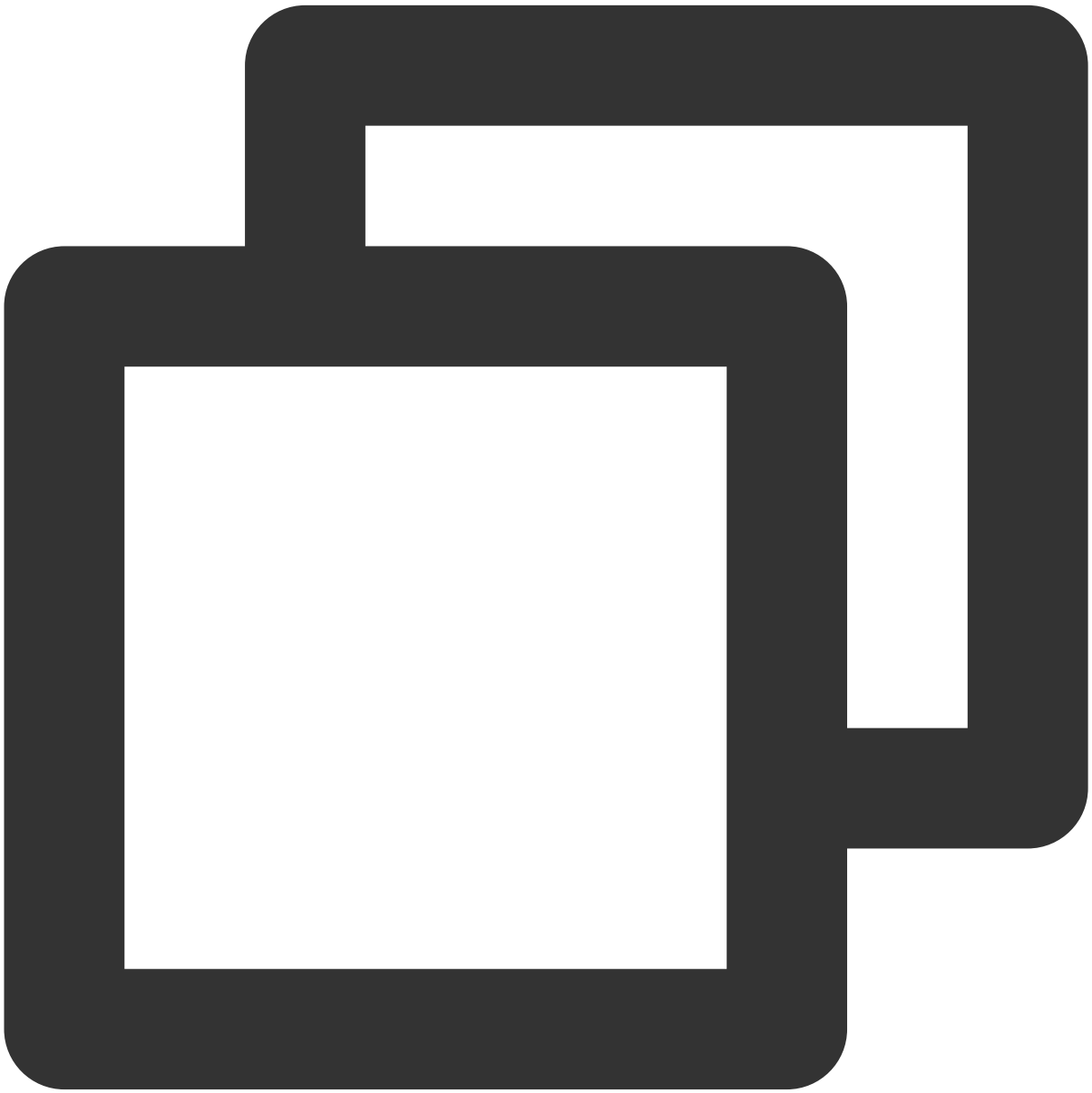
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/rotate/<rotateDegree>
                      /auto-orient"
```

```
    }]
}
```

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/rotate/<rotateDegree>
                      /auto-orient"
  }]
}
```

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。
/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、参数超限等导致处理失败的场景，会直接返回原图而不报错

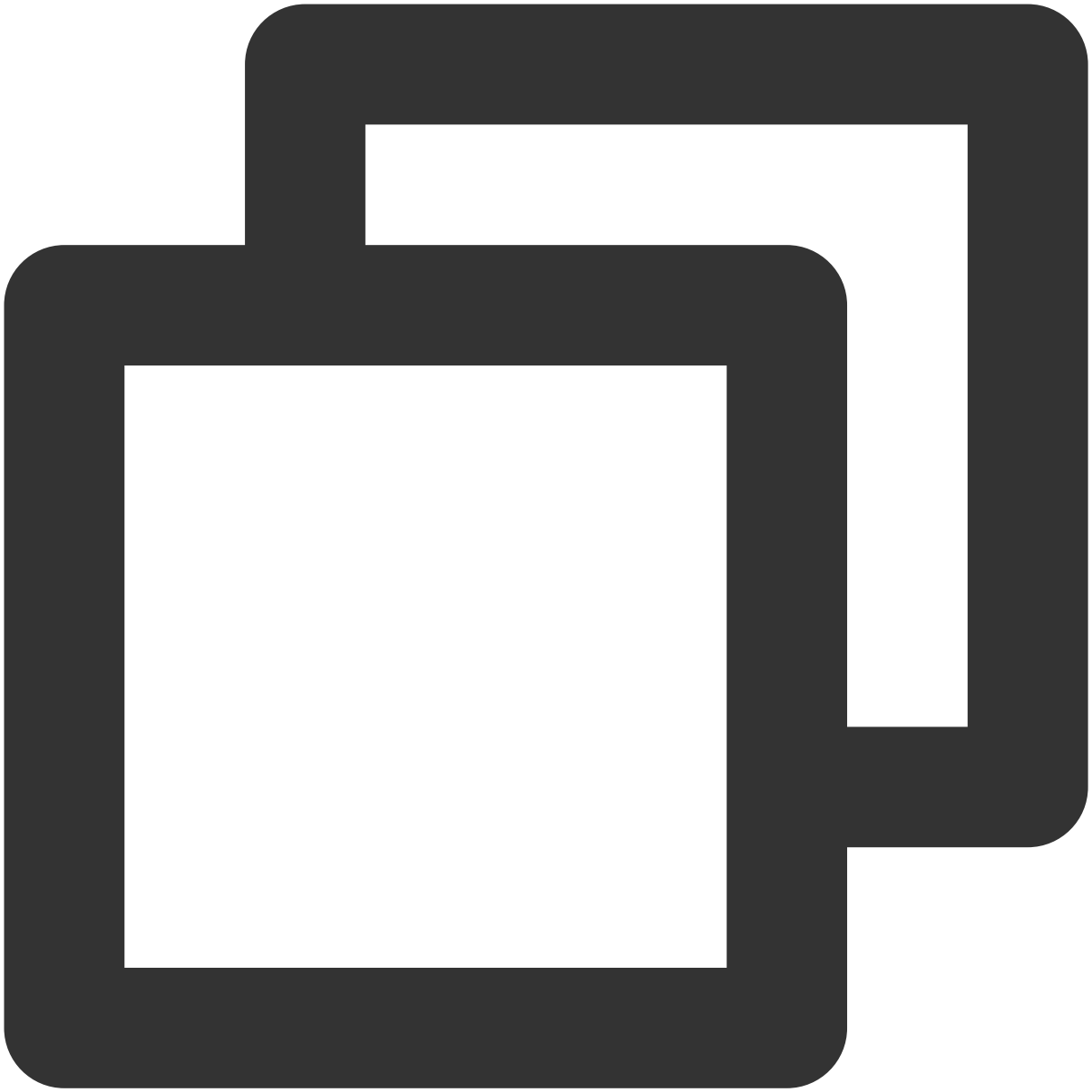
实际案例

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
```

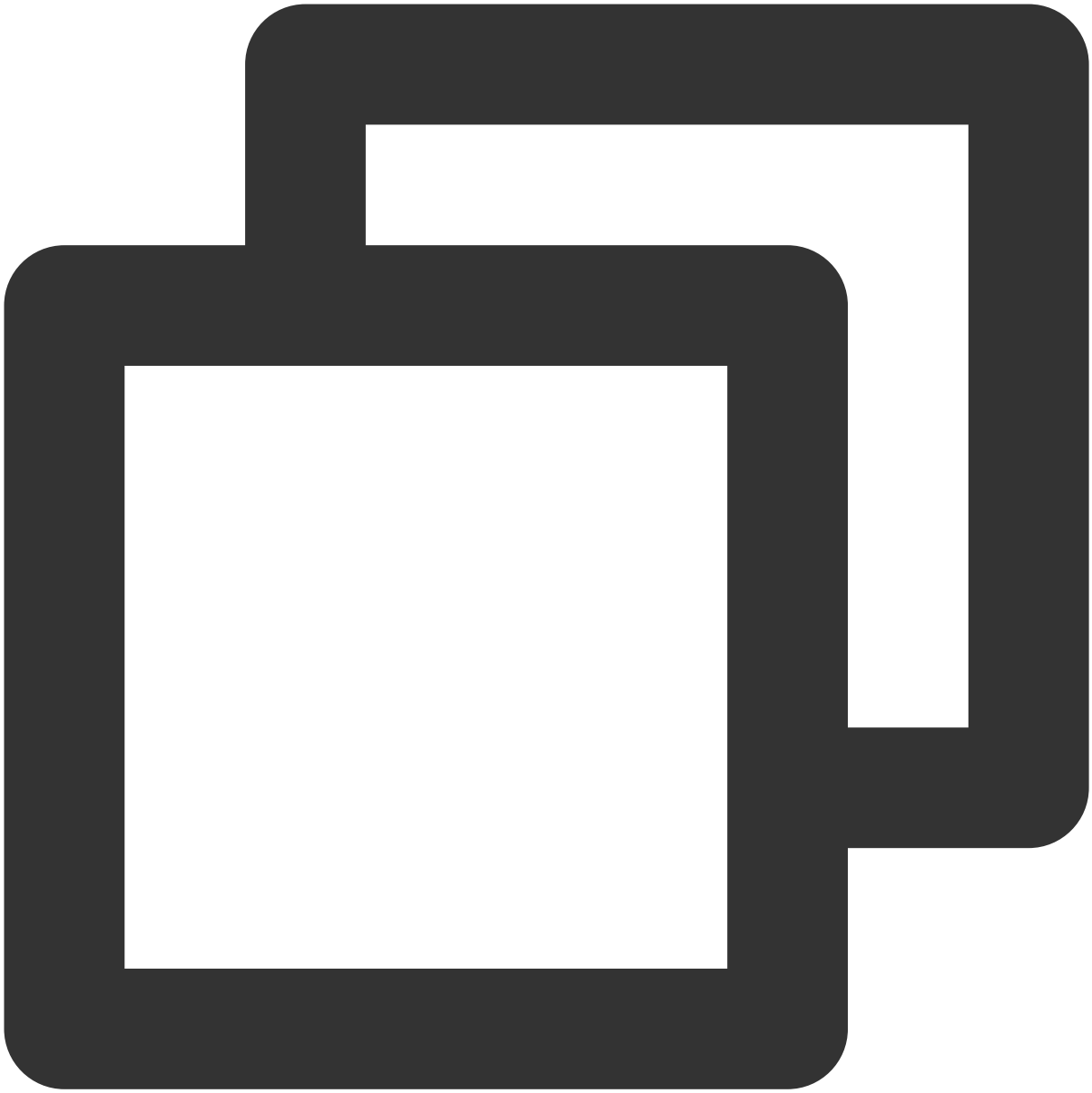
最终效果如下：





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：



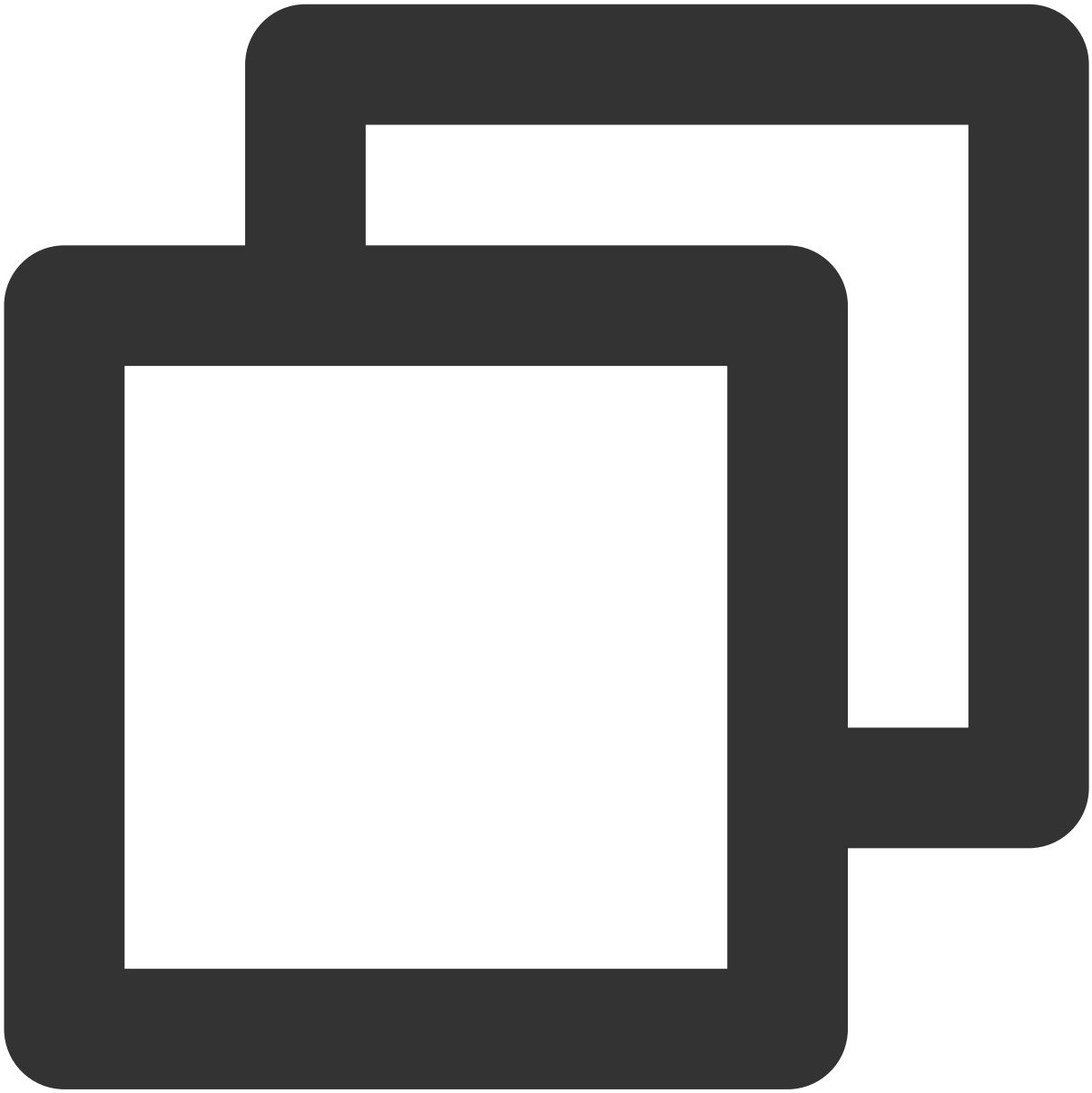
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供格式转换、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

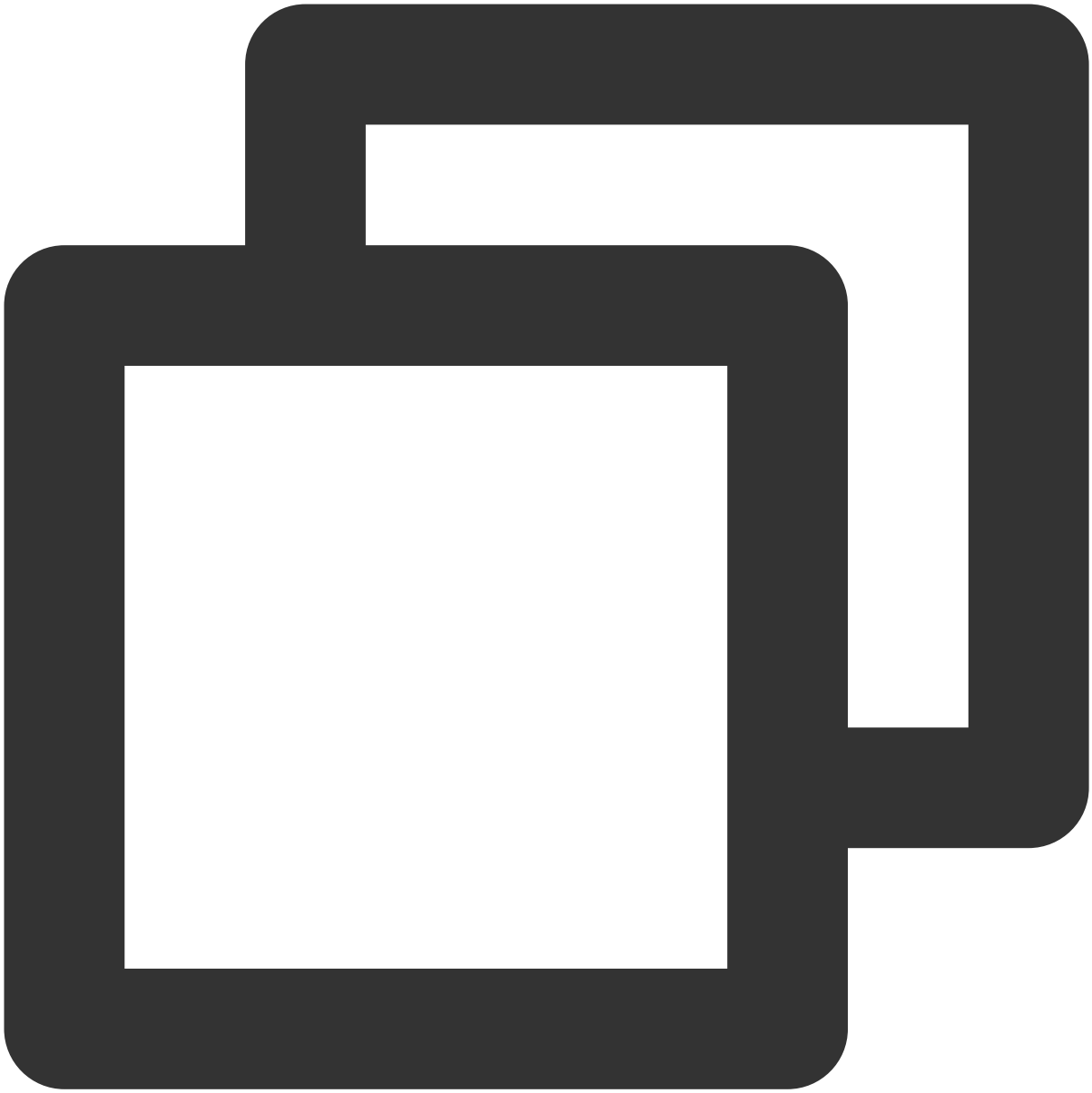
限制说明

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

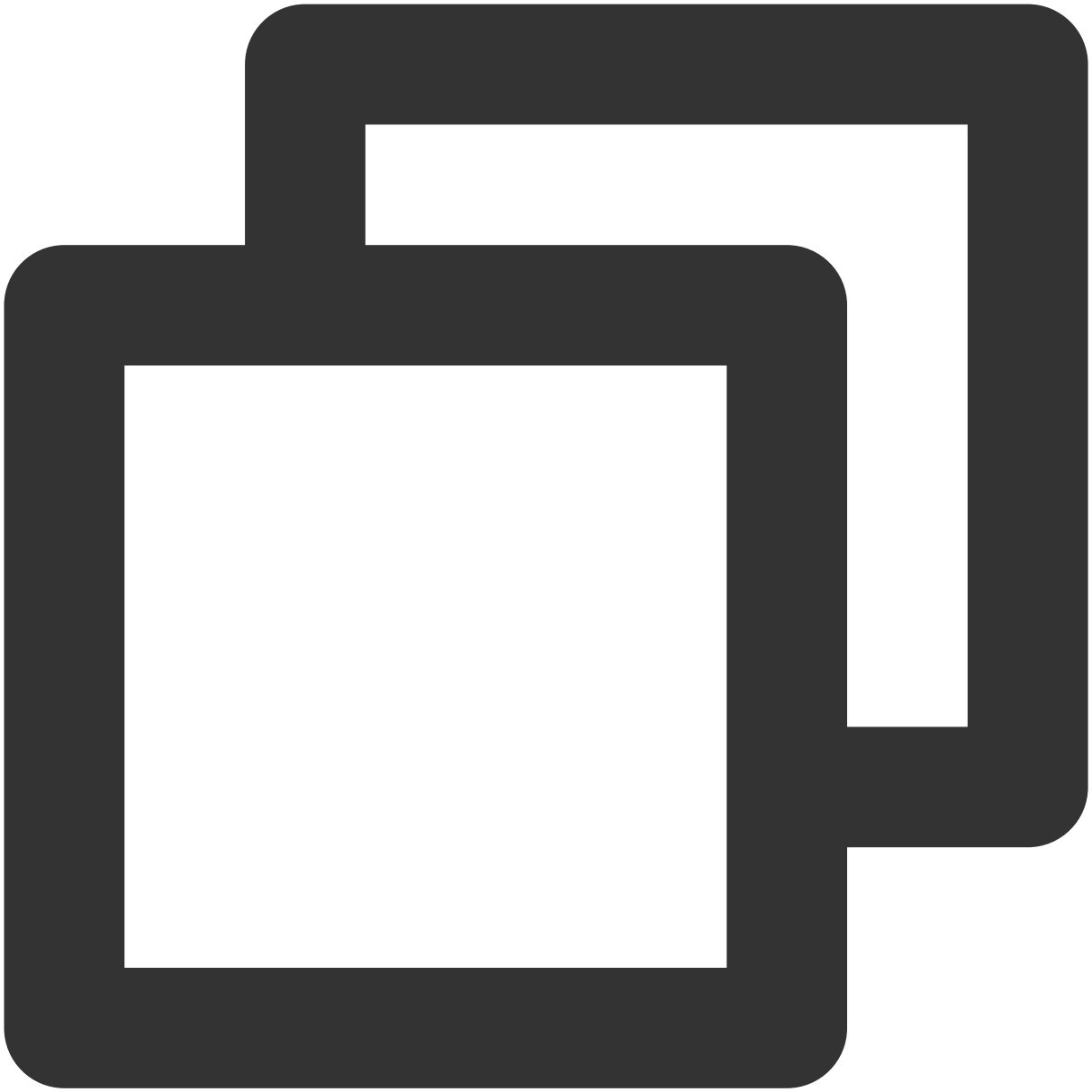
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/format/<Format>
               /cgif/<FrameNumber>
               /interlace/<Mode> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



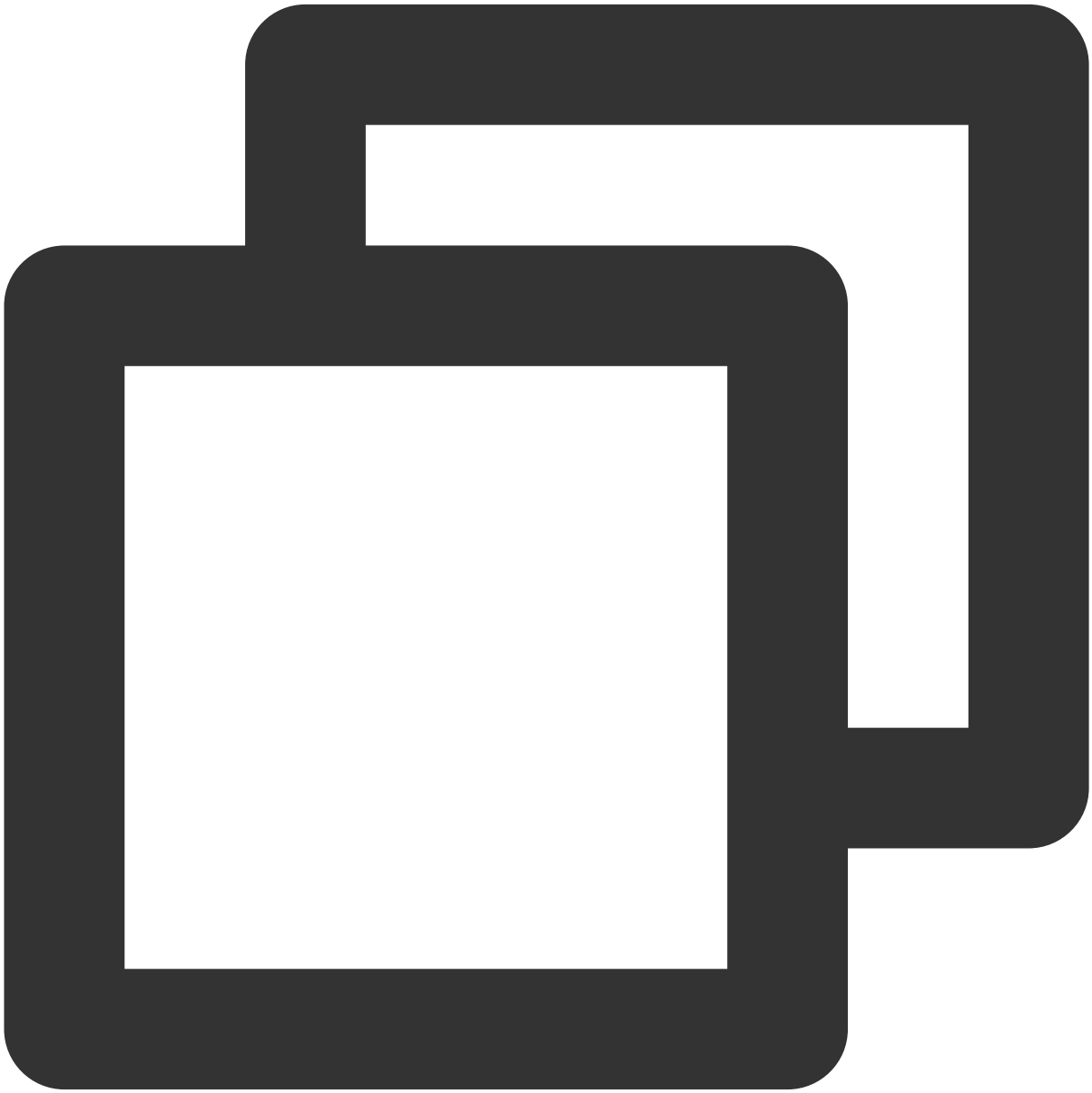
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/format/<Format>
                      /cgif/<FrameNumber>
```

```
                        /interlace/<Mode>"
    }]
}
```

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/<Format>
                        /cgif/<FrameNumber>
                        /interlace/<Mode>"
  }]
}
```

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

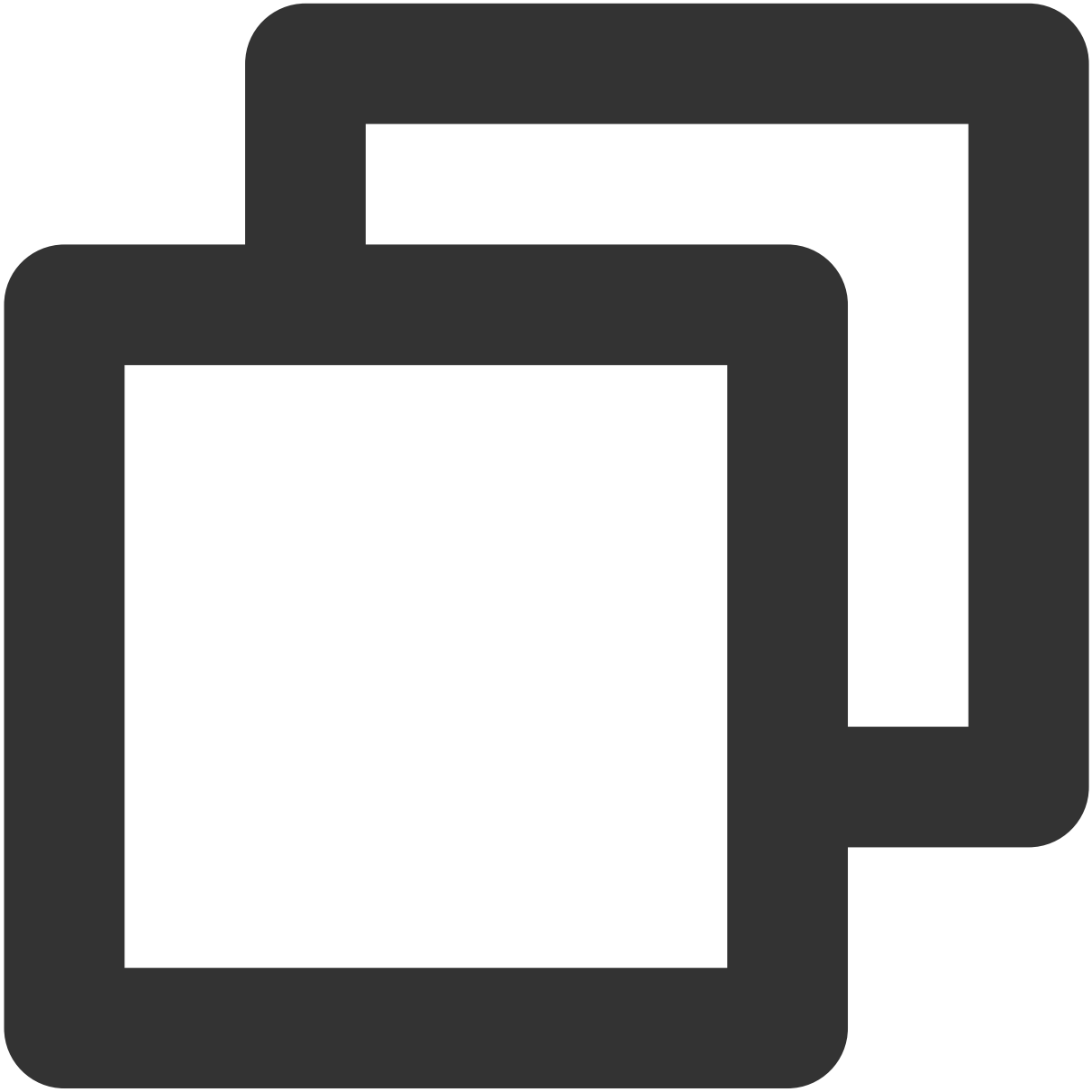
参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/format/<Format>	格式转换：目标缩略图的图片格式可为：jpg，bmp，gif，png，webp，yjpeg 等，其中 yjpeg 为数据万象针对 jpeg 格式进行的优化，本质为 jpg 格式；缺省为原图格式。
/cgif/<FrameNumber>	gif 格式优化：只针对原图为 gif 格式，对 gif 图片格式进行的优化，降帧降颜色。分为以下两种情况： FrameNumber=1，则按照默认帧数30处理，如果图片帧数大于该帧数则截取。 FrameNumber 取值(1,100]，则将图片压缩到指定帧数 （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、参数超限等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**。

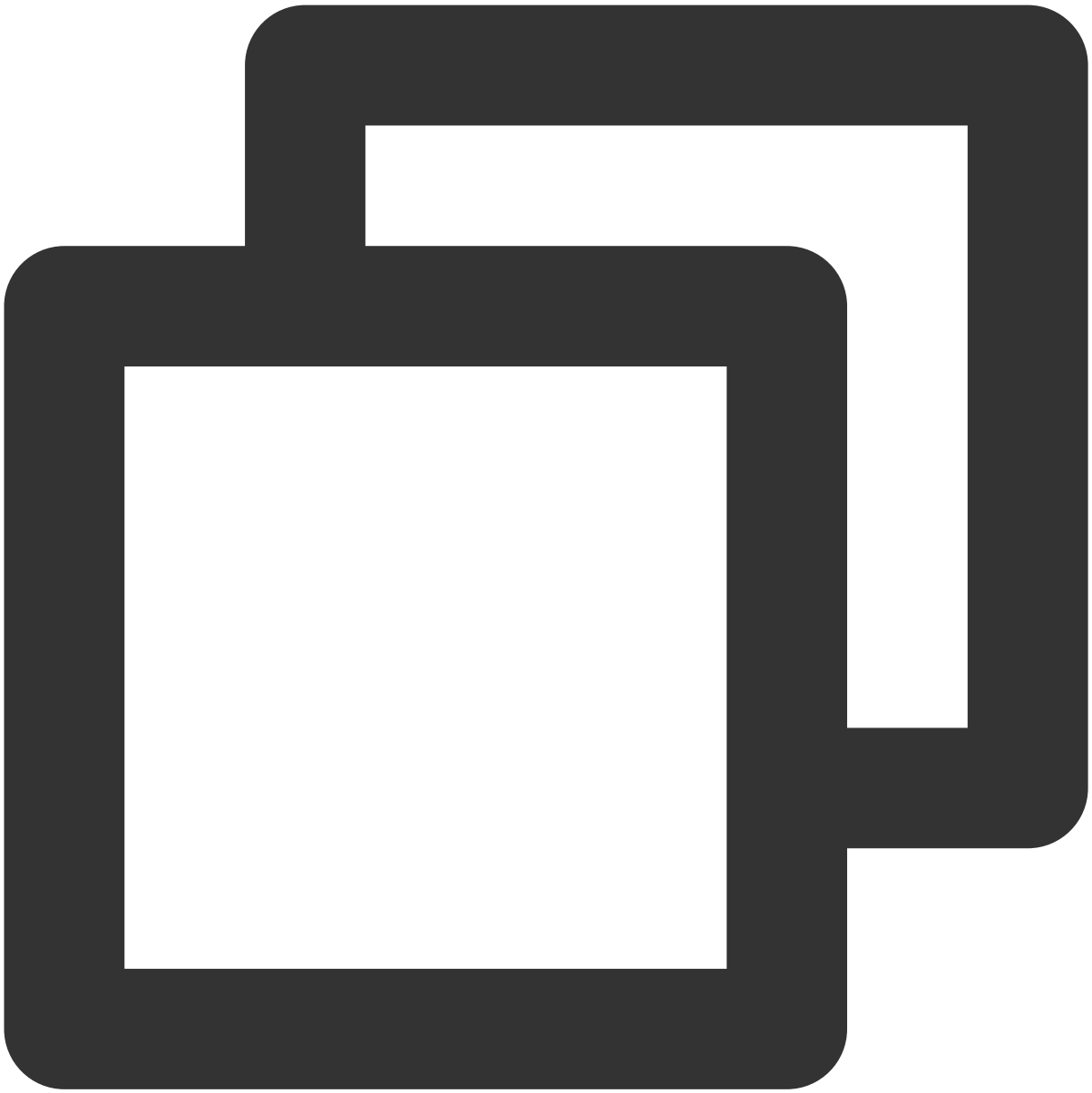
案例一：将 jpeg 格式的原图片转换为 png 格式



```
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：



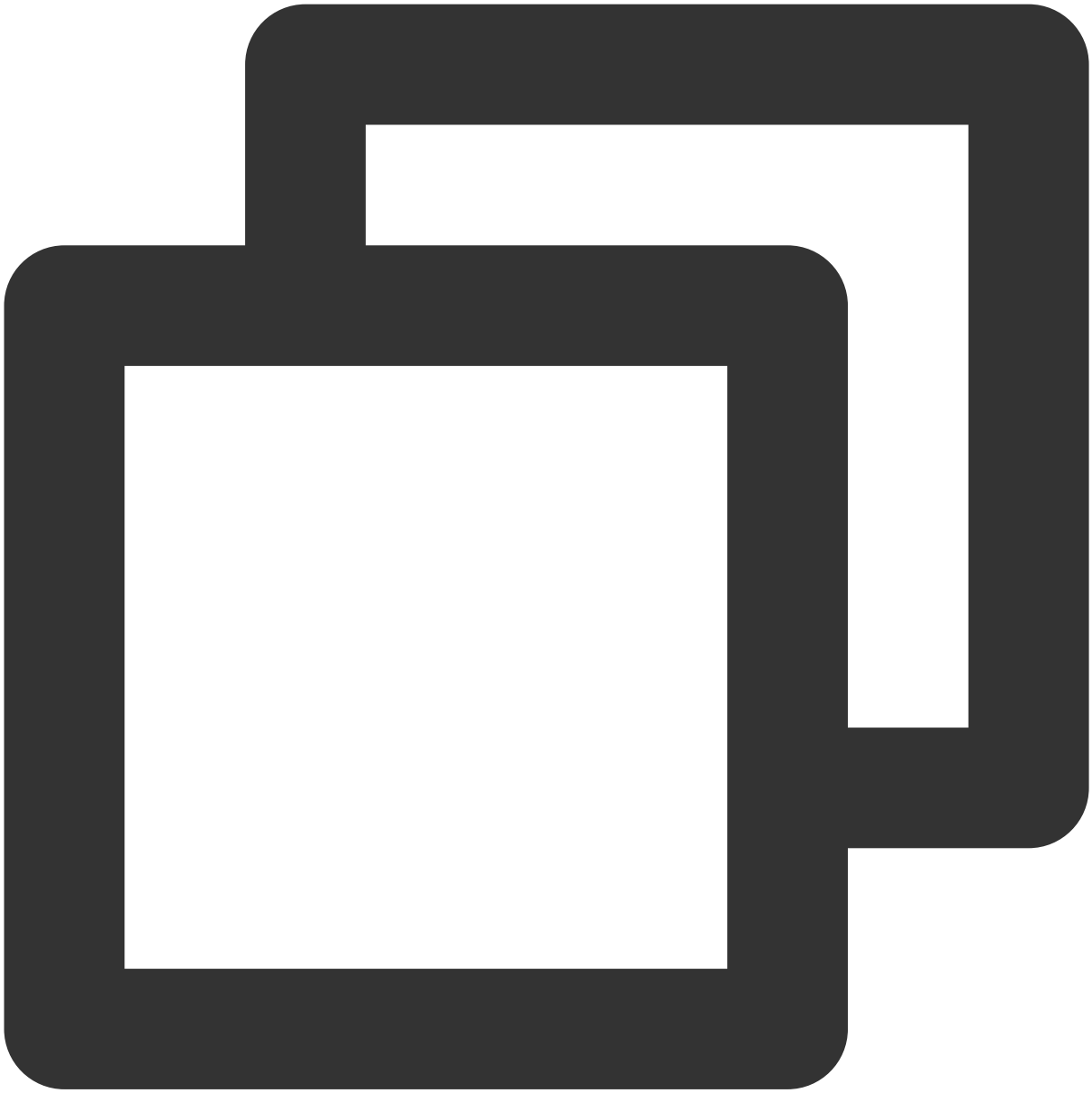
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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

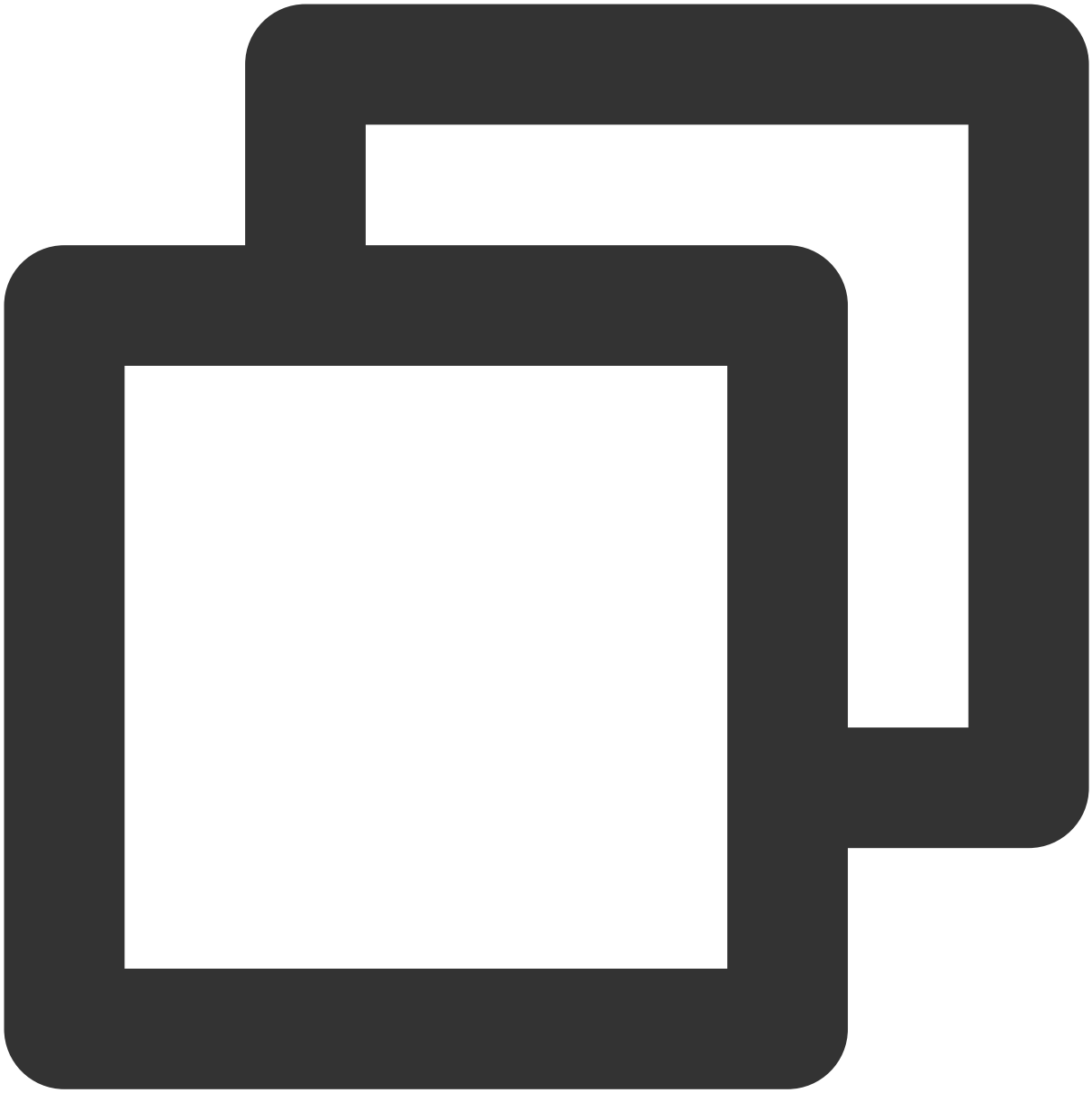
最近更新时间：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **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：
该接口支持 jpg、webp、tpg、heif、avif 格式图片。

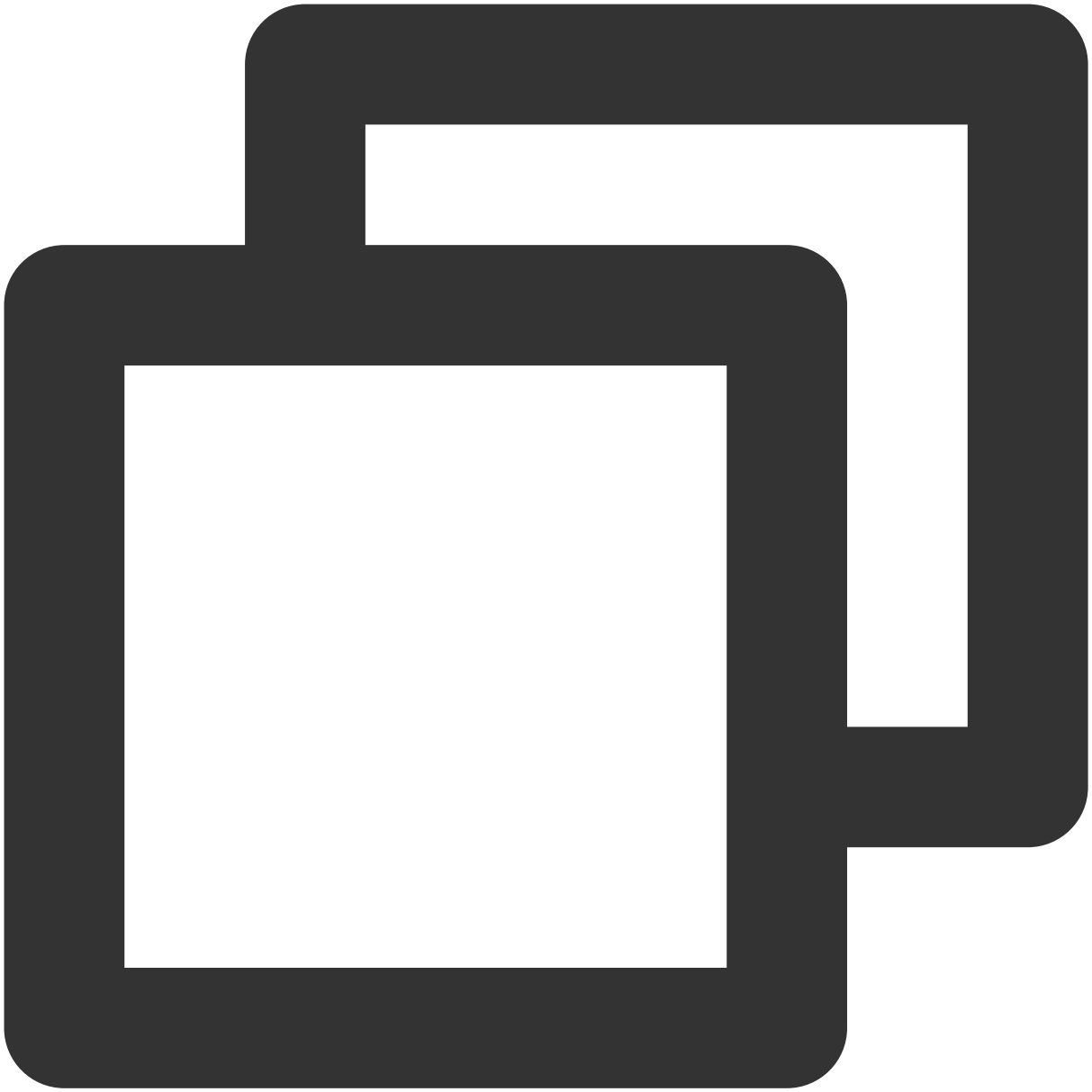
接口示例

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



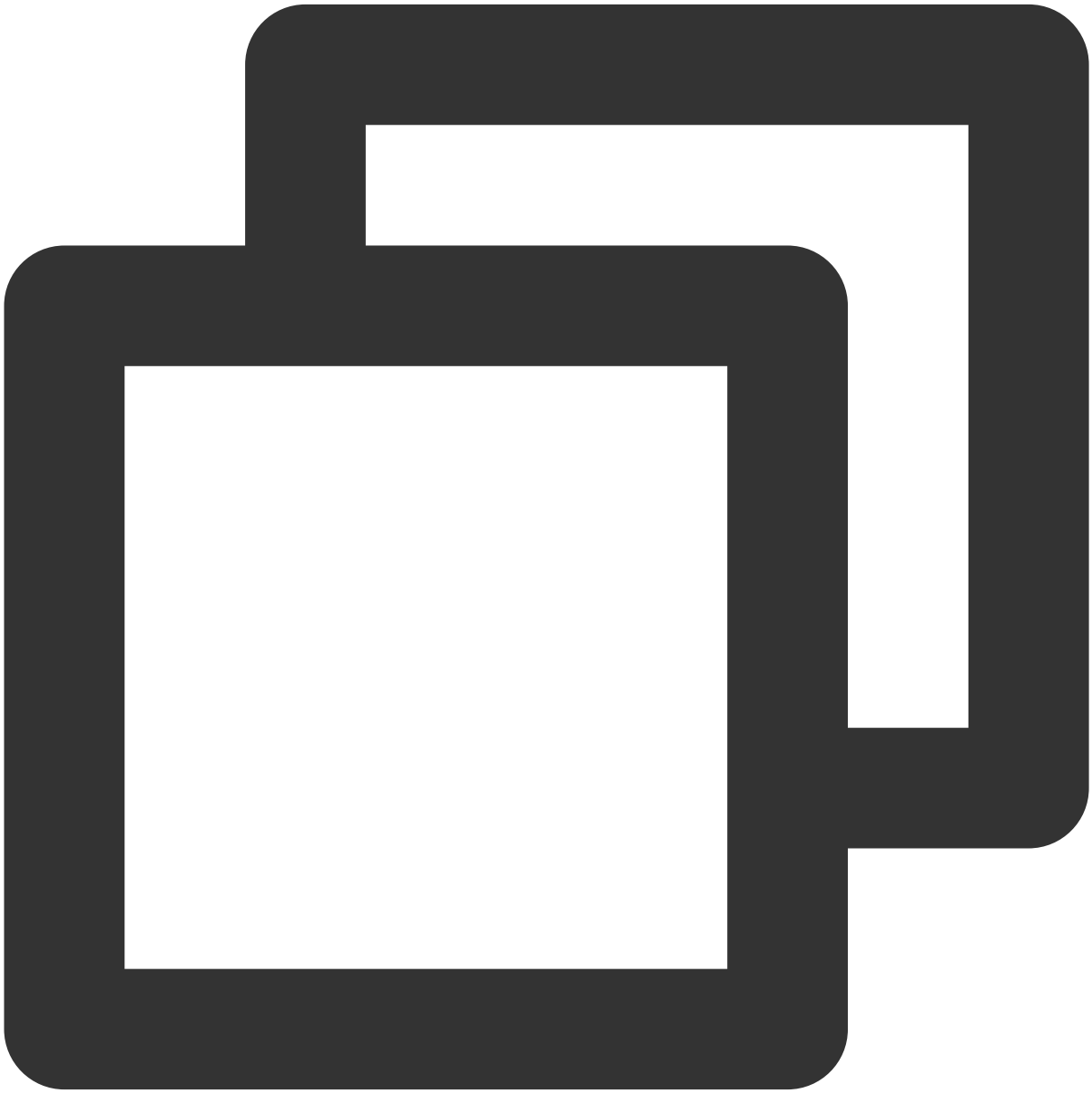
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/quality/<Quality>
                      /rquality/<quality>
```

```
                        /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、参数超限等导致处理失败的场景，会直接返回原图而不报错

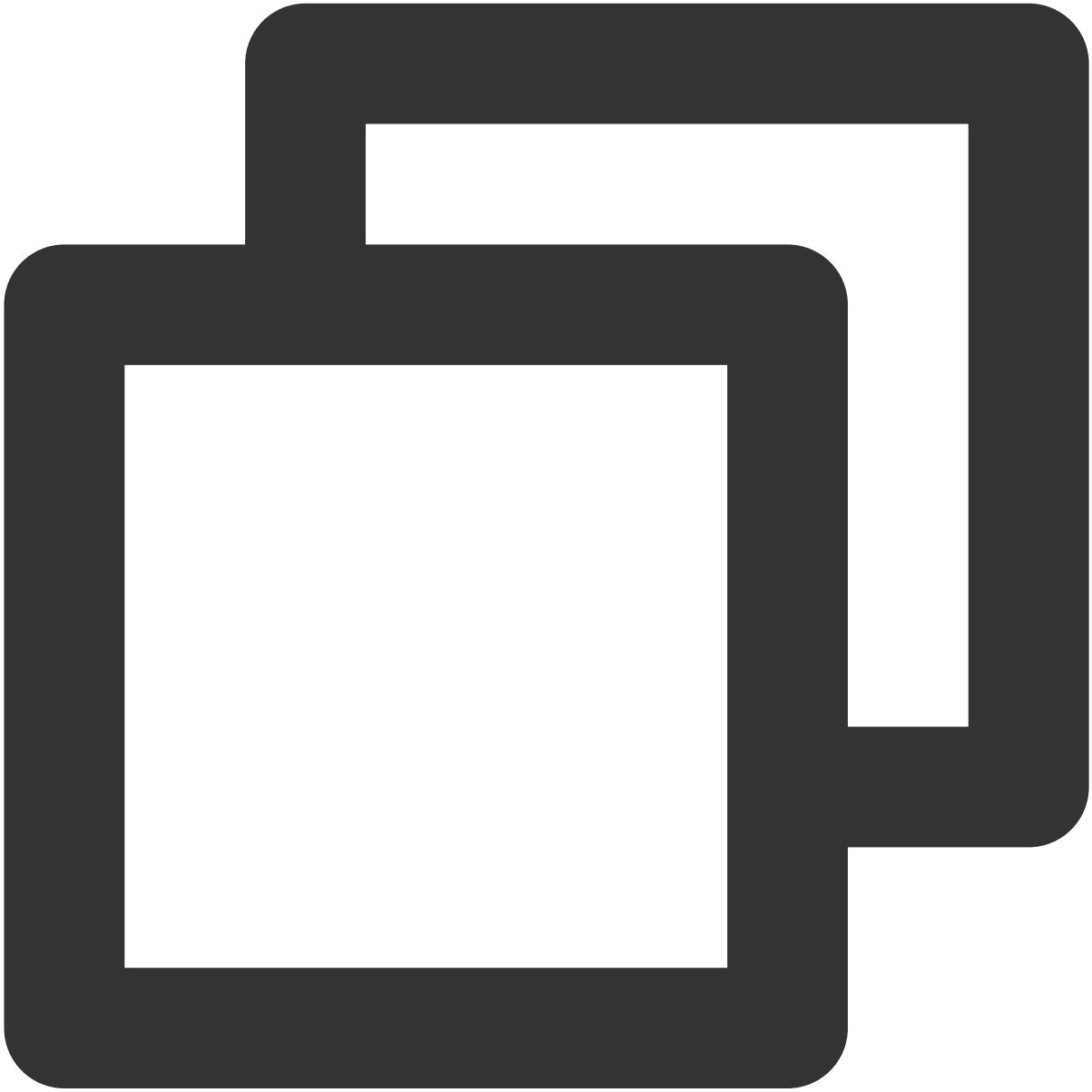
实际案例

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：



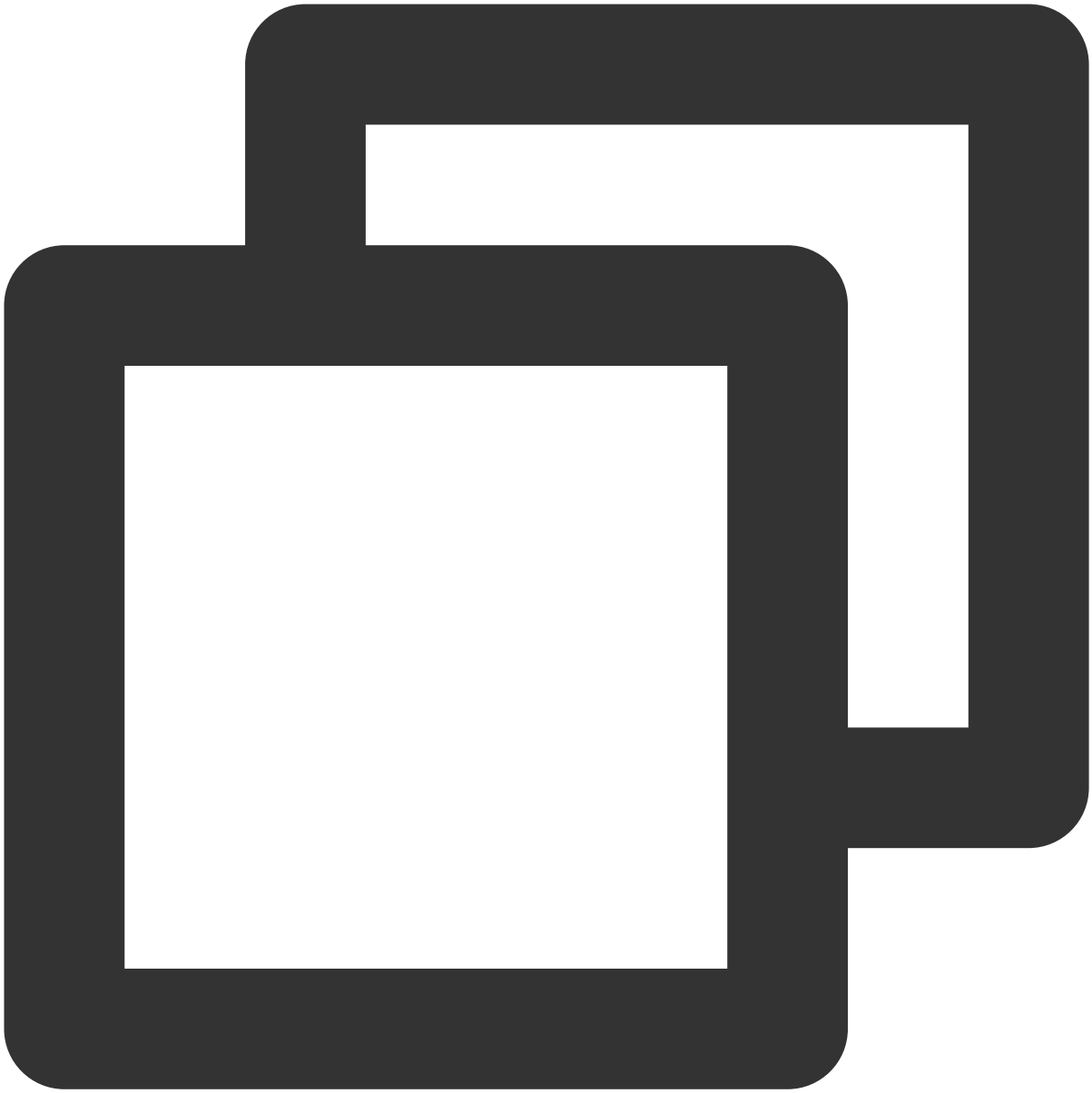
```
http://examples-1251000004.picsh.myqcloud.com/sample.jpeg?imageMogr2/quality/60
```

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：



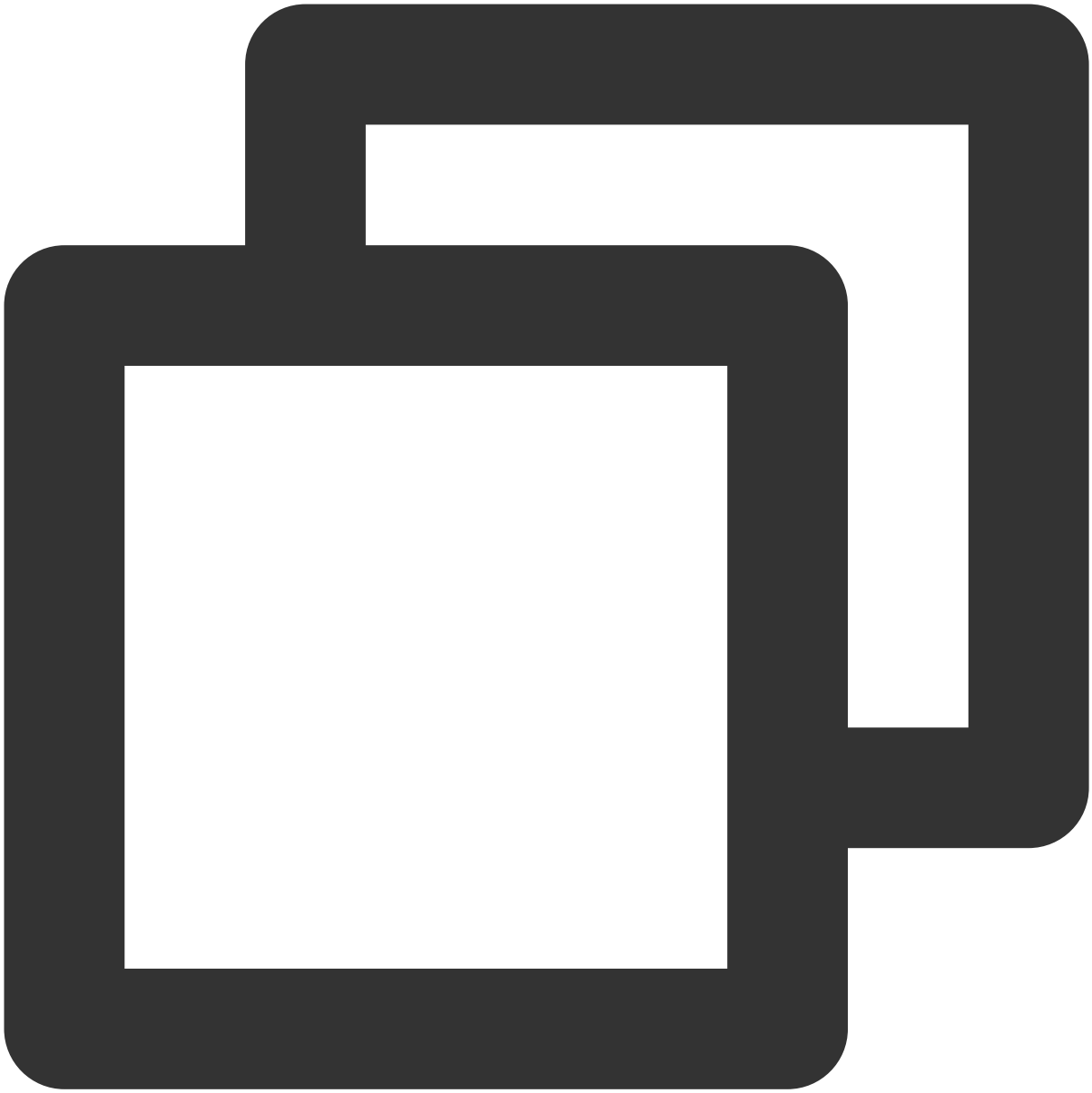
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/rqua
```

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：



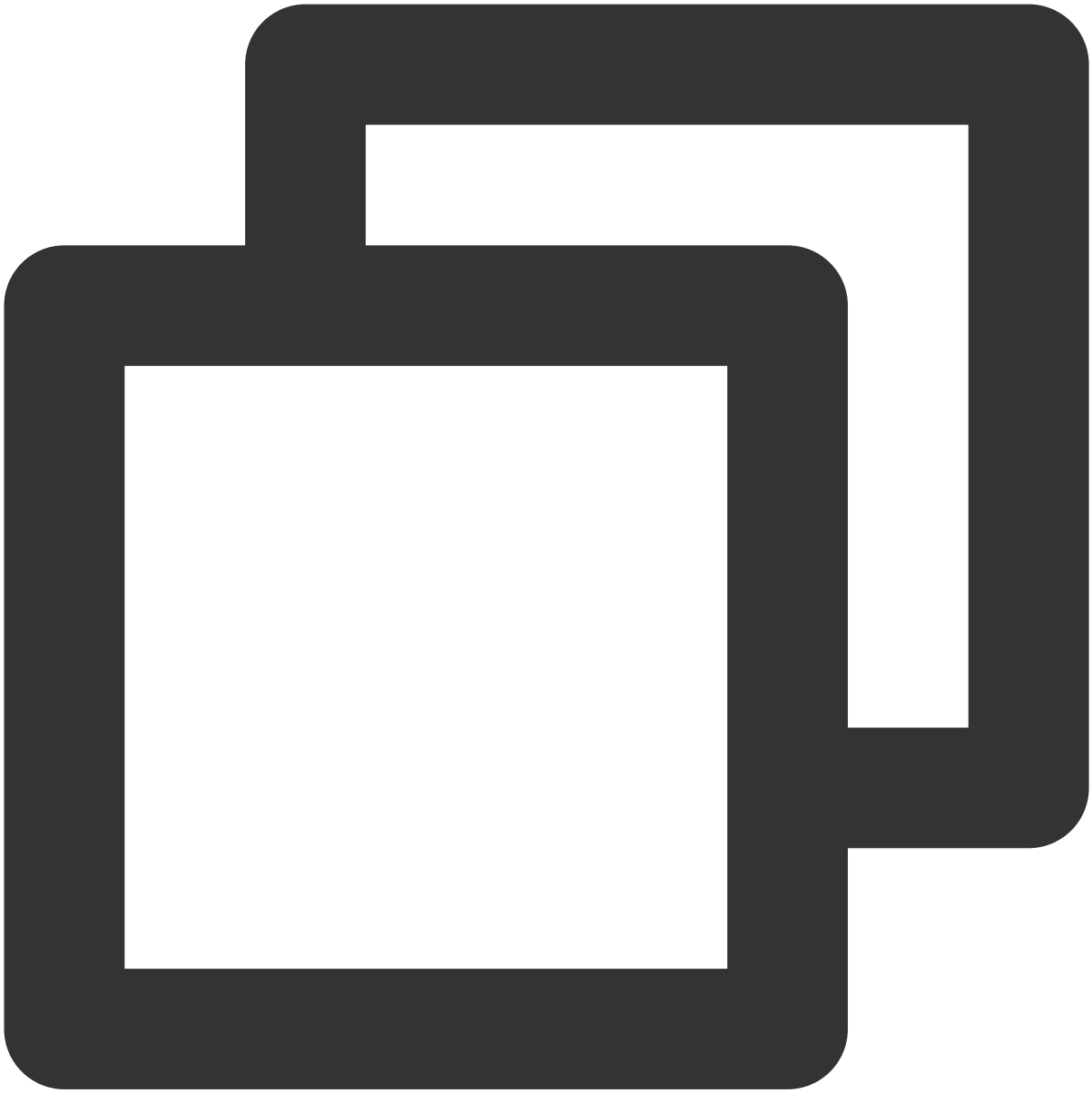
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **imageMogr2** 接口对图片进行模糊处理。
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

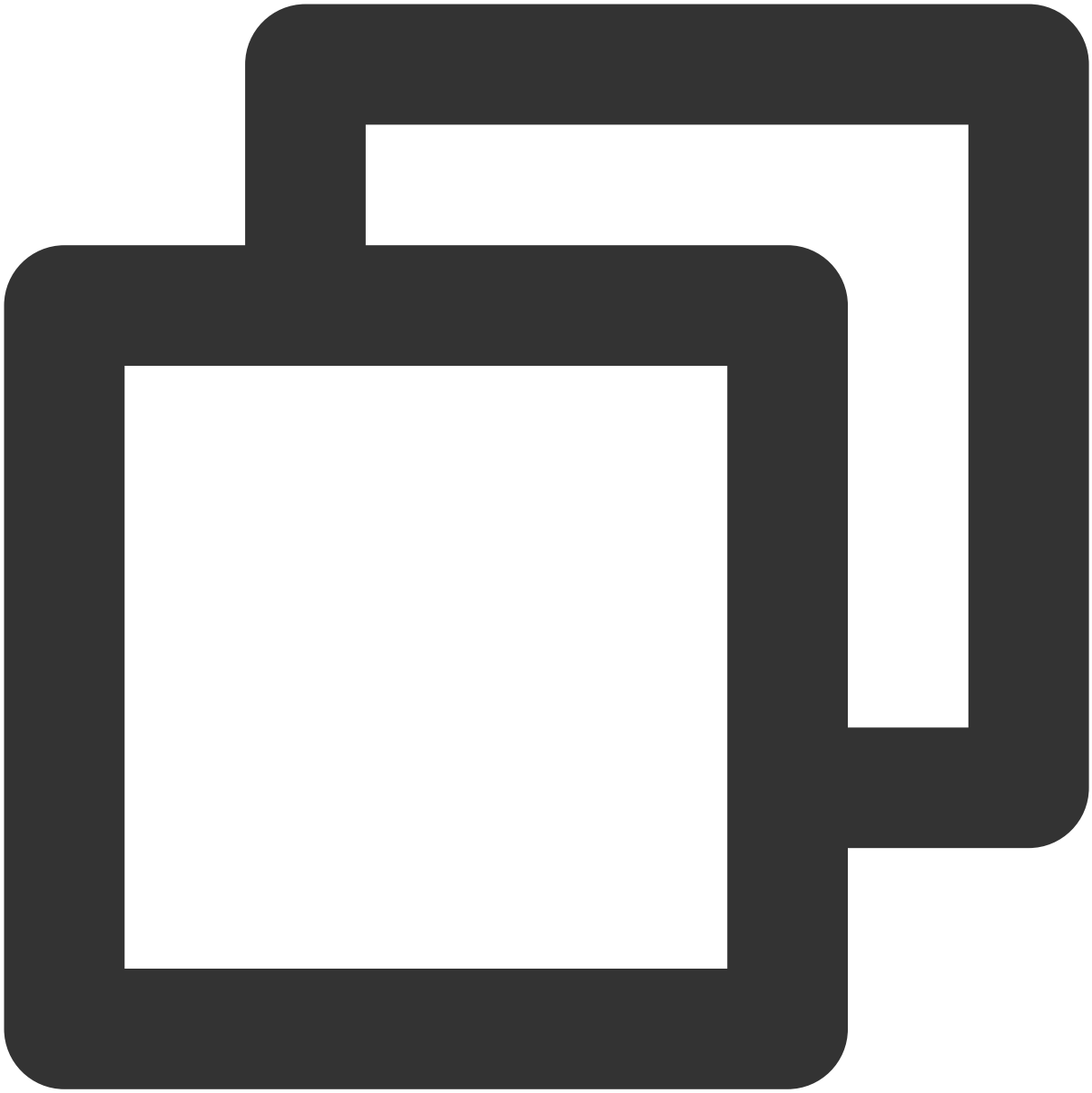
限制说明

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

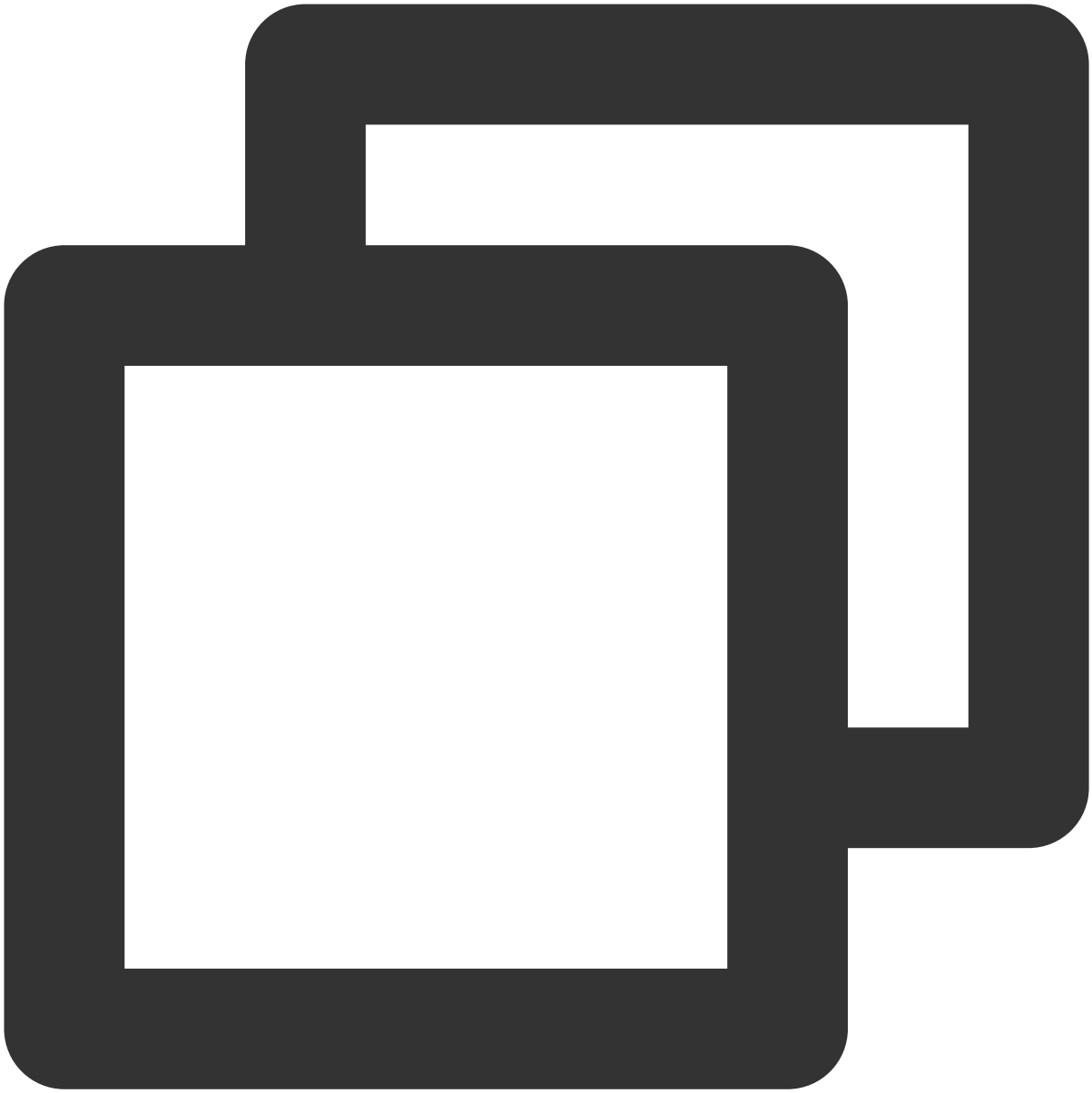
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/blur/<radius>x<sigma>        HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



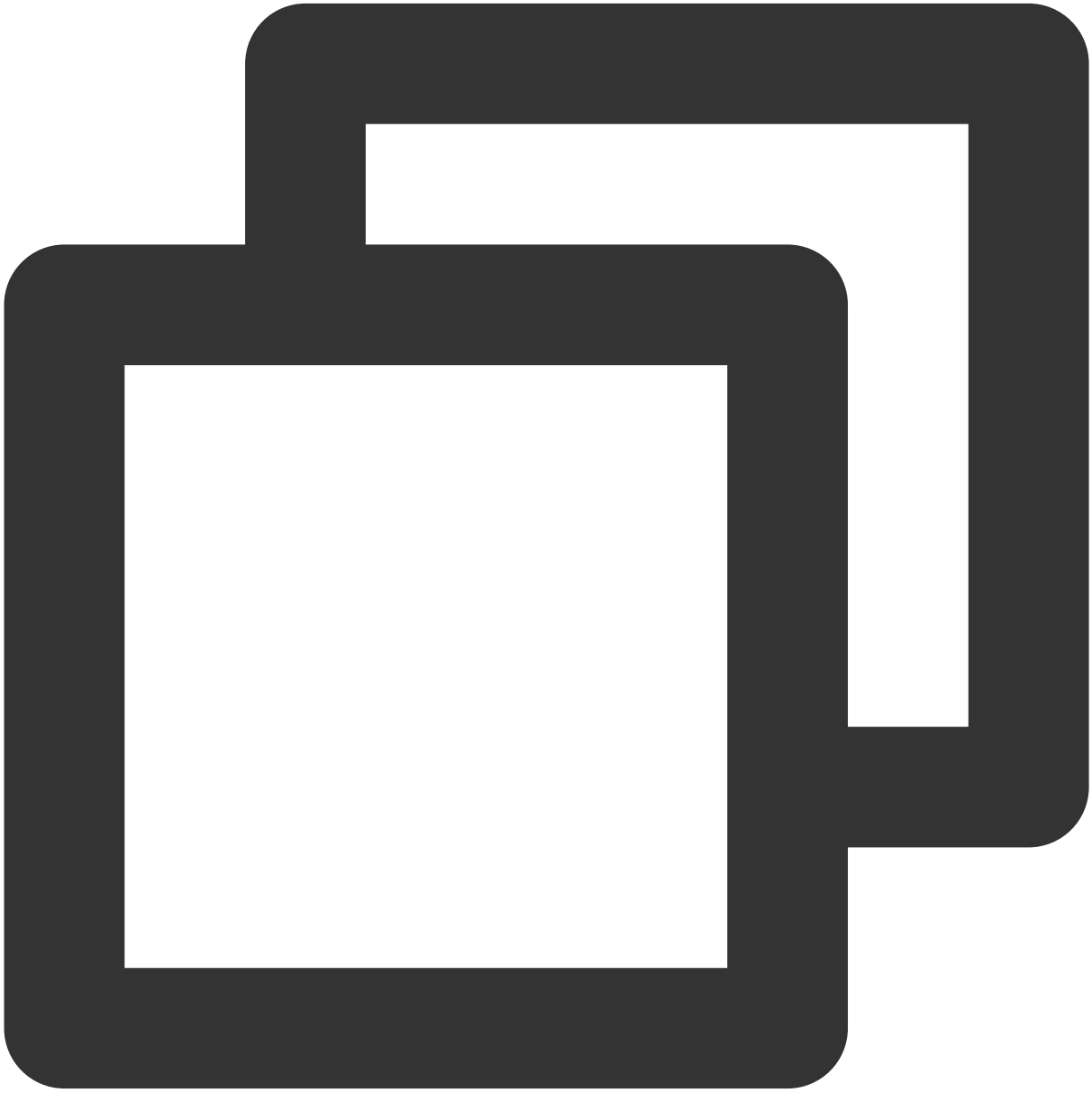
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/blur/<radius>x<sigma>"
  }]
```

```
}
```

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/blur/<radius>x<sigma>"
  }]
}
```

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：blur。

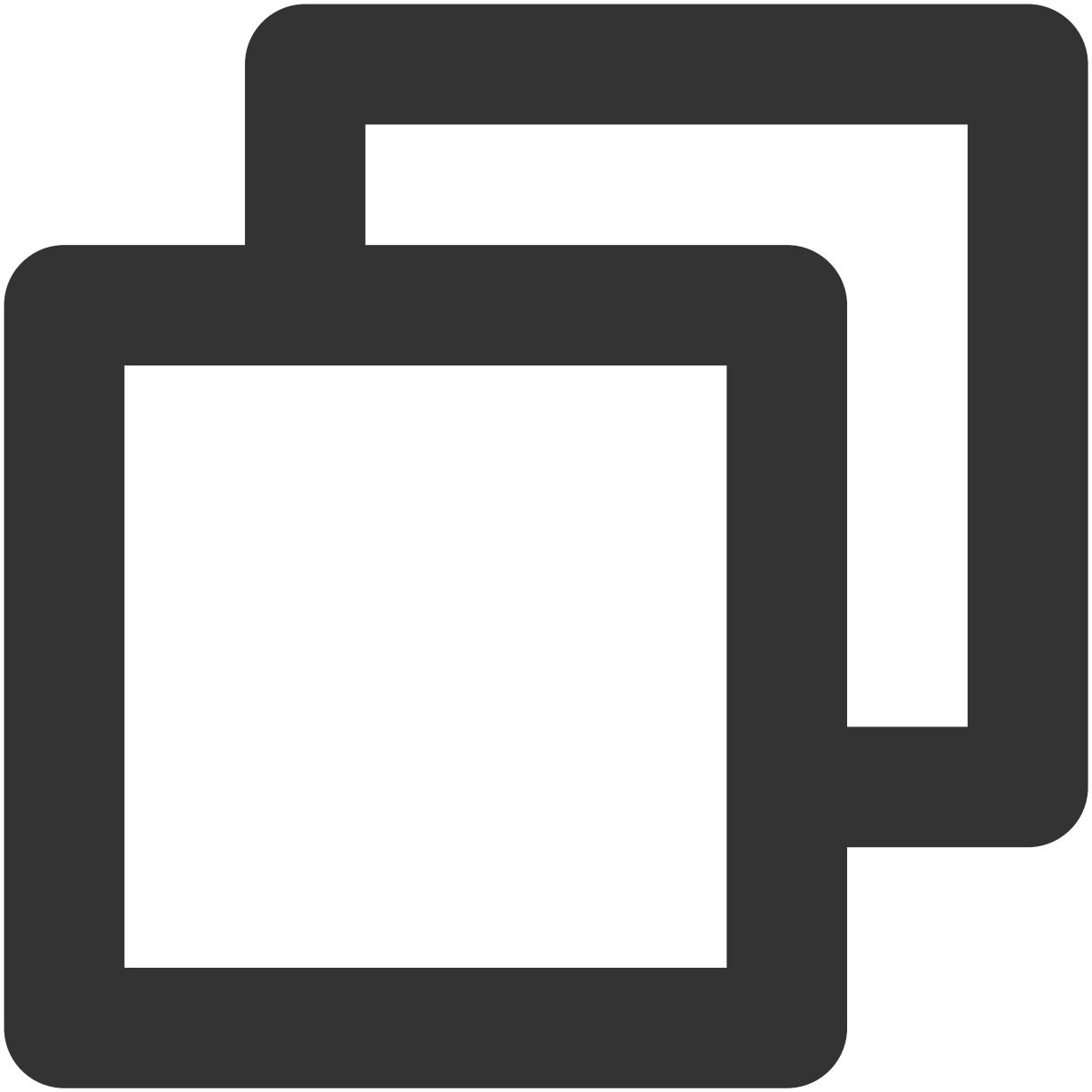
参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
radius<radius>	模糊半径，取值范围为1 - 50
sigma<sigma>	正态分布的标准差，必须大于0
/ignore-error/1	当处理参数中携带此参数时，针对文件过大、参数超限等导致处理失败的场景，会直接返回原图而不报错

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，您可参见 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



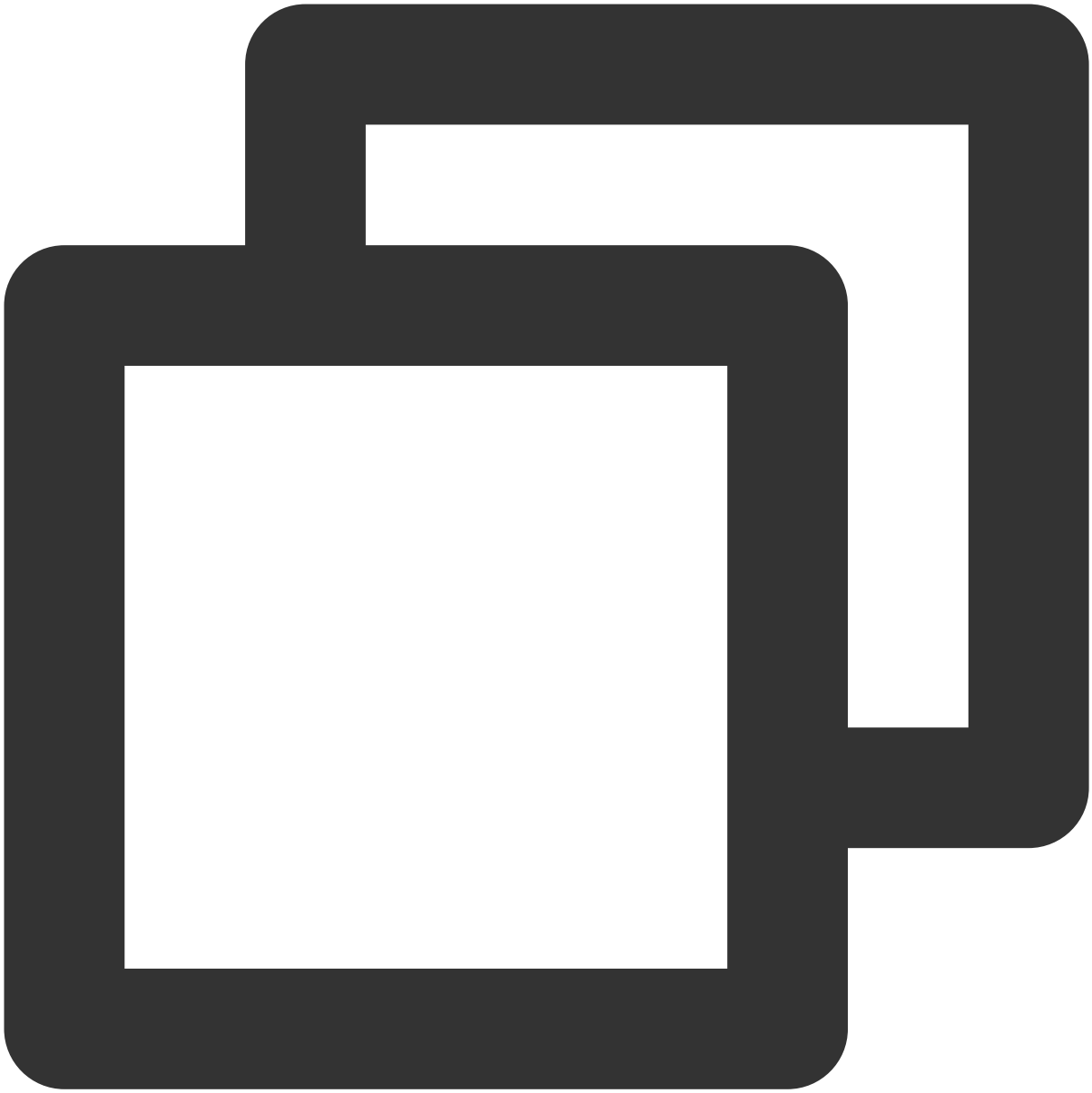
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/blur
```

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：



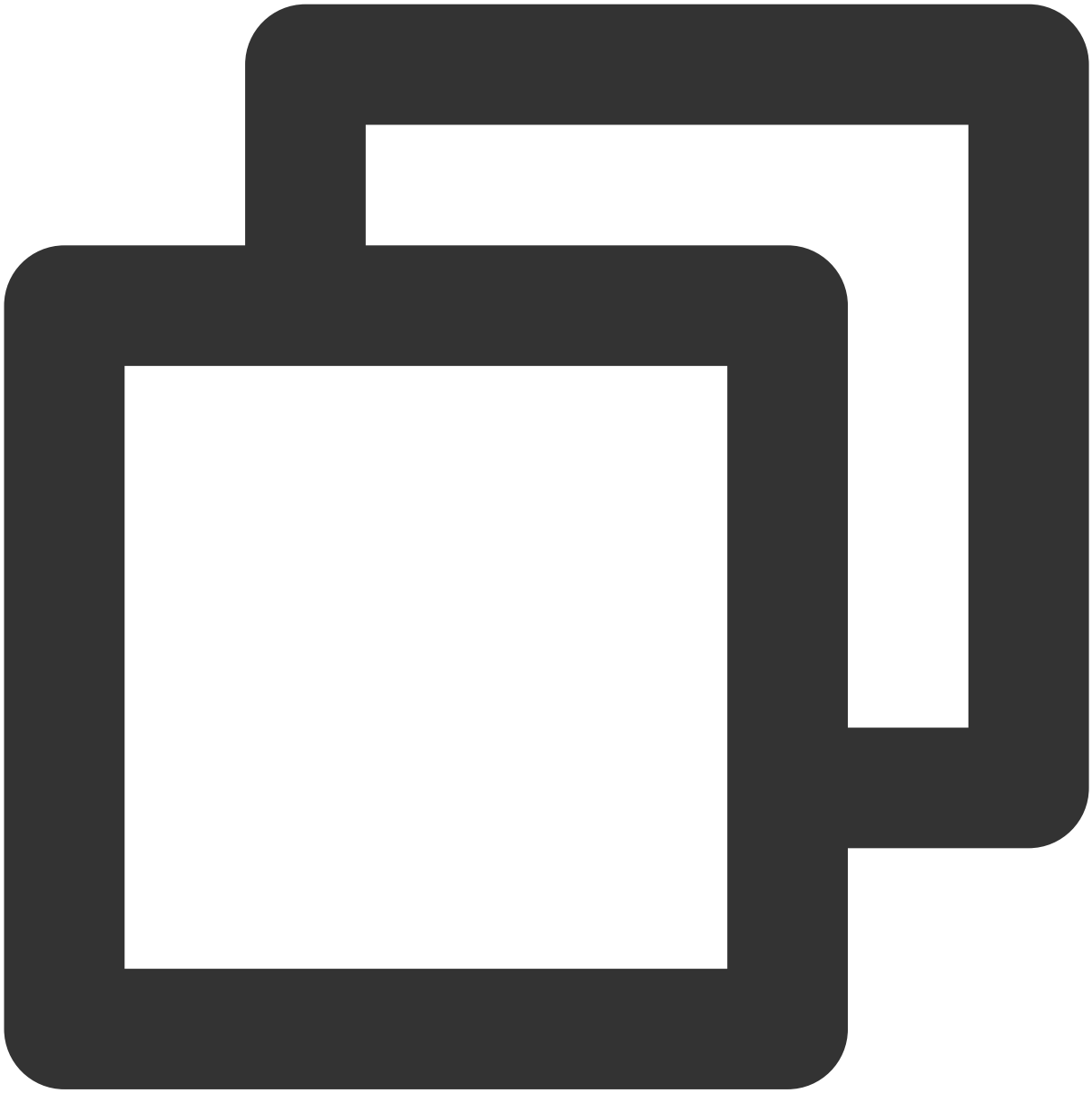
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:51

功能概述

腾讯云数据万象通过 **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

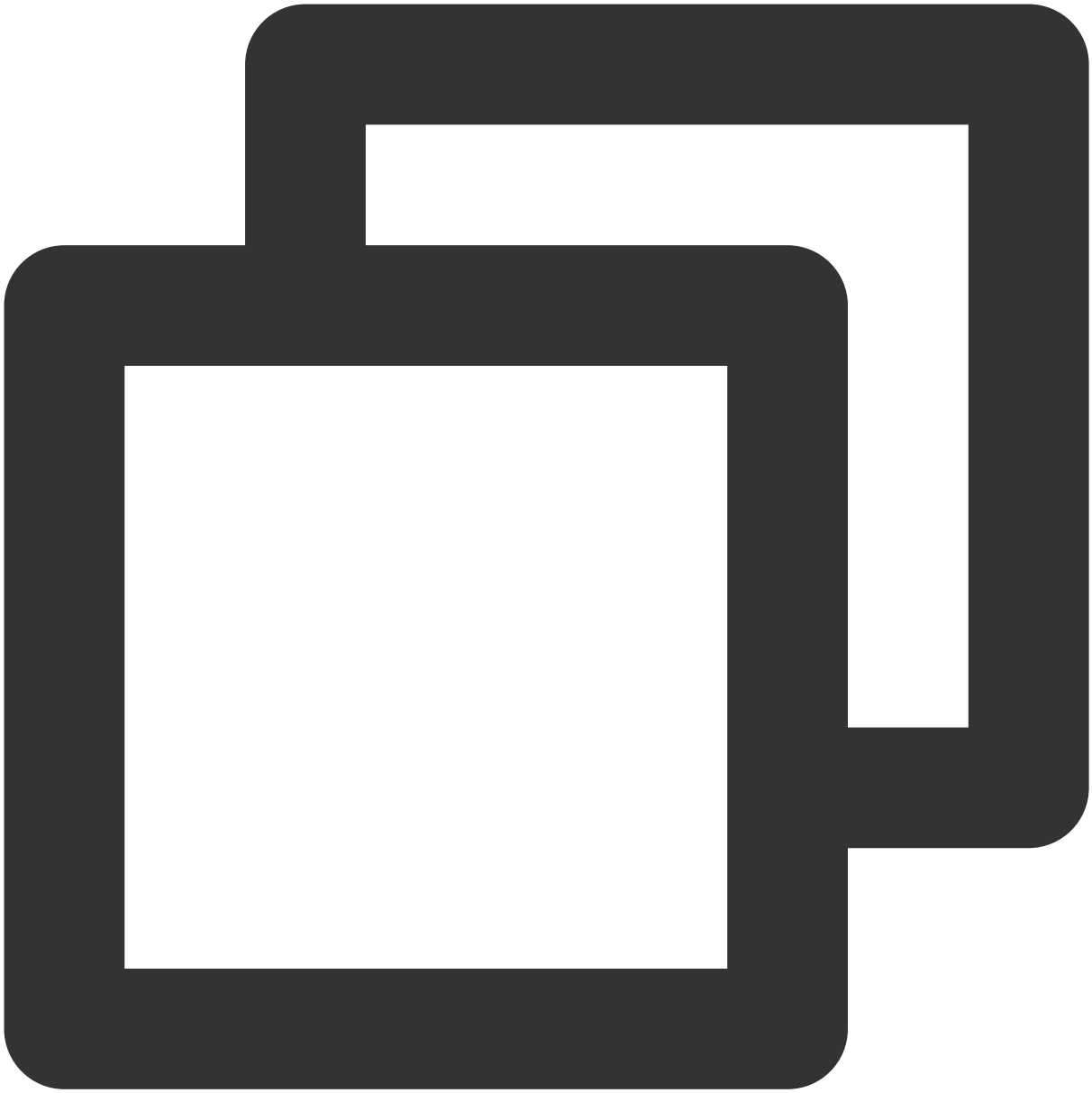
限制说明

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

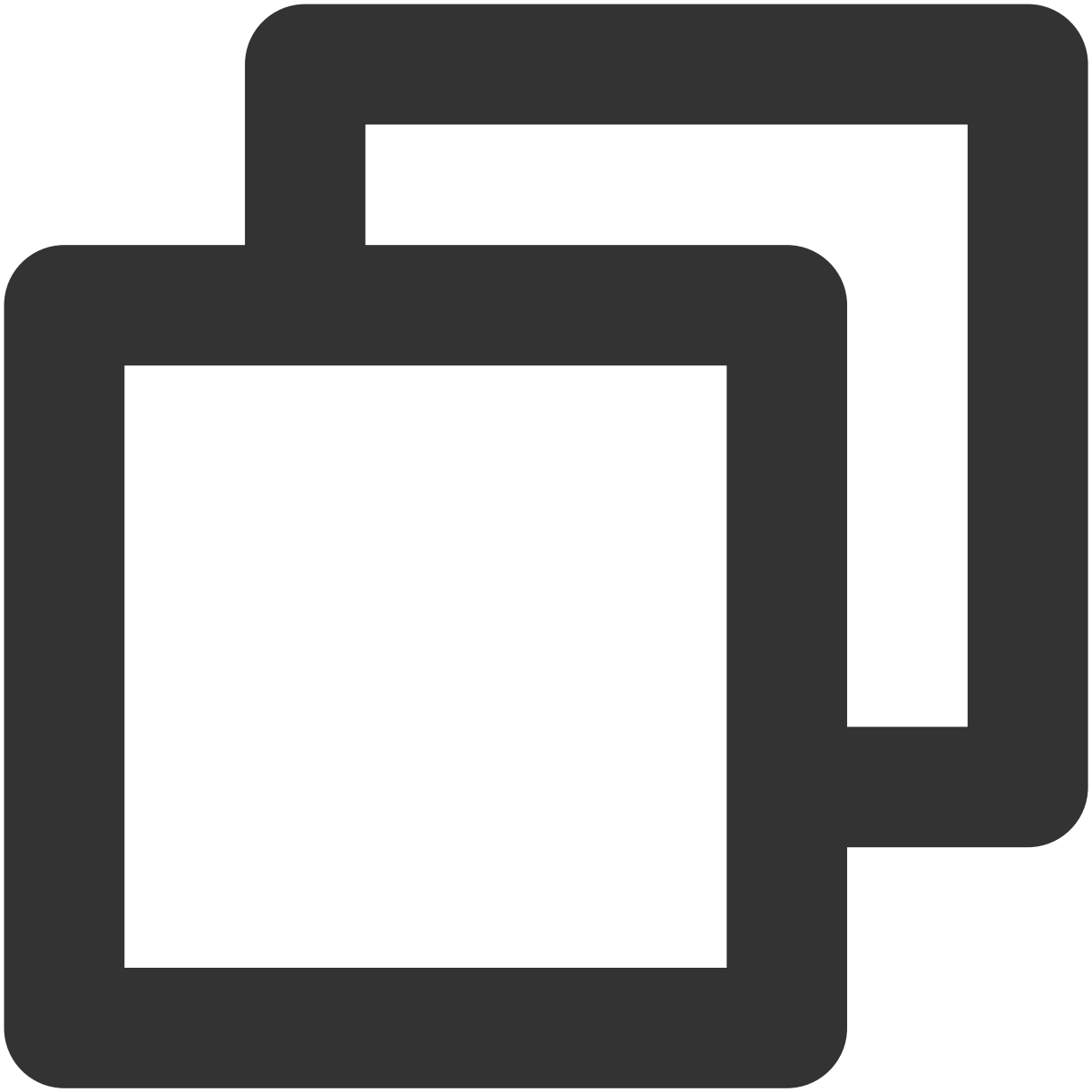
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/bright/<value> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



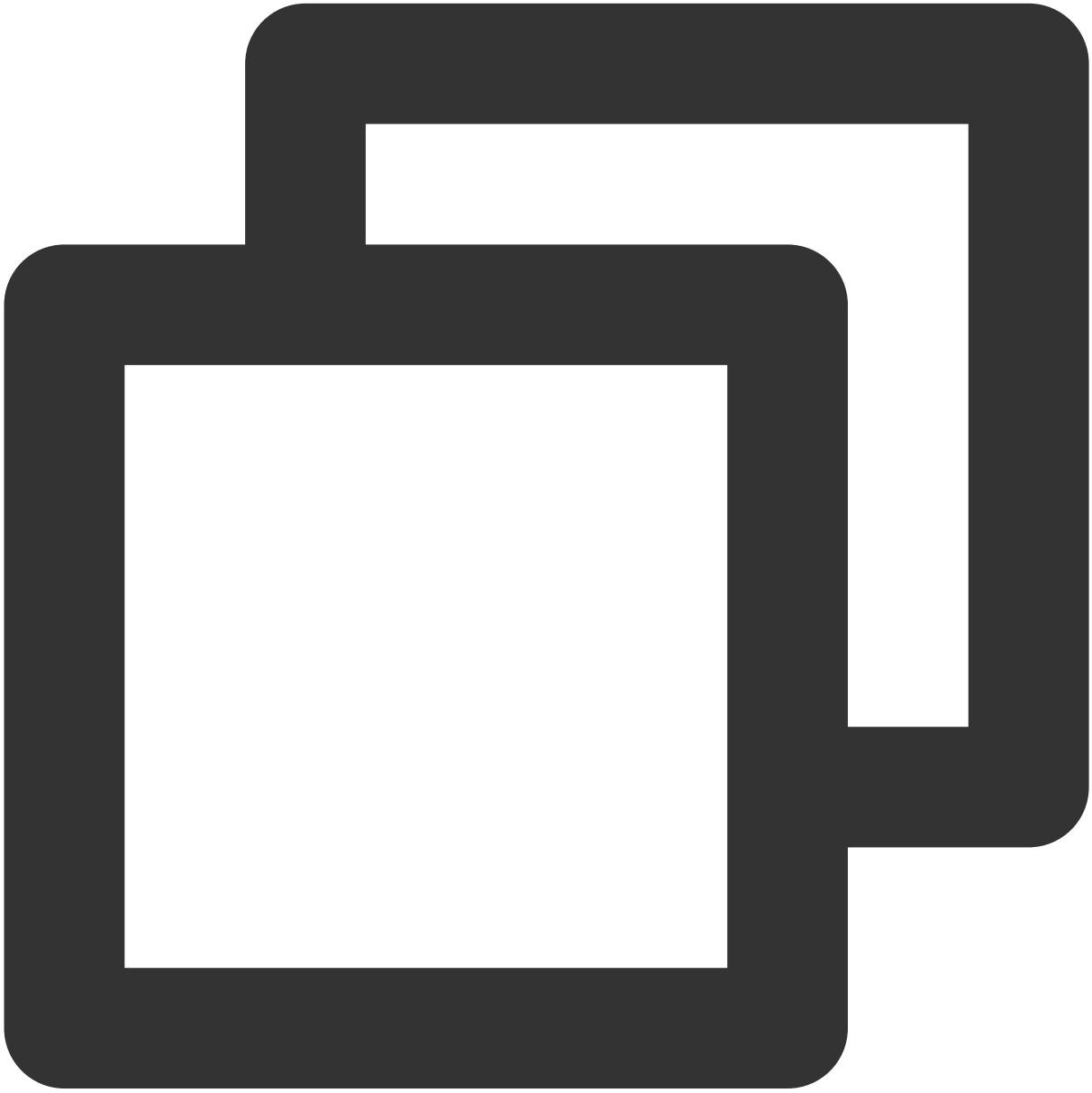
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/bright/<value>"
  }]
```

```
}
```

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/bright/<value>"
  }]
}
```

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：bright。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/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、参数超限等导致处理失败的场景，会直接返回原图而不报错。

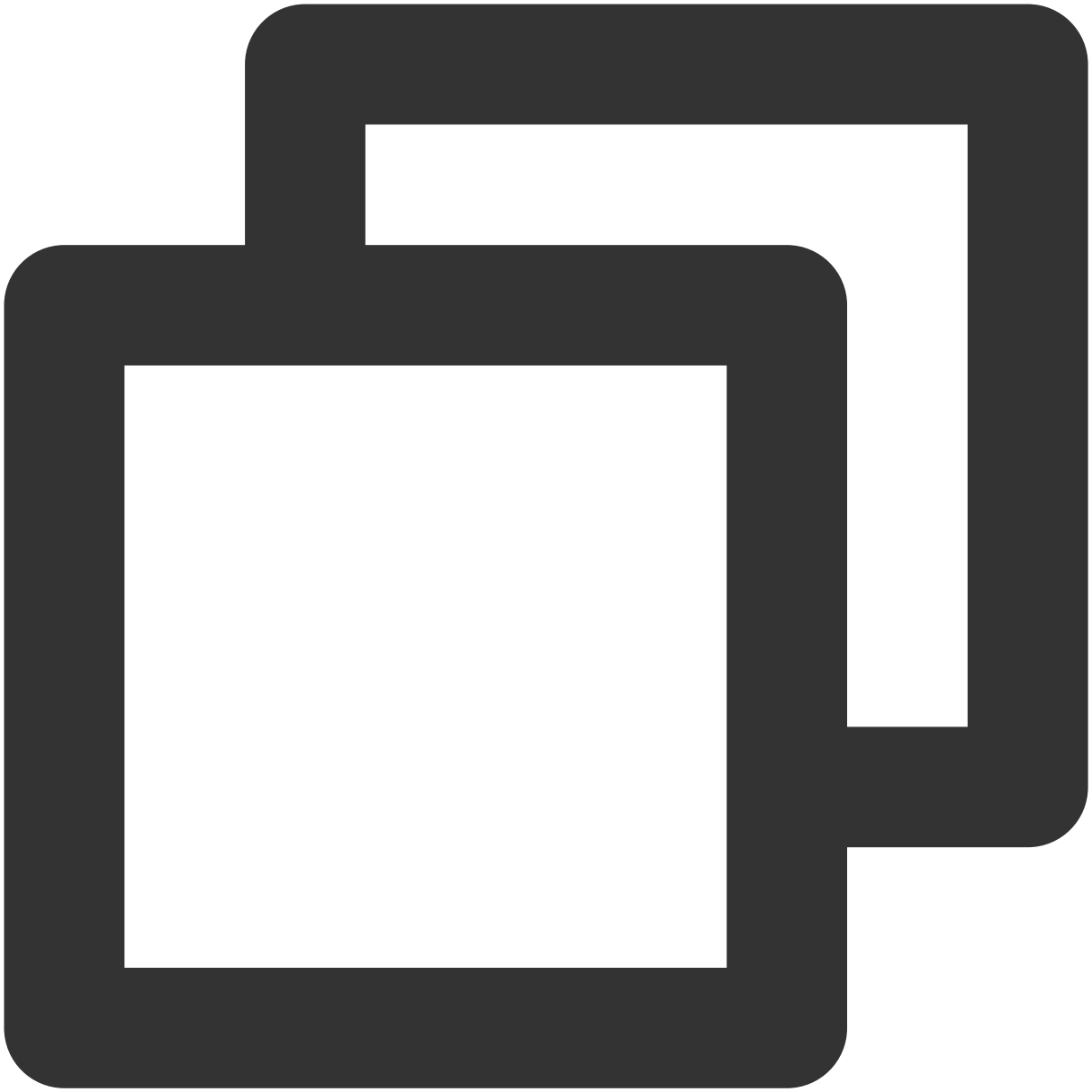
实际案例

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：



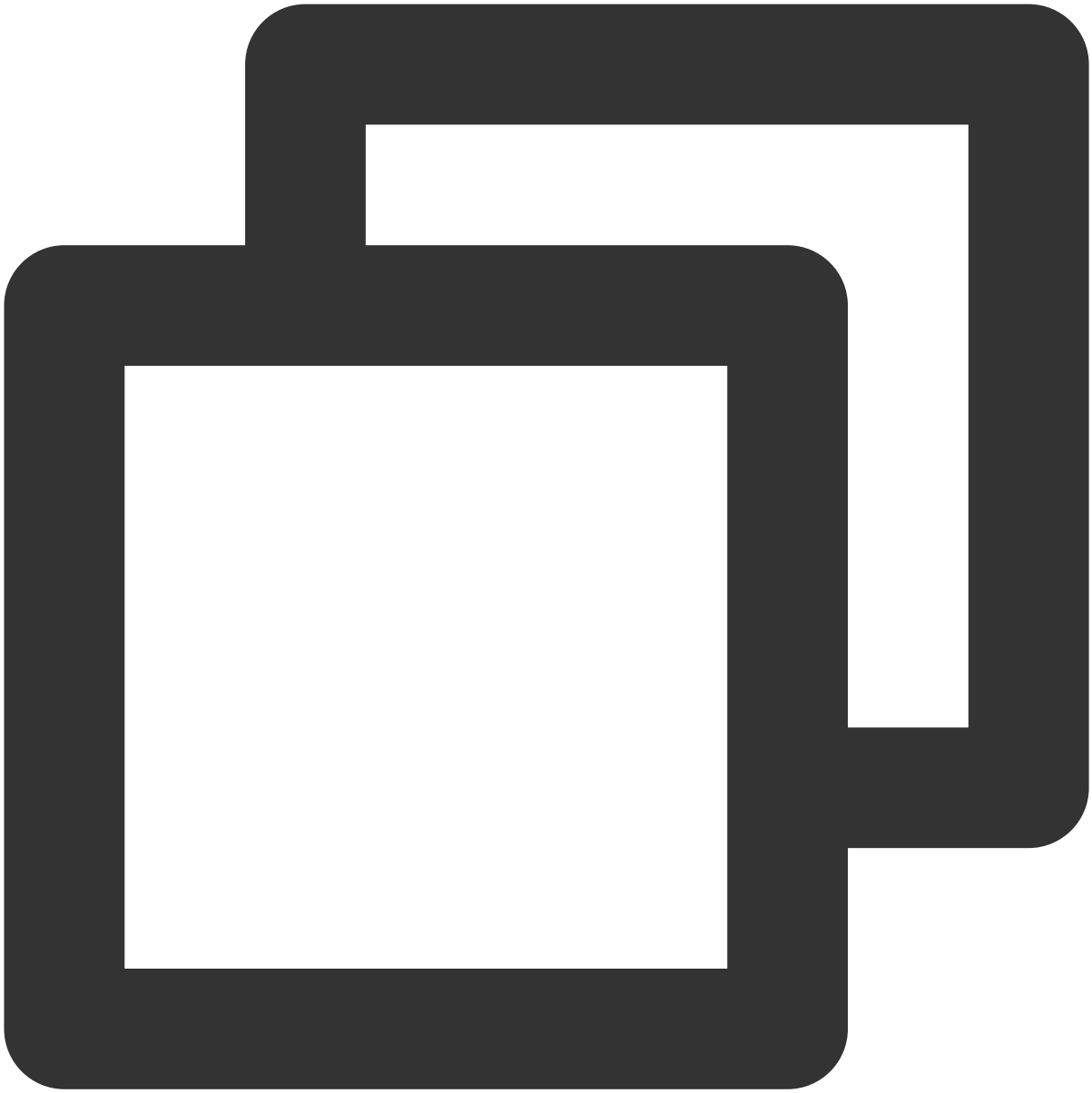
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/brig
```

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：



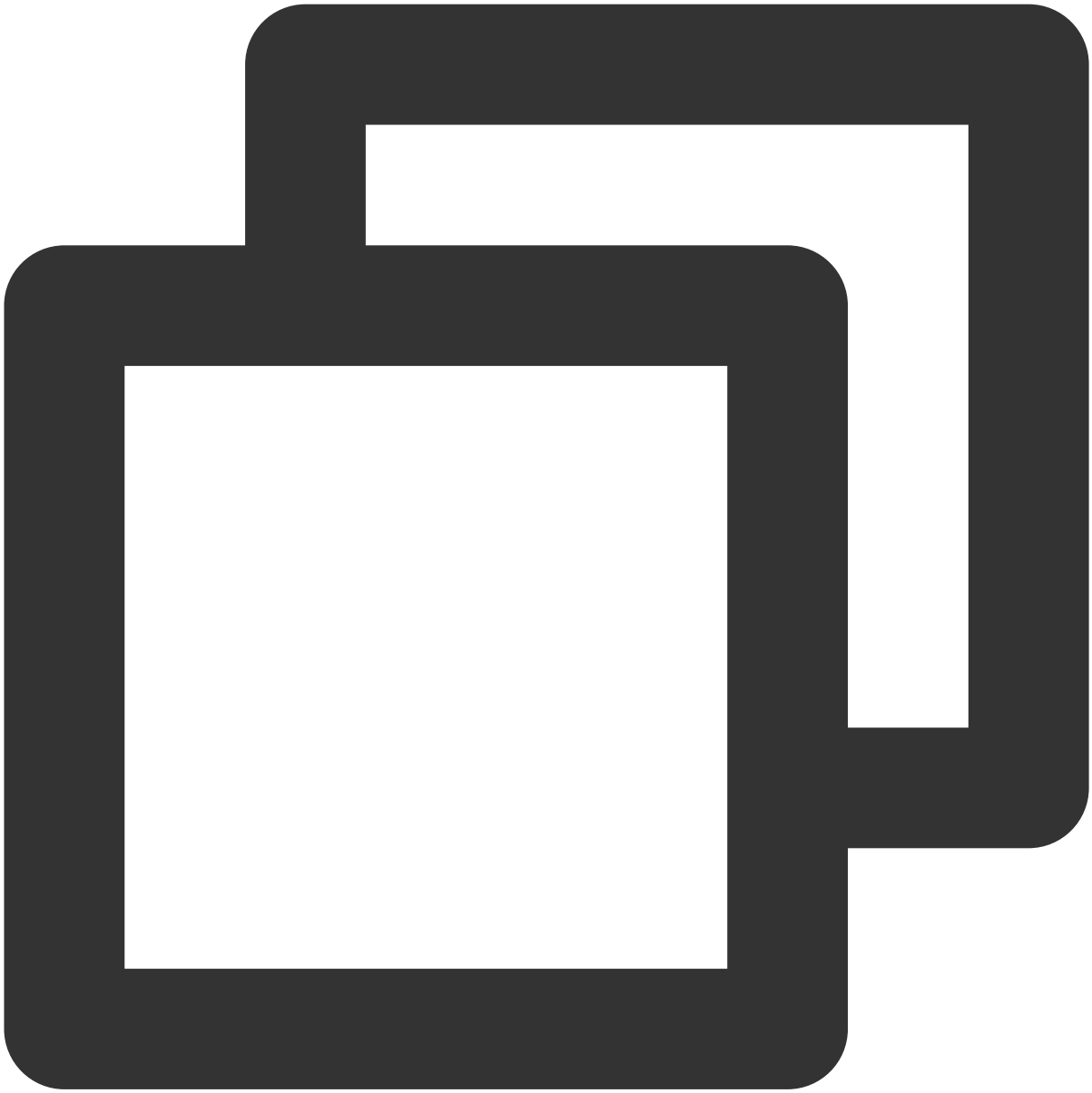
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:51

功能概述

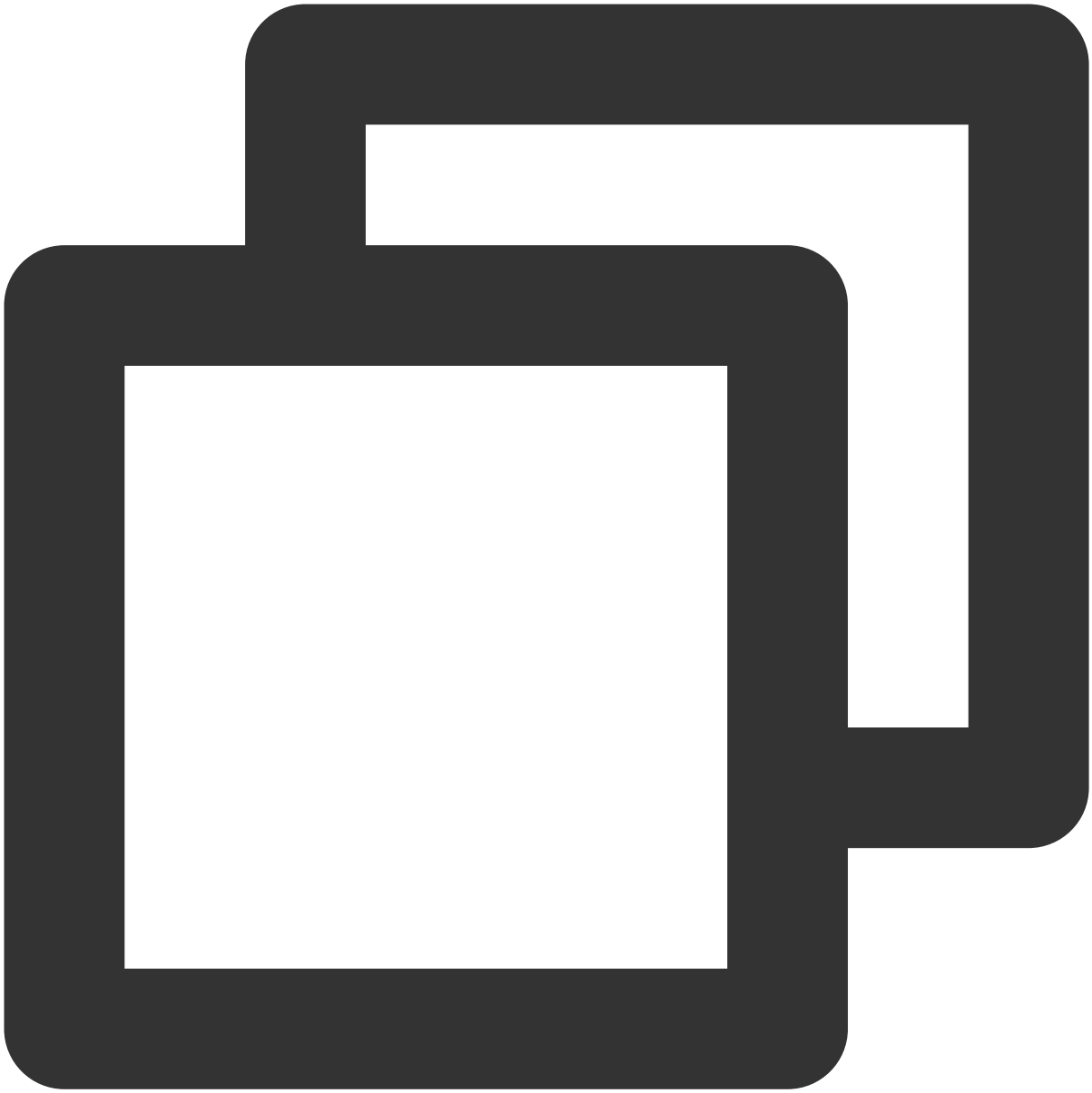
对比度指一幅图像中明暗区域最亮的白和最暗的黑之间不同亮度层级的测量，即指一幅图像灰度反差的大小。腾讯云数据万象通过 **imageMogr2** 接口调节图片对比度。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理

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。

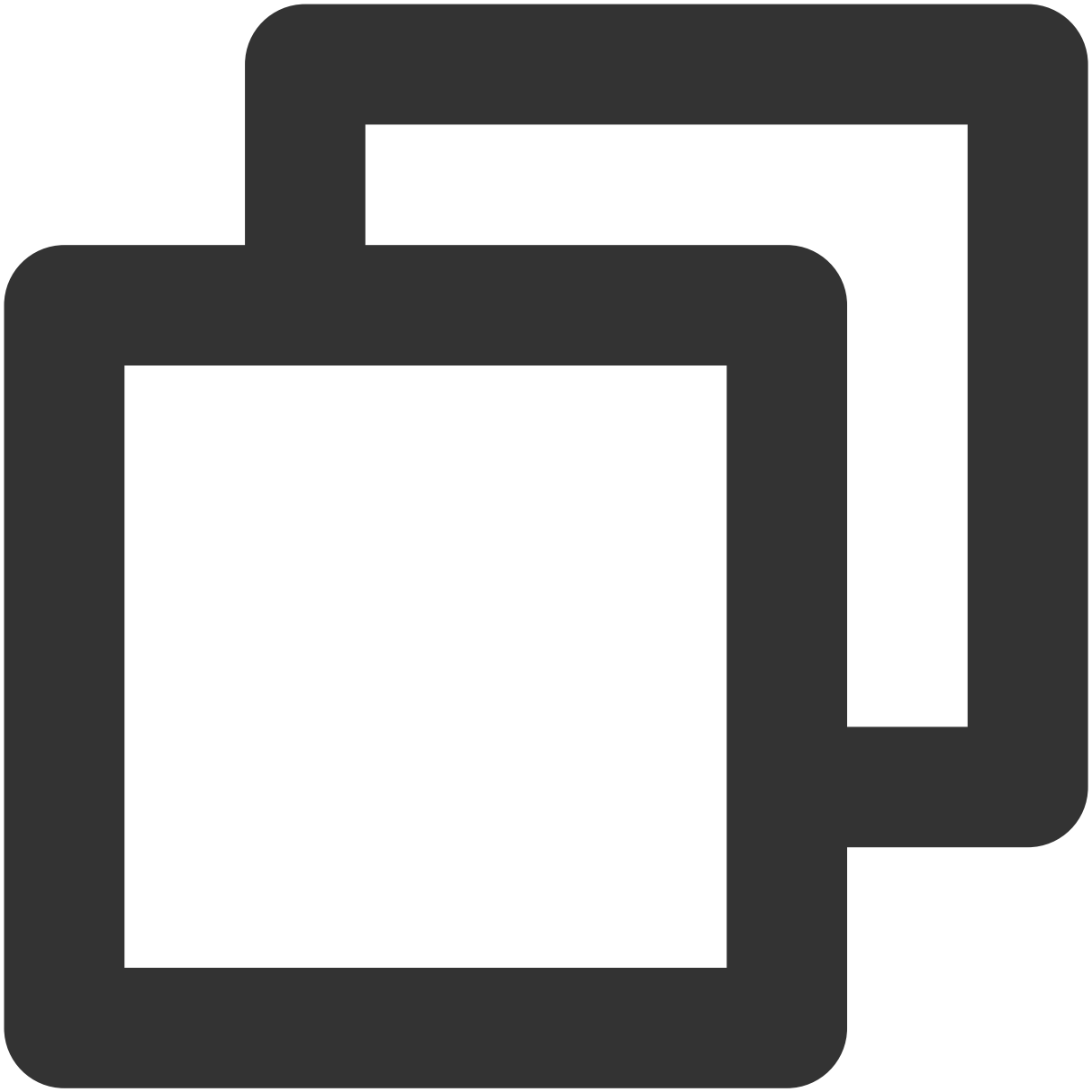
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/contrast/<value> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



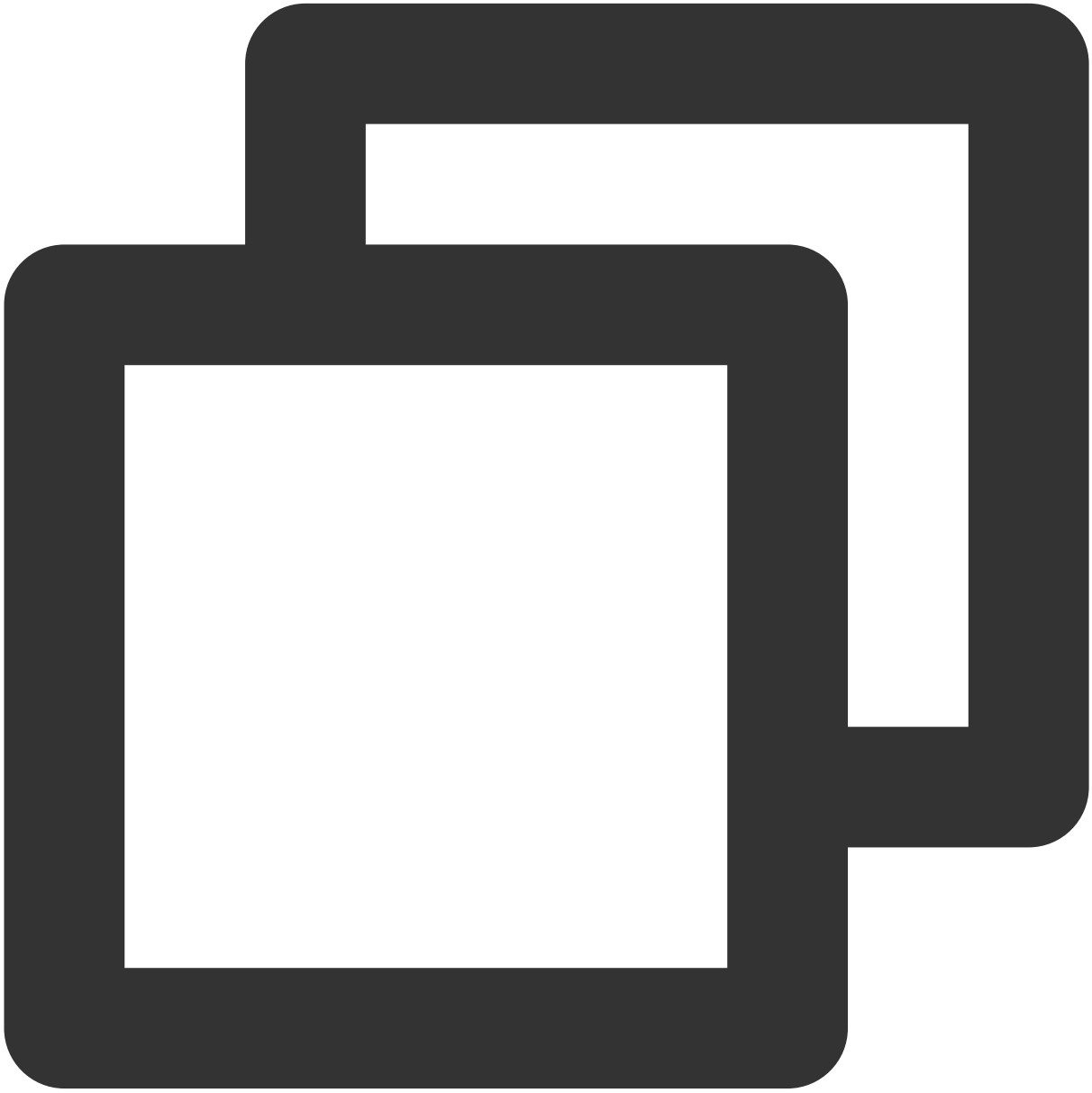
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/contrast/<value>"
  }]
```

```
}
```

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/contrast/<value>"
  }]
}
```

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：contrast。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/contrast/<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、参数超限等导致处理失败的场景，会直接返回原图而不报错。

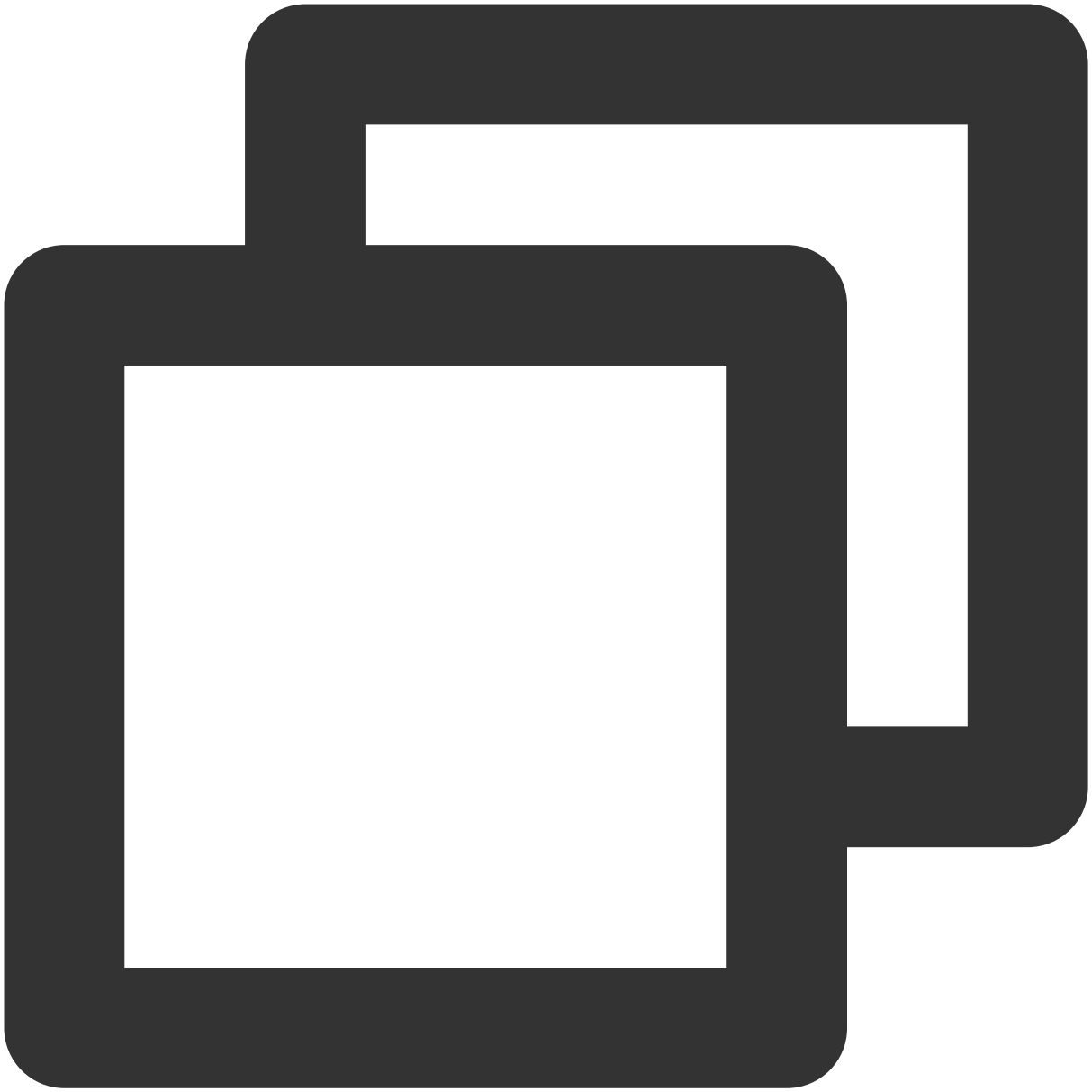
实际案例

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：



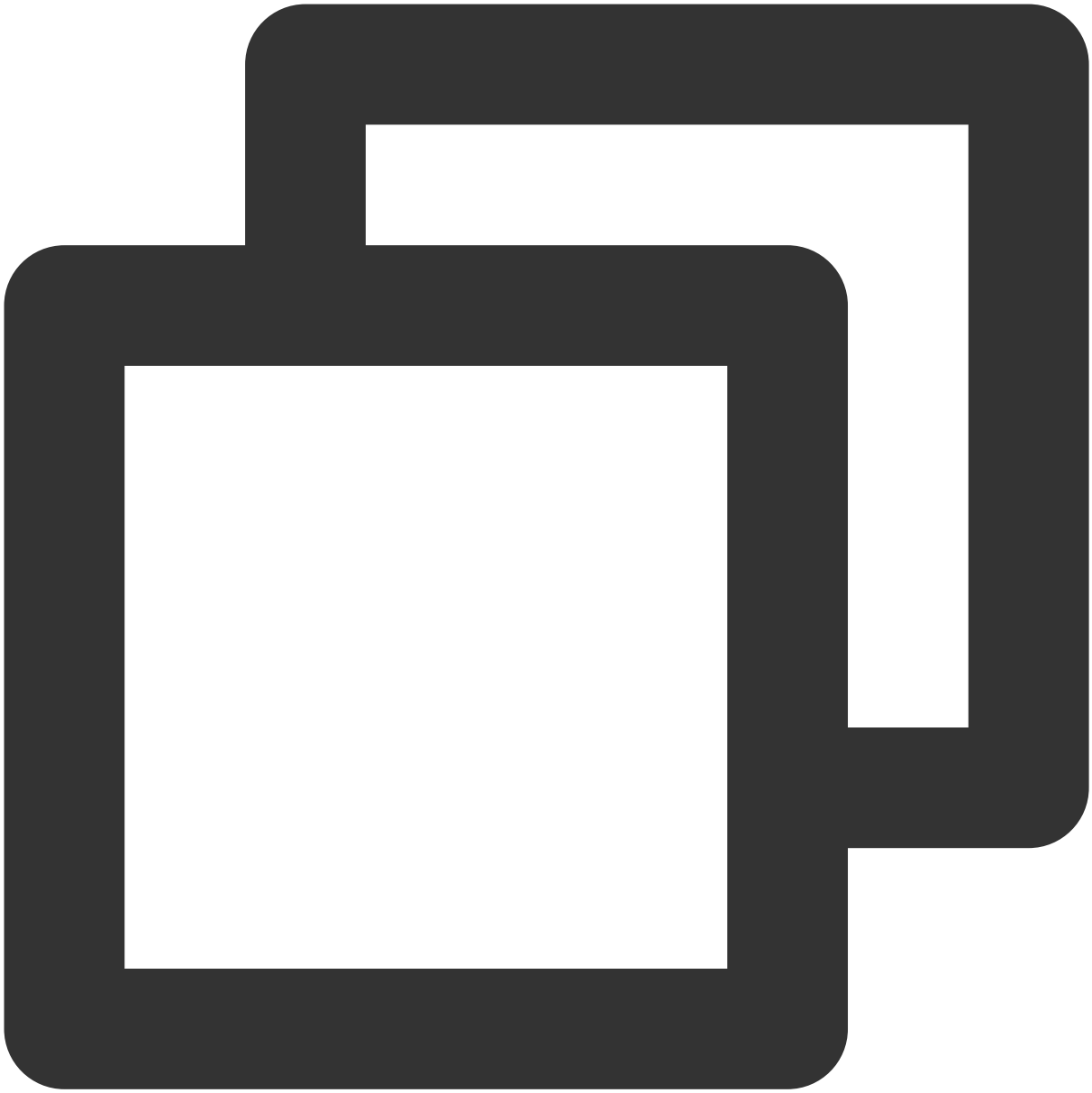
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/cont
```

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：



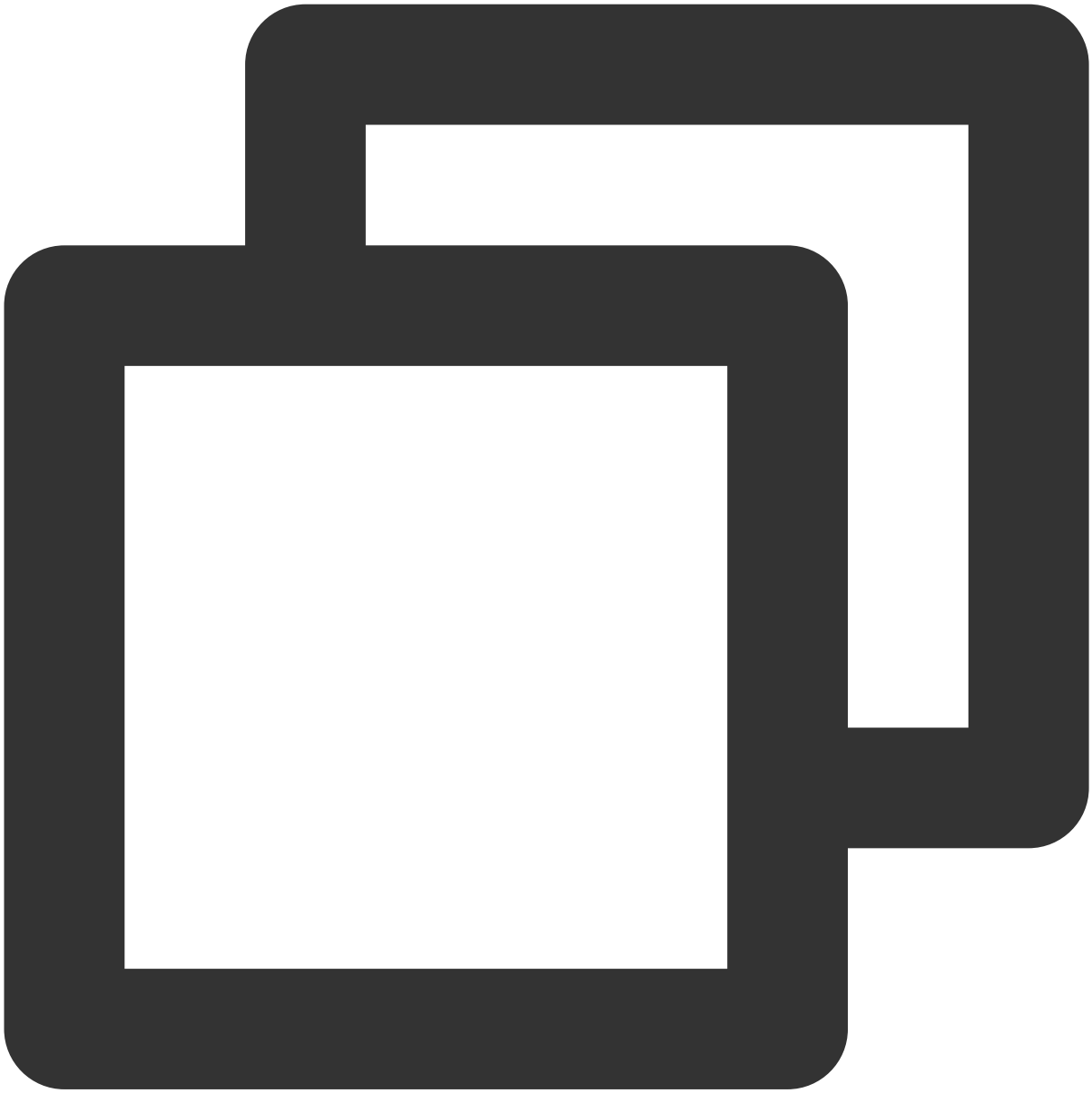
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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
```

锐化

最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象通过 **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

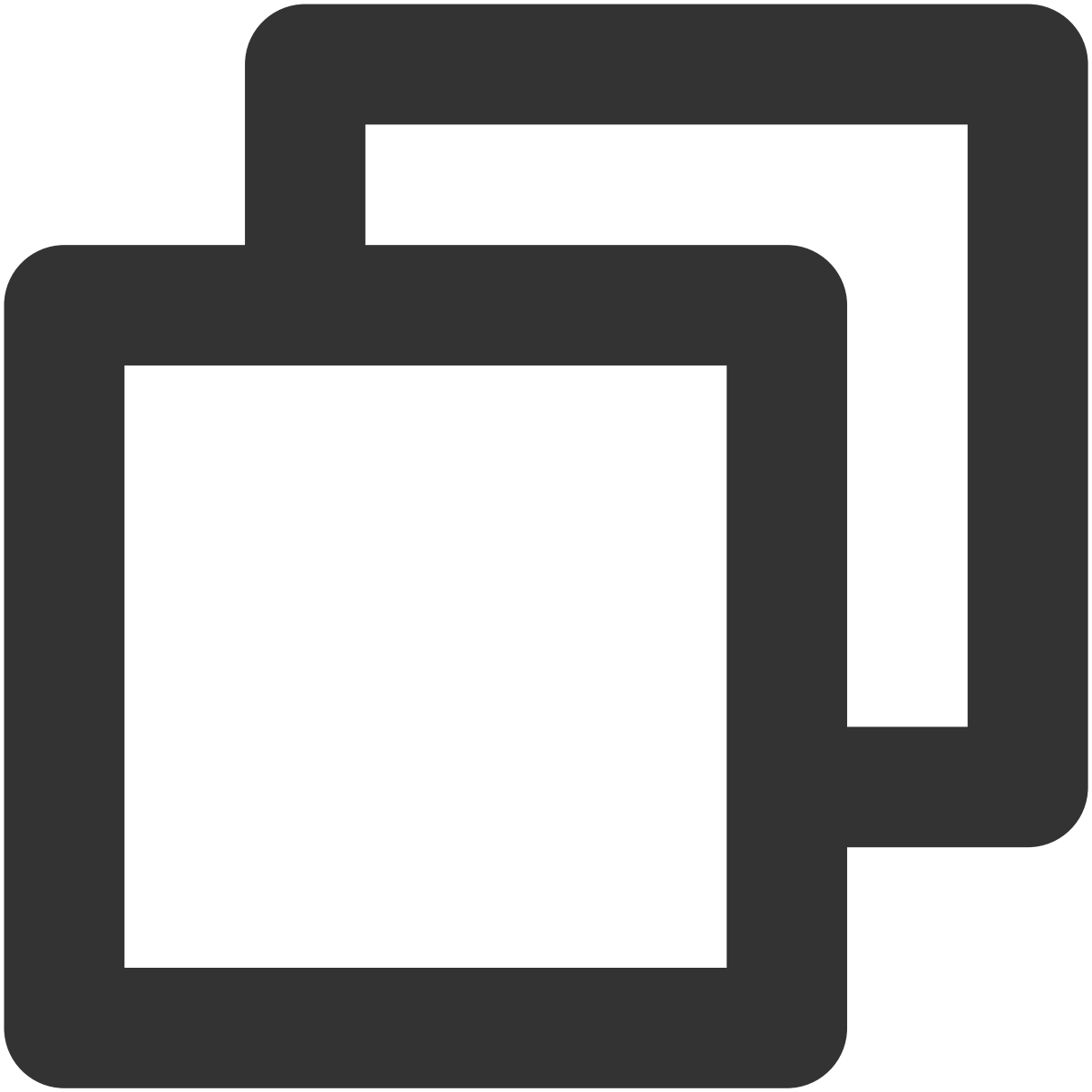
限制说明

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

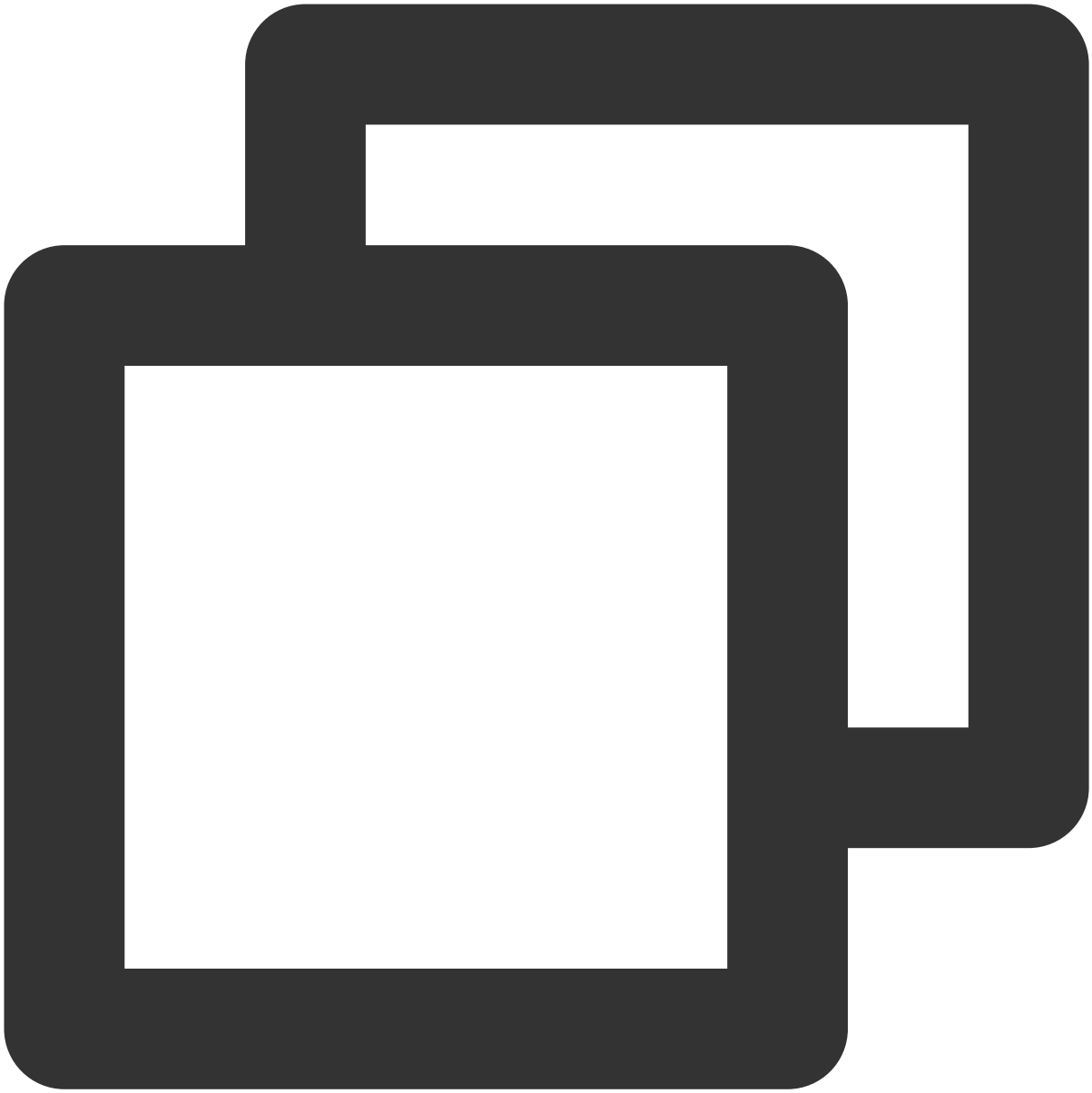
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/sharpen/<value> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



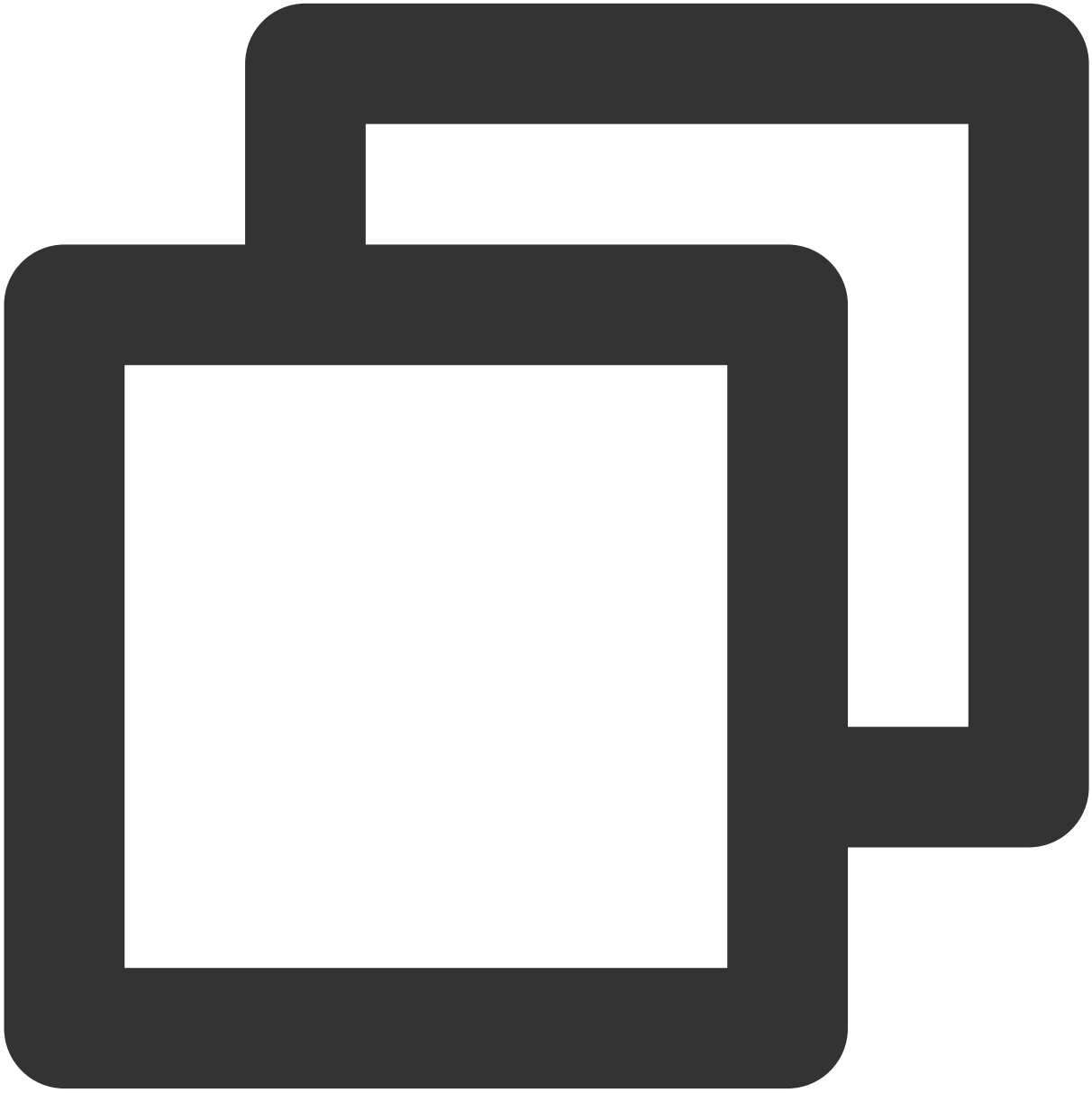
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/sharpen/<value>"
  }]
```

```
}
```

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/sharpen/<value>"
  }]
}
```

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：sharpen。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/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、参数超限等导致处理失败的场景，会直接返回原图而不报错。

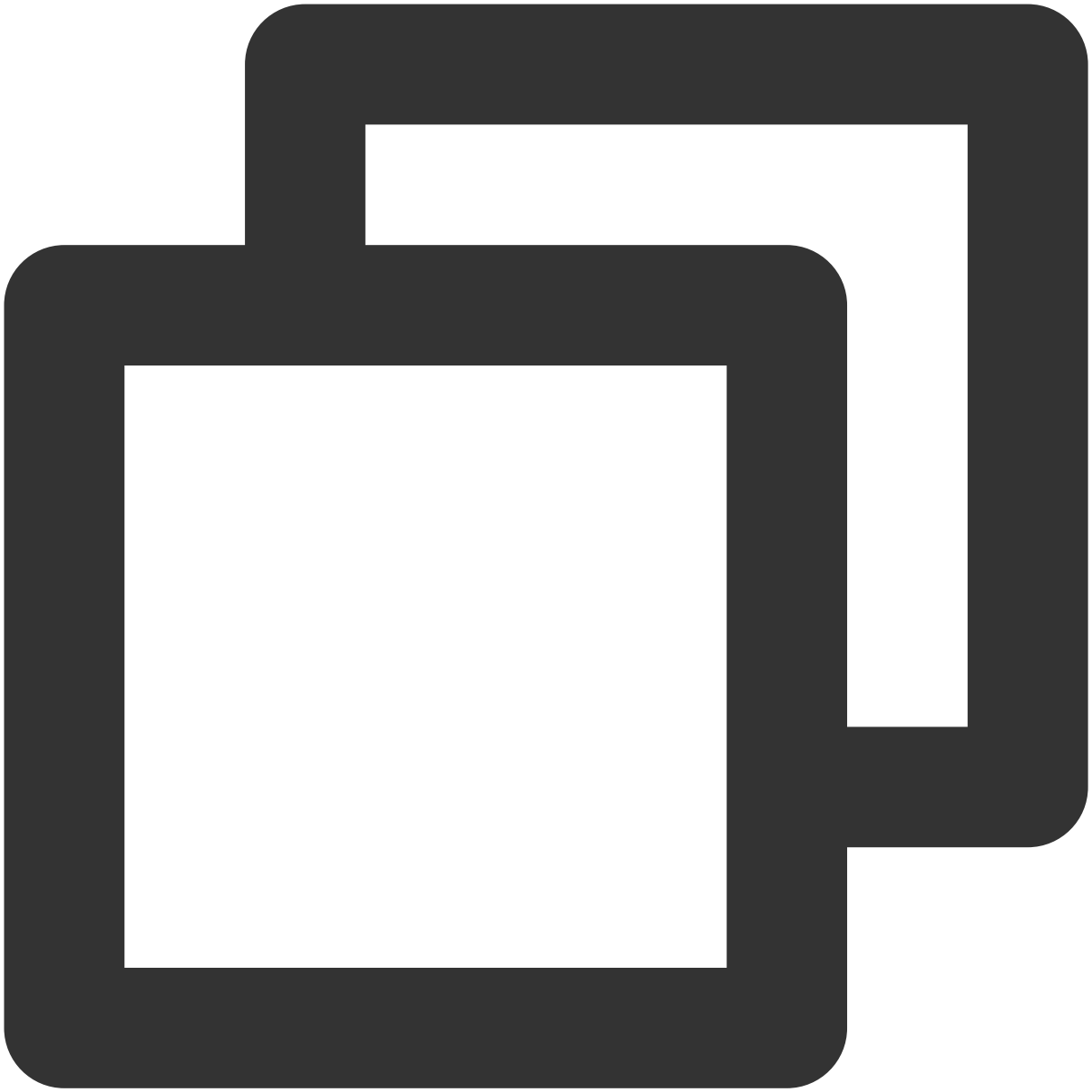
实际案例

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：



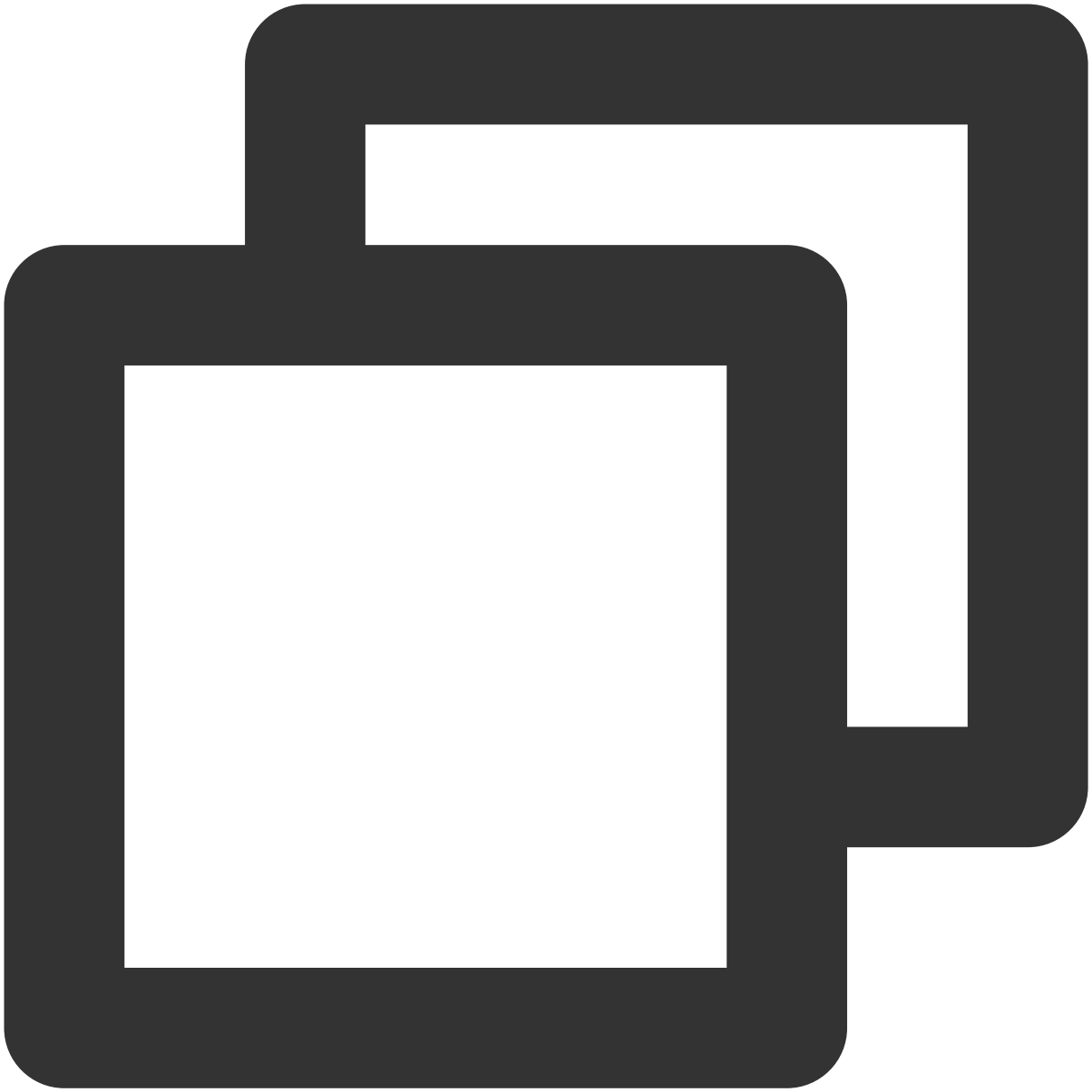
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/shar
```

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：



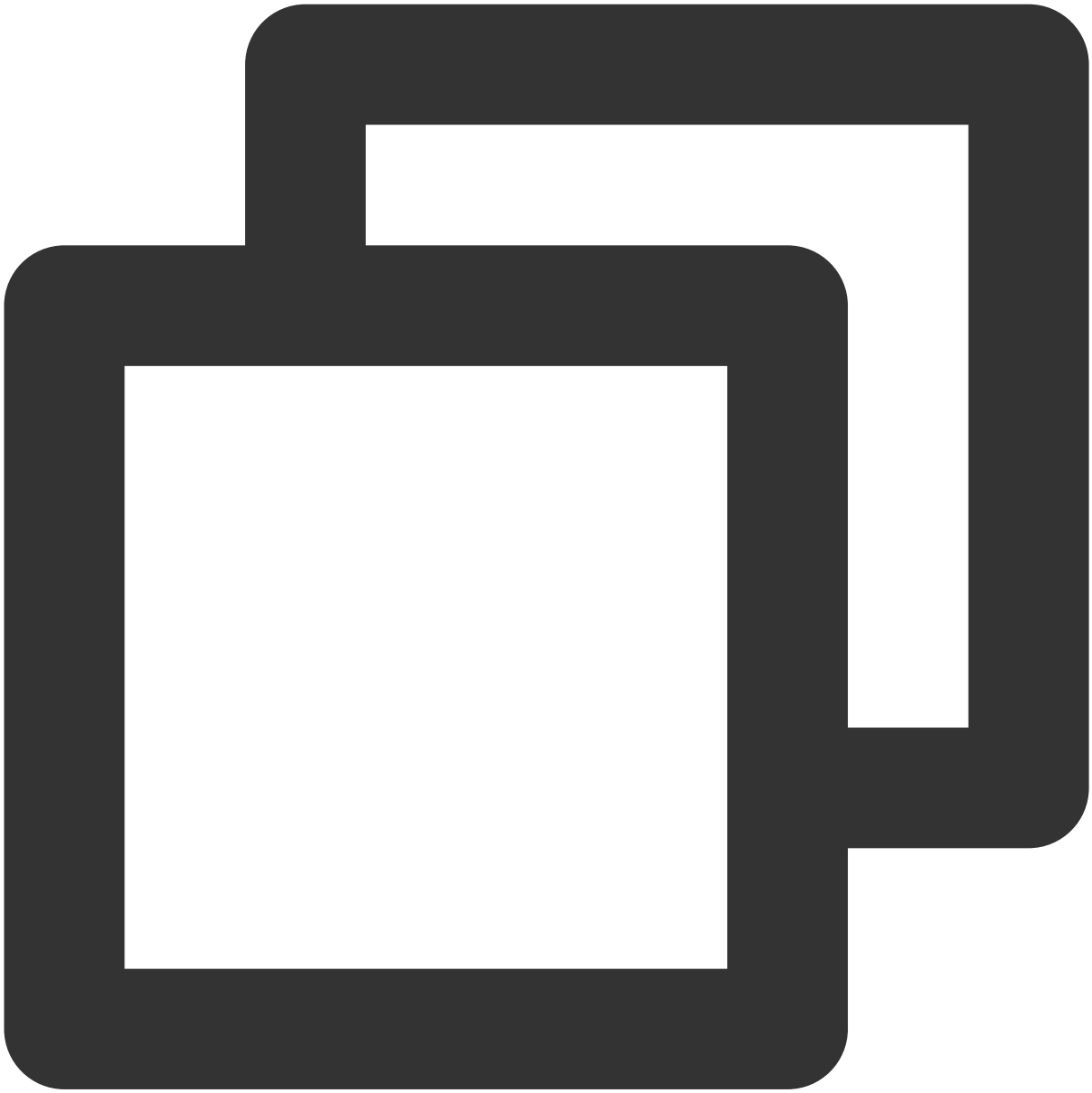
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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

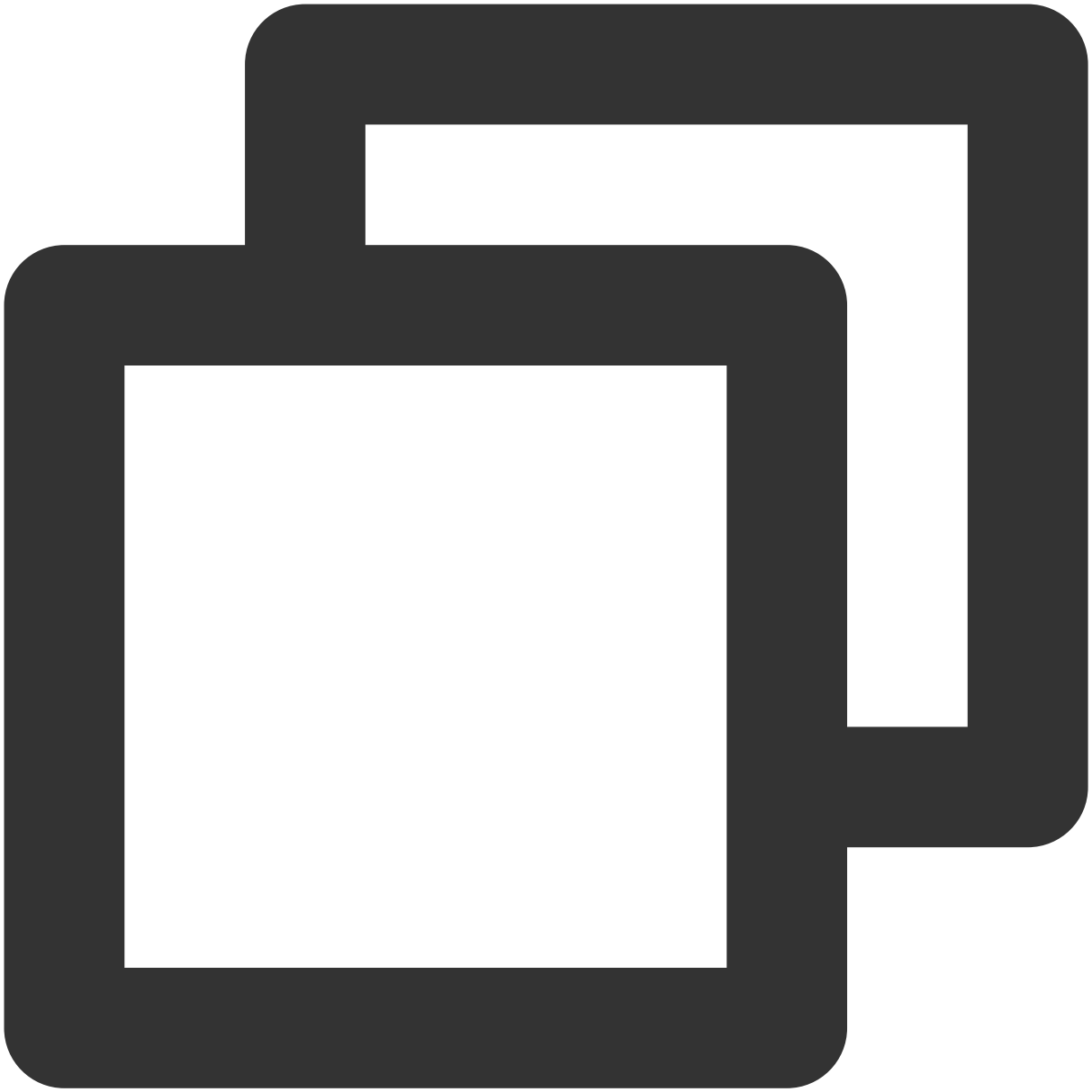
最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象通过 **imageMogr2/grayscale** 接口将图片设置为灰度图。

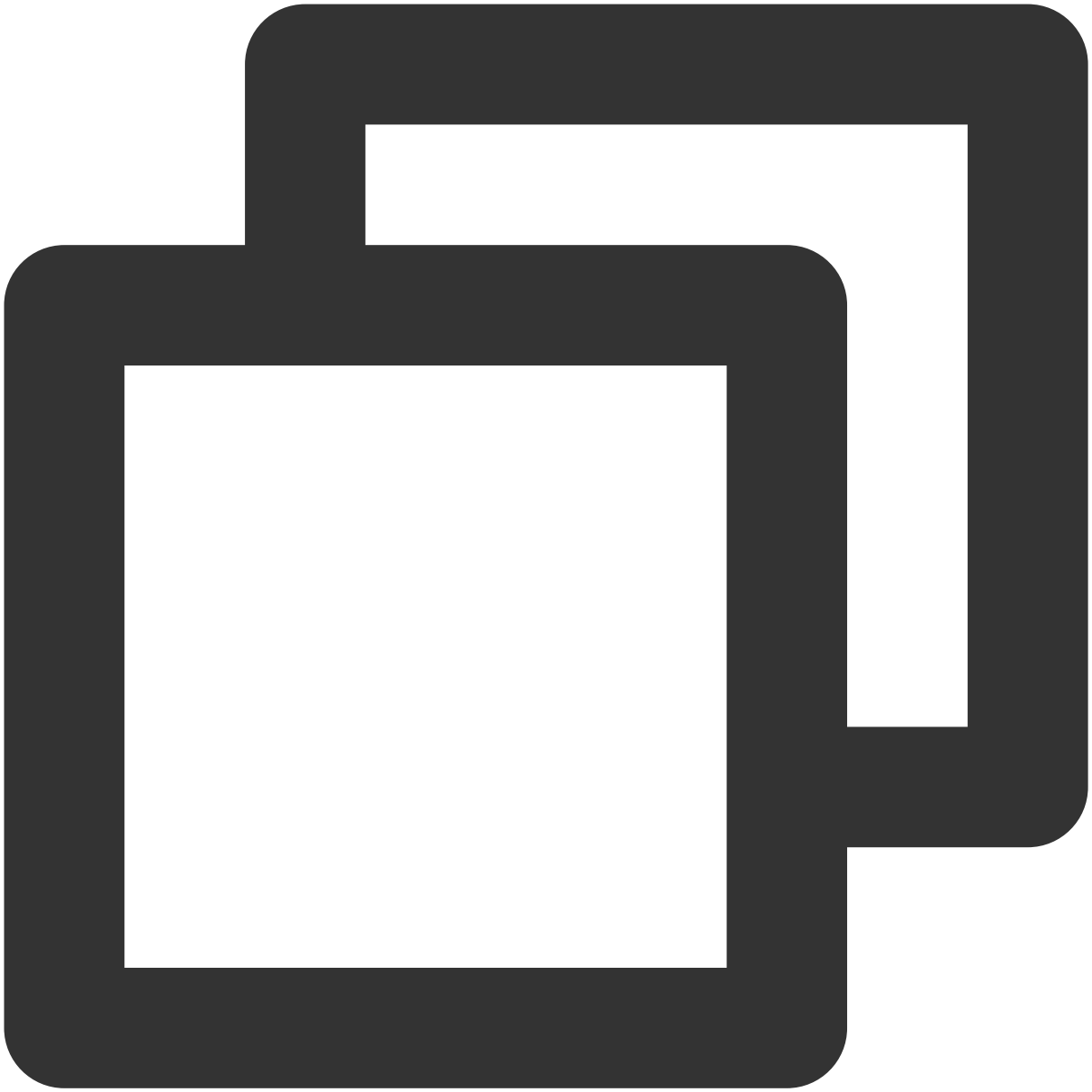
接口示例

1. 下载时处理



```
GET /<ObjectKey>?imageMogr2/grayscale/<value> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



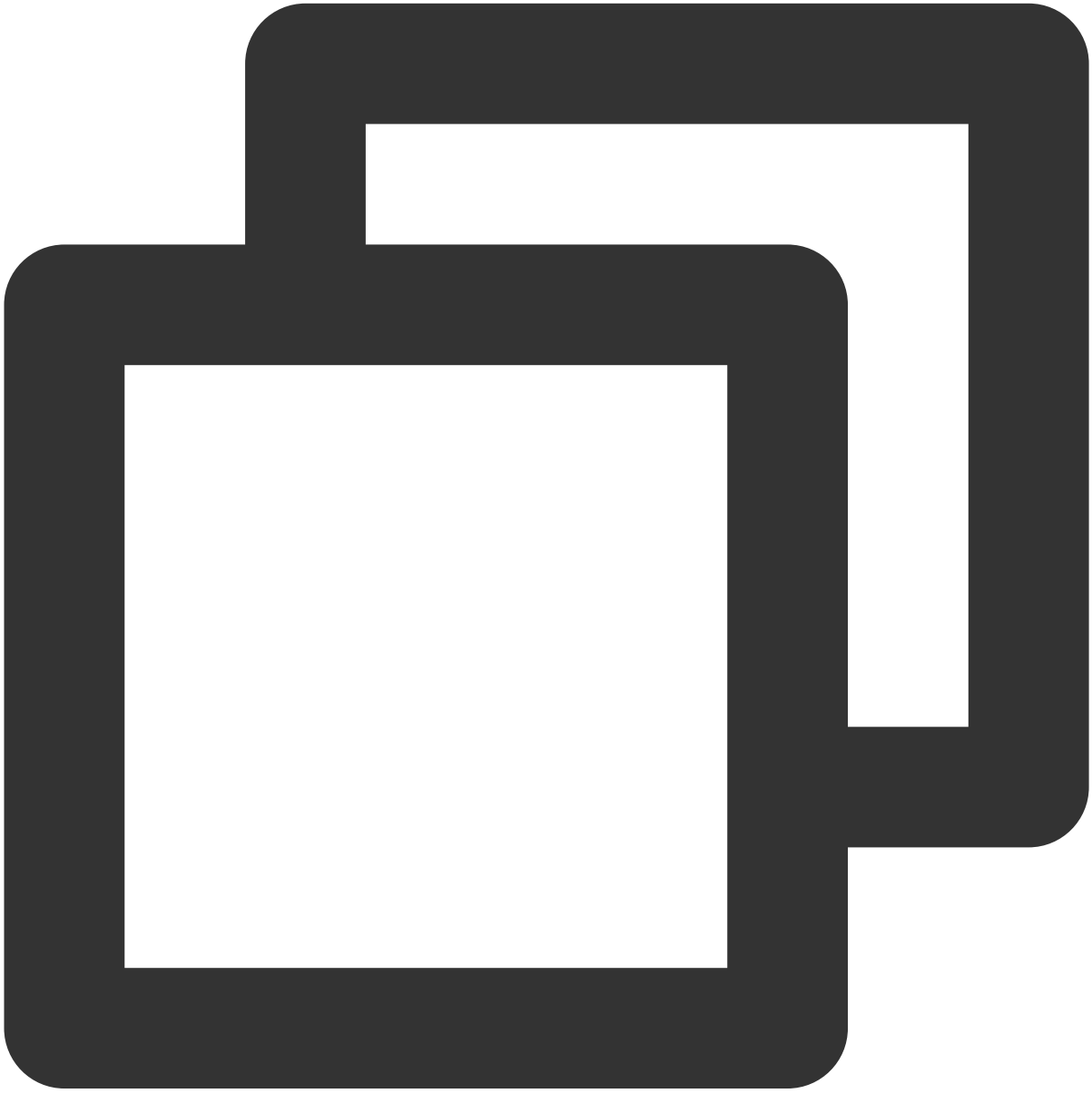
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/grayscale/<value>"
  }]
```

```
}
```

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/grayscale/<value>"
  }]
}
```

处理参数说明

操作名称：grayscale。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/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、参数超限等导致处理失败的场景，会直接返回原图而不报错。

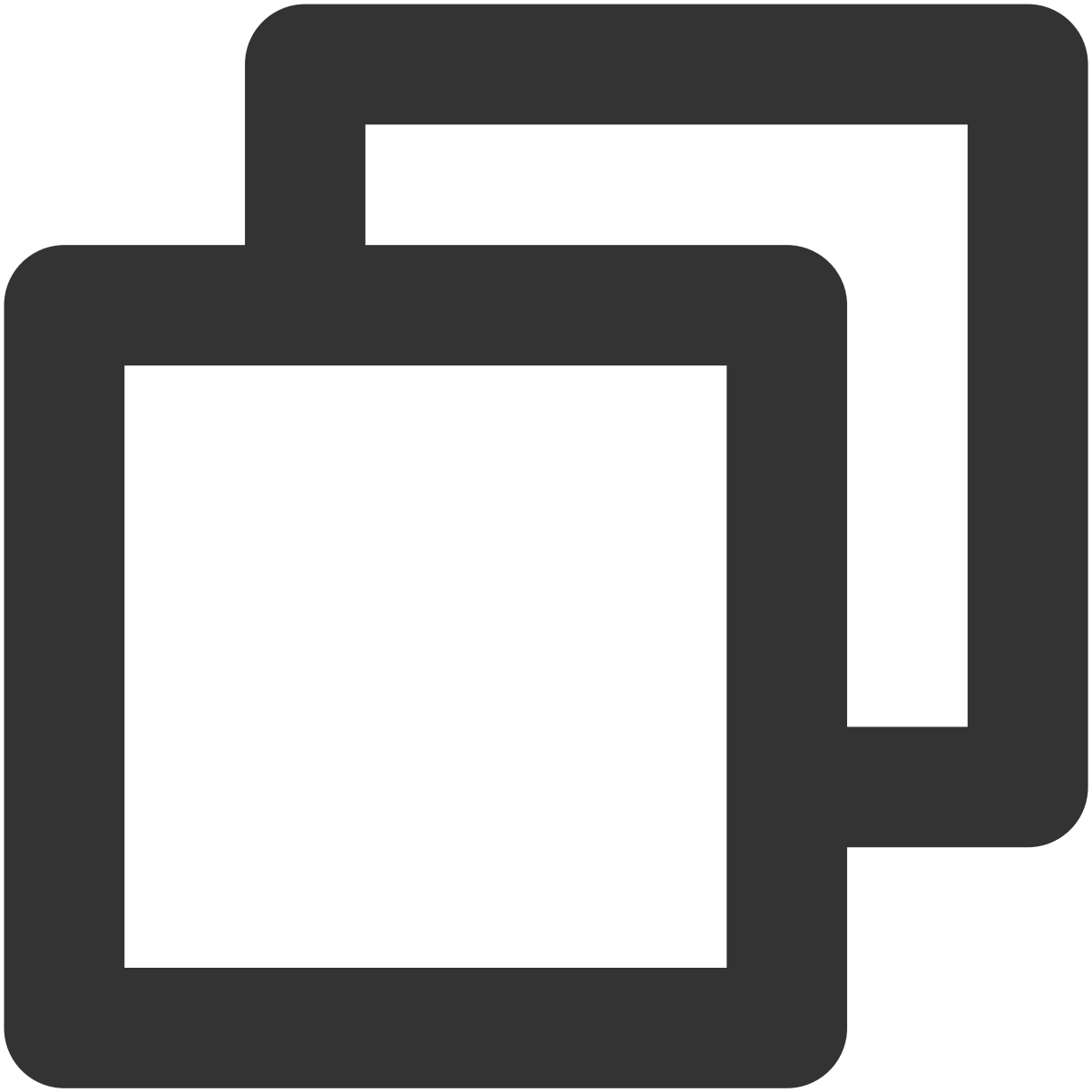
实际案例

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：转为灰度图

将图片变为灰度图，示例如下：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/gray
```

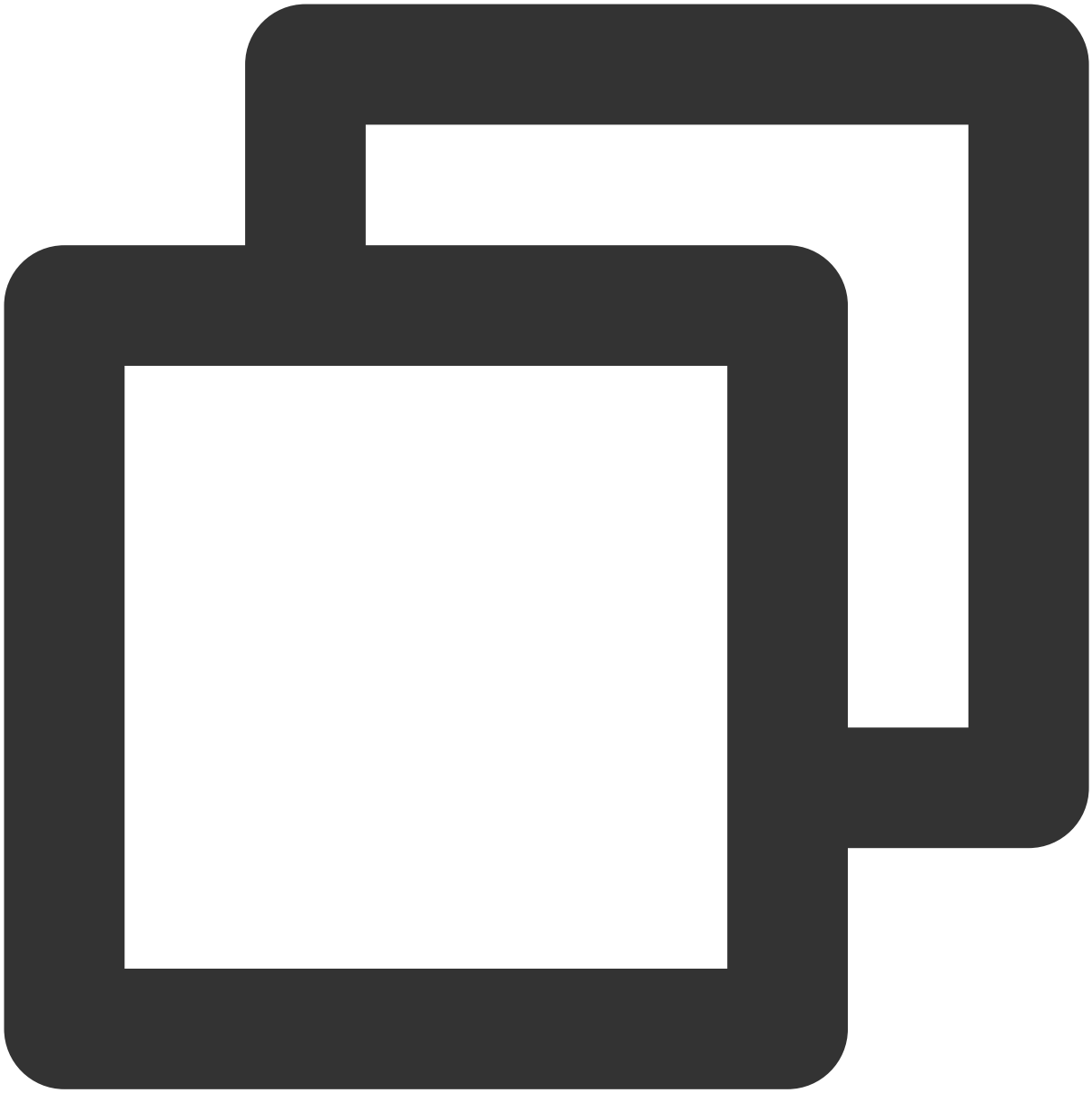
最终效果如下：



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



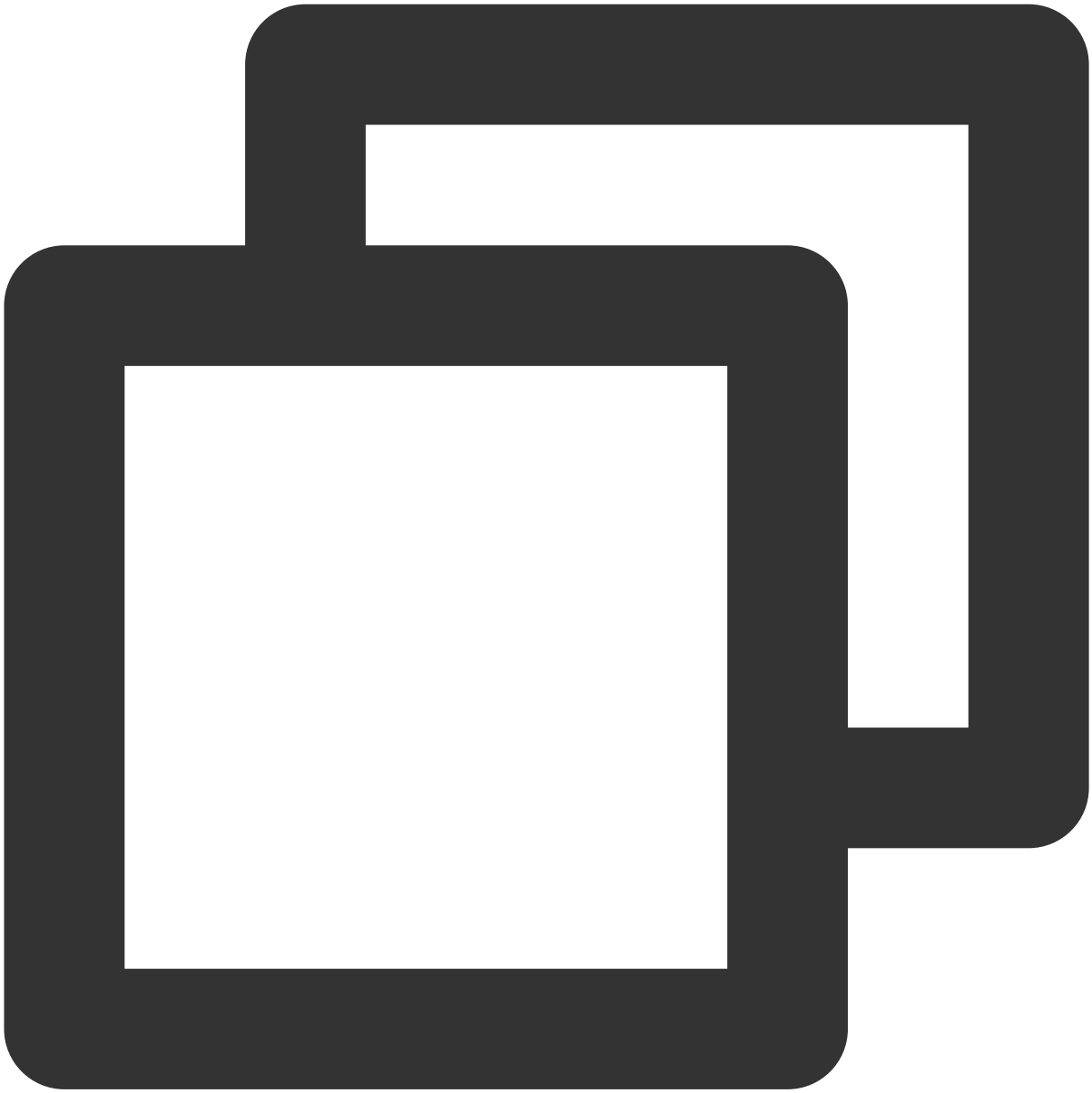
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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
```

图片水印

最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象通过 **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中的图片。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

说明：

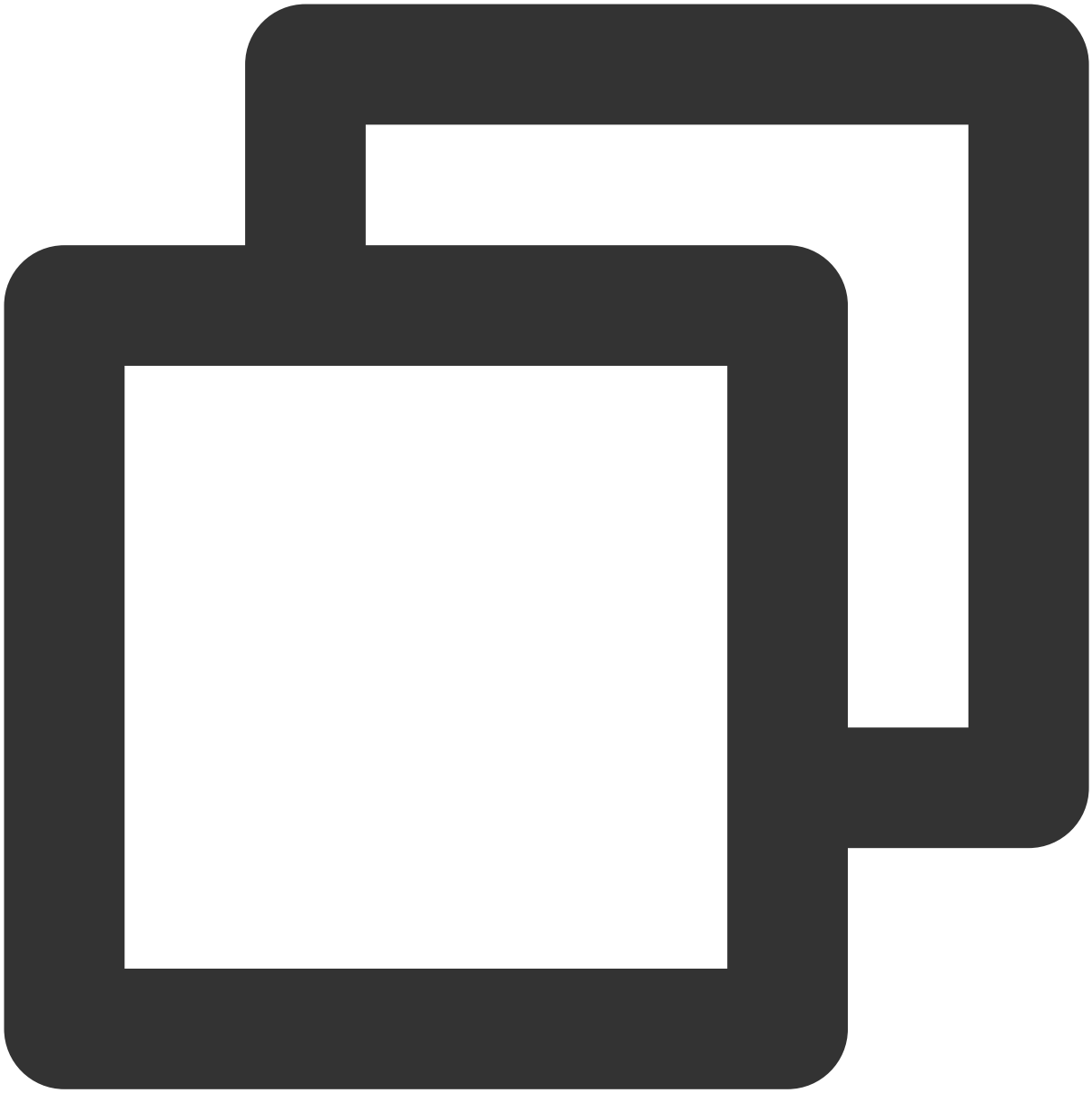
一张图片上，最多添加10张不同的图片水印。

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

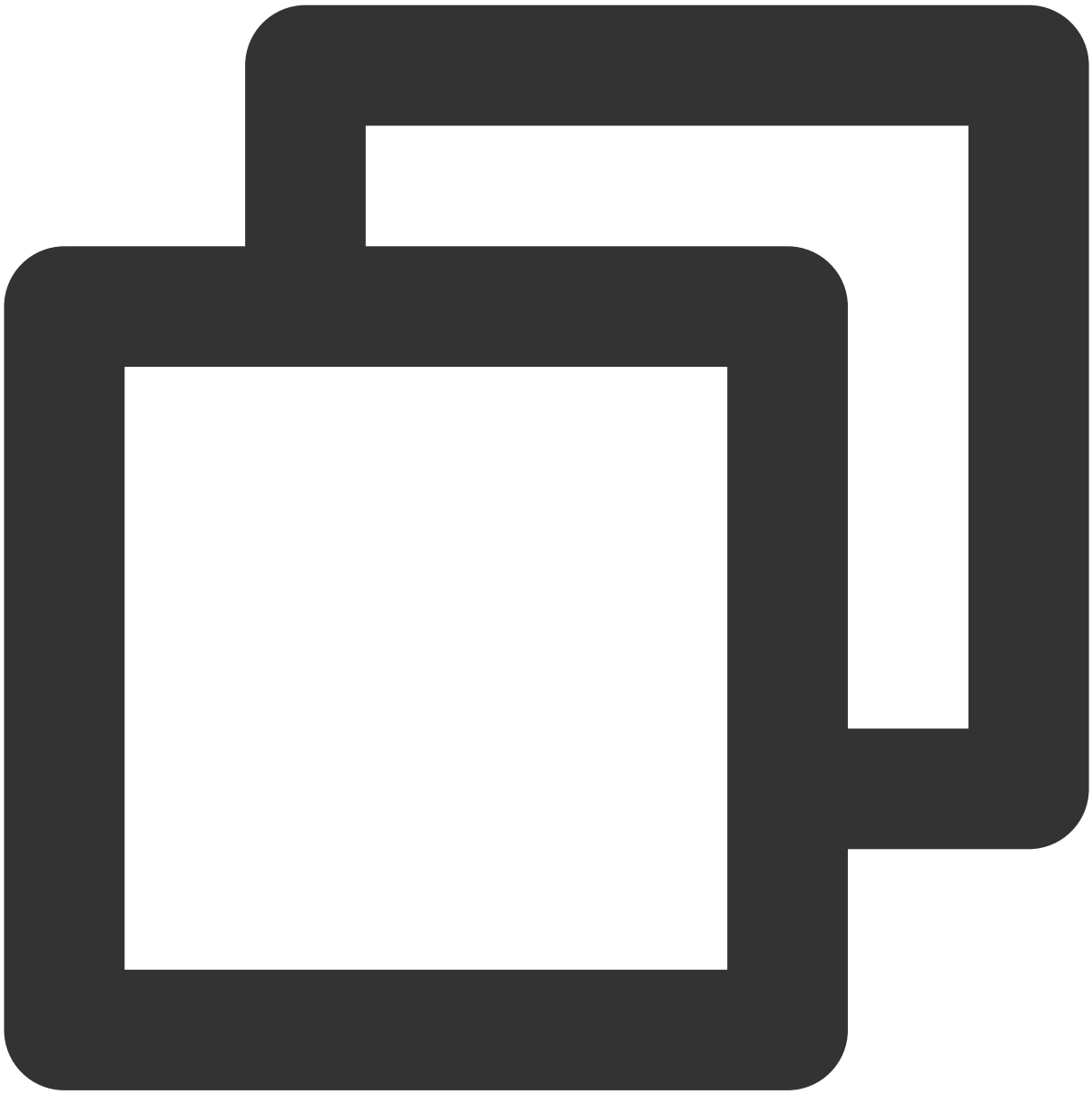


```
GET /<ObjectKey>?watermark/1/image/<encodedURL>
                            /gravity/<gravity>
                            /dx/<dx>
                            /dy/<dy>
                            /blogo/<type> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明：

请忽略上面的空格与换行符。

2. 上传时处理



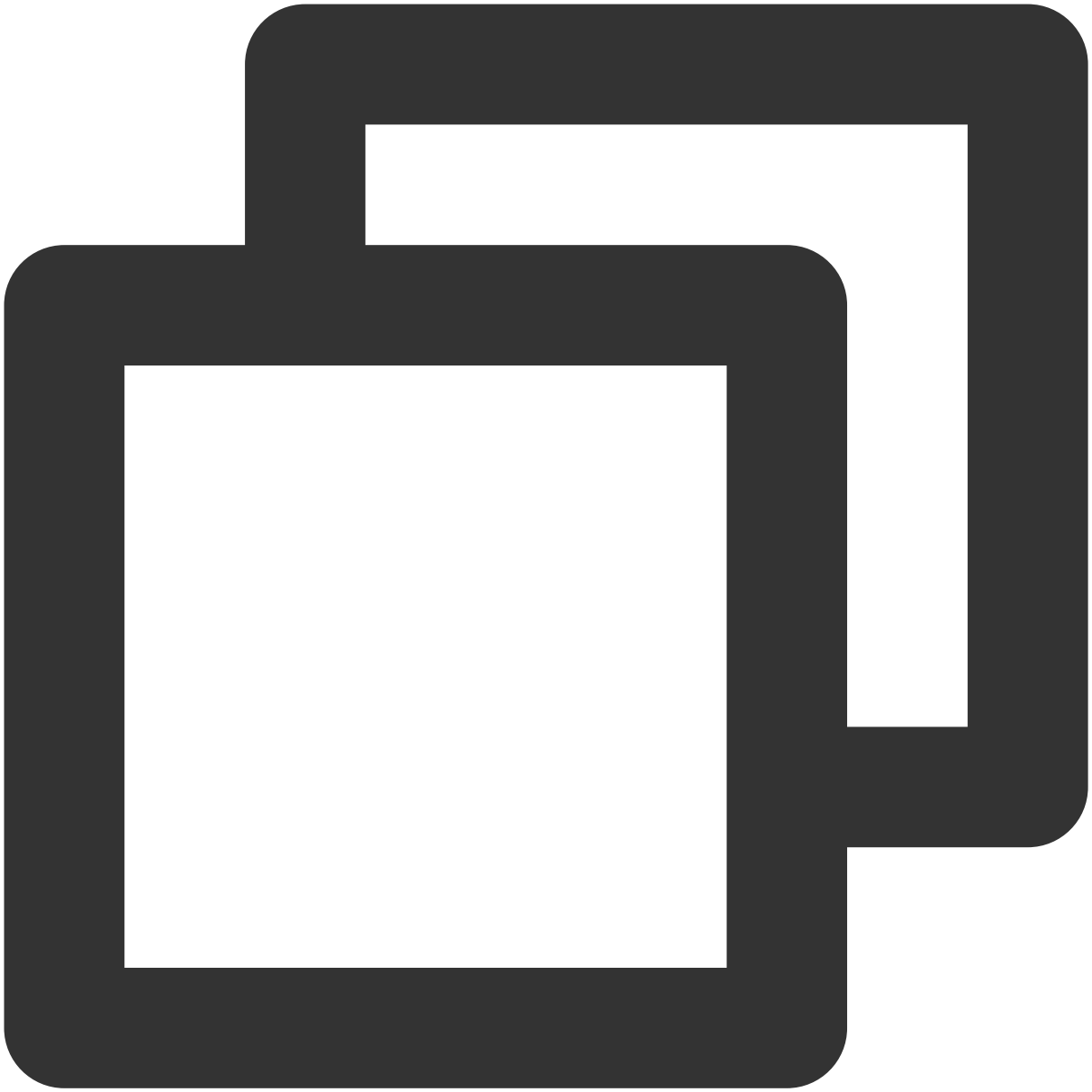
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
```

```
    "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 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）。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-125100` `shanghai.myqcloud.com/sample.jpeg?watermark/1/image/xxxxxxx/scatype/3/sp`
/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：

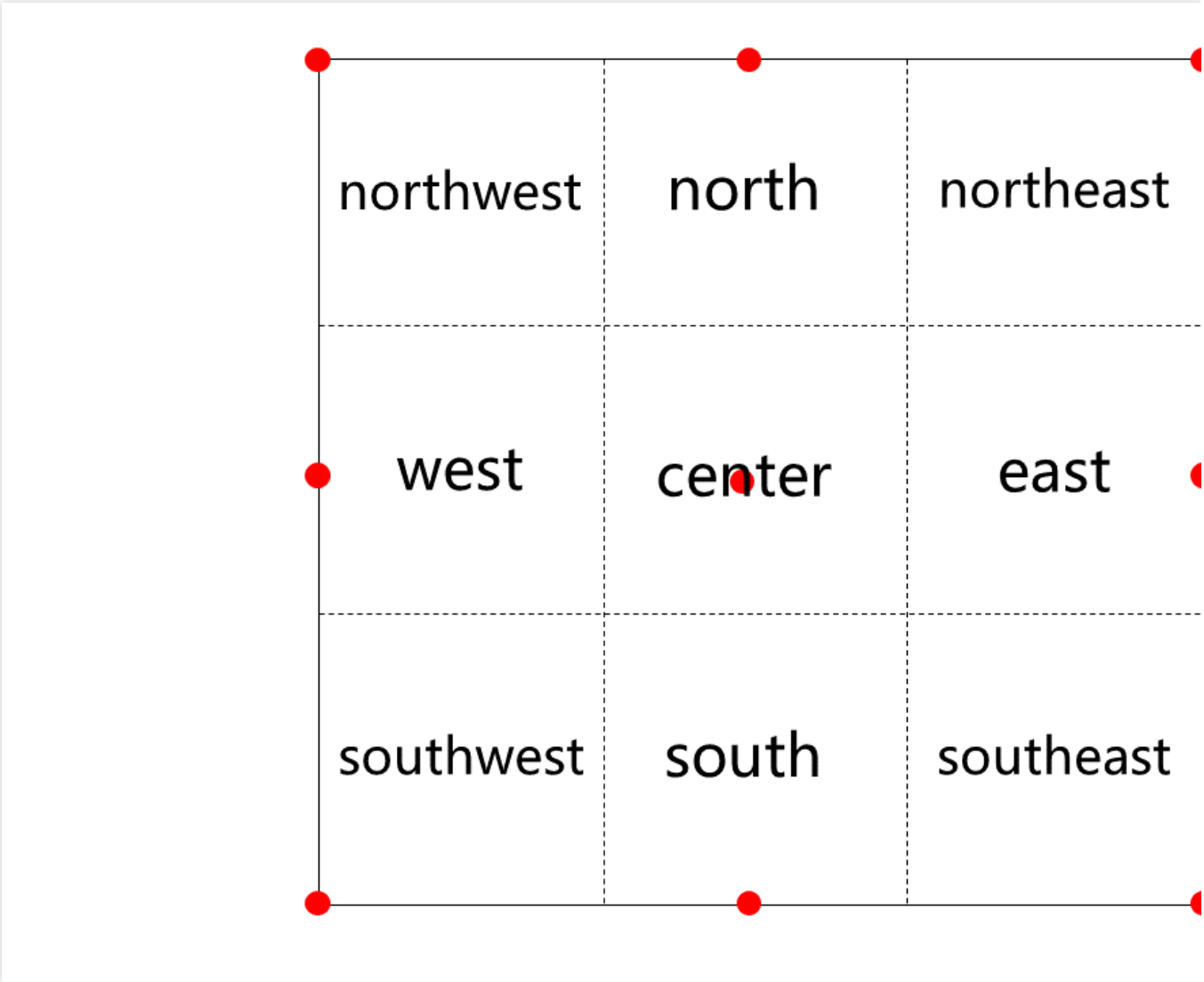
水印图片与源图片必须位于同一个存储桶下。

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）。

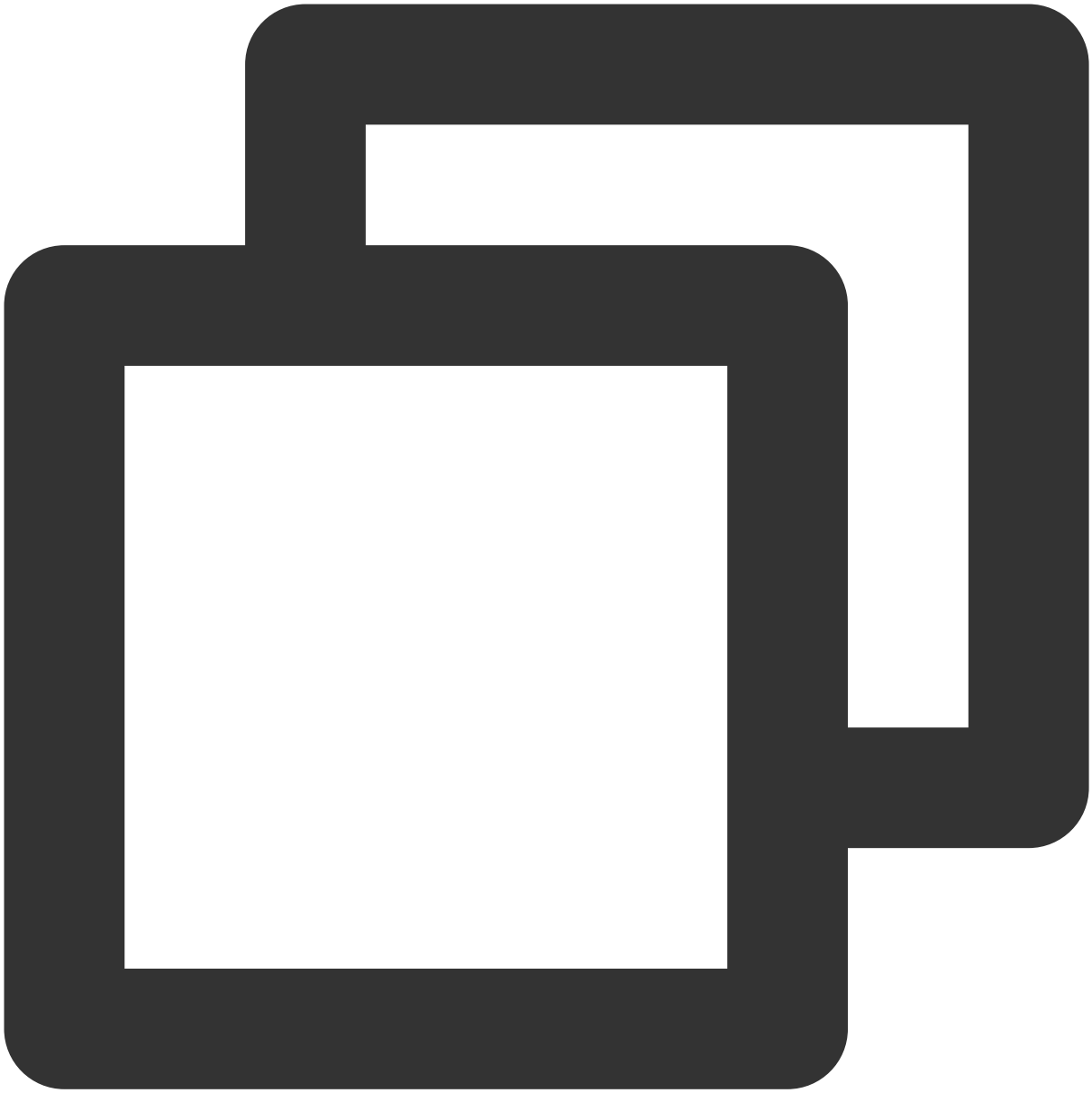
当 gravity 参数设置为 west 或 east 时，dy 参数无效（水印会垂直居中）。

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



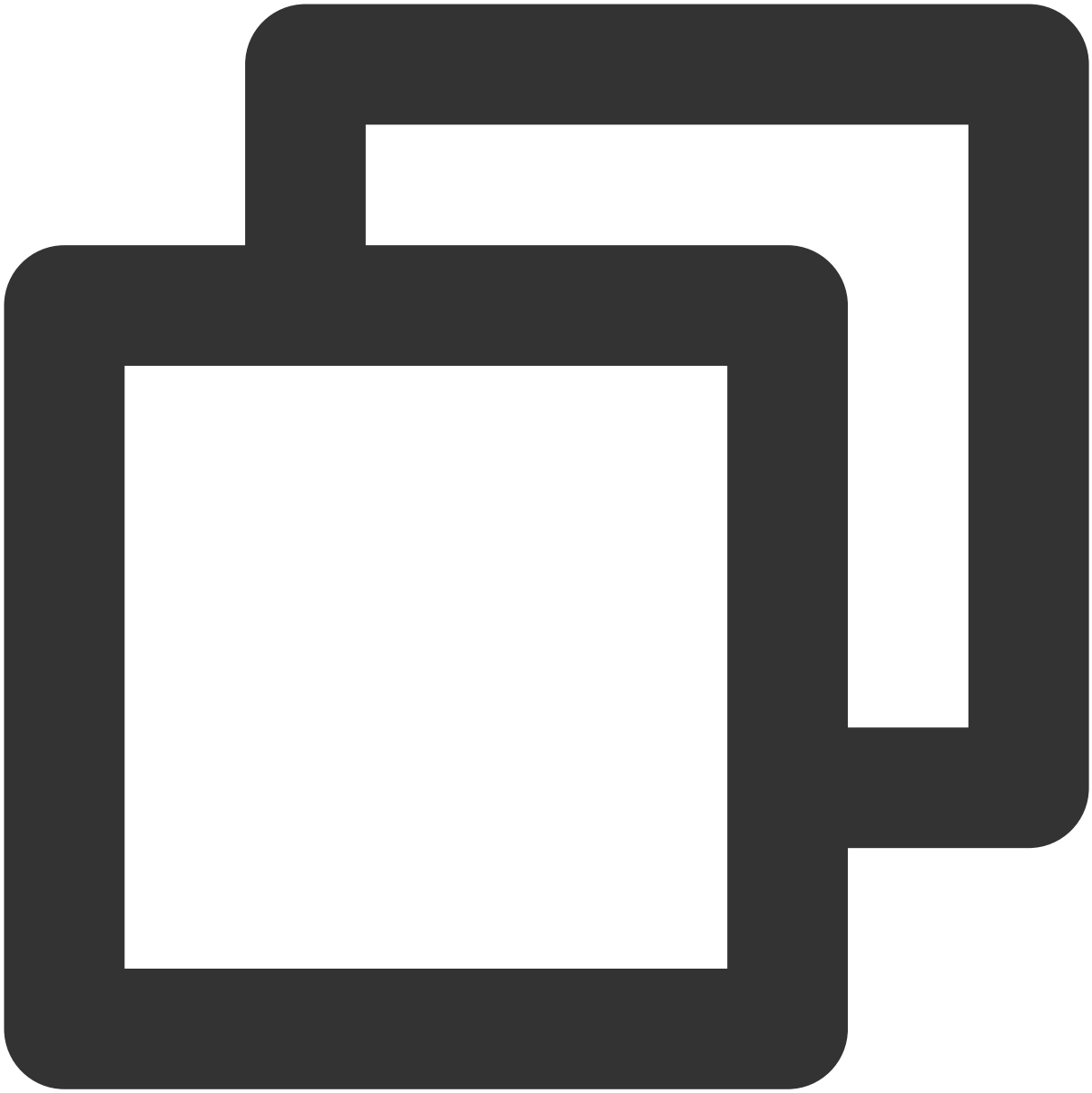
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/1/ima
```

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

版权所有：腾讯云计算（北京）有限责任公司



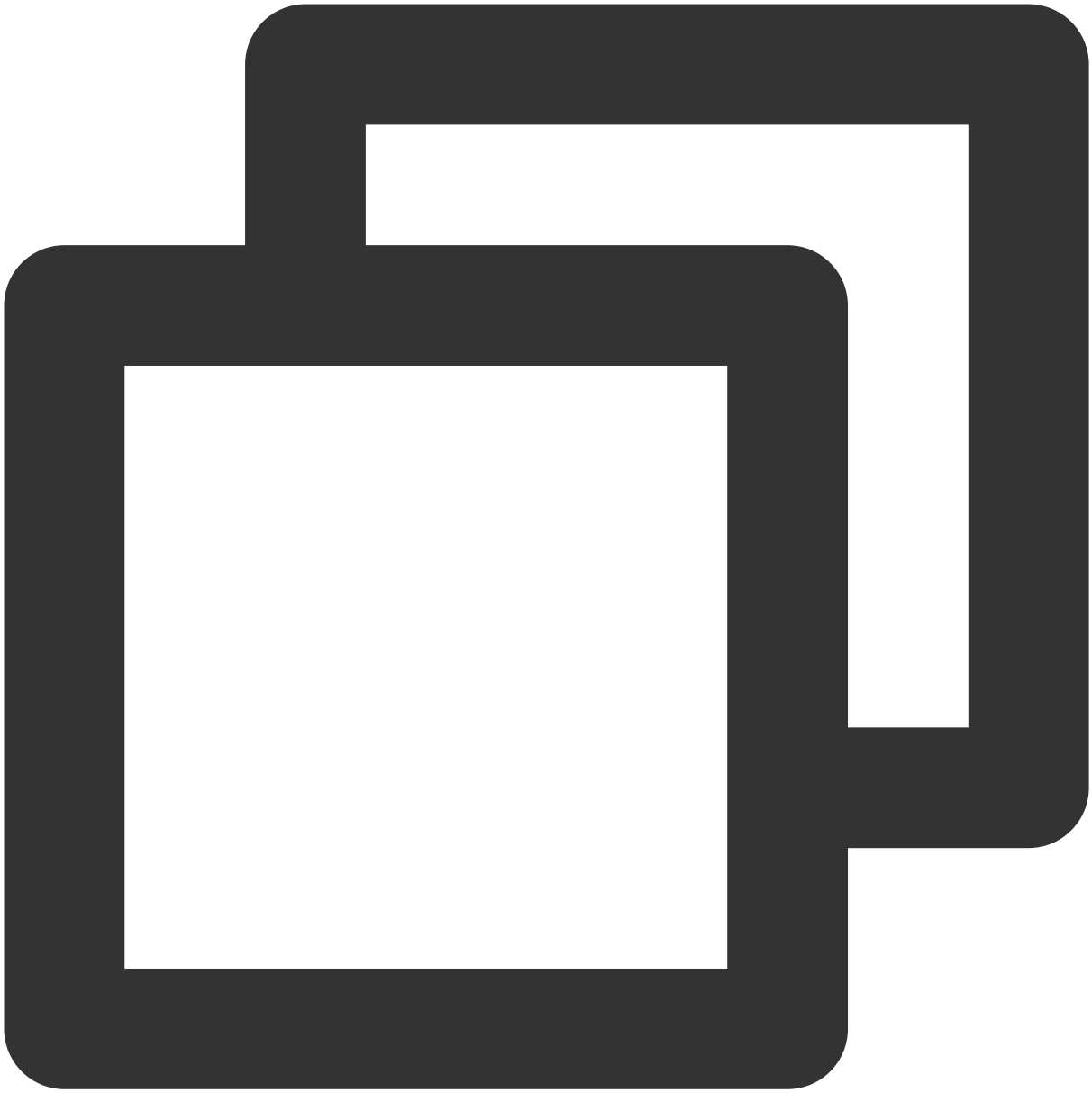
```
https://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/1/im
```

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：



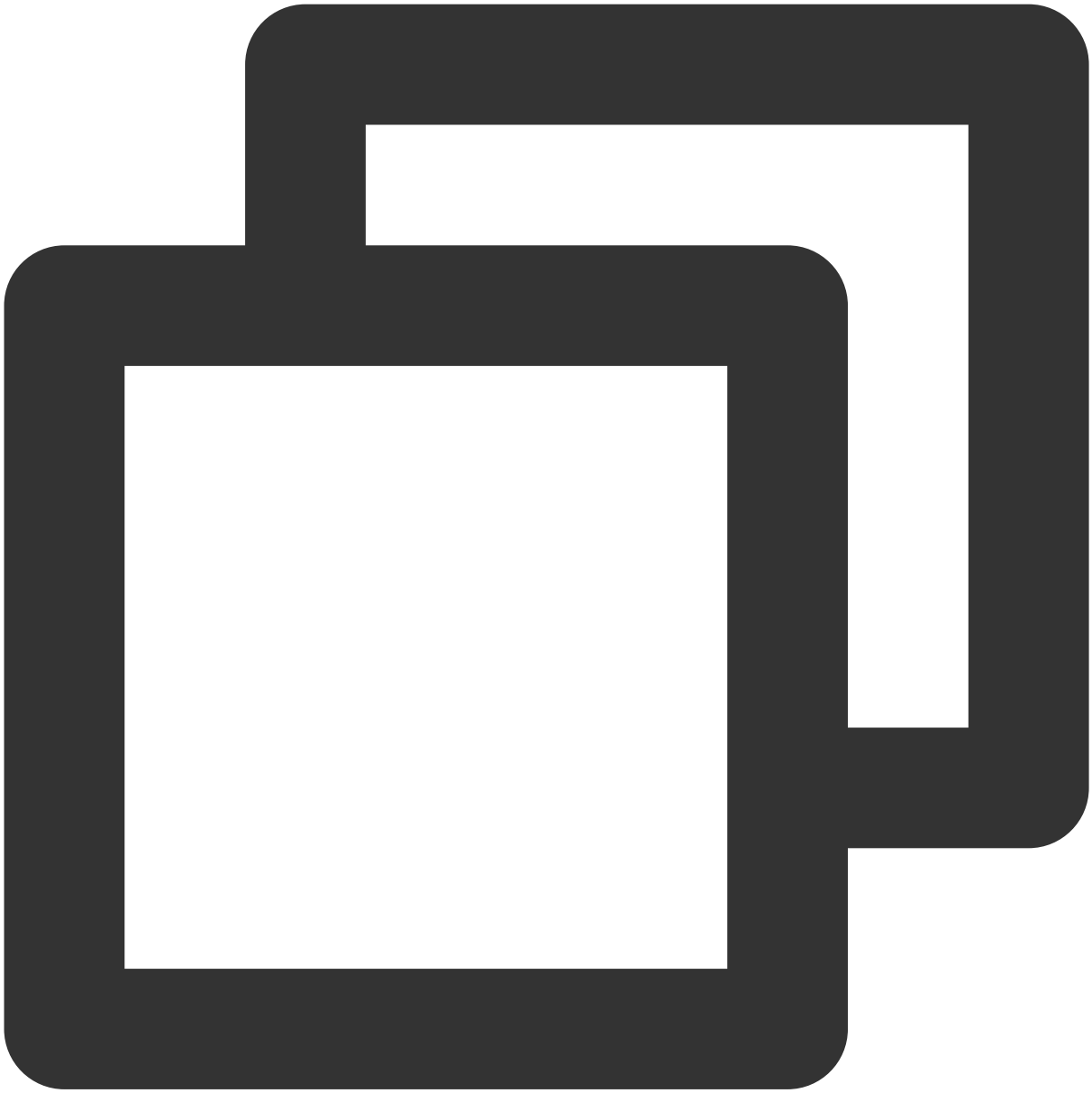
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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

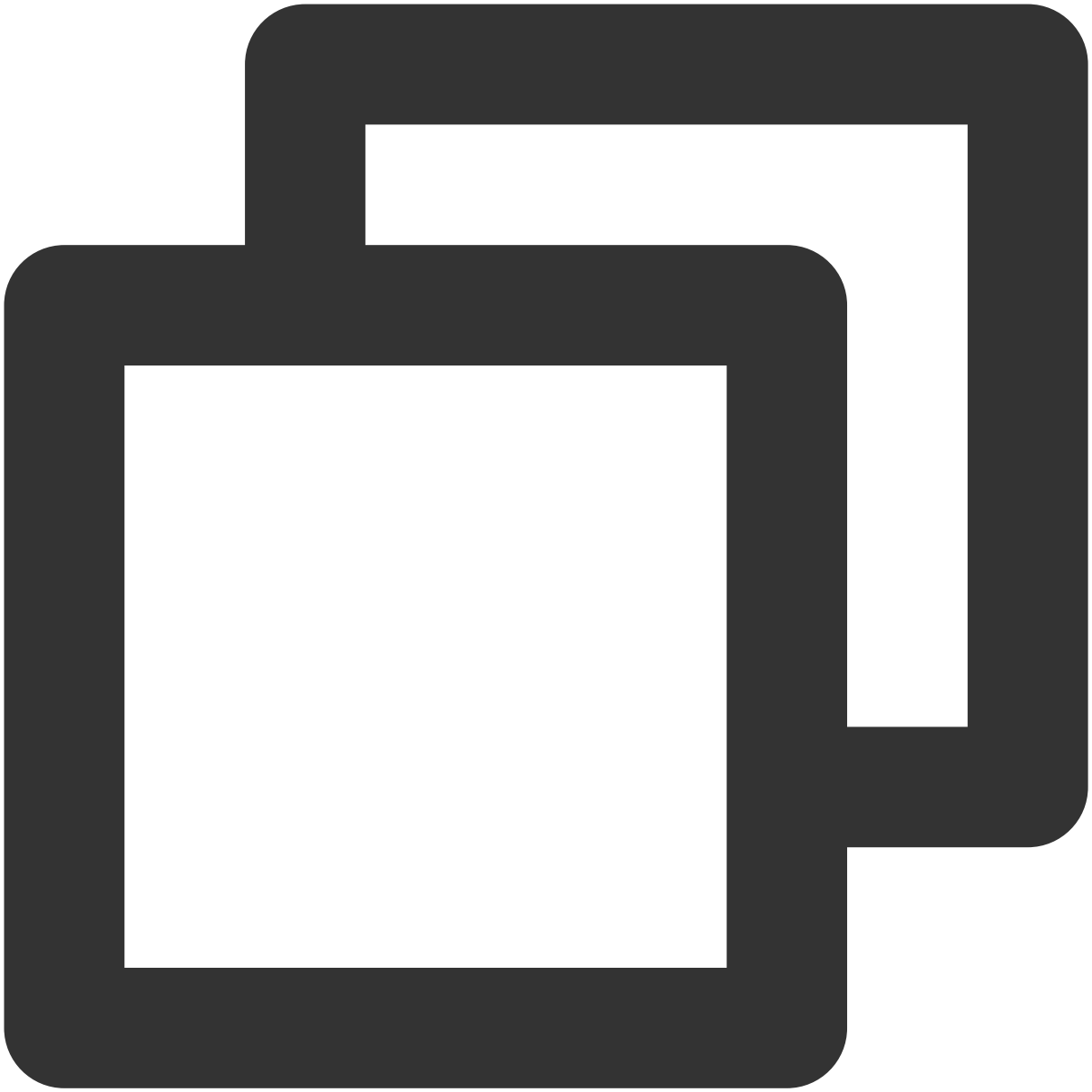
最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象通过 **watermark** 接口提供实时文字水印处理功能。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。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理

接口示例

1. 下载时处理



```
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
```

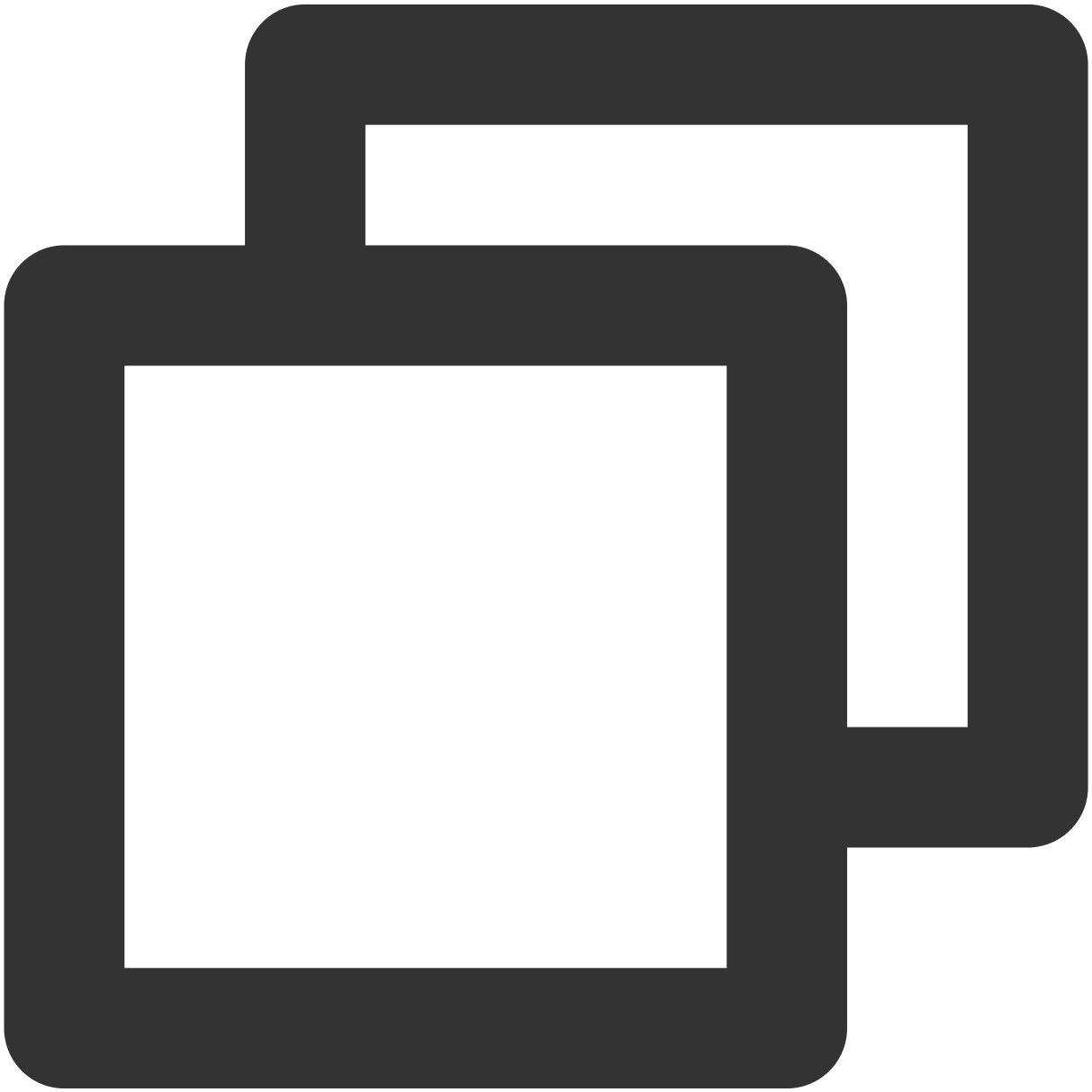
版权所有：腾讯云计算（北京）有限责任公司

```
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



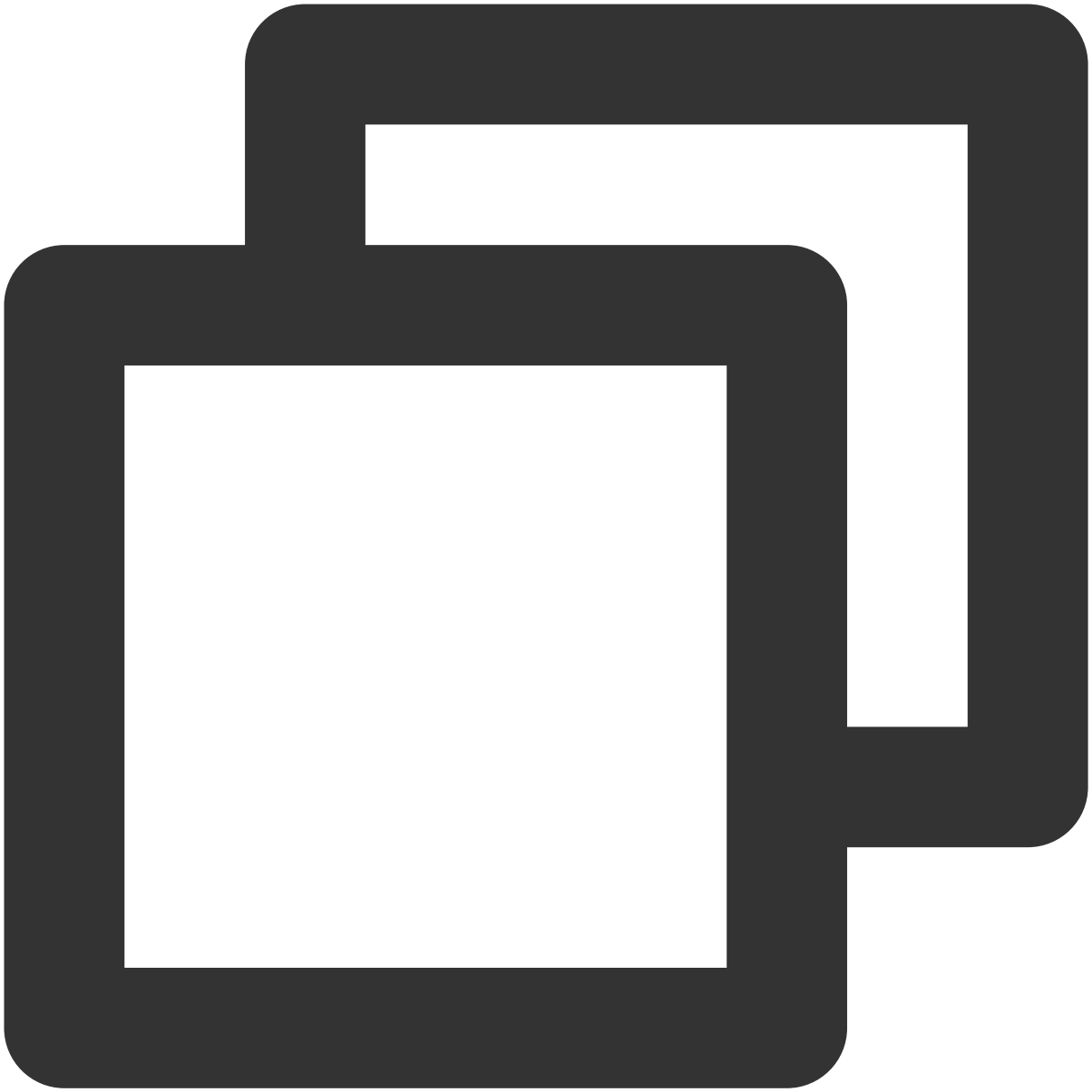
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
```

```
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，具体参数信息请参见 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

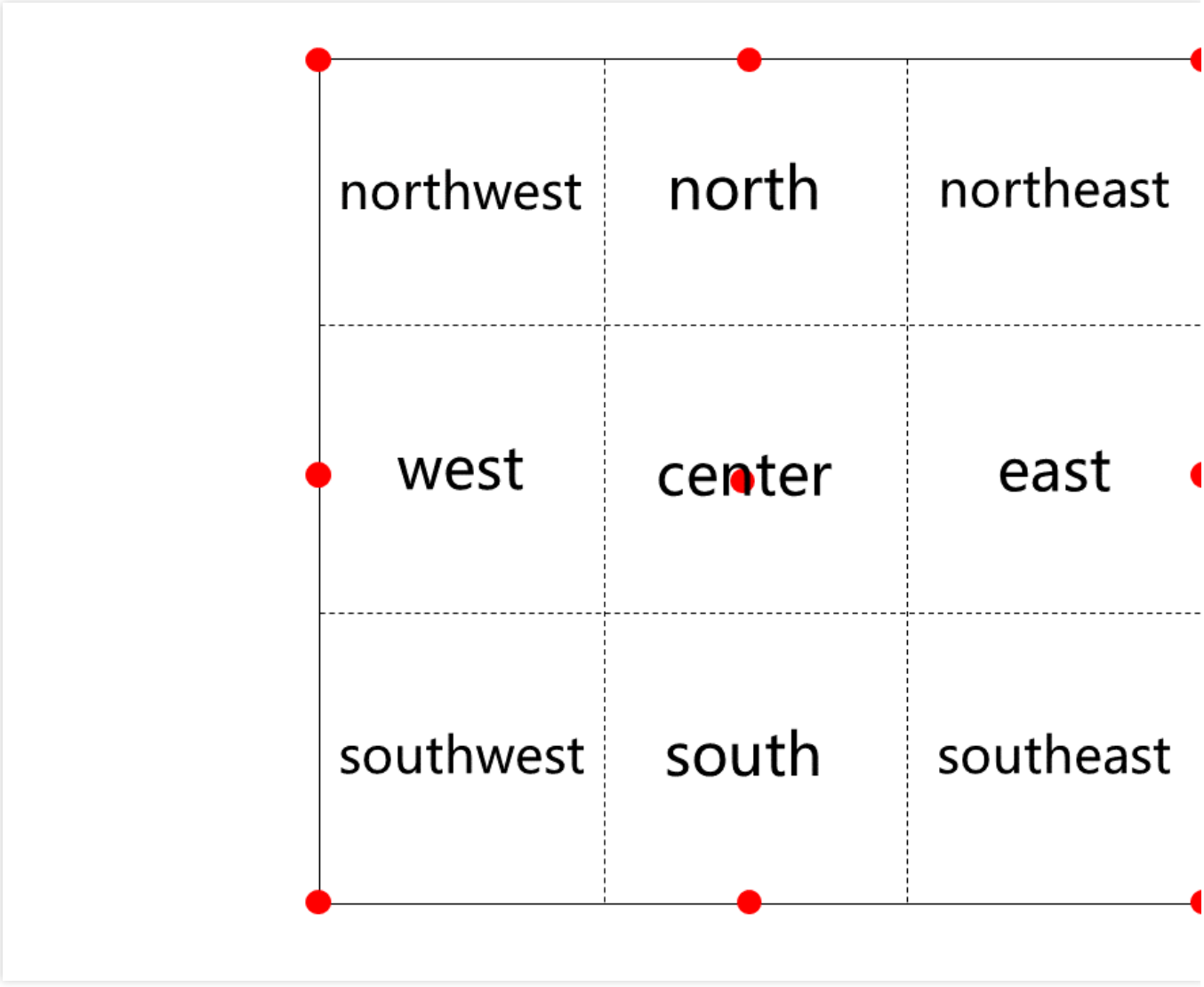
操作名称：watermark，数字2，表示水印类型为文字水印。

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 。水印字体列表请参见 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，并参见 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），详情请参见 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

/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

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圆点为参照）。



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）。

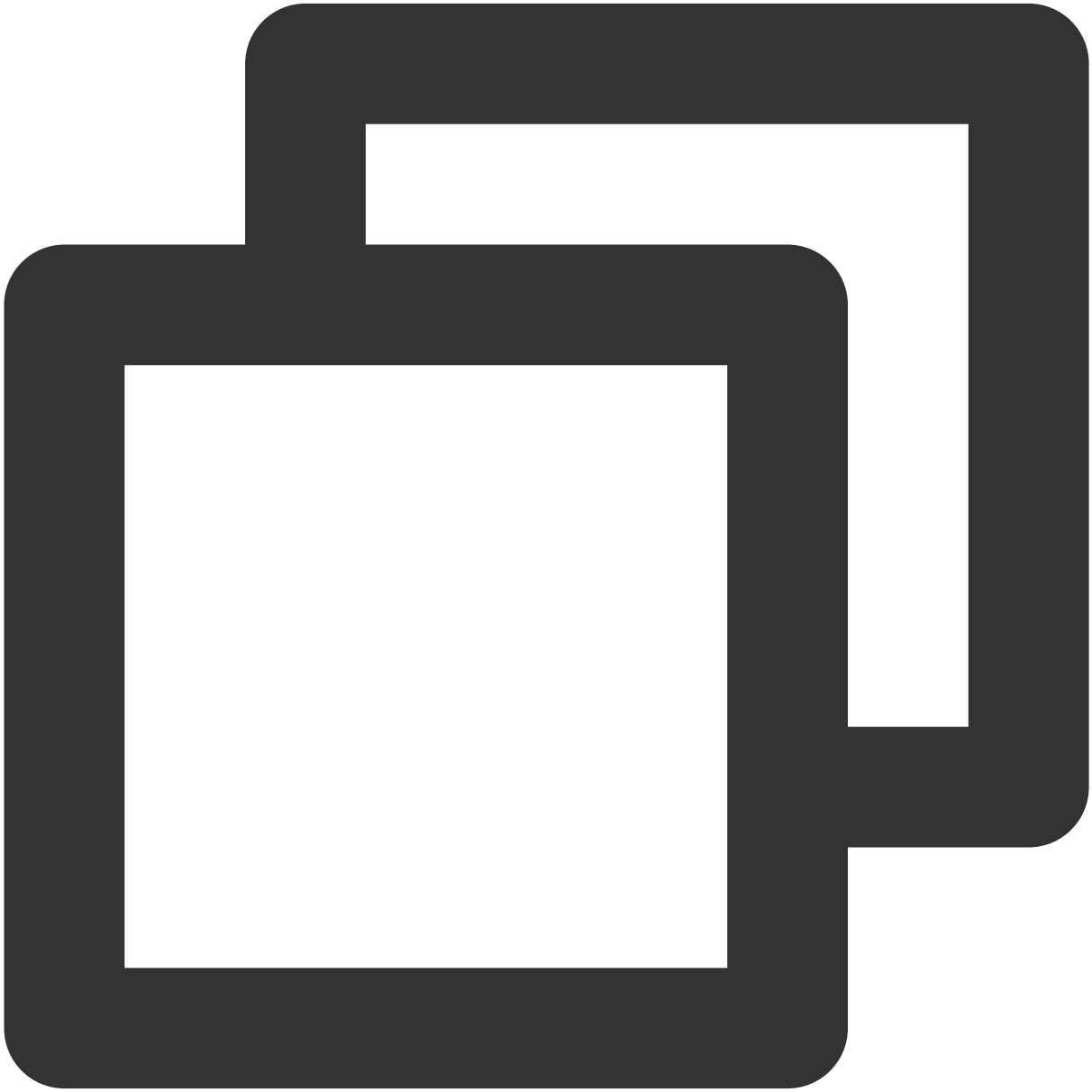
版权所有：腾讯云计算（北京）有限责任公司

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



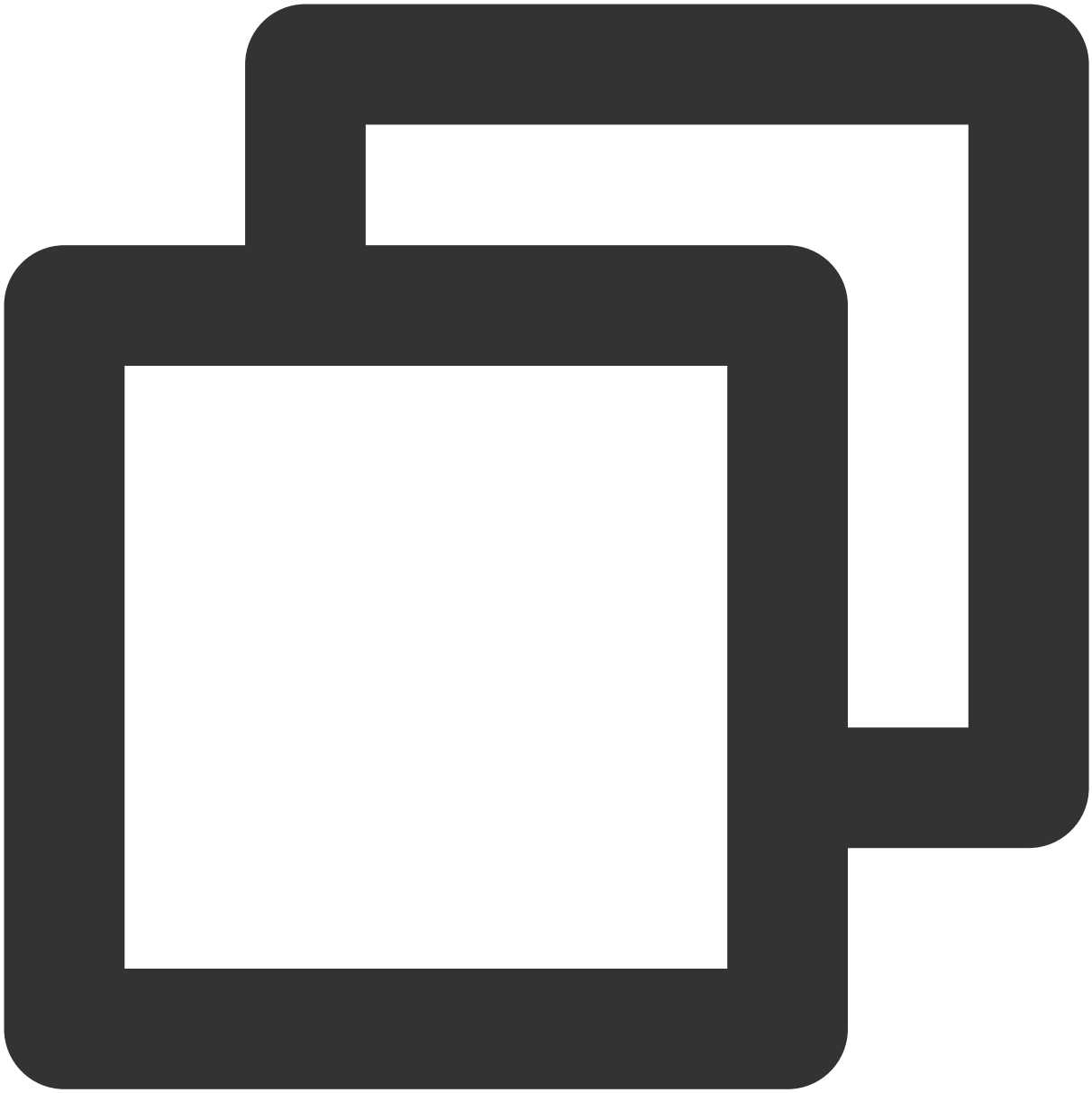
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/2/tex
```

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：



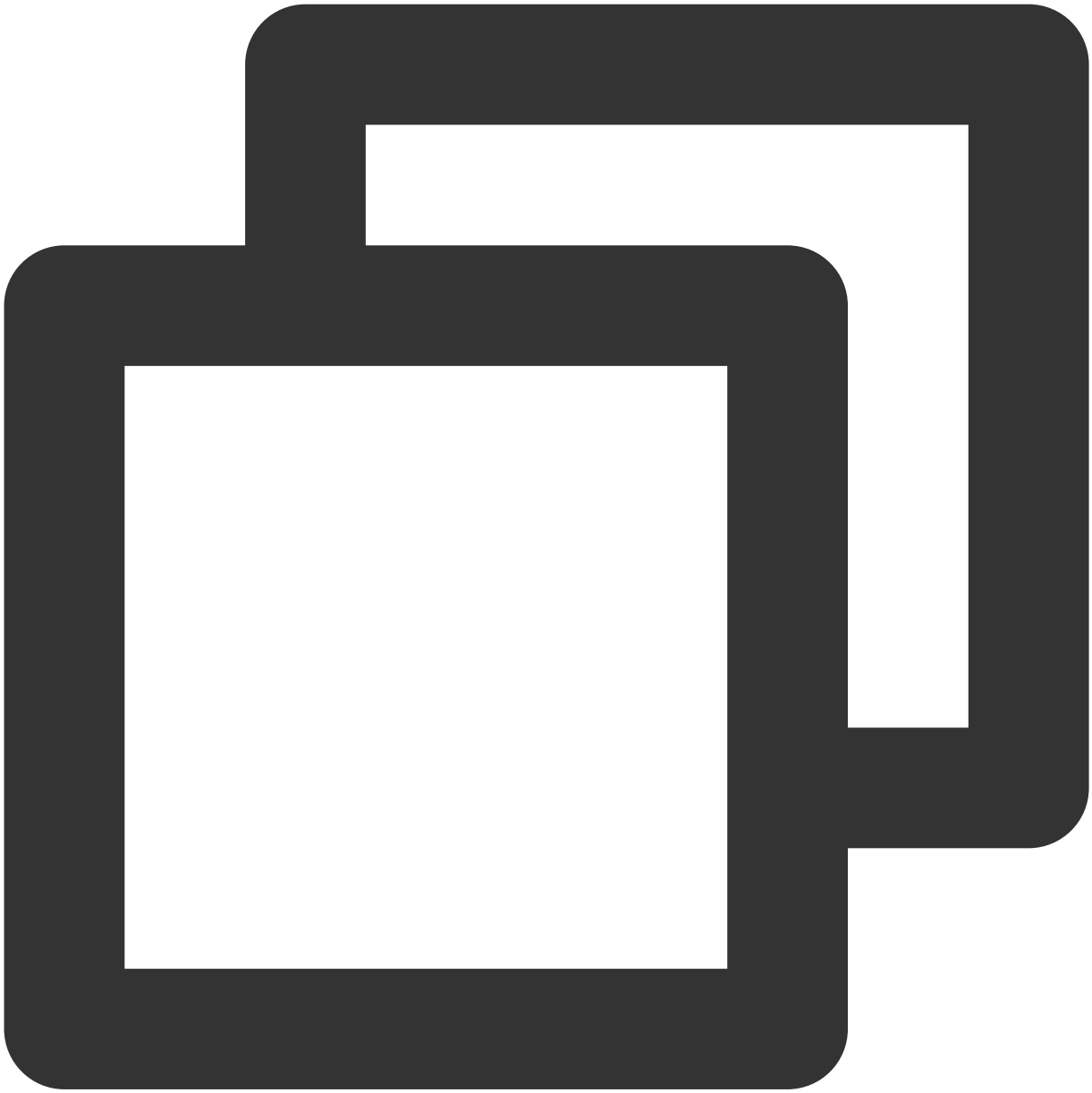
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

说明：

`<signature>` 为签名部分，获取方式请参见 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：2024-01-18 15:11:52

功能概述

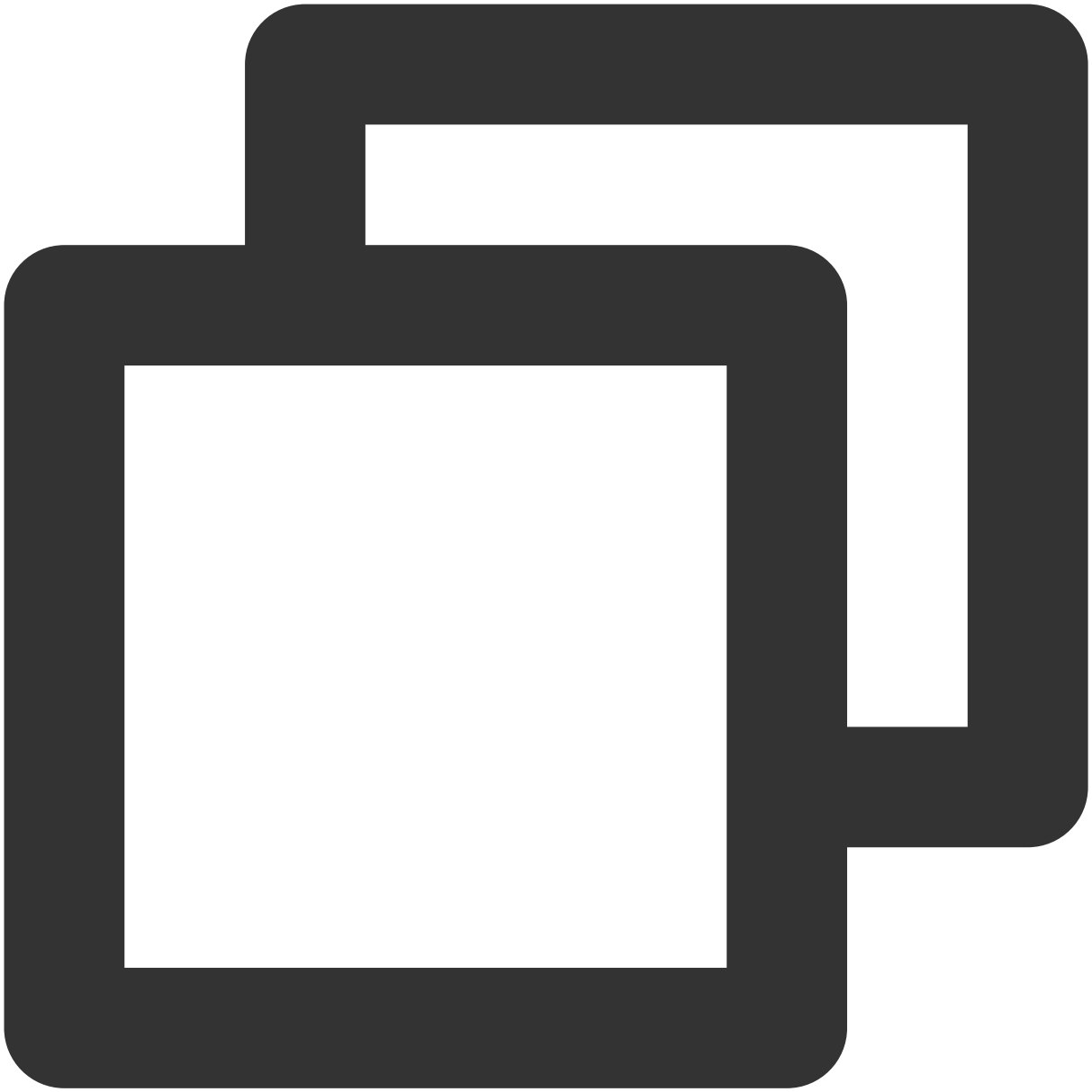
腾讯云数据万象通过 **imageInfo** 接口查询图片基本信息，包括格式、长、宽等。

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`

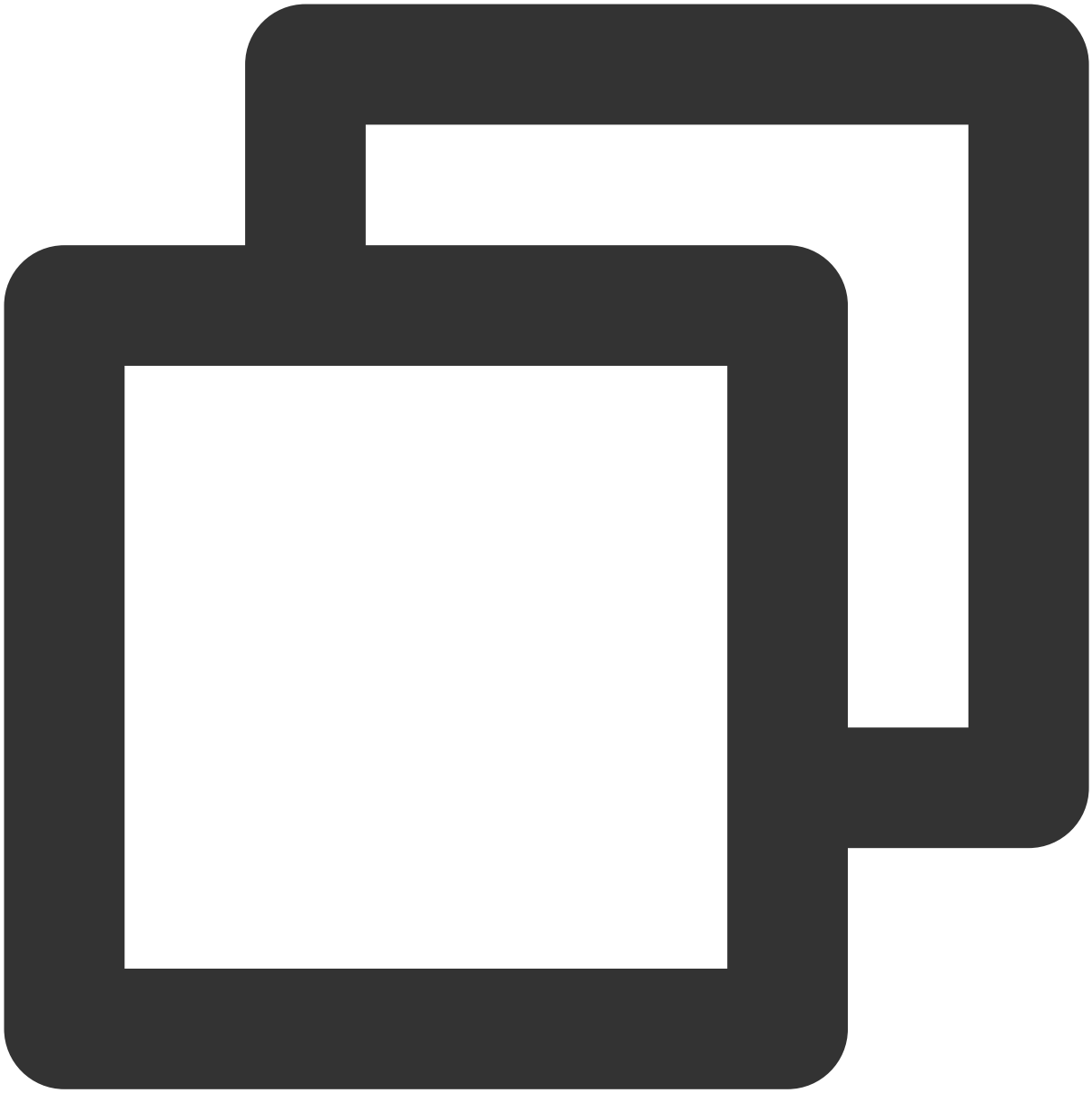
响应参数

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

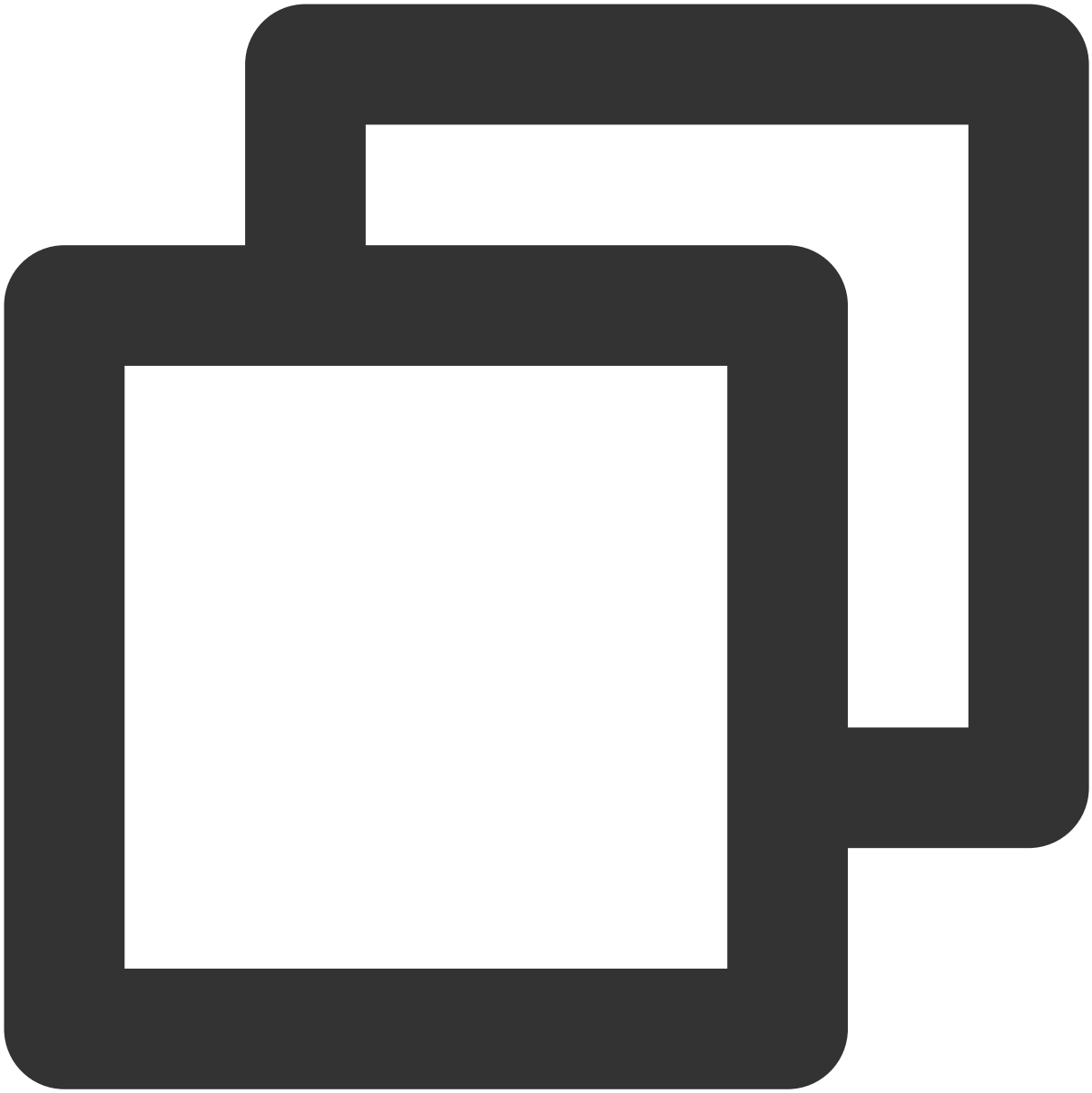
案例一：公有读

请求



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageInfo
```

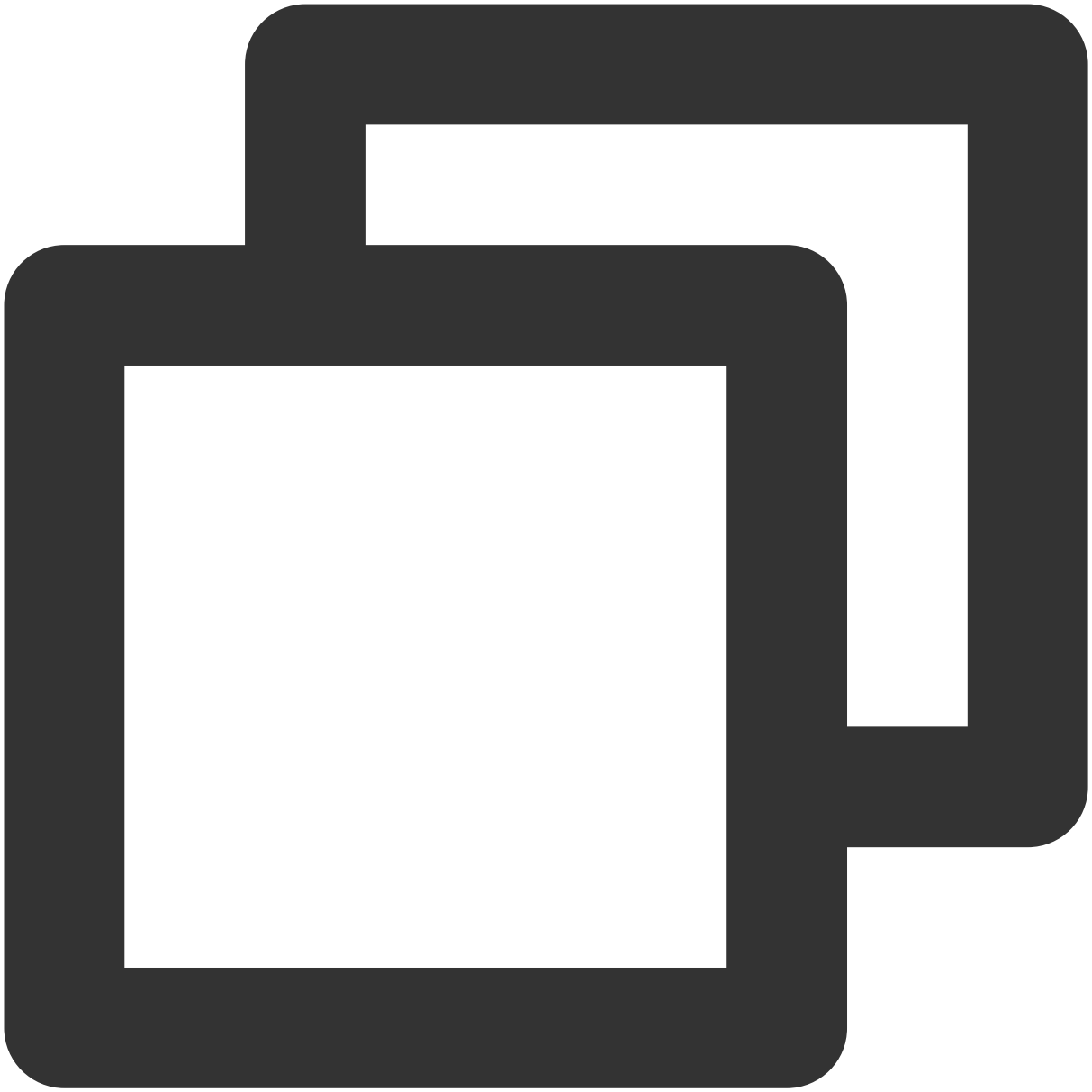
响应



{"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：



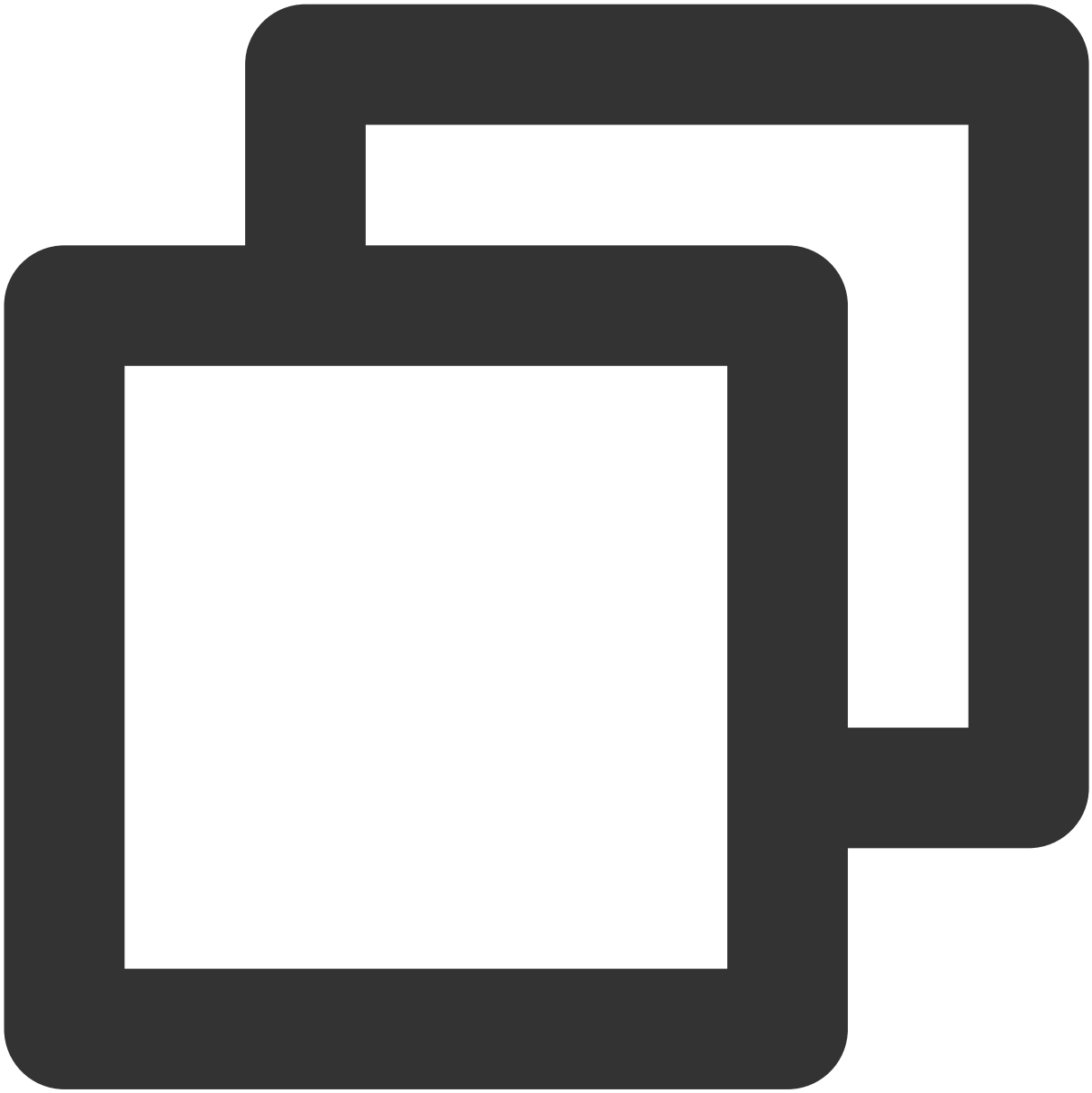
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:52

功能概述

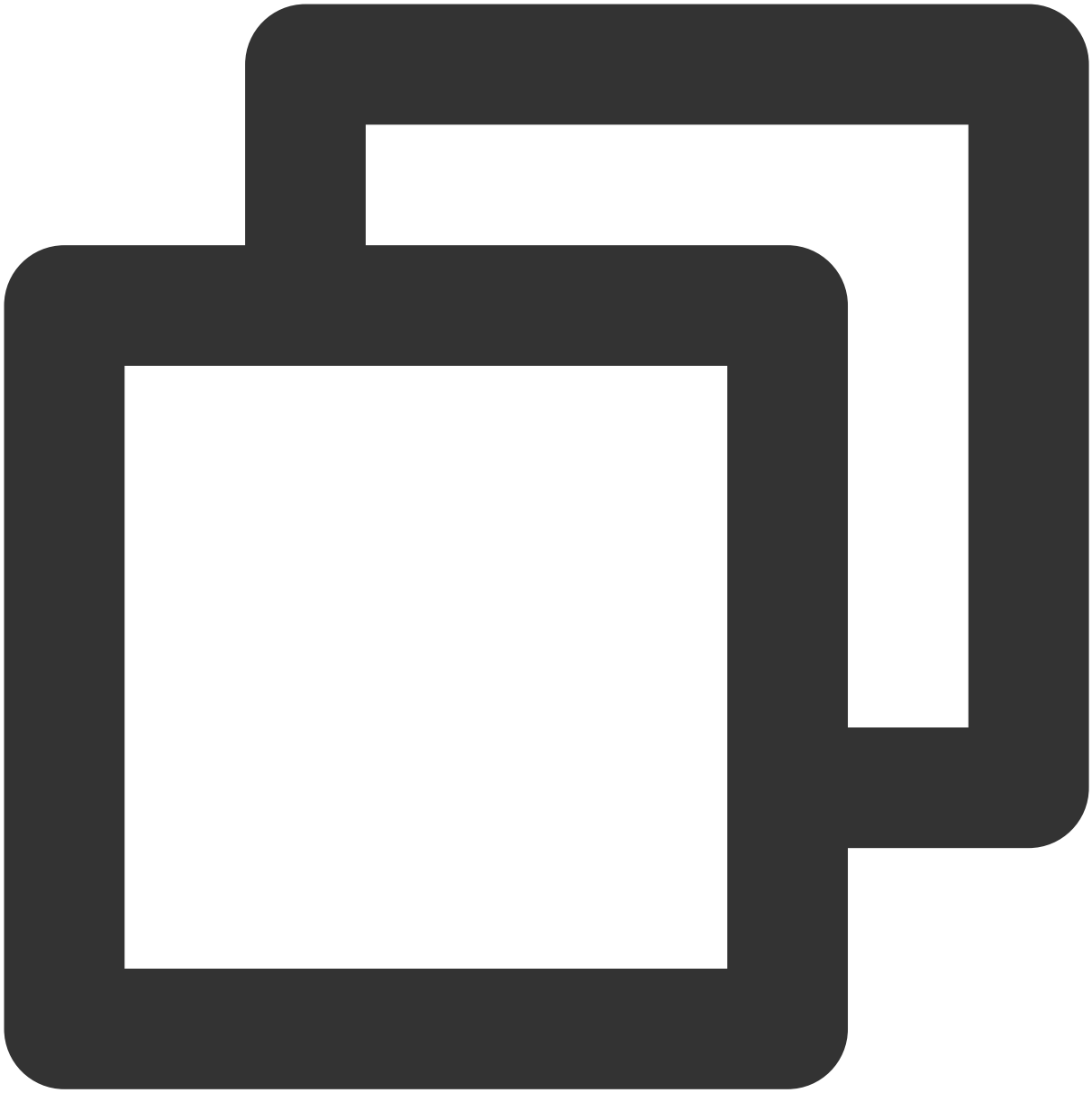
EXIF（Exchangeable Image File）全称为可交换图像文件，可记录数码照片的拍摄参数、缩略图及其他属性信息。腾讯云数据万象通过 **exif** 接口获取 EXIF 信息。处理图片原图大小不超过32MB、宽高不超过30000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素；针对动图，原图宽 x 高 x 帧数不超过2.5亿像素。

注意：

如图片无 exif 信息，将返回 `{"error" : "no exif data"}` 。

接口示例

版权所有：腾讯云计算（北京）有限责任公司



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

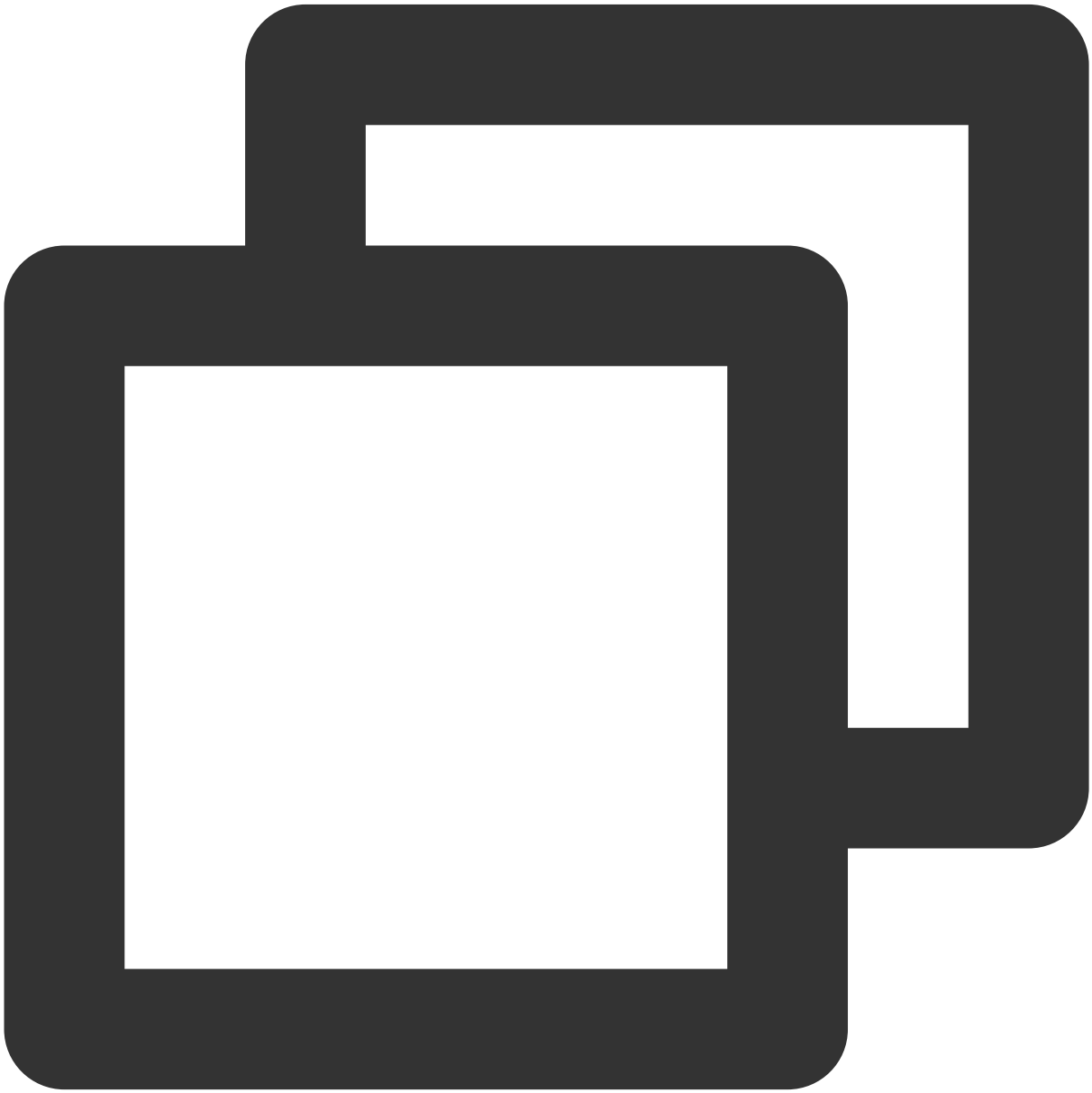
处理参数说明

操作名称：exif。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。

实际案例

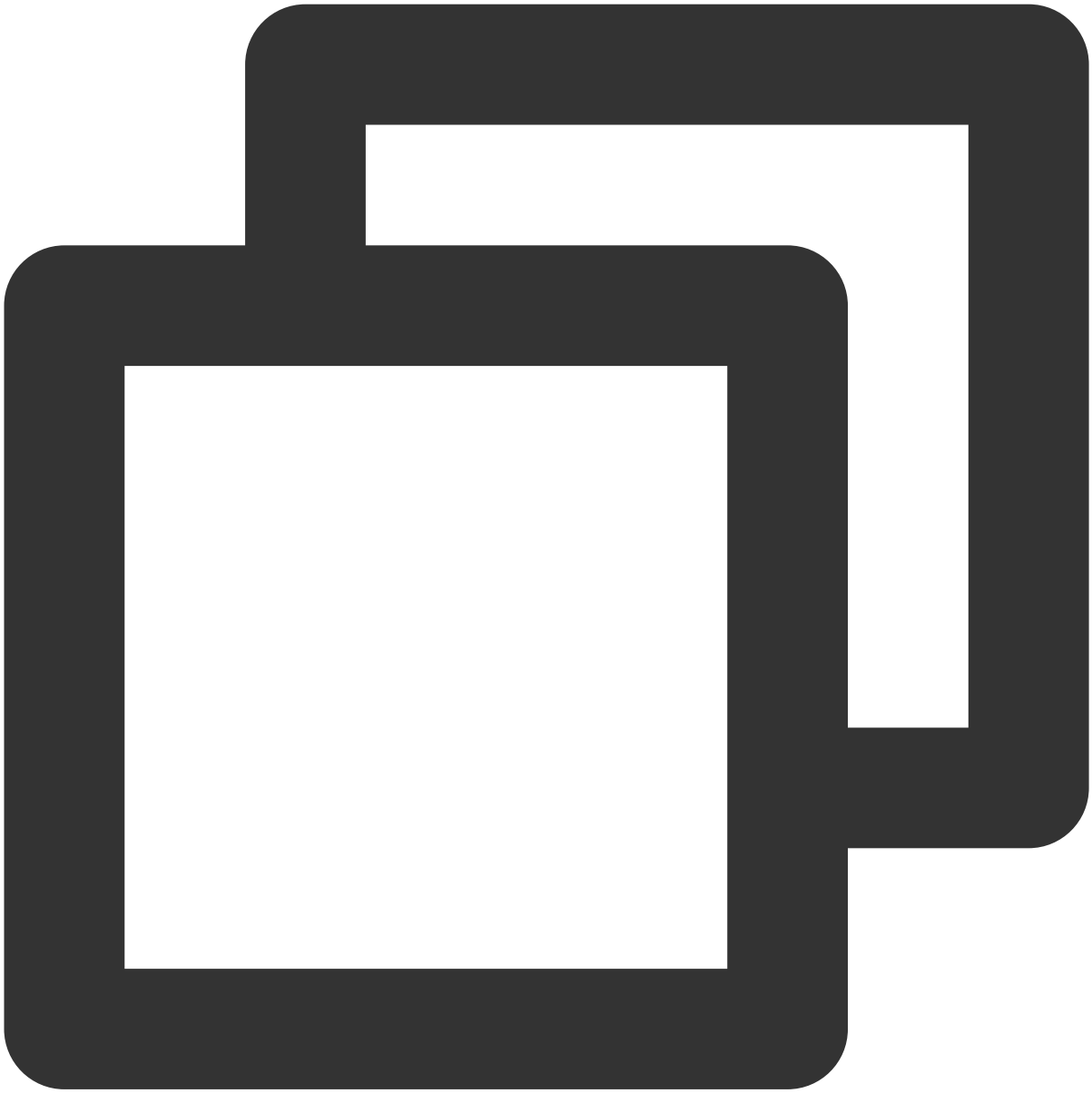
案例一：公有读



```
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：



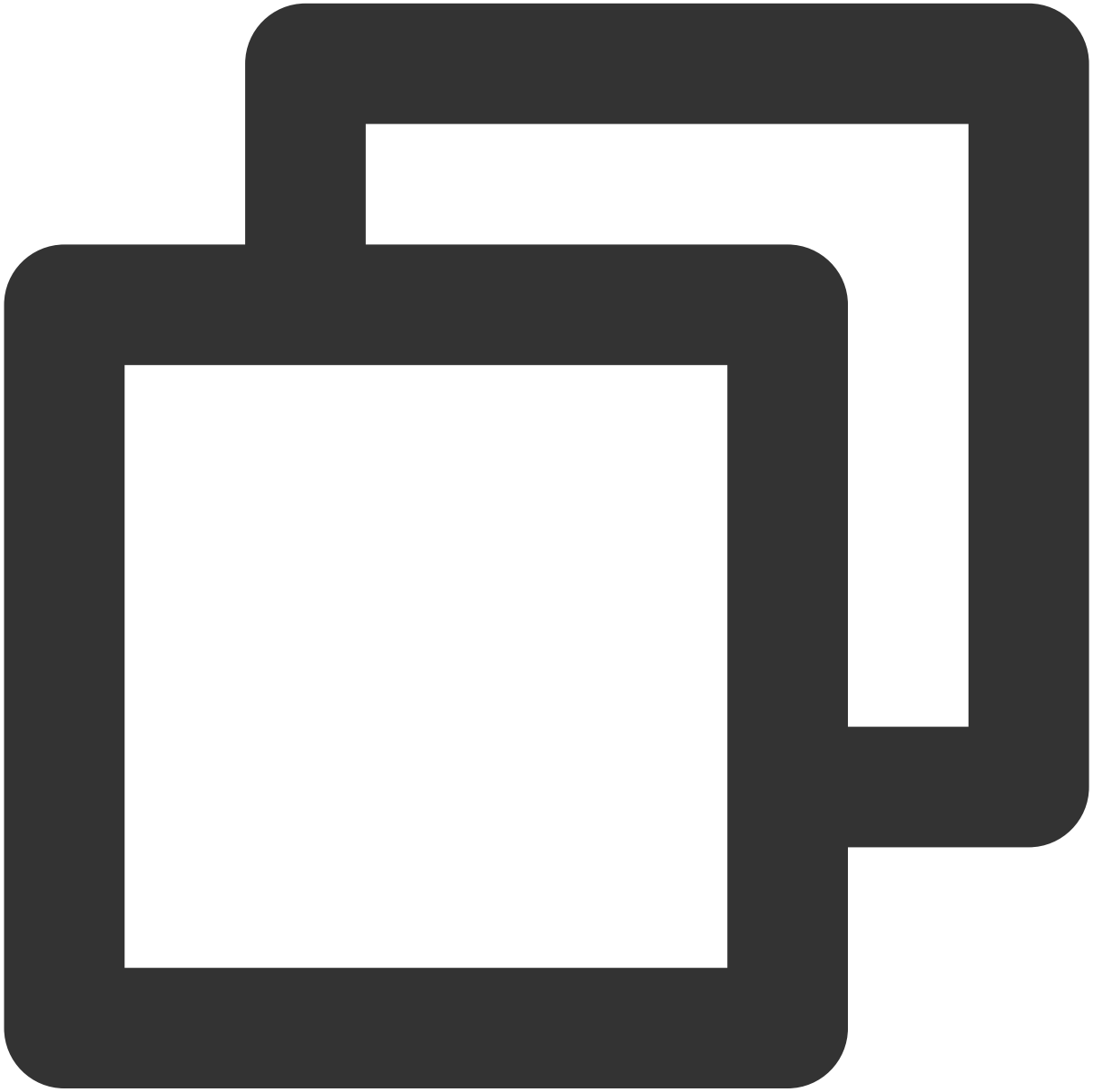
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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
```

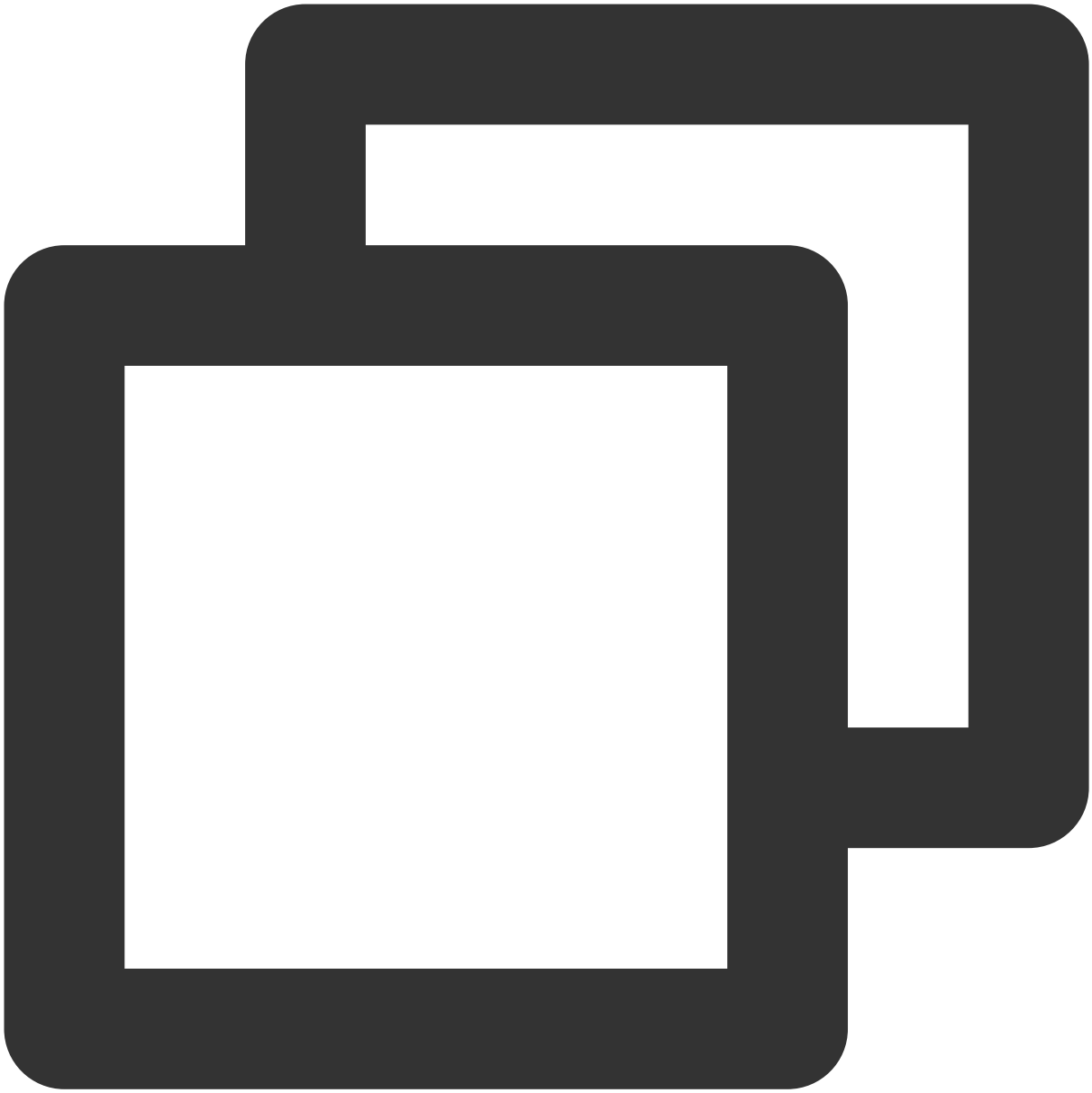
获取图片主色调

最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象通过 **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。

接口示例



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

参数说明

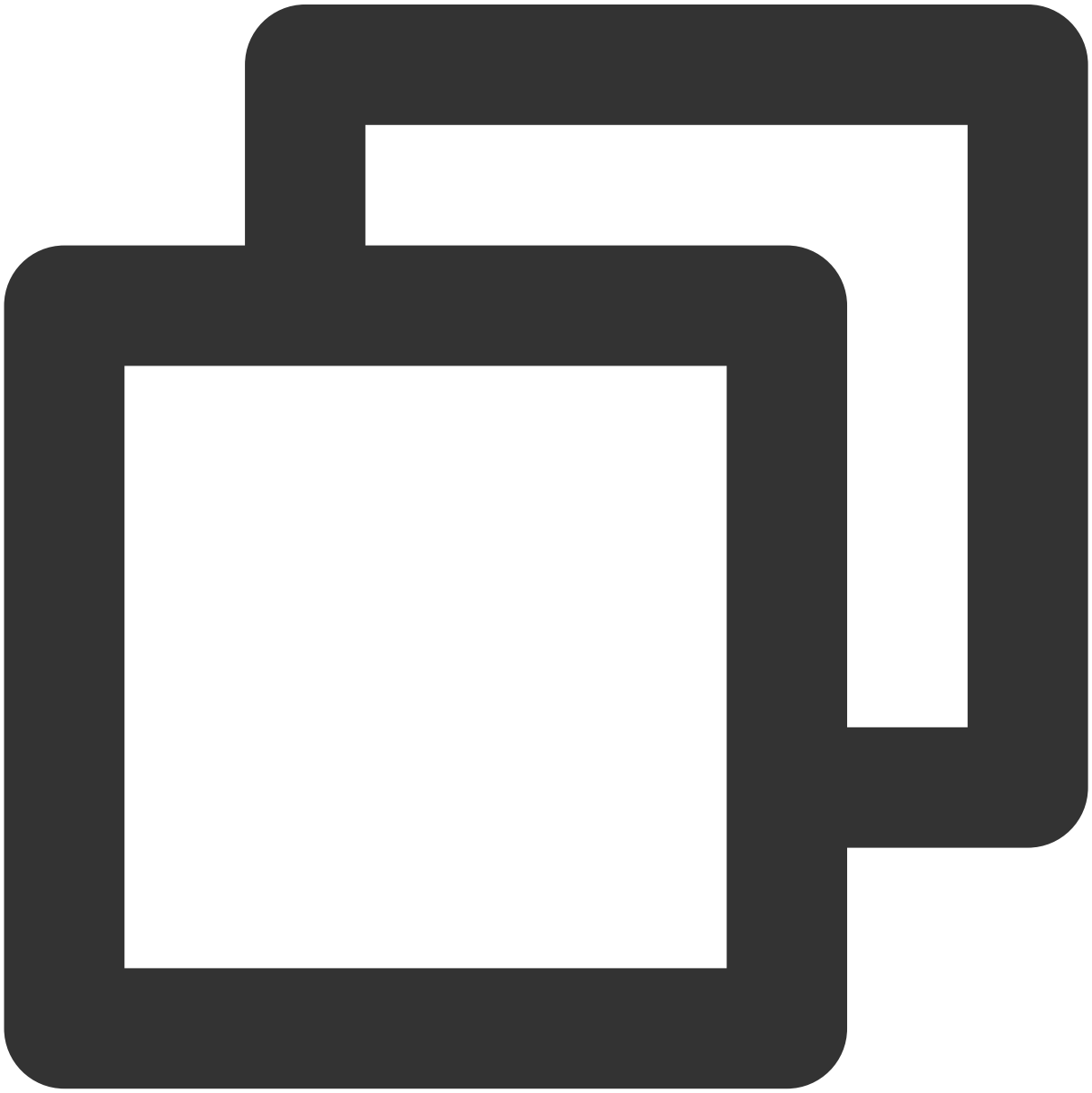
操作名称：imageAve

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。

实际案例

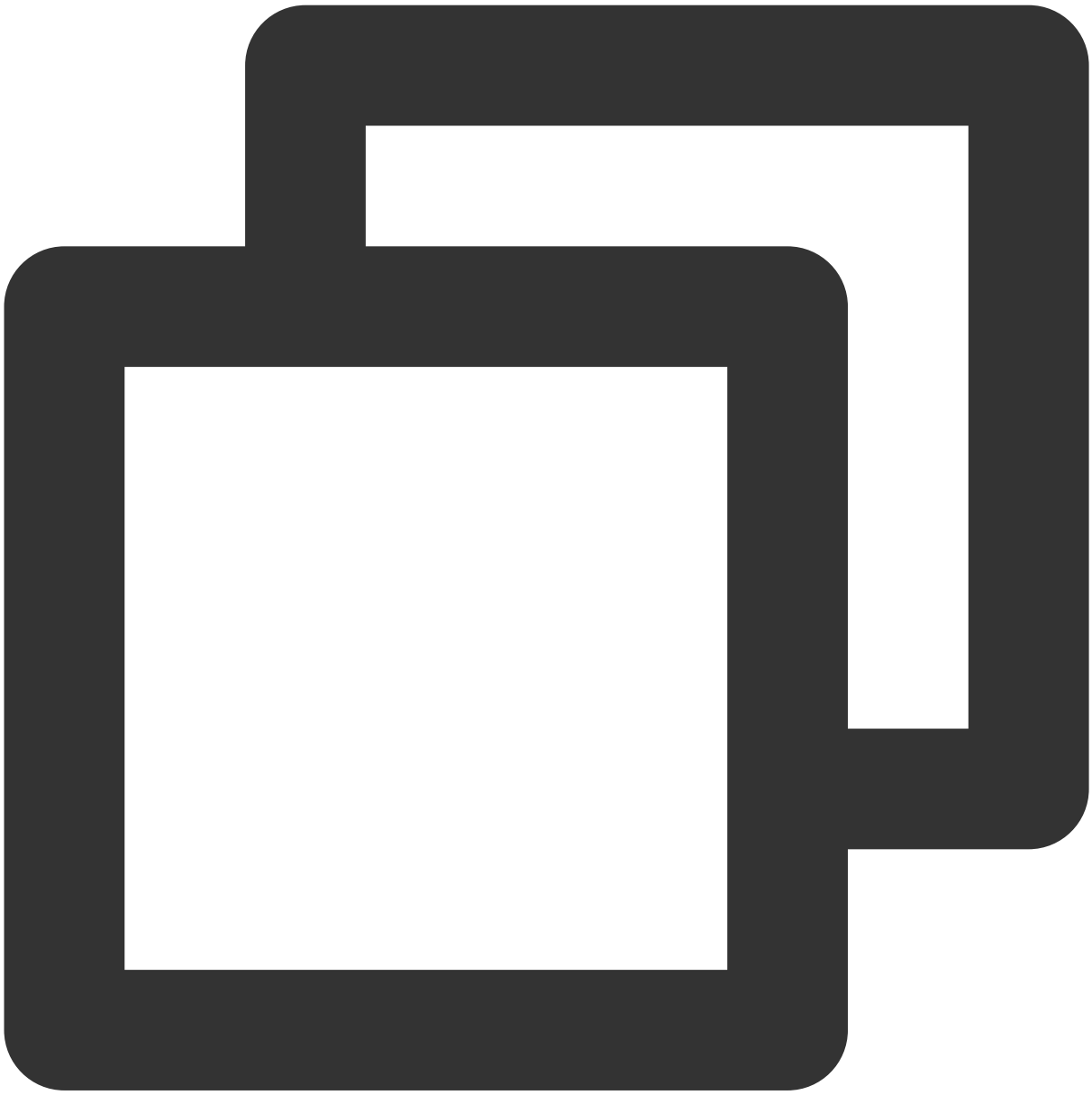
案例一：公有读

请求



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageAve
```

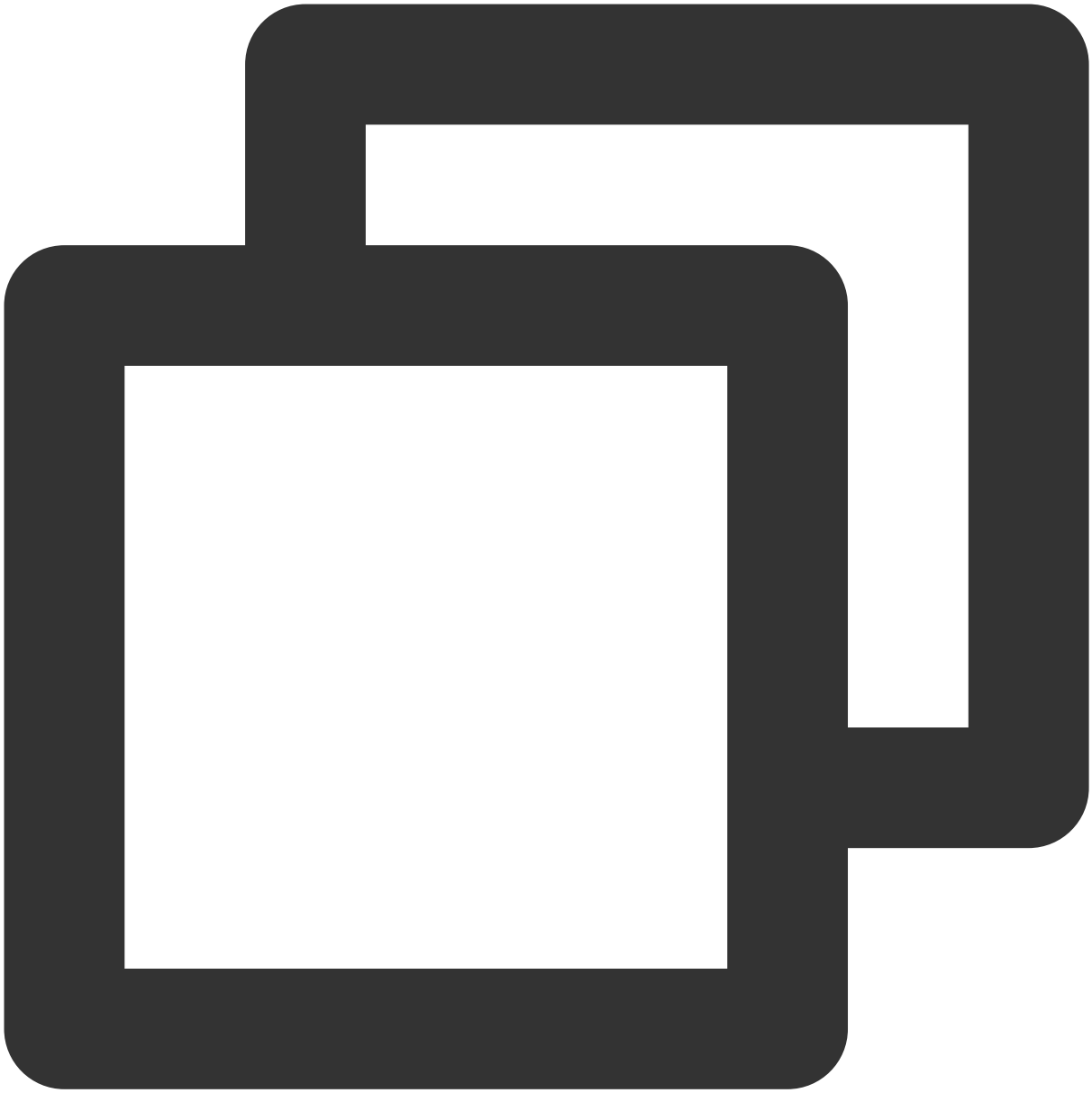
响应



```
{"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：



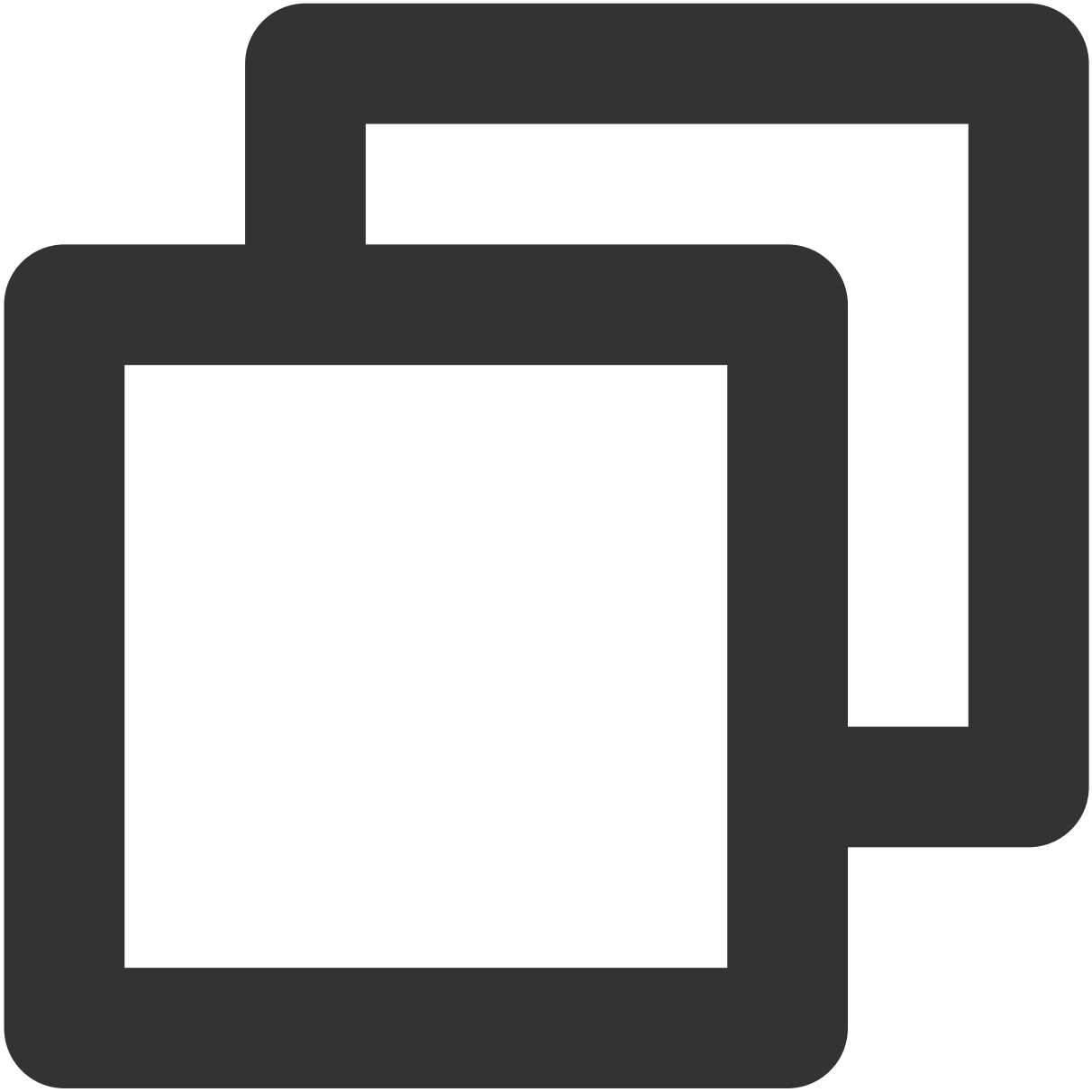
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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：2024-01-18 15:11:52

功能概述

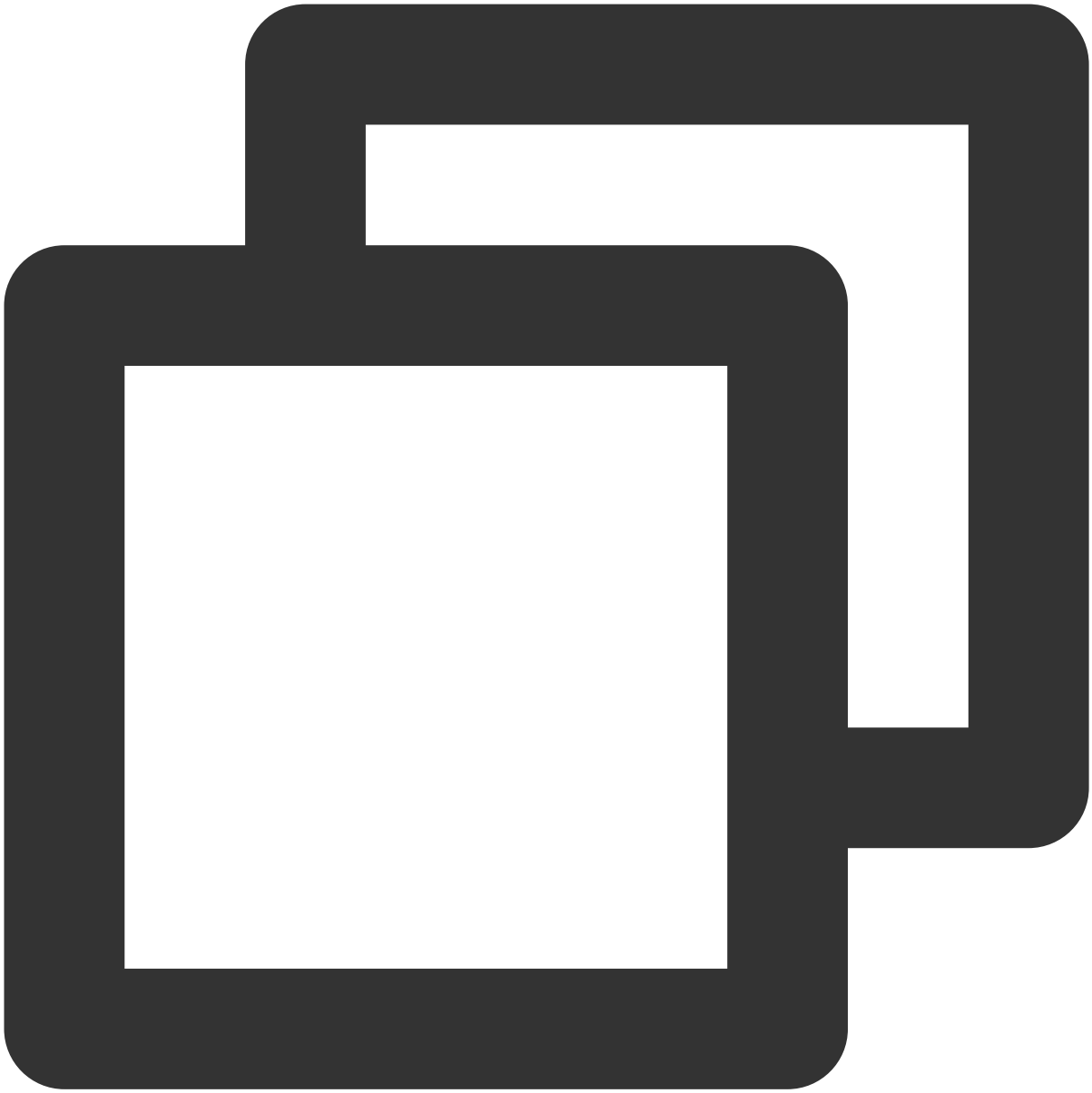
腾讯云数据万象通过 **imageMogr2** 接口可去除图片元信息，包括 exif 信息。
该功能支持以下处理方式：
下载时处理
上传时处理
云上数据处理

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。

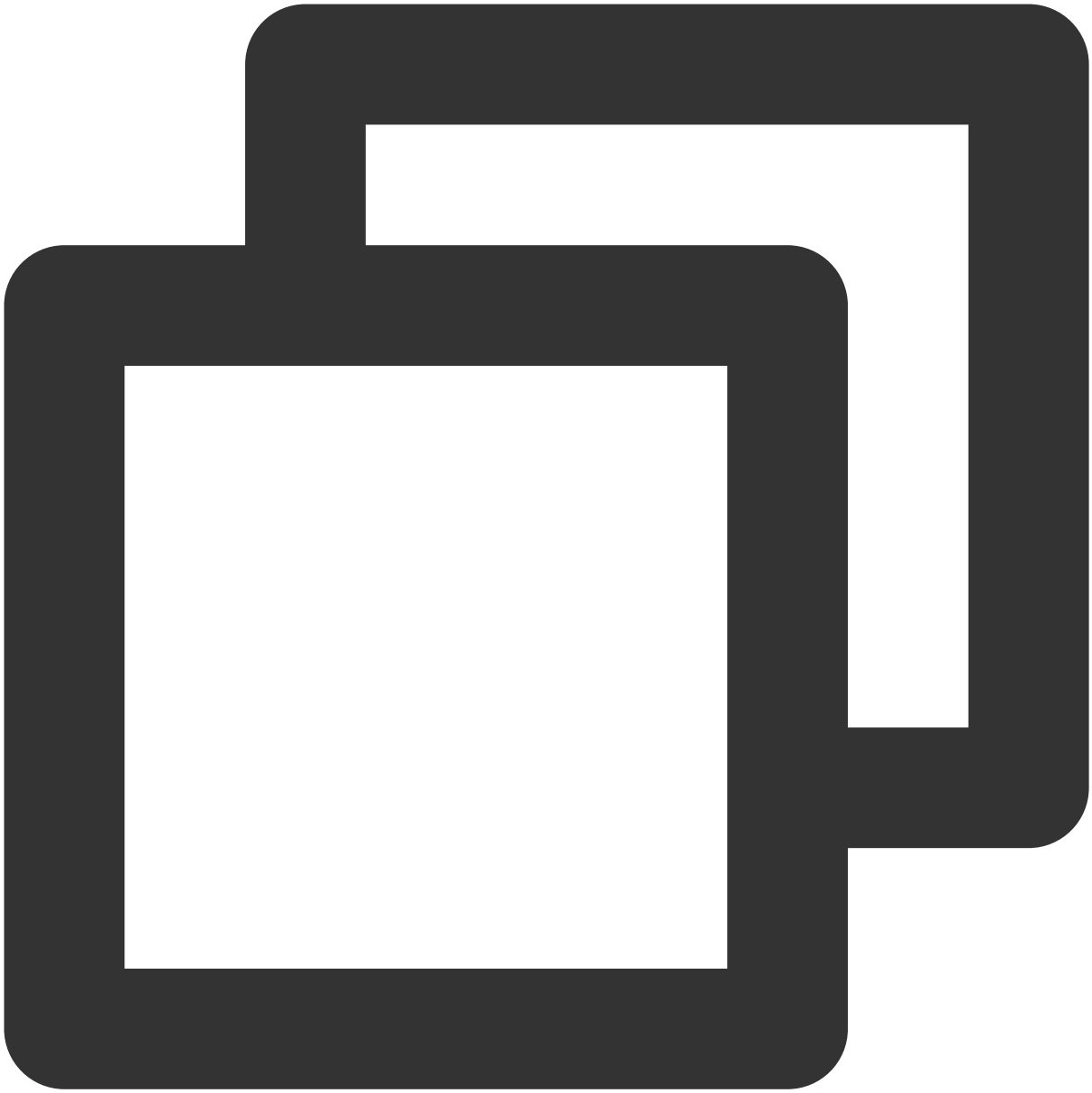
接口示例

1. 下载时处理



```
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



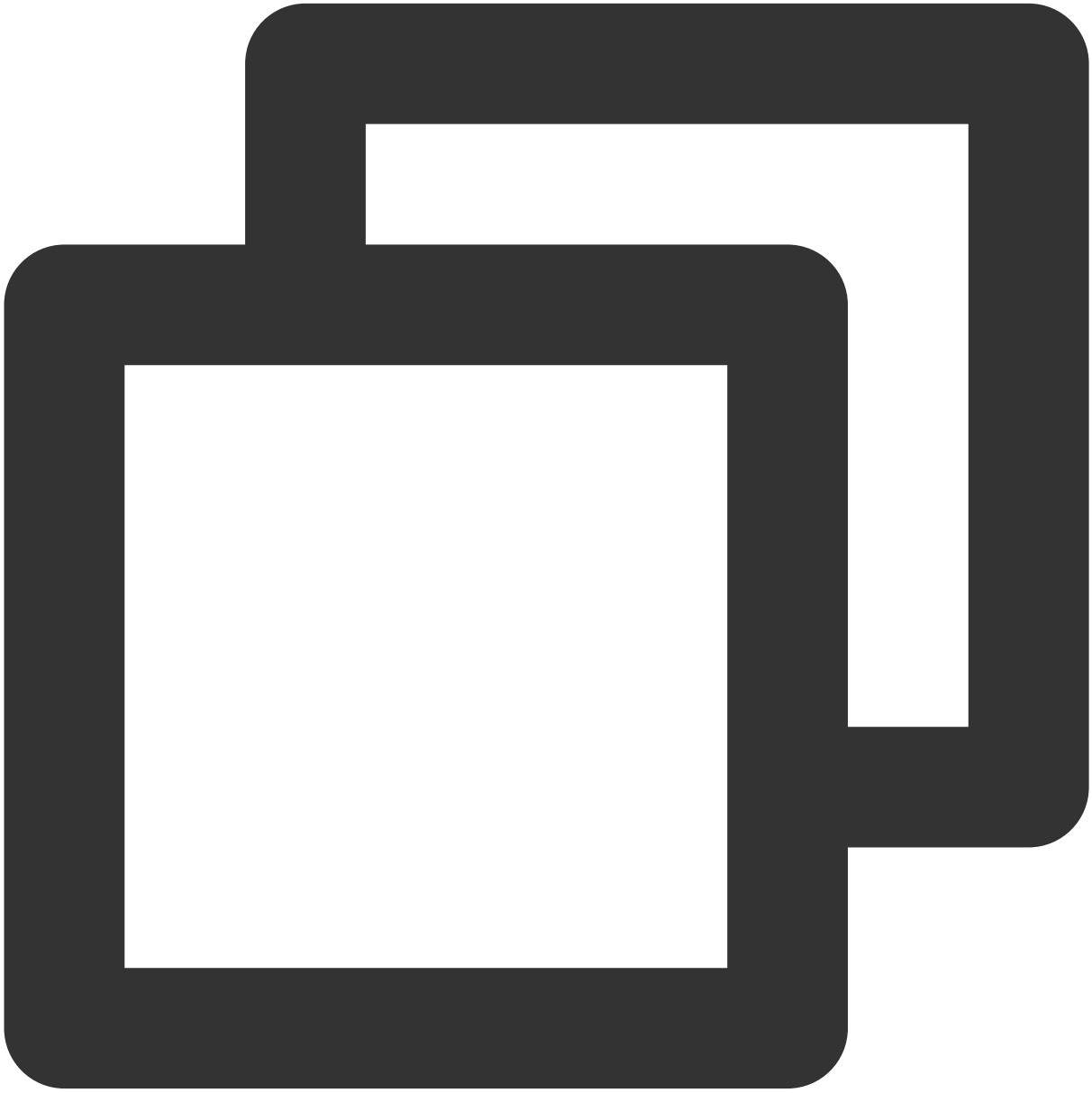
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
      "fileid": "exampleobject",
      "rule": "imageMogr2/strip"
  }]
```

```
}
```

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。

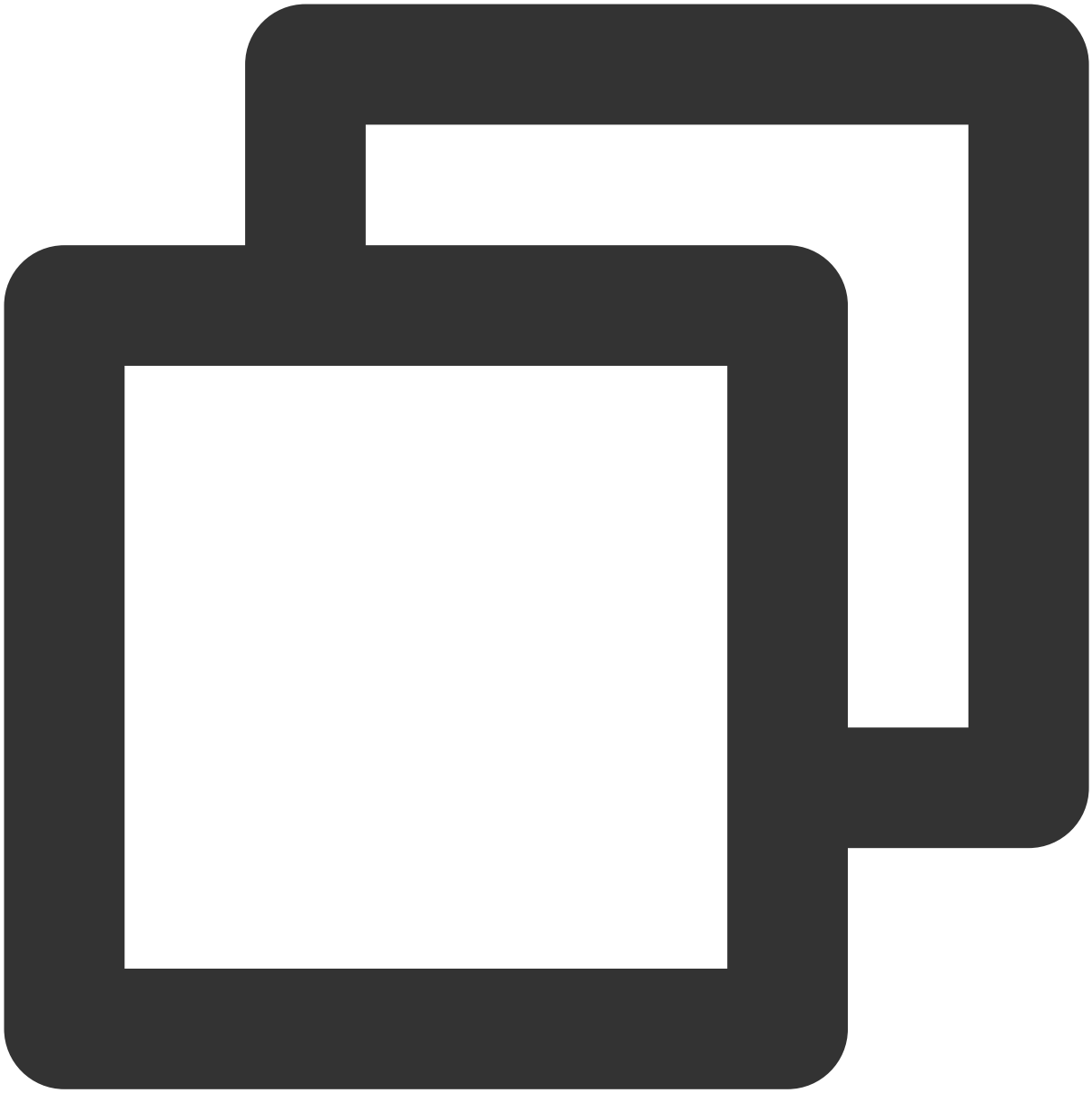
参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
/ignore-error/1	当处理参数中携带此参数时，针对文件过大、参数超限等导致处理失败的场景，会直接返回原图而不报错

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**。

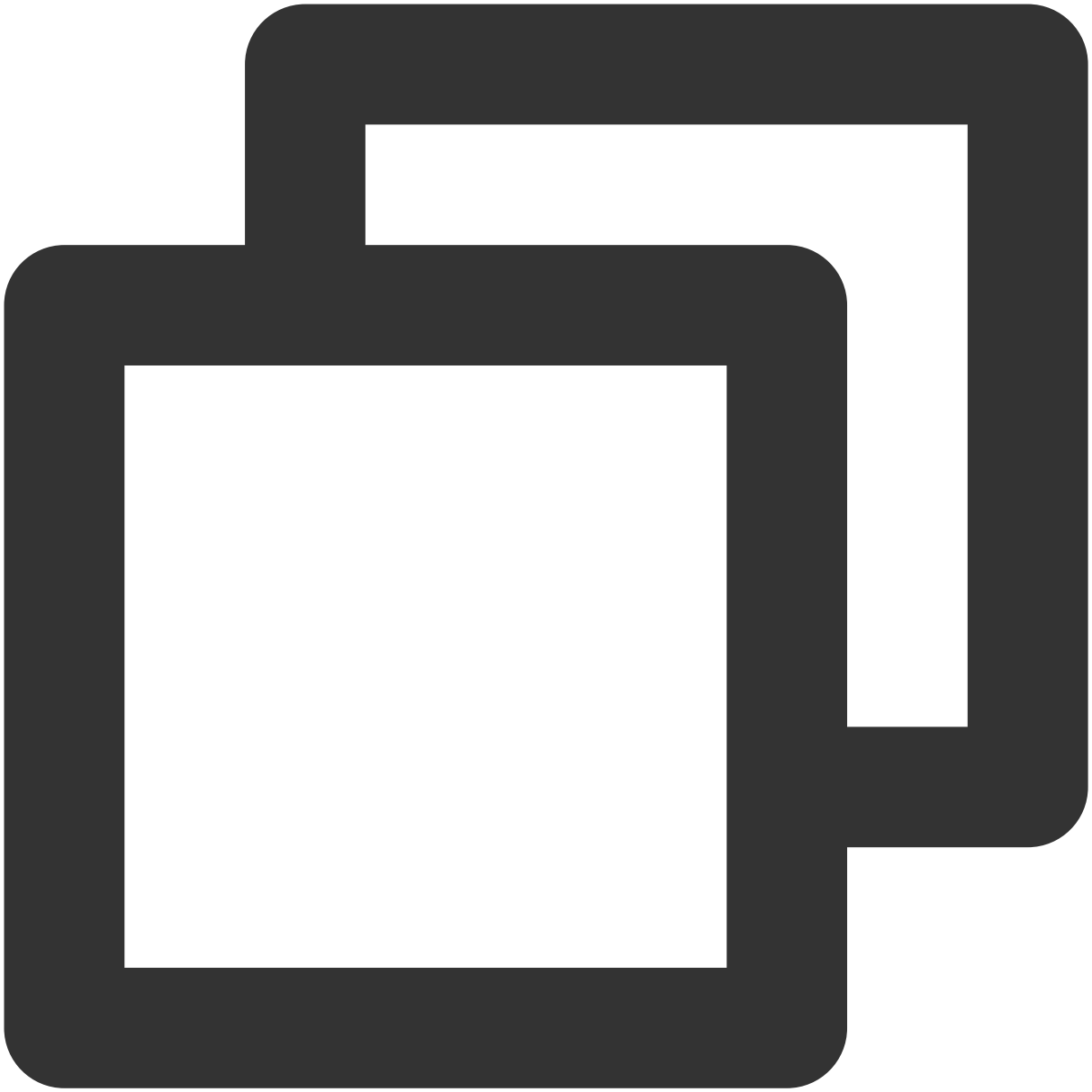
案例一：公有读



```
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：



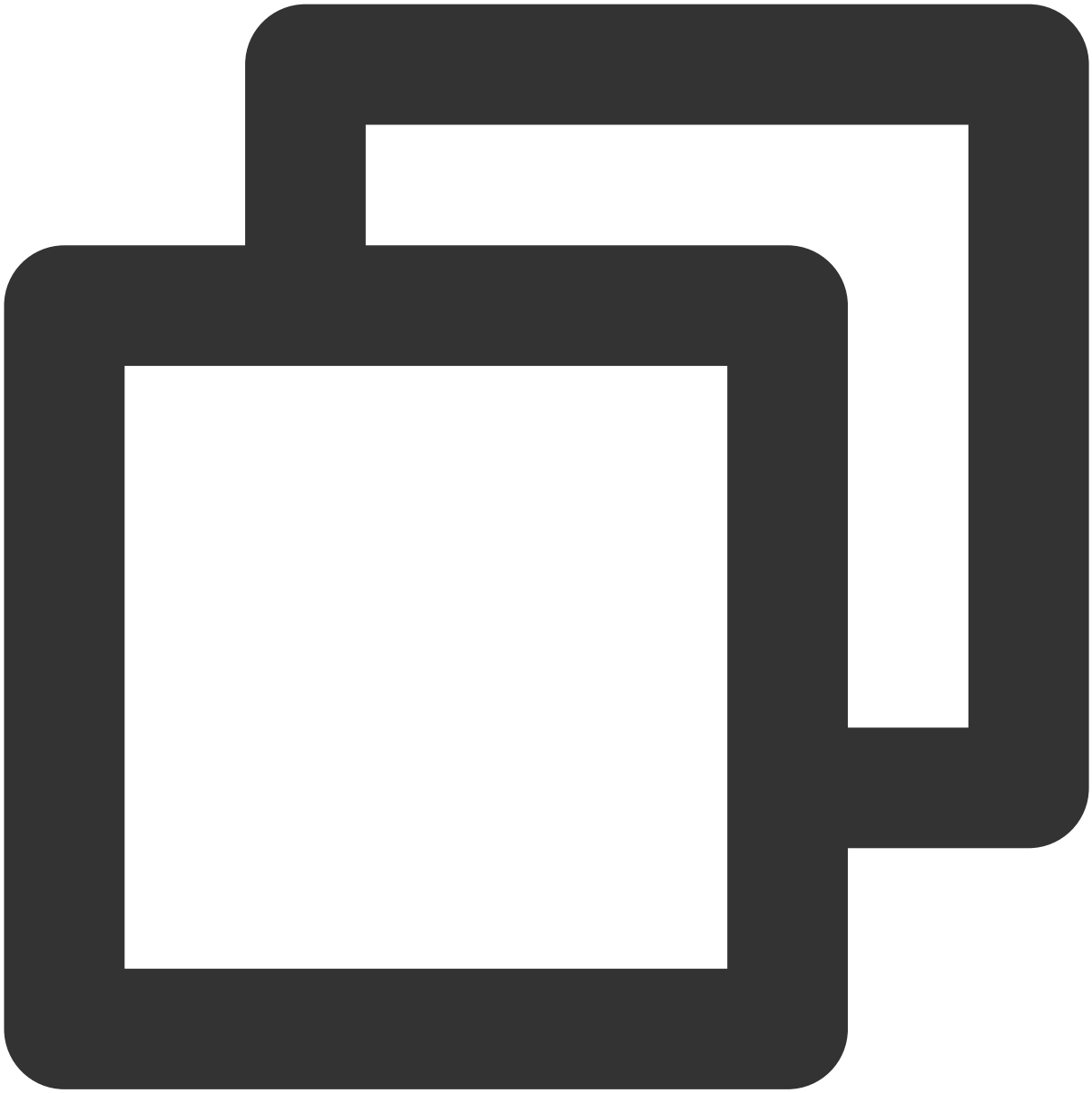
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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

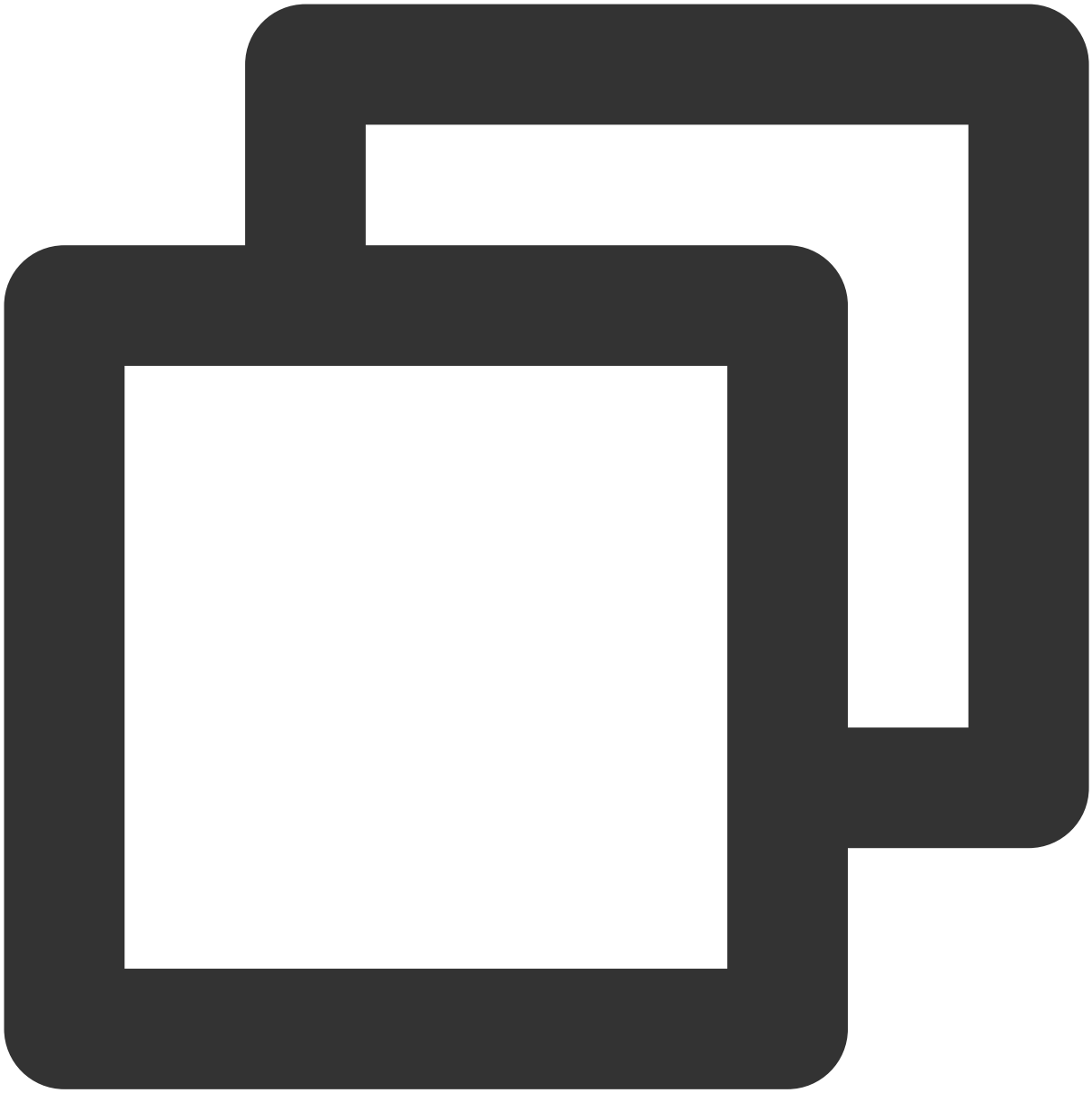
最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象通过 **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。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 gif 帧数限300帧。
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

接口示例

1. 下载时处理

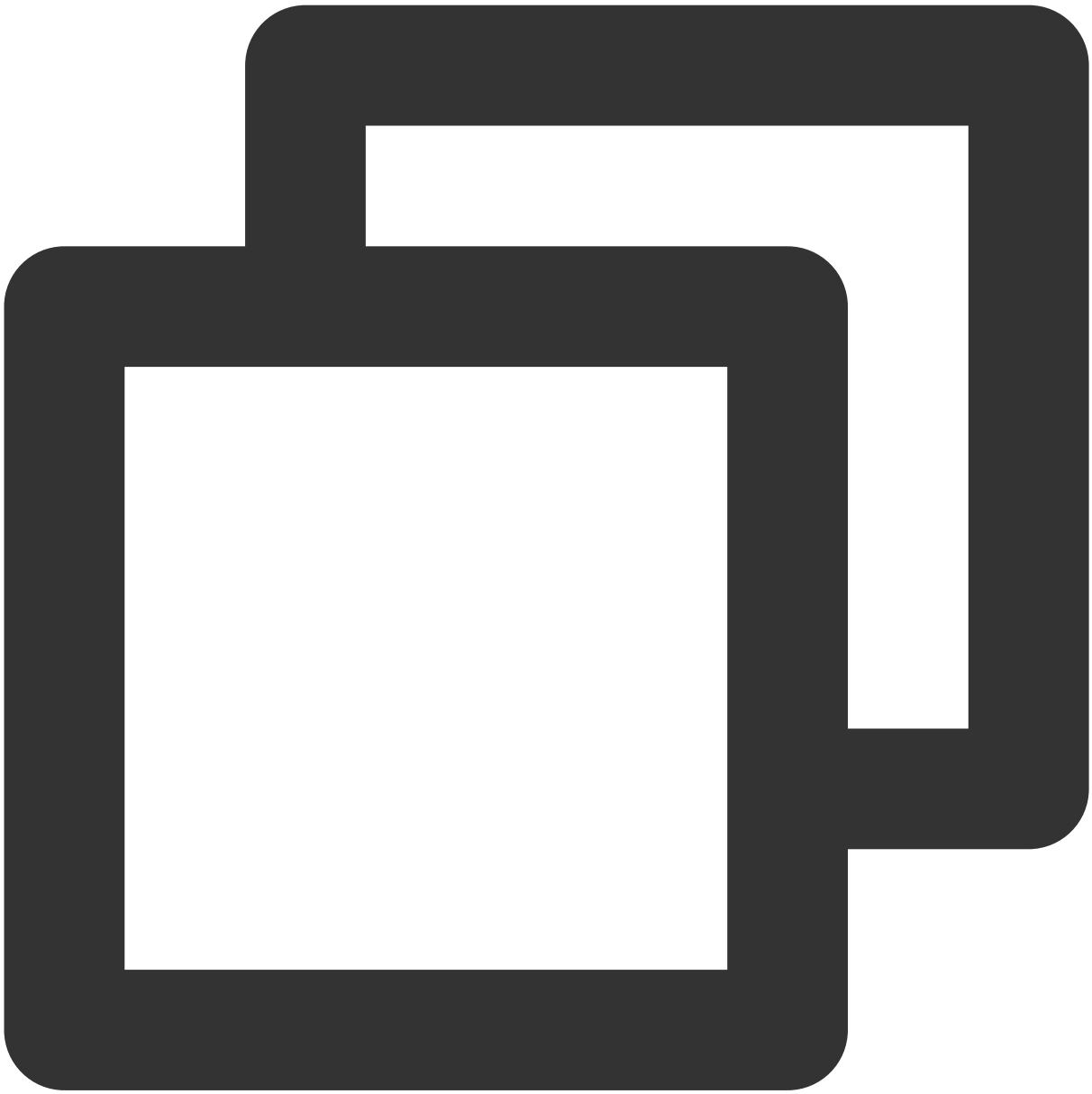


```
GET /<ObjectKey>?imageView2/<mode>/w/<Width>
                                  /h/<Height>
                                  /format/<Format>
                                  /q/<Quality>
                                  /rq/<Quality>
                                  /lq/<Quality> HTTP/1.1
Host: <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



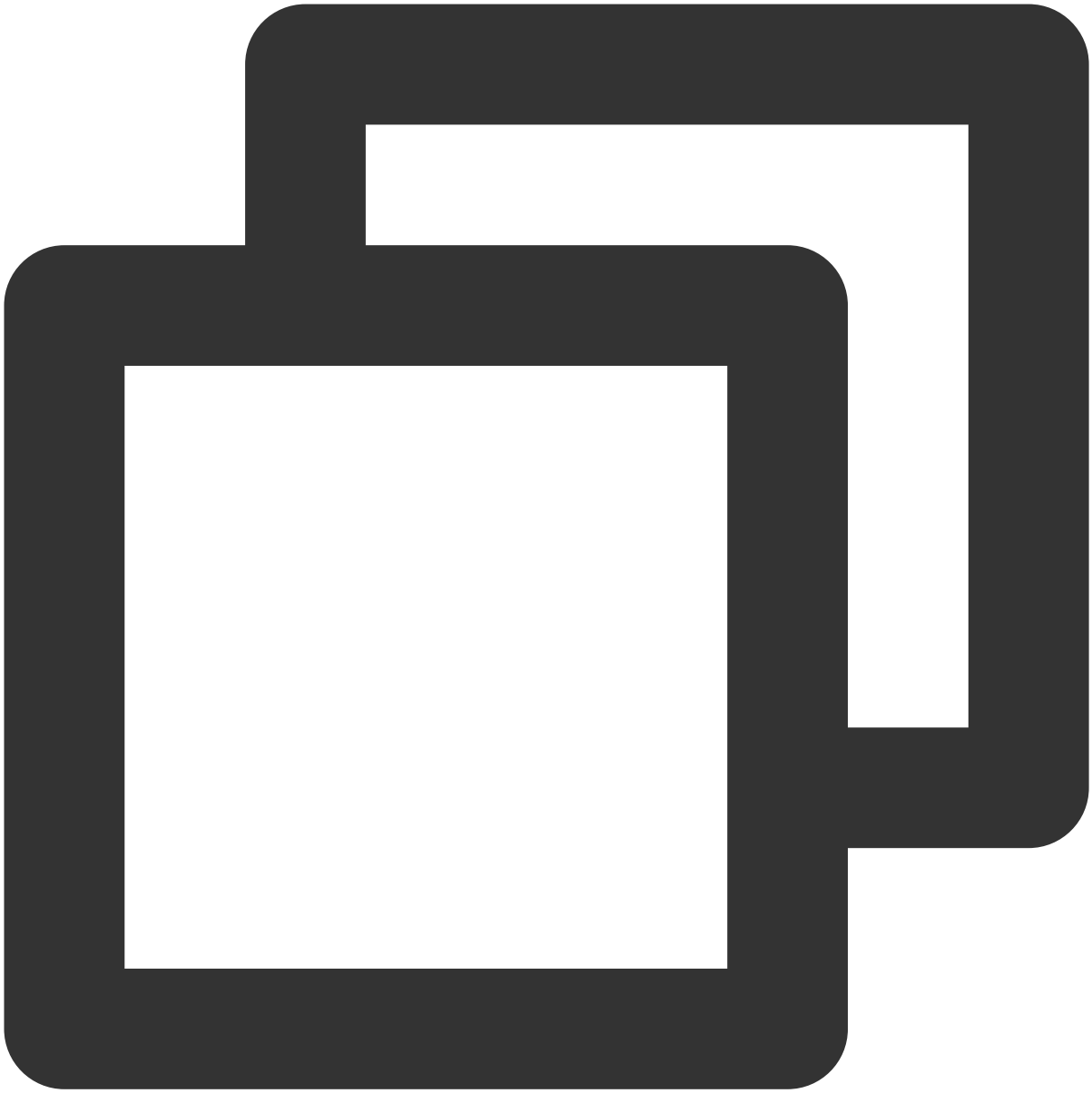
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
```

```
    "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>
```

```
                    /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

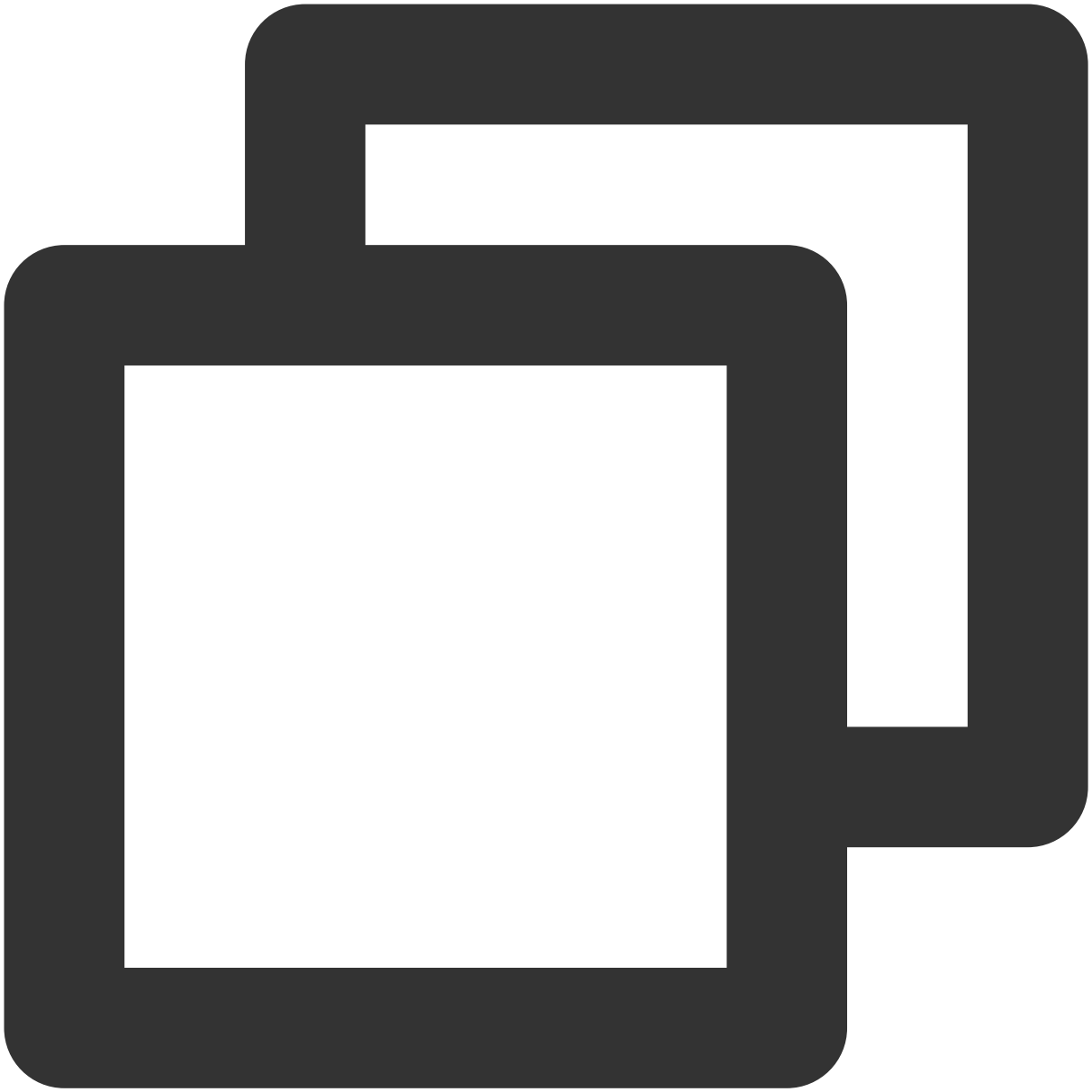
实际案例

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：



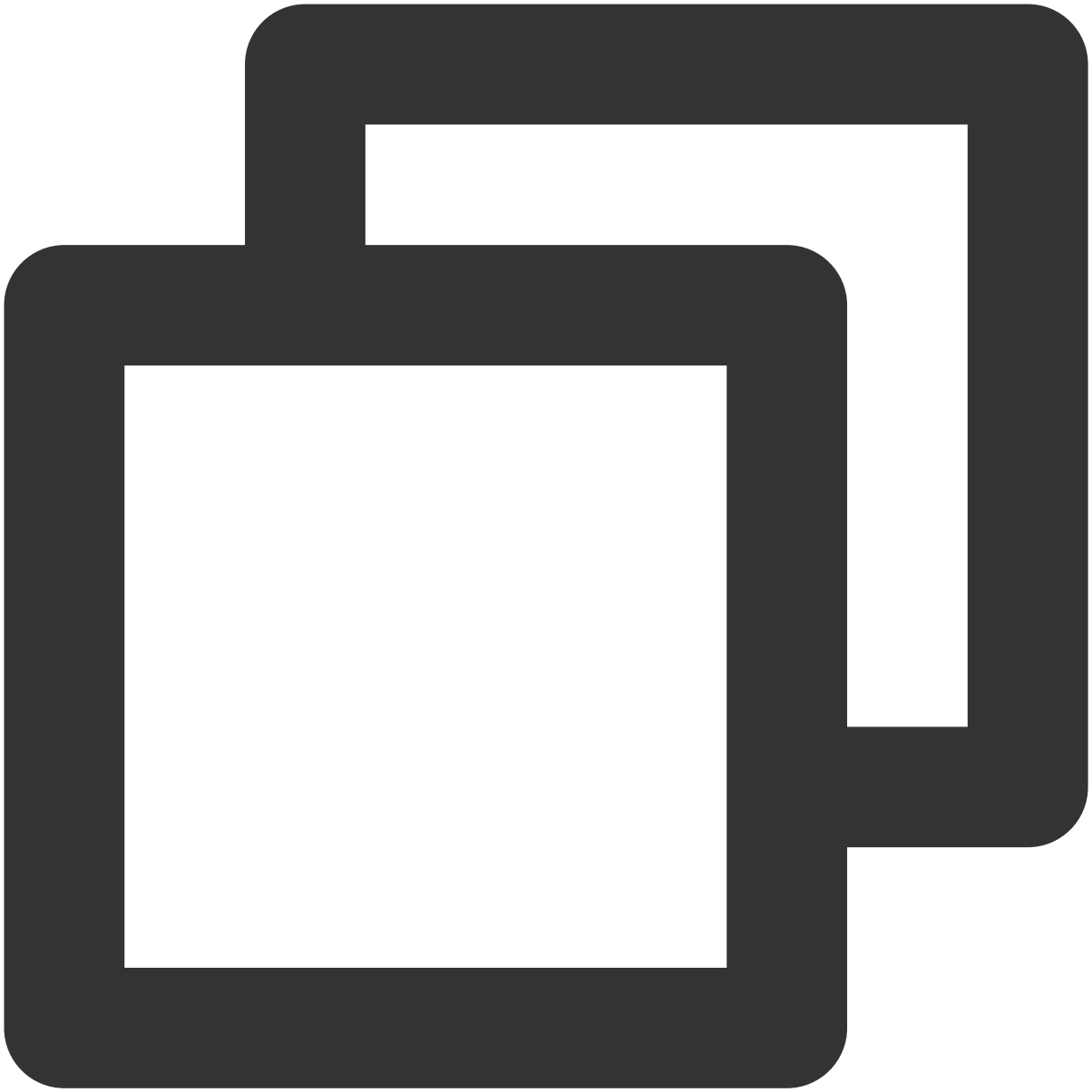
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageView2/1/w/
```

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：



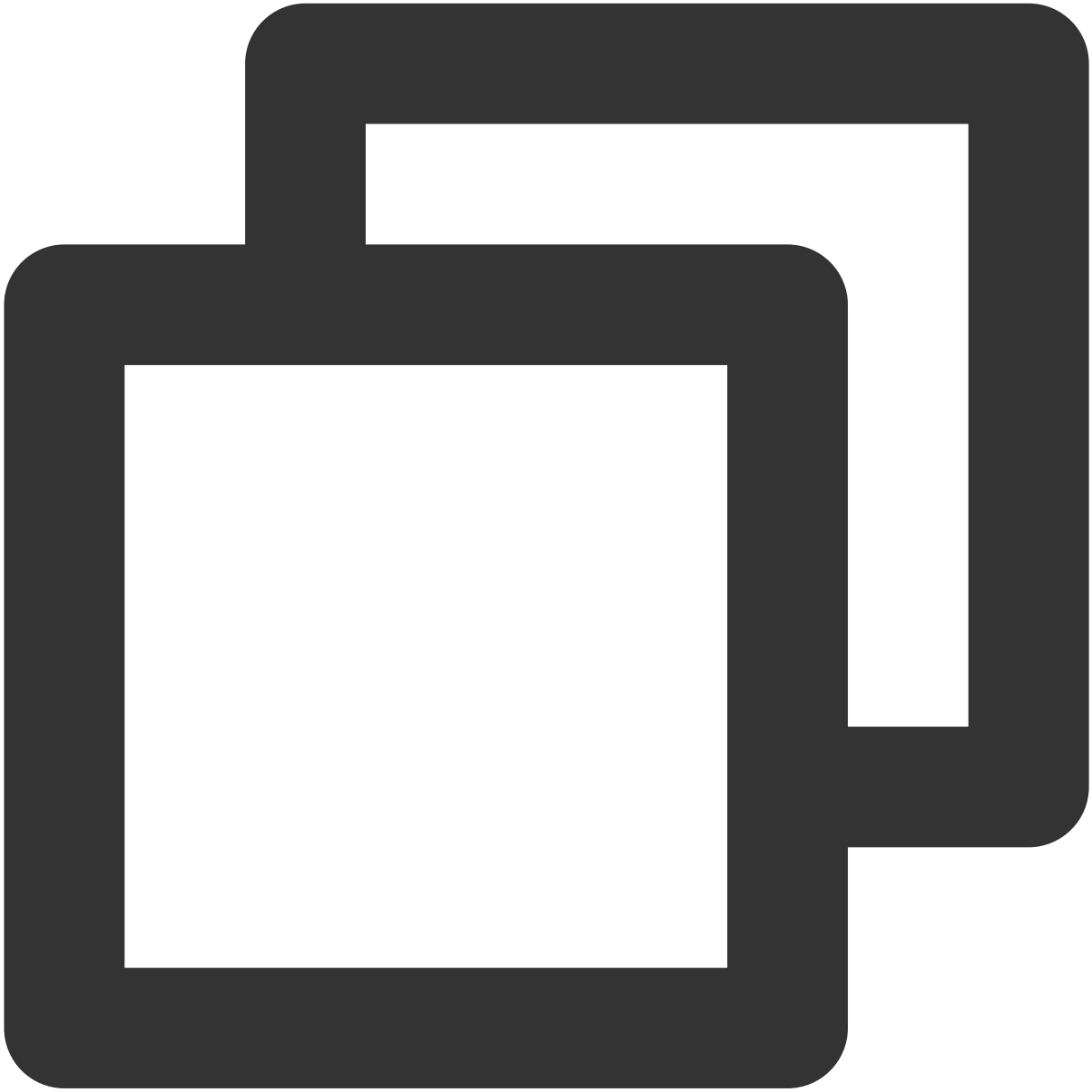
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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
```

管道操作符

最近更新时间：2024-01-18 15:11:52

功能概述

腾讯云数据万象的管道操作符 `|` 能够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。

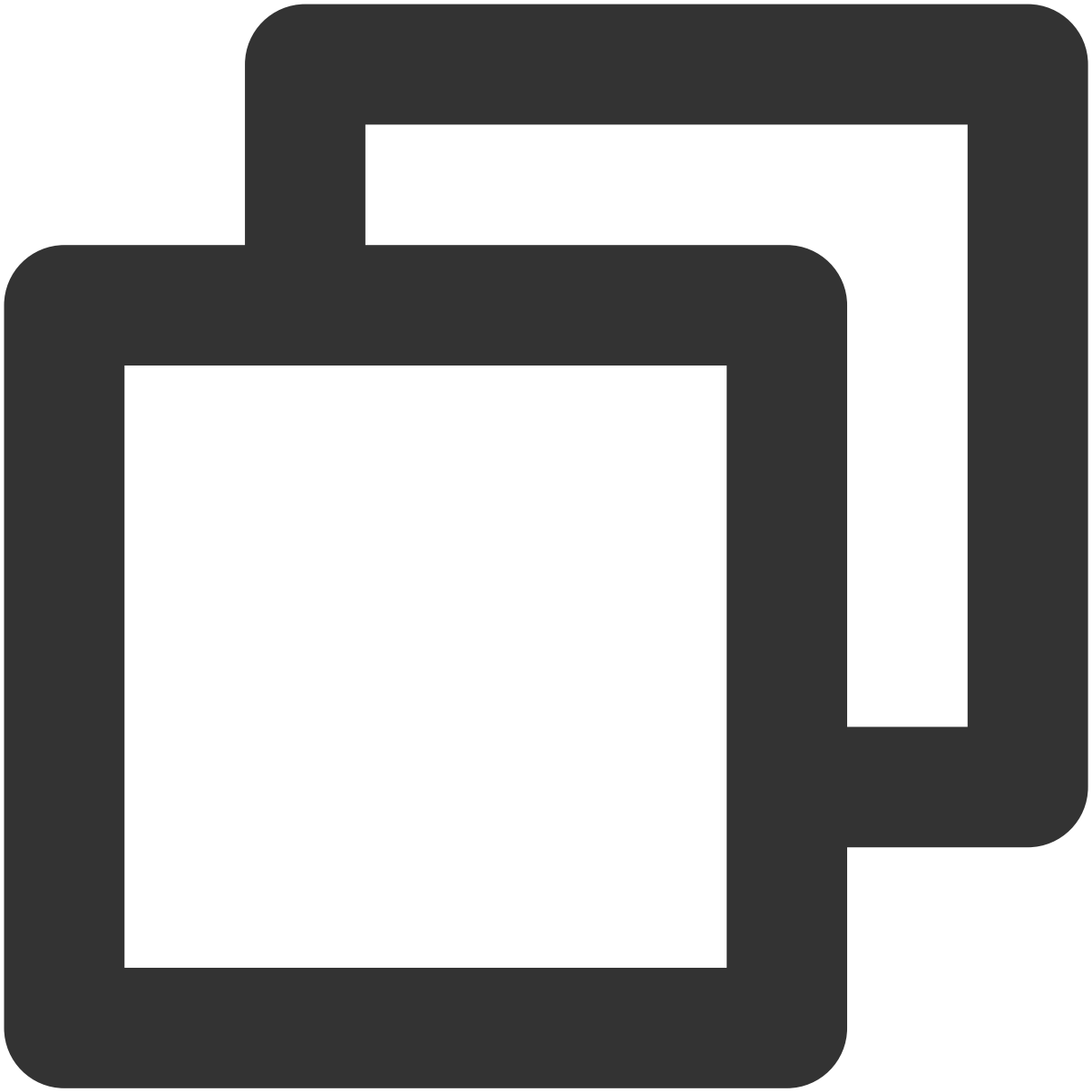
接口示例

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：

版权所有：腾讯云计算（北京）有限责任公司



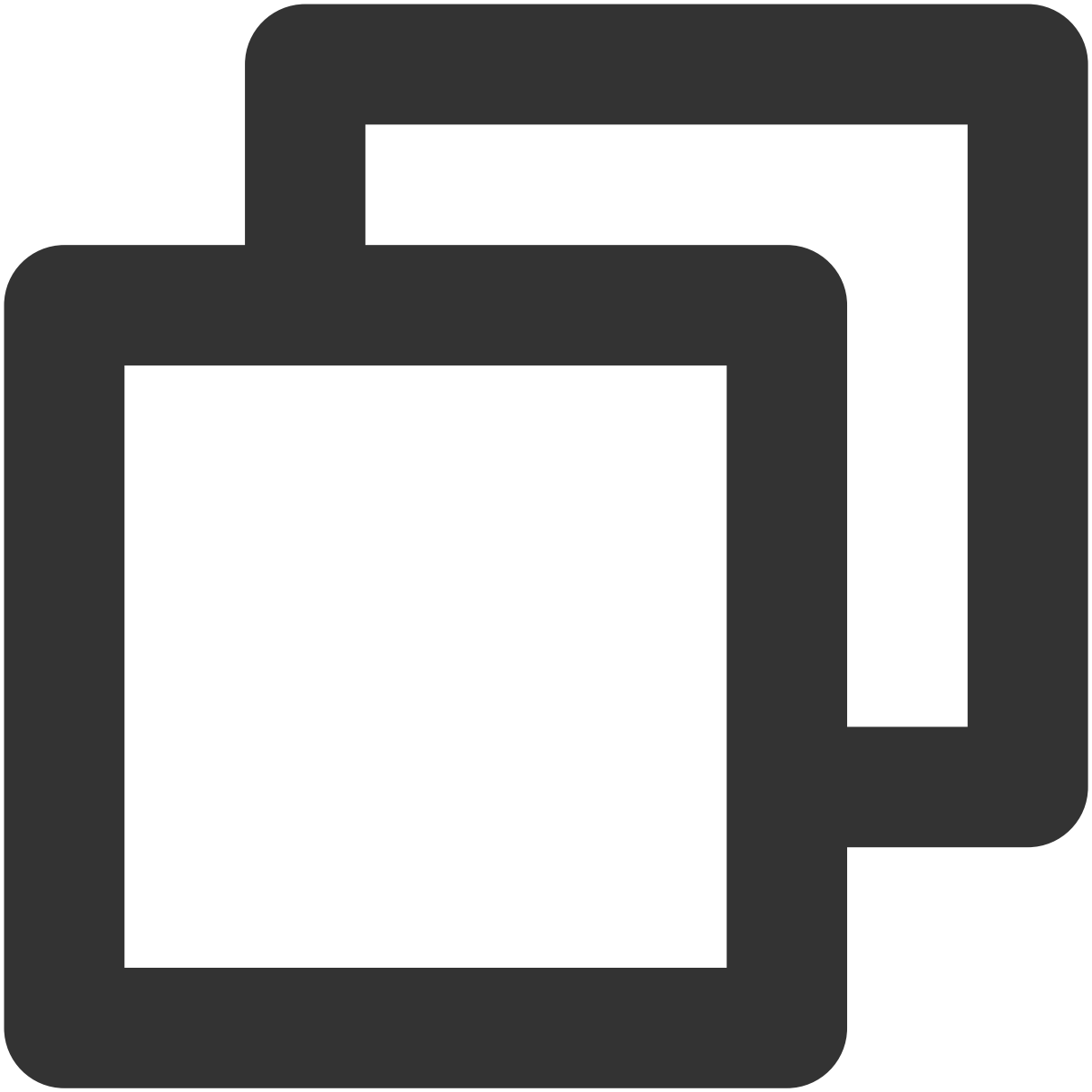
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg
```

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：



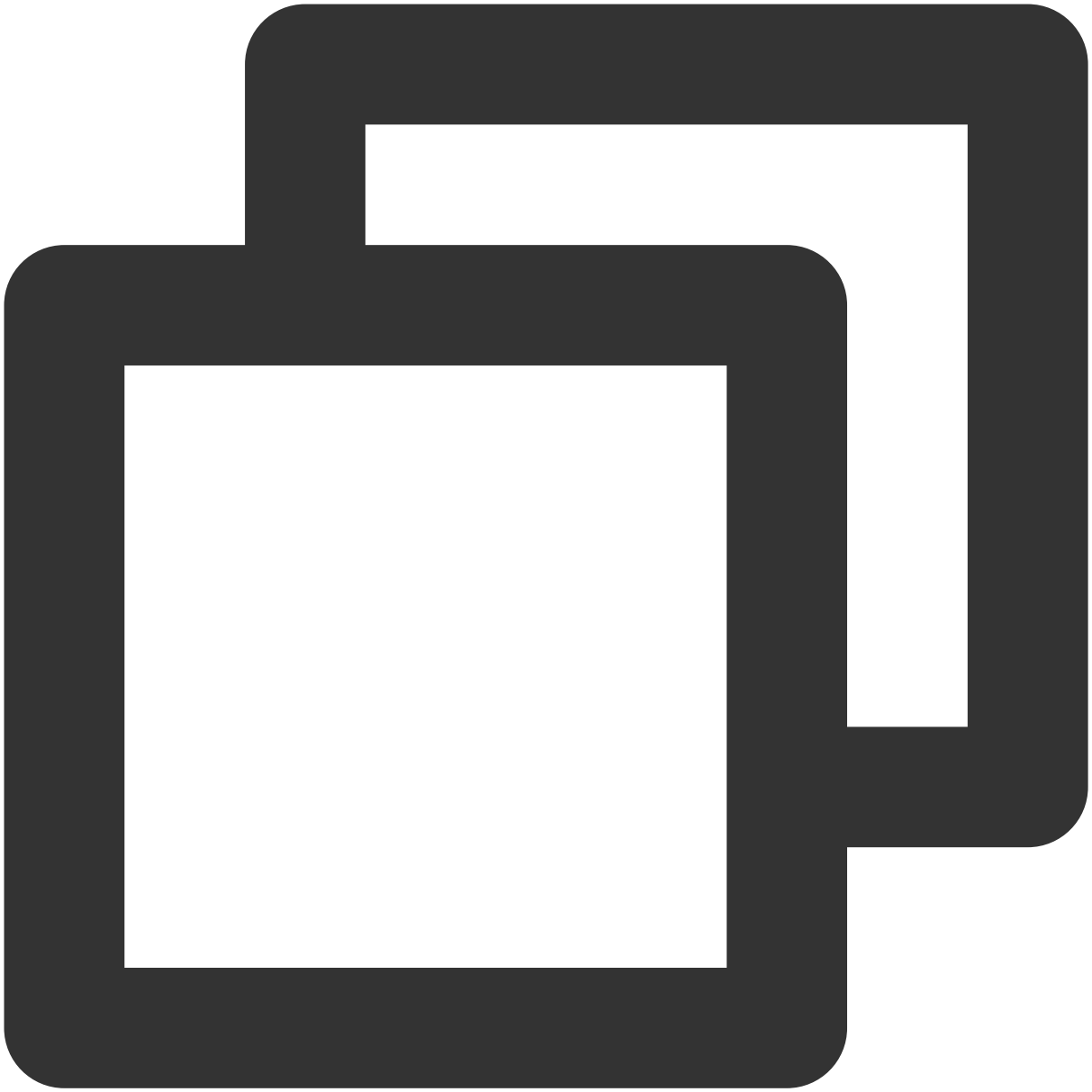
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

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：

版权所有：腾讯云计算（北京）有限责任公司



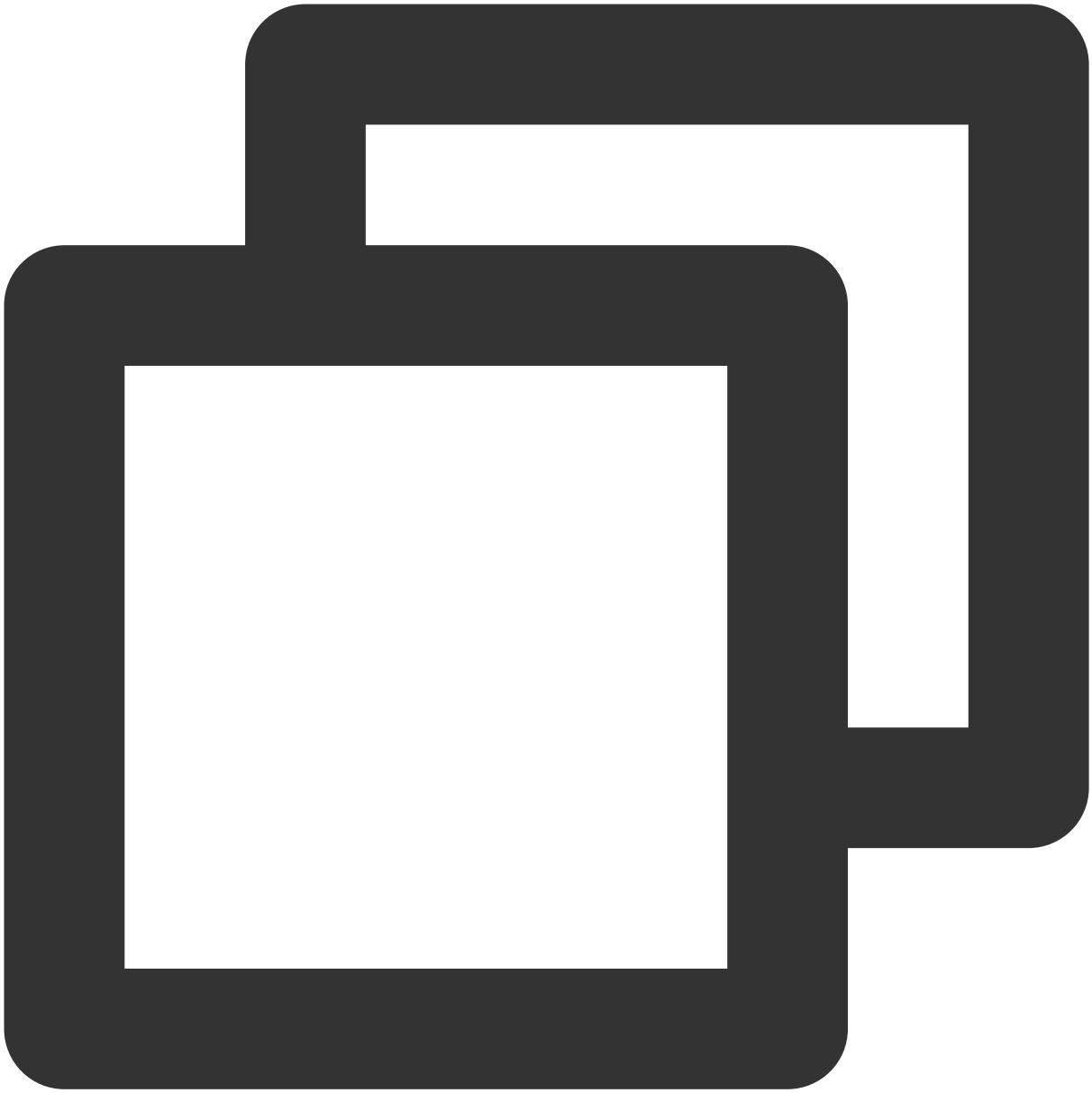
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?imageMogr2/thum
```

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：



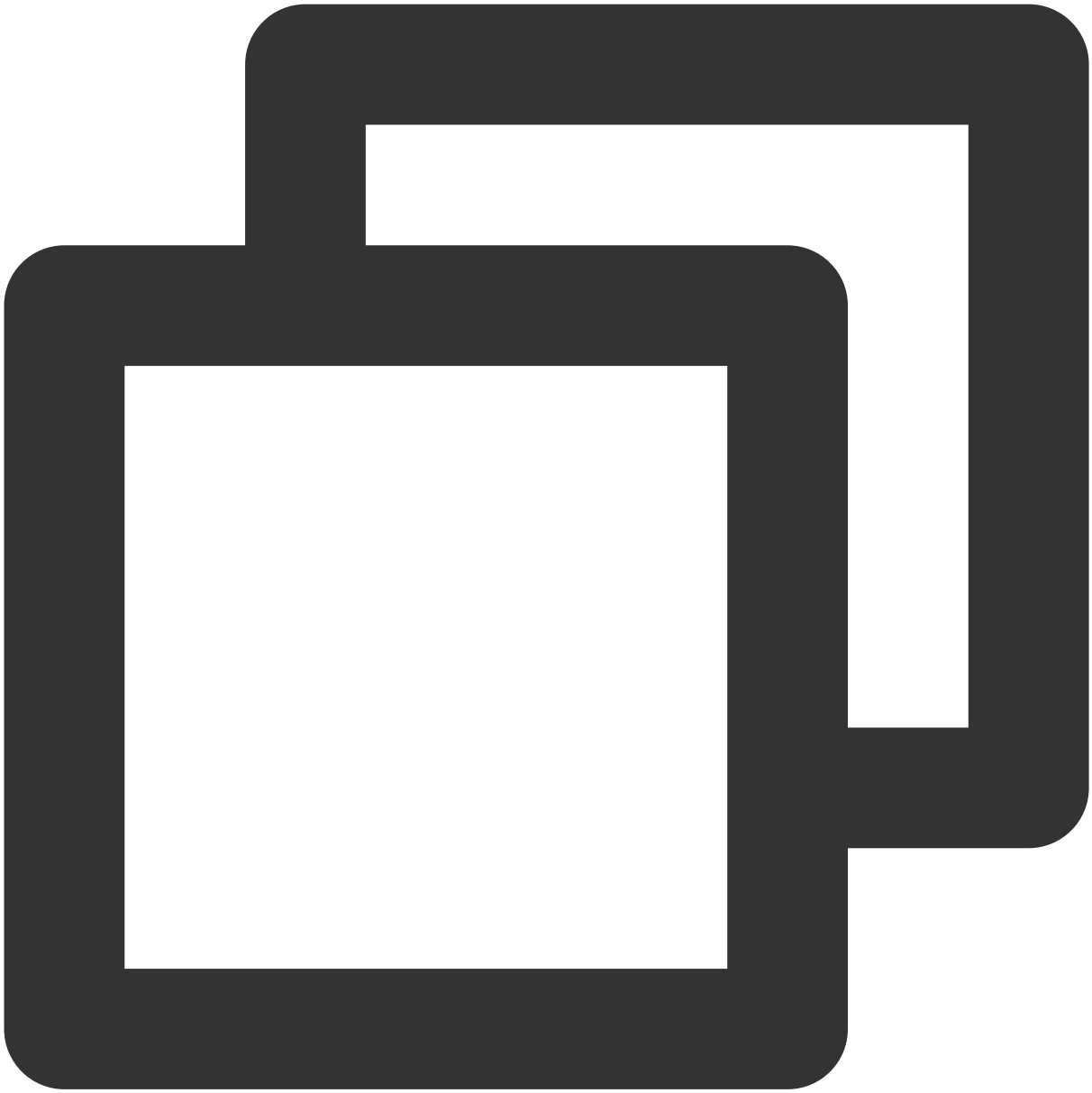
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?q-sign-algorith
```

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
```

限制图片大小

最近更新时间：2024-01-18 15:11:52

功能概述

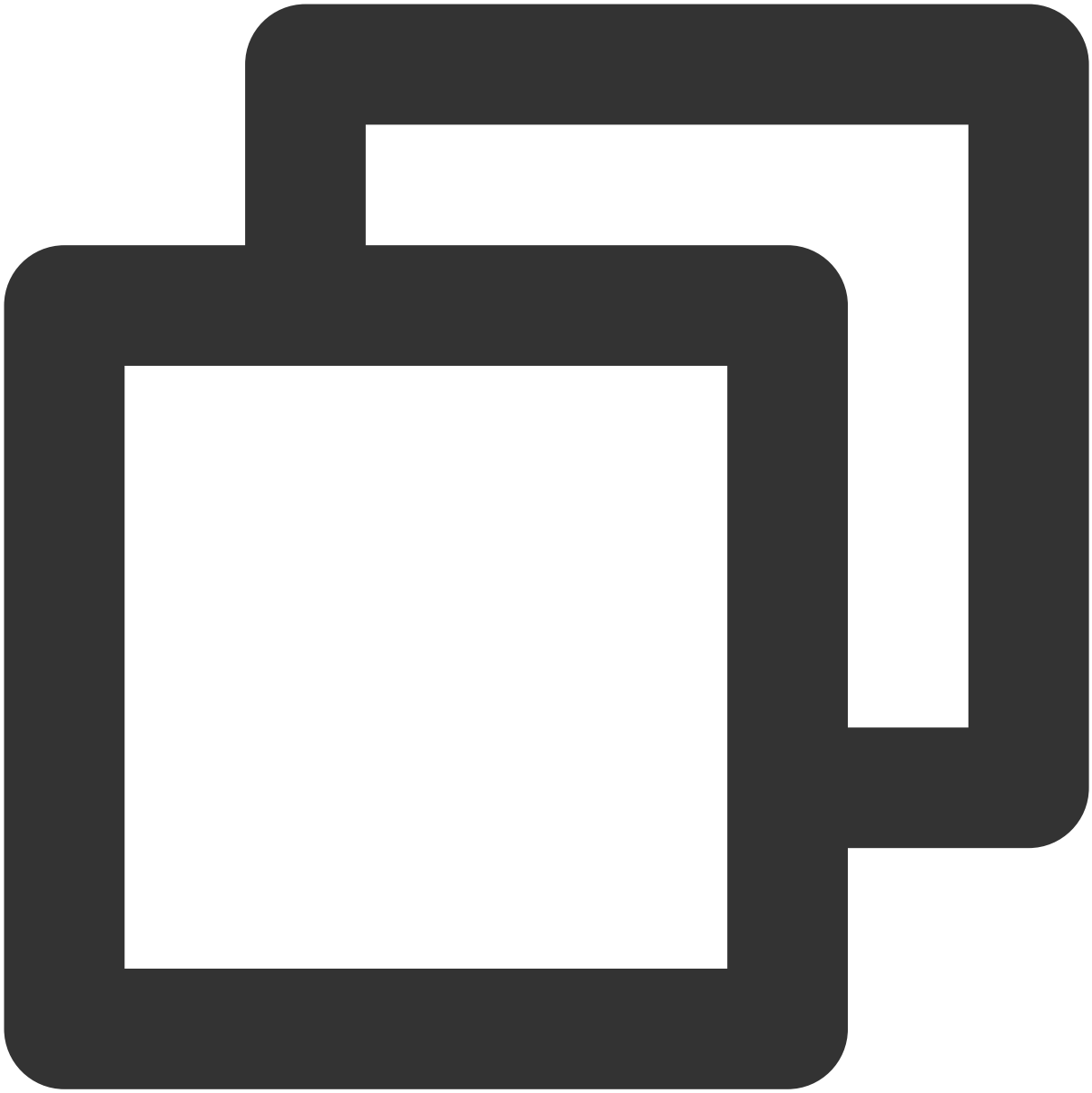
腾讯云数据万象通过 **imageMogr2/size-limit** 接口可限制图片处理（例如缩放、压缩等）后的文件大小。

限制说明

支持格式：仅支持对处理后为 JPG 格式的图片限制大小。
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。

接口示例

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/size-limit 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

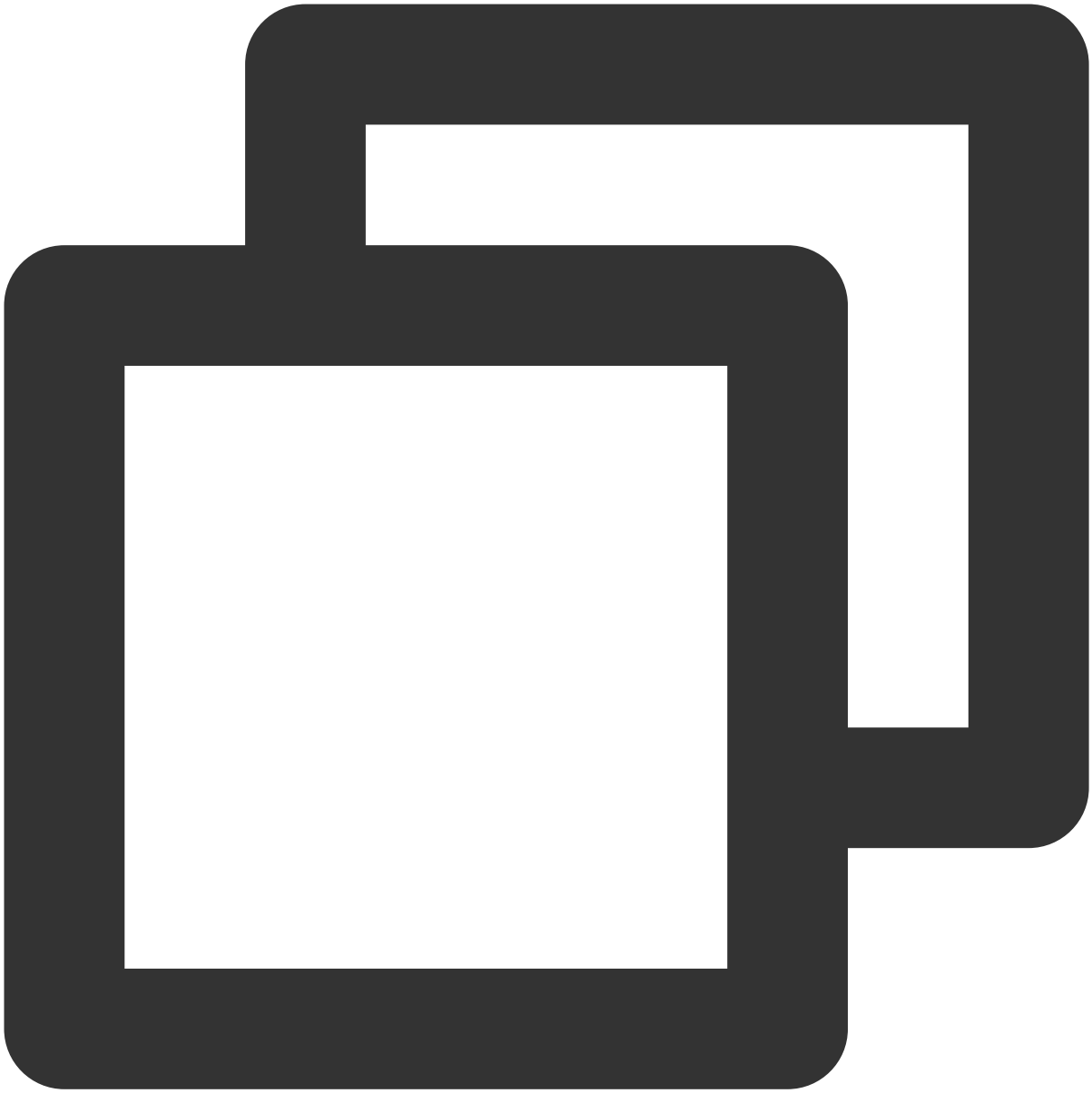
操作名称：size-limit。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。
size-limit	限制图片转换后的大小，支持以兆字节（m）和千字节（k）为单位 仅支持 JPG 格式的图片，可以用于限制处理后图片的大小。 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! 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.png?imageMogr2/strip/format/jpg/size-limit/15k!
/ignore-error/1	当处理参数中携带此参数时，针对文件过大、参数超限等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

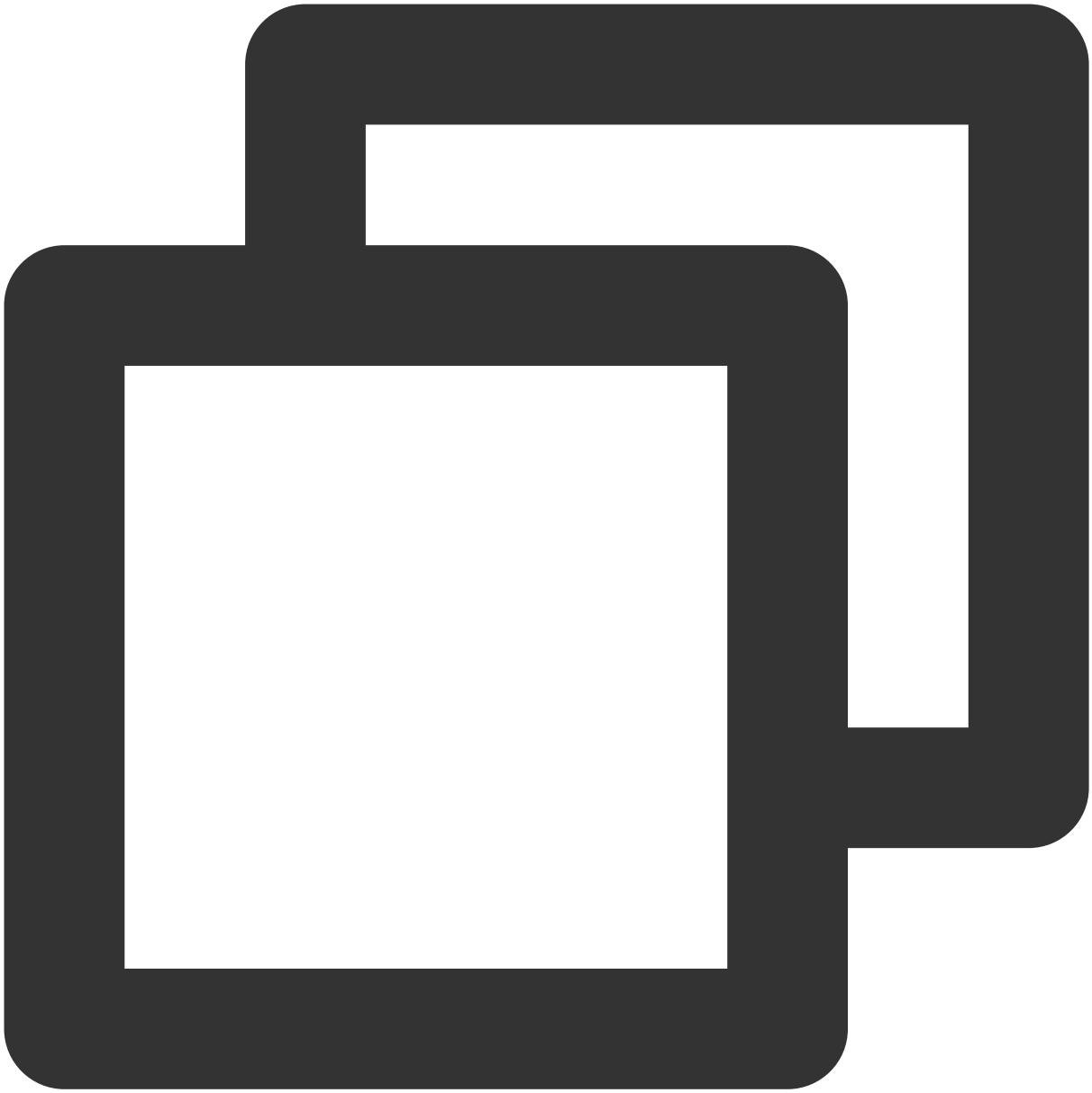
将 PNG 图片转换为 JPG 格式，并限制图片大小为15KB，示例如下：



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/strip
```

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：



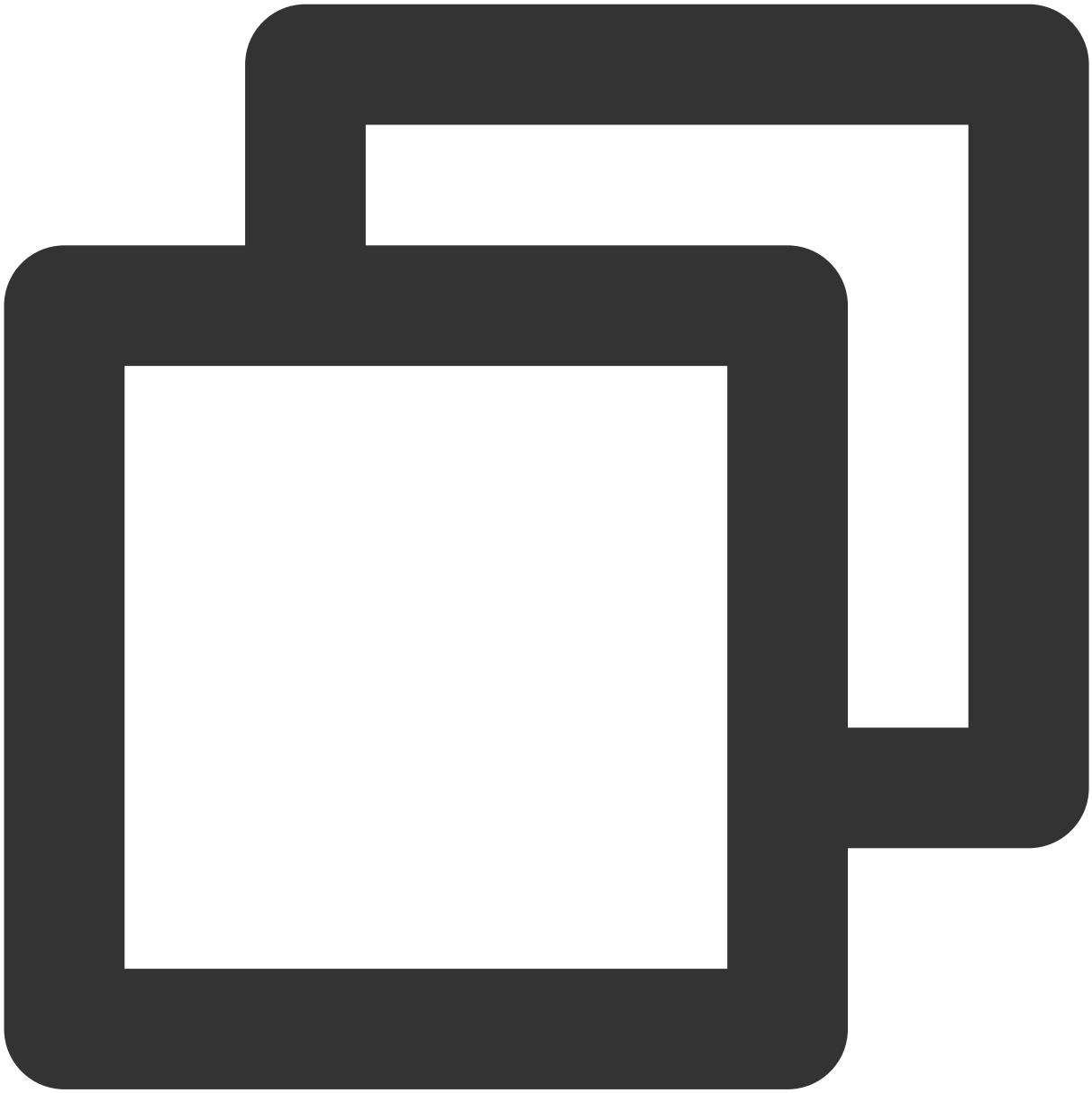
```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.png?q-sign-algorithm
```

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
```

支持字体列表

最近更新时间：2024-01-18 15:11:52

数据万象支持字体列表如下。

字体名称	支持语言
simfang仿宋.ttf	中/英
simhei黑体.ttf	中/英
simkai楷体.ttf	中/英
simsun宋体.ttc	中/英
STHeiti Light华文黑体.ttc	中/英
STHeiti Medium.ttc	中/英
幼圆.TTF	中/英
ahronbd.ttf	英文
arial.ttf	英文
ariblk.ttf	英文
Helvetica.dfont	英文
HelveticaNeue.dfont	英文

版权所有：腾讯云计算（北京）有限责任公司

图片样式
增加样式

最近更新时间：2024-01-18 15:11:52

功能描述

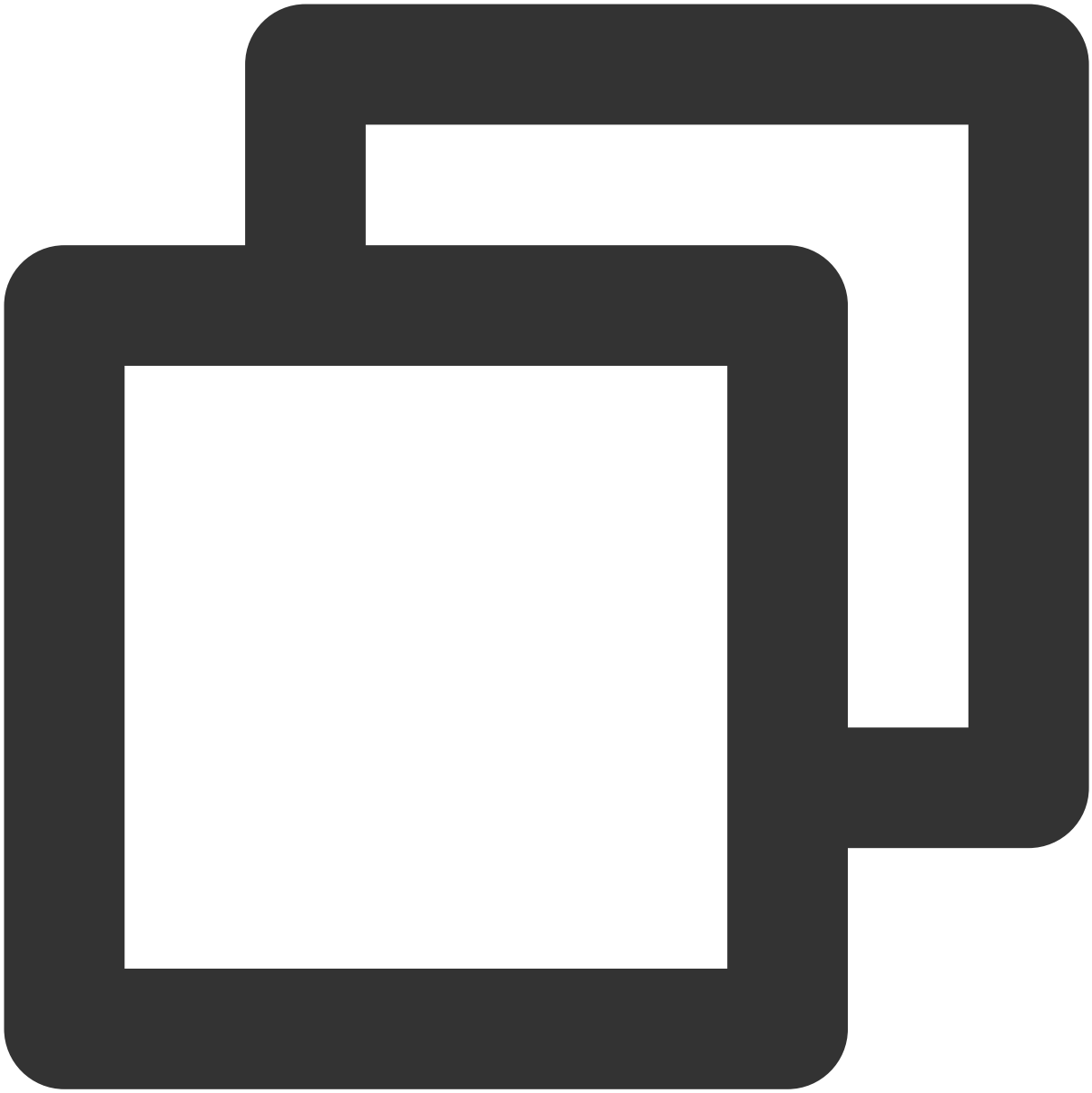
该接口用于对某一个存储桶设置样式功能，设置样式后，可以方便地对图片进行统一的处理。

说明：

Bucket 接口目前频控存在限制要求，写请求最多1qps，读请求最多10qps。

请求

请求示例



```
PUT /?style HTTP/1.1
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
<XML File>
```

说明：

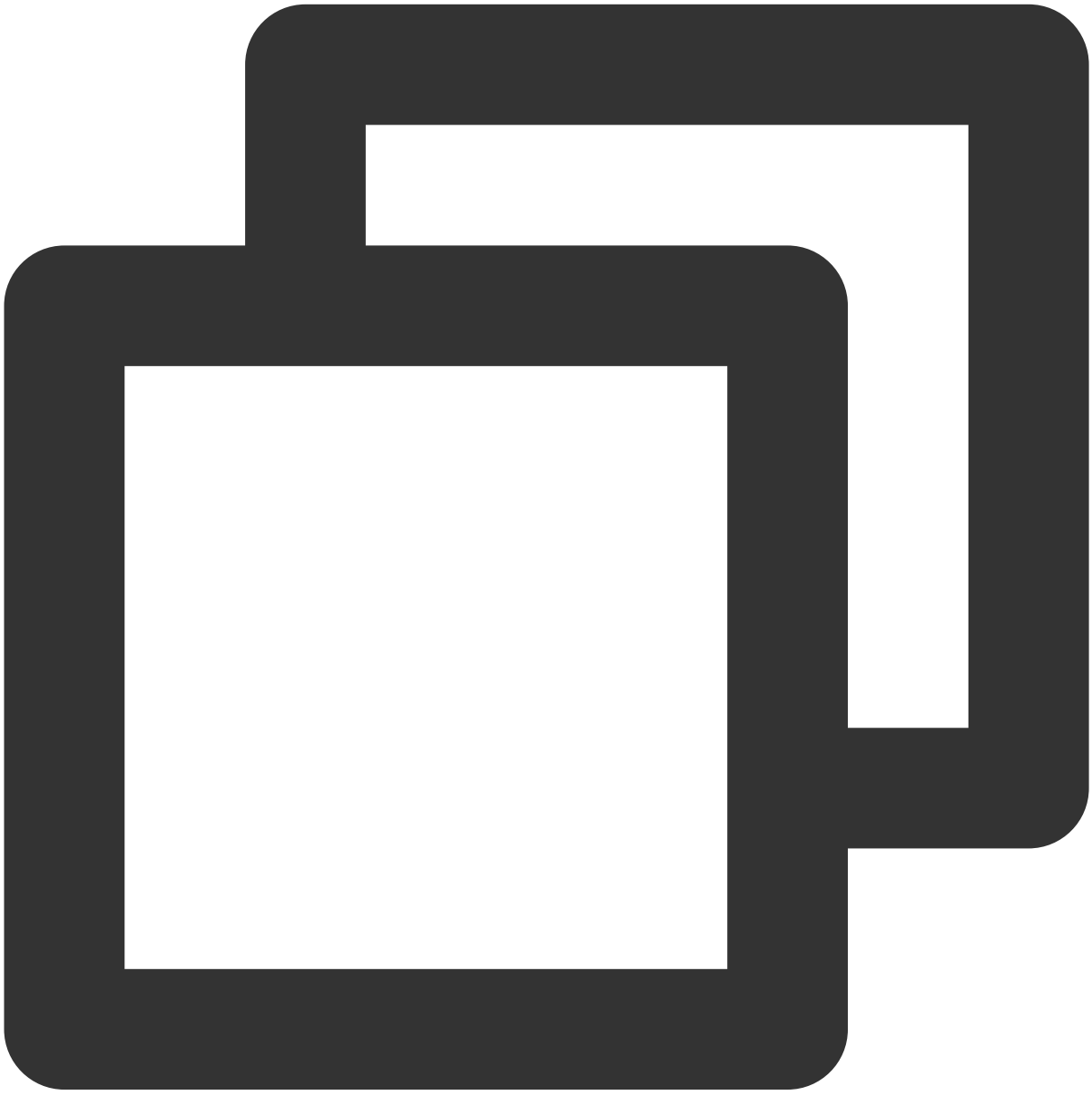
Authorization: Auth String（详情请参见 请求签名 文档）。

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，详情请参阅 公共请求头部 文档。

请求体



```
<?xml version="1.0" encoding="UTF-8" ?>
<AddStyle>
    <StyleName>string</StyleName>
    <StyleBody>string</StyleBody>
</AddStyle>
```

具体节点描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
AddStyle	无	待添加的样式信息	Container	是

Container 节点 AddStyle 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
StyleName	AddStyle	样式名称	string	是
StyleBody	AddStyle	样式详情	string	是

响应

响应头

此接口仅返回公共响应头部，详情请参阅 公共响应头部 文档。

响应体

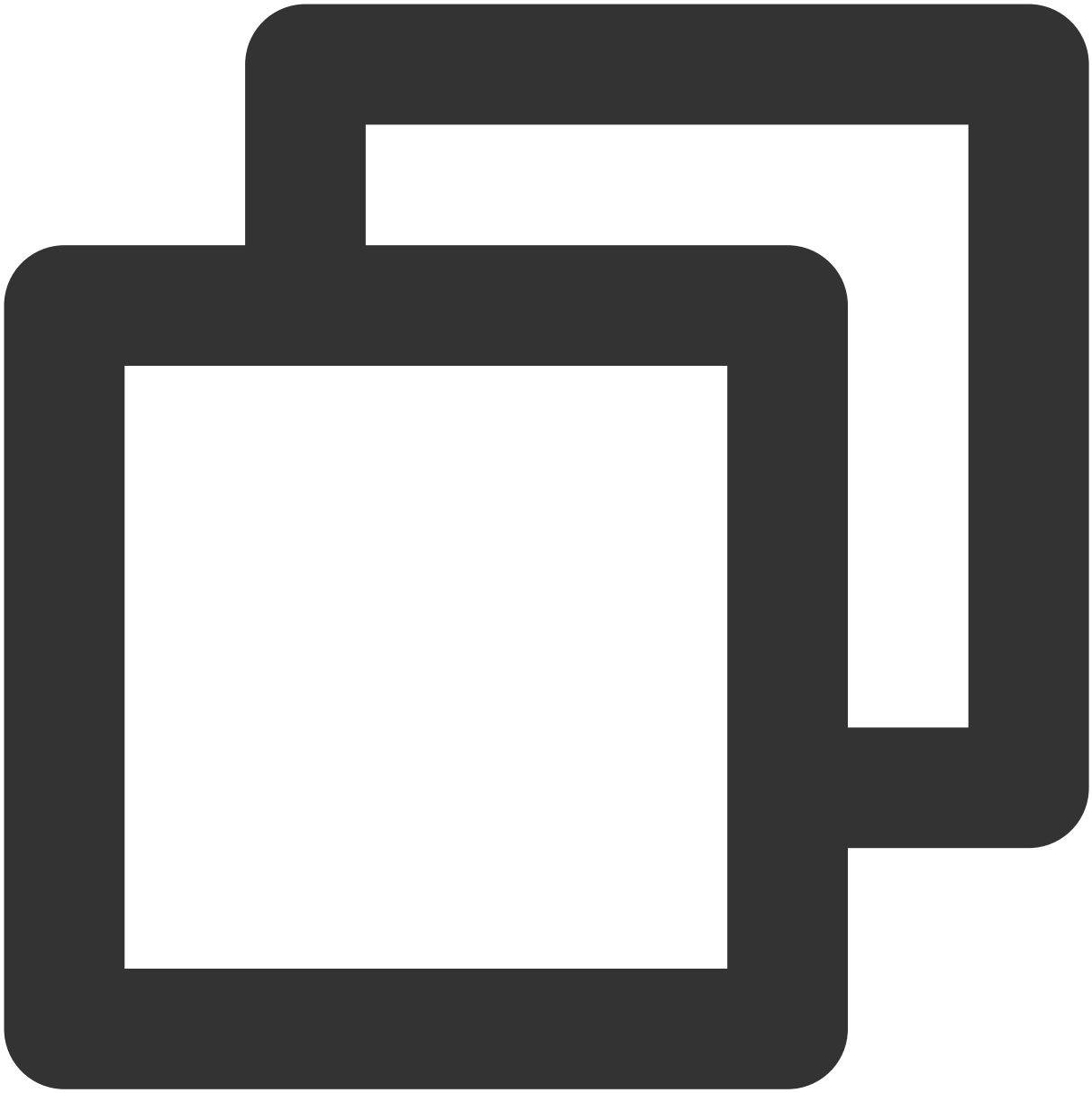
该响应体返回为空。

错误码

此接口无特殊错误信息，全部错误信息请参见 错误码 文档。

实际案例

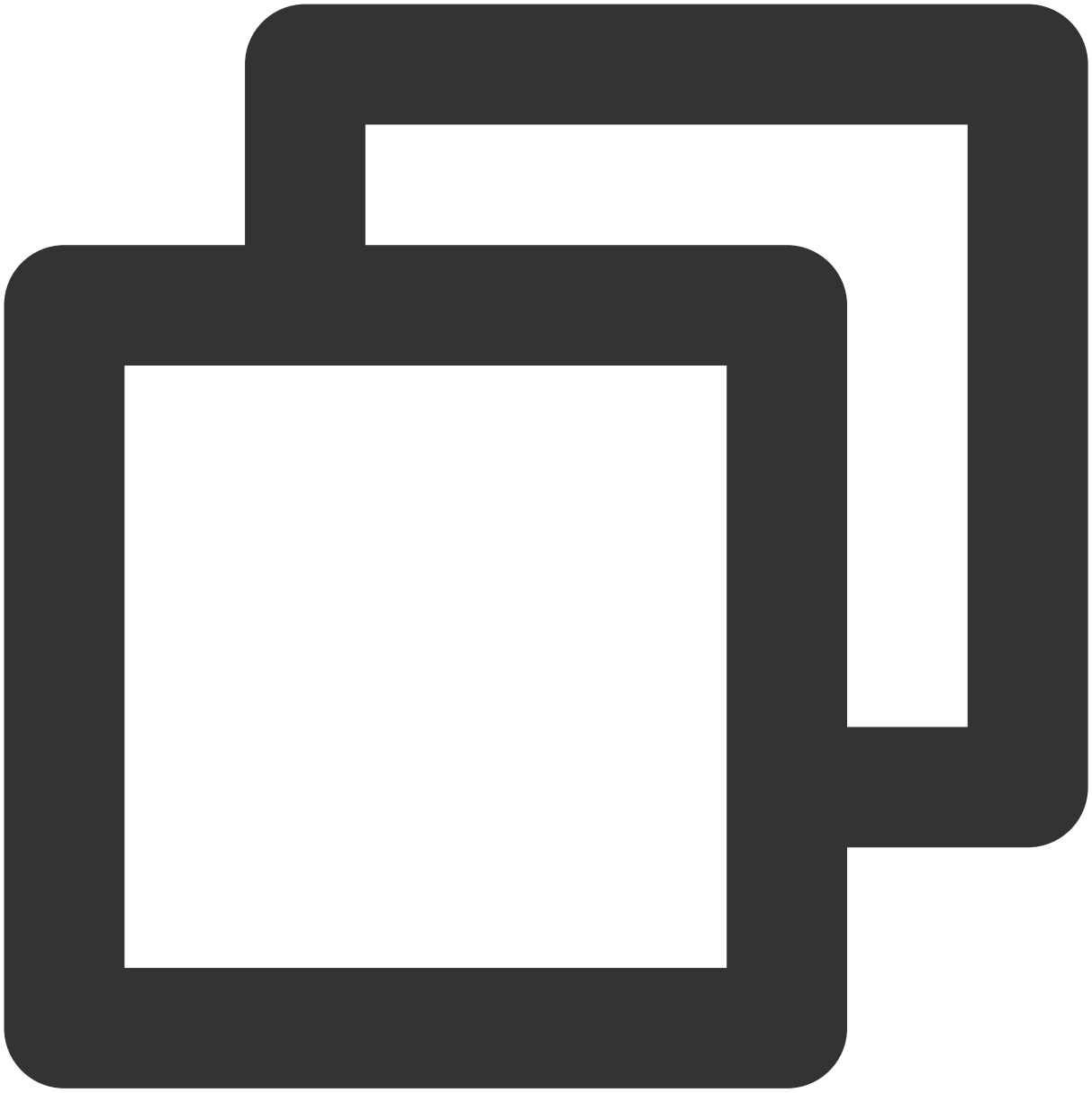
请求



```
PUT /?style HTTP/1.1
Host: examplebucket-1250000000.pic.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX

<?xml version="1.0" encoding="UTF-8" ?>
<AddStyle>
    <StyleName>style_name</StyleName>
    <StyleBody>imageMogr2/thumbnail/!50px</StyleBody>
</AddStyle>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Tue, 03 Apr 2019 09:06:16 GMT
x-cos-request-id: xxxxxxxxxxxxxxxxxx
```

查询样式

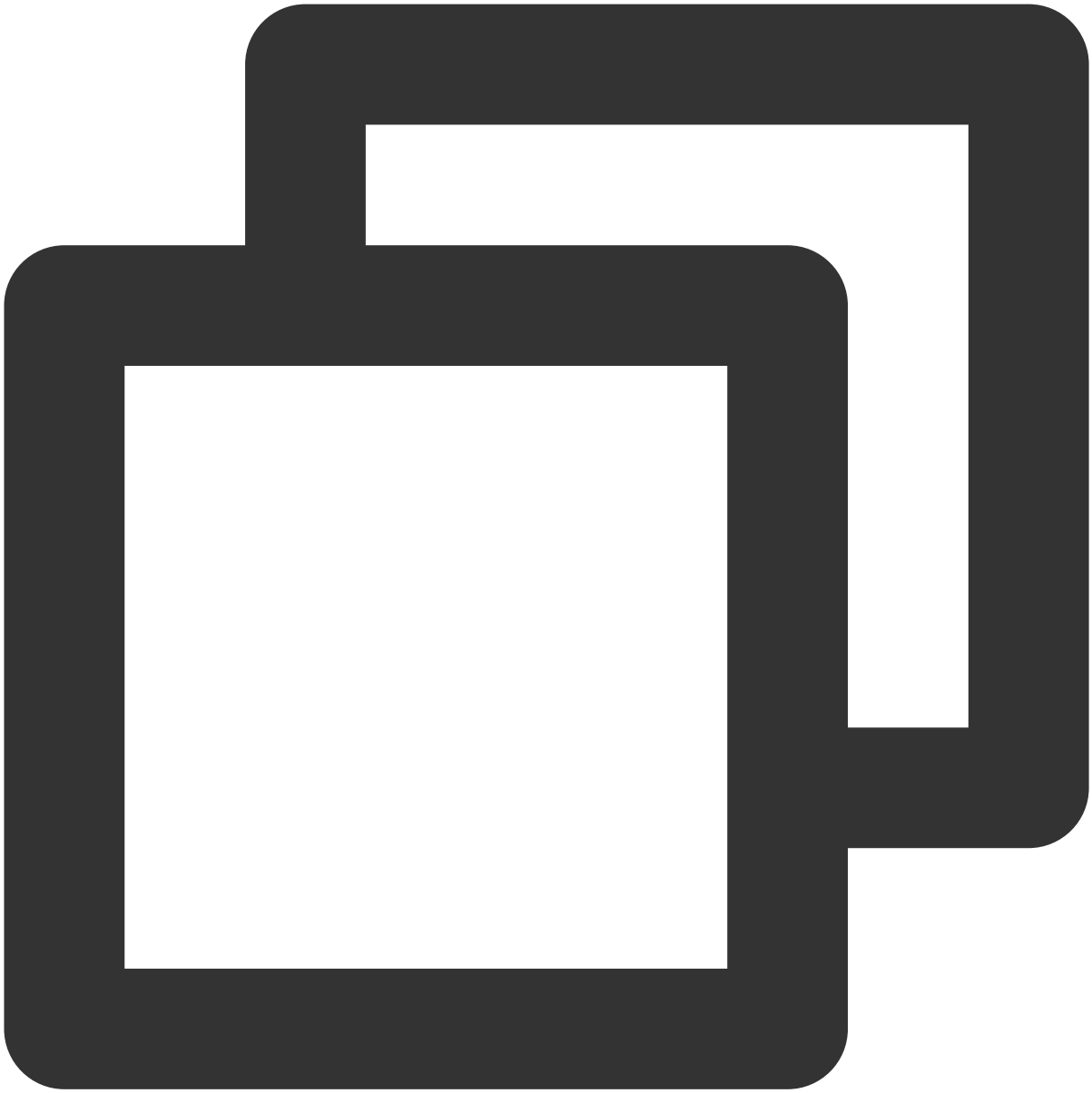
最近更新时间：2024-01-18 15:11:52

功能描述

用于查询该 Bucket 下已有的样式。若不使用请求体，则查询该 Bucket 下的全部样式，若使用请求体，则查询请求体中携带的样式名信息。

请求

请求示例



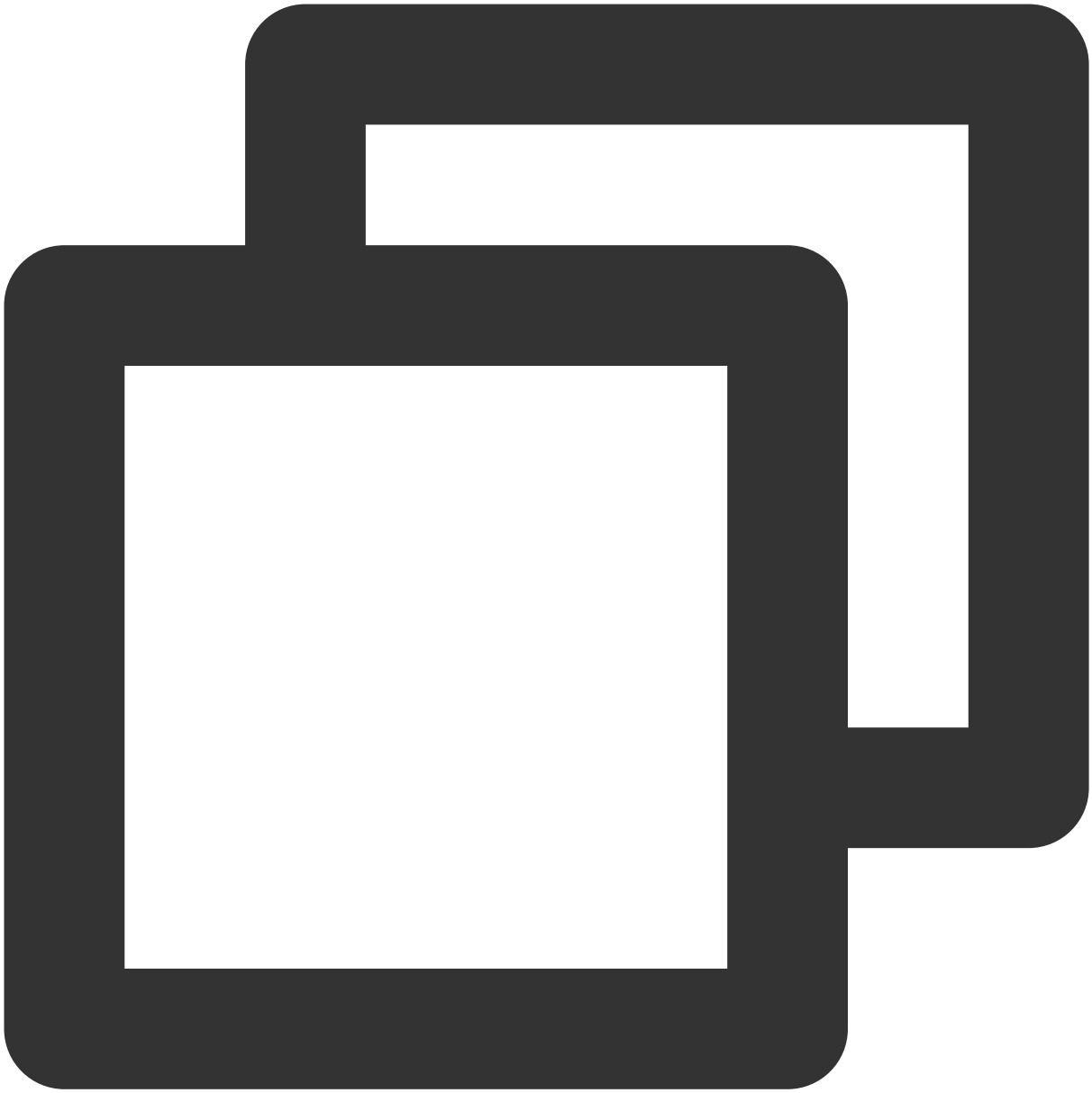
```
GET /?style HTTP/1.1
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

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。

请求行



```
GET /?style HTTP/1.1
```

该 API 接口接受 GET 请求。

请求头

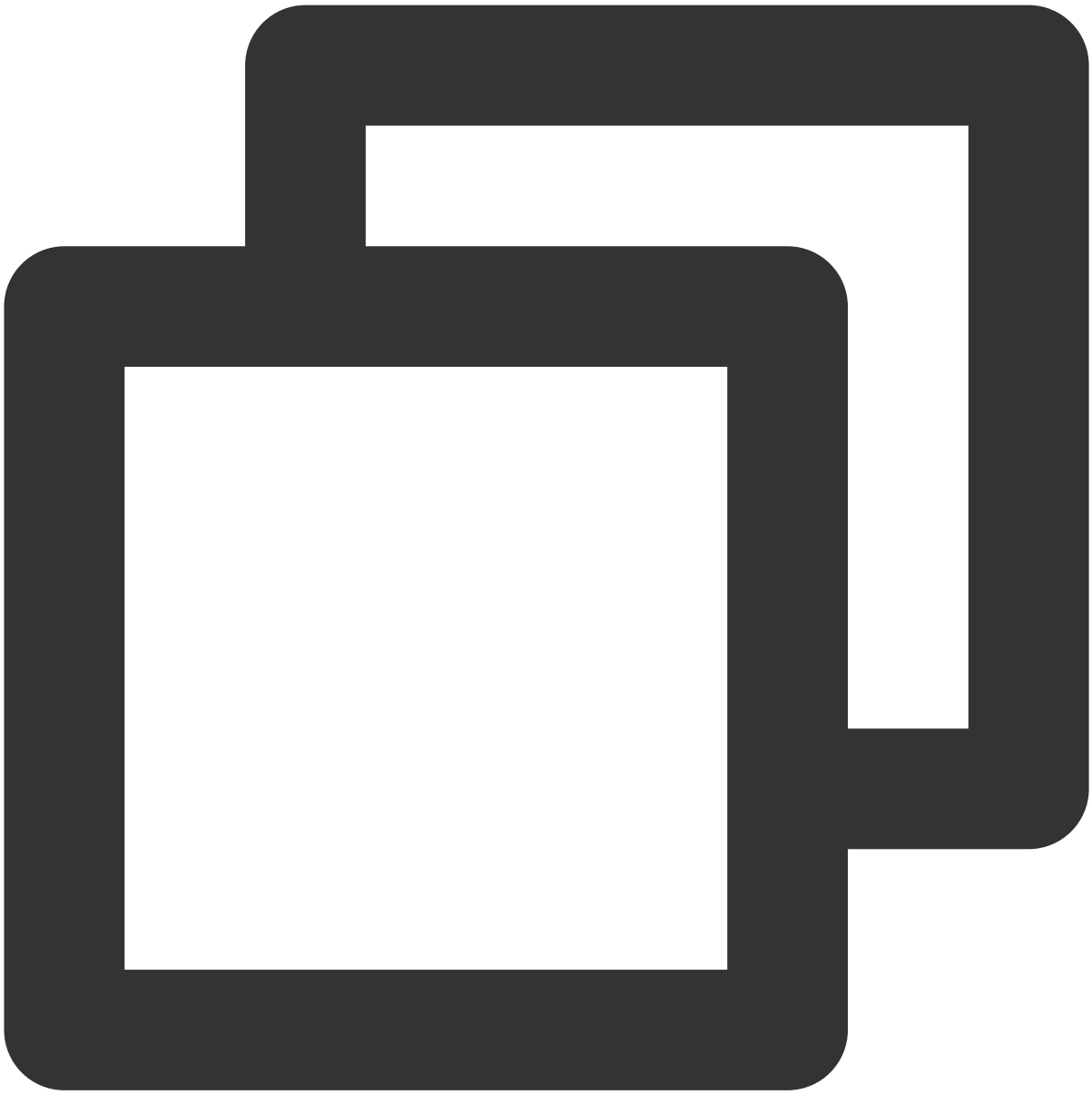
公共头部

该请求操作的实现使用公共请求头，了解公共请求头详情，请参阅 公共请求头部 章节。

非公共头部

该请求操作无特殊的请求头部信息。

请求体



```
<?xml version="1.0" encoding="UTF-8" ?>
<GetStyle>
  <StyleName>string</StyleName>
</GetStyle>
```

响应

响应头

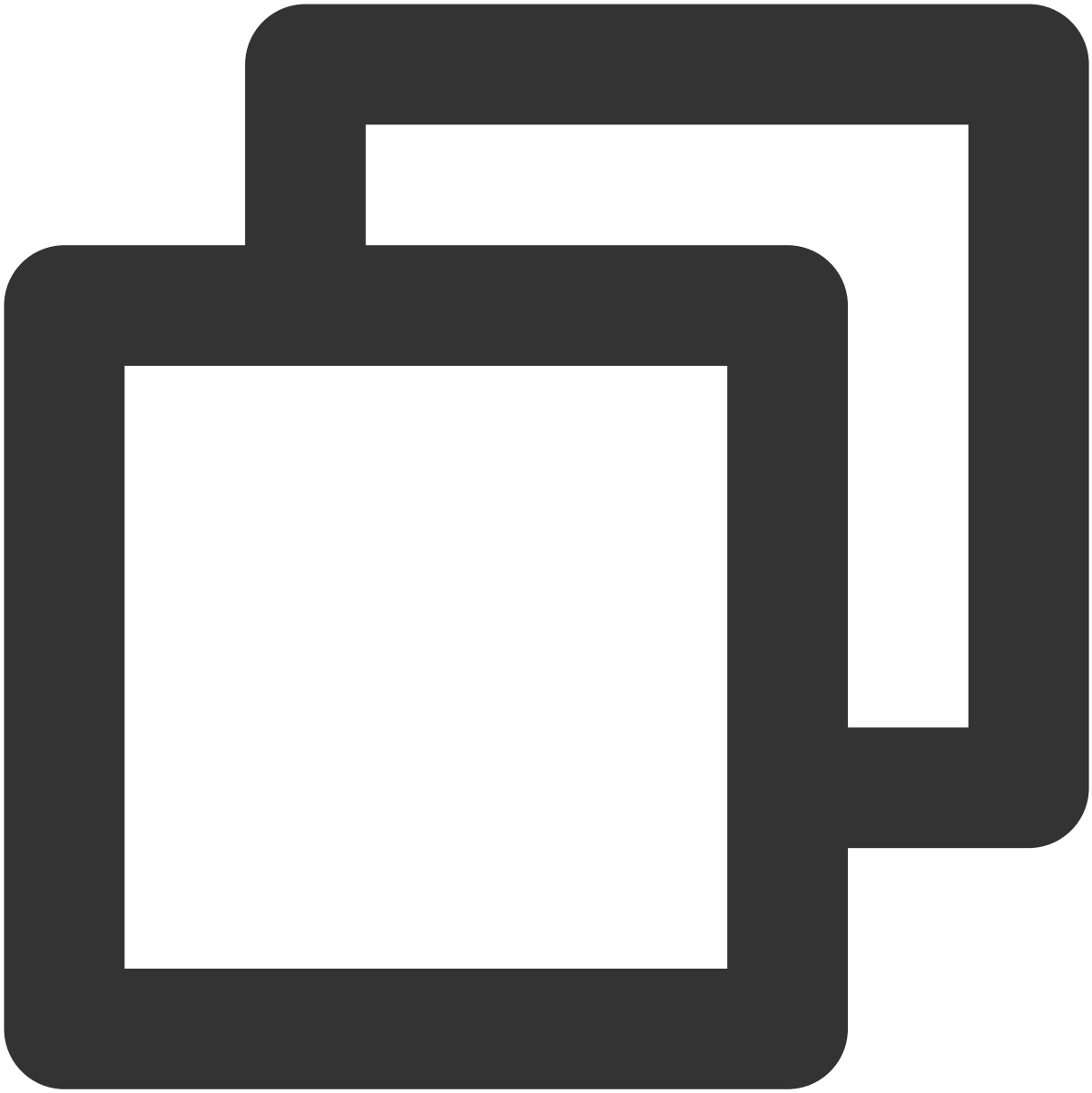
公共响应头

该响应使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

特有响应头

该响应无特殊的响应头。

响应体



```
<StyleList>
  <StyleRule>
  <StyleName>string</StyleName>
  <StyleBody>string</StyleBody>
</StyleRule>
  <StyleRule>
  <StyleName>string</StyleName>
  <StyleBody>string</StyleBody>
</StyleRule>
</StyleList>
```

版权所有：腾讯云计算（北京）有限责任公司

删除样式

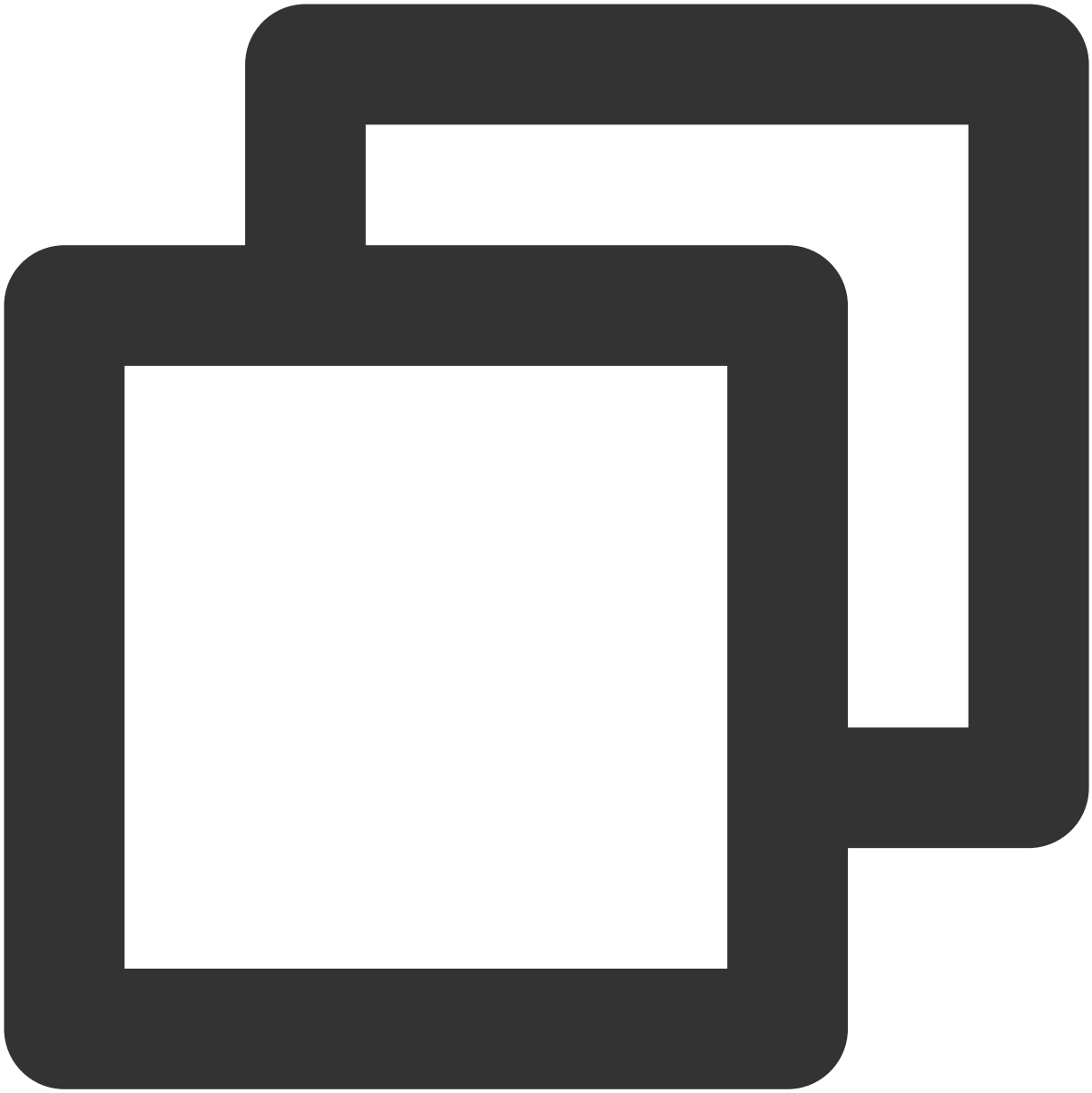
最近更新时间：2024-01-18 15:11:52

功能描述

用于删除某一特定样式。

请求

请求示例



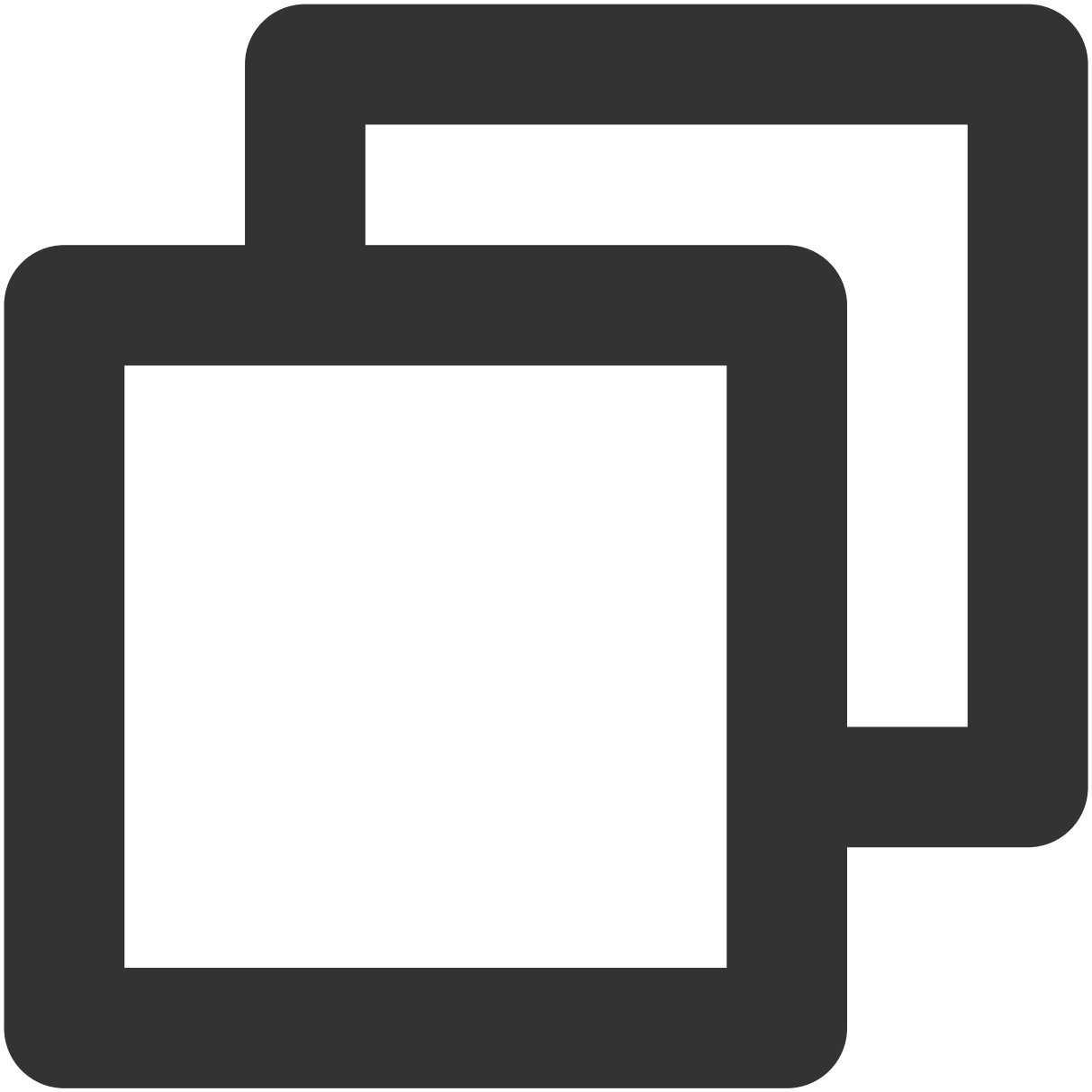
```
DELETE /?style HTTP/1.1
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
<XML File>
```

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。

请求行



```
DELETE /?style HTTP/1.1
```

该 API 接口接受 DELETE 请求。

请求头

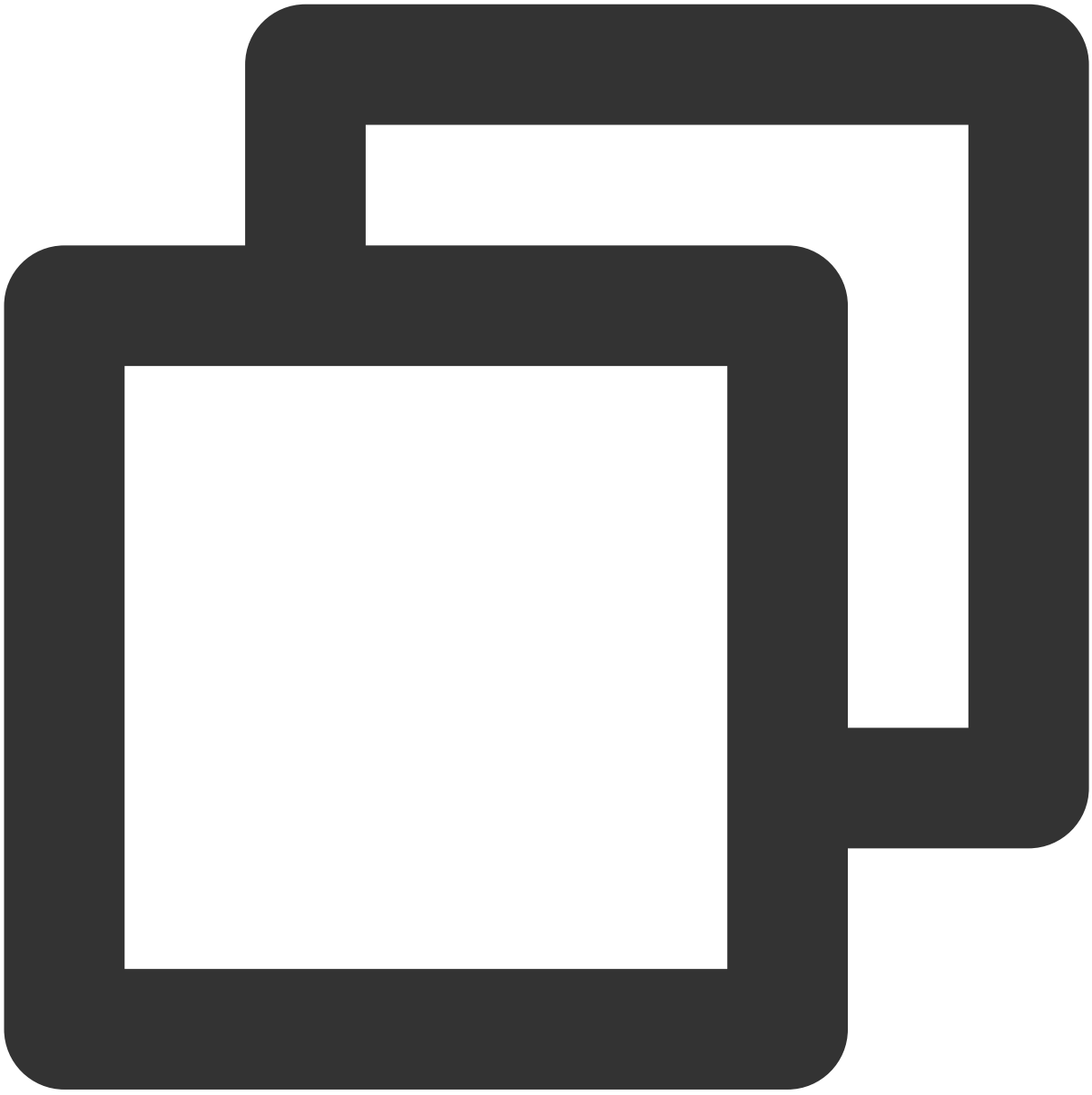
公共头部

该请求操作的实现使用公共请求头，了解公共请求头详情，请参阅 公共请求头部 章节。

非公共头部

该请求操作无特殊的请求头部信息。

请求体



```
<?xml version="1.0" encoding="UTF-8" ?>
<DeleteStyle>
  <StyleName>string</StyleName>
</DeleteStyle>
```

响应

响应头

公共响应头

该响应使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

特有响应头

该响应无特殊的响应头。

响应体

该响应体返回为空。

图片压缩
Guetzli 压缩
开通 Guetzli 压缩

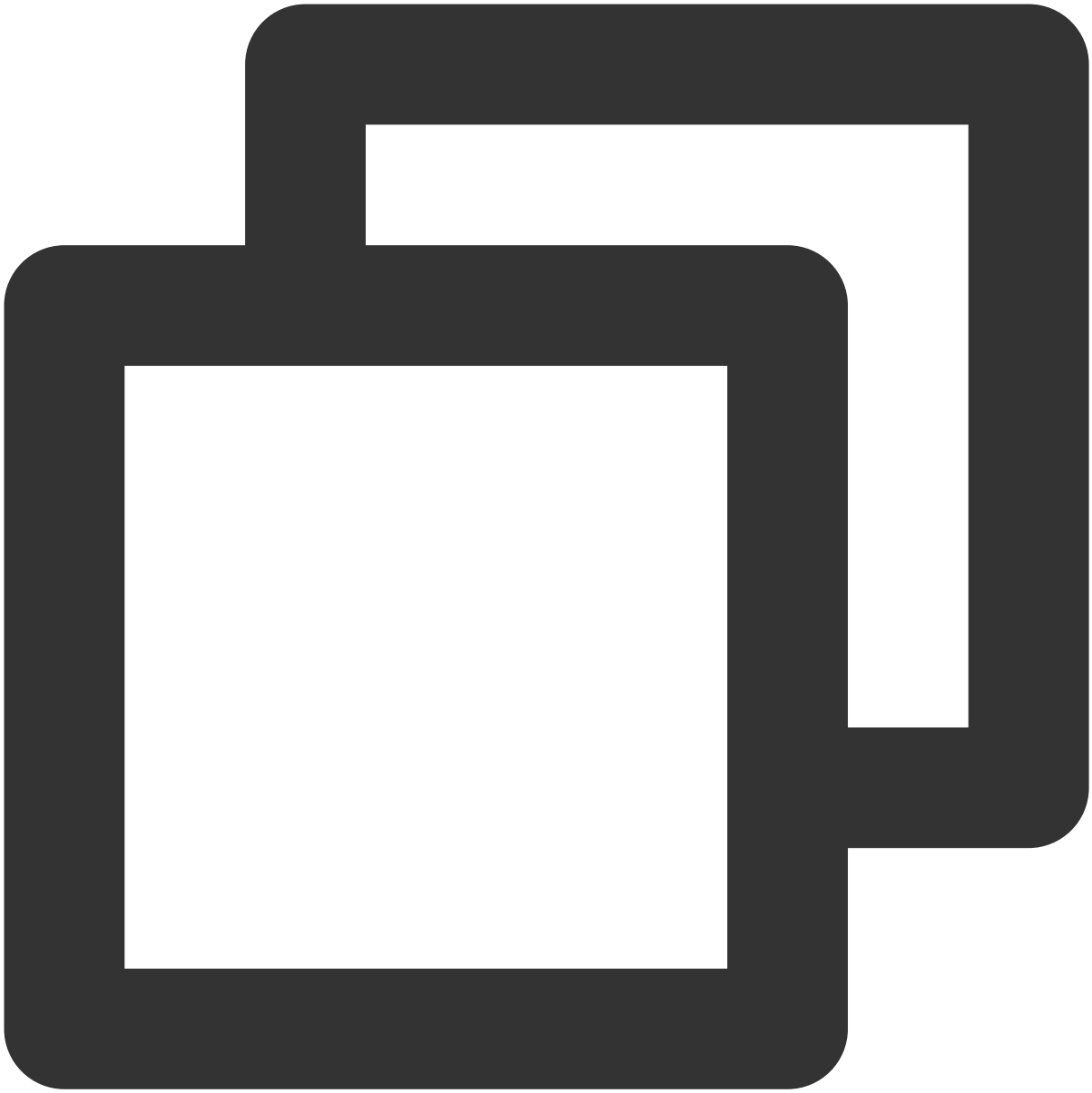
最近更新时间：2024-01-18 15:11:52

功能描述

对 Bucket 开通 Guetzli 压缩功能。

请求

请求示例



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

说明：

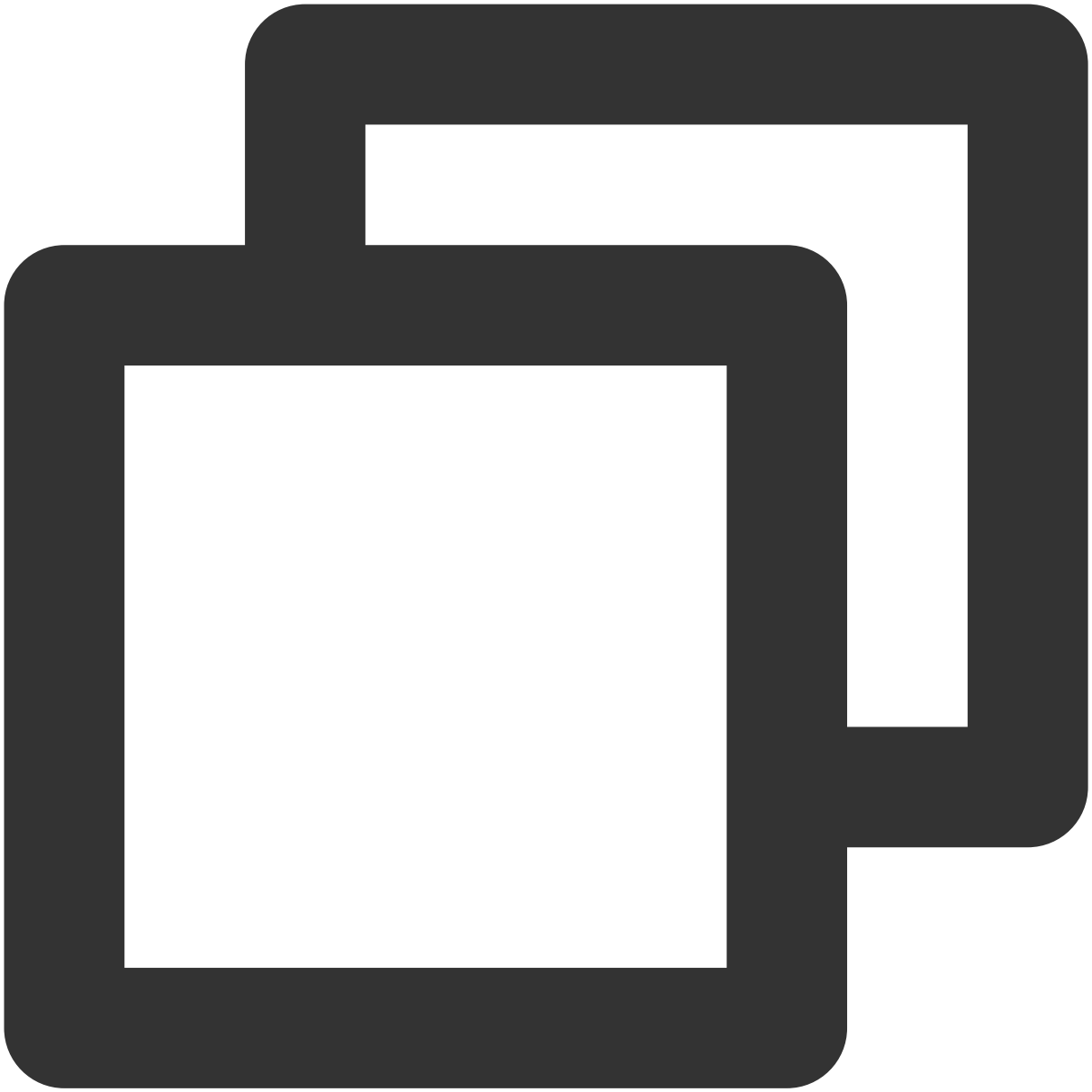
Authorization: Auth String（详情请参见 请求签名 文档）。

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。

Guetzli 压缩目前暂不支持通过 COS 域名使用，请参考示例使用 pic 域名。

版权所有：腾讯云计算（北京）有限责任公司

请求行



```
PUT /?guetzli HTTP/1.1
```

该 API 接口接受 PUT 请求。

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，请参阅 公共请求头部 章节。

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

该响应使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

特有响应头

该响应无特殊的响应头。

响应体

该响应体返回为空。

查询 Guetzli 状态

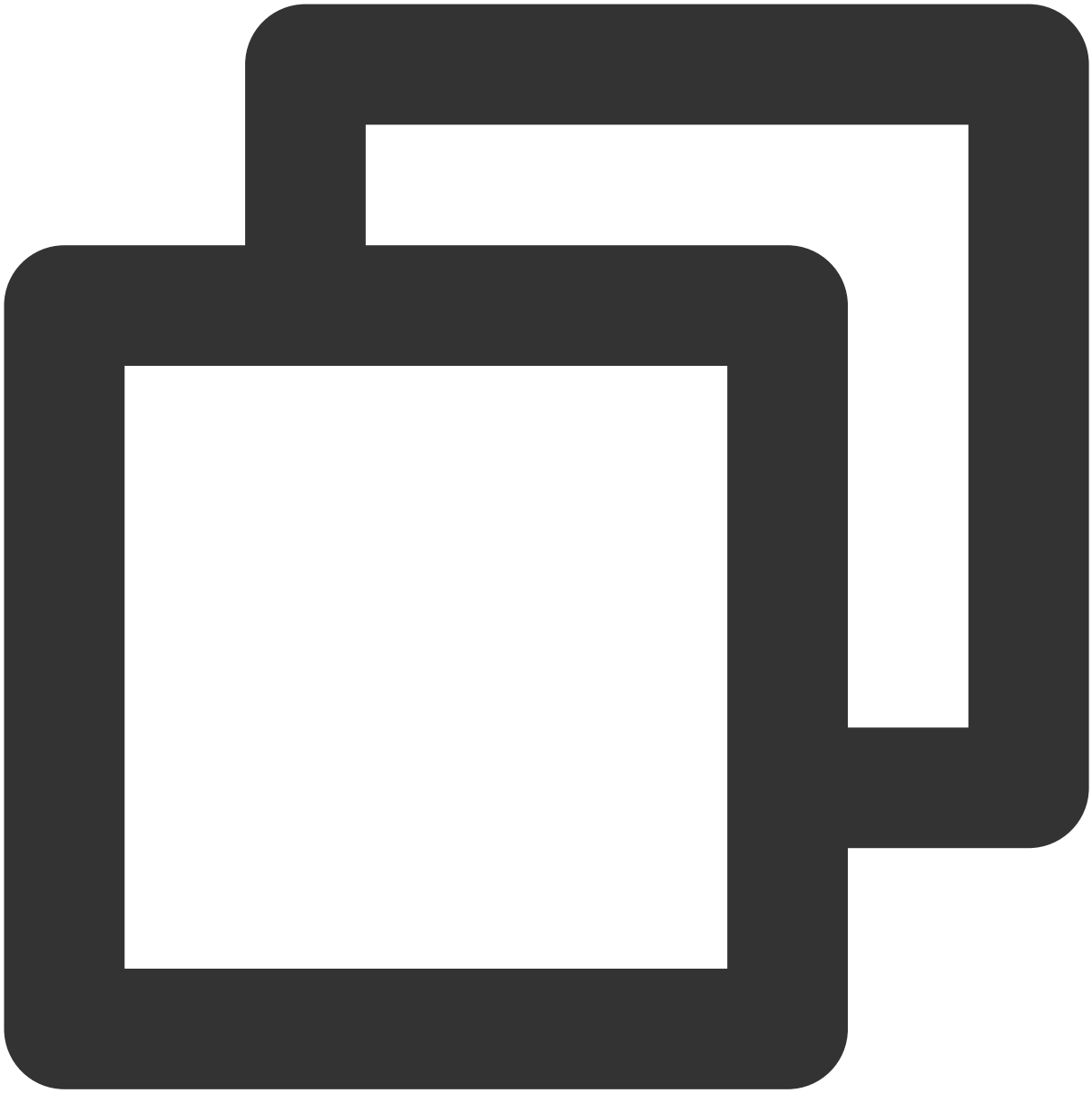
最近更新时间：2024-01-18 15:11:52

功能描述

用于查询 Guetzli 压缩功能是否开启。

请求

请求语法



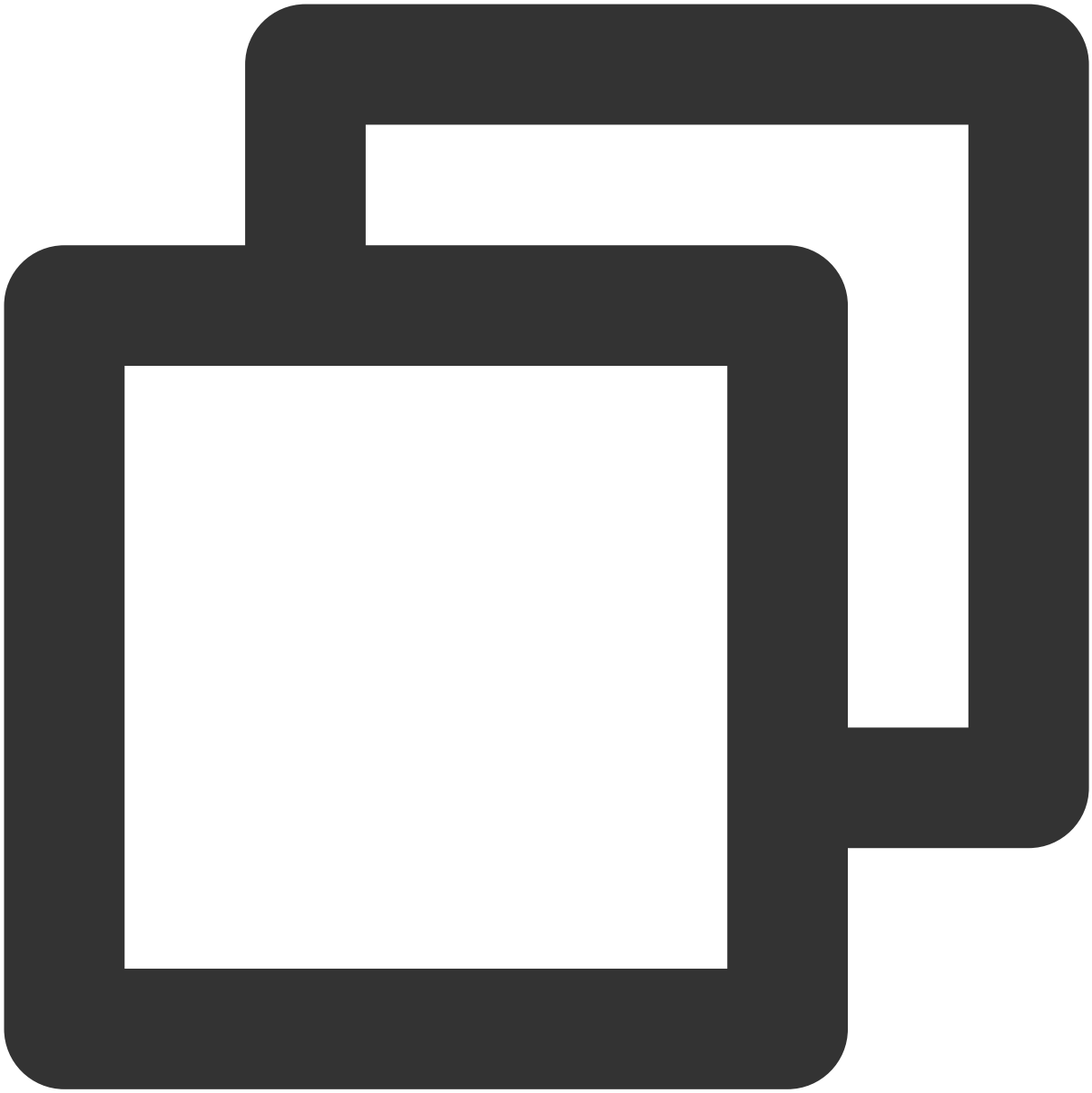
```
GET /?guetzli HTTP/1.1
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

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。

请求行



```
GET /?guetzli HTTP/1.1
```

该 API 接口接受 GET 请求。

请求头

公共头部

该请求操作的实现使用公共请求头，了解公共请求头详细请参见 公共请求头部 章节。

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

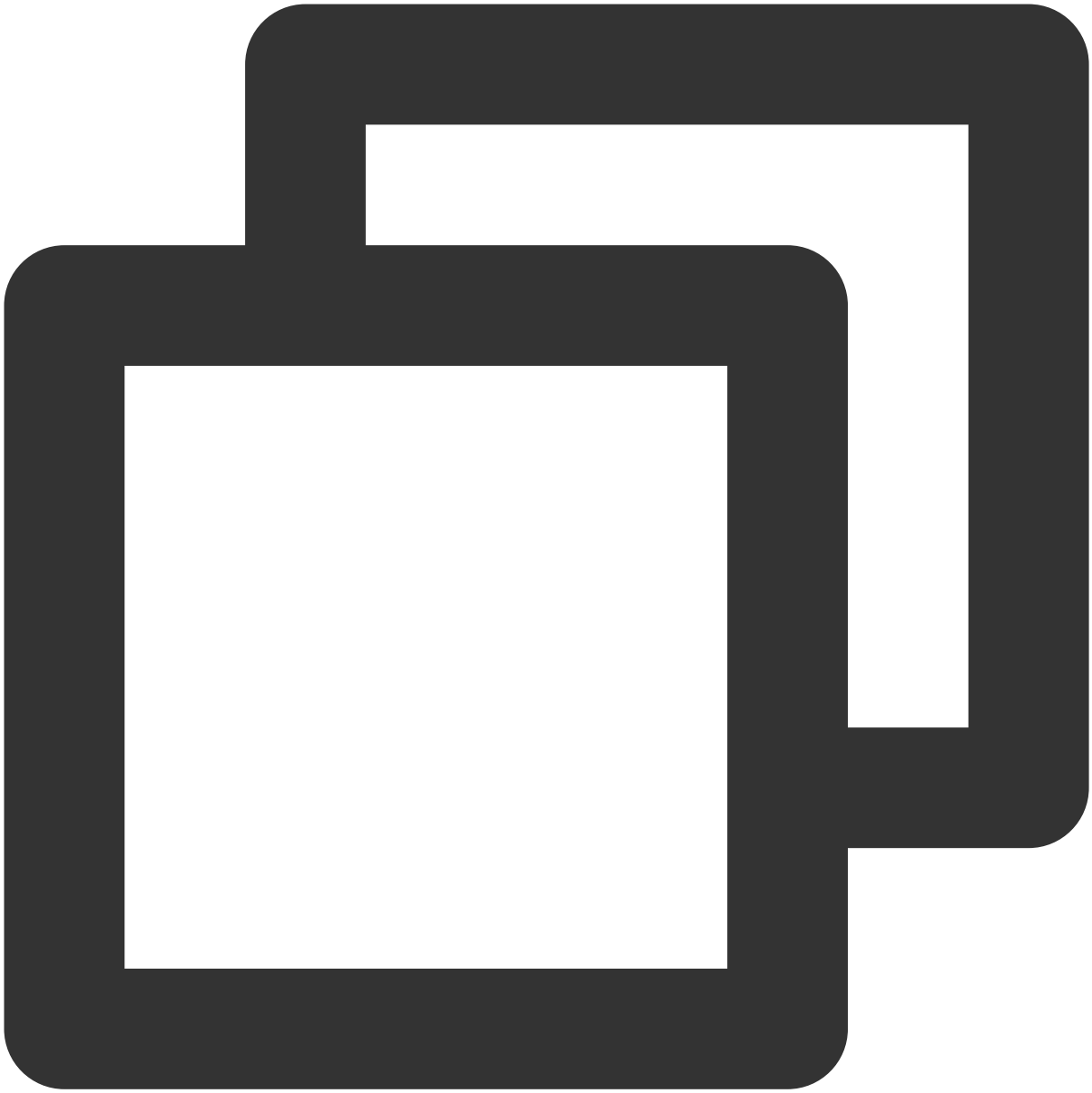
公共响应头

该响应使用公共响应头，了解公共响应头详细请参见 公共响应头部 章节。

特有响应头

该响应无特殊的响应头。

响应体



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

拥有 `on` 与 `off` 两种状态。

关闭 Guetzli 压缩

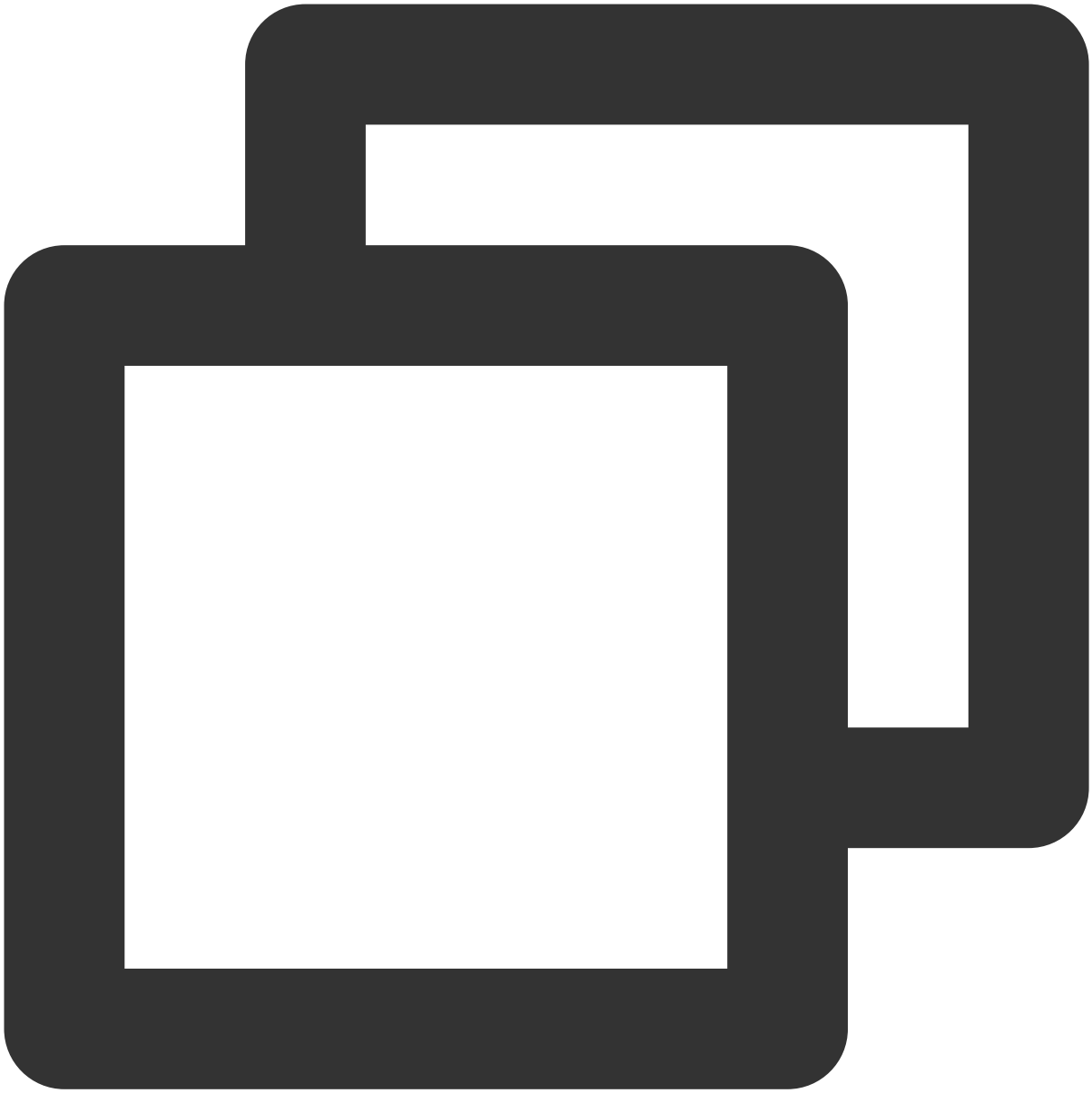
最近更新时间：2024-01-18 15:11:52

功能描述

用于关闭 Guetzli 压缩功能。

请求

请求语法



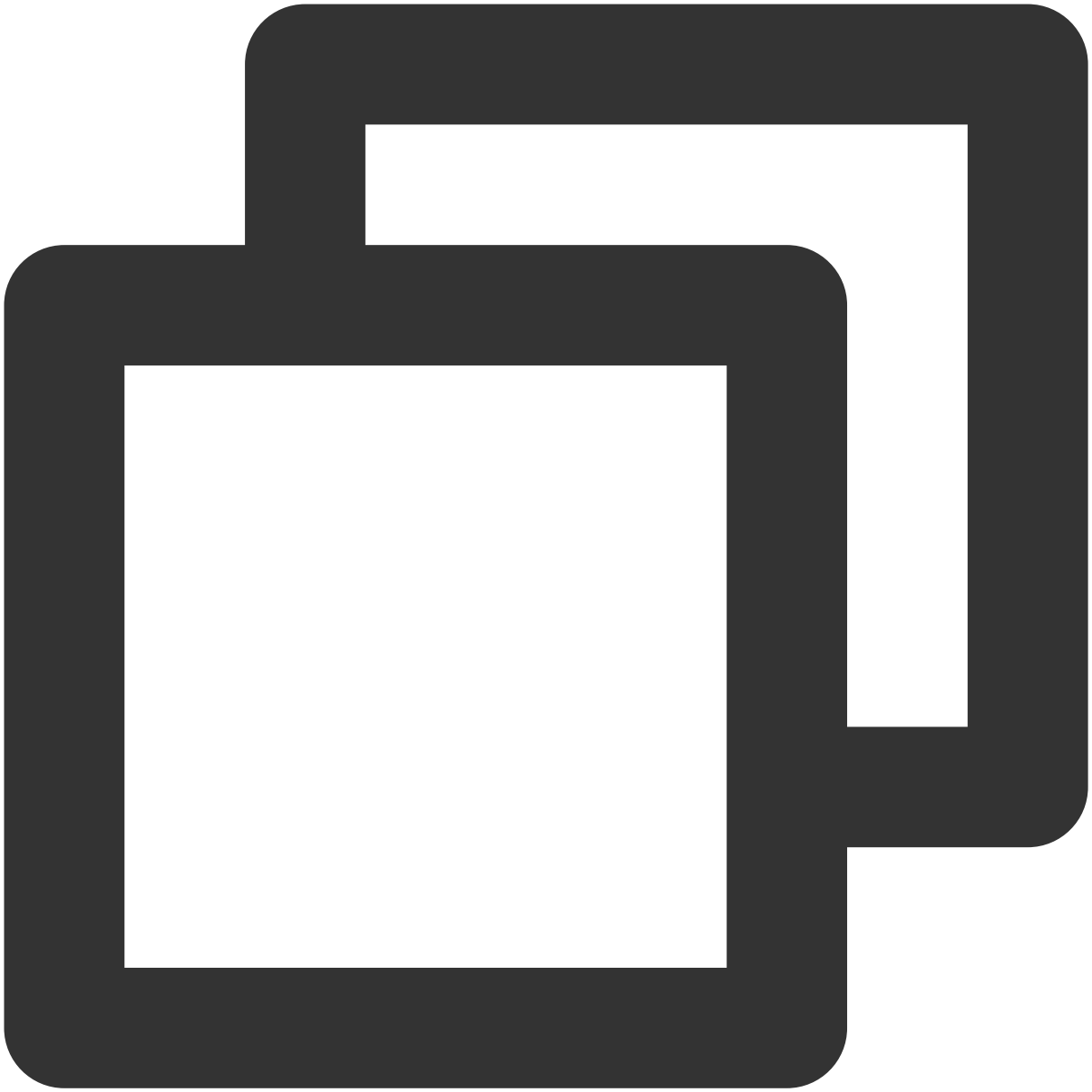
```
DELETE /?guetzli HTTP/1.1
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

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。

请求行



```
DELETE /?guetzli HTTP/1.1
```

该 API 接口接受 DELETE 请求。

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，请参阅 公共请求头部 章节。

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

该响应使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

特有响应头

该响应无特殊的响应头。

响应体

该响应体返回为空。

WebP 压缩

最近更新时间：2024-01-18 15:11:52

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。
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。

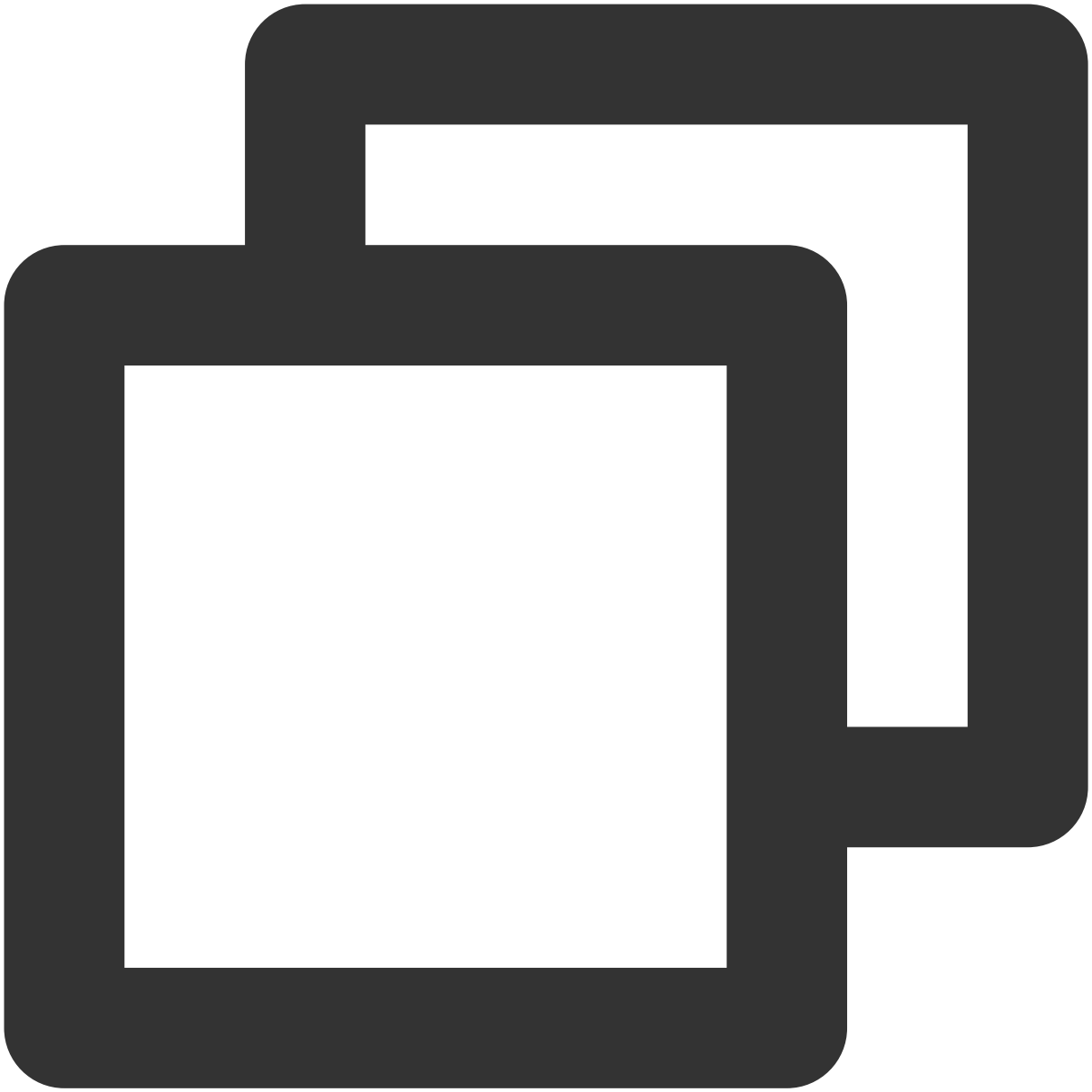
使用方式

数据万象通过 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，具体费用请参见 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

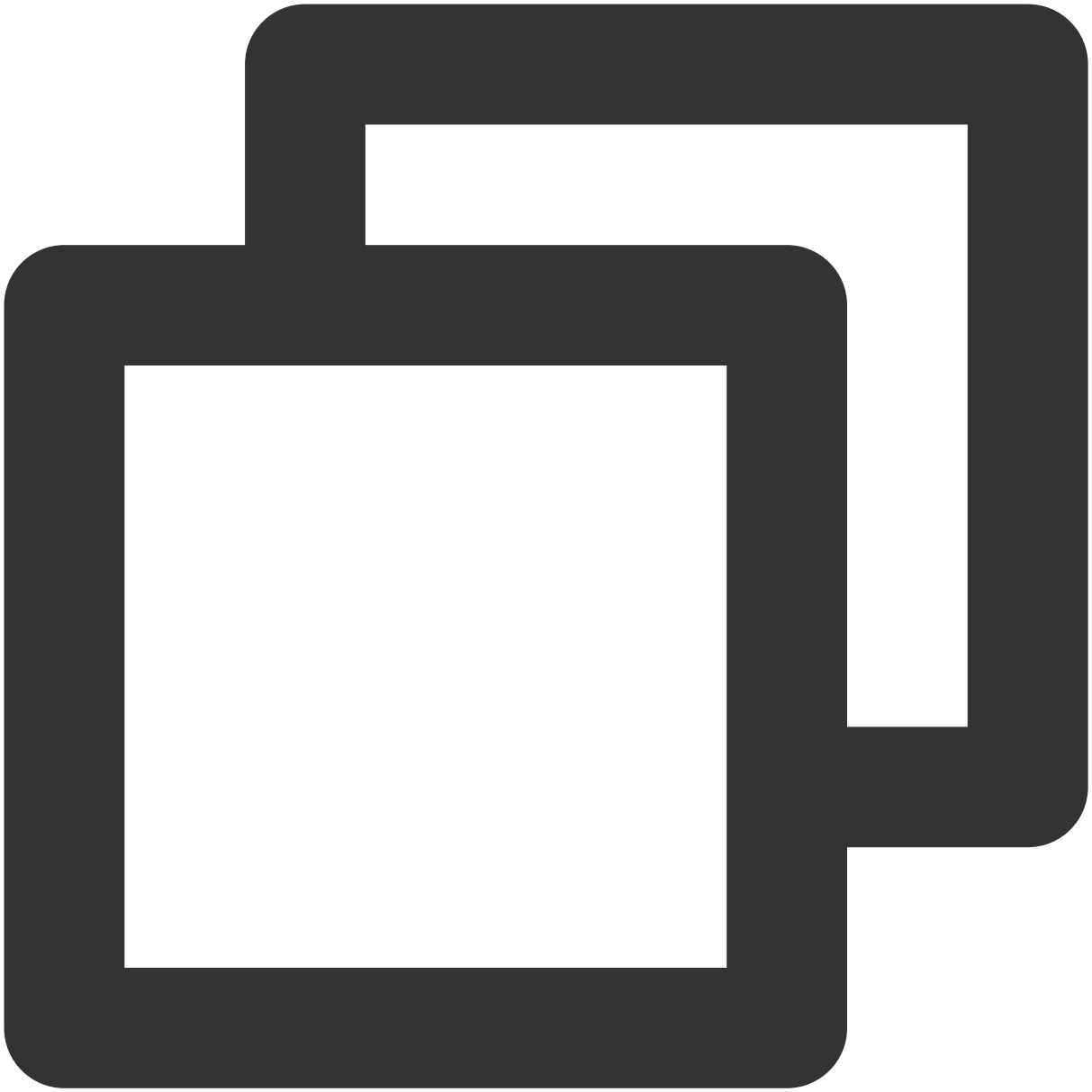
1. 下载时处理

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?imageMogr2/format/webp HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

2. 上传时处理



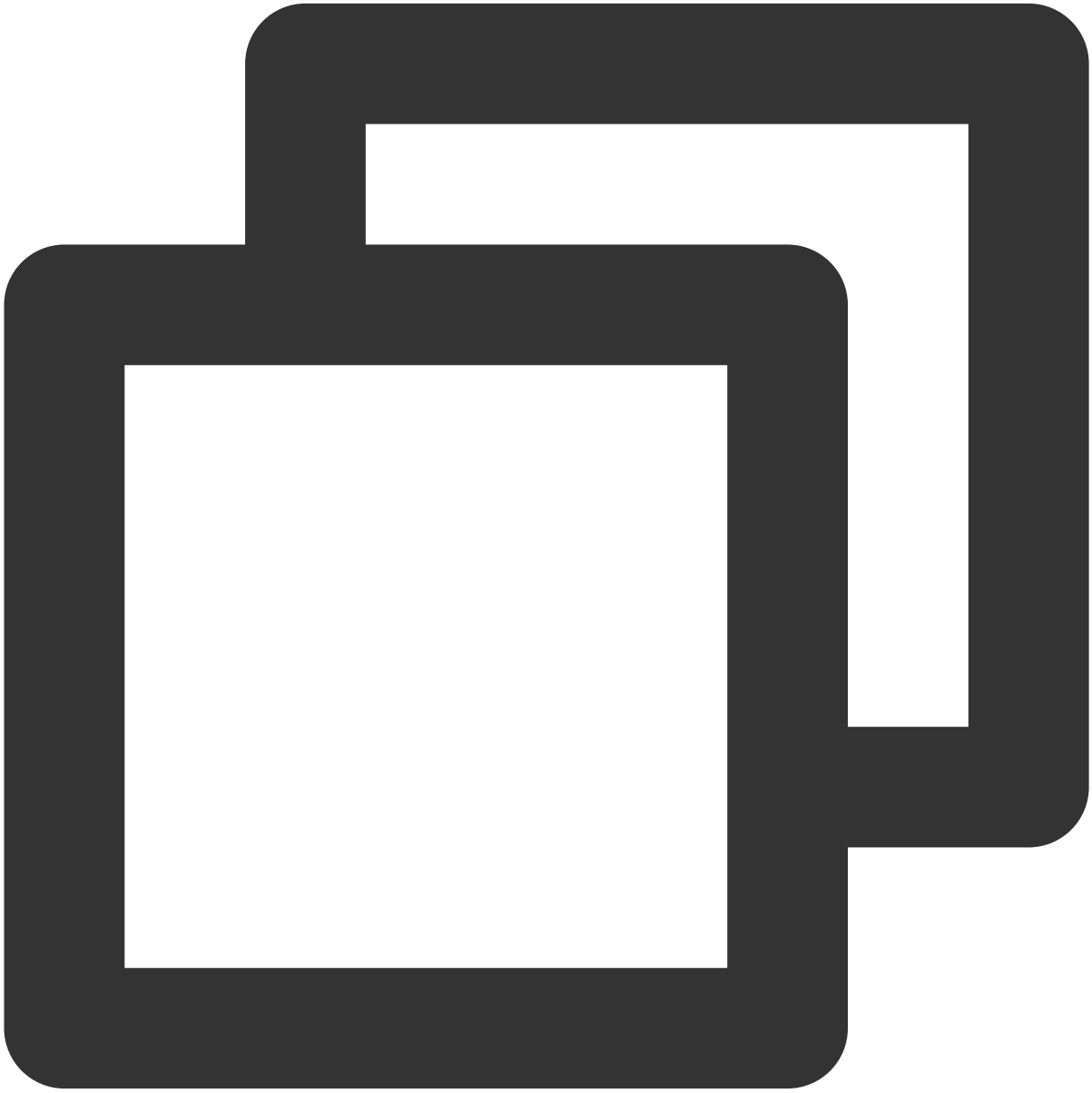
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/webp"
}]
```

```
}
```

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

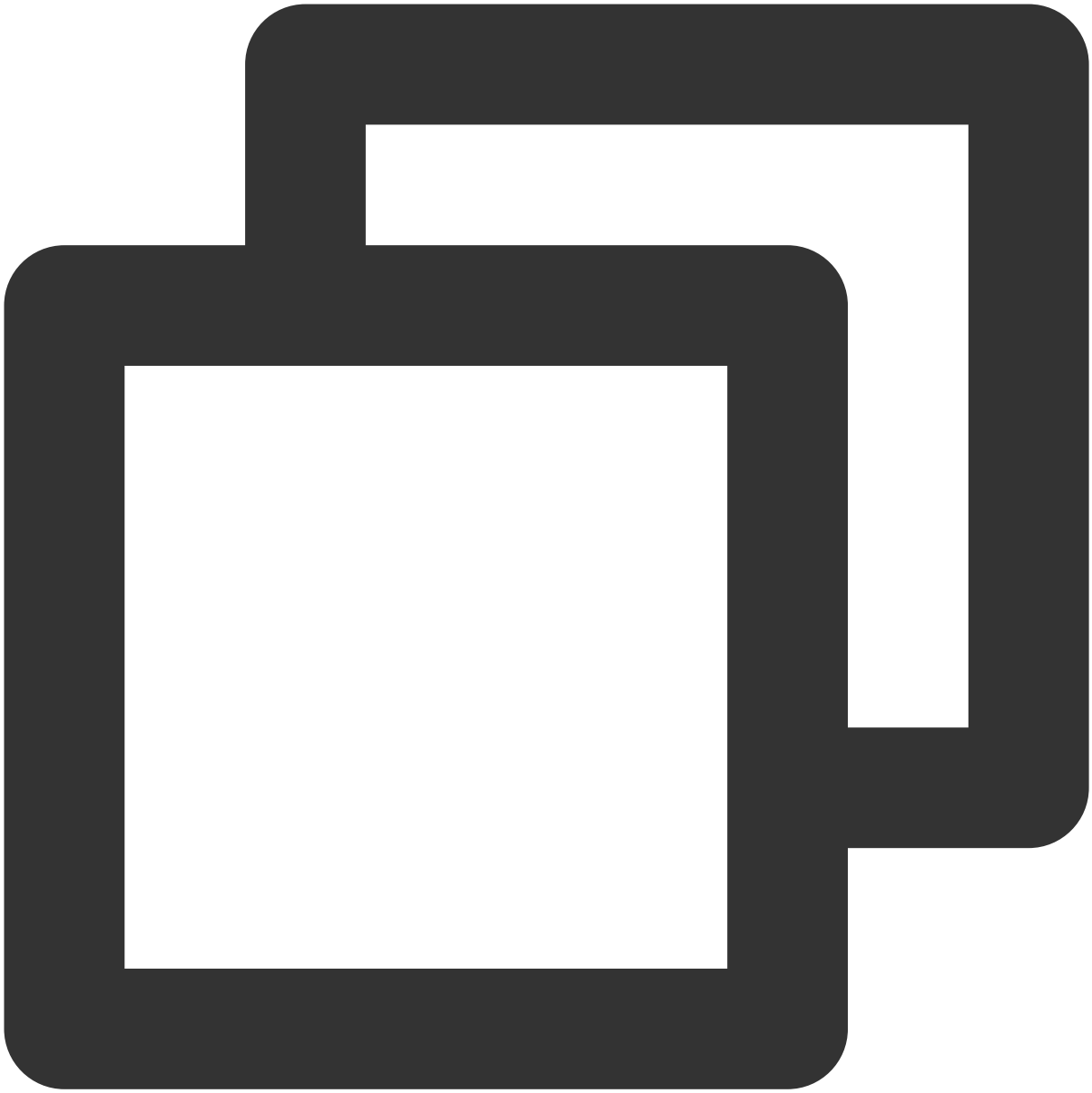
说明：

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：



```
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%）

HEIF 压缩

最近更新时间：2024-01-18 15:11:52

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。
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

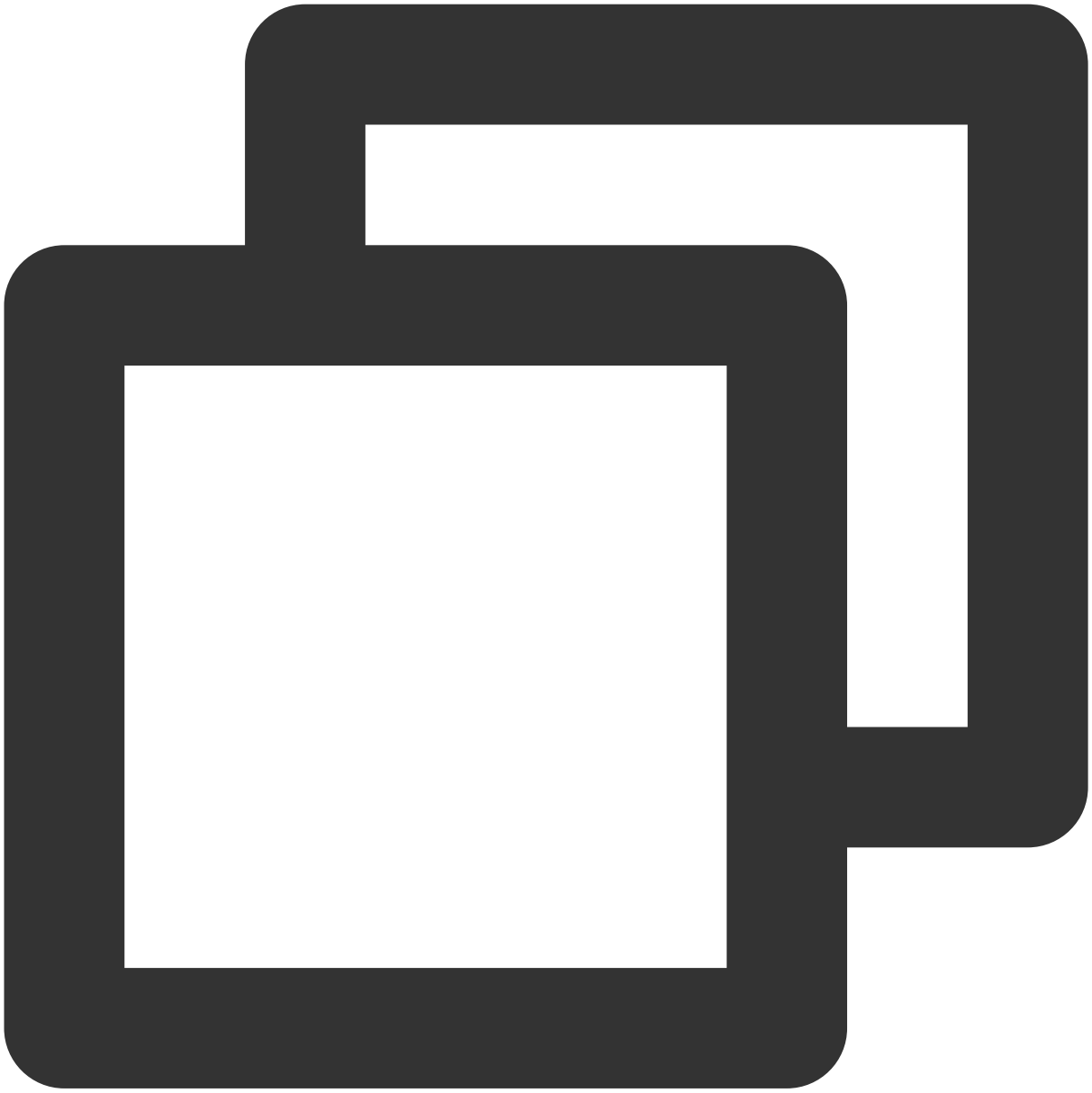
使用 HEIF 压缩，需要先开通存储桶的图片高级压缩功能，您需要在相应的存储桶配置页中通过开关按钮开启功能，详情请参见 设置图片高级压缩。

使用方式

数据万象通过 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，具体费用请参见图片处理费用。

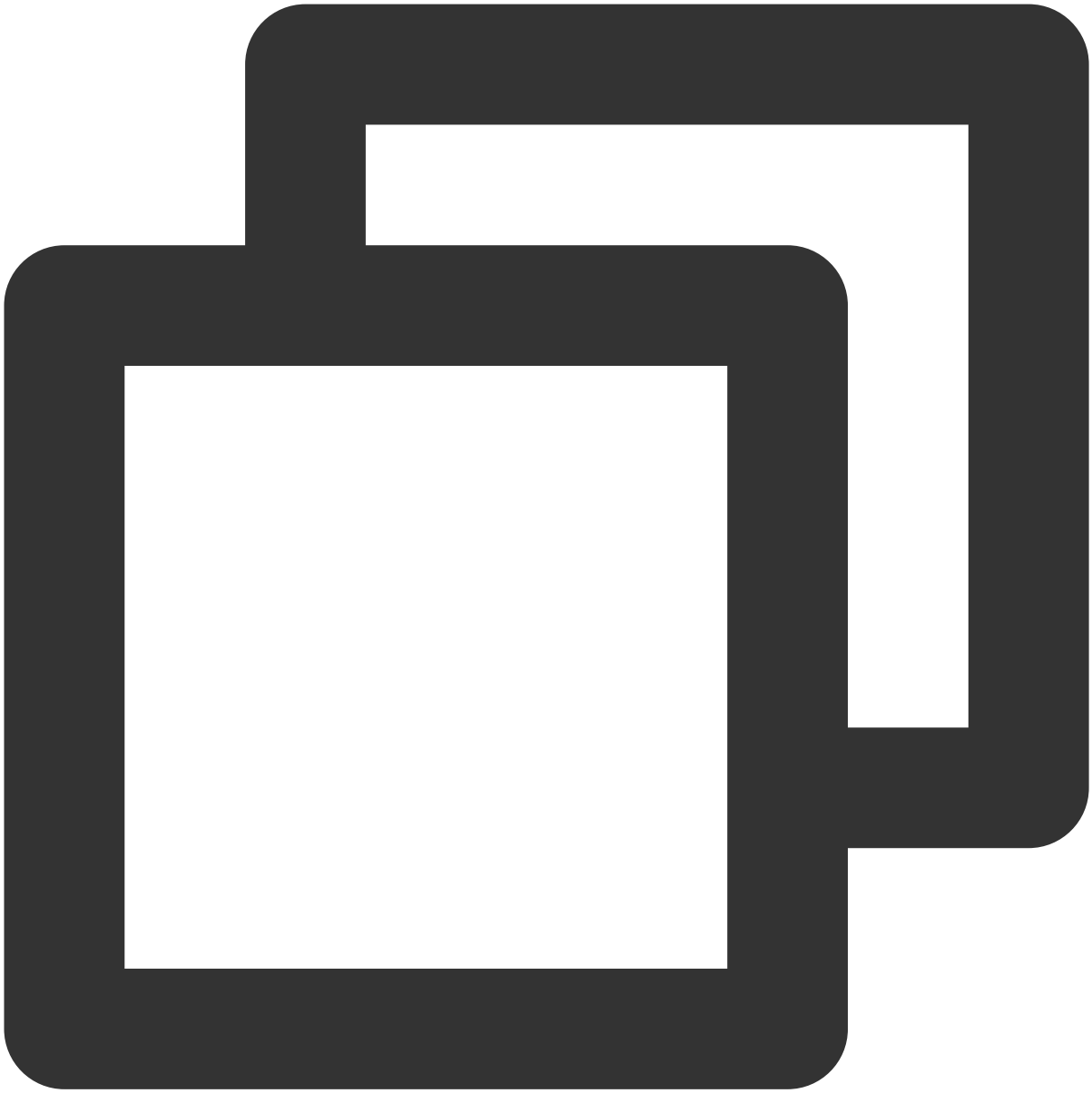
接口示例

1. 下载时处理



```
download_url?imageMogr2/format/heif
```

2. 上传时处理



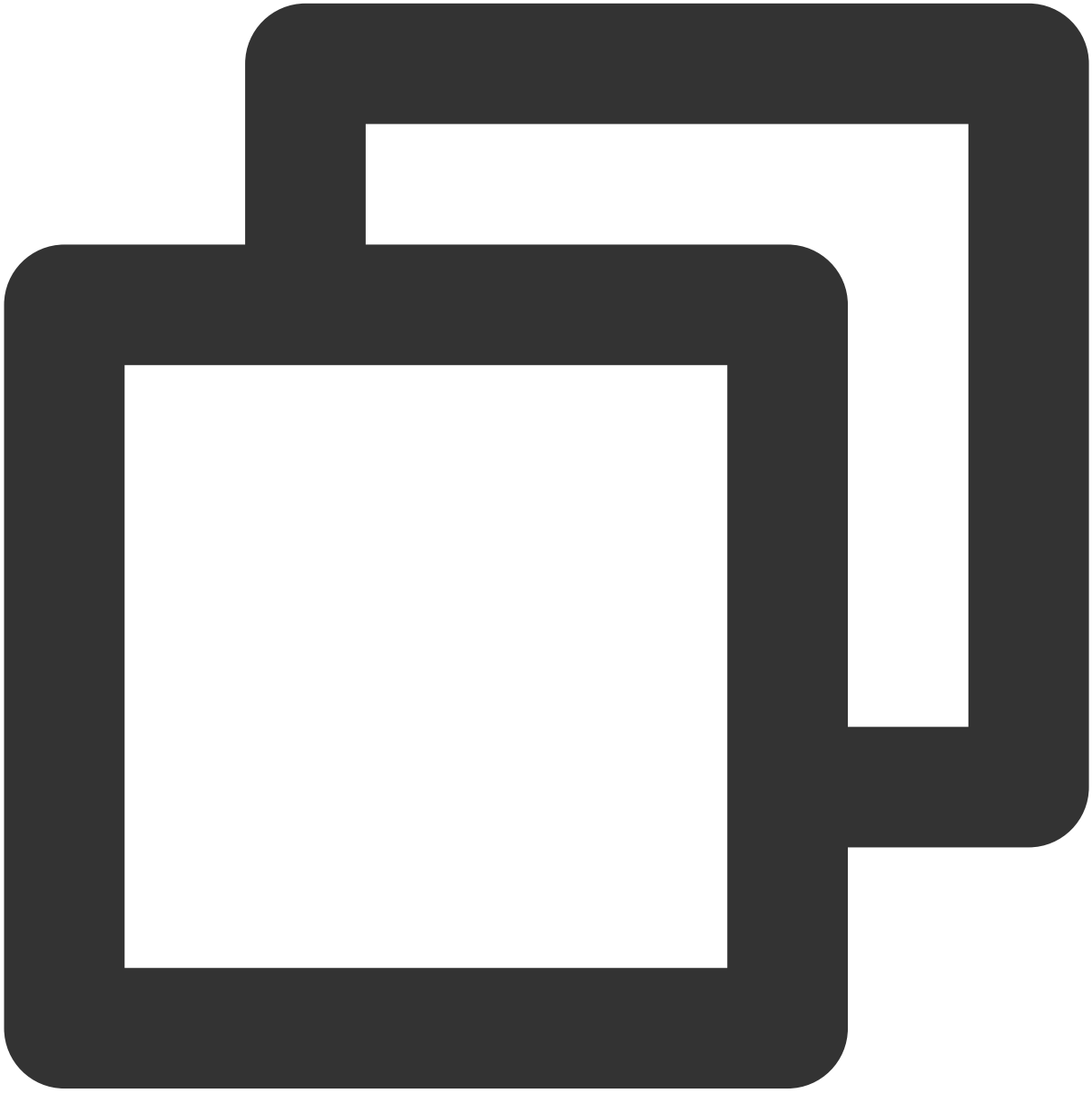
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/heif"
}]
```

```
}
```

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/heif"
}]
}
```

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 heif。

实际案例

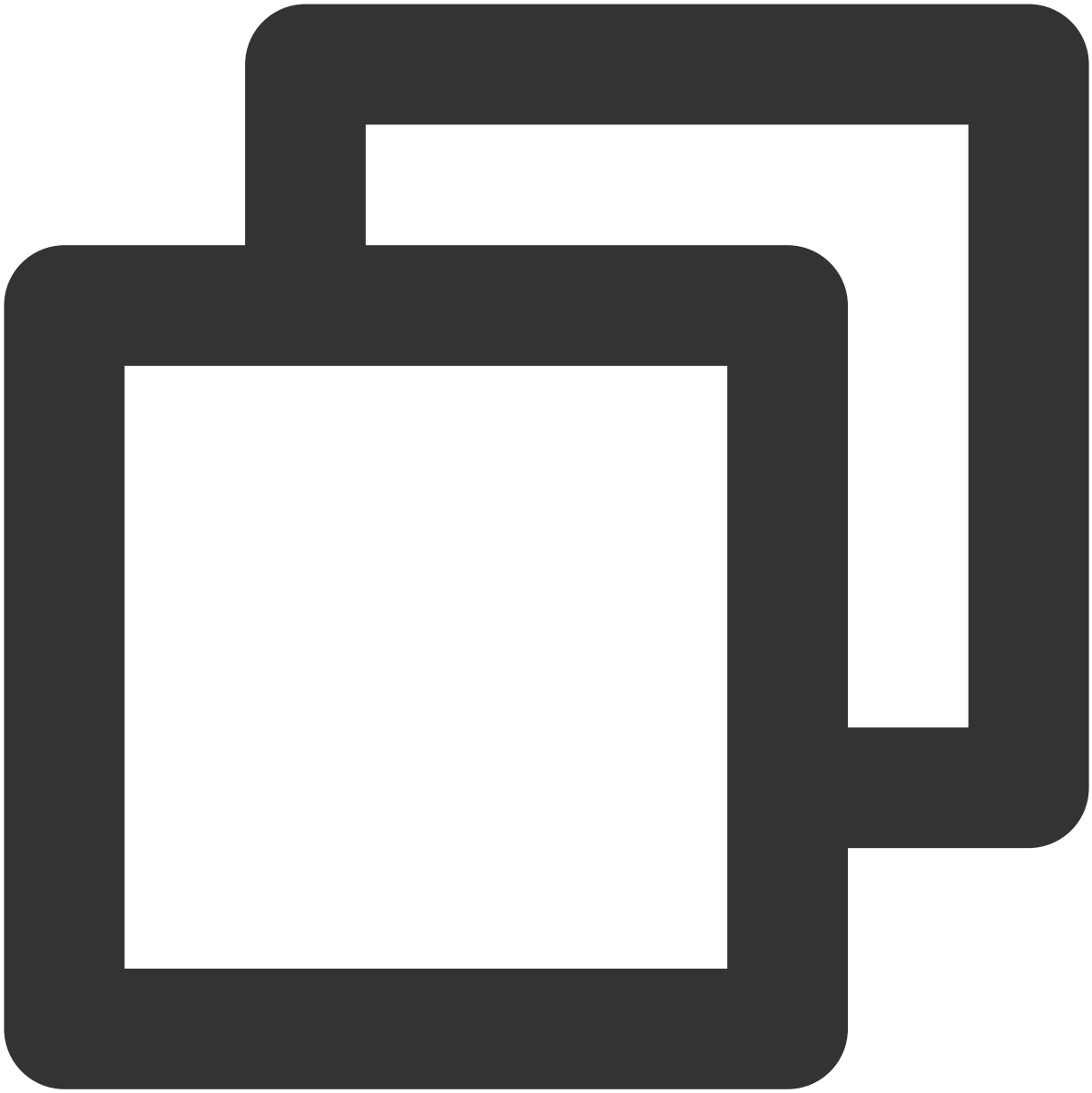
说明：

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 heif 格式，请求 URL 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
heif	52.87KB（压缩率96.0%）

TPG 压缩

最近更新时间：2024-01-18 15:11:52

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。
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。

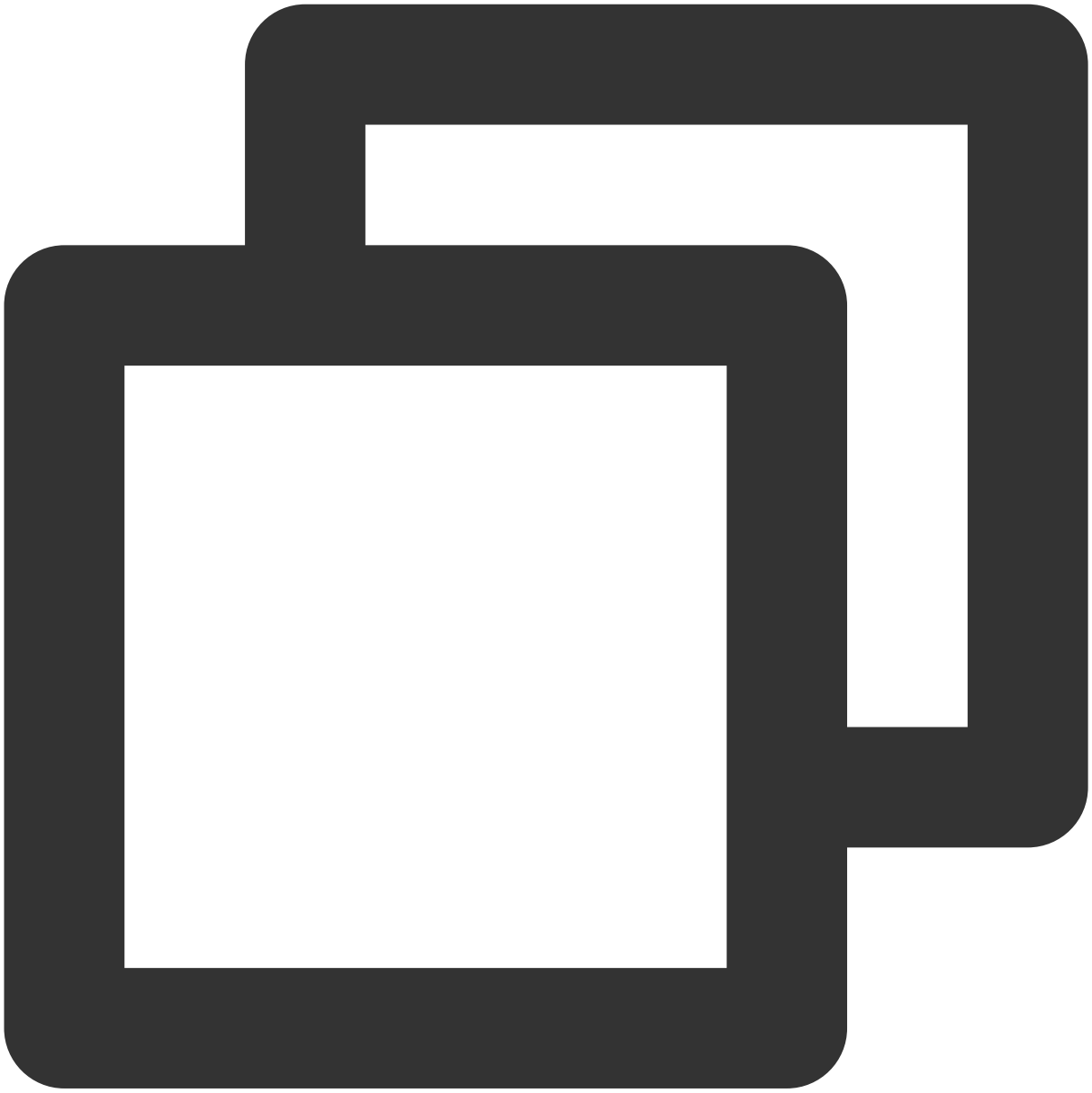
使用方式

数据万象通过 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

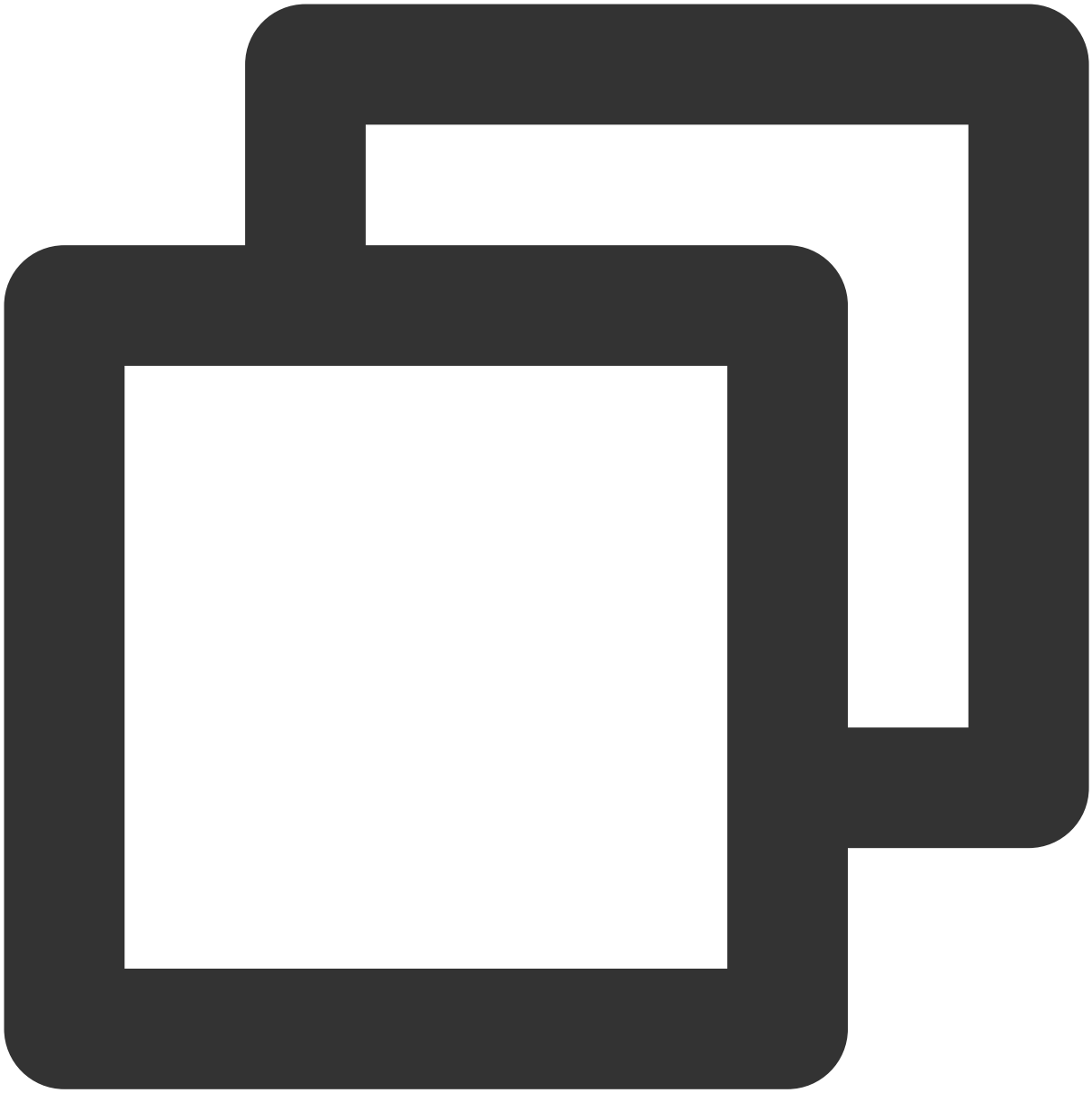
接口示例

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



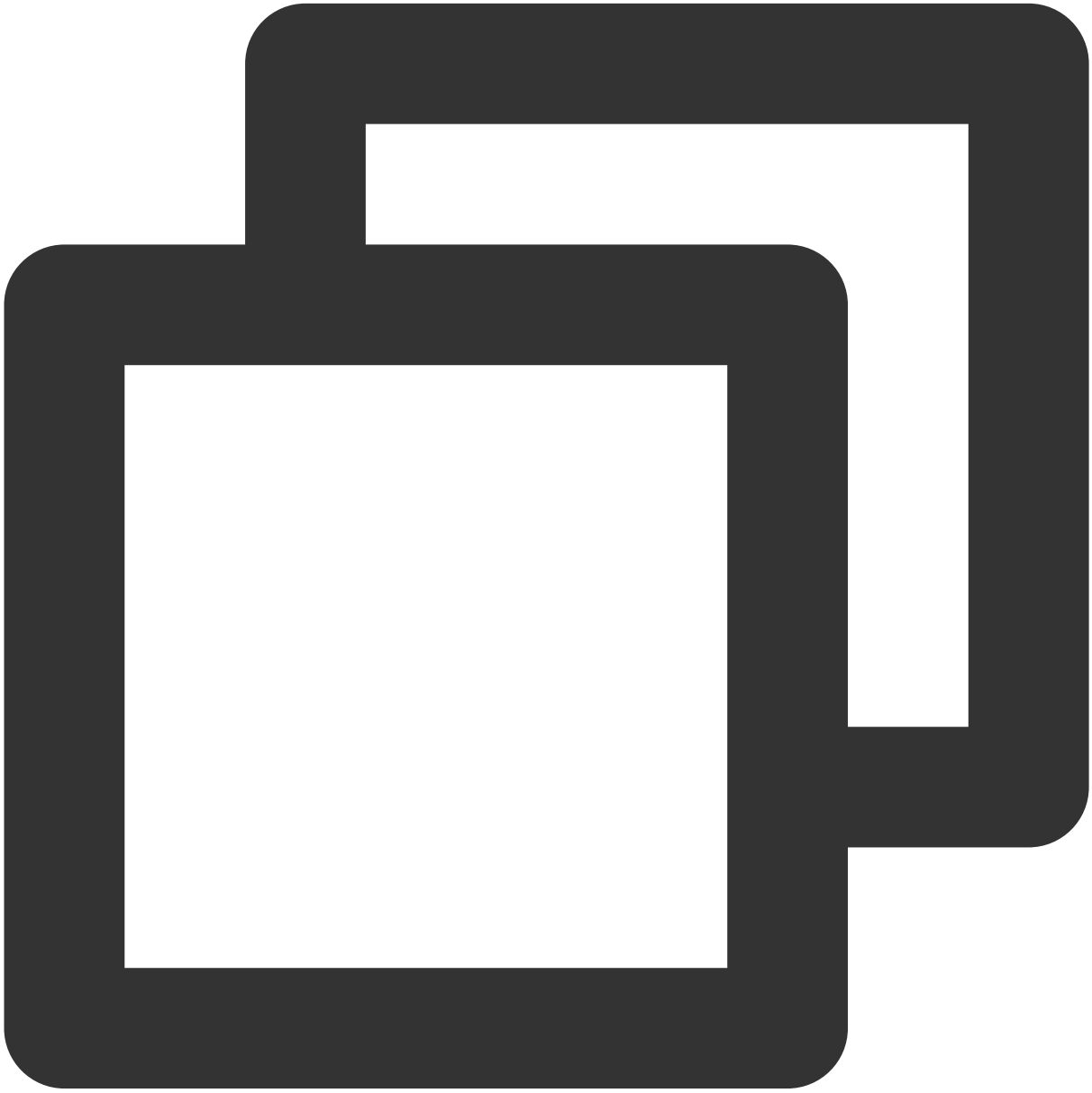
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/tpg"
}]
```

```
}
```

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
ObjectKey	对象文件名，例如 folder/sample.jpg。
/format/<Format>	压缩格式，此处为 tpg。

实际案例

说明

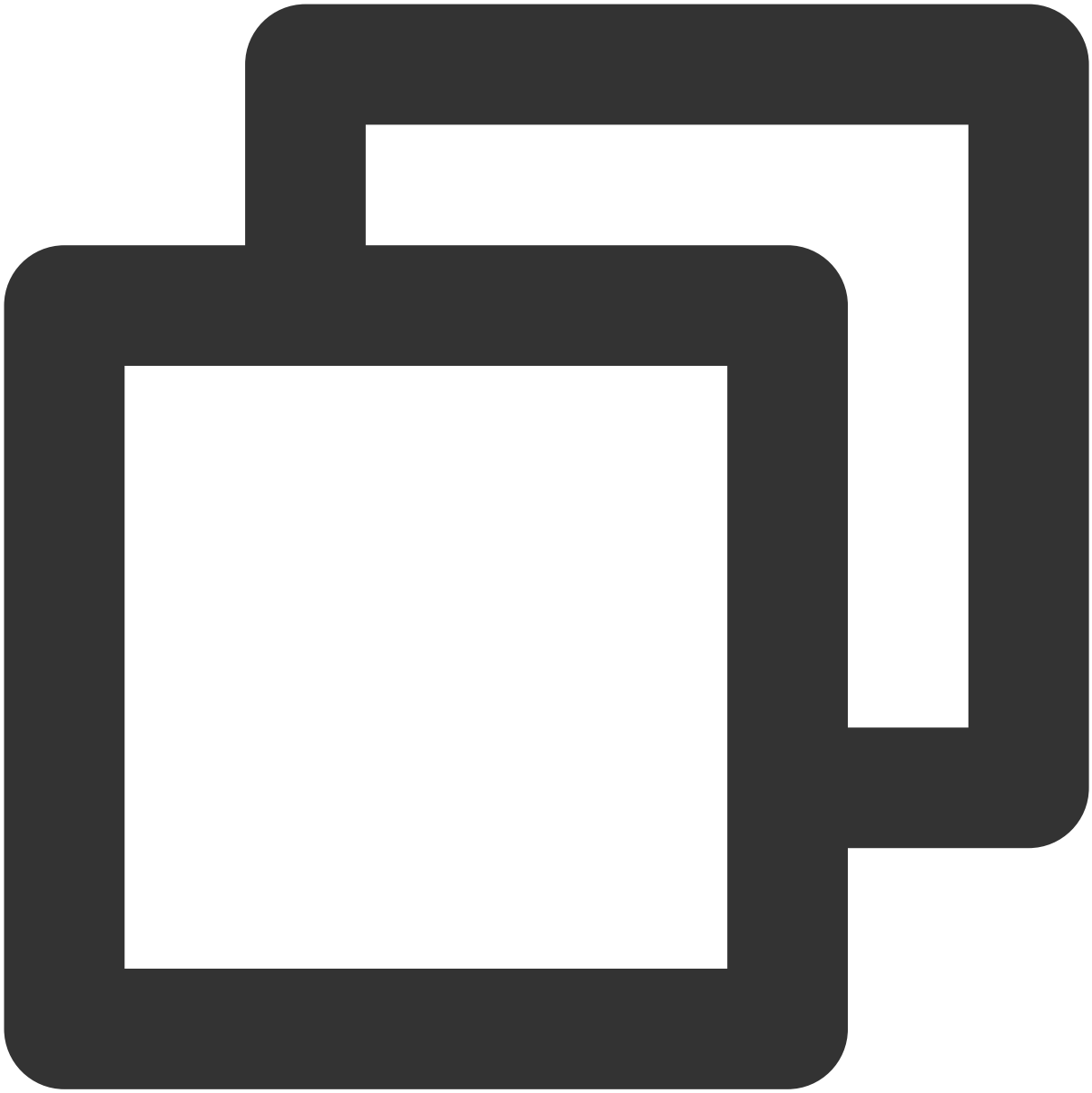
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，请使用**上传时处理**或**云上数据处理**方式。

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%）

AVIF 压缩

最近更新时间：2024-01-18 15:11:52

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。
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

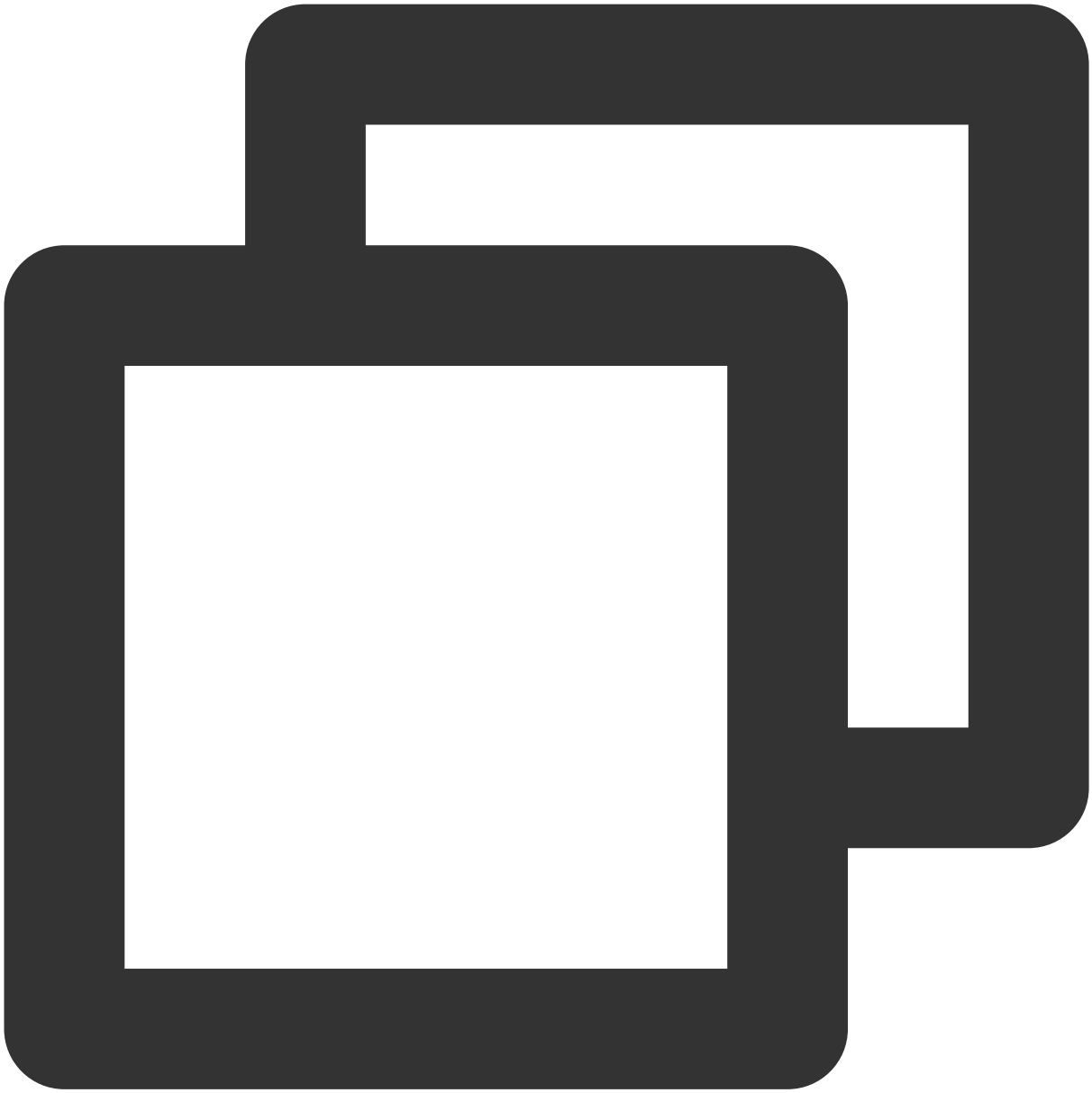
使用 AVIF 压缩，需要先开通存储桶的图片高级压缩功能，您需要在相应的存储桶配置页中通过开关按钮开启功能，详情请参见 图片高级压缩。

使用方式

数据万象通过 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，具体费用请参见 图片处理费用。

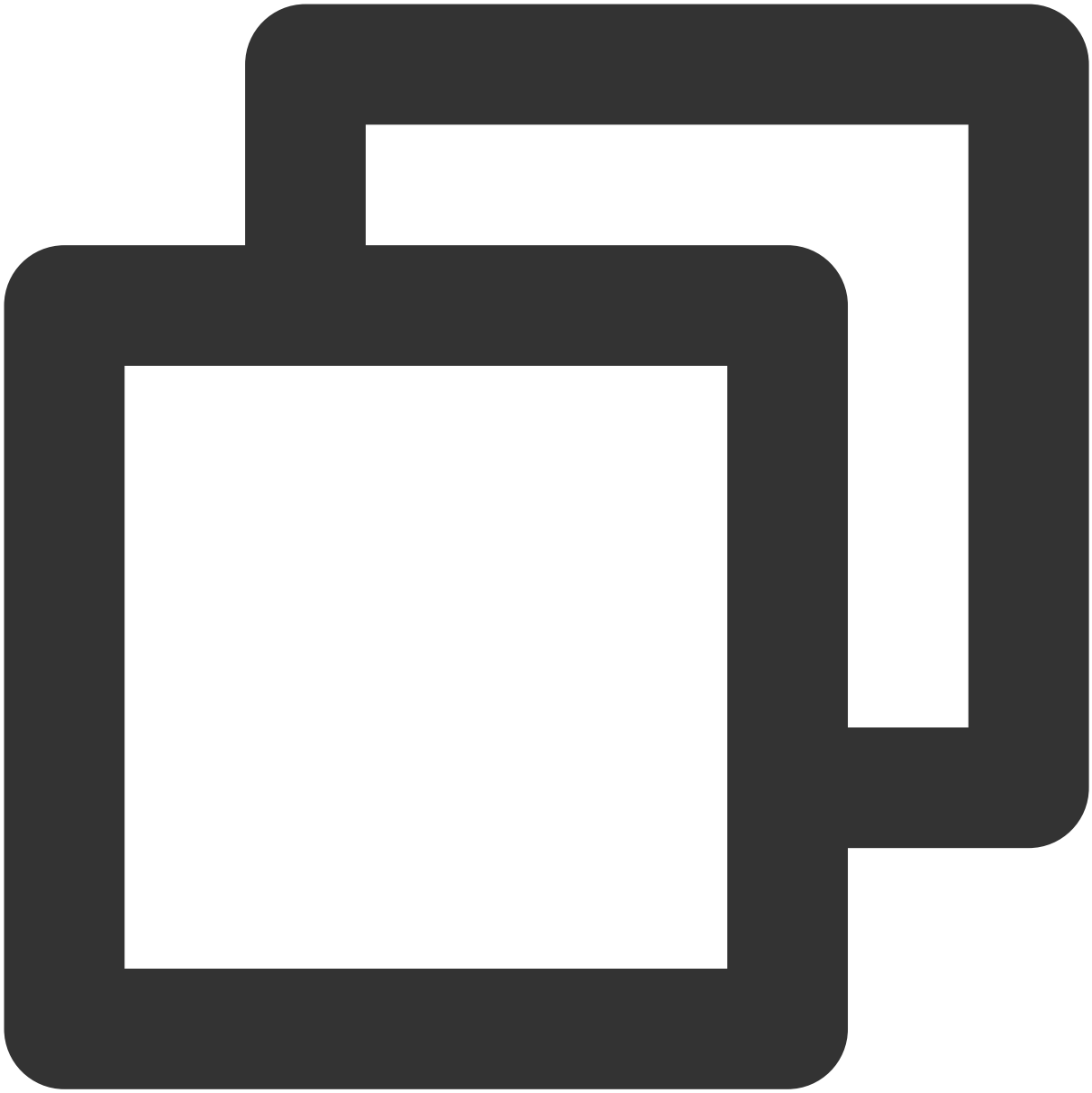
接口示例

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



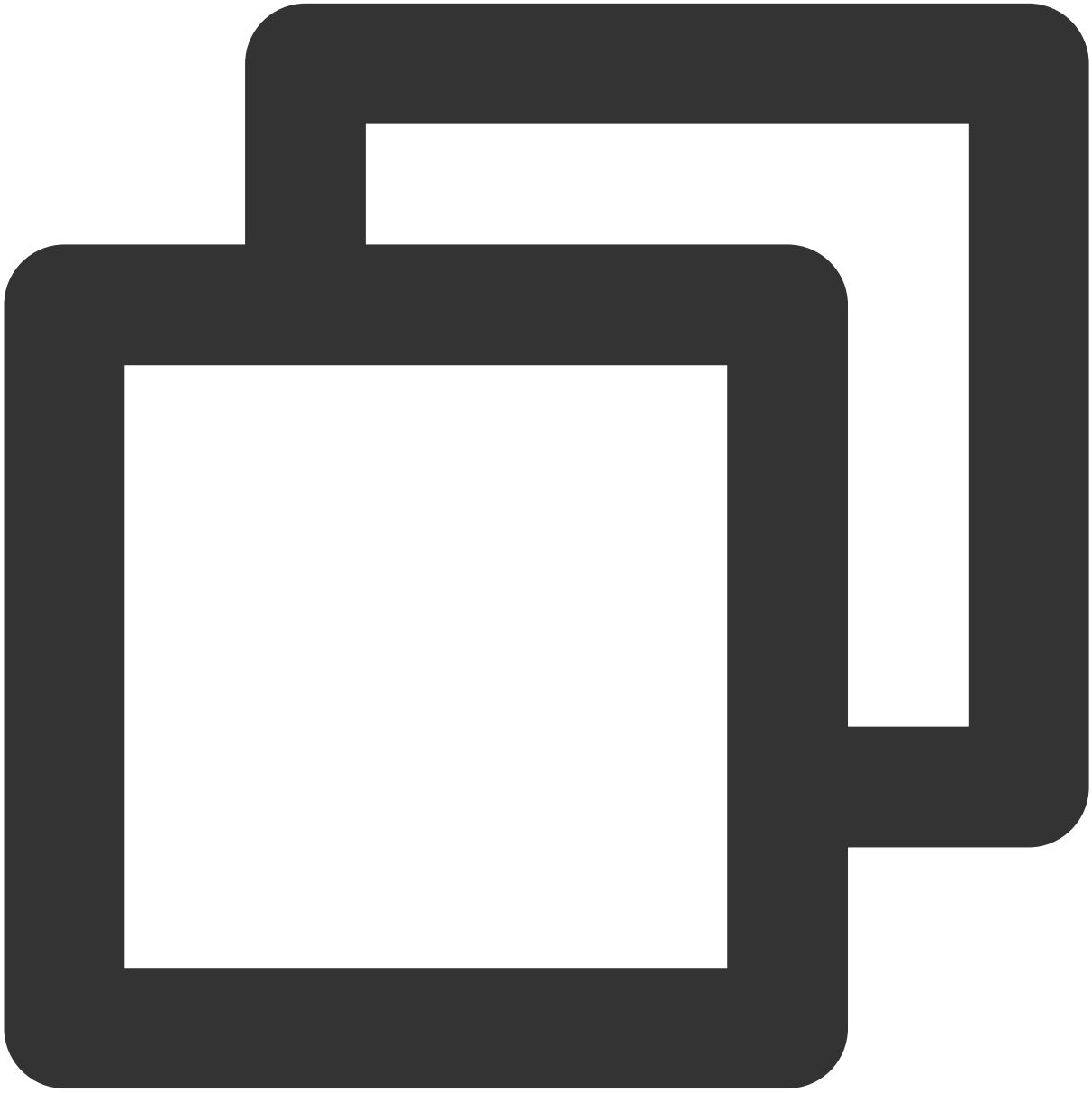
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/avif"
}]
```

```
}
```

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：

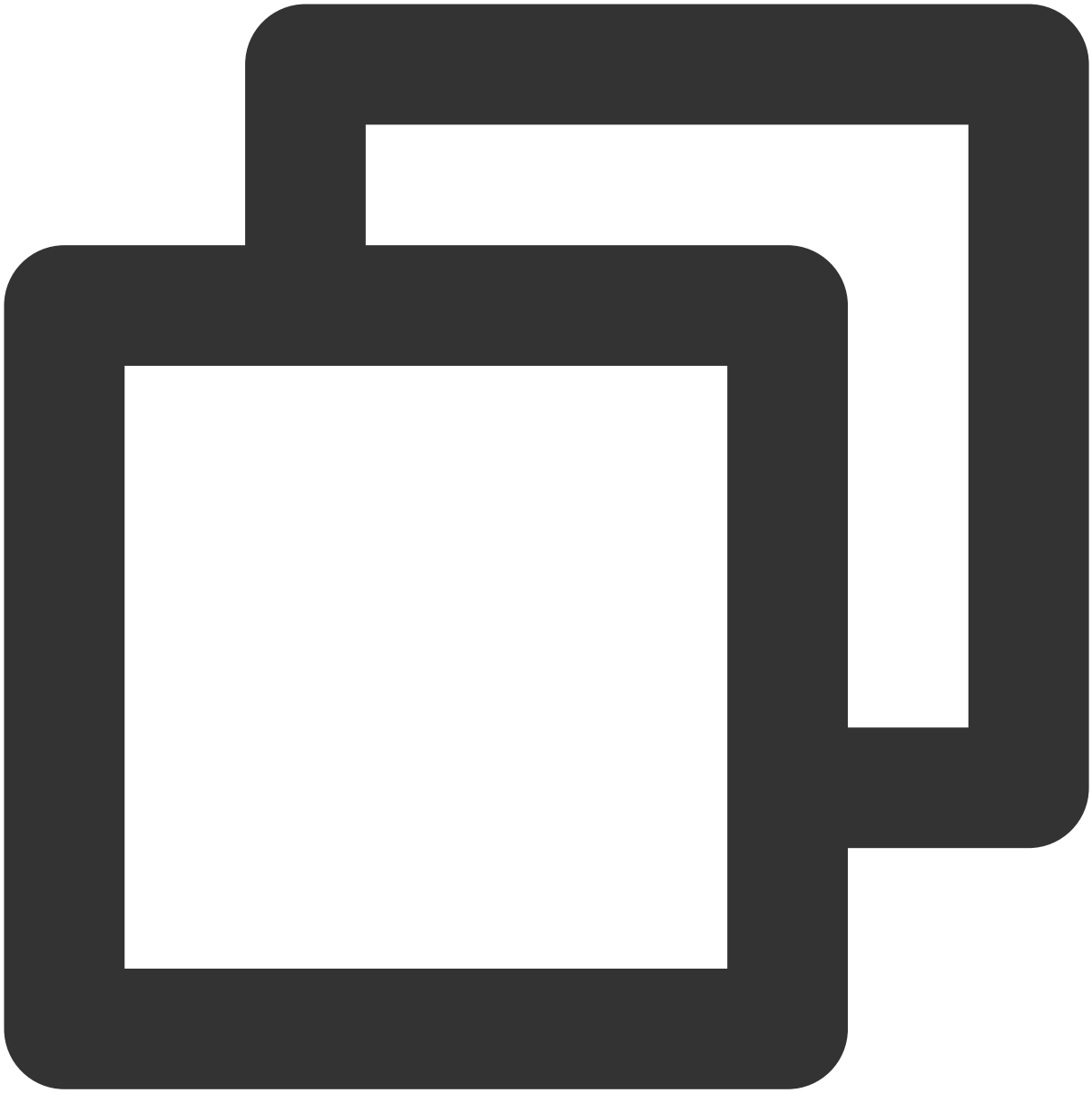
本篇文档中的实际案例仅包含**下载时处理**，该类处理不会保存处理后的图片至存储桶。如有保存需求，请使用**上传时处理**或**云上数据处理**方式。

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%）

高级压缩

最近更新时间：2024-01-18 15:11:52

功能概述

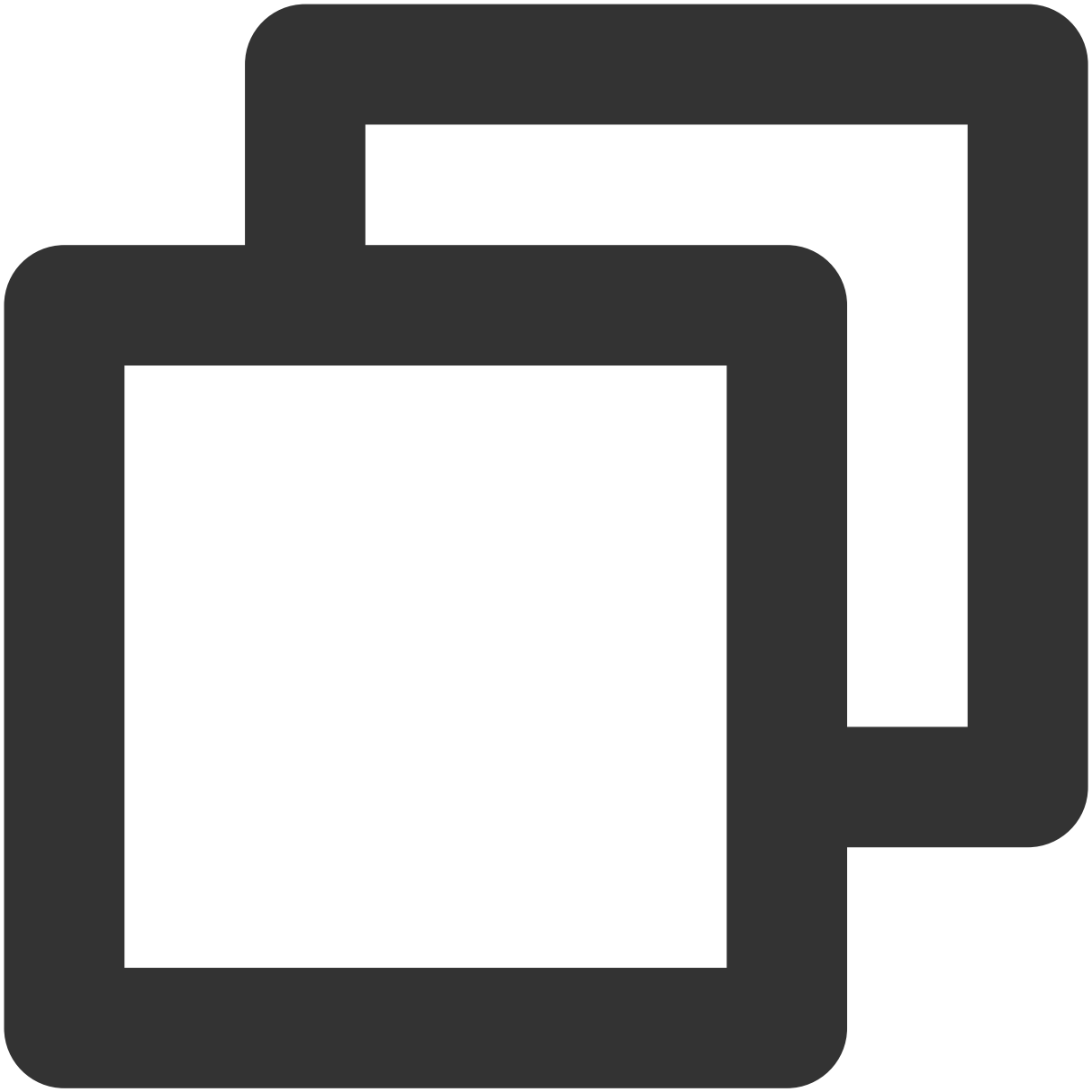
数据万象提供图片高级压缩功能，可以更加高效地将图片转换为 TPG 或 HEIF 等高压缩比格式，有效降低图片传输链路及加载耗时，降低带宽及流量成本。
该功能支持以下的处理方式：
下载时处理
上传时处理
云上数据处理

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 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，具体费用可查看 计费与定价。

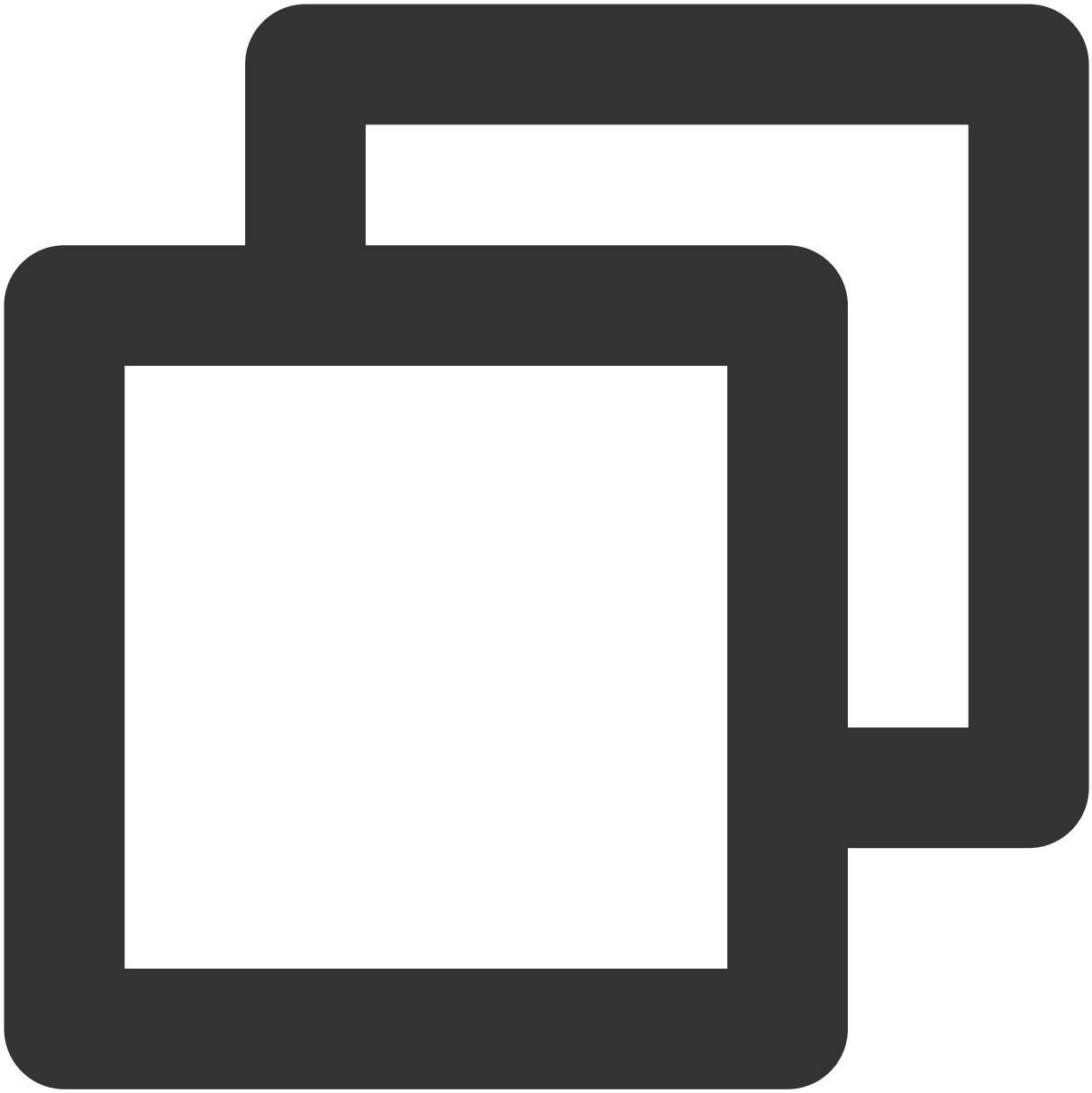
接口示例

1. 下载时处理



```
download_url?imageMogr2/format/<Format>
```

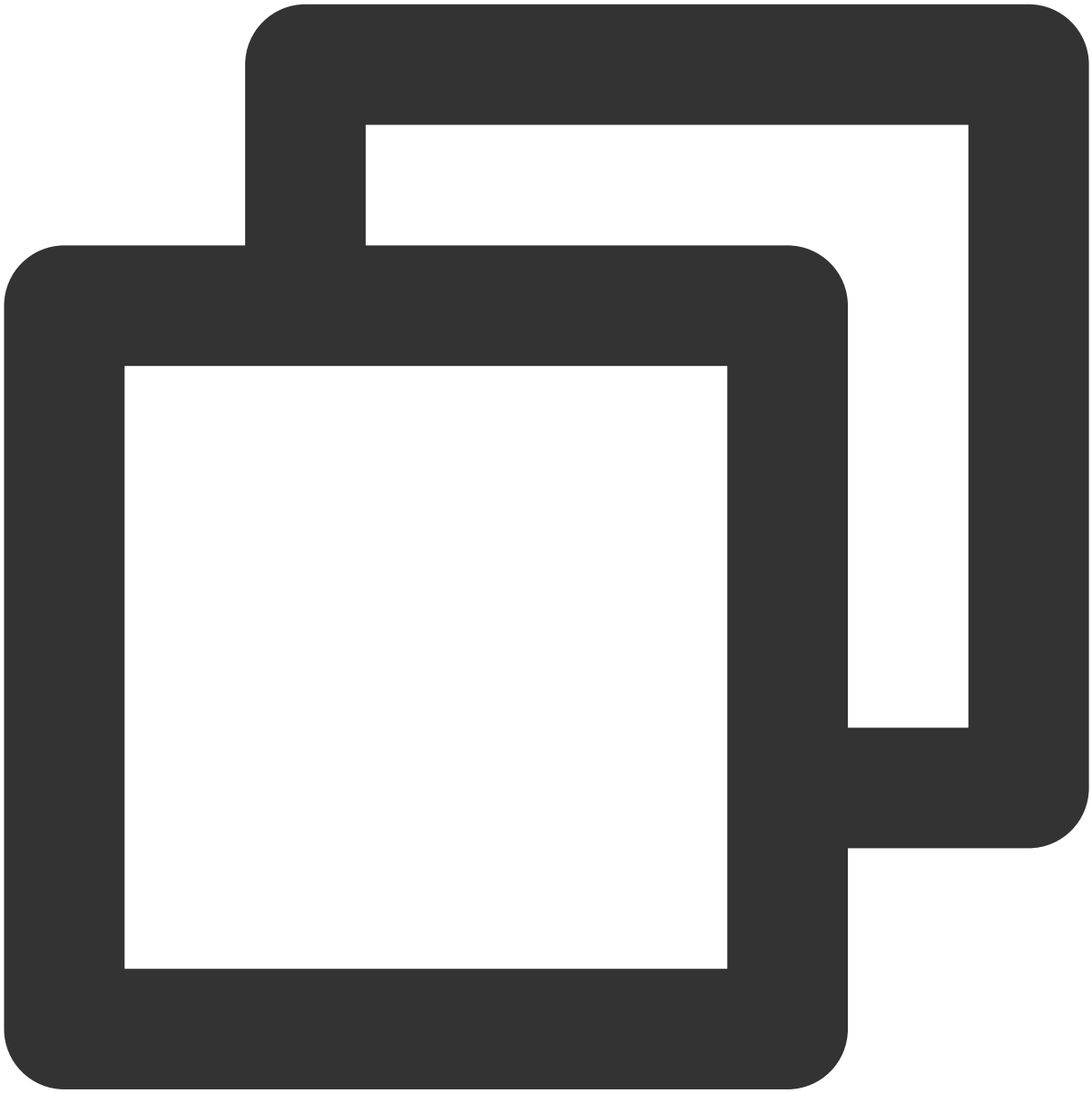
2. 上传时处理



```
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。

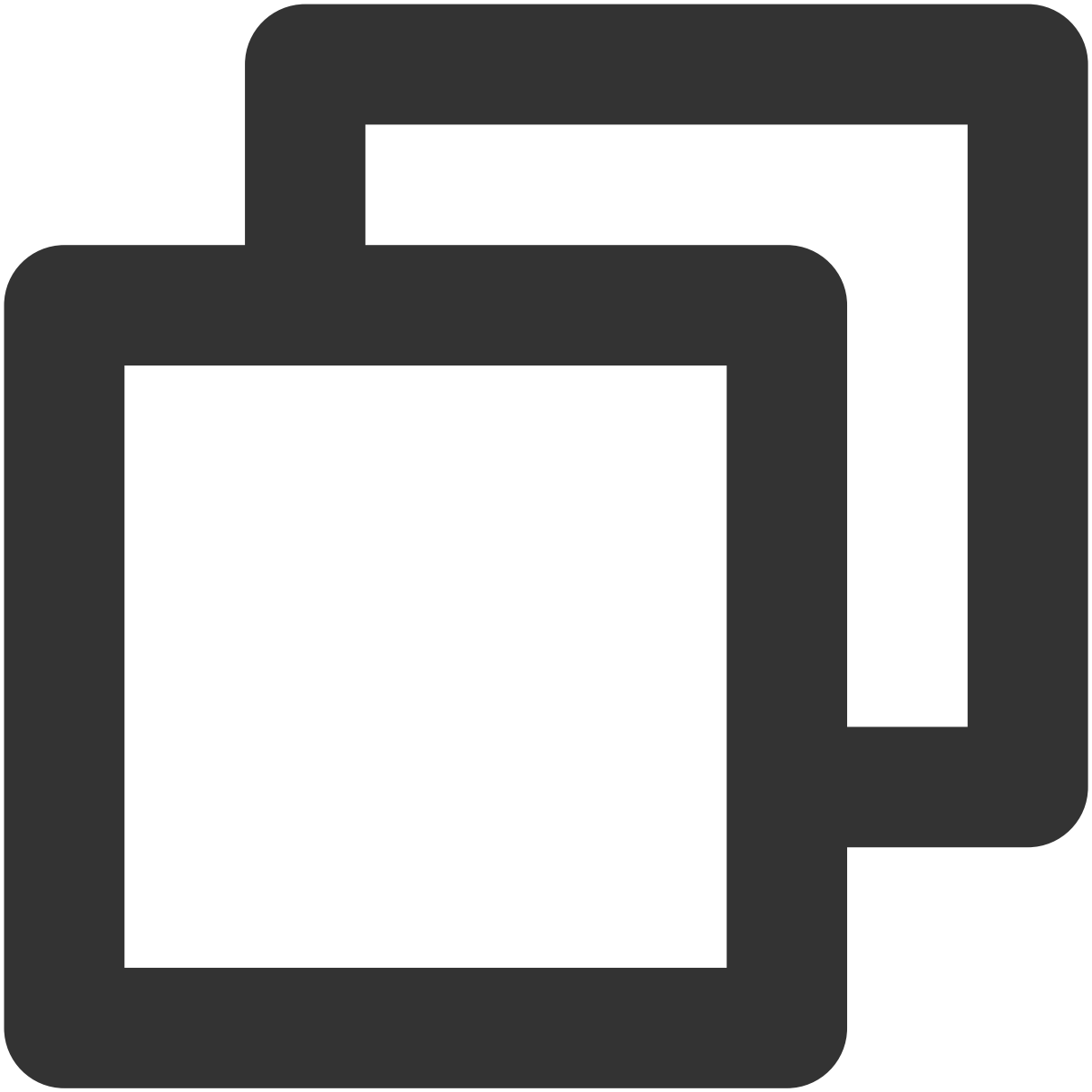
实际案例

假设原图格式为 PNG，图片大小为1335.2KB，如下图所示。



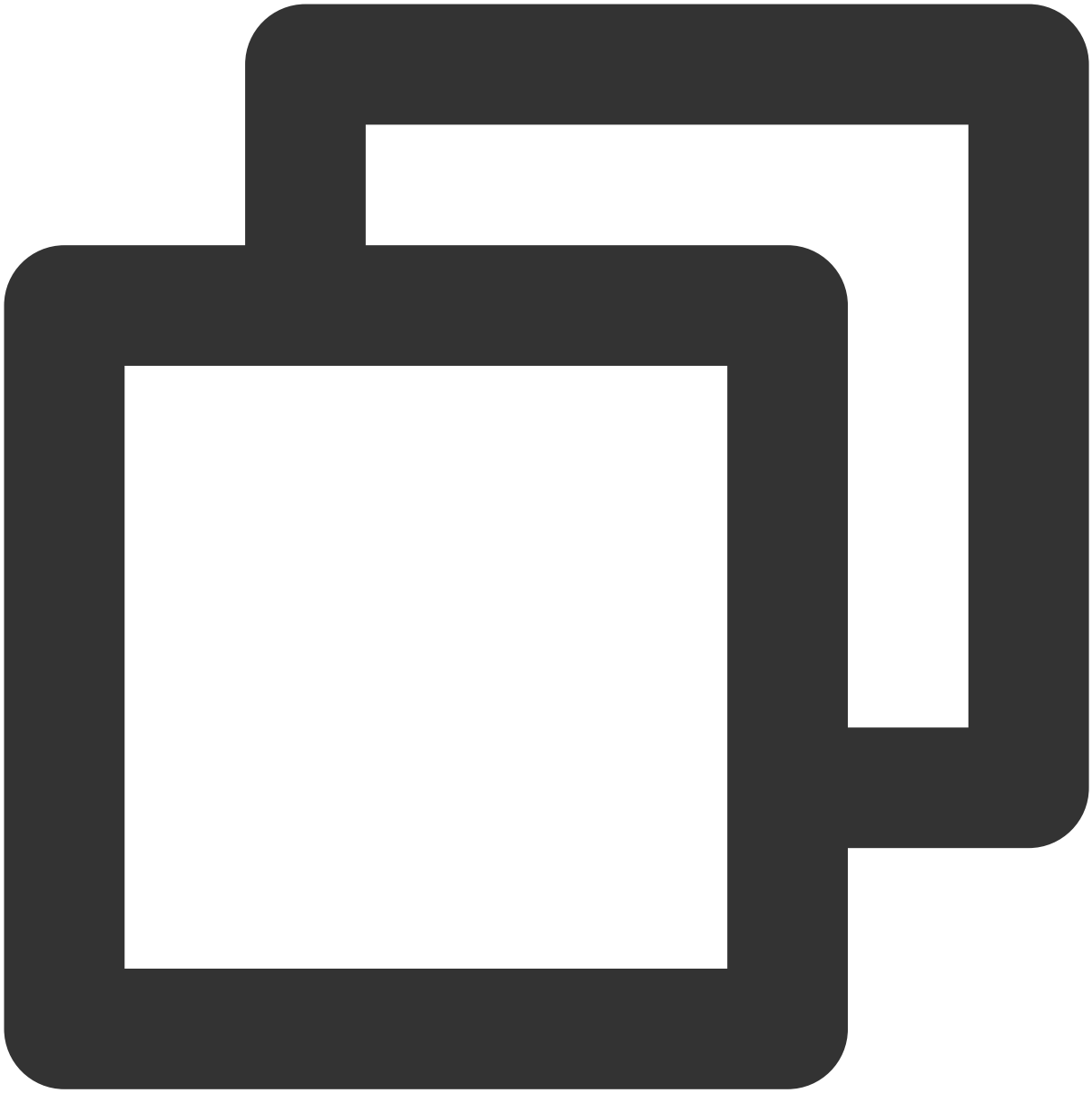
将原图转换为 TPG 格式，URL 地址如下。

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

SVG 压缩

最近更新时间：2024-01-18 15:11:52

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。
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

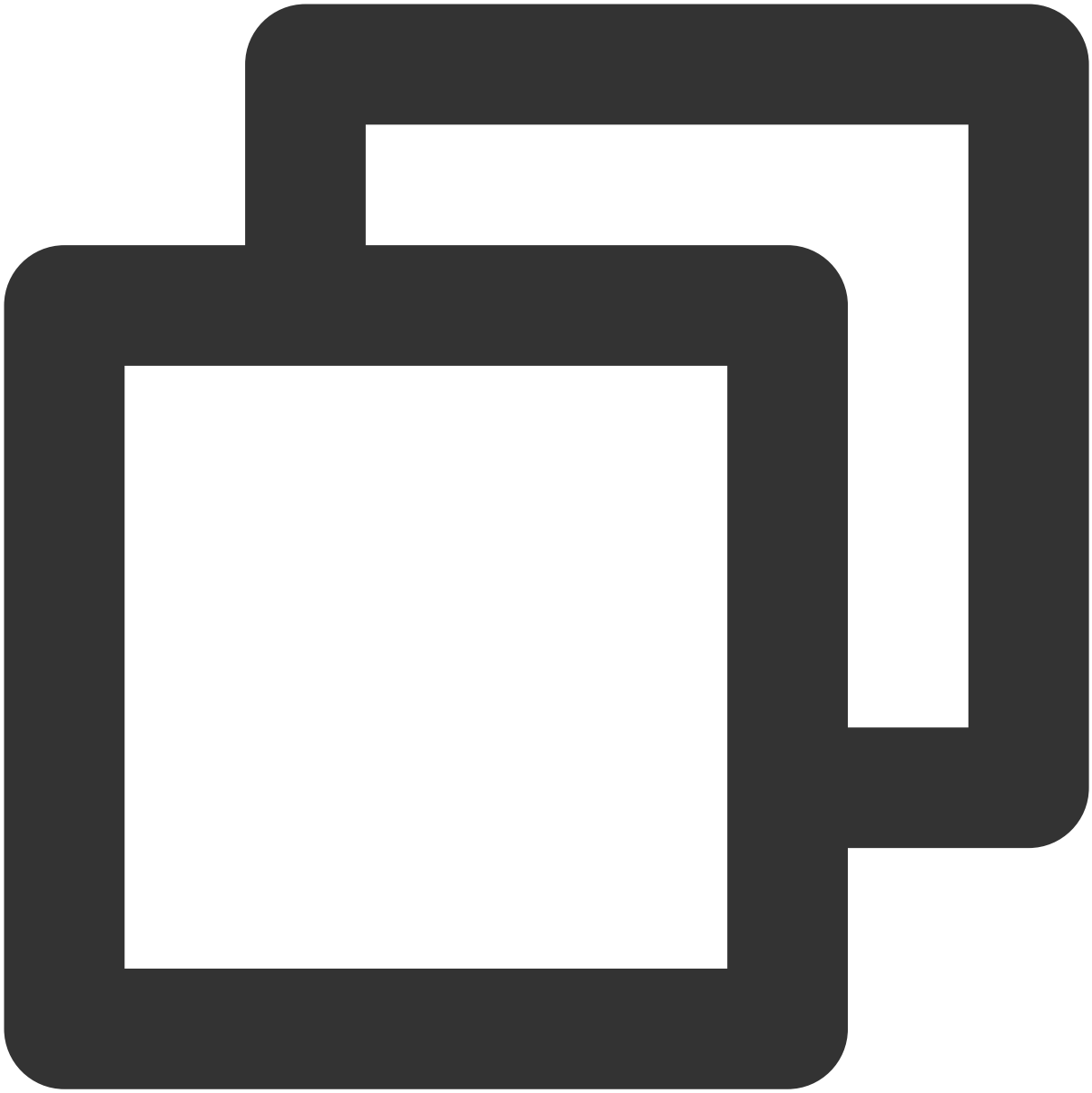
支持格式：仅支持将 svg 格式图片作为输入。
体积限制：处理图片原图大小不超过32MB。

使用方式

数据万象通过 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，具体费用请参见 图片处理费用。

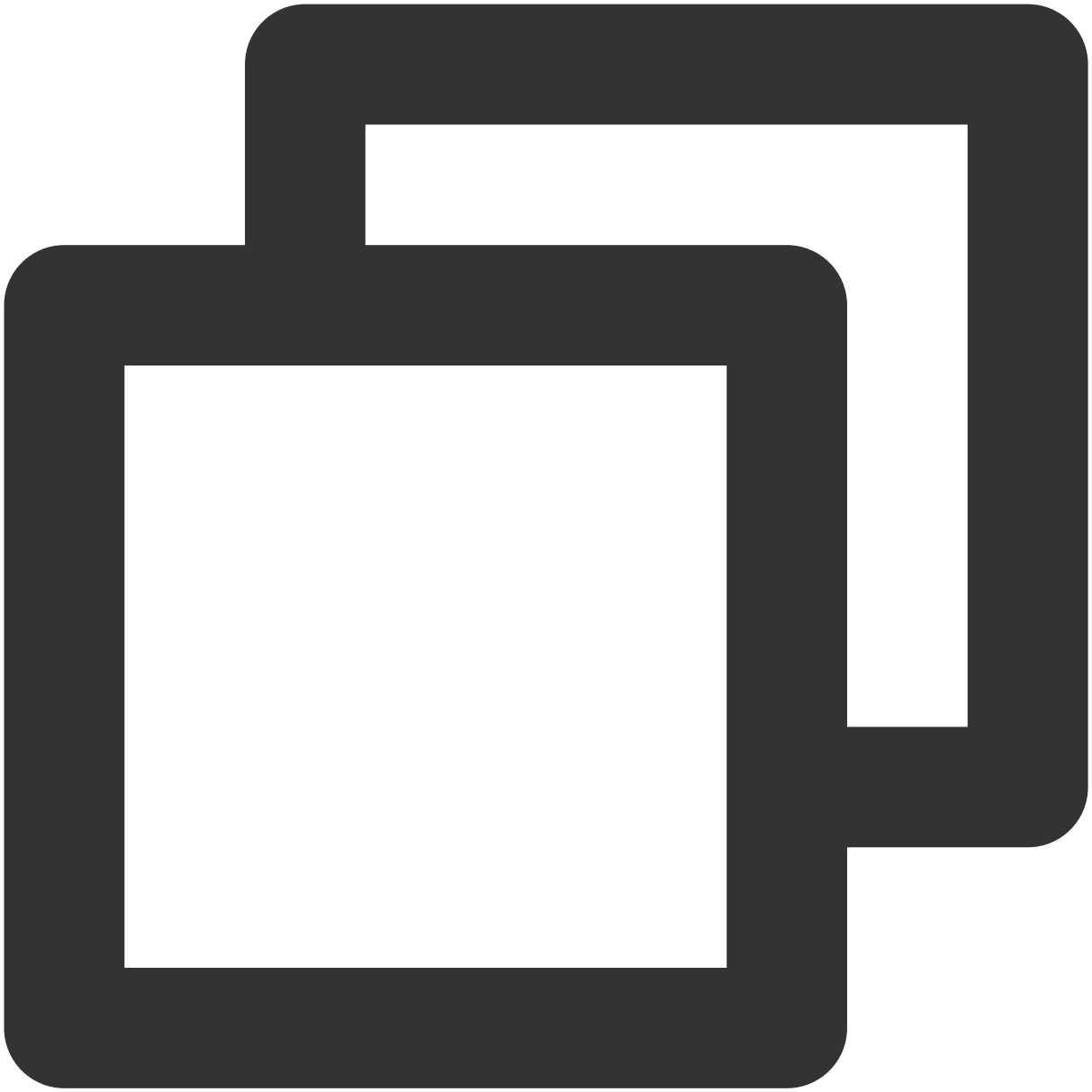
接口示例

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



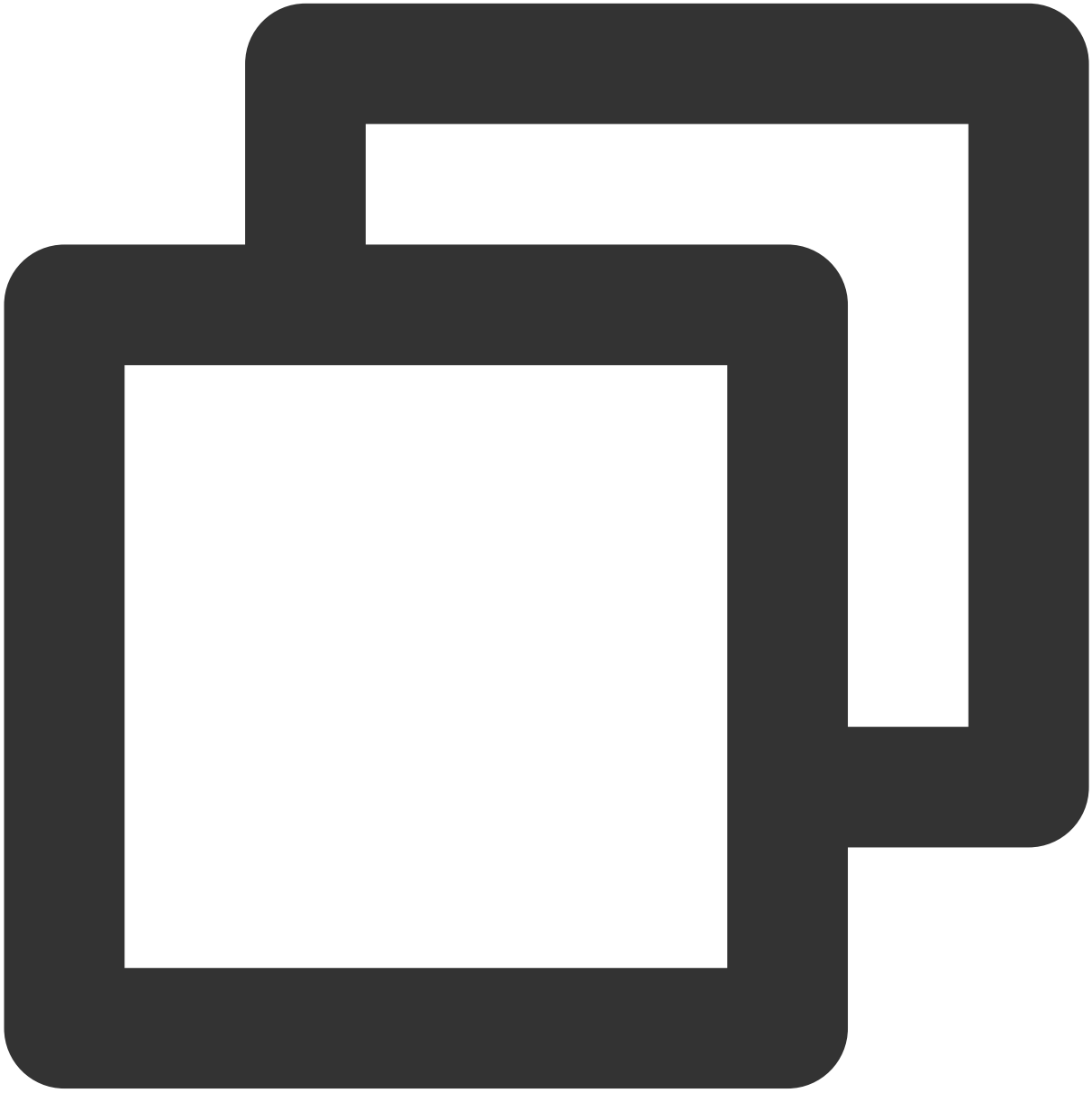
```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
"is_pic_info": 1,
"rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/format/svgc"
}]
```

```
}
```

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。

图片持久化处理

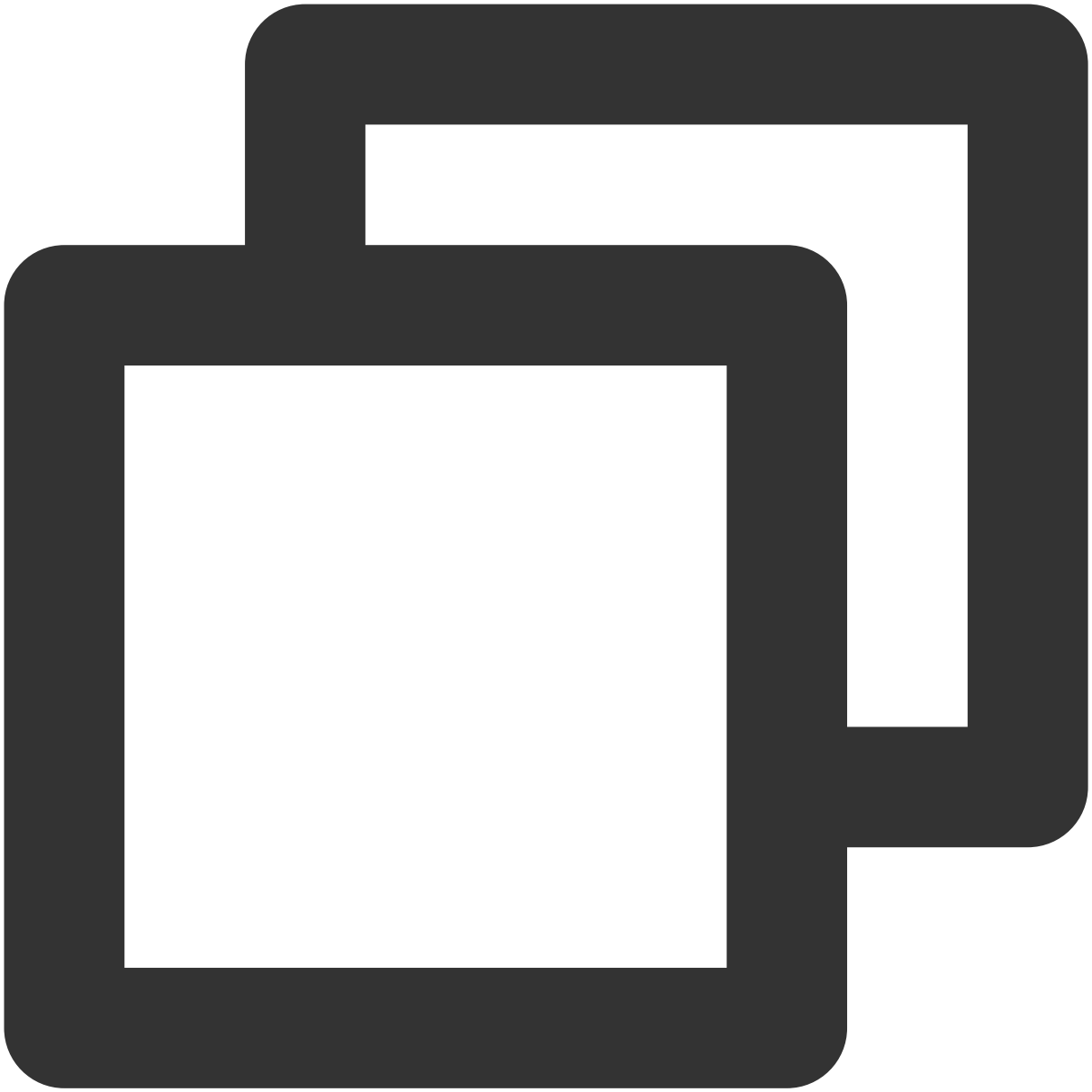
最近更新时间：2024-01-18 15:11:52

上传时处理

数据万象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32M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。



```
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）。

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。

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/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，参见数据万象图片处理 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：

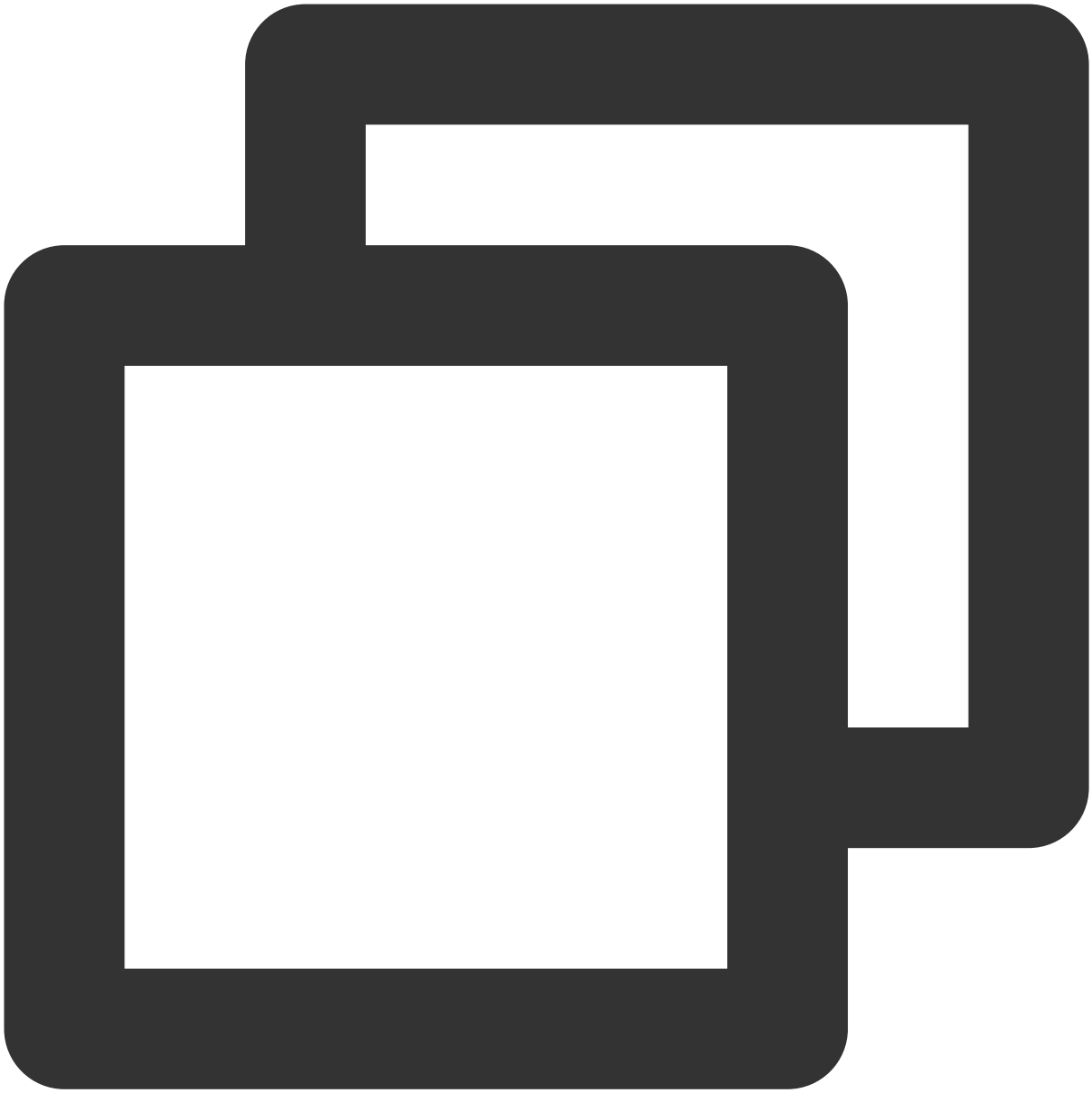
节点名称	类型	描述
Key	String	文件名
Location	String	图片路径
Format	String	图片格式

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

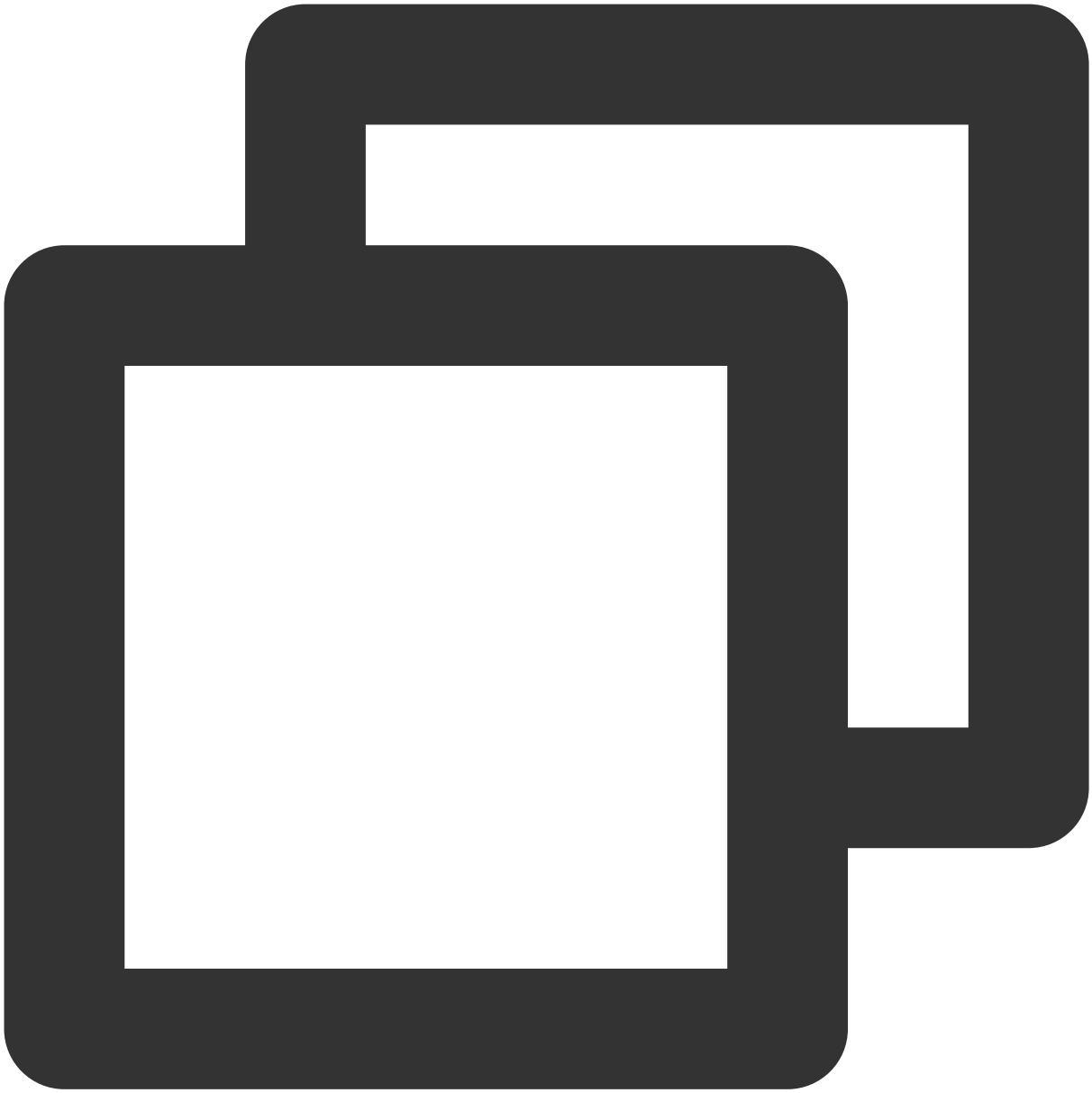
请求



```
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



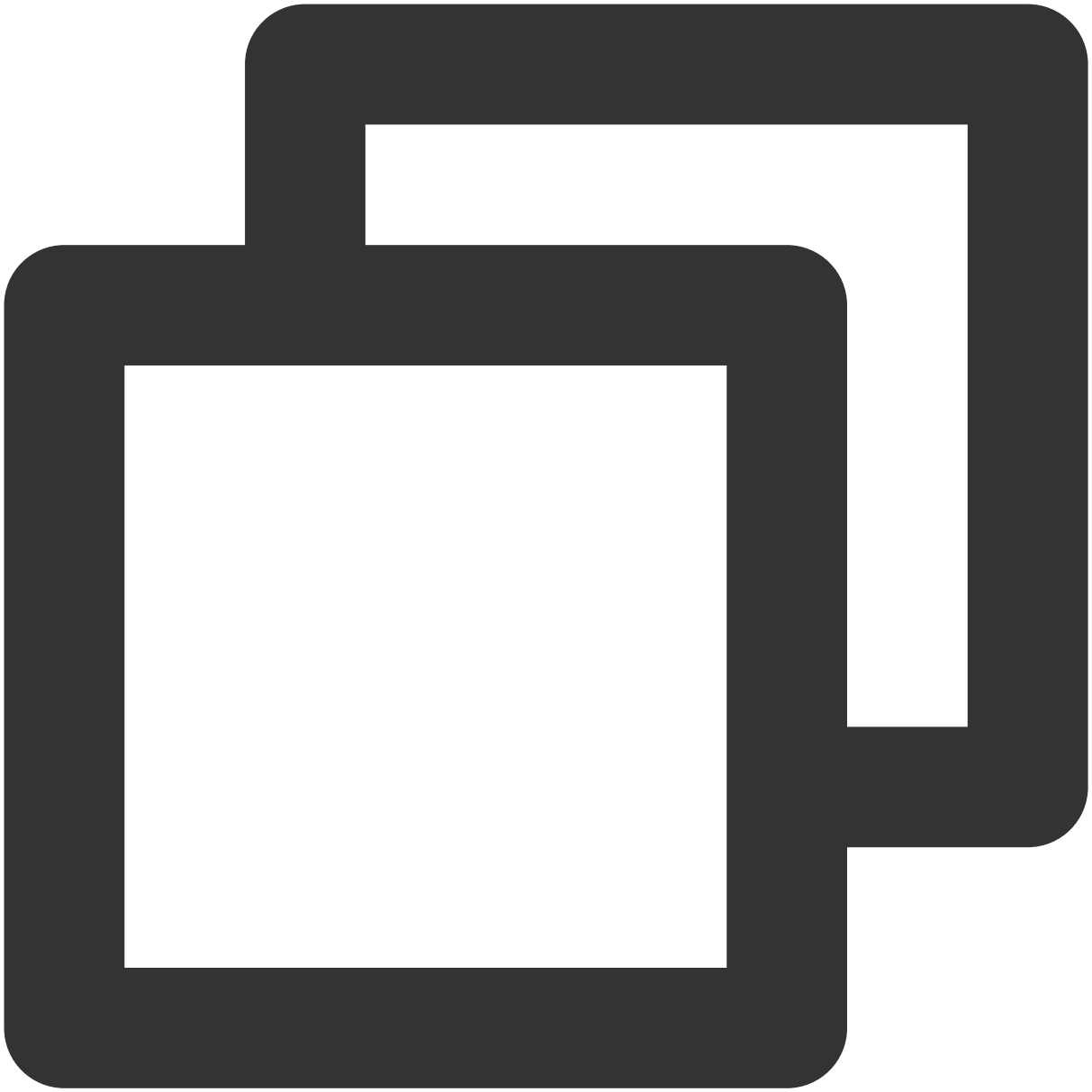
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<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>
```

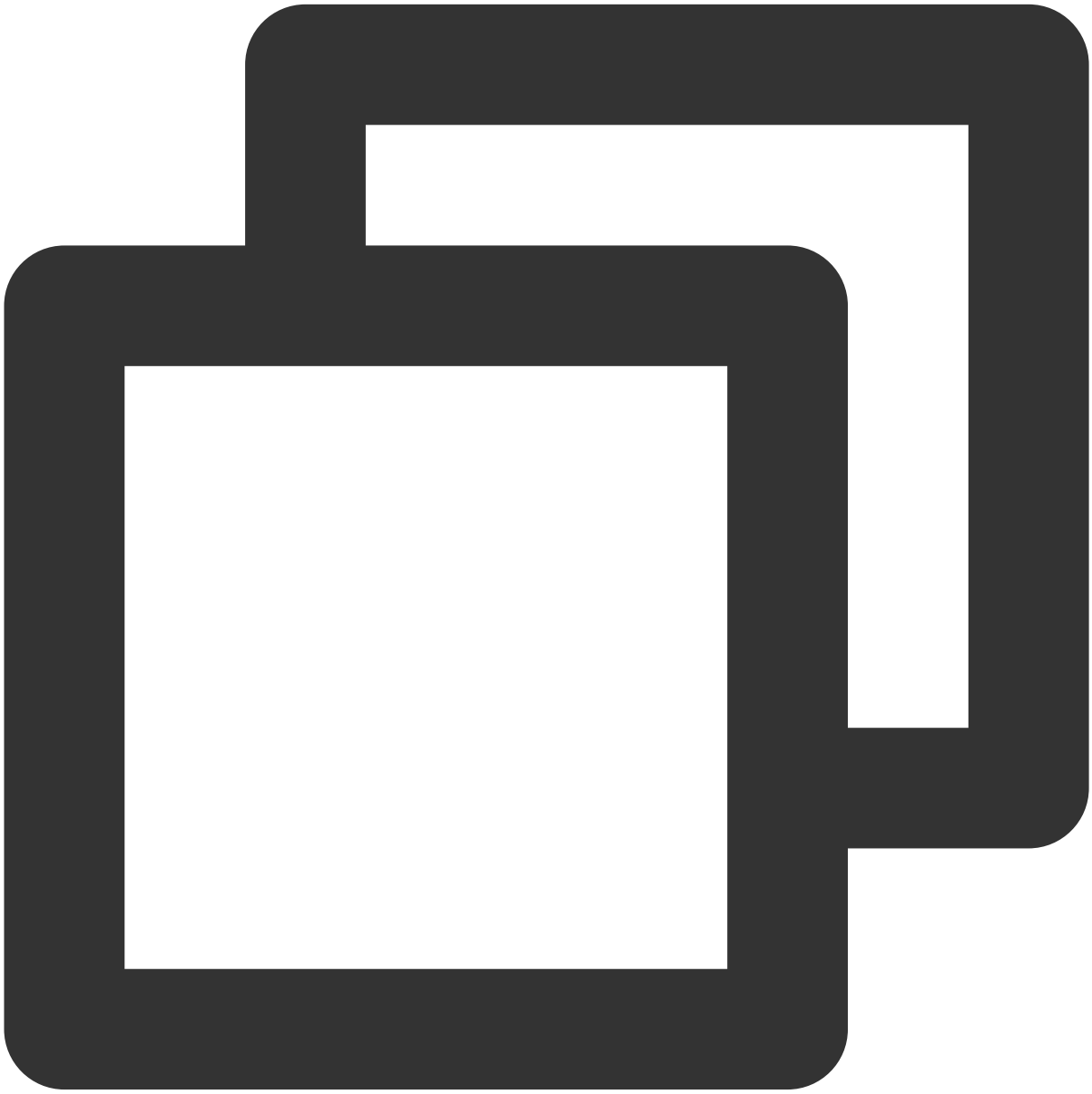
说明：

COS 接口请参见 COS Complete Multipart Upload 文档。

云上数据处理

数据万象的图片处理 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，参见数据万象 基础图片处理 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

版权所有：腾讯云计算（北京）有限责任公司

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：

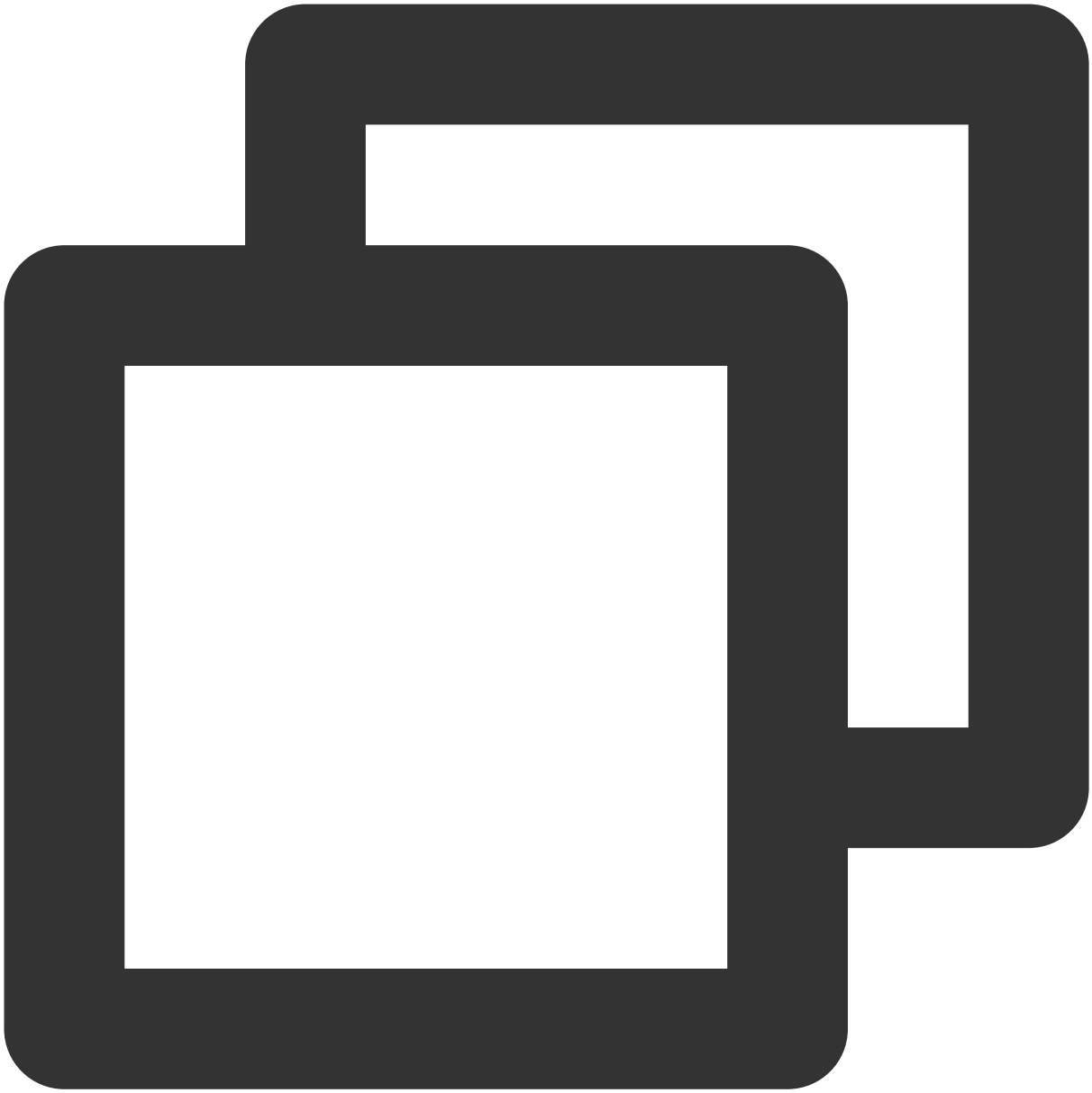
节点名称	类型	描述
Key	String	文件名
Location	String	图片路径
Format	String	图片格式

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

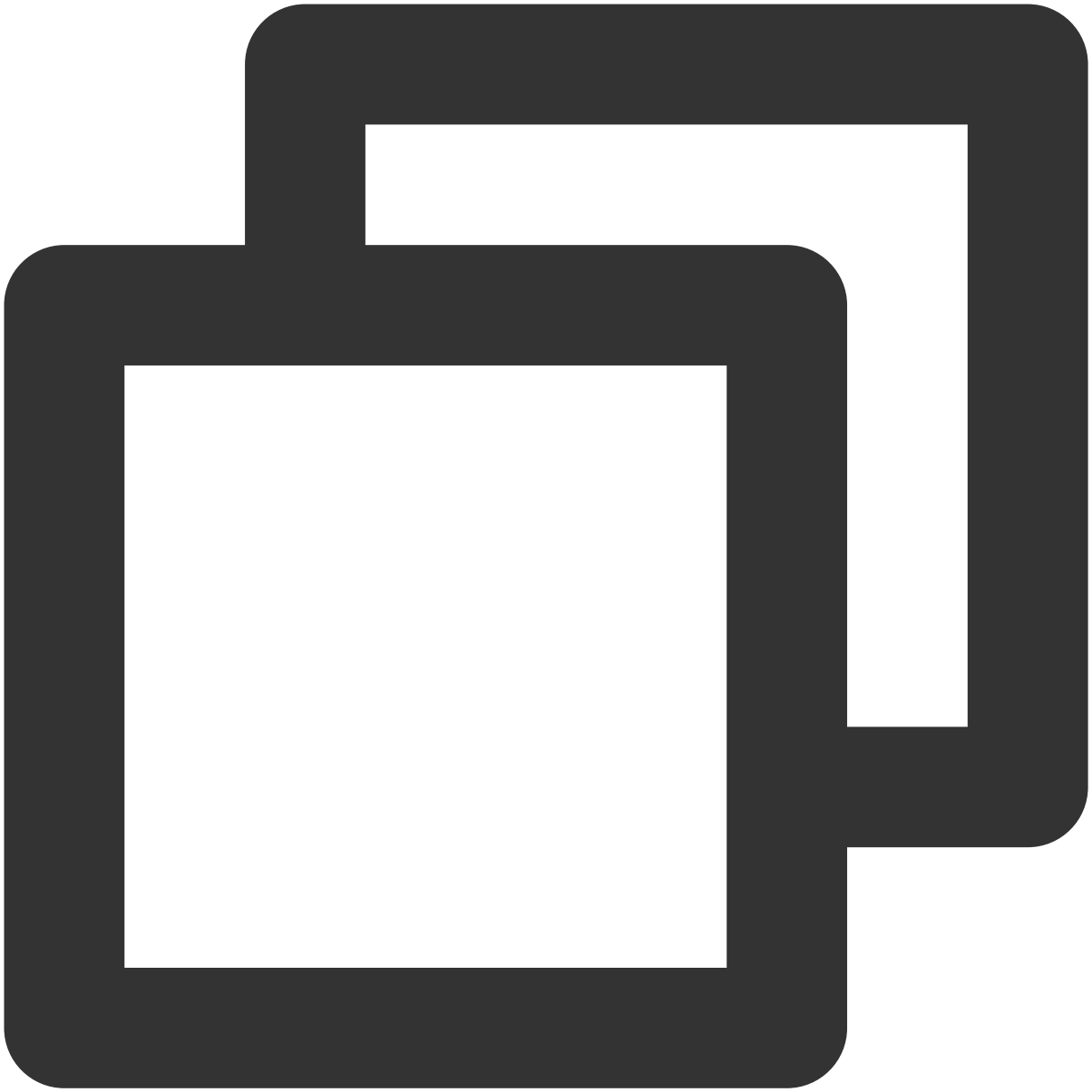
示例

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>
```

盲水印

最近更新时间：2024-01-18 15:11:52

简介

盲水印功能是指将水印以不可见的形式添加到图片中，这样的水印不会图片观感产生影响，同时也保证了图片的原创性。当发现图片被盗取后，您可以通过 提取盲水印，将不可见的水印提取出来，验证图片的归属。
通过该 API 接口，您可以：
在上传时为图片添加盲水印、提取盲水印
对 COS 上的图片添加盲水印、提取盲水印
在下载时为图片添加盲水印（不支持下载时提取盲水印）
数据万象提供的盲水印功能拥有半盲水印、全盲水印和文字盲水印三种类型：

水印类型	特性	适用场景
半盲水印（type1）	抗攻击性强，但提取水印需原图	小图（640px x 640px以下）使用
全盲水印（type2）	提取方便，提取水印仅需水印图，无需对比原图	批量添加，批量校验
文字盲水印（type3）	可直接将文字信息添加至图片中	终端信息添加

盲水印是付费服务，使用时需在相应存储桶配置页中通过开关开通服务。具体费用请参见 图片处理费用。
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。

适用场景

鉴权追责

您可对图片资源增加半盲水印，在发现恶意攻击方盗取您的资源后将疑似被盗取图取回，并与相应原图进行盲水印提取操作，若能够得到有效水印图即可证明资源归属。

上传查重

为解决部分用户使用其他用户资源重复上传相同信息的问题（如房产图、汽车图、商品图等），您可在用户上传图片资源前先进行全盲水印提取，若提取到水印图信息则证明该图片来自之前已有资源，并进行相应操作（如提醒用户请勿重复上传资源）；若不存在全盲水印则添加全盲水印，保护图片资源不被其他用户下载后重复上传。

资源防泄露

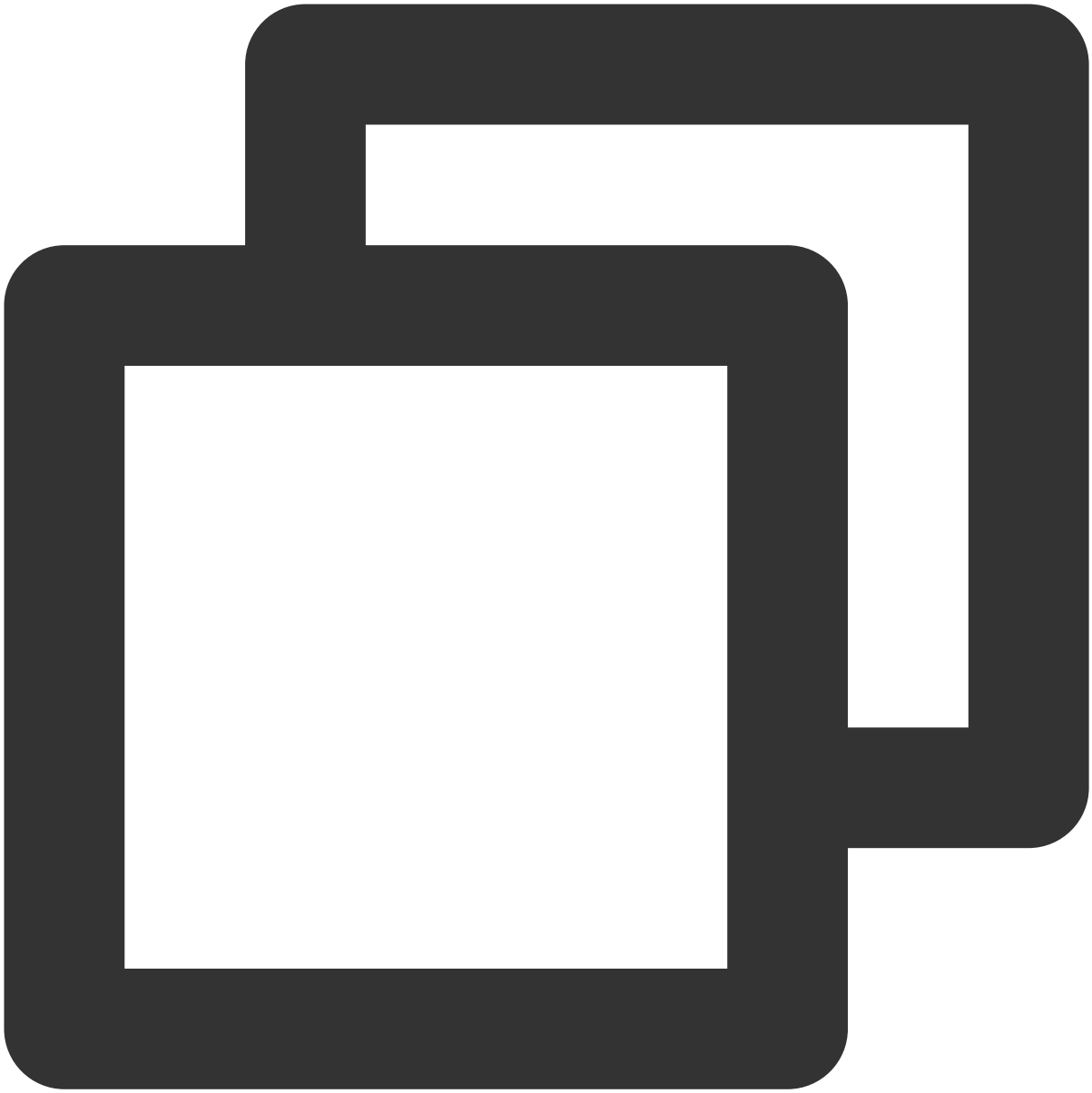
对于内部分享的图片资源，您可通过文字盲水印将访问方的信息在请求图片时添加至图片中，当资源泄露后可通过流传出的资源图提取出盲水印，进而得到泄露方信息。

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）。

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。

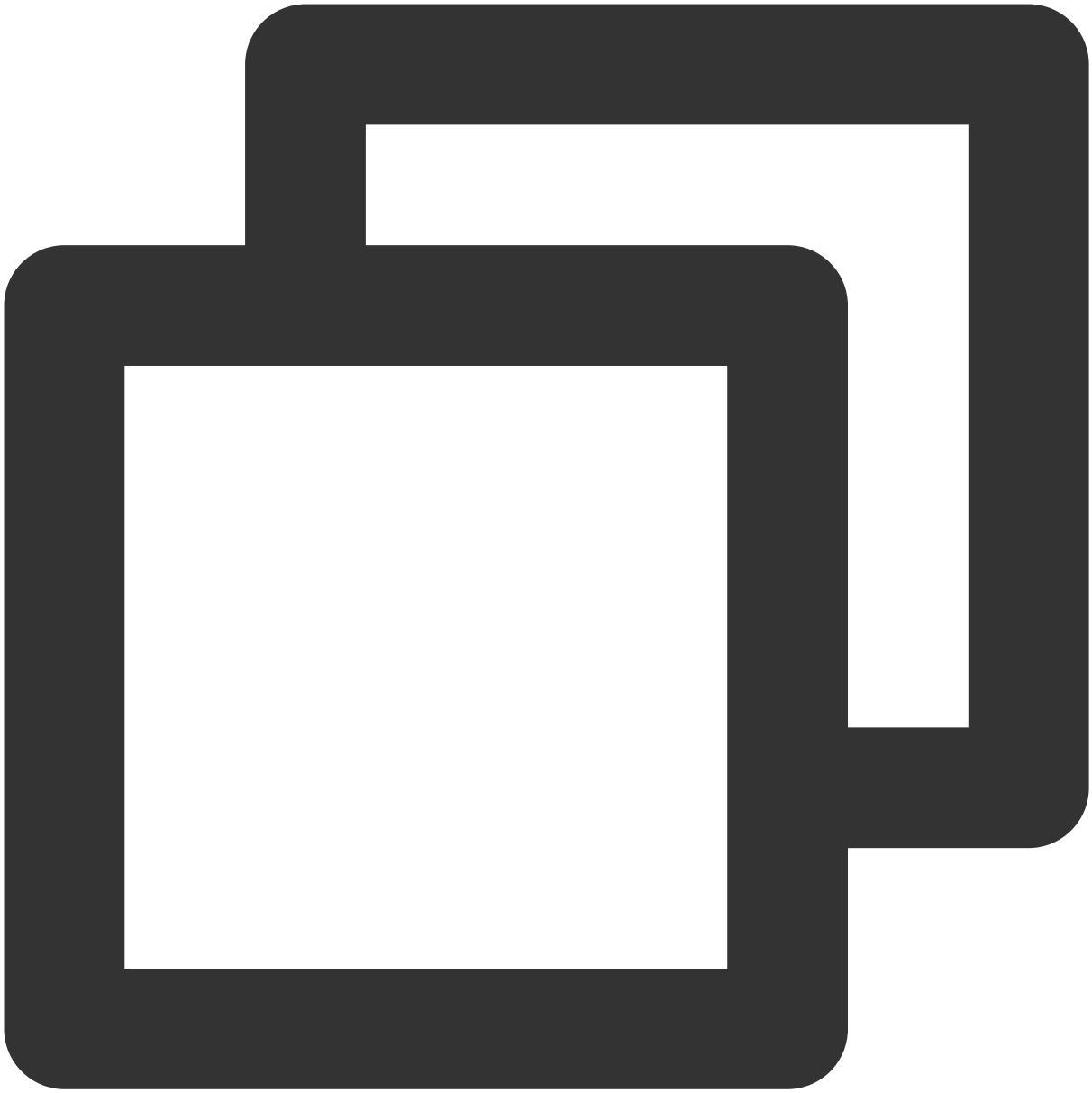
请求包头部增加图片处理参数 Pic-Operations，Pic-Operations 为 json 格式的字符串，具体参数如下：

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：	String	否

	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

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

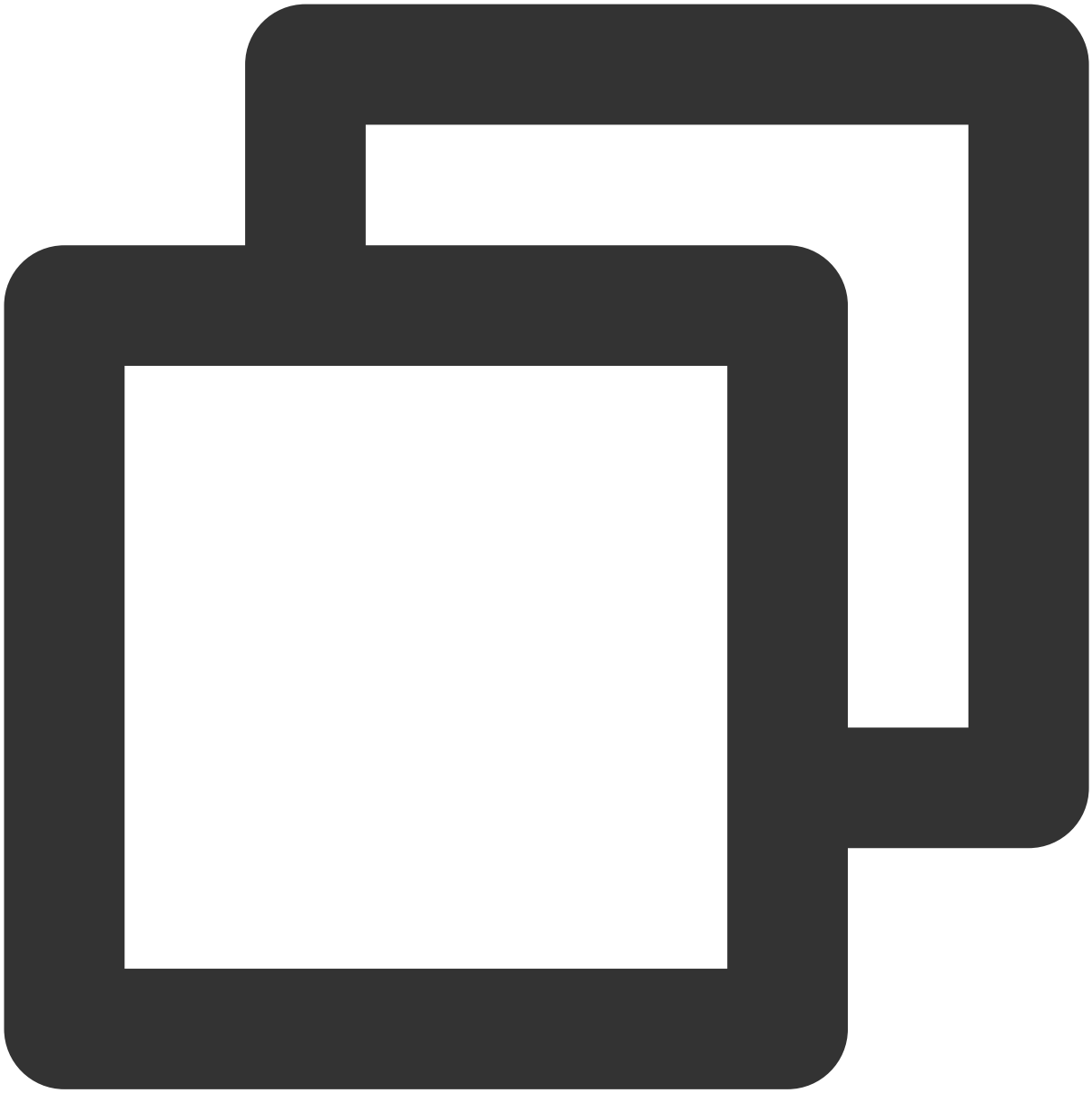
Object 节点内容：

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



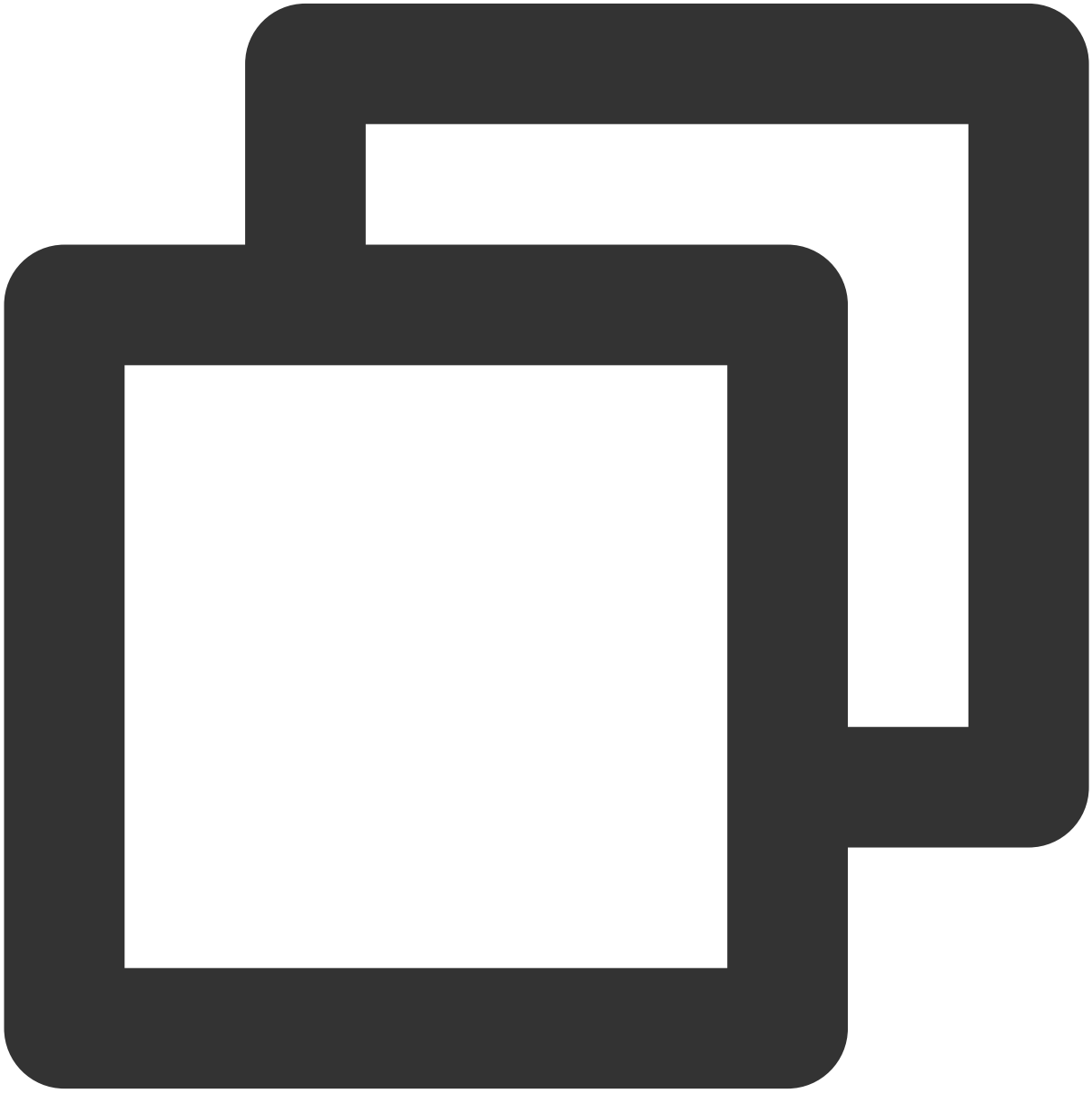
```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2018 09:06:15 GMT
Authorization:XXXXXXXXXXXX

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：

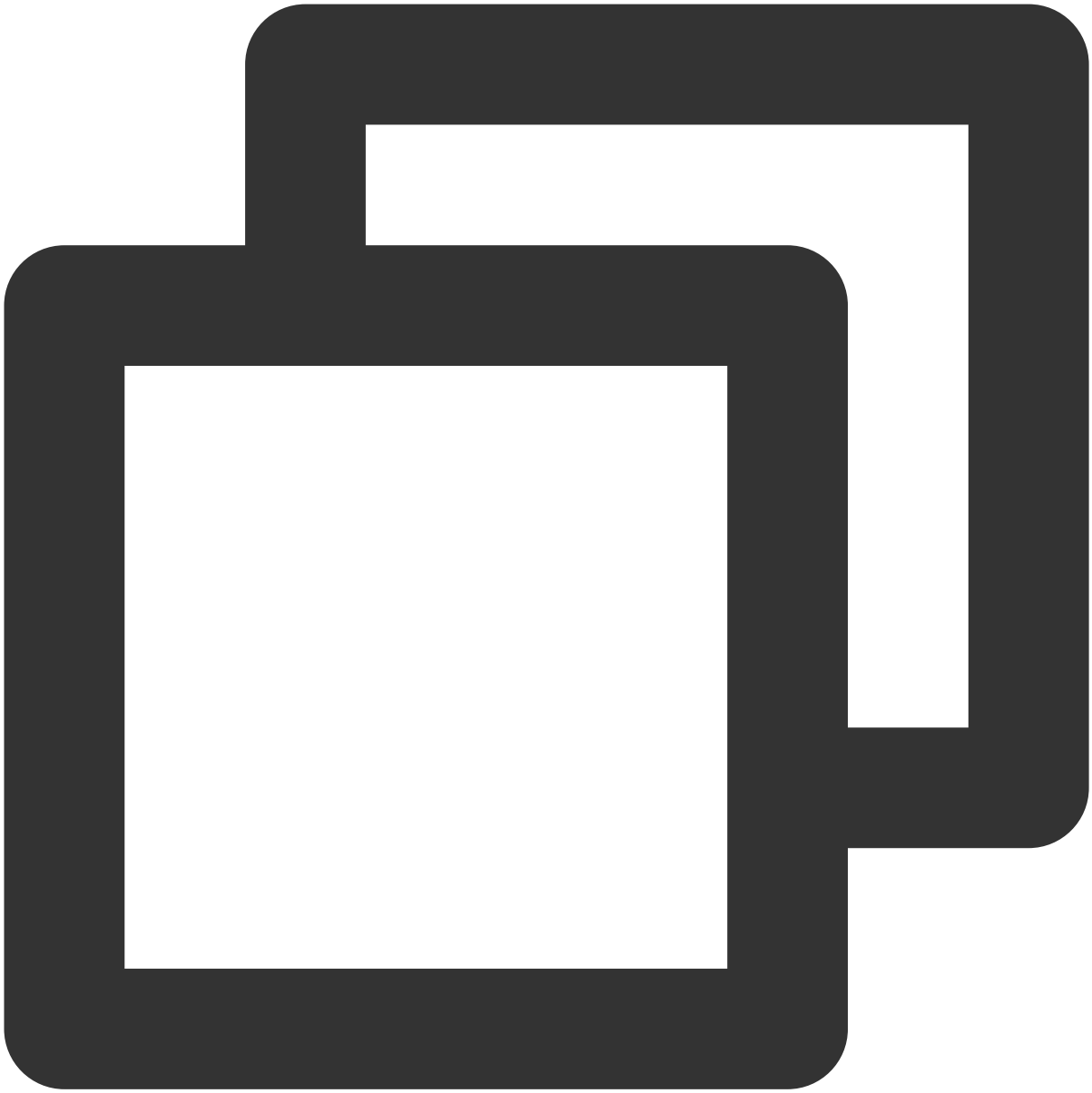
请求体的 [Object Content] 字段为原图的图片内容(二进制字符串)，请求头 Pic-Operations 的 rule 字段中，imageUrl 为水印图片的地址。

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>
```

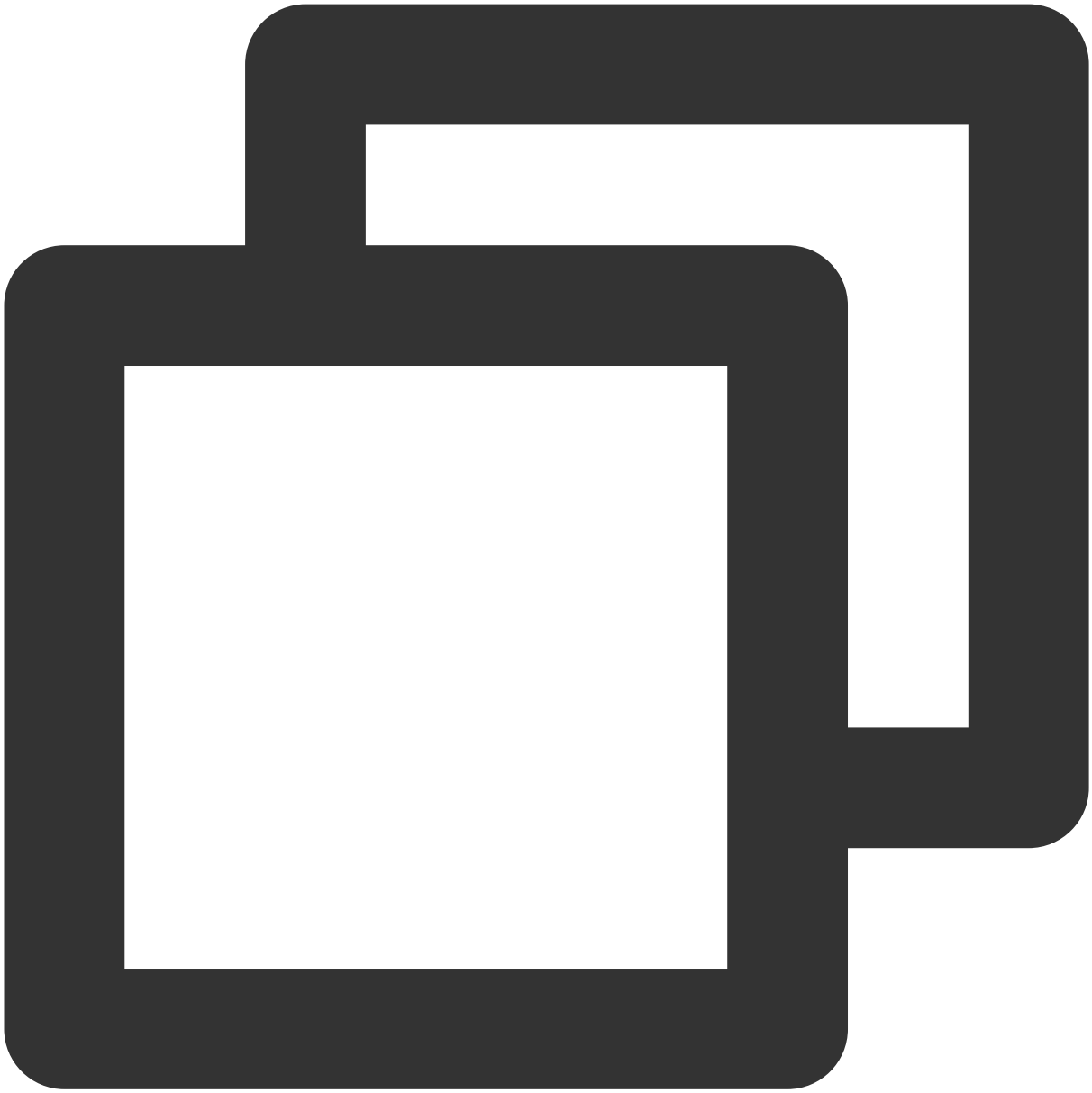
请求参数

请求参数说明如下：

参数	描述	类型	是否必选

ObjectKey	对象文件名，例如 folder/sample.jpg。	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

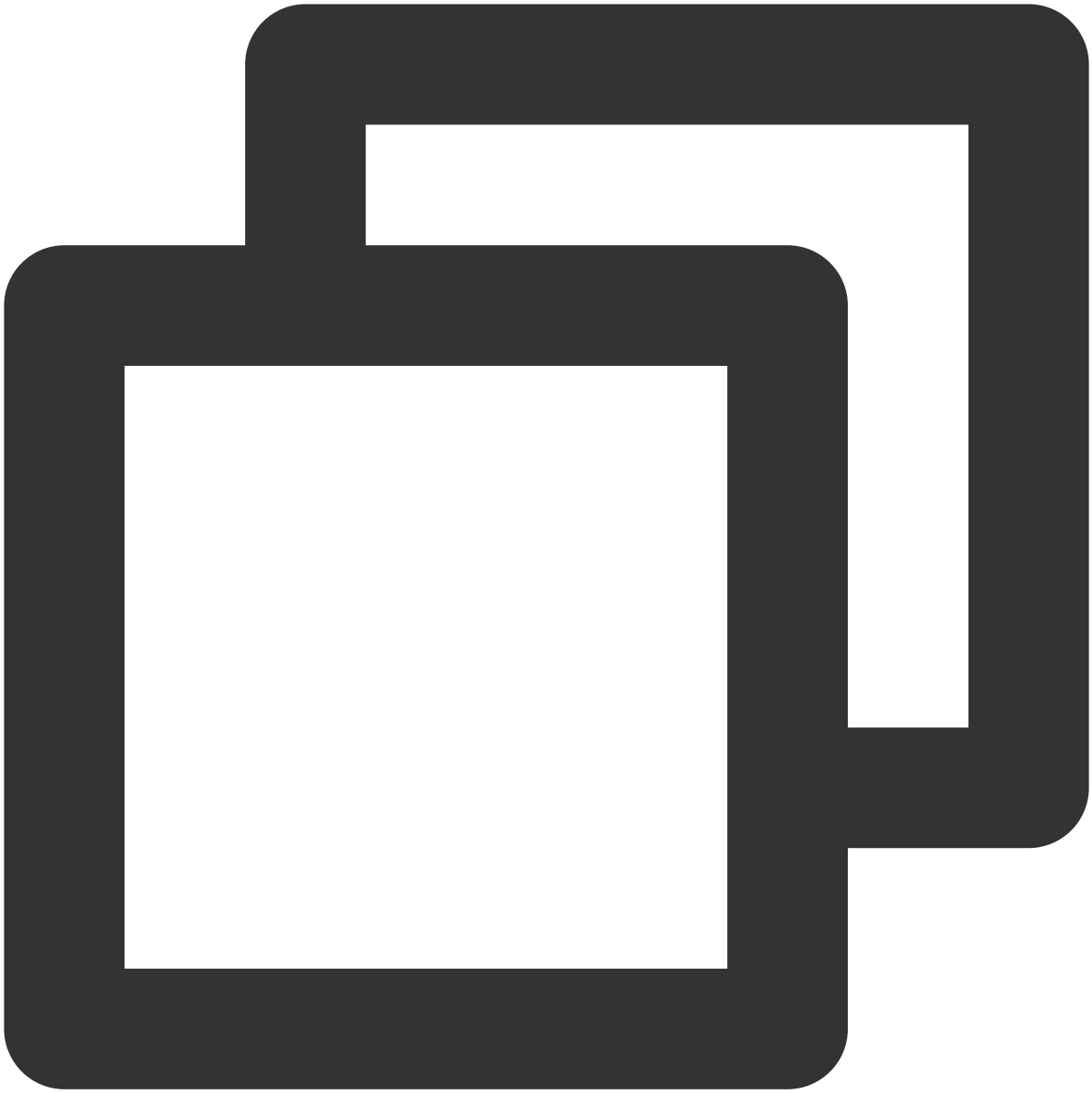


```
https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark
```

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）。

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。

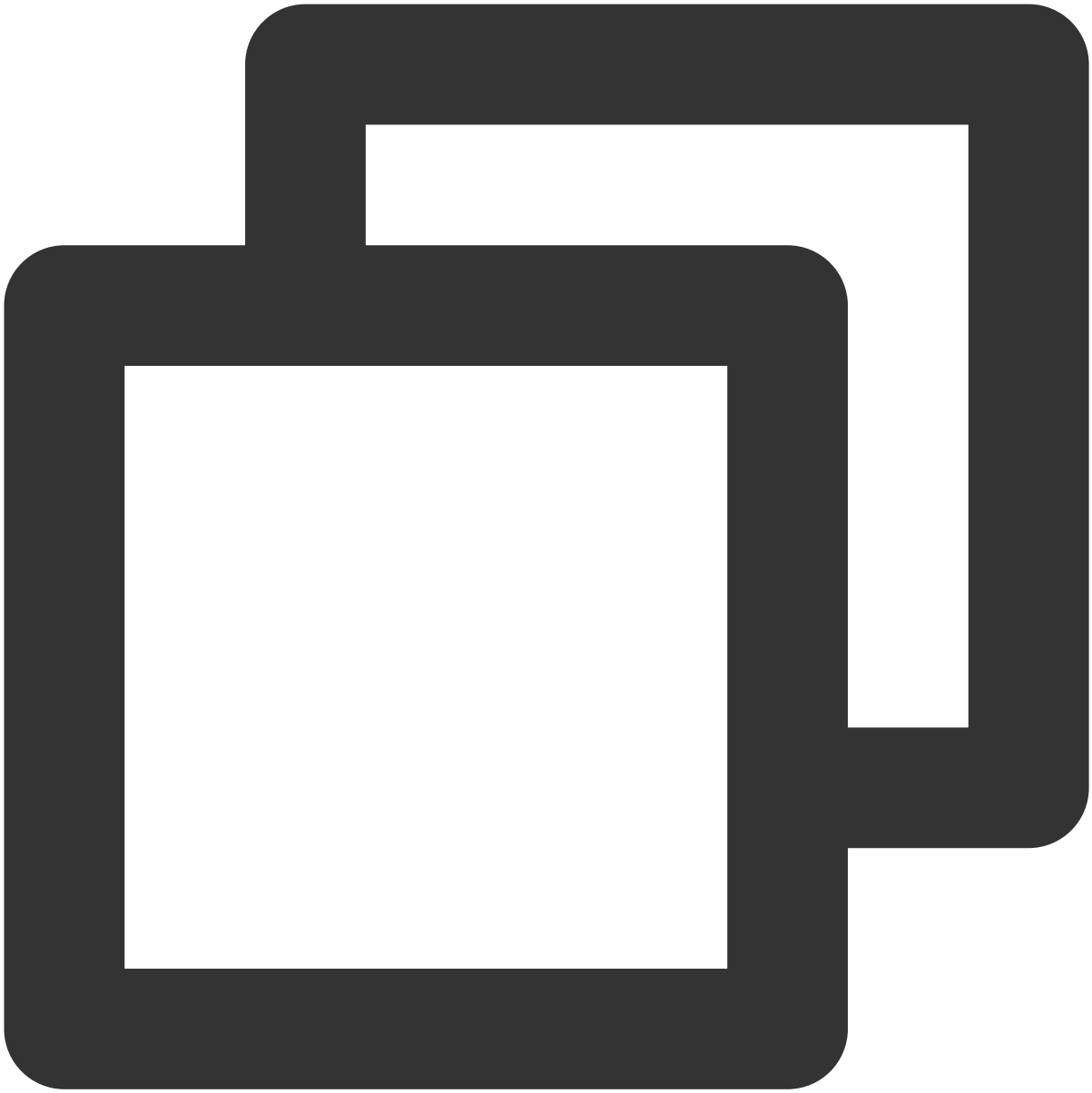
请求包头部增加图片处理参数 Pic-Operations，Pic-Operations 为 json 格式的字符串，具体参数如下：

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

	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
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：

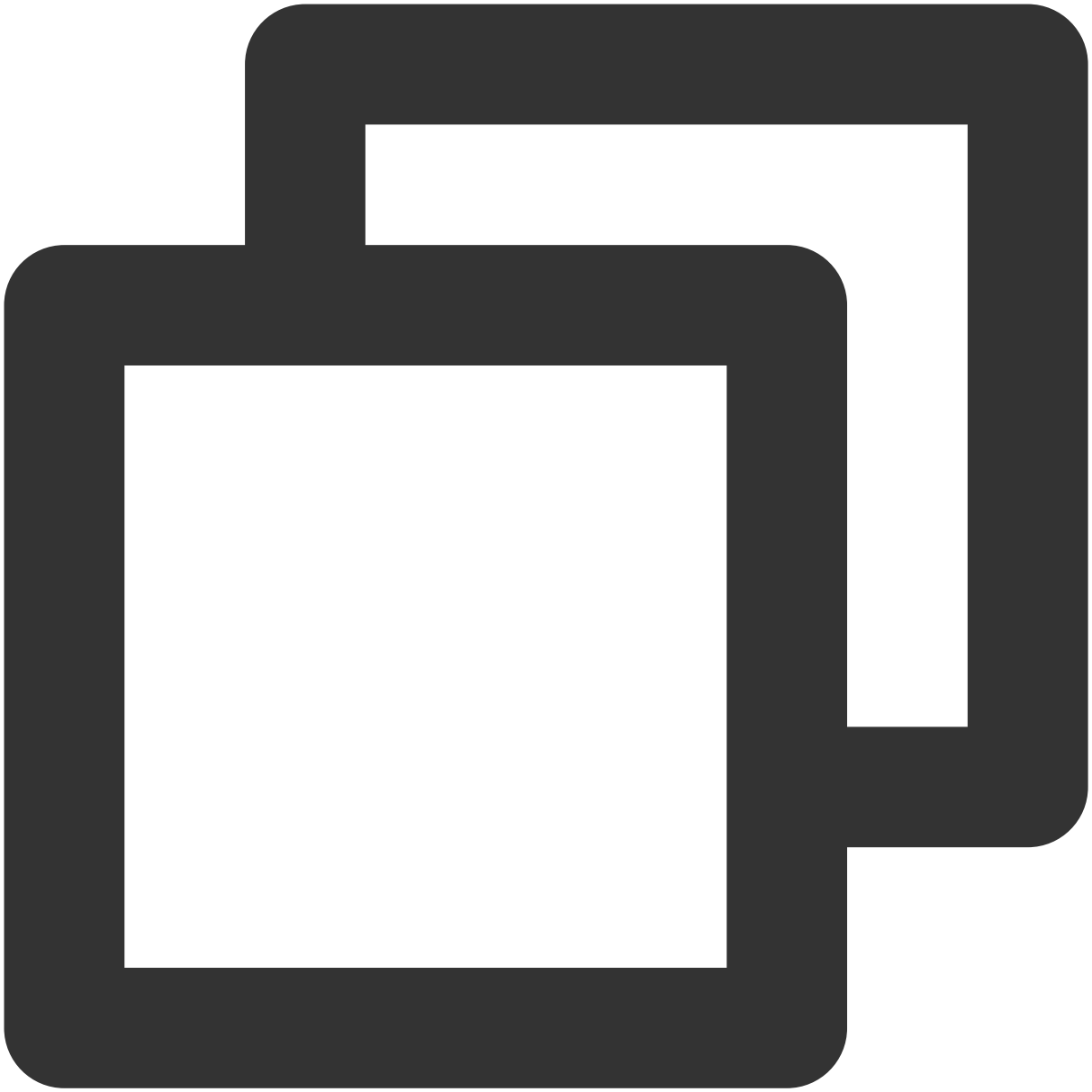
节点名称	描述	类型
Object	每一个图片处理结果	Container

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

请求

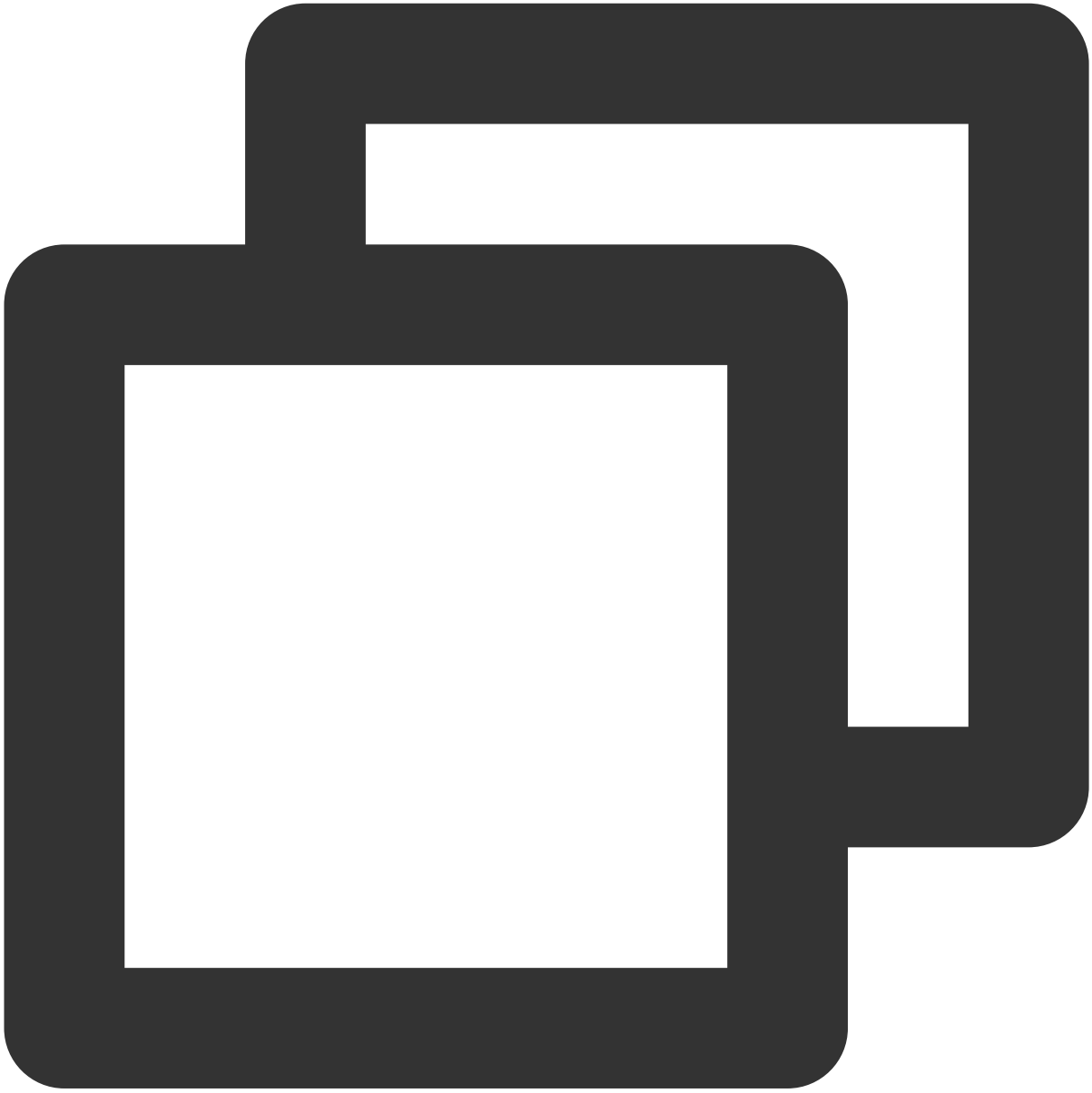


```
POST /exampleobject?image_process HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2018 09:06:15 GMT
Authorization:XXXXXXXXXXXX

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****

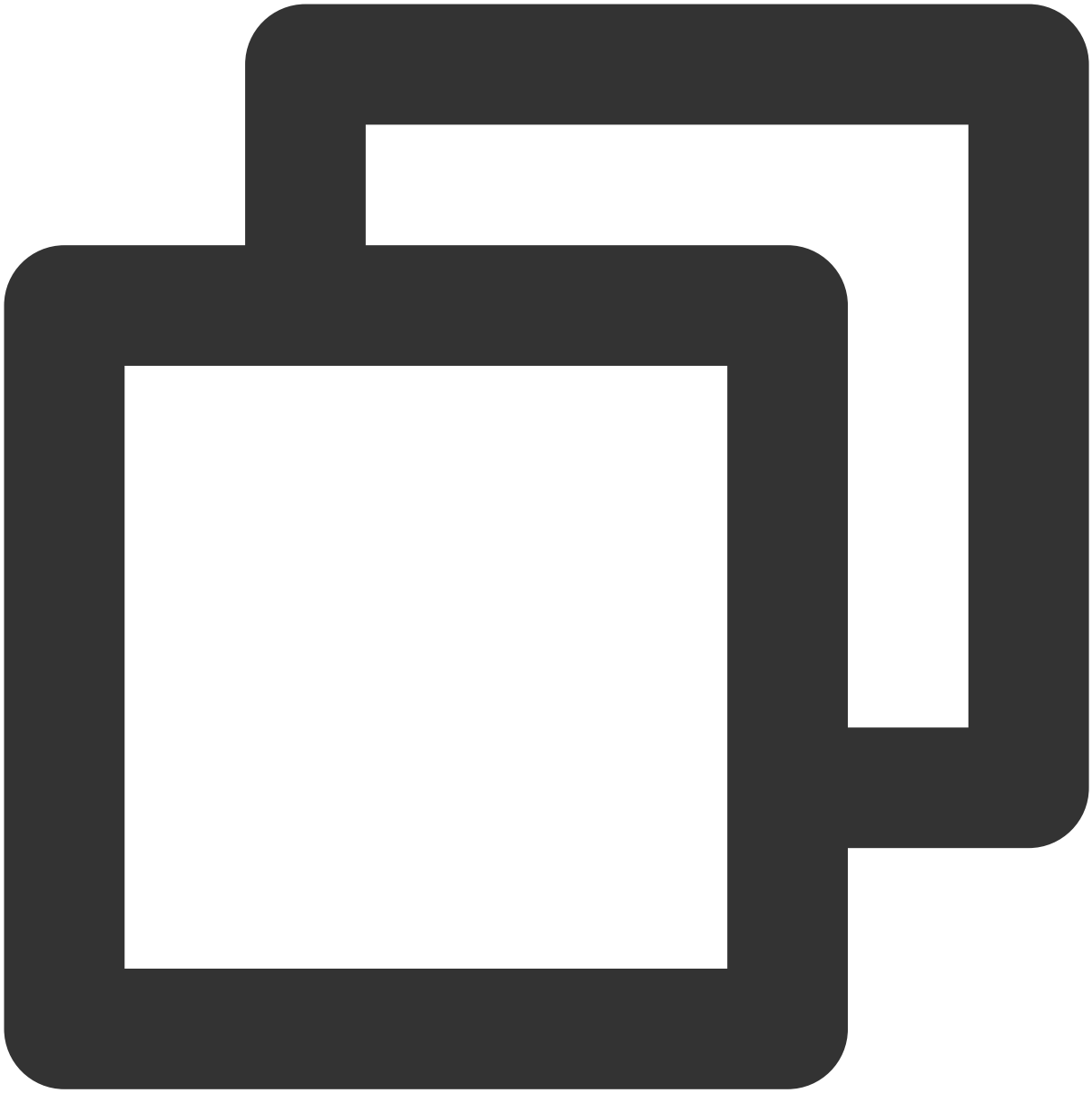
<UploadResult>
          <OriginalInfo>
              <Key>exampleobject</Key>
              <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/exampl
              <ImageInfo>
                   <Format>JPEG</Format>
                   <Width>640</Width>
                   <Height>427</Height>
                   <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）。

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。

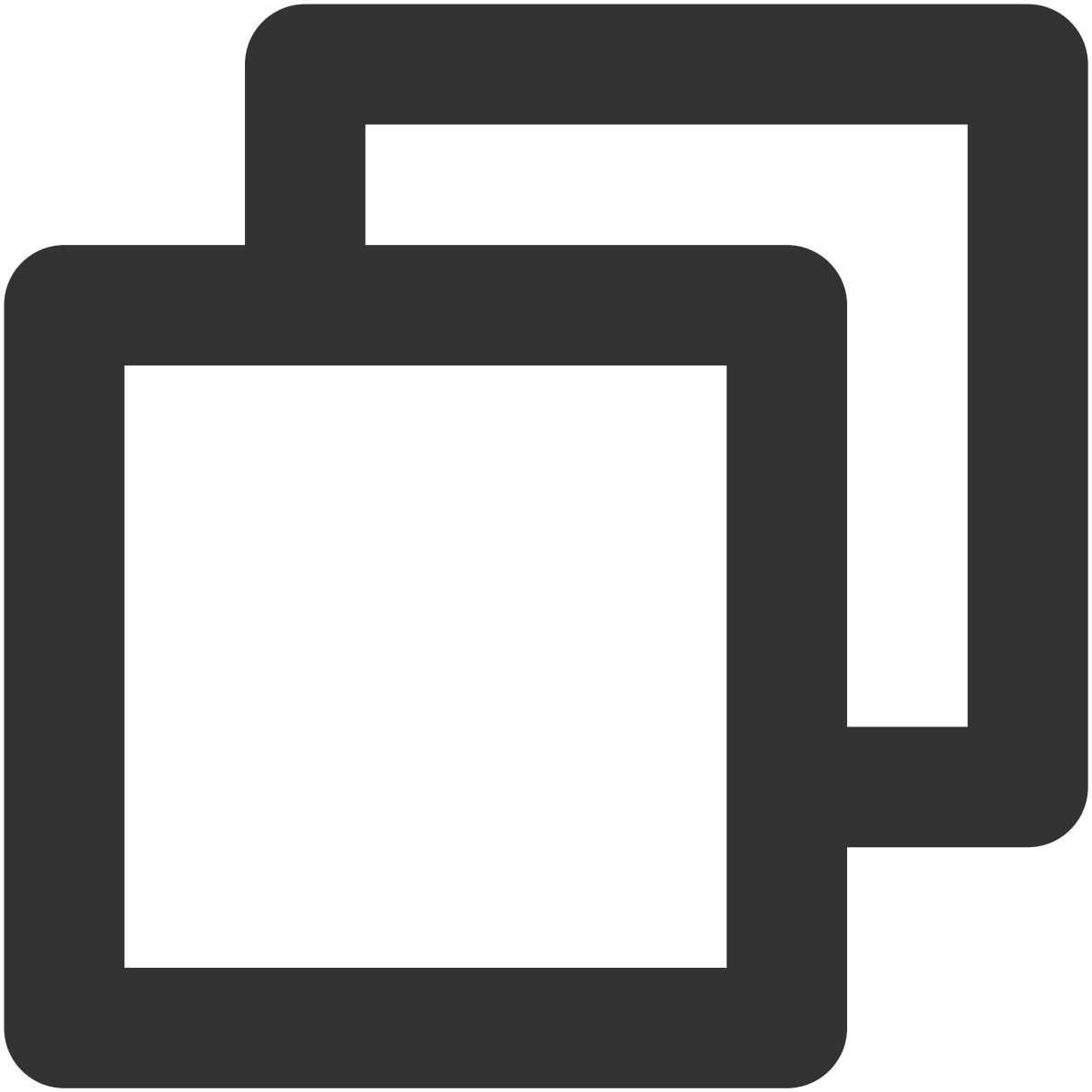
请求包头部增加图片处理参数 Pic-Operations，Pic-Operations 为 json 格式的字符串，具体参数如下：

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	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。

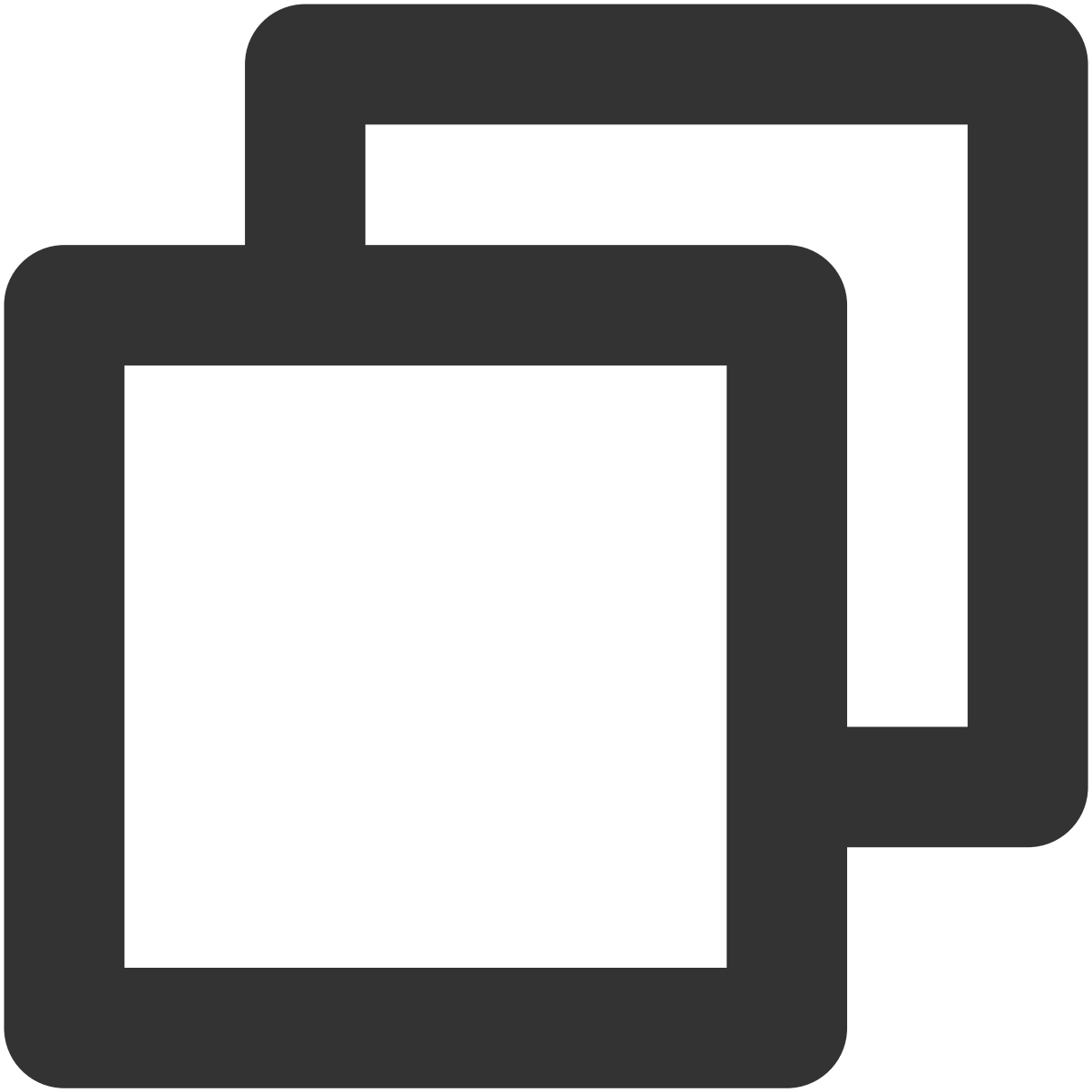
响应体

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	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



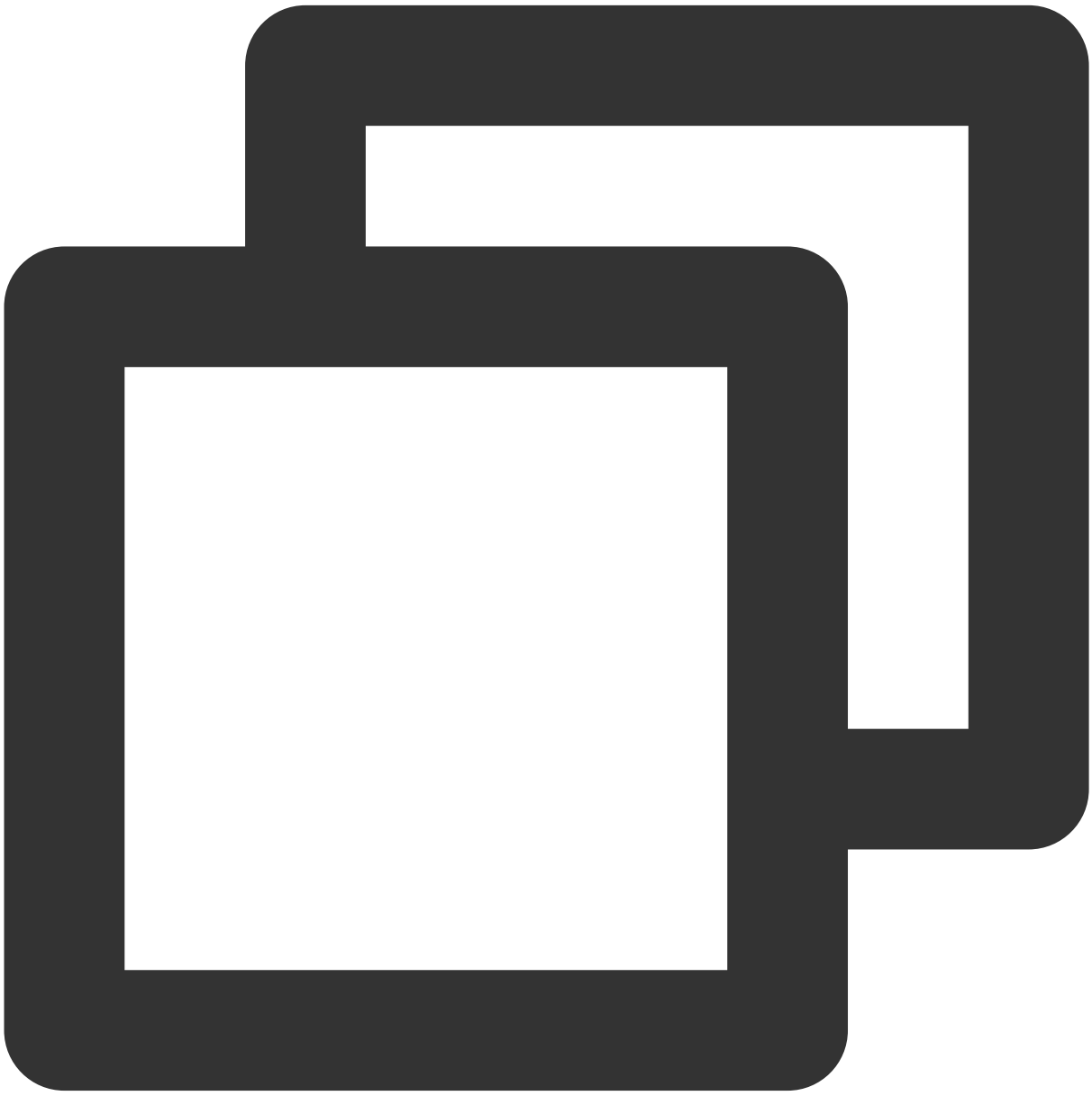
```
PUT /exampleobject1 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2018 09:06:15 GMT
Authorization:XXXXXXXXXXXX

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。

异常图片检测

最近更新时间：2024-01-18 15:11:52

功能概述

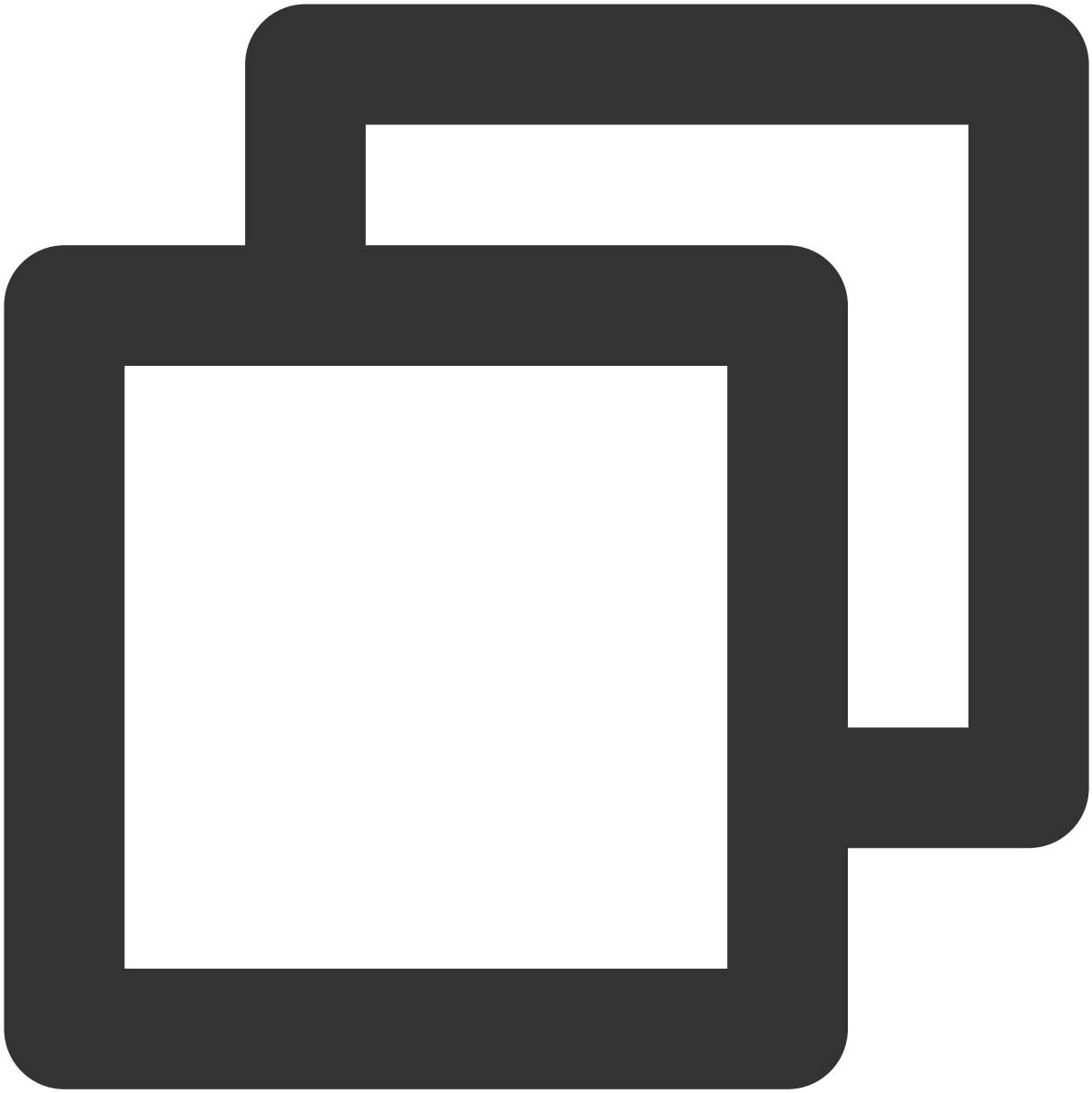
腾讯云数据万象通过 **ImageInspect** 接口检测图片中是否隐含其他类型的可疑文件。

限制说明

支持格式：支持检测 PNG 后带 TS 视频流、BMP 带 TS 视频流、纯 TS 视频流（仅更改后缀）三种类型的异常图片。
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。

接口示例

版权所有：腾讯云计算（北京）有限责任公司



```
GET /<ObjectKey>?ci-process=ImageInspect HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

处理参数说明

请求参数

操作名称：ImageInspect。

参数	含义
ObjectKey	对象文件名，例如 folder/sample.jpg。

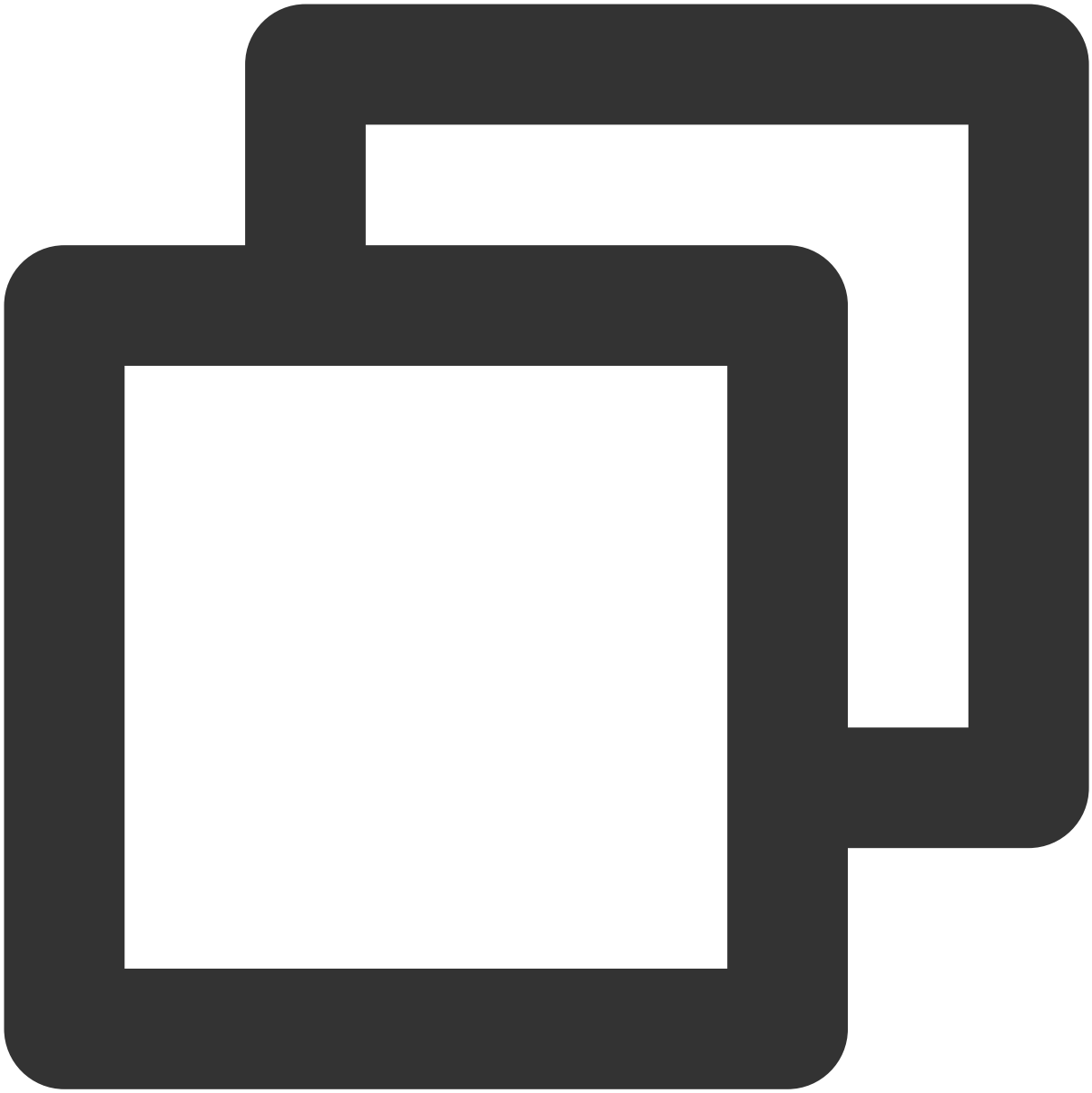
响应参数

参数	含义
picSize	检测到的可疑文件的大小，单位为 Bytes
picType	检测到的可疑文件类型。例如 ts、mkv 等
suspicious	是否检测到图片格式以外的文件。false 为正常图片文件，true 表示含有其他格式的可疑文件
suspiciousBeginByte	检测到的可疑文件的起始字节位置，单位为 Bytes
suspiciousEndByte	检测到的可疑文件的末尾字节位置，单位为 Bytes
suspiciousSize	检测到的可疑文件的大小
suspiciousType	检测到的可疑文件的类型，例如 MPEG-TS

实际案例

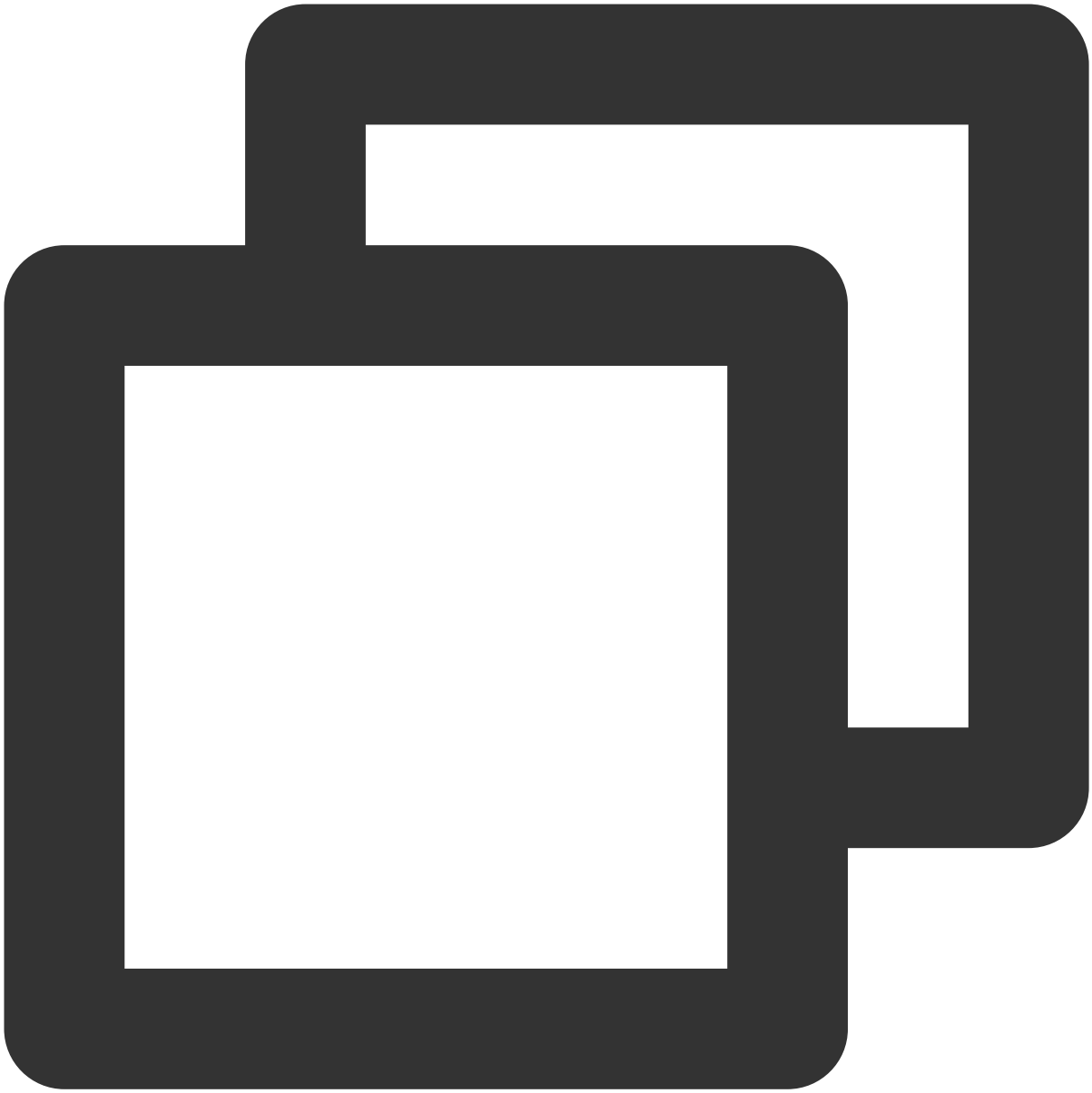
案例一：公有读

请求



```
http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?ci-process=Imag
```

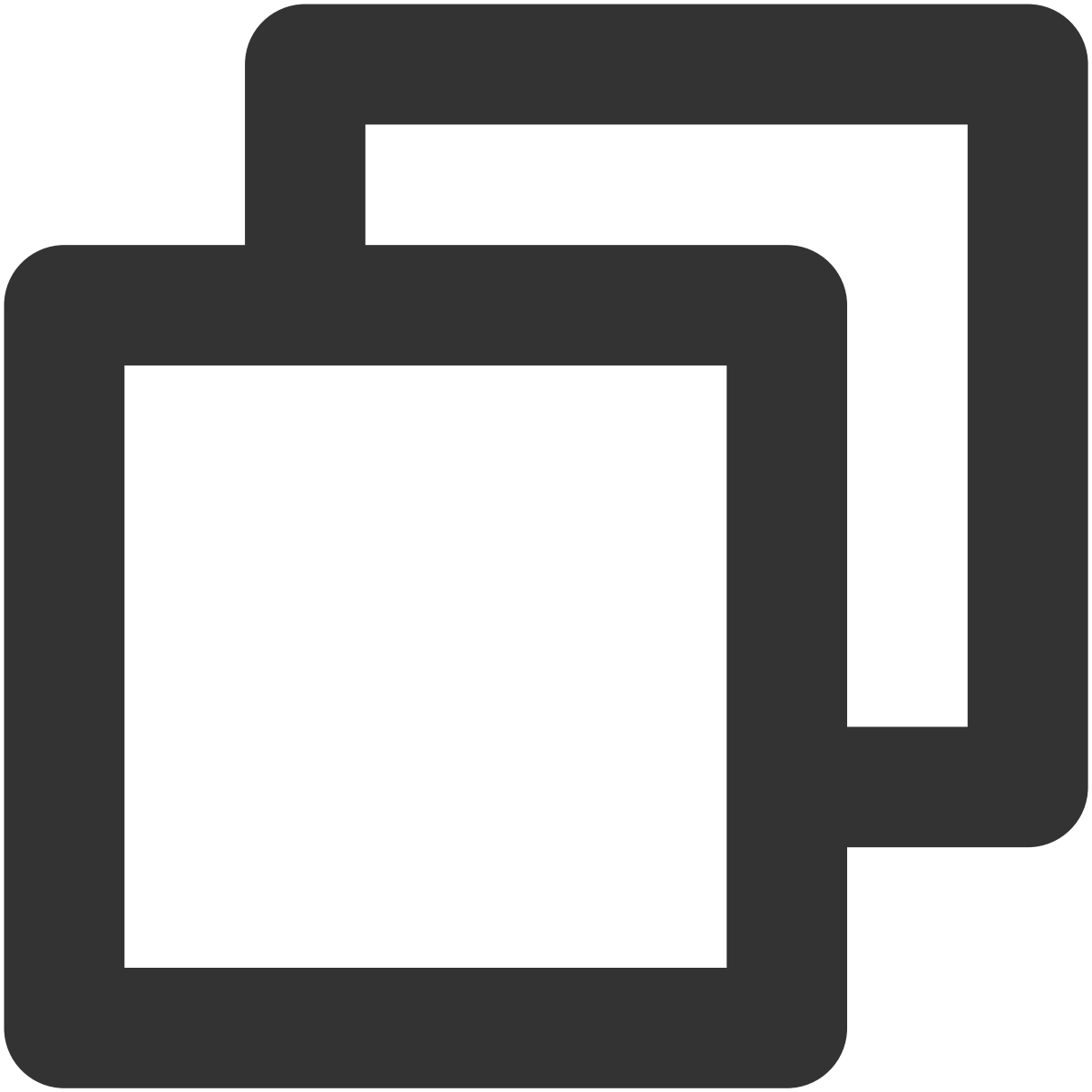
响应



{"picSize":1097476,"picType":"PNG","suspicious":true,"suspiciousBeginByte":120,"sus

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。

队列接口
查询队列

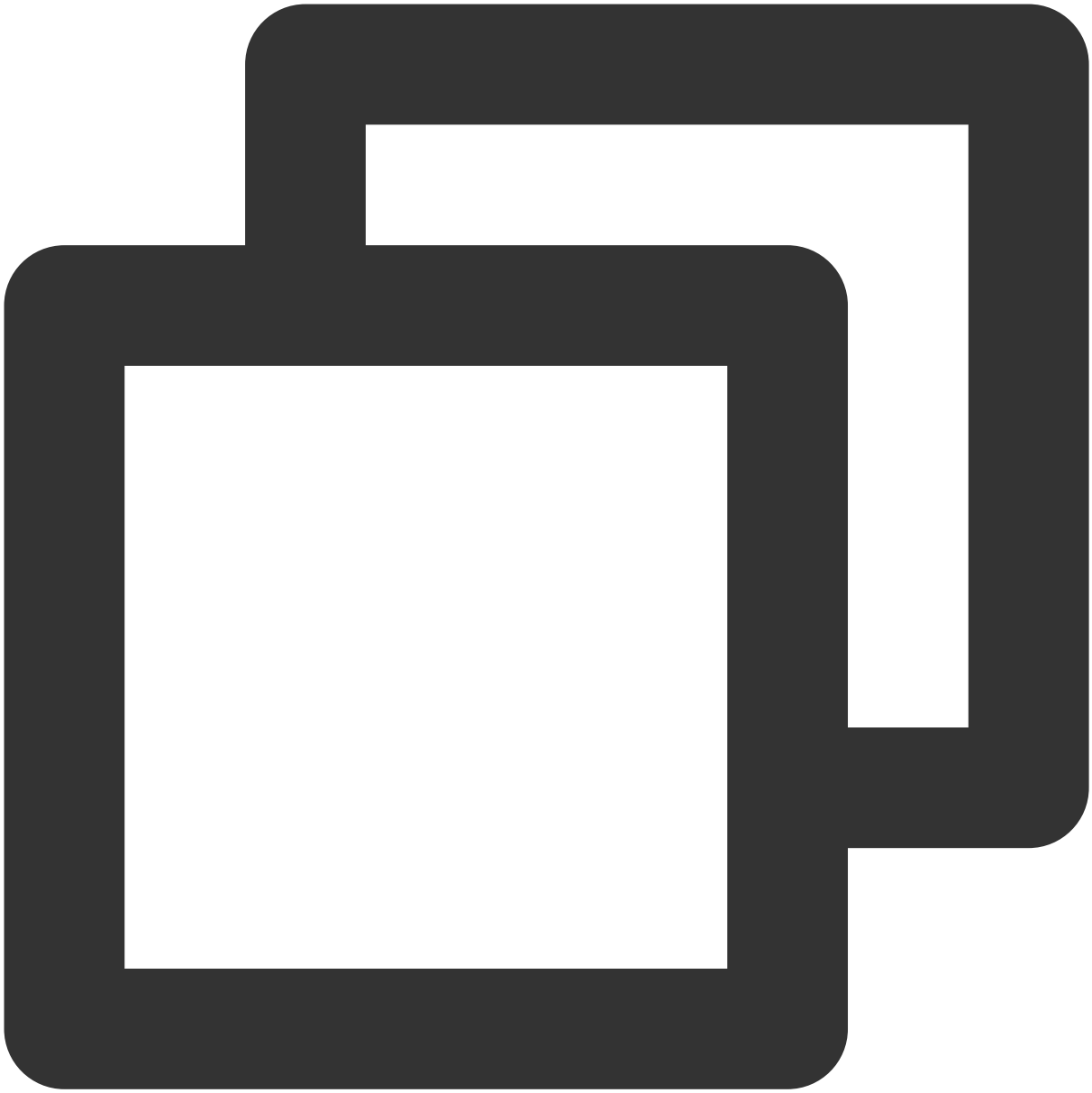
最近更新时间：2024-01-18 15:11:52

功能描述

搜索图片处理队列。

请求

请求示例



```
GET /picqueue 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
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

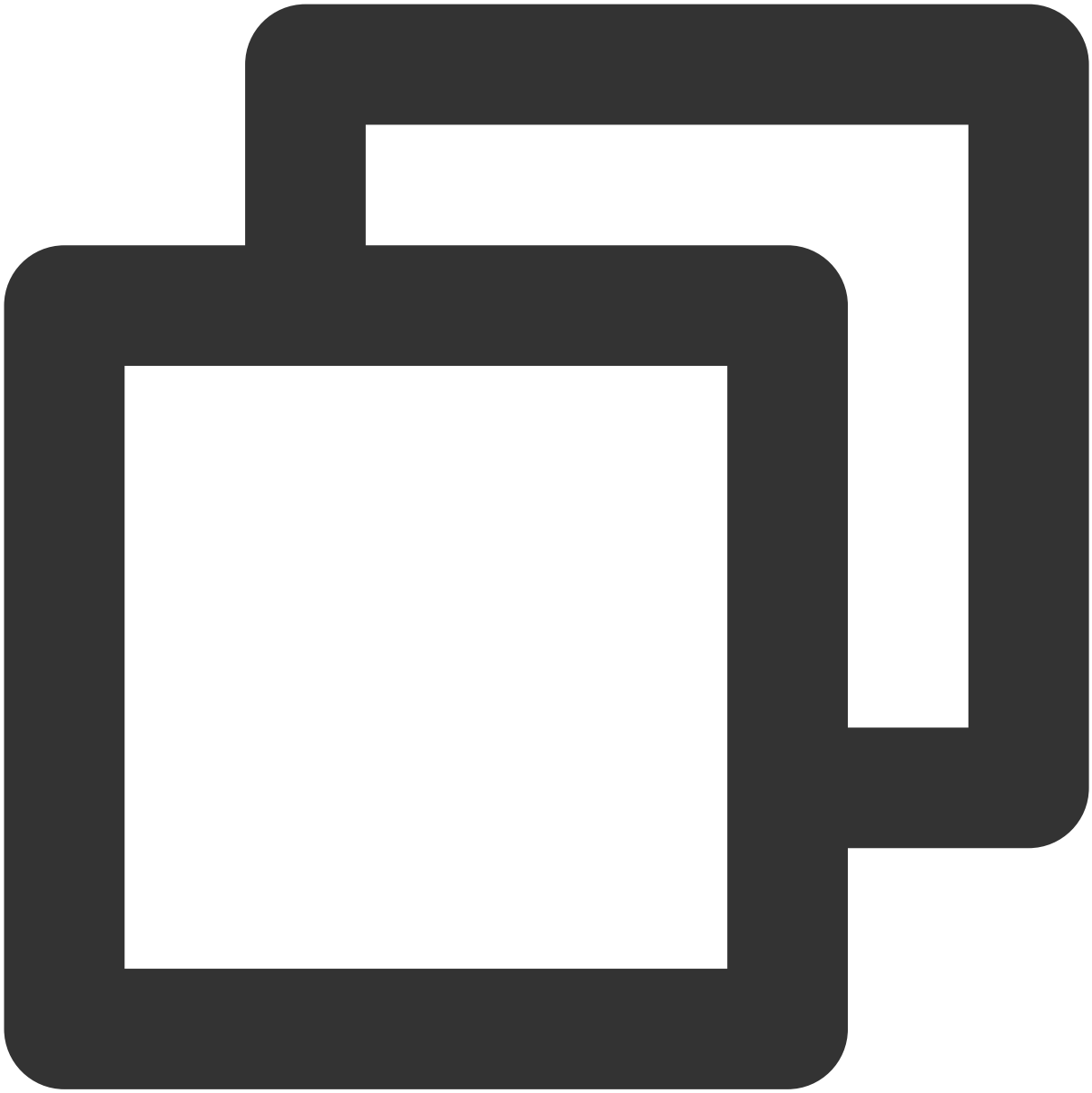
响应

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>p8366c460c67b4c65a9bf81b41eaa3c4b</QueueId>
        <Name>queue-pic-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>PicProcessing</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

版权所有：腾讯云计算（北京）有限责任公司

MaxSize	Response.QueueList	队列最大长度	Int
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：

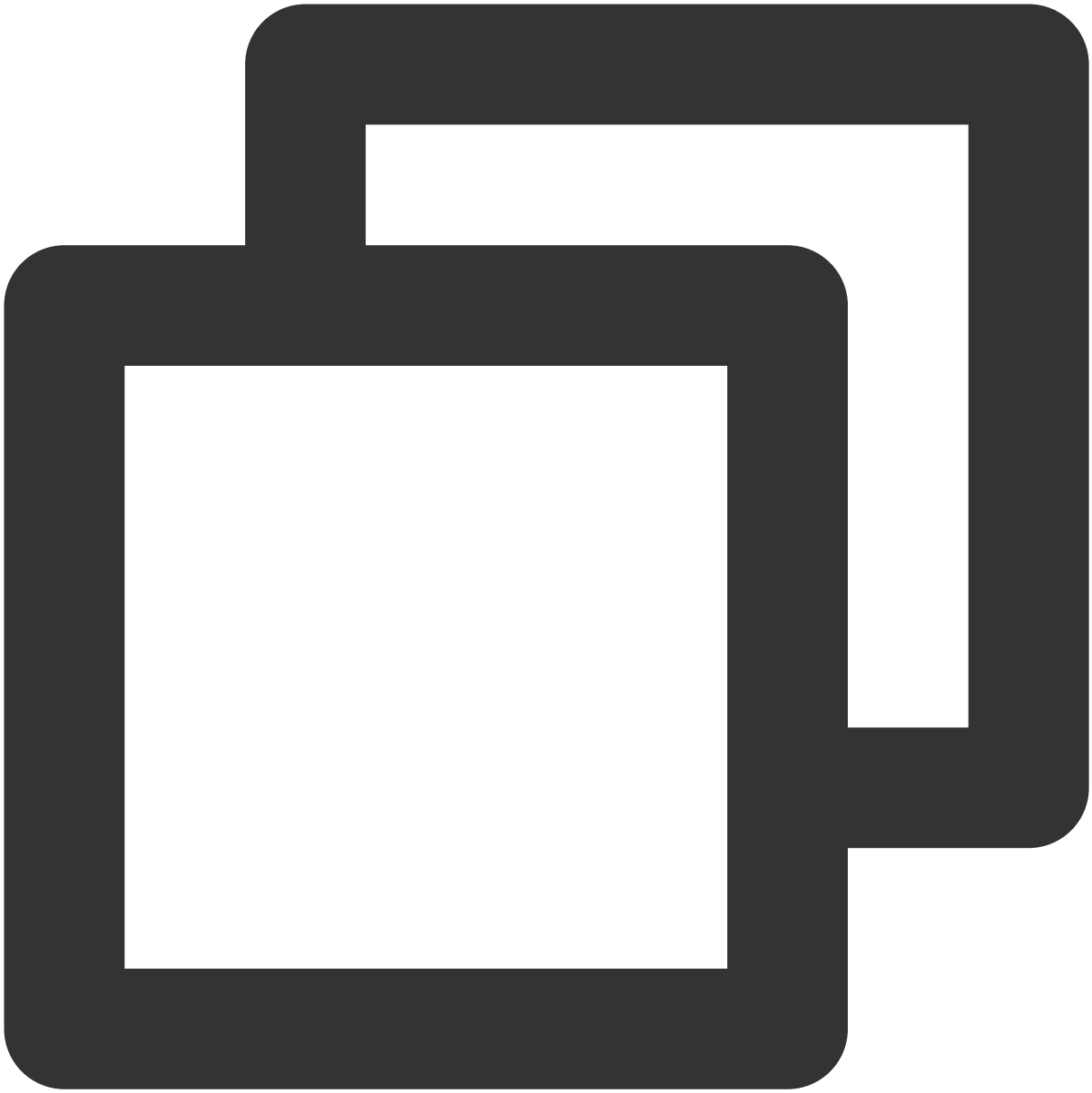
节点名称（关键字）	父节点	描述	类型
Url	Response.QueueList.NotifyConfig	回调地址	String
State	Response.QueueList.NotifyConfig	开关状态，On 或者 Off	String
Type	Response.QueueList.NotifyConfig	回调类型，Url	String
Event	Response.QueueList.NotifyConfig	任务完成：TaskFinish；工作流完成：WorkflowFinish	String
ResultFormat	Response.QueueList.NotifyConfig	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。

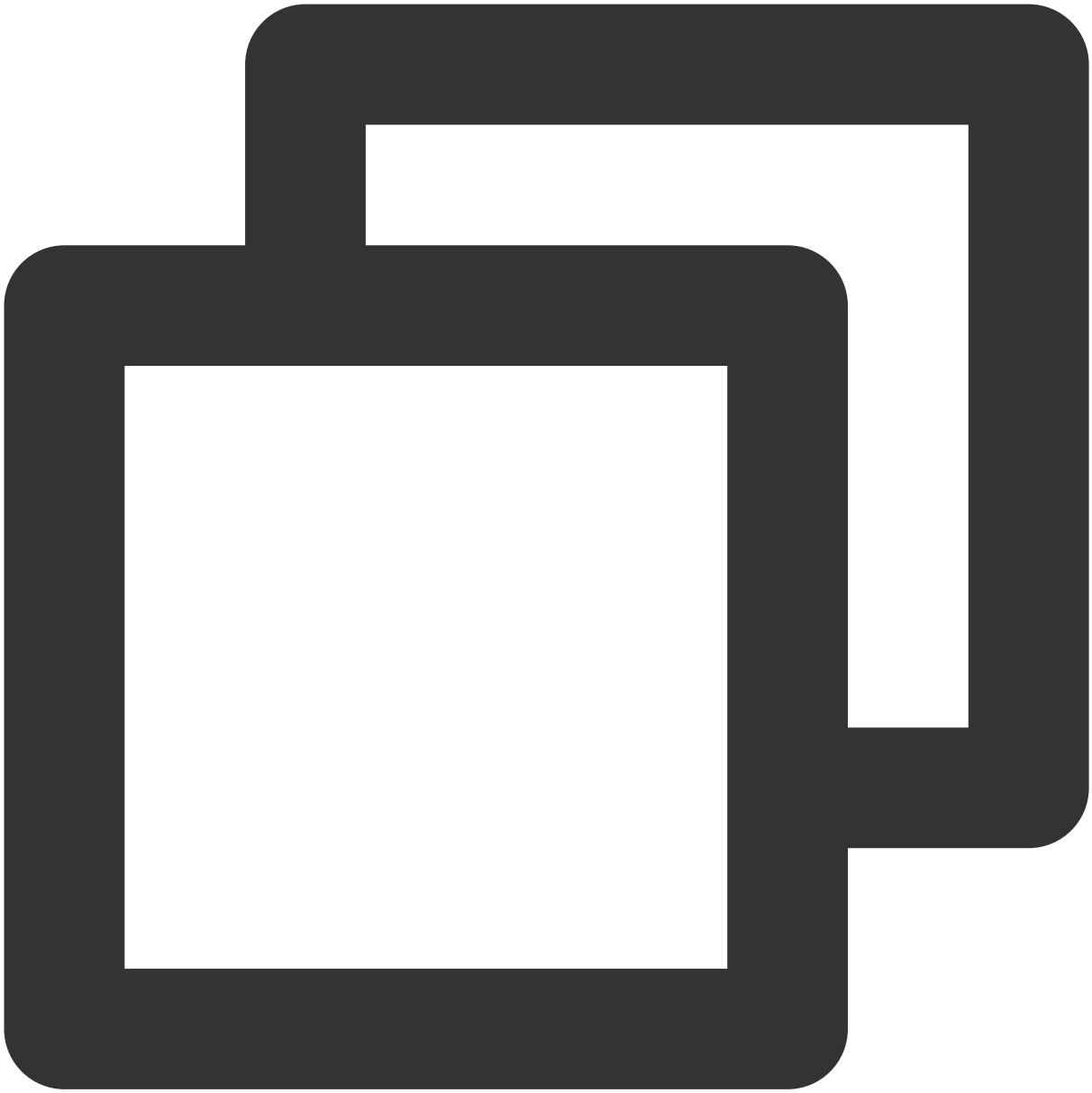
实际案例

请求1：搜索指定队列



```
GET /picqueue?queueIds=pb926e1dce3ed4e47a4f08692164f4869,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

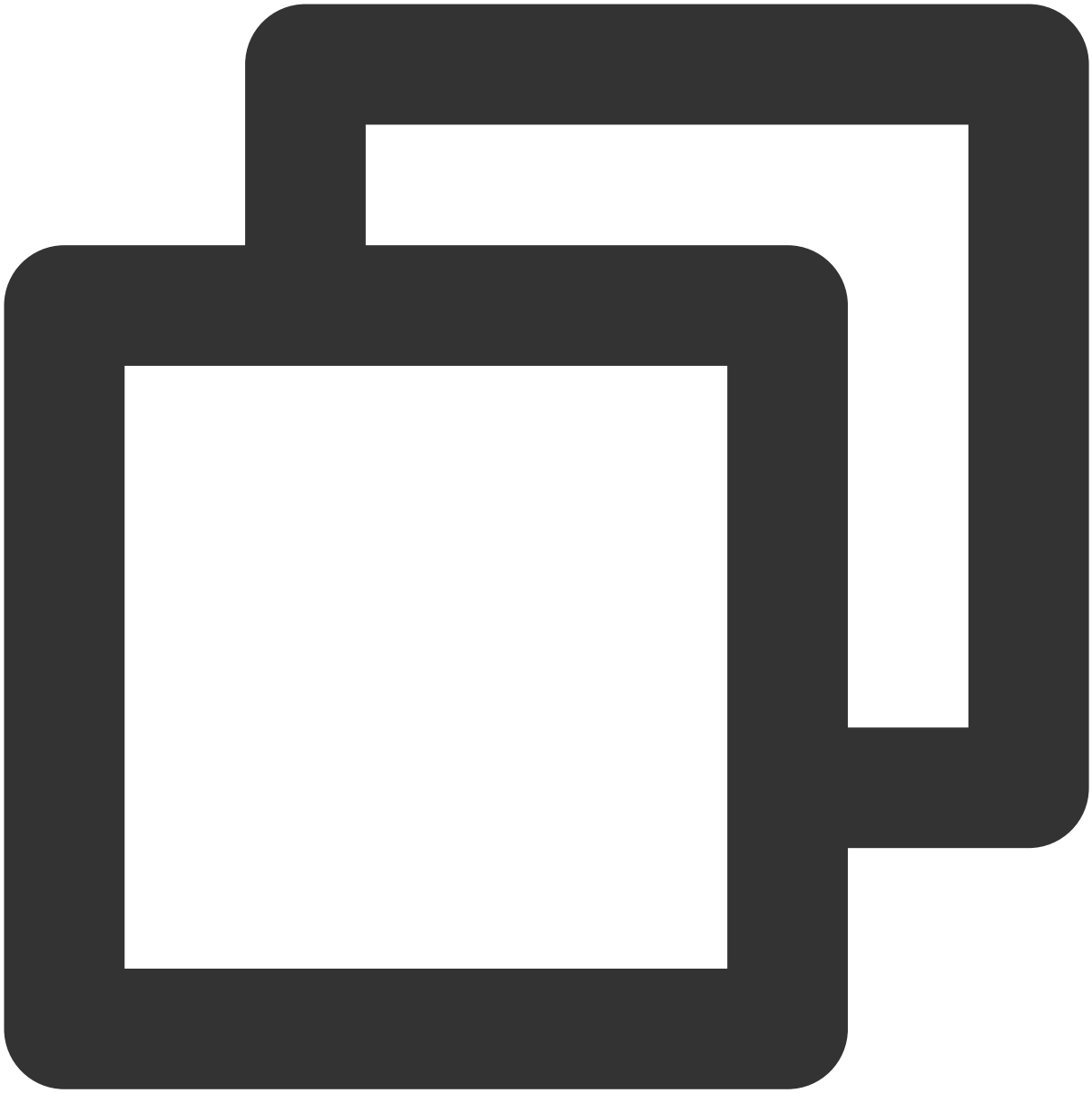


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x

<Response>
    <RequestId>NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x</RequestId>
    <QueueList>
```

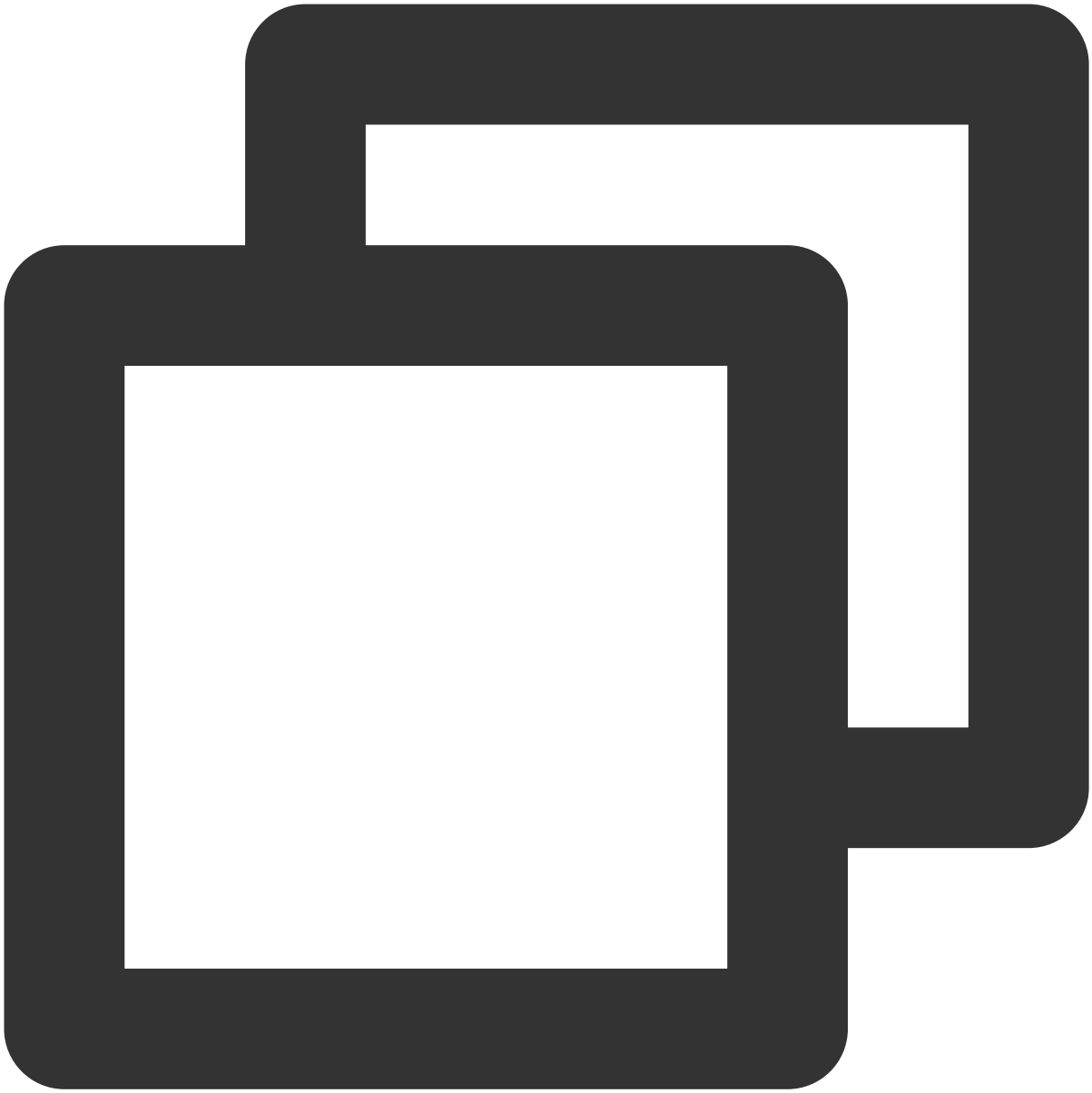
```
        <QueueId>p8366c460c67b4c65a9bf81b41eaa3c4b</QueueId>
        <Name>queue-pic-process-1</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <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>PicProcessing</Category>
    </QueueList>
    <NonExistPIDs>A</NonExistPIDs>
    <NonExistPIDs>B</NonExistPIDs>
</Response>
```

请求2：查询队列列表



```
GET /picqueue?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>p8366c460c67b4c65a9bf81b41eaa3c4b</QueueId>
        <Name>queue-pic-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>PicProcessing</Category>
    </QueueList>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新队列

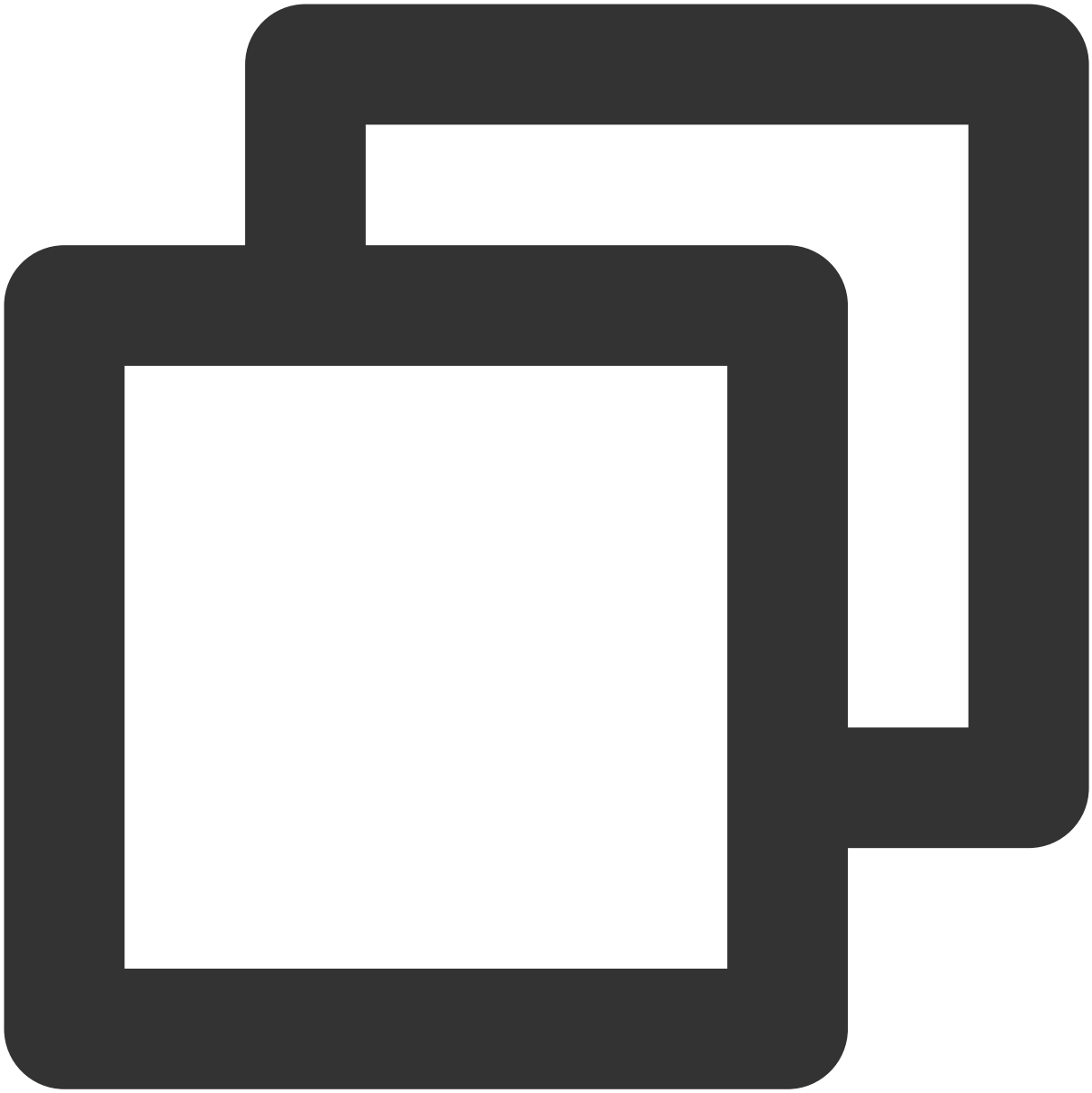
最近更新时间：2024-01-18 15:11:52

功能描述

更新图片处理队列。

请求

请求示例



```
PUT /picqueue/<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：

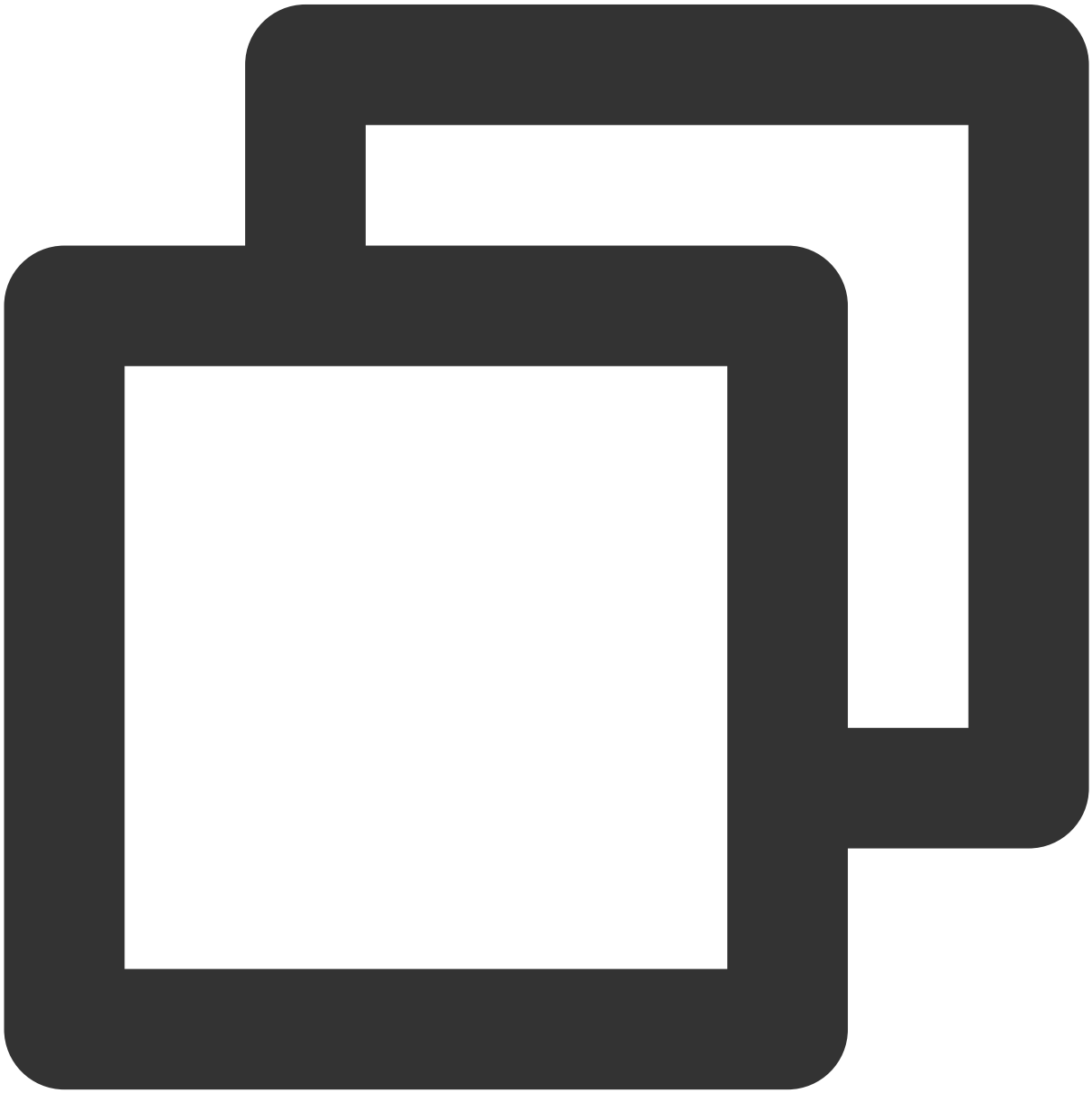
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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-Pic</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	On/Off
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
Type	Request.NotifyConfig	回调类型	String	当 State=On 时, 必选	无	Url 或 TDMQ
Url	Request.NotifyConfig	回调地址	String	当 State=On, 且Type=Url 时, 必选	无	不能为内网地址
MqMode	Request.NotifyConfig	TDMQ 使用模式	String	当 State=On, 且 Type=TDMQ 时, 必选	Queue	主题订阅：Topic 队列服务: Queue
MqRegion	Request.NotifyConfig	TDMQ 所属园区	String	当 State=On, 且 Type=TDMQ 时, 必选	无	目前支持园区 sh（上海）、bj（北京）、gz（广州）、cd（成都）、hk（香港）
MqName	Request.NotifyConfig	TDMQ 主题名称	String	当 State=On, 且 Type=TDMQ 时, 必选	无	无

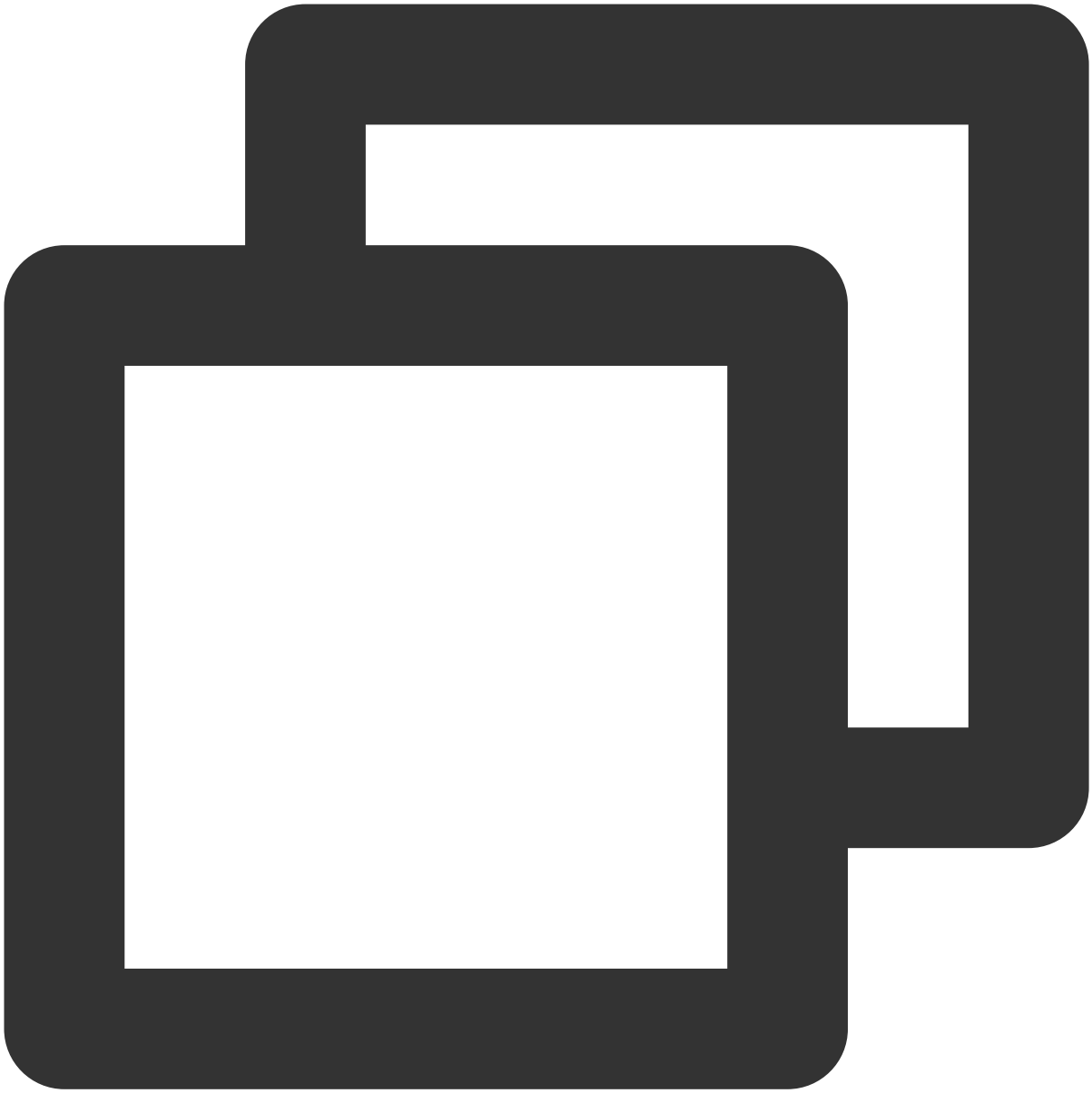
响应

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>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <Queue>
        <QueueId>pb926e1dce3ed4e47a4f08692164f4869</QueueId>
        <Name>My-Queue-Pic</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：

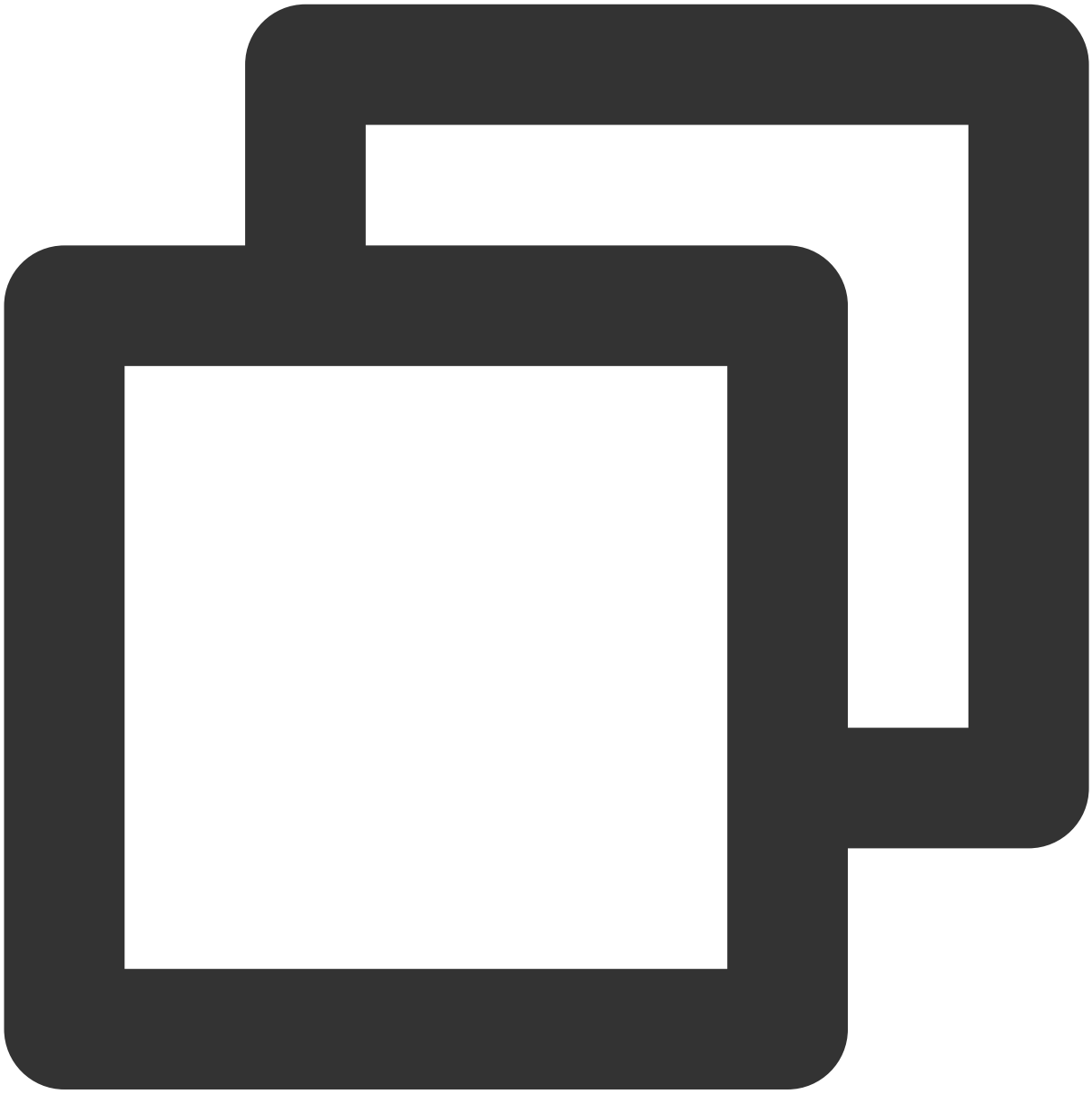
节点名称（关键字）	父节点	描述	类型
QueueId	Response.Queue	队列 ID	String
Name	Response.Queue	队列名字	String
State	Response.Queue	当前状态，Active 或者 Paused	String
NotifyConfig	Response.Queue	回调配置,同 Request.NotifyConfig	Container
MaxSize	Response.Queue	队列最大长度	Int
MaxConcurrent	Response.Queue	当前队列最大并行执行的任务数	Int
Category	Response.Queue	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

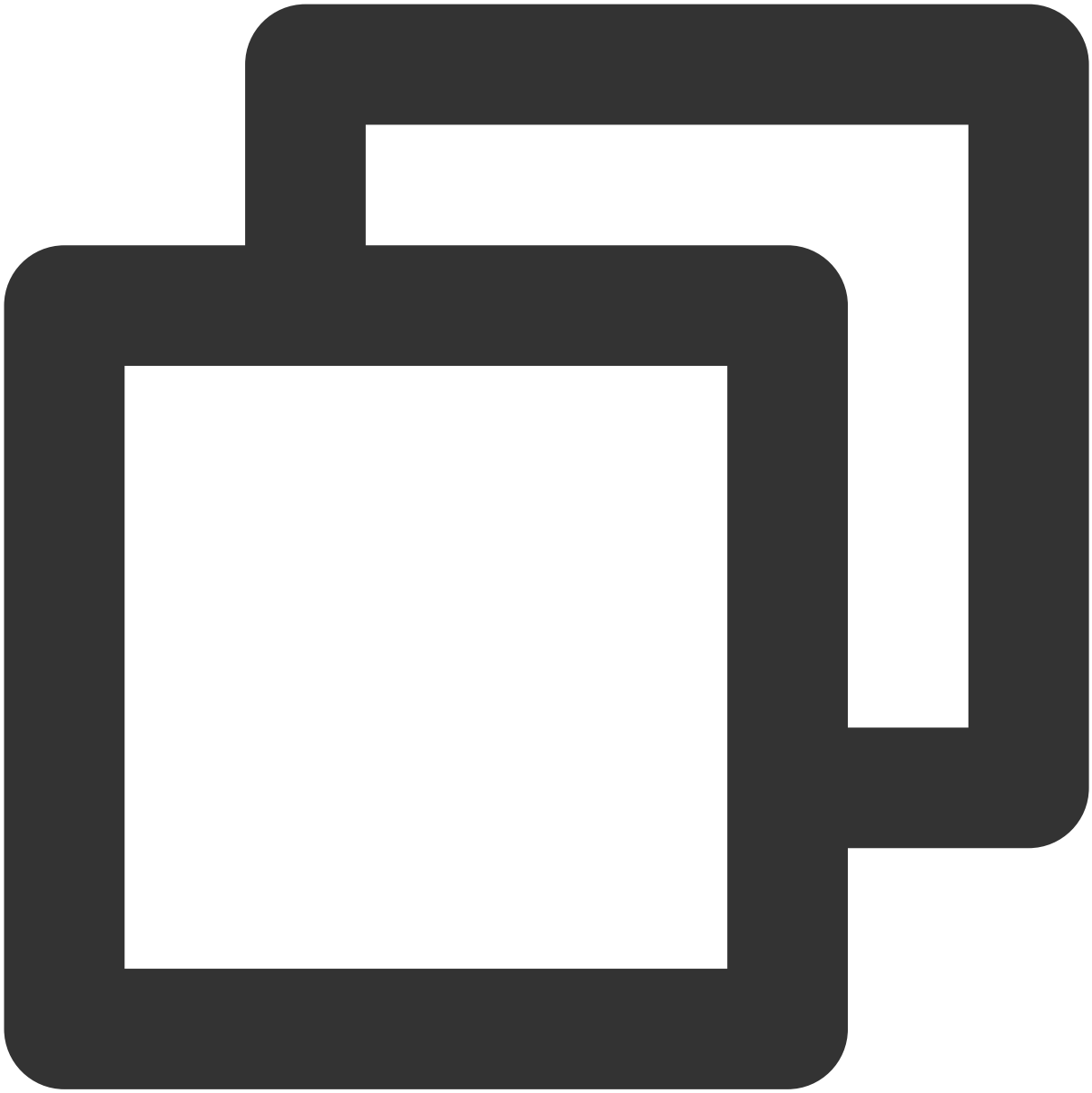


```
PUT /picqueue/pb926e1dce3ed4e47a4f08692164f4869 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

```
Content-Length: 162
Content-Type: application/xml

<Request>
    <Name>My-Queue-Pic</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>pb926e1dce3ed4e47a4f08692164f4869</QueueId>
        <Name>My-Queue-Pic</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>
```

AI 内容识别
人脸识别
人脸特效

最近更新时间：2024-01-20 09:22:44

功能描述

人脸特效支持人脸美颜、人脸性别变换、人脸年龄变化、人像分割的特效功能，适用于社交娱乐、广告营销、互动传播等场景。

功能	说明
人脸美颜	可用于自拍照片等人像美化场景，上传照片即可一键智能实现美白、磨皮、瘦脸、大眼、美型，支持自定义调整参数，帮助提升个人形象。
人脸性别变换	用户上传一张人脸图片，基于人脸编辑与生成算法，输出一张人脸性别转换的图片。男变女可实现美颜、淡妆、加刘海和长发的效果；女变男可实现加毛须、变短发的效果。
人脸年龄变化	用户上传一张人脸图片，基于人脸编辑与生成算法，输出一张人脸变老或变年轻的图片，支持实现人脸不同年龄的变化。
人像分割	指对图片中的人体轮廓范围进行识别，将其与背景进行分离，返回分割后的二值图、灰度图、前景人像图。可应用于人像抠图、照片合成、人像特效等场景，极大提升工具效率。

说明：

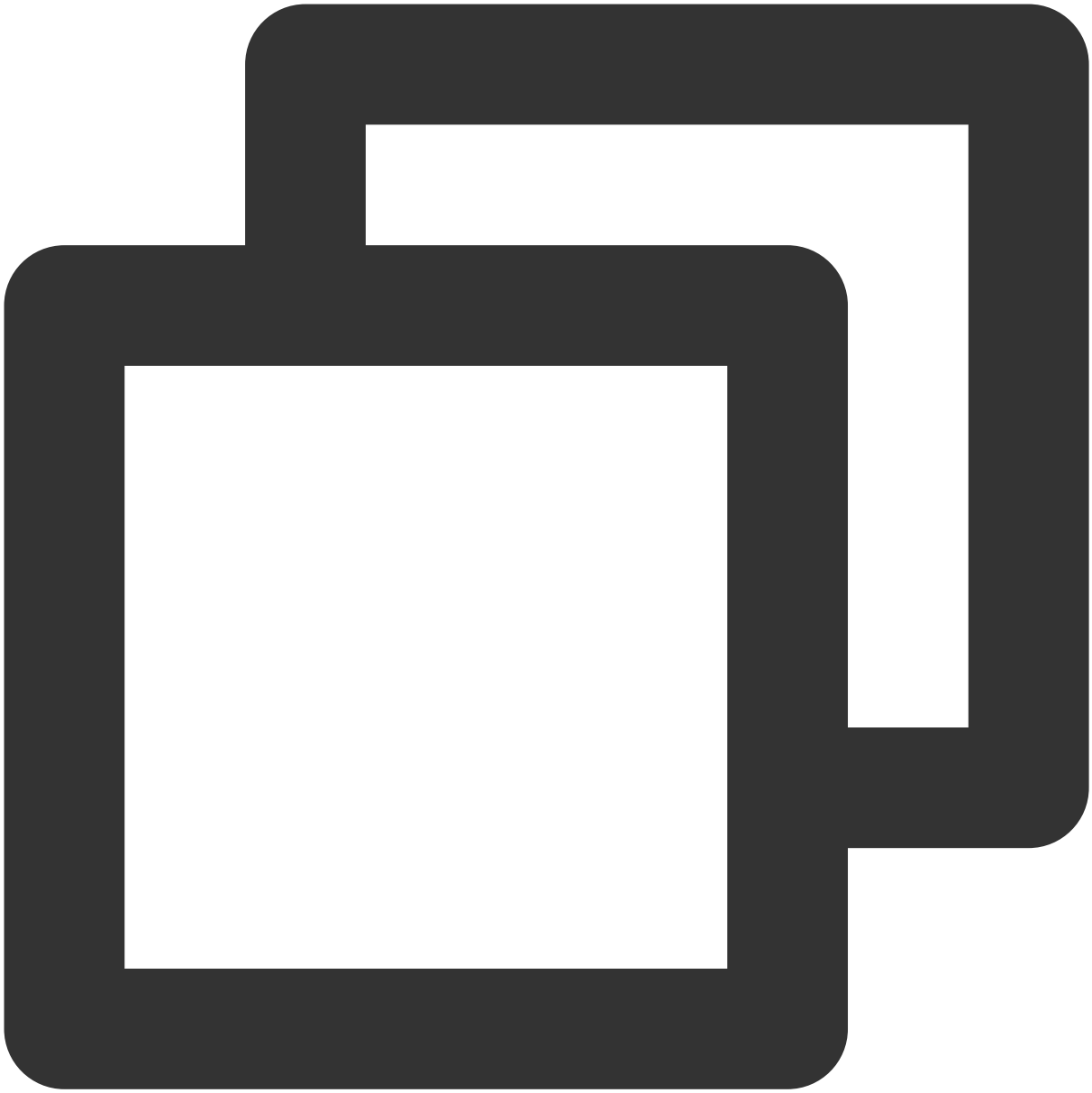
当前人脸特效的图片限制为：图片 base64 编码后大小不可超过5MB。支持 PNG、JPG、JPEG、BMP 格式，不支持 GIF 图片。当使用人像分割时，图片分辨率须小于2000 × 2000px。

人脸特效为付费服务，计费详情请查看内容识别费用。

该功能目前仅提供 API 使用方式。

请求

请求示例



```
GET /<ObjectKey>?ci-process=face-effect&type=<type> 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。

请求参数

参数名称	描述	类型	是否必选
ObjectKey	对象文件名，例如：folder/document.jpg。	String	是
ci-process	数据万象处理能力，人脸特效固定为 face-effect。	String	是
type	人脸特效类型。人脸美颜：face-beautify；人脸性别转换：face-gender-transformation；人脸年龄变化：face-age-transformation；人像分割：face-segmentation。	String	是
whitening	type 为 face-beautify 时生效，美白程度，取值范围[0,100]。0表示不美白，100表示最高程度。默认值30。	Integer	否
smoothing	type 为 face-beautify 时生效，磨皮程度，取值范围[0,100]。0表示不磨皮，100表示最高程度。默认值10。	Integer	否
faceLifting	type 为 face-beautify 时生效，瘦脸程度，取值范围[0,100]。0表示不瘦脸，100表示最高程度。默认值70。	Integer	否
eyeEnlarging	type 为 face-beautify 时生效，大眼程度，取值范围[0,100]。0表示不大眼，100表示最高程度。默认值70。	Integer	否
gender	type 为 face-gender-transformation 时生效，选择转换方向，0表示男变女，1表示女变男。无默认值。注意：仅对图片中面积最大的人脸进行转换。	Integer	是
age	type 为 face-age-transformation 时生效，变化到的人脸年龄，范围为[10,80]，无默认值。注意：仅对图片中面积最大的人脸进行转换。	Integer	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

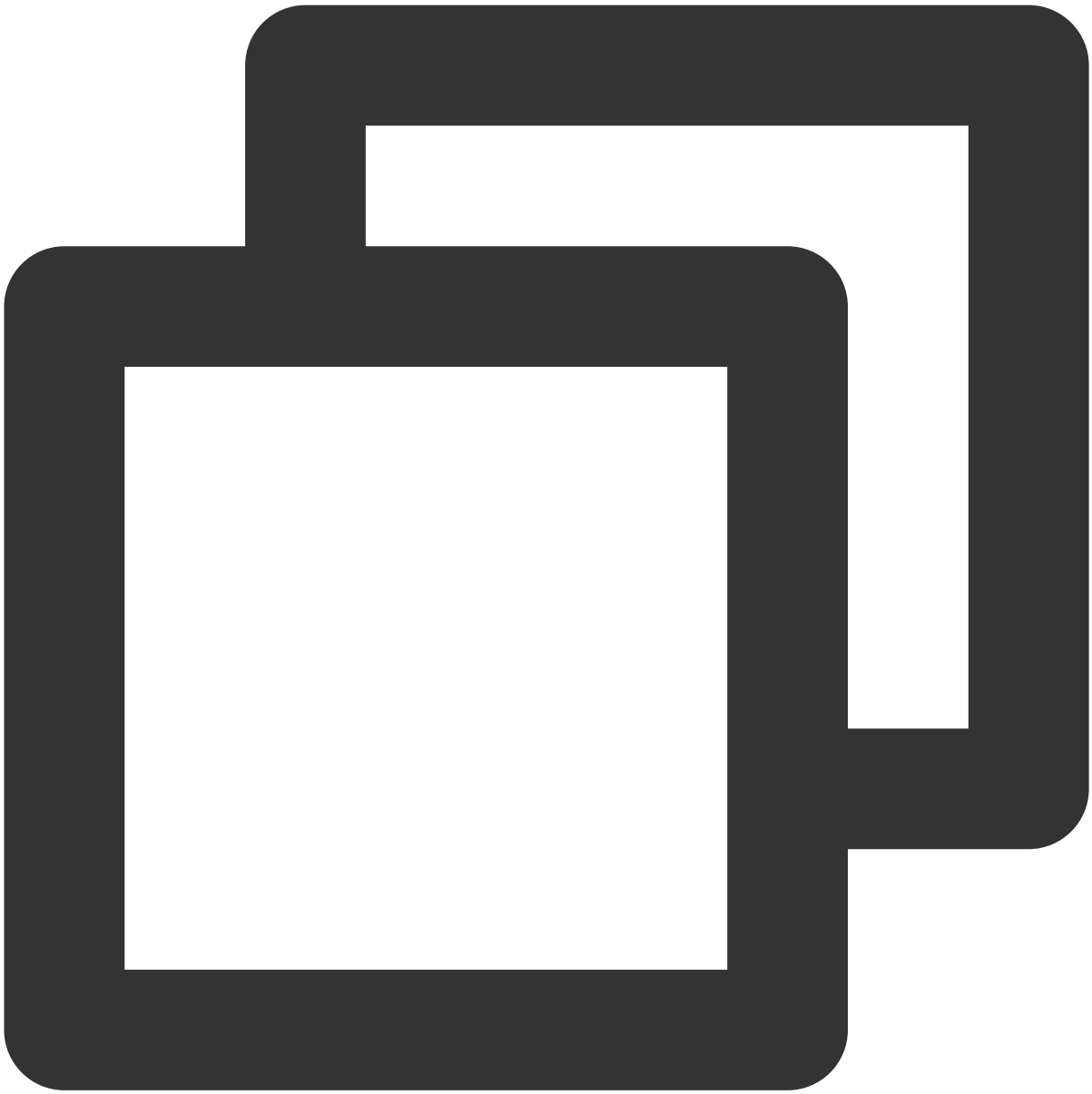
响应头

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

版权所有：腾讯云计算（北京）有限责任公司

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>
  <ResultImage>base64编码的图片</ResultImage>
  <ResultMask>base64编码的文件</ResultMask>
</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 的内容：

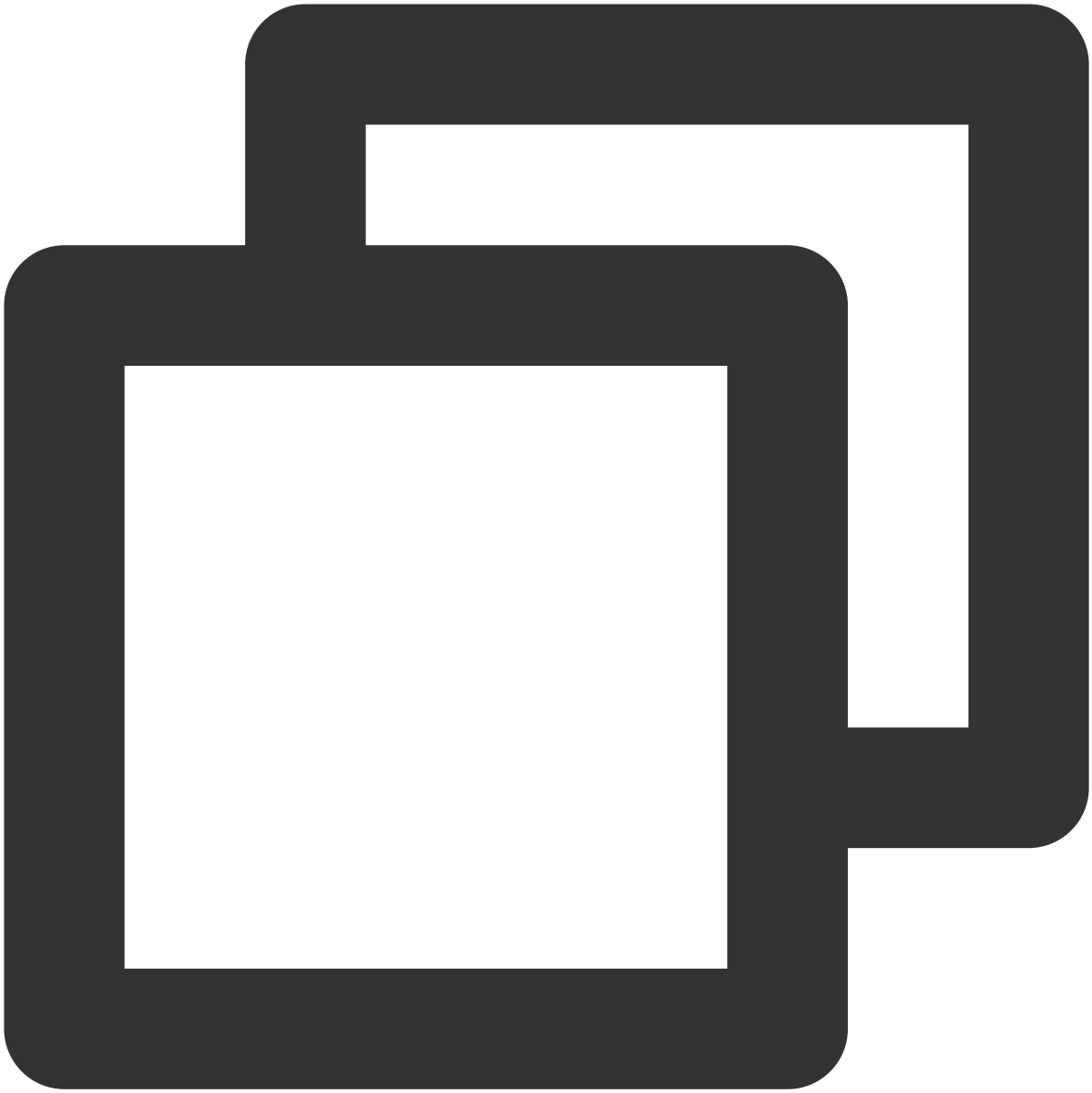
节点名称（关键字）	父节点	描述	类型
ResultImage	Response	处理后的图片 base64 数据。	String
ResultMask	Response	人像分割输出参数，一个通过 Base64 编码的文件，解码后文件由 Float 型浮点数组成。这些浮点数代表原图从左上角开始的每一行的每一个像素点，每一个浮点数的值是原图相应像素点位于人体轮廓内的置信度（0-1）转化的灰度值（0-255）。	String

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。

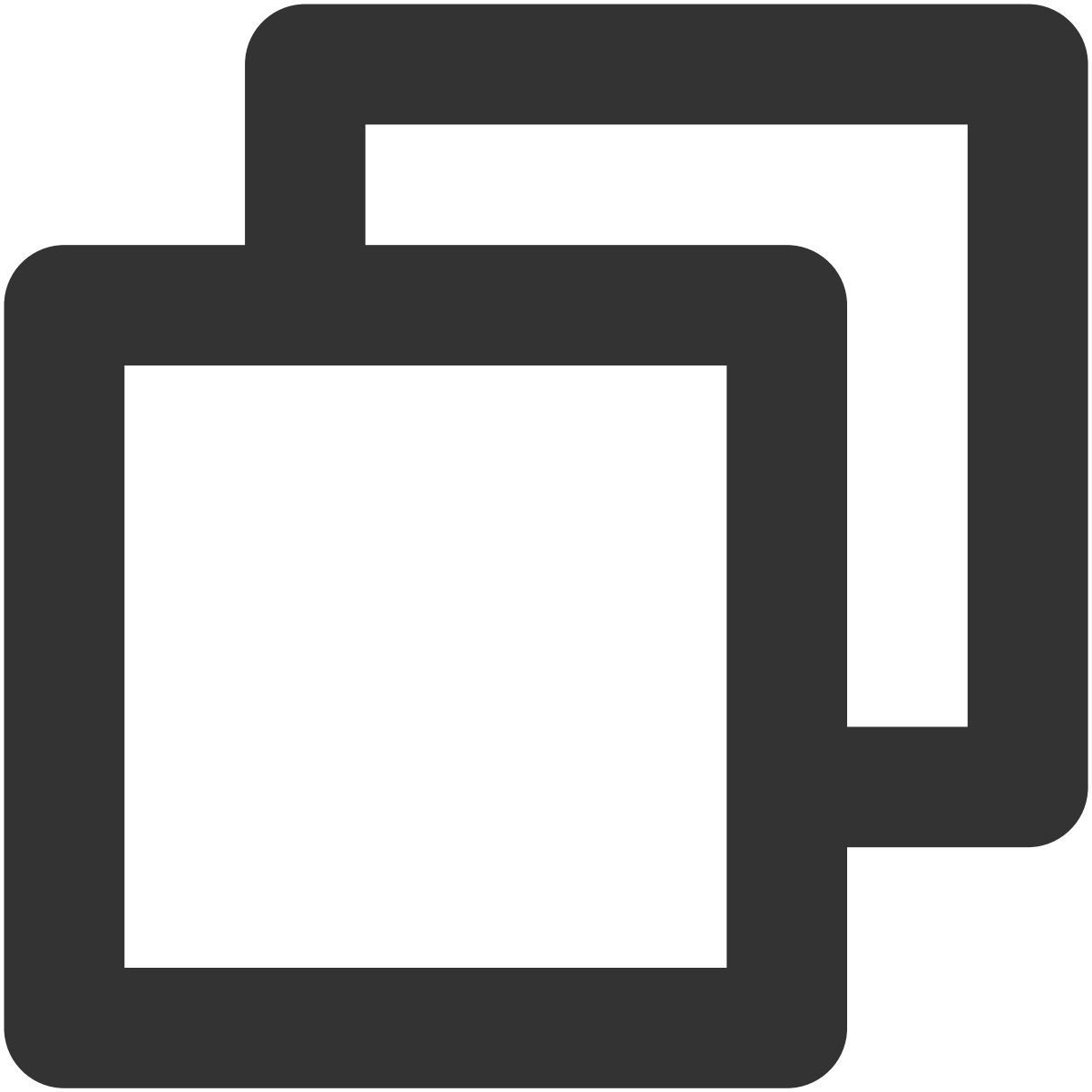
实际案例

请求



```
GET /test.jpg?ci-process=face-effect&type=face-beautify&whitening=70&smoothing=80&f
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 414641
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-image
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <ResultImage>base64编码的图片</ResultImage>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

人脸核身
身份证识别

最近更新时间：2024-01-20 09:22:44

功能描述

支持中国大陆居民二代身份证正反面所有字段的识别，包括姓名、性别、民族、出生日期、住址、公民身份证号、签发机关、有效期限；具备身份证照片、人像照片的裁剪功能和翻拍、PS、复印件告警功能，以及边框和框内遮挡告警、临时身份证告警和身份证有效期不合法告警等扩展功能。

说明：

当前身份证识别的图片限制为：要求图片经 Base64 编码后不超过7M，分辨率建议500 × 800以上，支持 PNG、JPG、JPEG、BMP 格式。建议卡片部分占据图片2/3以上。

身份证识别为付费服务，计费详情请参见内容识别费用。

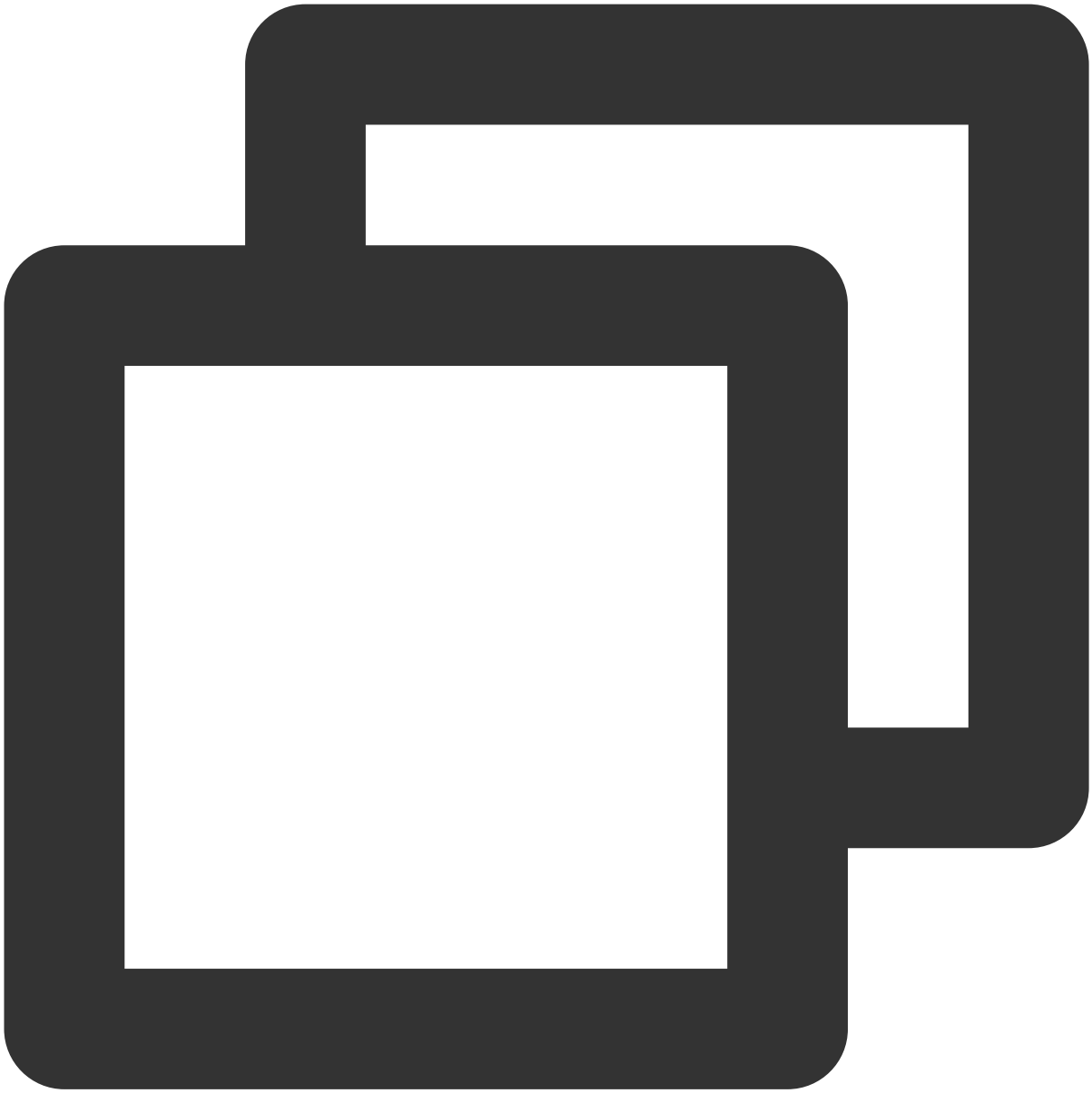
该功能目前仅提供 API 使用方式。

请求

上传时处理

身份证图片上传时识别的请求包与 COS 简单上传文件接口一致，只需增加相关处理参数，即可在图片上传时返回识别结果，并可将原图存入到 COS。

版权所有：腾讯云计算（北京）有限责任公司



```
PUT /<ObjectKey>?ci-process=IDCardOCR&CardSide=<CardSide>&Config=<Config> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

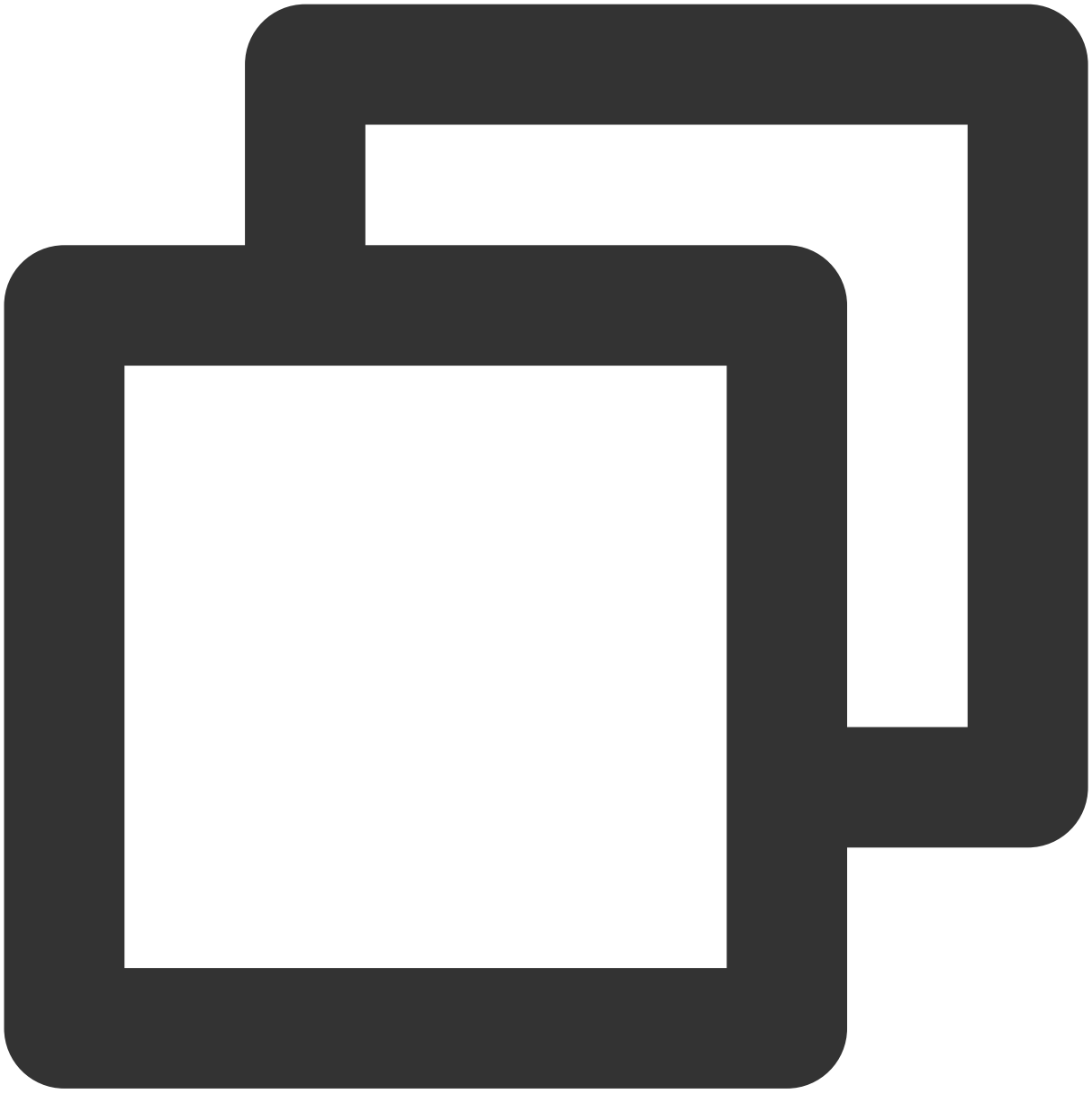
说明：

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。

云上数据处理

云上数据处理请求能够对已存储在 COS 的身份证图片进行相应处理操作，返回识别结果。



```
GET /<ObjectKey>?ci-process=IDCardOCR&CardSide=<CardSide>&Config=<Config> 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。

请求参数

参数名称	描述	类型	是否必选
ObjectKey	对象文件名，例如：folder/document.jpg	String	是
ci-process	数据万象处理能力，身份证识别固定为 IDCardOCR	String	是
CardSide	FRONT：身份证有照片的一面（人像面） BACK：身份证有国徽的一面（国徽面） 该参数如果不填，将为您自动判断身份证正反面	String	否
Config	以下可选字段均为 bool 类型，默认 false： CropIdCard，身份证照片裁剪（去掉证件外多余的边缘、自动矫正拍摄角度） CropPortrait，人像照片裁剪（自动抠取身份证头像区域） CopyWarn，复印件告警 BorderCheckWarn，边框和框内遮挡告警 ReshootWarn，翻拍告警 DetectPsWarn，PS 检测告警 TempIdWarn，临时身份证告警 InvalidDateWarn，身份证有效日期不合法告警 Quality，图片质量分数（评价图片的模糊程度） MultiCardDetect，是否开启多卡证检测 参数设置方式参考： Config = {"CropIdCard":true,"CropPortrait":true}	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。

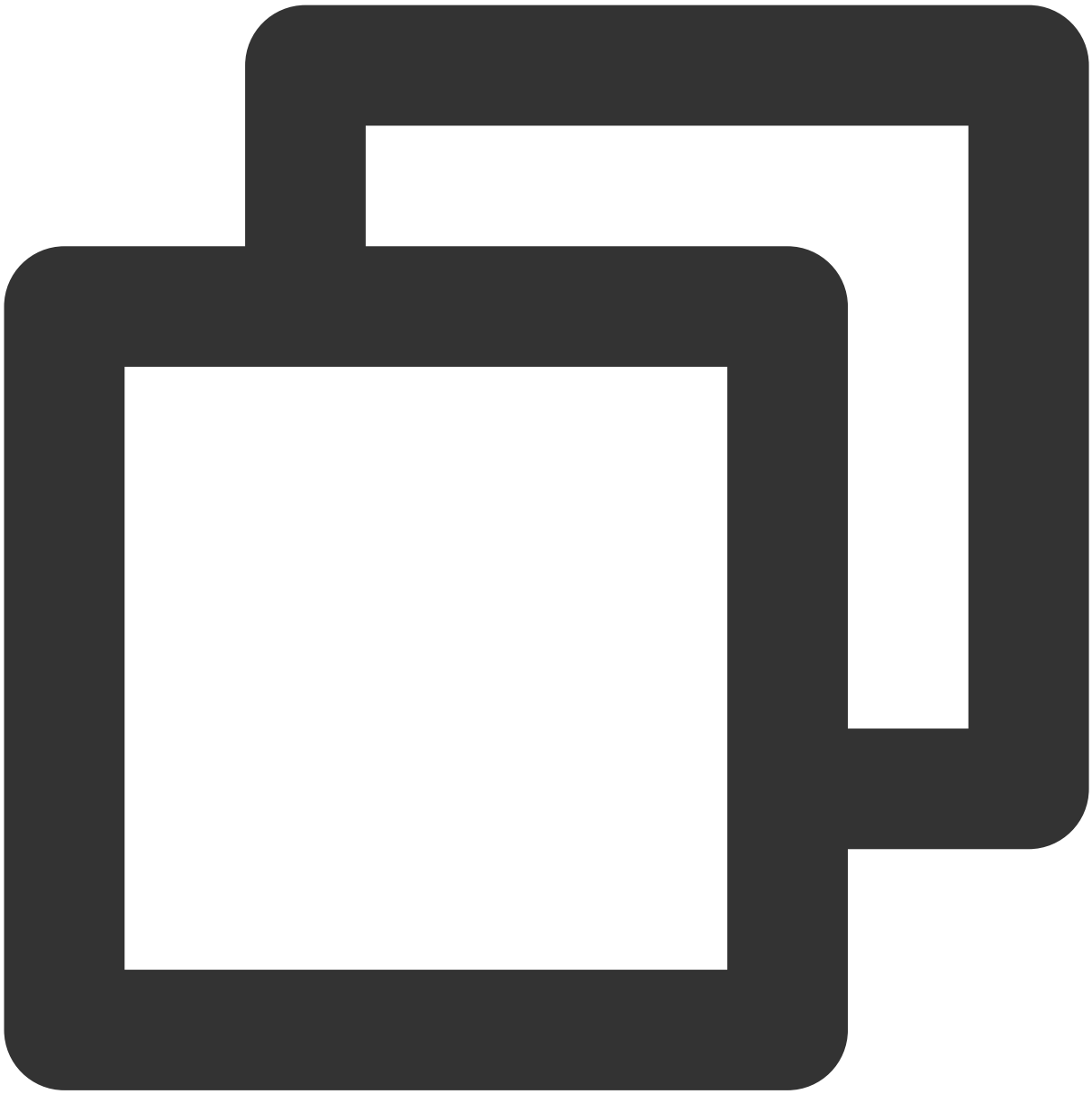
响应

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>
  <IdInfo>
    身份证识别信息
  </IdInfo>
  <AdvancedInfo>
    身份证识别信息
  </AdvancedInfo>
</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 的内容：

节点名称（关键字）	父节点	描述	类型
IdInfo	Response	身份证识别信息	Container
AdvancedInfo	Response	扩展信息，不请求则不返回	Container

IdInfo身份证识别信息中包含如下内容：

节点名称（关键字）	父节点	描述	类型
Name	IdInfo	姓名（人像面）	String
Sex	IdInfo	性别（人像面）	String
Nation	IdInfo	民族（人像面）	String
Birth	IdInfo	出生日期（人像面）	String
Address	IdInfo	地址（人像面）	String
IdNum	IdInfo	身份证号（人像面）	String
Authority	IdInfo	发证机关（国徽面）	String
ValidDate	IdInfo	证件有效期（国徽面）	String

AdvancedInfo扩展信息信息中包含如下内容：

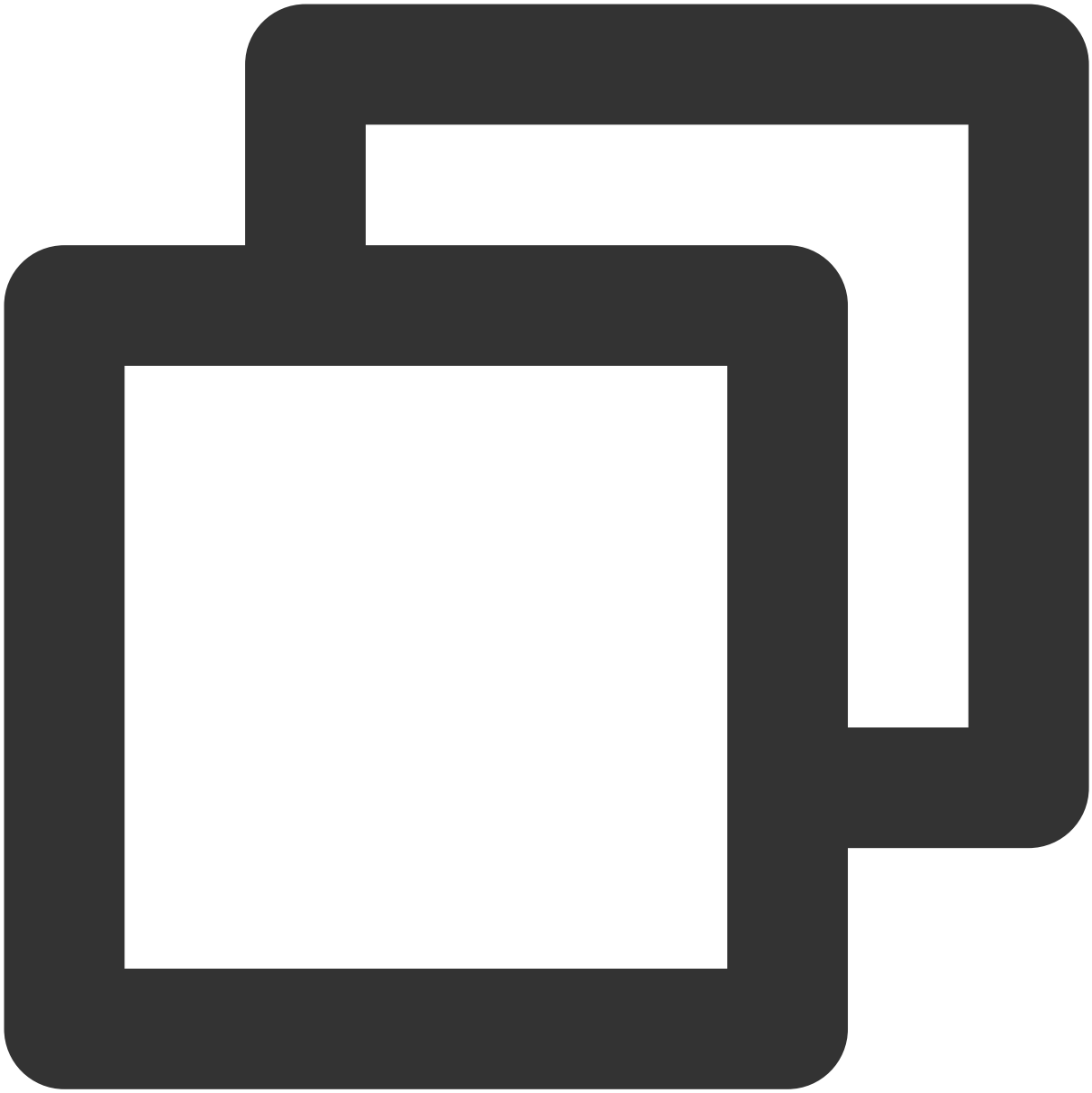
节点名称（关键字）	父节点	描述	类型
IdCard	AdvancedInfo	裁剪后身份证照片的 Base64 编码，设置 Config.CropIdCard 为 true 时返回	String
Portrait	AdvancedInfo	身份证头像照片的 Base64 编码，设置 Config.CropPortrait 为 true 时返回	String
Quality	AdvancedInfo	图片质量分数，设置 Config.Quality 为 true 时返回（取值范围：0~100，分数越低越模糊，建议阈值≥50）	String

版权所有：腾讯云计算（北京）有限责任公司

BorderCodeValue	AdvancedInfo	身份证边框不完整告警阈值分数，设置 Config.BorderCheckWarn 为 true 时返回（取值范围：0~100，分数越低边框遮挡可能性越低，建议阈值≥50）	String
WarnInfos	IdInfo	告警信息，Code 告警码列表和释义： 9100 身份证有效日期不合法告警 9101 身份证边框不完整告警 9102 身份证复印件告警 9103 身份证翻拍告警 9104 临时身份证告警 9105 身份证框内遮挡告警 9106 身份证 PS 告警 可能存在多个 WarnInfos	String

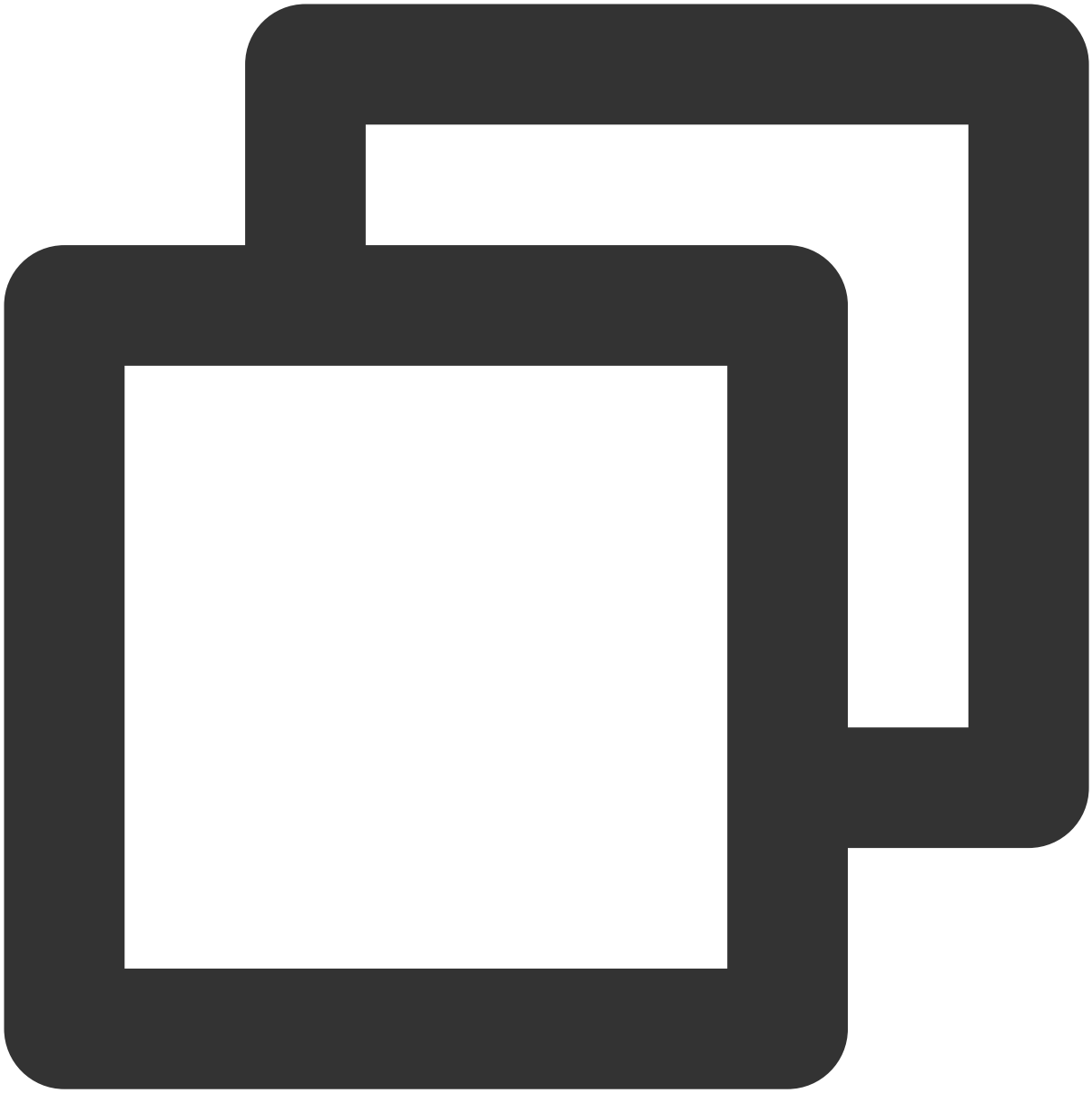
实际案例

请求



```
GET /test.jpg?ci-process=IDCardOCR&CardSide=FRONT&Config={"CropIdCard"：true, "Crop
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 414641
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-image
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <IdInfo>
     <Name>张明</Name>
     <Sex>男</Sex>
```

```
      <Nation>汉</Nation>
      <Birth>1987/1/1</Birth>
      <Address>北京市石景山区高新技术园腾讯大楼</Address>
      <IdNum>440524198701010014</IdNum>
   </IdInfo>
   <AdvancedInfo>
      <IdCard>裁剪后身份证照片的 Base64 编码</IdCard>
      <Portrait>身份证头像照片的 Base64 编码</Portrait>
   </AdvancedInfo>
</Response>
```

错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误信息请参见 错误码 文档。

错误码	描述
FailedOperation.DownLoadError	文件下载失败
FailedOperation.EmptyImageError	图片内容为空
FailedOperation.IdCardInfoIllegal	身份证信息不合法（身份证号、姓名字段校验非法等）
FailedOperation.ImageBlur	图片模糊
FailedOperation.ImageDecodeFailed	图片解码失败
FailedOperation.ImageNoIdCard	图片中未检测到身份证
FailedOperation.ImageSizeTooLarge	图片尺寸过大，请参考输出参数中关于图片大小限制的说明
FailedOperation.MultiCardError	照片中存在多张卡
FailedOperation.OcrFailed	OCR 识别失败
FailedOperation.UnKnowError	未知错误
FailedOperation.UnOpenError	服务未开通
InvalidParameter.ConfigFormatError	Config 格式错误
InvalidParameterValue.InvalidParameterValueLimit	参数值错误
LimitExceeded.TooLargeFileError	文件内容太大

ResourcesSoldOut.ChargeStatusException	计费状态异常

图片文字识别（OCR）
通用文字识别

最近更新时间：2024-01-20 09:22:44

功能描述

通用文字识别功能（Optical Character Recognition，OCR）基于行业前沿的深度学习技术，将图片上的文字内容，智能识别为可编辑的文本，可应用于随手拍扫描、纸质文档电子化、电商广告审核等多种场景，大幅提升信息处理效率。
本接口属于 GET 请求，为同步请求方式。

费用说明

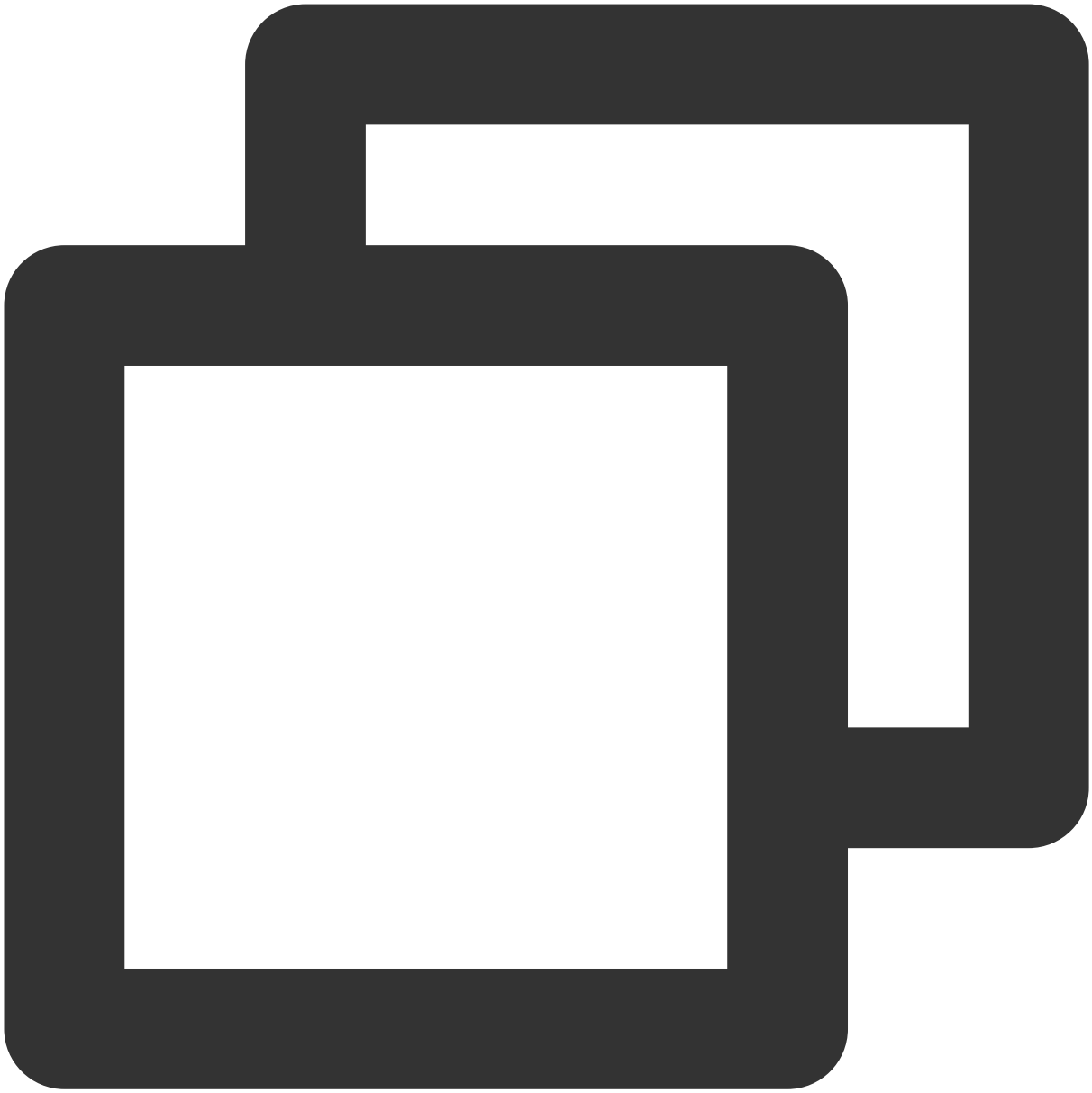
成功调用接口会产生 通用文字识别费用 和 COS 读请求费用。
如果图片属于 COS 上的低频存储类型，成功调用接口会产生 COS 低频数据取回费用。
不支持对 COS 上的归档存储类型和深度归档存储类型的图片进行处理，如果需要处理此类型图片，请先 恢复归档文件。

限制说明

图片支持格式：PNG、JPG、JPEG、BMP 及 PDF 文件。
图片大小：所下载文件经 Base64 编码后**不超过7MB**。
图片分辨率：建议大于**600*800像素**，否则影响识别效果。
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



```
GET /<ObjectKey>?ci-process=OCR&type=general&language-type=zh&ispdf=true&pdf-pagenu
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。

请求参数

参数名称	描述	类型	是否必选
ObjectKey	对象文件名，例如：folder/document.jpg。	String	是
ci-process	数据万象处理能力，图片文字识别固定为 OCR。	String	是
type	OCR 的识别类型，有效值为 general，accurate，efficient，fast，handwriting。general 表示通用印刷体识别；accurate 表示印刷体高精度版；efficient 表示印刷体精简版；fast 表示印刷体高速版；handwriting 表示手写体识别。默认值为 general。	String	否
language-type	type 值为 general 时有效，表示识别语言类型。支持自动识别语言类型，同时支持自选语言种类，默认中英文混合(zh)，各种语言均支持与英文混合的文字识别。可选值请参见 可识别的语言类型。	String	否
ispdf	type 值为 general、fast 时有效，表示是否开启 PDF 识别，有效值为 true 和 false，默认值为 false，开启后可同时支持图片和 PDF 的识别。	Boolean	否
pdf-pagenumber	type 值为 general、fast 时有效，表示需要识别的 PDF 页面的对应页码，仅支持 PDF 单页识别，当上传文件为 PDF 且 ispdf 参数值为 true 时有效，默认值为1。	Integer	否
isword	type 值为 general、accurate 时有效，表示识别后是否需要返回单字信息，有效值为 true 和 false，默认为 false。	Boolean	否
enable-word-polygon	type 值为 handwriting 时有效，表示是否开启单字的四点定位坐标输出，有效值为 true 和 false，默认值为 false。	Boolean	否

可识别的语言类型

可选值	语言	可选值	语言
zh	中英混合	rus	俄语
zh_rare	支持英文、数字、中文生僻字、繁体字，特殊符号等	ita	意大利语
auto	自动	hol	荷兰语
mix	混合语种	swe	瑞典语
jap	日语	fin	芬兰语
kor	韩语	dan	丹麦语
spa	西班牙语	nor	挪威语

版权所有：腾讯云计算（北京）有限责任公司

fre	法语	hun	匈牙利语
ger	德语	tha	泰语
por	葡萄牙语	hi	印地语
vie	越语	ara	阿拉伯语
may	马来语		

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。

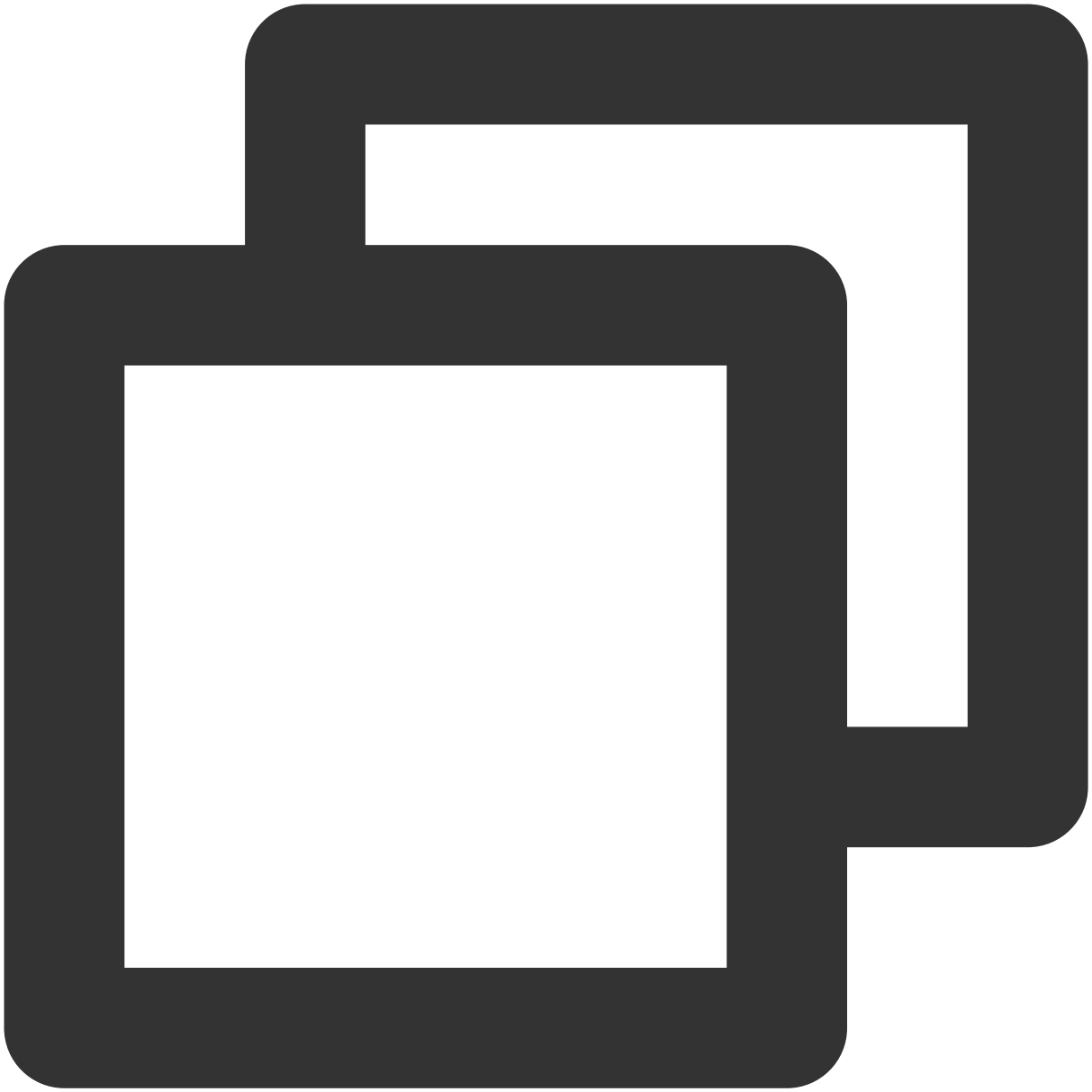
响应

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>
  <TextDetections>
    <DetectedText></DetectedText>
    <Confidence></Confidence>
    <Polygon>
      <X></X>
      <Y></Y>
    </Polygon>
    <ItemPolygon>
      <X></X>
      <Y></Y>
```

```
        <Width></Width>
        <Height></Height>
      </ItemPolygon>
      <Words>
        <Confidence></Confidence>
        <Character></Character>
        <WordCoordPoint>
          <WordCoordinate>
            <X></X>
            <Y></Y>
          </WordCoordinate>
        </WordCoordPoint>
      </Words>
    </TextDetections>
    <Language></Language>
    <Angel></Angel>
    <PdfPageSize></PdfPageSize>
    <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	保存结果的容器。	Container

Response 的内容：

节点名称（关键字）	父节点	描述	类型
TextDetections	Response	检测到的文本信息，包括文本行内容、置信度、文本行坐标以及文本行旋转纠正后的坐标。	Container
Language	Response	检测到的语言类型，目前支持的语言类型可参考入参 language-type 说明。	String
Angel	Response	图片旋转角度（角度制），文本的水平方向为0°；顺时针为正，逆时针为负。	Float
PdfPageSize	Response	图片为 PDF 时，返回 PDF 的总页数，默认为0。	Integer
RequestId	Response	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。	String

TextDetections 节点的内容：

节点名称（关键字）	父节点	描述	类型
DetectedText	TextDetections	识别出的文本行内容。	String
Confidence	TextDetections	置信度 0 - 100。	Integer
Polygon	TextDetections	文本行坐标，以顶点坐标表示 注意：此字段可能返回 null，表示取不到有效值。	Container
ItemPolygon	TextDetections	文本行在旋转纠正之后的图像中的像素坐标，表示为（左上角 x, 左上角 y，宽 width，高 height）。	Container
Words	TextDetections	识别出来的单字信息包括单字（包括单字 Character 和单字置信度 confidence），支持识别的接口：general、accurate。	Container
WordPolygon	TextDetections	字的坐标数组，以四个顶点坐标表示。 注意：此字段可能返回 null，表示取不到有效值。支持识别的类型：handwriting。	Container

Polygon 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	Polygon	横坐标。	Integer
Y	Polygon	纵坐标。	Integer

ItemPolygon 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	ItemPolygon	左上角 x。	Integer
Y	ItemPolygon	左上角 y。	Integer
Width	ItemPolygon	宽 width。	Integer
Height	ItemPolygon	高 height。	Integer

Words 节点的内容：

节点名称（关键字）	父节点	描述	类型
Confidence	Words	置信度 0 - 100。	Integer

版权所有：腾讯云计算（北京）有限责任公司

Character	Words	识别出来的单词信息。	String
WordCoordPoint	Words	单字在原图中的四点坐标，支持识别的接口：general、accurate。	Container

WordCoordPoint 节点的内容：

节点名称（关键字）	父节点	描述	类型
WordCoordinate	WordCoordPoint	单字在原图中的坐标，以四个顶点坐标表示，以左上角为起点，顺时针返回。	Container

WordCoordinate 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	WordCoordinate	横坐标。	Integer
Y	WordCoordinate	纵坐标。	Integer

WordPolygon 节点的内容：

节点名称（关键字）	父节点	描述	类型
LeftTop	WordPolygon	左上顶点坐标。	Container
RightTop	WordPolygon	右上顶点坐标。	Container
RightBottom	WordPolygon	右下顶点坐标。	Container
LeftBottom	WordPolygon	左下顶点坐标。	Container

LeftTop，RightTop，RightBottom，LeftBottom 节点的内容：

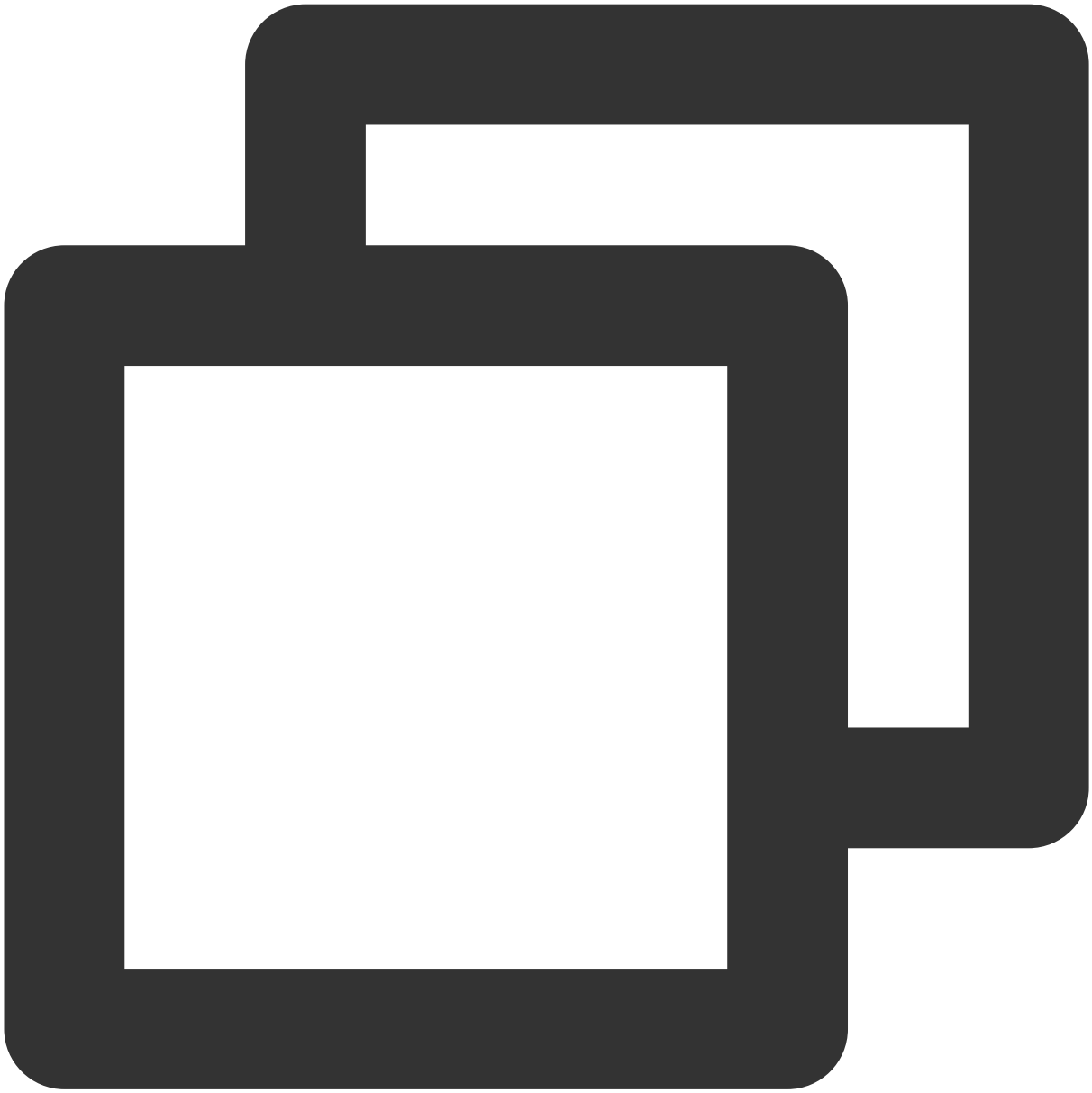
节点名称（关键字）	父节点	描述	类型
X	WordCoordinate	横坐标。	Integer
Y	WordCoordinate	纵坐标。	Integer

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。

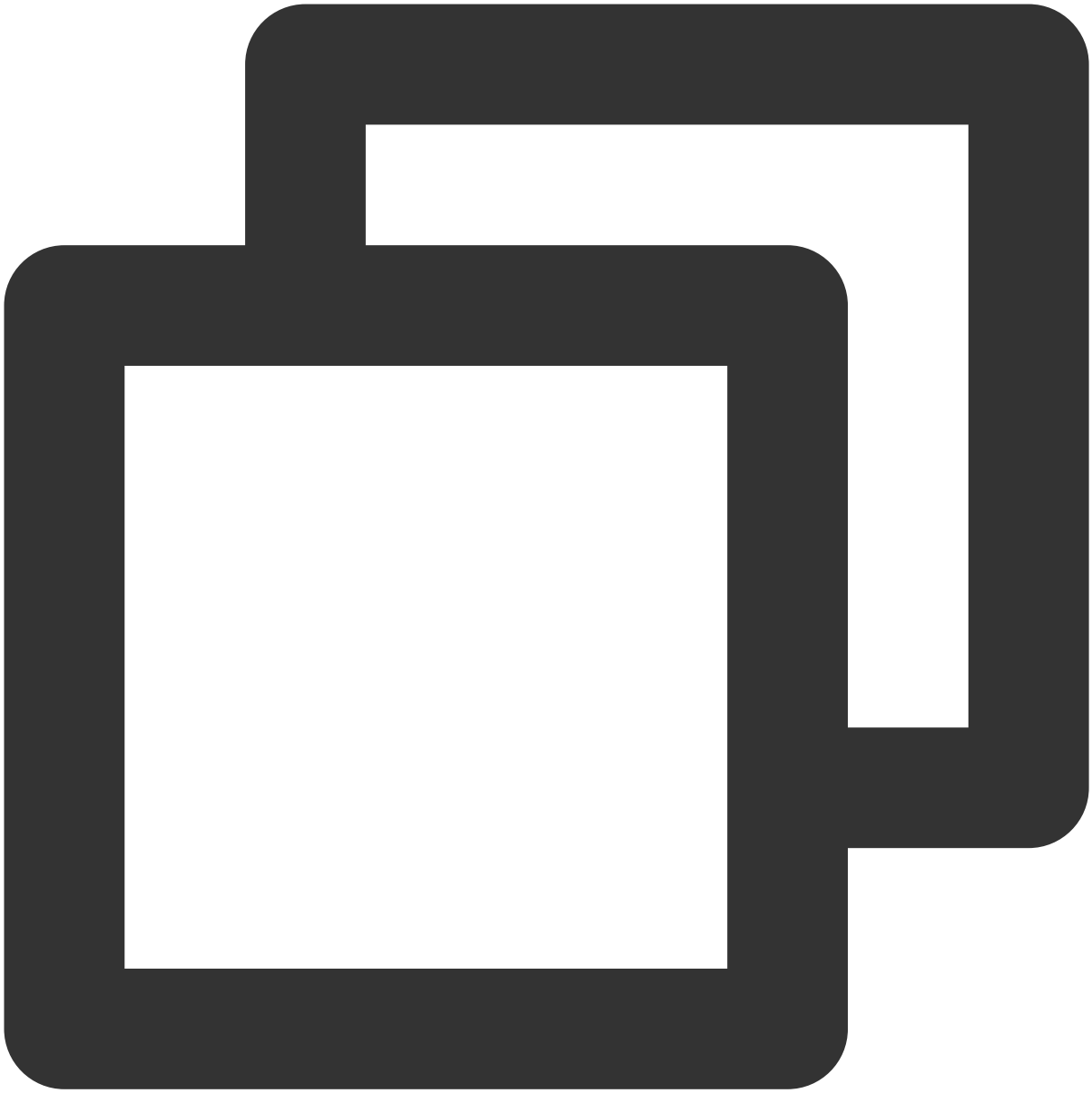
实际案例

请求



```
GET /<ObjectKey>?ci-process=OCR&type=general&language-type=zh&ispdf=true&isword=tru
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 414641
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-image
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<Response>
    <Angel>359.99</Angel>
    <Language>mix</Language>
    <PdfPageSize>0</PdfPageSize>
```

```
<RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
<TextDetections>
    <Confidence>99</Confidence>
    <DetectedText>你好</DetectedText>
    <ItemPolygon>
        <Height>64</Height>
        <Width>123</Width>
        <X>140</X>
        <Y>167</Y>
    </ItemPolygon>
    <Polygon>
        <X>140</X>
        <Y>167</Y>
    </Polygon>
    <Polygon>
        <X>263</X>
        <Y>167</Y>
    </Polygon>
    <Polygon>
        <X>263</X>
        <Y>231</Y>
    </Polygon>
    <Polygon>
        <X>140</X>
        <Y>231</Y>
    </Polygon>
    <Words>
        <Character>你</Character>
        <Confidence>99</Confidence>
        <WordCoordPoint>
            <WordCoordinate>
                <X>212</X>
                <Y>167</Y>
            </WordCoordinate>
            <WordCoordinate>
                <X>341</X>
                <Y>167</Y>
            </WordCoordinate>
            <WordCoordinate>
                <X>341</X>
                <Y>231</Y>
            </WordCoordinate>
            <WordCoordinate>
                <X>212</X>
                <Y>231</Y>
            </WordCoordinate>
        </WordCoordPoint>
```

```
        </Words>
        <Words>
            <Character>好</Character>
            <Confidence>99</Confidence>
            <WordCoordPoint>
                <WordCoordinate>
                    <X>341</X>
                    <Y>167</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>263</X>
                    <Y>167</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>263</X>
                    <Y>231</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>341</X>
                    <Y>230</Y>
                </WordCoordinate>
            </WordCoordPoint>
        </Words>
    </TextDetections>
    <TextDetections>
        <Confidence>99</Confidence>
        <DetectedText>再见</DetectedText>
        <ItemPolygon>
            <Height>43</Height>
            <Width>245</Width>
            <X>526</X>
            <Y>1444</Y>
        </ItemPolygon>
        <Polygon>
            <X>526</X>
            <Y>1444</Y>
        </Polygon>
        <Polygon>
            <X>771</X>
            <Y>1444</Y>
        </Polygon>
        <Polygon>
            <X>771</X>
            <Y>1487</Y>
        </Polygon>
        <Polygon>
            <X>526</X>
```

```
            <Y>1487</Y>
        </Polygon>
        <Words>
            <Character>再</Character>
            <Confidence>99</Confidence>
            <WordCoordPoint>
                <WordCoordinate>
                    <X>564</X>
                    <Y>1444</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>608</X>
                    <Y>1444</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>608</X>
                    <Y>1487</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>564</X>
                    <Y>1487</Y>
                </WordCoordinate>
            </WordCoordPoint>
        </Words>
        <Words>
            <Character>见</Character>
            <Confidence>99</Confidence>
            <WordCoordPoint>
                <WordCoordinate>
                    <X>608</X>
                    <Y>1444</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>641</X>
                    <Y>1444</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>641</X>
                    <Y>1487</Y>
                </WordCoordinate>
                <WordCoordinate>
                    <X>608</X>
                    <Y>1487</Y>
                </WordCoordinate>
            </WordCoordPoint>
        </Words>
    </TextDetections>
```

```
</Response>
```

语音识别
任务接口
提交语音识别任务

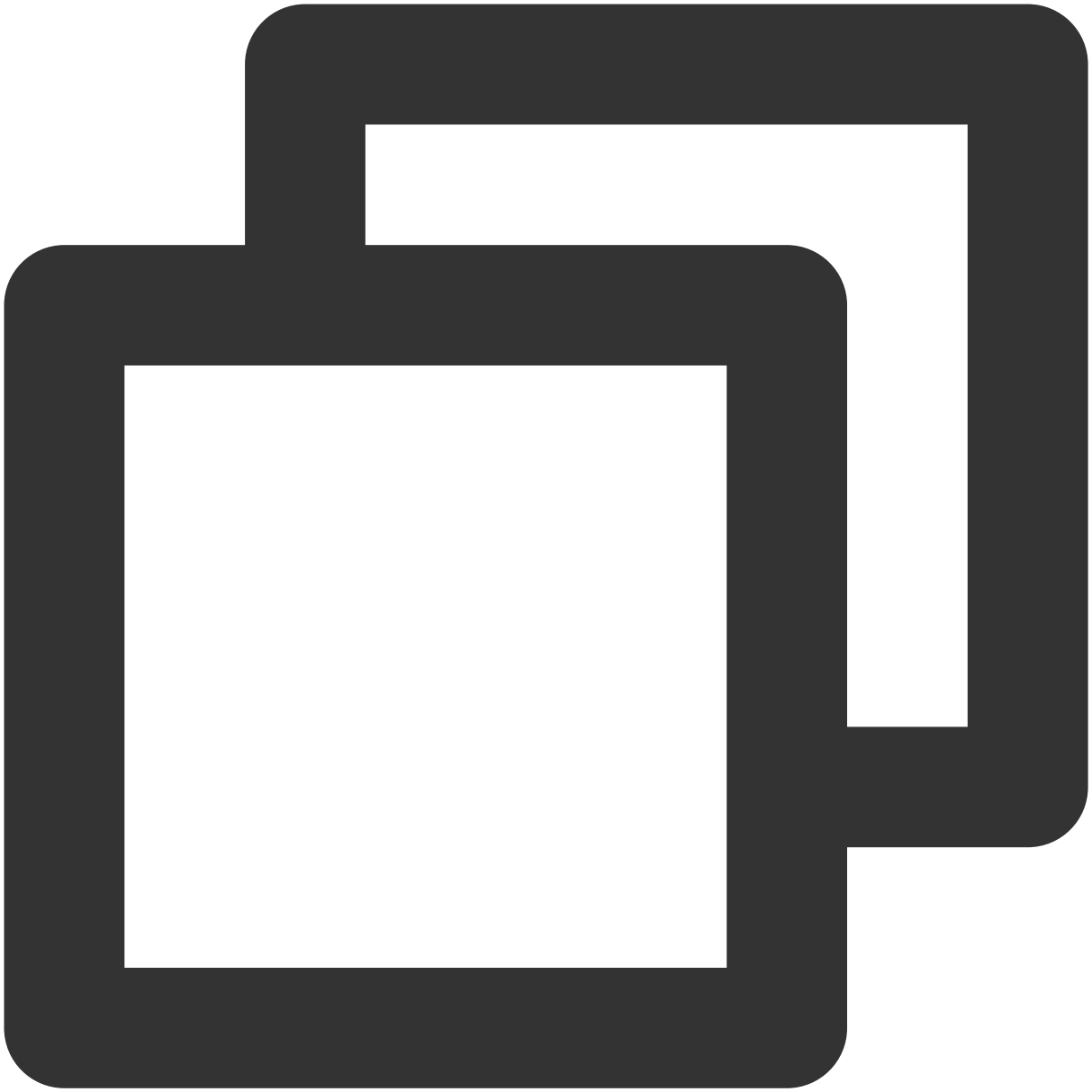
最近更新时间：2024-01-20 09:22:44

功能描述

CreateSpeechJobs 接口用于提交一个语音识别任务。

请求

请求示例



```
POST /asr_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：

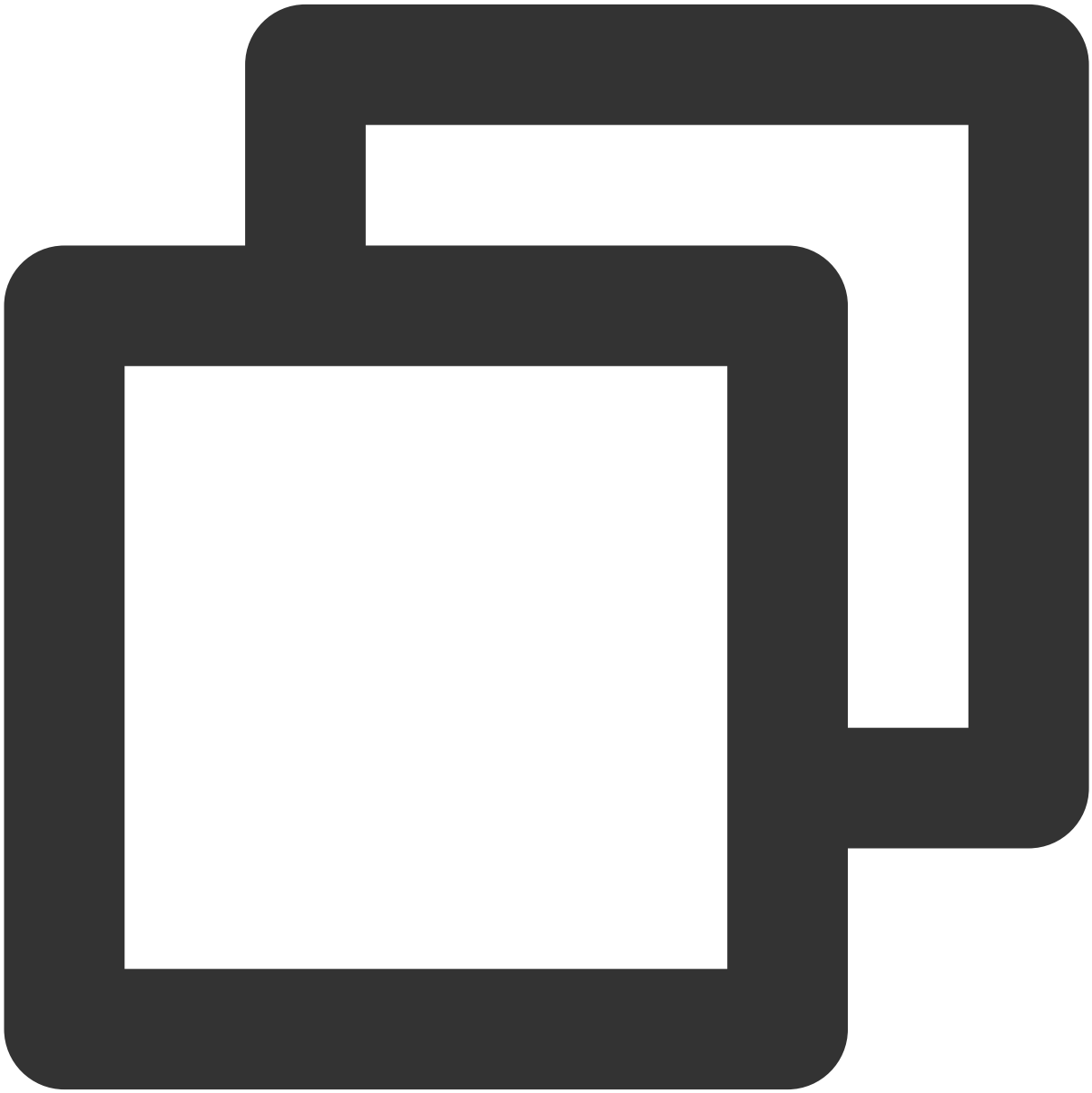
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

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>SpeechRecognition</Tag>
  <Input>
    <Object></Object>
  </Input>
  <Operation>
    <SpeechRecognition></SpeechRecognition>
    <Output>
      <Region></Region>
      <Bucket></Bucket>
      <Object></Object>
    </Output>
  </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：SpeechRecognition	String	是
Input	Request	待操作的语音文件	Container	是
Operation	Request	操作规则	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	语音文件在 COS 上的 key，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
SpeechRecognition	Request.Operation	当 Tag 为 SpeechRecognition 时有效，指定该任务的参数	Container	否
Output	Request.Operation	结果输出地址	Container	是

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.Operation.SpeechRecognition	引擎模型类型。 电话场景： • 8k_zh：电话 8k 中文普通话通用（可用于双声道音频）； • 8k_zh_s：电话 8k 中文普通话话者分离（仅适用于单声道音频）； 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。	String	是
ChannelNum	Request.Operation.SpeechRecognition	语音声道数。1：单声道；2：双声道（仅支持 8k_zh 引擎模型）。	Integer	是
ResTextFormat	Request.Operation.SpeechRecognition	识别结果返回形式。0：识别结果文本(含分段时间戳)；1：仅支持16k中文引擎，含识别结果详情(词时间戳列表，一般用于生成字幕场景)。	Integer	是

FilterDirty	Request.Operation.SpeechRecognition	是否过滤脏词（目前支持中文普通话引擎）。0：不过滤脏词；1：过滤脏词；2：将脏词替换为 * 。默认值为0。	Integer	否
FilterModal	Request.Operation.SpeechRecognition	是否过语气词（目前支持中文普通话引擎）。0：不过滤语气词；1：部分过滤；2：严格过滤 。默认值为 0。	Integer	否
ConvertNumMode	Request.Operation.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：不转换，直接输出中文数字，1：根据场景智能转换为阿拉伯数字。默认值为1。	Integer	否

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

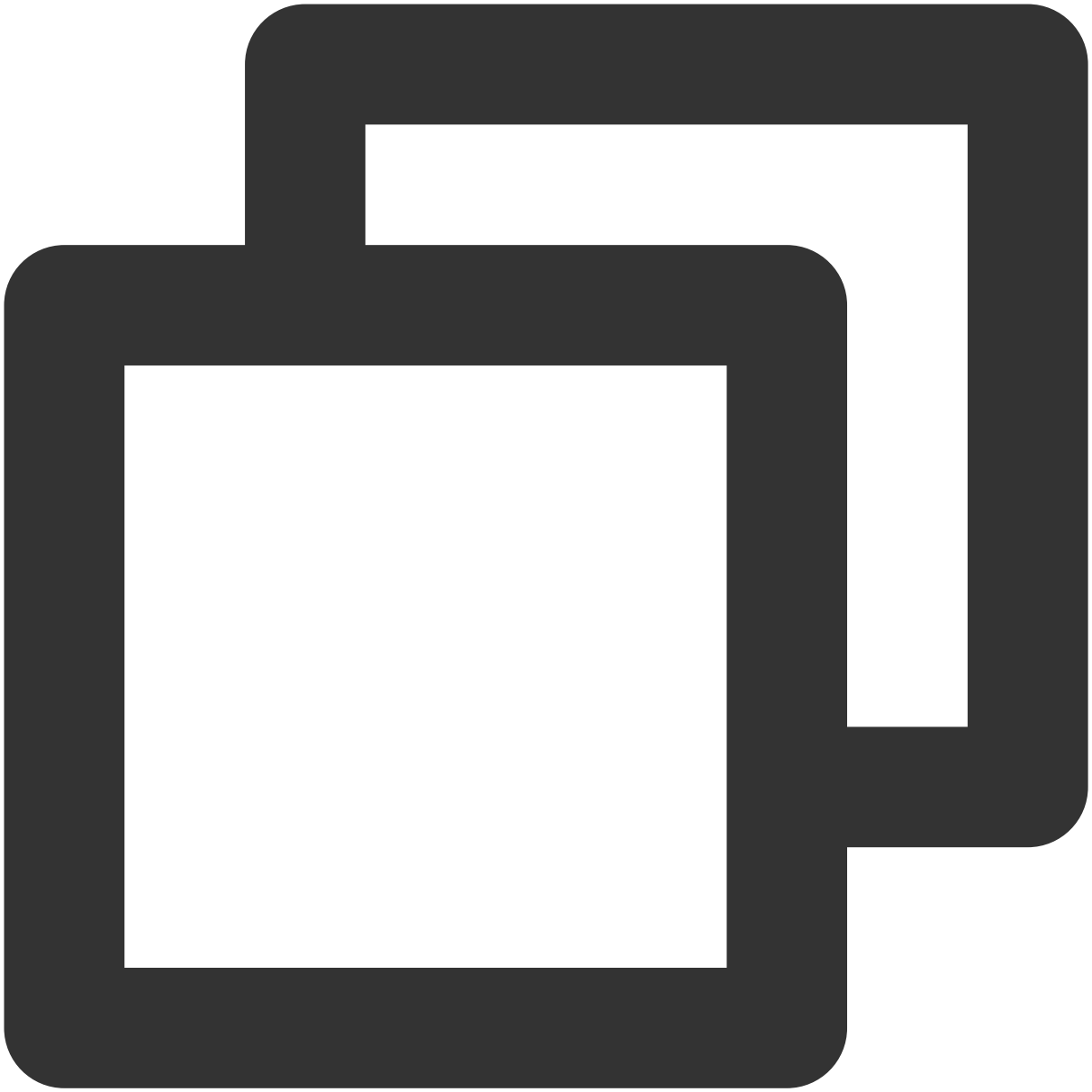
响应

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>
  <JobsDetail>
    <Code></Code>
    <Message></Message>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <QueueId></QueueId>
    <Tag><Tag>
    <Input>
      <Object></Object>
```

```
    </Input>
    <Operation>
      <SpeechRecognition></SpeechRecognition>
      <Output>
        <Region></Region>
        <Bucket></Bucket>
        <Object></Object>
      </Output>
      <MediaInfo>
      </MeidaInfo>
    </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：SpeechRecognition	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	该任务的输入资源地址	Container

版权所有：腾讯云计算（北京）有限责任公司

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
TemplateId	Response.JobsDetail.Operation	任务的模板 ID	String
Output	Response.JobsDetail.Operation	文件的输出地址	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：

同上面请求中的 Request.Operation.Output 节点。

Container 节点 SpeechRecognition 的内容：

同上面请求中的 Request.Operation.SpeechRecognition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。

查询指定的语音识别任务

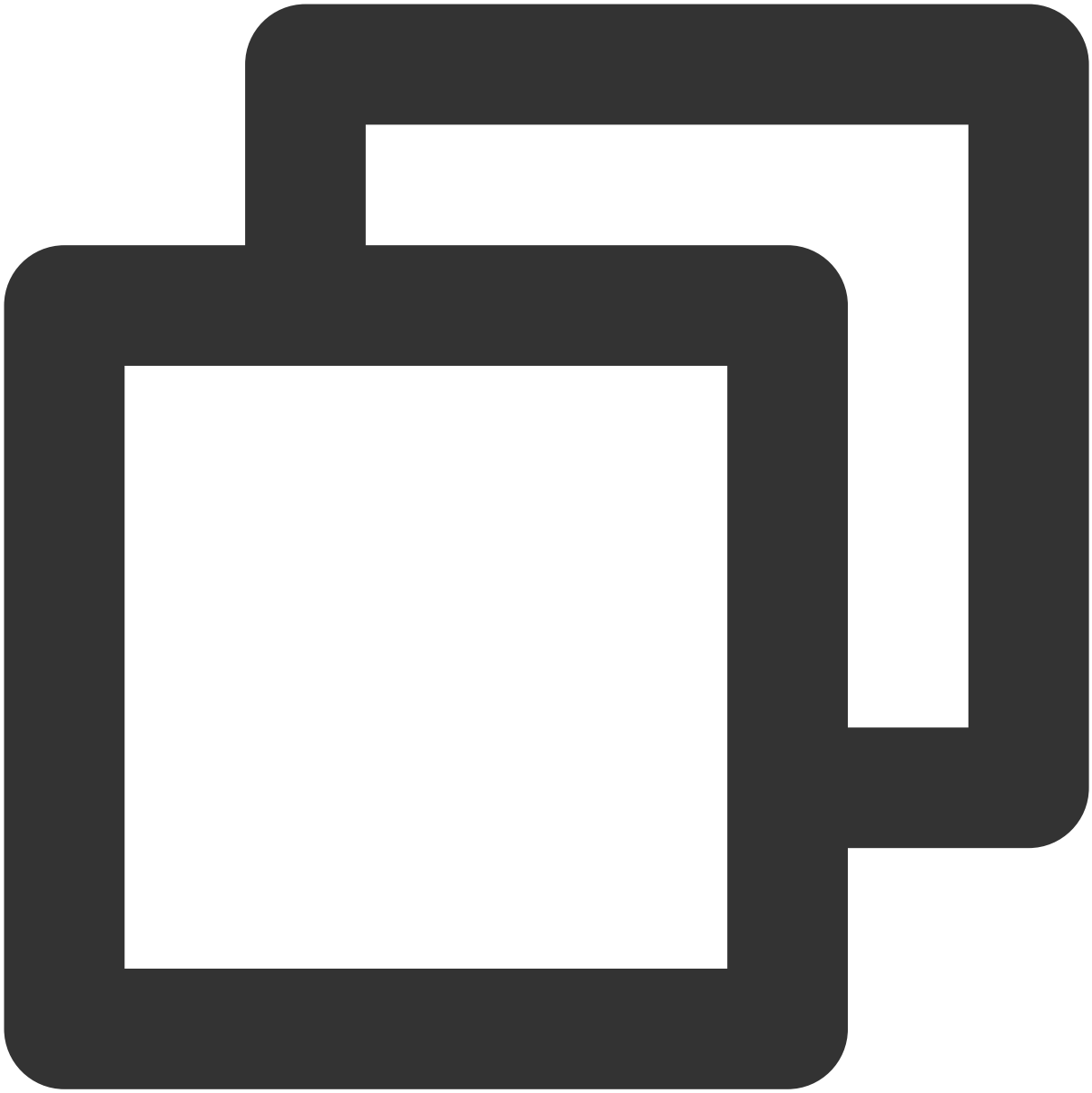
最近更新时间：2024-01-20 09:22:44

功能描述

DescribeSpeechJob 用于查询指定的语音识别任务。

请求

请求示例



```
GET /asr_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。

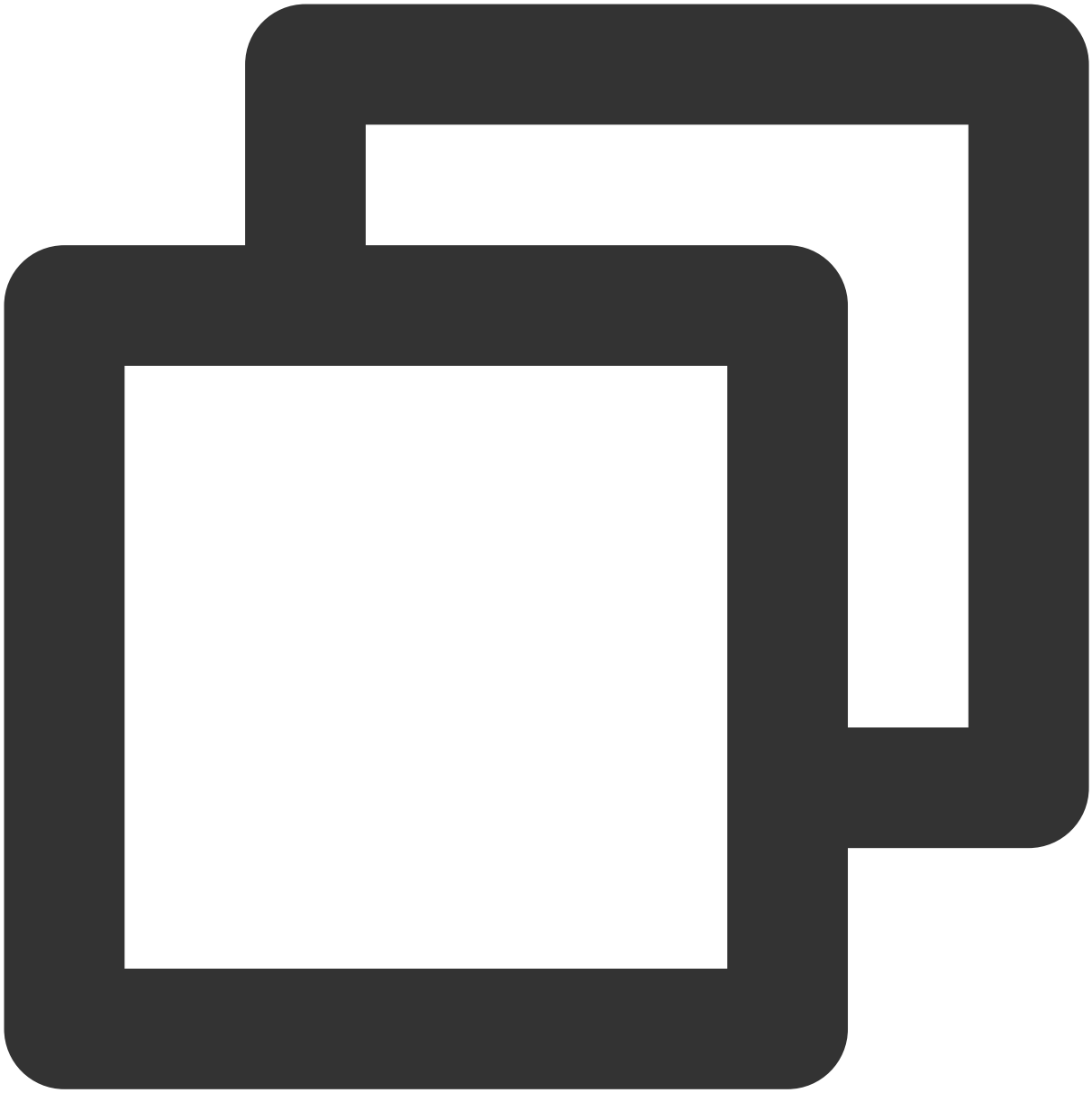
响应

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，同 CreateSpeechJobs 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
SpeechRecognitionResult	Response	在 job 的类型为 SpeechRecognition 且 job 状态为 success 时，返回语音识别的识别结果详情。	Container

Container 节点 SpeechRecognitionResult 的内容：

节点名称（关键字）	父节点	描述	类型
Result	Response.SpeechRecognitionResult	识别结果	Container
ResultDetail	Response.SpeechRecognition.ResultDetail	识别结果详情，包含每个句子中的词时间偏移，一般用于生成字幕的场景。(识别请求中ResTextFormat=1时该字段不为空) 注意：此字段可能返回null，表示取不到有效值。	Array of SentenceDetail
AudioTime	Response.SpeechRecognition.audioTime	语音时长	double

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。

批量拉取语音识别任务

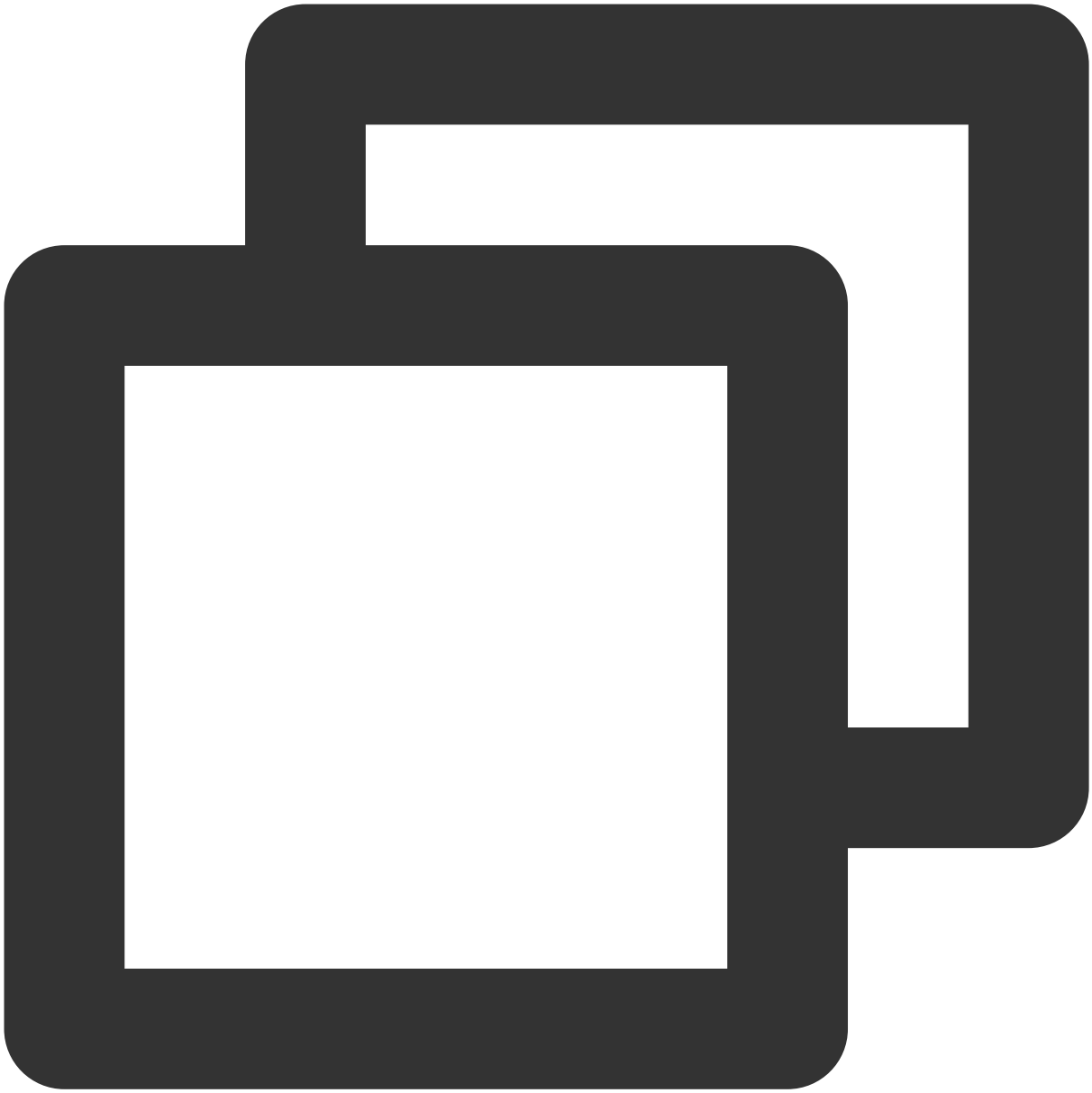
最近更新时间：2024-01-20 09:22:44

功能描述

DescribeSpeechJobs 用于拉取符合条件的语音识别任务。

请求

请求示例



```
GET /asr_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：SpeechRecognition。	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

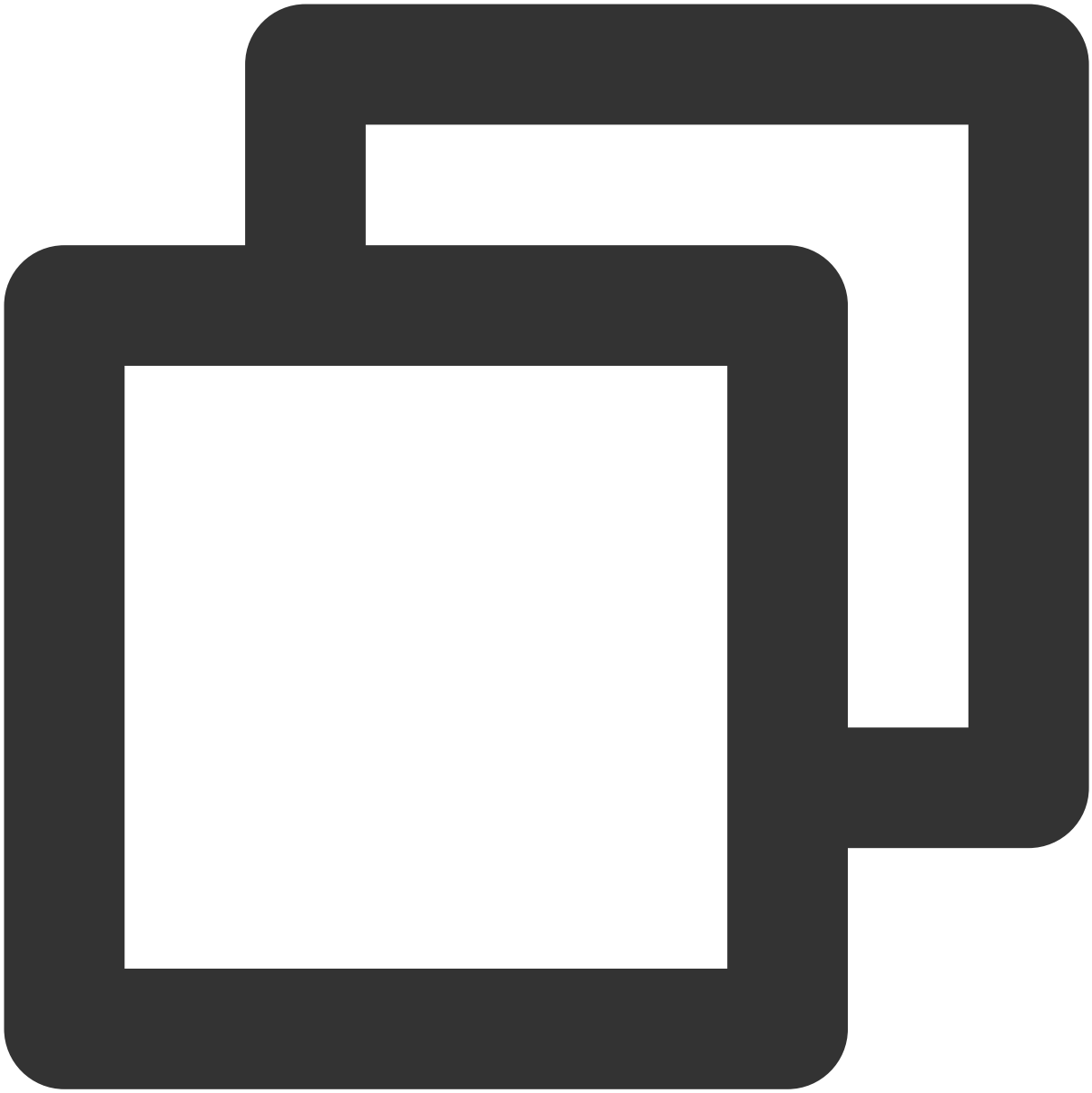
响应

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 CreateSpeechJobs 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。

队列接口
查询语音识别队列

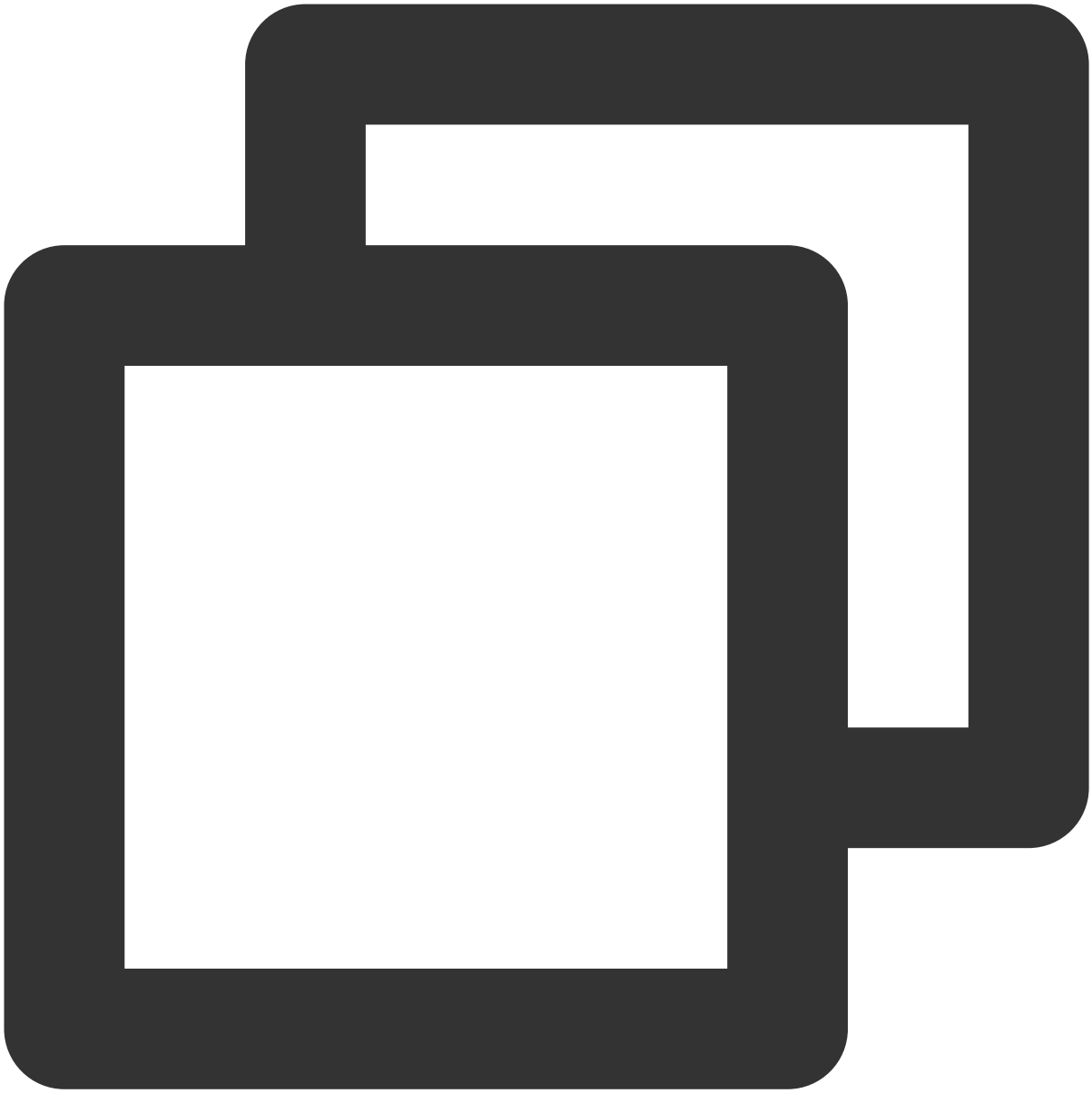
最近更新时间：2024-01-20 09:22:44

功能描述

DescribeSpeechQueues 接口用于查询语音识别队列。

请求

请求示例



```
GET /asrqueue 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	类型	描述	是否必选
queueIds	string	队列 ID，以“,”符号分割字符串。	否
state	string	1. Active 表示队列内的作业会被语音识别服务调度执行。 2. Paused 表示队列暂停，作业不再会被语音识别服务调度执行，队列内的所有作业状态维持在暂停状态，已经处于识别中的任务将继续执行，不受影响。	否
pageNumber	string	第几页。	否
pageSize	string	每页个数。	否

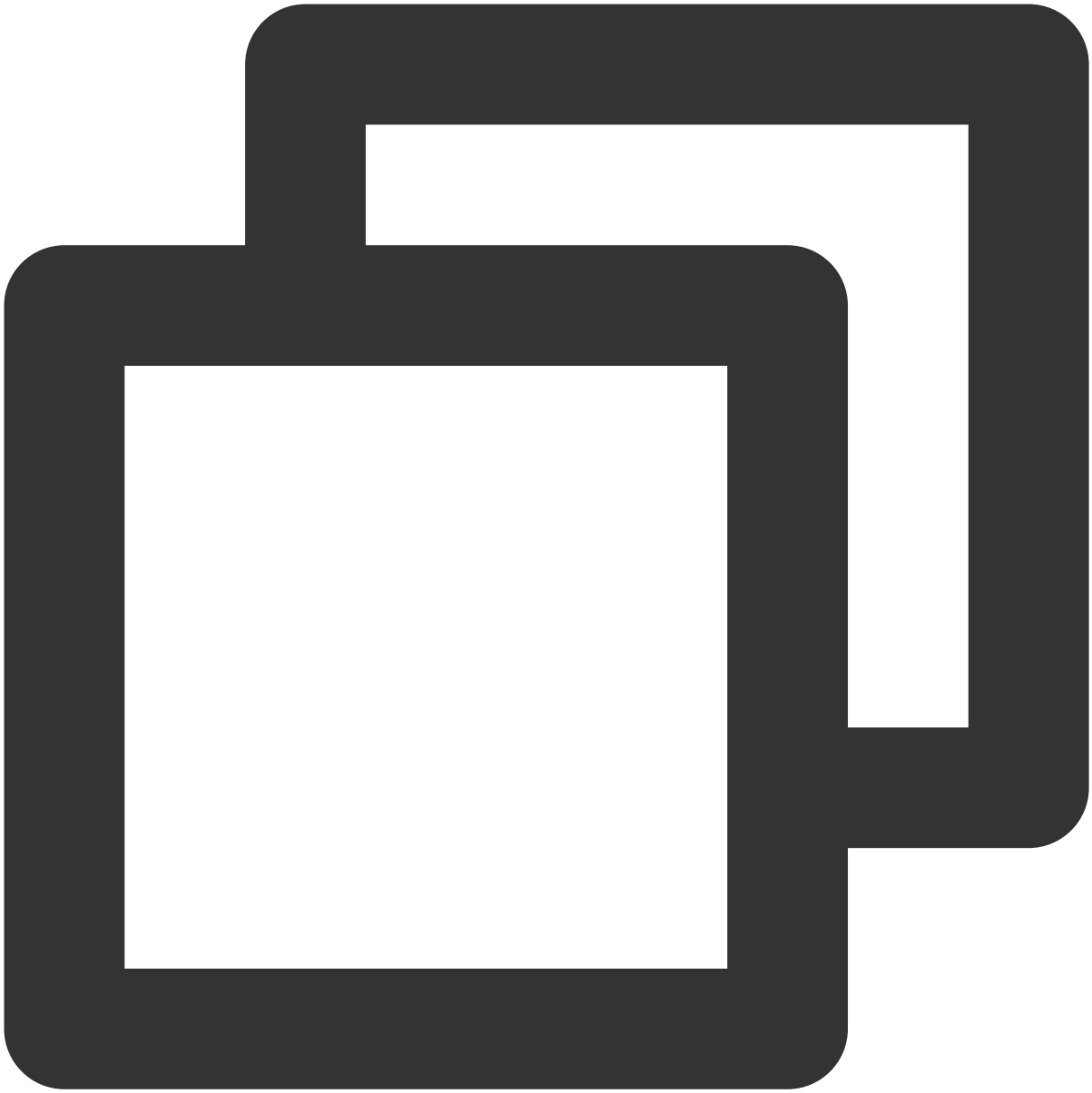
响应

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>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TotalCount>1</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <QueueList>
        <QueueId>A</QueueId>
        <Name></Name>
        <State>Active</State>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
```

```
        <UpdateTime></UpdateTime>
        <CreateTime></CreateTime>
        <NotifyConfig>
            <Url></Url>
            <Type></Type>
            <State></State>
            <Event></Event>
        </NotifyConfig>
    </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
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

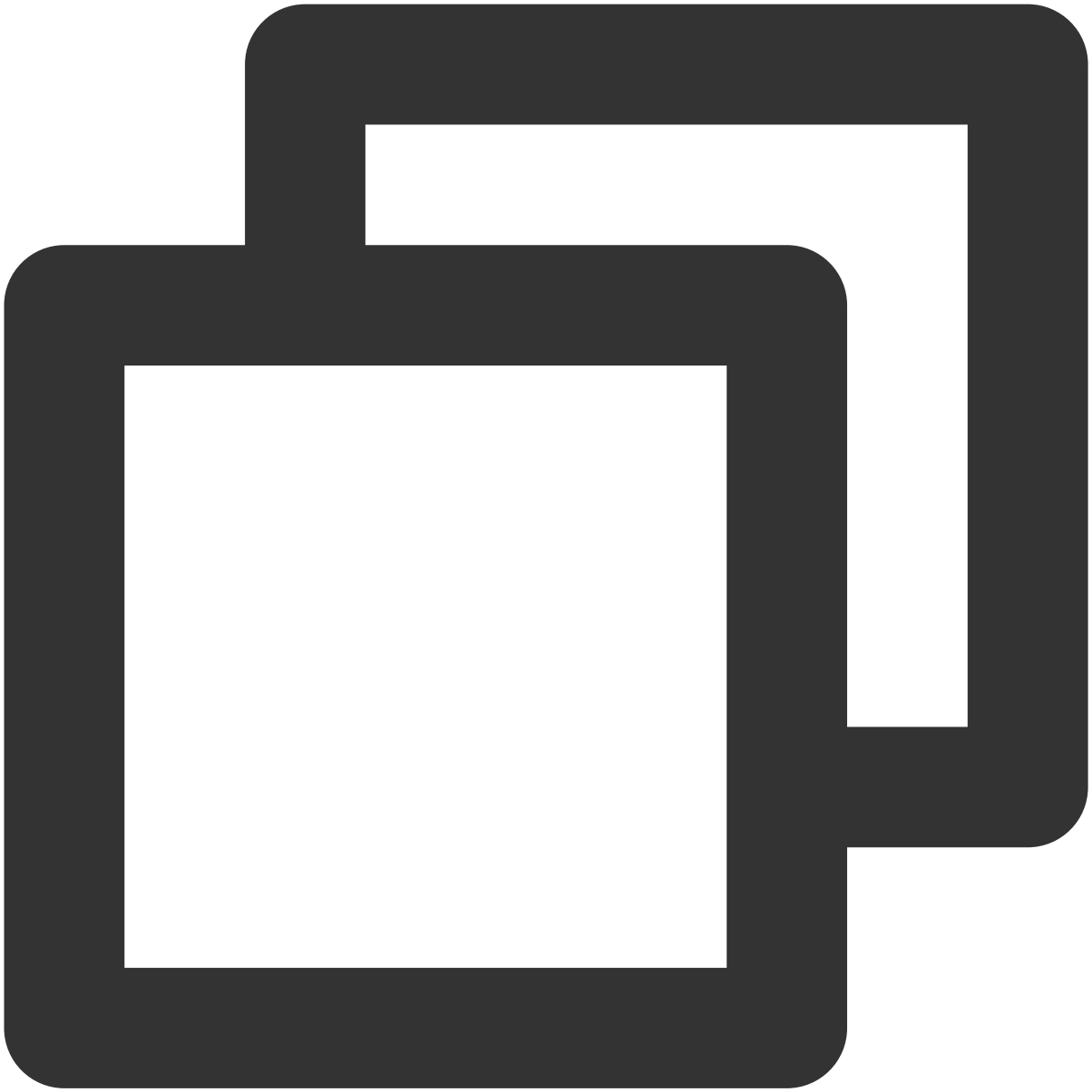
版权所有：腾讯云计算（北京）有限责任公司

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

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



```
<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。

更新语音识别队列

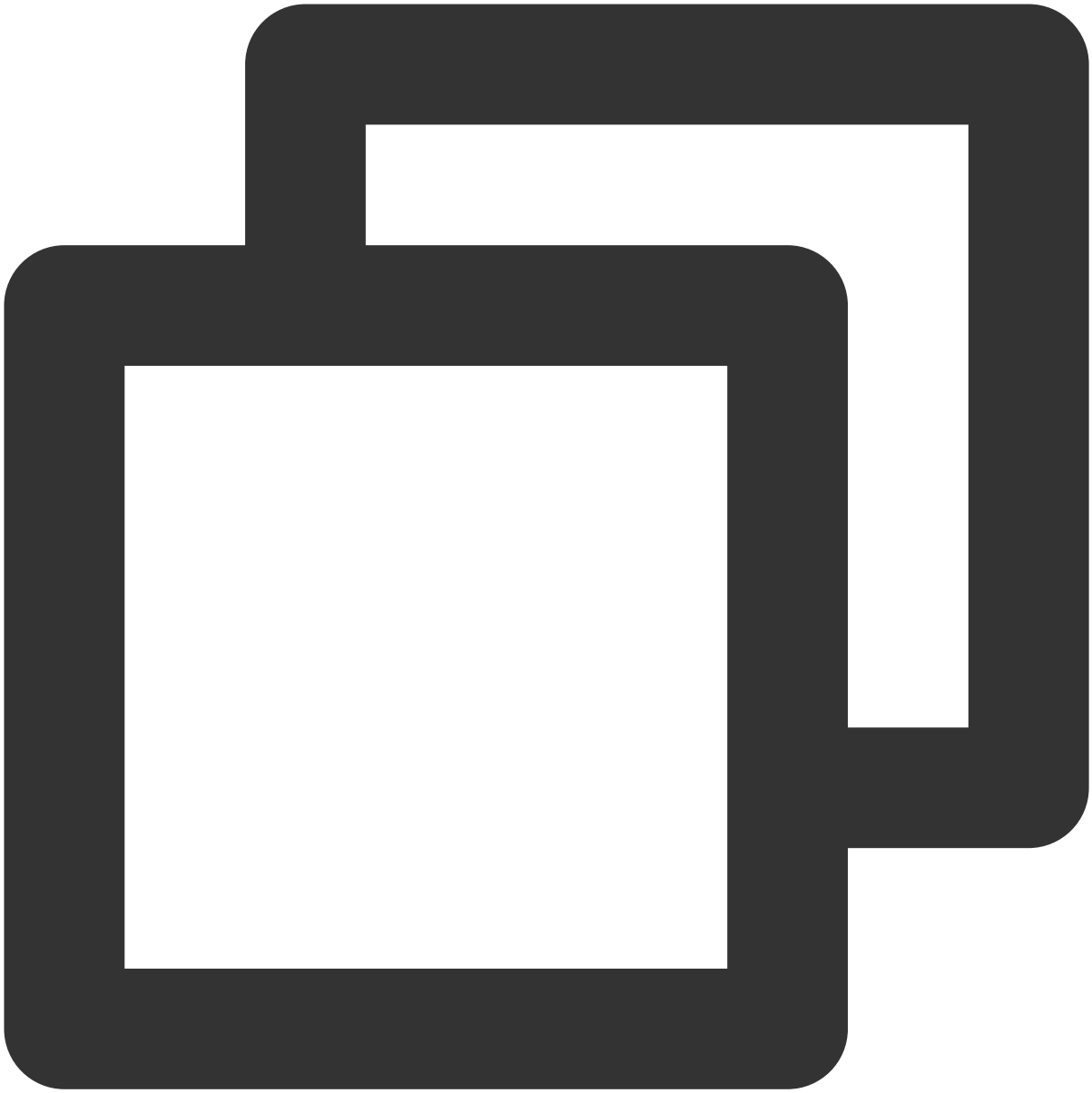
最近更新时间：2024-01-20 09:22:44

功能描述

UpdateSpeechQueue 接口用于更新语音识别队列。

请求

请求示例



```
PUT /asrqueue/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：

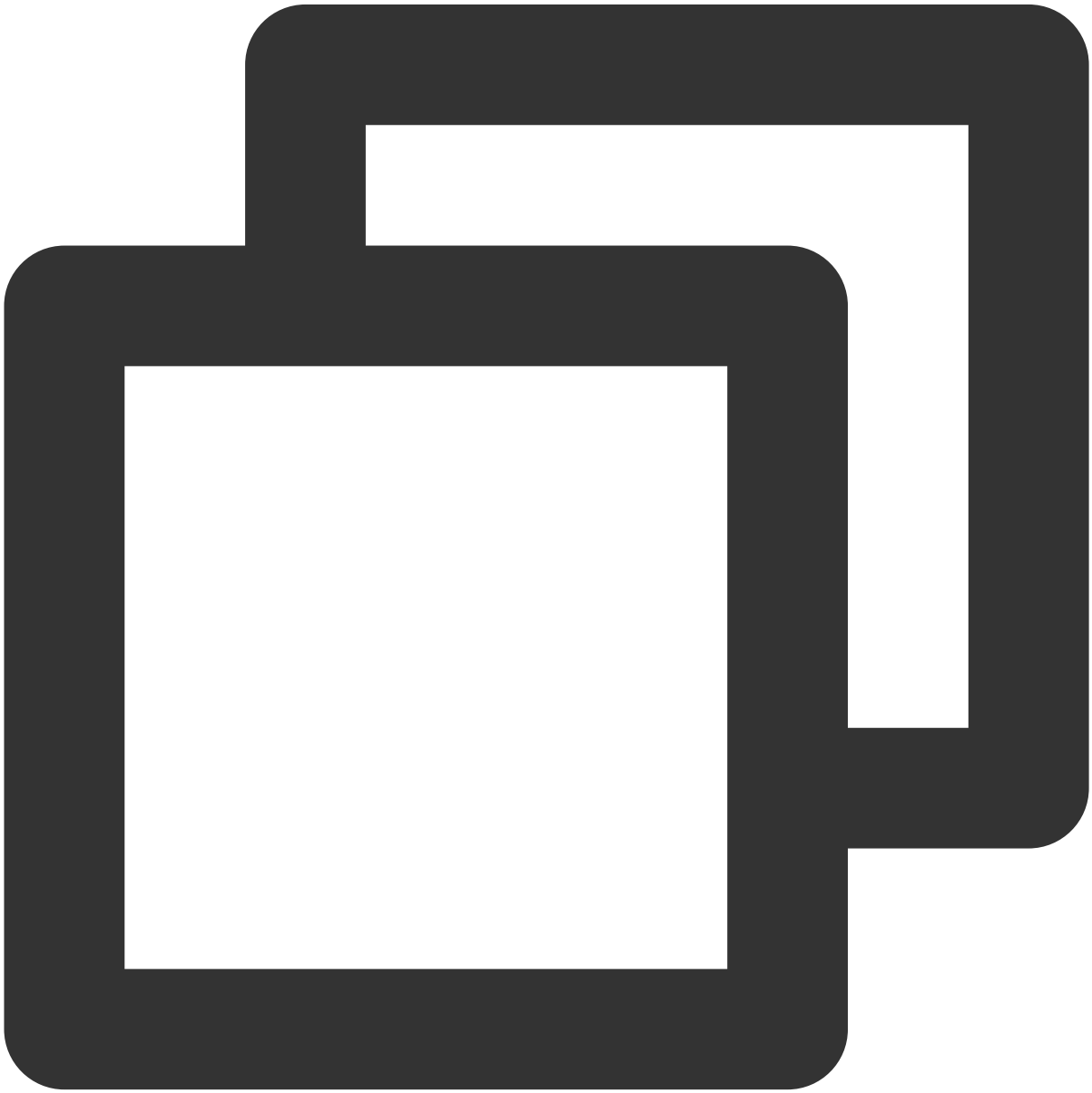
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

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	模板名称	String	是	长度限制100字符
QueueID	Request	管道 ID	String	是	-
State	Request	管道状态	String	是	1. Active 表示管道内的作业会被语音识别服务调度执行 2. Paused 表示管道暂停，作业不再会被语音识别服务调度执行，管道内的所有作业状态维持在已提交状态，已经处于识别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

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	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

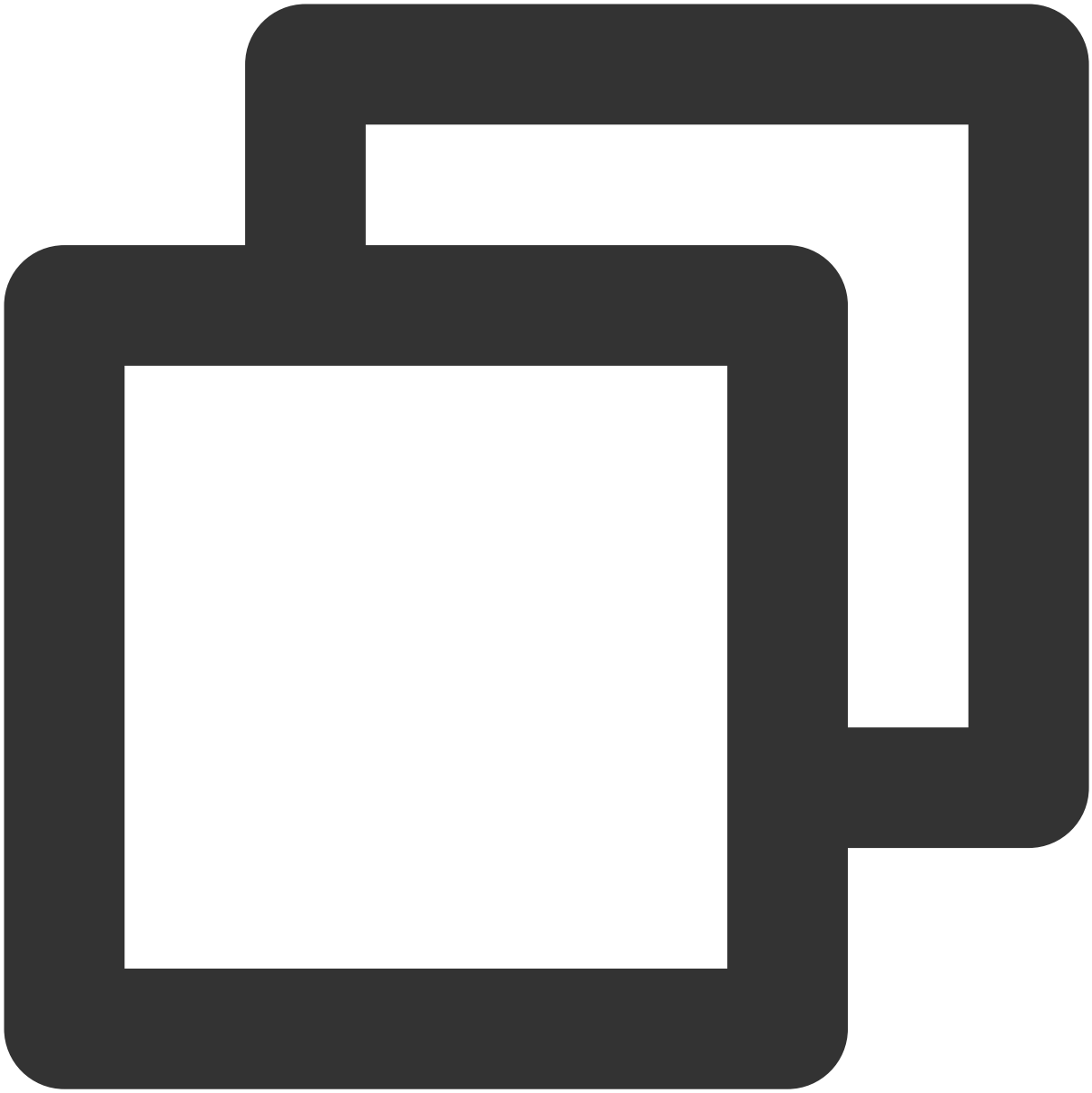
响应

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，详情同 [DescribeSpeechQueues]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。

以图搜图

开通以图搜图

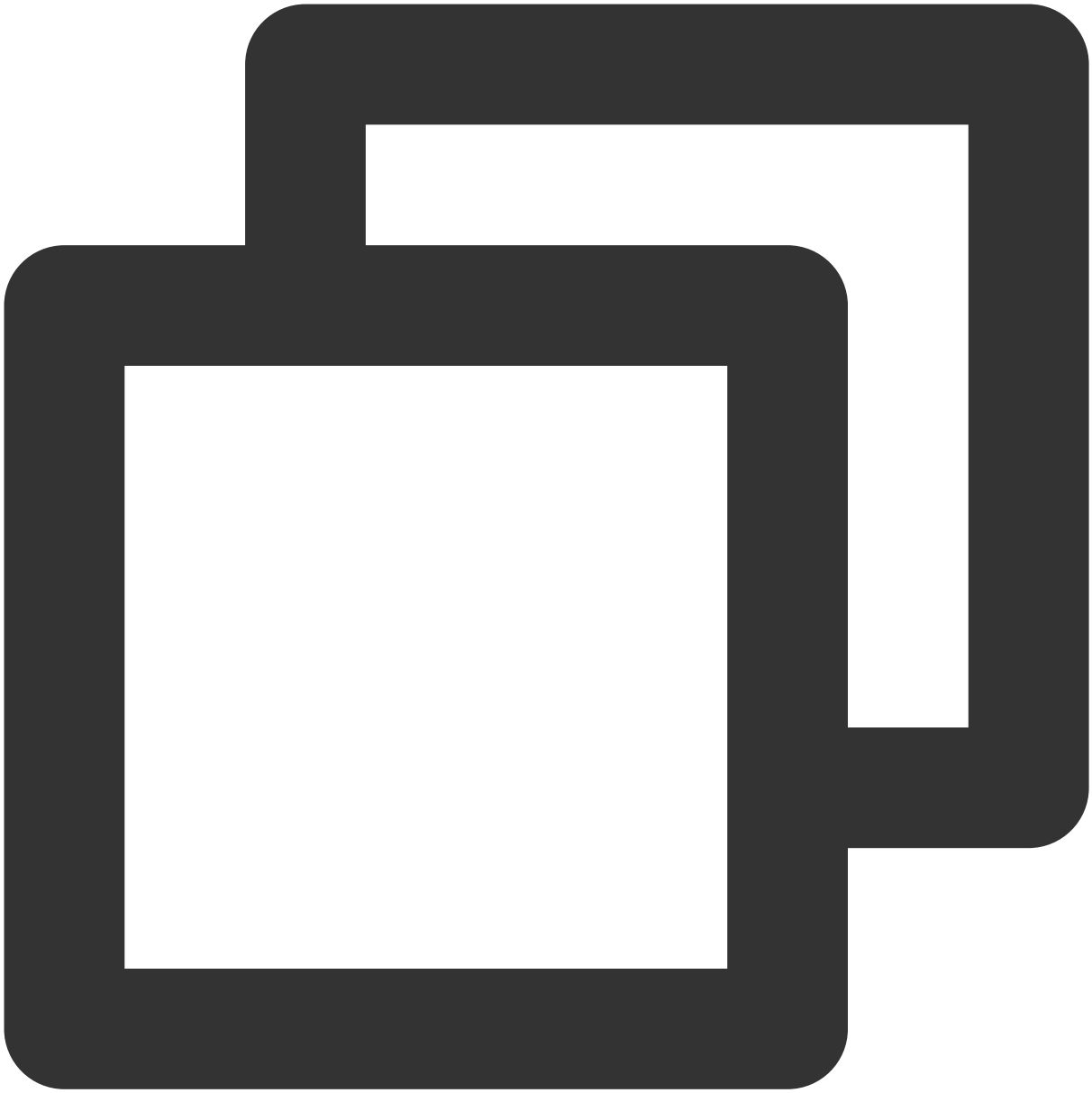
最近更新时间：2024-01-20 09:22:44

功能描述

该接口用于开通 Bucket 搜图功能。

请求

请求示例



```
POST /ImageSearchBucket 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

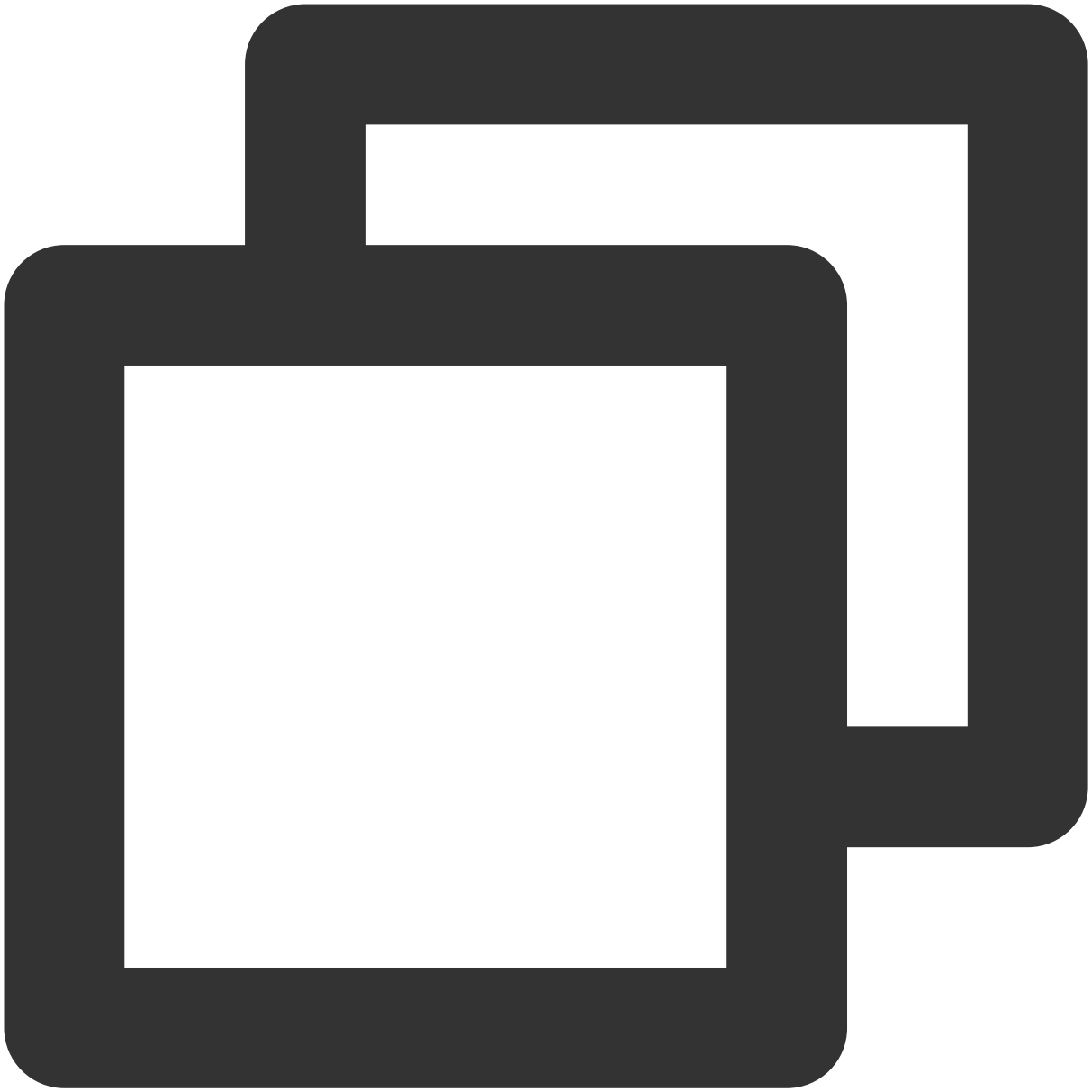
此接口无请求参数。

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

该请求操作的实现需要有如下请求体。



```
<Request>
    <MaxCapacity>10000</MaxCapacity>
    <MaxQps>10</MaxQps>
</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

Request 节点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
MaxCapacity	Request	图库容量限制	Int	是
MaxQps	Request	图库访问限制，默认10	Int	否

说明：

图库容量设置后不允许修改。

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

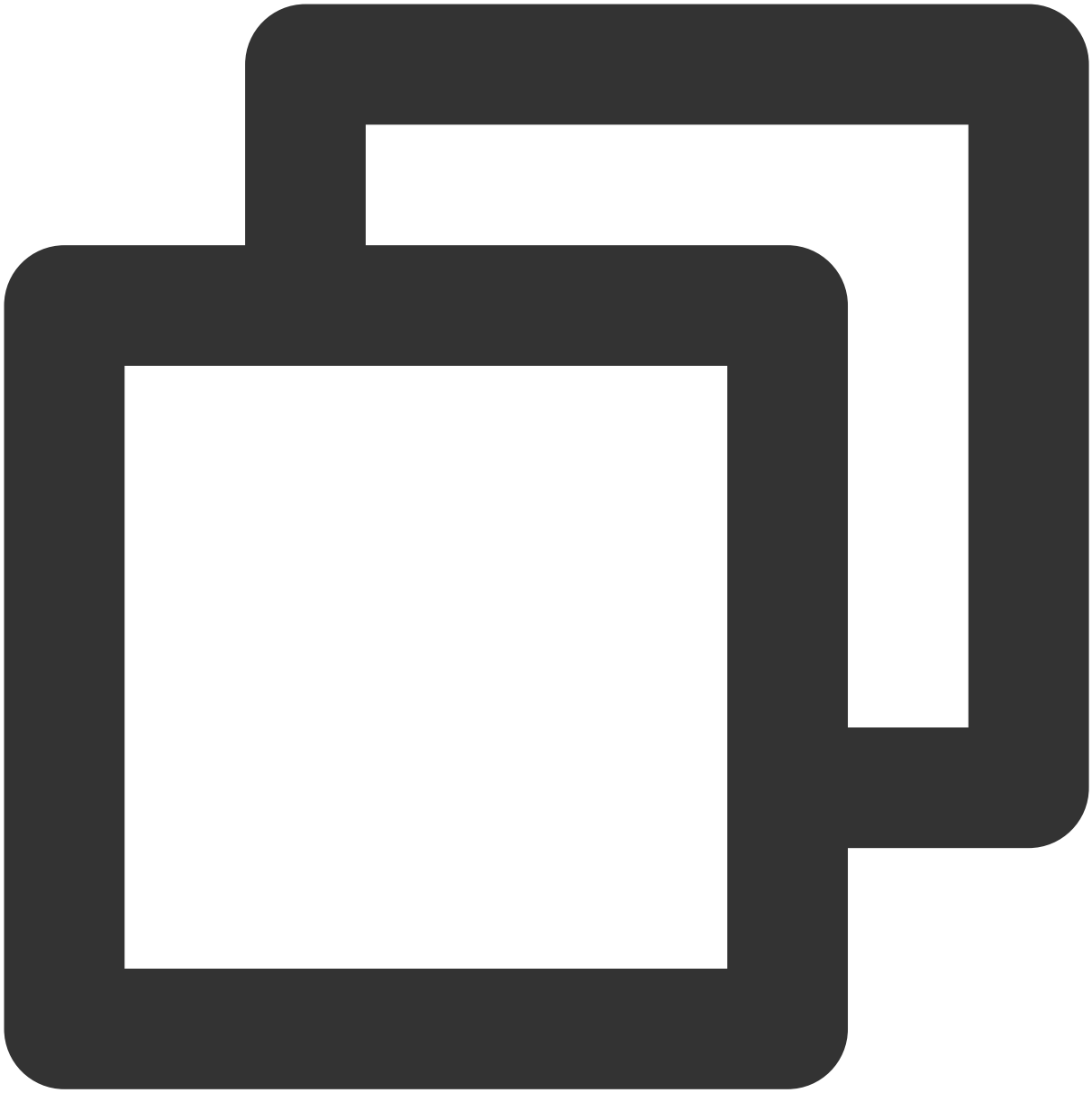
该响应无响应体。

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

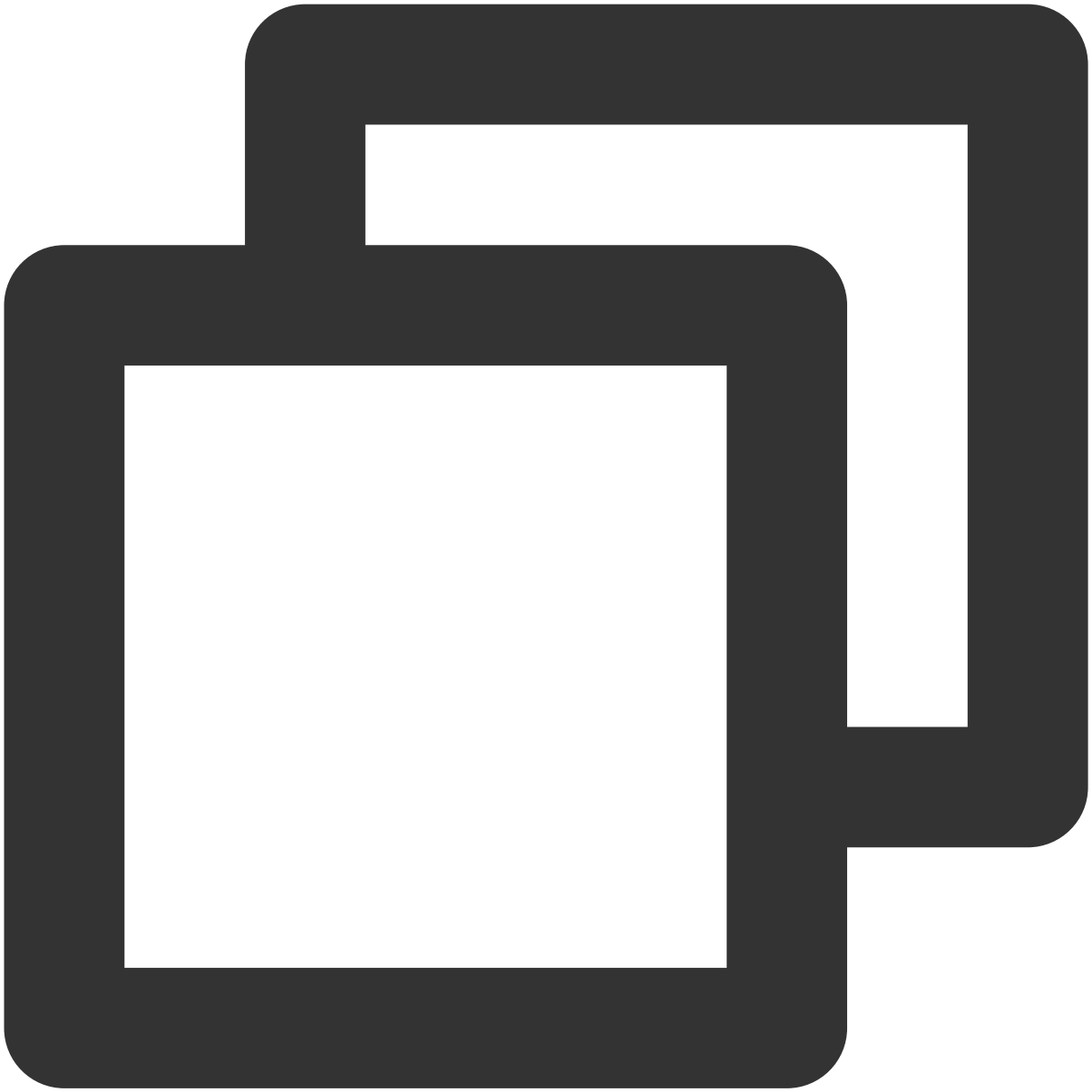
请求



```
POST /ImageSearchBucket HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 50
Content-Type: application/xml

<Request>
    <MaxCapacity>10000</MaxCapacity>
    <MaxQps>10</MaxQps>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
```

添加图库图片

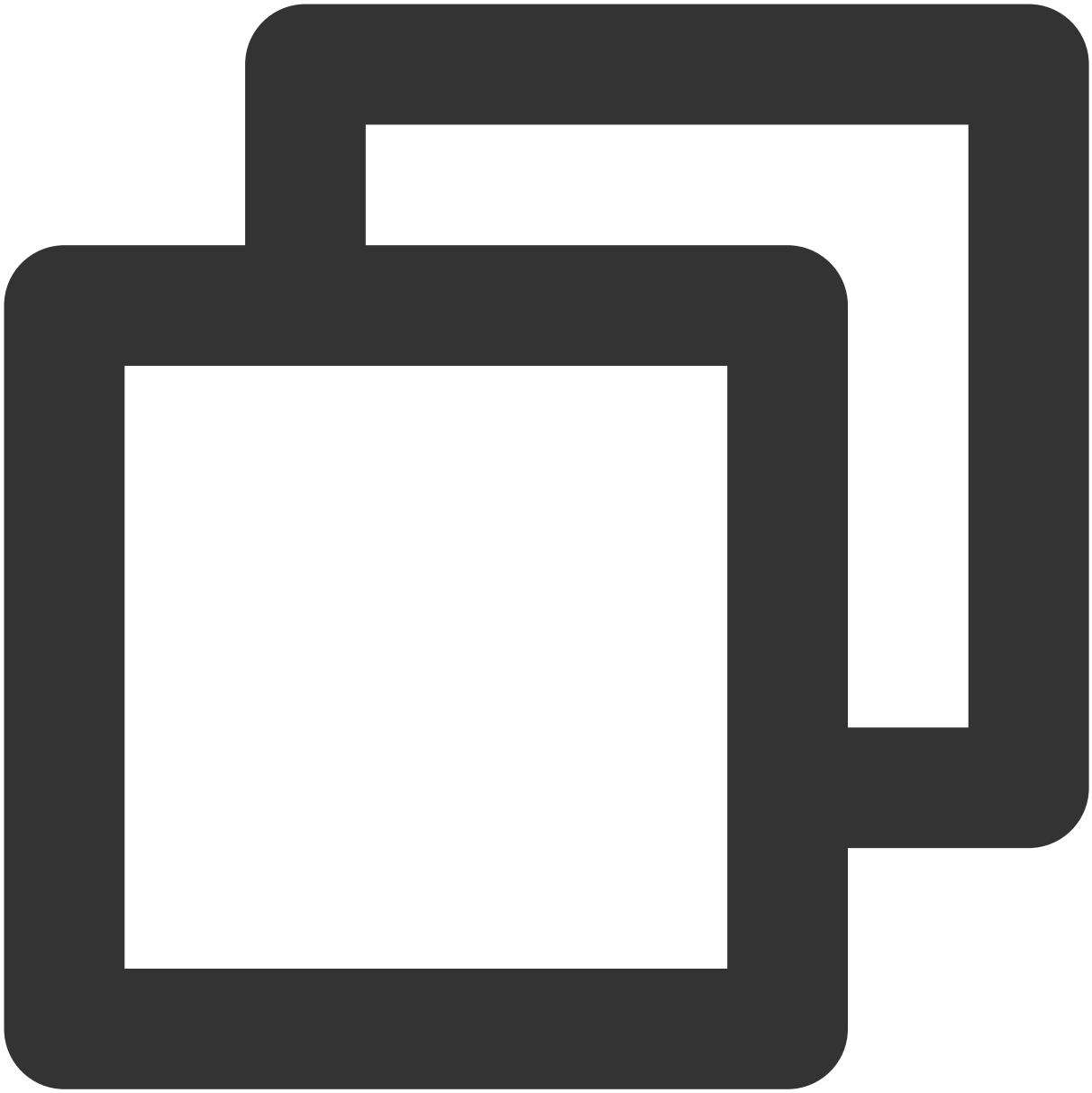
最近更新时间：2024-01-20 09:22:44

功能描述

该接口用于添加图库图片。

请求

请求示例



```
POST /<ObjectKey>?ci-process=ImageSearch&action=AddImage HTTP/1.1
Host: <BucketName-APPID>.cos.<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

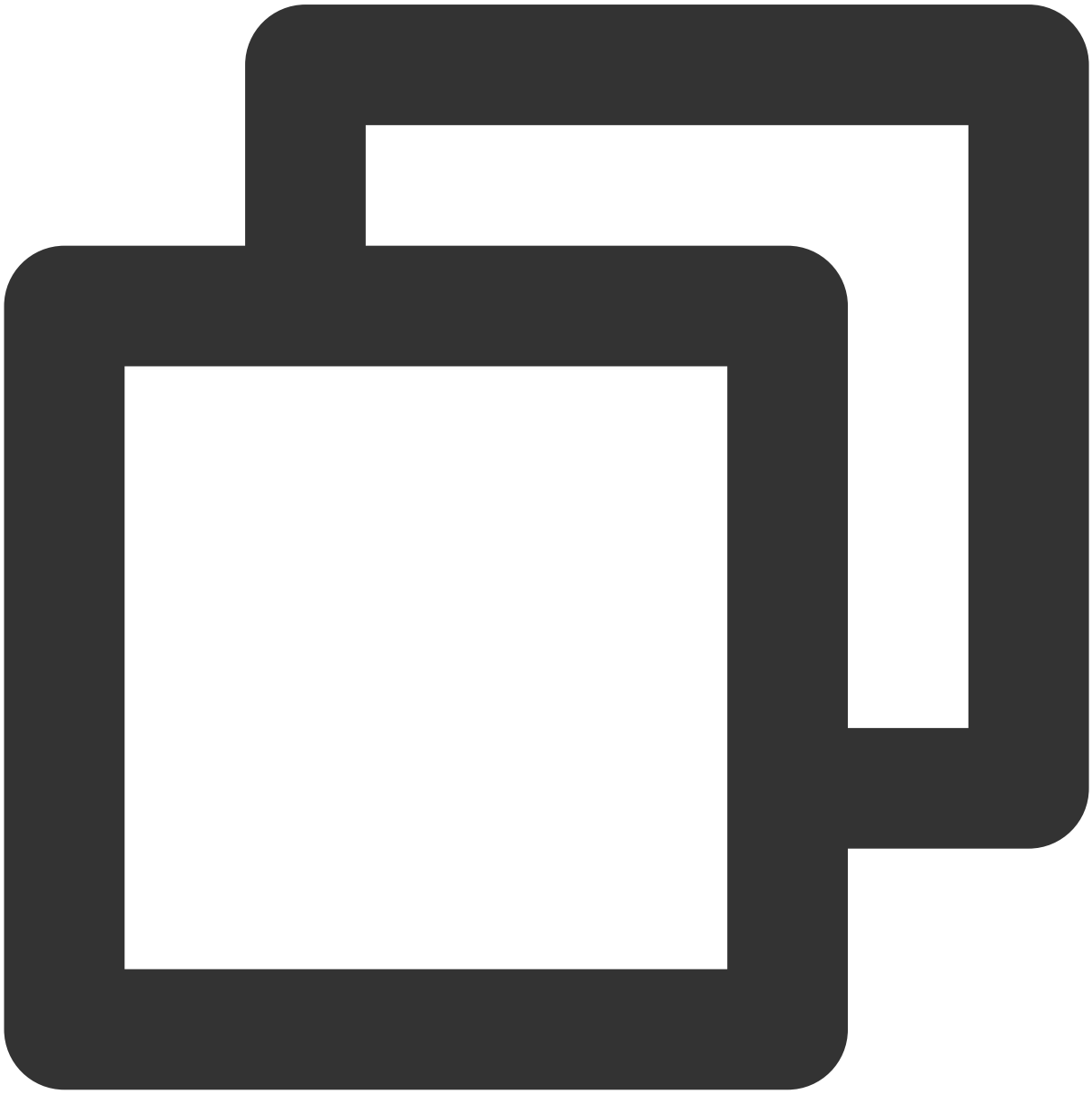
此接口无请求参数。

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

该请求操作的实现需要有如下请求体。



```
<Request>
    <EntityId>apple</EntityId>
    <CustomContent>myapple</CustomContent>
    <Tags>{"key1":"val1","key2":"val2"}</Tags>
</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

Request节点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
EntityId	Request	物品 ID，最多支持64个字符。若 EntityId 已存在，则对其追加图片	String	是
CustomContent	Request	用户自定义的内容，最多支持4096个字符，查询时原样带回	String	否
Tags	Request	图片自定义标签，最多不超过10个，json 字符串，格式为 key:value 对	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

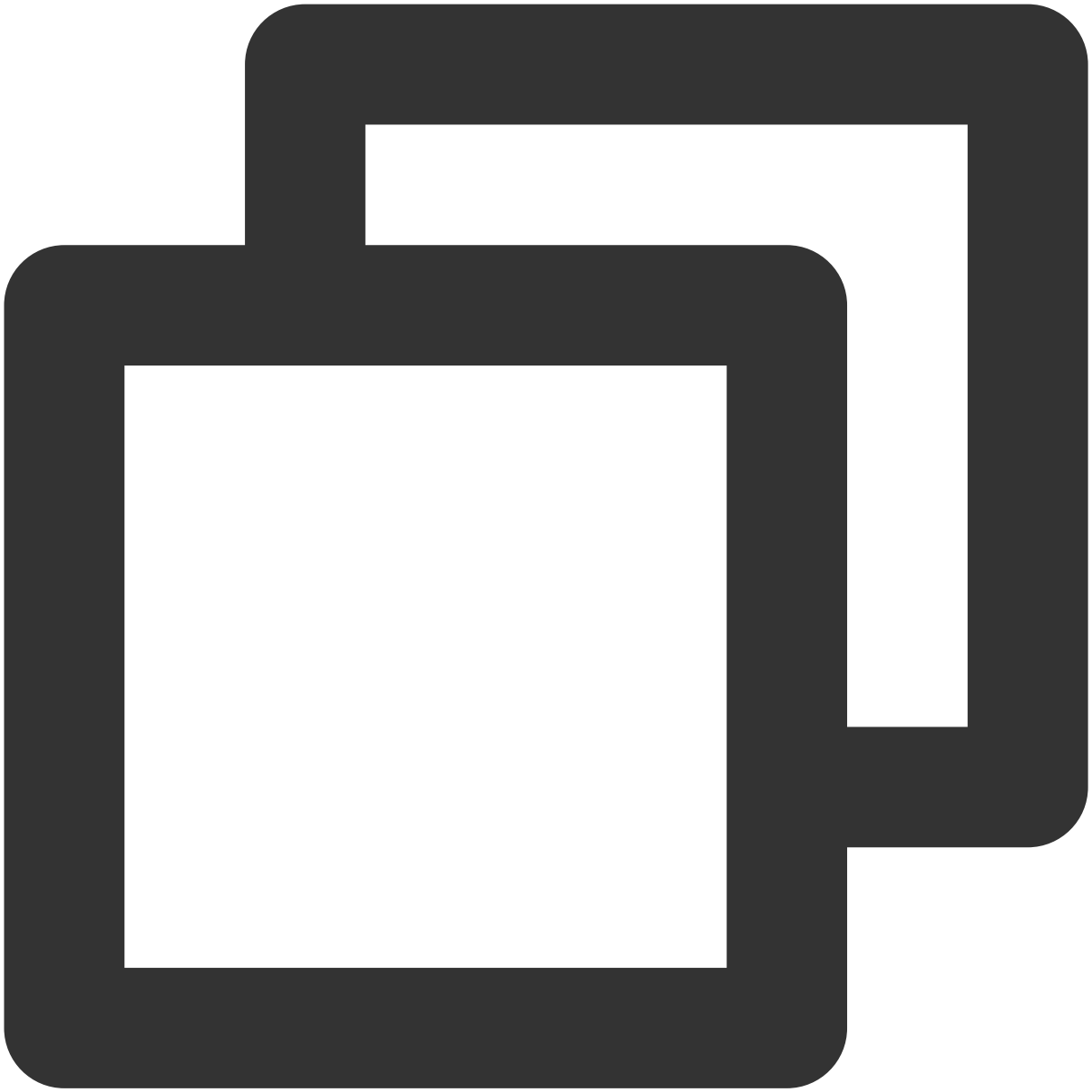
该响应无响应体。

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

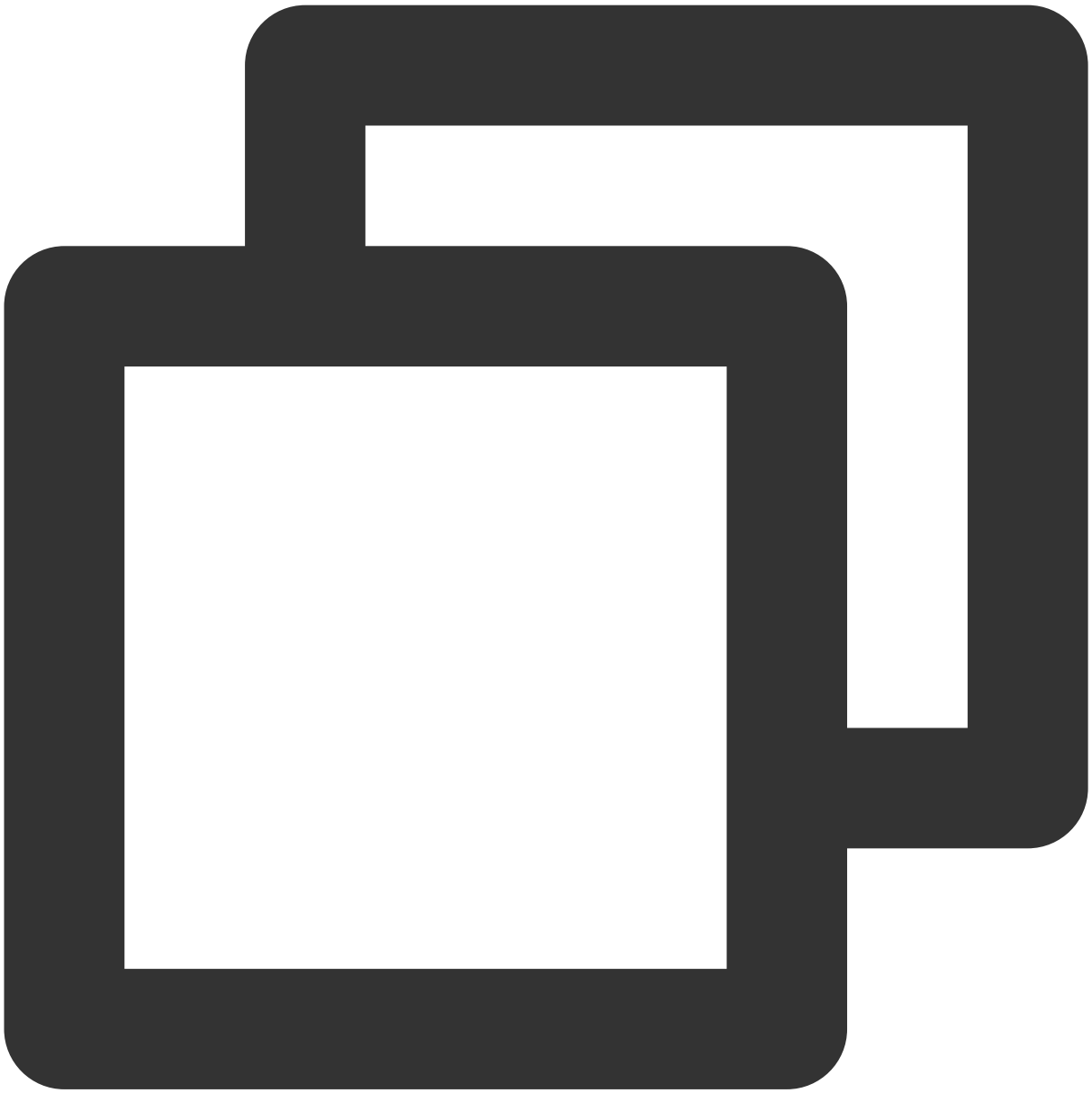


```
POST /<ObjectKey>?ci-process=ImageSearch&action=AddImage HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Content-Length: 100
Content-Type: application/xml

<Request>
    <EntityId>apple</EntityId>
    <CustomContent>myapple</CustomContent>
    <Tags>{"key1":"val1","key2":"val2"}</Tags>
</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: 0
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
```

版权所有：腾讯云计算（北京）有限责任公司

图片搜索接口

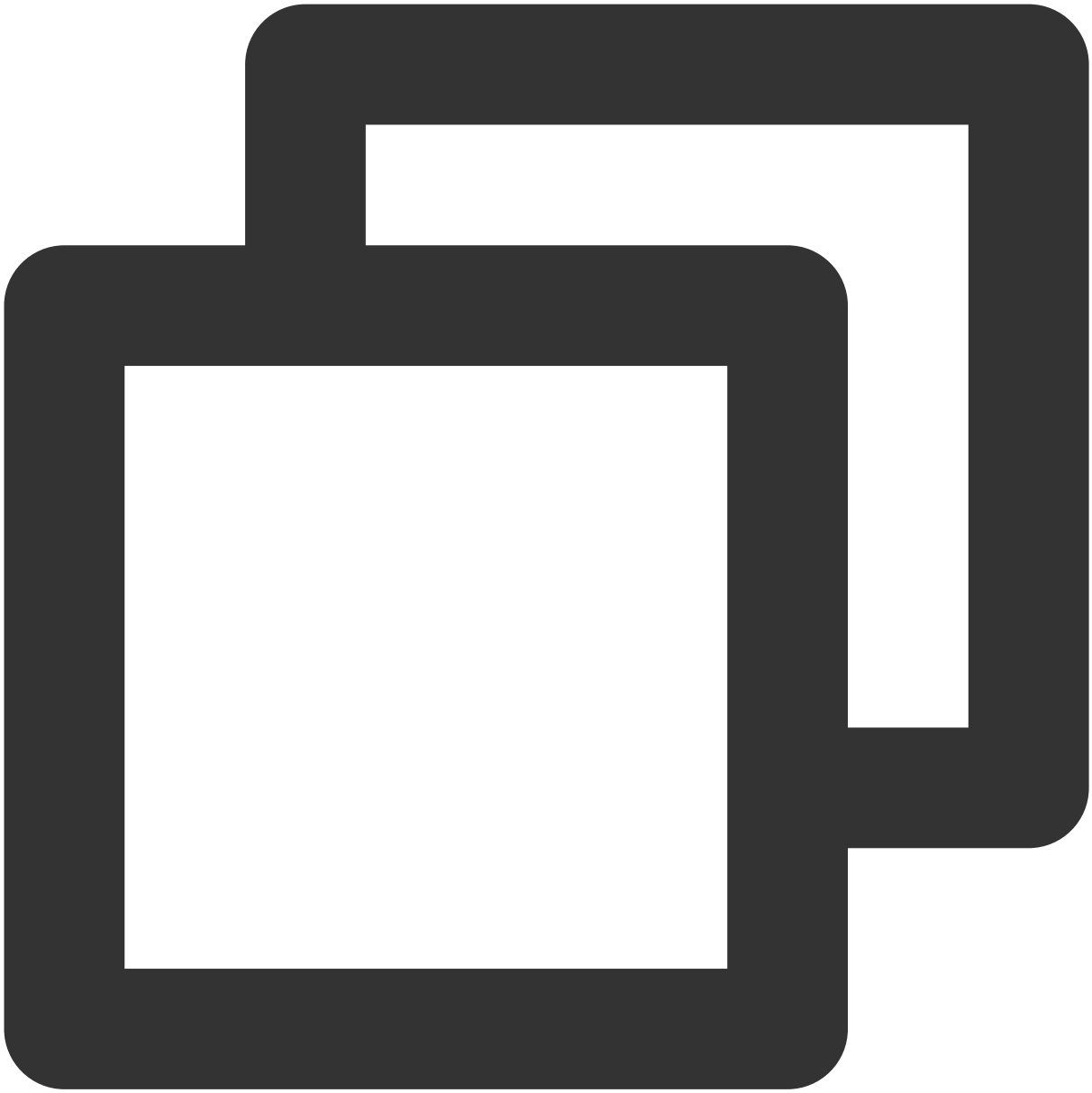
最近更新时间：2024-01-20 09:22:44

功能描述

该接口用于检索图片。

请求

请求示例



```
GET /<ObjectKey>?ci-process=ImageSearch&action=SearchImage&MatchThreshold=&Offset=
Host: <BucketName-APPID>.cos.<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）。

请求参数

参数的具体内容如下：

节点名称（关键字）	描述	类型	是否必选
MatchThreshold	出参 Score 中，只有超过 MatchThreshold 值的结果才会返回。默认为0	Int	否
Offset	起始序号，默认值为0	Int	否
Limit	返回数量，默认值为10，最大值为100	Int	否
Filter	针对入库时提交的 Tags 信息进行条件过滤。支持>、>=、<、<=、=、!=，多个条件之间支持 AND 和 OR 进行连接	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。

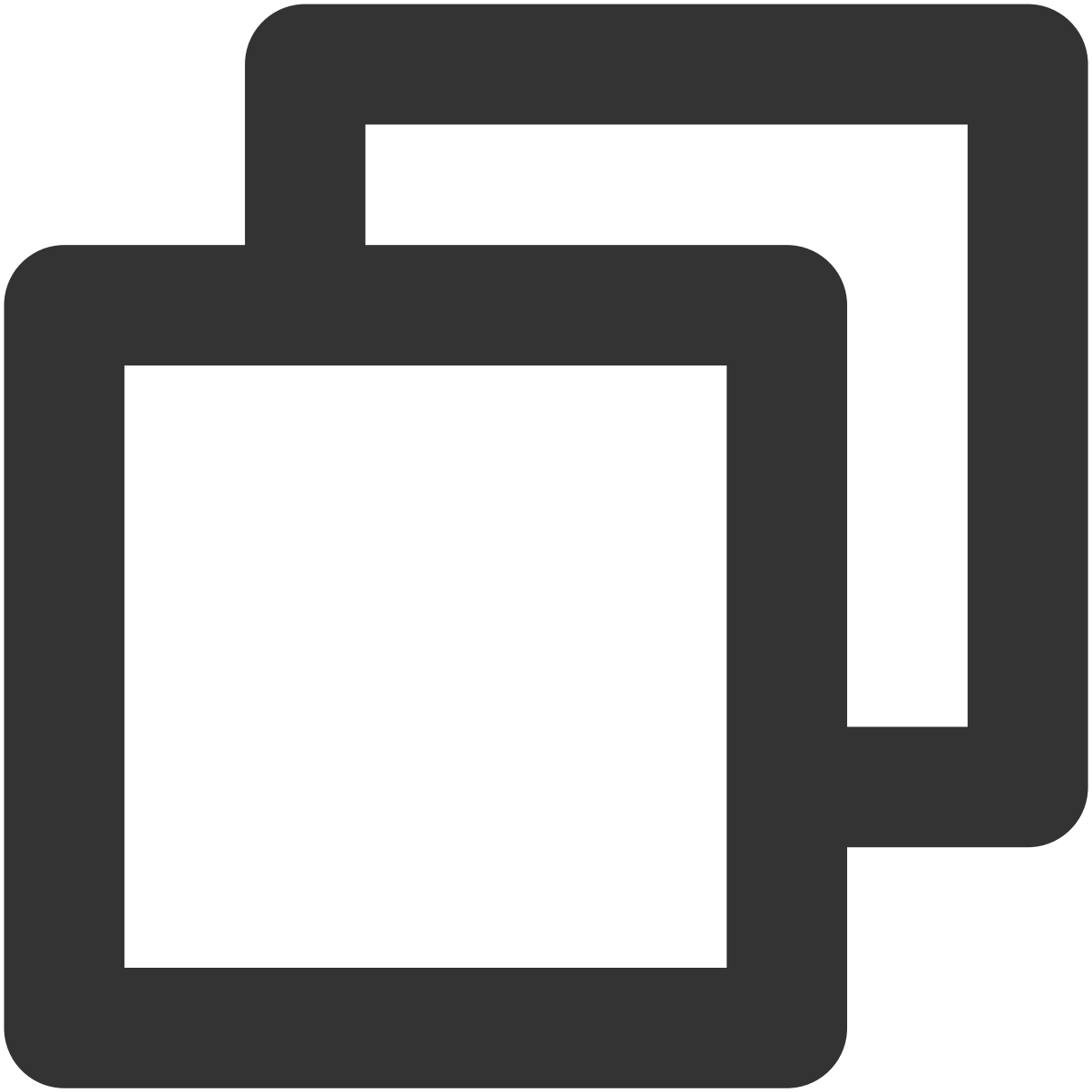
响应

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>
    <Count>2</Count>
    <ImageInfos>
        <EntityId>apple</EntityId>
        <PicName>ObjectKey1</PicName>
        <Score>99</Score>
        <CustomContent>XXX</CustomContent>
        <Tags>{"key1":"value1","key2":"value2"}</Tags>
    </ImageInfos>
    <ImageInfos>
        <EntityId>apple</EntityId>
```

```
        <PicName>ObjectKey2</PicName>
        <Score>90</Score>
        <CustomContent>XXX</CustomContent>
        <Tags>{"key1":"value1","key2":"value2"}</Tags>
    </ImageInfos>
</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 节点的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
Count	Response	数量	Int
ImageInfos	Response	图片信息	Container

ImageInfos 节点为数组，其中每一项内容为

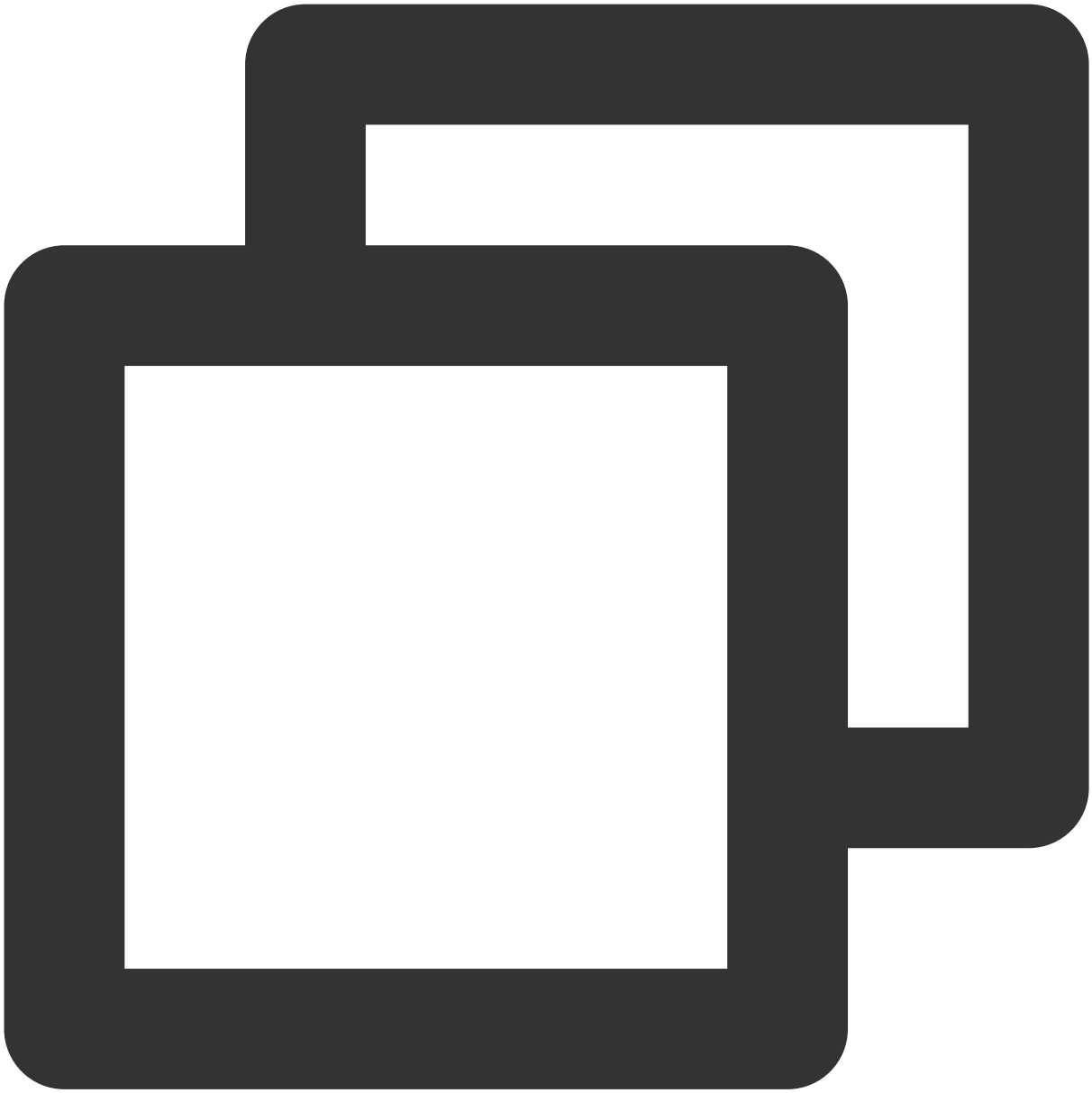
节点名称（关键字）	父节点	描述	类型
EntityId	ImageInfos	物品 ID	String
CustomContent	ImageInfos	图片自定义的内容	String
Tags	ImageInfos	自定义标签信息，json 字符串	String
PicName	ImageInfos	图片名称	String
Score	ImageInfos	相似度	Int

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。

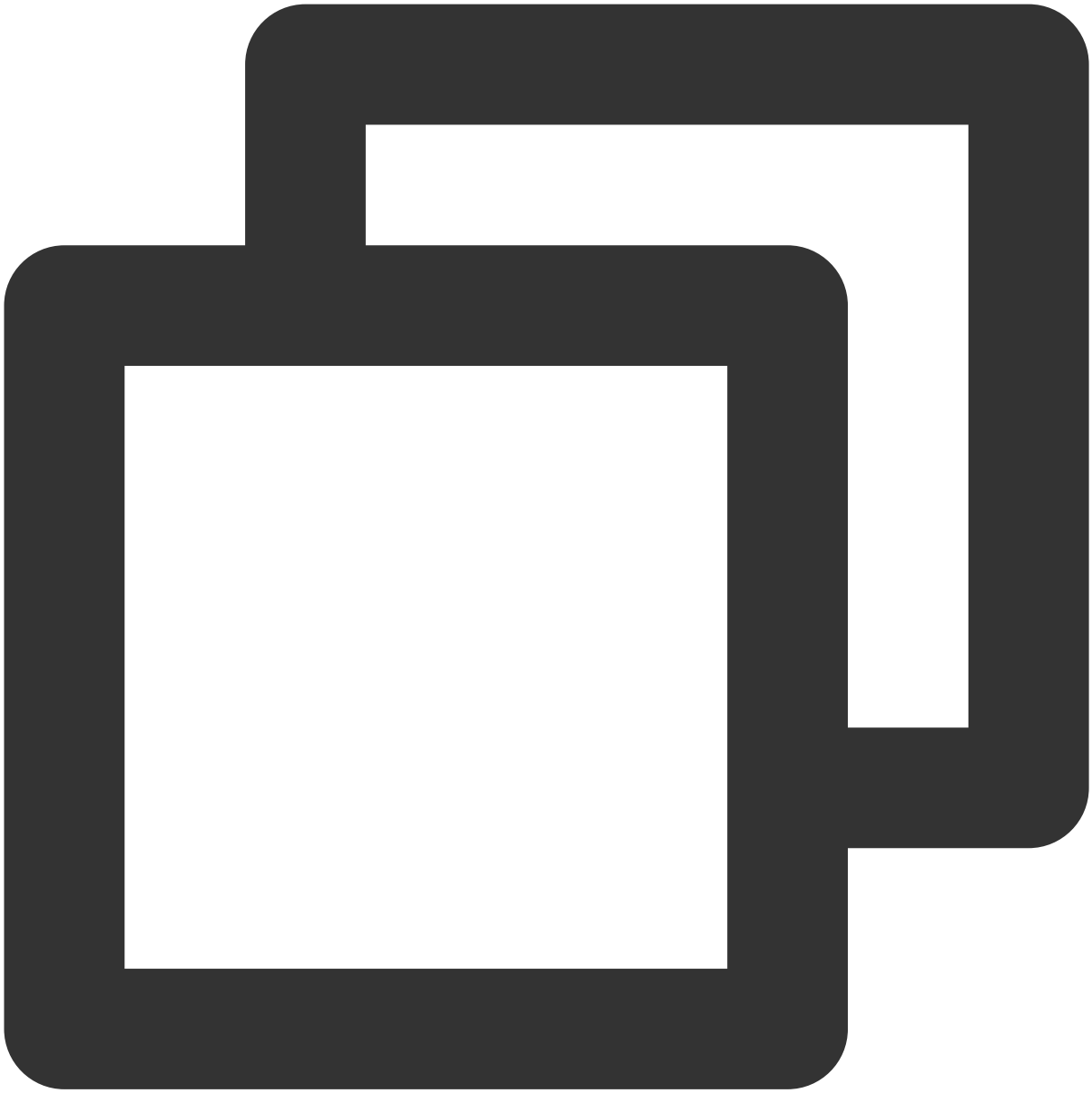
实际案例

请求



```
GET /<ObjectKey>?ci-process=ImageSearch&action=SearchImage&MatchThreshold=10&Offset
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 150
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <Count>2</Count>
    <ImageInfos>
```

```
        <EntityId>apple</EntityId>
        <PicName>ObjectKey</PicName>
        <Score>99</Score>
        <CustomContent>XXX</CustomContent>
        <Tags>{"key1":"value1","key2":"value2"}</Tags>
    </ImageInfos>
    <ImageInfos>
        <EntityId>apple</EntityId>
        <PicName>ObjectKey2</PicName>
        <Score>90</Score>
        <CustomContent>XXX</CustomContent>
        <Tags>{"key1":"value1","key2":"value2"}</Tags>
    </ImageInfos>
</Response>
```

删除图库图片

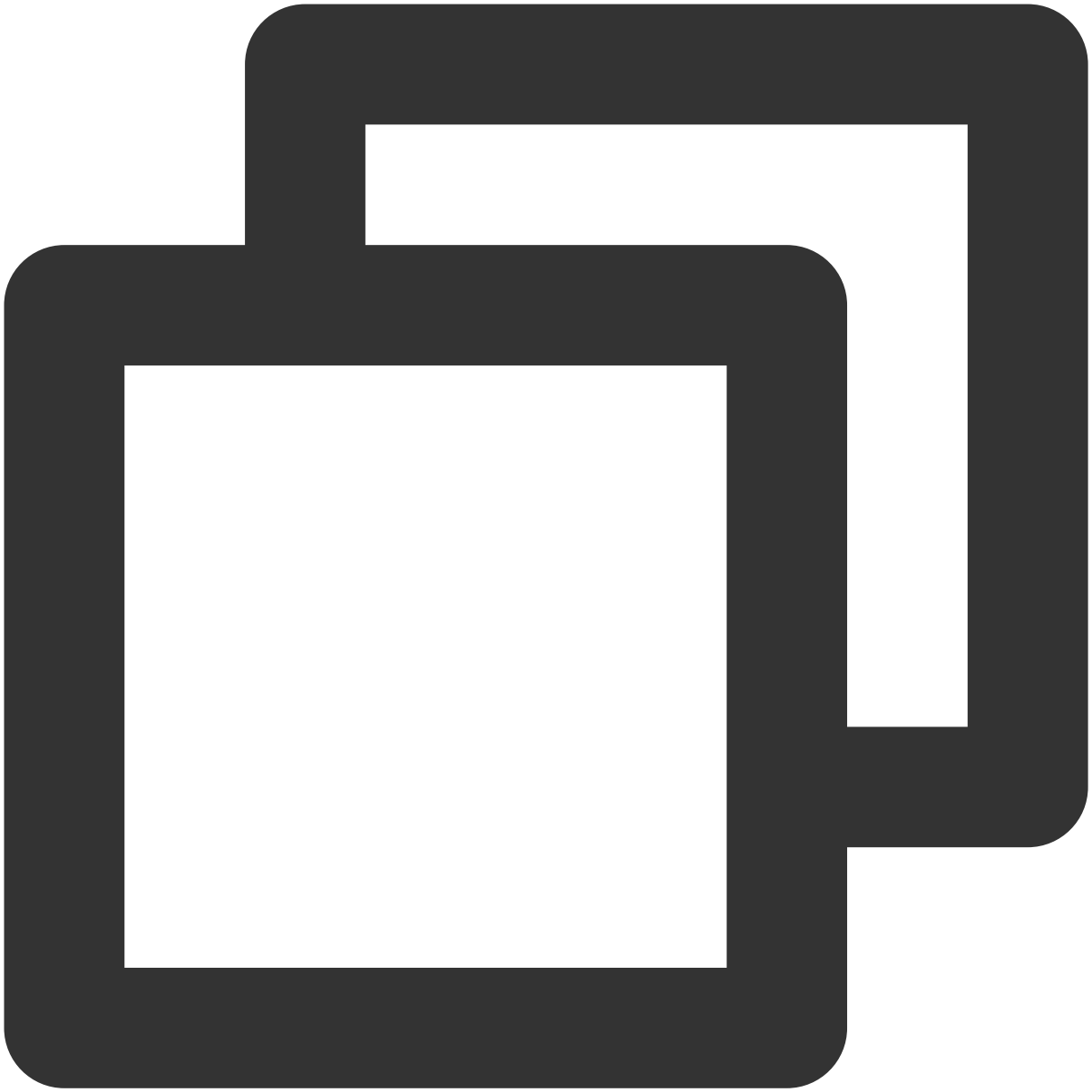
最近更新时间：2024-01-20 09:22:44

功能描述

该接口用于删除图库图片。

请求

请求示例



```
POST /<ObjectKey>?ci-process=ImageSearch&action=DeleteImage HTTP/1.1
Host: <BucketName-APPID>.cos.<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

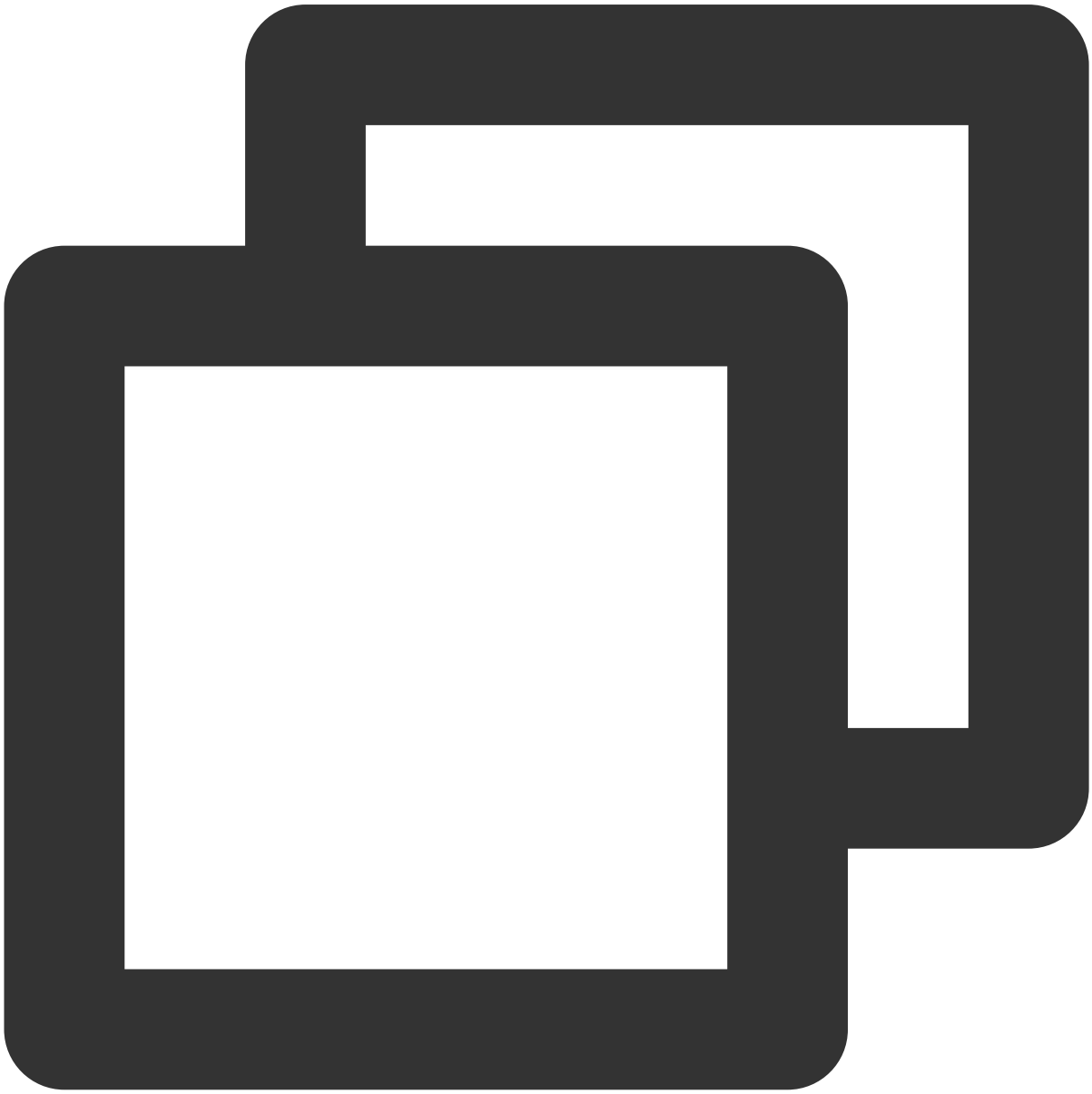
此接口无请求参数。

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

该请求操作的实现需要有如下请求体。



```
<Request>
    <EntityId>apple</EntityId>
</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

Request 节点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
EntityId	Request	物品 ID	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

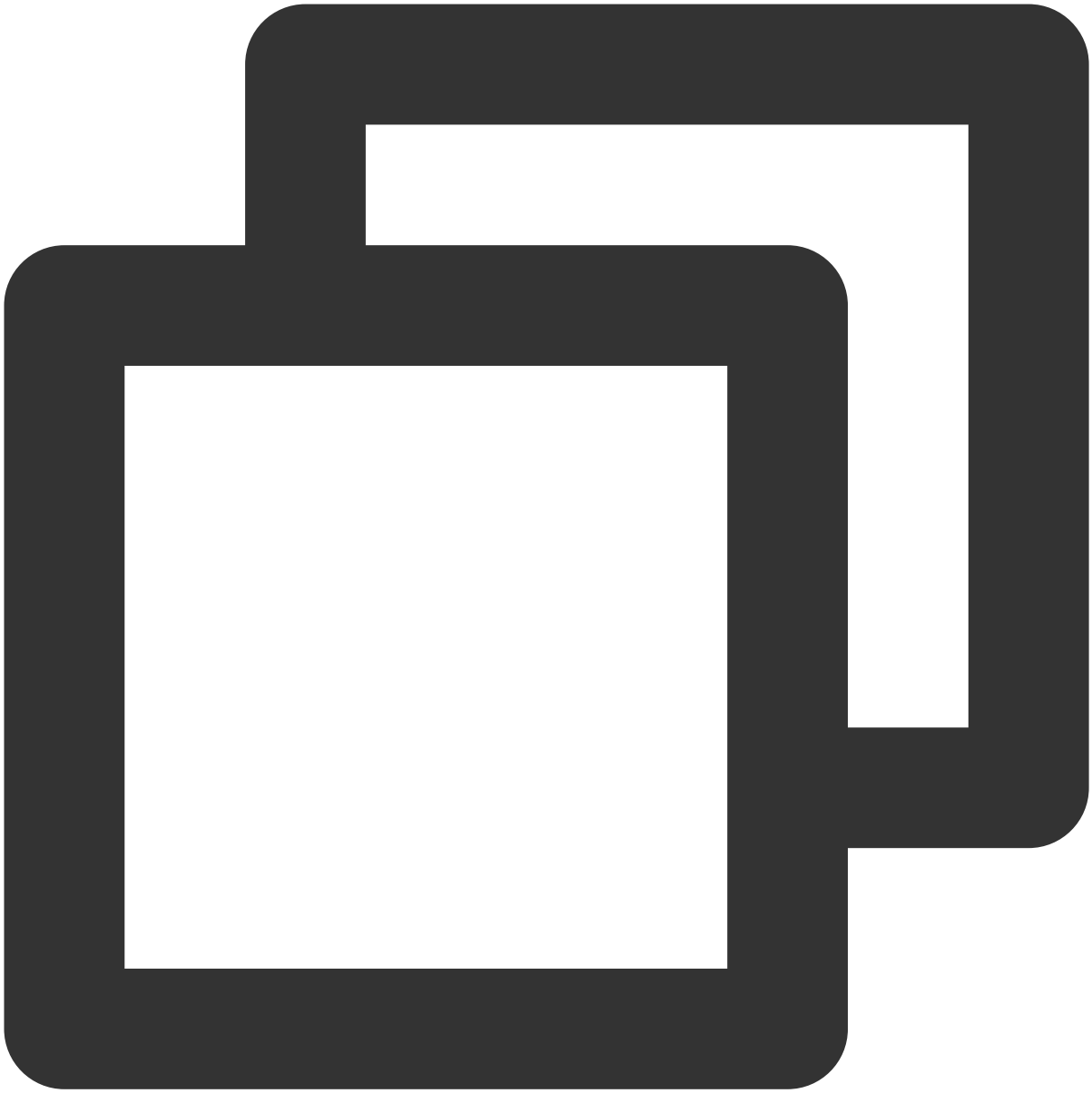
该响应无响应体。

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

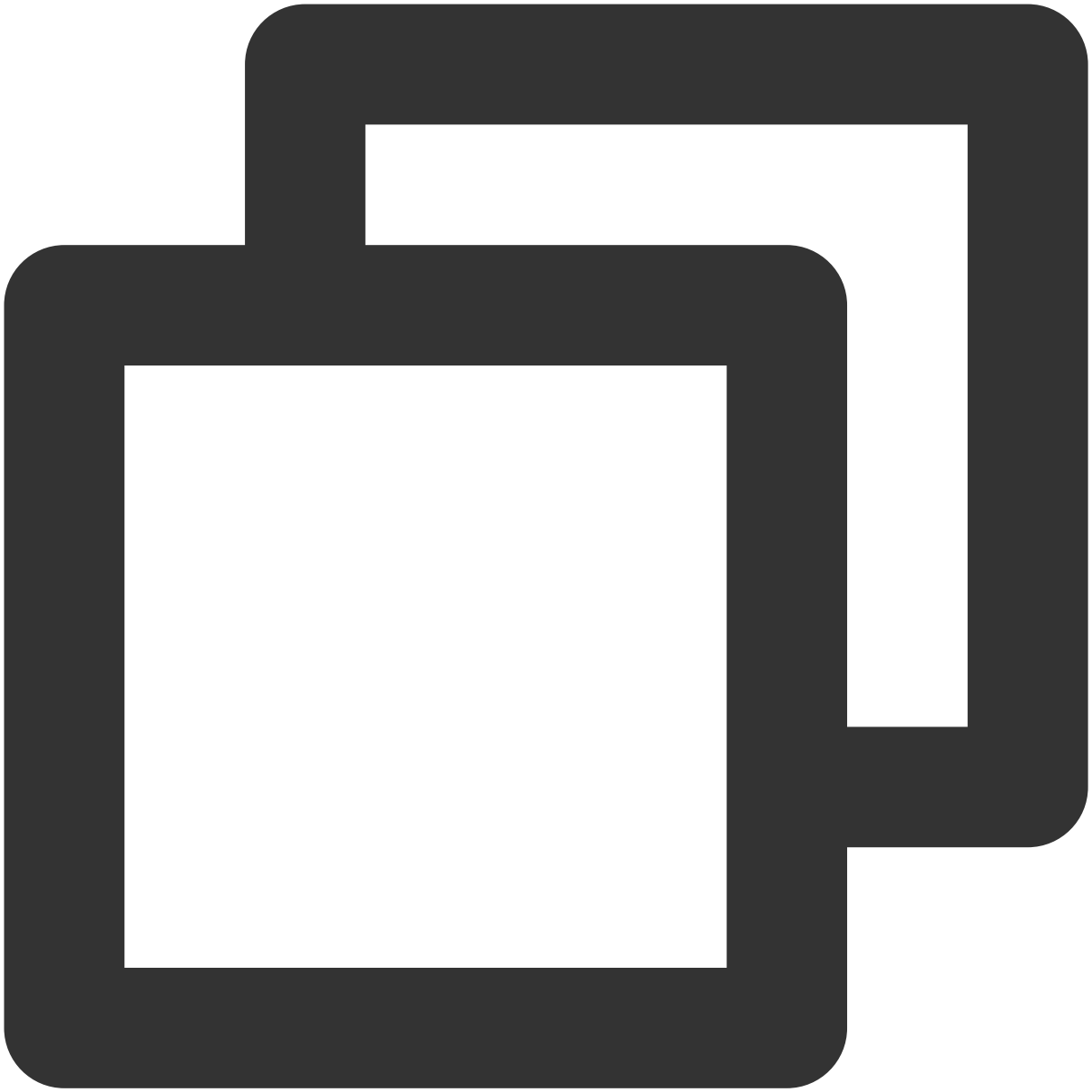
请求



```
POST /<ObjectKey>?ci-process=ImageSearch&action=DeleteImage HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Content-Length: 20
Content-Type: application/xml

<Request>
    <EntityId>apple</EntityId>
</Request>
```

响应



```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
```

队列接口
查询队列

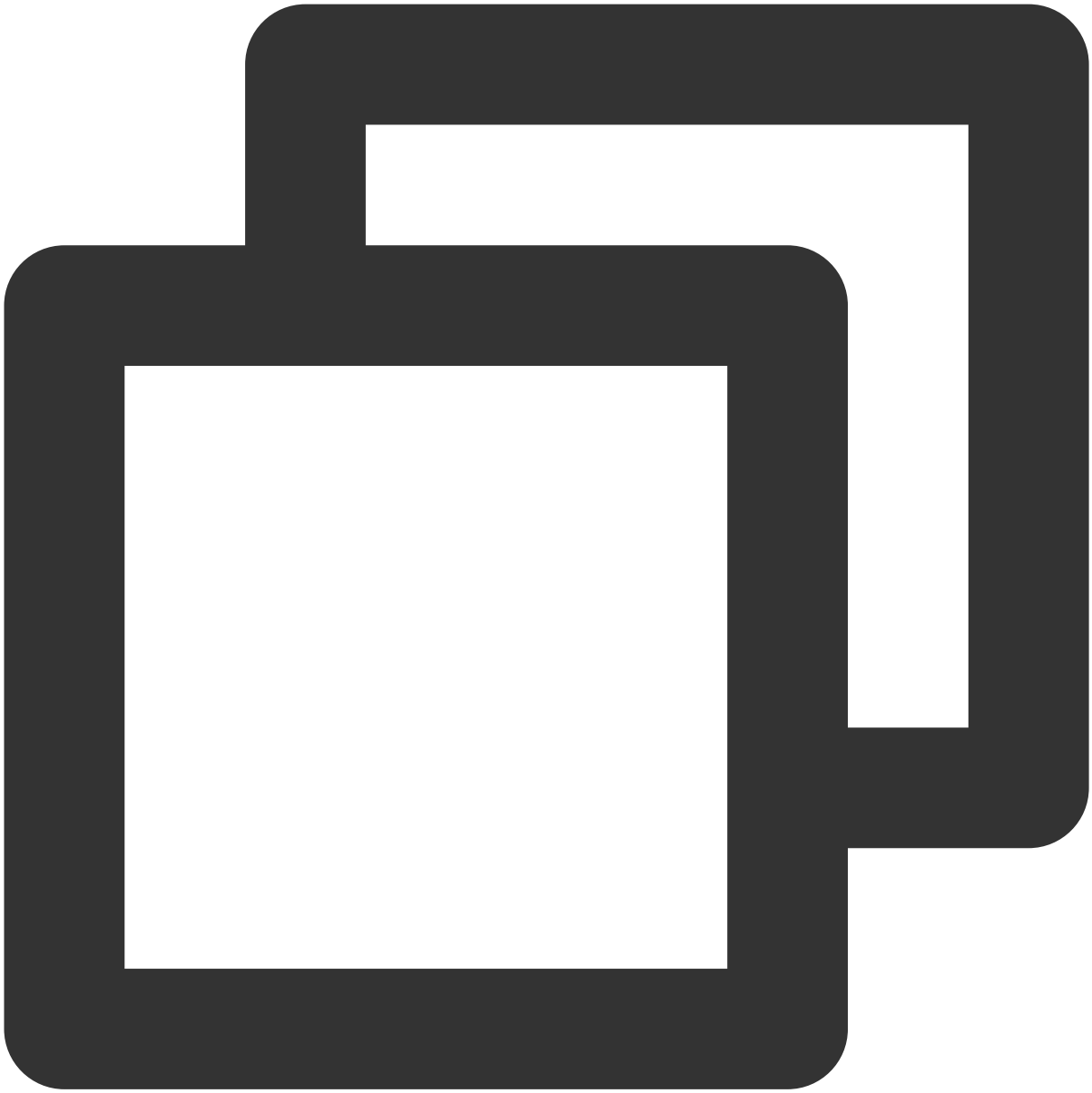
最近更新时间：2024-01-20 09:22:44

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
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

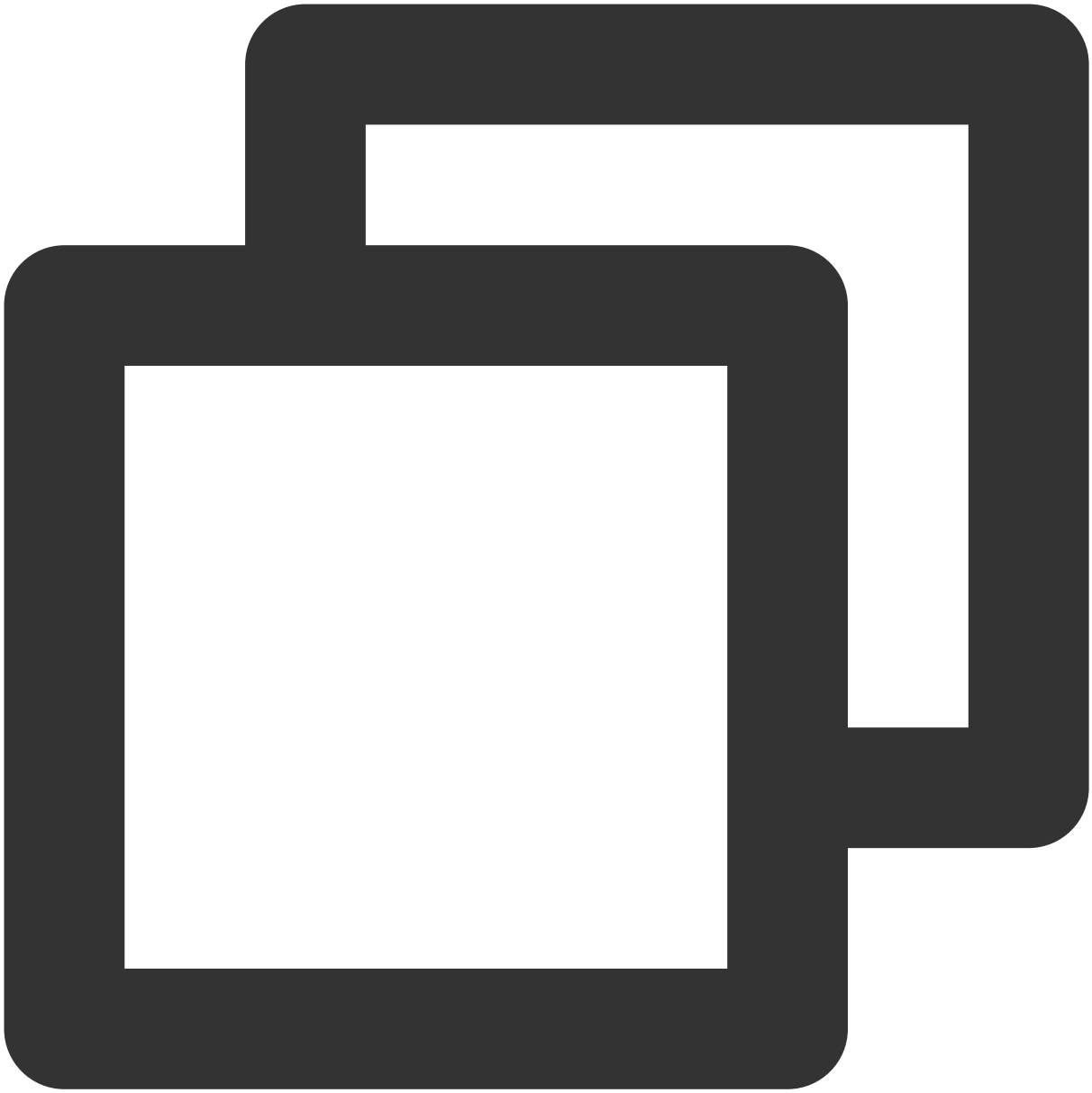
响应

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>
```

```
            <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

版权所有：腾讯云计算（北京）有限责任公司

NotifyConfig	Response.QueueList	回调配置	Container
MaxSize	Response.QueueList	队列最大长度	Int
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：

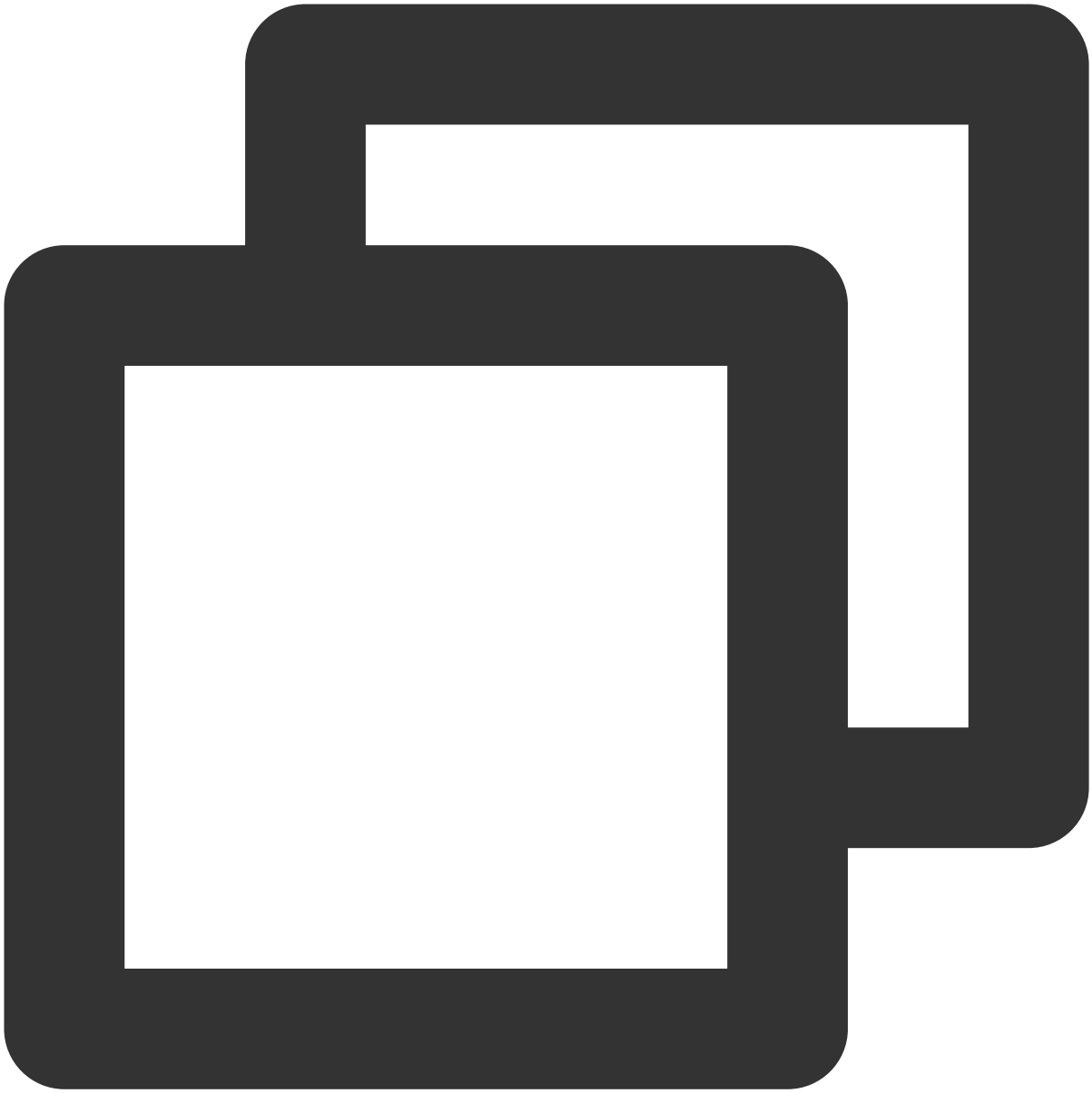
节点名称（关键字）	父节点	描述	类型
Url	Response.QueueList.NotifyConfig	回调地址	String
State	Response.QueueList.NotifyConfig	开关状态，On 或者 Off	String
Type	Response.QueueList.NotifyConfig	回调类型，Url	String
Event	Response.QueueList.NotifyConfig	任务完成：TaskFinish；工作流完成：WorkflowFinish	String
ResultFormat	Response.QueueList.NotifyConfig	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。

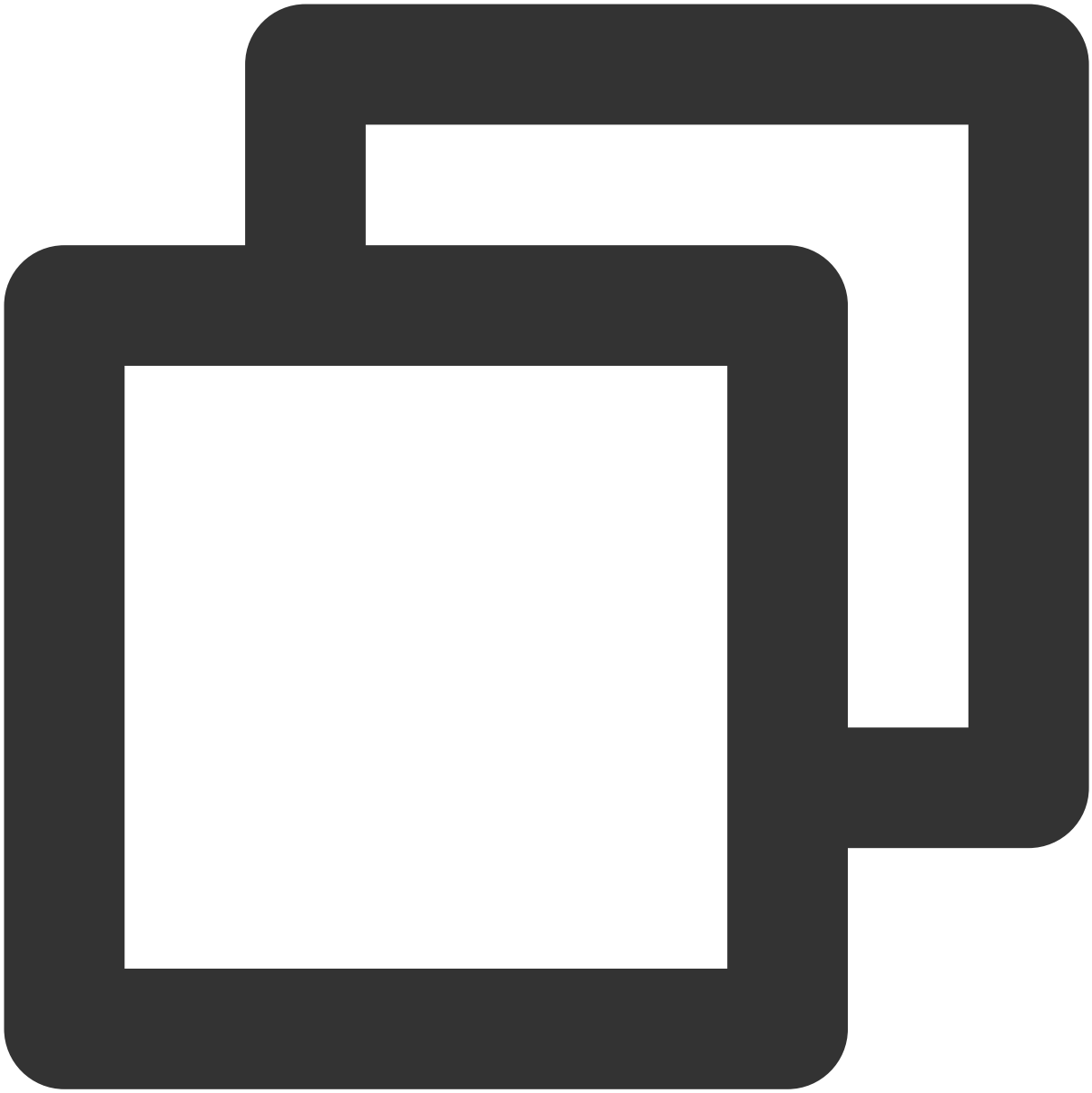
实际案例

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

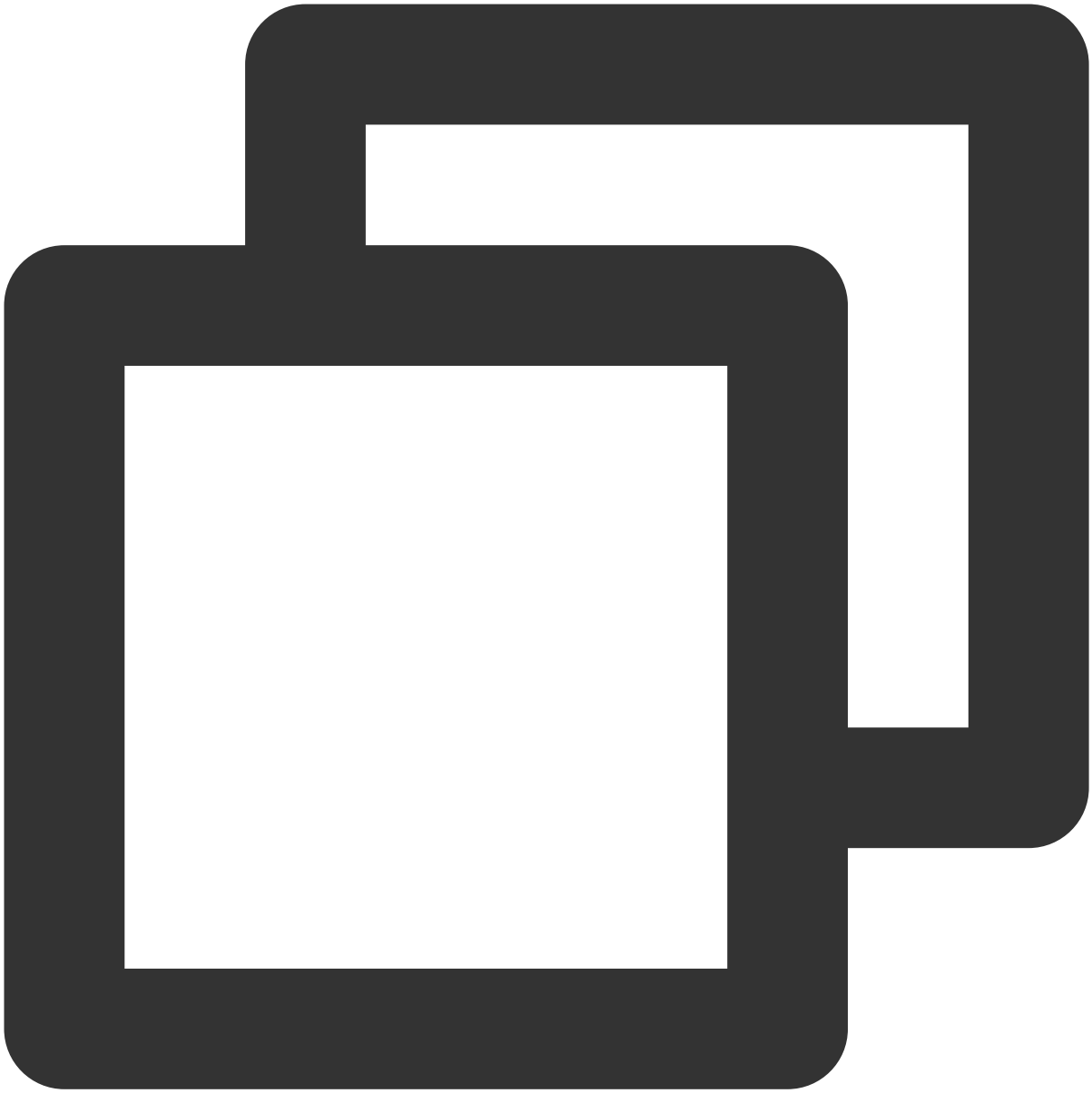


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x

<Response>
    <RequestId>NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x</RequestId>
    <QueueList>
```

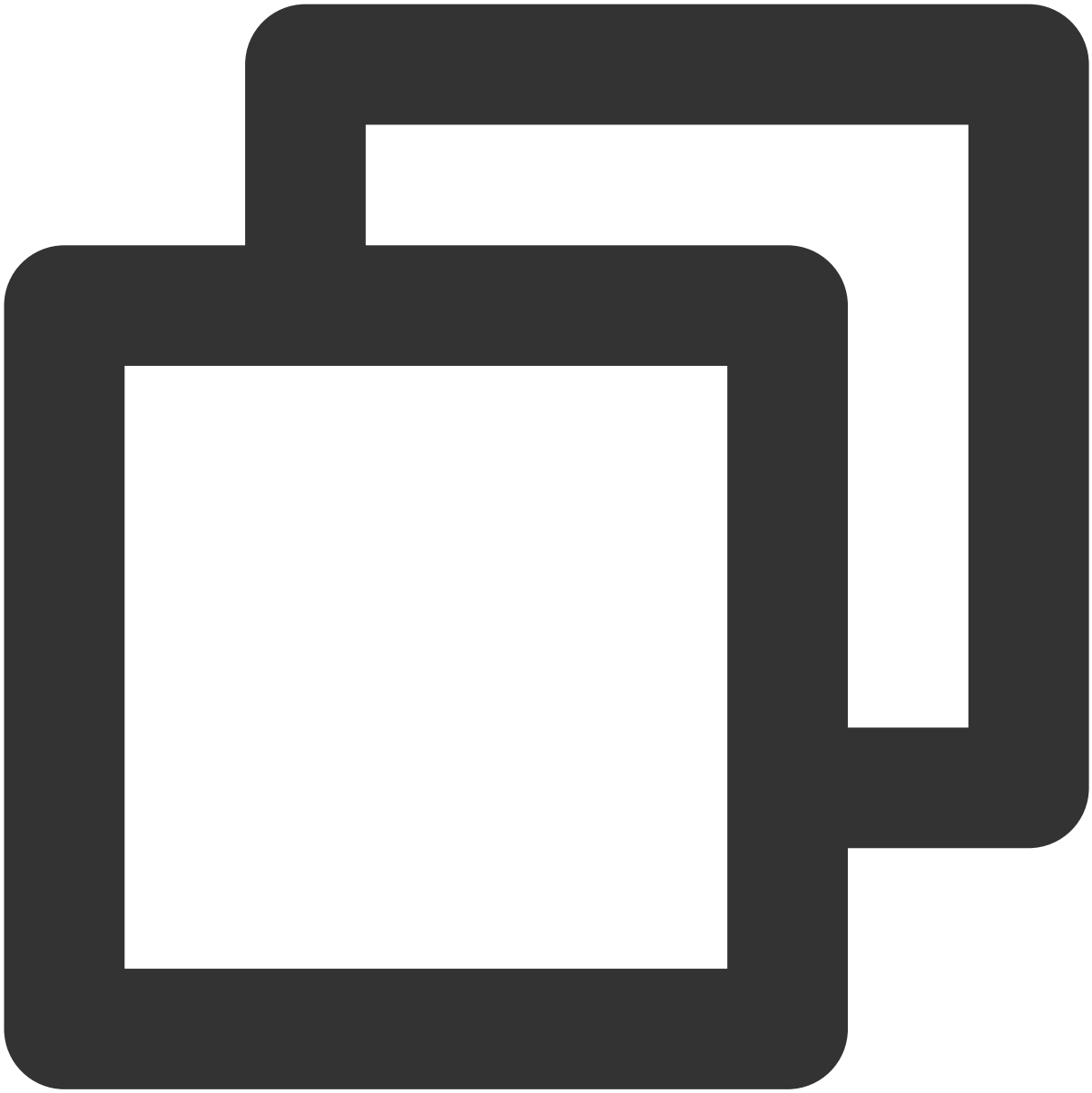
```
        <QueueId>p23f69f4779e64b18aa61e70d5416c071</QueueId>
        <Name>queue-ai-process-1</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <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

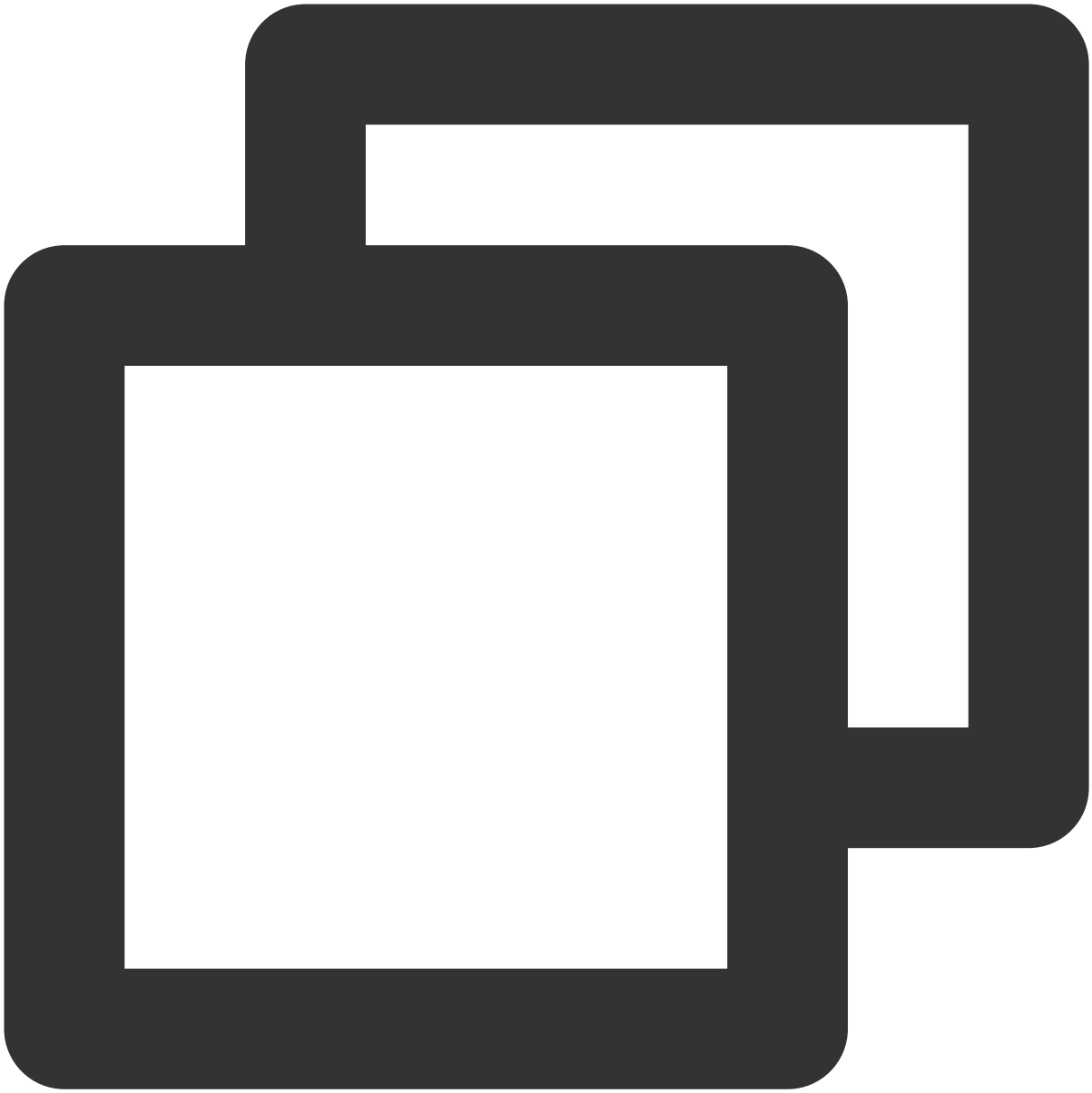
最近更新时间：2024-01-20 09:22:44

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：

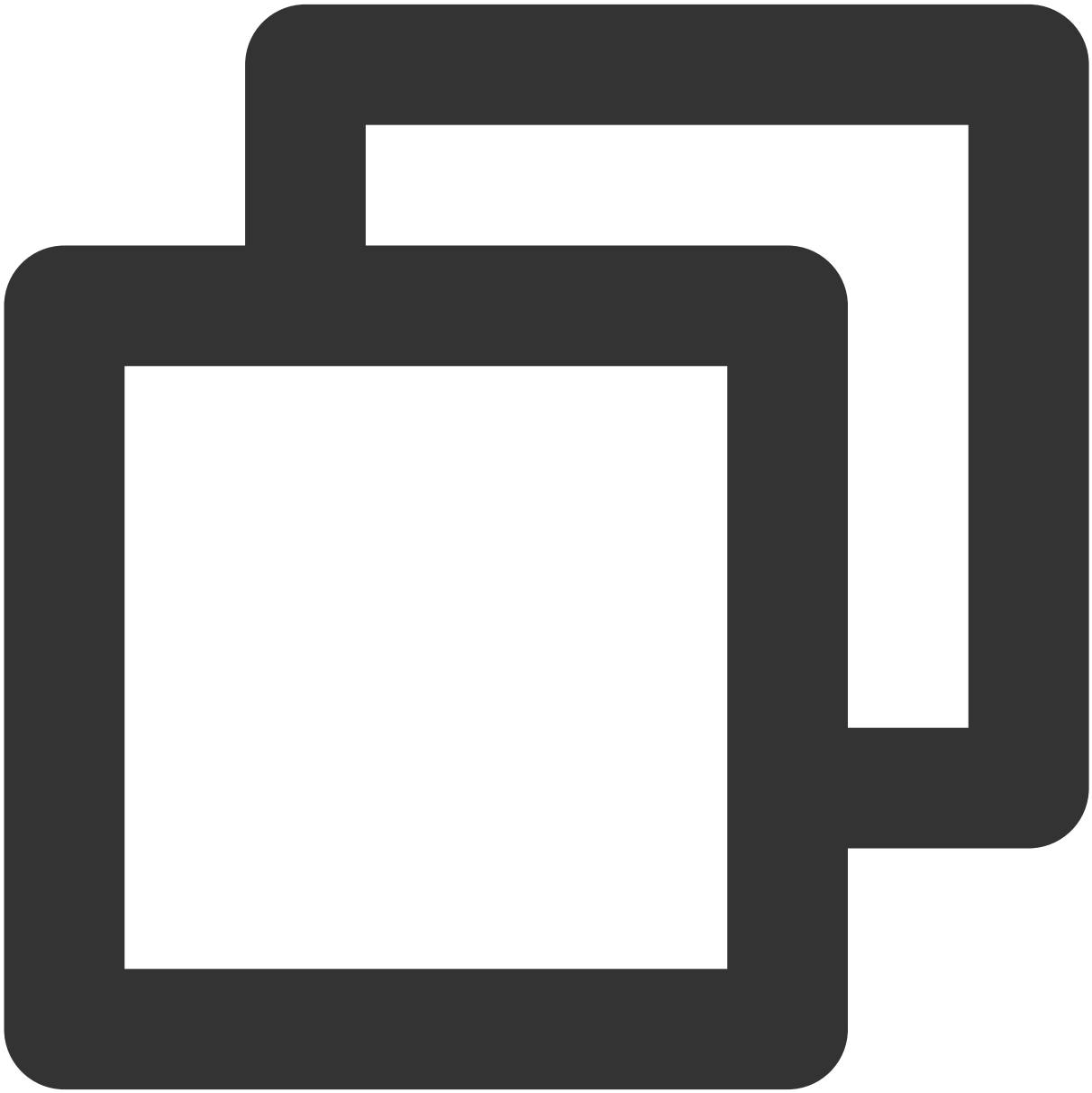
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	On/Off
Event	Request.NotifyConfig	回调事件	String	当 State=On 时，必选	无	任务完成：TaskFinish；工

版权所有：腾讯云计算（北京）有限责任公司

						作流完成：WorkflowFinish
ResultFormat	Request.NotifyConfig	回调格式	String	否	XML	JSON/XML
Type	Request.NotifyConfig	回调类型	String	当 State=On 时, 必选	无	Url 或 TDMQ
Url	Request.NotifyConfig	回调地址	String	当 State=On, 且 Type=Url 时, 必选	无	不能为内网地址
MqMode	Request.NotifyConfig	TDMQ 使用模式	String	当 State=On, 且 Type=TDMQ 时, 必选	Queue	主题订阅：Topic 队列服务：Queue
MqRegion	Request.NotifyConfig	TDMQ 所属园区	String	当 State=On, 且 Type=TDMQ 时, 必选	无	目前支持园区 sh（上海）、bj（北京）、gz（广州）、cd（成都）、hk（中国香港）
MqName	Request.NotifyConfig	TDMQ 主题名称	String	当 State=On, 且 Type=TDMQ 时, 必选	无	无

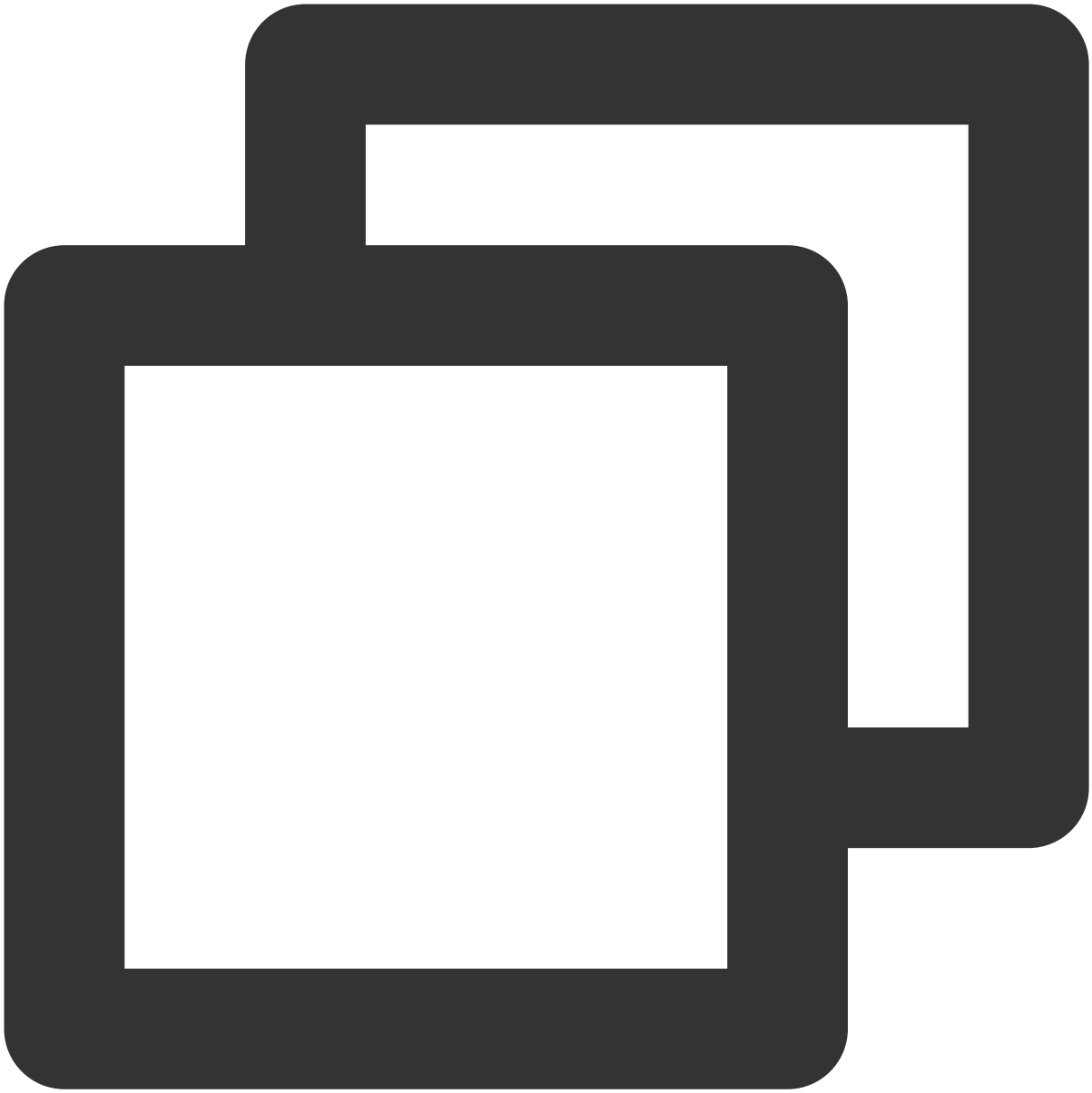
响应

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>
    <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：

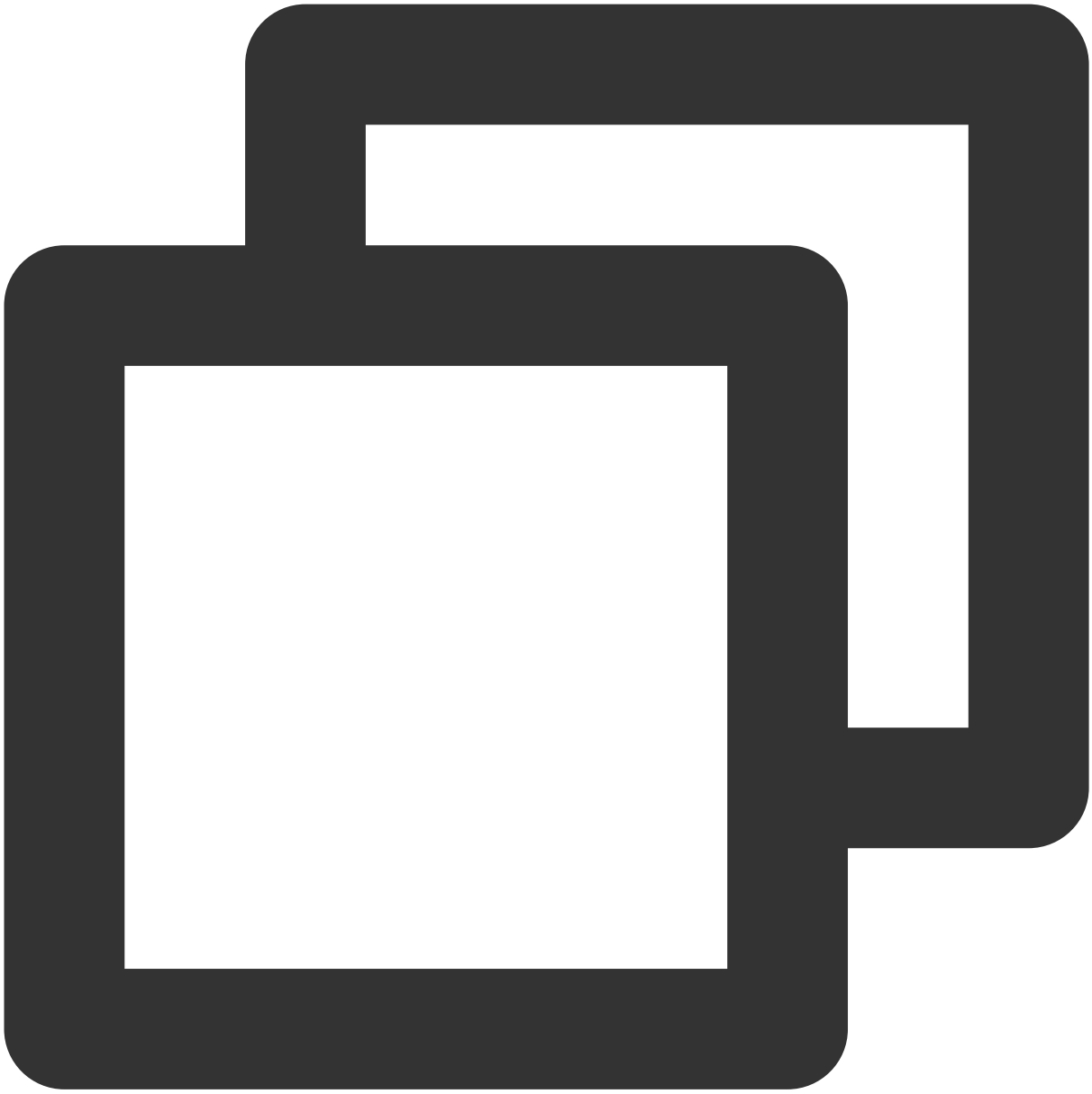
节点名称（关键字）	父节点	描述	类型
QueueId	Response.Queue	队列 ID	String
Name	Response.Queue	队列名字	String
State	Response.Queue	当前状态，Active 或者 Paused	String
NotifyConfig	Response.Queue	回调配置，同 Request.NotifyConfig	Container
MaxSize	Response.Queue	队列最大长度	Int
MaxConcurrent	Response.Queue	当前队列最大并行执行的任务数	Int
Category	Response.Queue	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

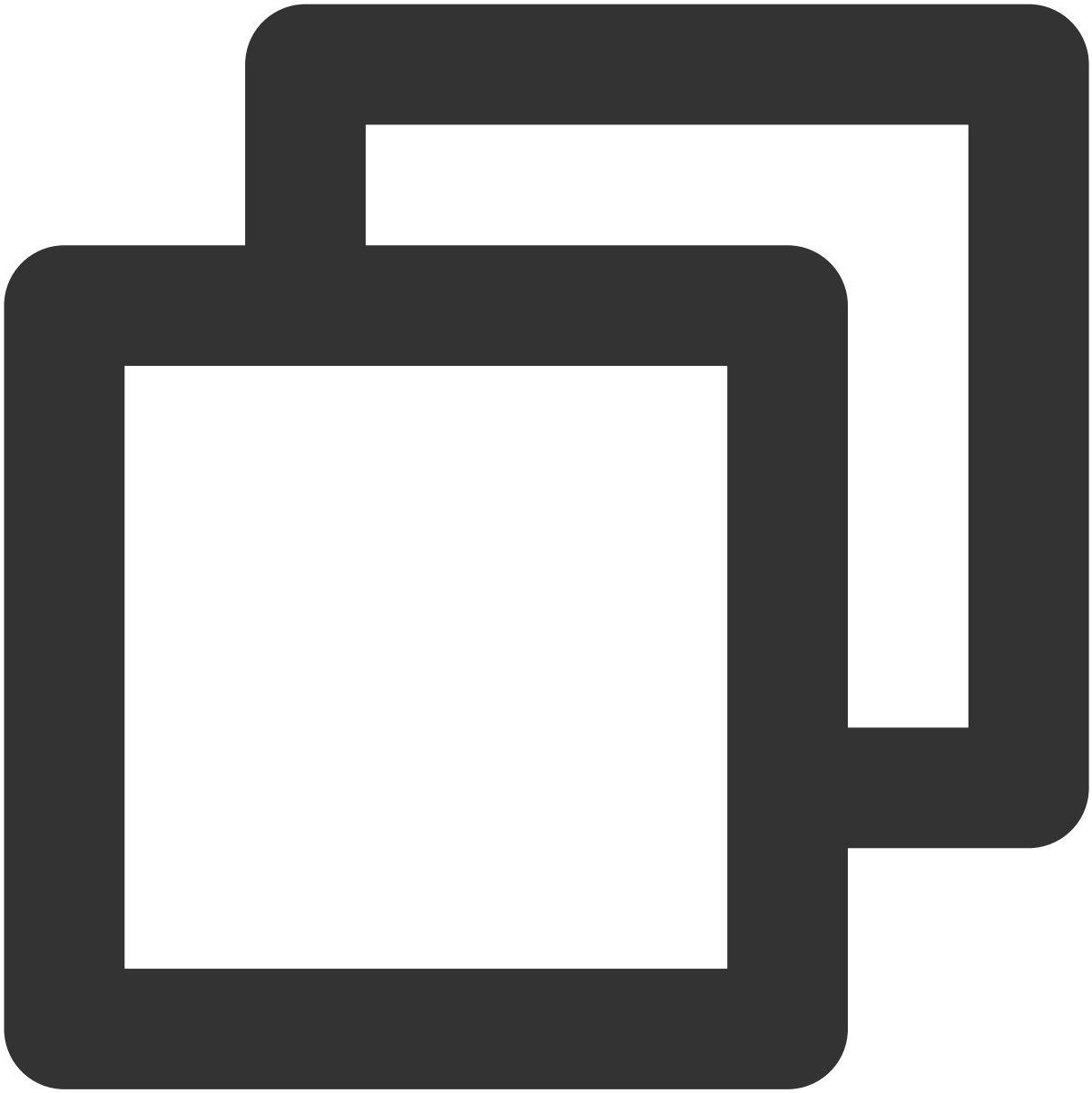
请求



```
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>
```

服务配置
开通 AI 内容识别

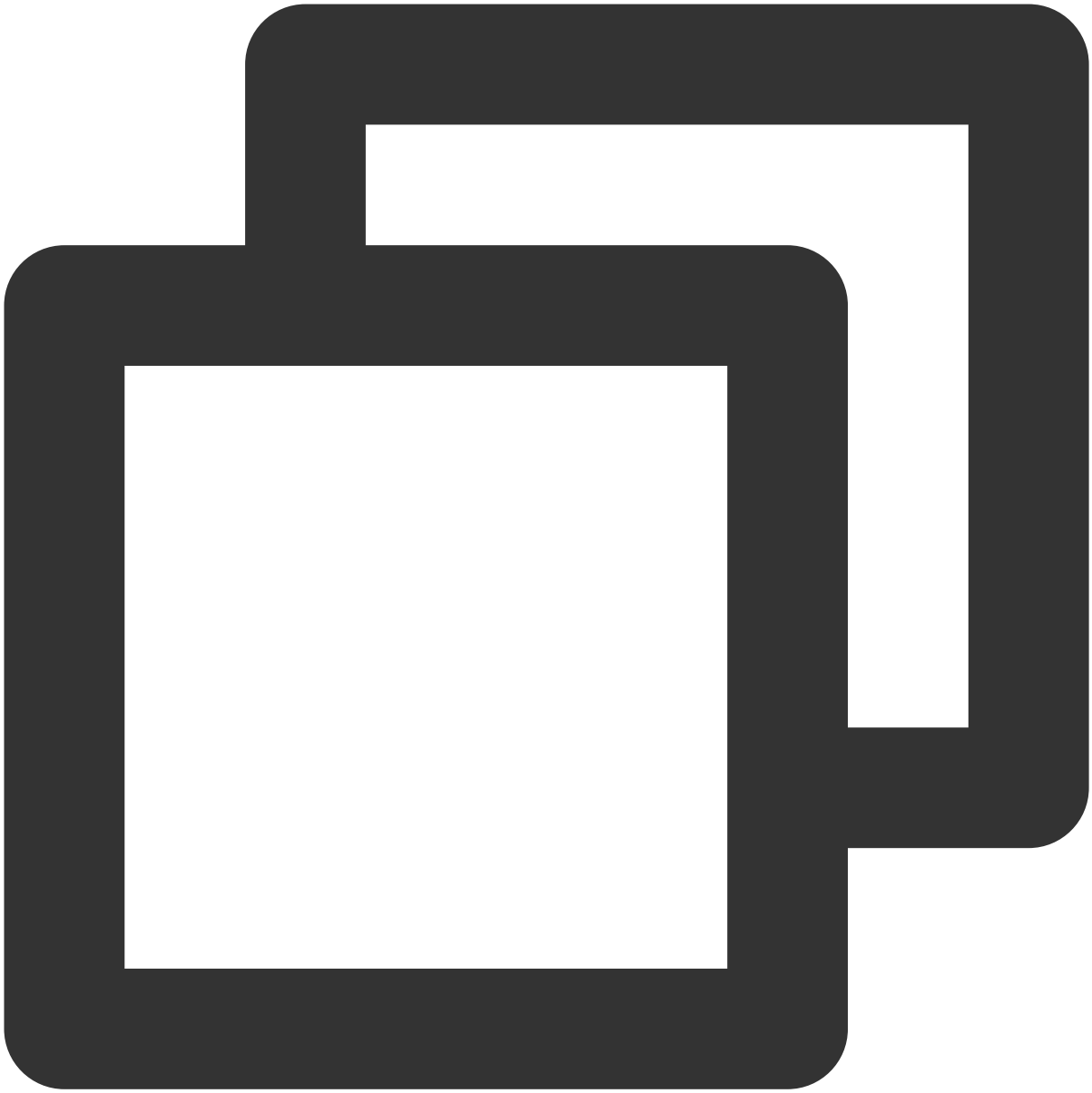
最近更新时间：2024-01-20 09:22:44

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。

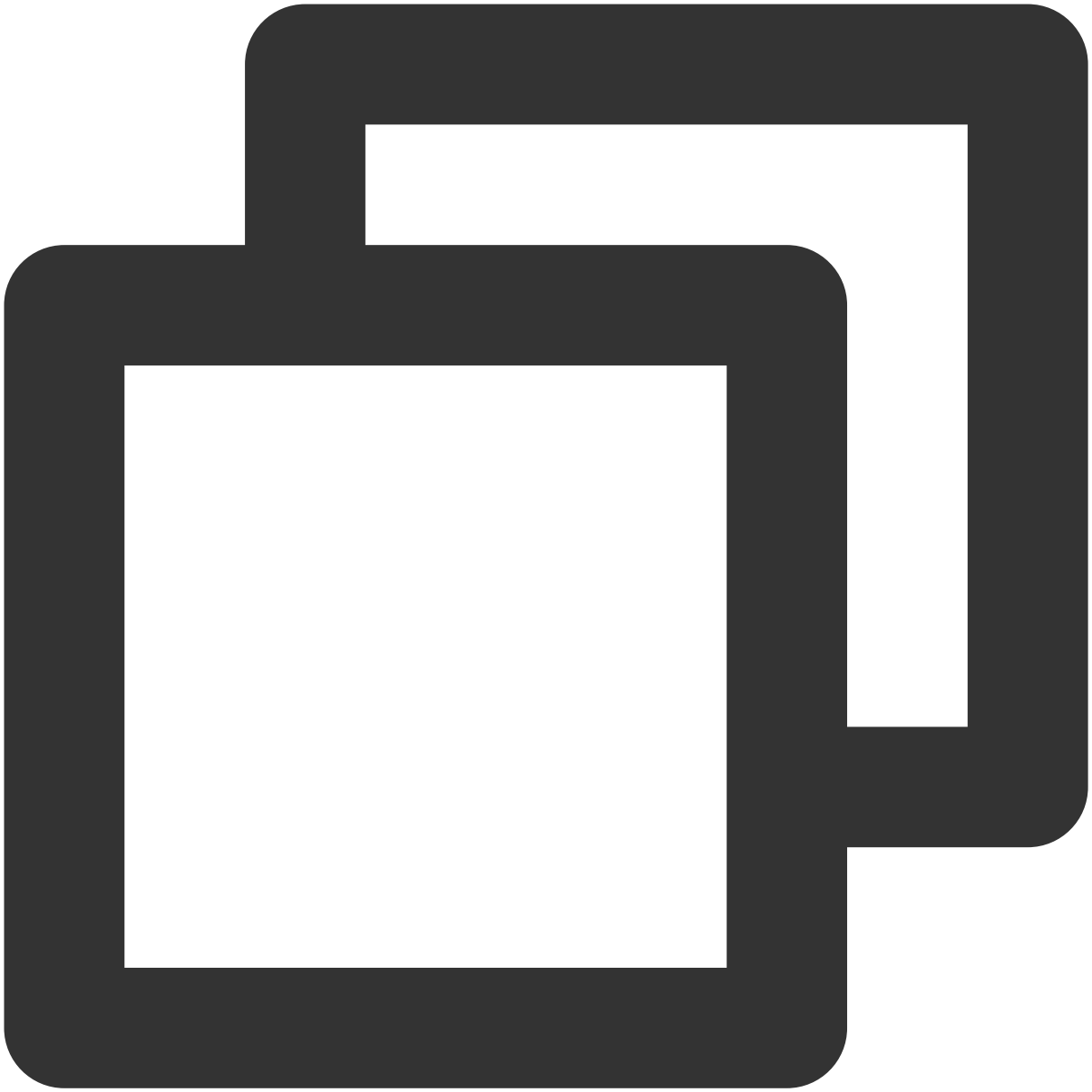
响应

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：

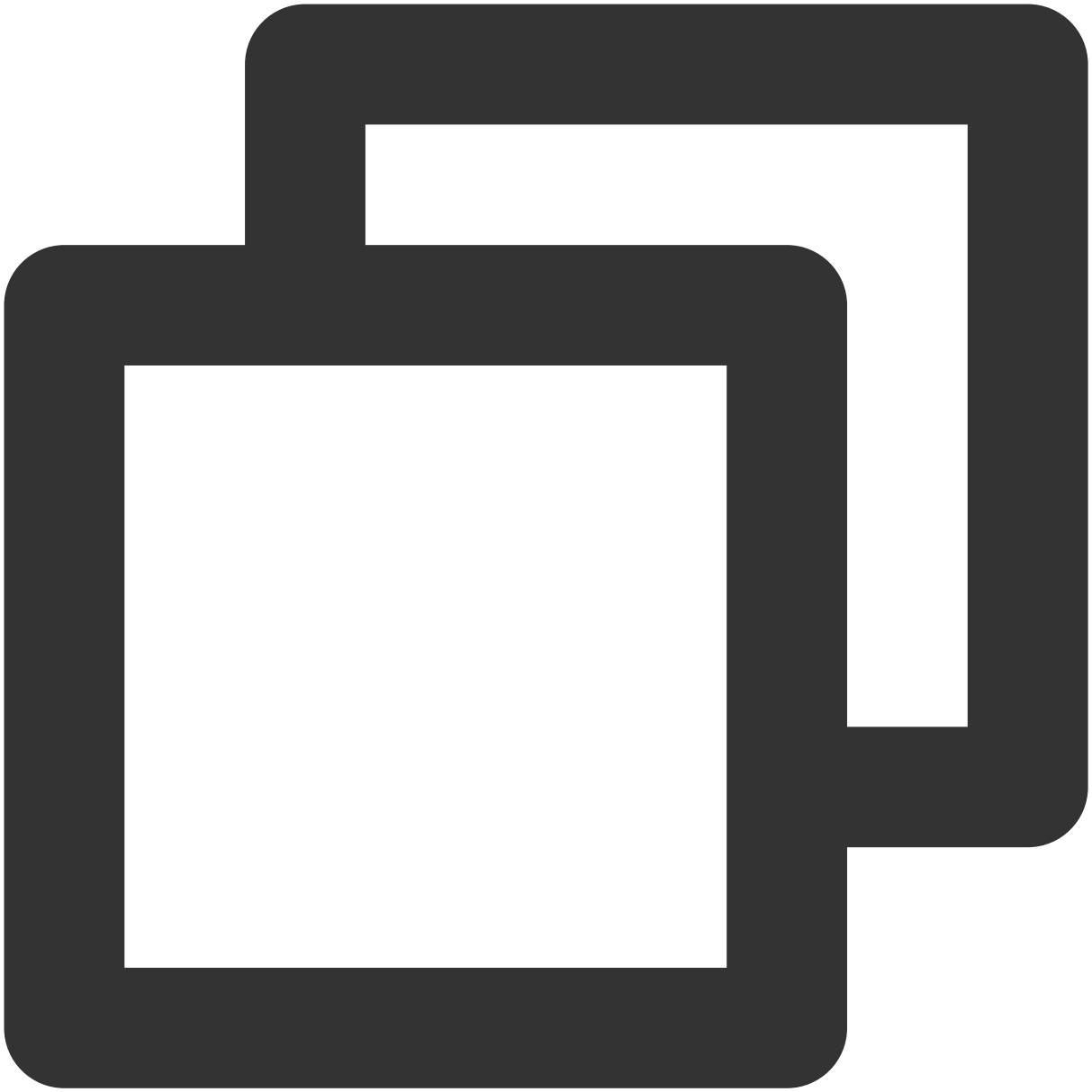
节点名称（关键字）	父节点	描述	类型
BucketId	Response.AiBucket	存储桶 ID	String
Name	Response.AiBucket	存储桶名称，同 BucketId	String
Region	Response.AiBucket	所在的地域	String
CreateTime	Response.AiBucket	创建时间	String

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。

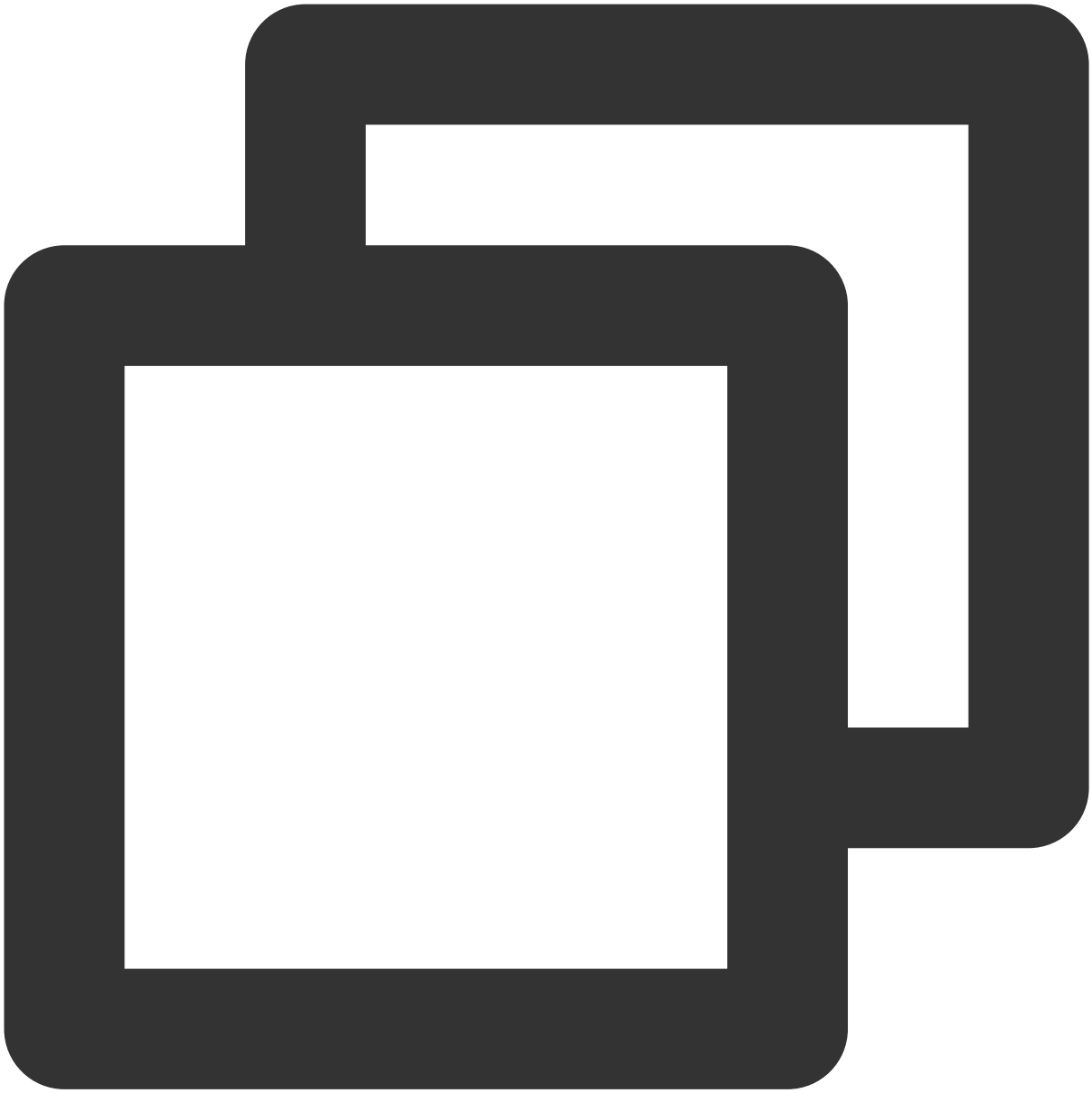
实际案例

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

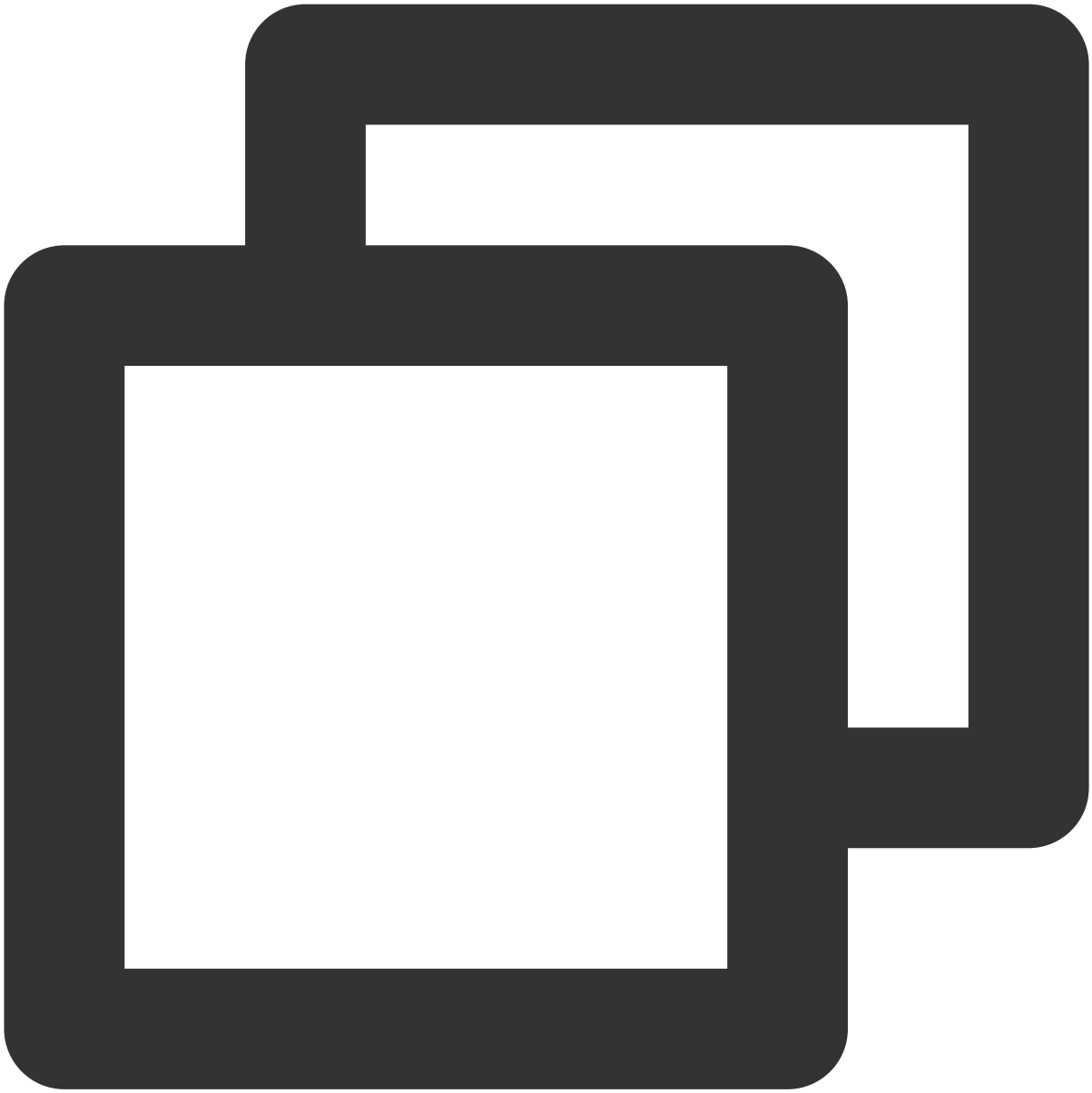
最近更新时间：2024-01-20 09:22:44

功能描述

查询 AI 内容识别服务状态。

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
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

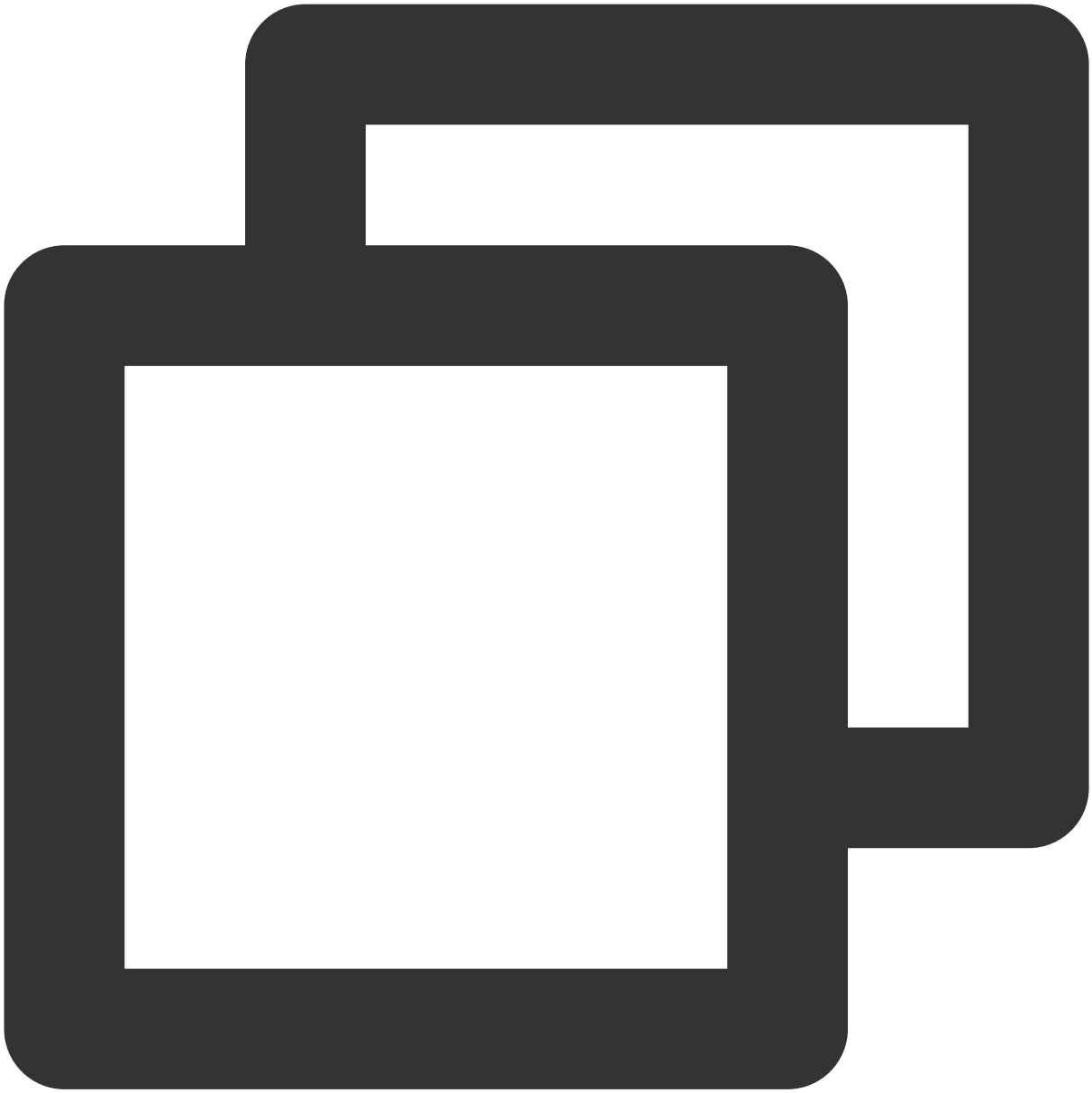
响应

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
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：

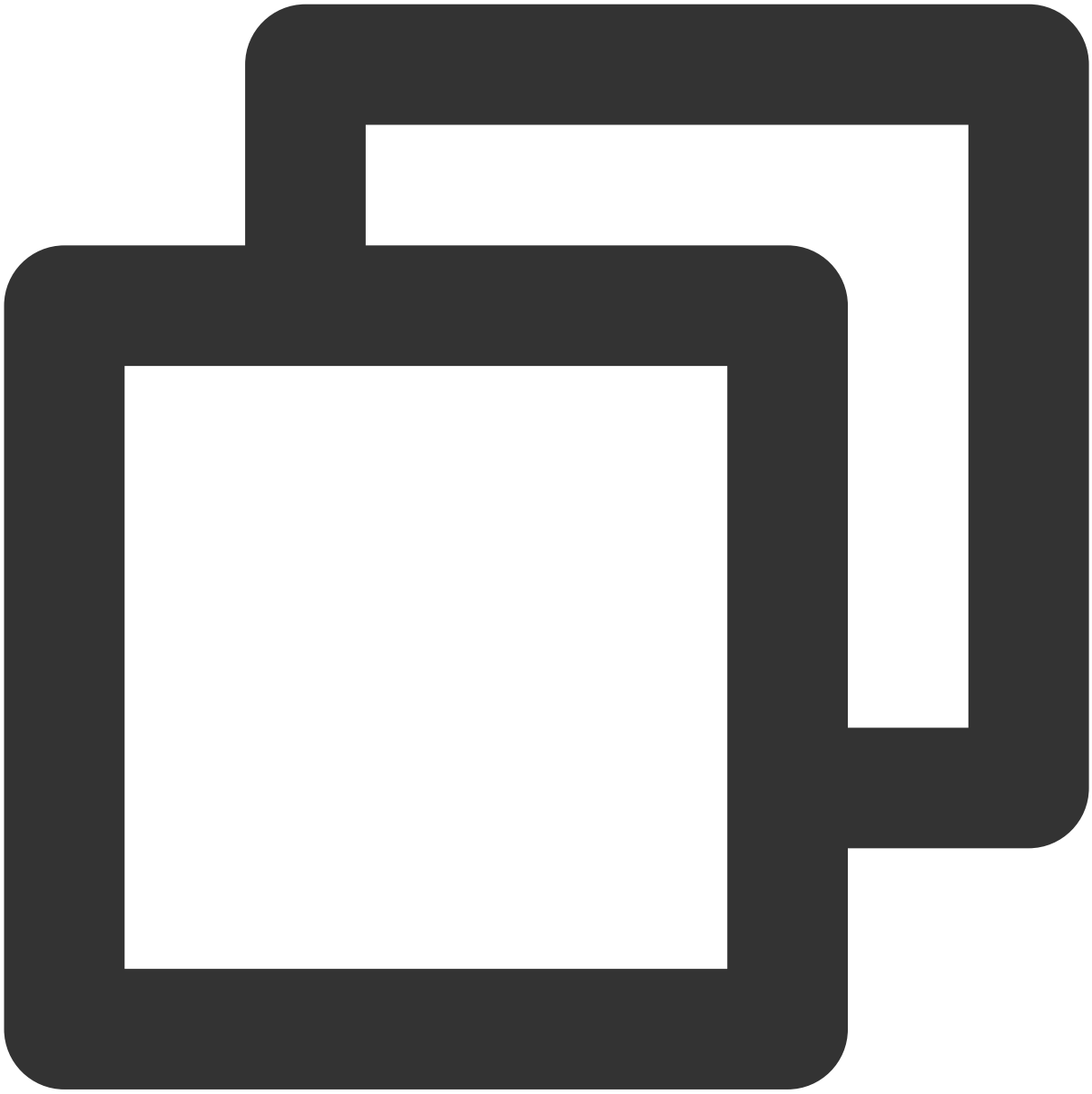
节点名称（关键字）	父节点	描述	类型
BucketId	Response.AiBucketList	存储桶 ID	String
Name	Response.AiBucketList	存储桶名称，同 BucketId	String
Region	Response.AiBucketList	所在的地域	String
CreateTime	Response.AiBucketList	创建时间	String

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。

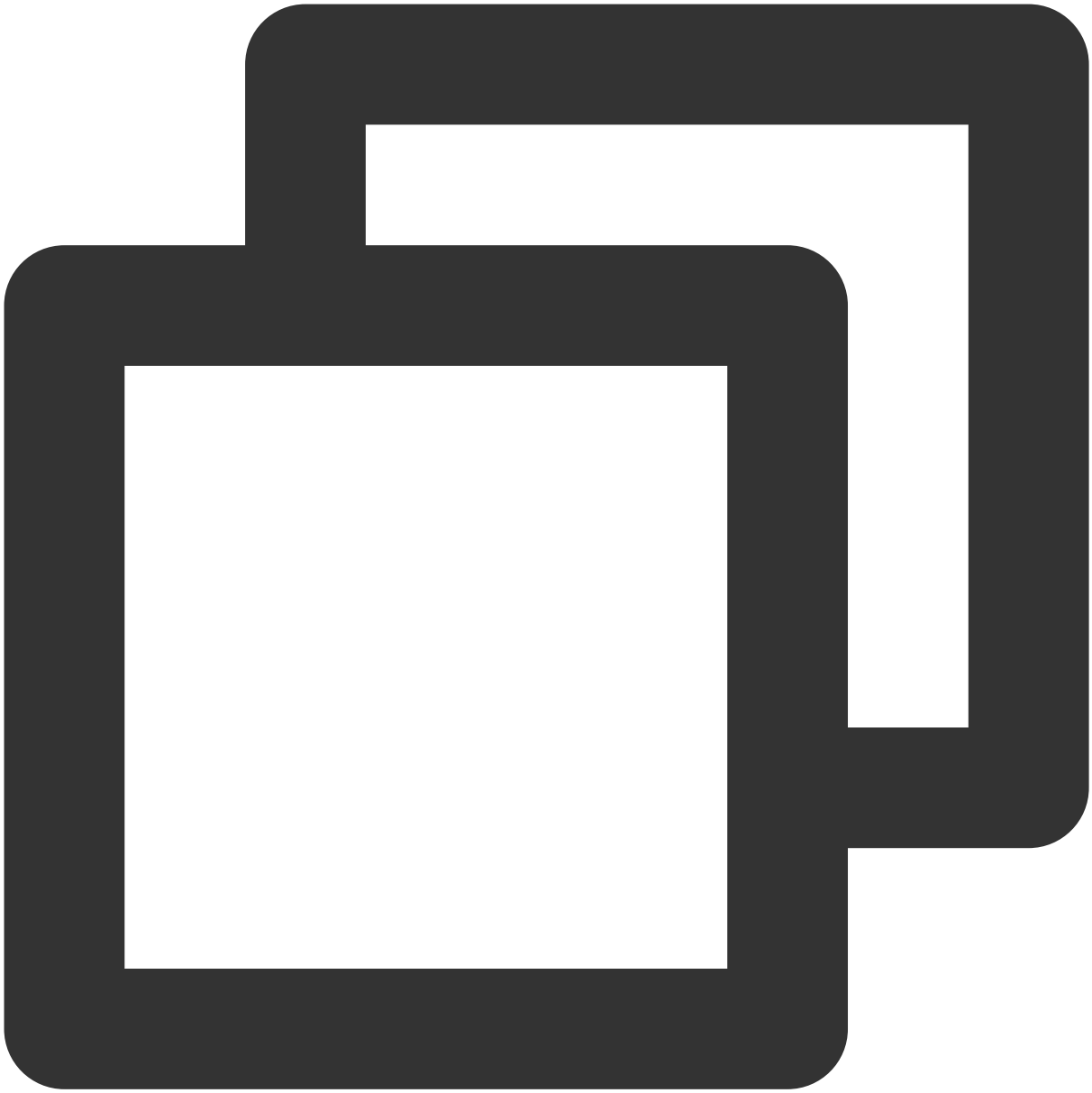
实际案例

请求



```
GET /ai_bucket?regions=ap-chongqing&pageSize=5&bucketName=t HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-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: 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>
```

智能语音
智能语音接口列表

最近更新时间：2024-01-20 10:09:09

任务接口

任务是一个抽象的概念，智能语音包含多种类型的任务，如语音合成、音频降噪等。
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。

工作流接口

智能语音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。

功能		任务接口
智能语音	音频降噪	提交音频降噪任务
	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

版权所有：腾讯云计算（北京）有限责任公司

智能语音服务配置接口
查询语音识别开通状态

最近更新时间：2024-01-20 10:09:09

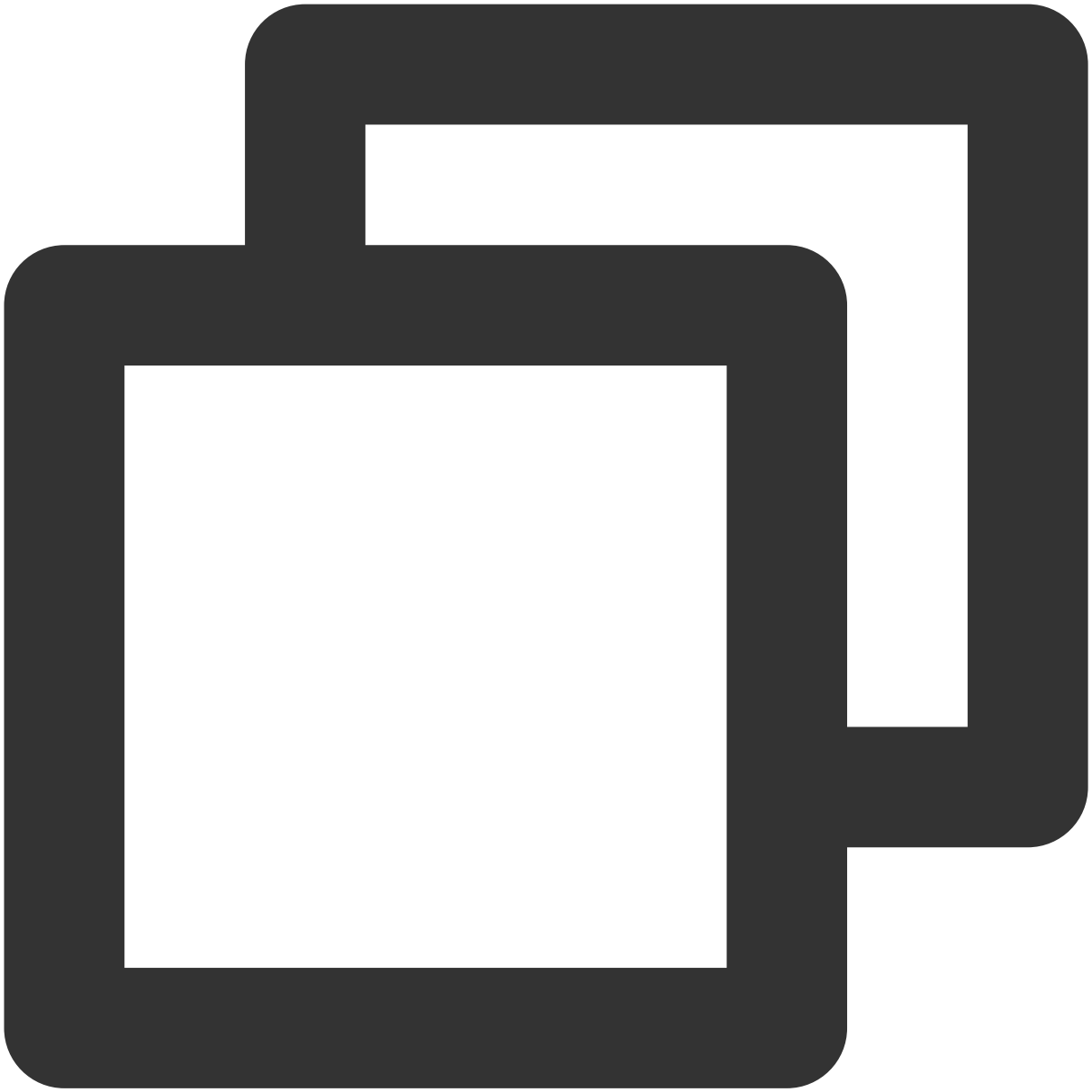
功能描述

DescribeSpeechBuckets 接口用于查询存储桶是否已开通语音识别功能。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
GET /asrbucket 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	类型	描述	是否必选
regions	string	地域信息，以“,”分隔字符串，支持 All、ap-shanghai、ap-beijing	否
bucketNames	string	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	否
bucketName	string	存储桶名称前缀，前缀搜索	否
pageNumber	string	第几页	否
pageSize	string	每页个数	否

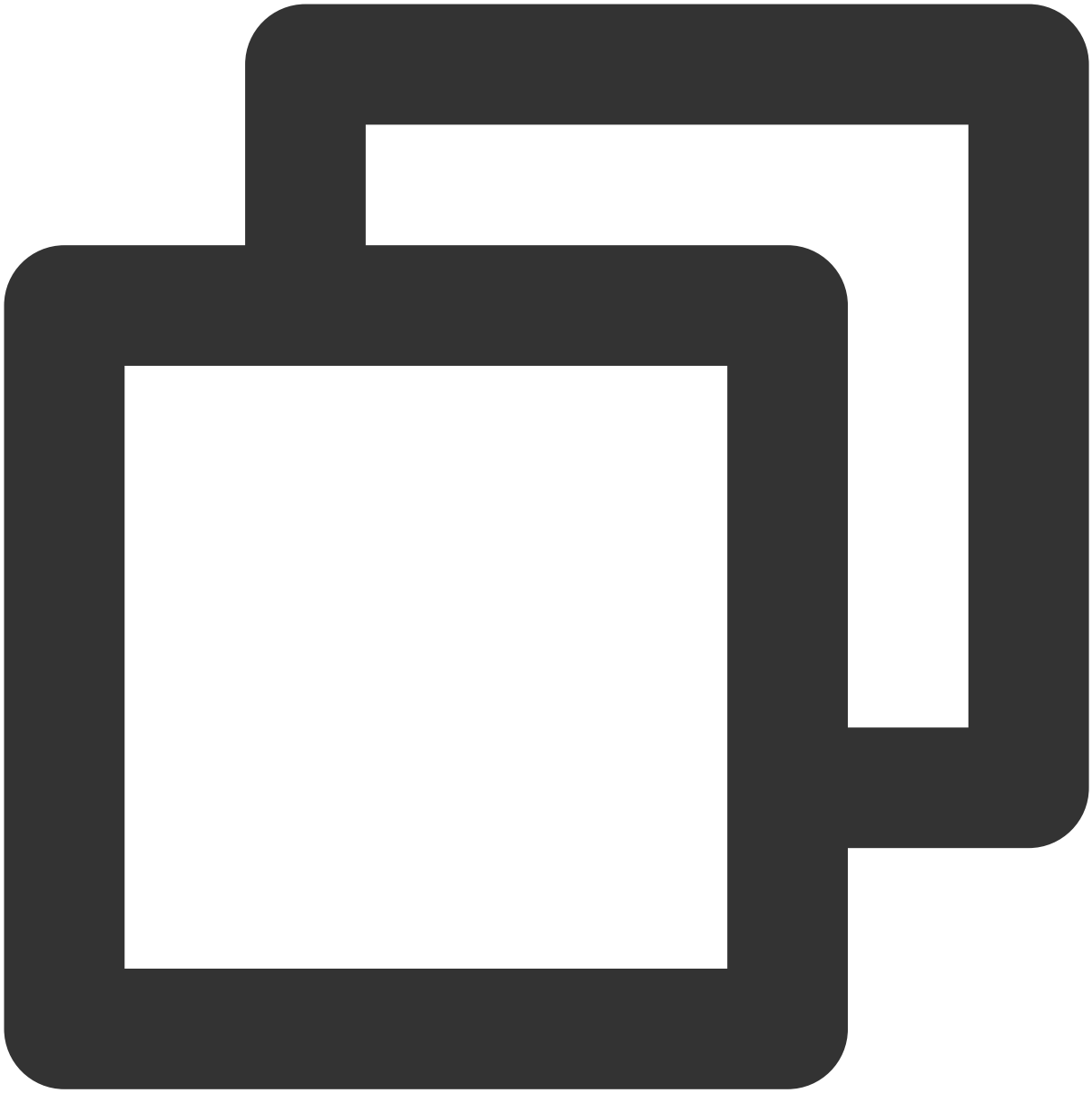
响应

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>
    <AsrBucketList>
        <BucketId></BucketId>
        <Region></Region>
        <CreateTime></CreateTime>
    </AsrBucketList>
</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
AsrBucketList	Response	语音 Bucket 列表	Container

Container 节点 AsrBucketList 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.AsrBucketList	存储桶 ID	String
Name	Response.AsrBucketList	存储桶名称，同 BucketId	String
Region	Response.AsrBucketList	所在的地域	String
CreateTime	Response.AsrBucketList	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。

媒体处理
媒体处理接口列表

最近更新时间：2024-01-20 10:09:09

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	画质增强（包含超分辨率、插帧、色彩增强等功能）	提交画质增强任务
	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
同步任务	视频截图	获取媒体截图任务
	获取媒体信息	获取媒体信息任务
	私有m3u8	读取m3u8任务

模板接口

针对媒体处理功能（例如音视频转码、音视频拼接、视频截帧和视频转动图等），您需要在数据万象创建**任务**或**工作流**时，指定使用的模板，模板页为您提供了系统预设模板，您也可根据自身业务需求自定义模板，媒体处理模板接口请参见 媒体处理模板。

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，工作流调用请参见 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，调用请参见 批量数据处理接口。

媒体处理服务配置接口
查询媒体处理服务

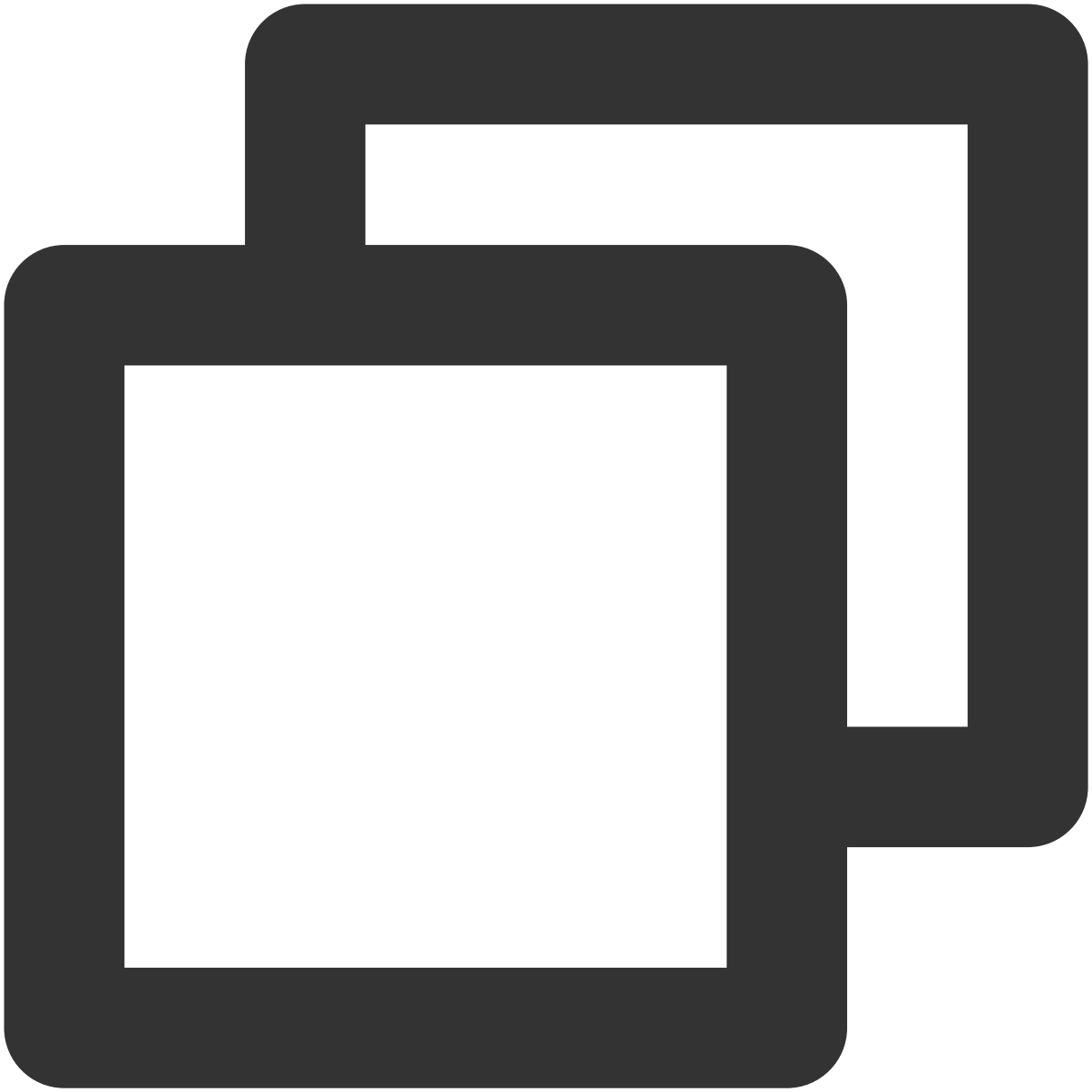
最近更新时间：2024-01-20 10:09:09

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）。

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，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

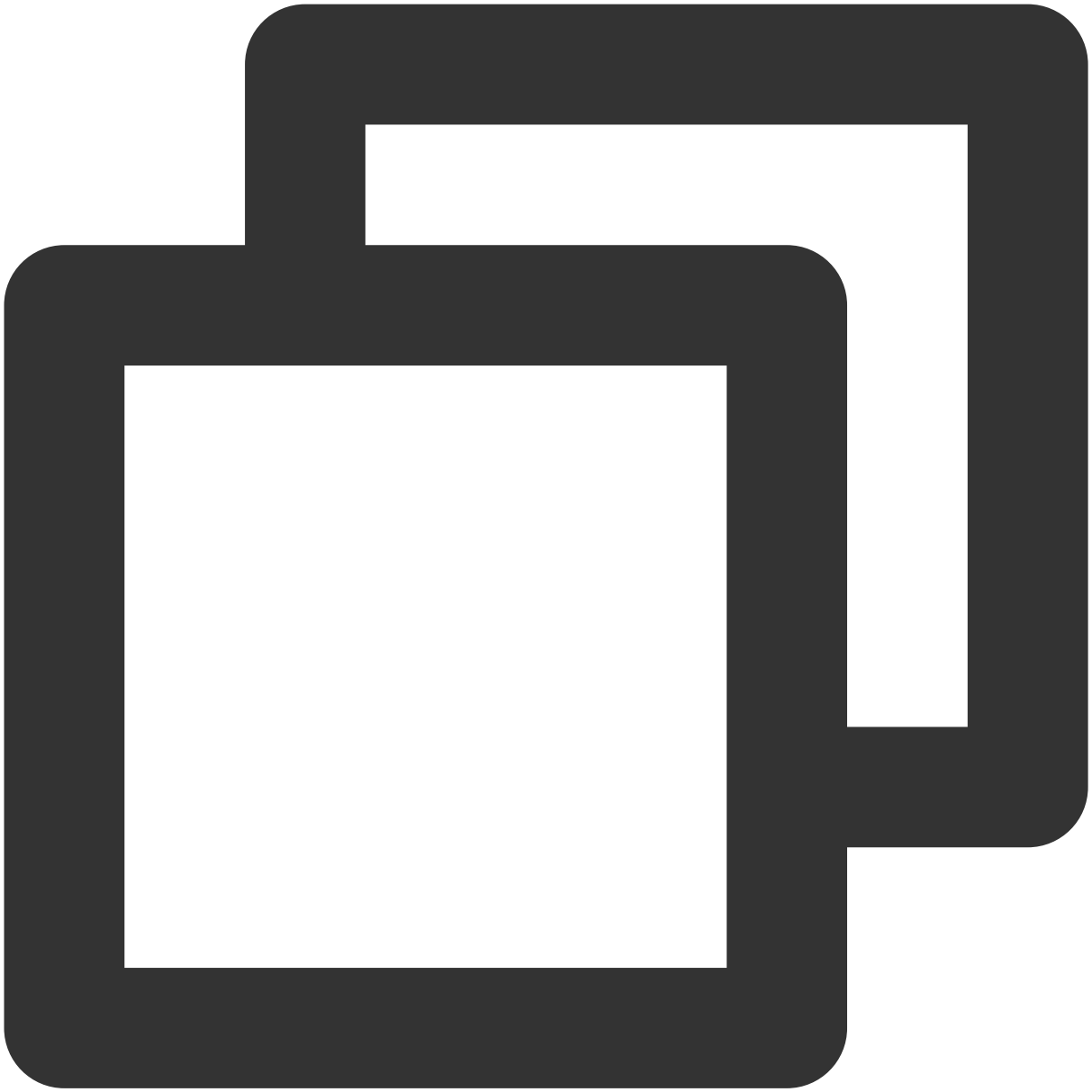
响应

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：

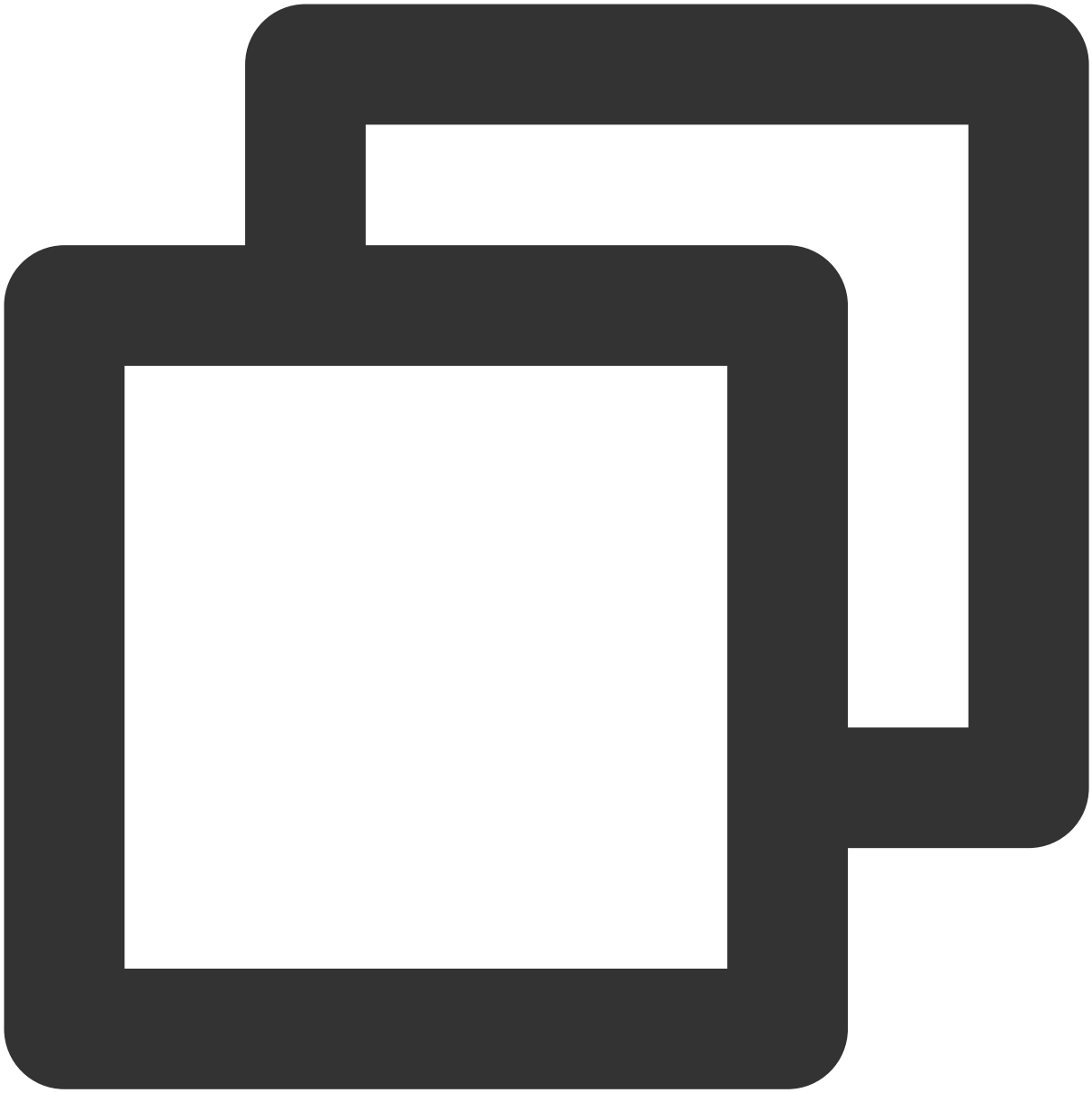
节点名称（关键字）	父节点	描述	类型
BucketId	Response.MediaBucketList	存储桶 ID	String
Name	Response.MediaBucketList	存储桶名称，同 BucketId	String
Region	Response.MediaBucketList	所在的地域	String
CreateTime	Response.MediaBucketList	创建时间	String

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。

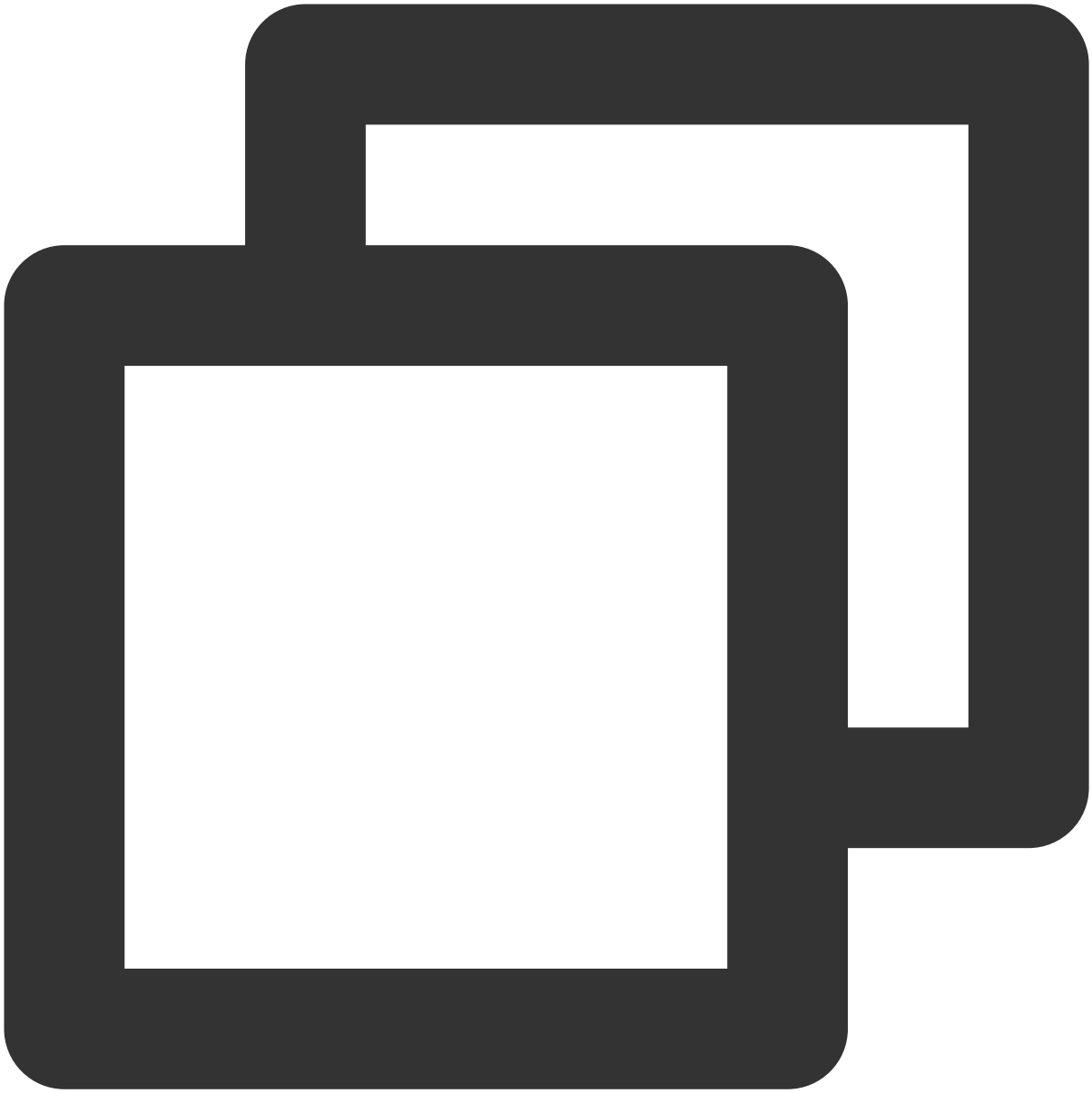
实际案例

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>
```

绑定媒体处理服务

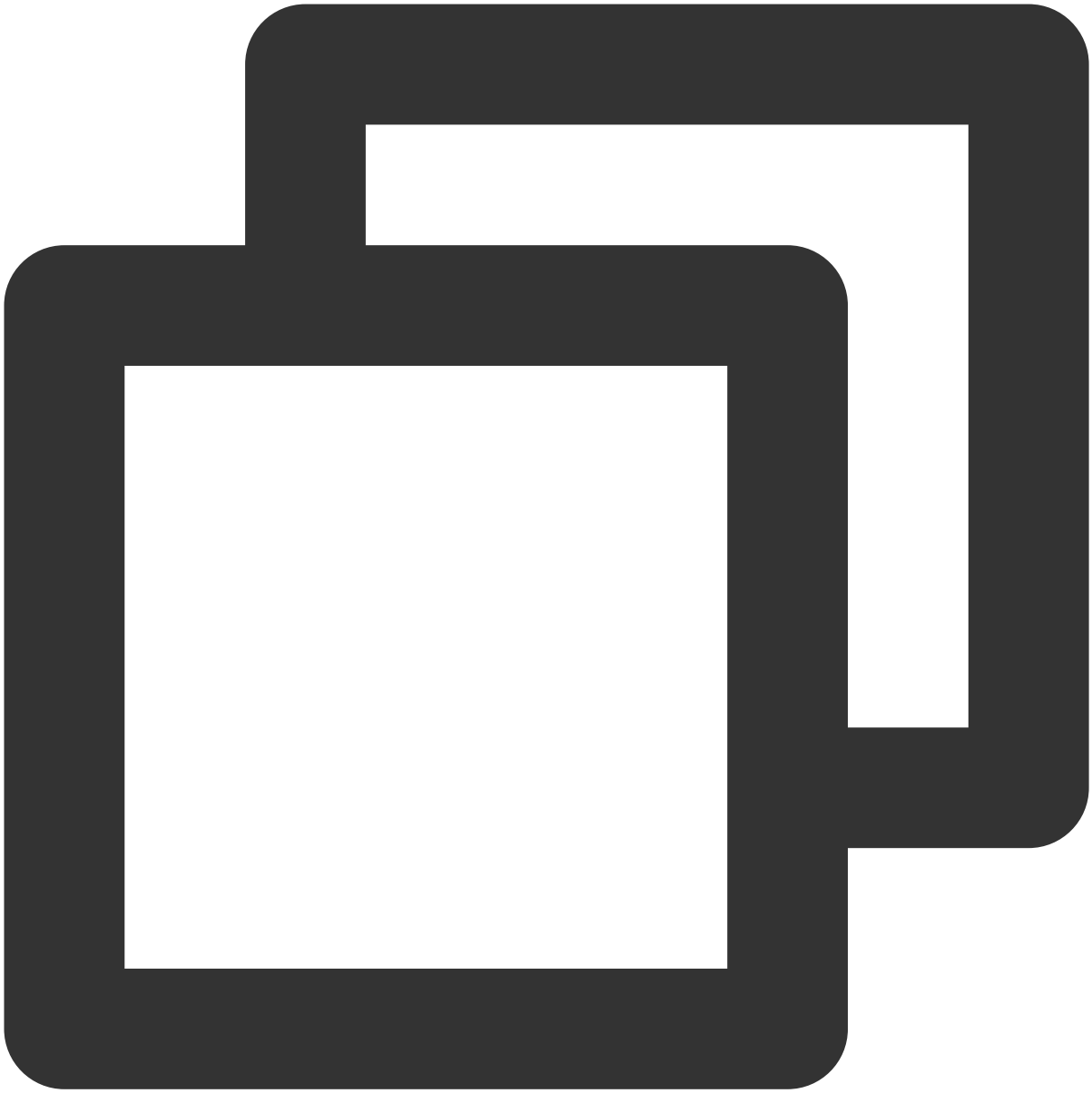
最近更新时间：2024-01-20 10:09:09

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。

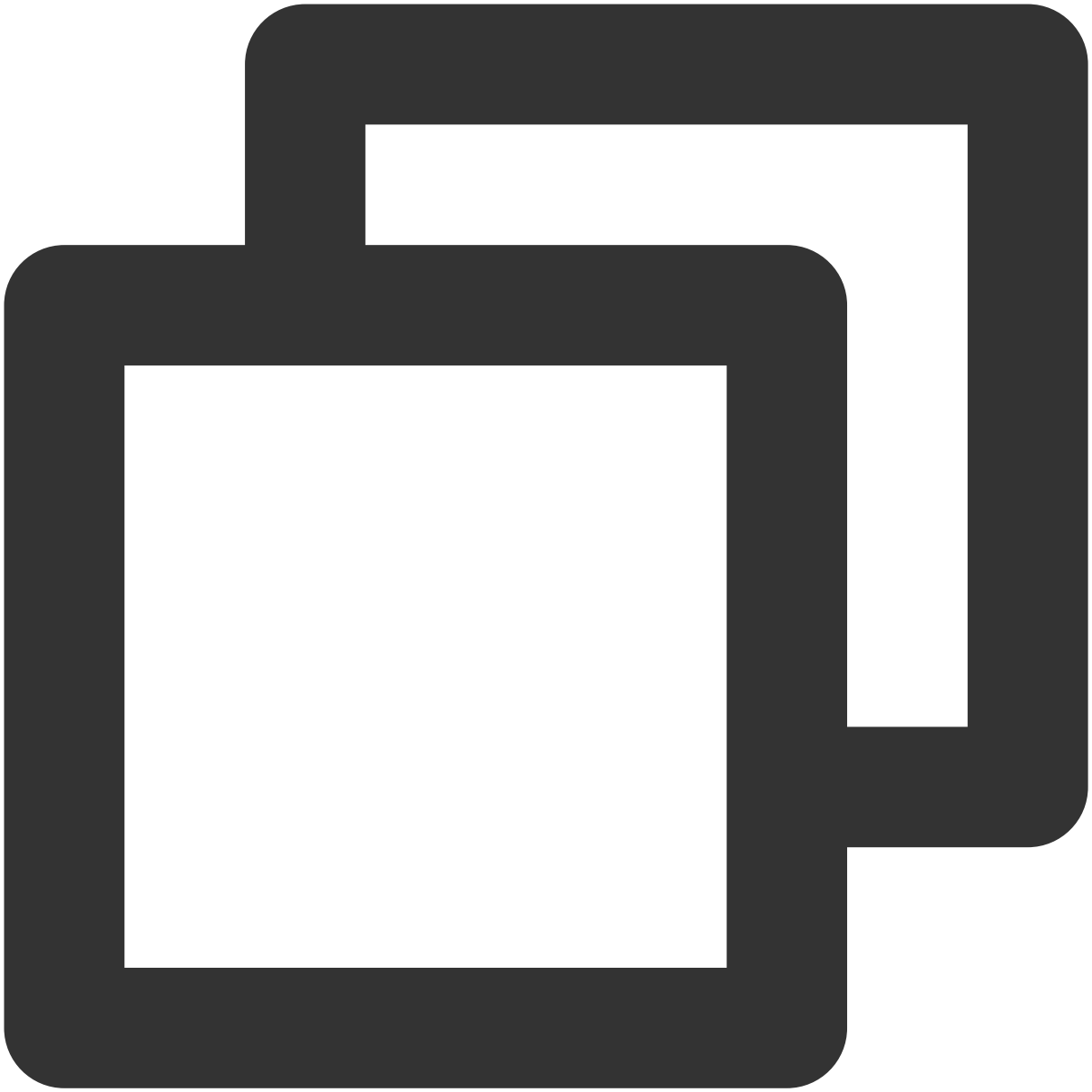
响应

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：

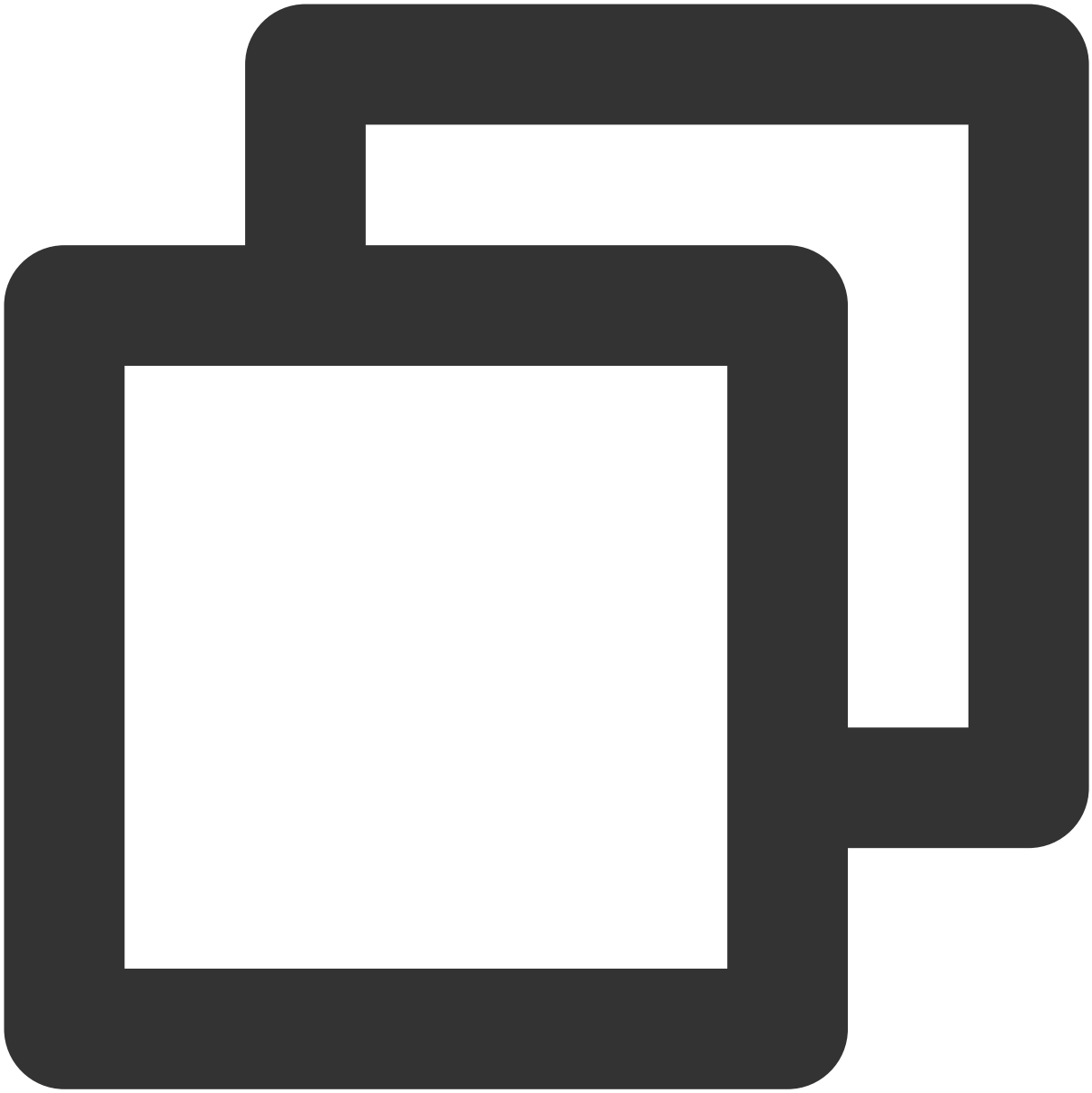
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
MediaBucket	Response	媒体 Bucket 详情，同 DescribeMediaBuckets 中的 Response.MediaBucketList	Container

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。

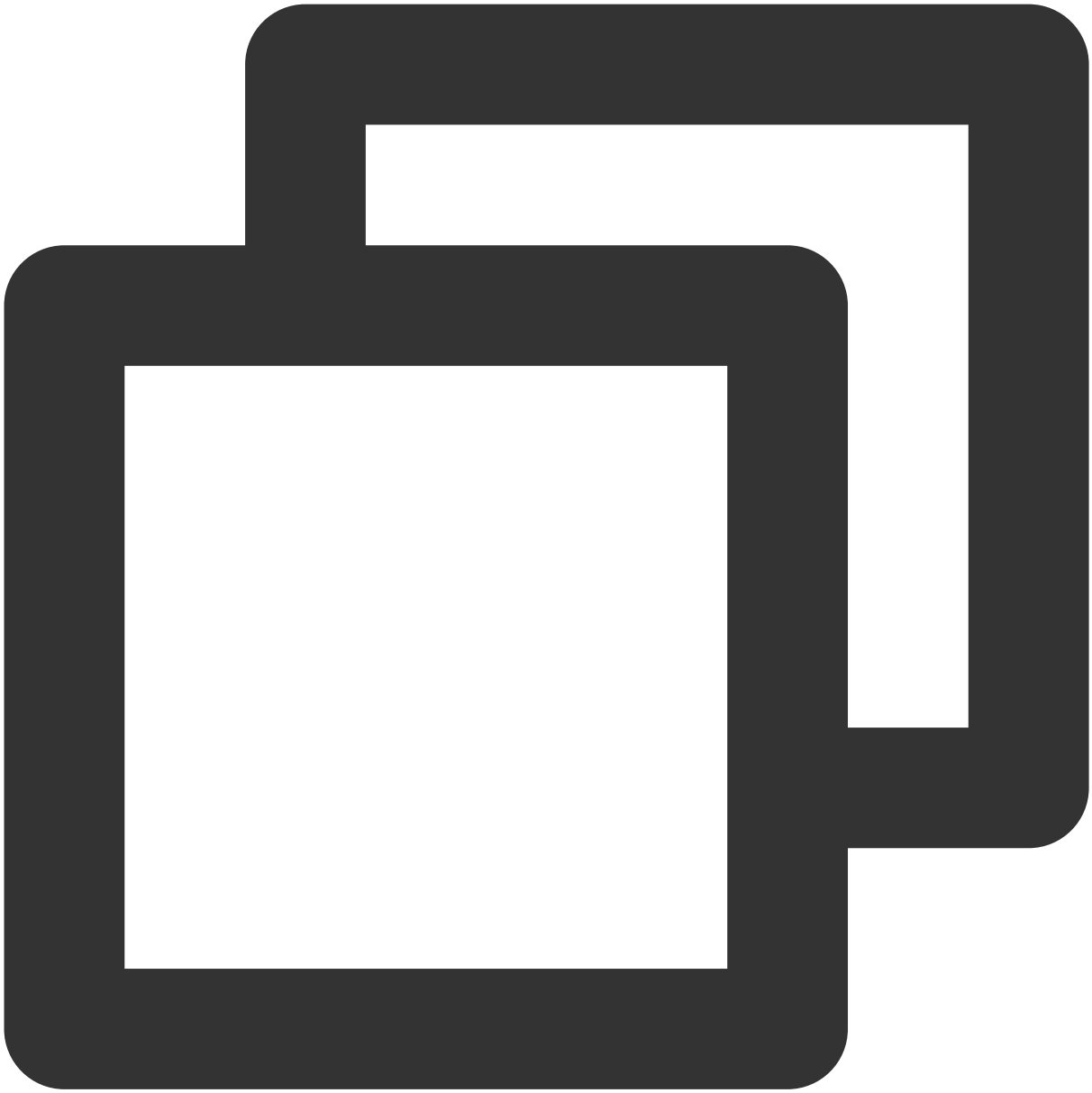
实际案例

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>
```

解绑媒体处理服务

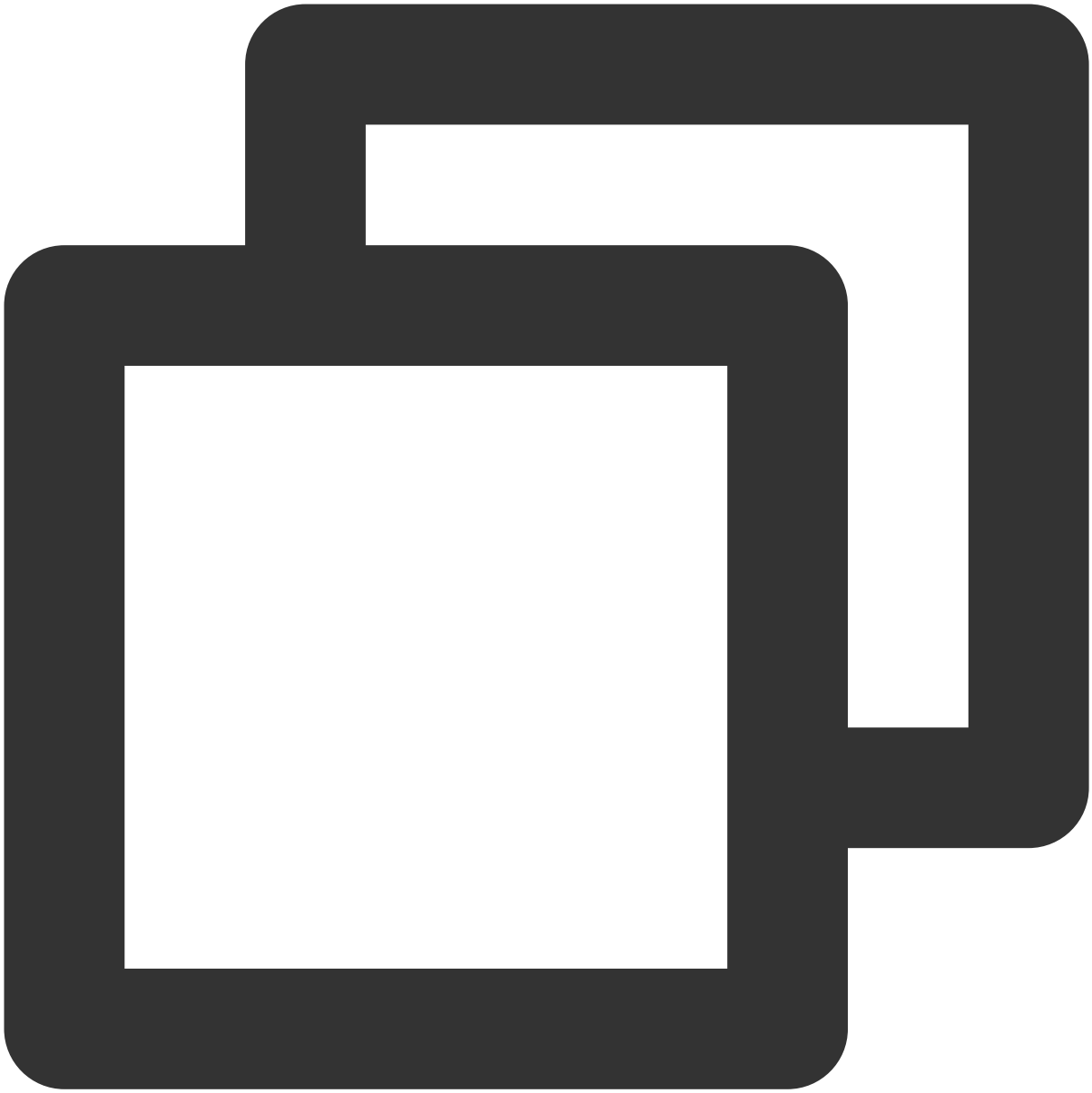
最近更新时间：2024-01-20 10:09:09

功能描述

DELETEMediaBucket 接口用于解绑媒体存储桶操作（不会删除存储桶内的对象）。

请求

请求示例



```
DELETE /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。

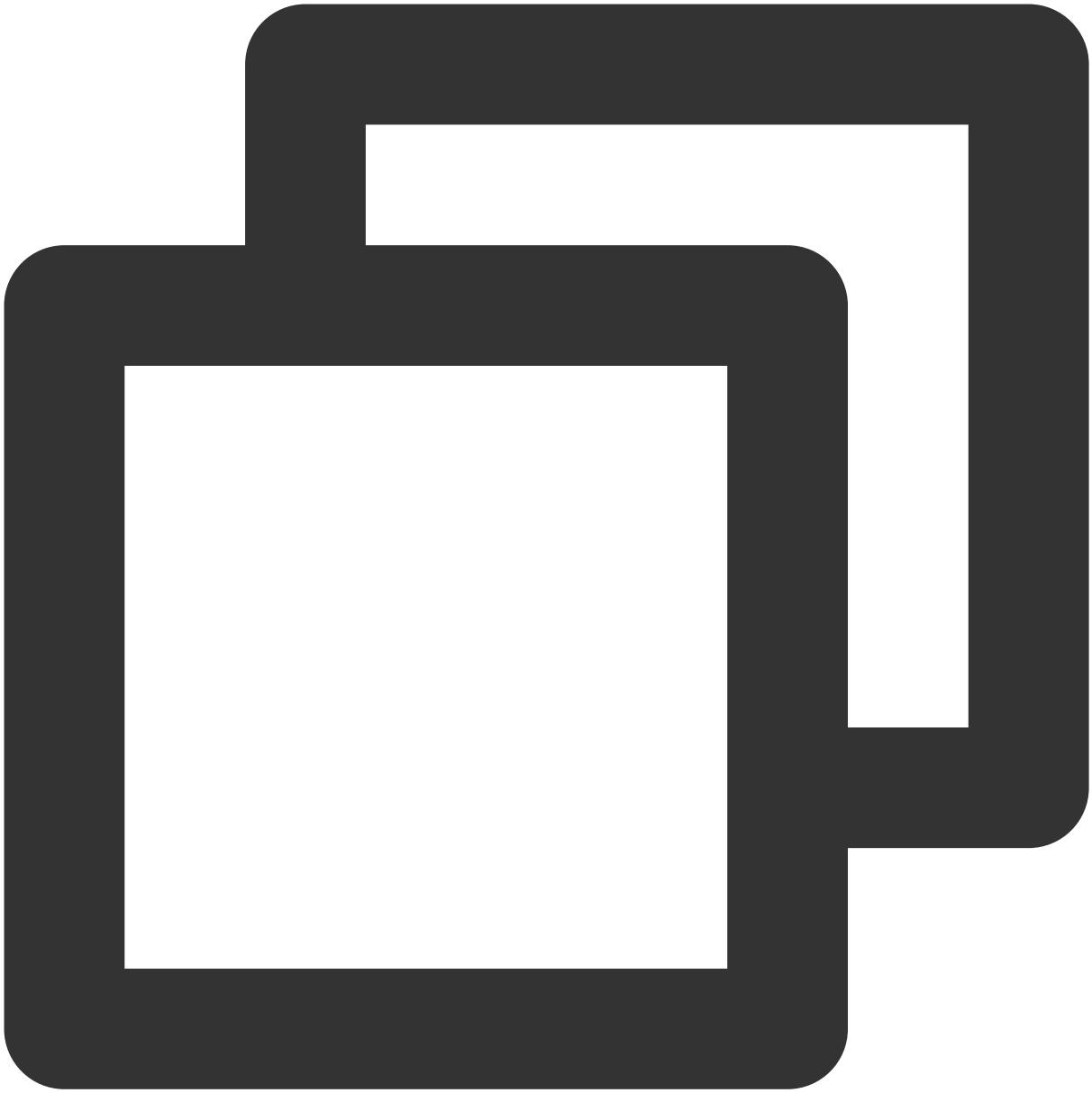
响应

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：



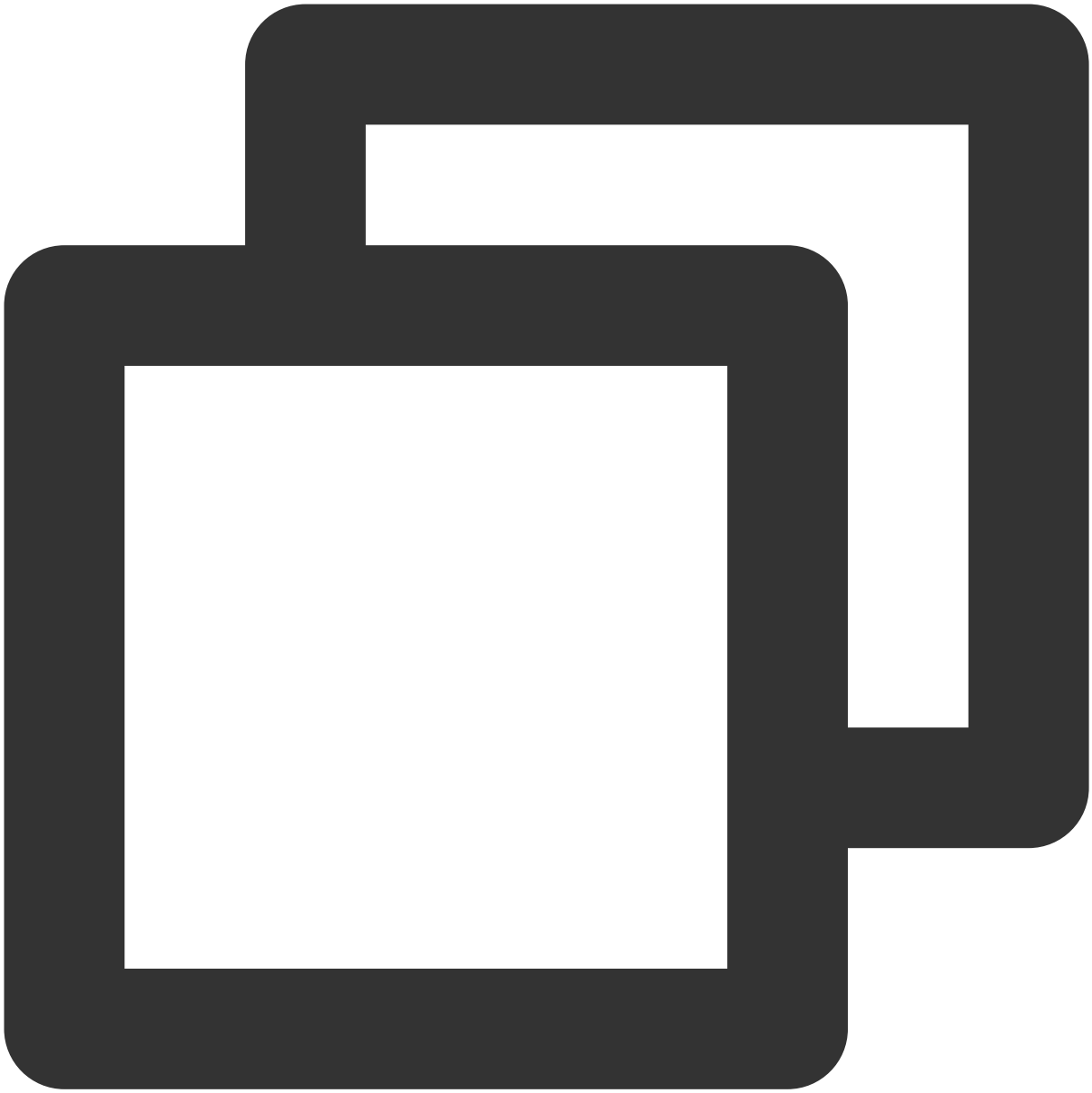
```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
</Response>
```

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。

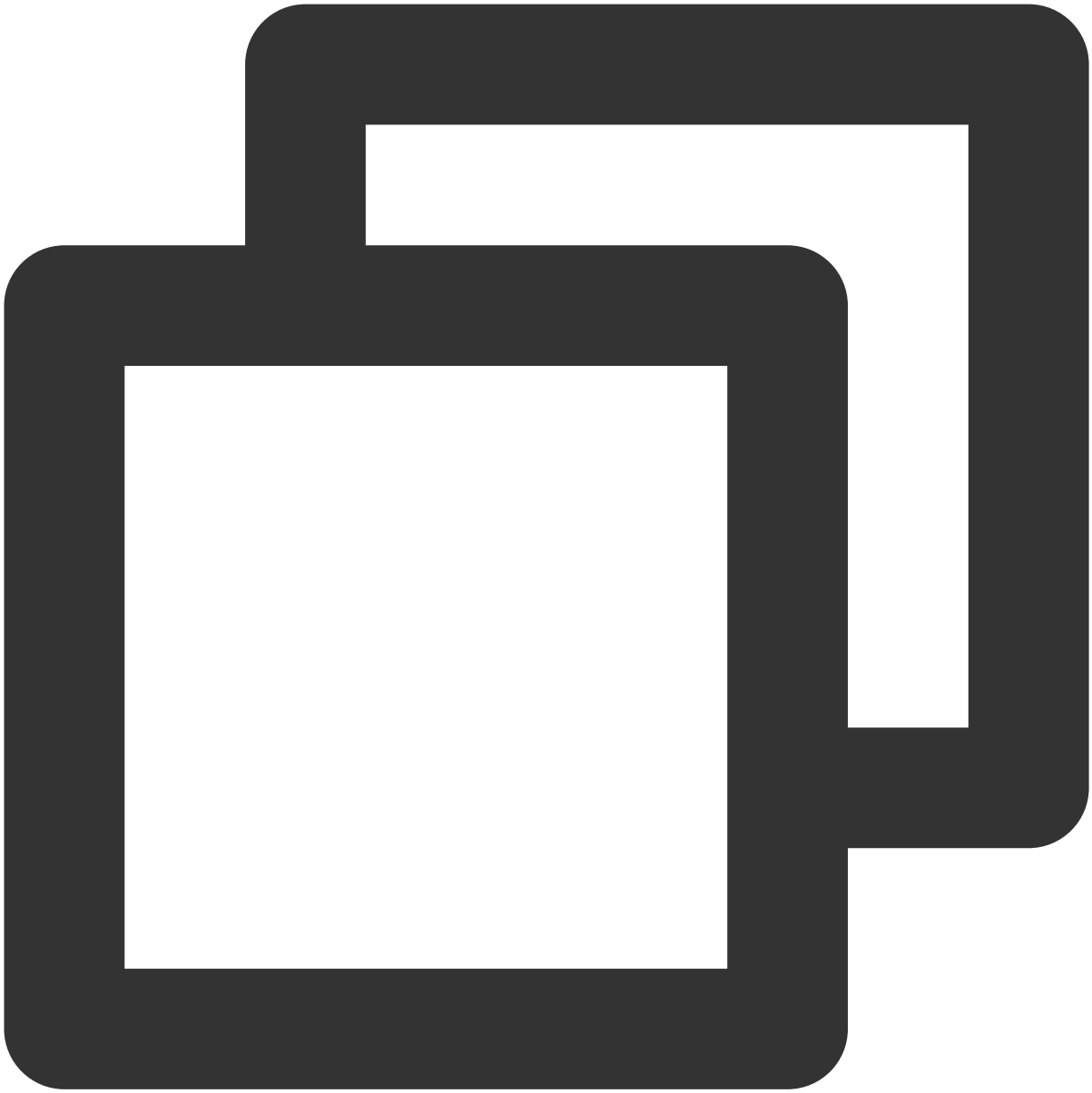
实际案例

请求



```
DELETE /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>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

媒体 bucket 接口
查询媒体处理开通状态

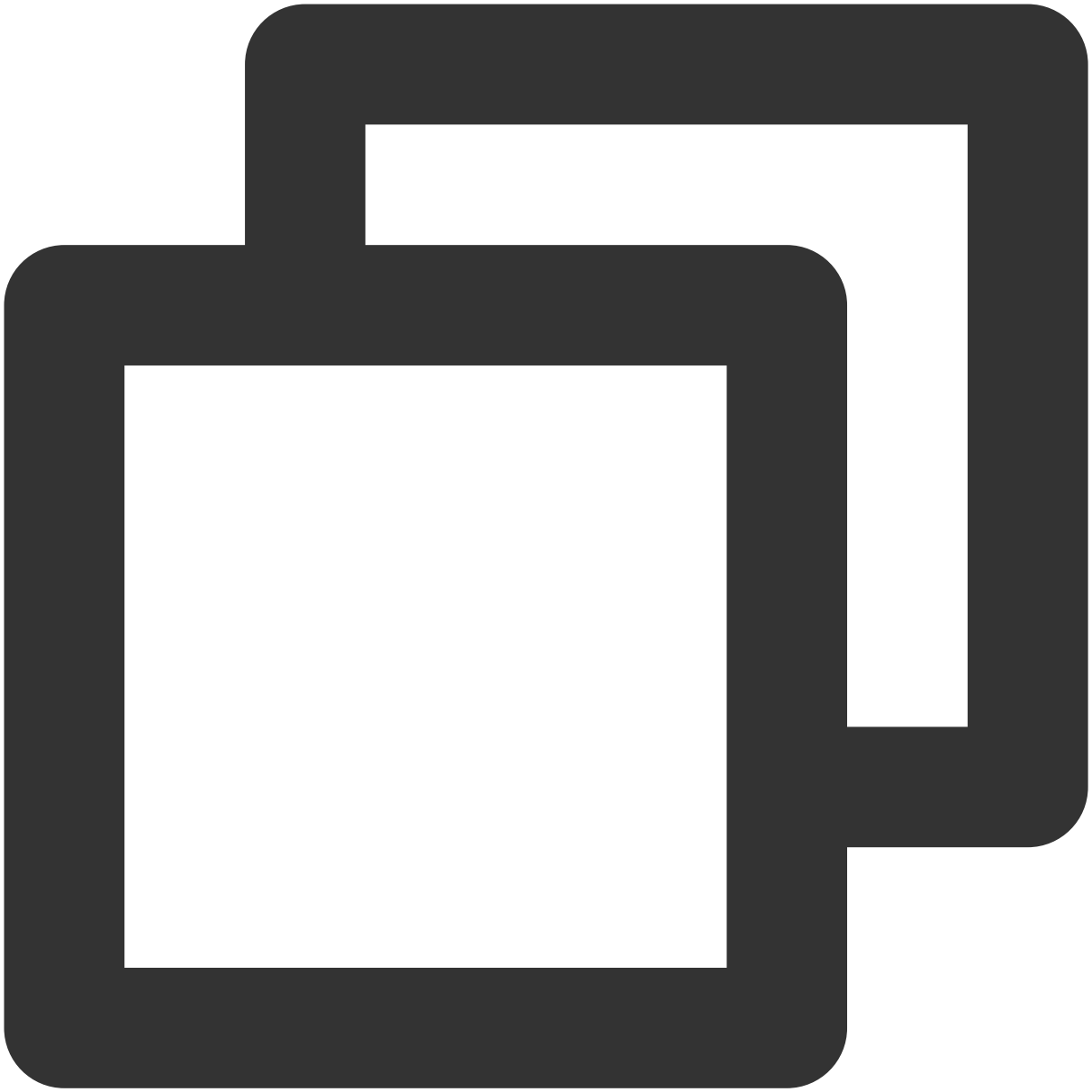
最近更新时间：2024-01-20 10:09:09

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）。

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，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

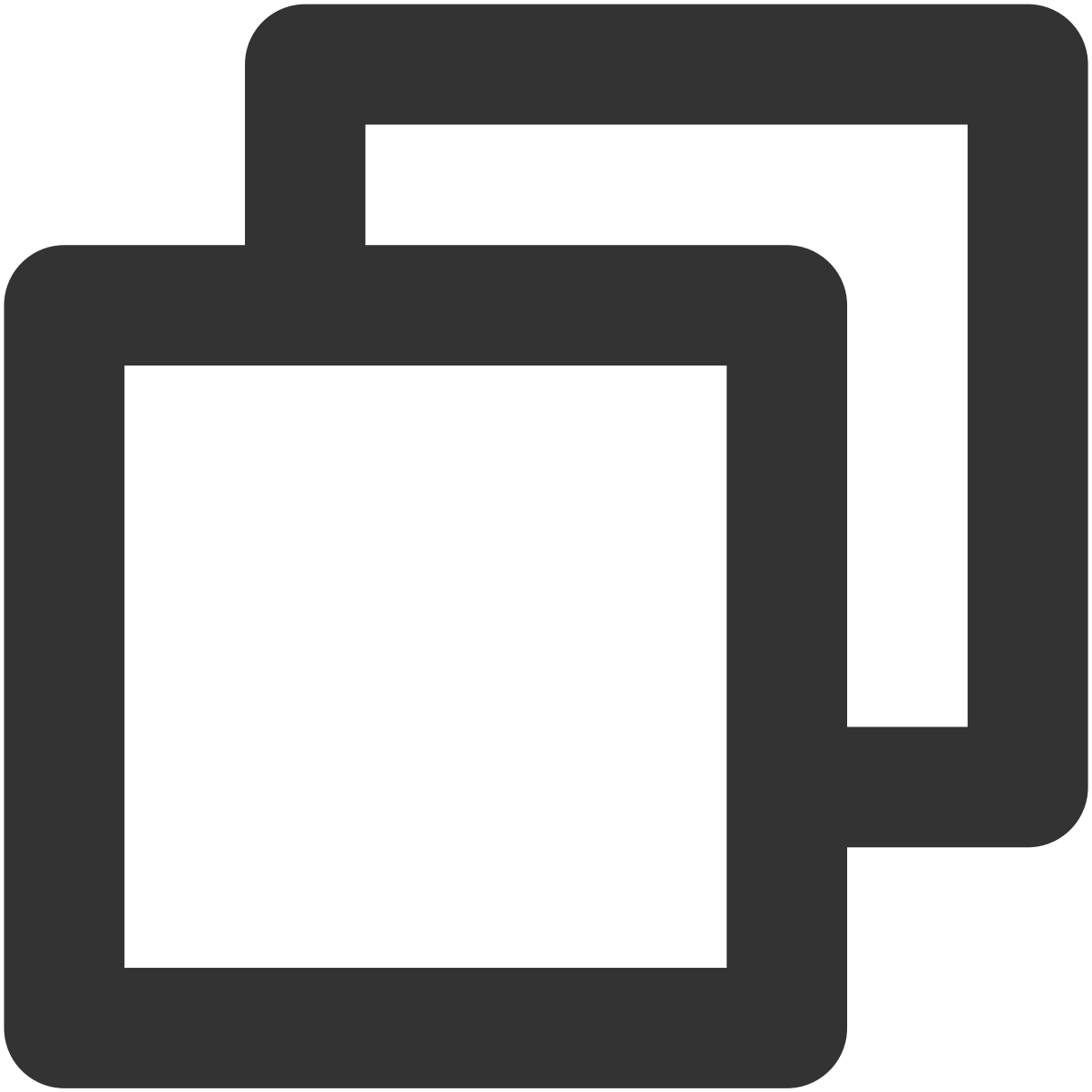
响应

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：

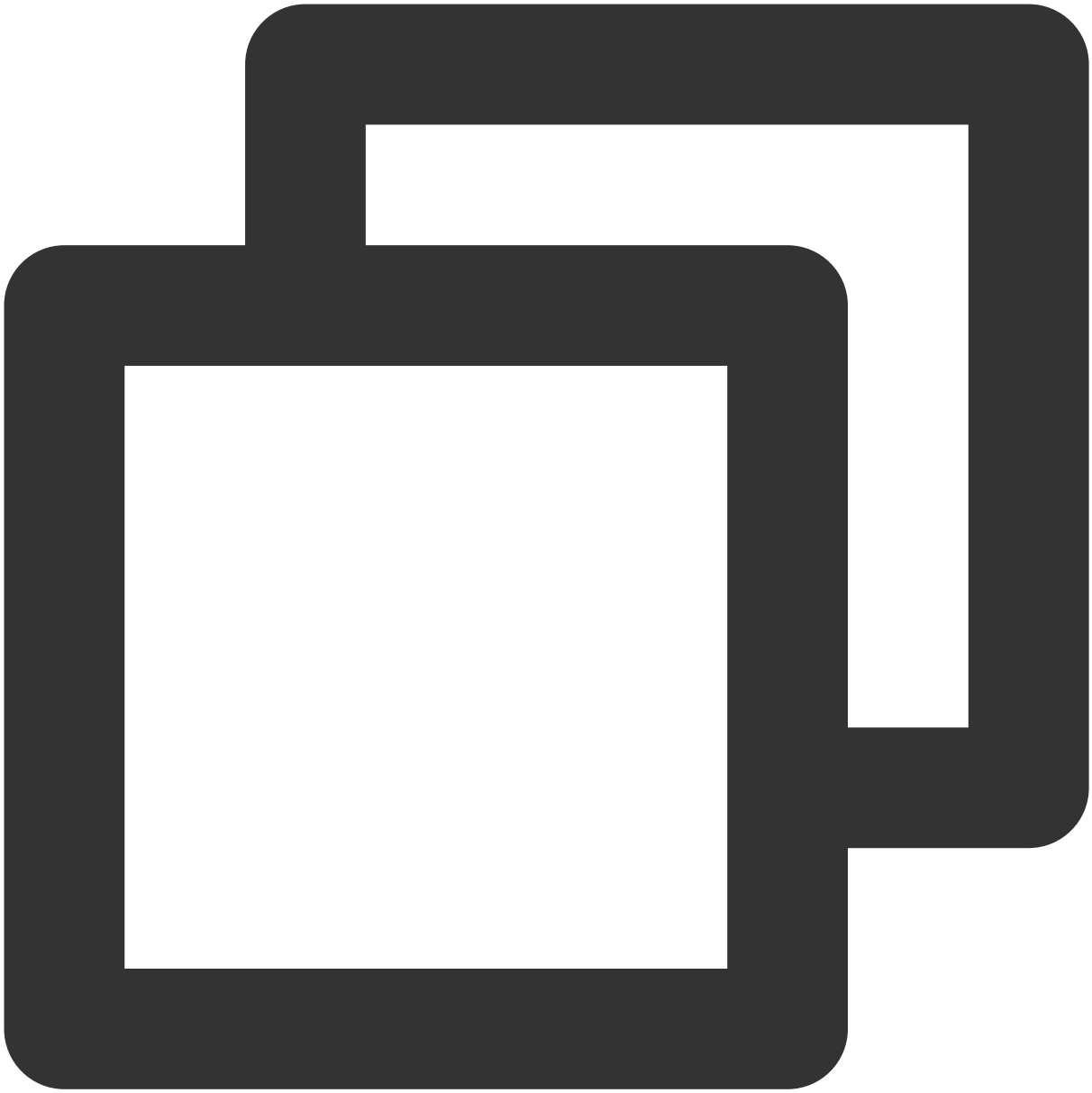
节点名称（关键字）	父节点	描述	类型
BucketId	Response.MediaBucketList	存储桶 ID	String
Name	Response.MediaBucketList	存储桶名称，同 BucketId	String
Region	Response.MediaBucketList	所在的地域	String
CreateTime	Response.MediaBucketList	创建时间	String

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。

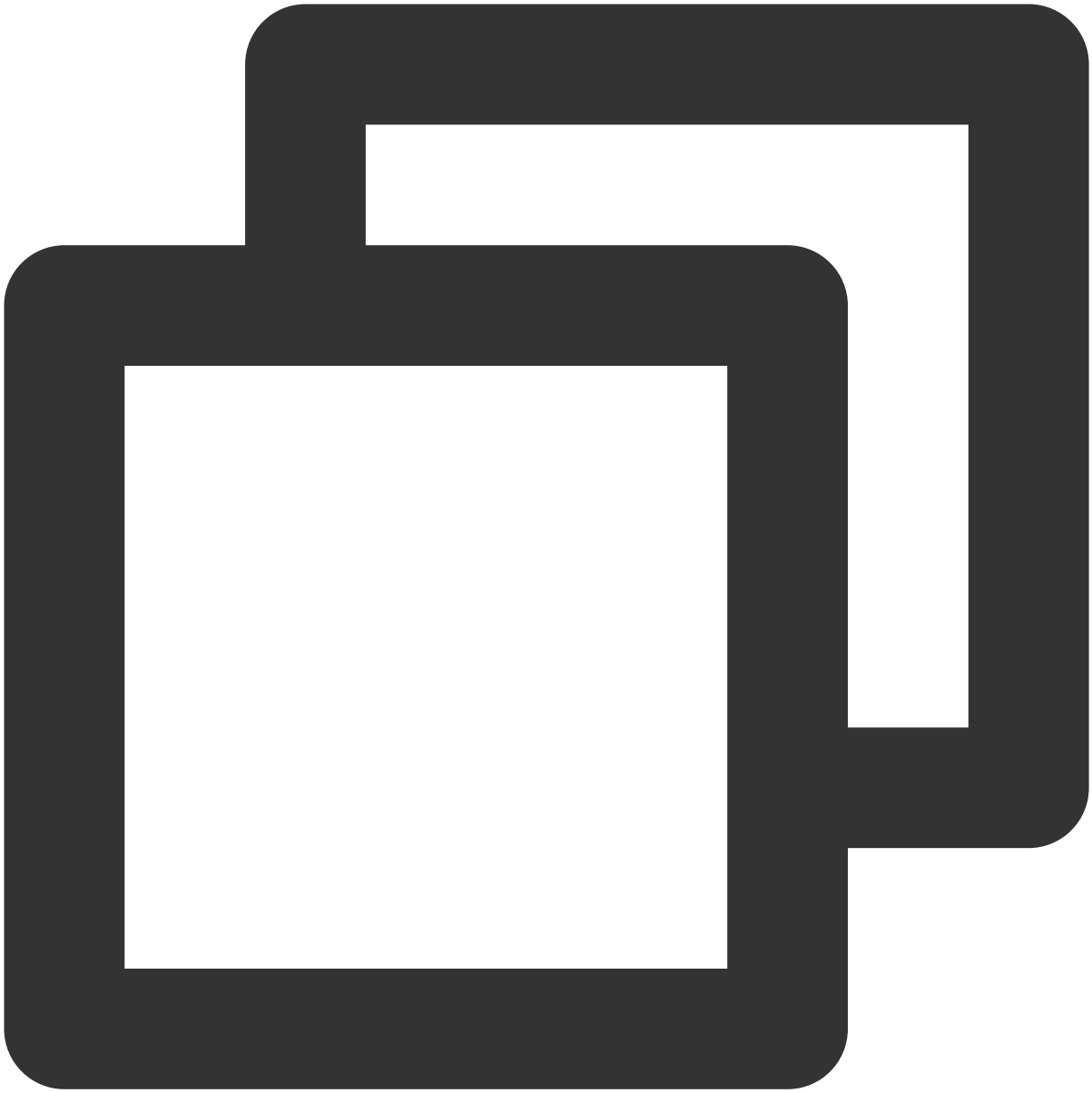
实际案例

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

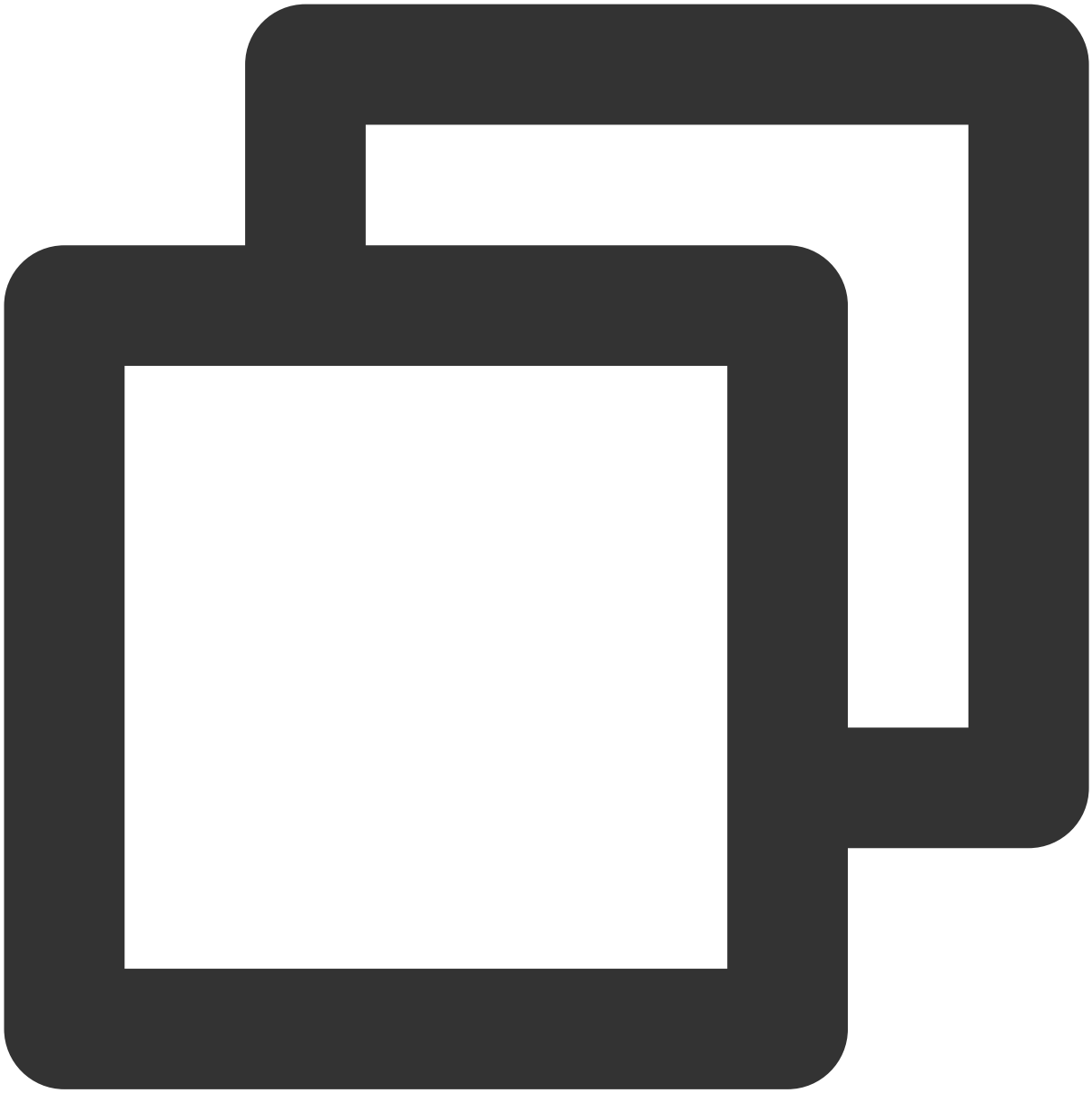
最近更新时间：2024-01-20 10:09:09

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。

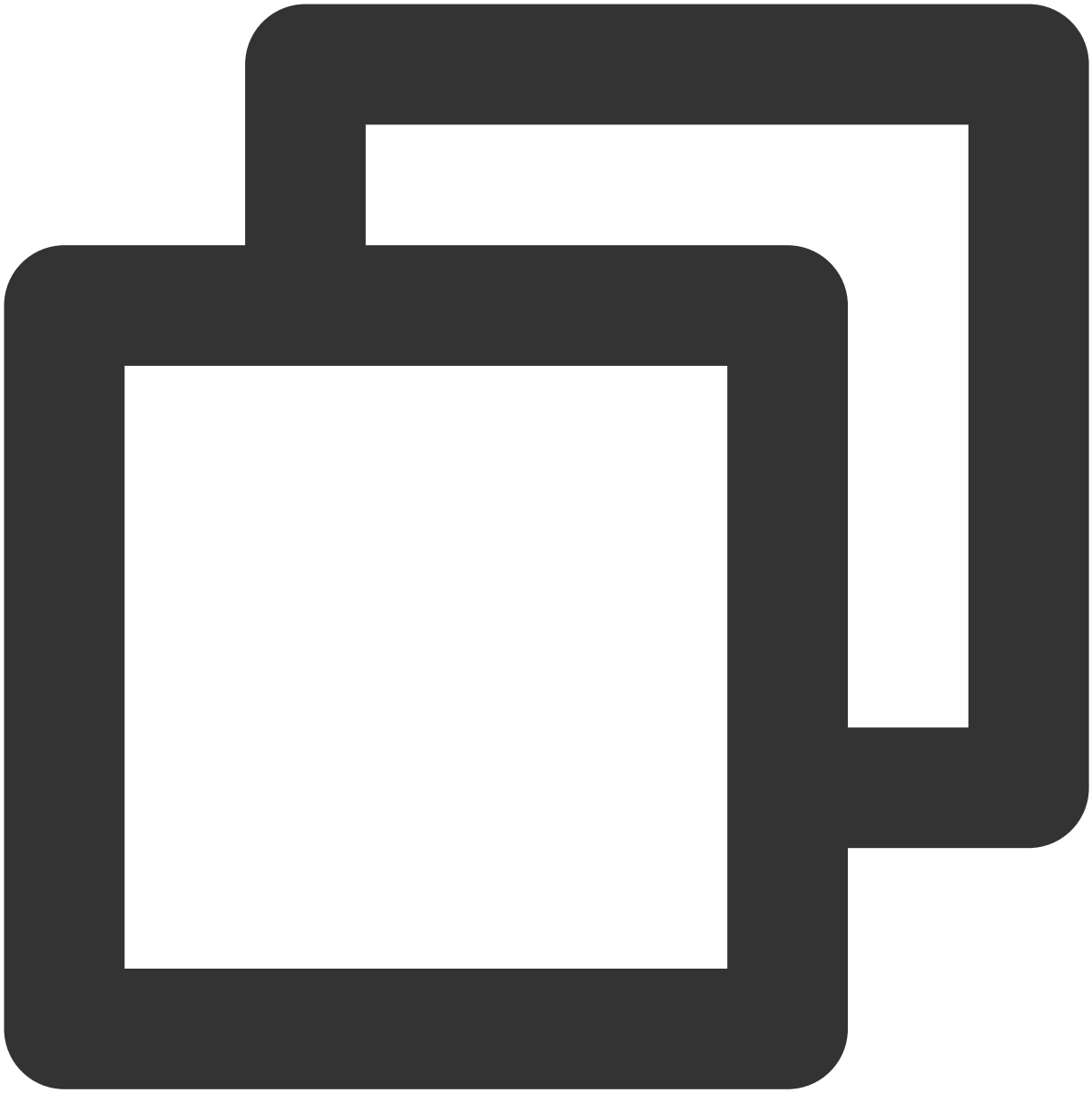
响应

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：

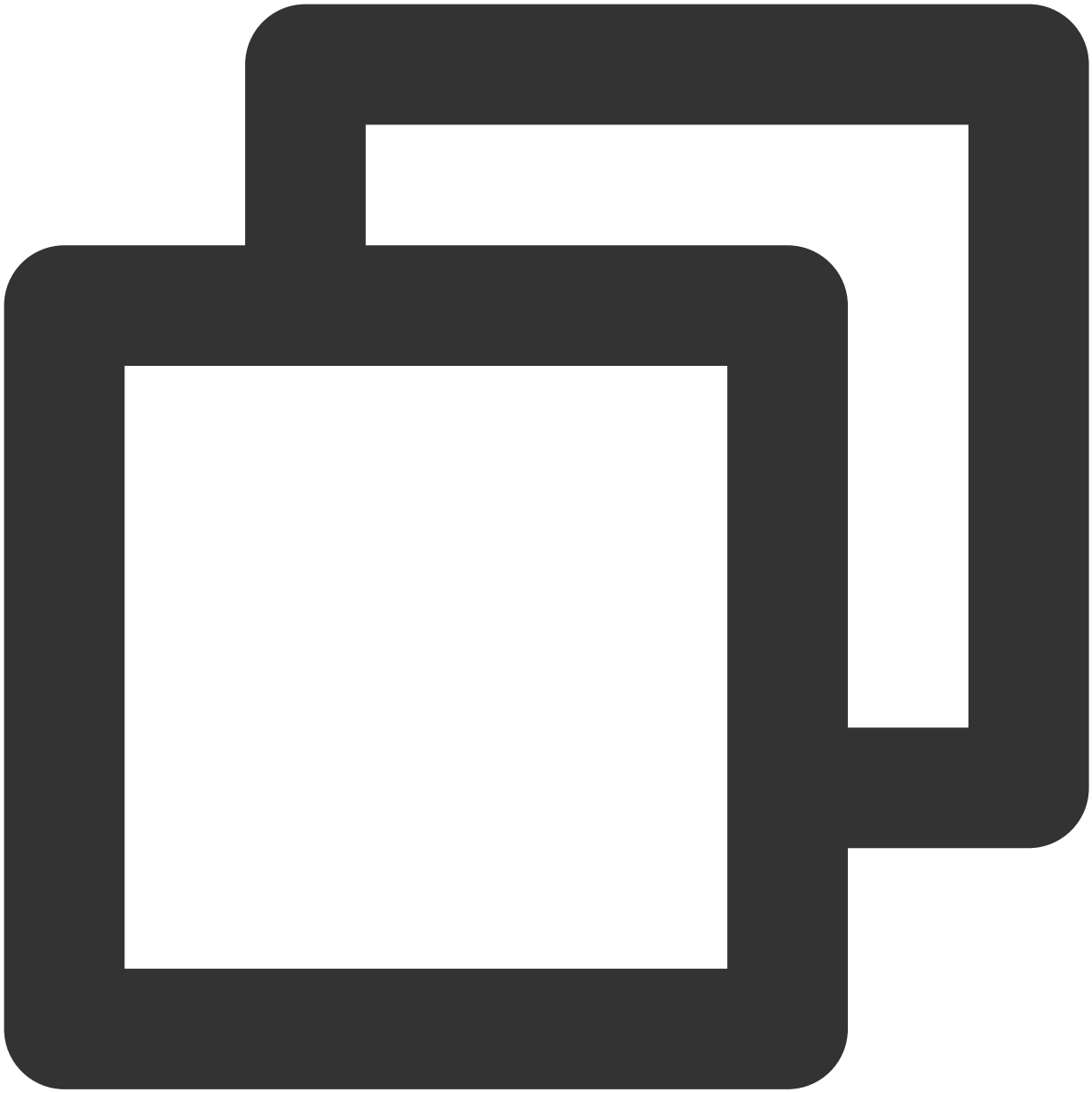
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
MediaBucket	Response	媒体 Bucket 详情，同 DescribeMediaBuckets 中的 Response.MediaBucketList	Container

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。

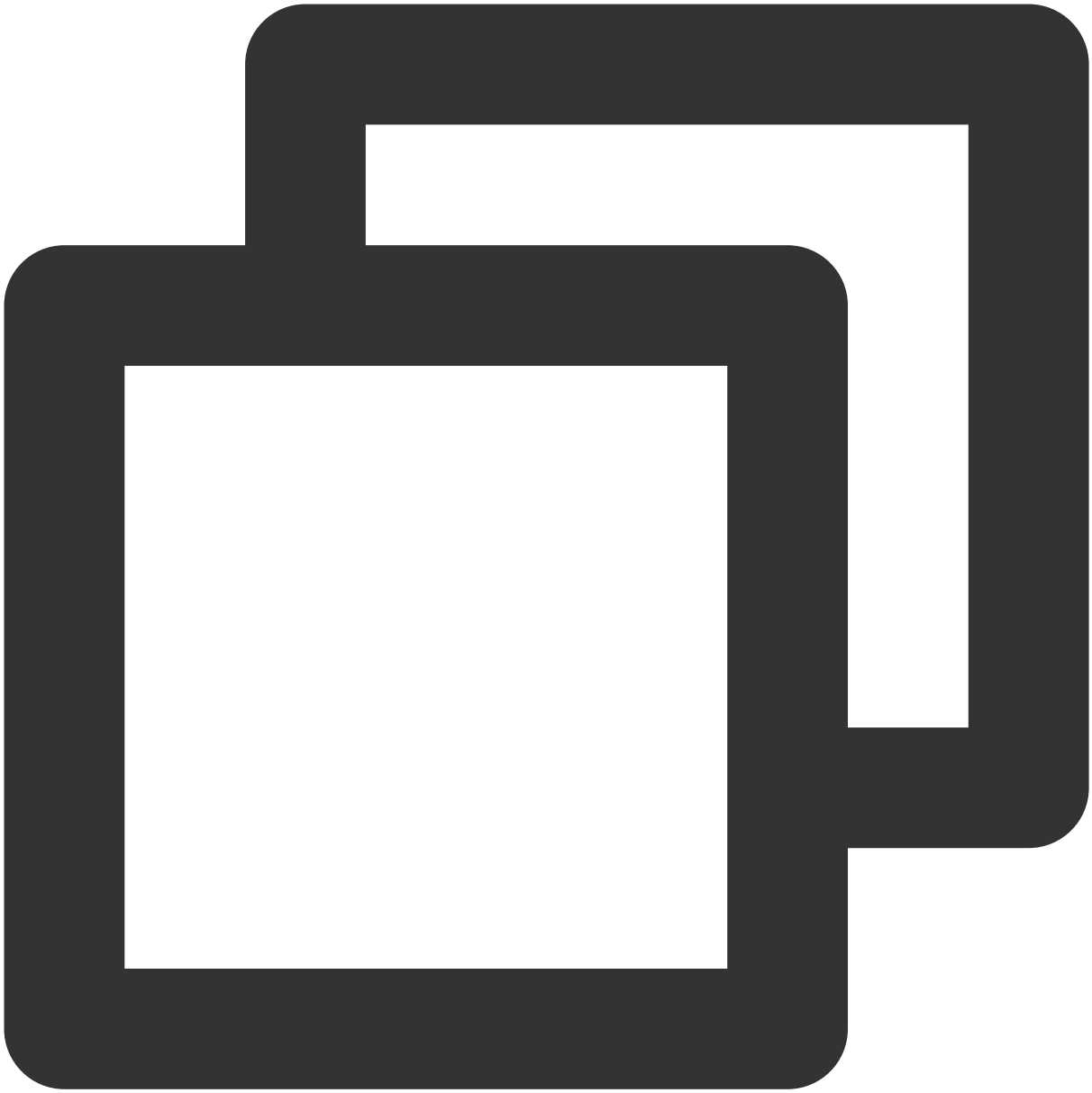
实际案例

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>
```

CreateMediaBucket

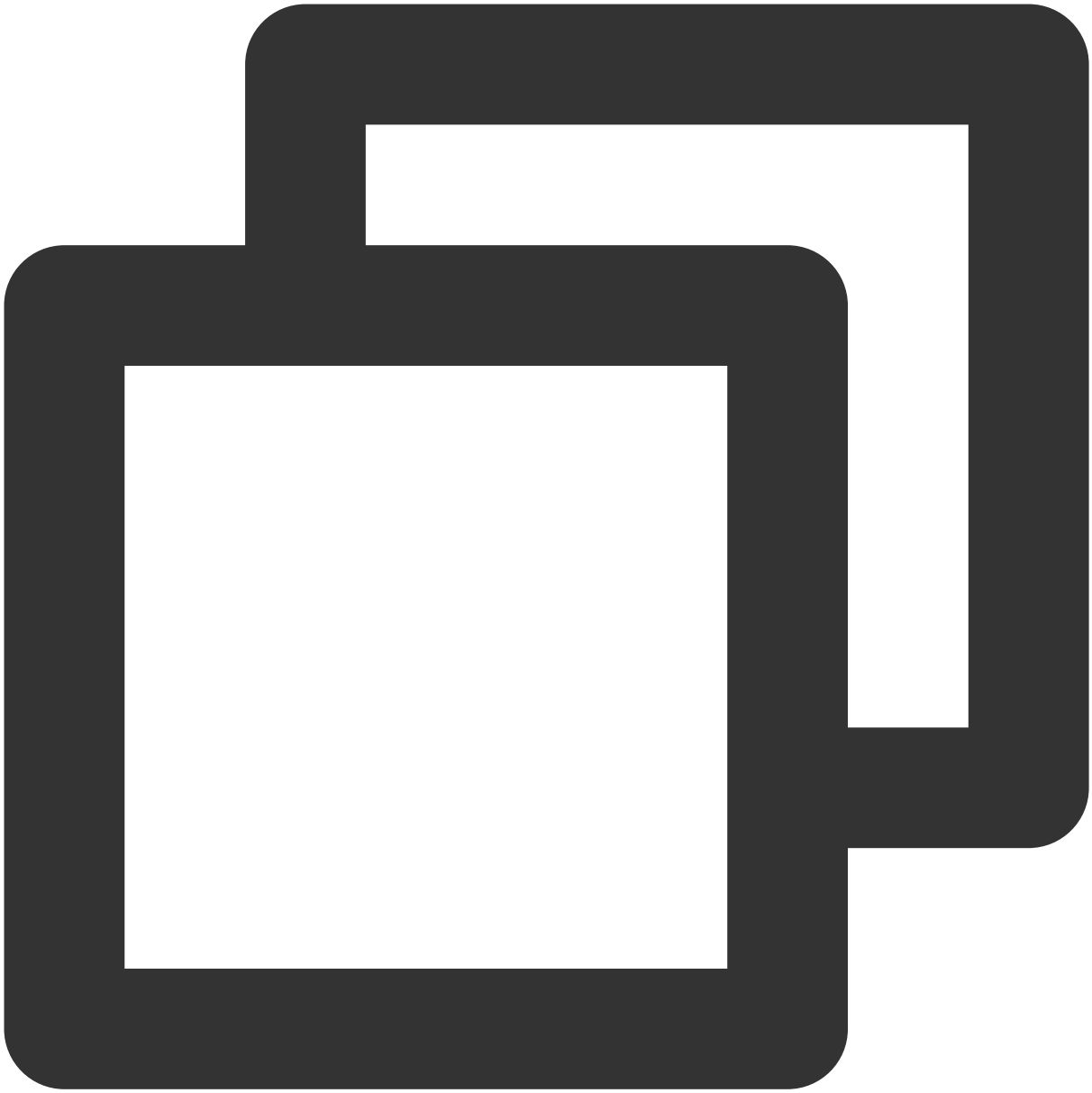
最近更新时间：2024-01-20 10:09:09

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。

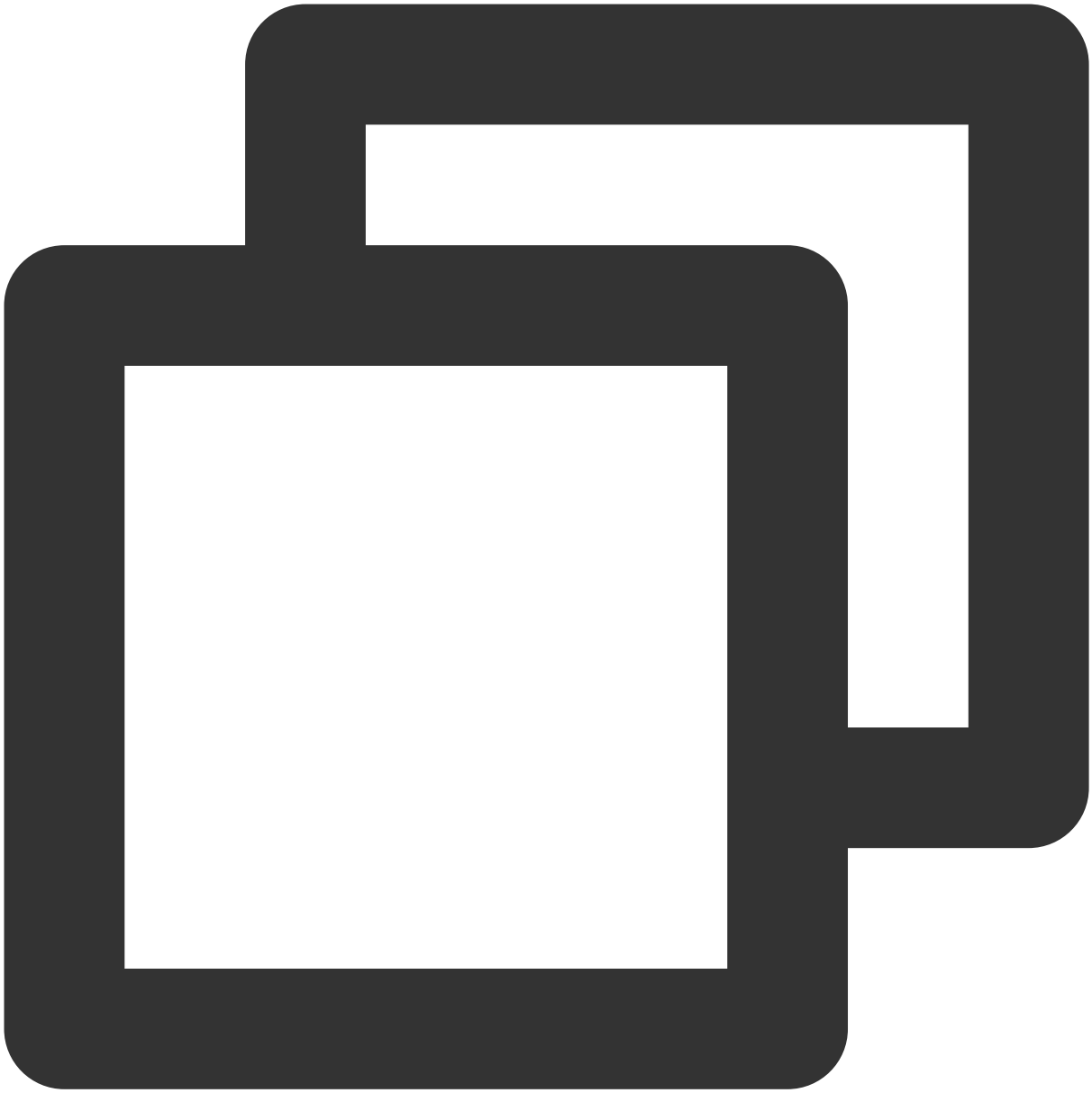
响应

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：

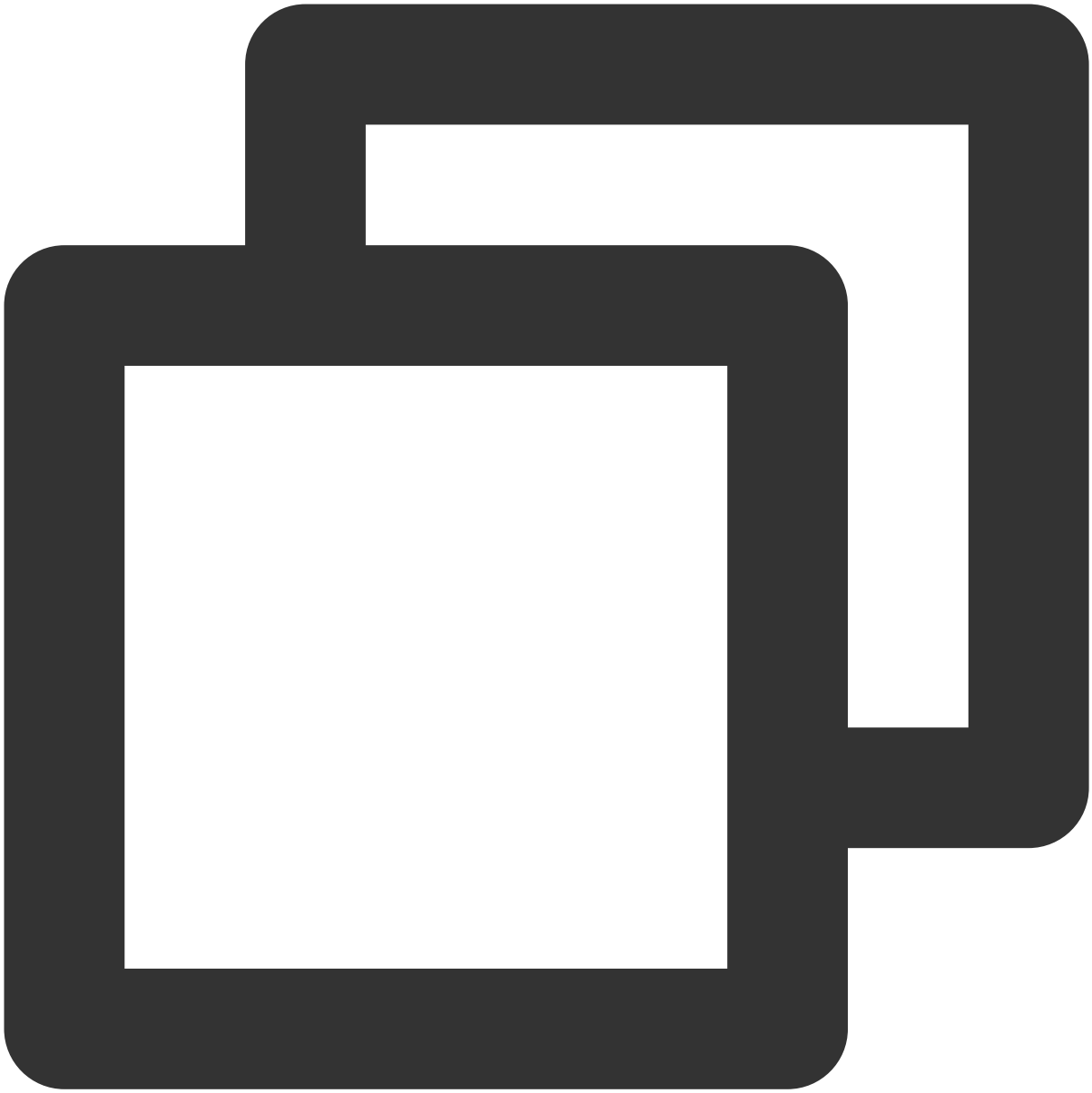
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
MediaBucket	Response	媒体 Bucket 详情，同 DescribeMediaBuckets 中的 Response.MediaBucketList	Container

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。

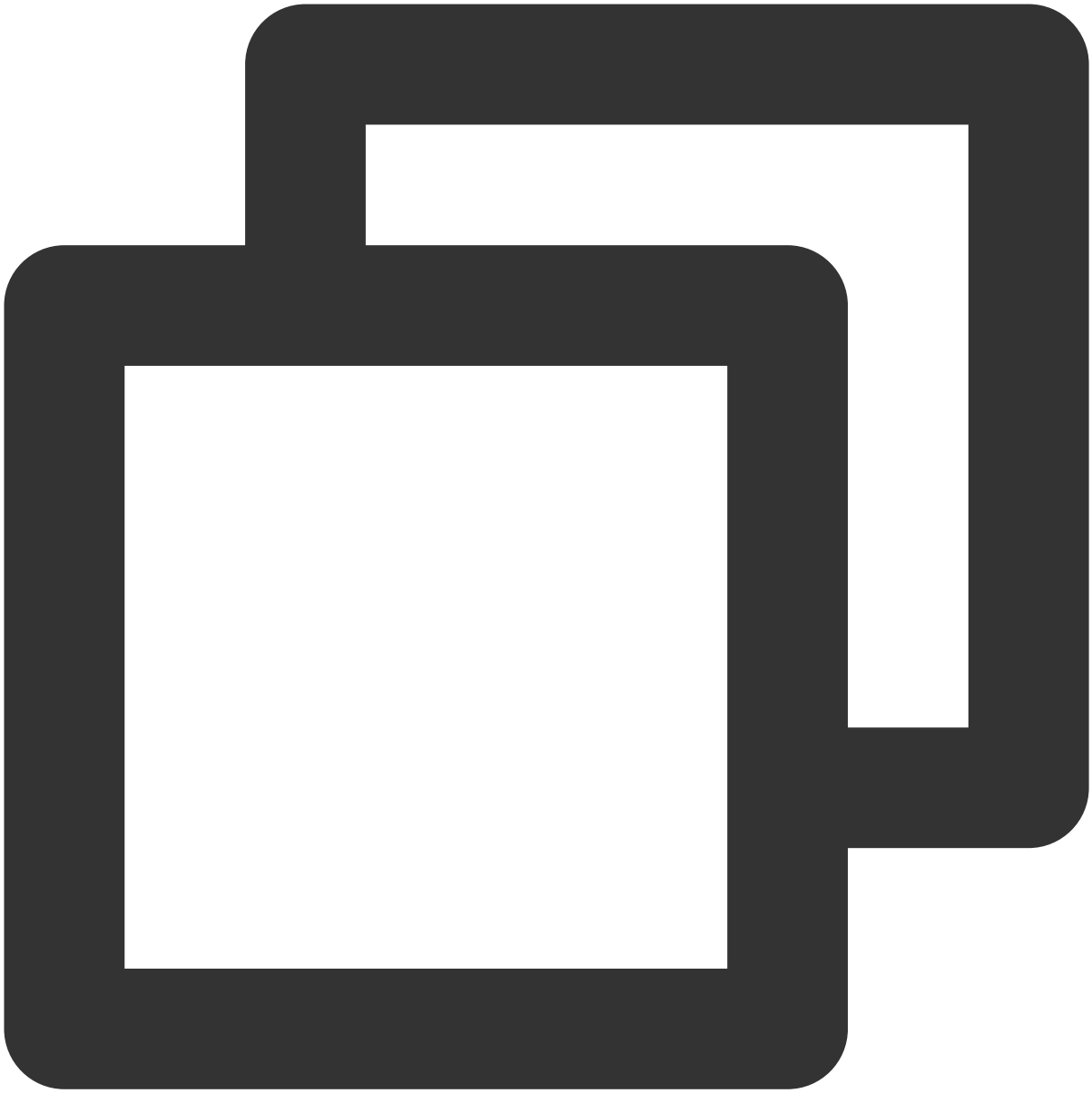
实际案例

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>
```

队列接口
搜索媒体处理队列

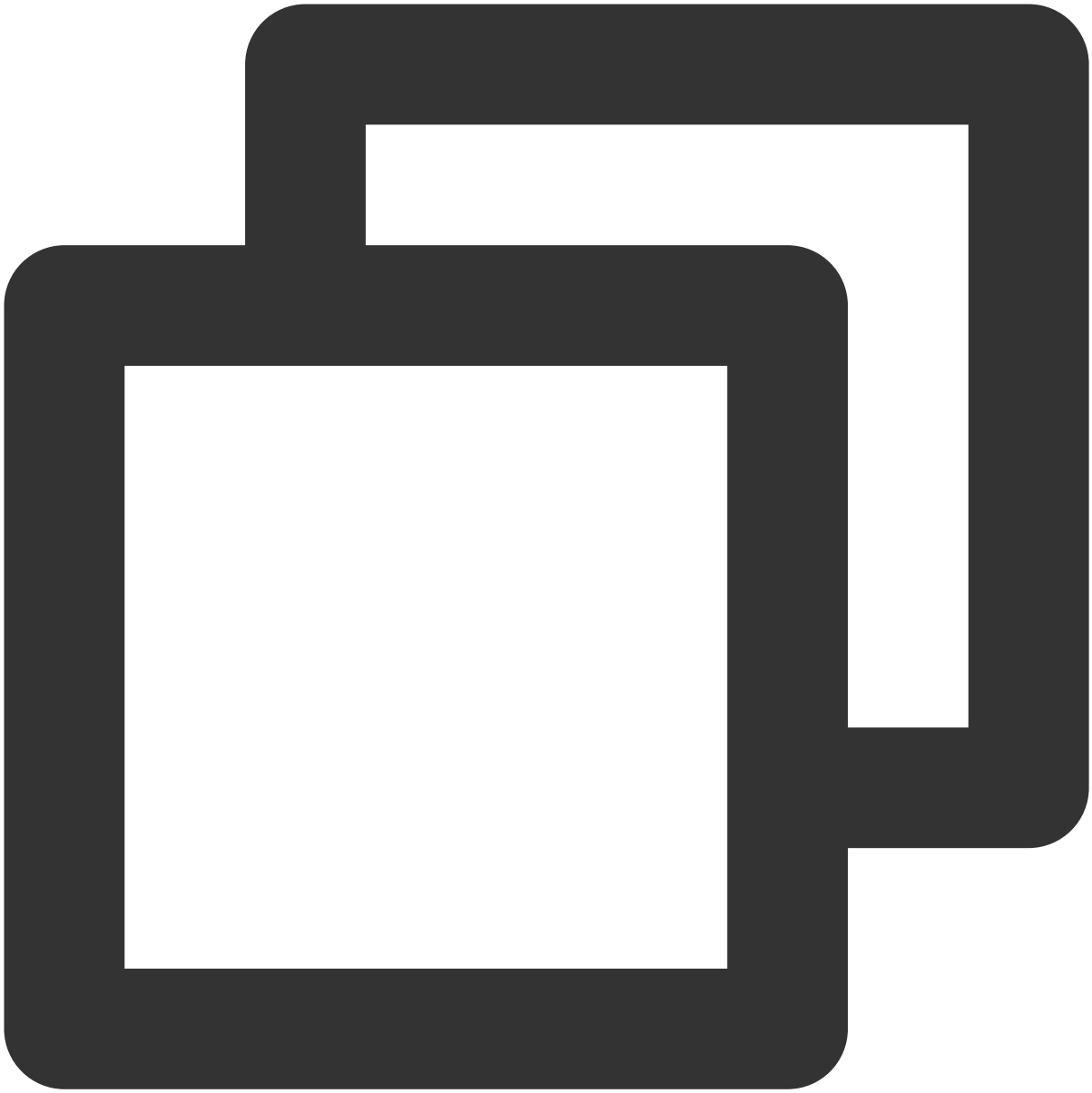
最近更新时间：2024-01-20 10:09:09

功能描述

搜索媒体处理队列。

请求

请求示例



```
GET /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
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
category	1. CateAll：所有类型。 2. Transcoding：媒体处理队列。 3. SpeedTranscoding：媒体处理倍速转码队列。 4. 默认为 Transcoding。	String	否
pageNumber	第几页，默认值1。	String	否
pageSize	每页个数，默认值10。	String	否

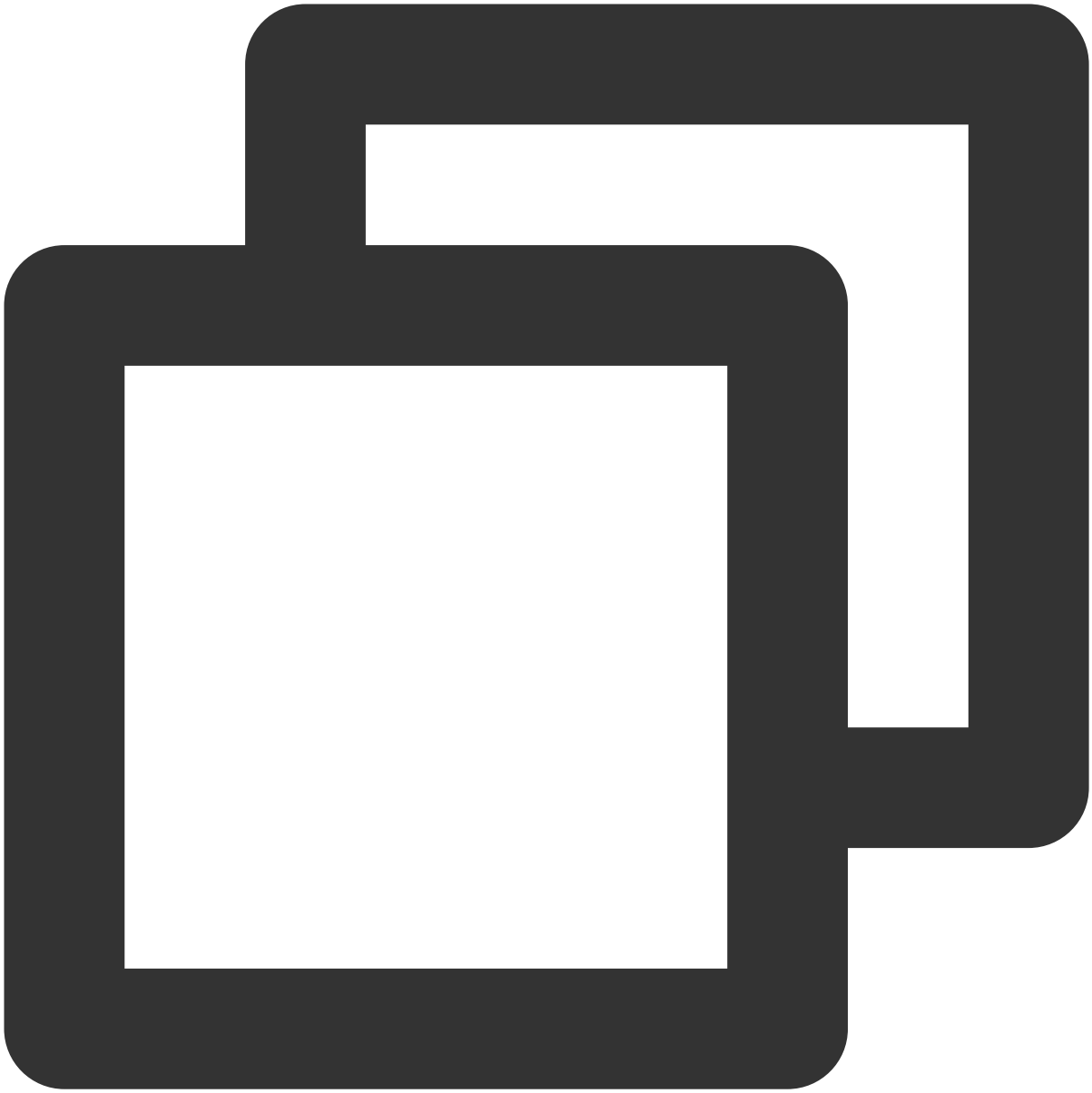
响应

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>2</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>queue-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>Transcoding</Category>
    </QueueList>
    <QueueList>
        <QueueId>pbf17a810dce74c18a641b663aede6803</QueueId>
        <Name>queue-speed-transcoding-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>SpeedTranscoding</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
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

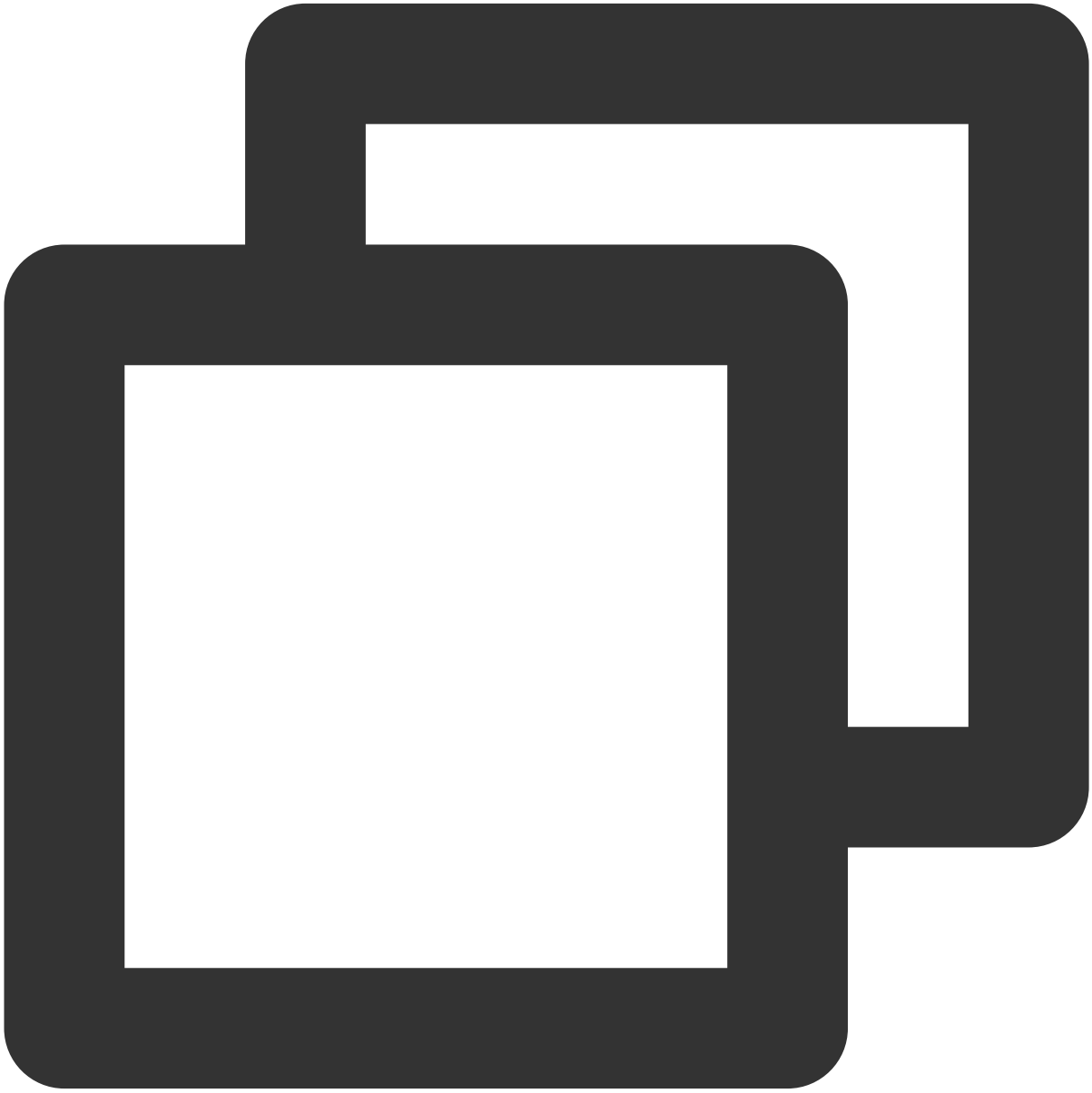
版权所有：腾讯云计算（北京）有限责任公司

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

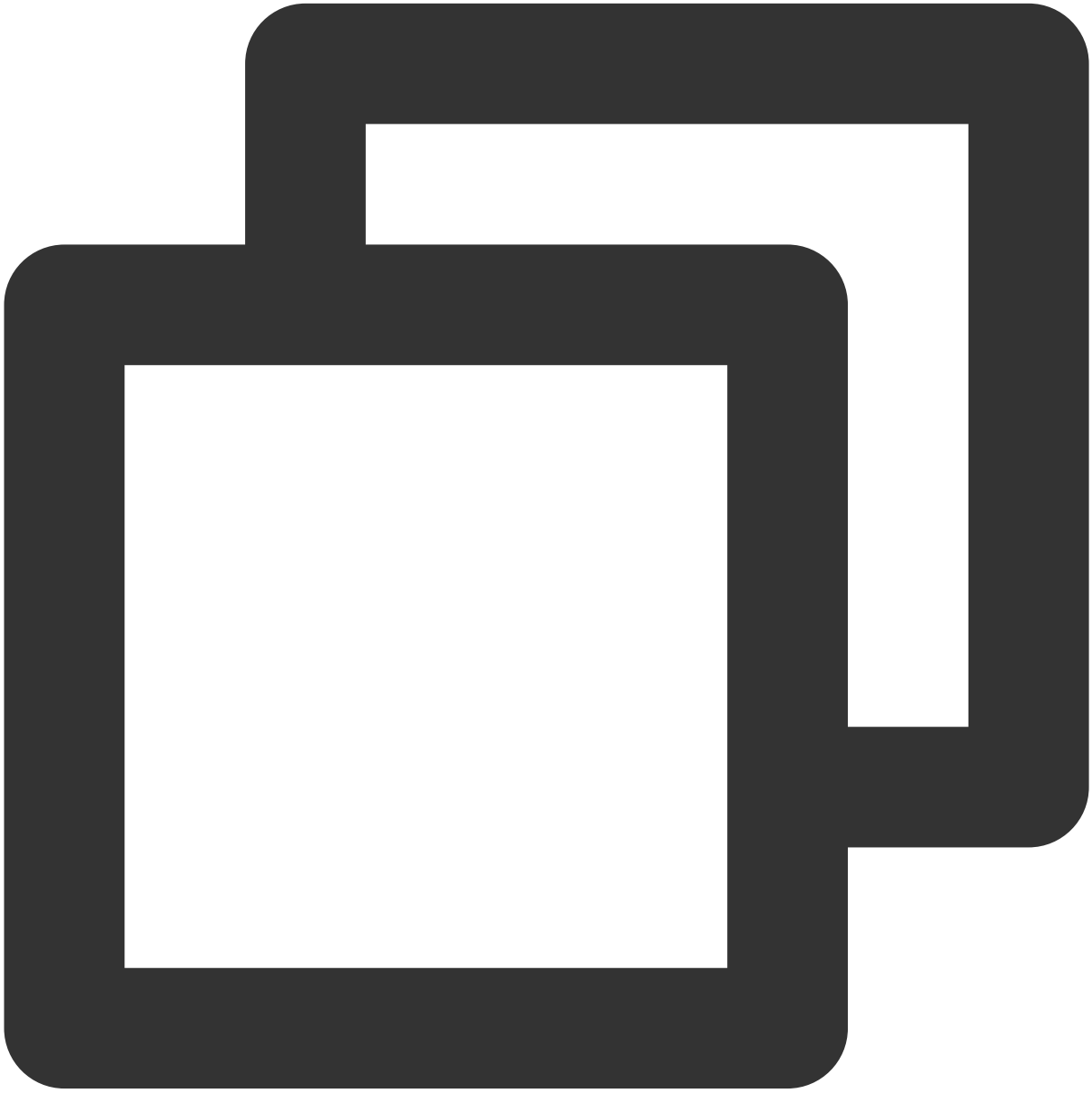
请求1：搜索指定队列



```
GET /queue?queueIds=p4a17eeea29334bf499b7e20e2fbfd99d,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

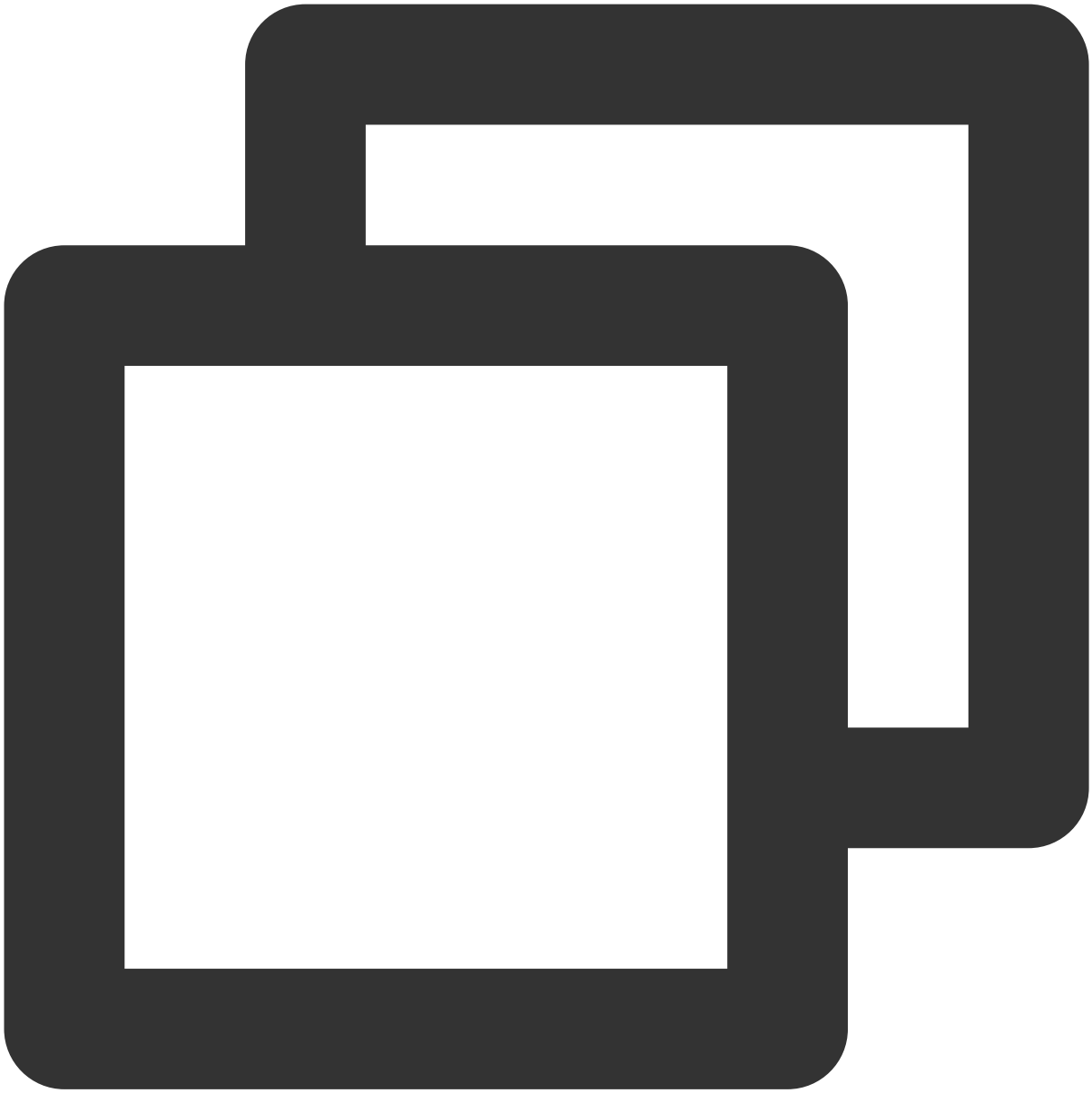


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 09 Aug 2022 16:23:12 GMT
```

```
Server: tencent-ci
x-ci-request-id: NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x

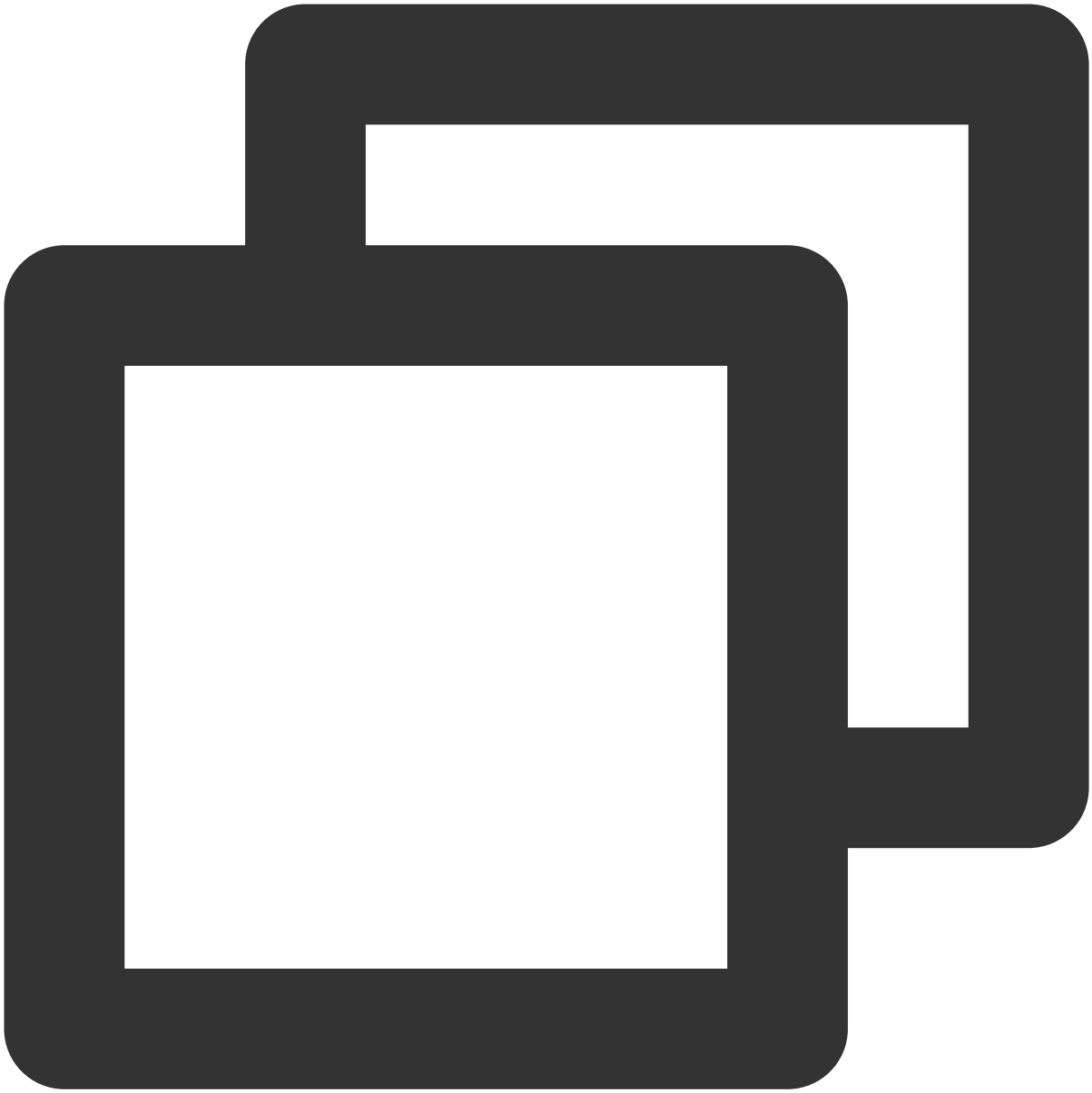
<Response>
    <RequestId>NjJmMzJjNDFfOTBmYTUwNjRfNjQ5NF8x</RequestId>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>queue-1</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</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>Transcoding</Category>
    </QueueList>
    <NonExistPIDs>A</NonExistPIDs>
    <NonExistPIDs>B</NonExistPIDs>
</Response>
```

请求2：查询队列列表



```
GET /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>2</TotalCount>
```

```
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>queue-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>Transcoding</Category>
    </QueueList>
    <QueueList>
        <QueueId>pbf17a810dce74c18a641b663aede6803</QueueId>
        <Name>queue-speed-transcoding-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>SpeedTranscoding</Category>
    </QueueList>
</Response>
```

更新媒体处理队列

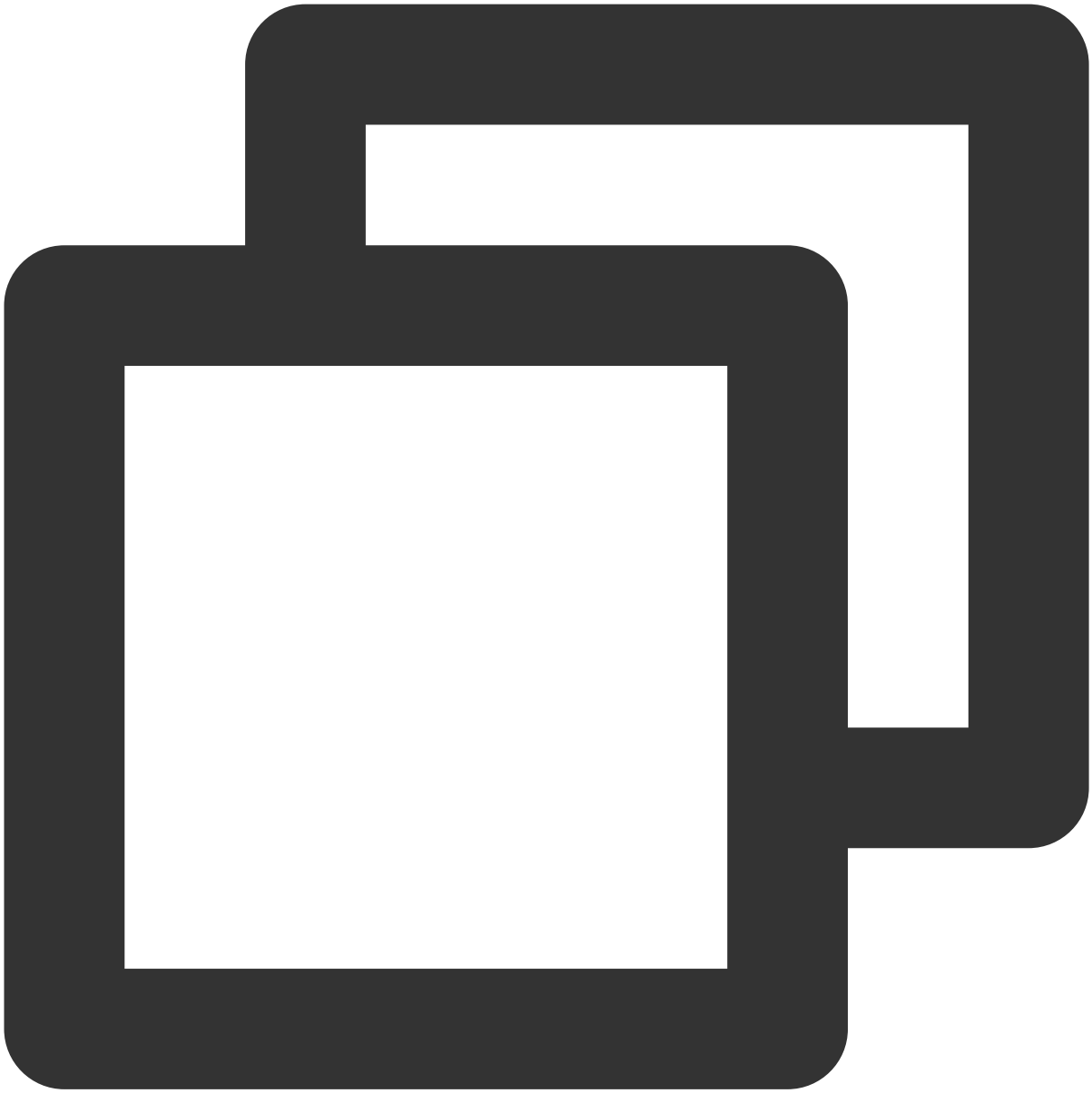
最近更新时间：2024-01-20 10:09:09

功能描述

更新媒体处理队列。

请求

请求示例



```
PUT /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：

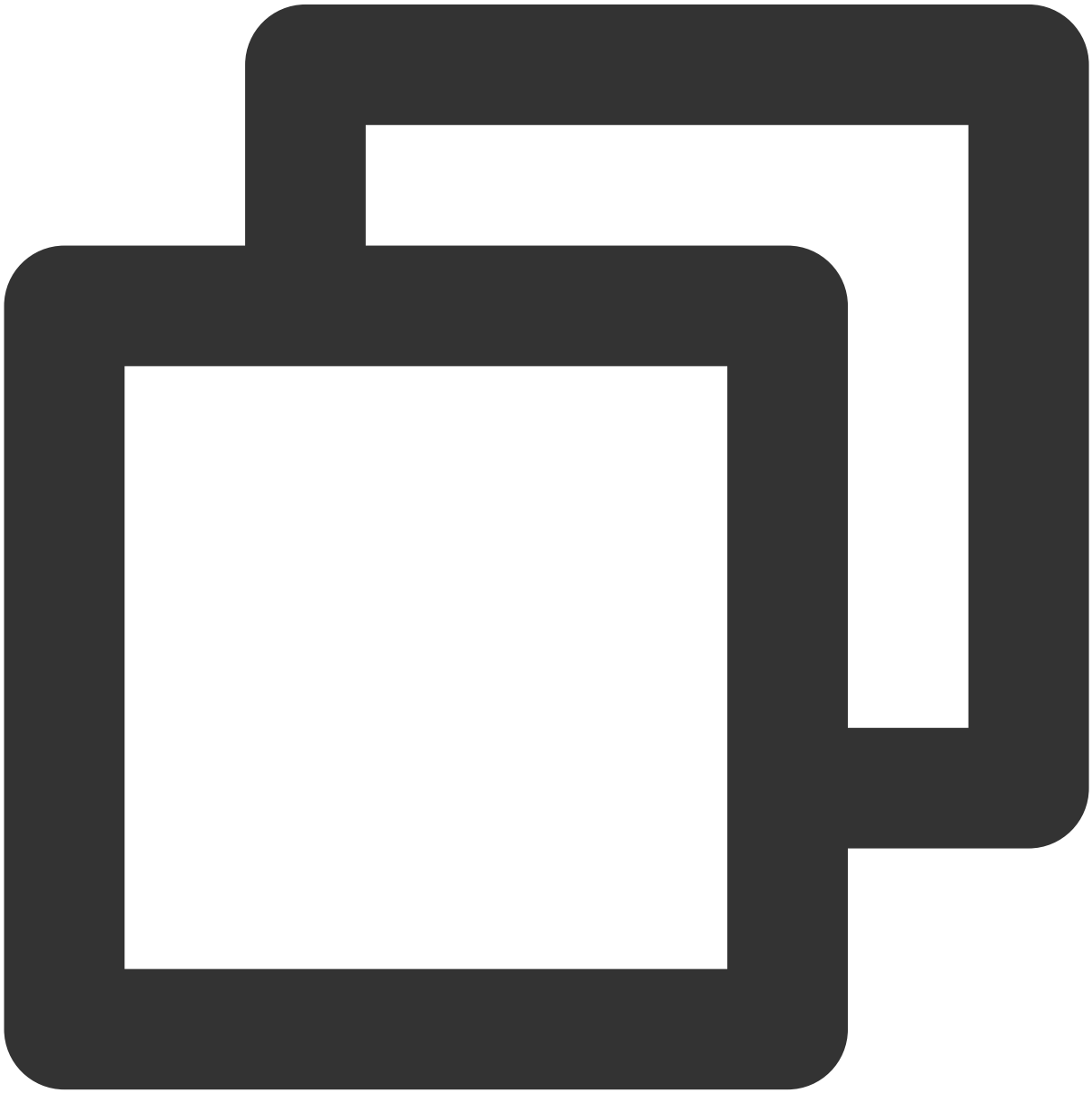
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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-Media</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	On/Off
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

版权所有：腾讯云计算（北京）有限责任公司

ResultFormat	Request.NotifyConfig	回调格式	String	否	XML	JSON/XML
Type	Request.NotifyConfig	回调类型	String	当 State=On 时, 必选	无	Url 或 TDMQ
Url	Request.NotifyConfig	回调地址	String	当 State=On, 且 Type=Url 时, 必选	无	不能为内网地址
MqMode	Request.NotifyConfig	TDMQ 使用模式	String	当 State=On, 且 Type=TDMQ 时, 必选	Queue	主题订阅：Topic 队列服务: Queue
MqRegion	Request.NotifyConfig	TDMQ 所属园区	String	当 State=On, 且 Type=TDMQ 时, 必选	无	目前支持园区 sh（上海）、bj（北京）、gz（广州）、cd（成都）、hk（中国香港）
MqName	Request.NotifyConfig	TDMQ 主题名称	String	当 State=On, 且 Type=TDMQ 时, 必选	无	无

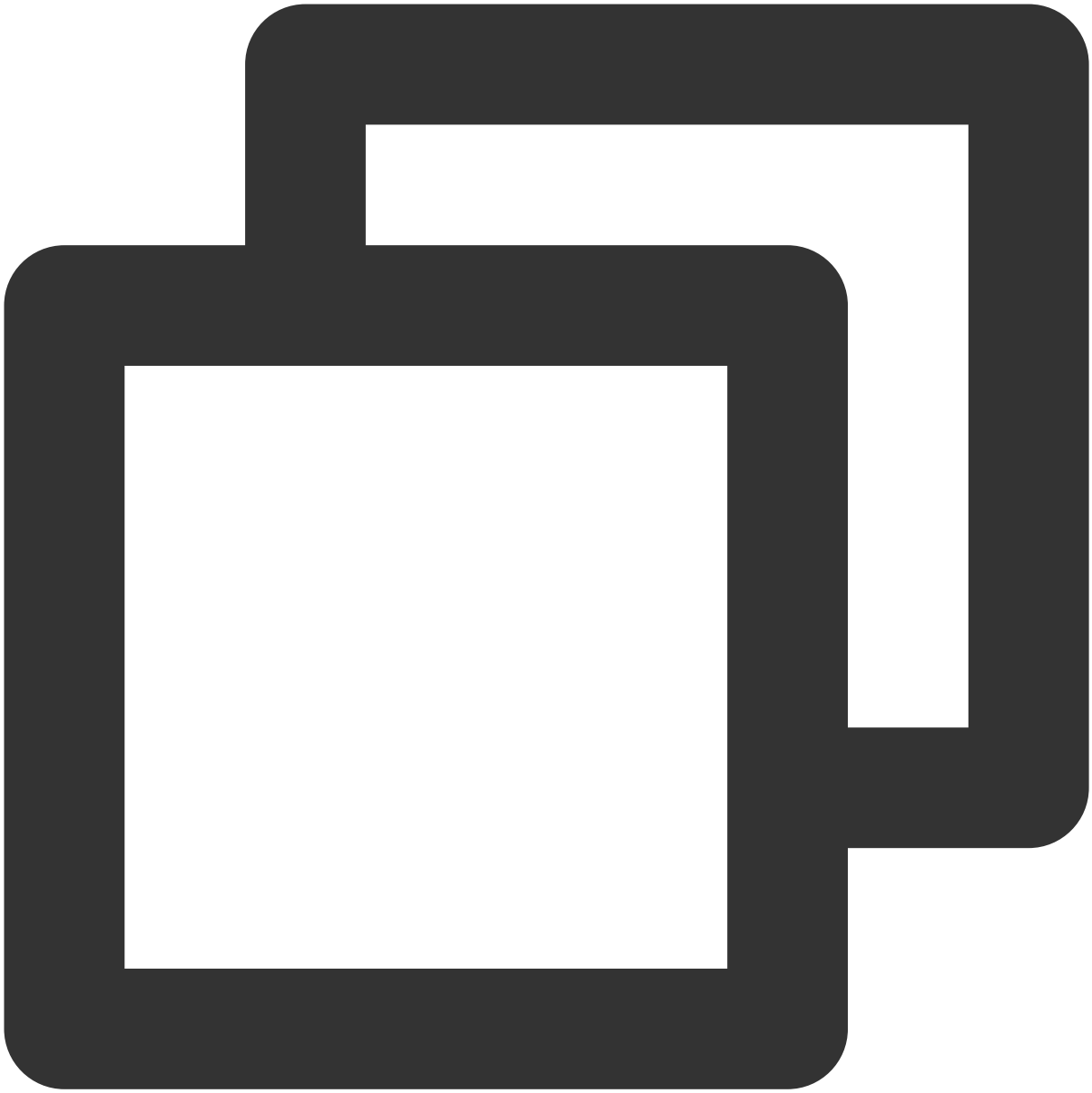
响应

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>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <Queue>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>My-Queue-Media</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：

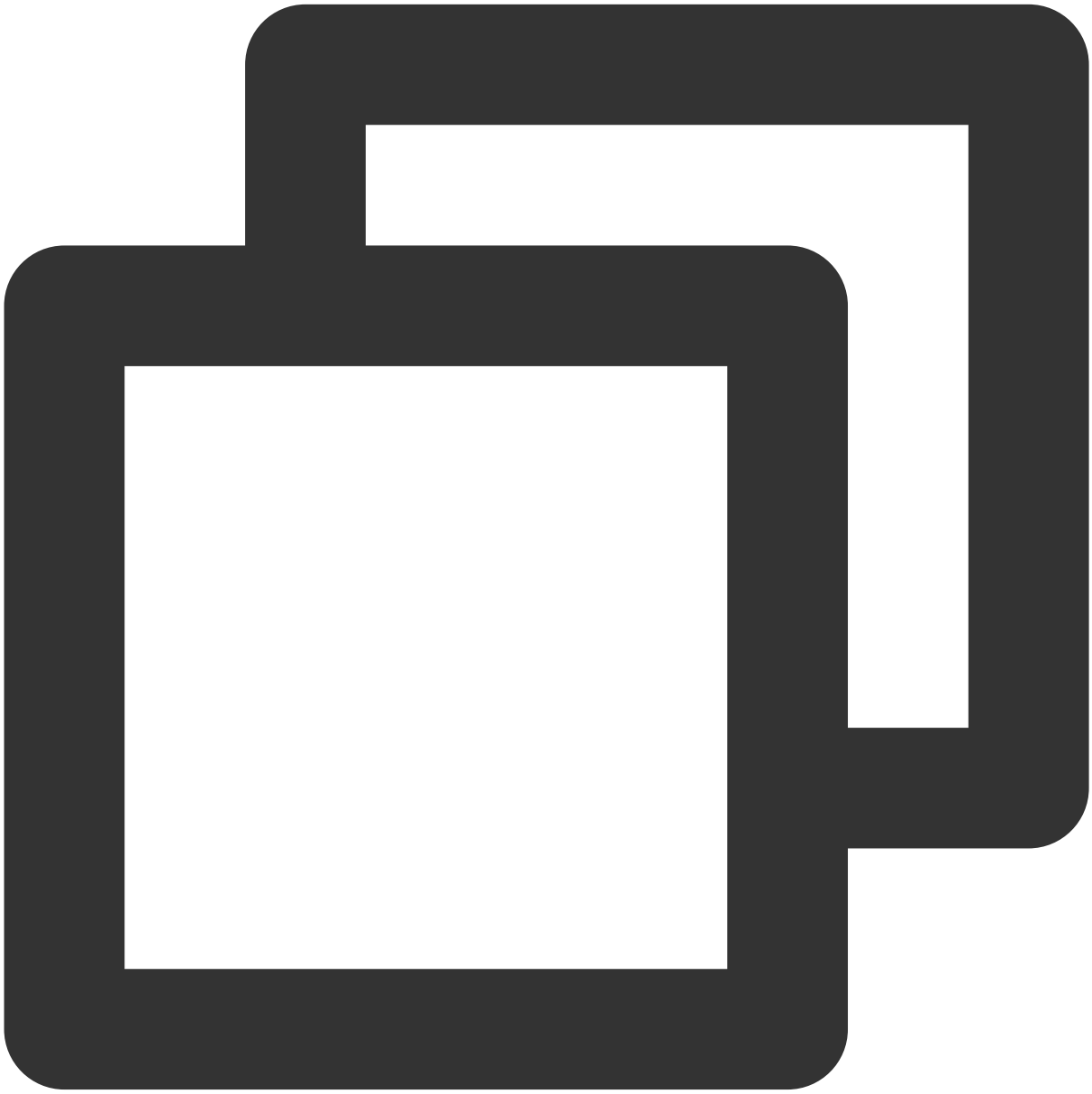
节点名称（关键字）	父节点	描述	类型
QueueId	Response.Queue	队列 ID	String
Name	Response.Queue	队列名字	String
State	Response.Queue	当前状态，Active 或者 Paused	String
NotifyConfig	Response.Queue	回调配置，同 Request.NotifyConfig	Container
MaxSize	Response.Queue	队列最大长度	Int
MaxConcurrent	Response.Queue	当前队列最大并行执行的任务数	Int
Category	Response.Queue	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

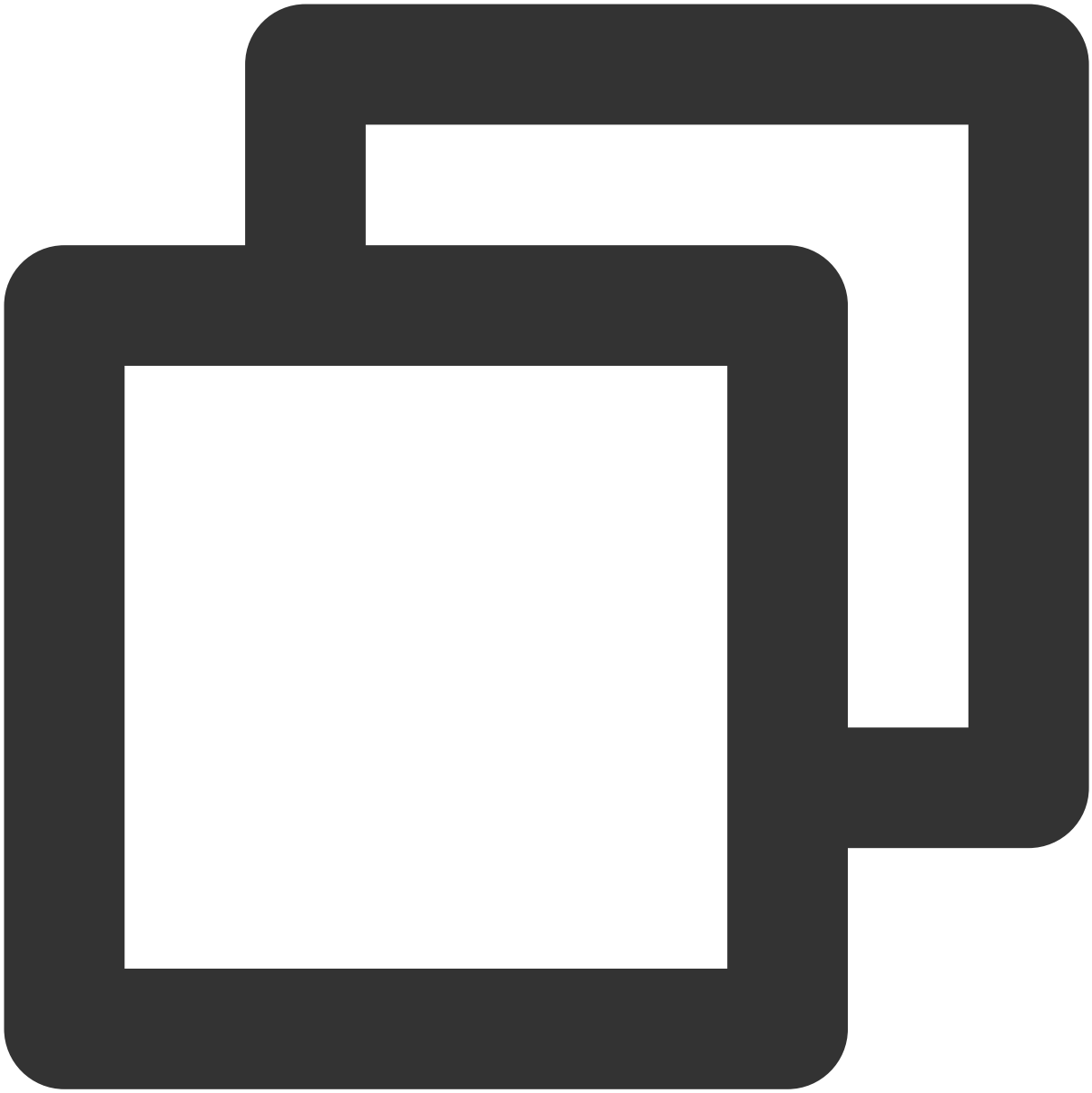


```
PUT /queue/p4a17eeea29334bf499b7e20e2fbfd99d HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

```
Content-Length: 162
Content-Type: application/xml

<Request>
    <Name>My-Queue-Media</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>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>My-Queue-Media</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>
```

同步媒体处理
获取媒体截图
获取媒体文件截图

最近更新时间：2024-01-20 10:09:09

功能描述

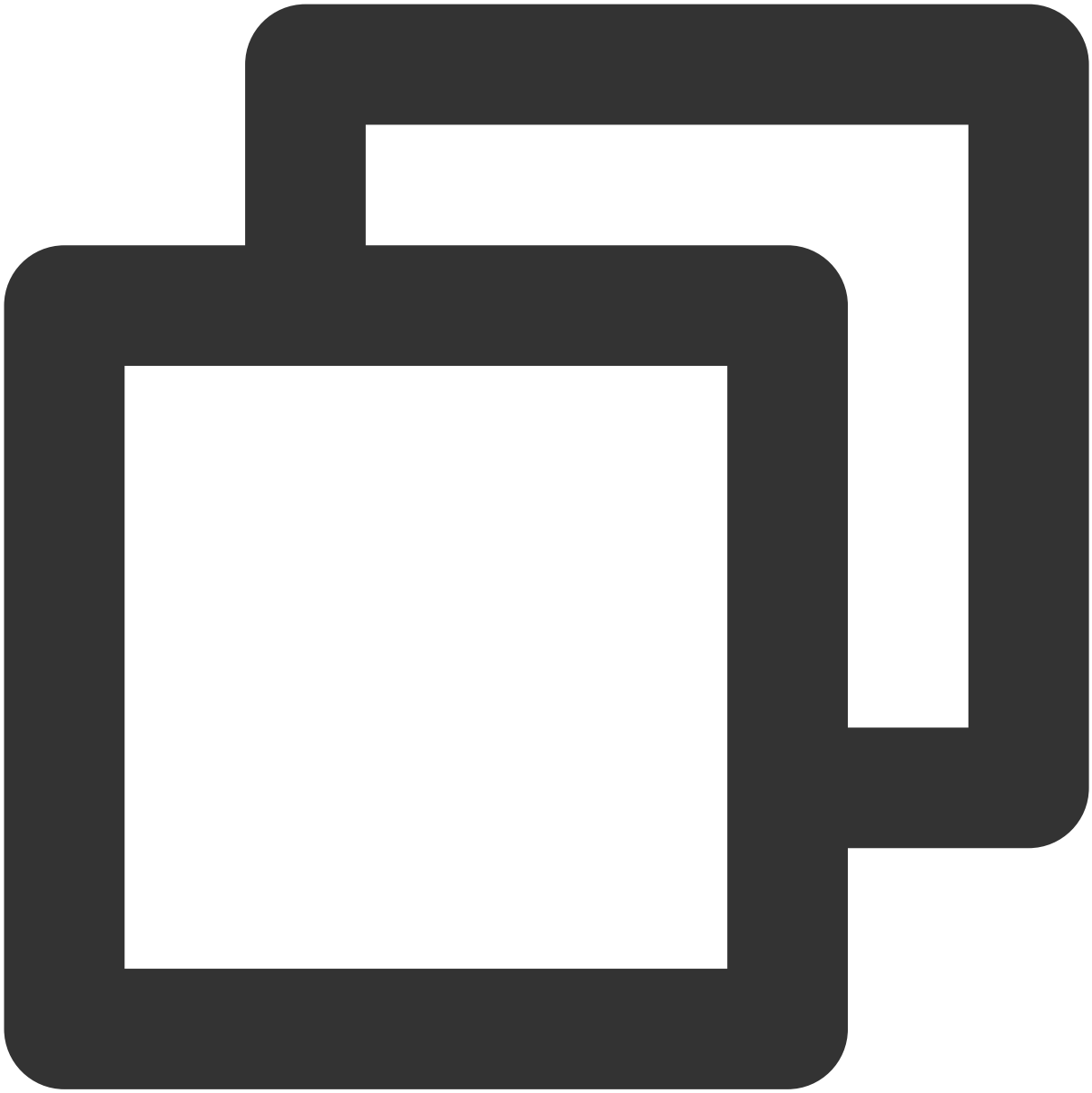
GetSnapshot 接口用于获取媒体文件某个时间的截图。

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。

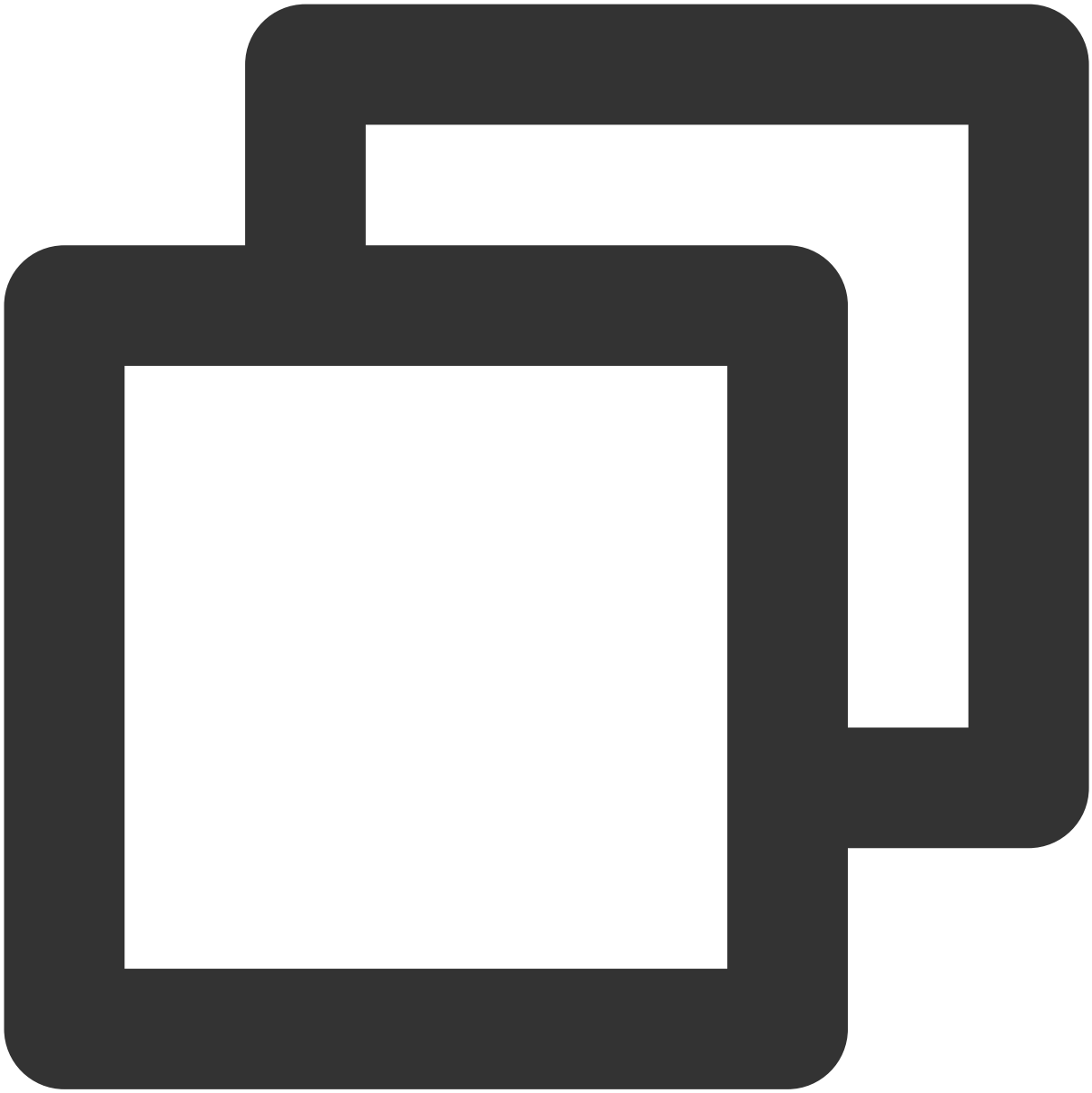
请求示例

1. 通过GET请求方式



```
GET /<ObjectKey>?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。

<ObjectKey>为存储桶对象。

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

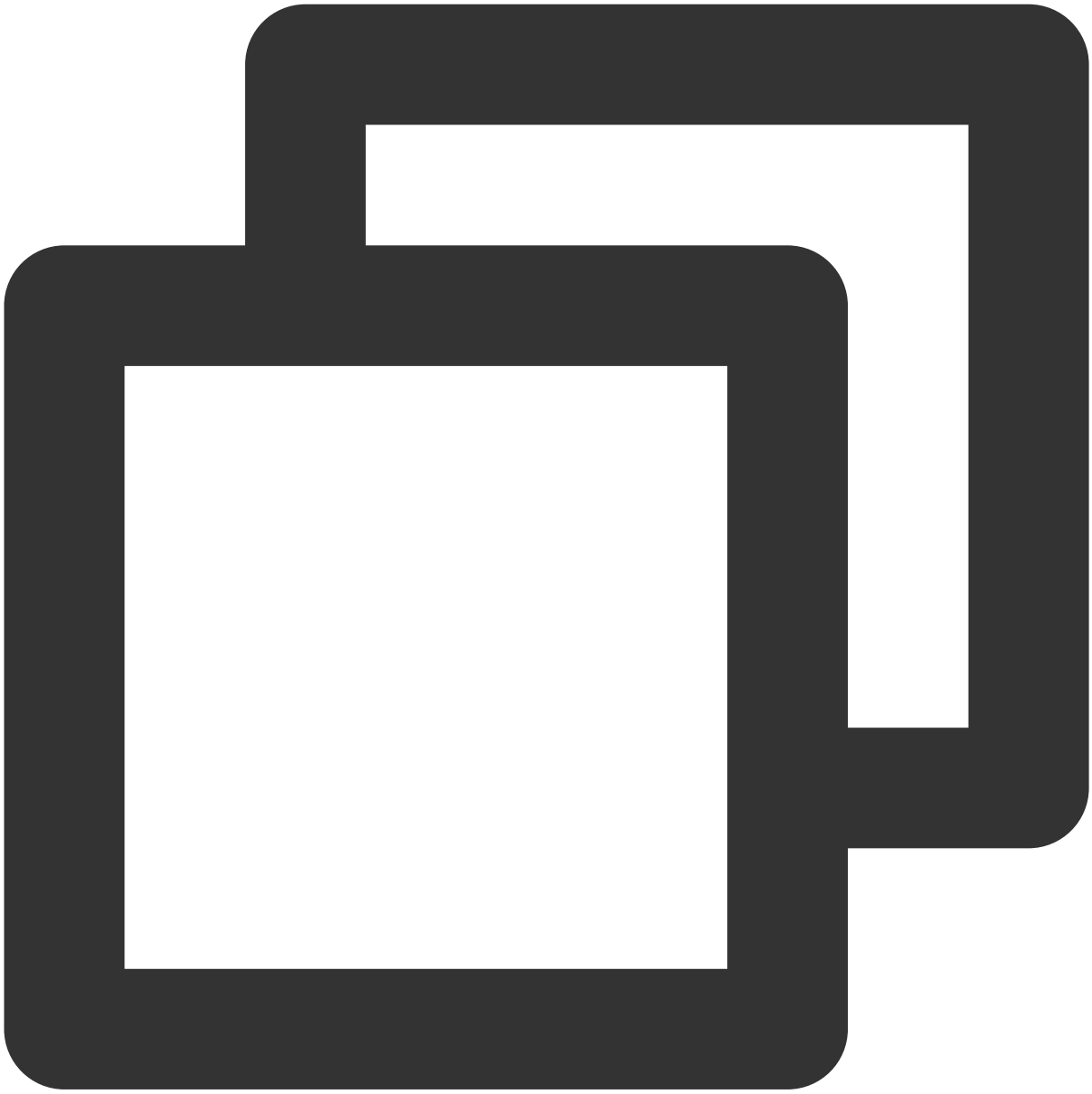
该响应体为截图文件内容。

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。

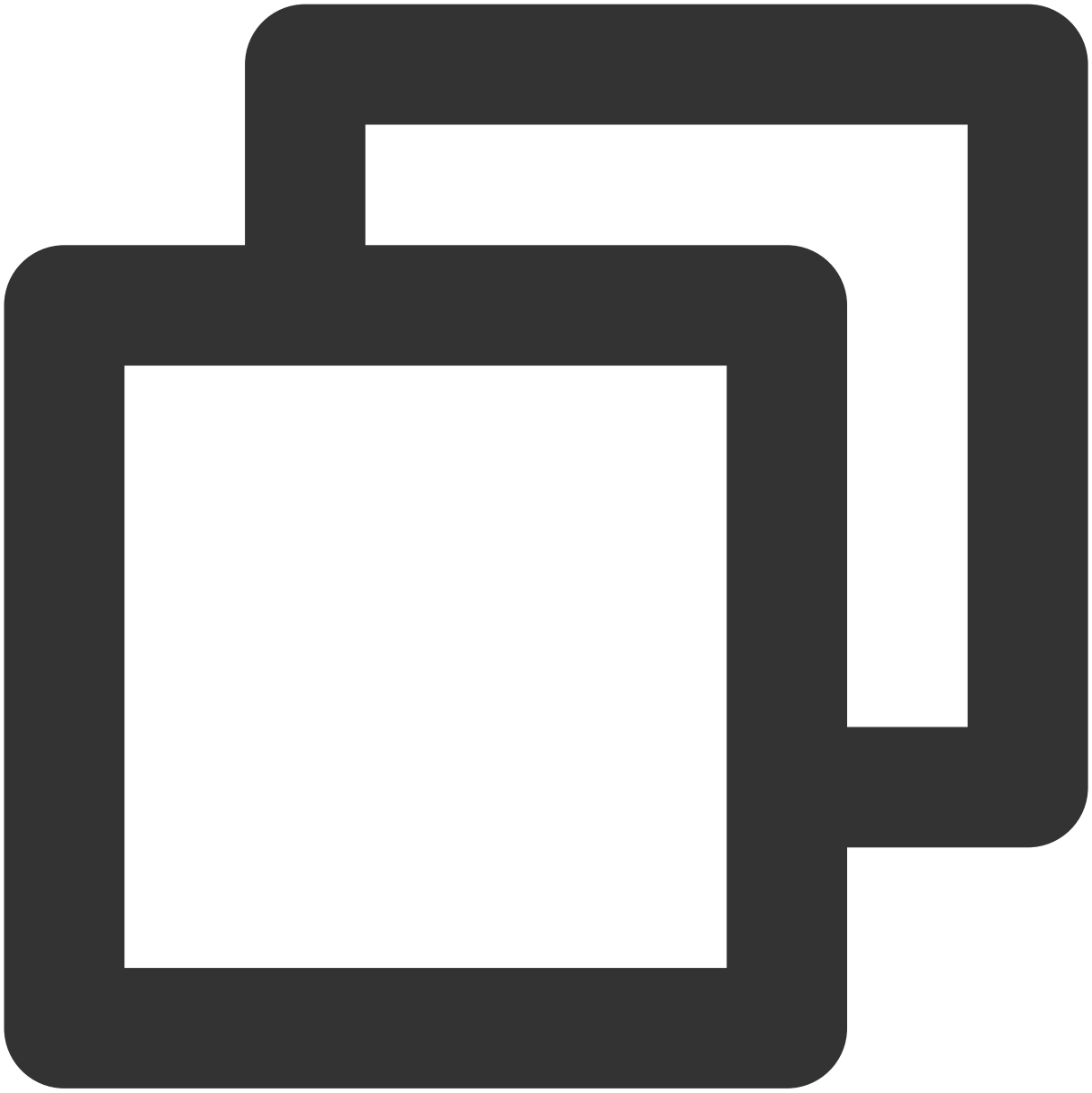
实际案例

请求1 通过GET请求方式



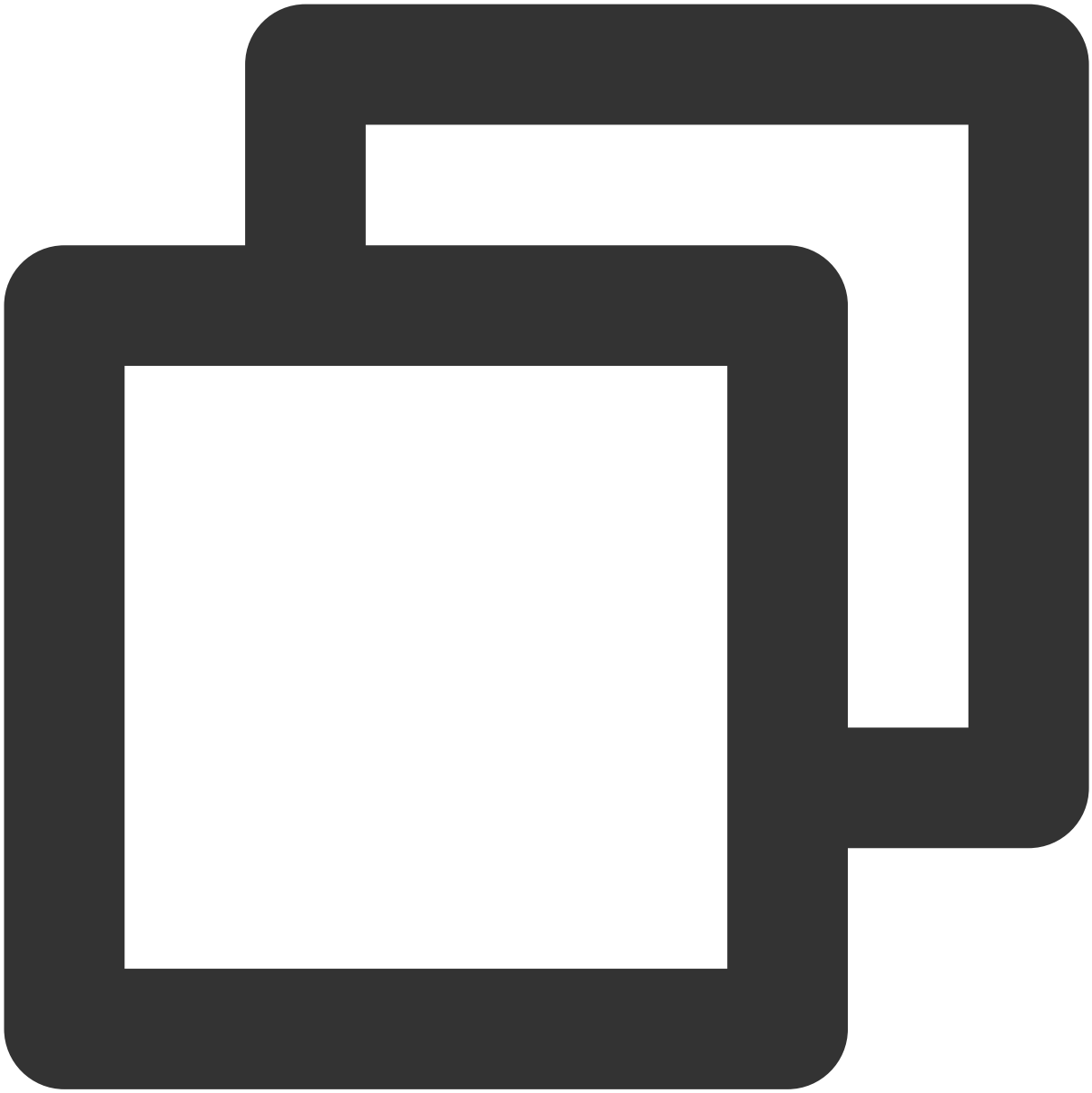
```
GET /for-test.mp4?ci-process=snapshot&time=1.5 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG****-si
Content-Length: 0
```

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-ci
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==


<图片内容>
```

获取媒体文件截图（万象域名）

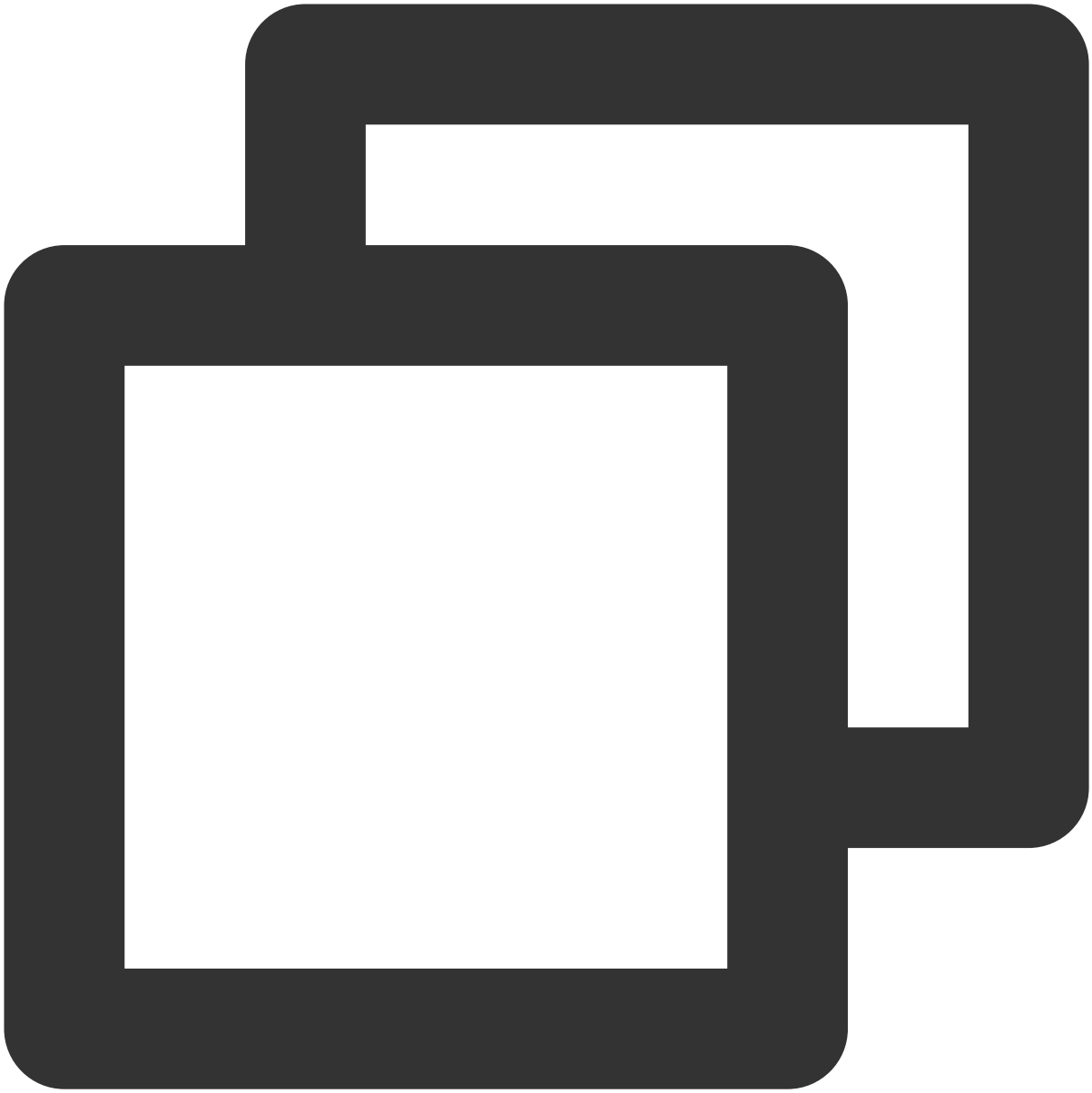
最近更新时间：2024-01-20 10:09:09

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：

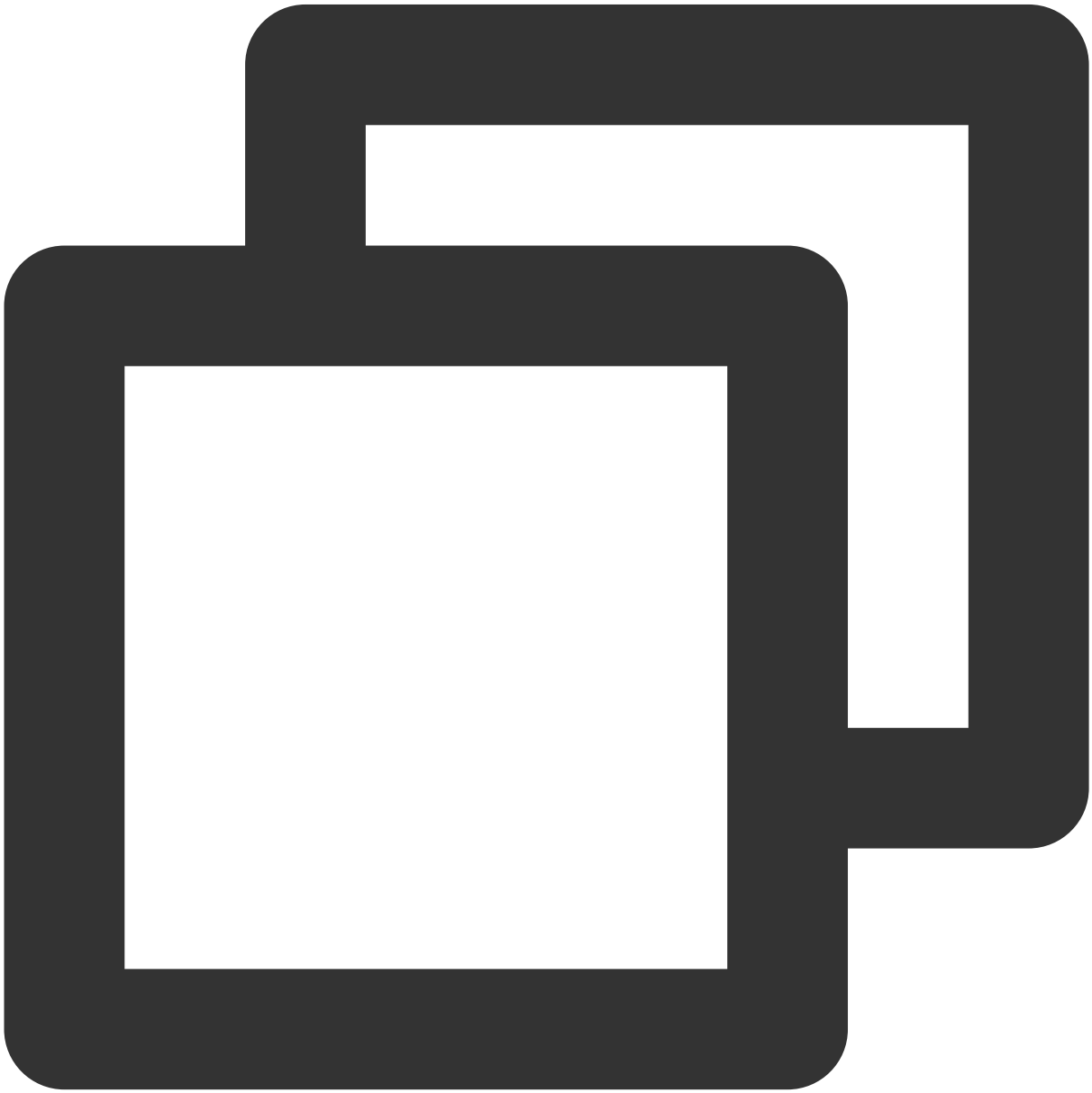
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

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 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	String	否

		默认值为 exactframe		
Rotate	Request	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

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

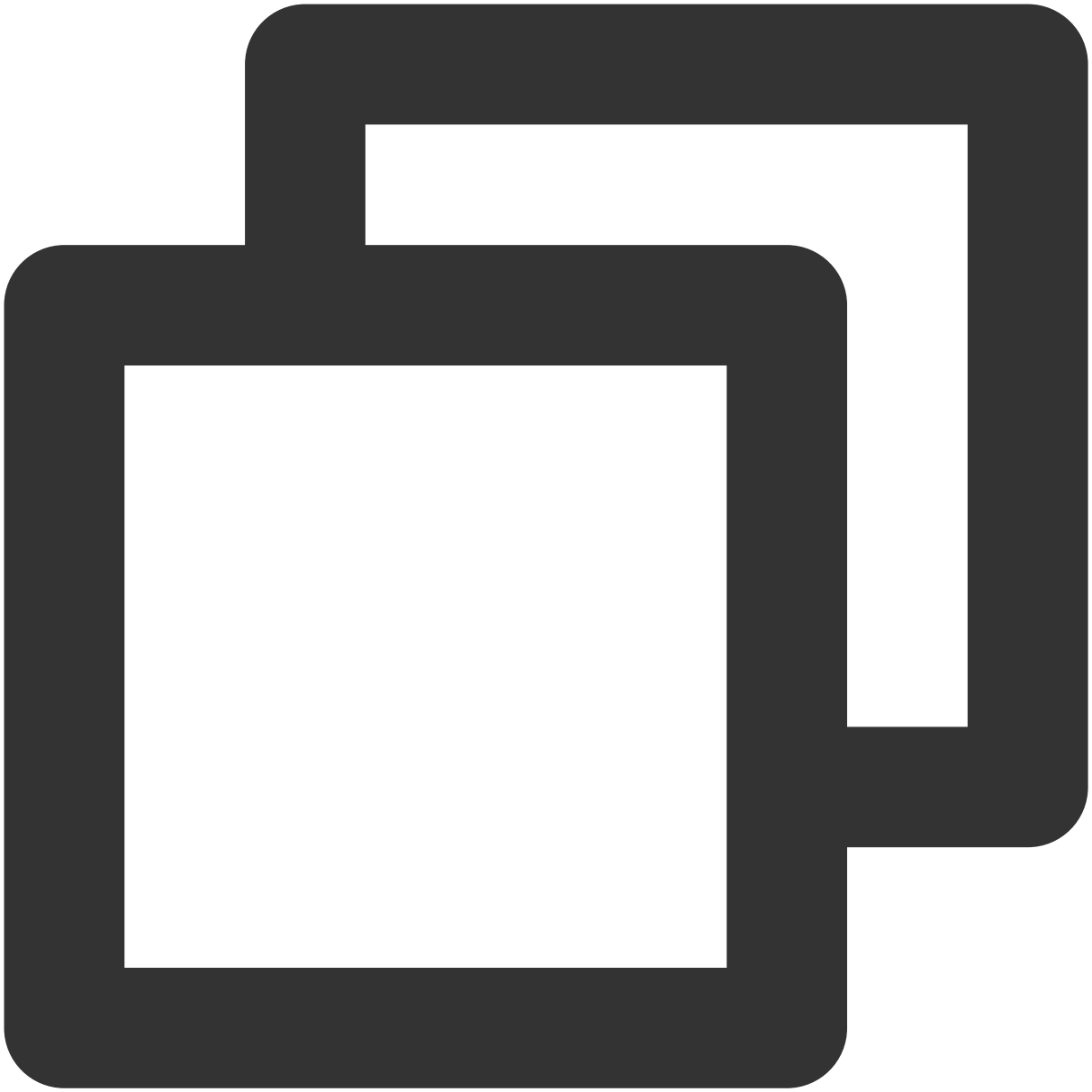
响应

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：

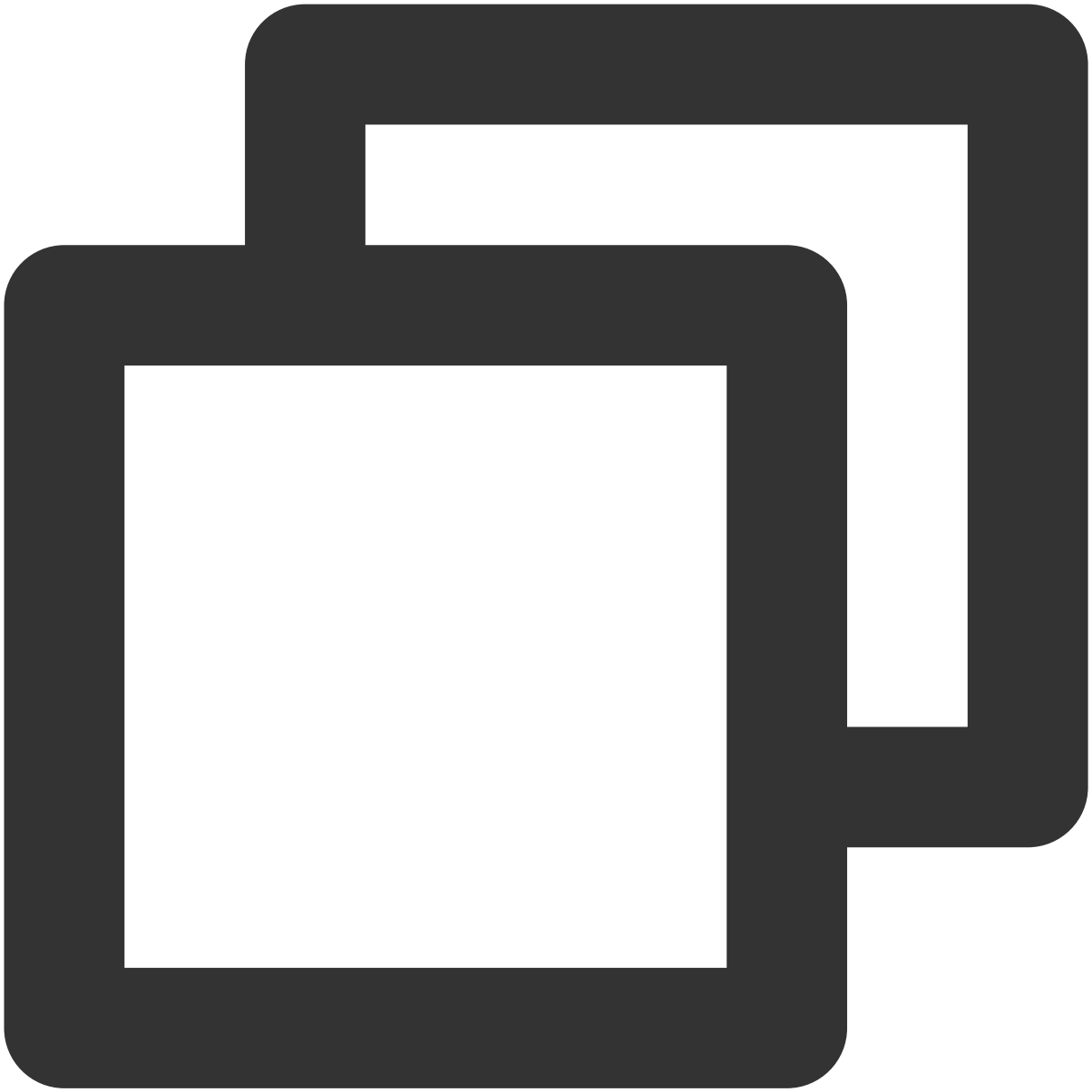
节点名称（关键字）	父节点	描述	类型
Region	Response.Output	存储桶所在的地域	String
Bucket	Response.Output	文件所属的 COS 存储桶	String
Object	Response.Output	文件的名称	String

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

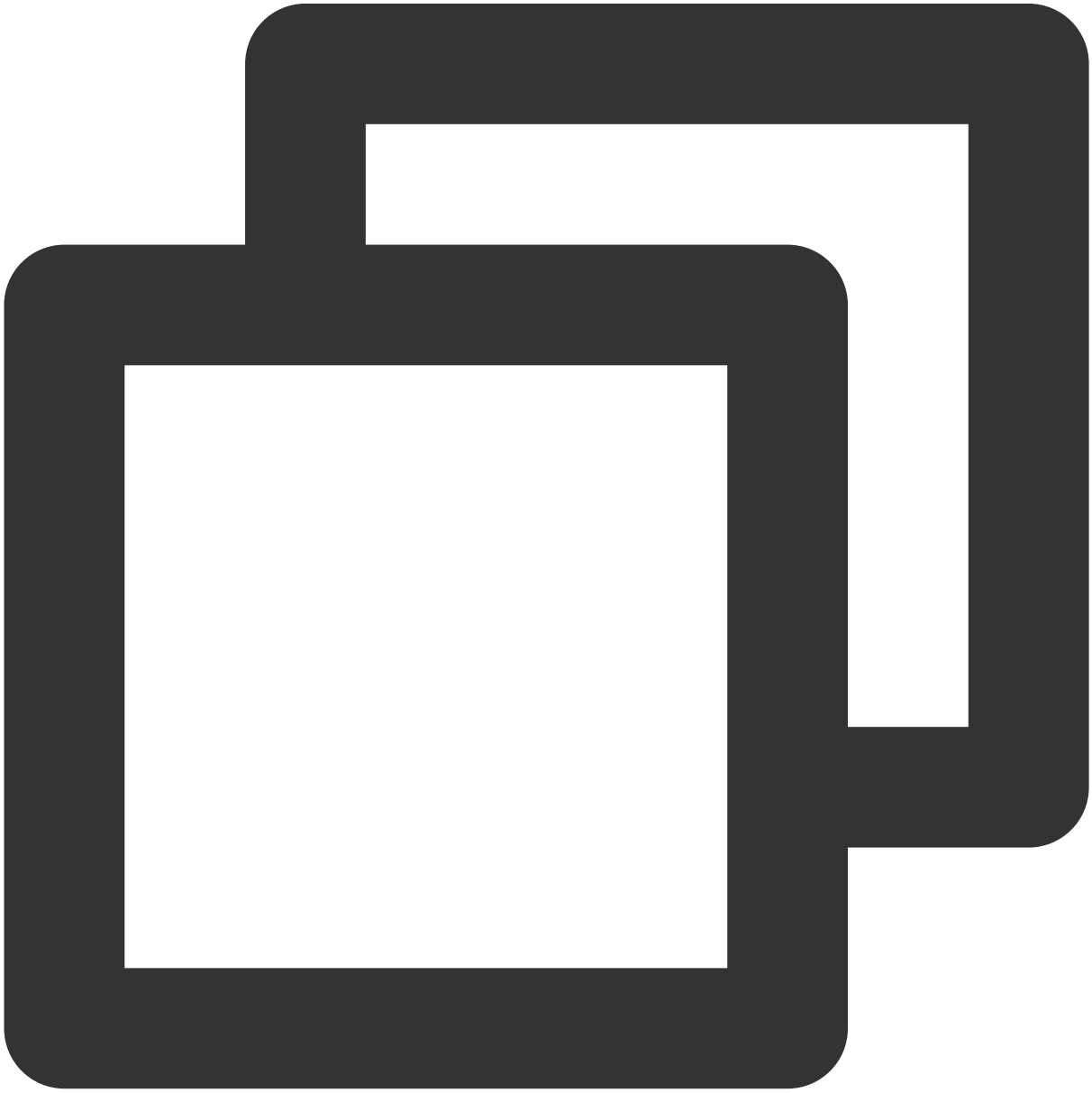


```
POST /snapshot HTTP/1.1
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
Content-Length: 552
Content-Type: application/xml

<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>
```

GenerateSnapshot

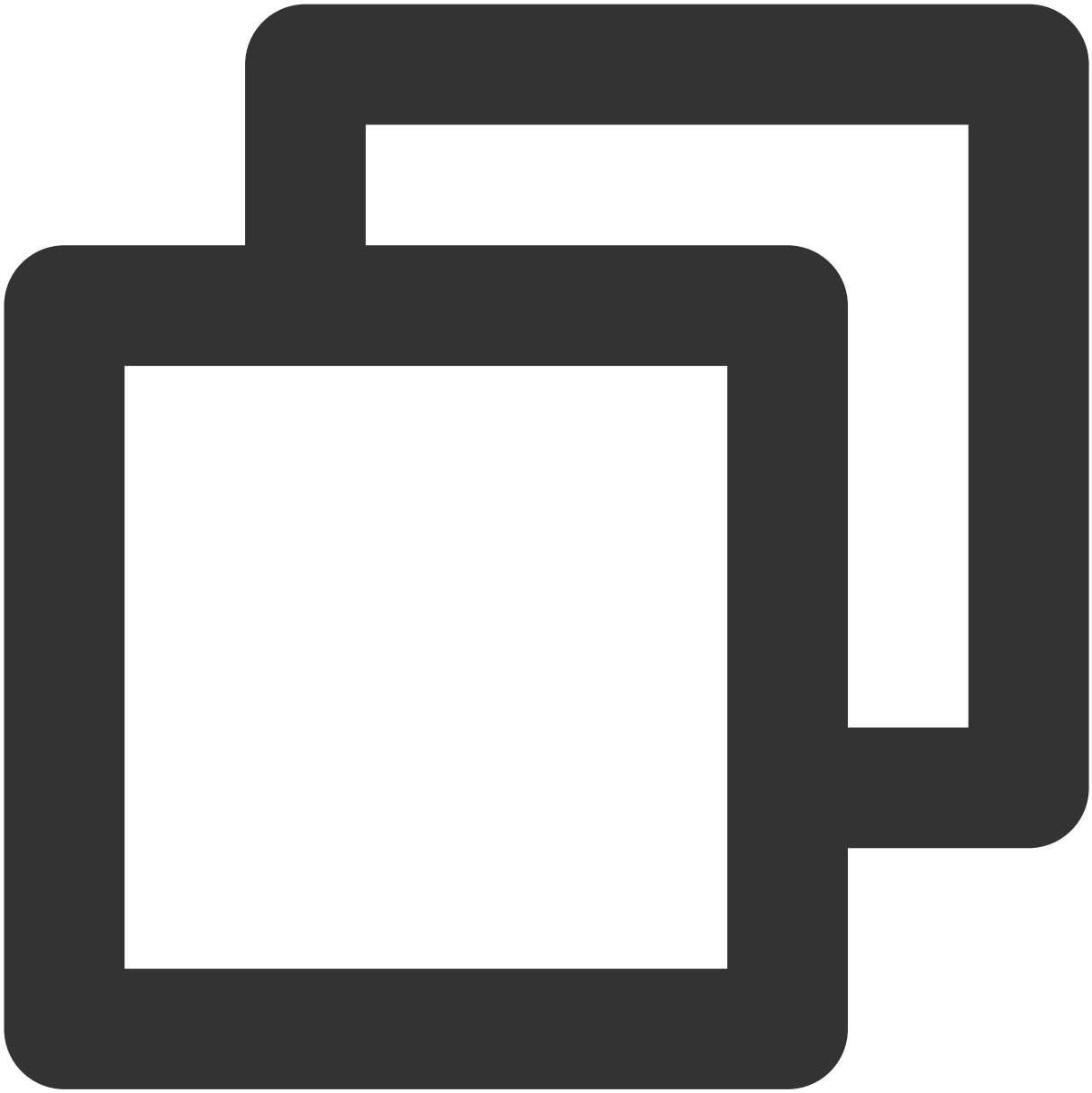
最近更新时间：2024-01-20 10:09:09

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：

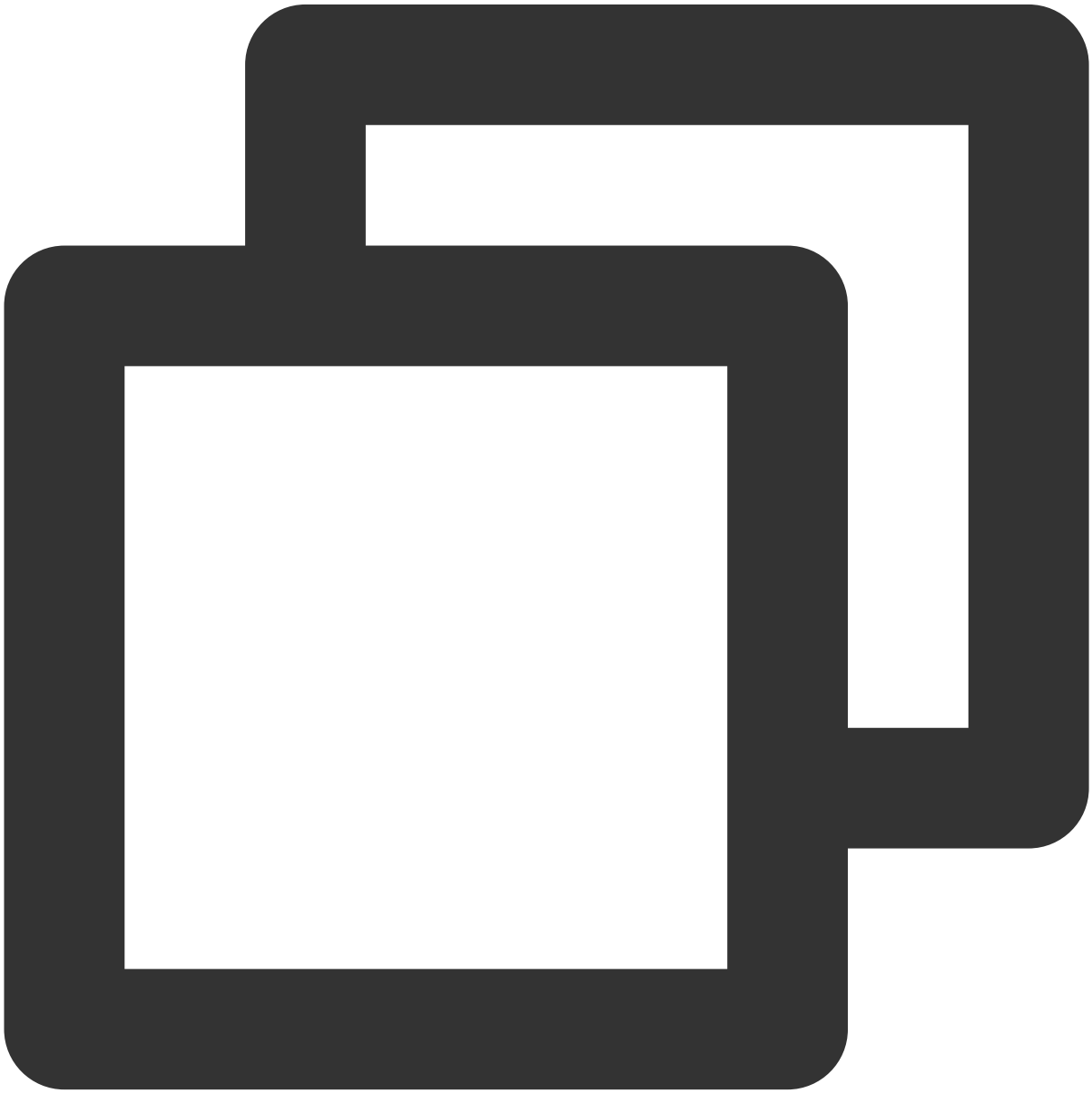
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见授权粒度详情文档。

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 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。	String	否

		auto：按视频旋转信息进行自动旋转 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

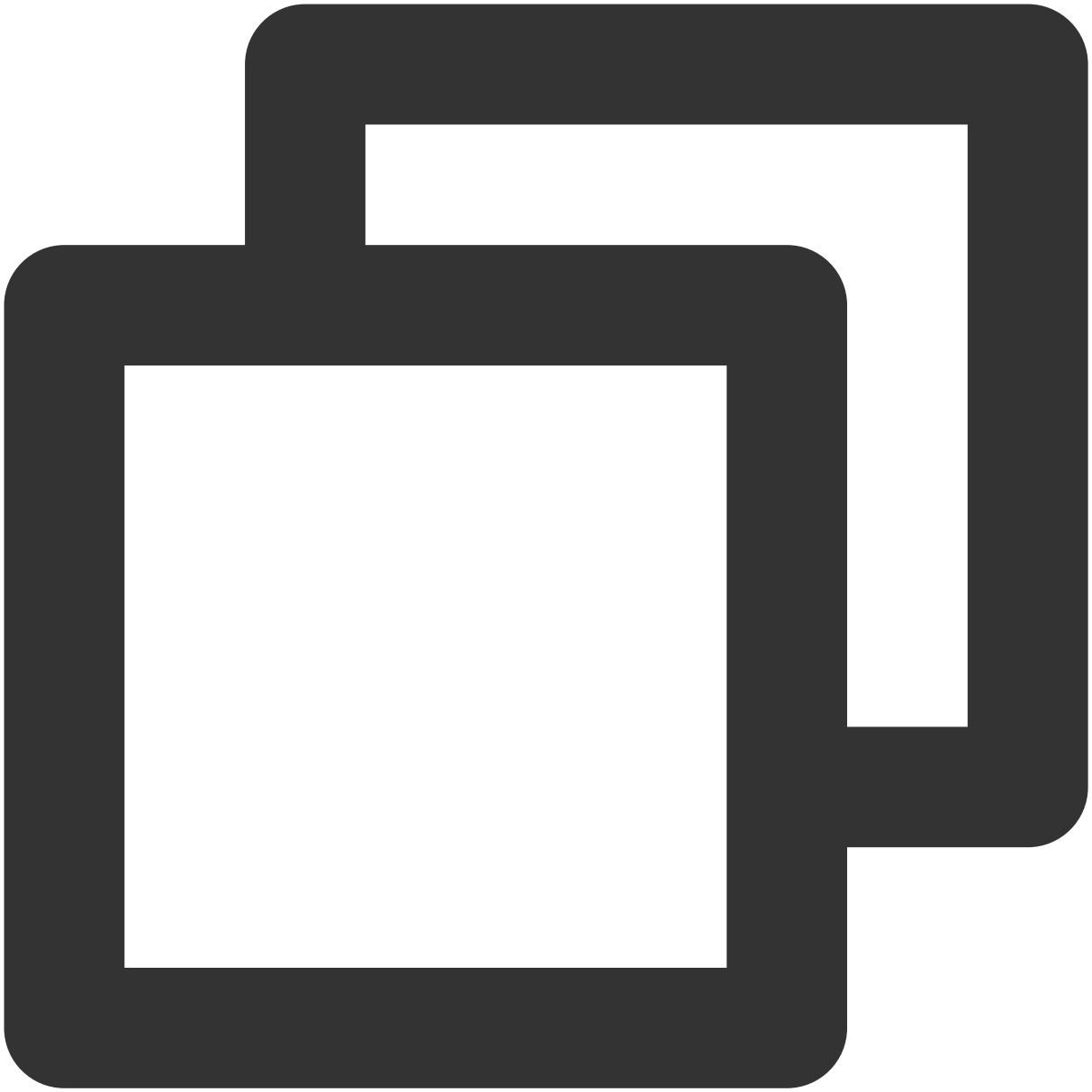
响应

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：

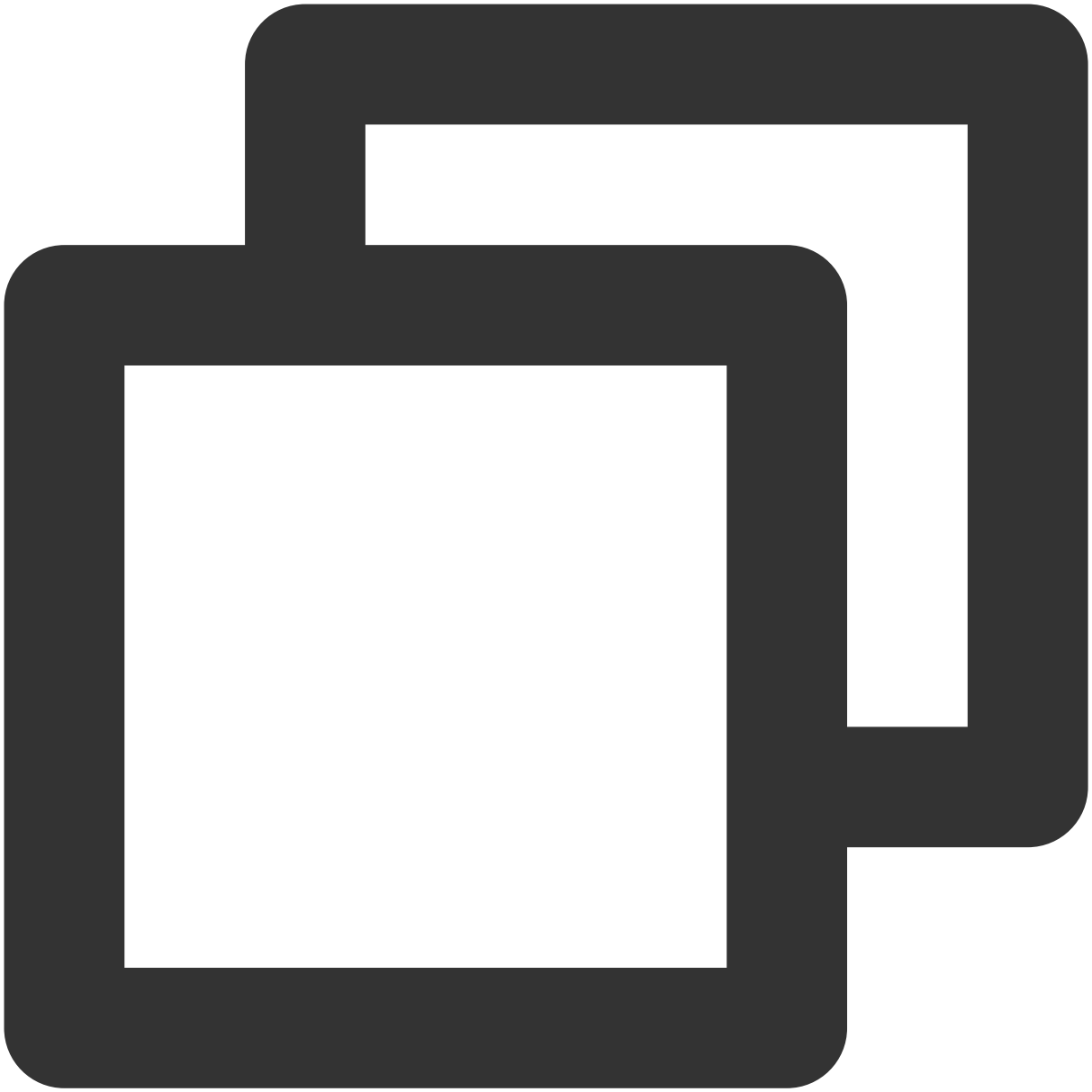
节点名称（关键字）	父节点	描述	类型
Region	Response.Output	存储桶所在的地域	String
Bucket	Response.Output	文件所属的 COS 存储桶	String
Object	Response.Output	文件的名称	String

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

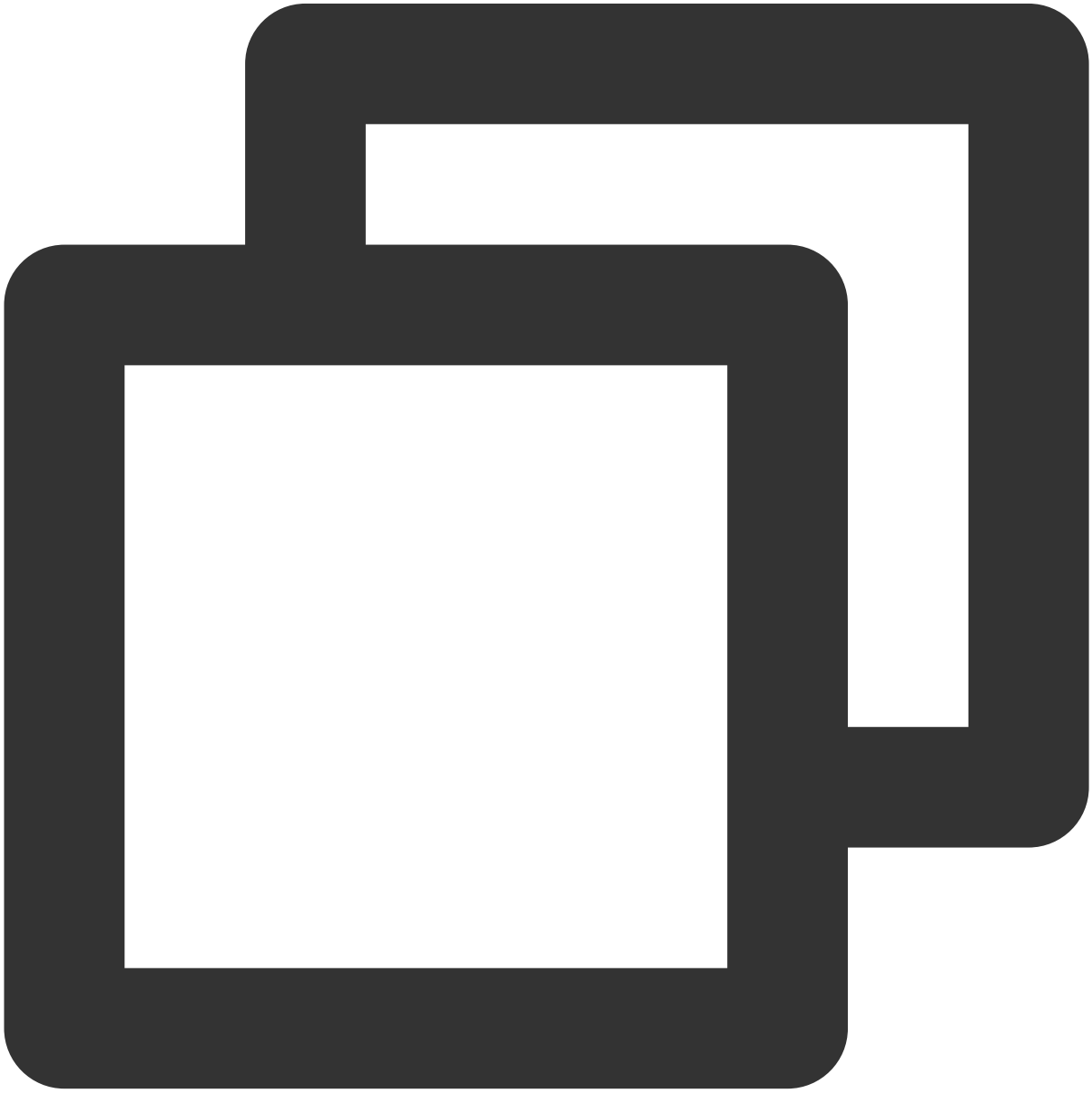


```
POST /snapshot HTTP/1.1
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-si
Content-Length: 552
Content-Type: application/xml

<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>
```

获取媒体信息
获取媒体文件信息

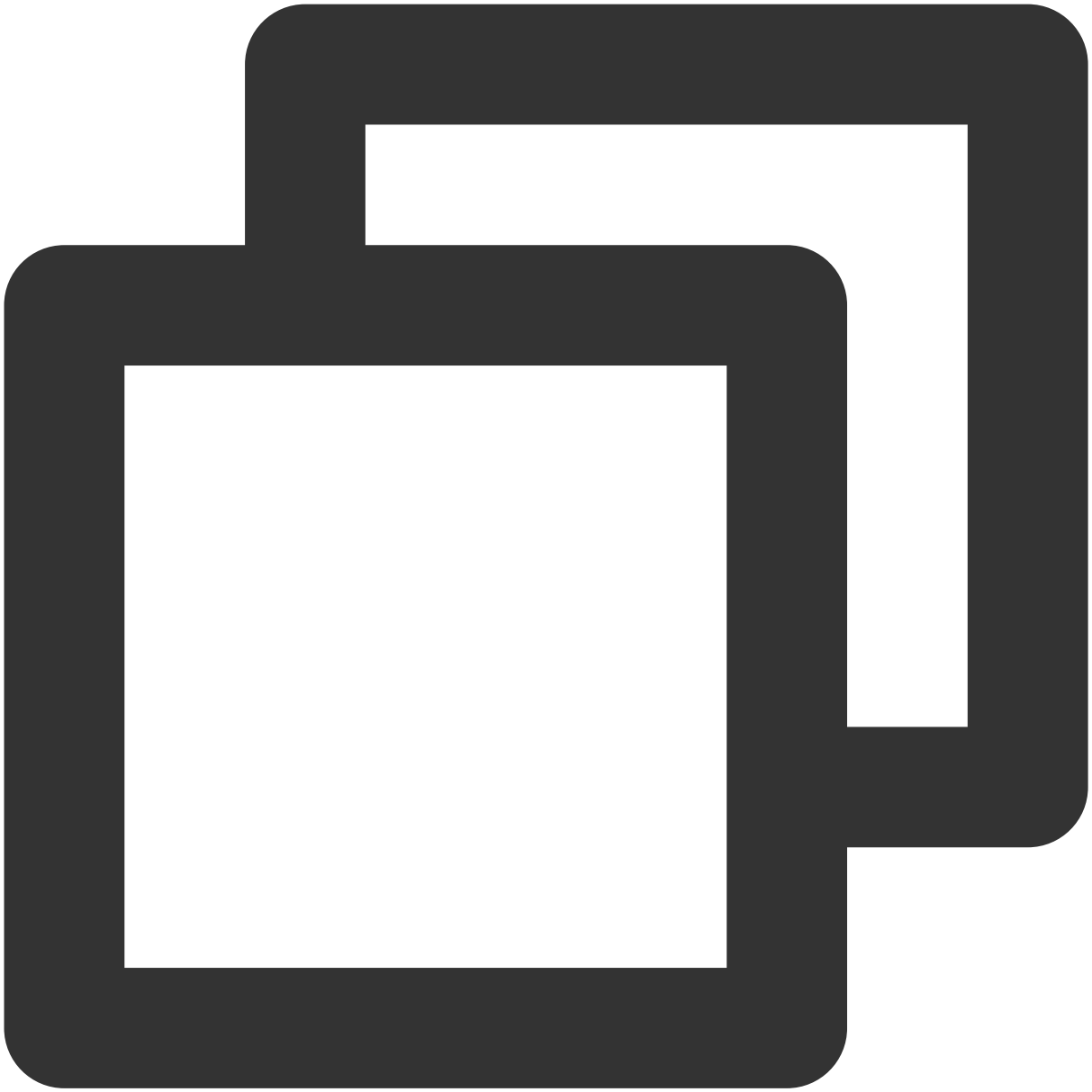
最近更新时间：2024-01-20 10:09:09

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 /<ObjectKey>?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）。

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。

<ObjectKey>为存储桶对象。

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。

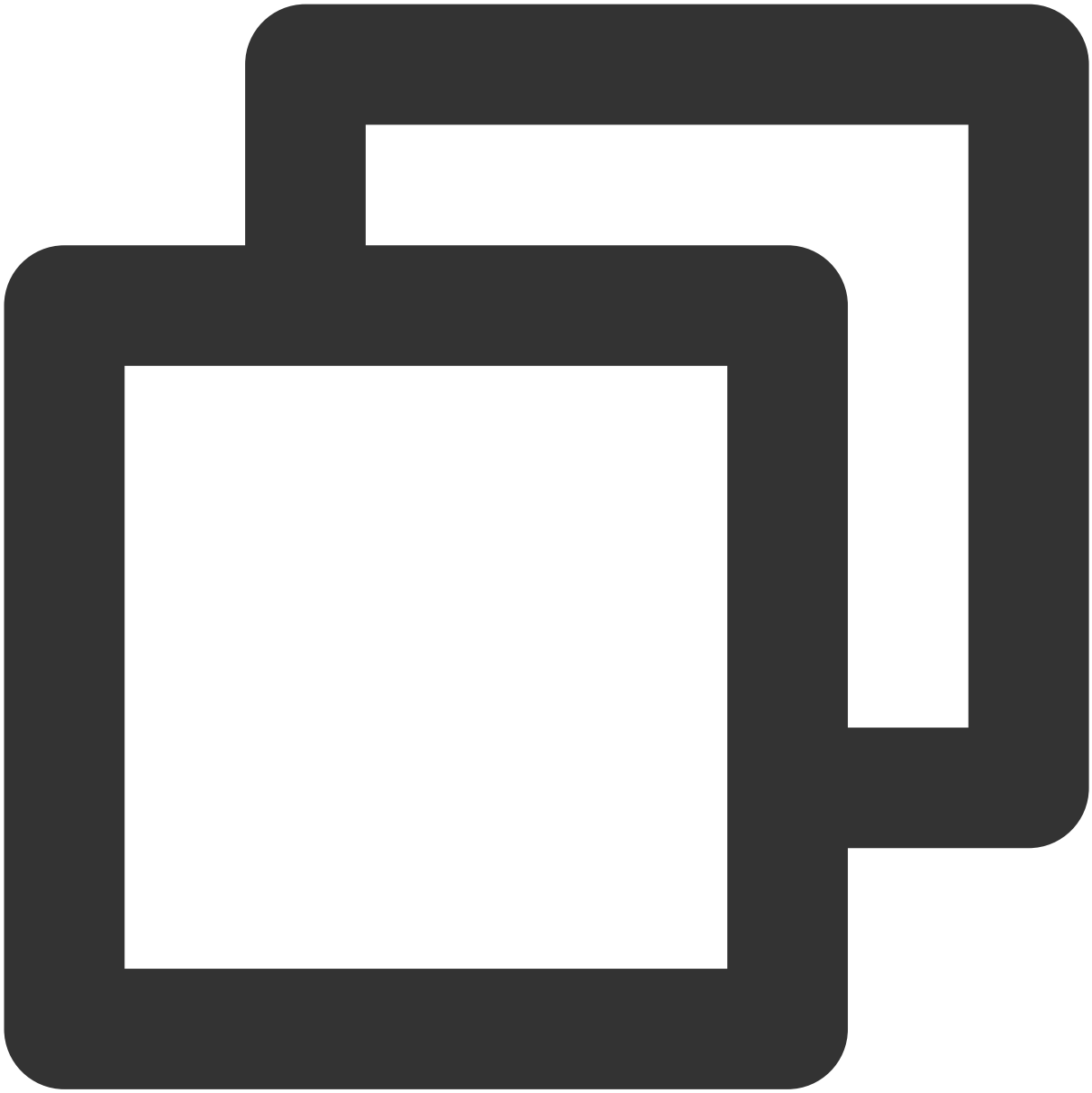
响应

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

版权所有：腾讯云计算（北京）有限责任公司

Sar	Response.MediaInfo.Stream.Video	采样宽高比	String
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

版权所有：腾讯云计算（北京）有限责任公司

Channel	Response.MediaInfo.Stream.Audio	通道数量	Int
ChannelLayout	Response.MediaInfo.Stream.Audio	通道格式	String
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：

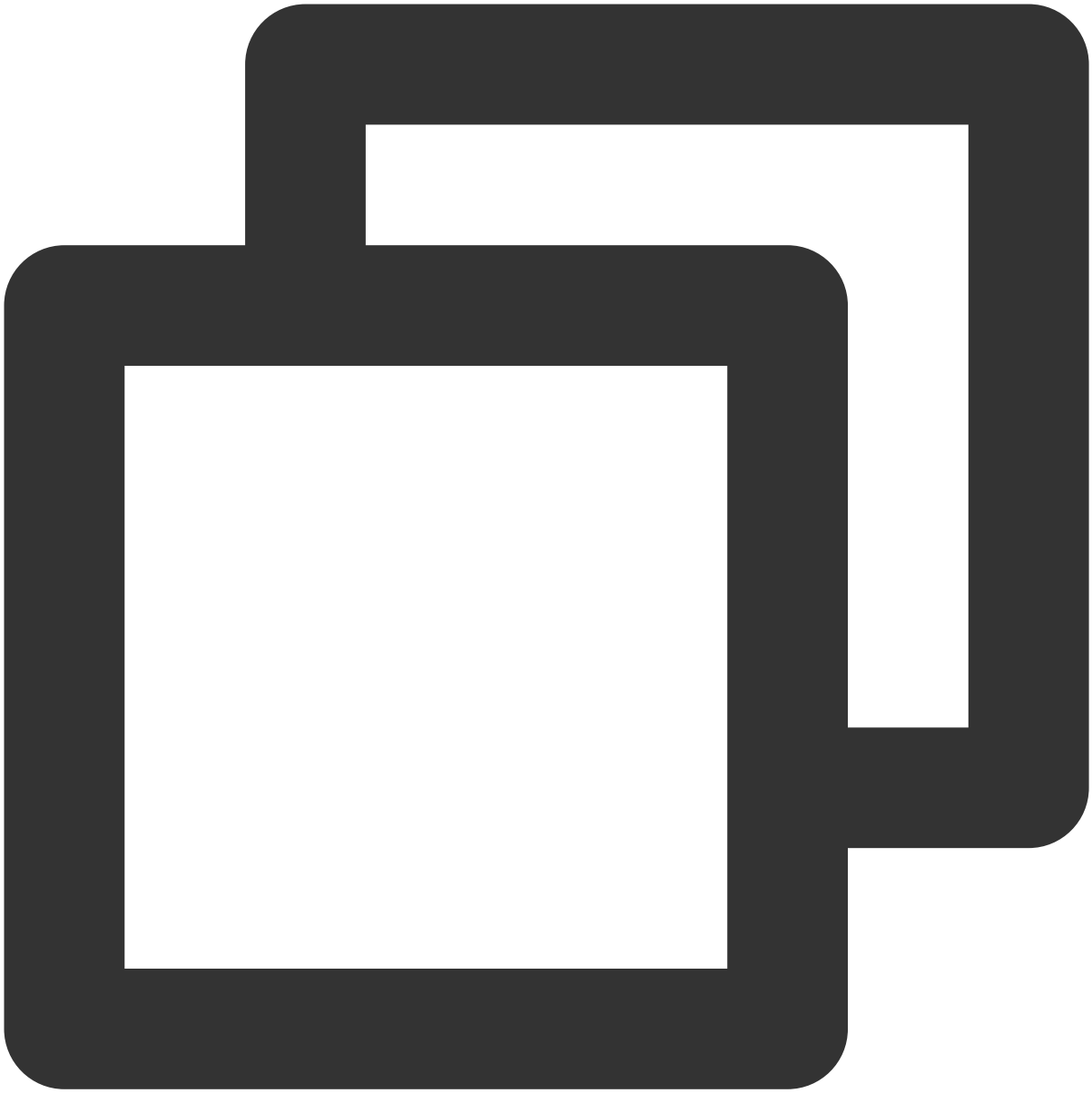
节点名称（关键字）	父节点	描述	类型
Index	Response.MediaInfo.Stream.Subtitle	该流的编号	Int
Language	Response.MediaInfo.Stream.Subtitle	语言，und 表示无查询结果	String

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。

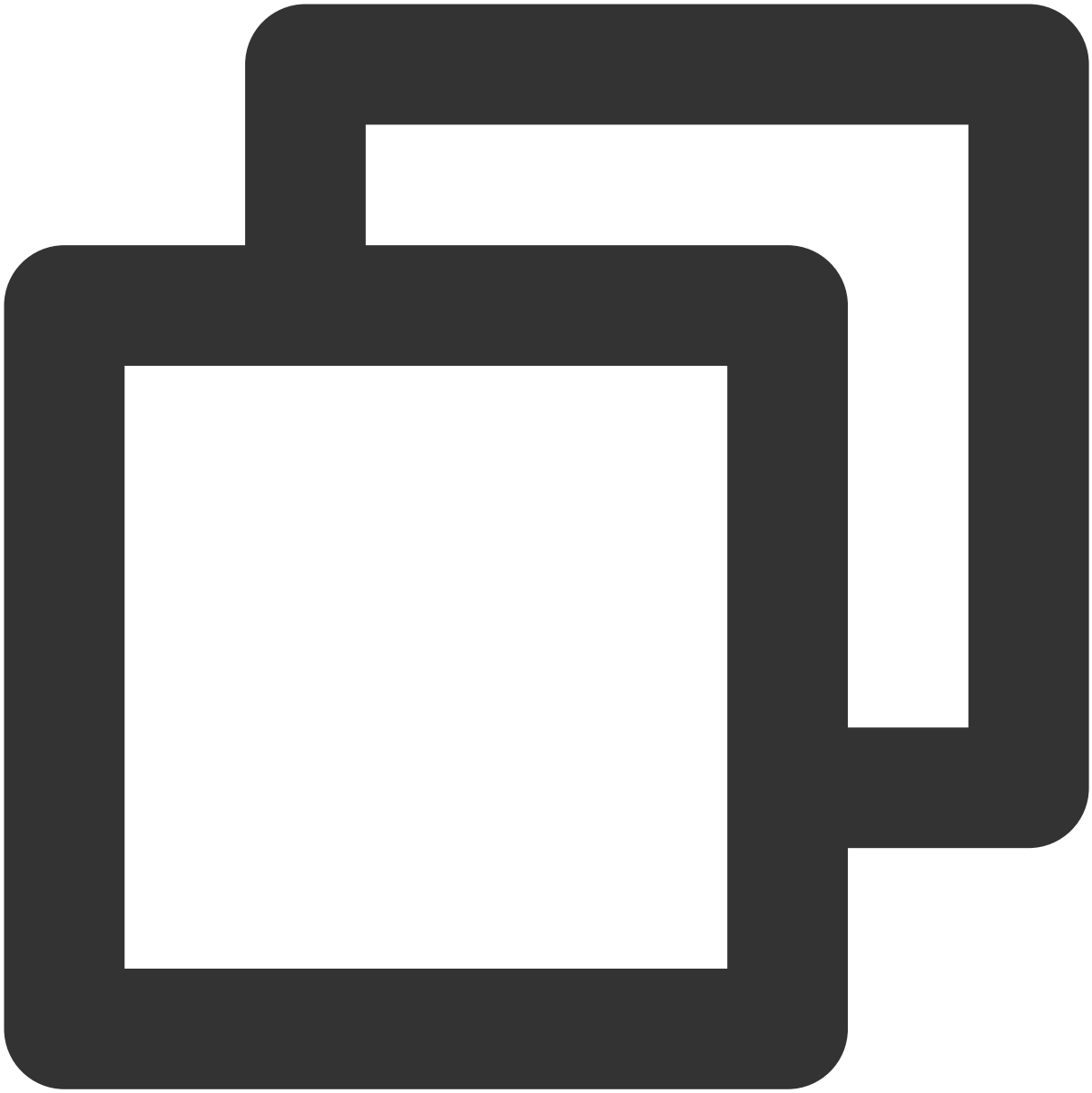
实际案例

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-ci
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>
```

读取私有 M3U8
获取私有 m3u8

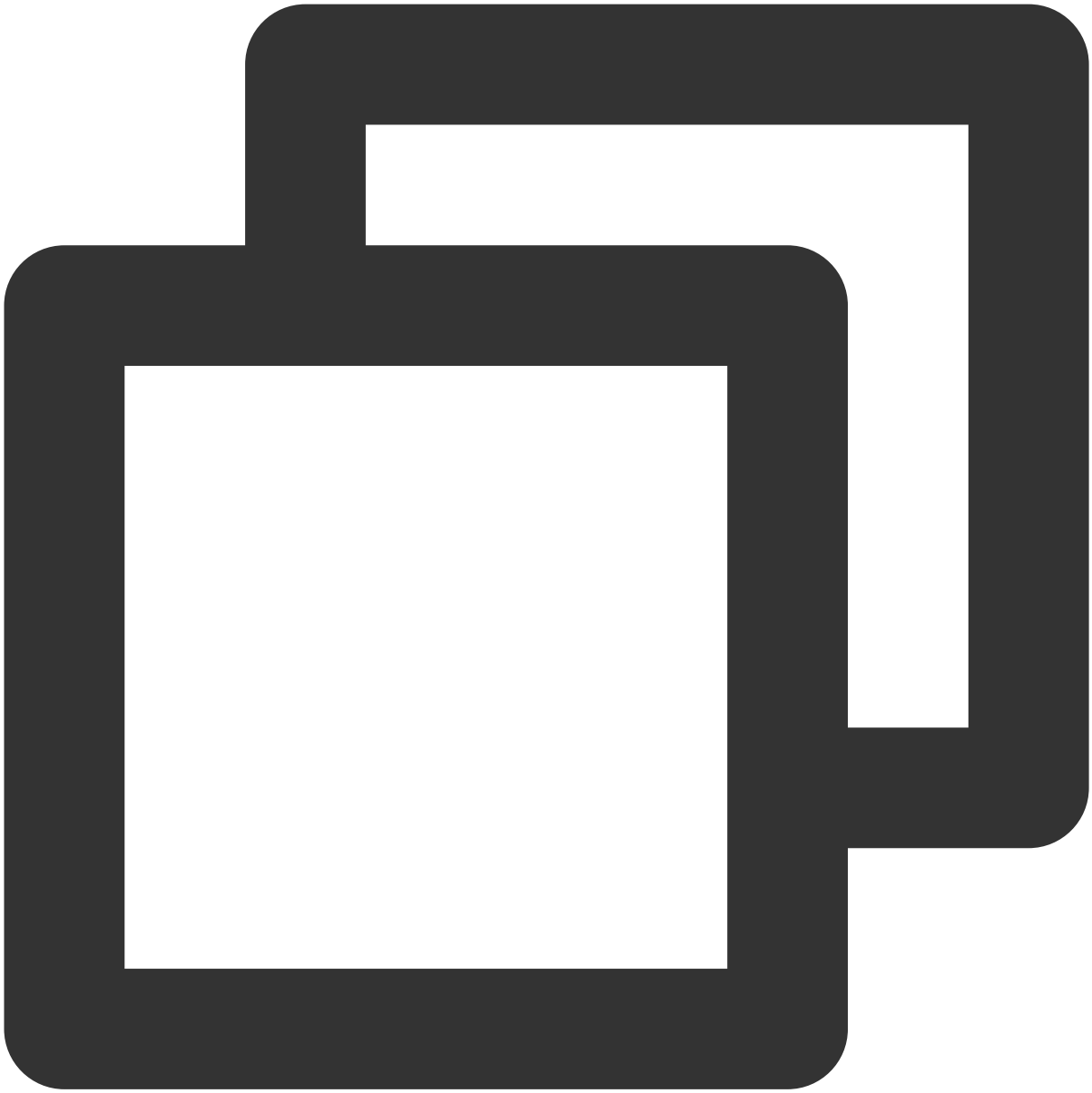
最近更新时间：2024-01-20 10:09:09

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 /<ObjectKey>?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）。

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。

<ObjectKey>为存储桶对象。

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

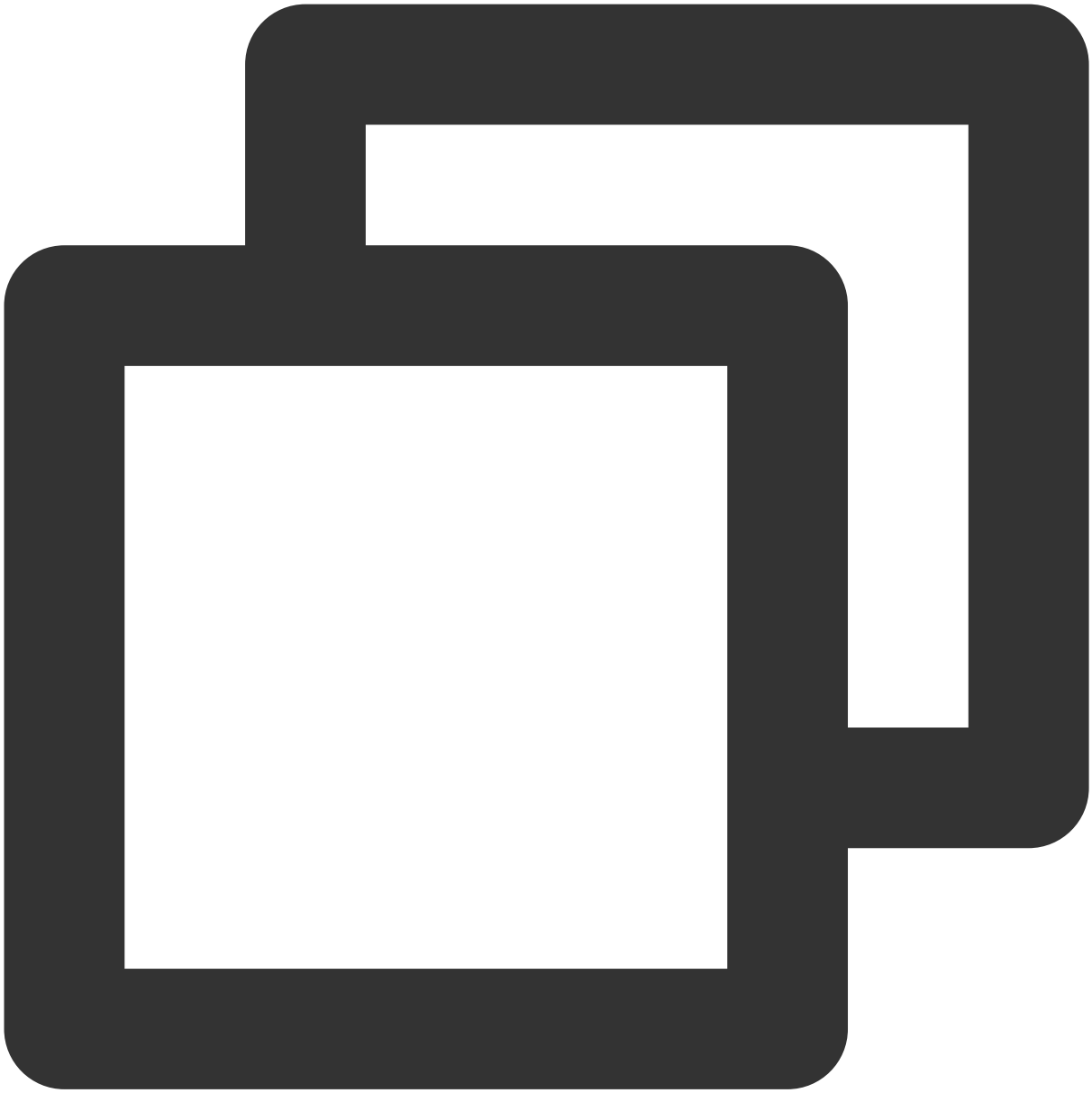
该响应体为截图文件内容。

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。

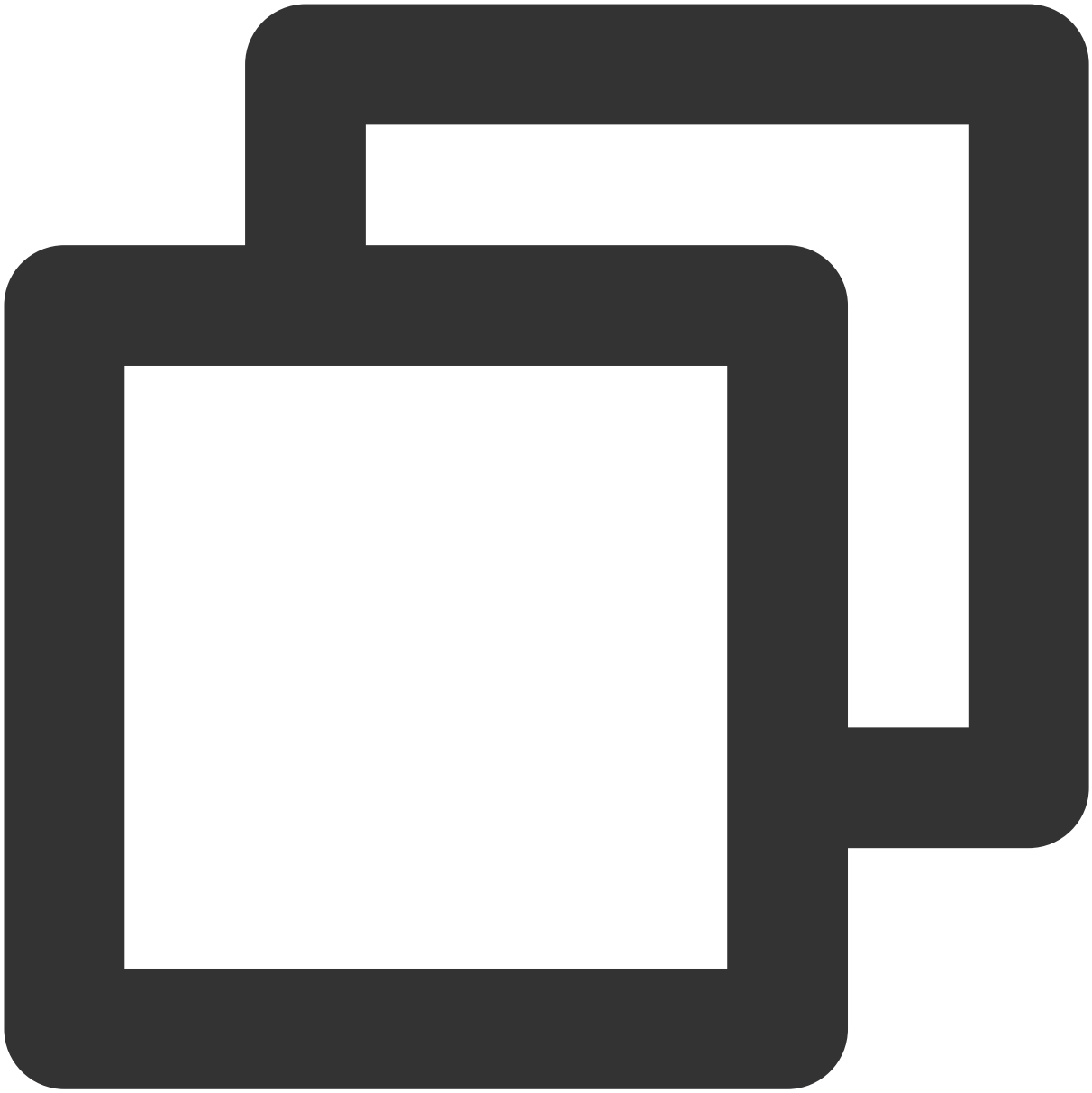
实际案例

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-ci
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZm****

<M3U8文件内容>
```

内容审核
音频审核
提交音频审核任务

最近更新时间：2024-01-20 11:24:09

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。

授权说明

授权策略中 action 设置为 `ci:CreateAuditingAudioJob` 。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。
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。
接口默认并发数量：10。
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。

费用说明

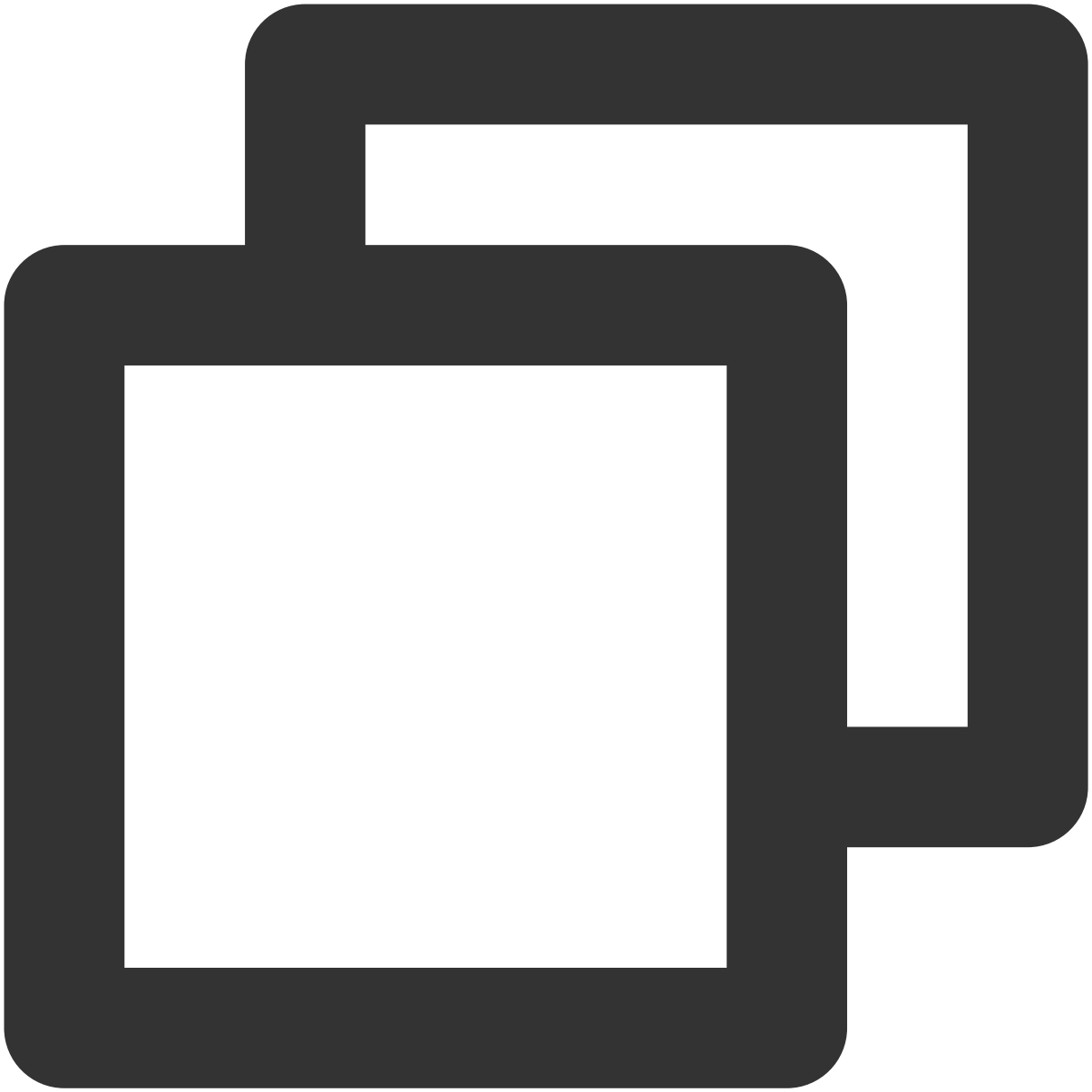
每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1个音频**，**计2次**审核费用。
调用接口会产生 音频审核费用 和 COS 读请求费用。
如果音频属于 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用。
不支持审核 COS 上的归档存储类型和深度归档存储类型的音频，如果需要审核此类型音频，请先 恢复归档文件。

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



```
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）。

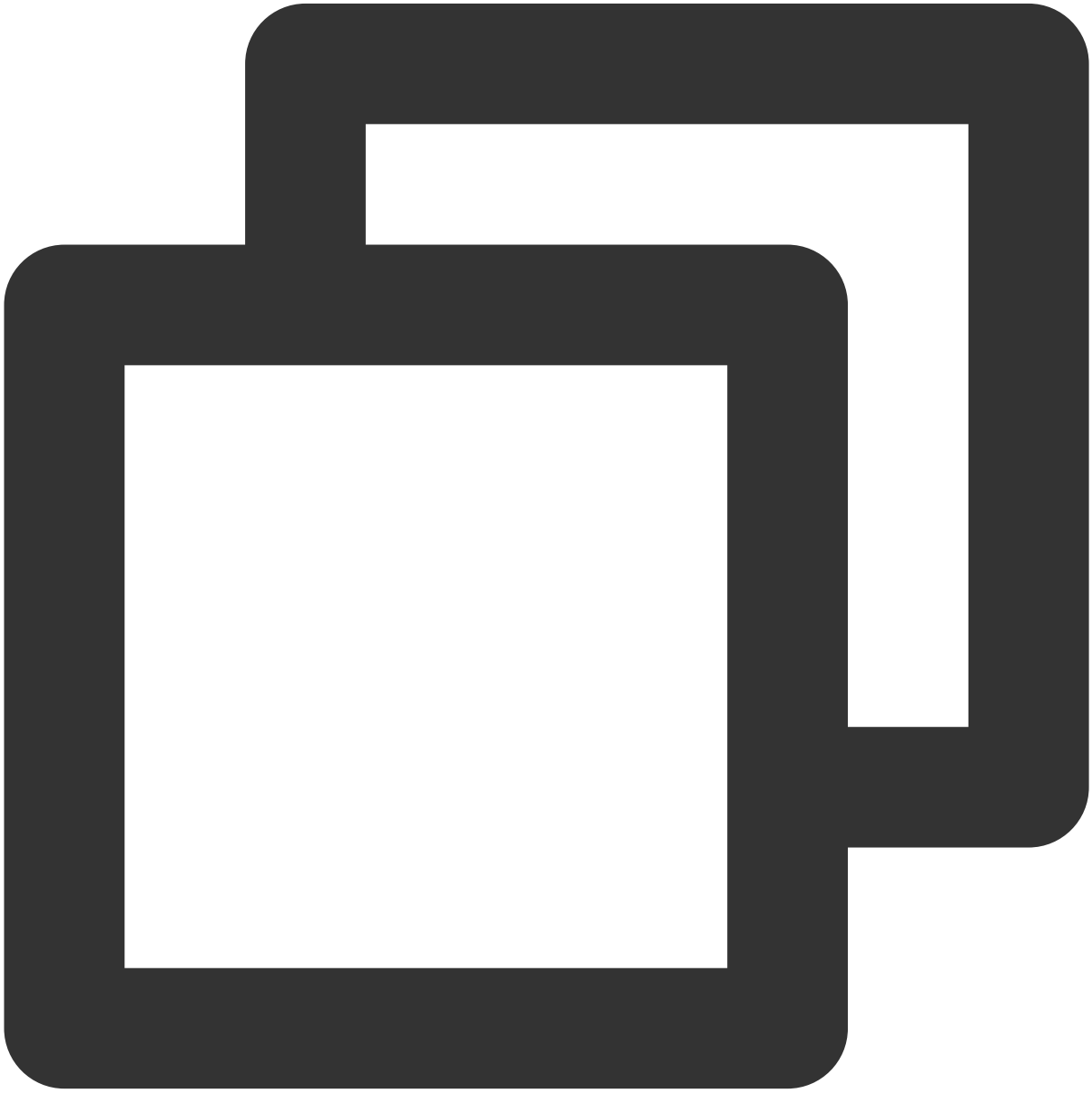
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

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）	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	String	否

		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。		
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。

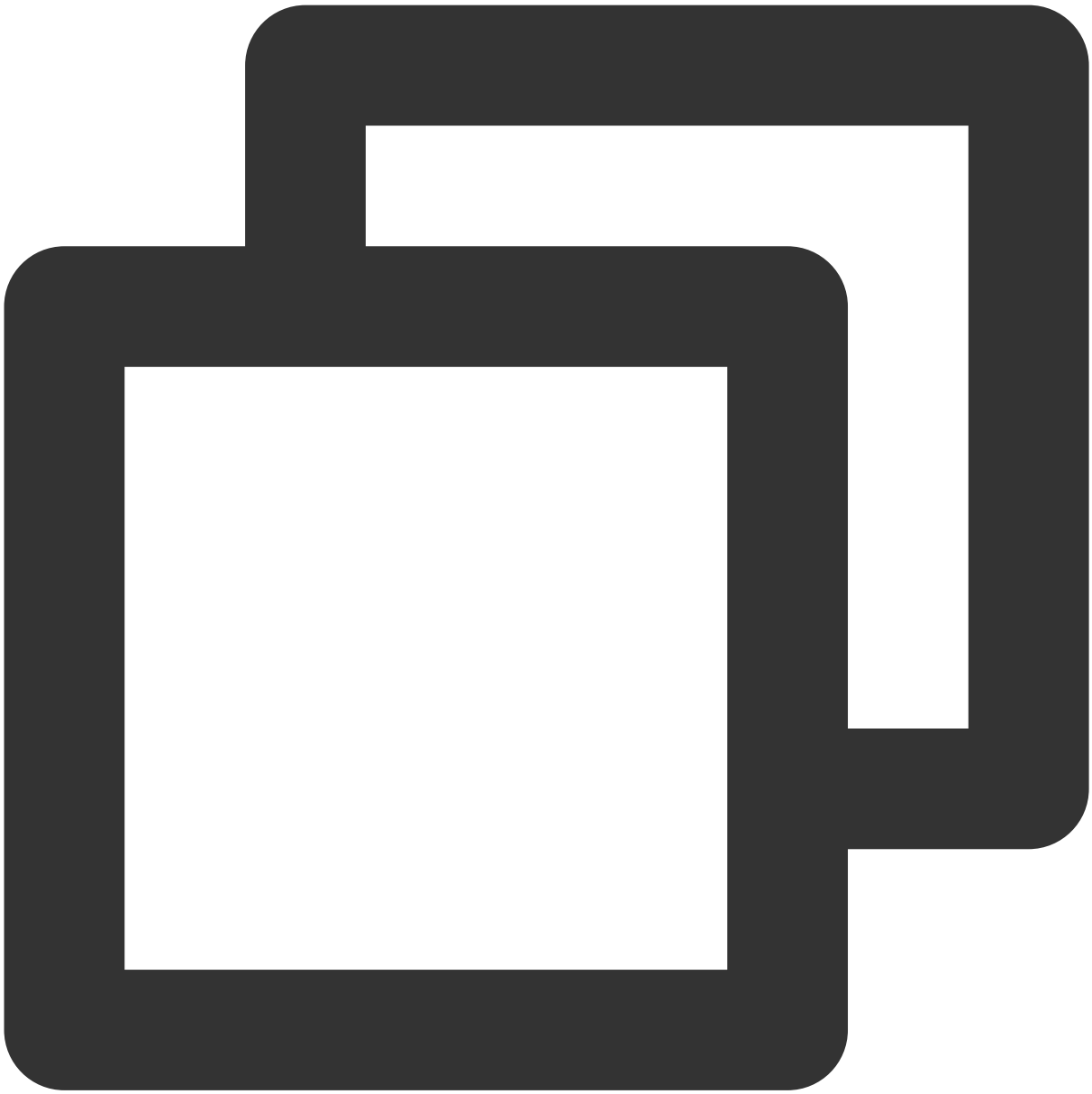
响应

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：

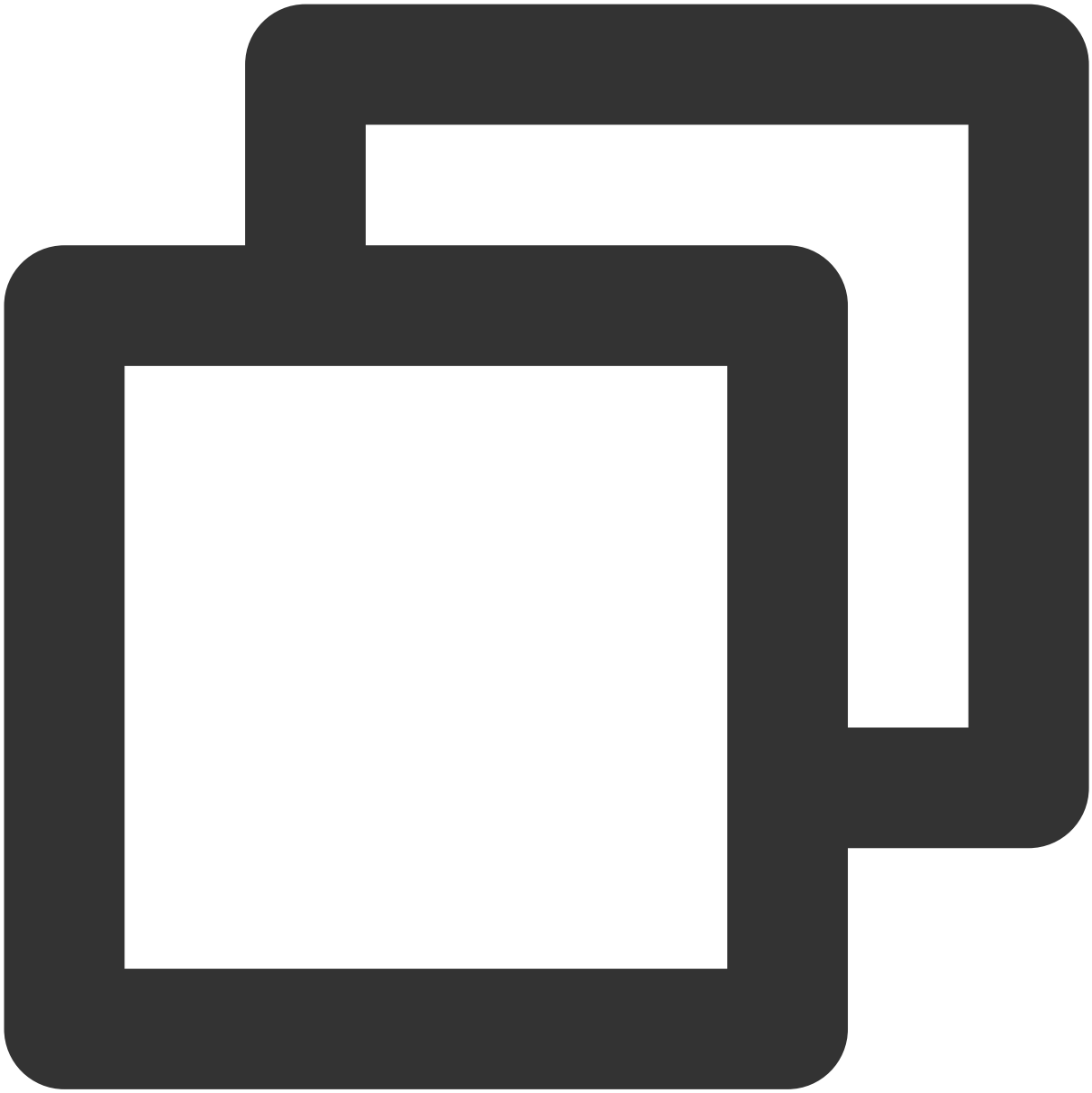
节点名称（关键字）	父节点	描述	类型
DataId	Response.JobsDetail	请求中添加的唯一业务标识。	String
JobId	Response.JobsDetail	本次音频审核任务的 ID。	String
State	Response.JobsDetail	音频审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	音频审核任务的创建时间。	String

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

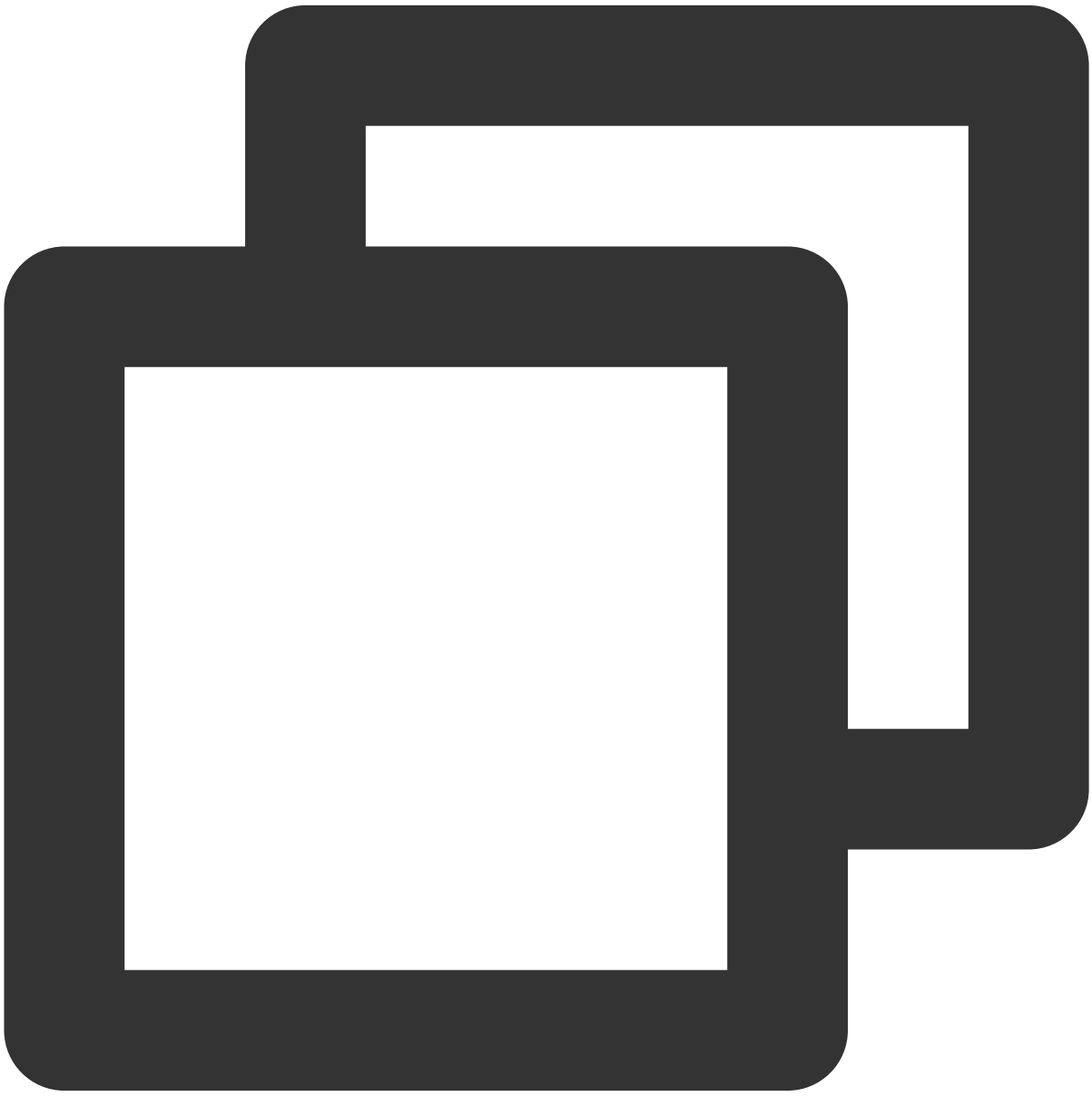


```
POST /audio/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Input>
        <Object>a.mp3</Object>
        <DataId>123-fdrsg-123</DataID>
    </Input>
```

```
    <Conf>
        <DetectType>Porn,Ads</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>
        <DataId>123-fdrsg-123</DataID>
        <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
        <State>Submitted</State>
        <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    </JobsDetail>
    <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

文本审核
文本内容同步审核

最近更新时间：2024-01-20 11:24:09

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。
接口默认并发数量：100。

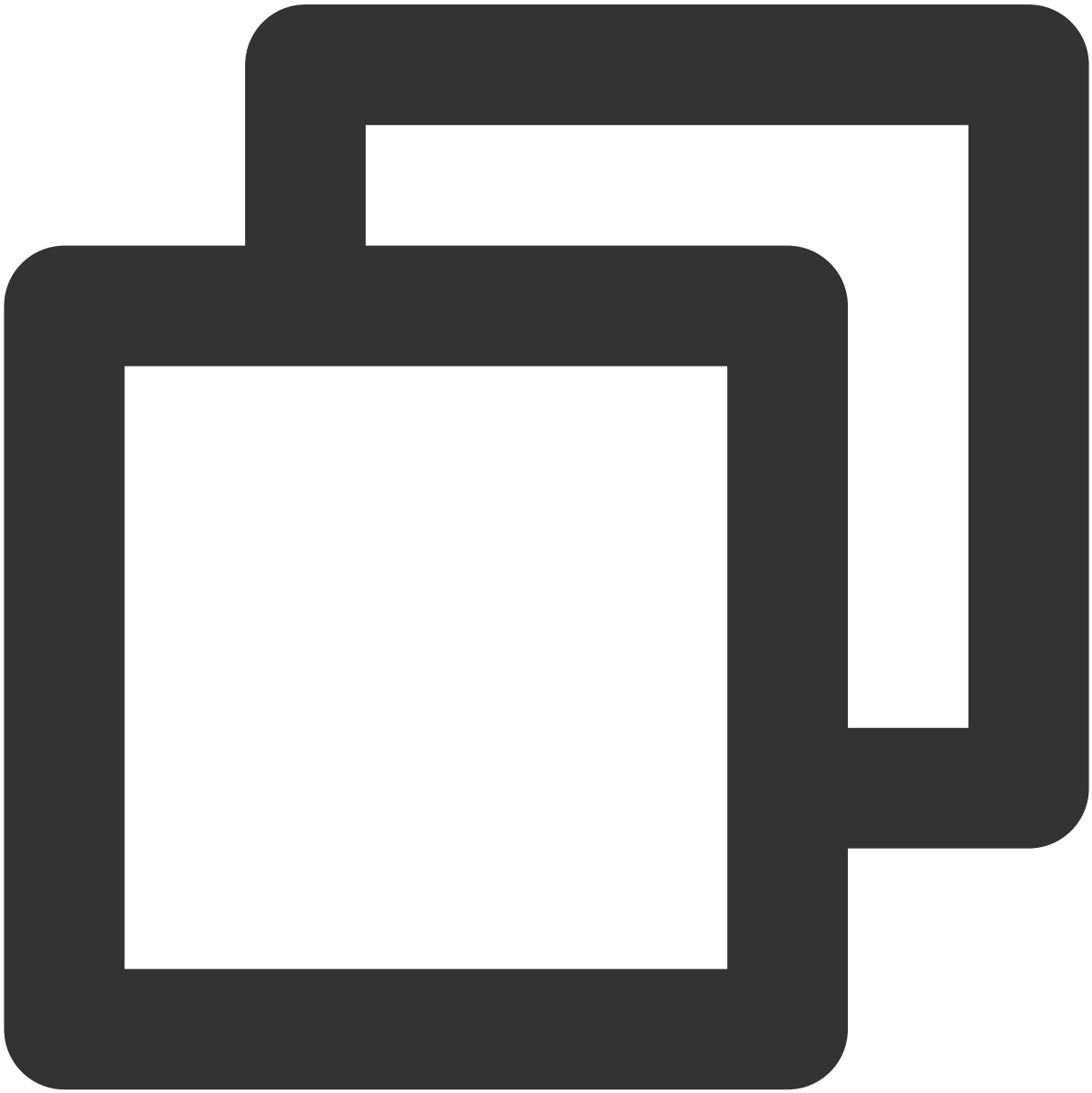
版权所有：腾讯云计算（北京）有限责任公司

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



```
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）。

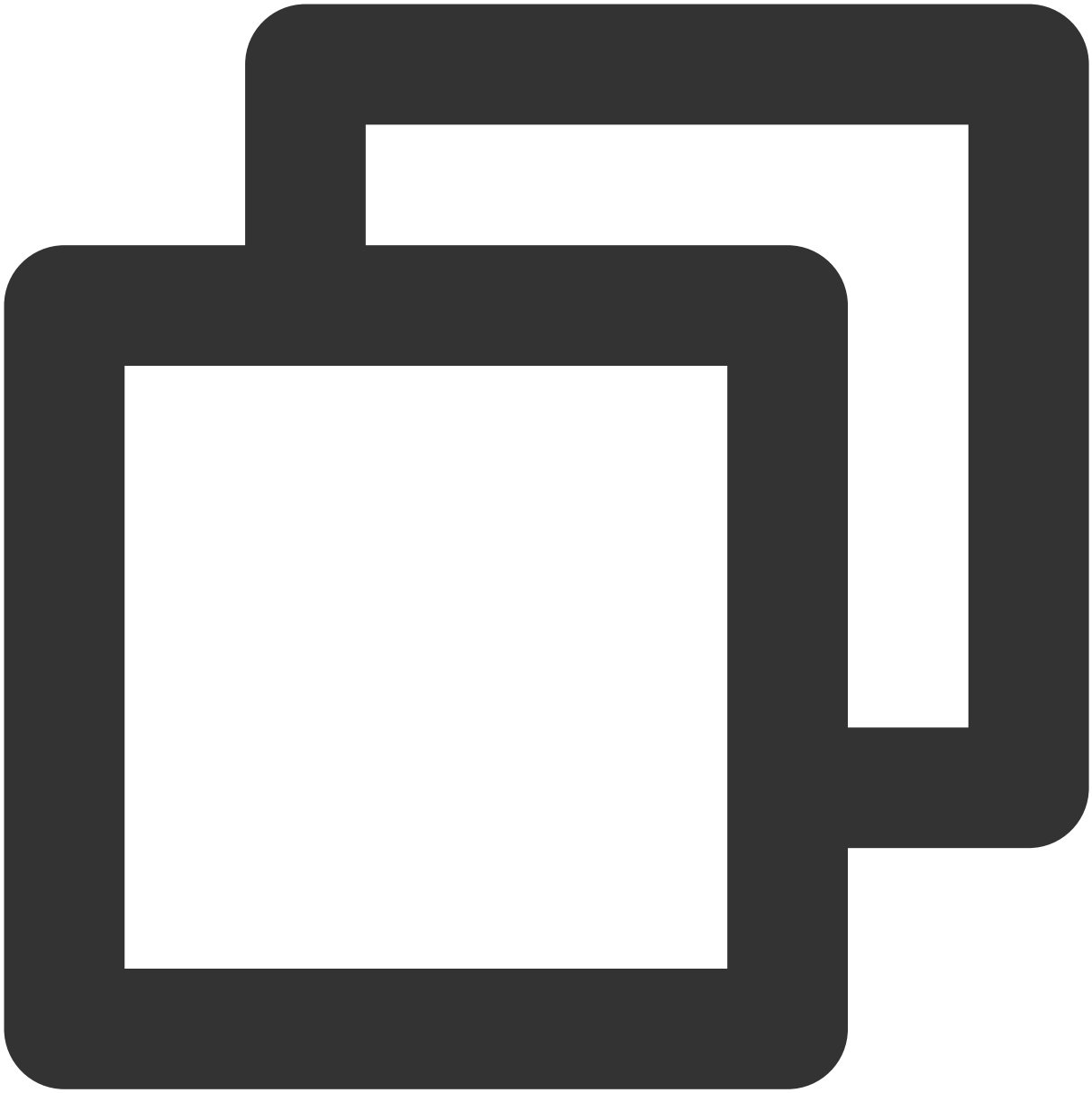
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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。 注意：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

版权所有：腾讯云计算（北京）有限责任公司

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	Container	否

版权所有：腾讯云计算（北京）有限责任公司

		`input` 中审核的文本为 `object` 时有效。		

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。

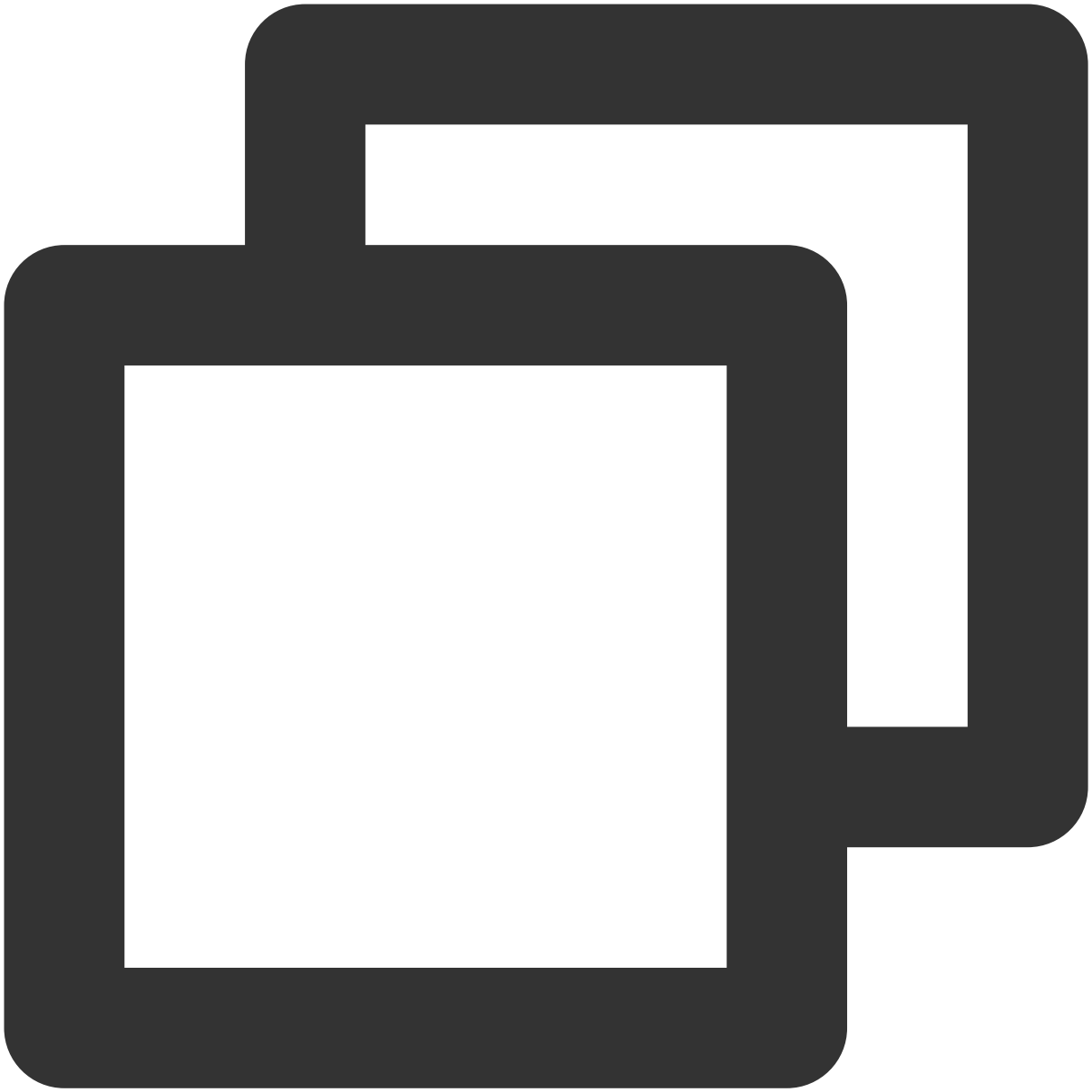
响应

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>
    <ContextText></ContextText>
```

```
    <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 Failed（审核失败）。	String
Content	Response.JobsDetail	本次提交审核的文本内容，以 Base64 编码形式返回。	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	父节点	描述	类型

（关键字）			
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	命中该审核分类的分片数。	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.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
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）	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：

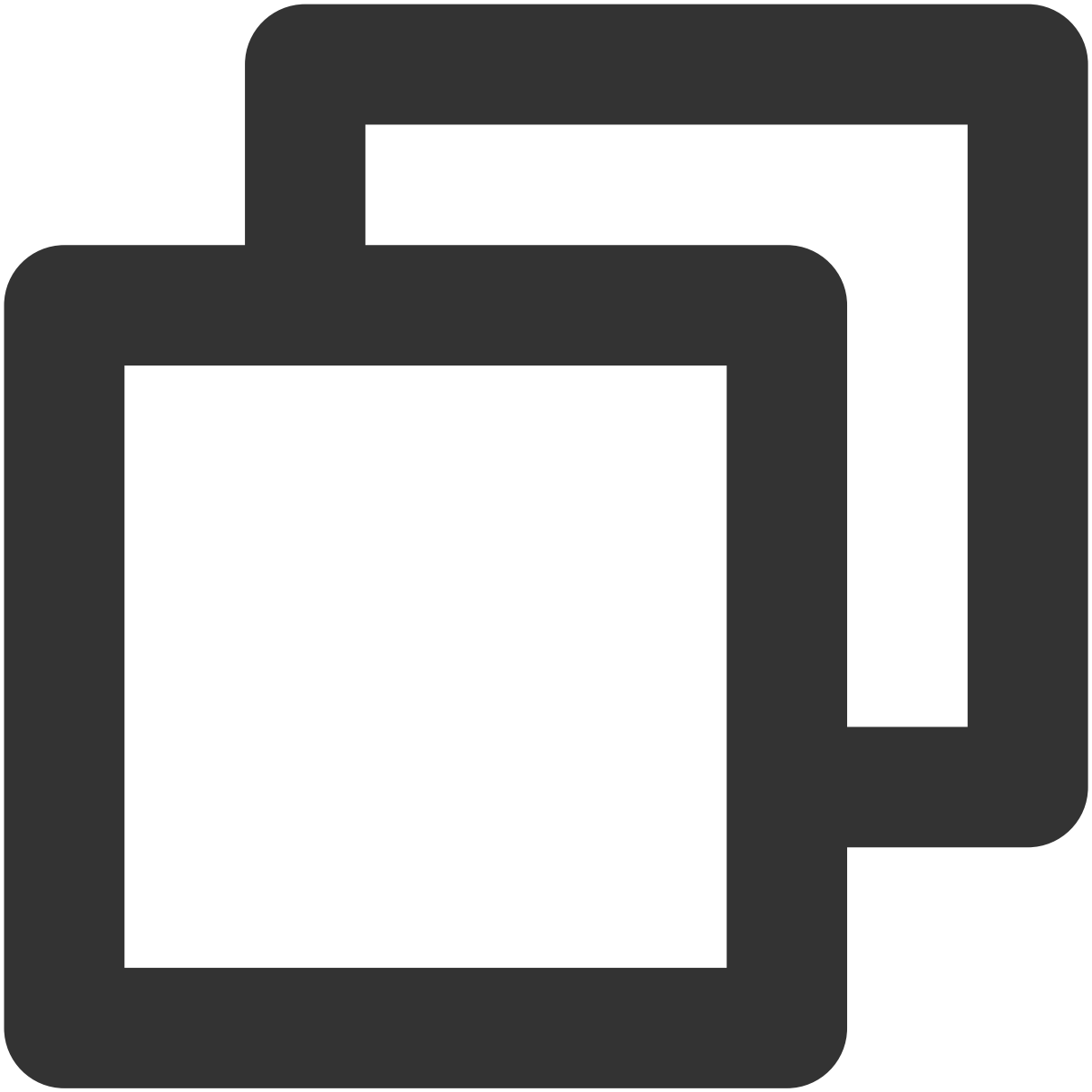
节点名称（关键字）	父节点	描述	类型
ListType	Response.JobsDetail.ListInfo.ListResults	命中的名单类型，取值为0（白名单）和1（黑名单）。	Integer
ListName	Response.JobsDetail.ListInfo.ListResults	命中的名单名称。	String
Entity	Response.JobsDetail.ListInfo.ListResults	命中了名单中的哪条内容。	String

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

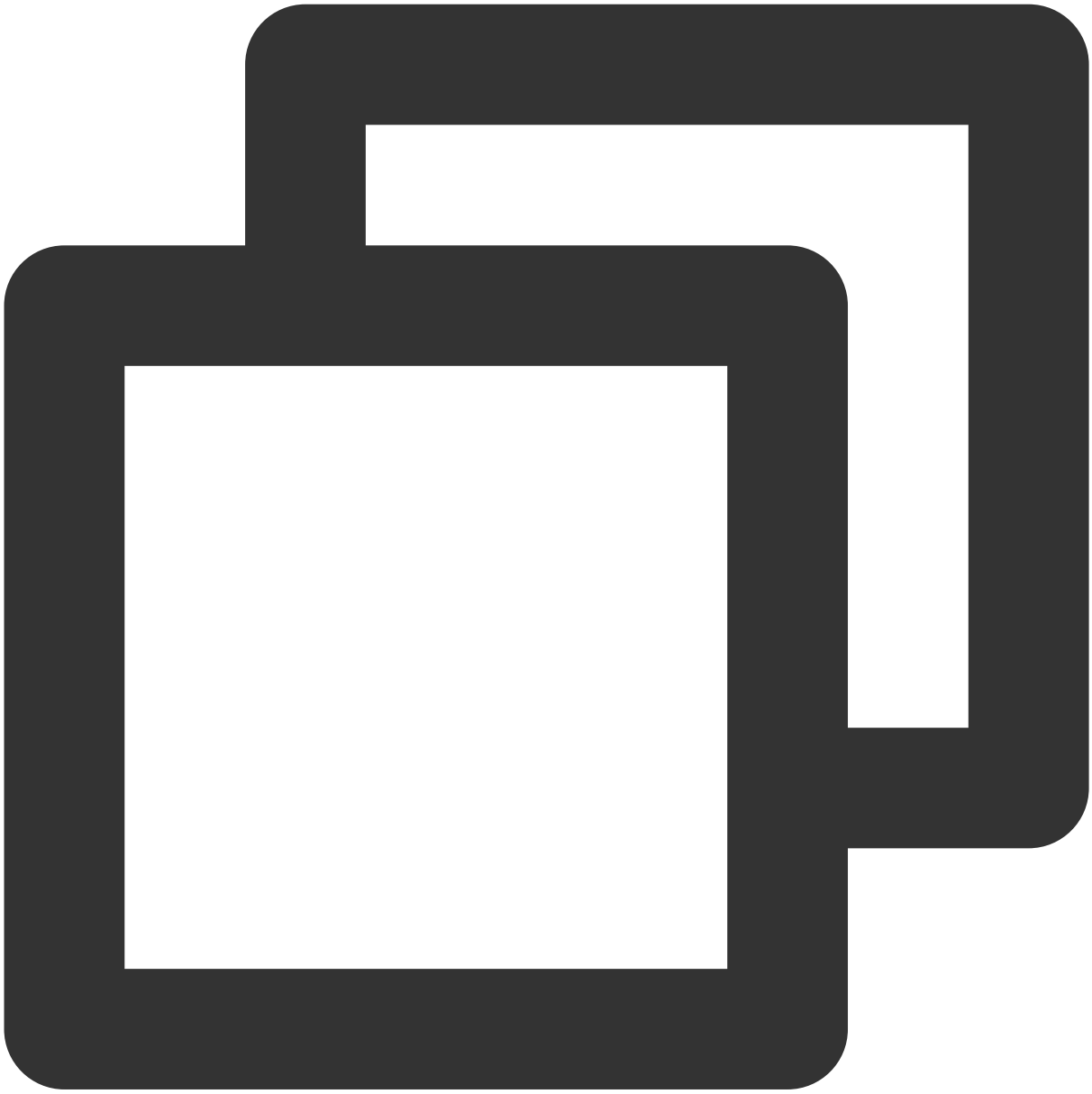
请求



```
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>
```

文档预览
查询文档预览开通状态

最近更新时间：2024-01-20 11:39:30

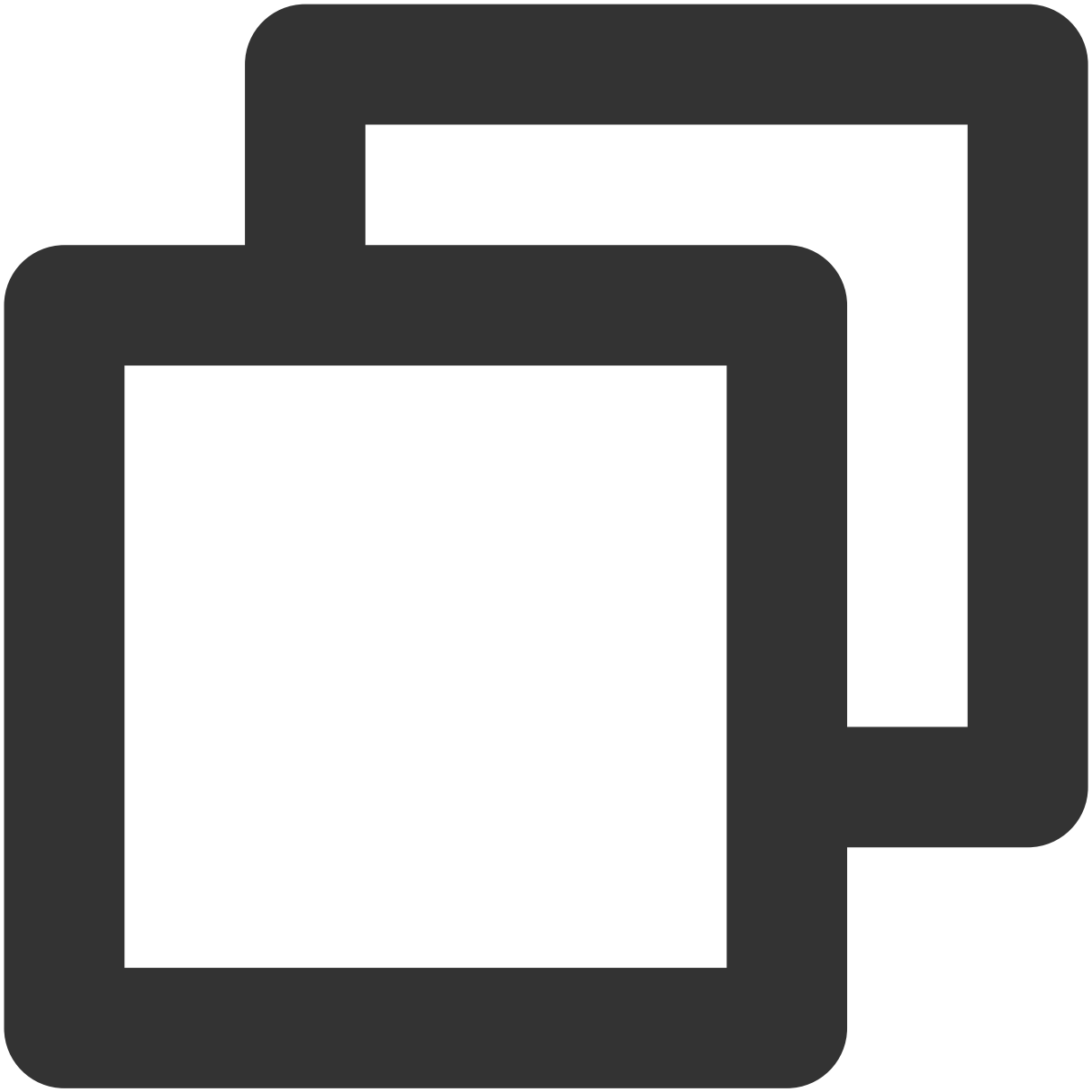
功能描述

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

版权所有：腾讯云计算（北京）有限责任公司



```
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

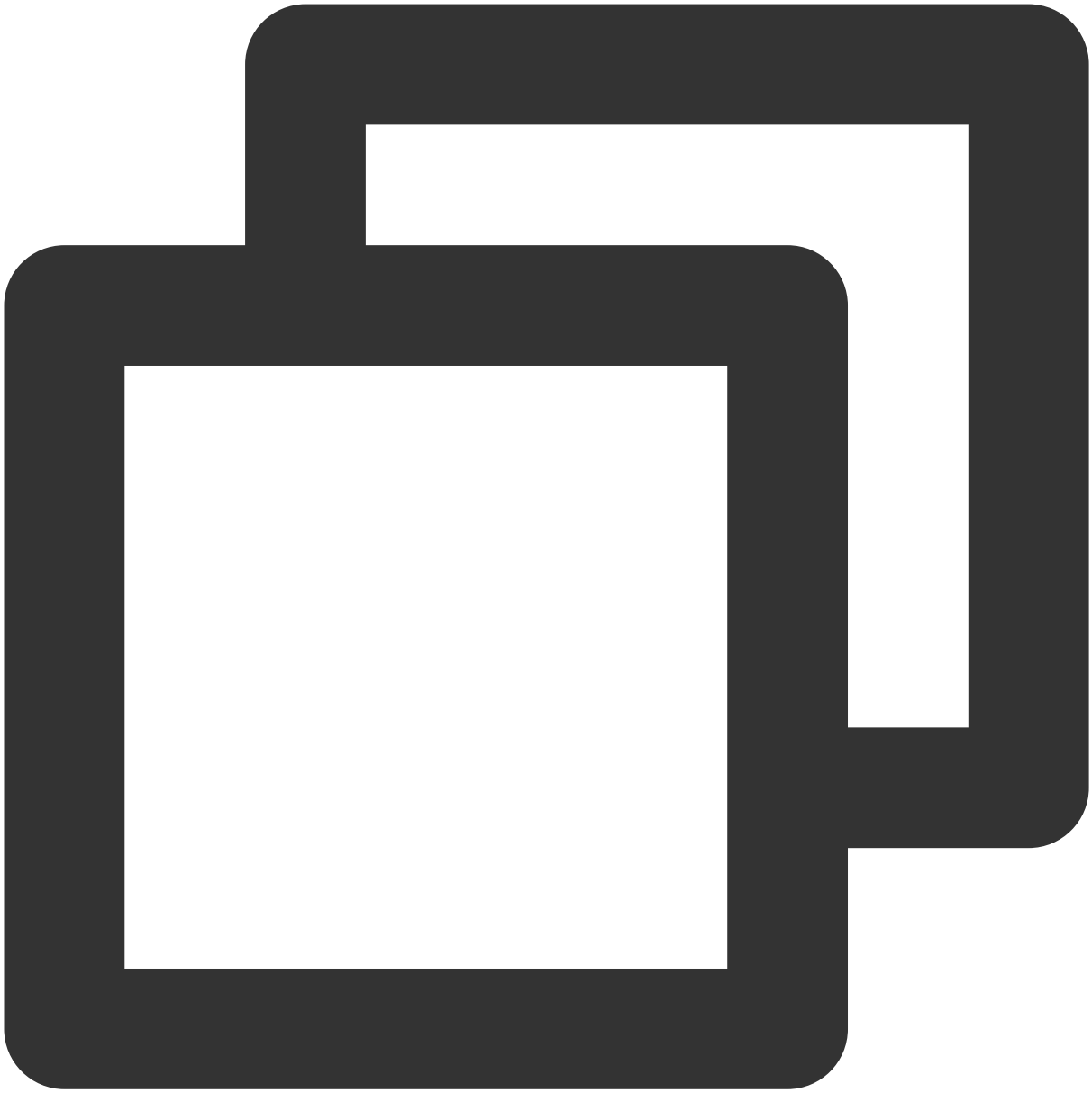
响应

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </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：

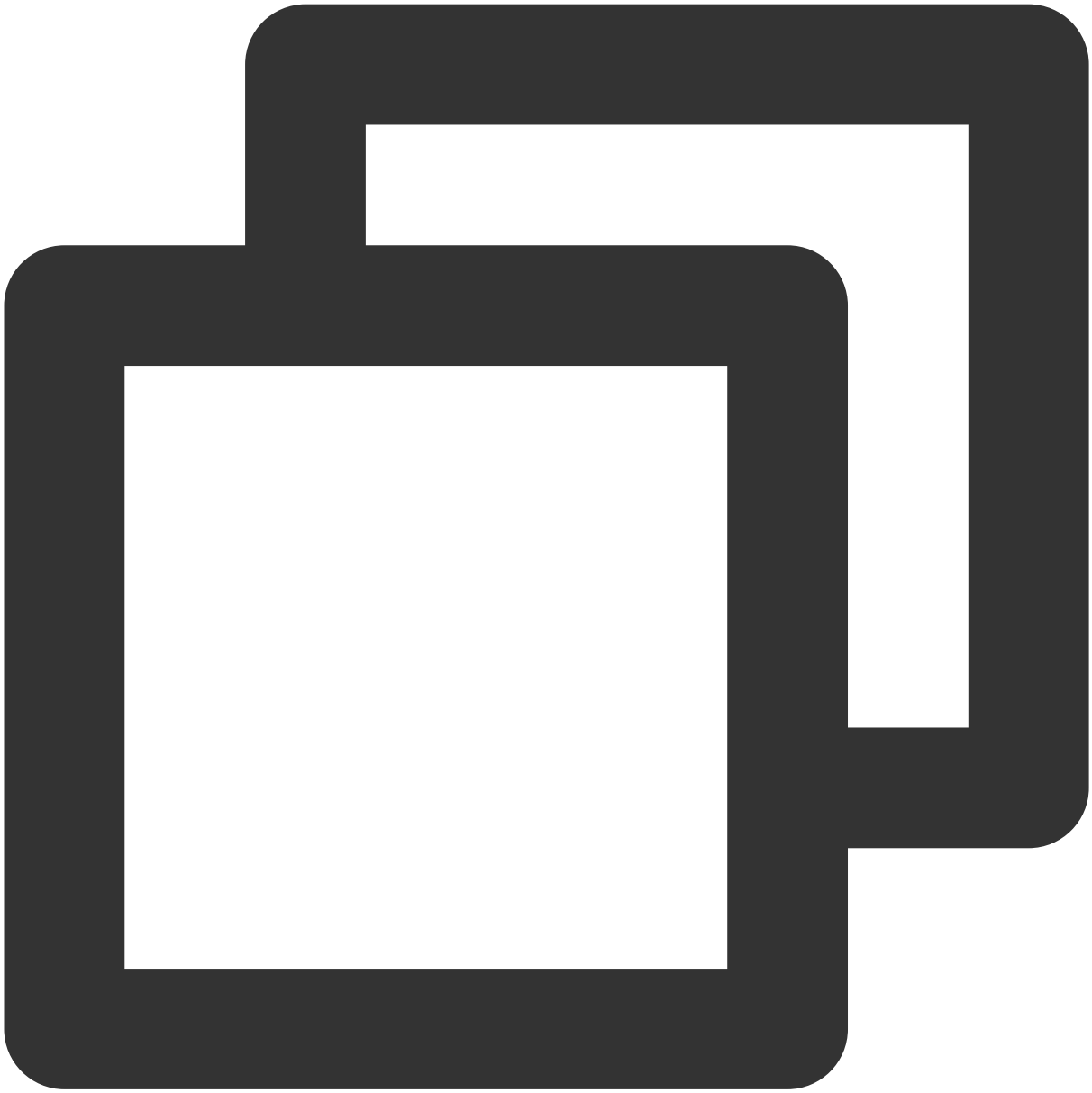
节点名称（关键字）	父节点	描述	类型
BucketId	Response.DocBucketList	存储桶 ID	String
Name	Response.DocBucketList	存储桶名称，同 BucketId	String
Region	Response.DocBucketList	所在的地域	String
CreateTime	Response.DocBucketList	创建时间	String
AliasBucketId	Response.DocBucketList	存储桶别名	String

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。

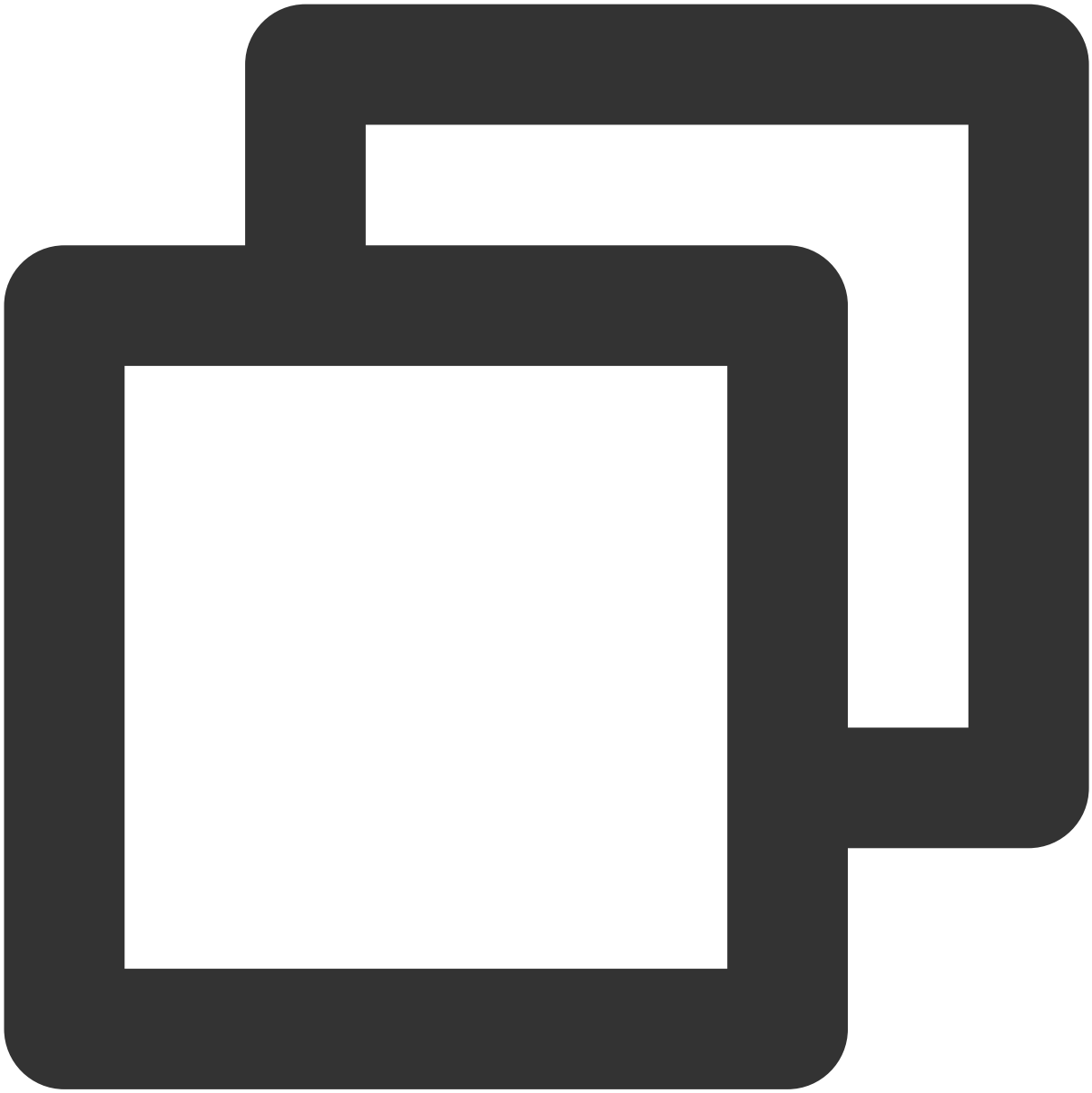
实际案例

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>
```

版权所有：腾讯云计算（北京）有限责任公司

文档转码
同步请求接口

最近更新时间：2024-01-20 11:39:30

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

使用时您需要在 数据万象控制台 相应的存储桶管理页面下，单击**文档处理**，找到**文档预览**配置项，并通过开启按钮开通服务。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。

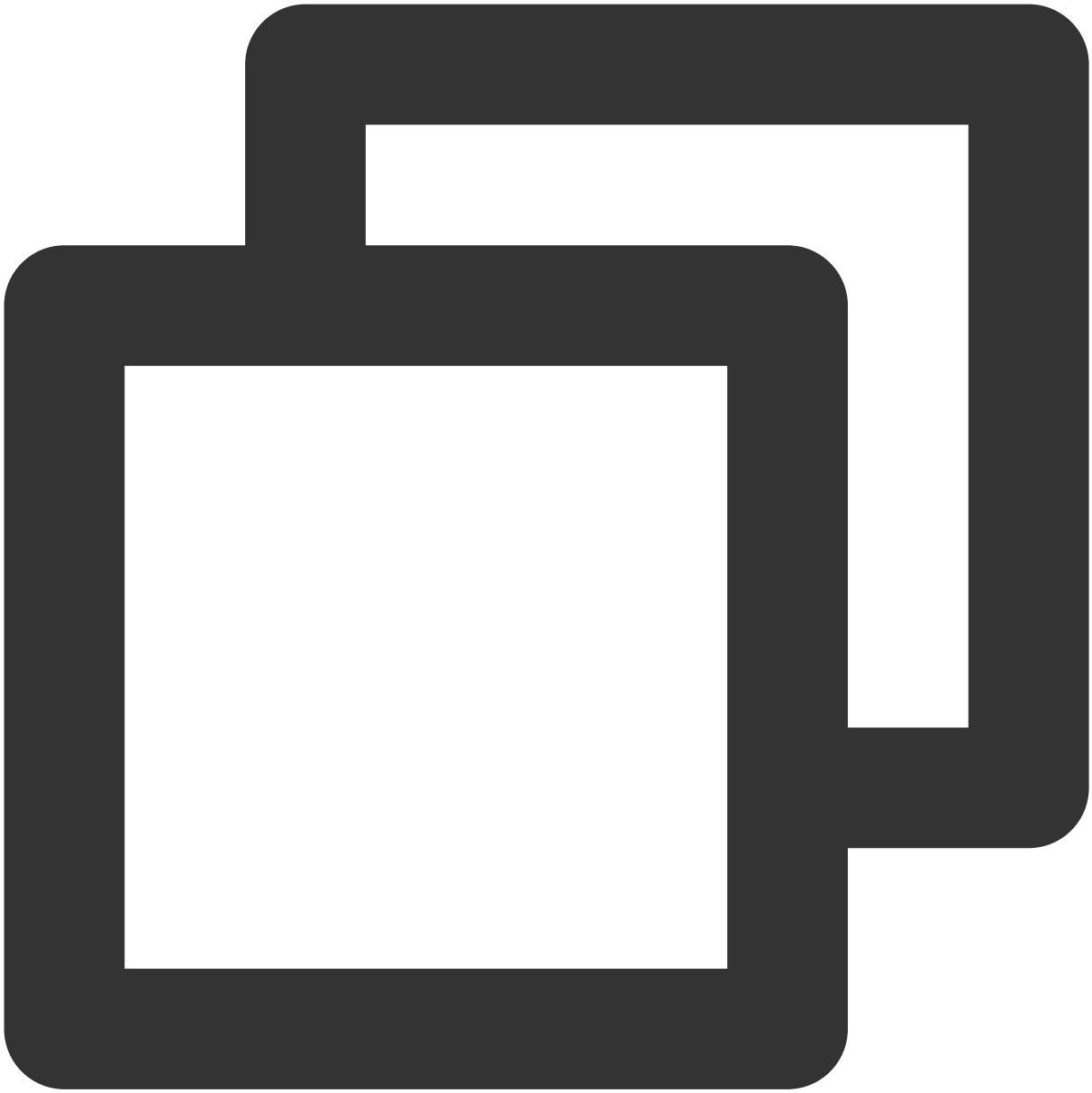
文档预览为付费功能，具体费用请参见 文档处理费用。数据万象在每个账户在首次产生**文档预览转图片**用量后，将发放一个用量为6000页，有效期为2个月的文档预览转图片免费额度资源包，超出用量或资源包到期后将正常计费。

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

版权所有：腾讯云计算（北京）有限责任公司



```
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）。

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。

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 格式文件。 无法选择页码，page 参数不生效 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	是否隐藏批注和应用修订，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
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

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。

同步接口适用于百页内的PDF文件或预览少数页的情况，若文件较大建议使用异步接口。

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

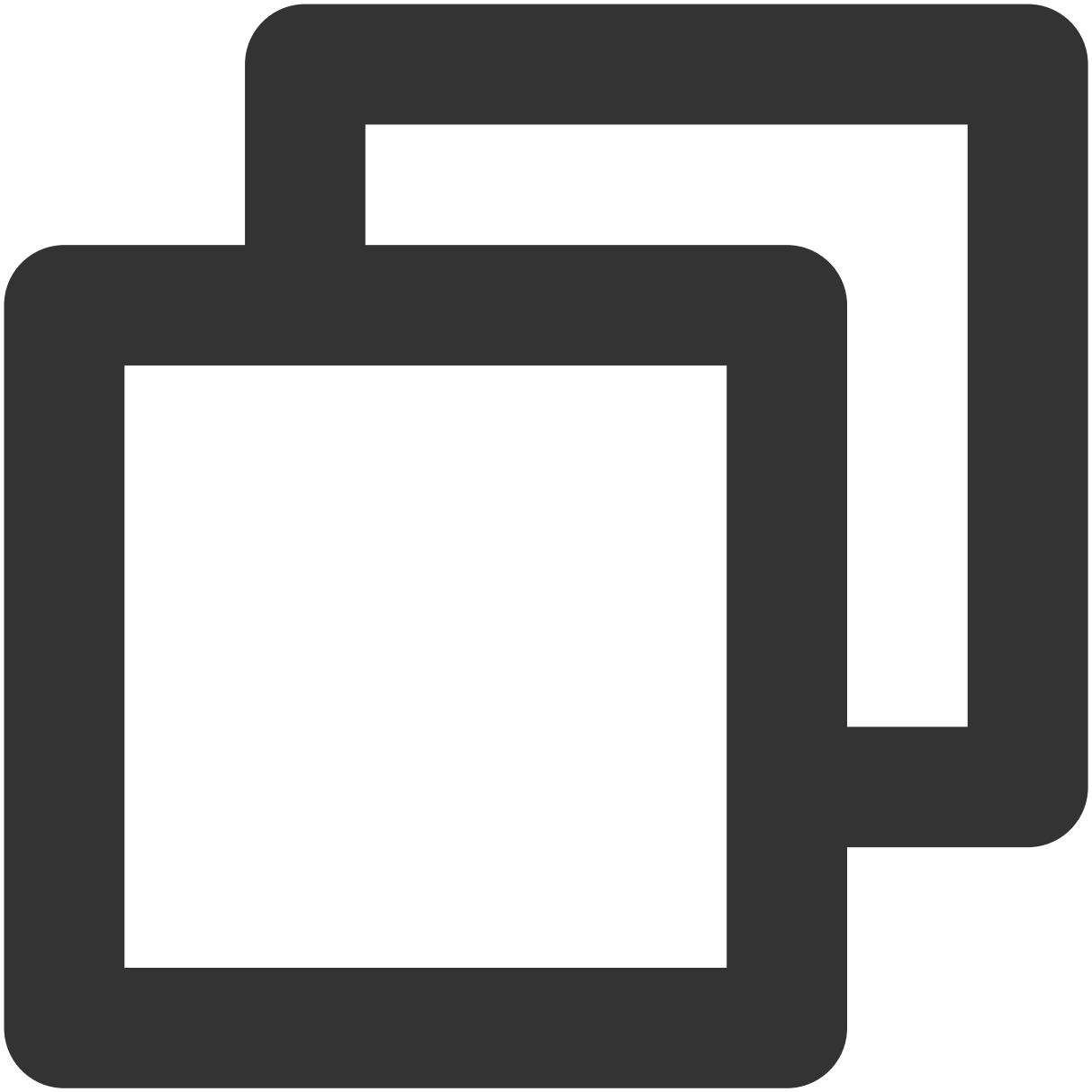
错误码

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

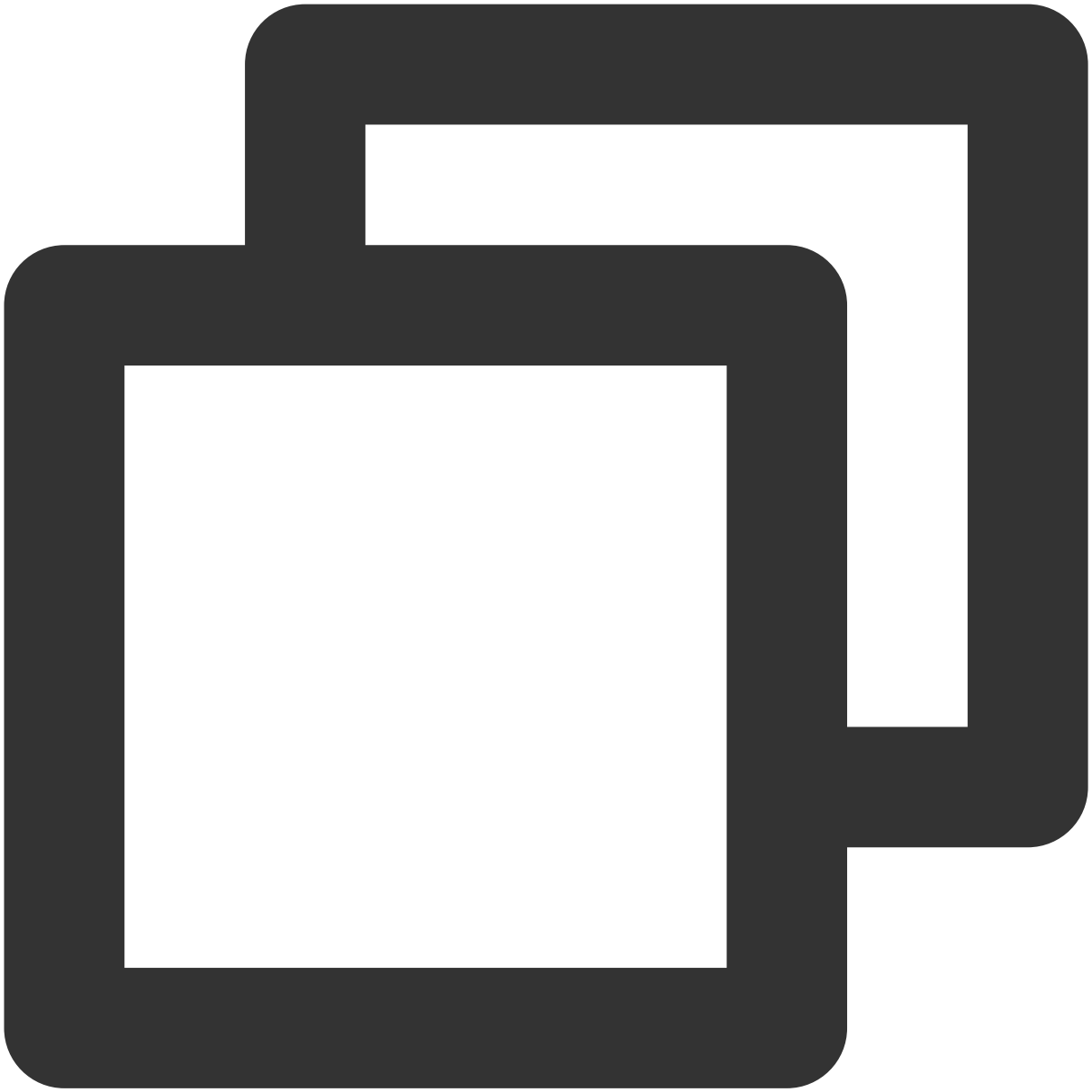
案例一：普通文档预览

请求



```
GET /sample.pdf?ci-process=doc-preview&page=1 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
```

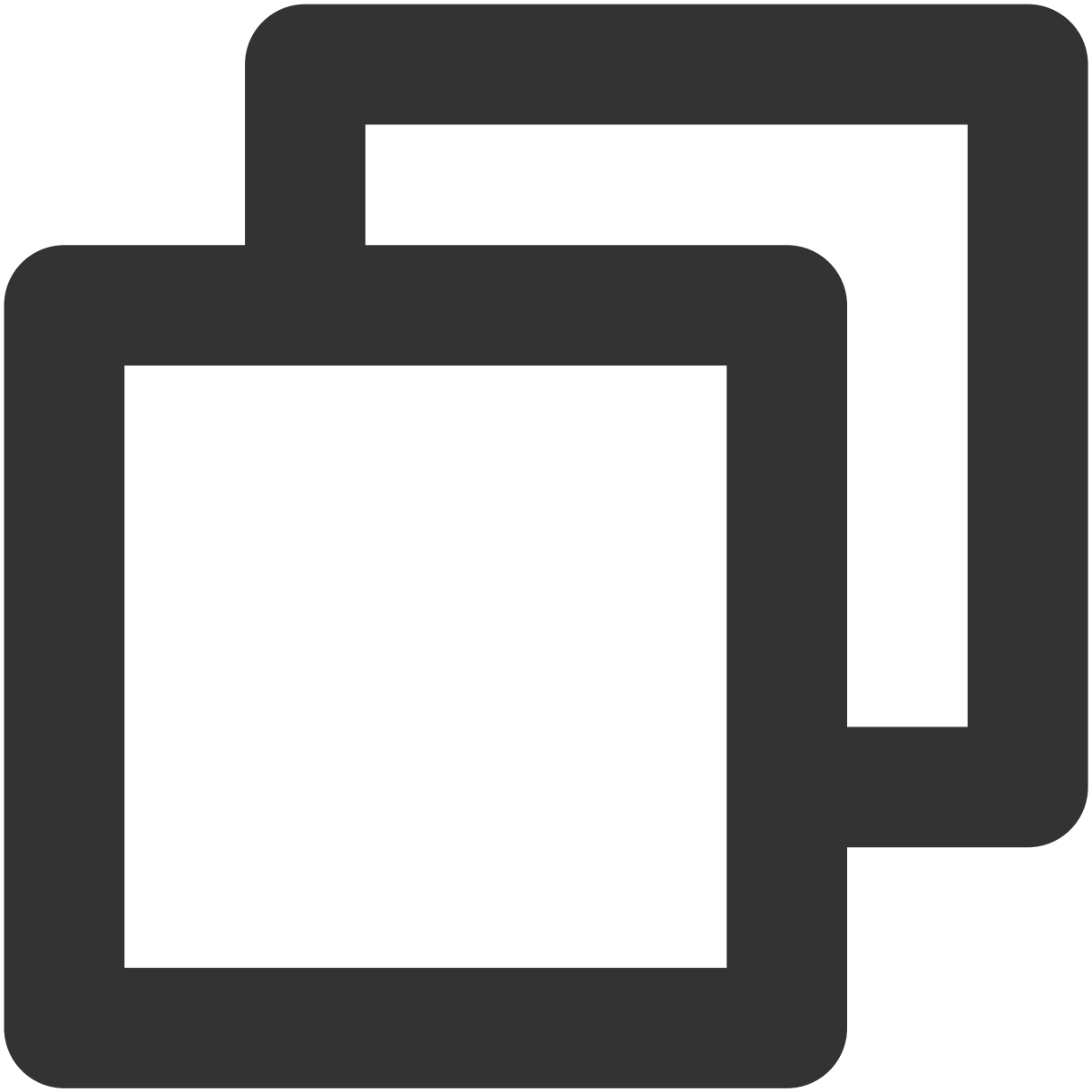
响应1：成功预览



```
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

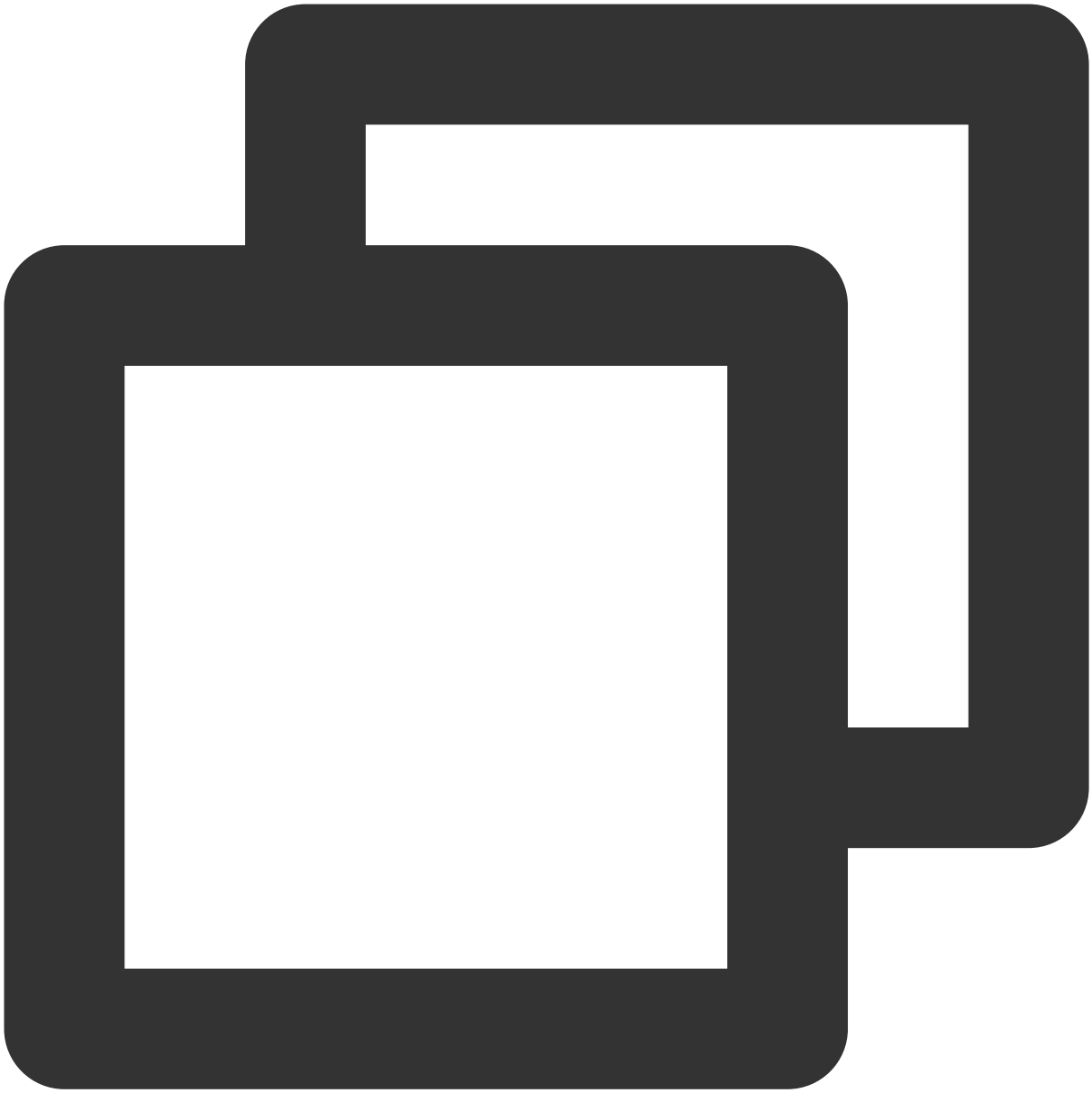
[转码后图片 data]
```

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
```

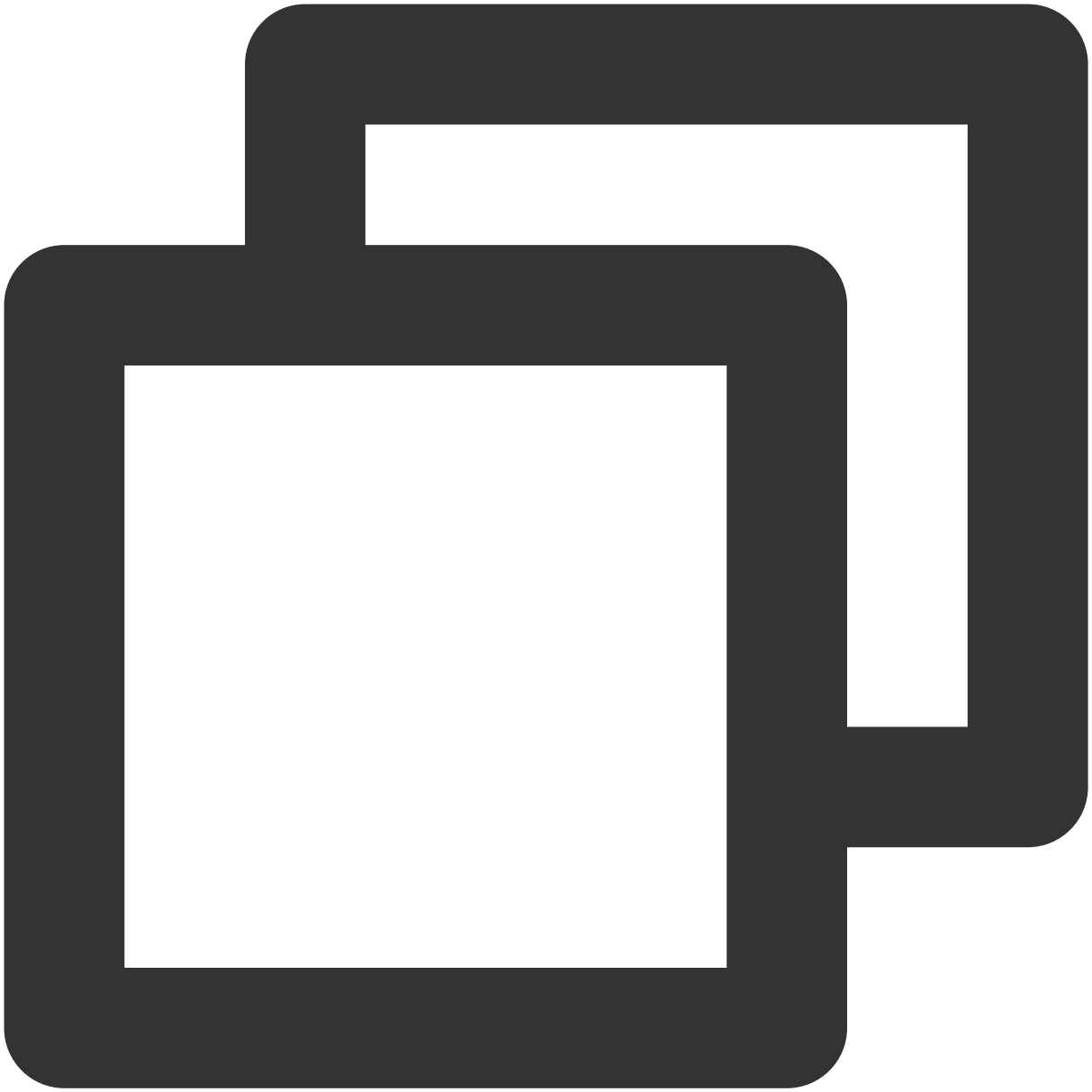
响应3：指定页码不存在



```
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
```

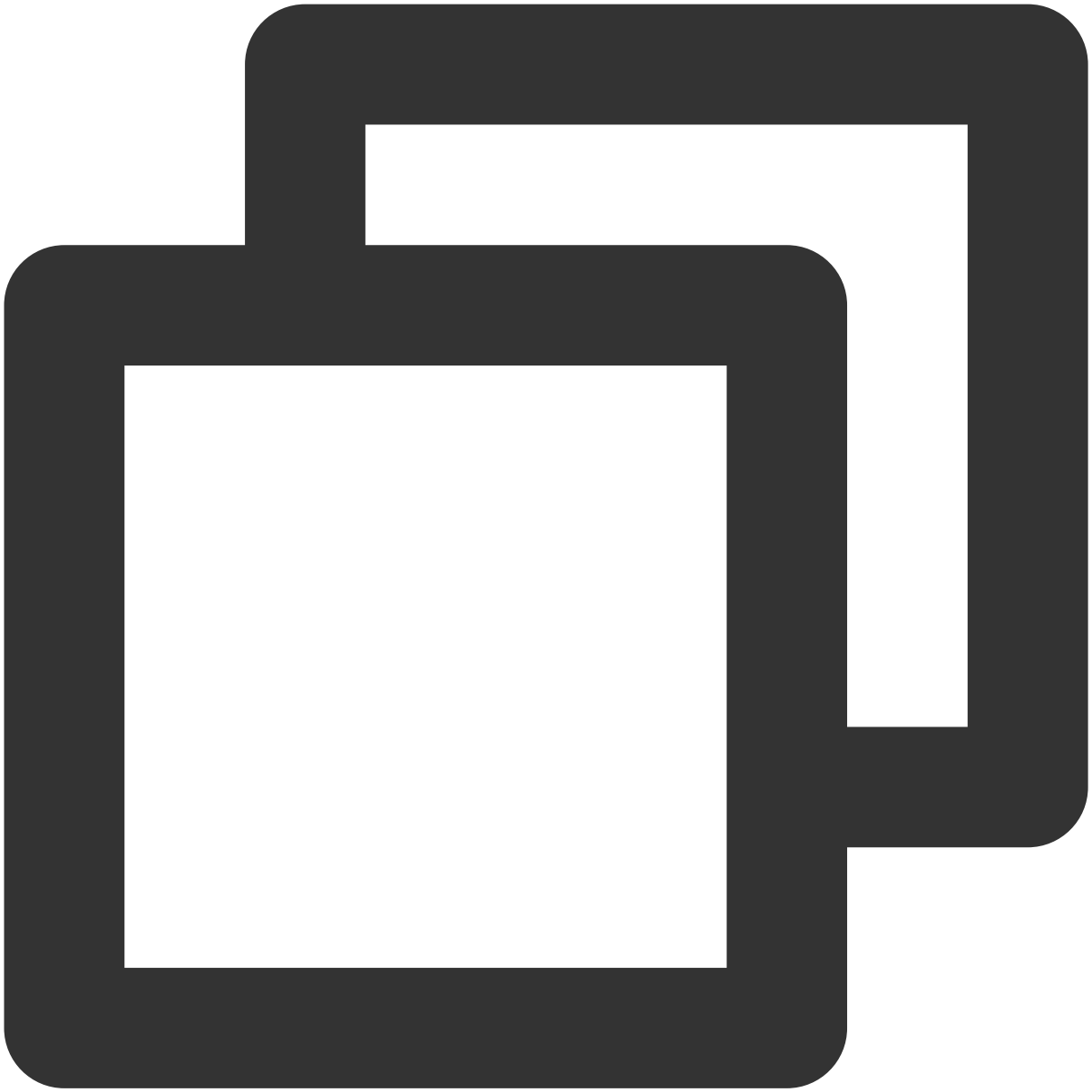
案例二：文档预览并进行缩放和添加文字水印的图片处理

请求



```
GET /sample.pdf?ci-process=doc-preview&page=1&ImageParams=imageMogr2/thumbnail/!50p
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
```

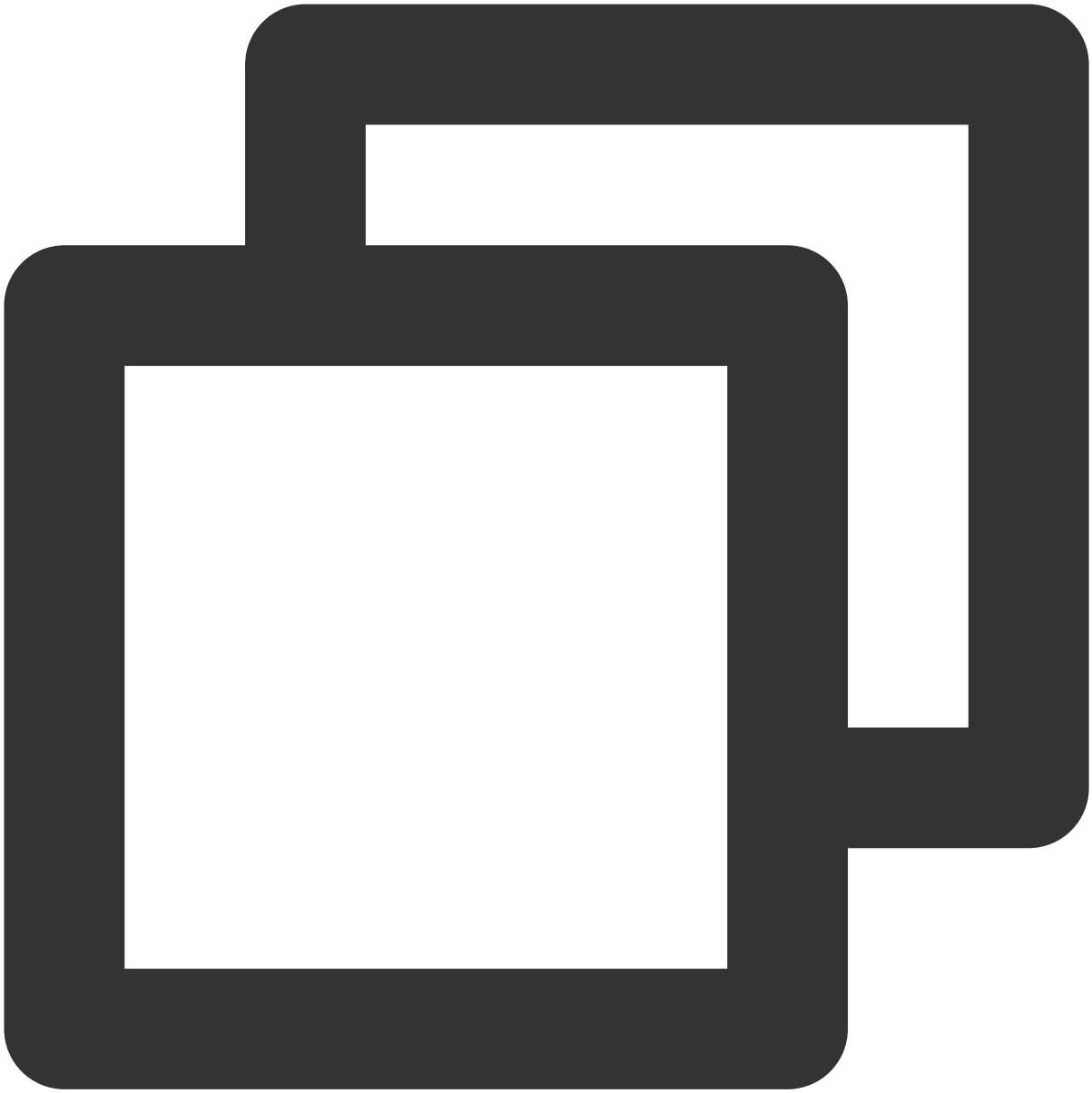
响应1：成功预览



```
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

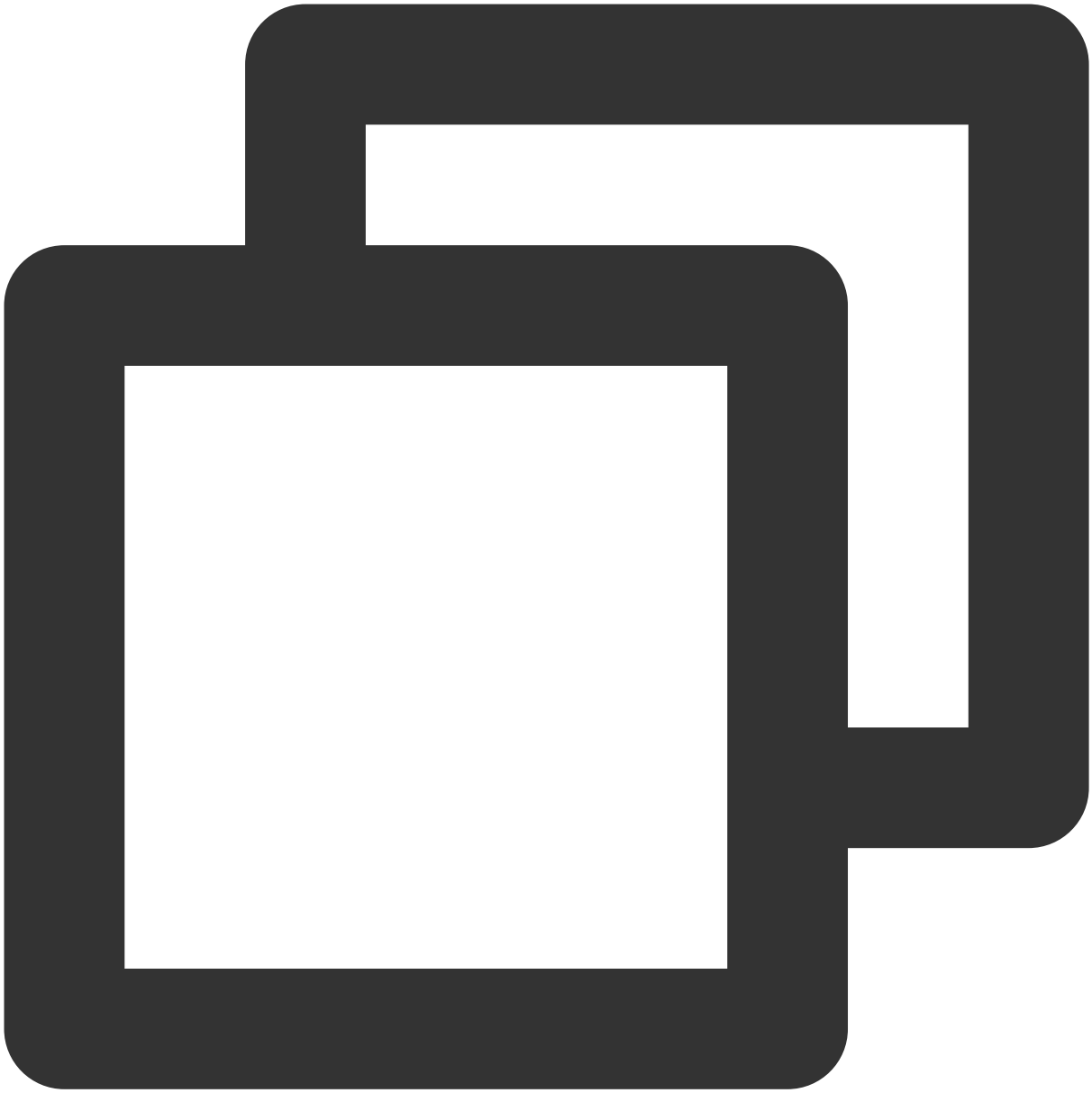
[转码后图片 data]
```

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：2024-01-20 11:39:30

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。

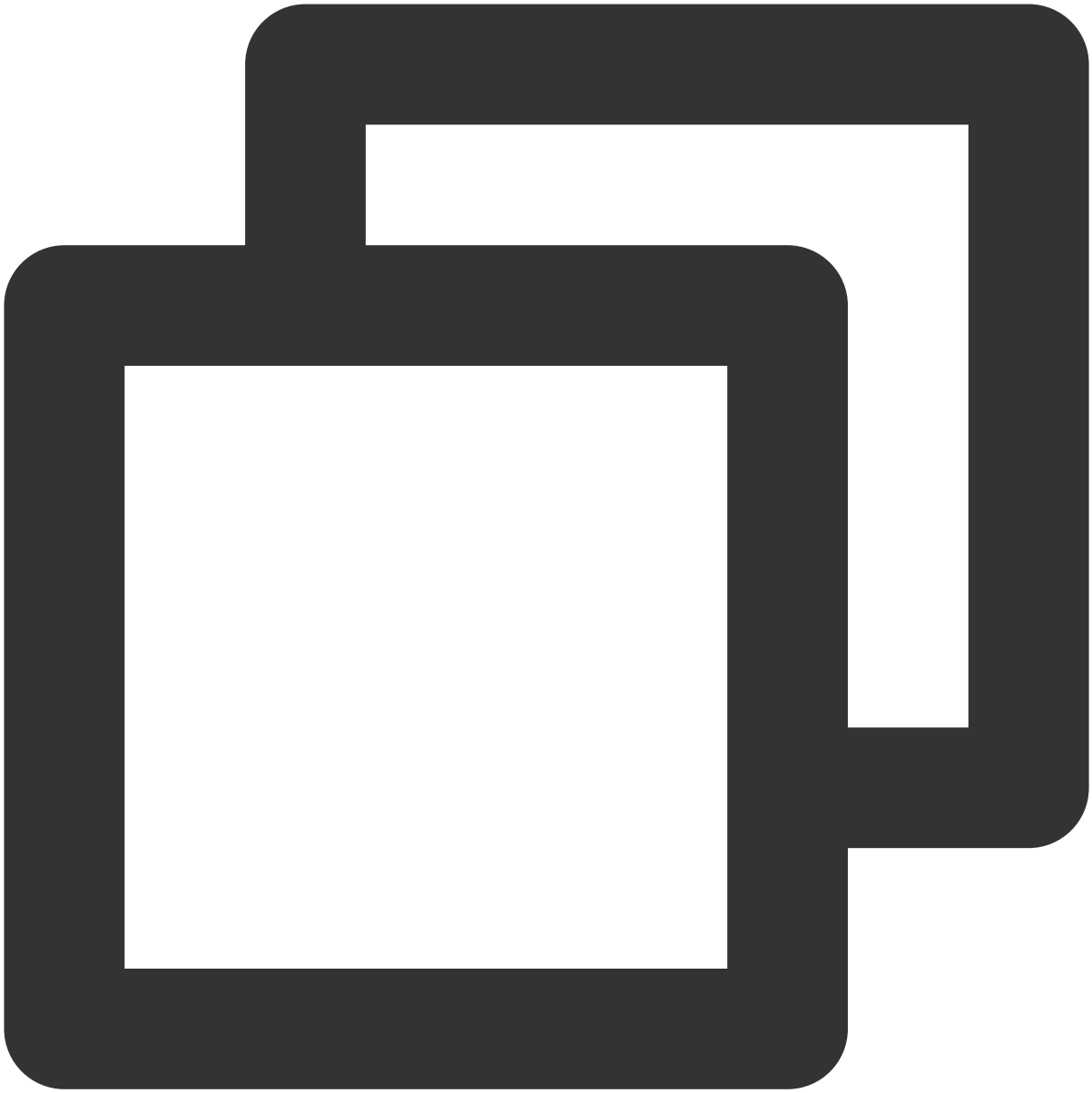
输入文件大小限制在200M 之内。

输入文件页数限制在5000页之内。

任务记录保留一个月，请及时保存任务记录。建议配置回调进行任务结果查询。

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：

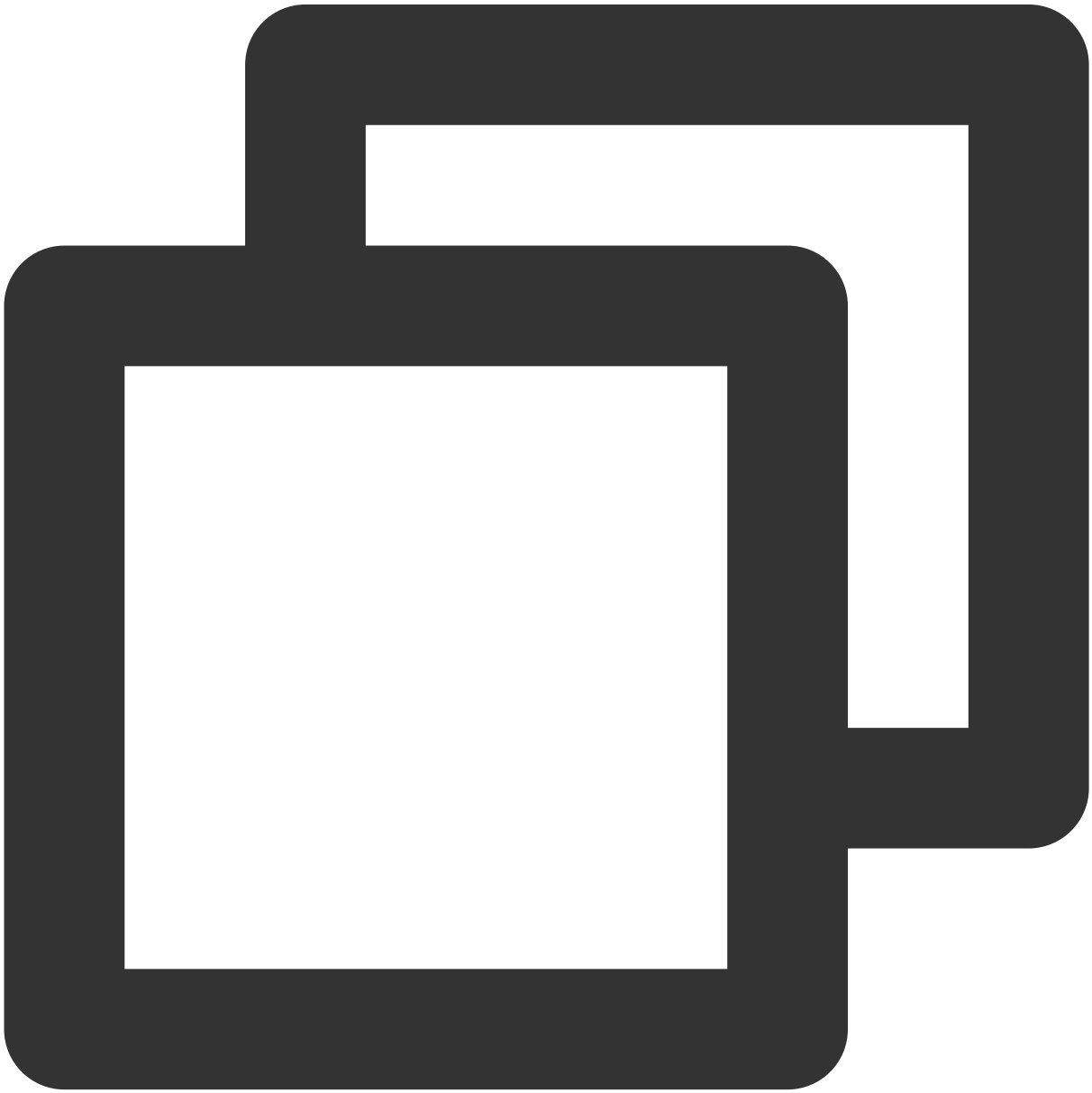
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	父节点	描述	类型	是否必选

字）				
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，分页范围可以通过 StartPage、EndPage 控制。默认值为 `0` ，按页导出图	Int	否

版权所有：腾讯云计算（北京）有限责任公司

		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

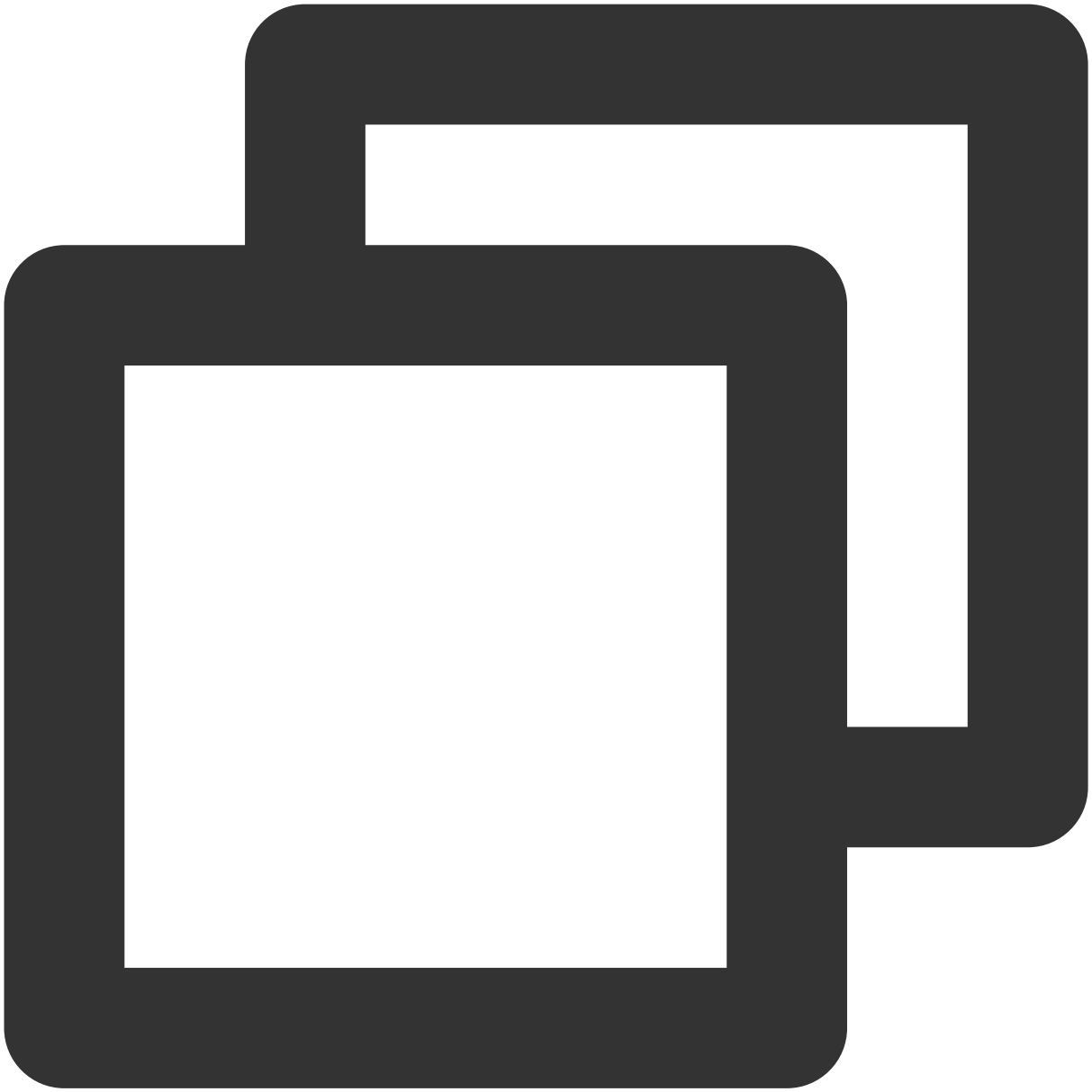
响应

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

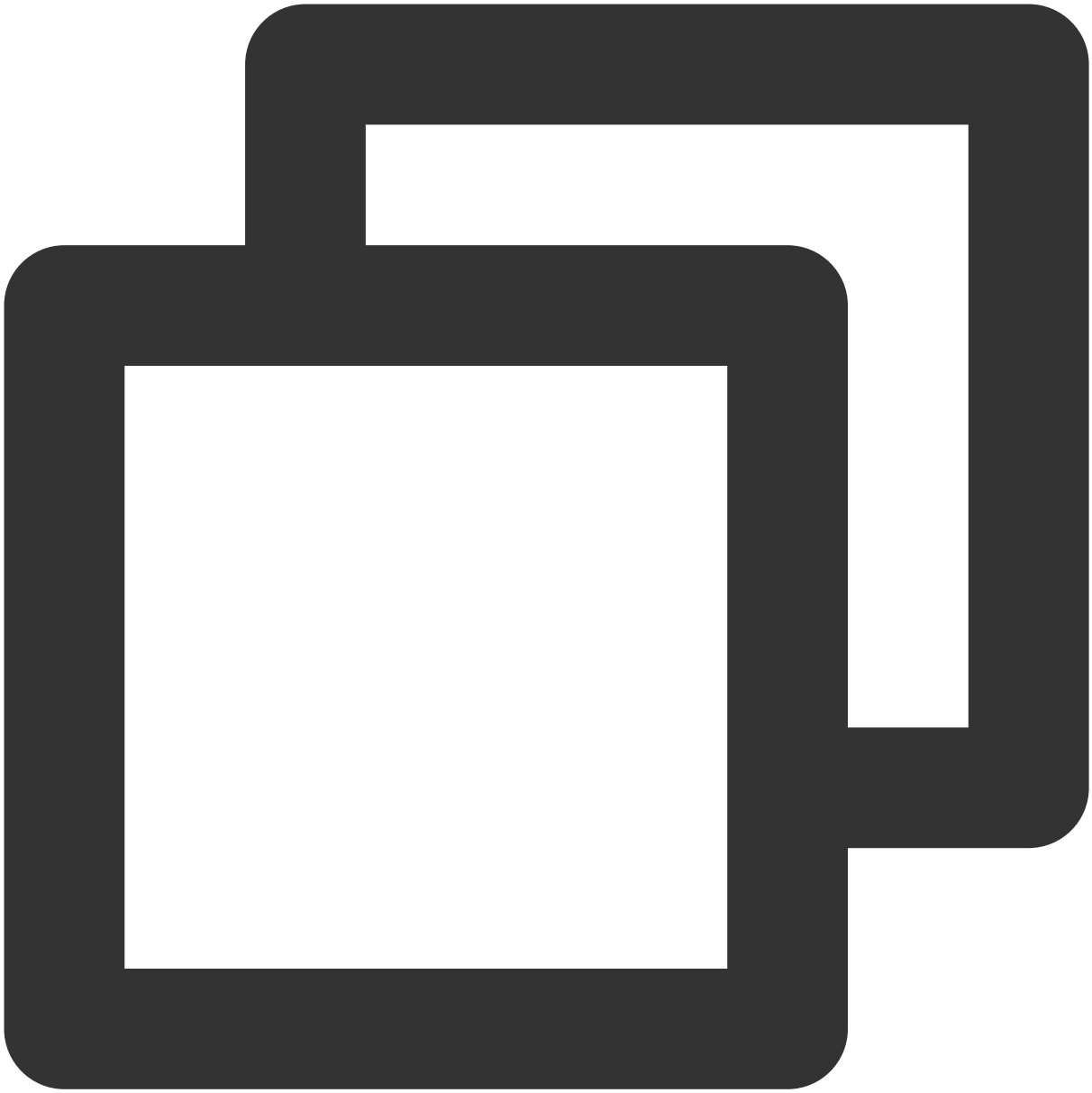
Container 节点 DocProcess 的内容：
同上面请求中的 Request.Operation.DocProcess 节点。
Container 节点 Output 的内容：
同上面请求中的 Request.Operation.Output 节点。

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

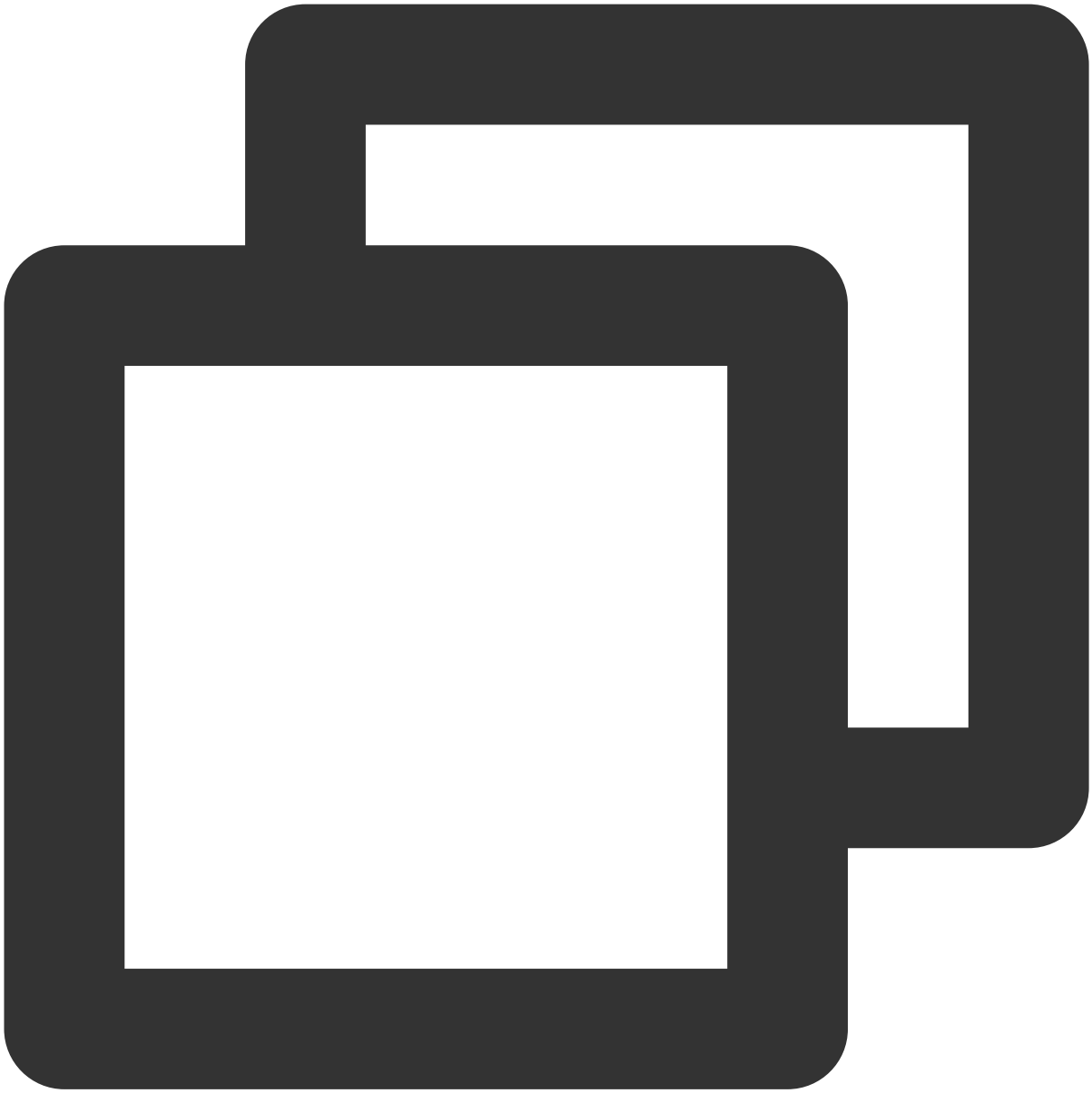


```
POST /doc_jobs HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: examplebucket-1250000000.ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Content-Length: 546
Authorization: Authorization

<?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：2024-01-20 11:39:30

功能描述

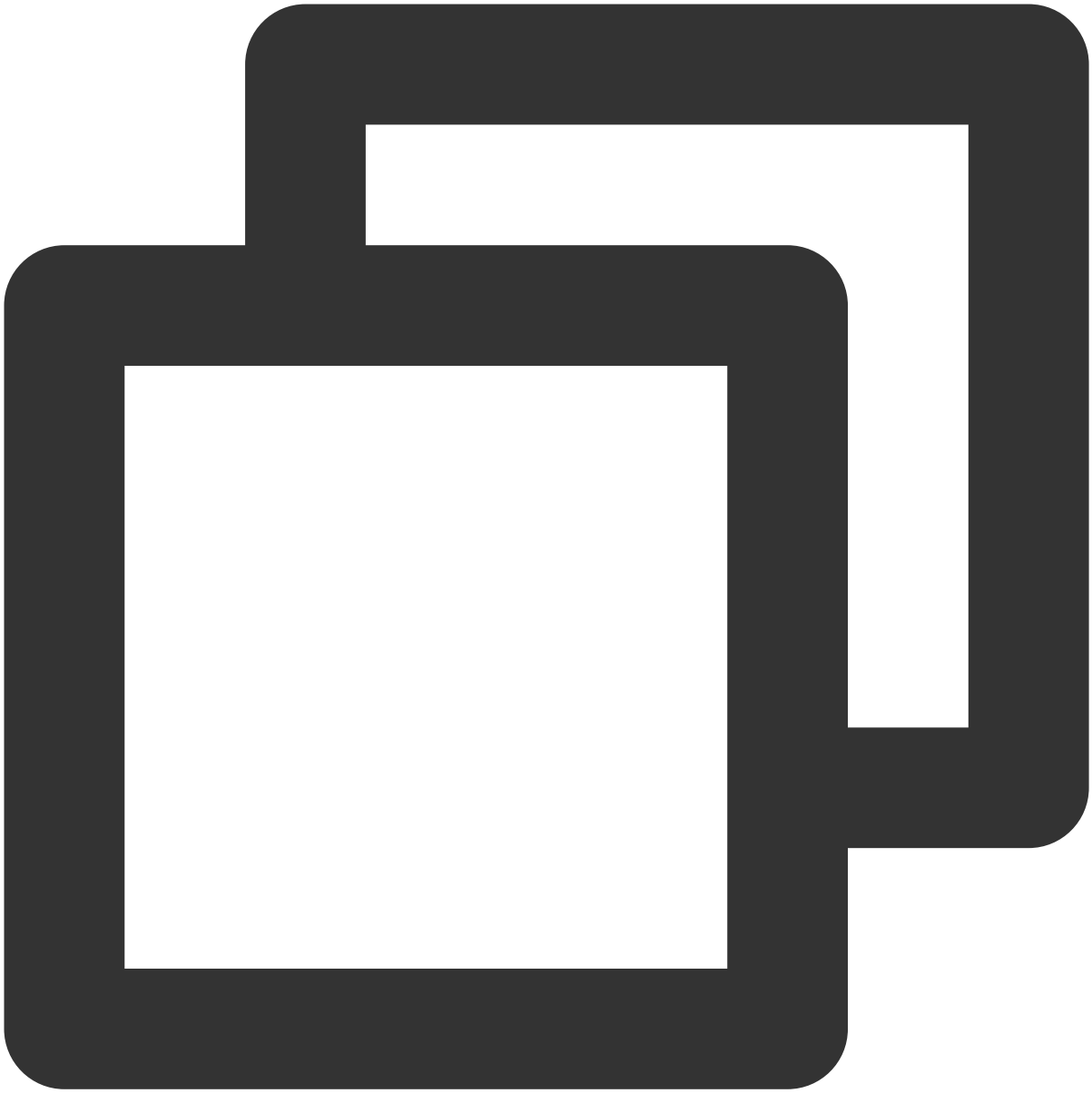
DescribeDocProcessJob 用于查询指定的文档转码任务。

请求

注意：

文档预览任务记录保留一个月，请及时保存任务记录。建议配置回调进行任务结果查询。

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。

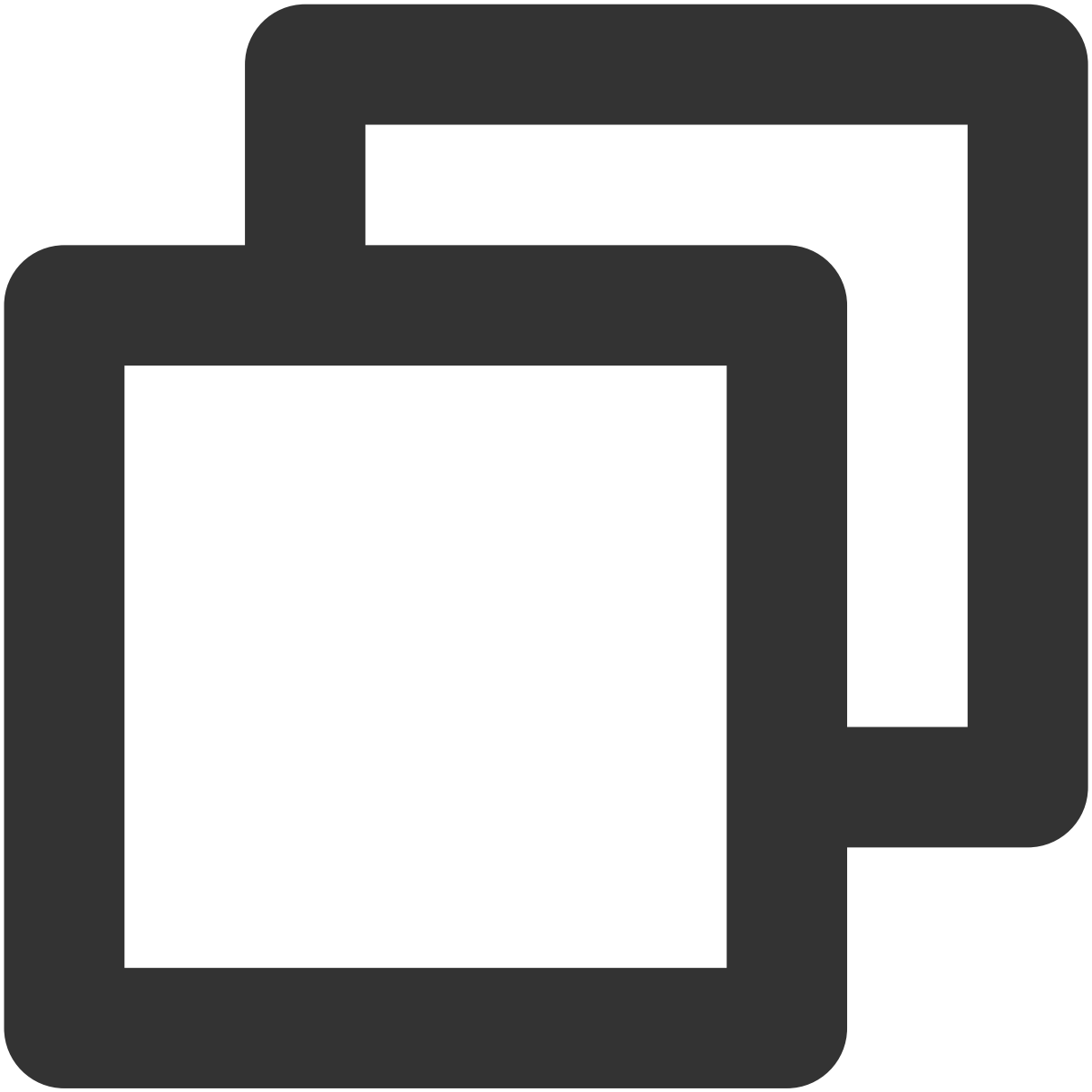
响应

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

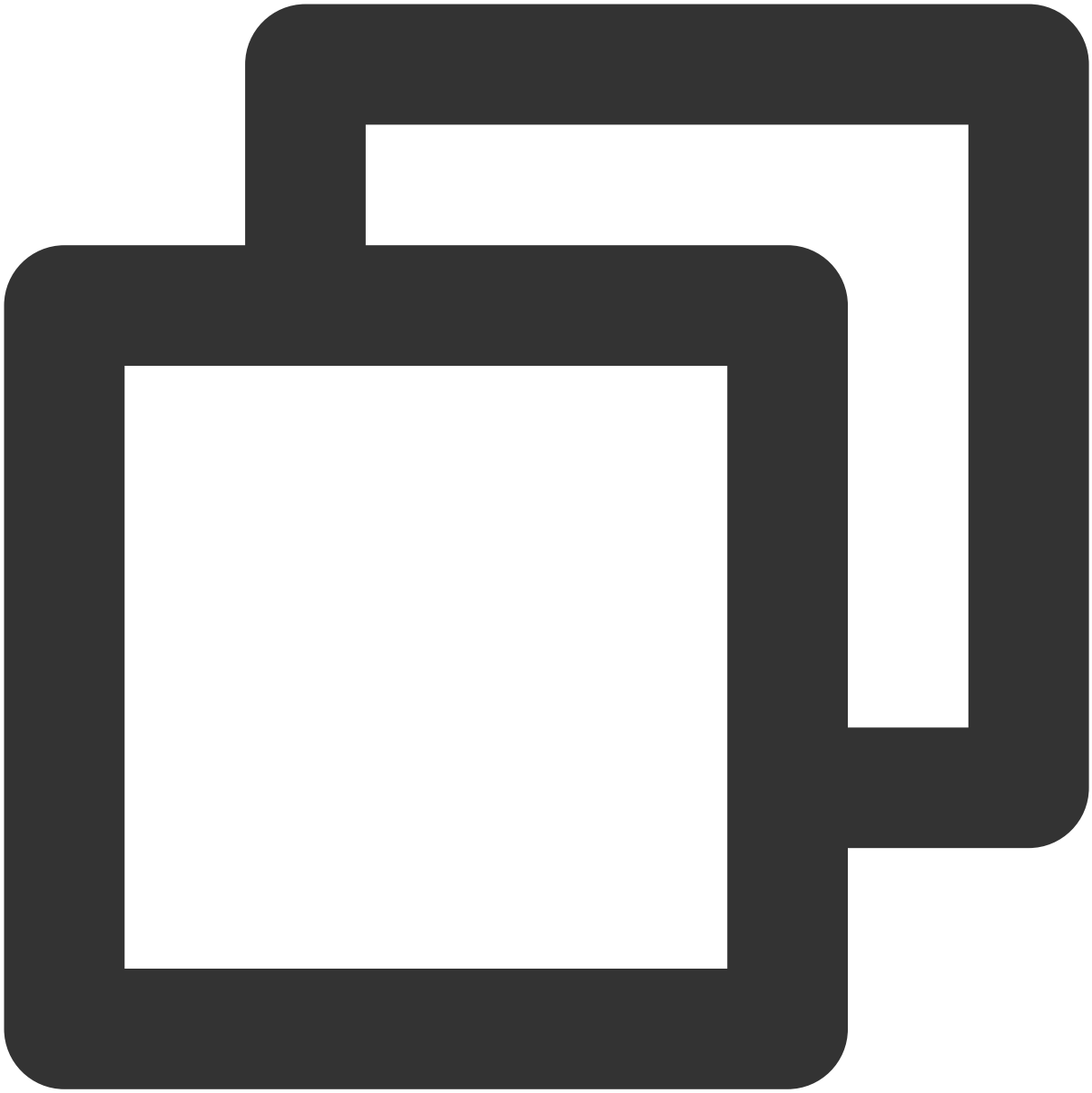
TgtUri	Response.JobsDetail.Operation.DocProcessResult.PageInfo	预览产物生成的 cos 桶路径	Int
X-SheetPics	Response.JobsDetail.Operation.DocProcessResult.PageInfo	当前 Sheet 生成的图片总数（源文件为 Excel 特有参数）	Int
PicIndex	Response.JobsDetail.Operation.DocProcessResult.PageInfo	当前预览产物在整个源文件中的序号（源文件为 Excel 特有参数）	Int
PicNum	Response.JobsDetail.Operation.DocProcessResult.PageInfo	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。

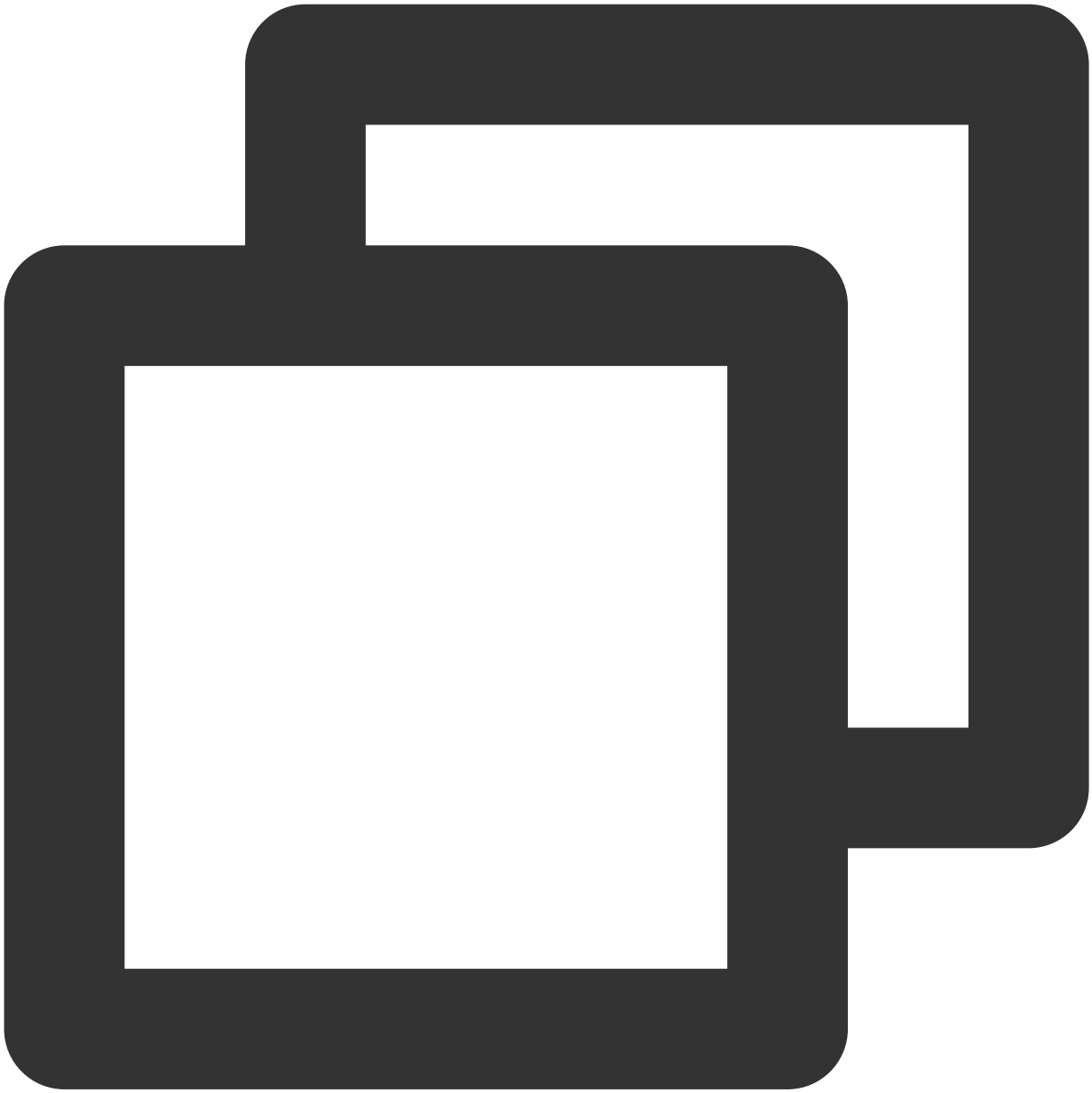
实际案例

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
```

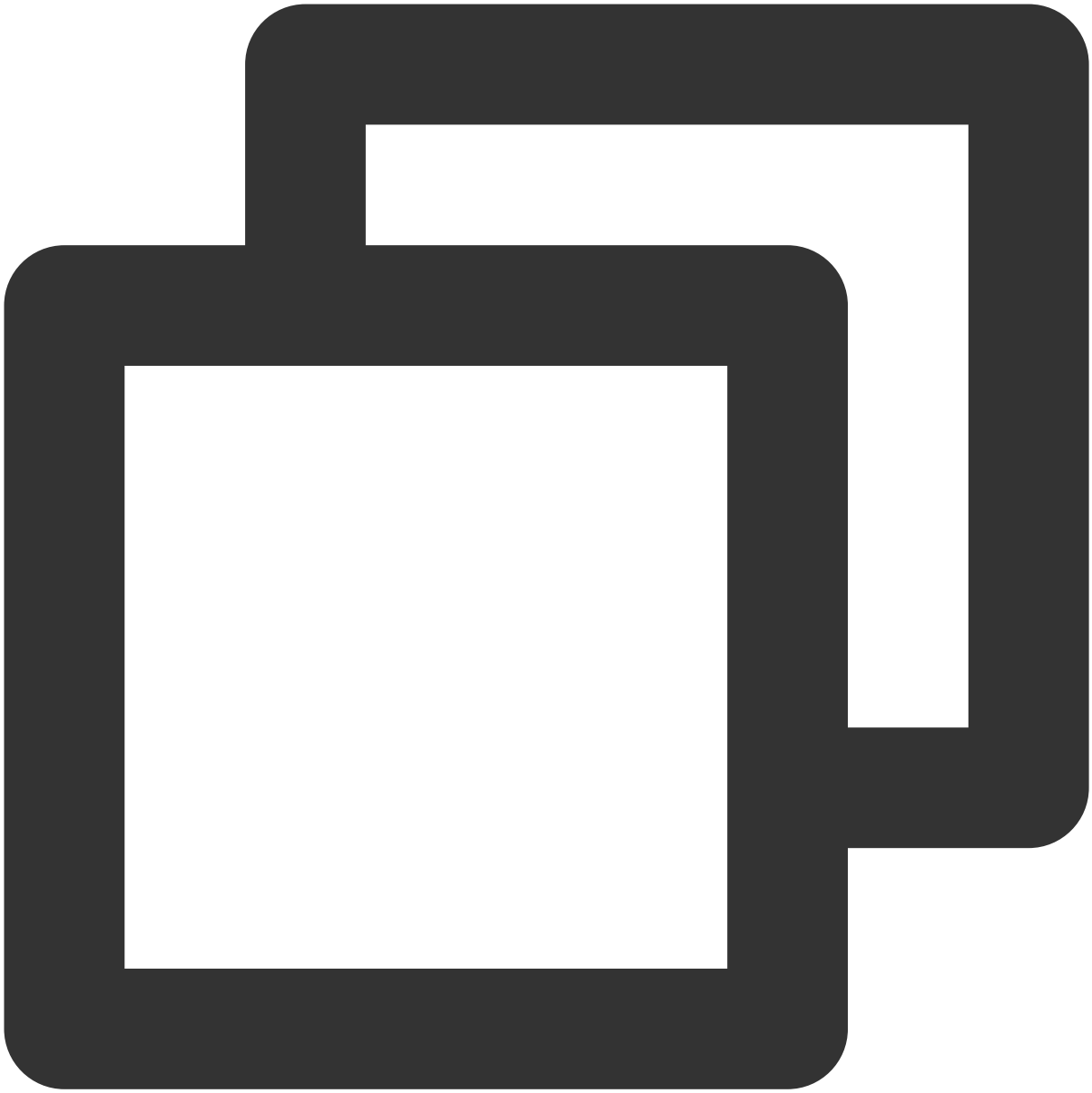
1.非 Excel 文档请求响应



```
<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>
```

拉取符合条件的文档转码任务

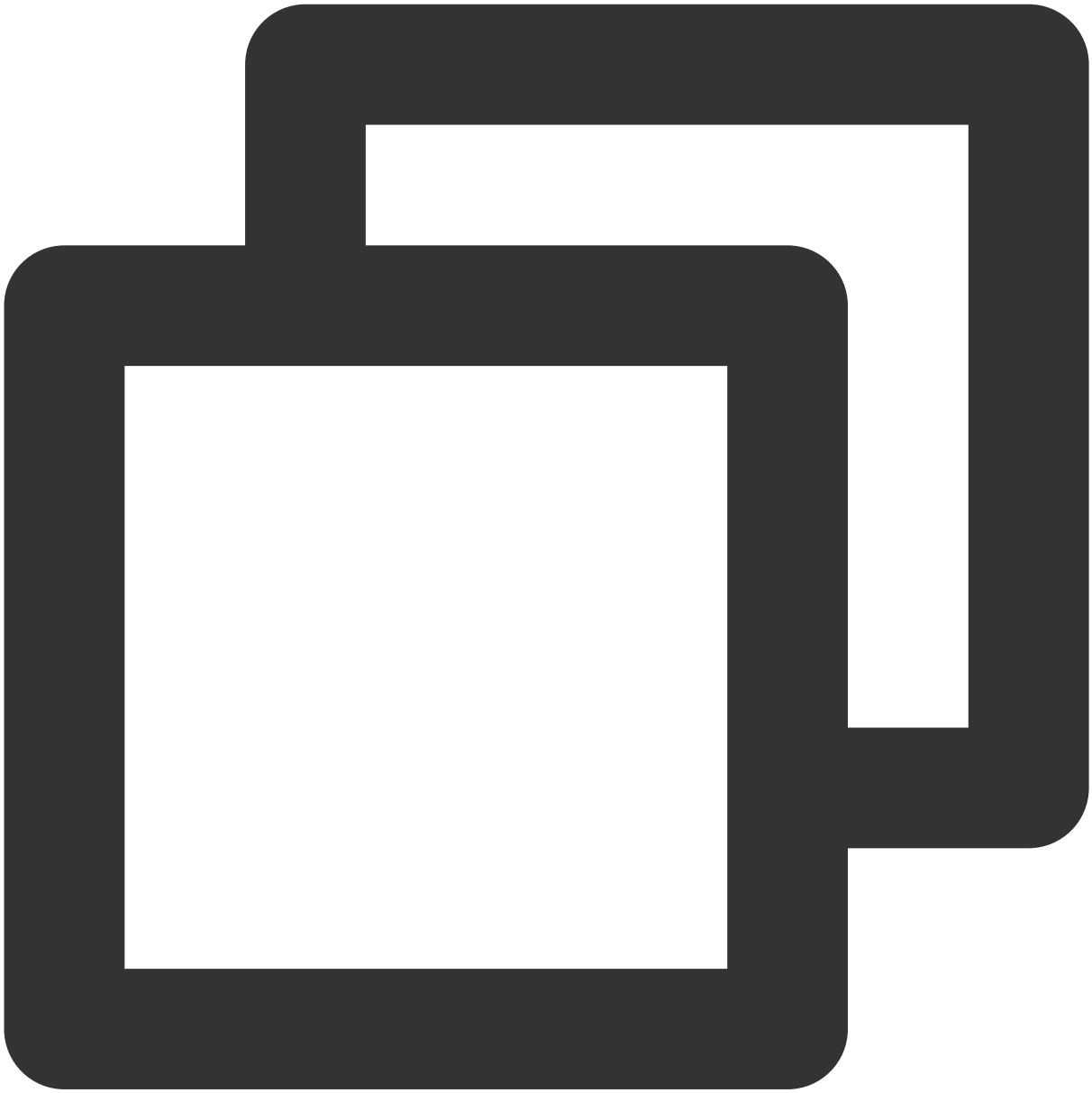
最近更新时间：2024-01-20 11:39:30

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

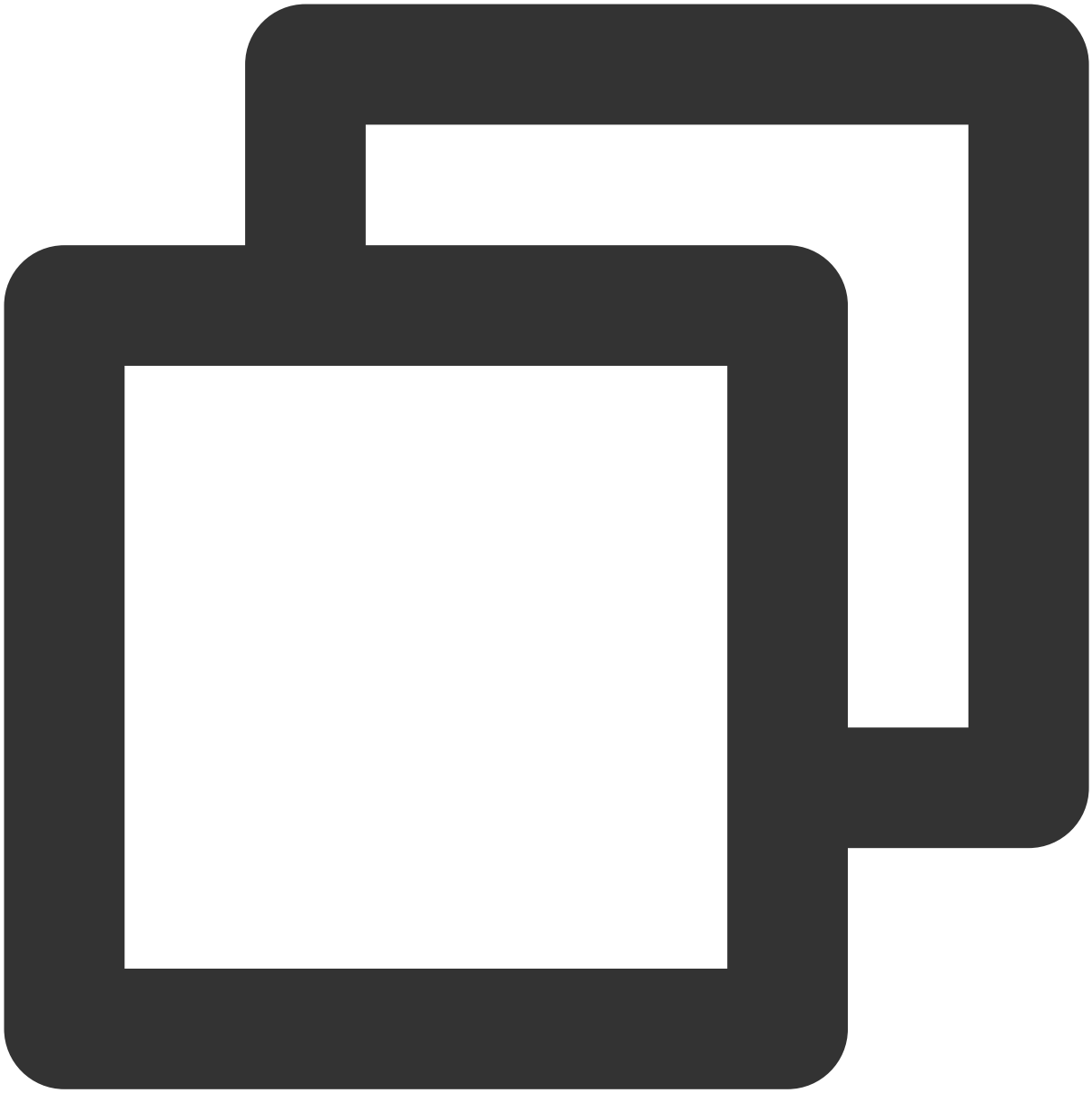
响应

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：

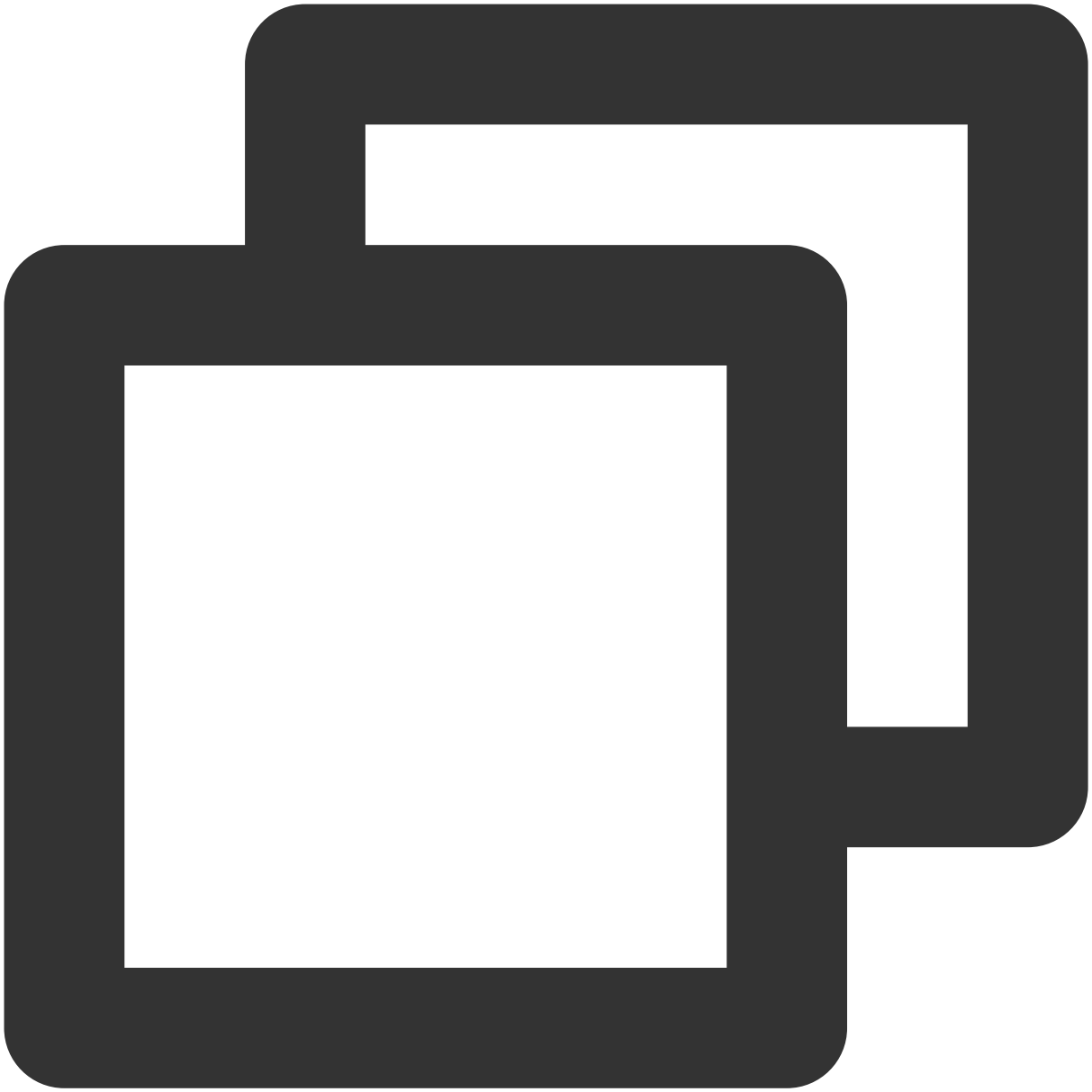
节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息，同 CreateDocProcessJobs 接口中的 Response.JobsDetail 节点	Container
NextToken	Response	翻页的上下文 Token	String

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。

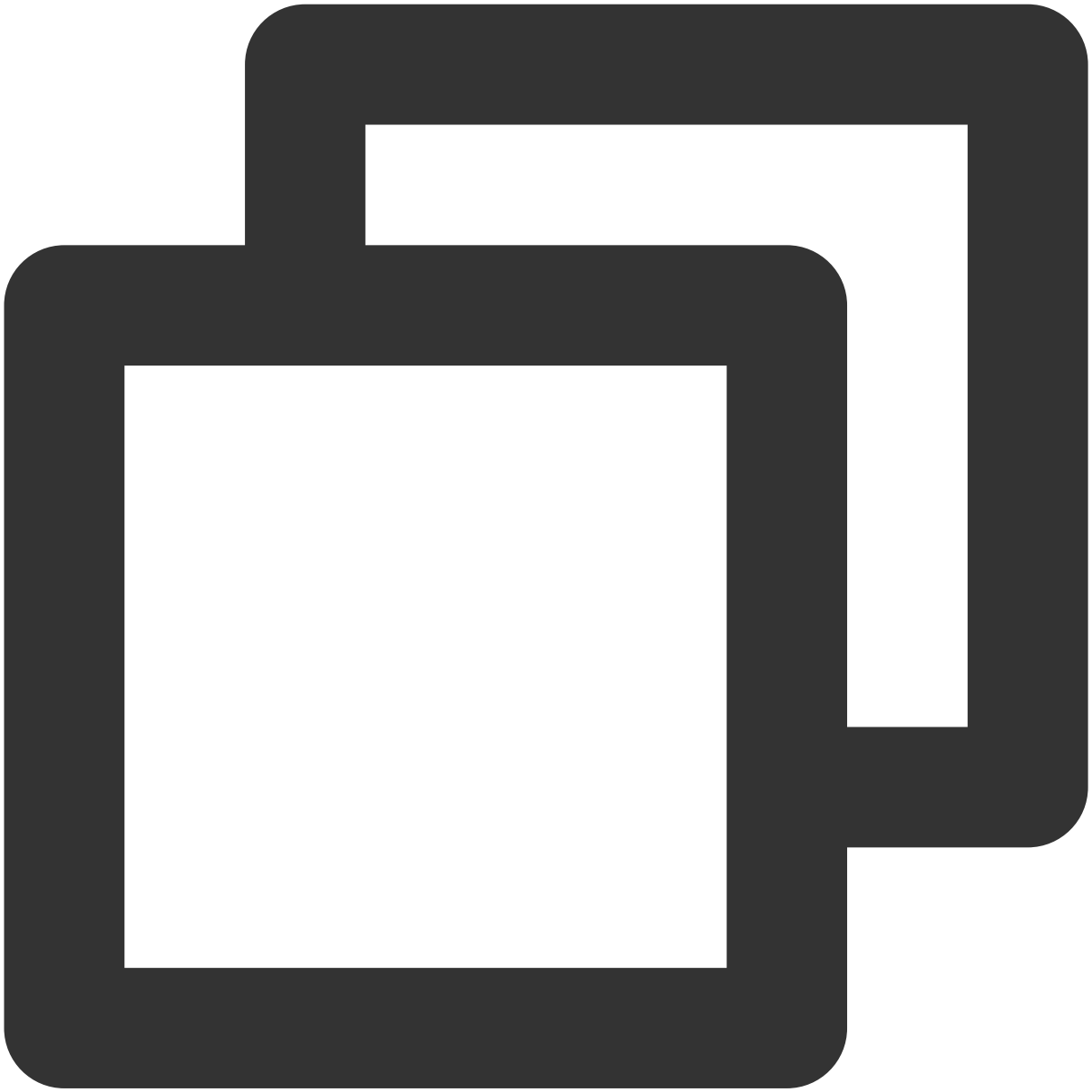
实际案例

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

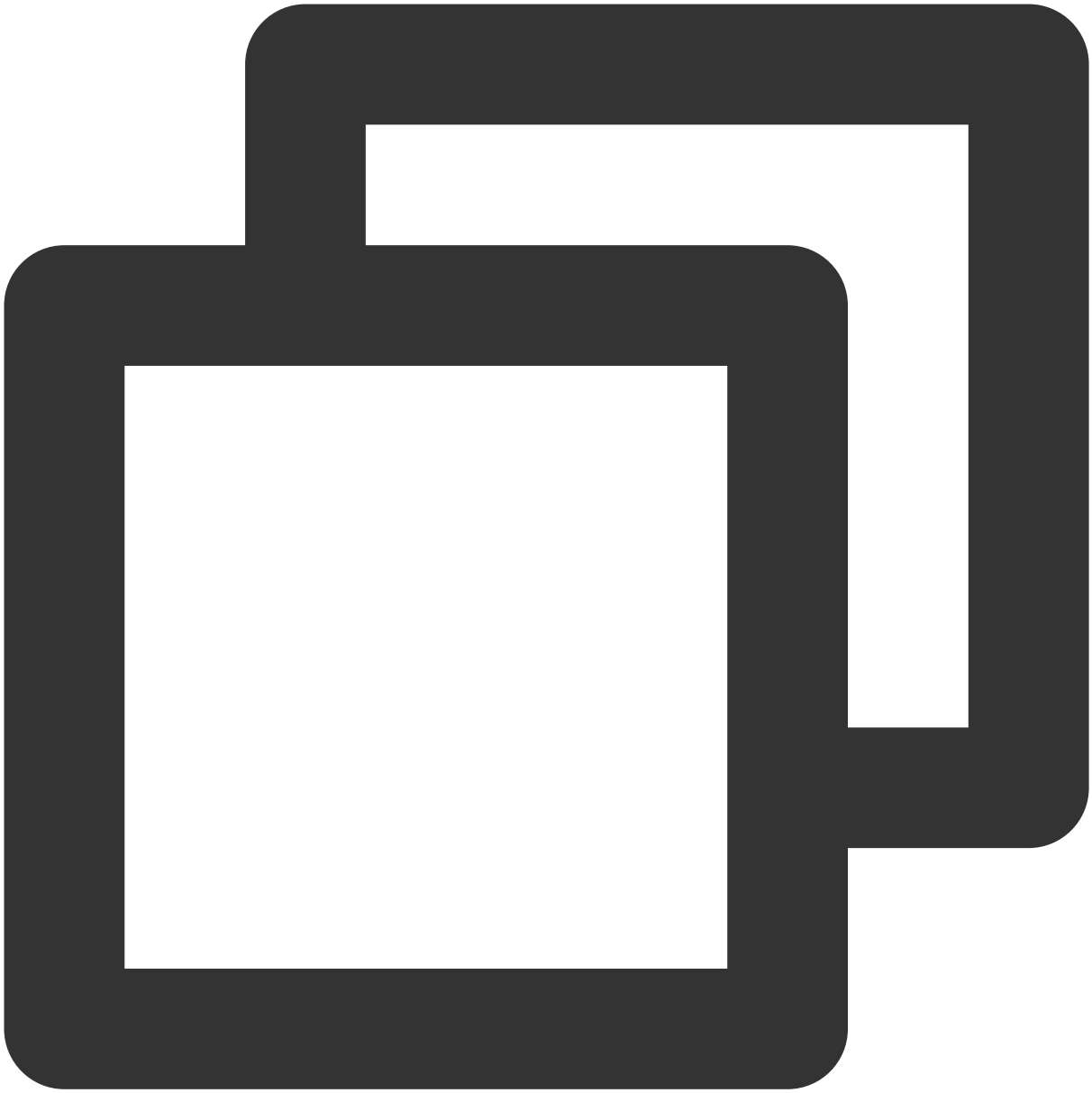
最近更新时间：2024-01-20 11:39:30

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

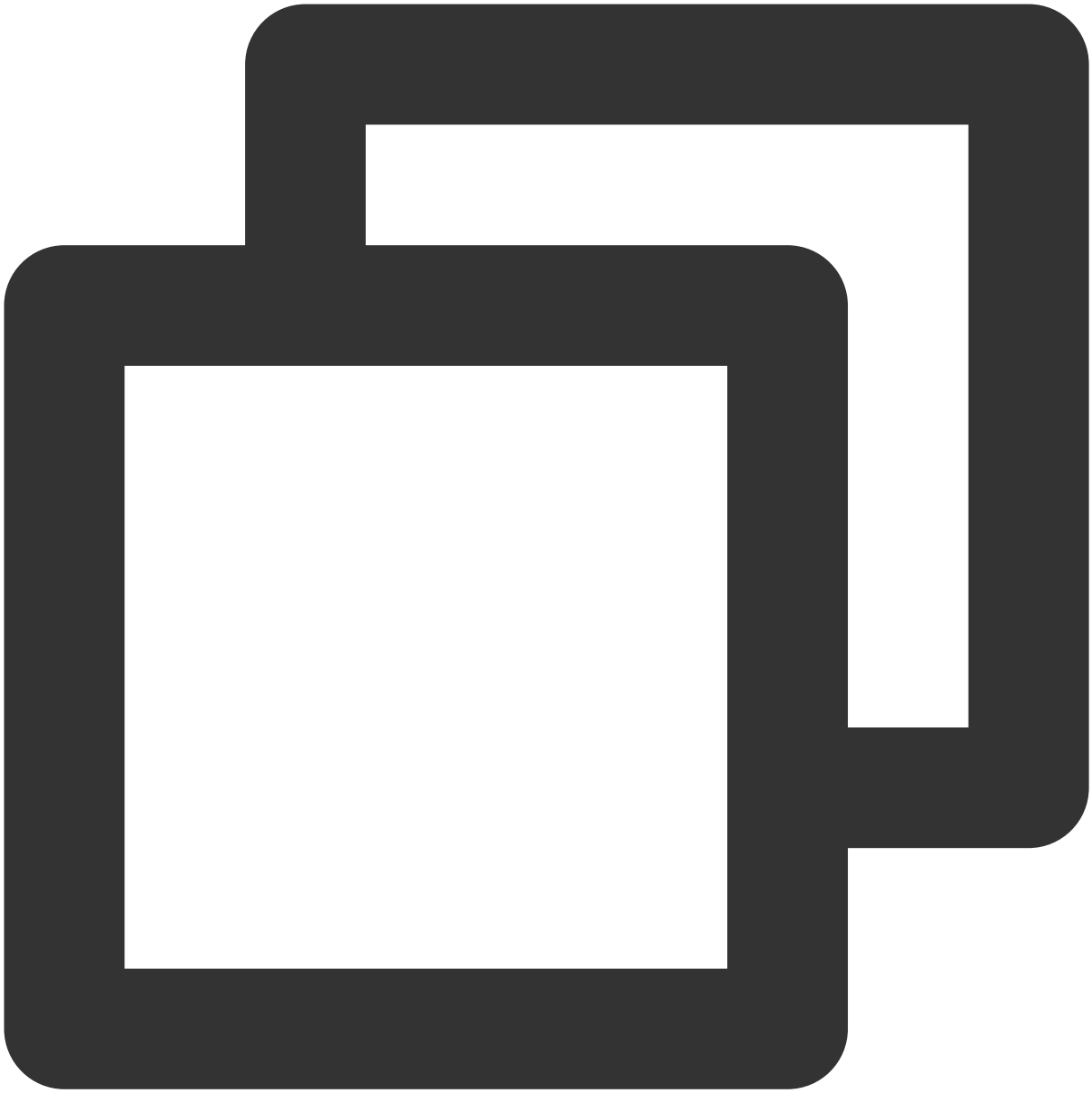
响应

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>
```

```
                        <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

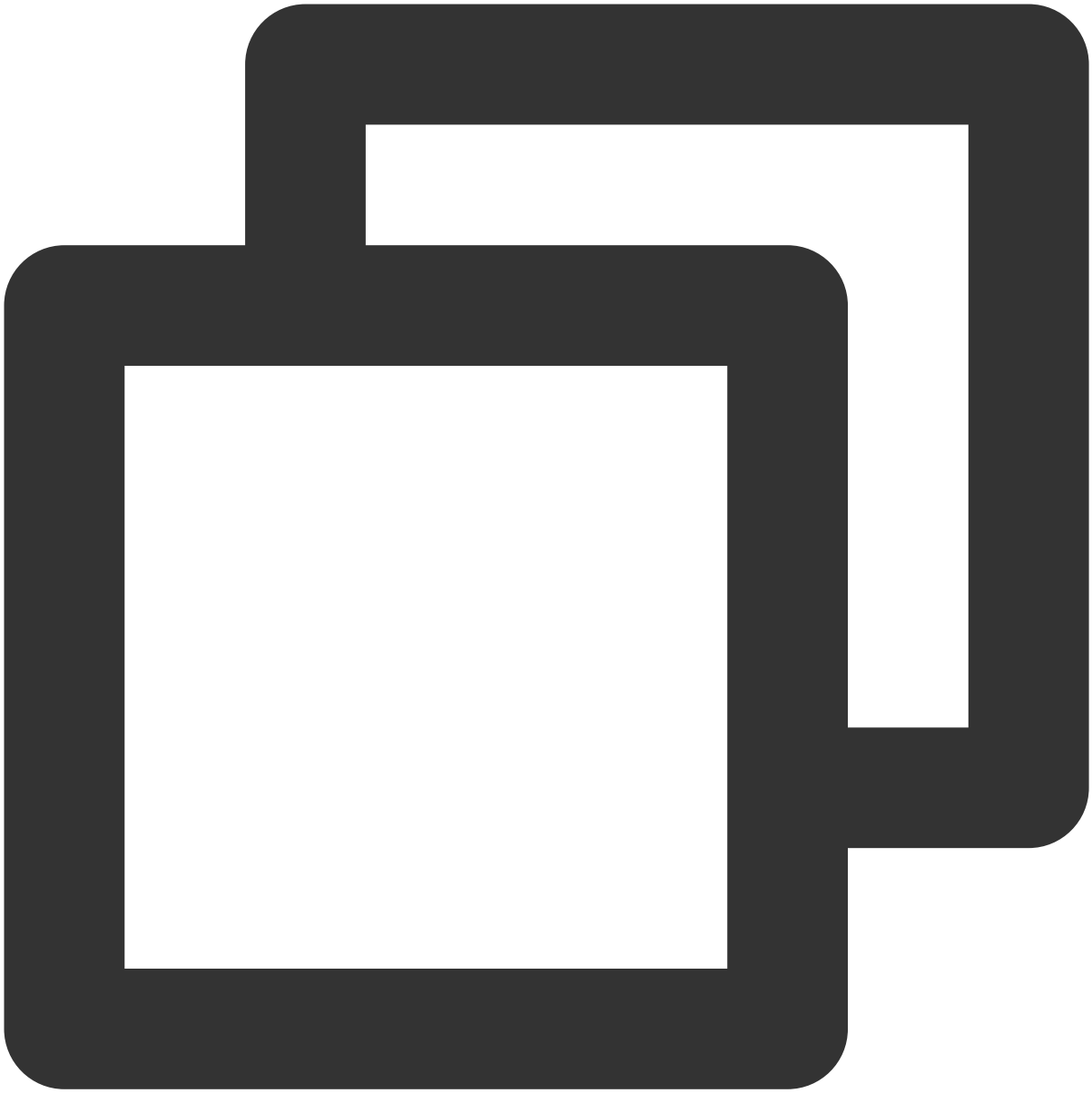
版权所有：腾讯云计算（北京）有限责任公司

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。



```
<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：

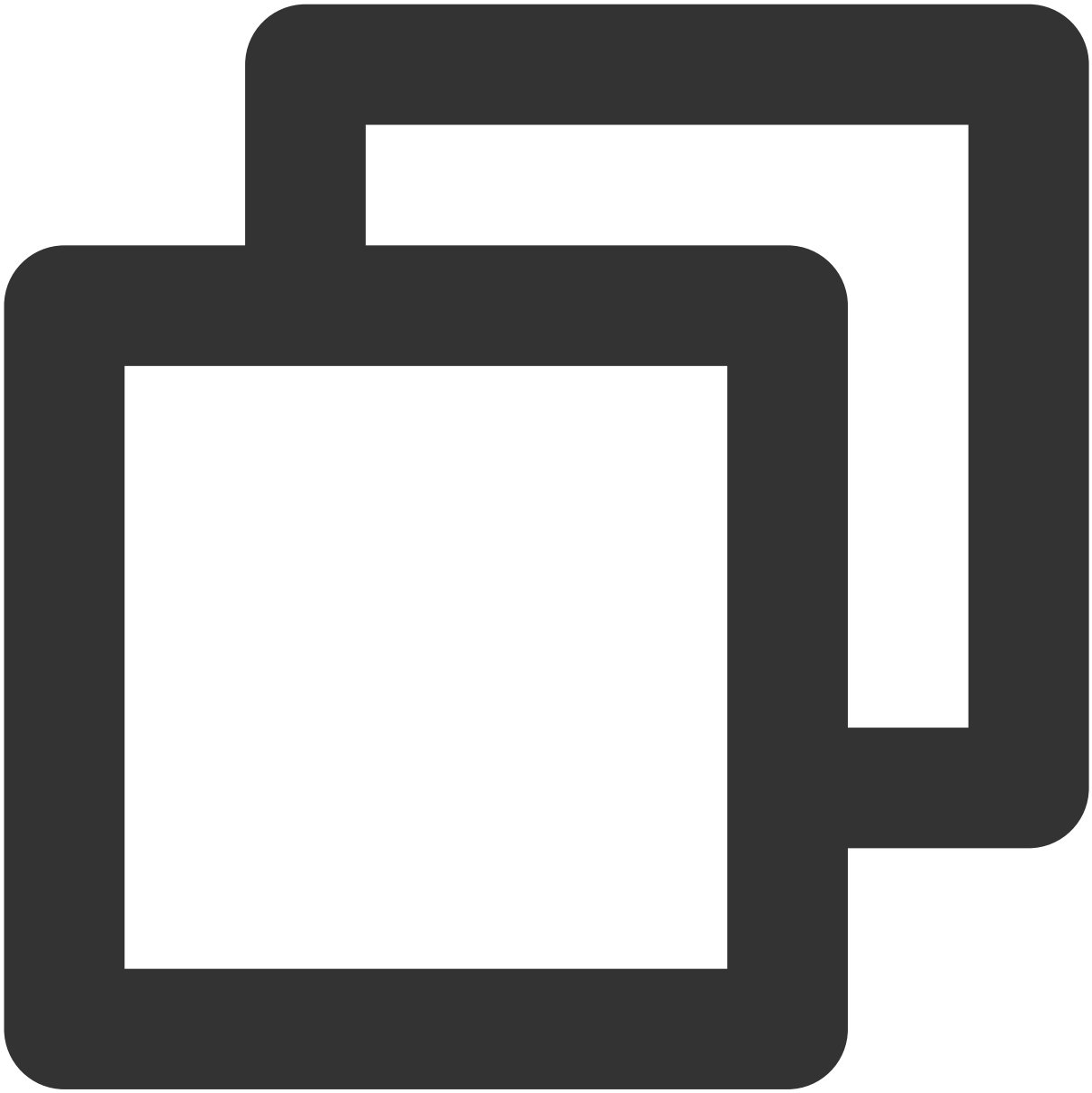
节点名称（关键字）	父节点	描述	类型
QueueID	Response.NonExistPIDs	队列 ID	String

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。

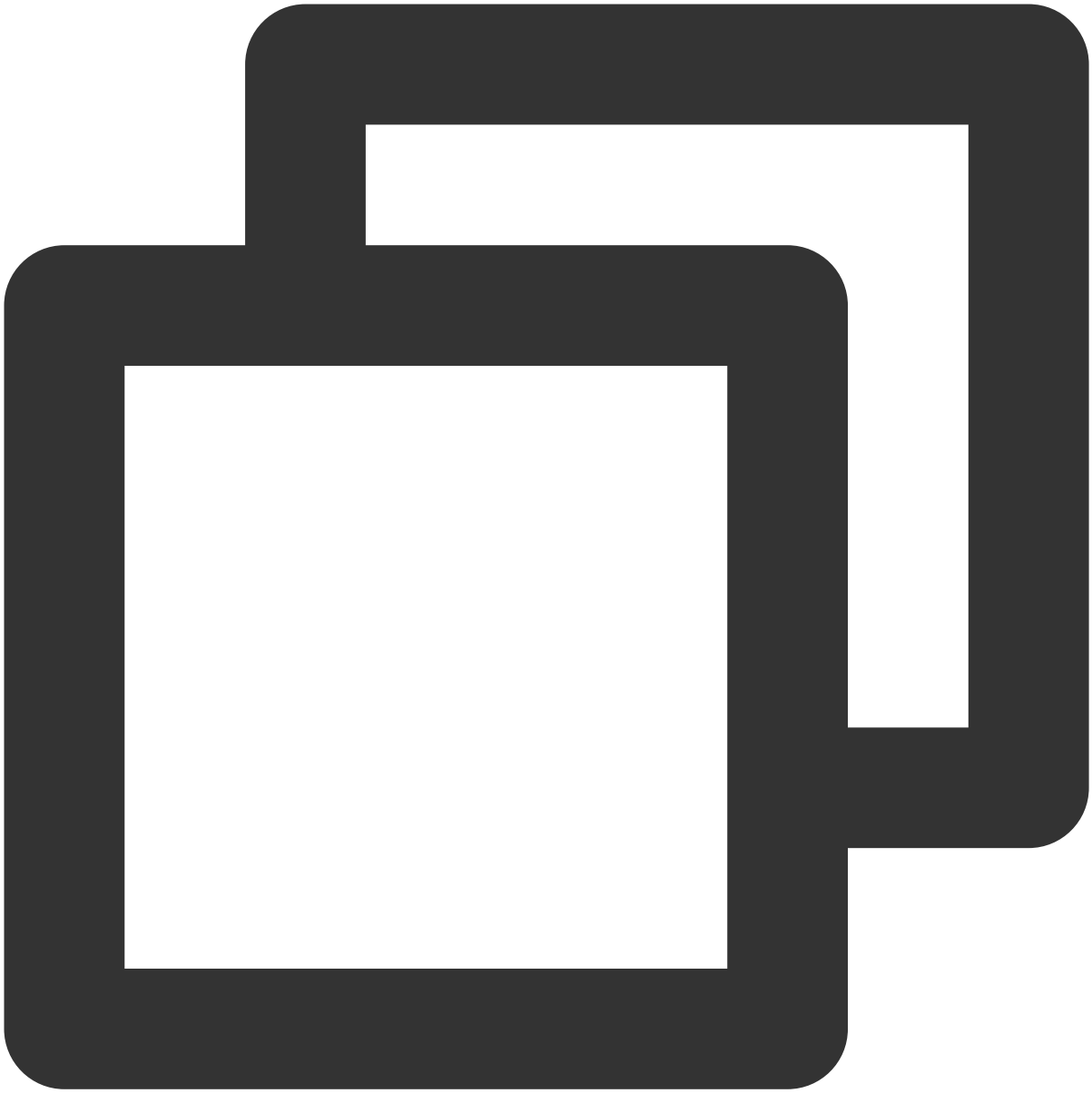
实际案例

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

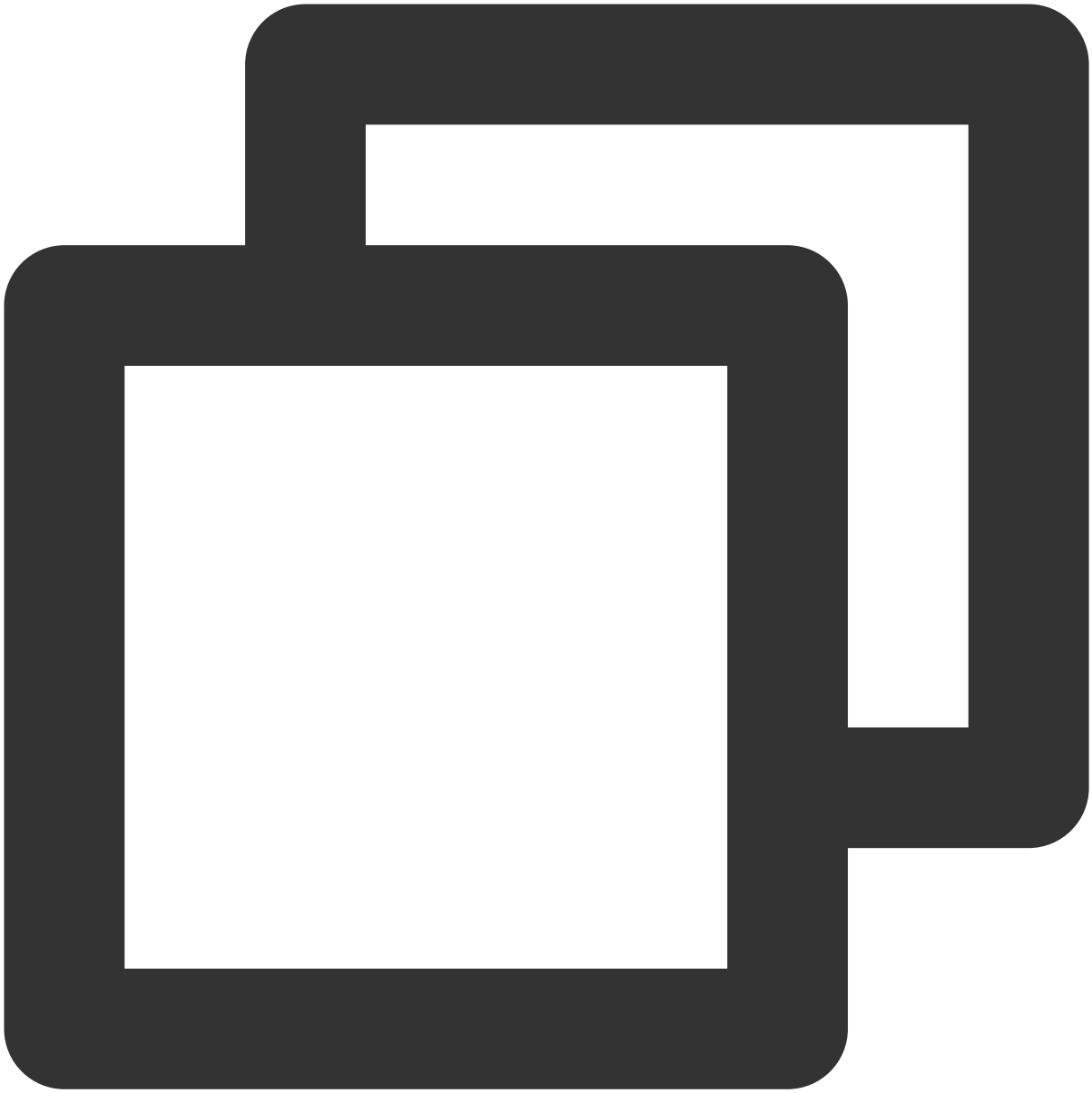
最近更新时间：2024-01-20 11:39:30

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：

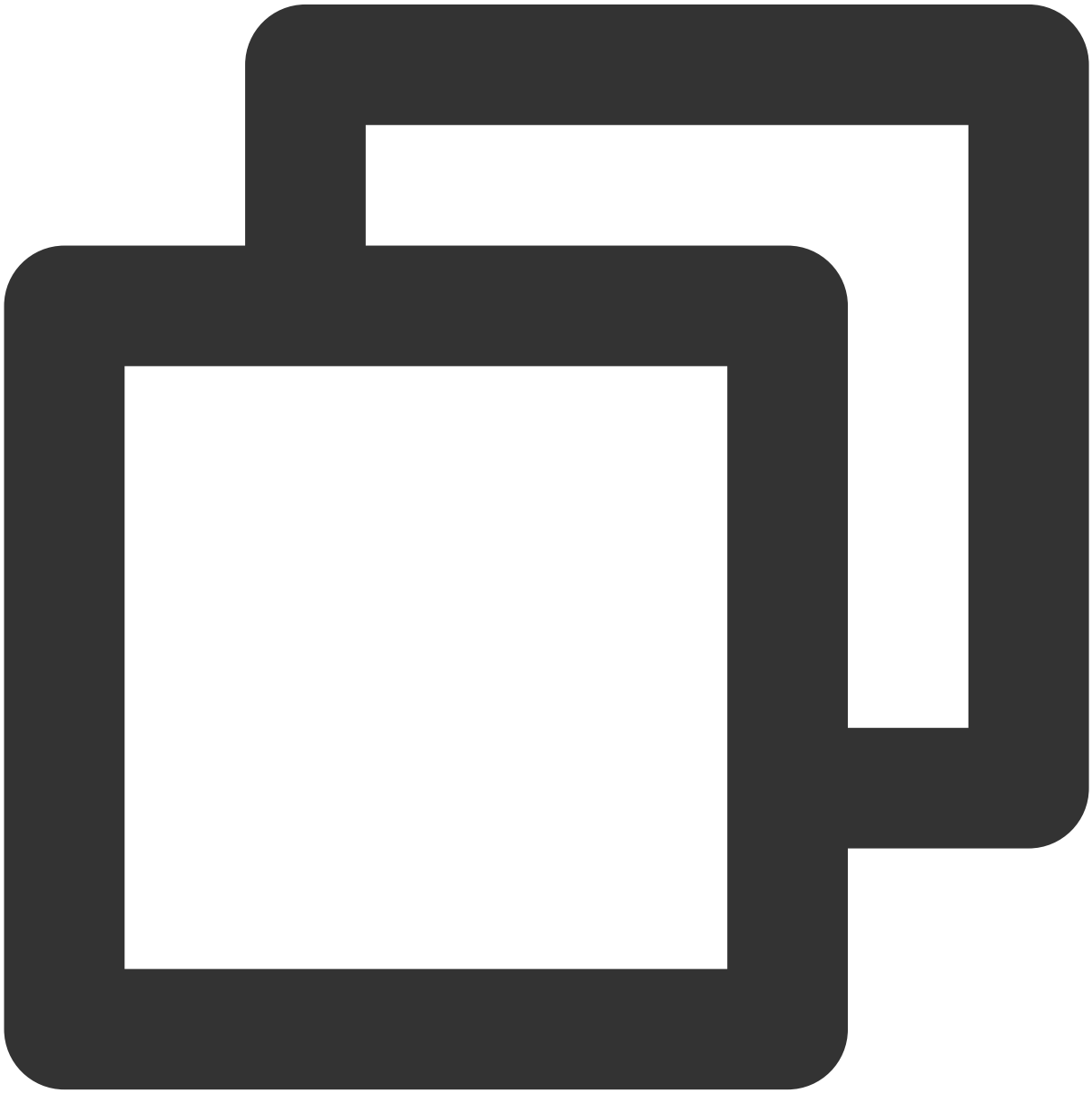
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

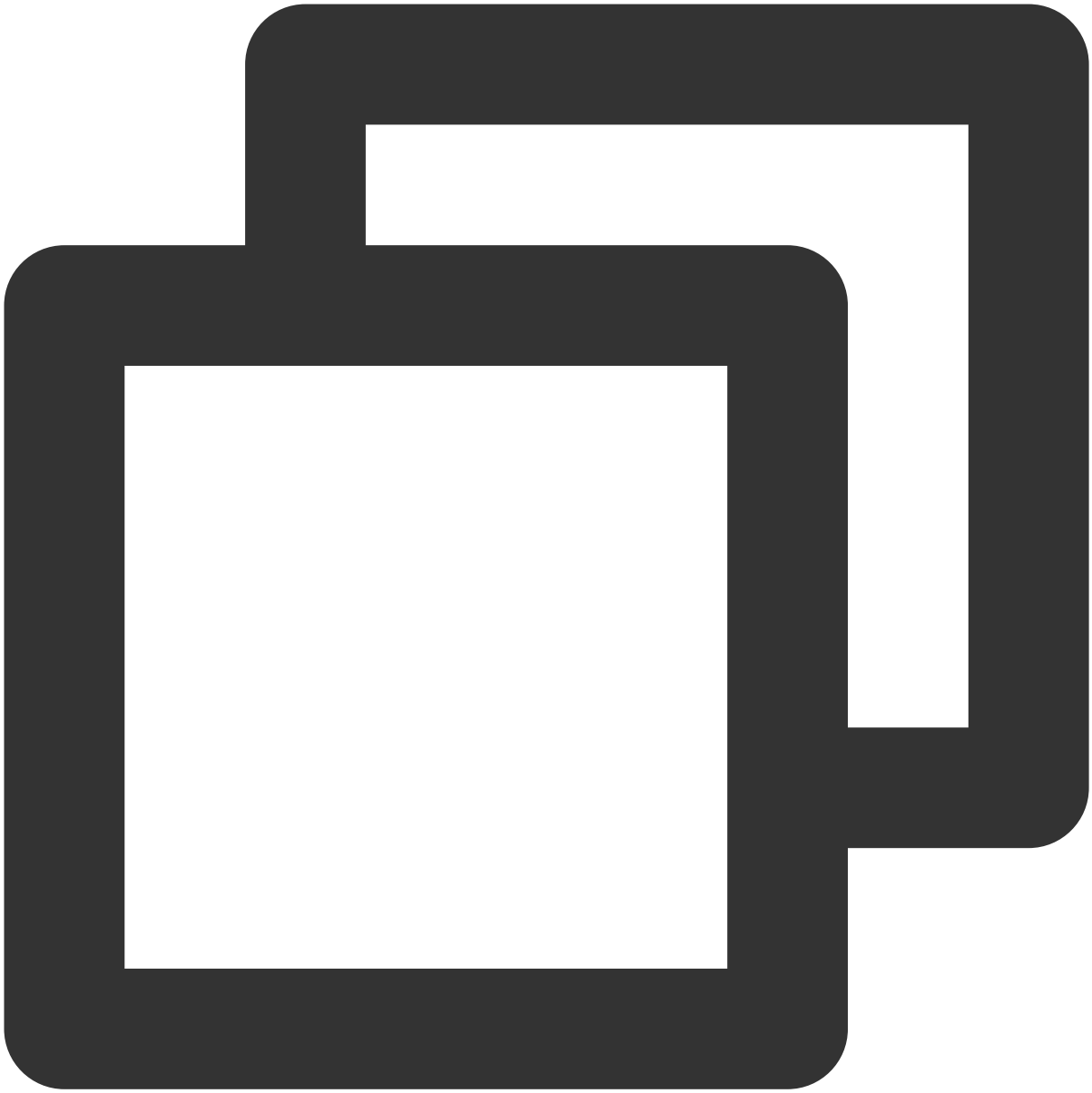
响应

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：

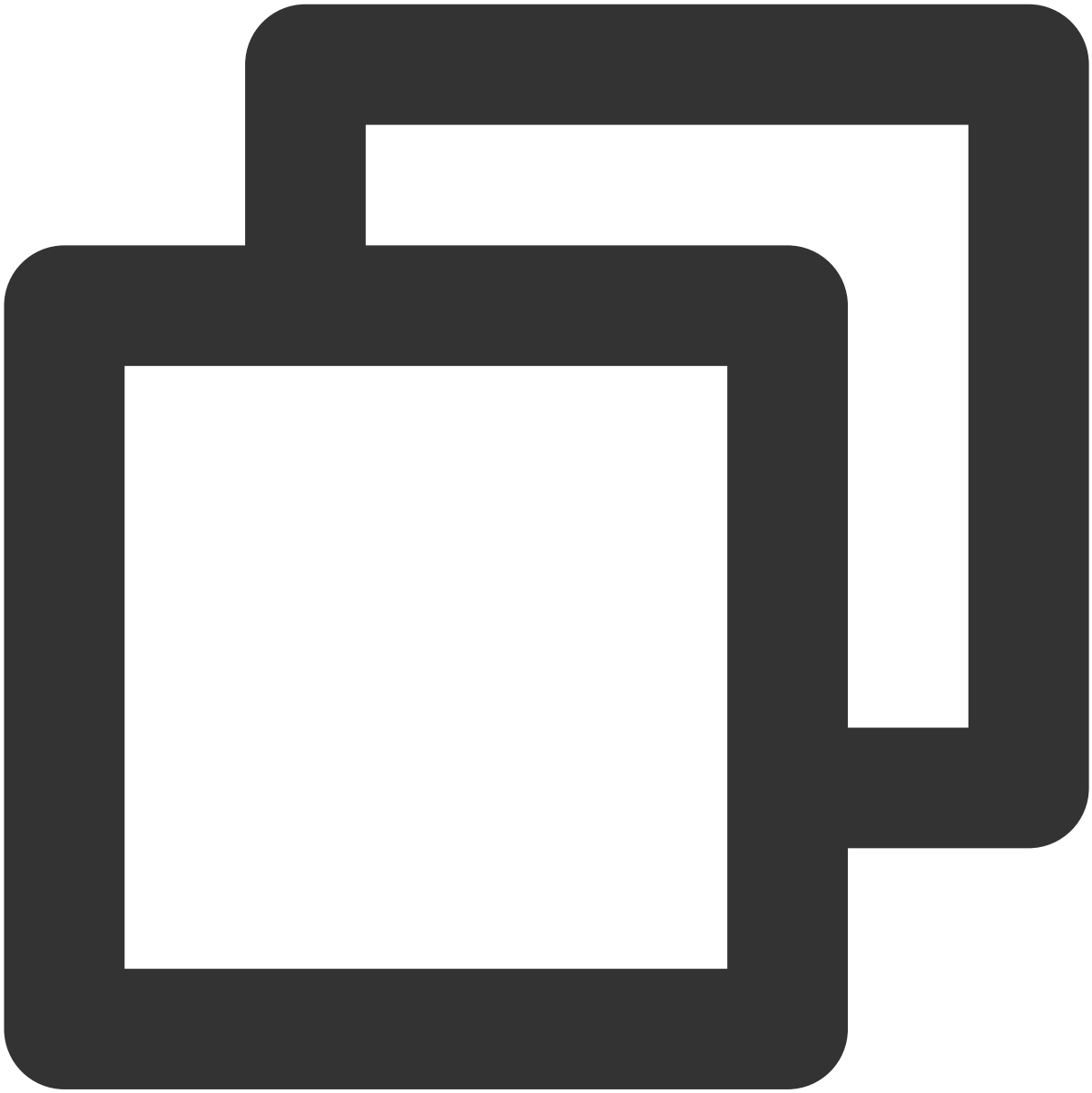
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
Queue	Response	队列信息，详情请参见 DescribeDocProcessQueues 中的 Response.QueueList	Container

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

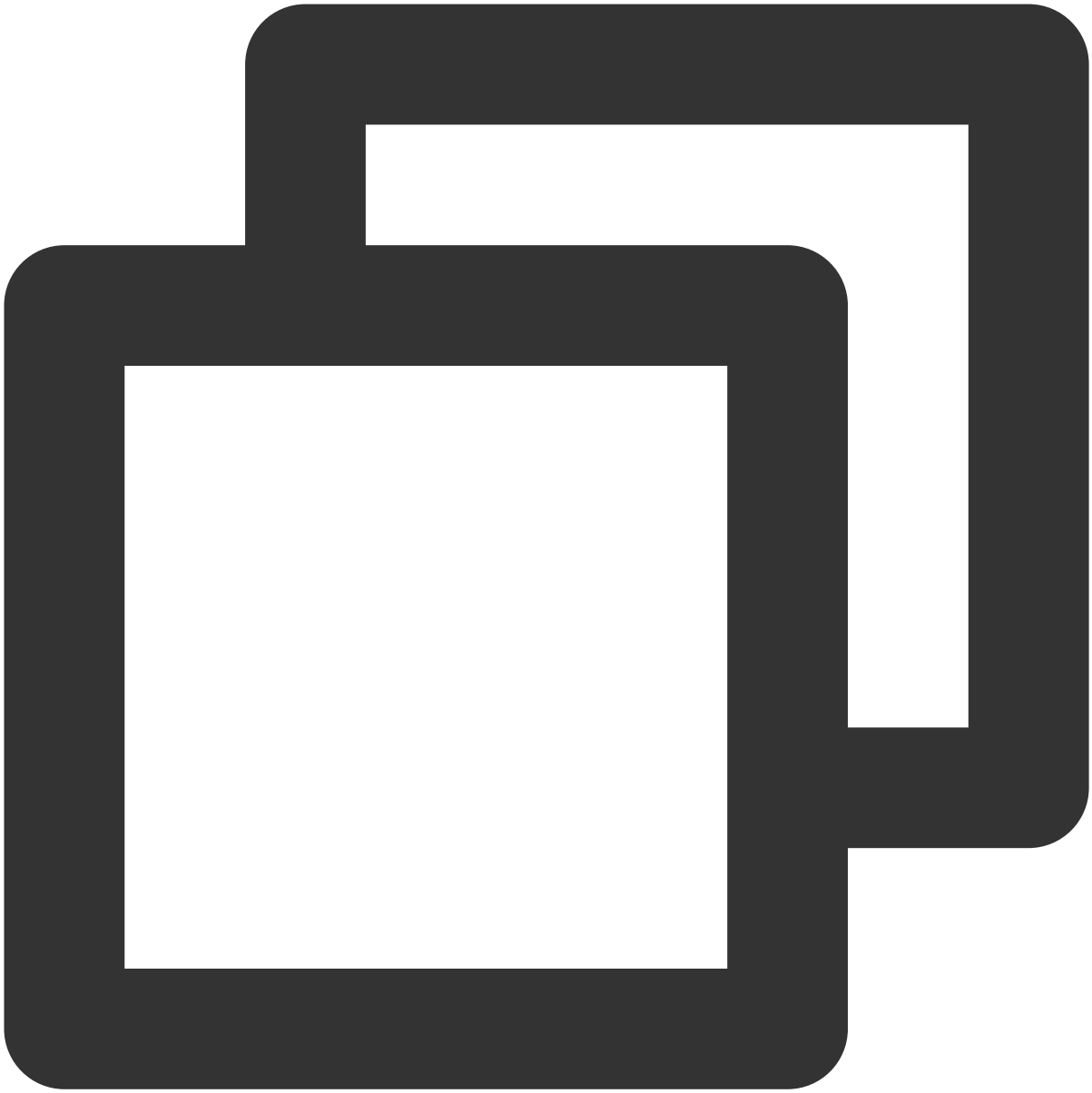


```
PUT /docqueue/p2505d57bdf4c4329804b58a6a5fb1572 HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: examplebucket-1250000000.ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Content-Length: 279
Authorization: Authorization

<?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>
                <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>
```

文档转 HTML 快速入门

最近更新时间：2024-01-20 11:39:30

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：

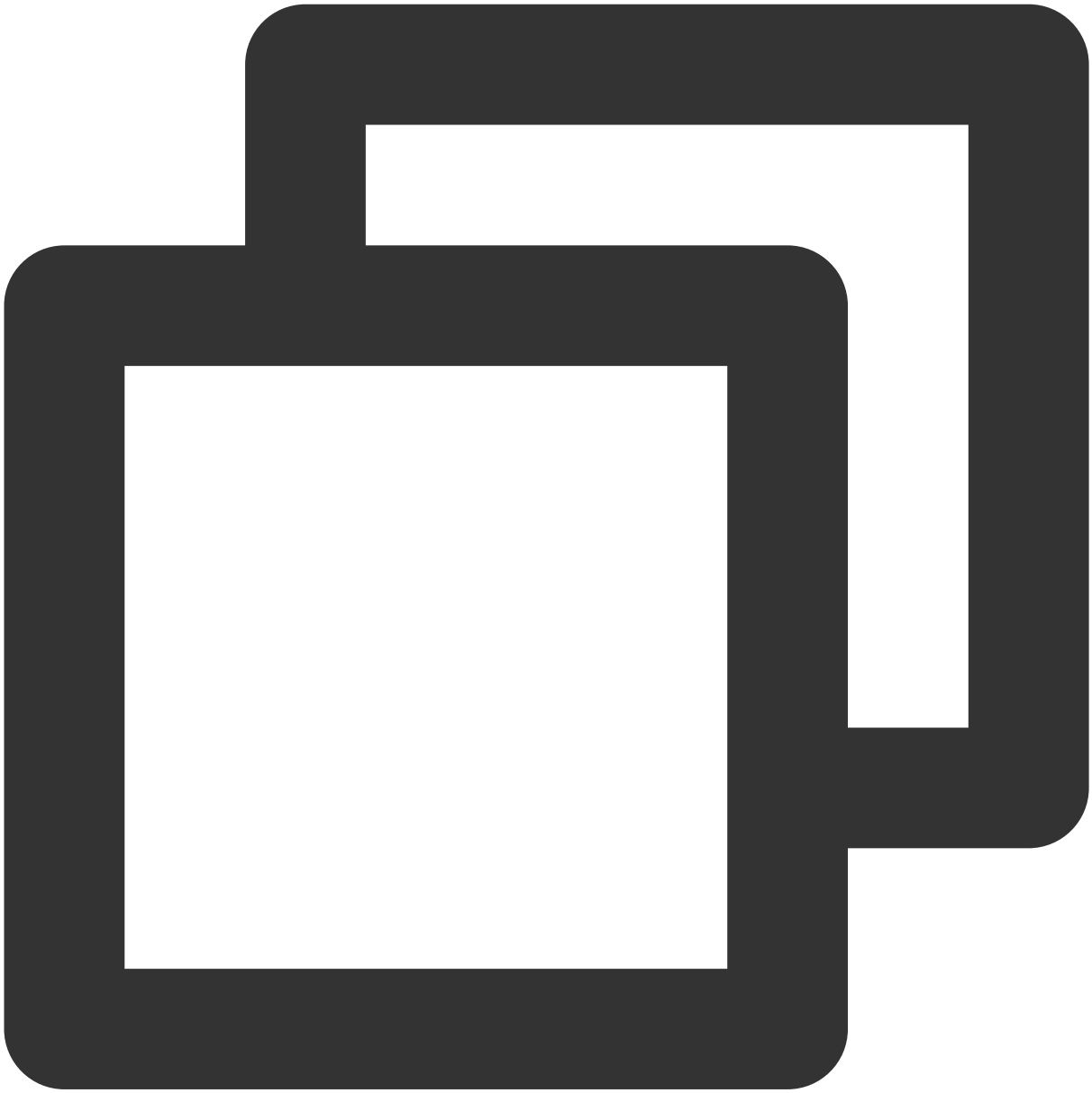
数据万象封装预览 JS SDK，只需要一条 URL 即可接入 HTML 预览服务。

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



```
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	数据万象处理能力，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。

版权所有：腾讯云计算（北京）有限责任公司

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。

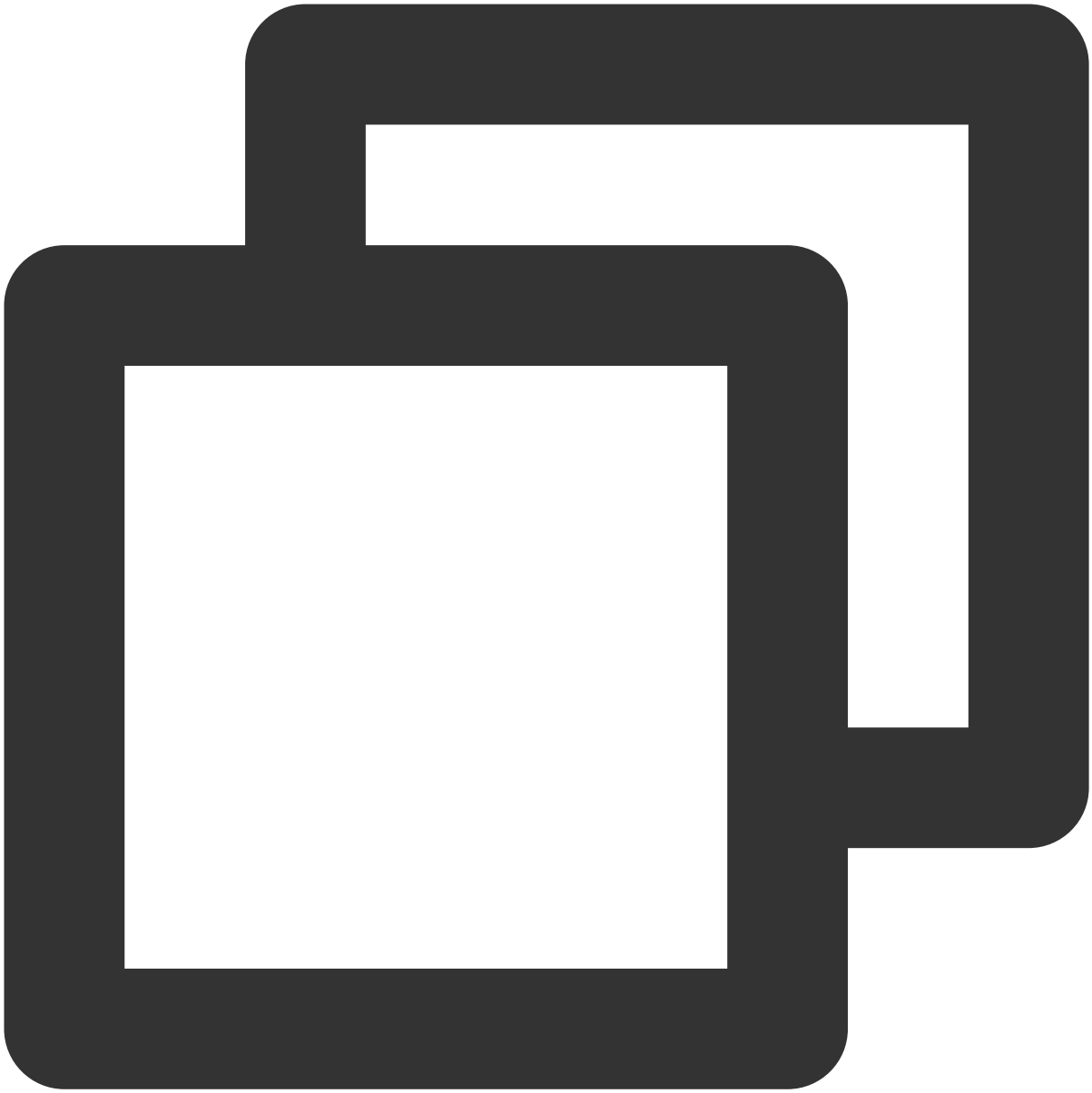
错误码

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。

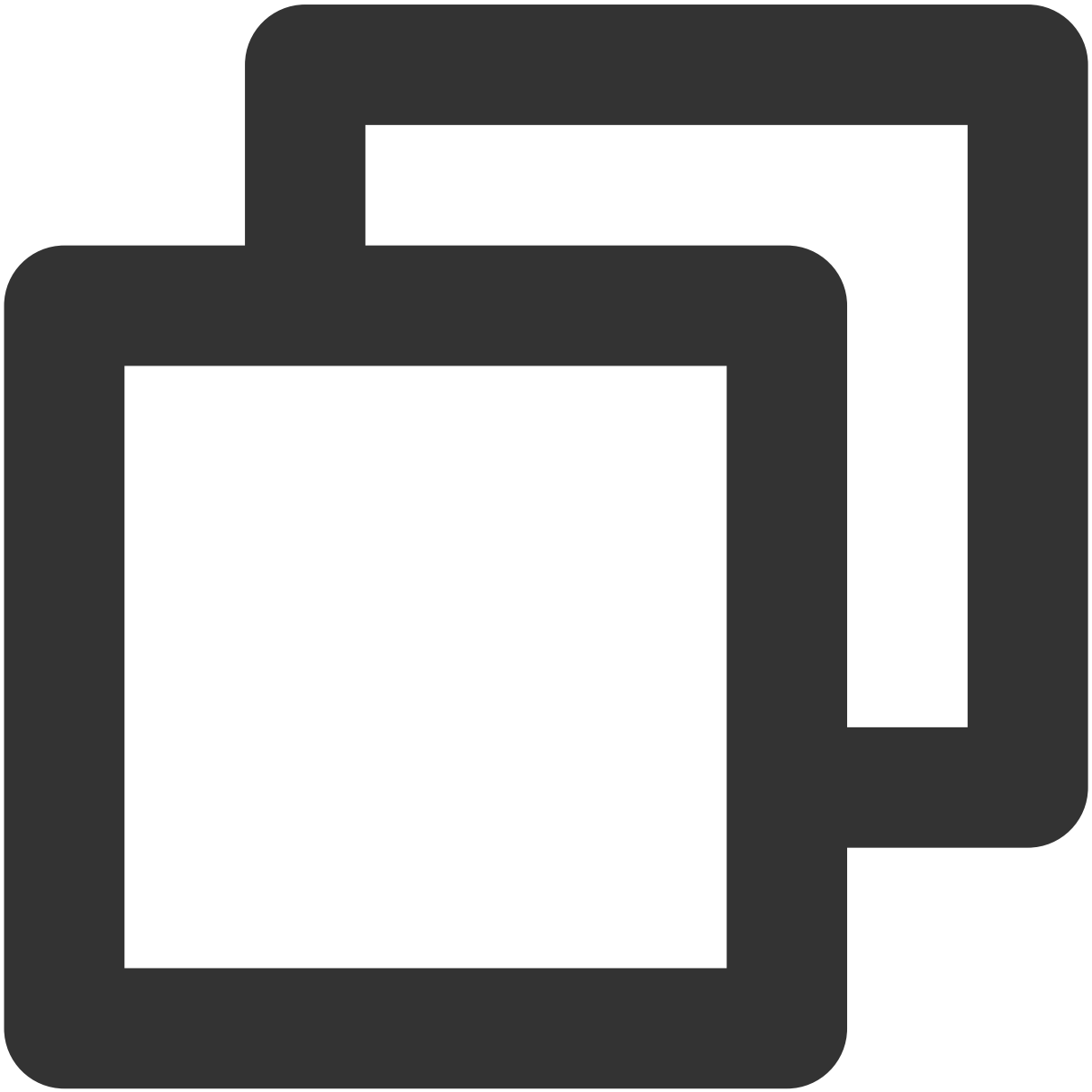
直接在浏览器中打开链接，生成的预览页面如下图所示，对于幻灯片文档，还可进行全屏的幻灯片放映，支持**动画、触发器**等效果，并提供**虚拟激光笔、演讲者模式**等功能。

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=）。

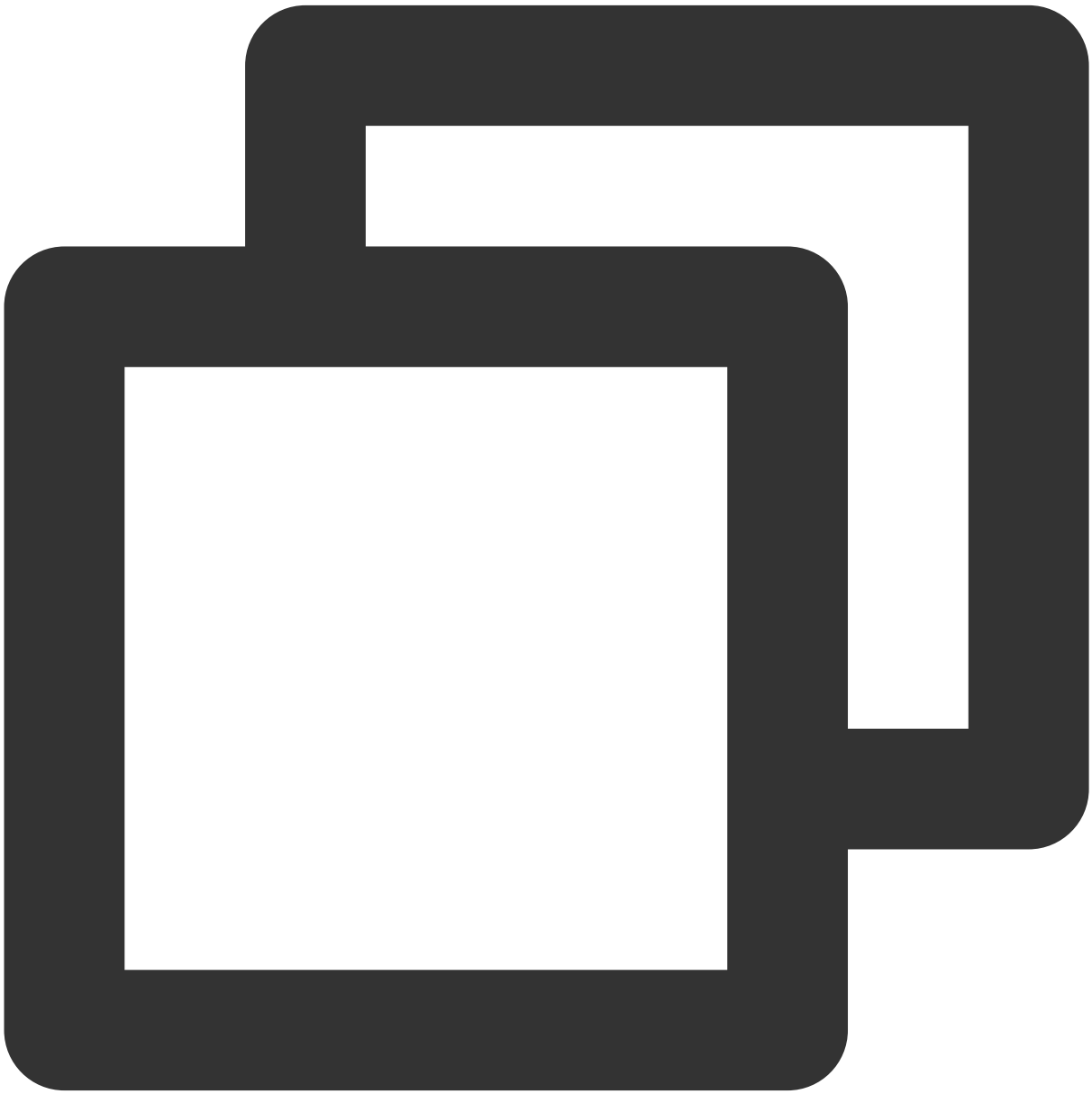
则最终 URL 如下：



```
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>
```

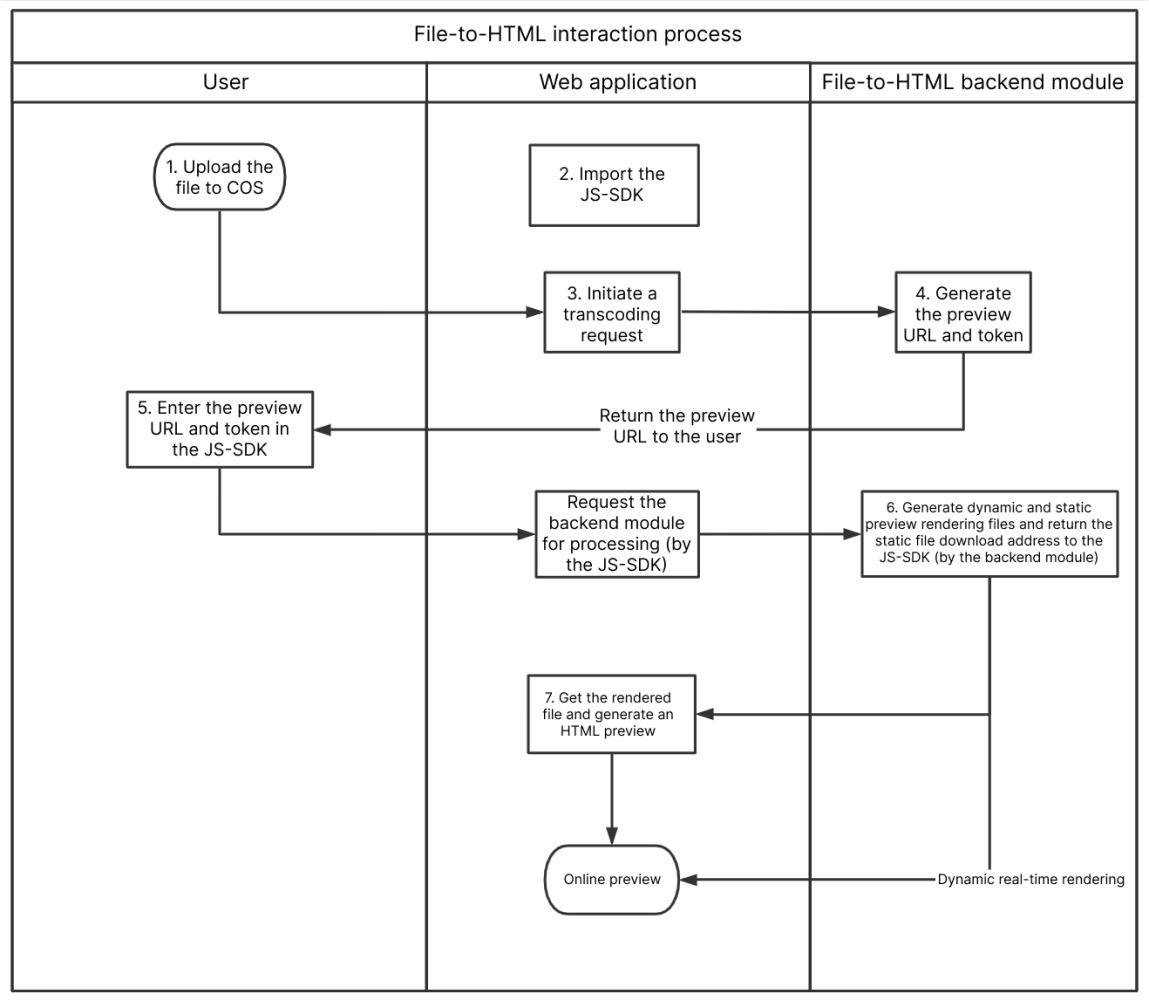
自定义配置概述

最近更新时间：2024-01-20 11:39:30

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

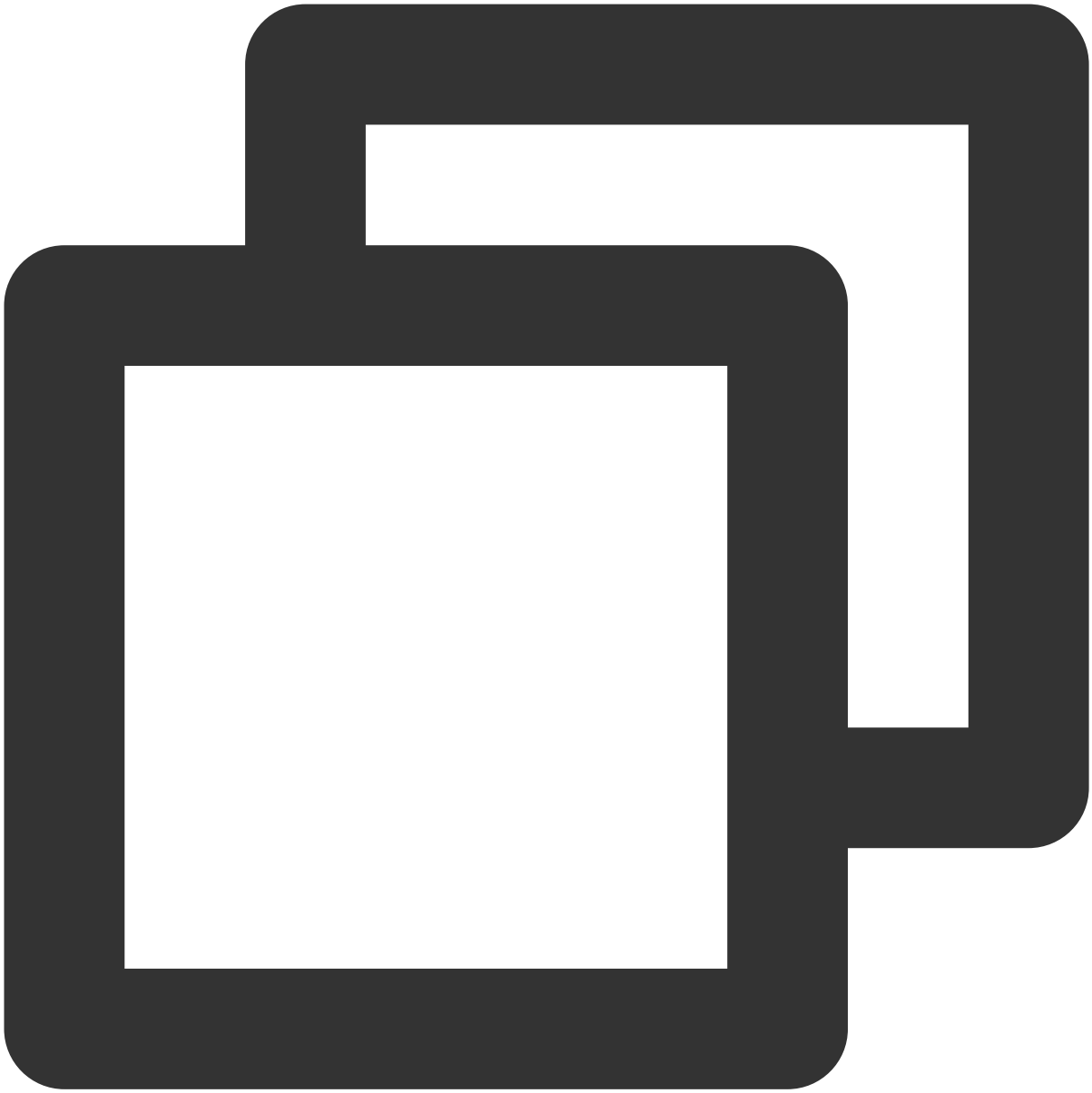
使用说明

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。

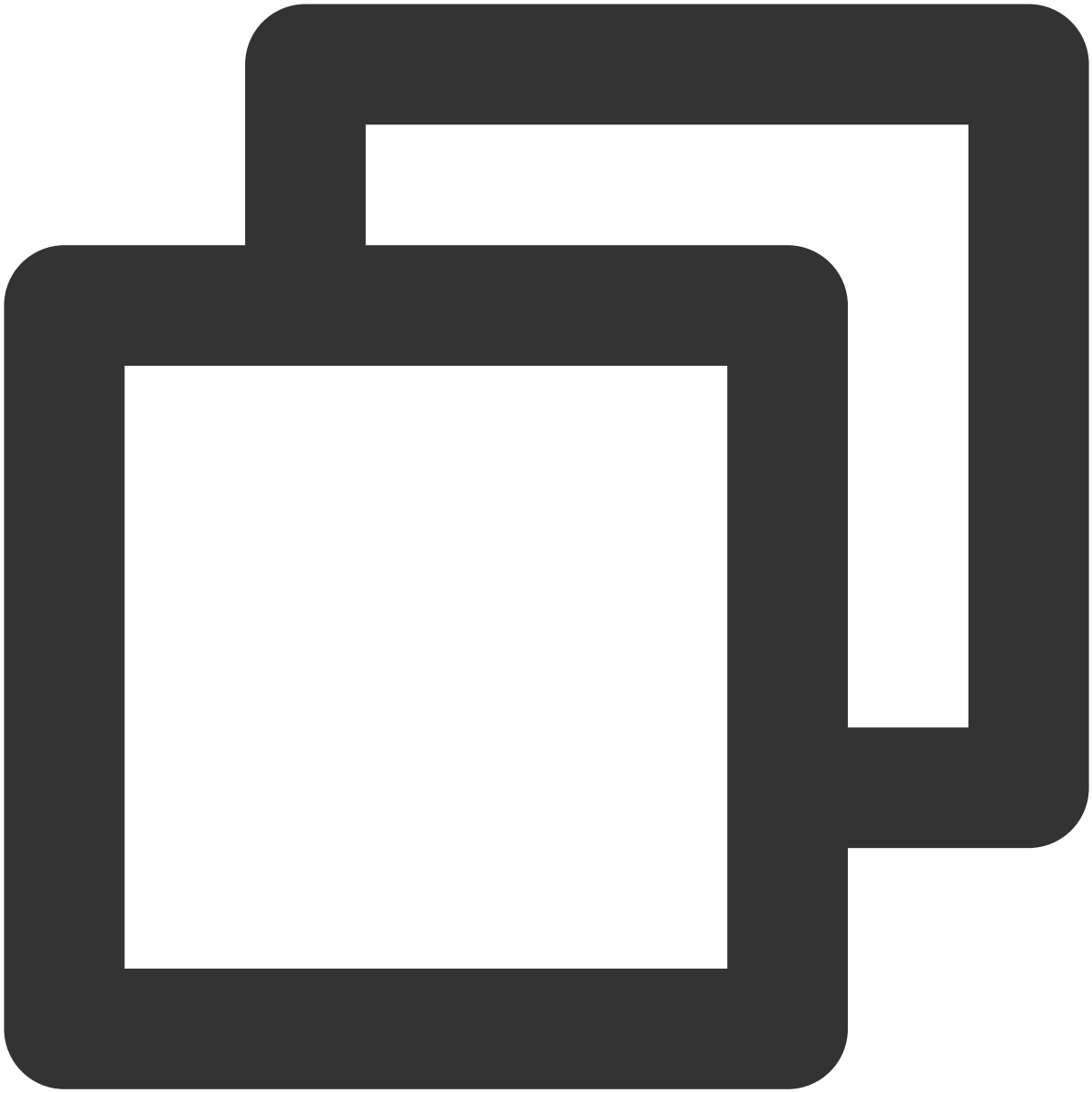


```
<script src="sdk-${version}.js"></script>
```

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：



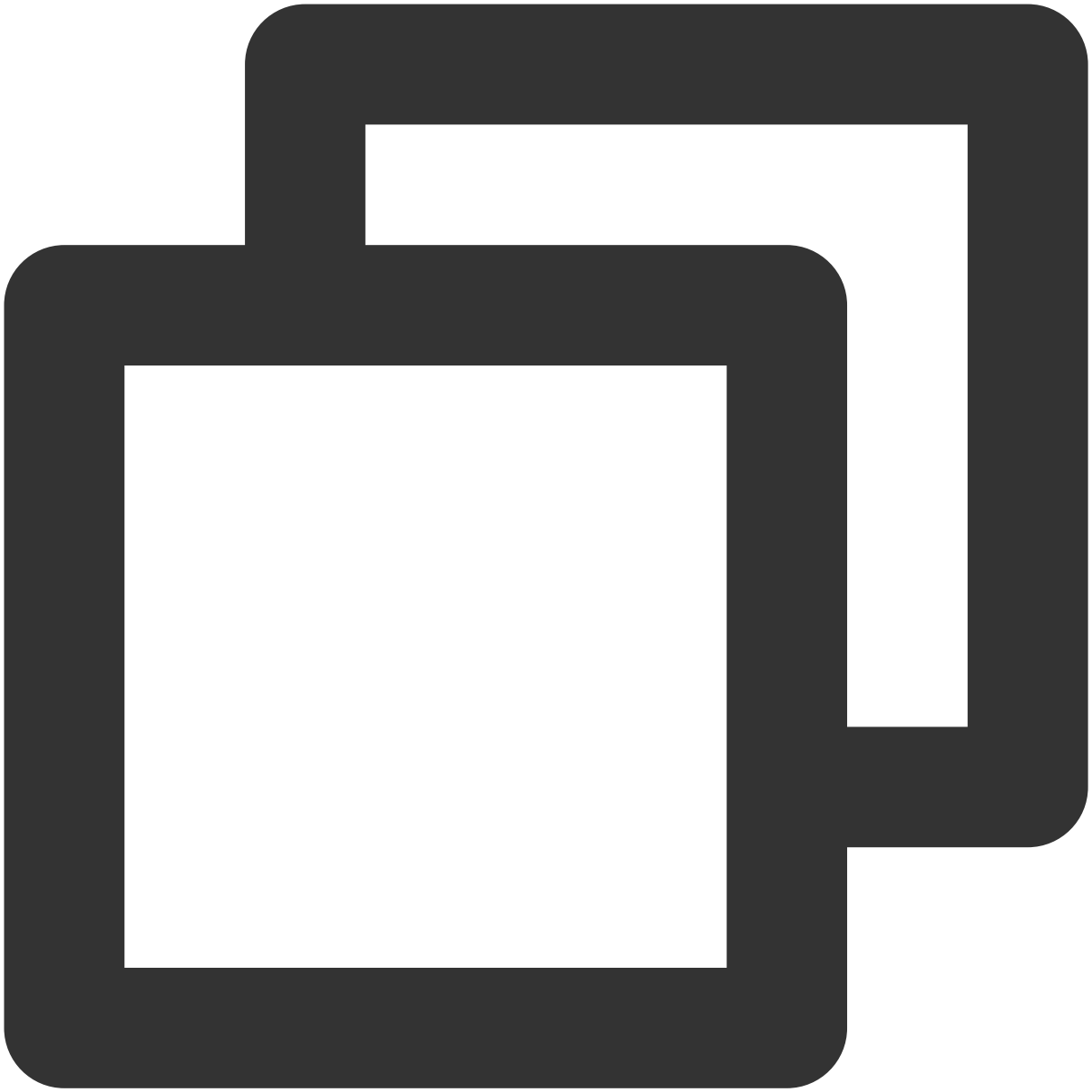
```
var url = await COSDocPreviewSDK.getPreviewUrl( {
      objectUrl: 'https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/tes
      credentials: {       // 如果预览文件是私有的，则需要携带访问凭证。
          secretId: '',
          secretKey: '',
          authorization: 'q-sign-algorithmxxxxxxxxxxxxxxx',
      },
      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	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
```

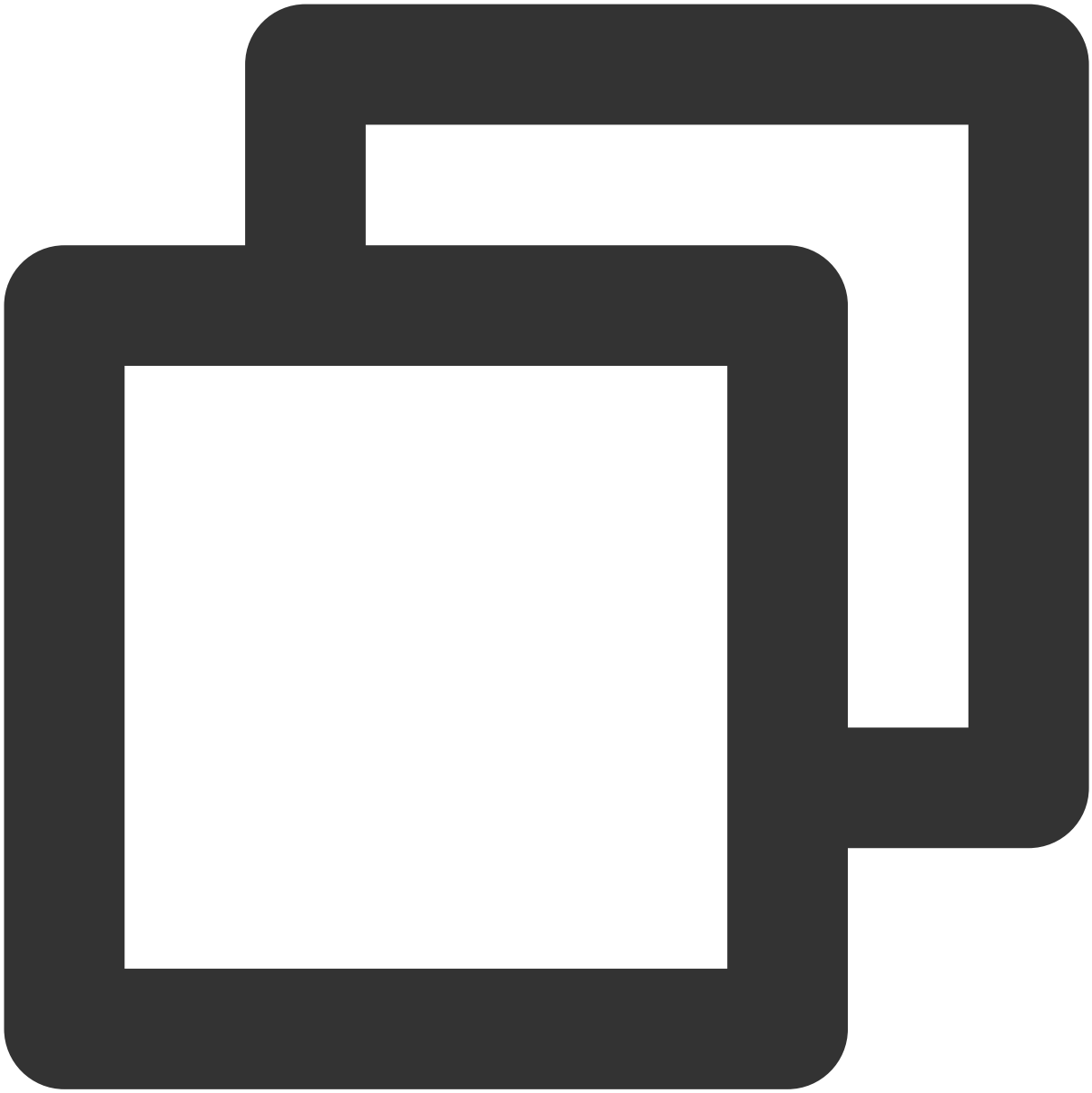
请求参数

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	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

版权所有：腾讯云计算（北京）有限责任公司



```
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****

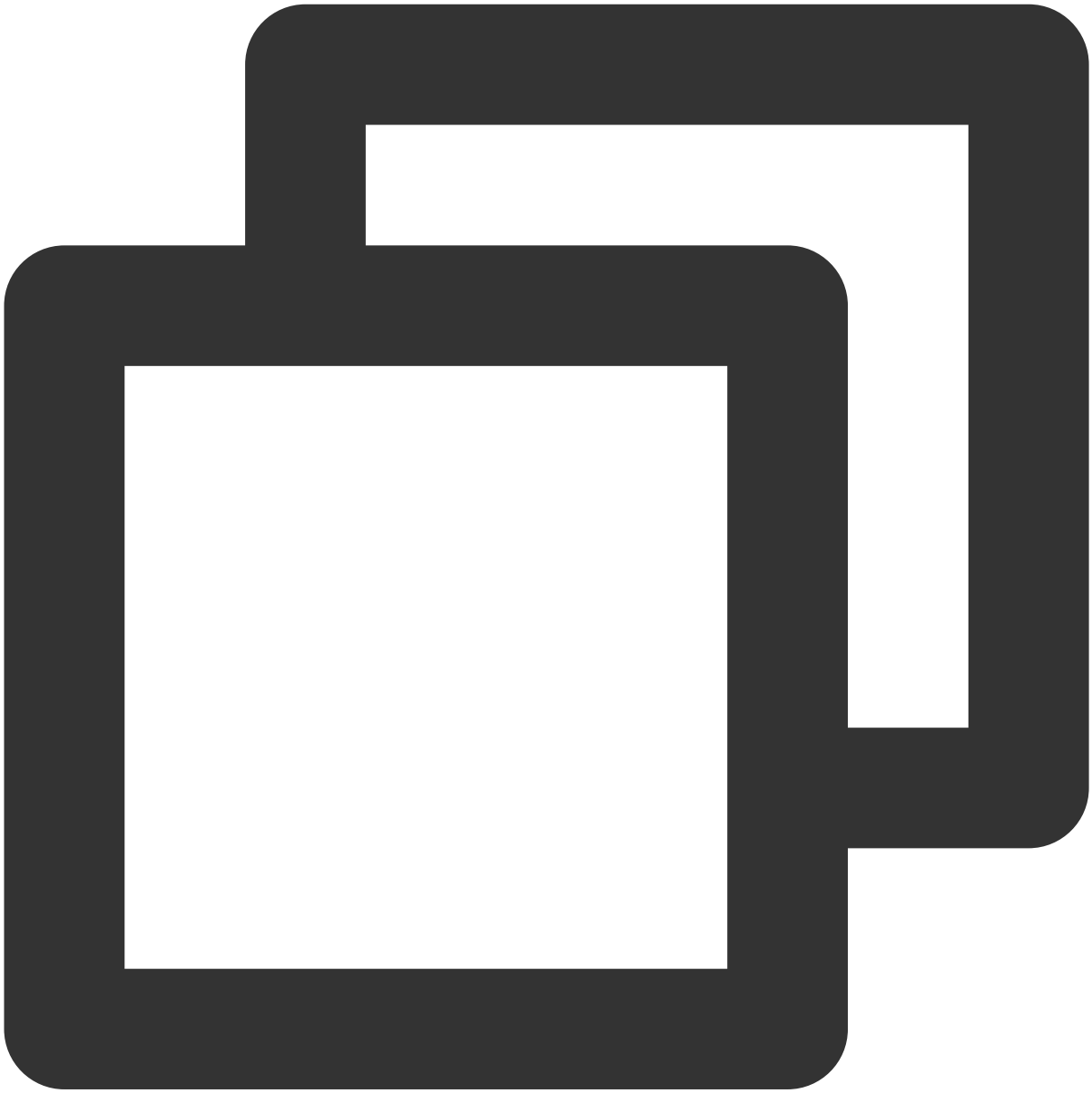


{
        "PreviewUrl": "https://prvsh.myqcloud.com/office/p/xxxx?_w_provider=XXX"
```

```
}
```

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。

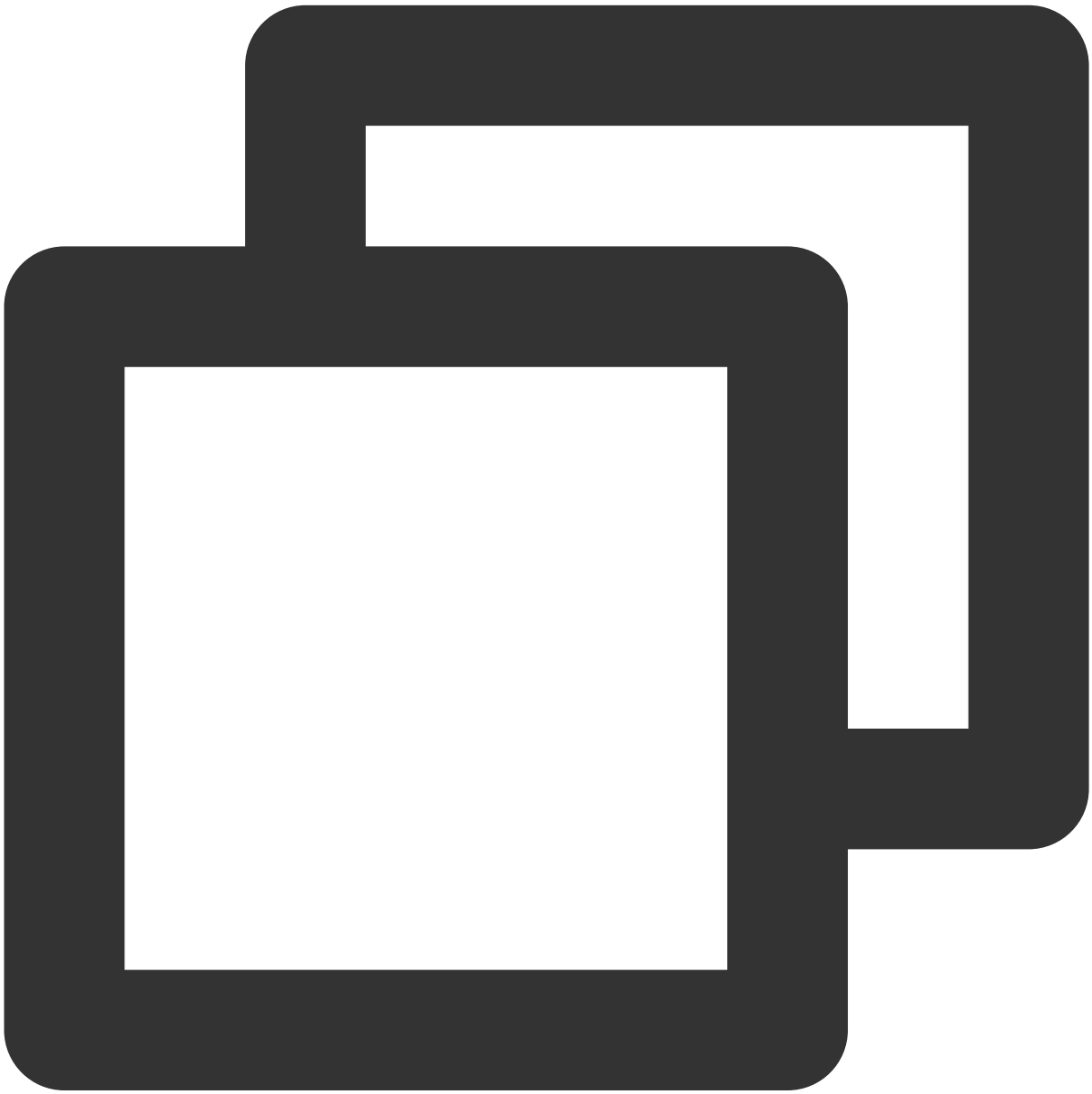


```
var demo = COSDocPreviewSDK.config({
    url: '在线文档预览地址',  // 步骤3中获取到的在线文档预览地址
})
// 如果需要对 iframe 进行特殊的处理，可以通过以下方式拿到 iframe 的 dom 对象
console.log(demo.iframe)
```

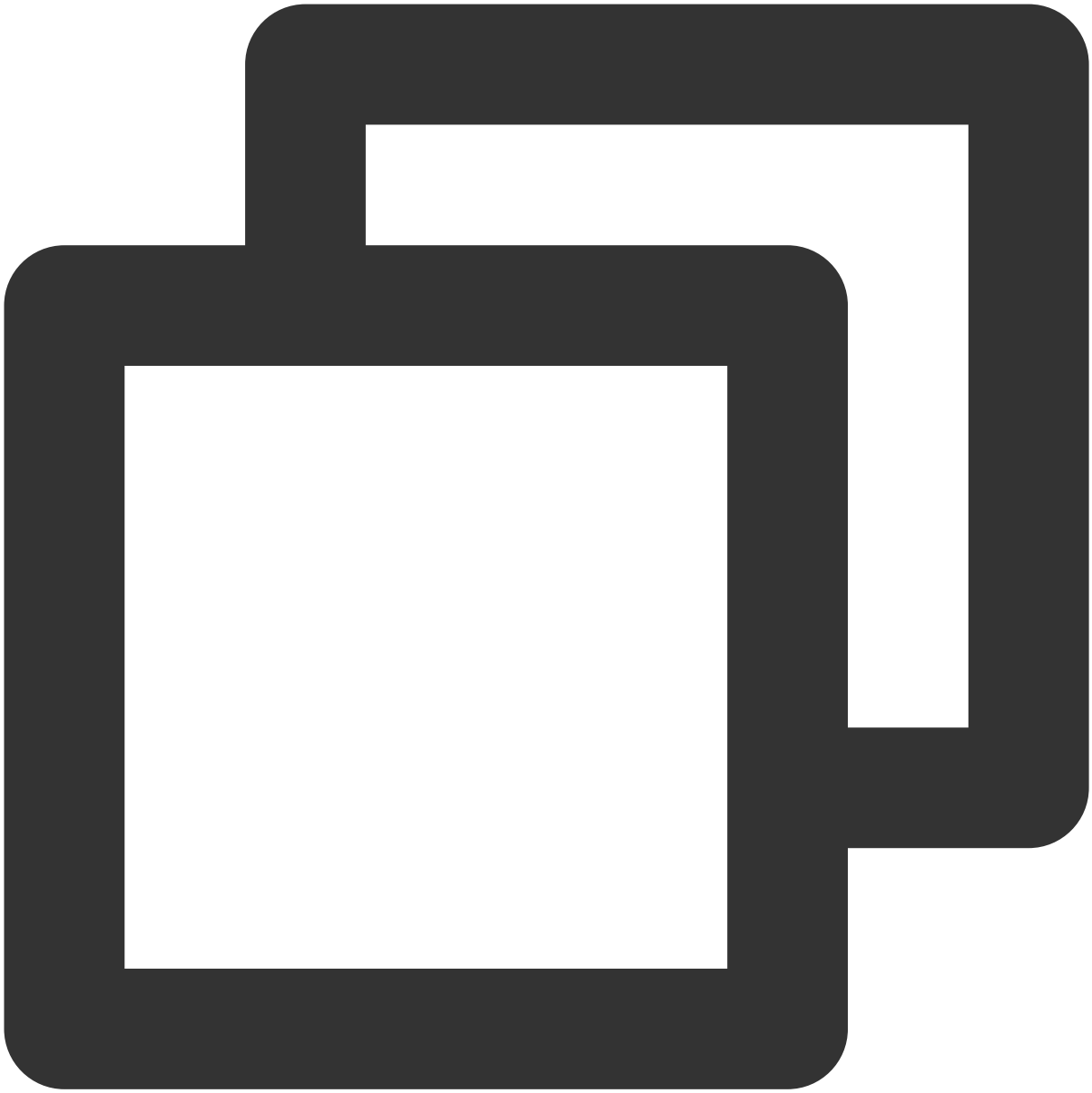
```
// 打开文档结果
demo.on('fileOpen', function(data) {
    console.log(data.success)
})
```

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。

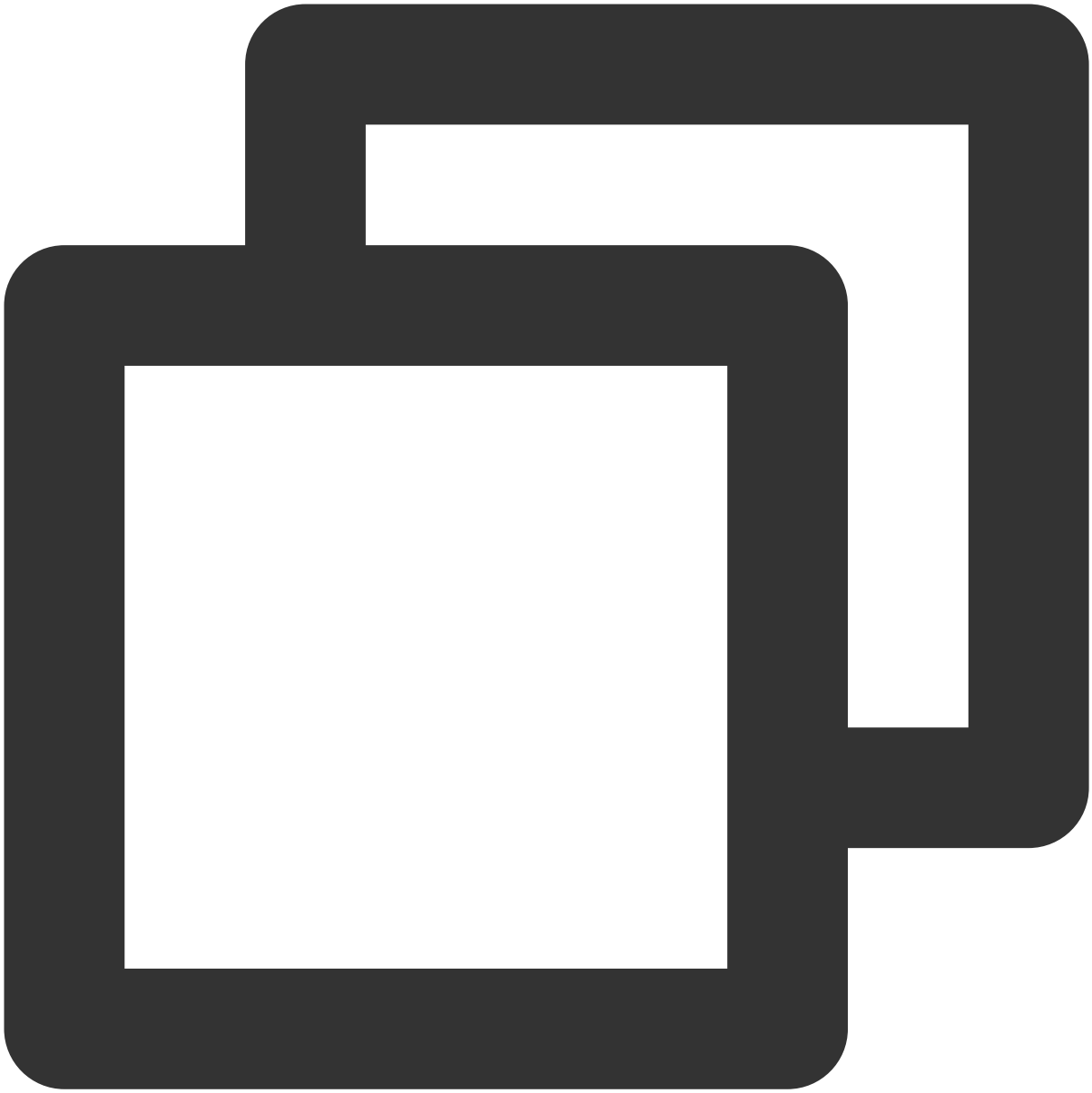


```
<div class="custom-mount"></div>
```



```
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
})
```

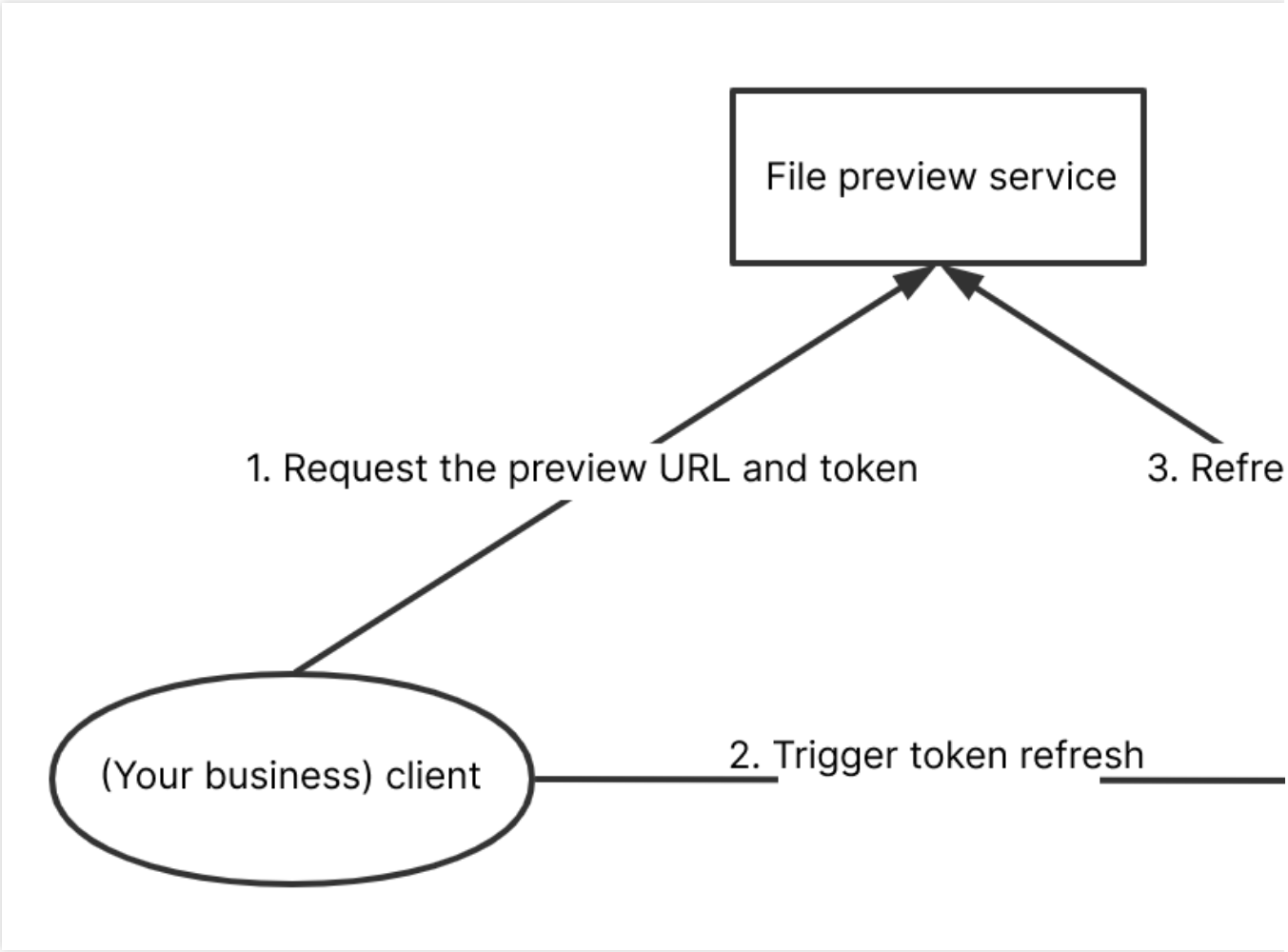
JS-SDK 初始化完成后会自动在挂载节点下面插入一个 iframe。



```
<div class="custom-mount">
    <iframe src="..."></iframe>
</div>
```

6. Token 鉴权

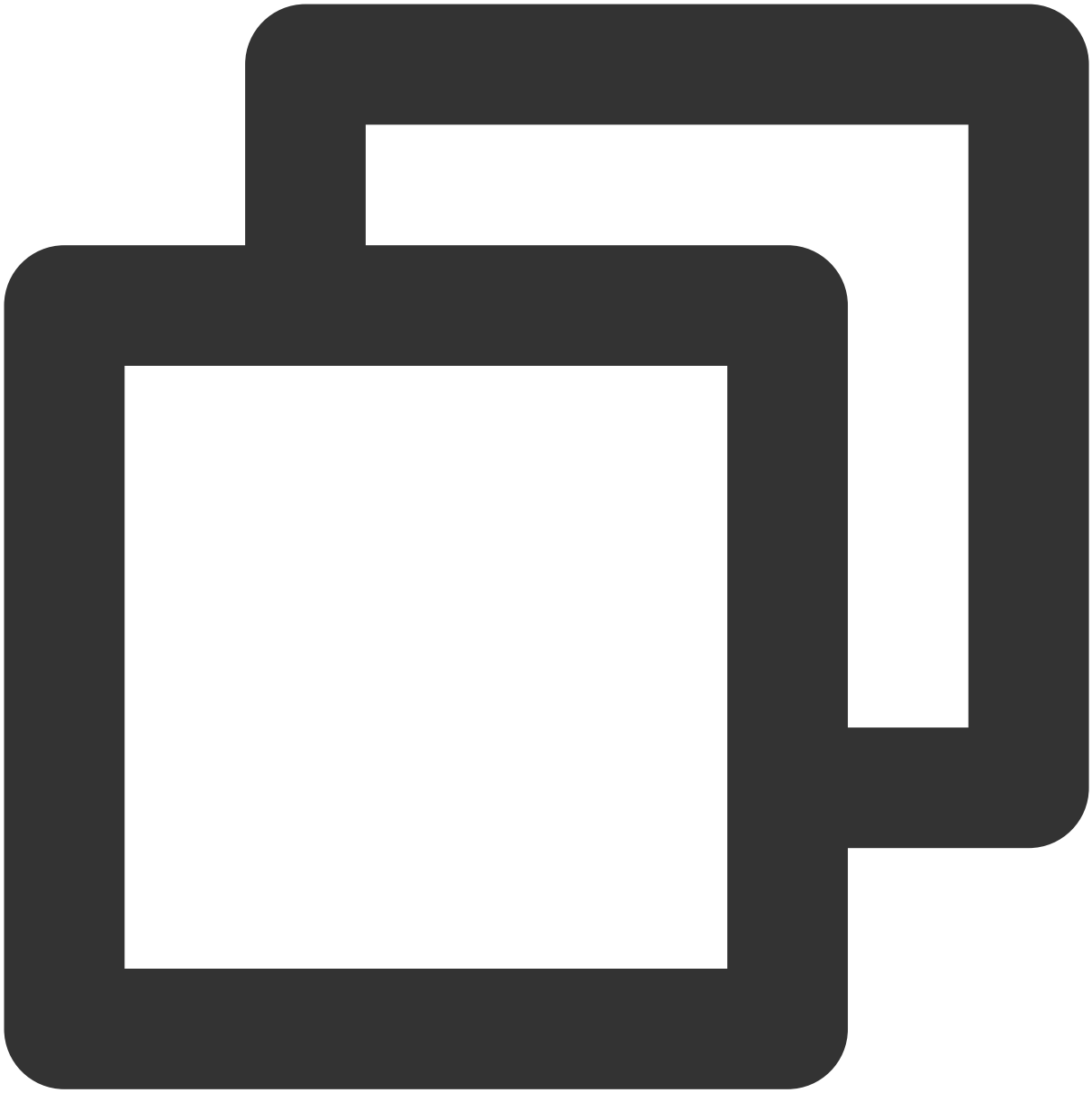
不使用 Token 时，默认使用文档的签名有效期鉴权，过期则无法预览文件。
使用 Token 鉴权可以在预览期间持续进行 Token 鉴权和 Token 续期，由业务侧自主控制预览授权。



Token 鉴权与续期分为以下三个步骤，详情请参见 demo 示例 。

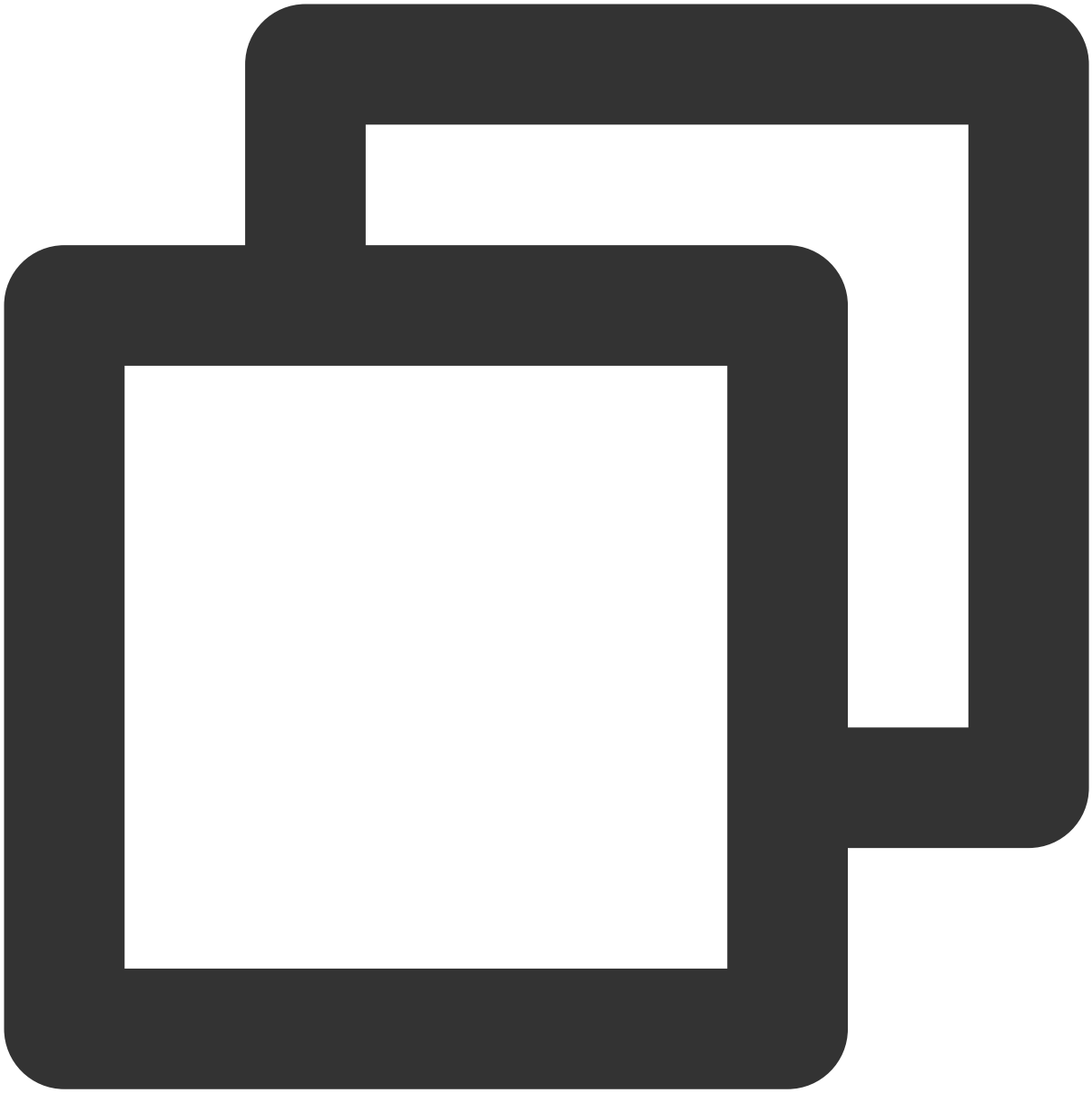
1. 获取 Token

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****



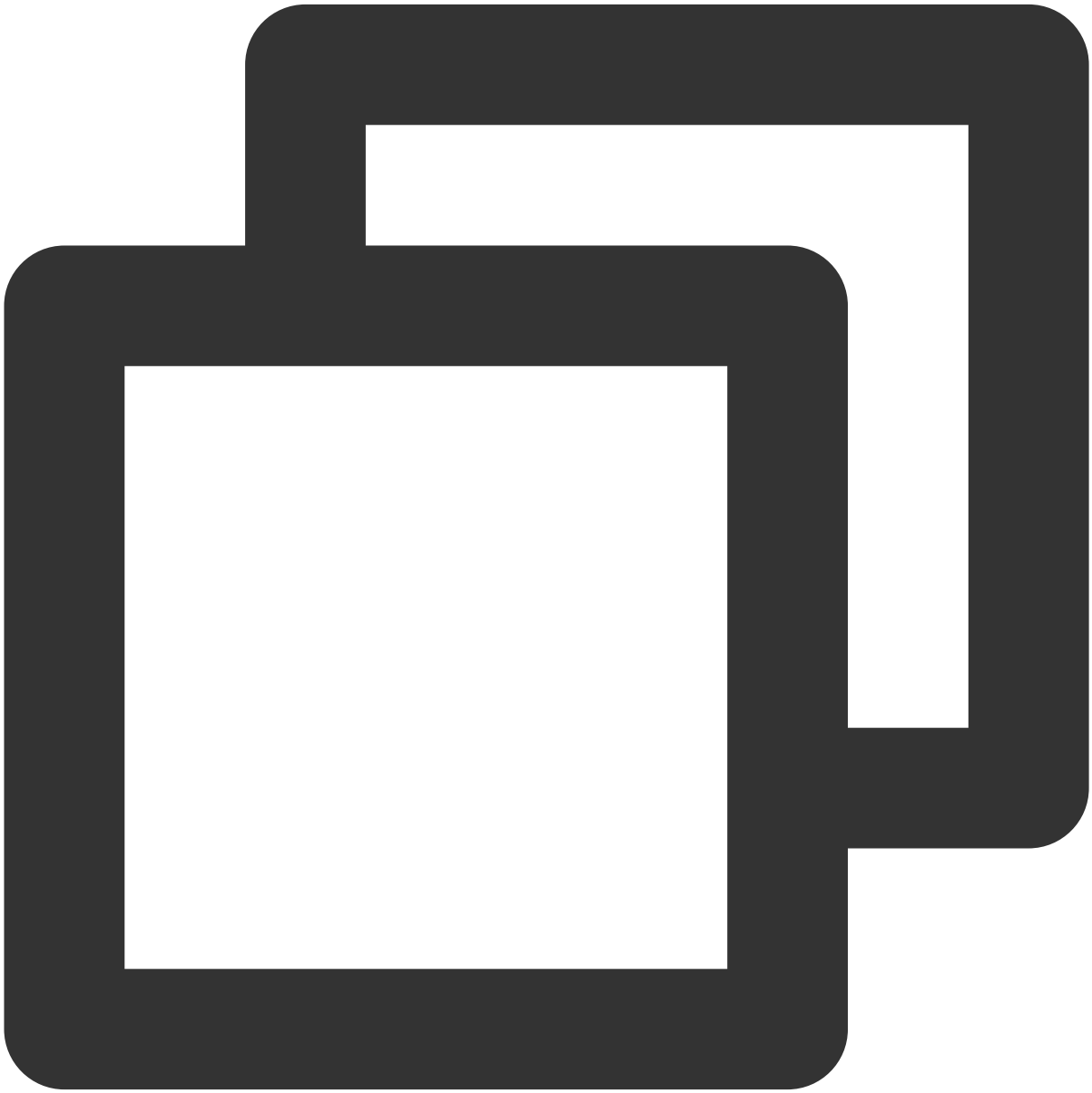
{
 Token: "ChAJj2vNRiHTc8reToMmJ7T2EJ7FxpUGGhS1mPJanpN03u8PSB17/YKBA/Vo****"
```

```
 TokenExpireTime: "1655808670"
}
```

2. 设置 Token

说明：

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。



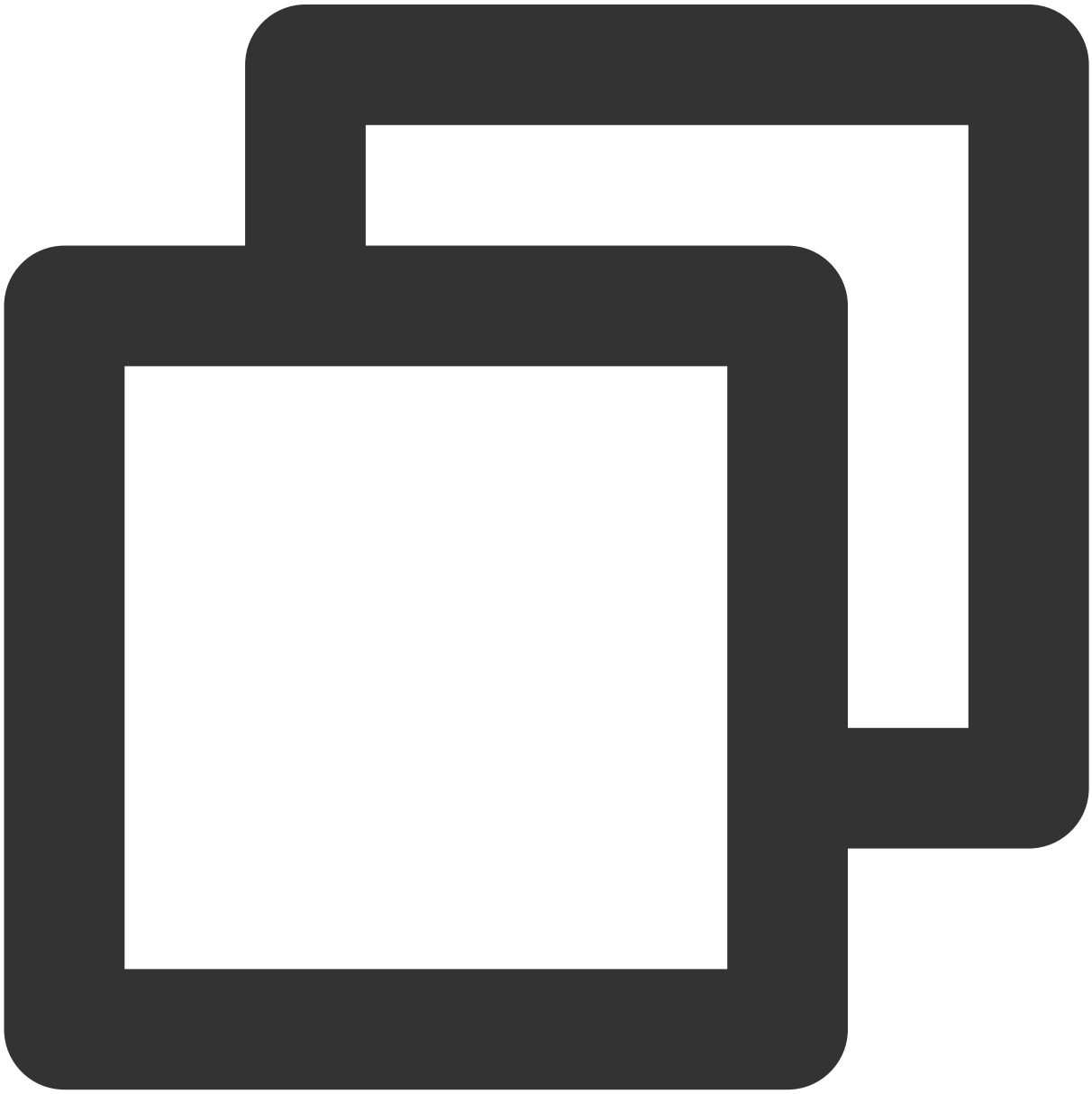
```
var demo = COSDocPreviewSDK.config({
    url: '在线文档预览地址',  // 步骤3中获取到的在线文档预览地址
})
```

```
// 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，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

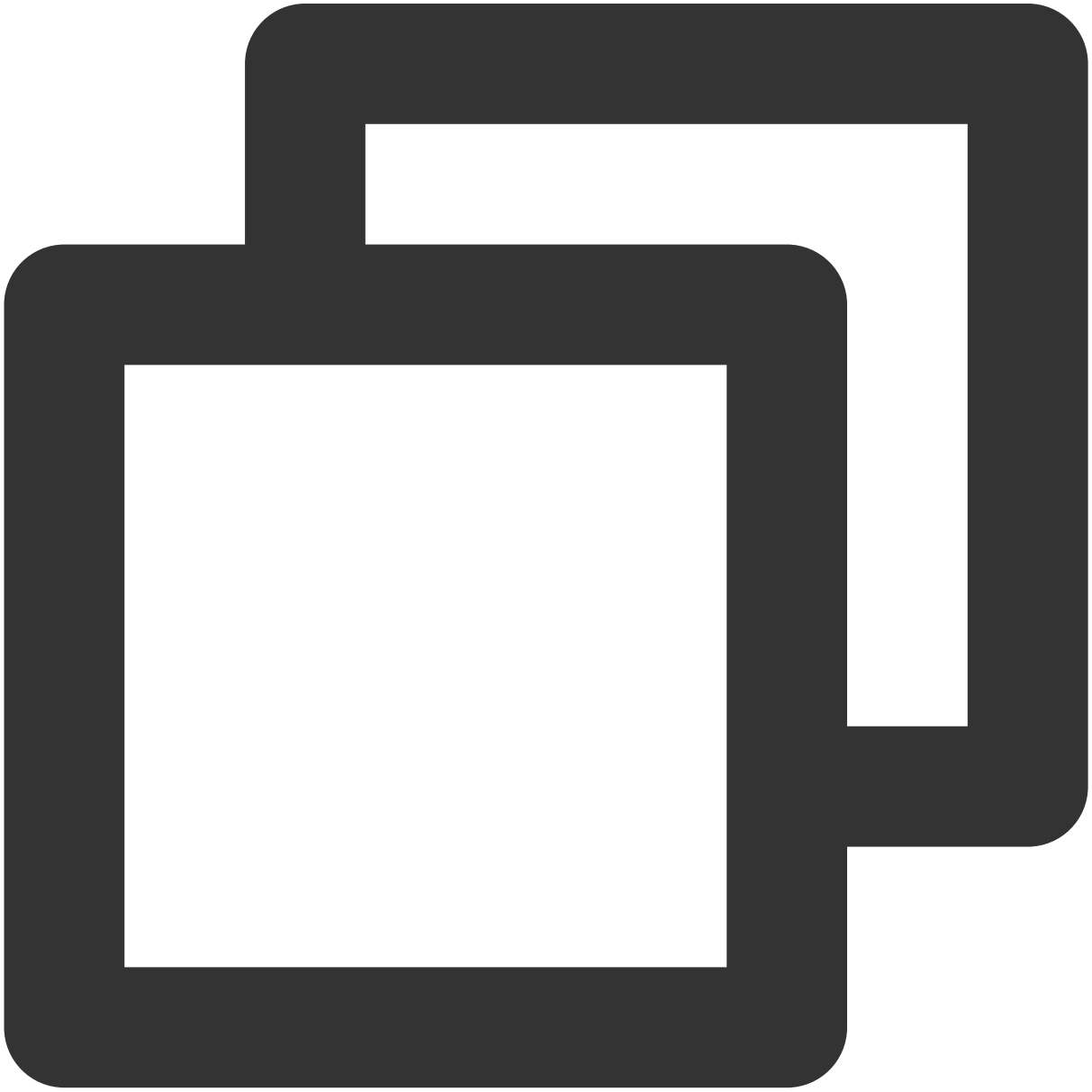
文字文件在线预览案例，参考 DOC 预览案例 。
演示文件在线预览案例，参考 PPT 预览案例 。
表格文件在线预览案例，参考 Excel 预览案例 。
PDF 文件在线预览案例，参考 PDF 预览案例 。
Token 鉴权与续期案例，参考 Token 鉴权与续期案例 。

配置项说明

最近更新时间：2024-01-20 11:39:30

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({
```

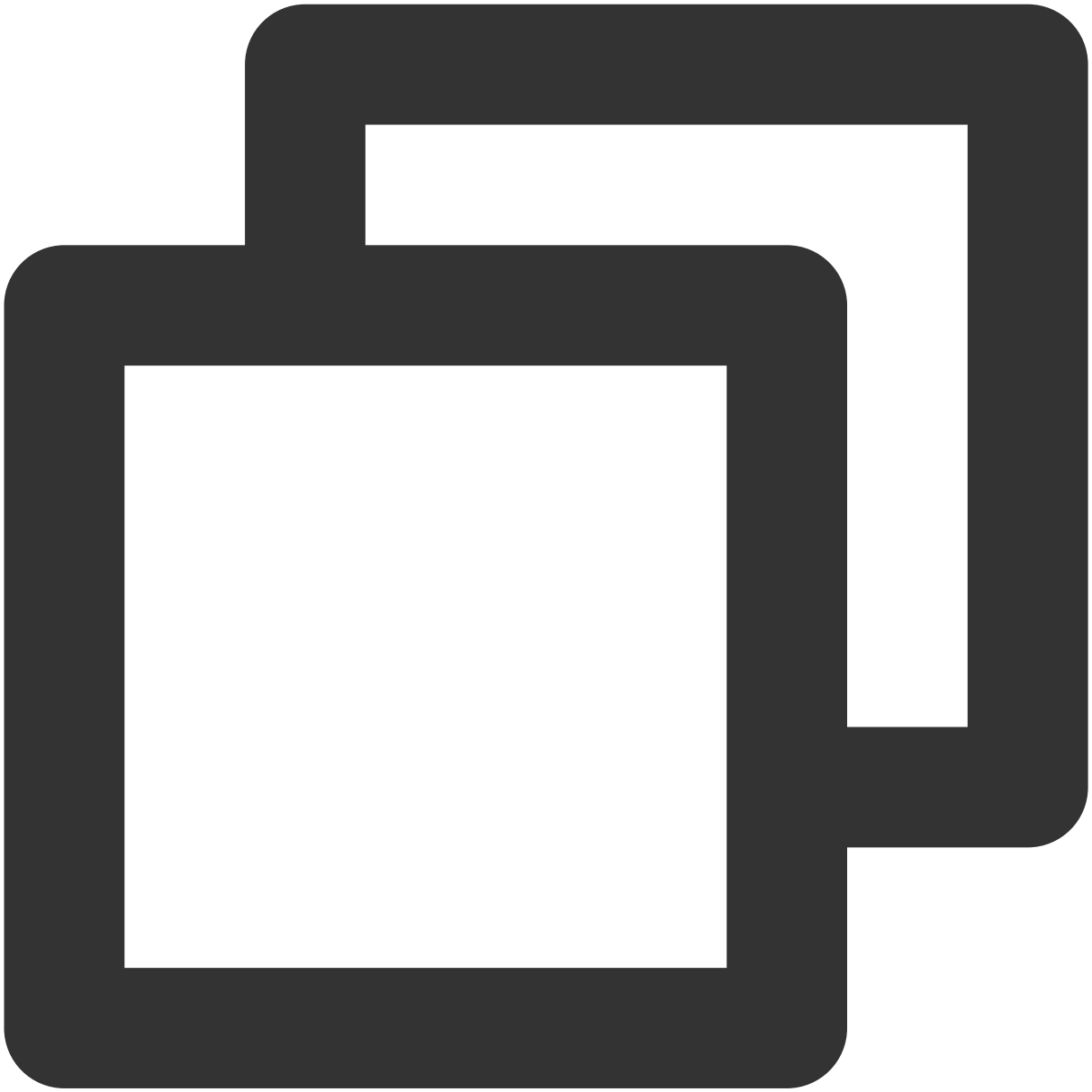
```
    mode: 'normal',
})
```

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

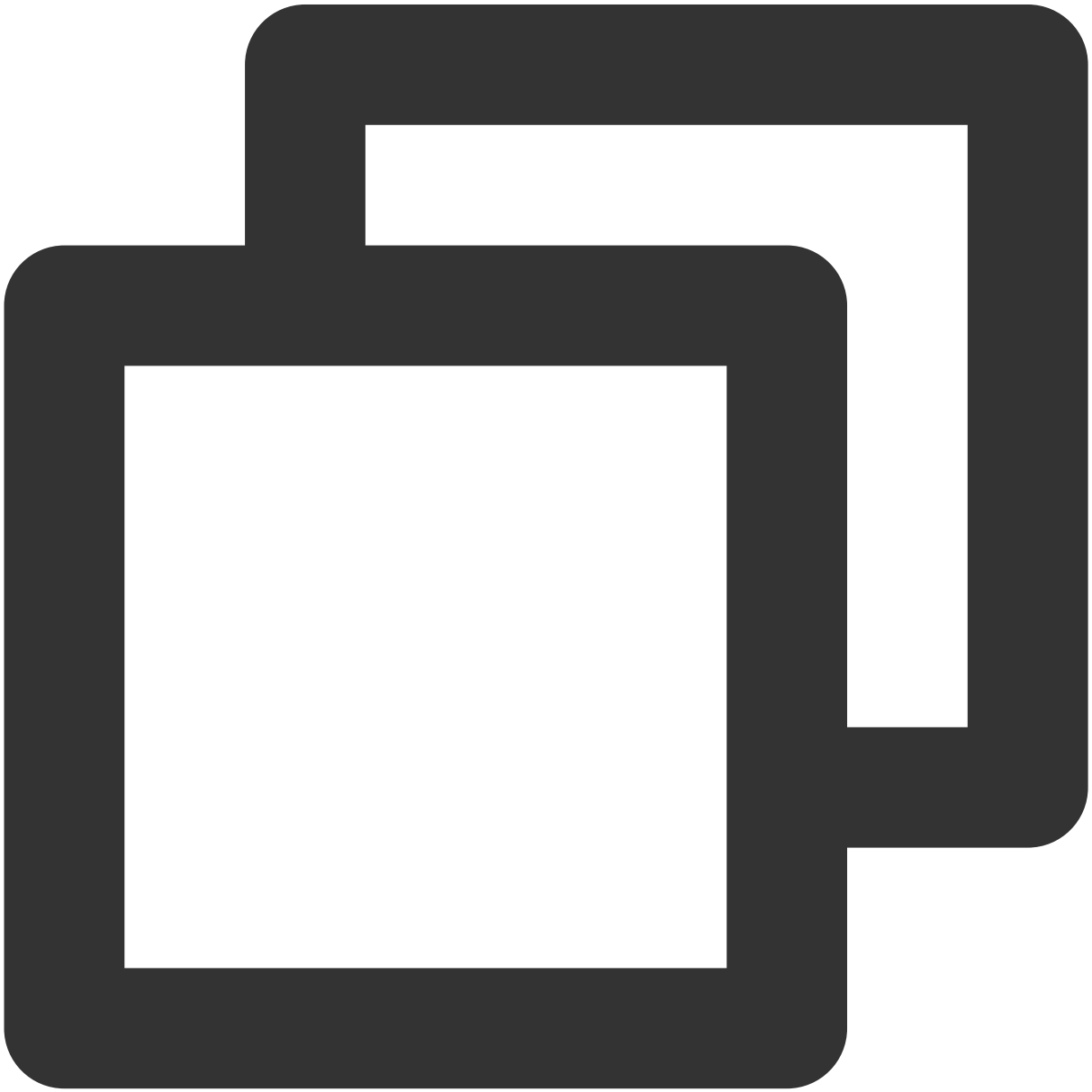
版权所有：腾讯云计算（北京）有限责任公司

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。



```
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

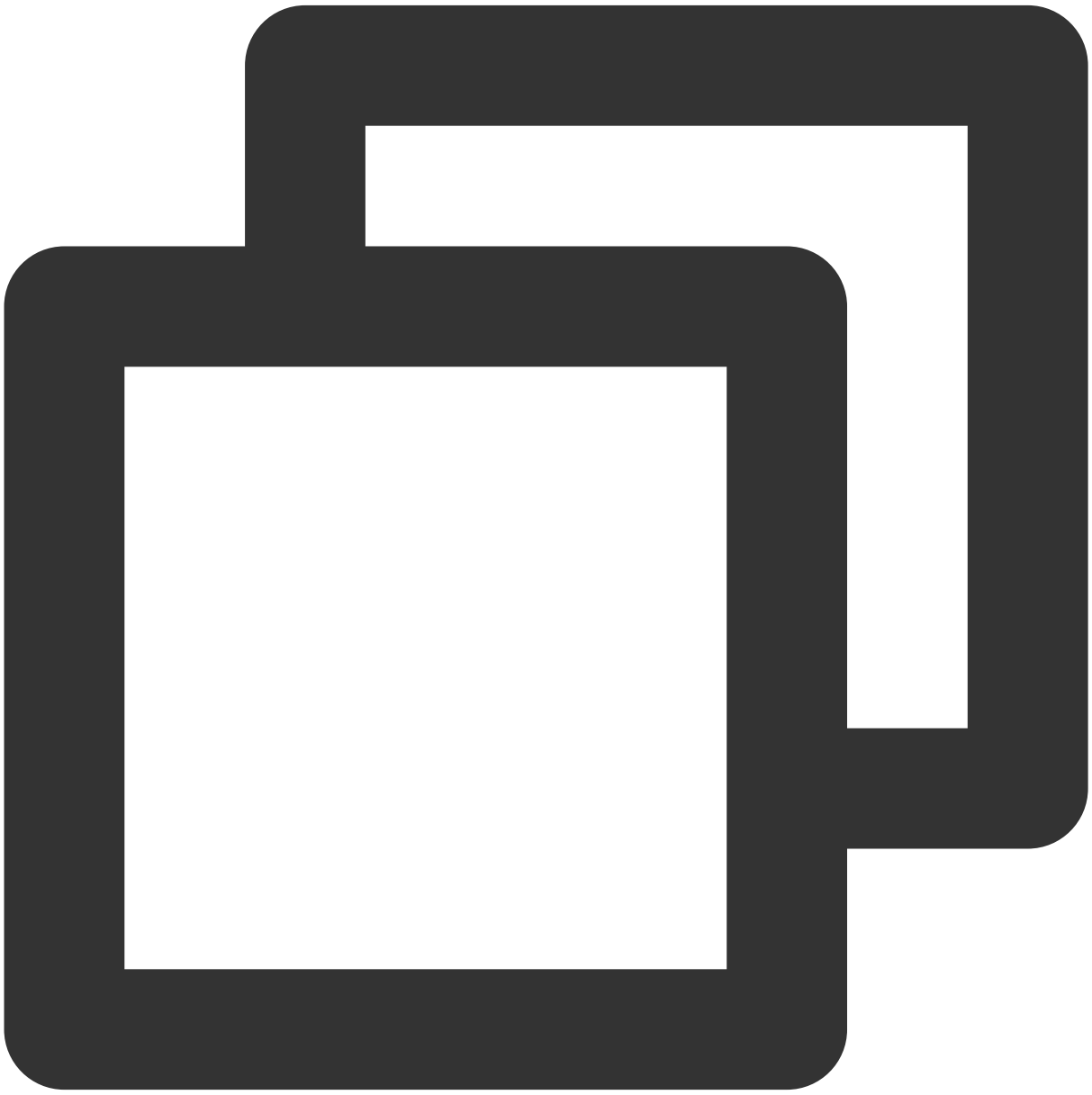
版权所有：腾讯云计算（北京）有限责任公司

高级功能

最近更新时间：2024-01-20 11:39:30

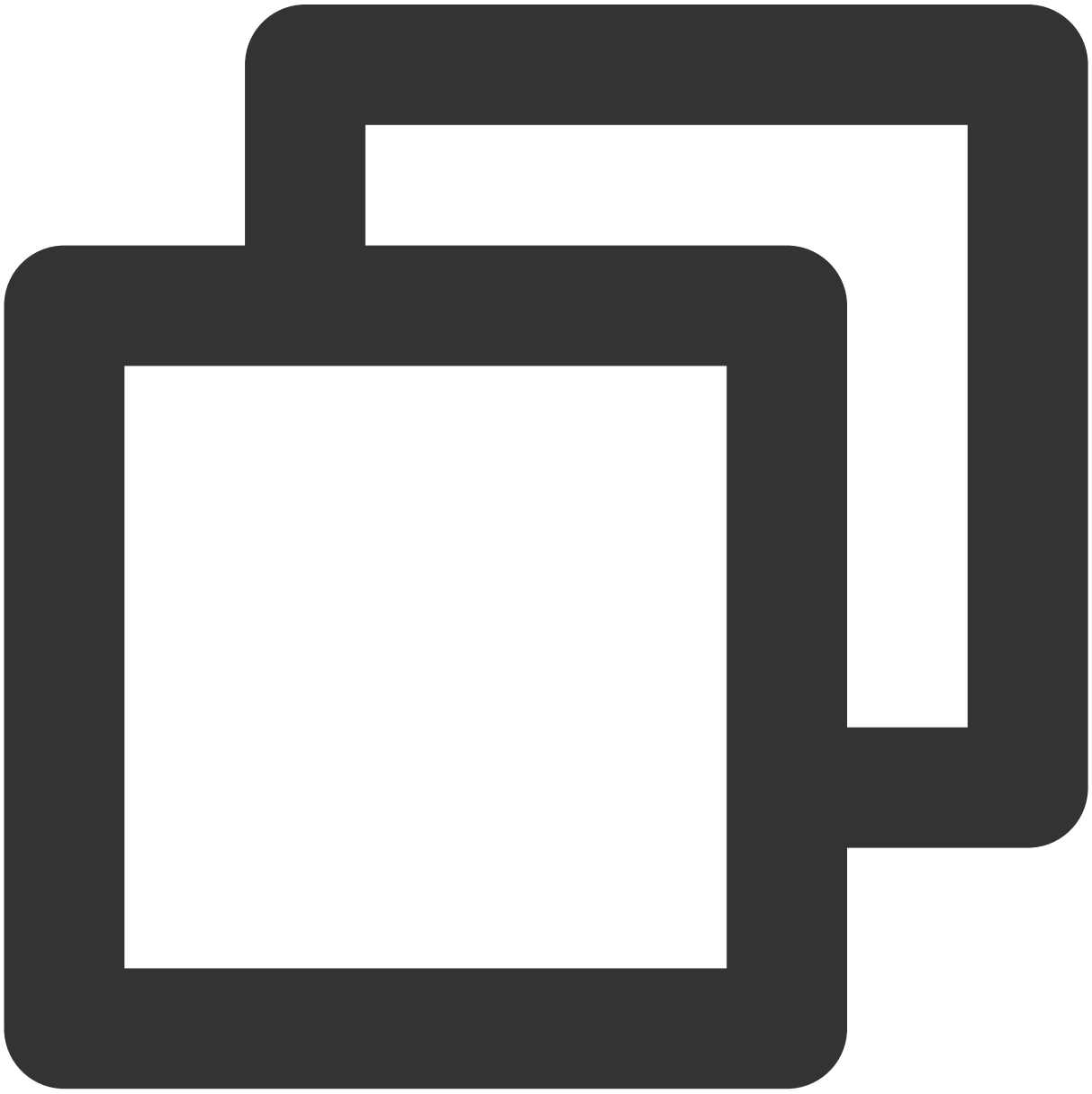
JS-SDK 提供了高级功能可直接对文档进行操作，以下是高级功能的调用流程。

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()

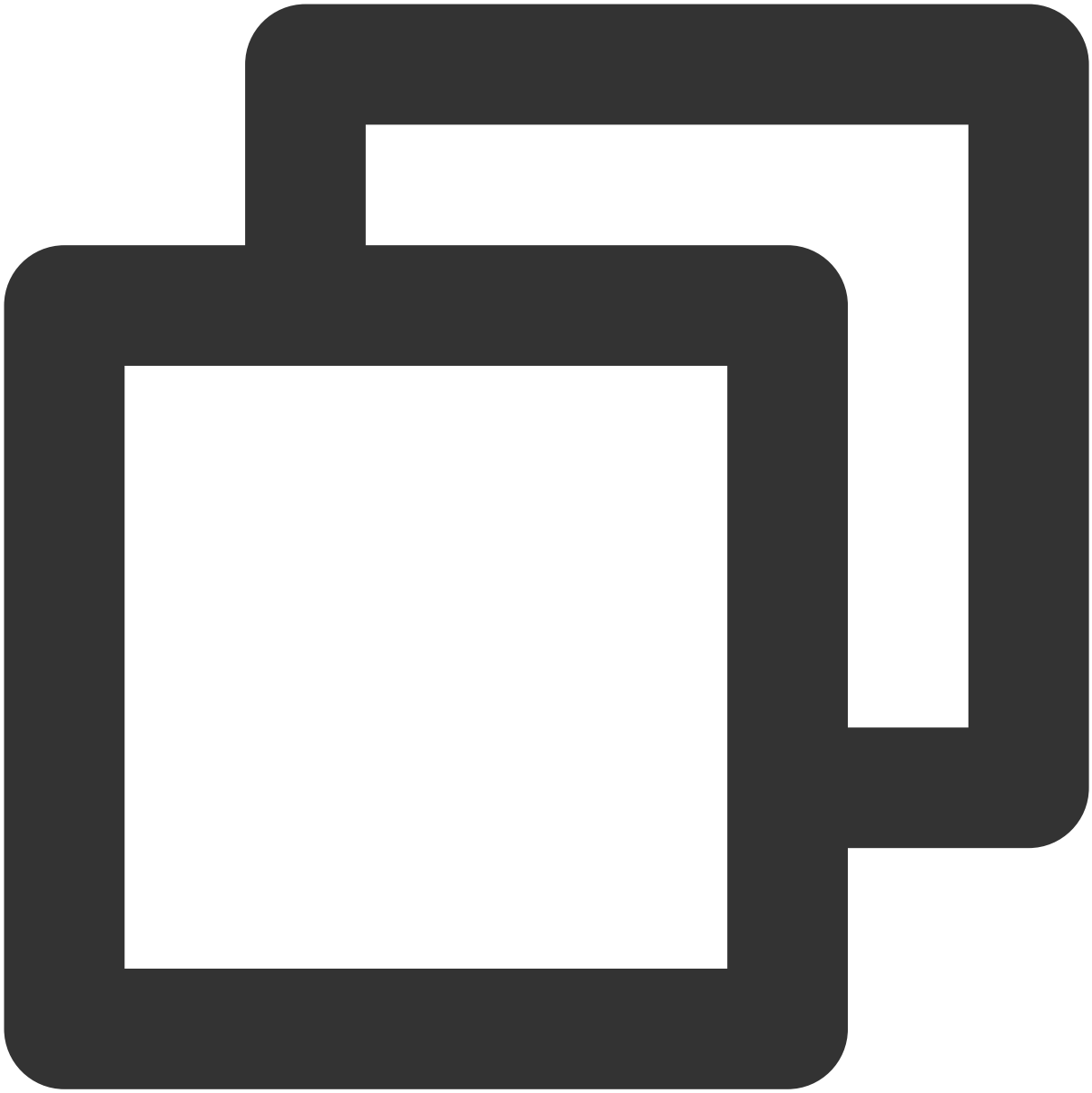
// 自动识别
const app = demo.Application
```

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

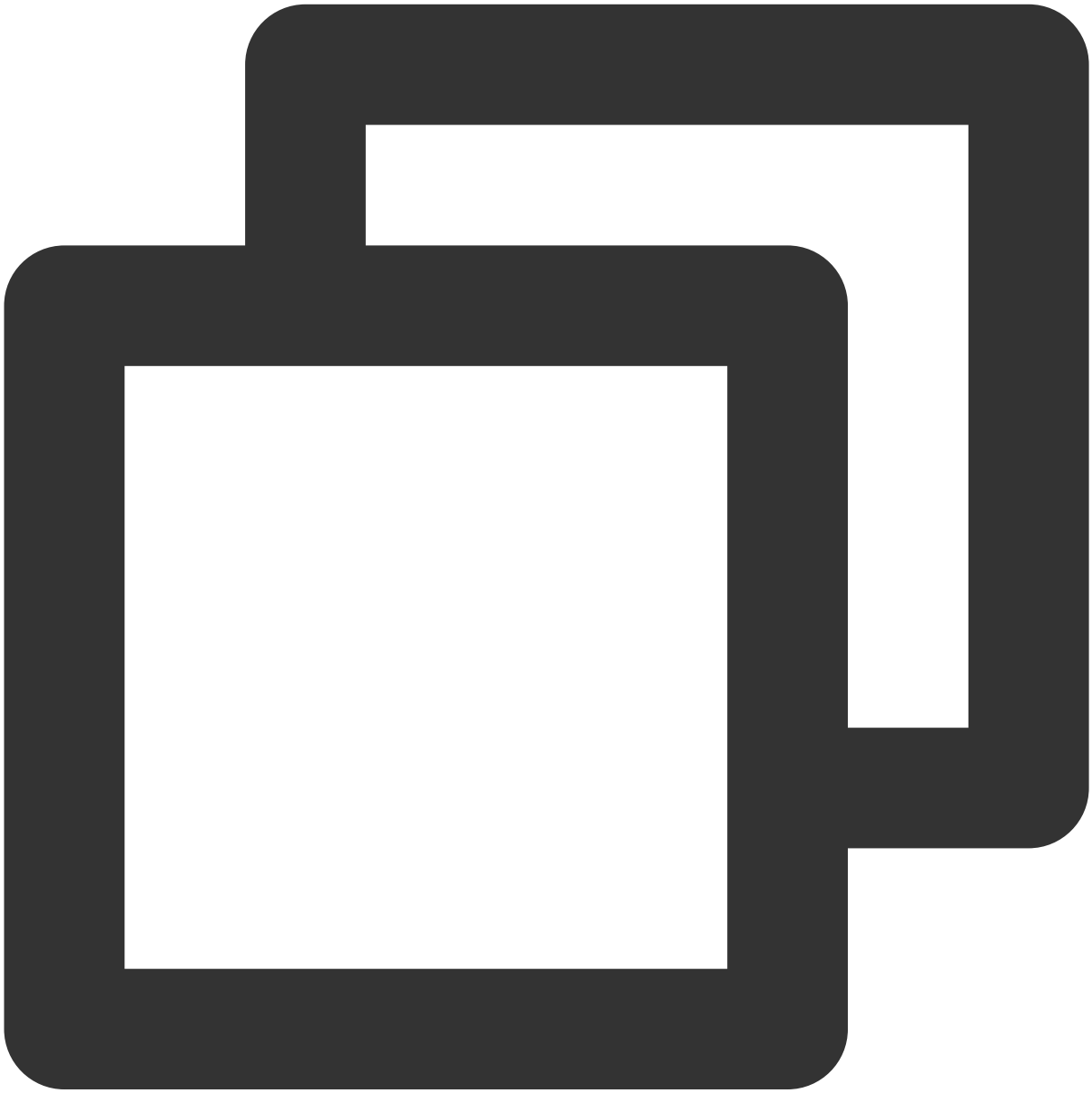
获取总页数



```
/*
* @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")
}
```

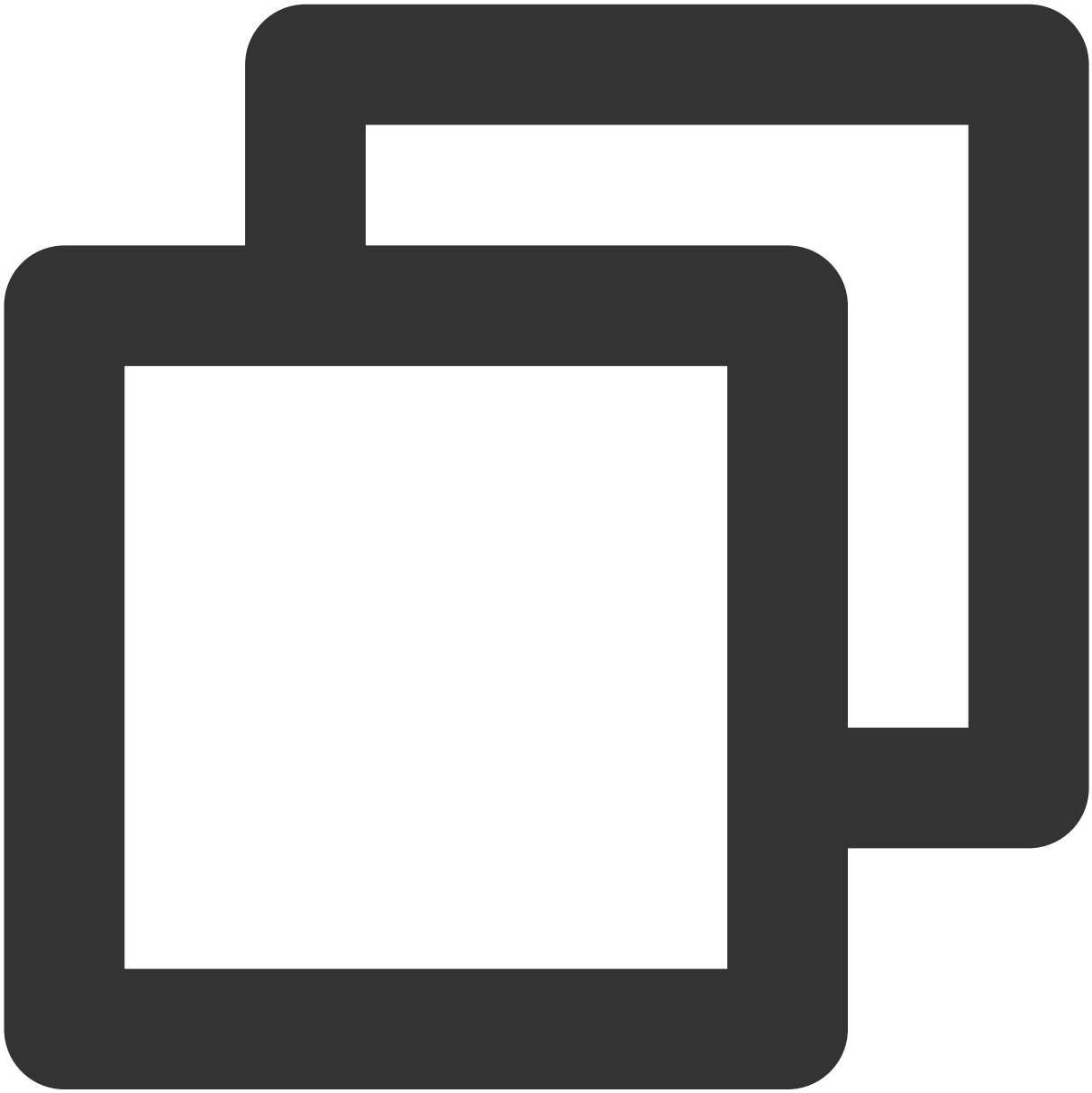
获取当前页



```
/*
* @param: WdInformation: {
*       wdActiveEndPageNumber: 3
*   }
* @return: number
*/
const app = demo.WordApplication()
```

```
const {Enum} = app
let currentPage = await app.ActiveDocument.Selection.Information(Enum.WdInformati
```

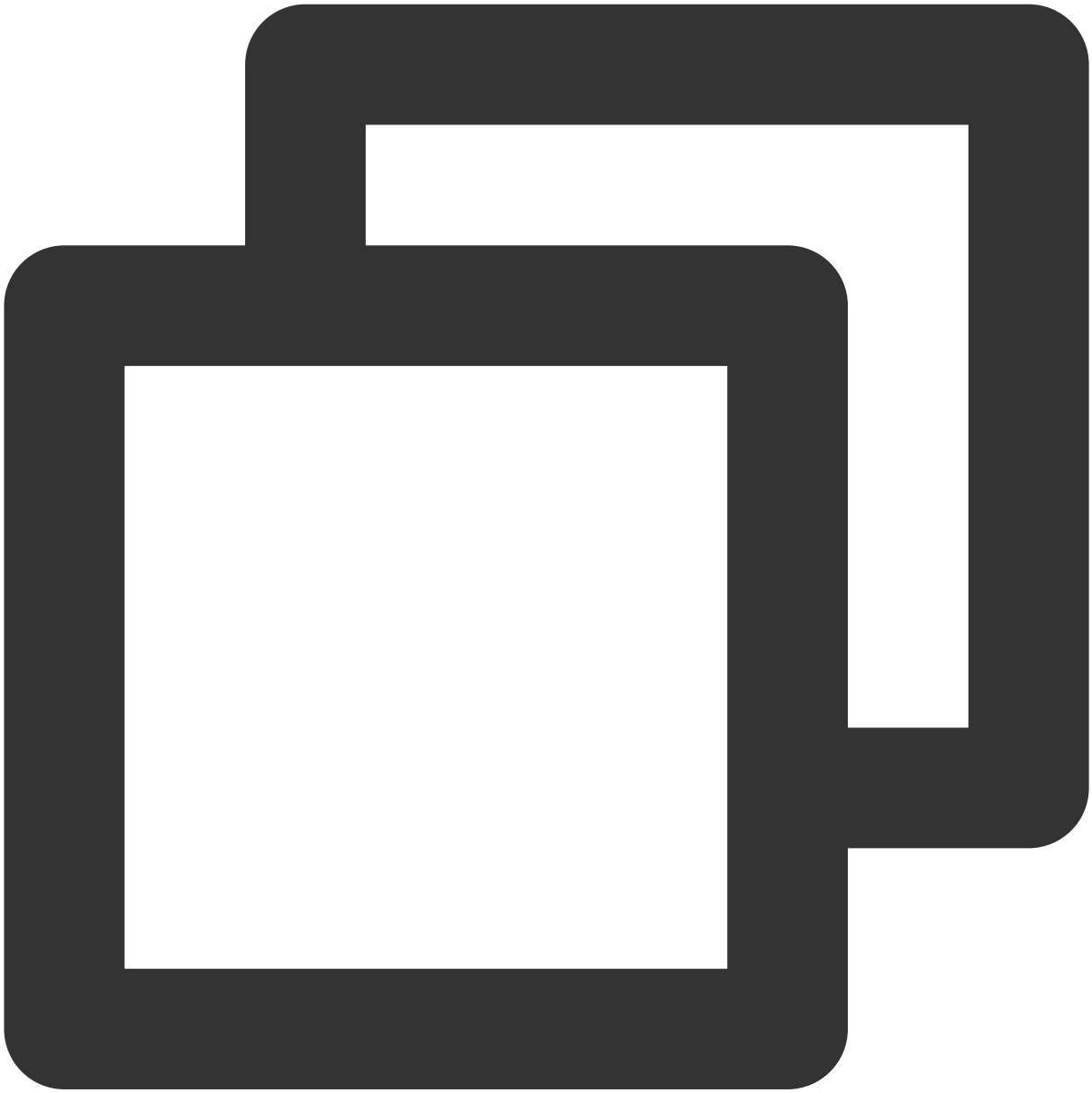
跳转到指定页



```
/*
* @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
    })
```

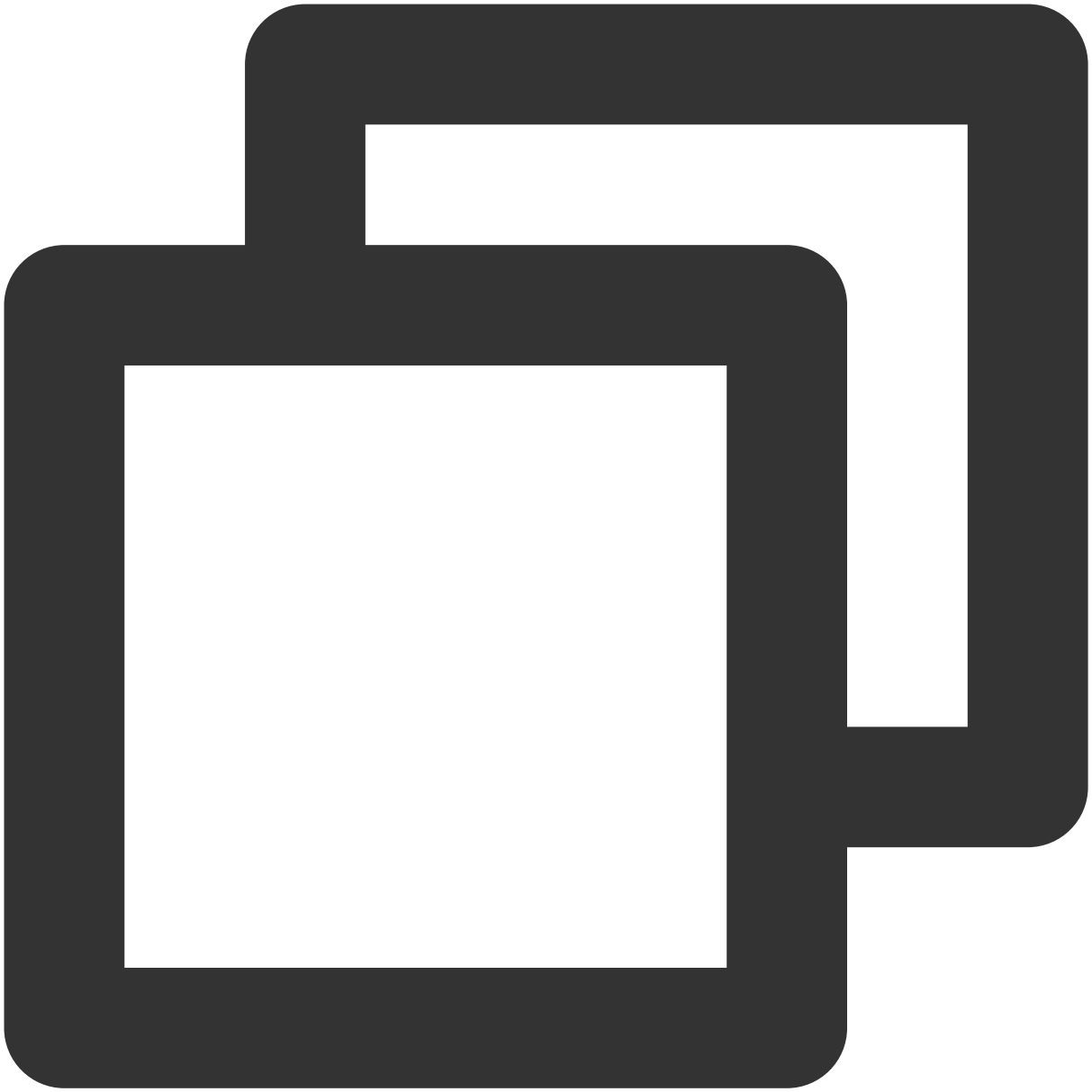
设置评论显示



```
/*
* @param: bool
* true 为显示,  false 为隐藏
*/

// 隐藏评论
demo.WordApplication().ActiveDocument.ActiveWindow.View.ShowComments = false
```

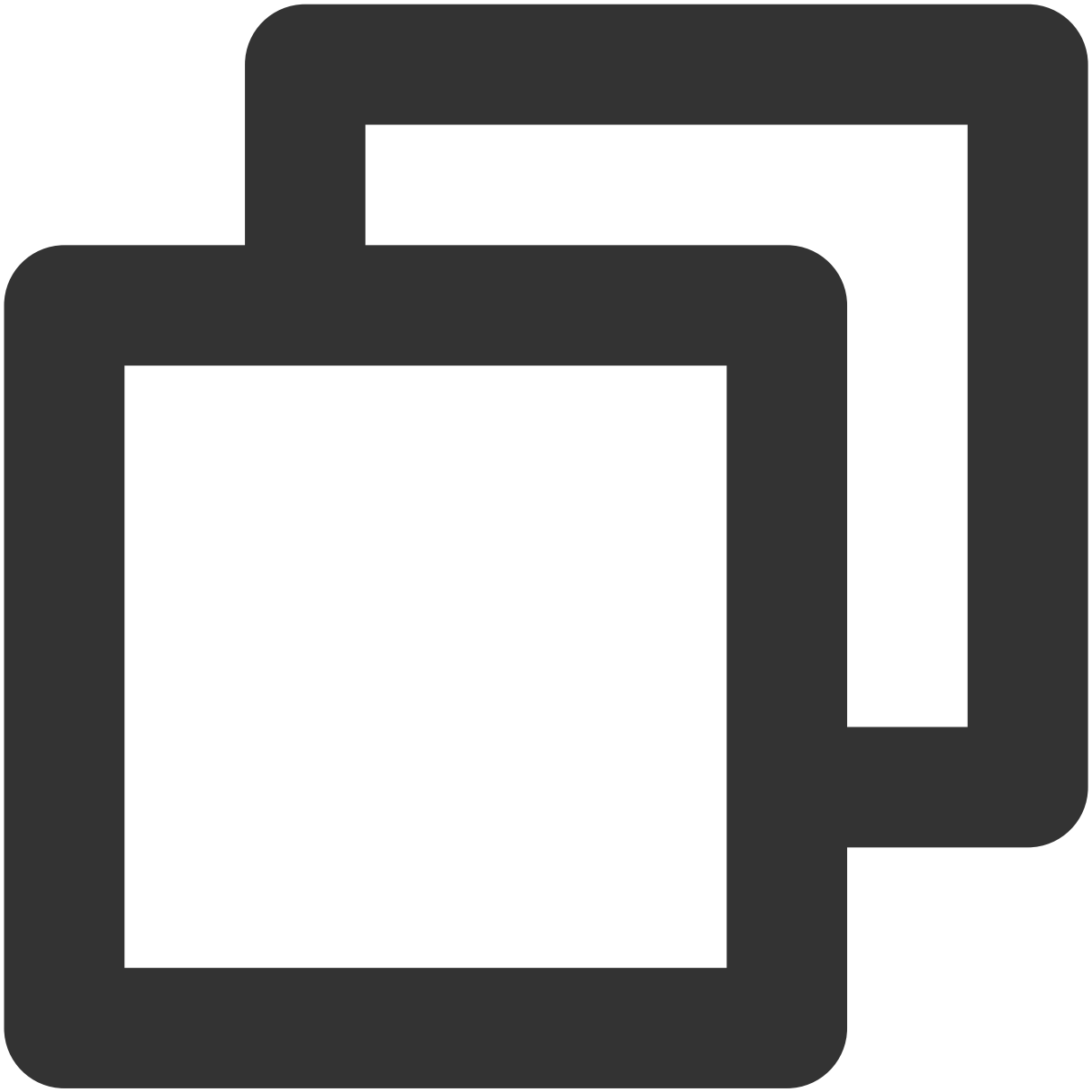
设置目录显示



```
/*
* @param: bool
* true 为显示,  false 为隐藏
*/

// 隐藏目录
demo.WordApplication().ActiveDocument.ActiveWindow.DocumentMap = false
```

视图缩放



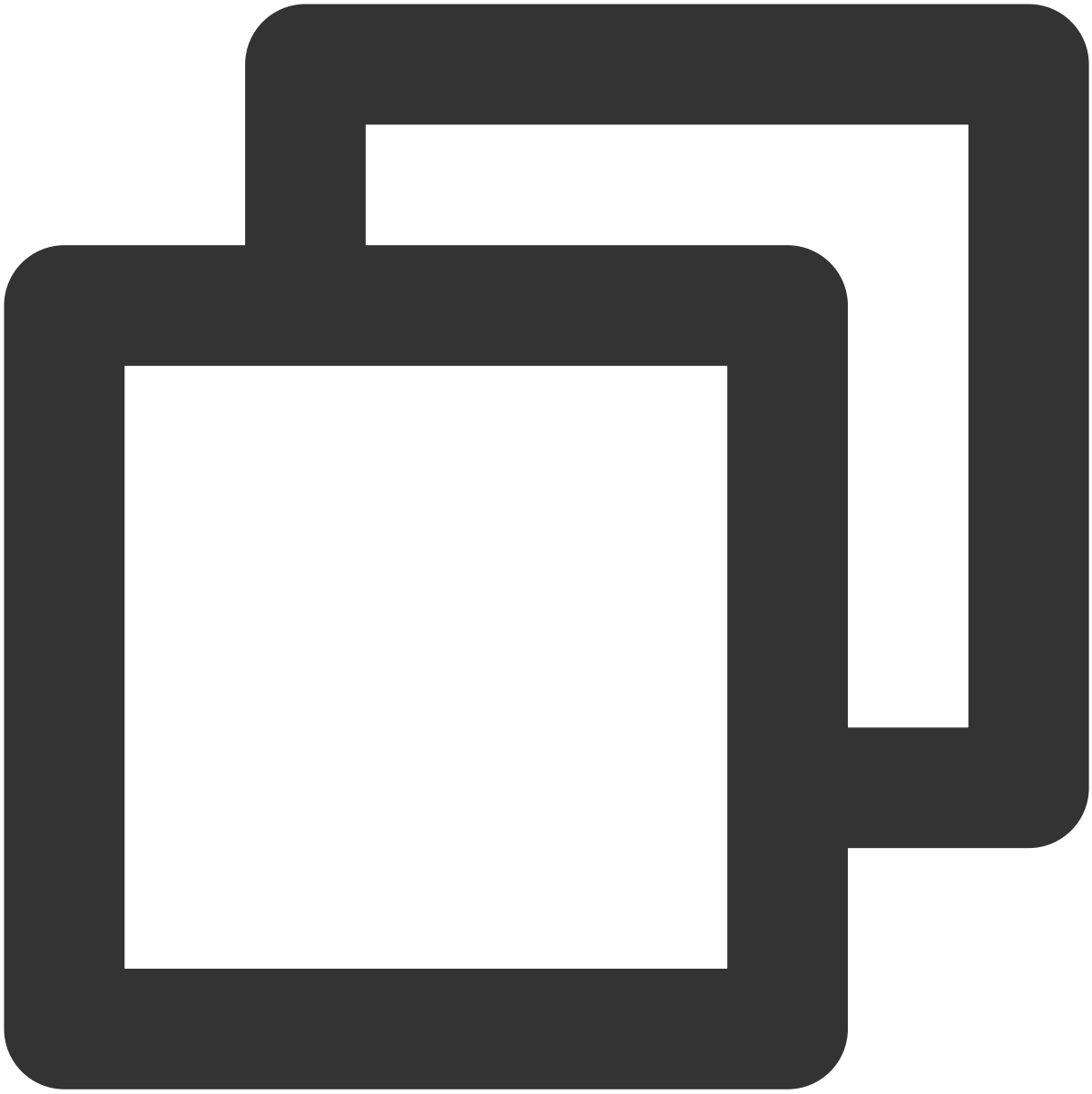
```
/*
* @return: number
*/
await demo.WordApplication().ActiveDocument.ActiveWindow.View.Zoom.Percentage  //

/*
* @param : 50 <= number <= 300 (缩放属性值在50%到 300%之间)
*/
demo.WordApplication().ActiveDocument.ActiveWindow.View.Zoom.Percentage = 100  //
```

版权所有：腾讯云计算（北京）有限责任公司

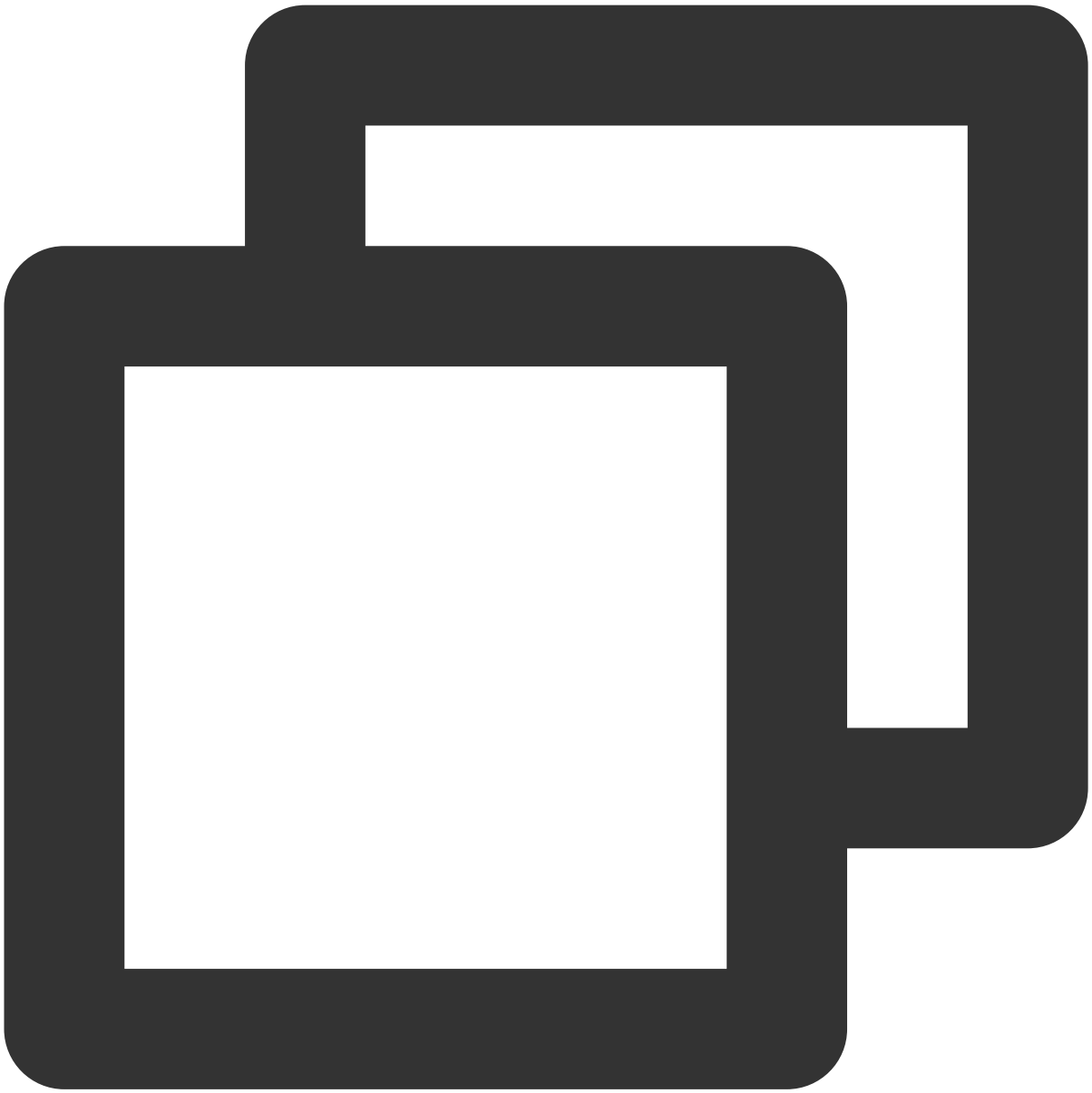
表格文件

获取所有 sheet 名称



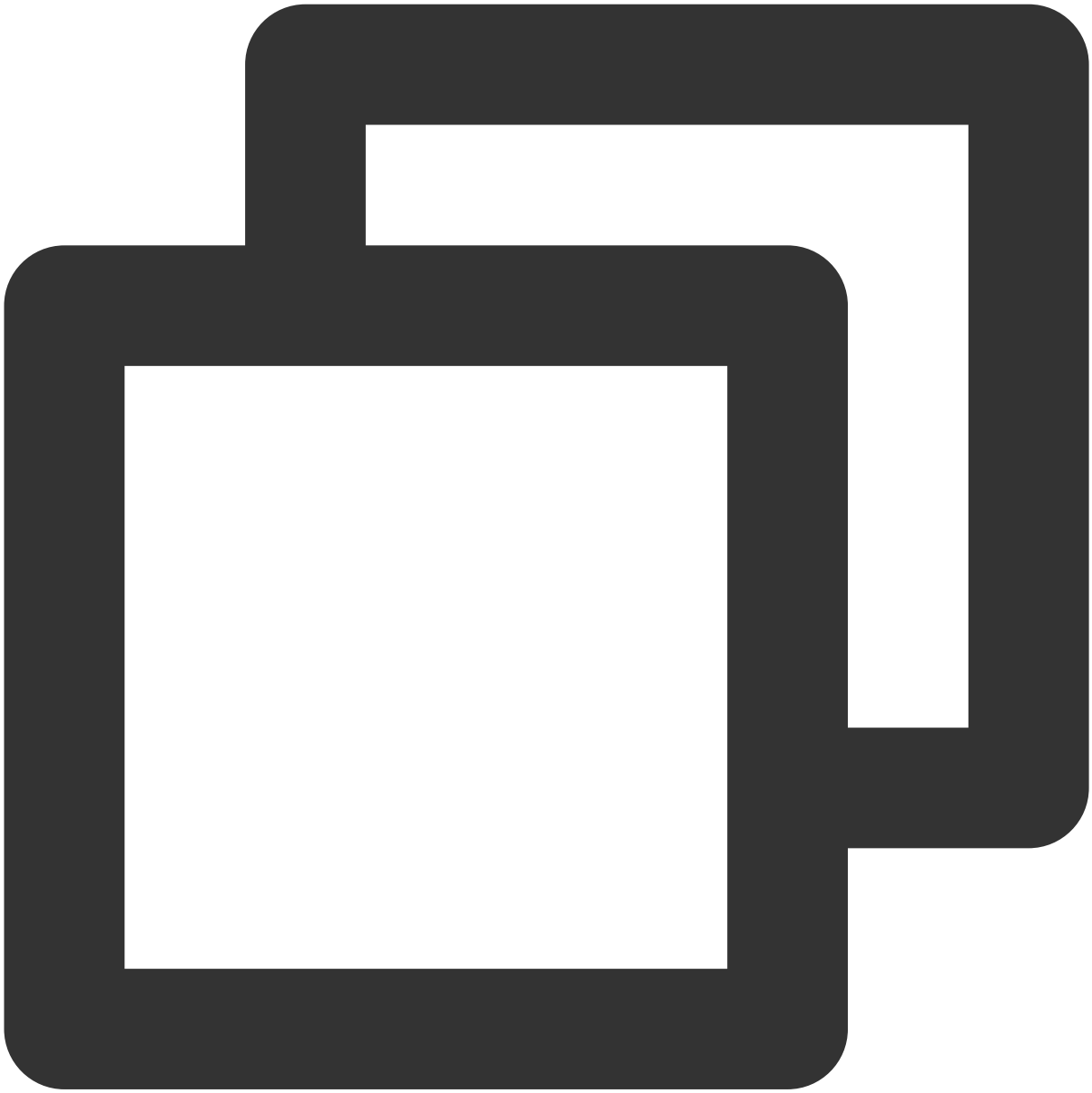
```
const app = demo.ExcelApplication()
const Names = []
const count = await app.Sheets.Count;
for(let index = 1; index <= count; index++) {
  Names.push(await app.Sheets.Item(index).Name);
}
console.log(Names)
```

获取当前 sheet 名称



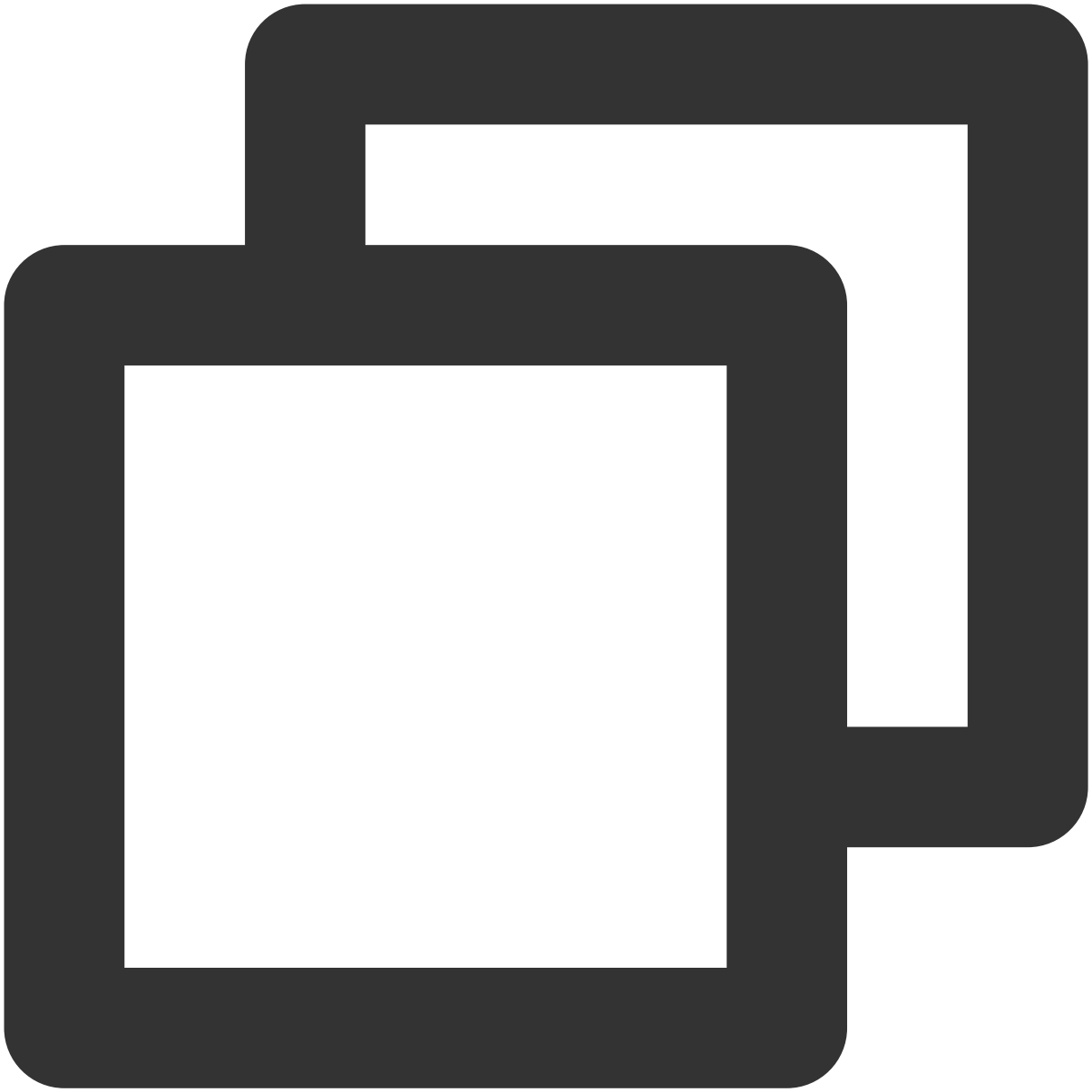
```
const app = demo.ExcelApplication()
const name = await app.ActiveSheet.Name
```

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
```

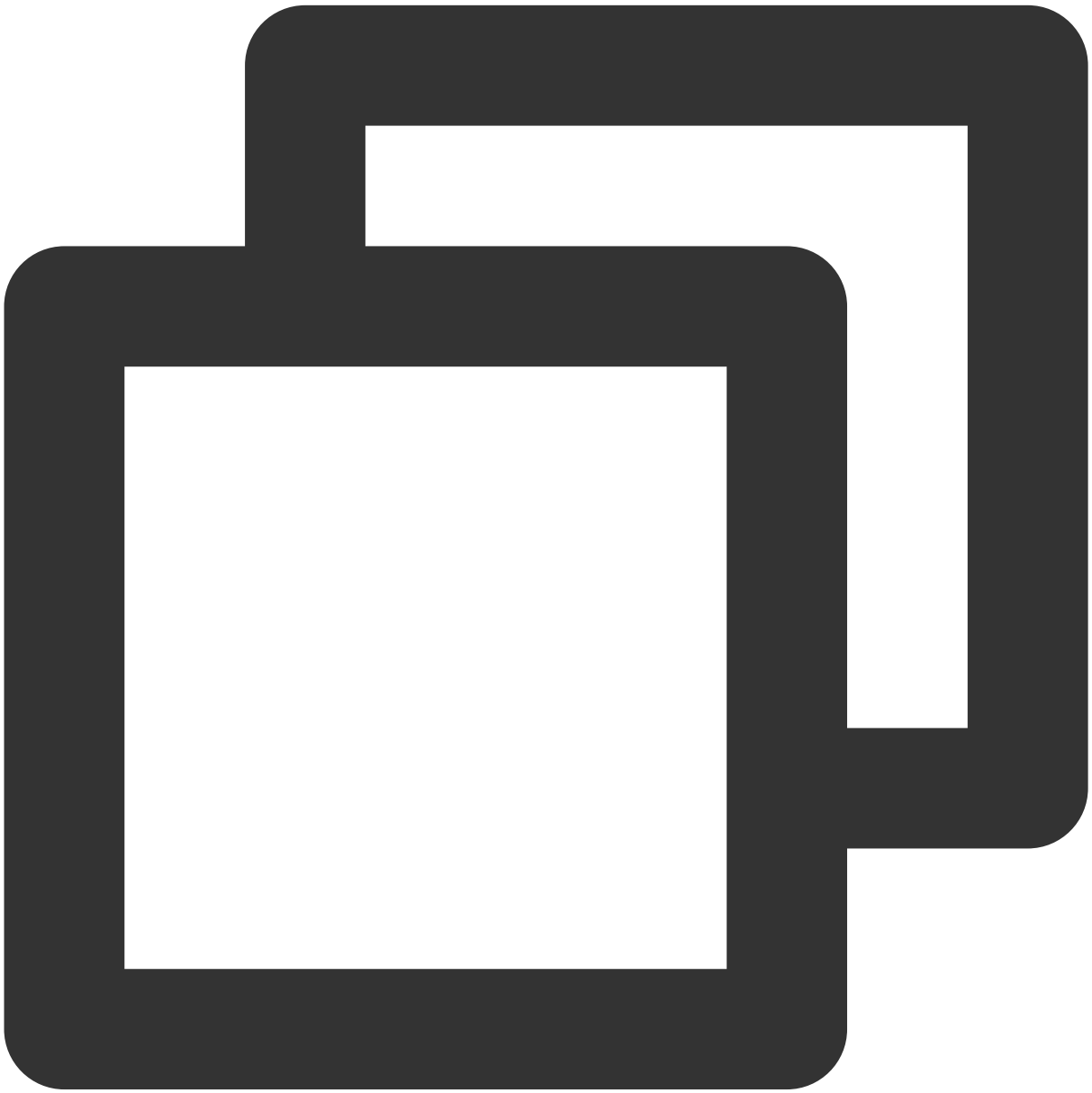
视图缩放



```
/*
* @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
```

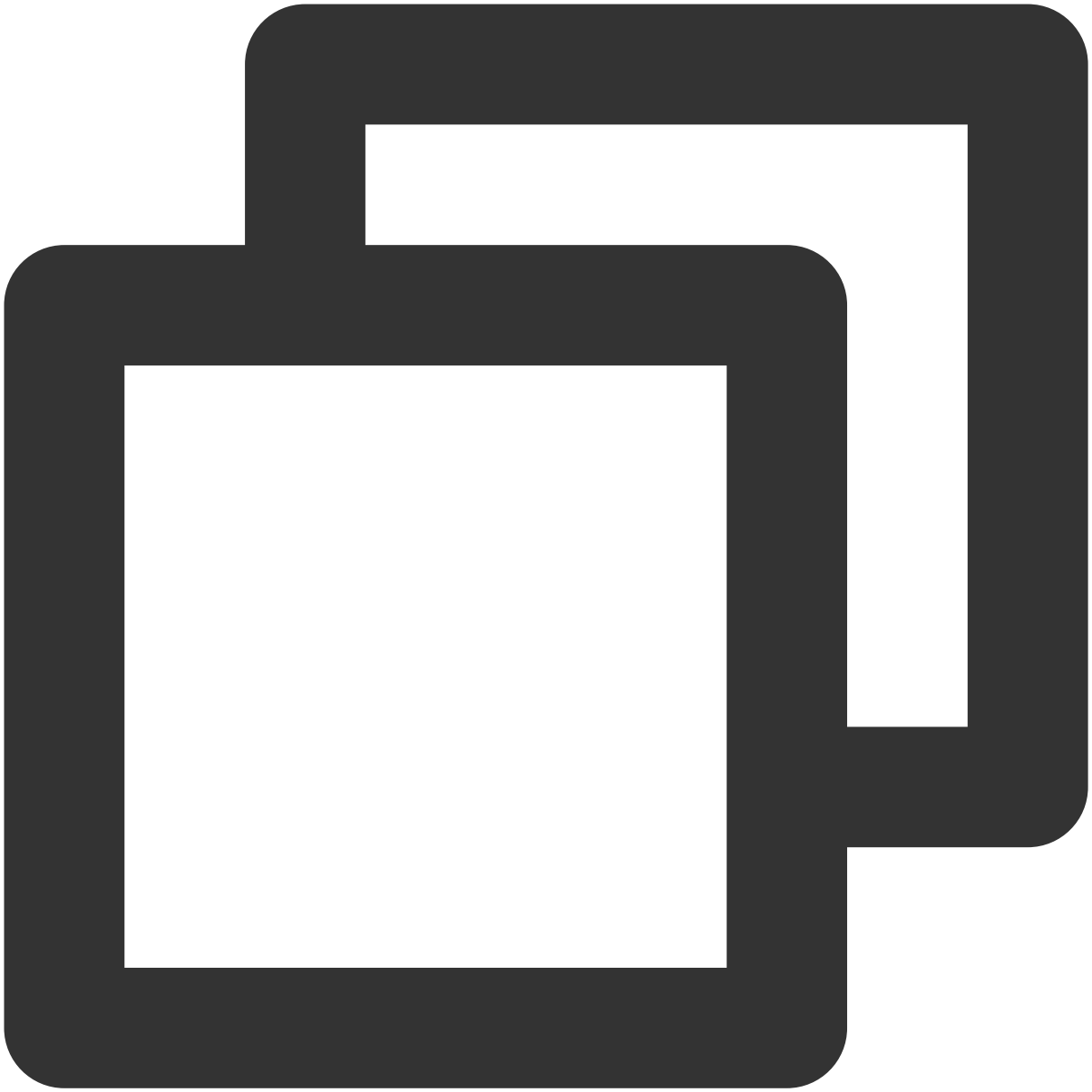
演示文件

获取总页数



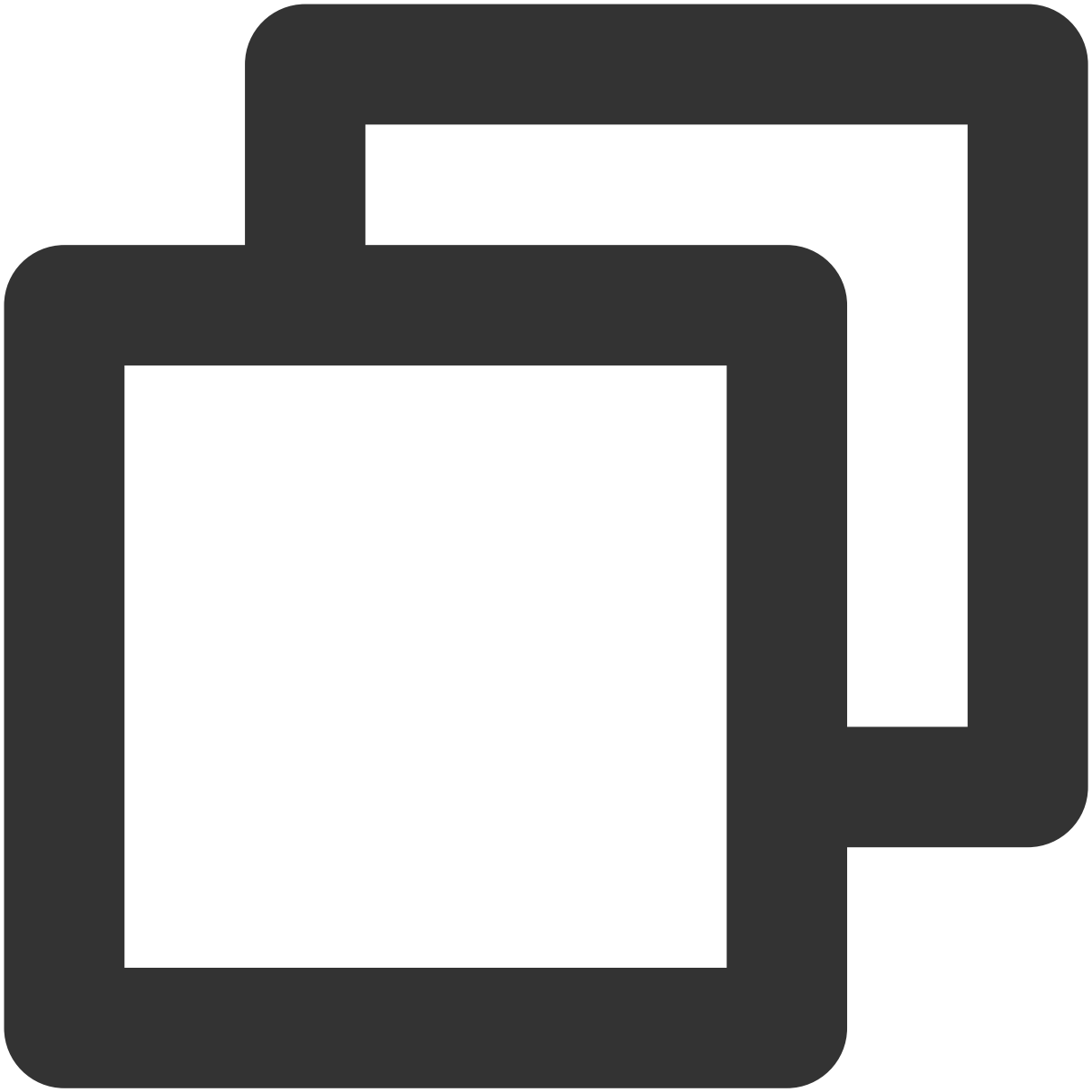
```
/*
* @return: number
*/
let totalPages = await demo.PPTApplication().ActivePresentation.Slides.Count
```

获取当前页



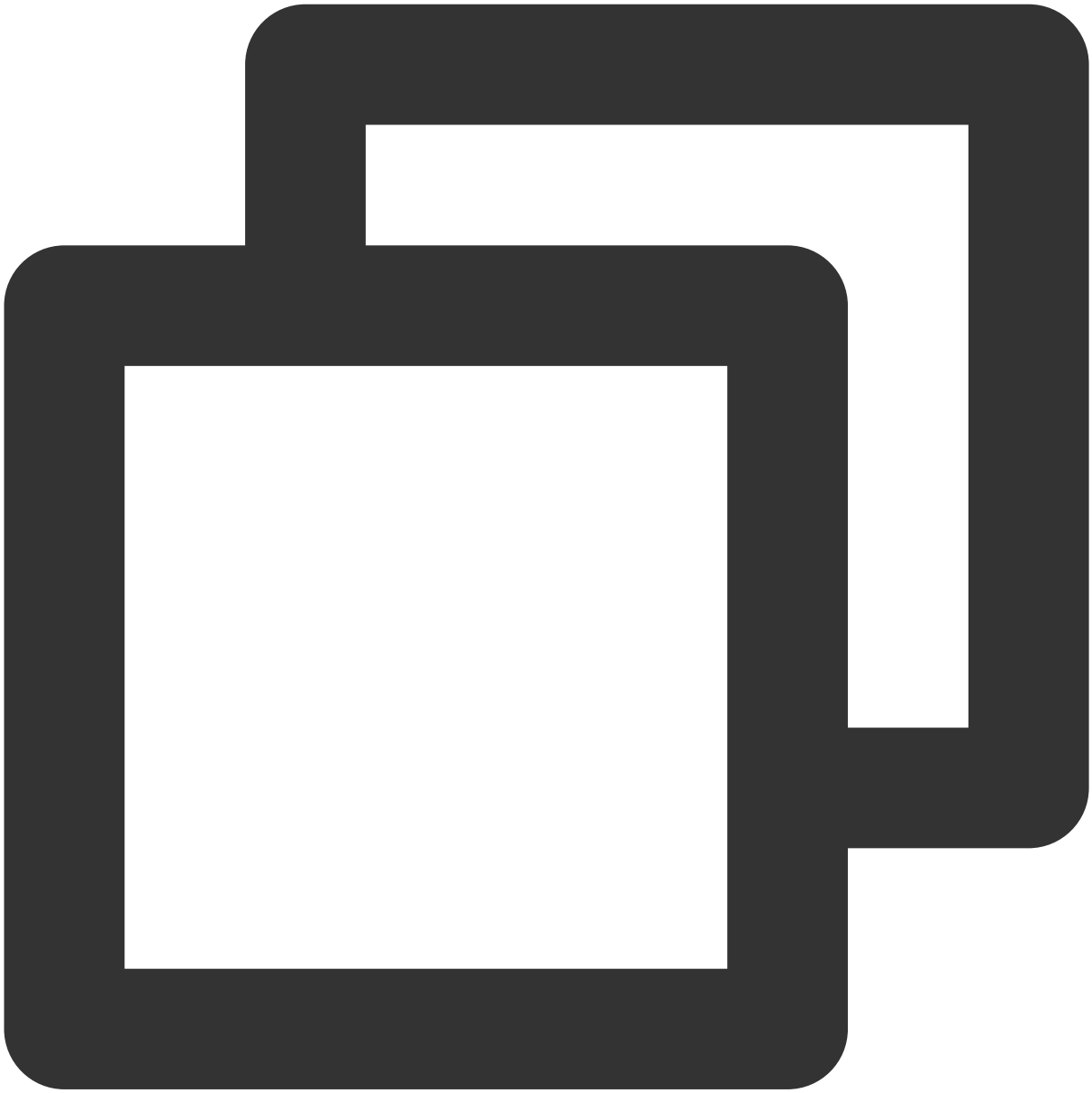
```
/*
* @return: number
*/
let totalPages = await demo.PPTApplication().ActivePresentation.SlideShowWindow.V
```

跳转到指定页



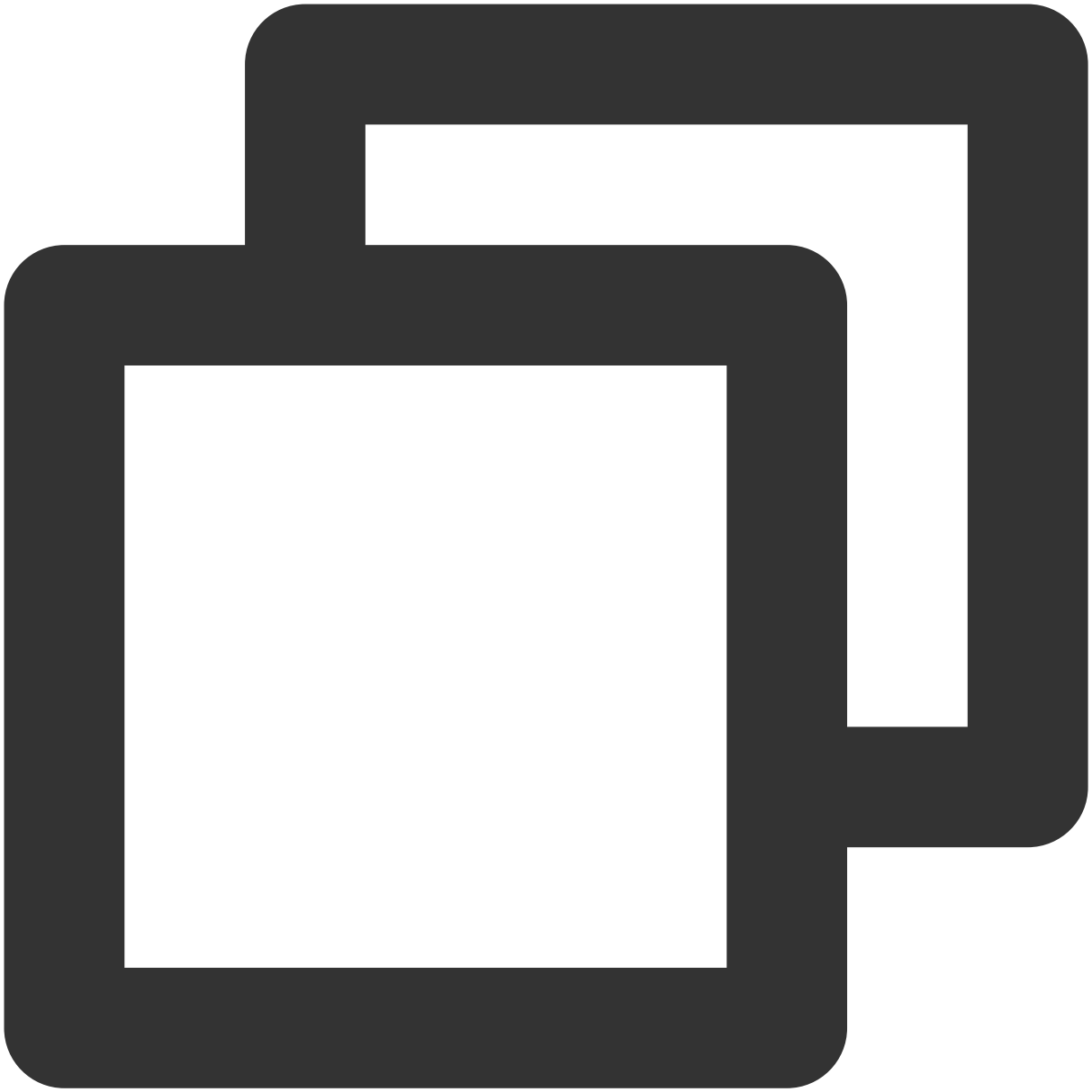
```
/*
* @param: number
*/
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.GotoSlide(3)
```

播放幻灯片



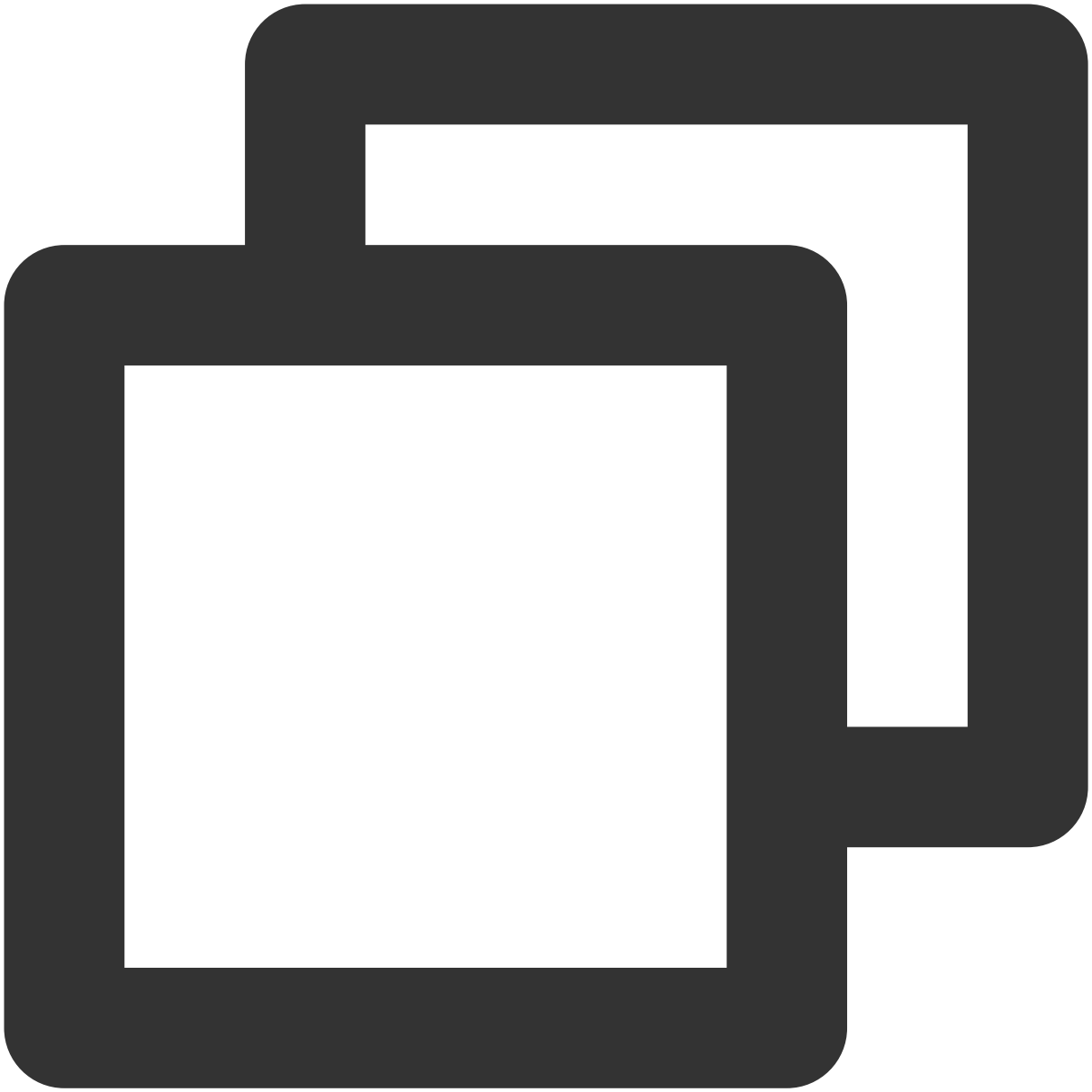
```
await demo.PPTApplication().ActivePresentation.SlideShowSettings.Run()
```

退出播放



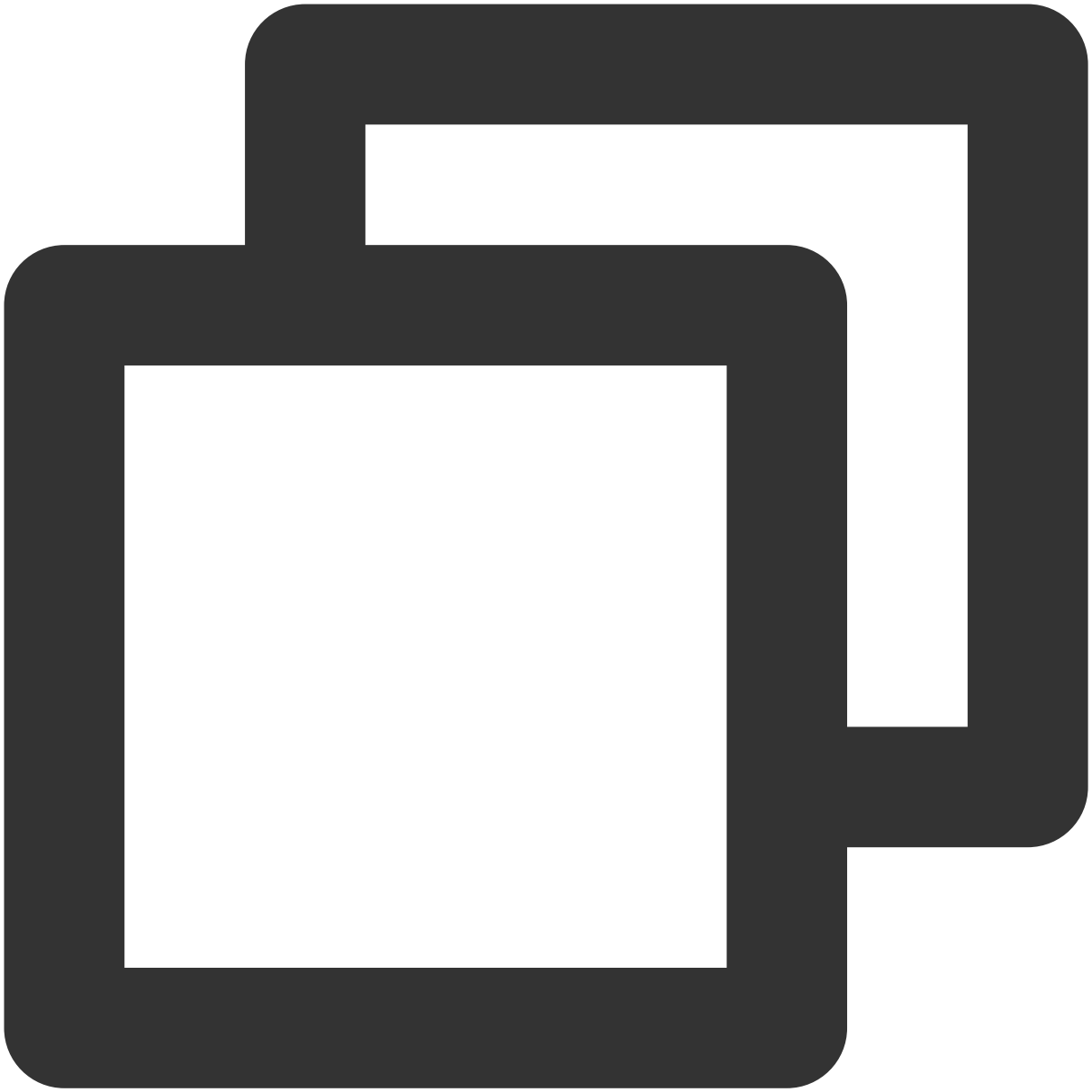
```
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.Exit()
```

幻灯片下一步动画



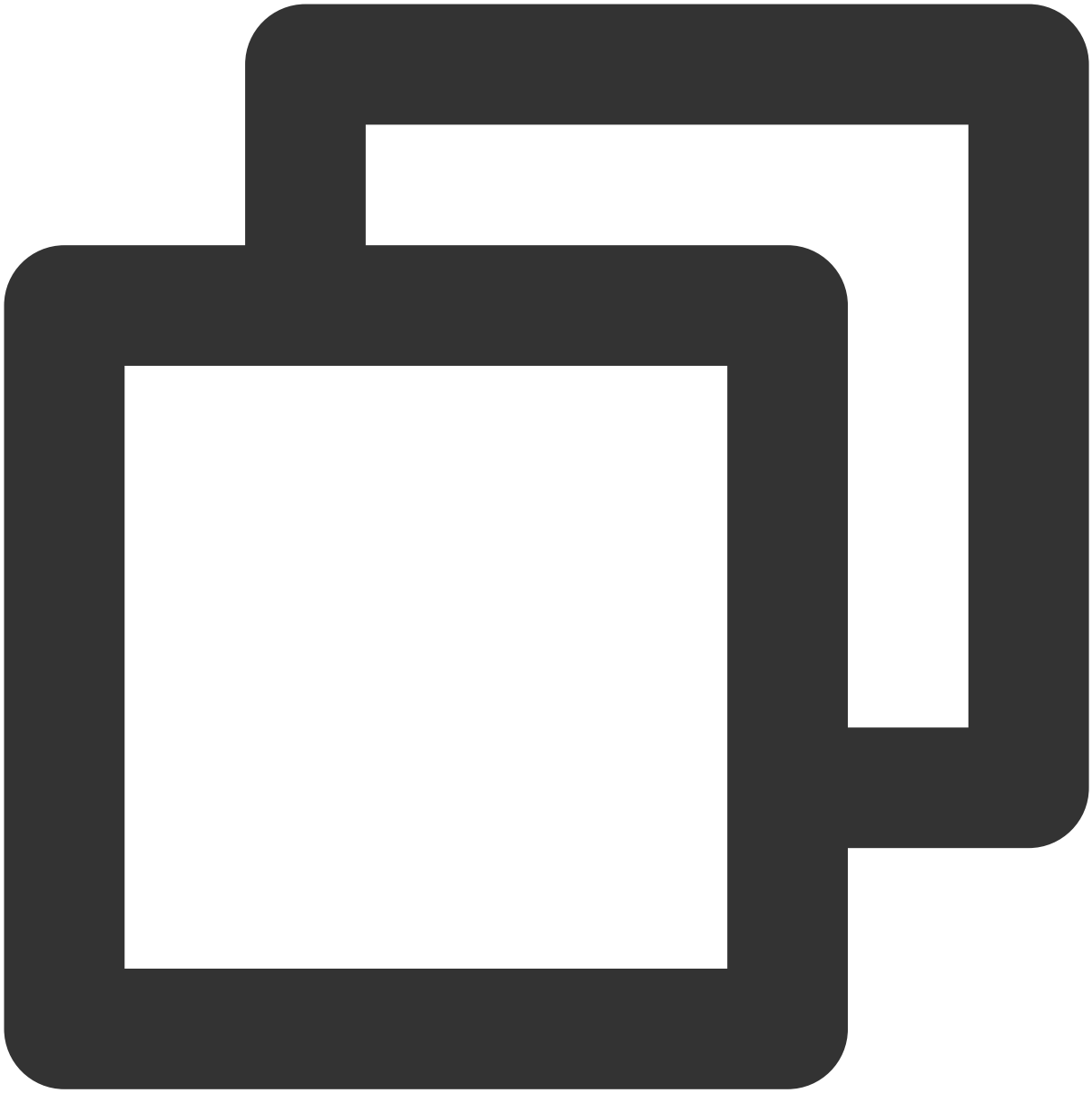
```
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.GotoNextClick
```

幻灯片上一步动画



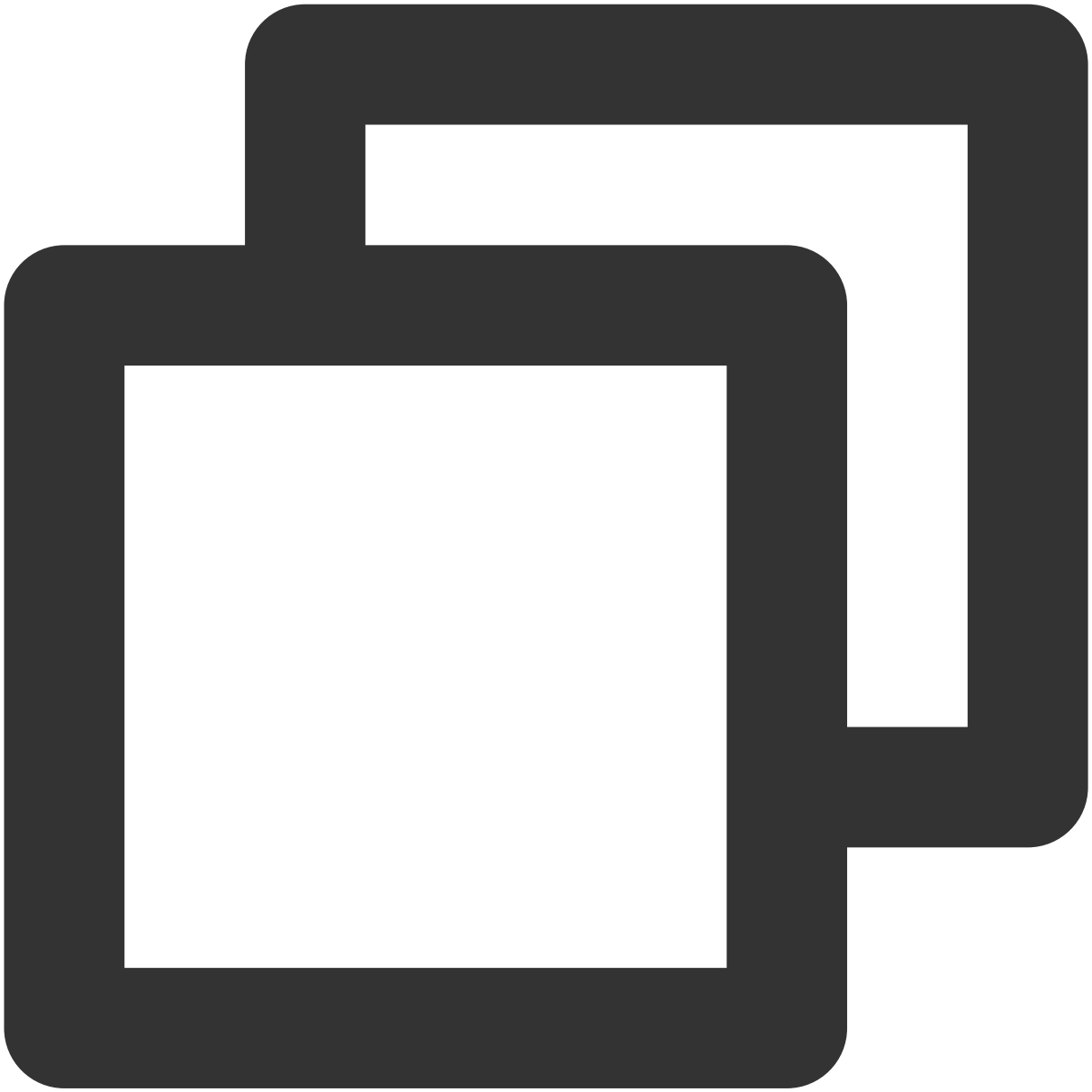
```
await demo.PPTApplication().ActivePresentation.SlideShowWindow.View.GotoPreClick(
```

获取播放状态



```
/*
* @return: string ('preview' | 'play')
*/
let currentState = await demo.PPTApplication().ActivePresentation.SlideShowWindow
```

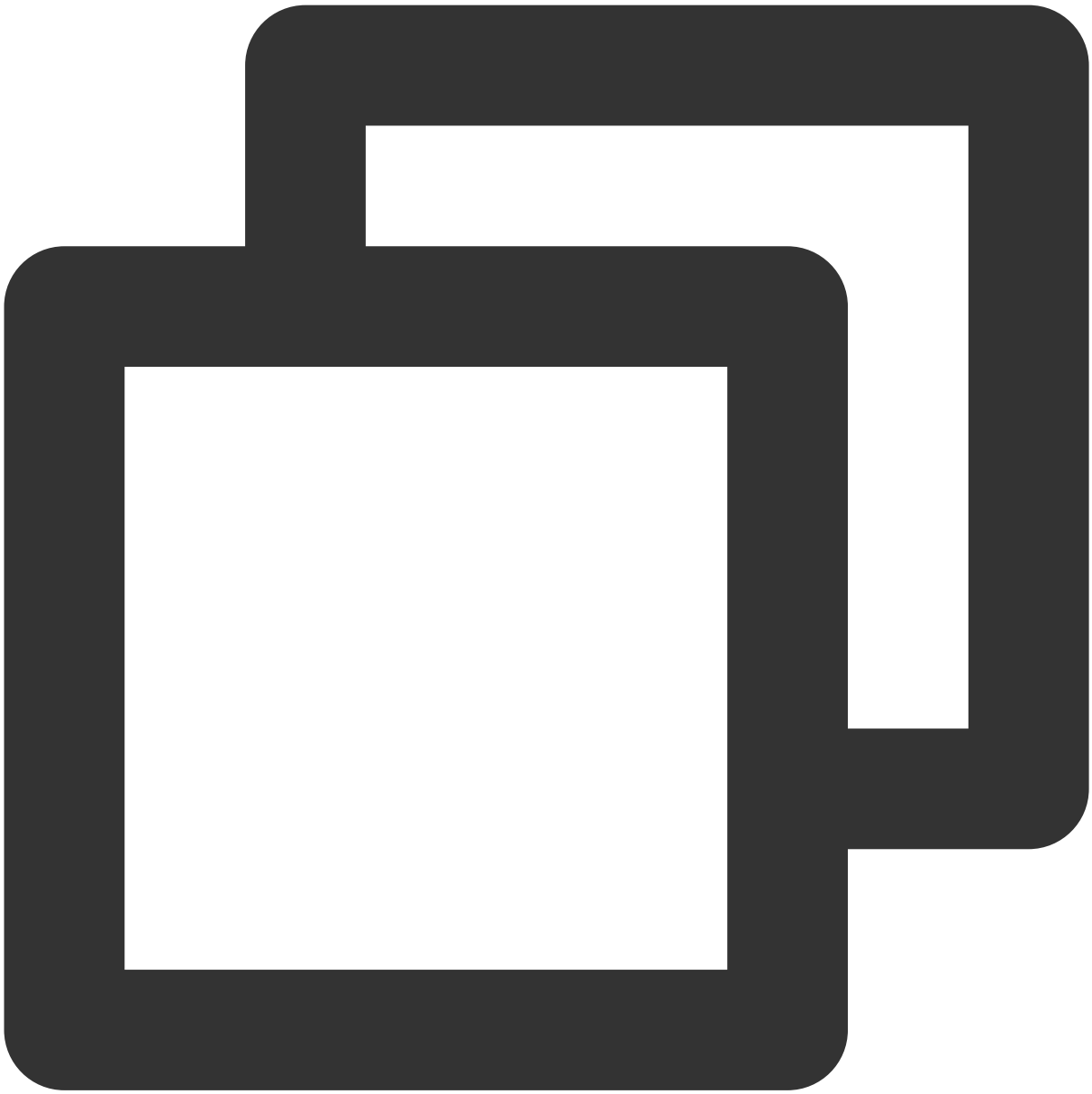
视图缩放



```
/*
* @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
```

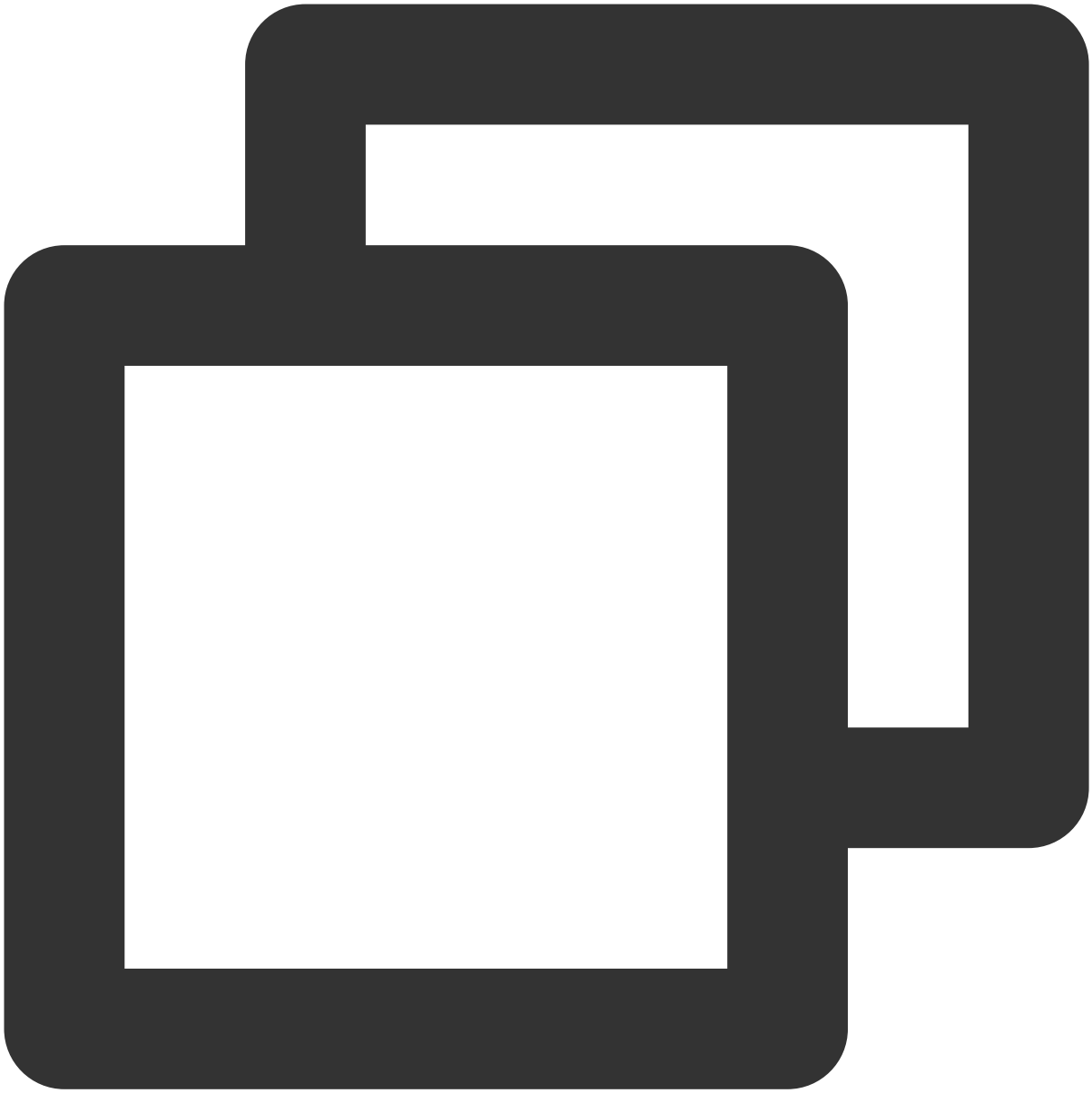
PDF 文件

获取总页数



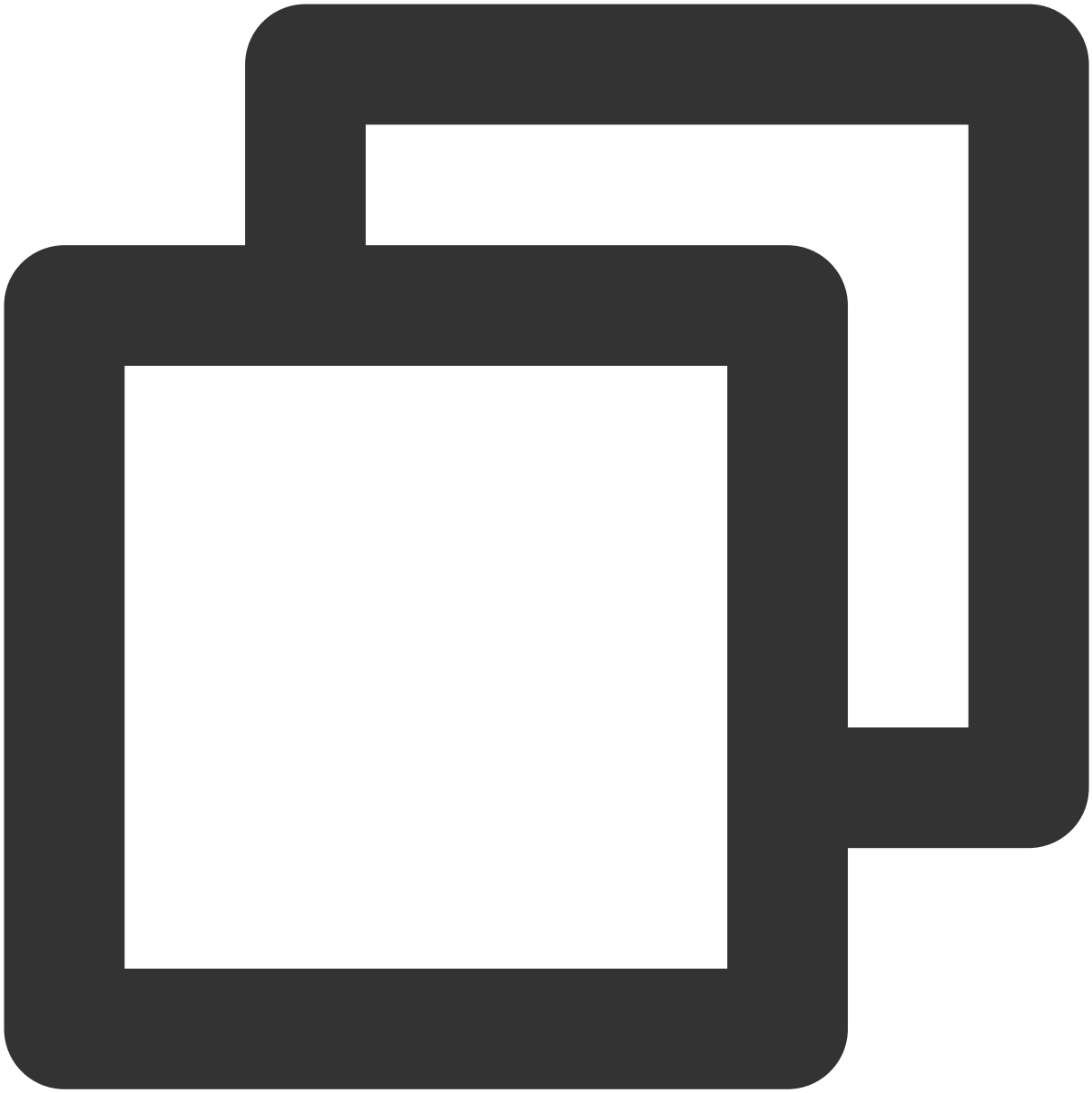
```
/*
* @return: number
*/
let totalPages = await demo.PDFApplication().ActivePDF.PagesCount
```

获取当前页



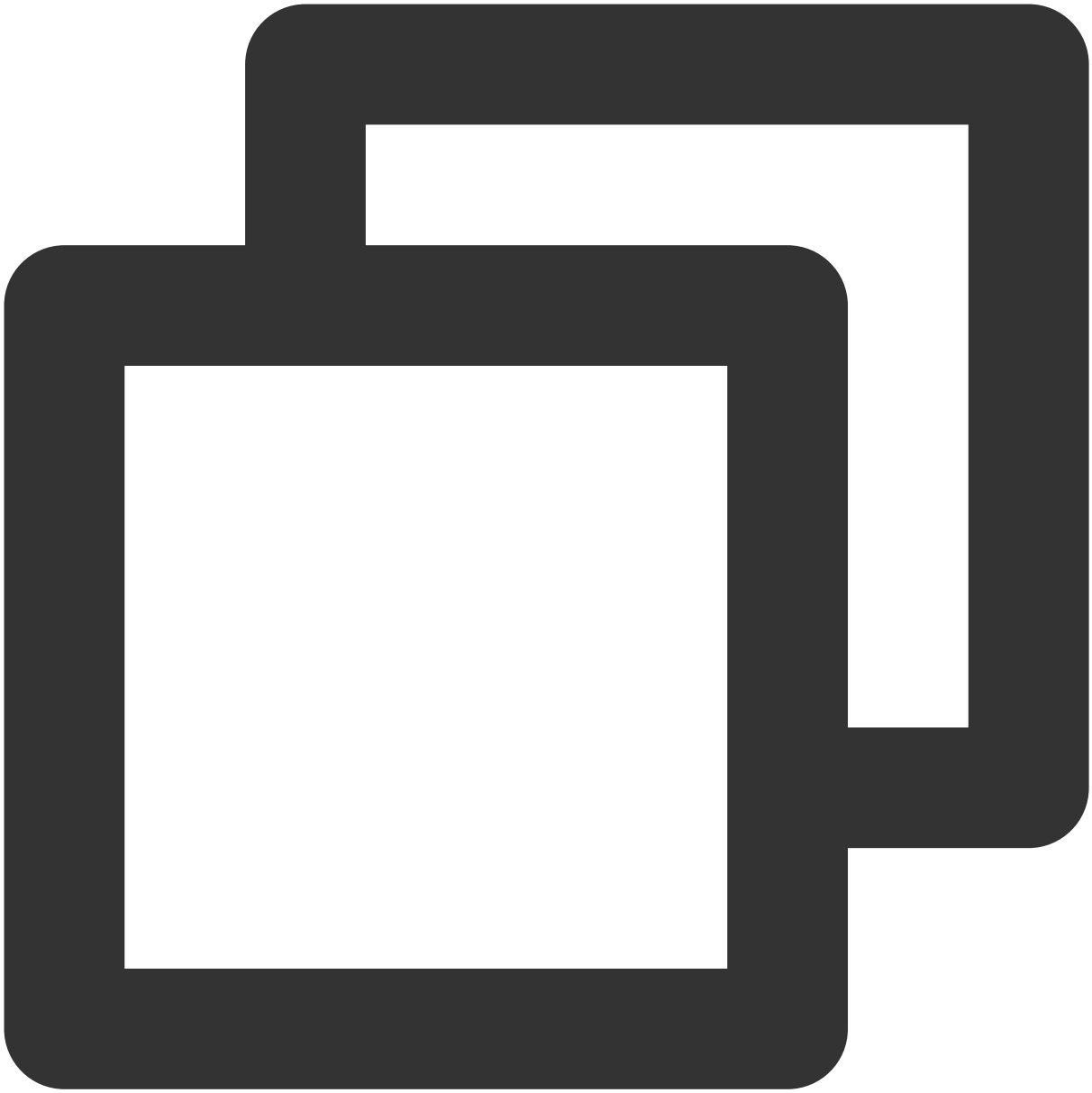
```
/*
* @return: number
*/
let totalPages = await demo.PDFApplication().ActivePDF.CurrentPage
```

跳转到指定页



```
/*
* @param : { PageNum: number }
*/
let PageNum = 10
await demo.PDFApplication().ActivePDF.JumpToPage({PageNum})
```

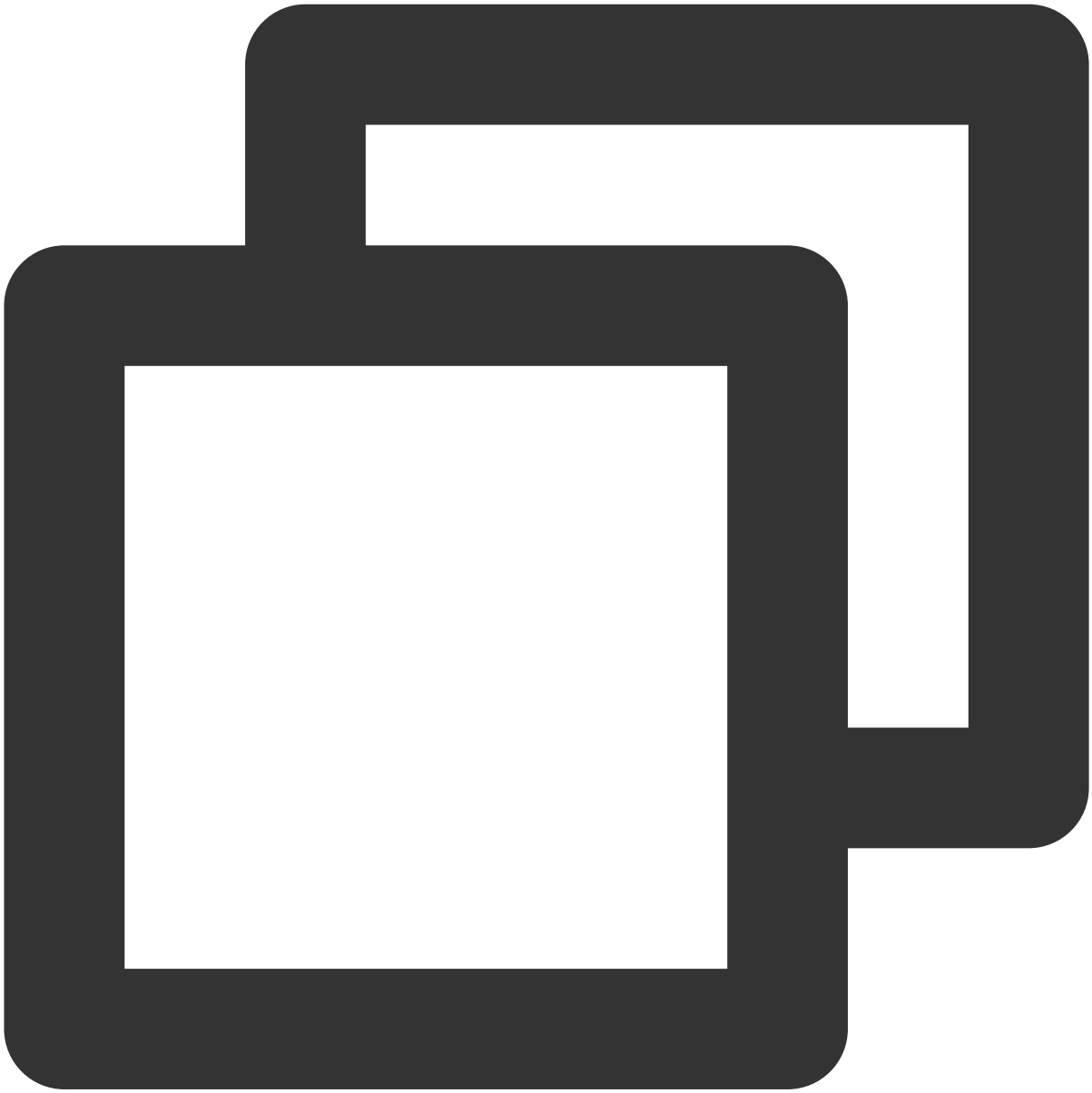
设置页面渲染模式



```
/*
* 1：单页模式
* 0：多页模式
*/

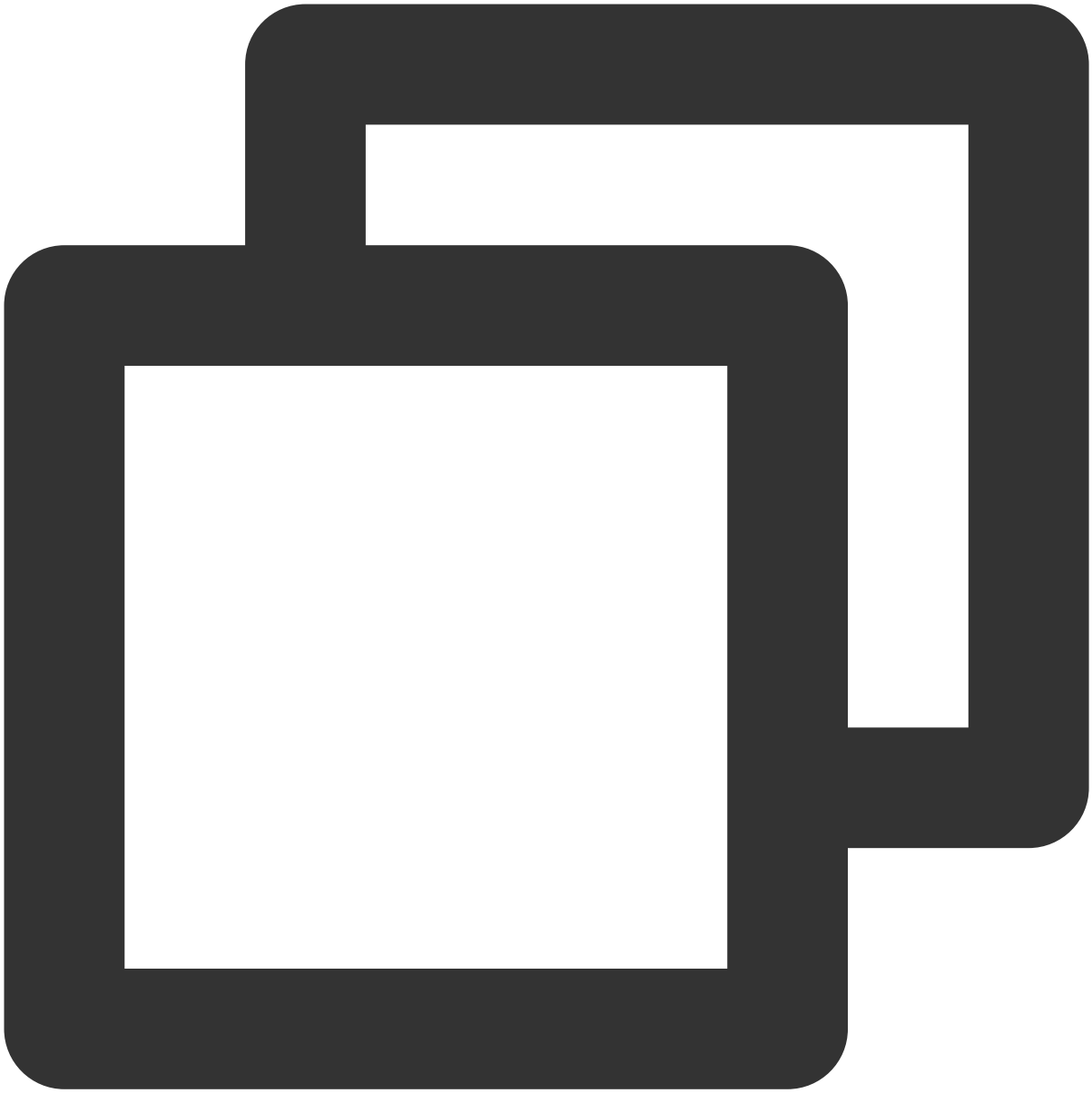
demo.PDFApplication().ActivePDF.PageMode = 1     // 设置为单页模式
```

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
```

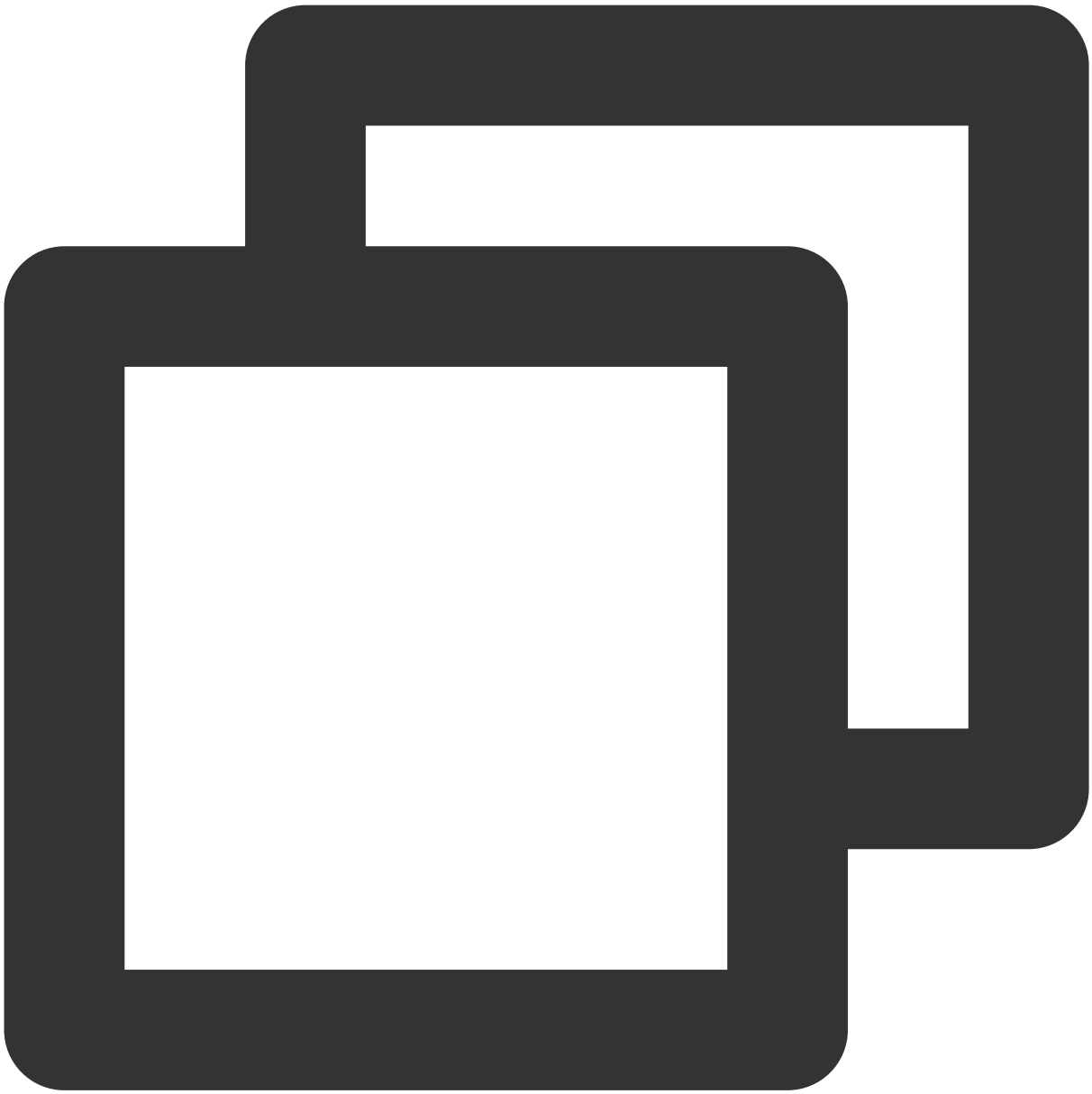
视图缩放



```
/*
* @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

更新日志

最近更新时间：2024-01-20 11:39:30

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

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：2024-01-20 11:52:52

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

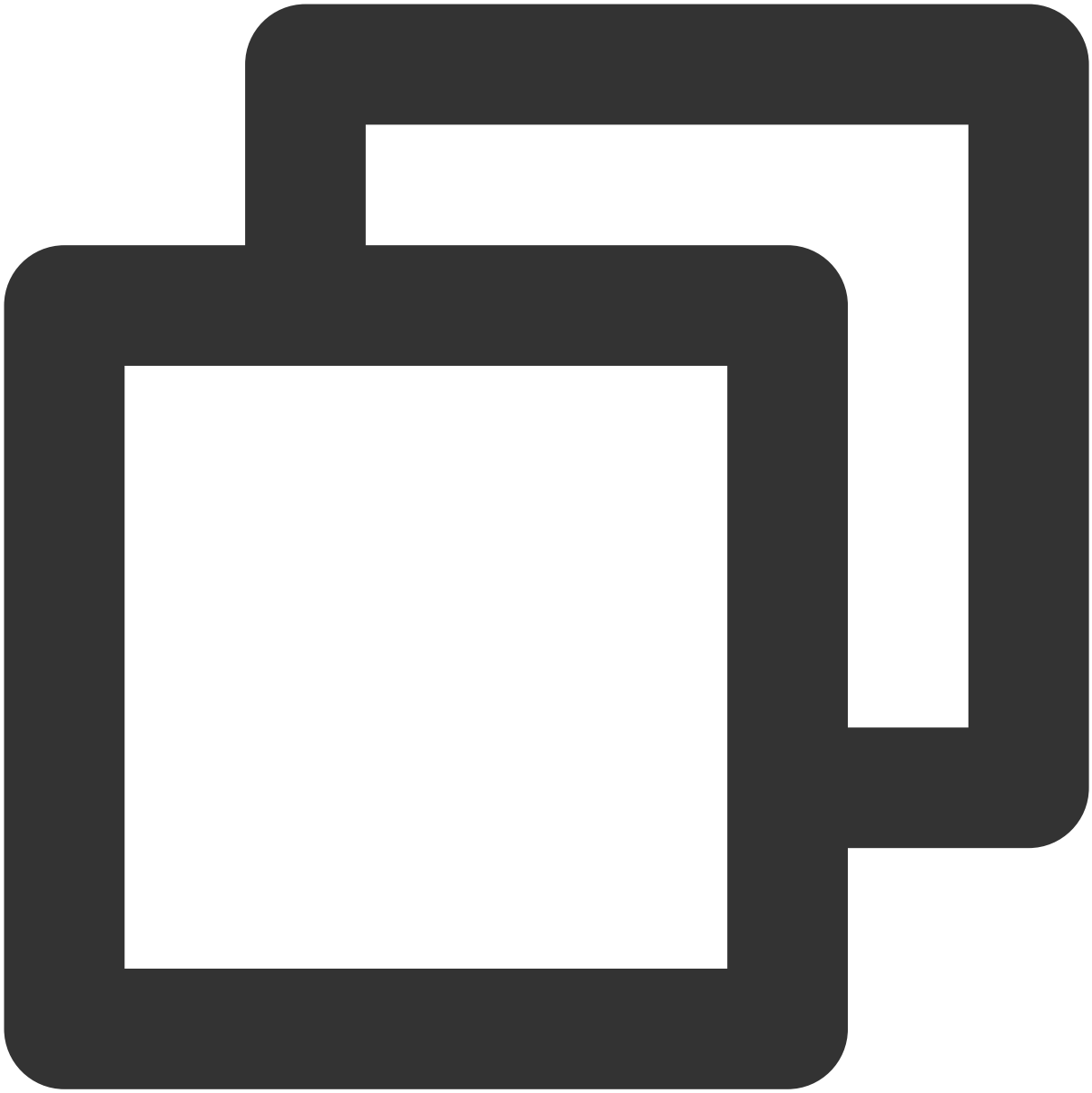
调用接口会产生哈希值计算费用和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持处理 COS 上的归档存储类型和深度归档存储类型的文件，如果需要处理此类型文件，请先 恢复归档文件。

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。

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。

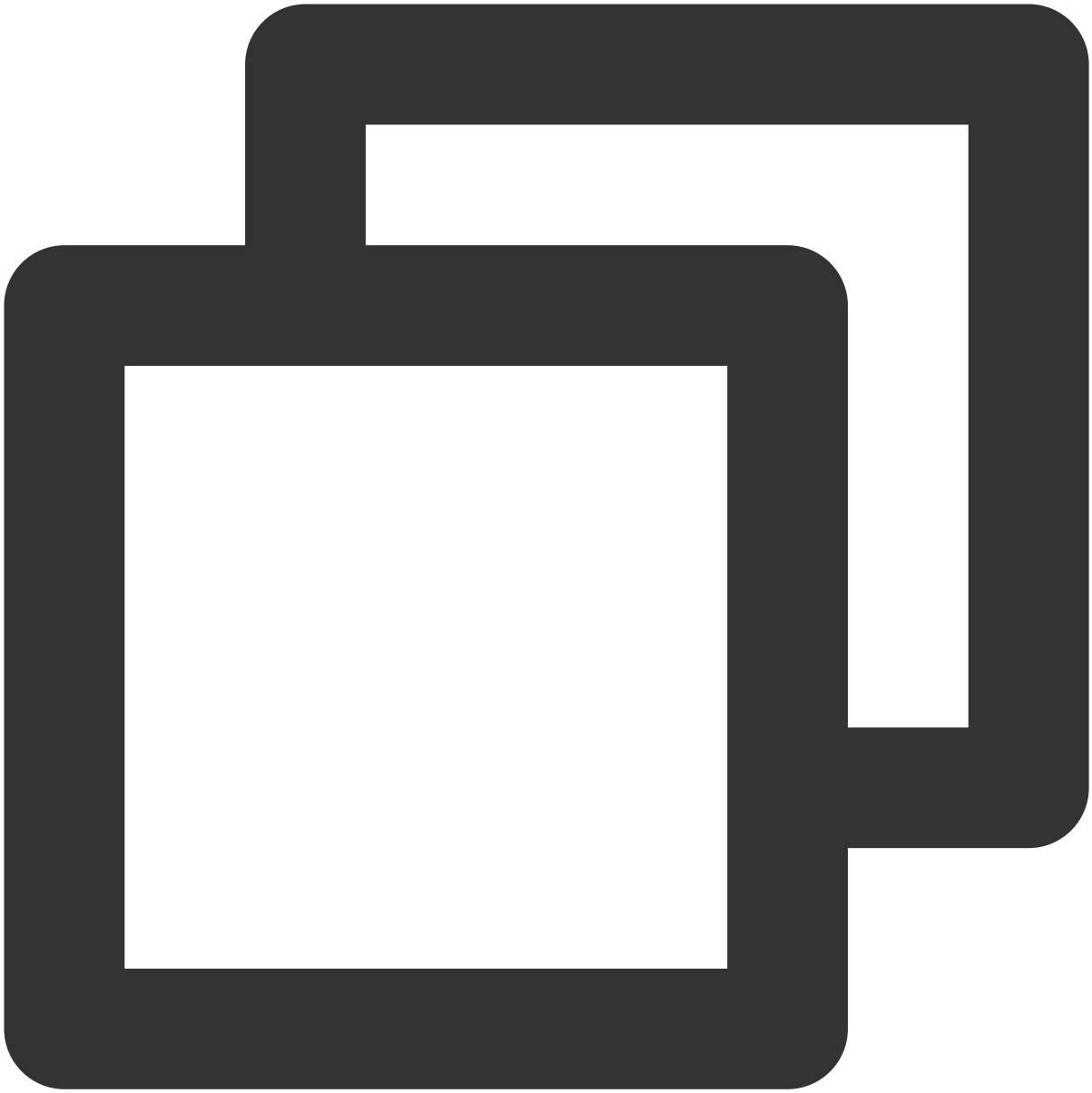
响应

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>
        <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

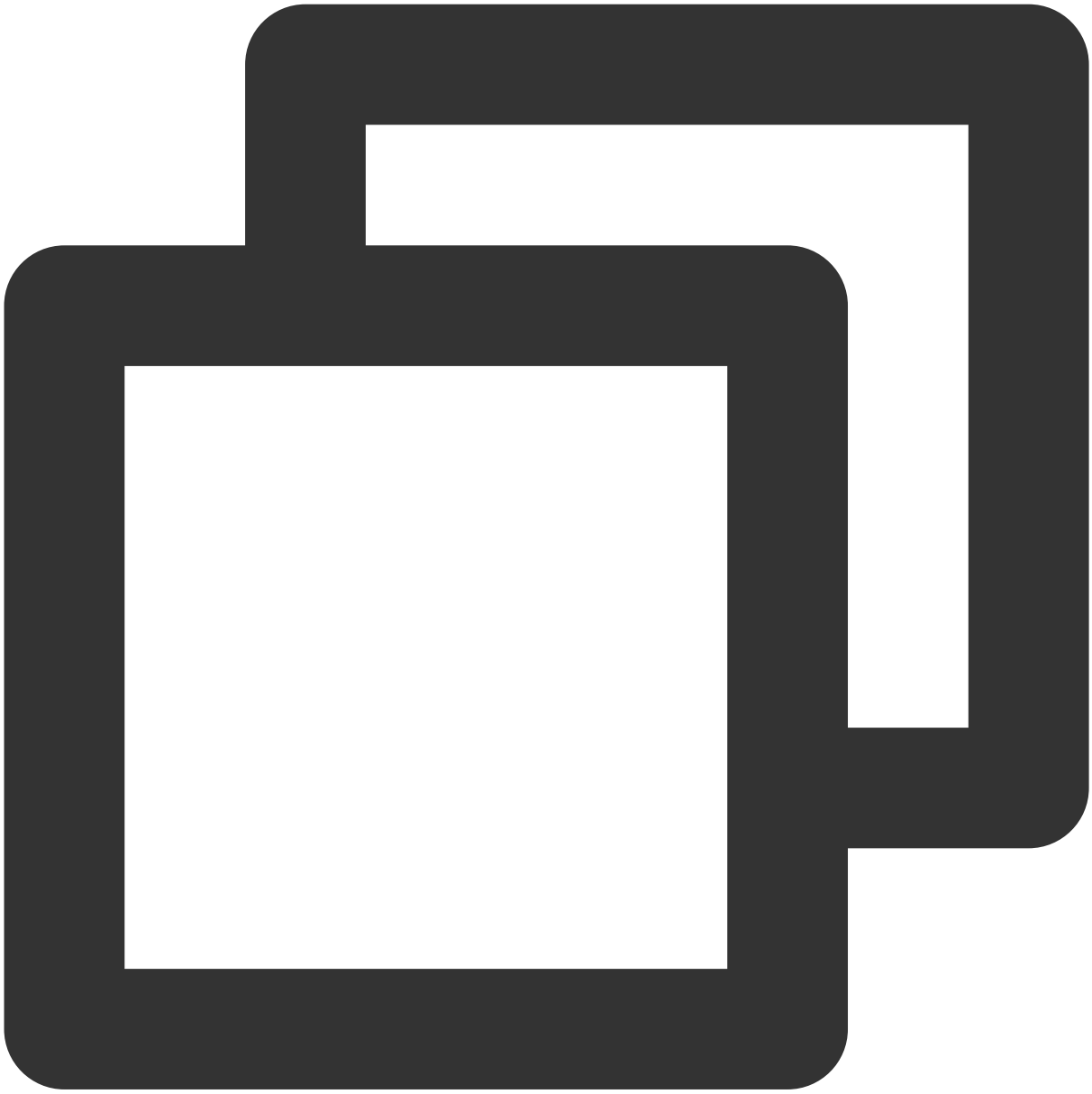
错误码

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。

版权所有：腾讯云计算（北京）有限责任公司

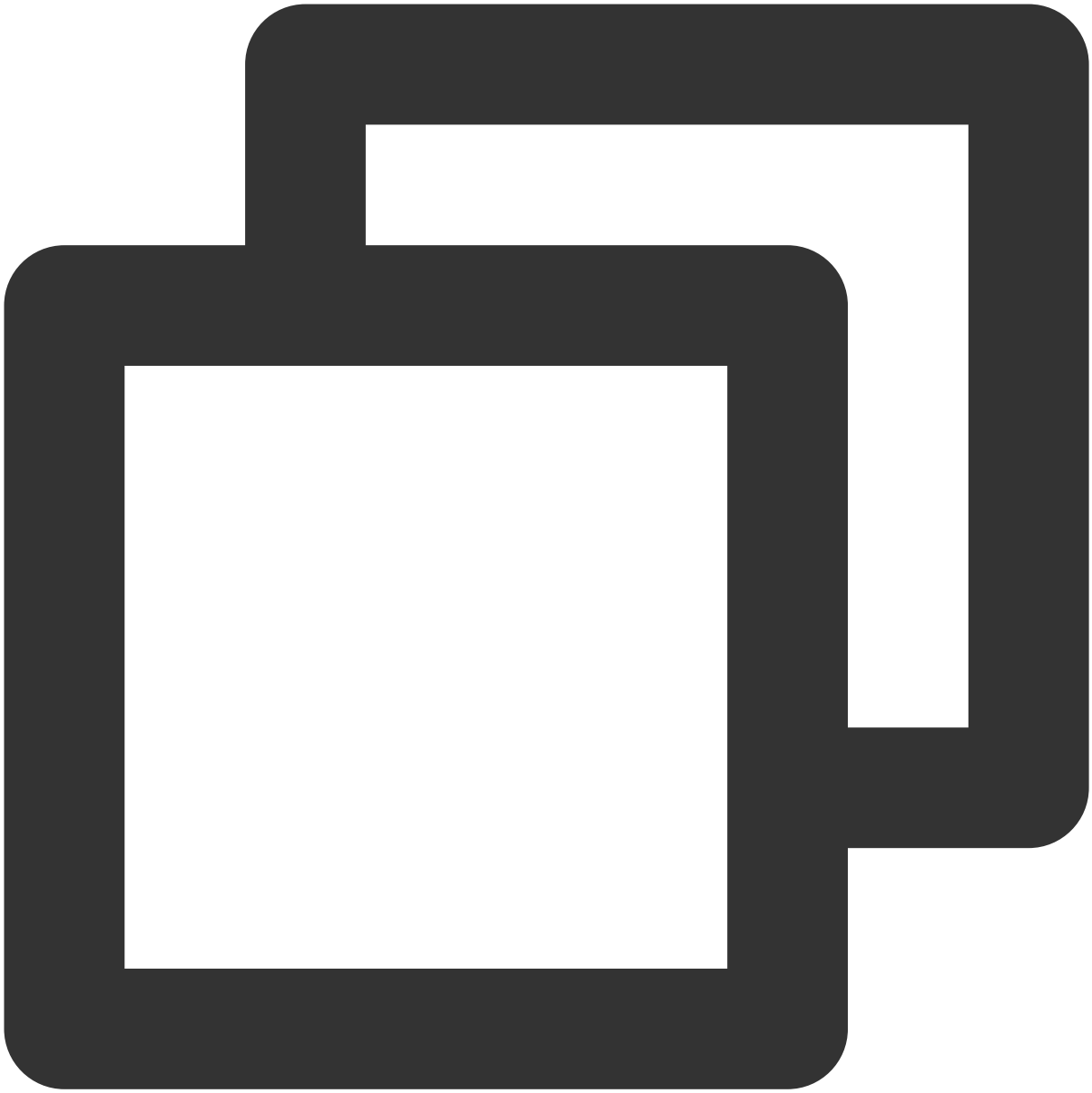
实际案例

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****-si
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>
```

提交哈希值计算任务

最近更新时间：2024-01-20 11:52:52

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

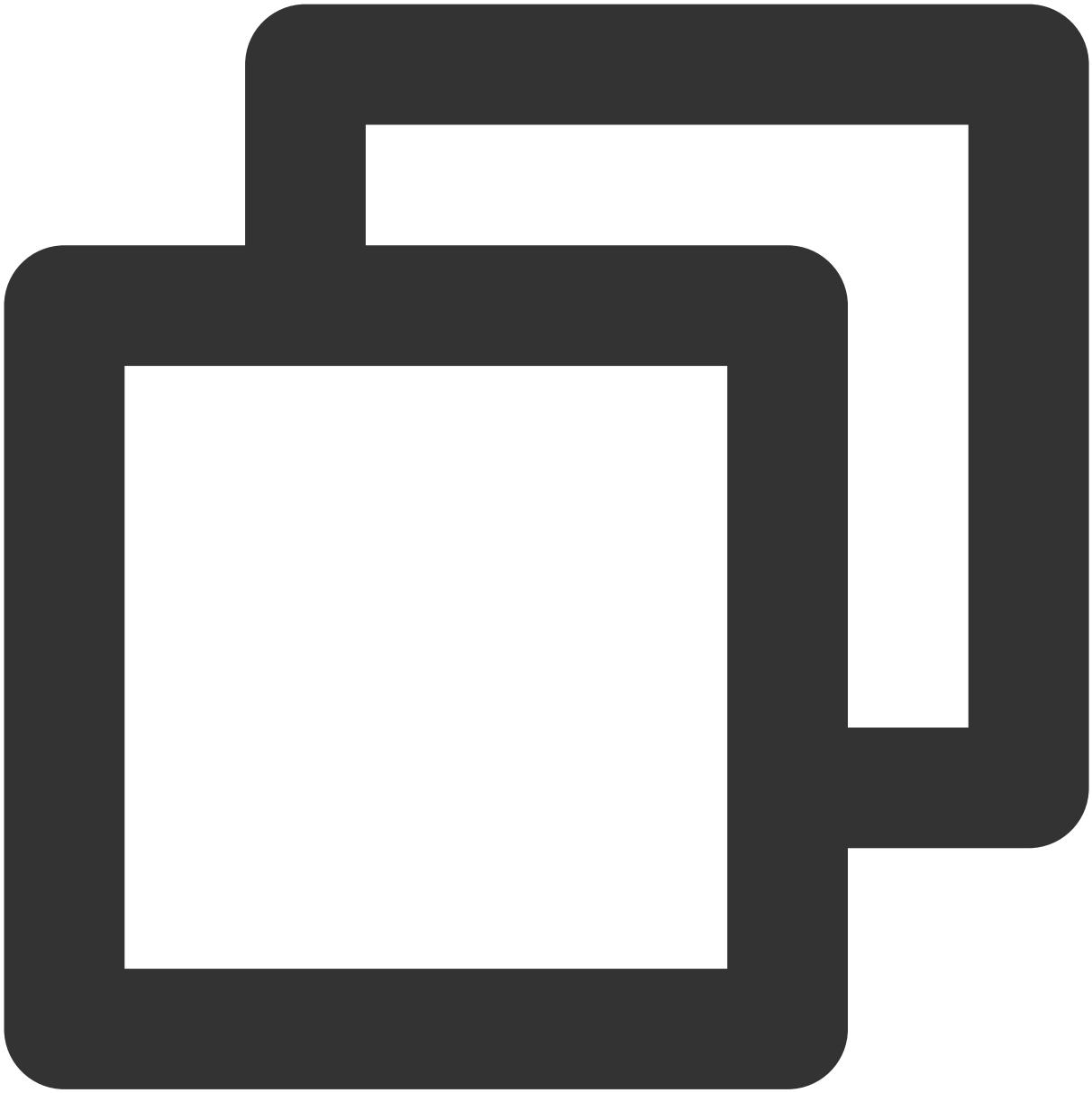
调用接口会产生哈希值计算费用和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持计算 COS 上的归档存储类型和深度归档存储类型的文件哈希值，如果需要计算此类型文件的哈希值，请先 恢复归档文件。

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：

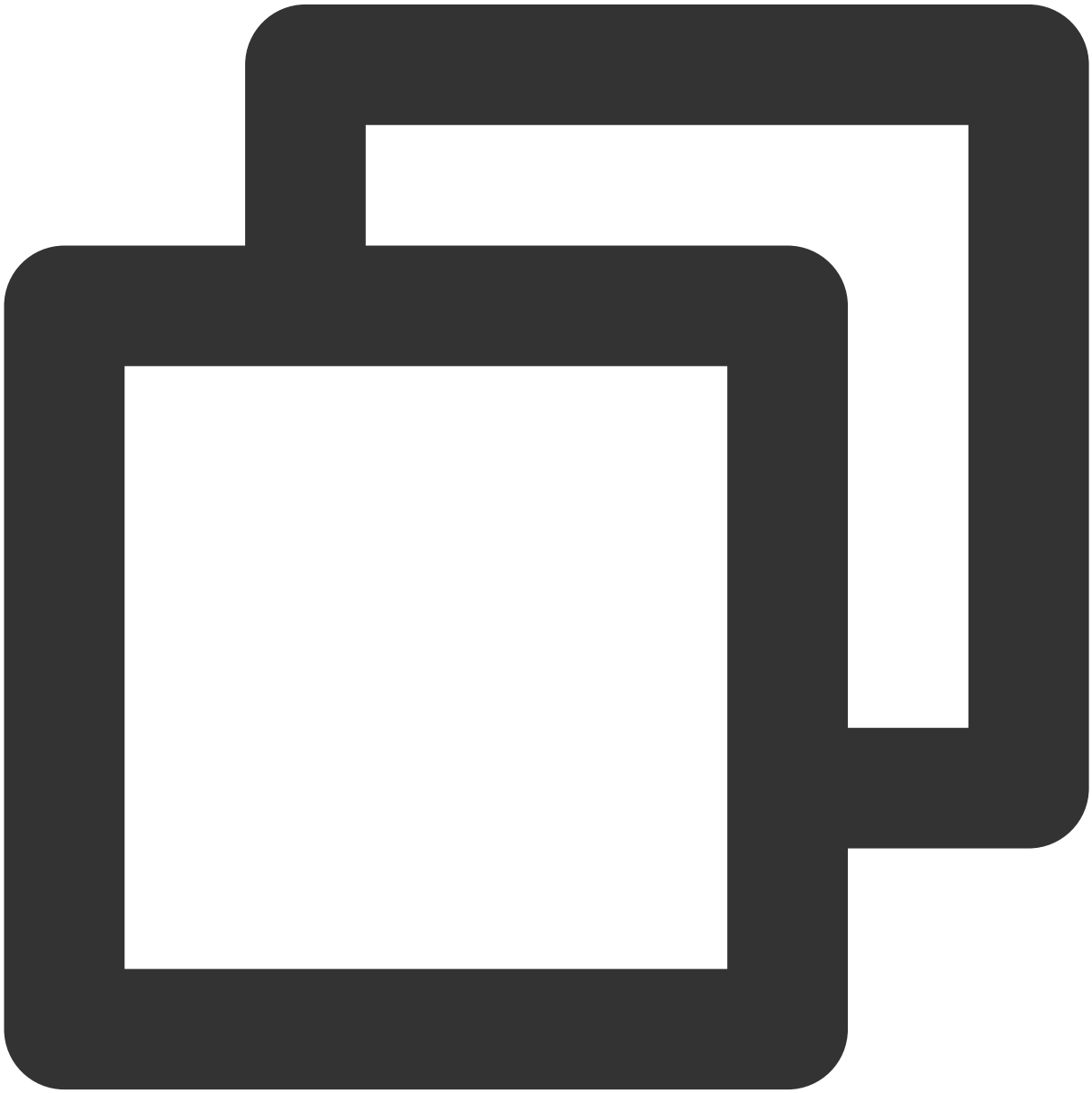
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

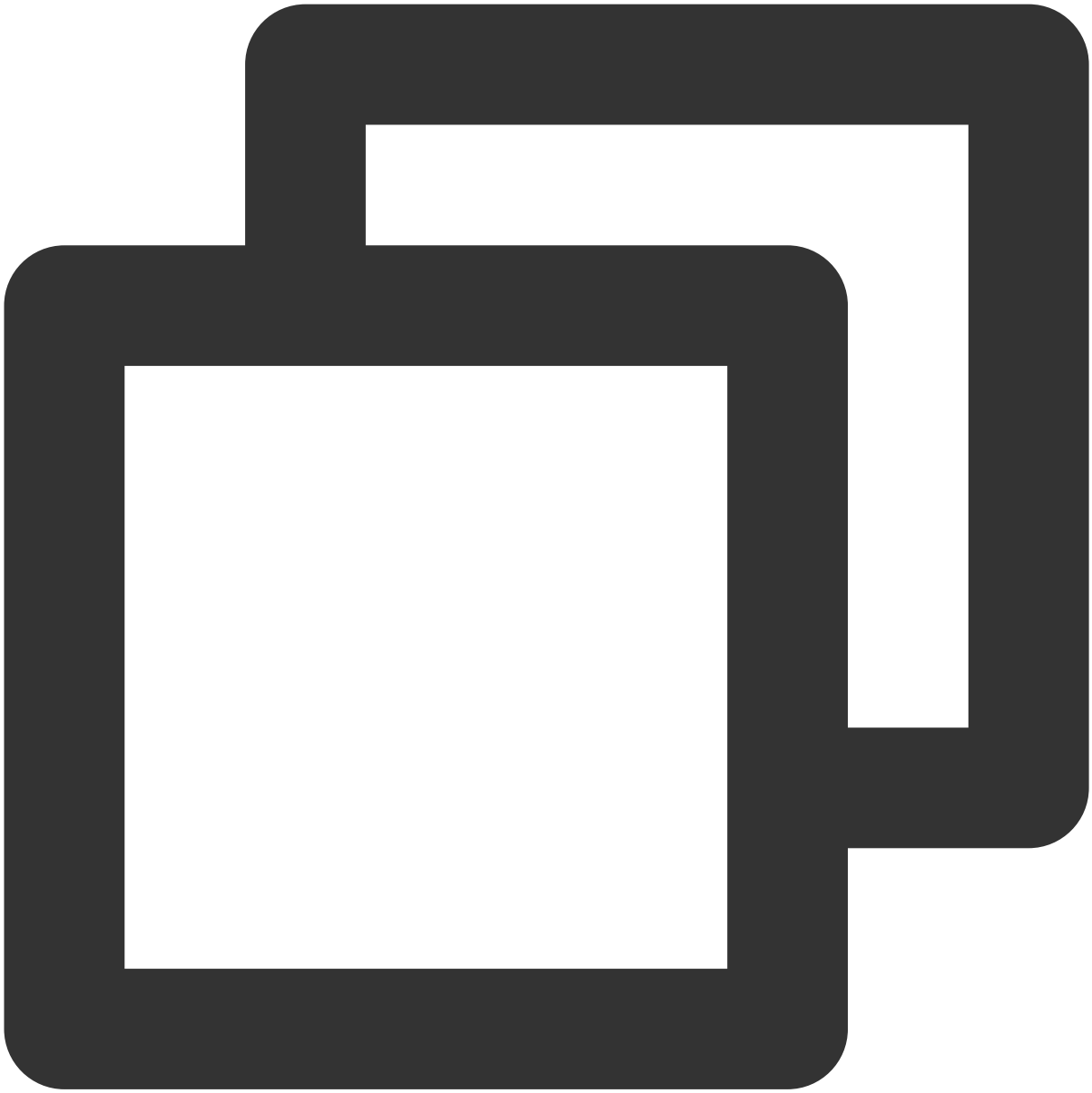
响应

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
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

版权所有：腾讯云计算（北京）有限责任公司

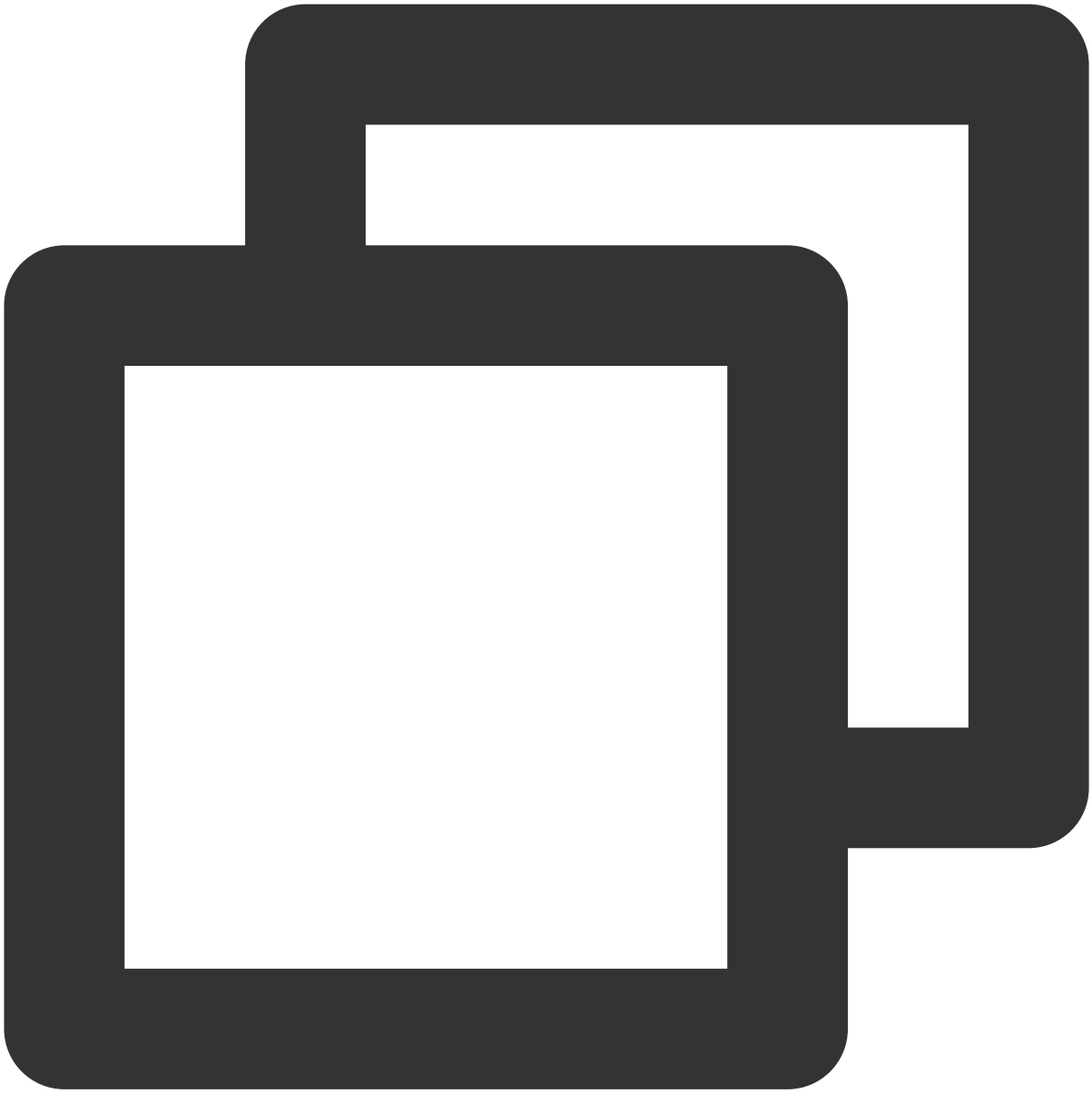
LastModified	Response.JobsDetail.Operation.FileHashCodeResult	文件的最后修改时间。	String
Etag	Response.JobsDetail.Operation.FileHashCodeResult	文件的Etag。	String

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

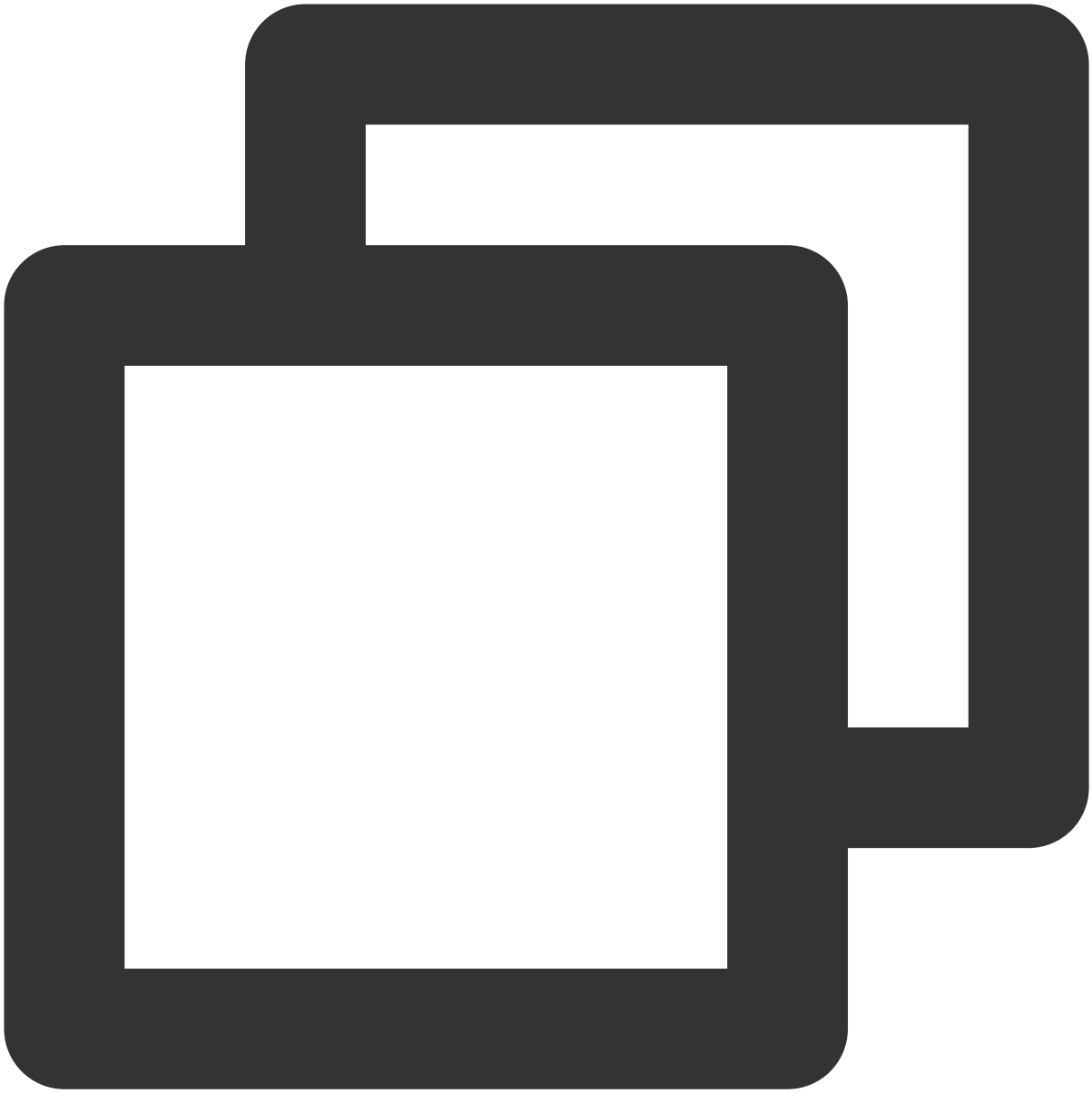


```
POST /file_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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

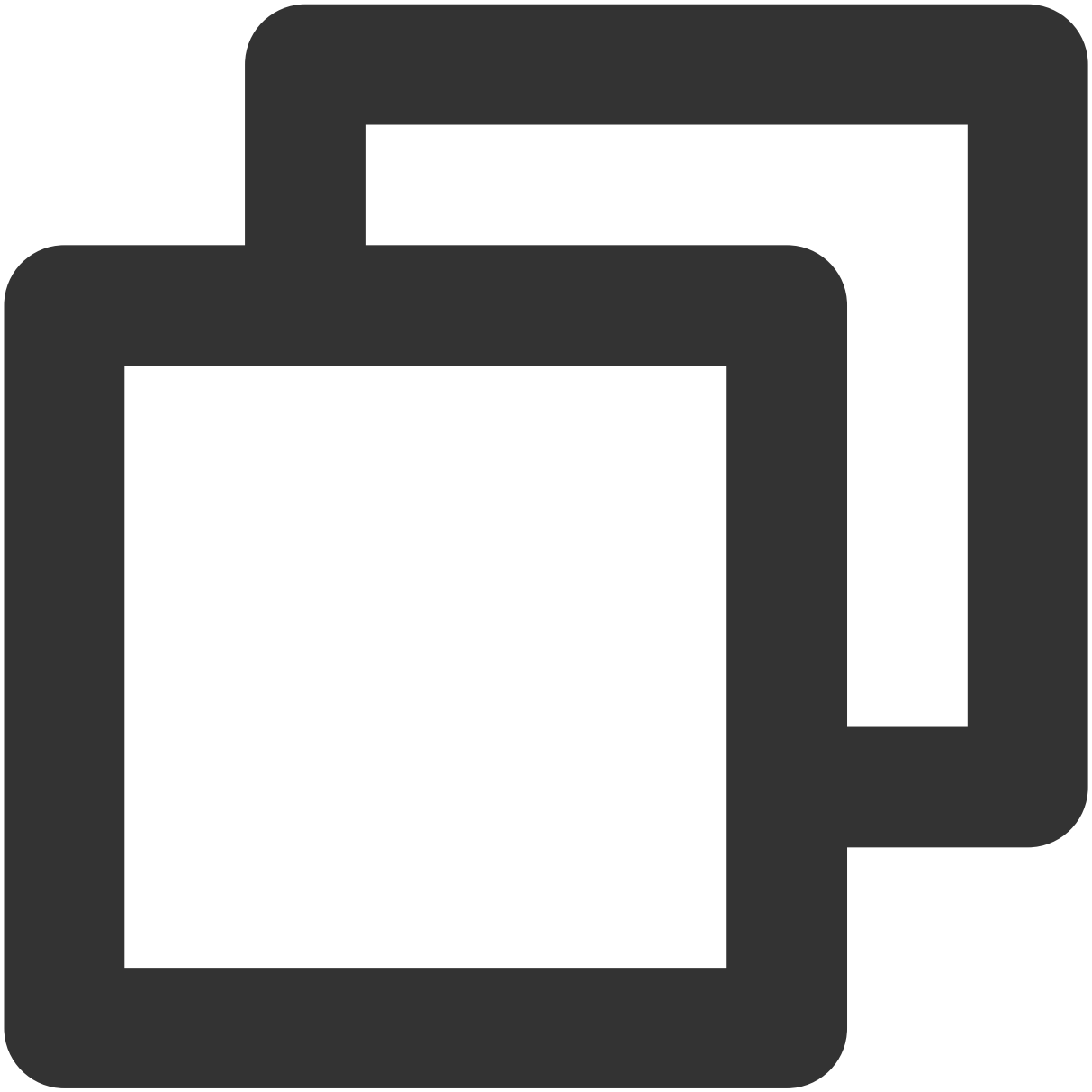
最近更新时间：2024-01-20 11:52:52

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。

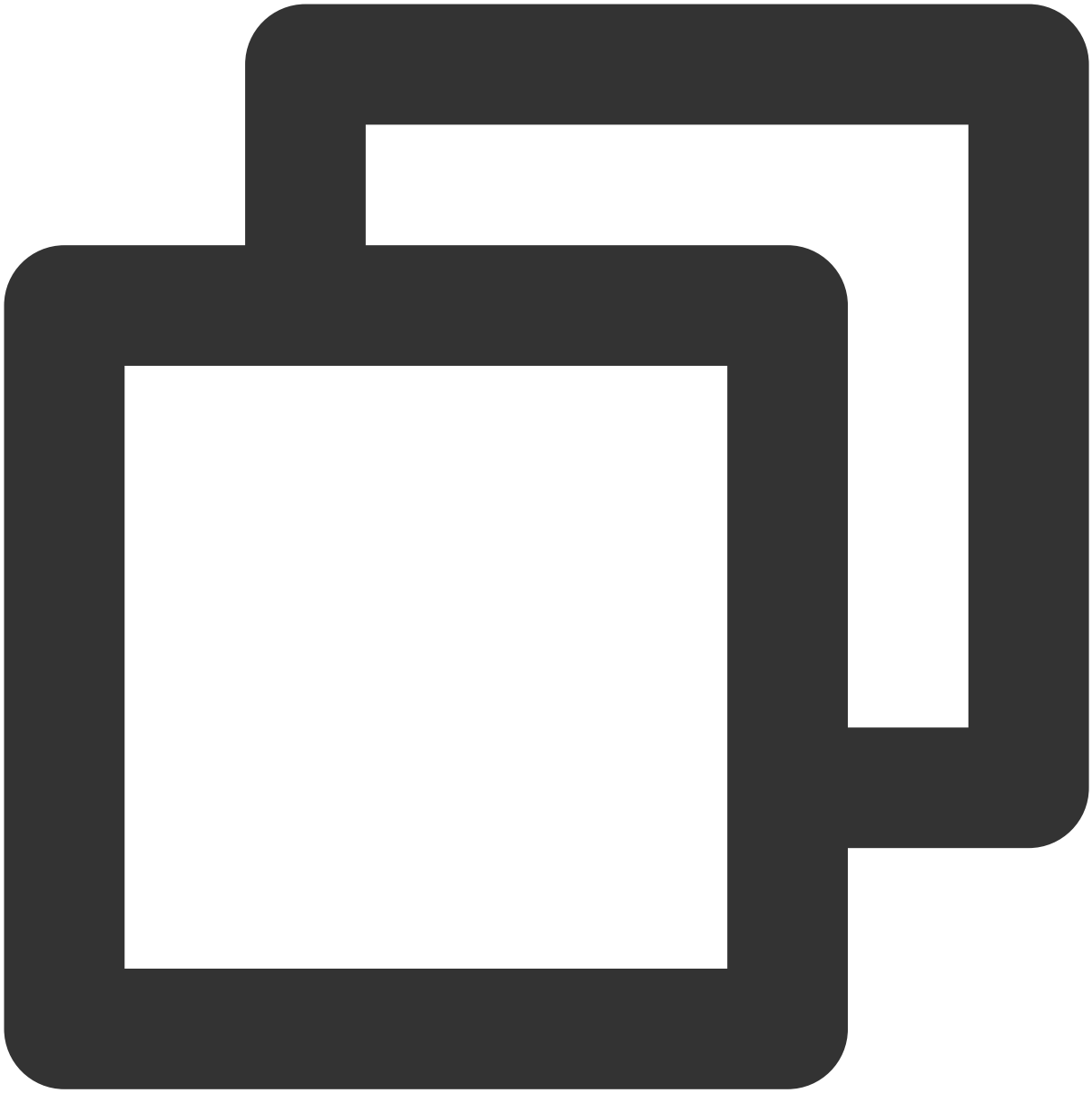
响应

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

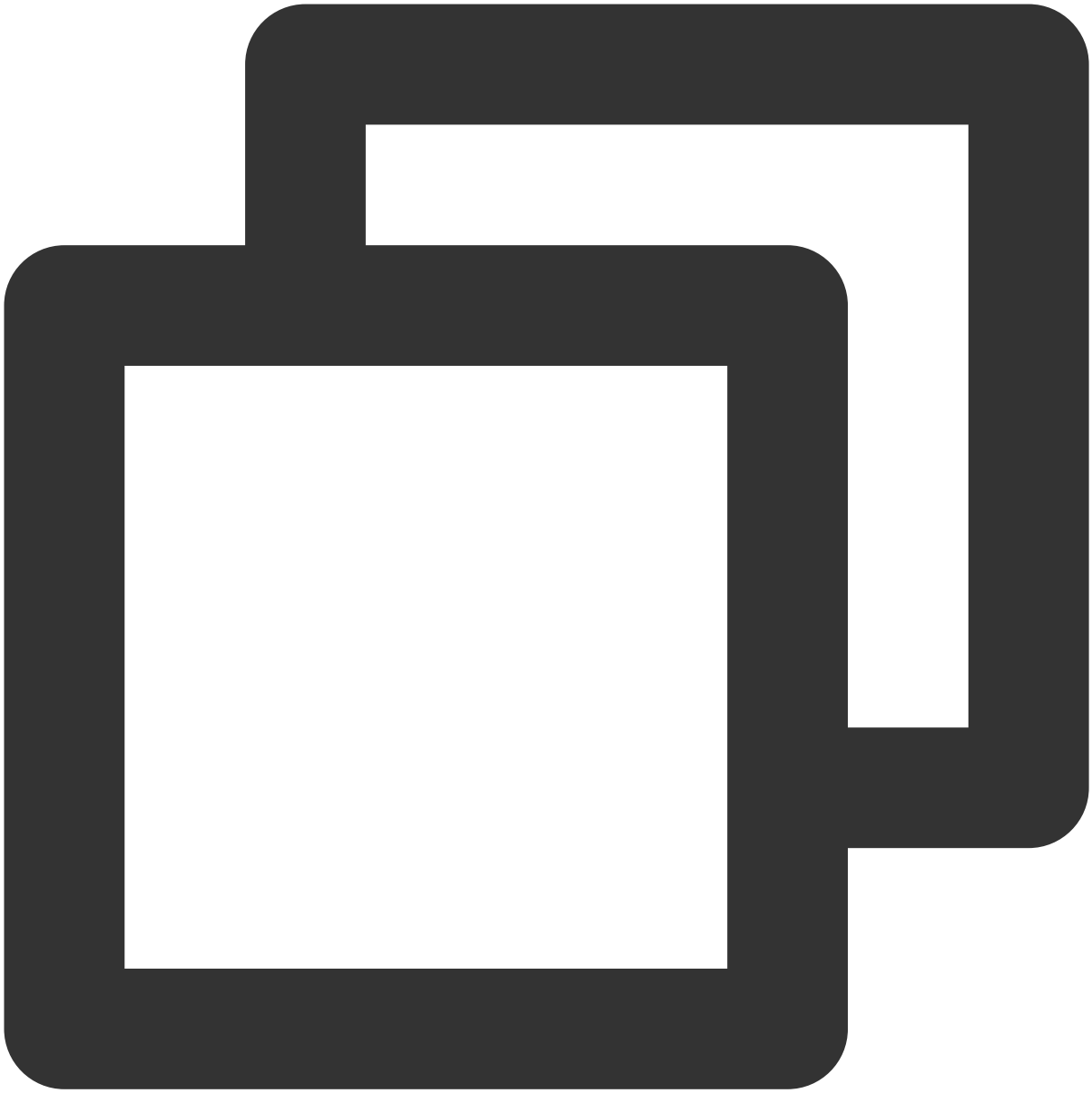
具体参数值，请前往 文件哈希值计算 `JobsDetail` 查看。

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。

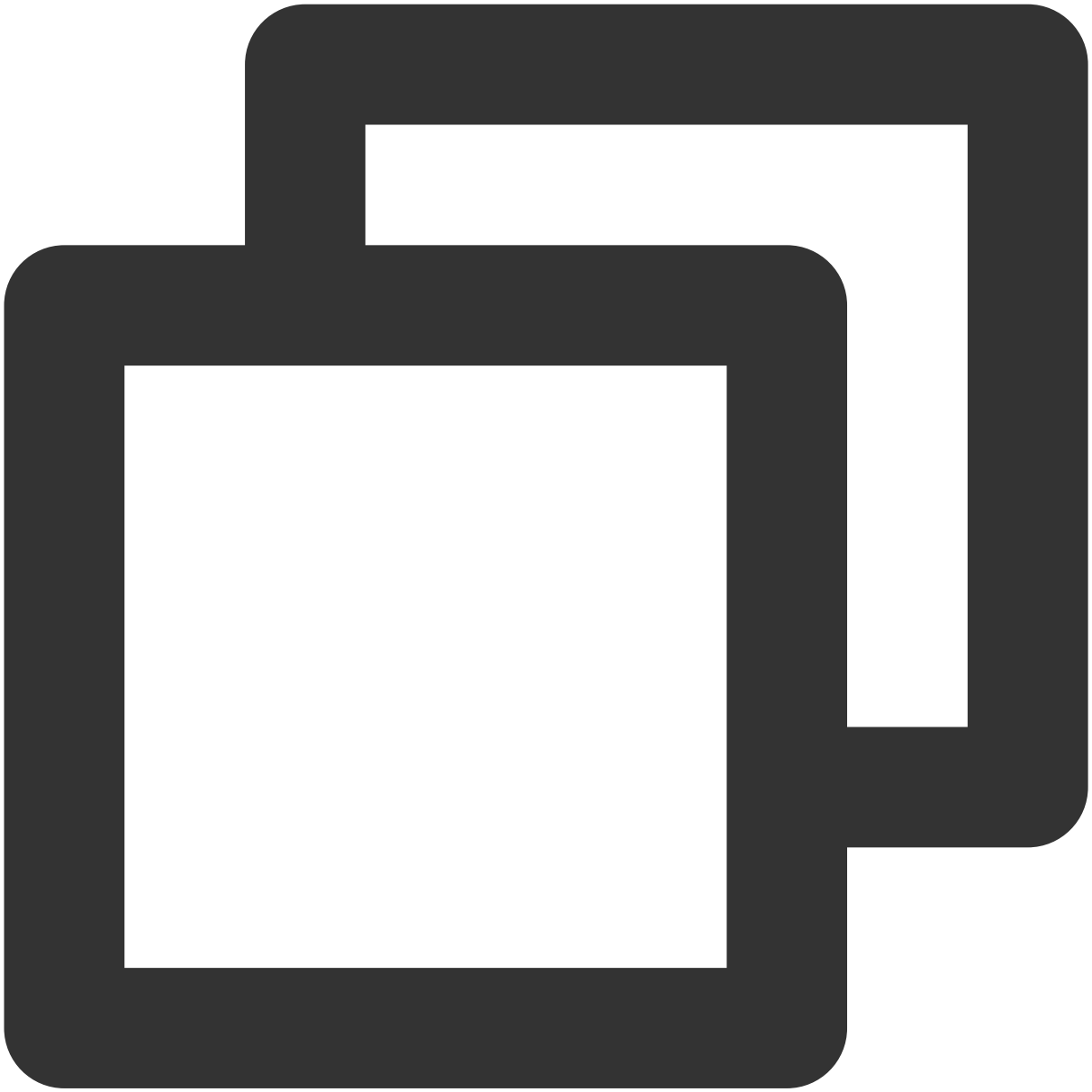
实际案例

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：2024-01-20 11:52:52

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

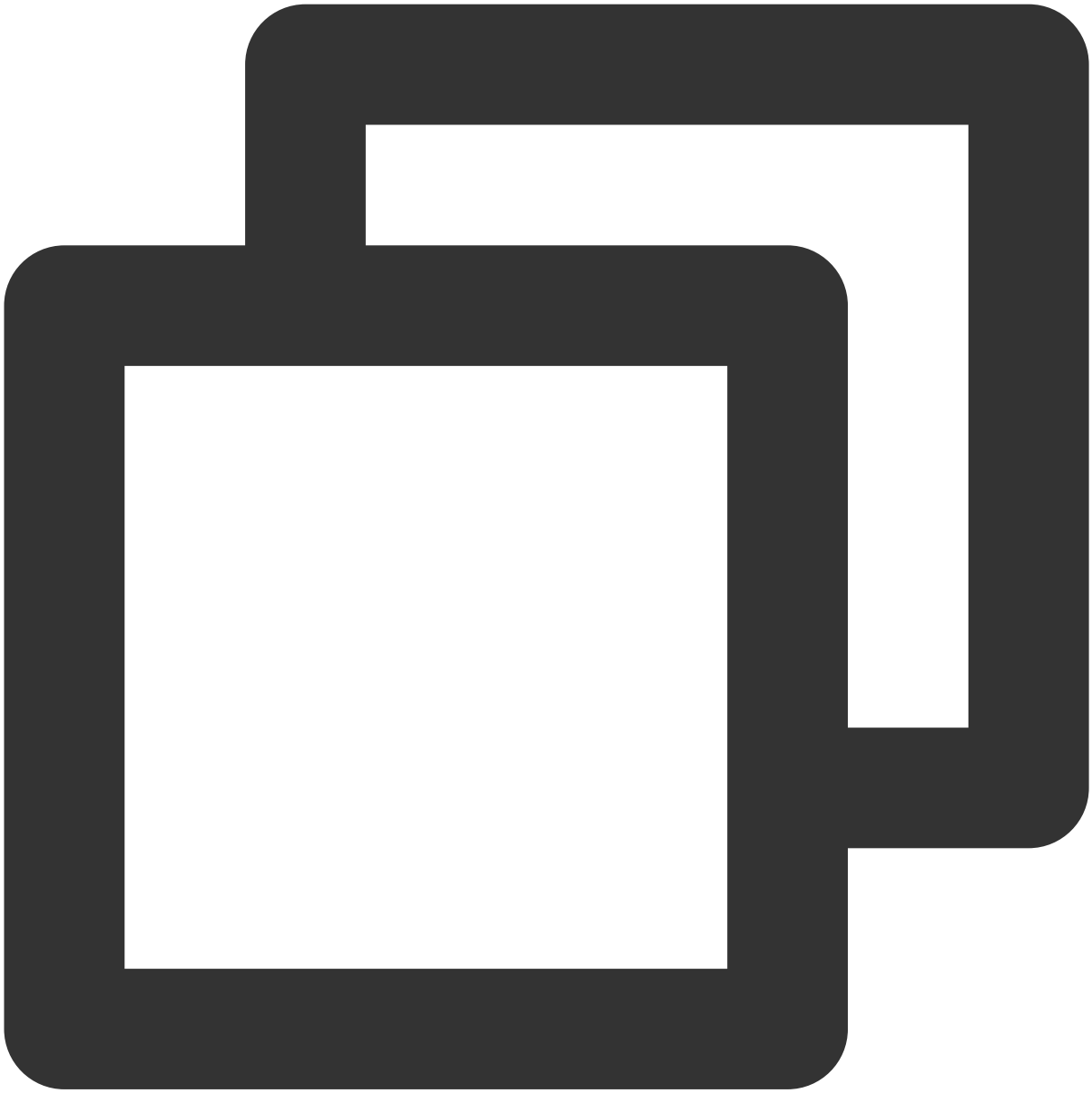
调用接口会产生文件解压费用和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持解压 COS 上的归档存储类型和深度归档存储类型的文件，如果解压需要此类型文件，请先 恢复归档文件。

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：

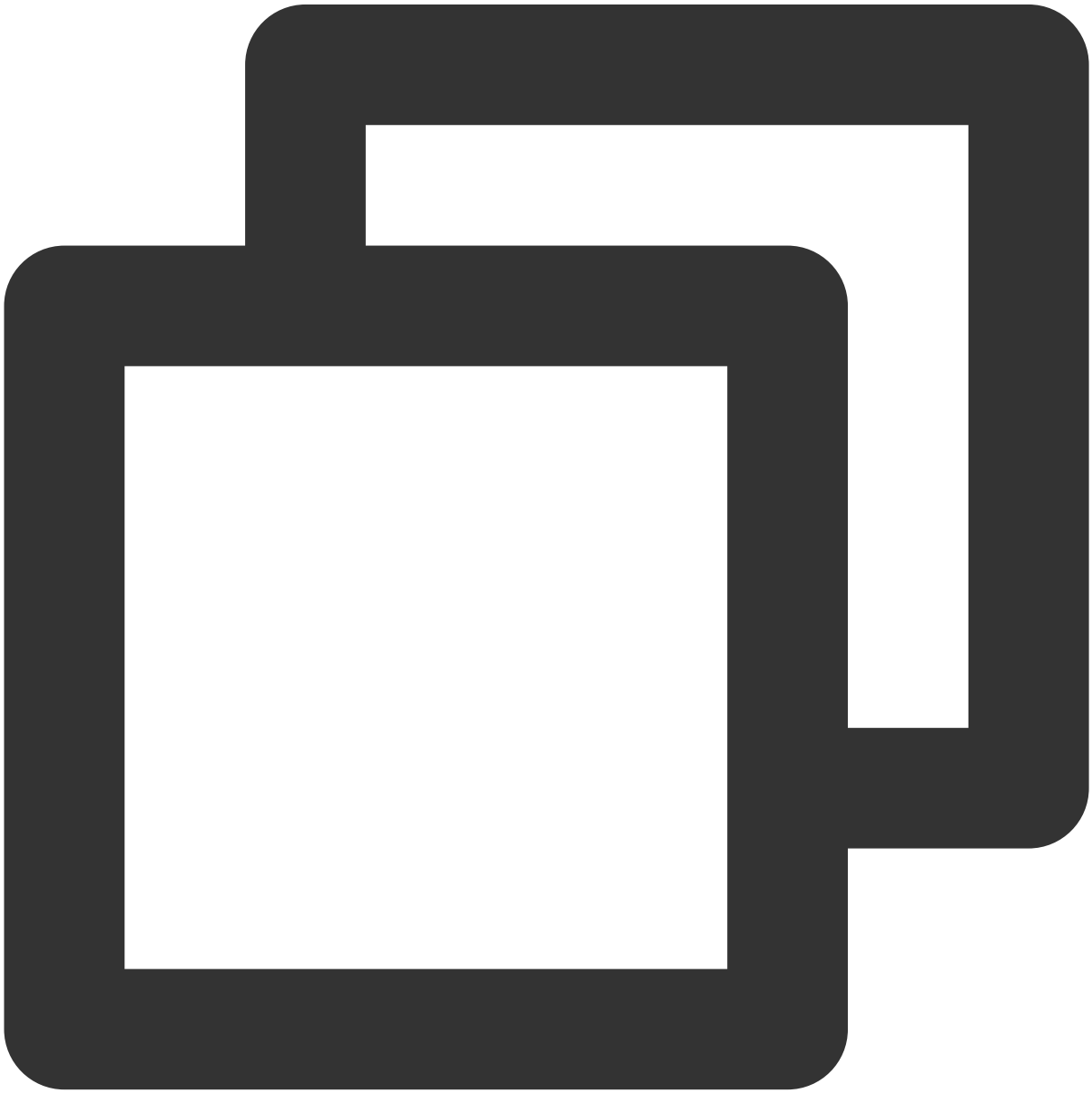
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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址。	String	否

版权所有：腾讯云计算（北京）有限责任公司

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）。 - 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。	String	否

		- 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。		

说明：

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

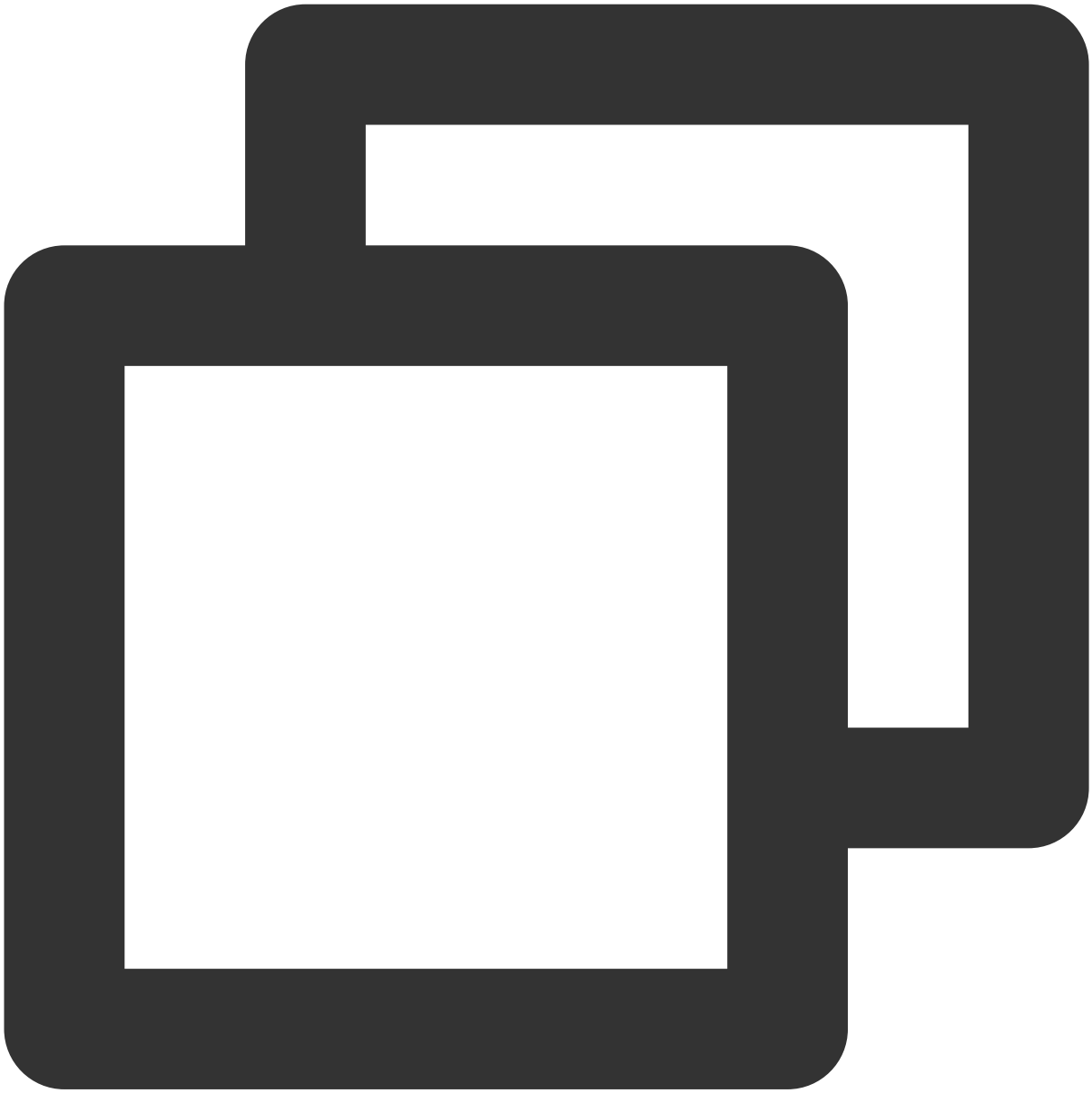
响应

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

		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

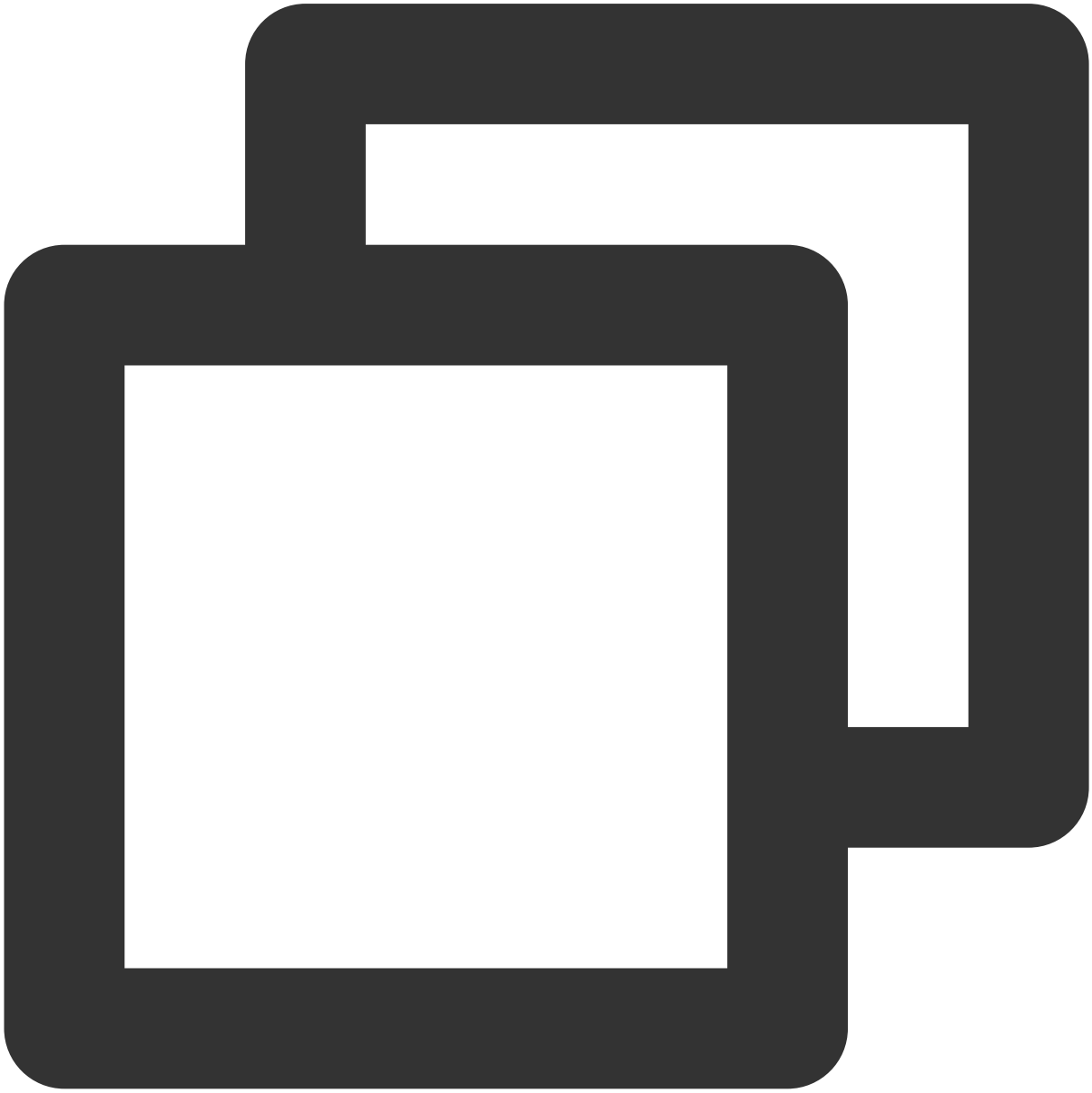
FileCount	Response.JobsDetail.Operation.FileUncompressResult	解压后文件的个数	String

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

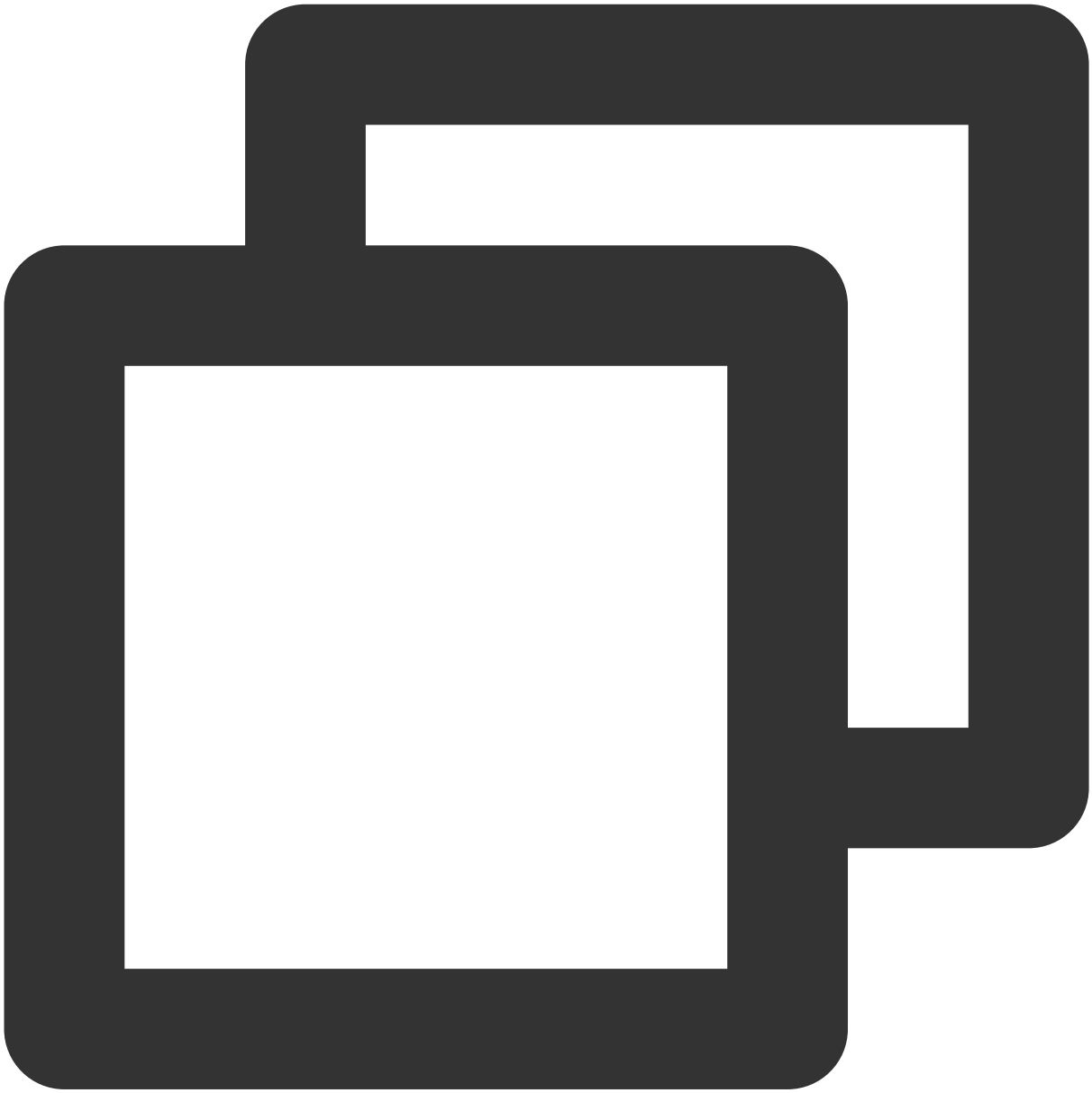
请求



```
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

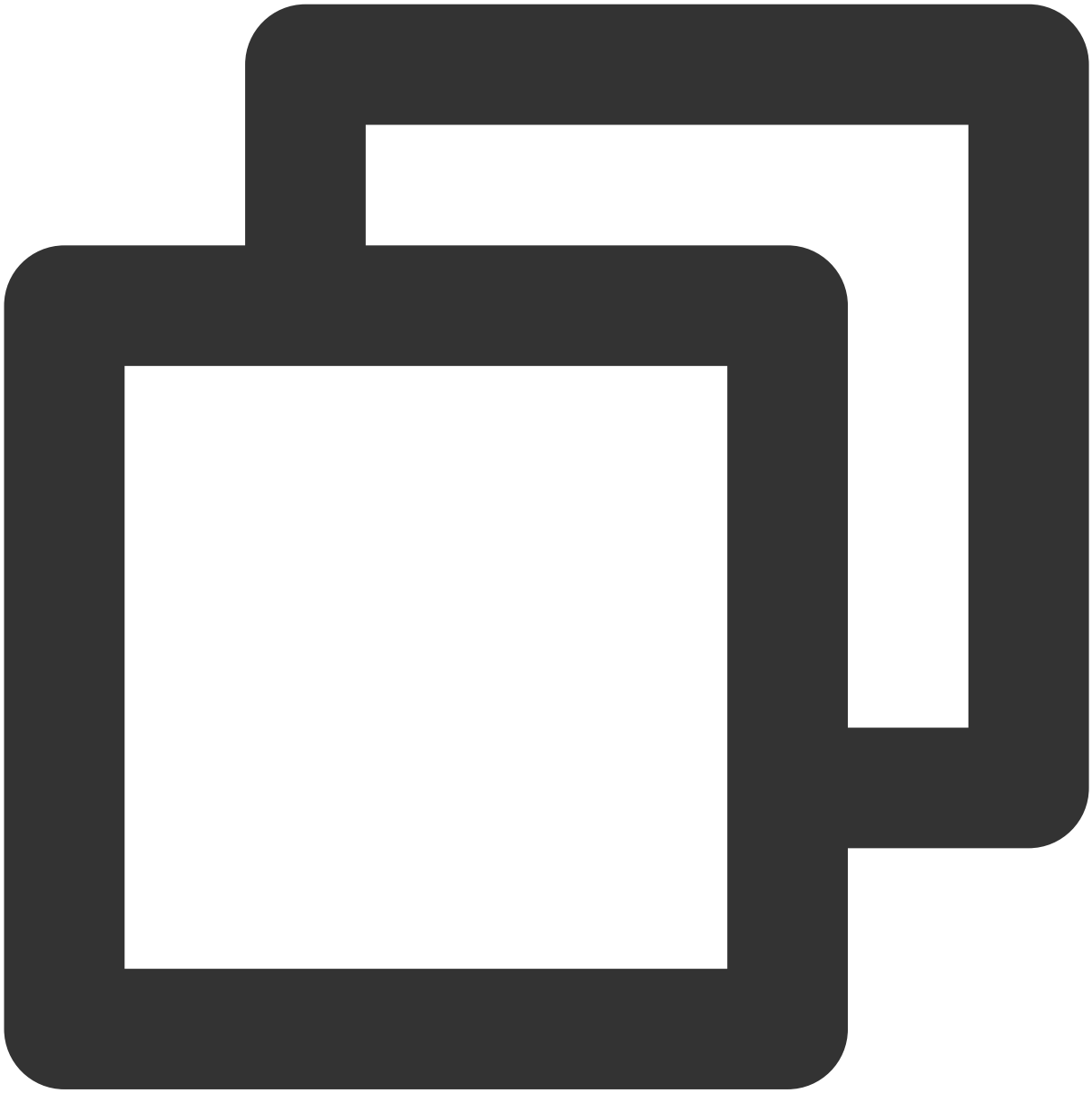
最近更新时间：2024-01-20 11:52:52

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。

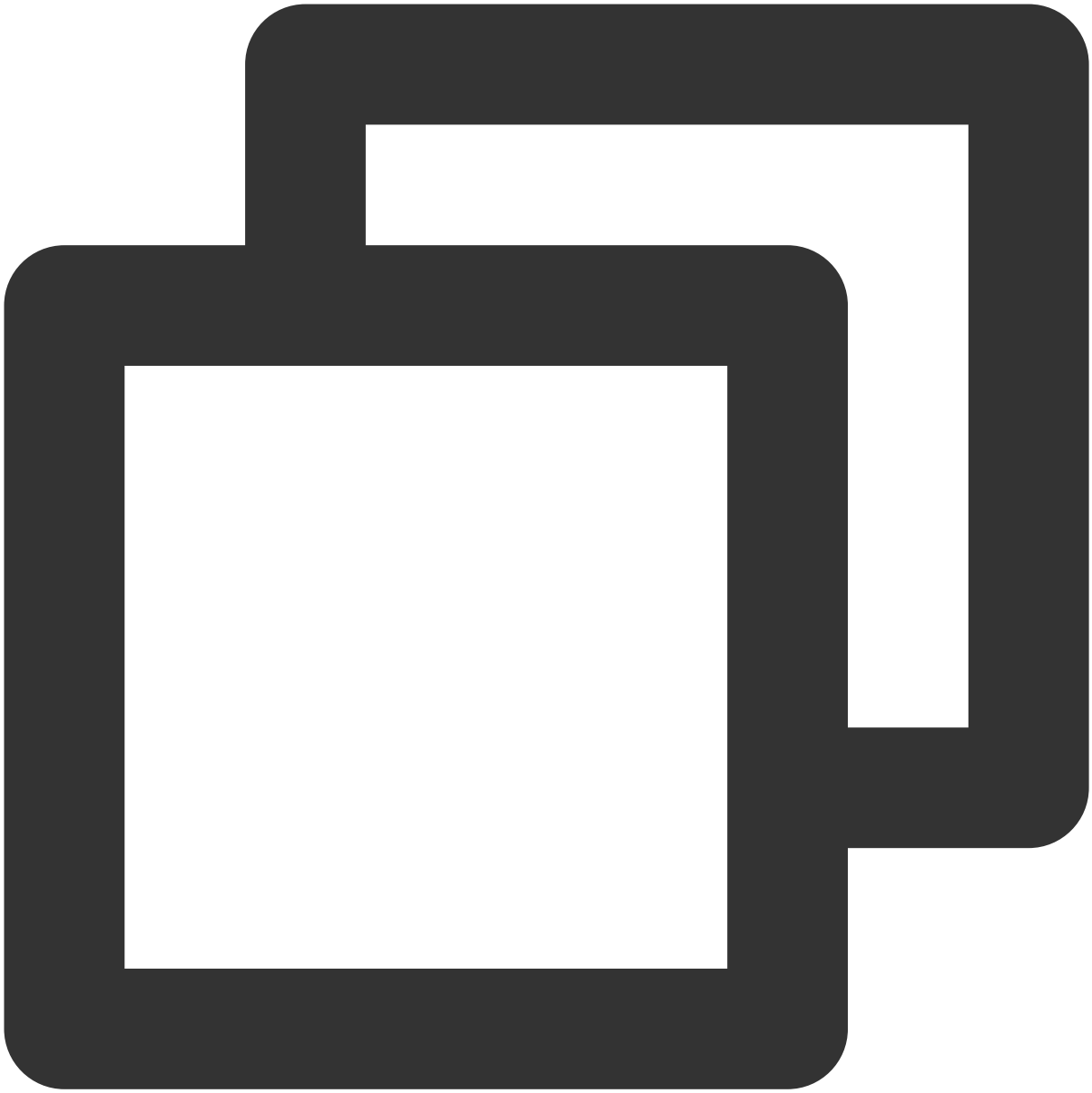
响应

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

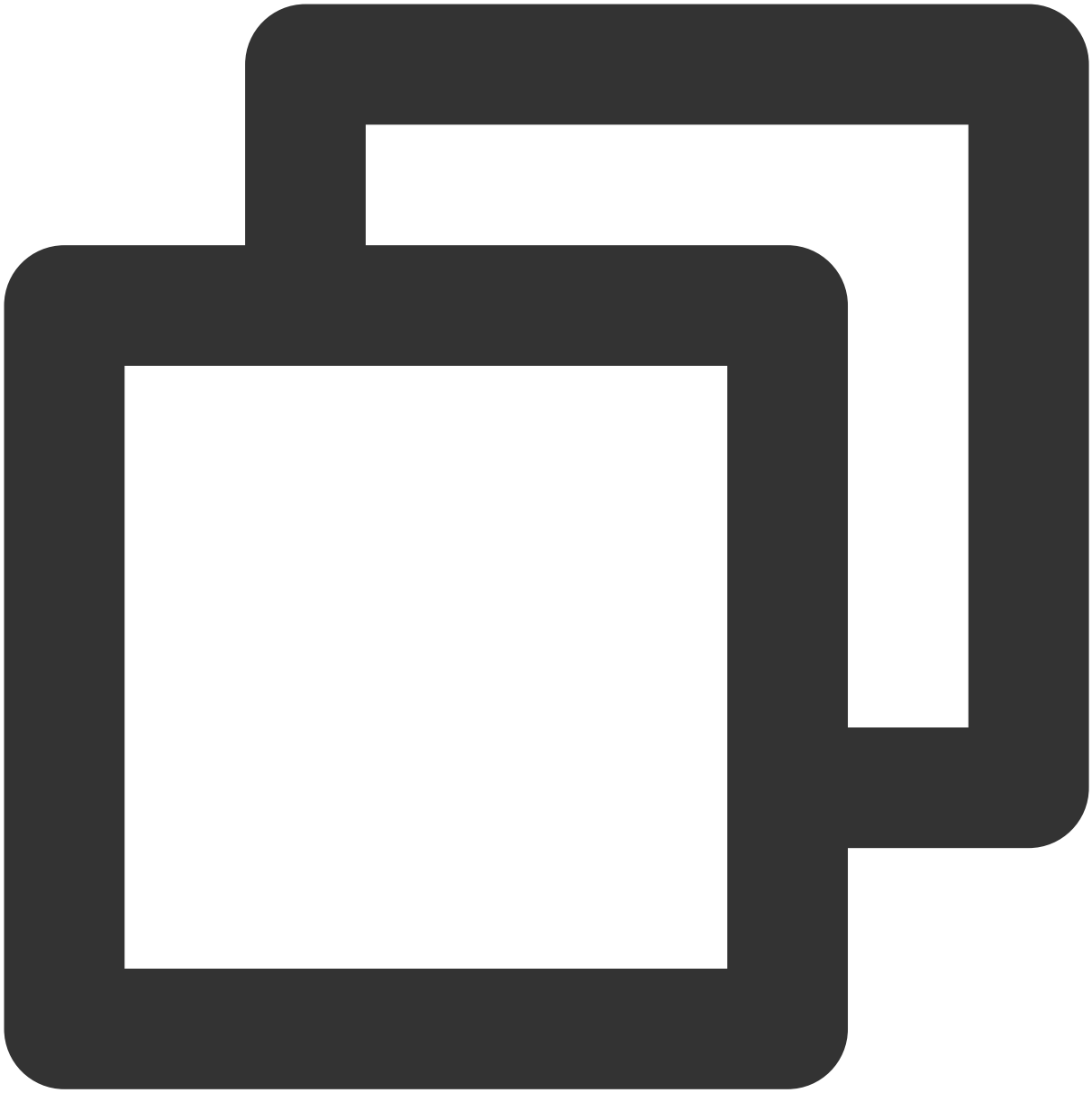
具体参数值，请前往 文件解压 `JobsDetail` 查看。

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。

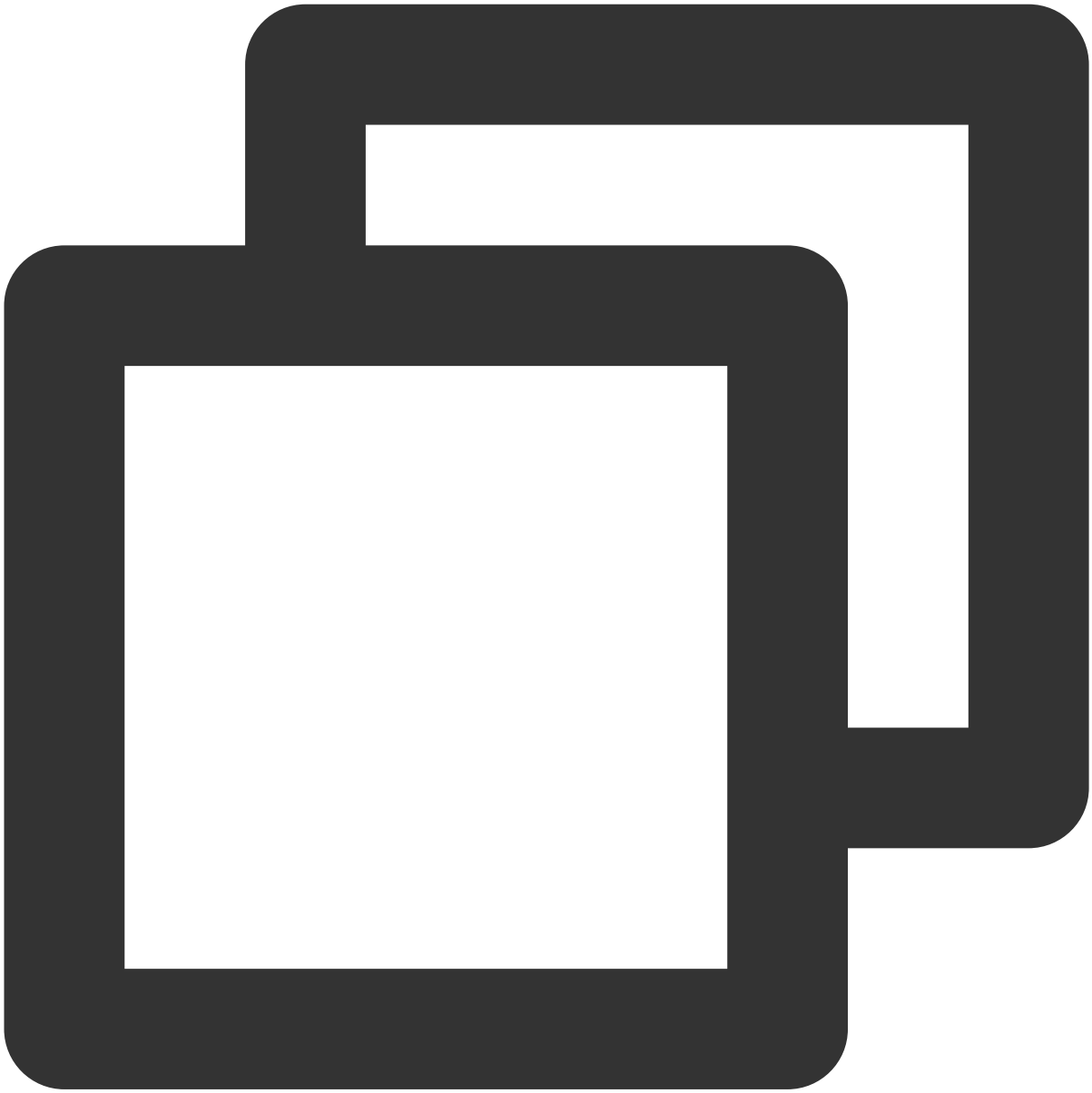
实际案例

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：2024-01-20 11:52:52

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

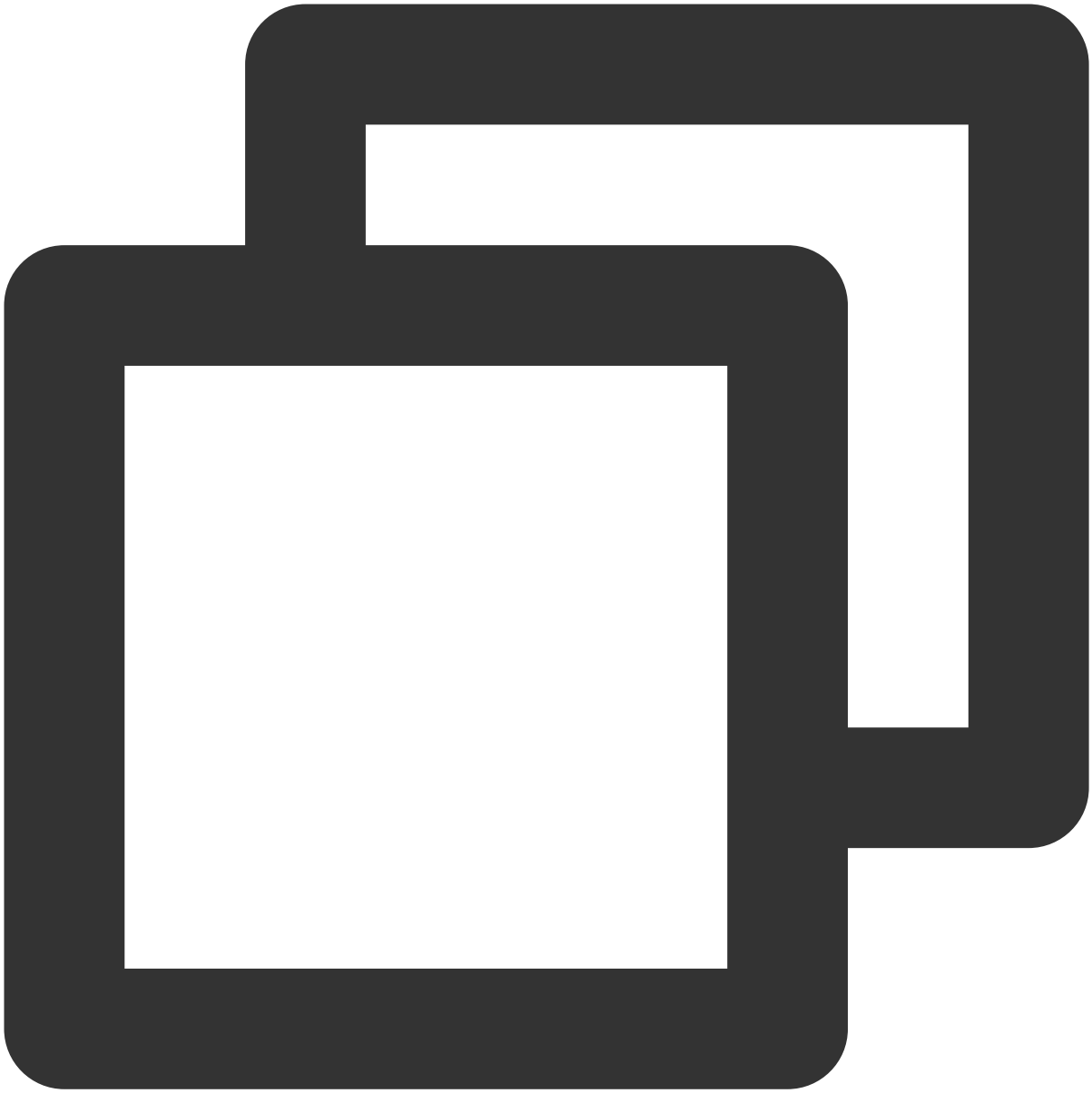
调用接口会产生多文件打包压缩费用和 COS 读请求费用。
如果文件为 COS 上的低频存储类型，调用会产生 COS 低频数据取回费用。
不支持打包 COS 上的归档存储类型和深度归档存储类型的文件，如果打包需要此类型文件，请先 恢复归档文件。

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。
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：

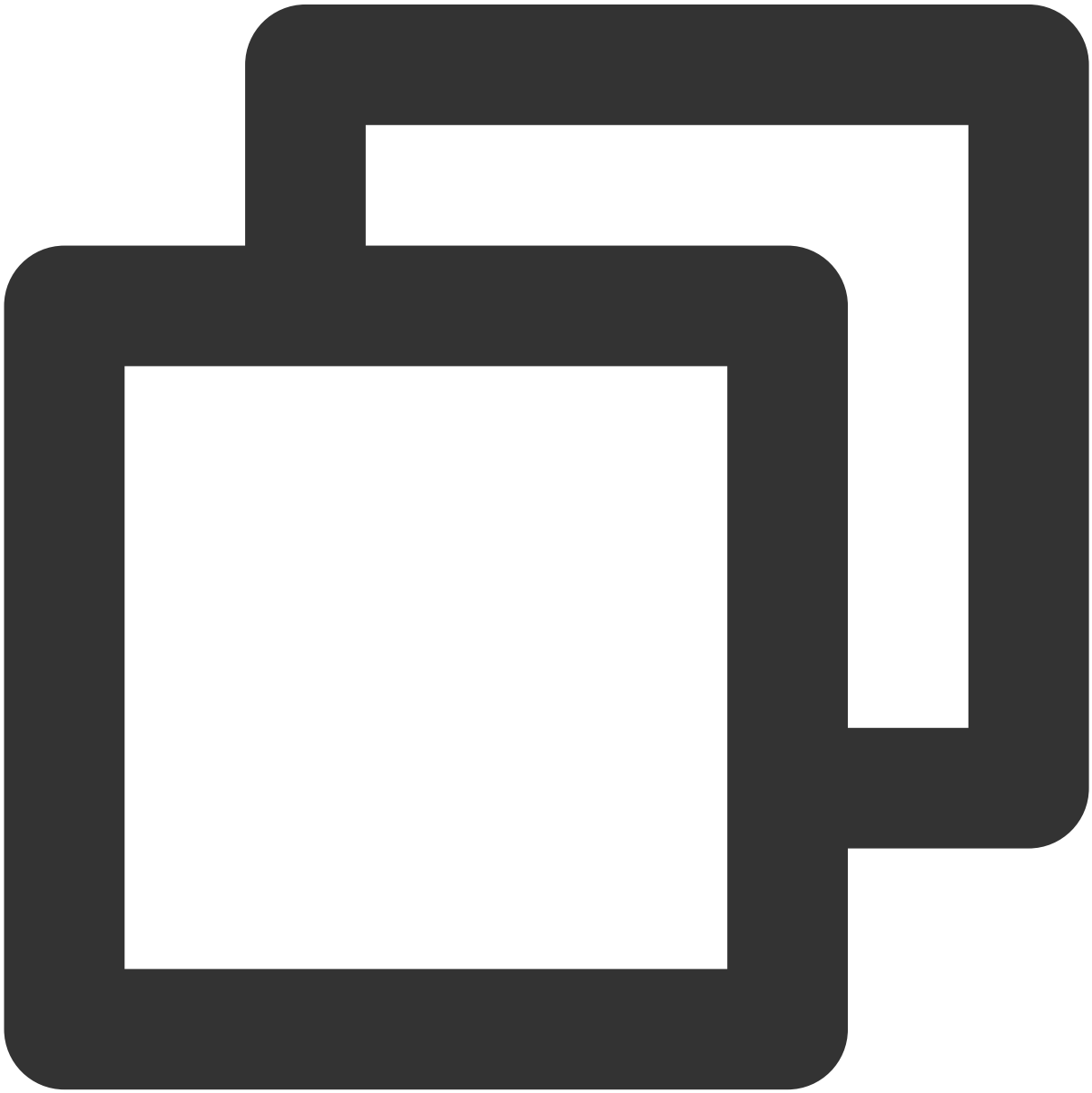
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>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，多文件打包压缩默认为：FileCompress。	String	是
Operation	Request	包含文件打包压缩的处理规则。	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。详情请参见 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	是否必选
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	是
Format	Request.Operation.FileCompressConfig	打包压缩的类型，有效值：zip、tar、tar.gz。	String	是
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。	String	否

		索引文件格式：仅支持 CSV 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	否
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	否

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

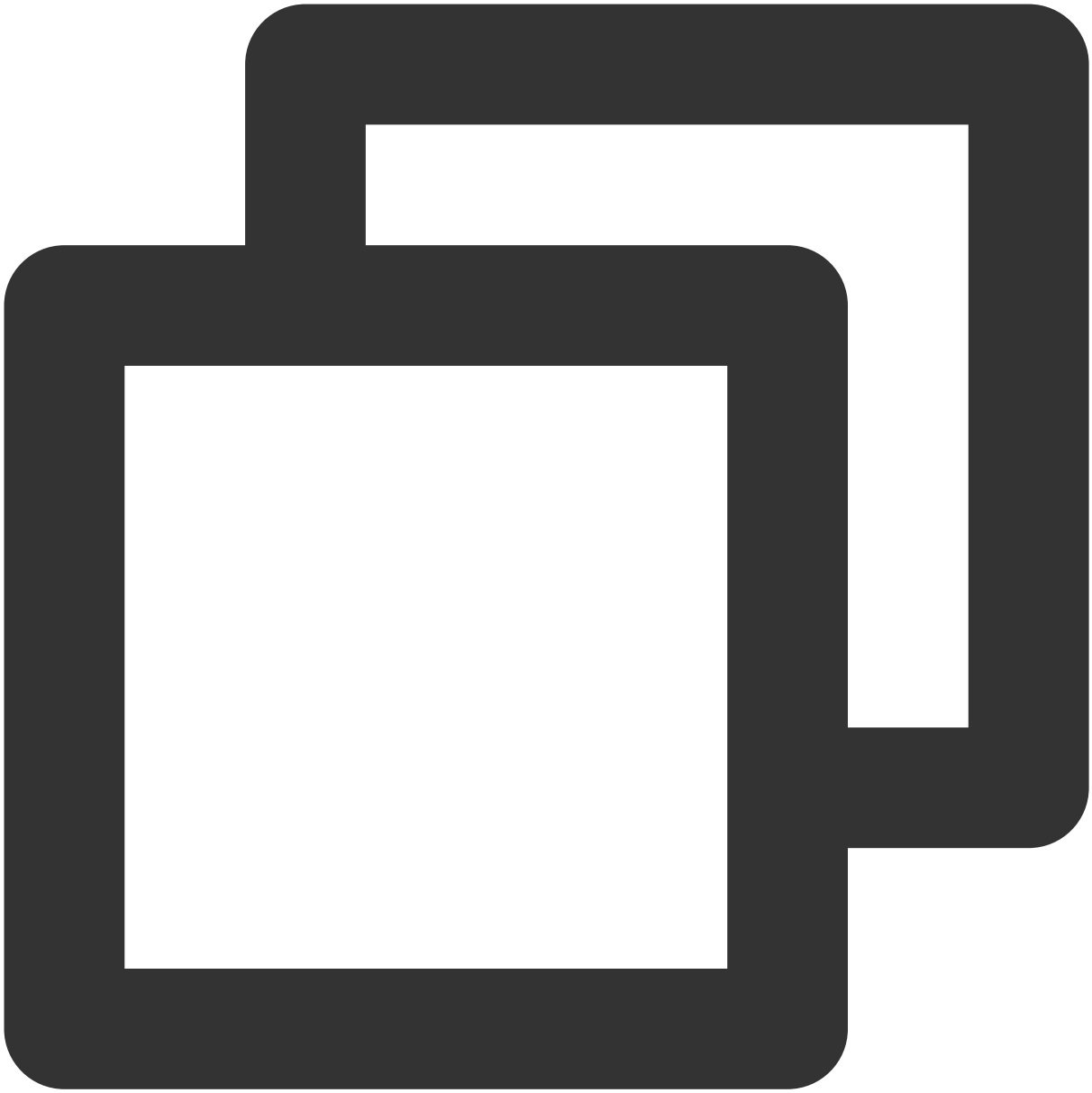
响应

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：

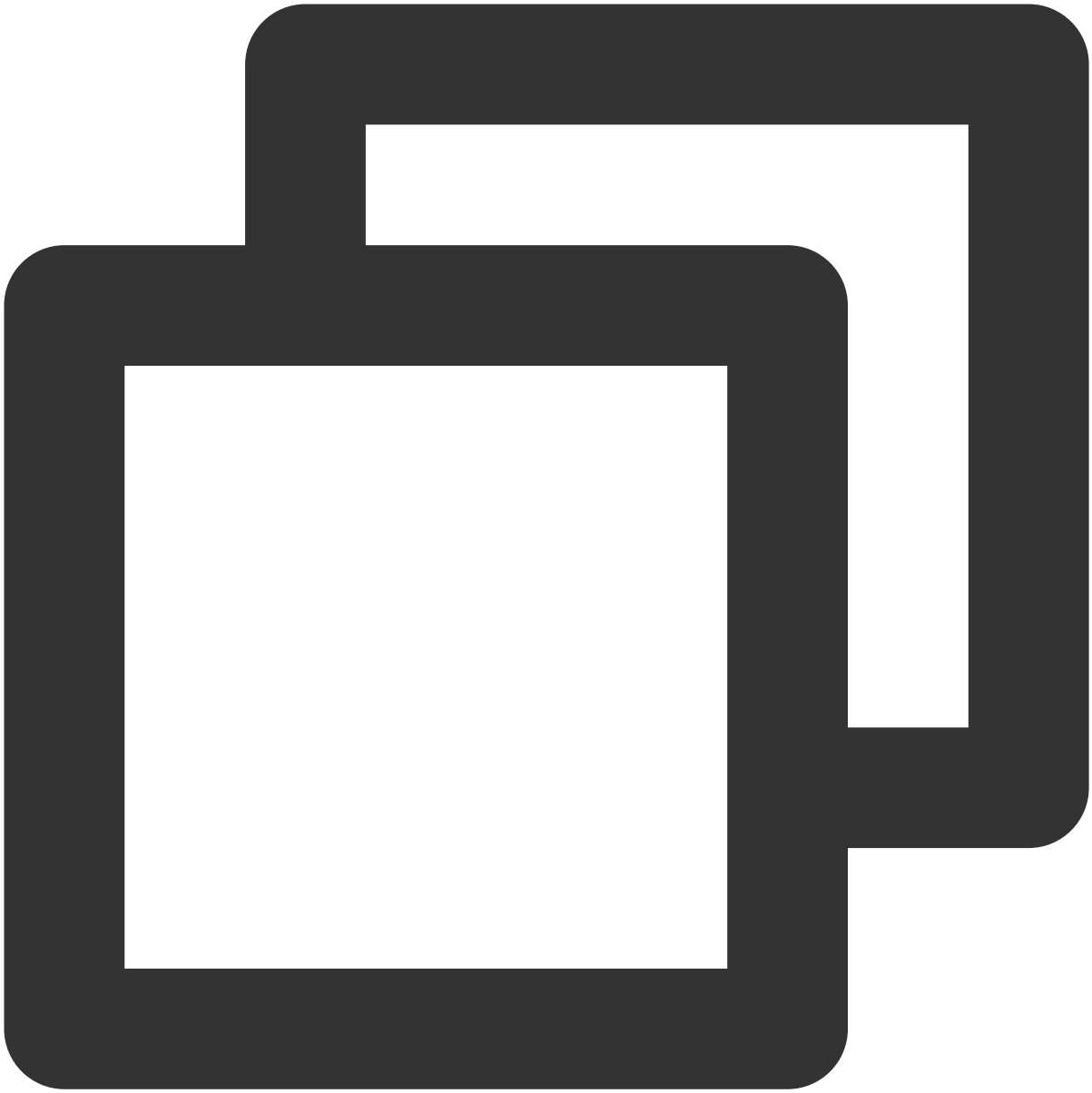
节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Operation.FileCompressResult	打包压缩后文件保存的存储桶的地域。	String
Bucket	Response.JobsDetail.Operation.FileCompressResult	打包压缩后文件保存的存储桶。	String
Object	Response.JobsDetail.Operation.FileCompressResult	打包压缩后文件的名称。	String

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

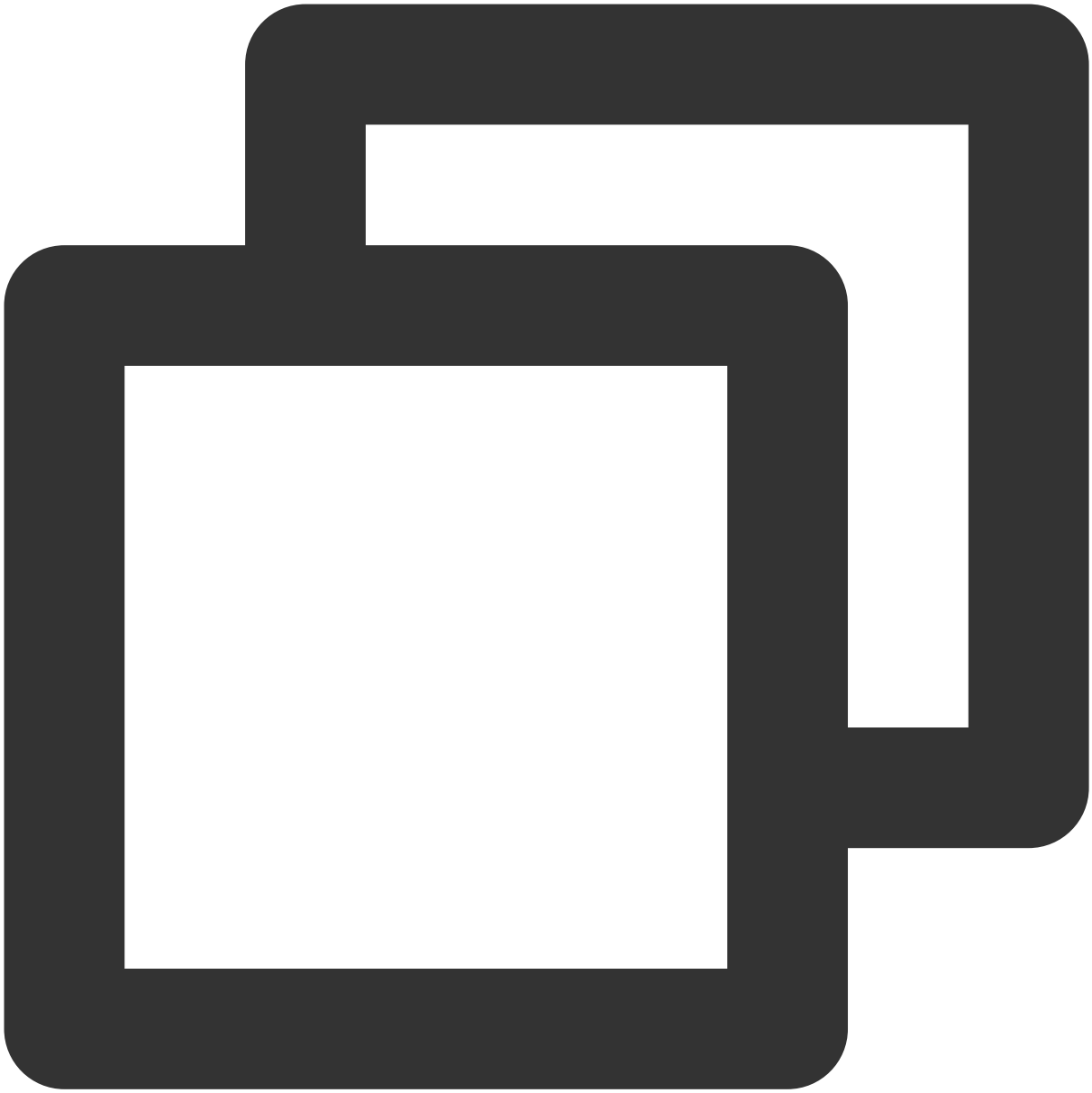


```
POST /file_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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

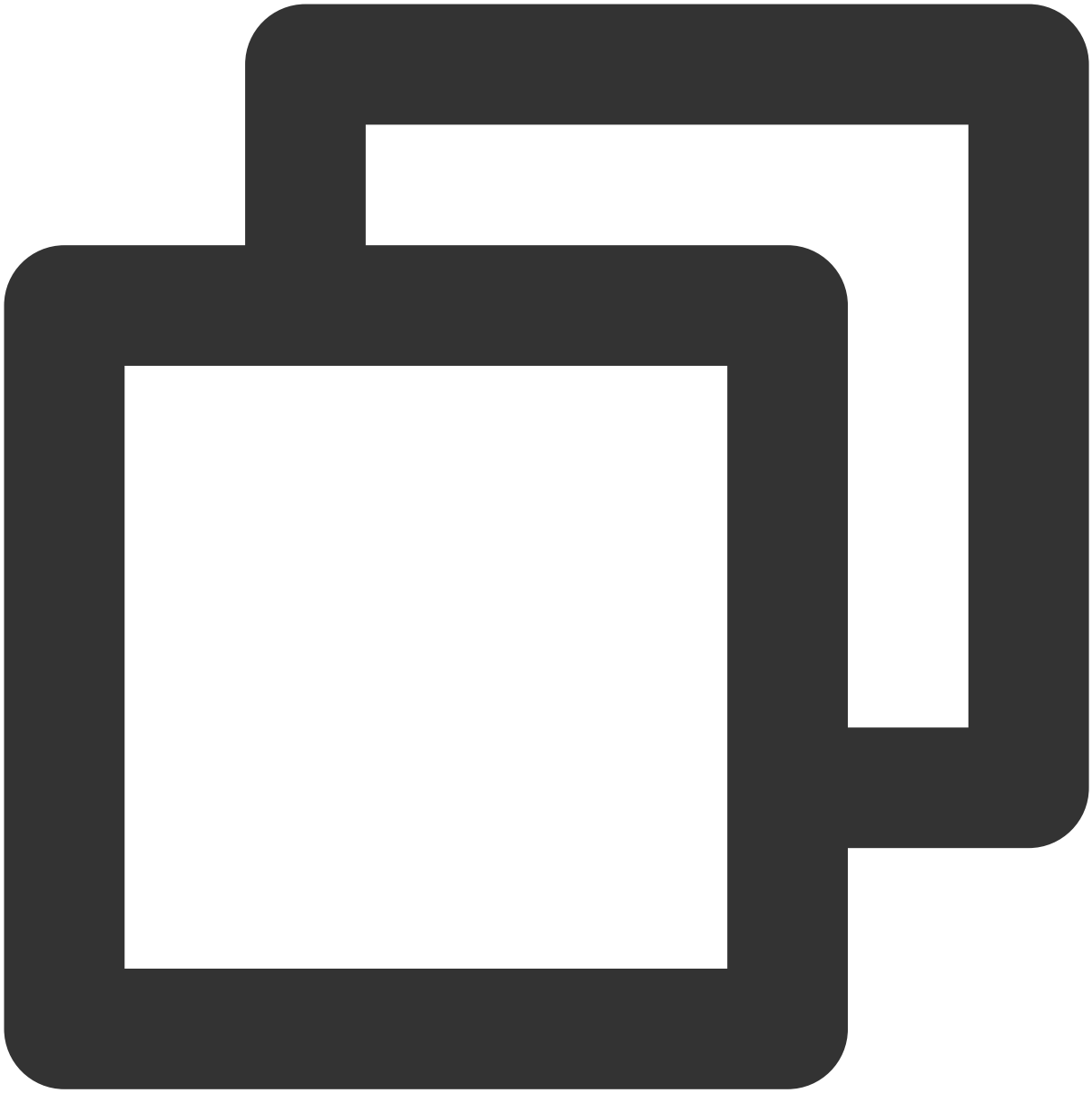
最近更新时间：2024-01-20 11:52:52

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。

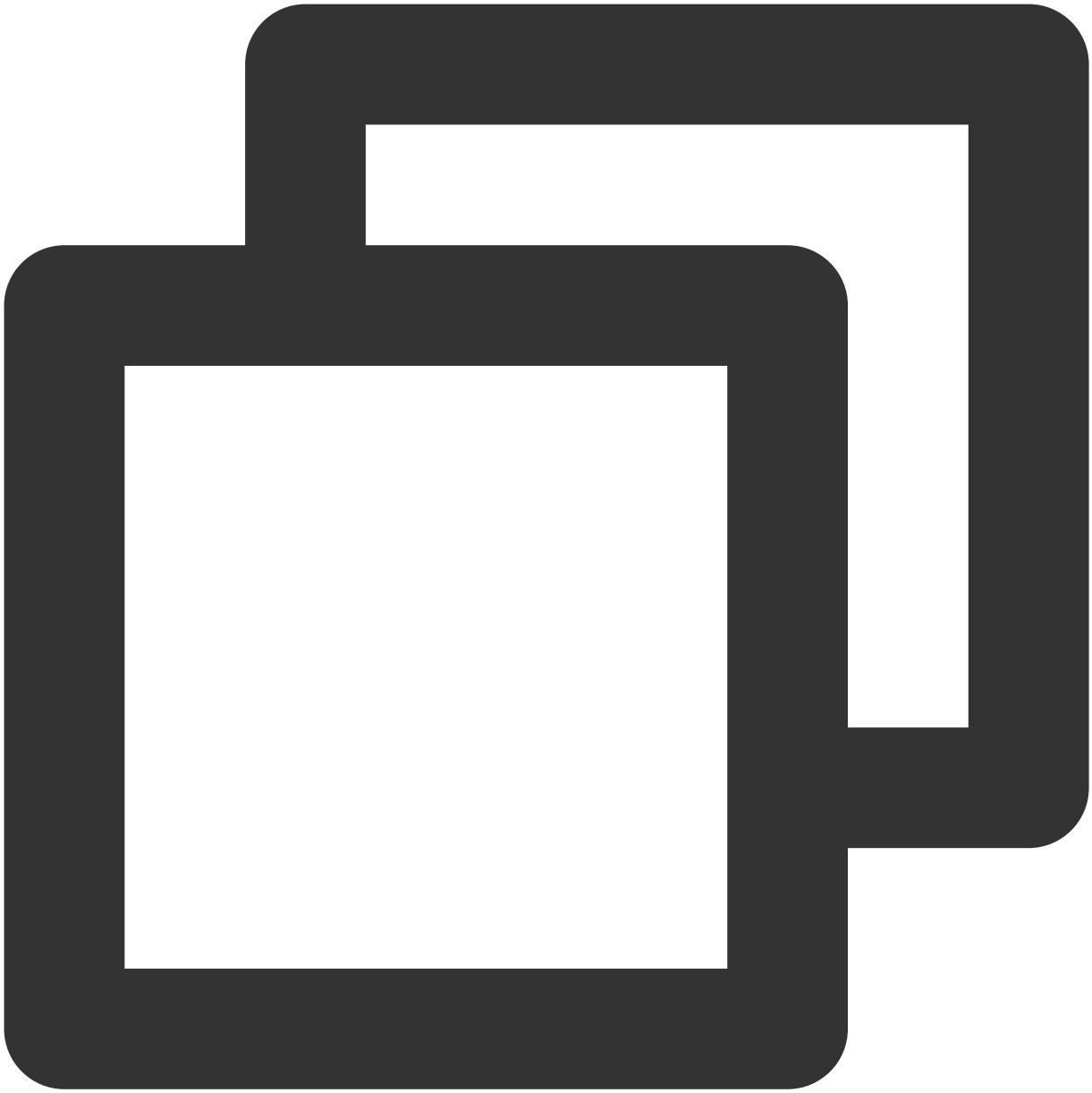
响应

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

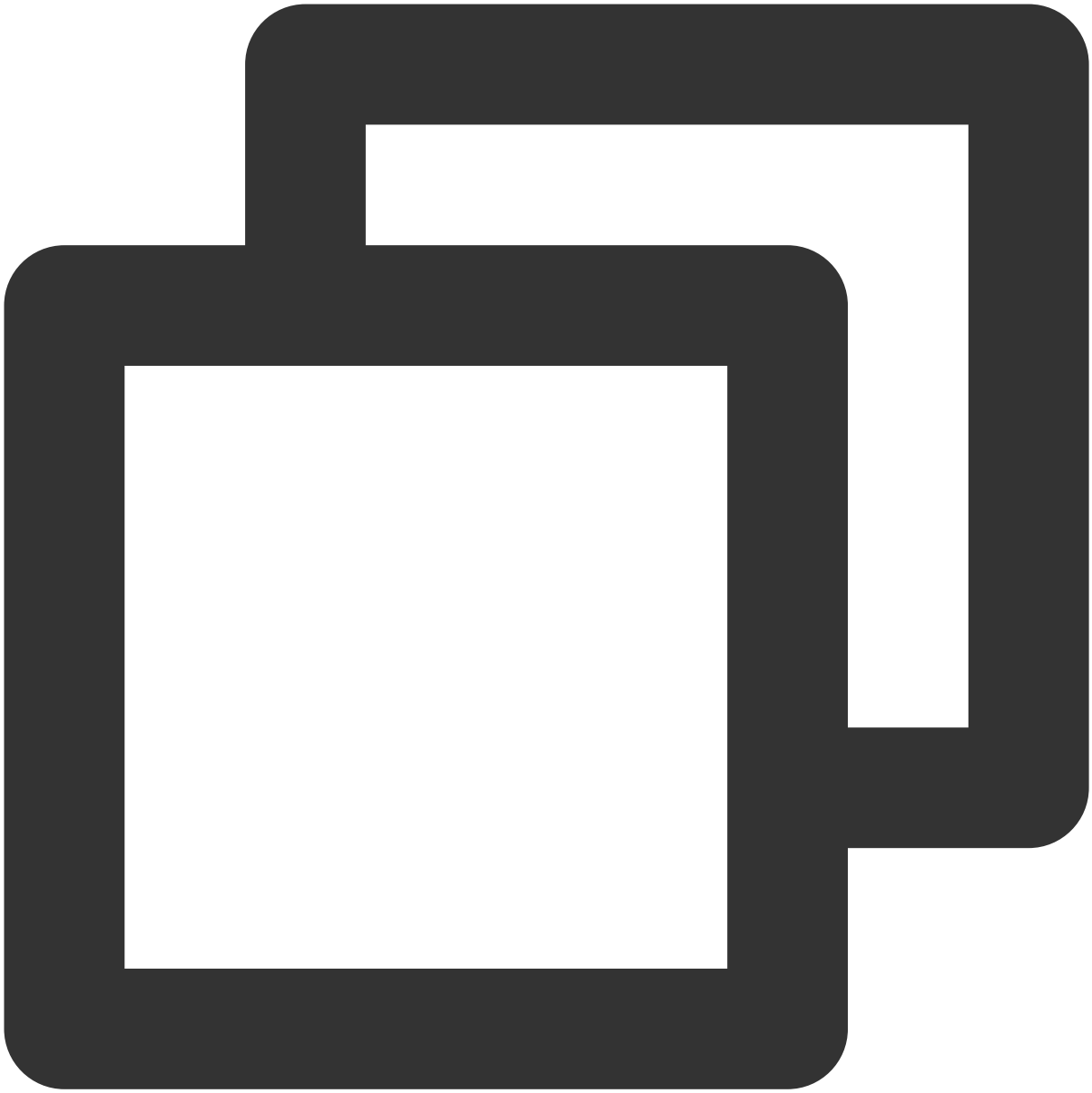
具体参数值，请前往 多文件打包压缩 `JobsDetail` 查看。

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。

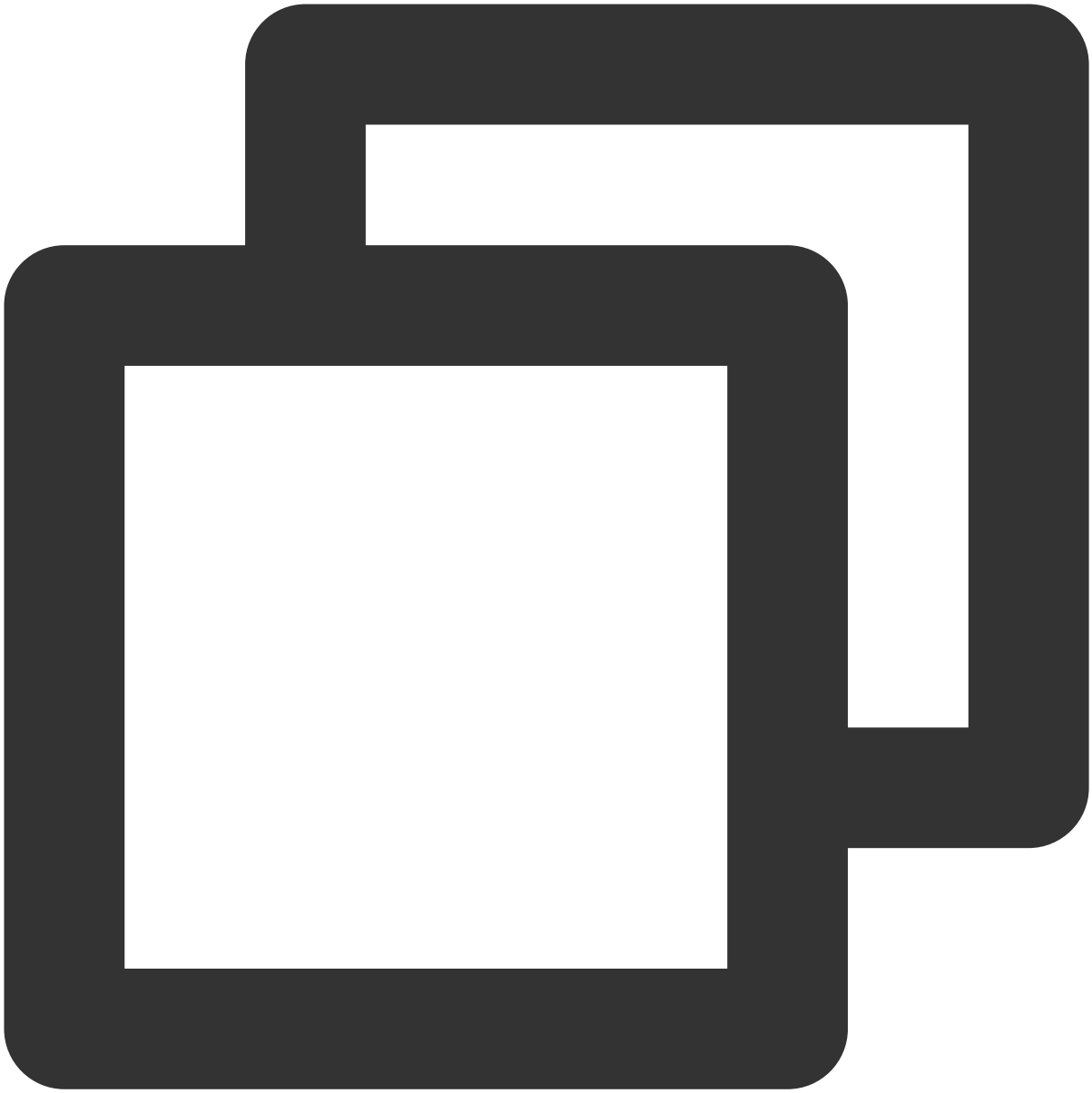
实际案例

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
工作流接口
创建工作流

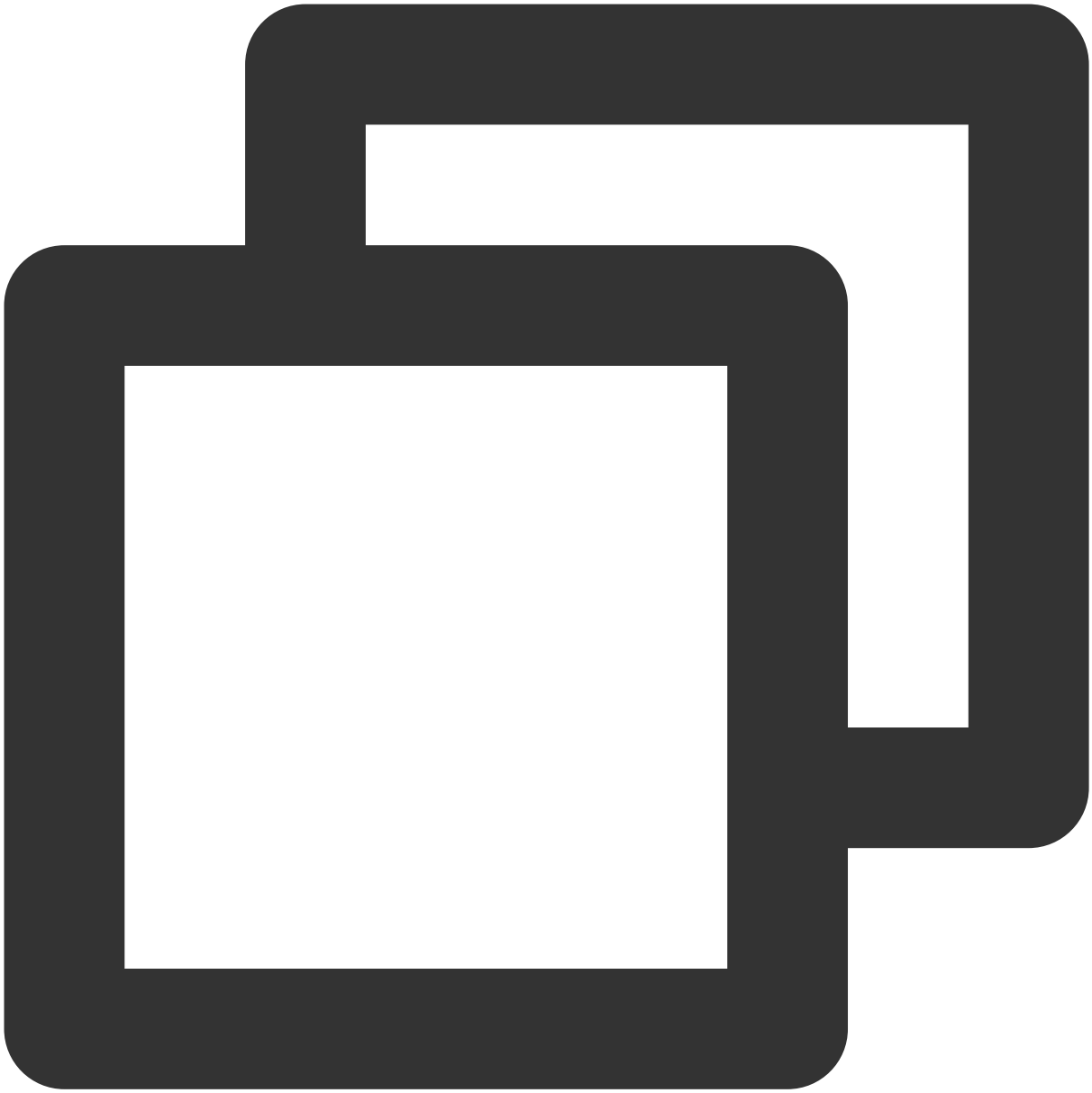
最近更新时间：2024-01-20 16:26:09

功能描述

创建工作流。

请求

请求示例



```
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：

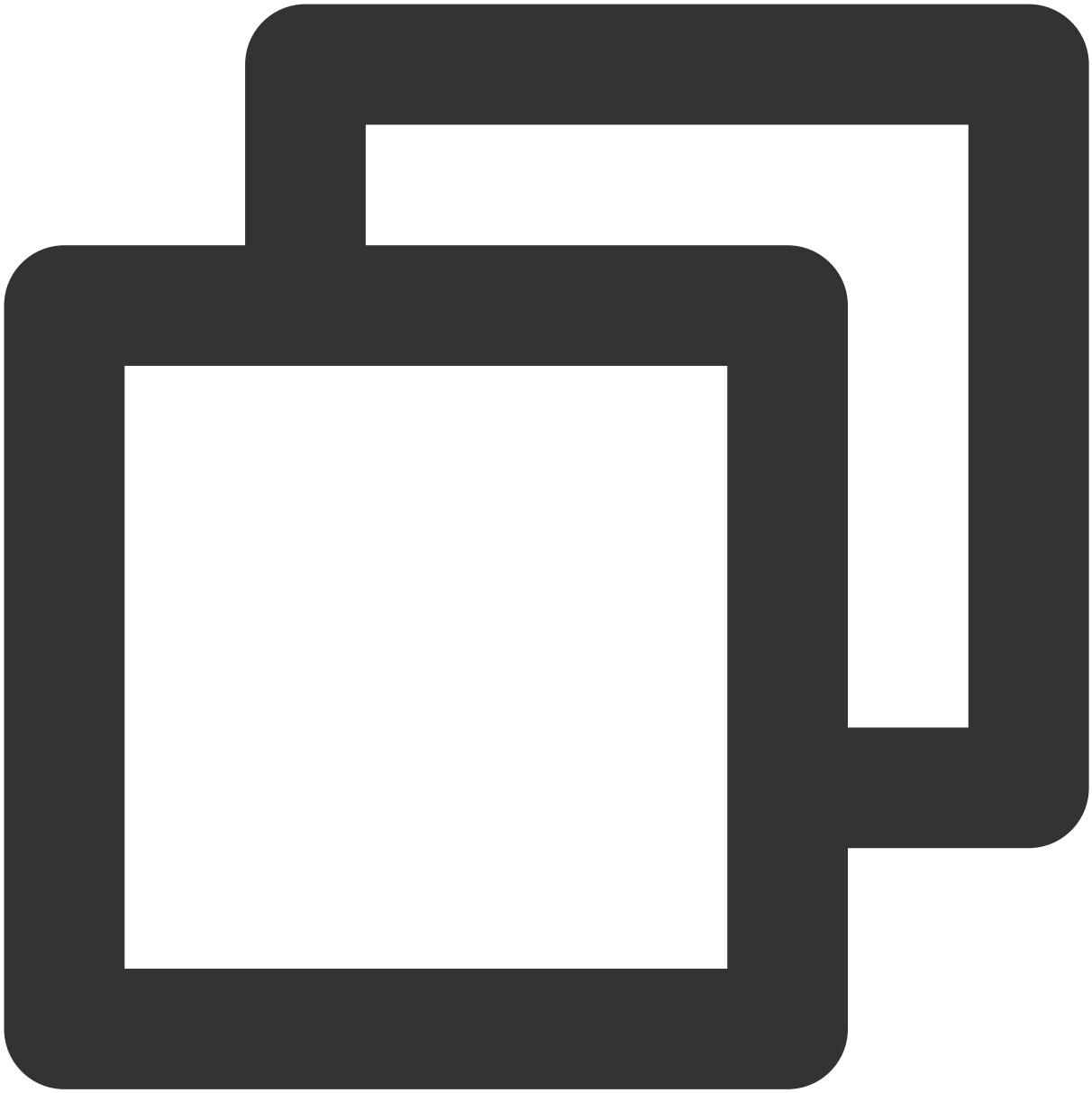
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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。

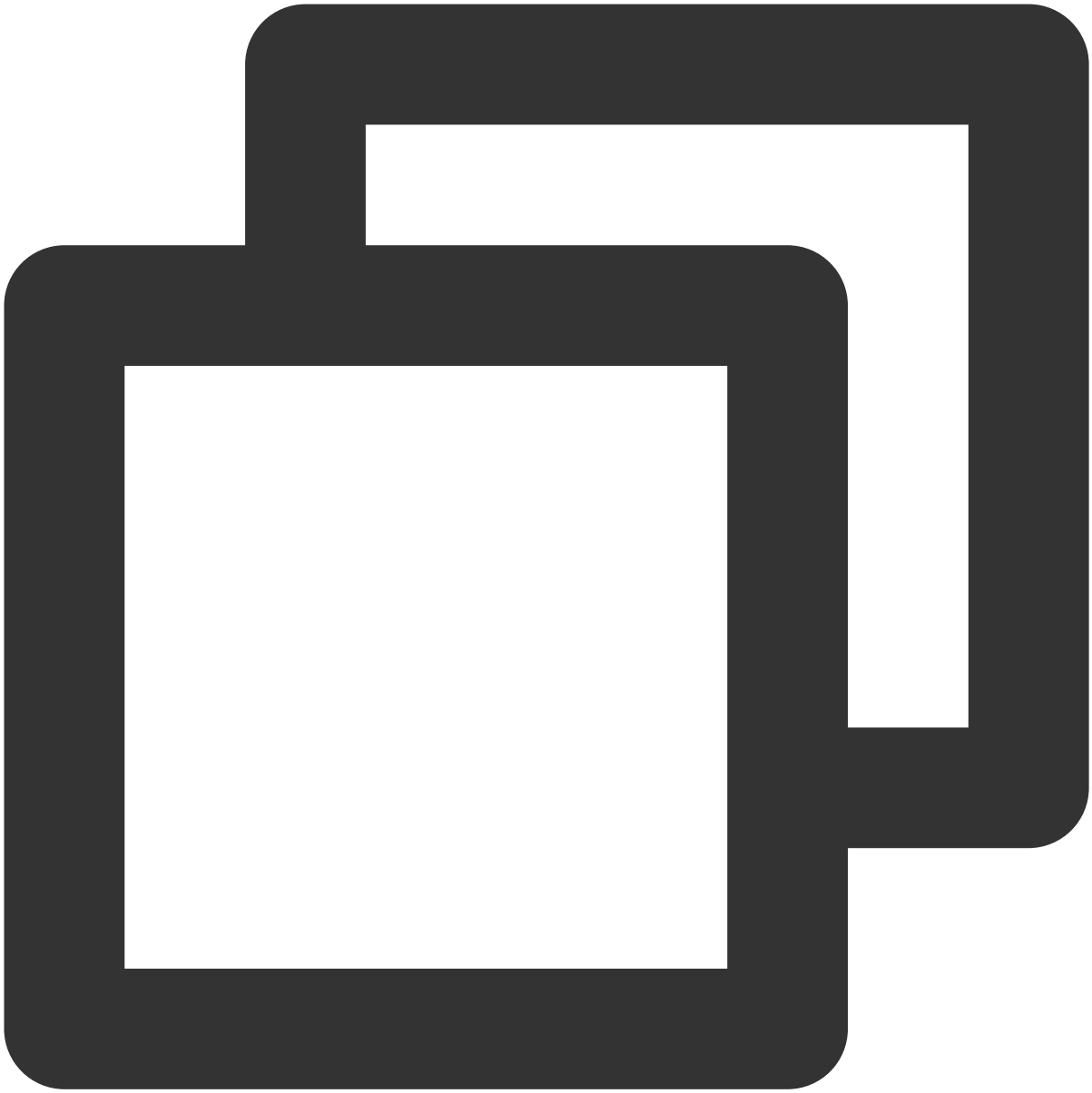


```
<Request>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <Topology>
            <Dependencies>
                <Start>Snapshot_1581665960536,Transcode_1581665960538</Start>
                <Snapshot_1581665960536>End</Snapshot_1581665960536>
                <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816659
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_158
```

```
            <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
Animation_***	Request.MediaWorkflow.Topology.Nodes	动图类型节点	Container	否	节 A 点
Snapshot_***	Request.MediaWorkflow.Topology.Nodes	截图类型节点	Container	否	节 S 点
SmartCover_***	Request.MediaWorkflow.Topology.Nodes	智能封面节点	Container	否	节 S 参

Transcode_***	Request.MediaWorkflow.Topology.Nodes	转码节点	Container	否	节 T 编 码
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 音 人
VideoMontage_***	Request.MediaWorkflow.Topology.Nodes	精彩集锦节点	Container	否	节 V 编 采
StreamPackConfig_***	Request.MediaWorkflow.Topology.Nodes	自适应码流节点	Container	否	节 S 入 点 点 类 自 流
VideoStream_***	Request.MediaWorkflow.Topology.Nodes	视频子流节点	Container	否	节 V 编 步 在 S 节 自 节
StreamPack_***	Request.MediaWorkflow.Topology.Nodes	自适应码流打包节点	Container	否	节 S 编 适 ， 节 自

SDRtoHDR_***	Request.MediaWorkflow.Topology.Nodes	SDRtoHDR 节点	Container	否	节 S 结 S
VideoProcess_***	Request.MediaWorkflow.Topology.Nodes	视频处理节点	Container	否	节 V 结 步
SCF_***	Request.MediaWorkflow.Topology.Nodes	SCF 函数节点	Container	否	节 前 S
SuperResolution_***	Request.MediaWorkflow.Topology.Nodes	超分辨率节点	Container	否	节 S 前 走
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 结 片
Tts_***	Request.MediaWorkflow.Topology.Nodes	语音合成节点	Container	否	节 前 语
SpeechRecognition_***	Request.MediaWorkflow.Topology.Nodes	语音识别节点	Container	否	节 S 为 /
VideoEnhance_***	Request.MediaWorkflow.Topology.Nodes	画质增强节点	Container	否	节 V 结 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
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

版权所有：腾讯云计算（北京）有限责任公司

Type	Request.MediaWorkflow.Topology.Nodes.Snapshot_***	节点类型	String	是	Snap
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	Container	是	无

版权所有：腾讯云计算（北京）有限责任公司

		情见 Concat			

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	阿
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	Container	是

版权所有：腾讯云计算（北京）有限责任公司

		VideoProcess			

Container 类型 SCF_*** 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.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	父节点	描述	类型	是	限

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.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

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Condition_*** 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.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 VideoEnhance	Container

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

版权所有：腾讯云计算（北京）有限责任公司

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。

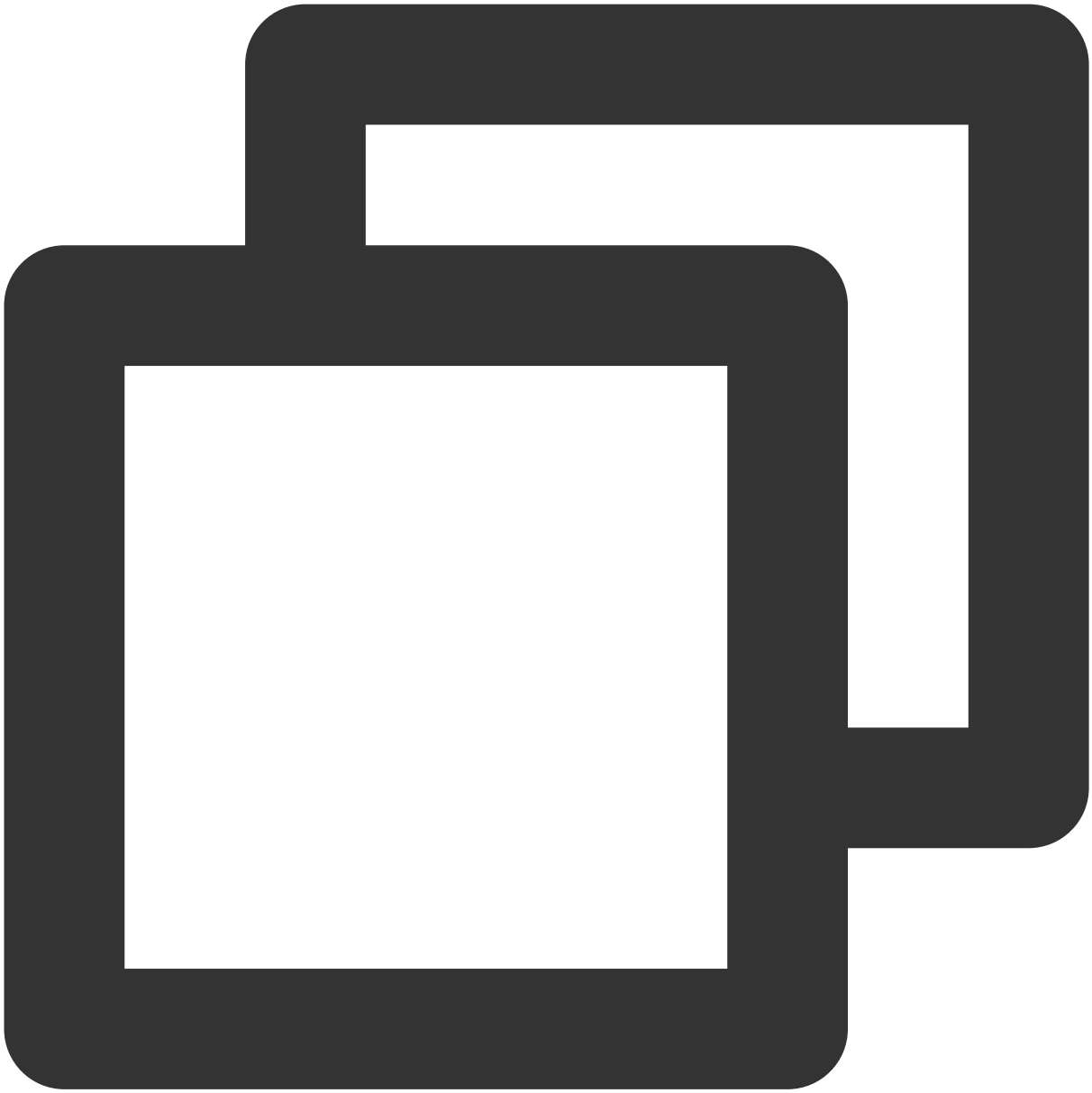
响应体

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。

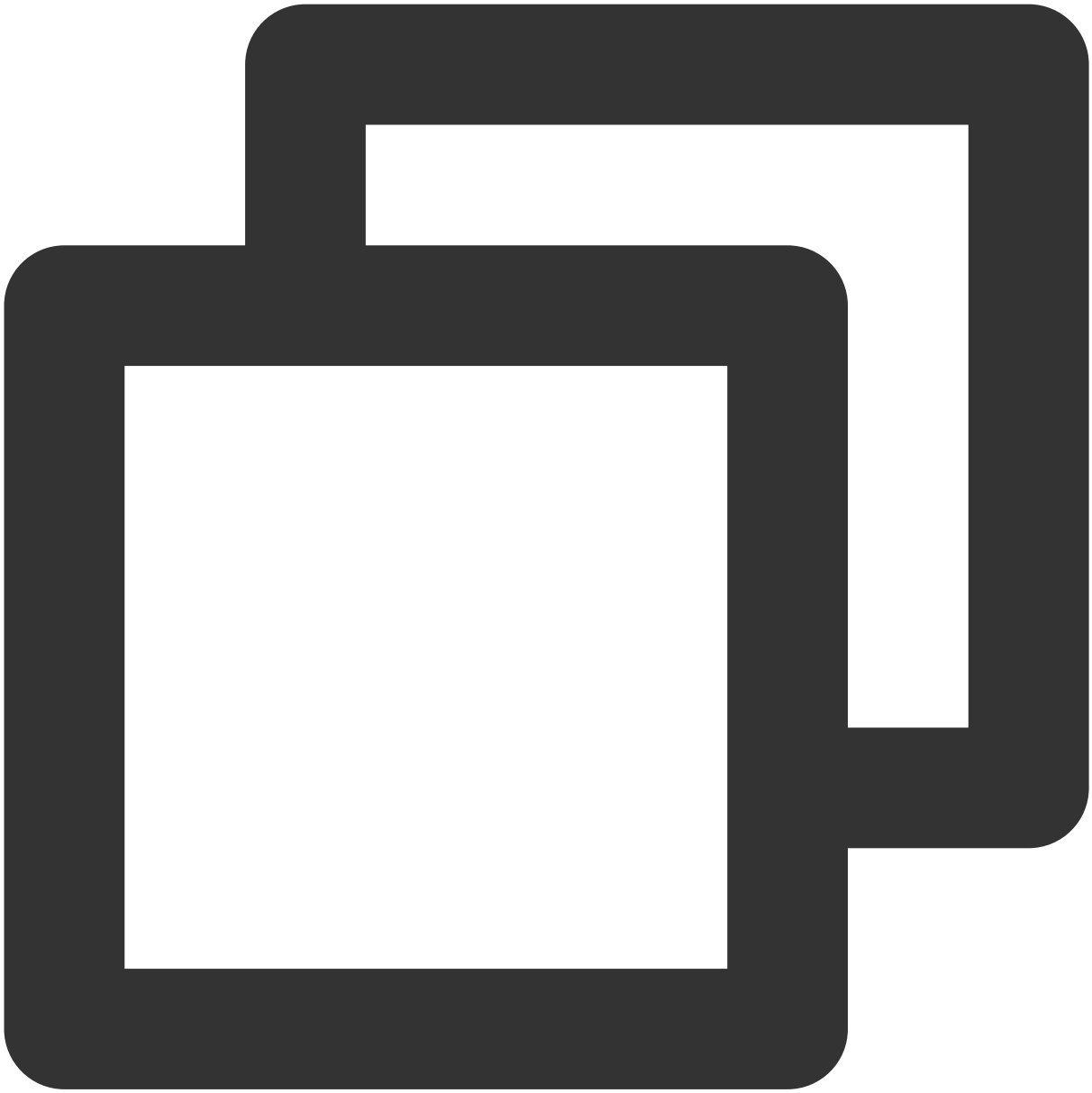


```
<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <MediaWorkflow>
        <Name>workflow-1</Name>
        <State>Active</State>
        <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
        <BucketId>test-1234567890</BucketId>
        <CreateTime>2022-07-14T12:37:28+0800</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>
```

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>
</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

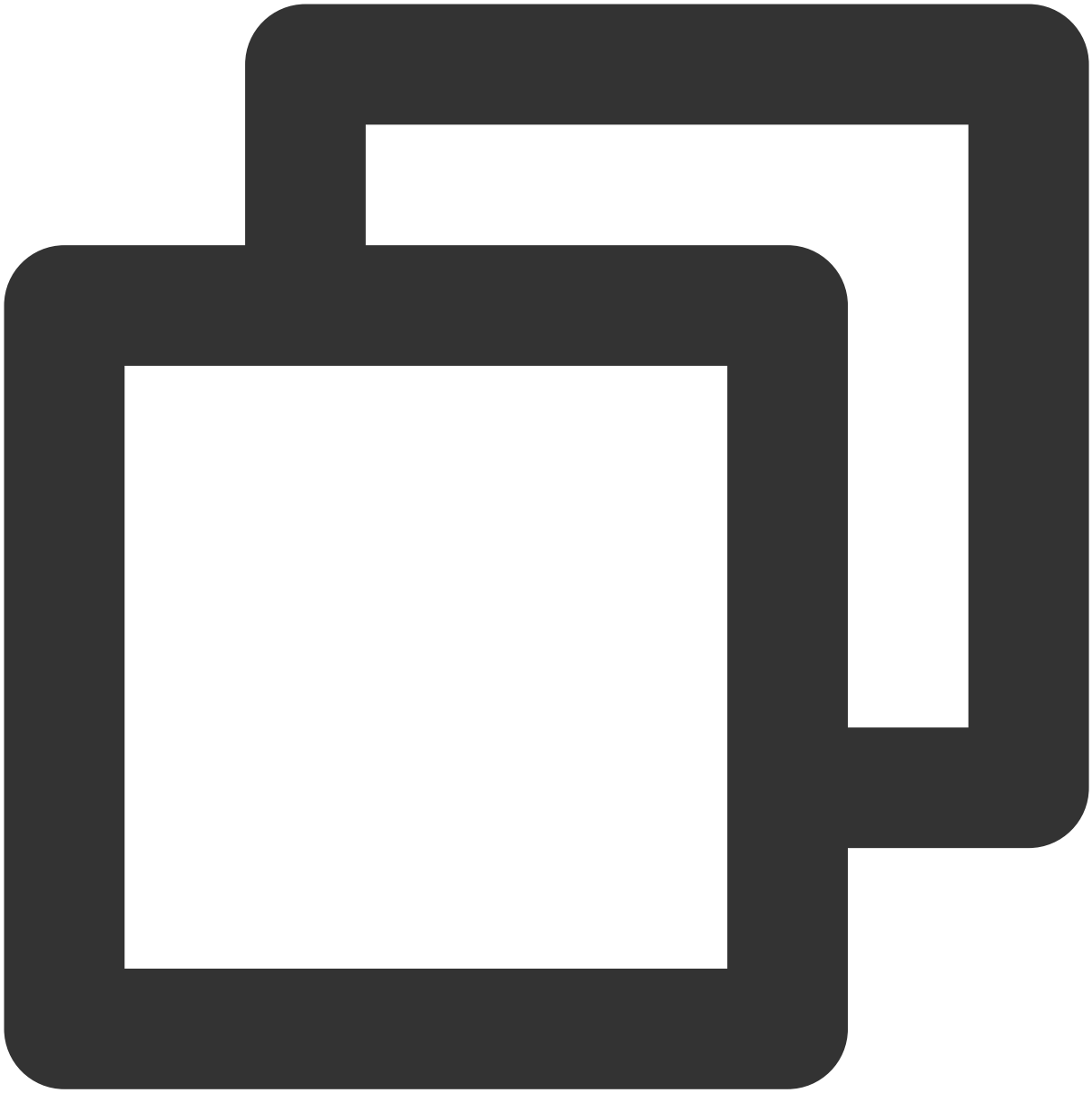
请求1

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。



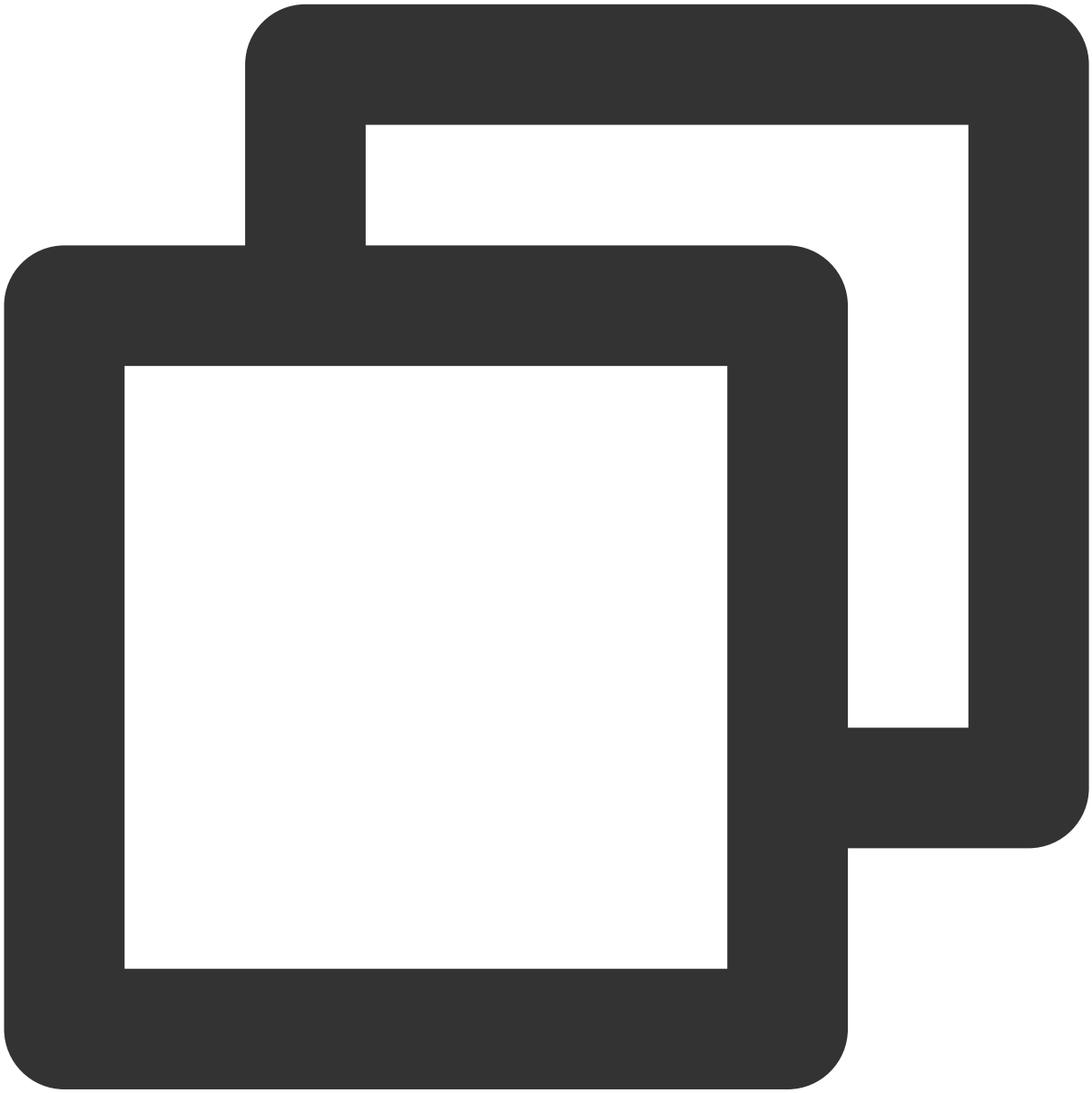
```
POST /workflow HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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

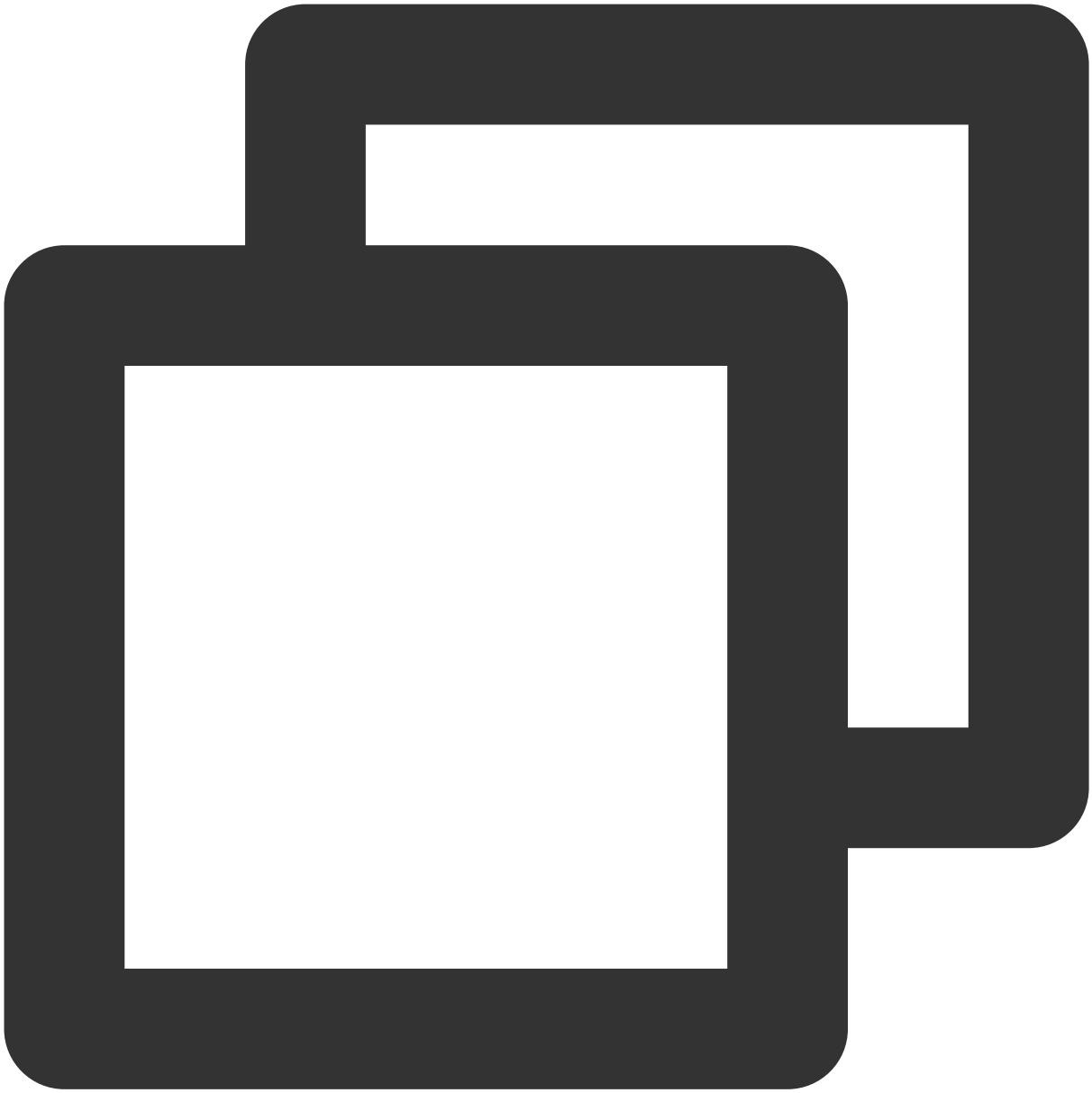
<Response>
```

```
<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

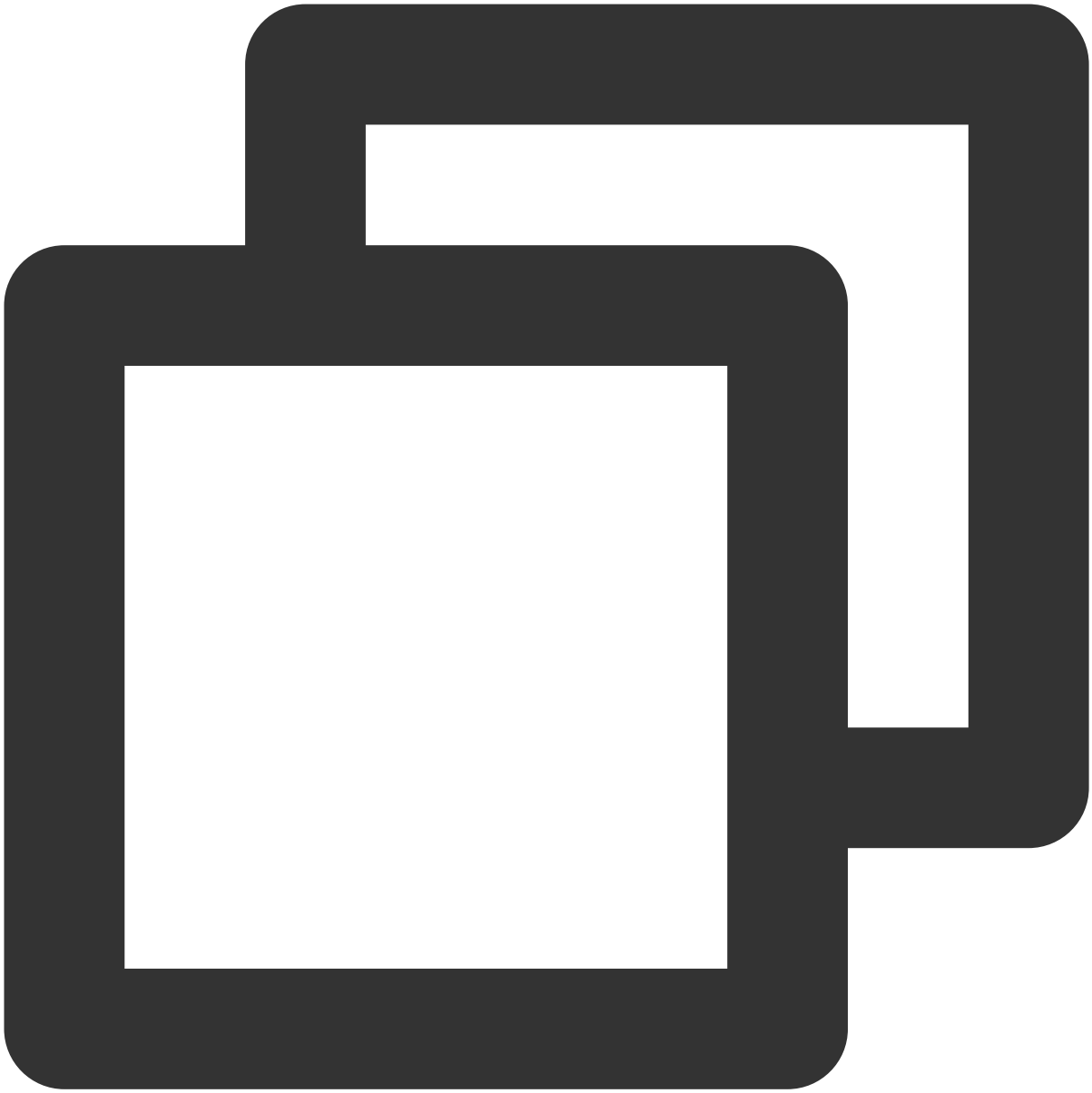


```
POST /workflow HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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>
```

```
                        <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

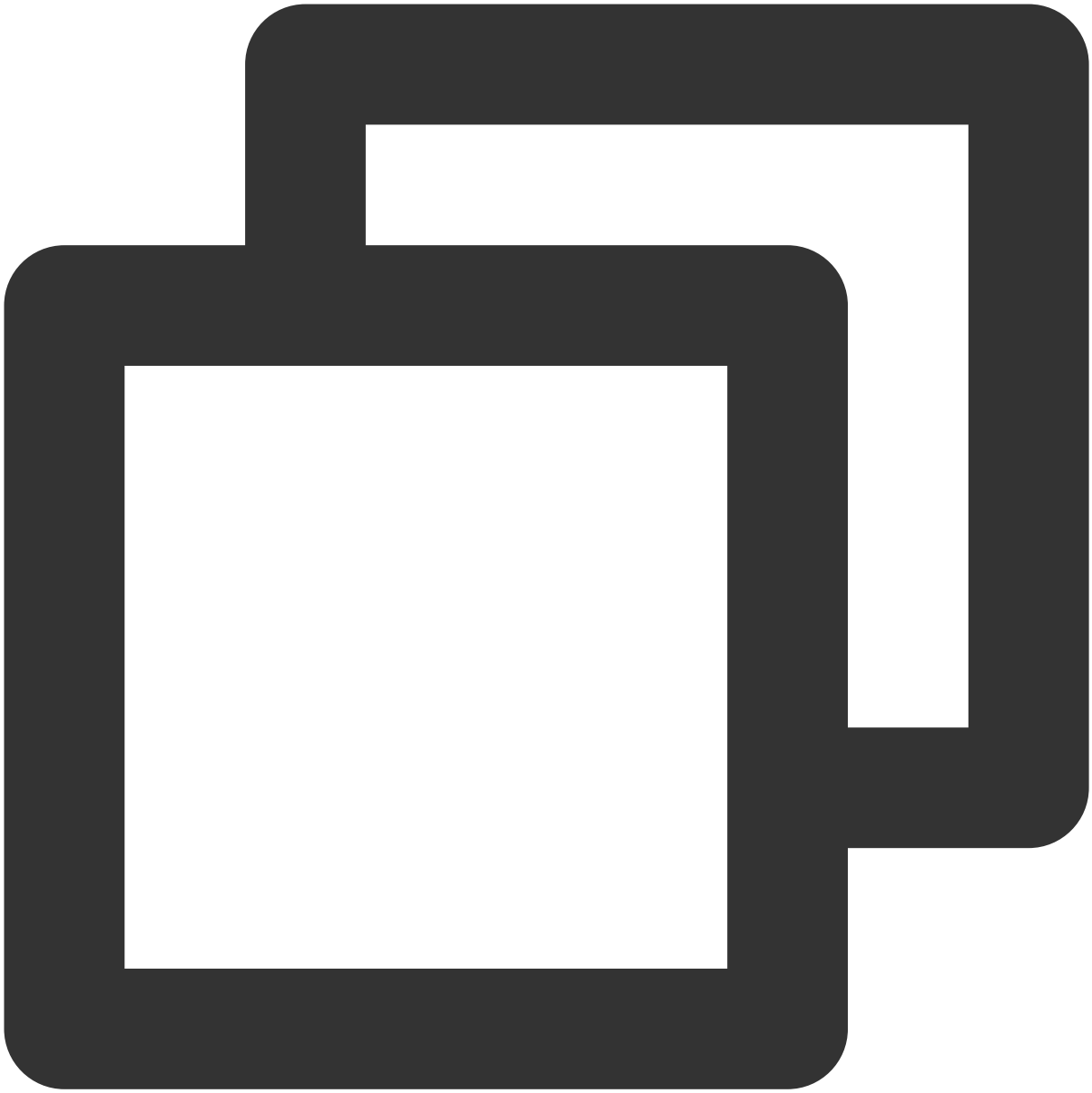
最近更新时间：2024-01-20 16:26:09

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

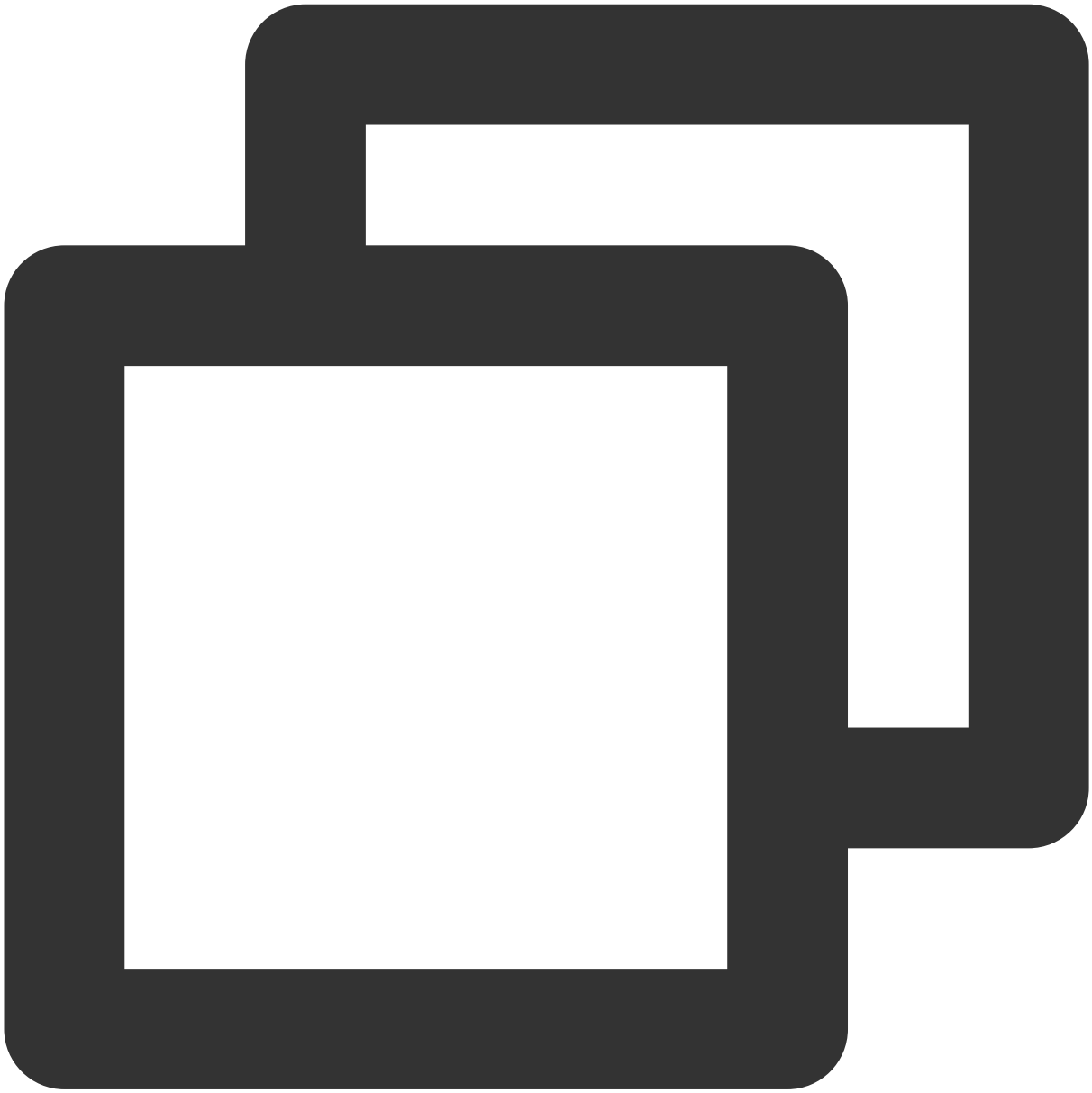
响应

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：

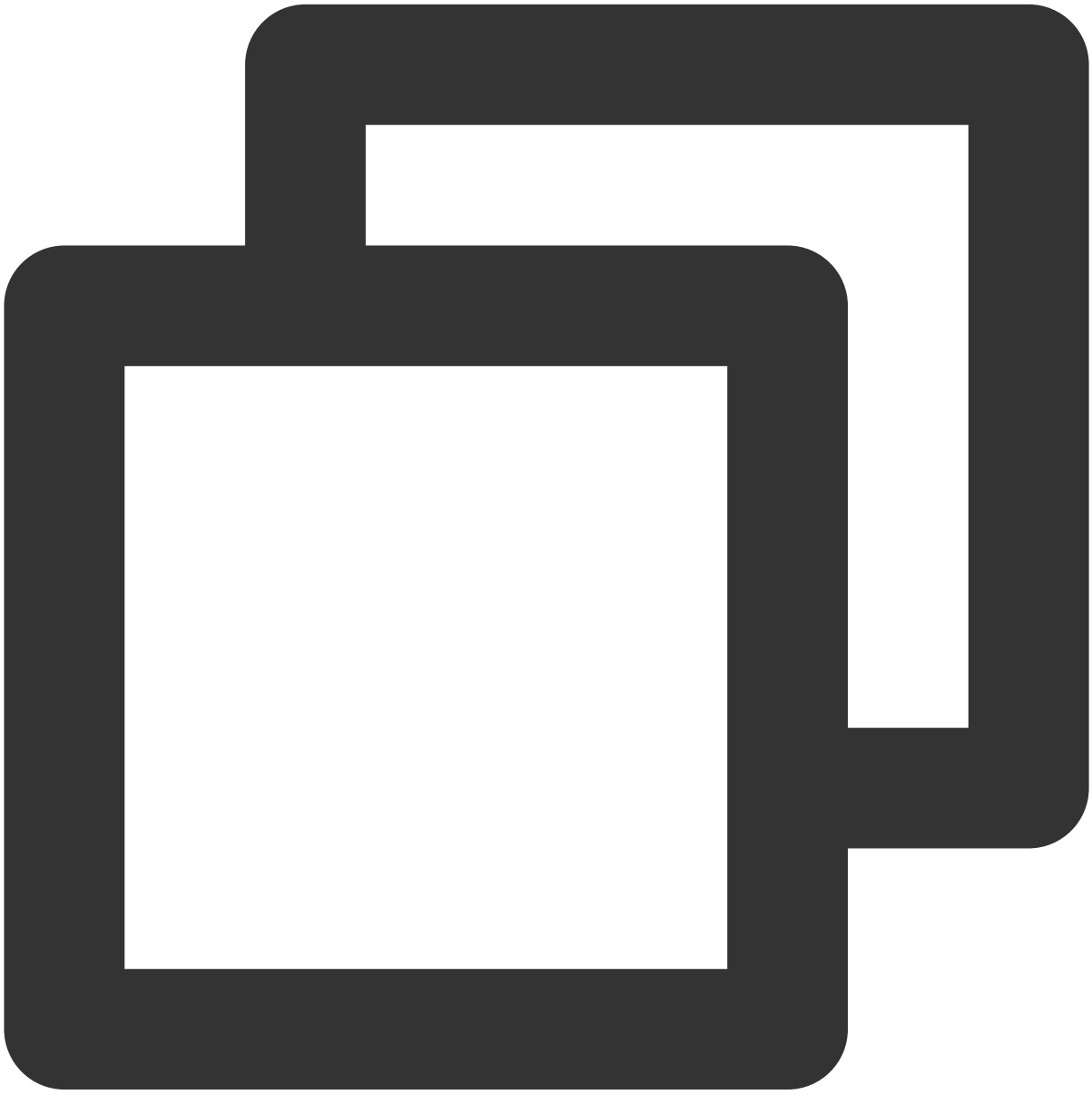
节点名称（关键字）	父节点	描述	类型	是否必选	限制
Name	Response.MediaWorkflowList	工作流名称	String	是	支持中文、英文、数字、—和_，长度限制128字符
WorkflowId	Response.MediaWorkflowList	工作流ID	String	是	工作流唯一 ID
State	Response.MediaWorkflowList	工作流状态	String	是	1. Active 2. Paused
CreateTime	Response.MediaWorkflowList	创建时间	String	是	无
UpdateTime	Response.MediaWorkflowList	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

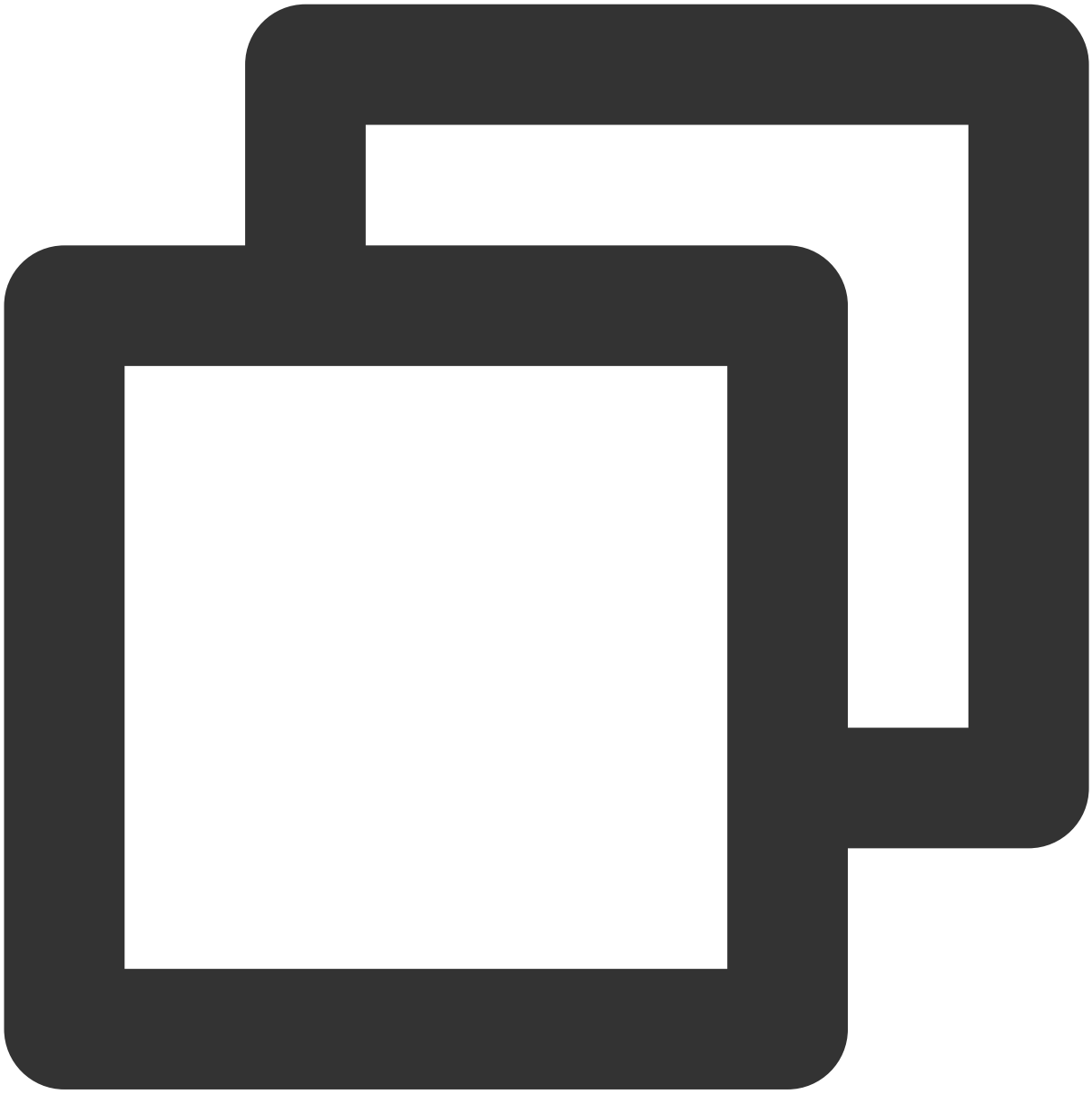
请求1：工作流 ID



```
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
```

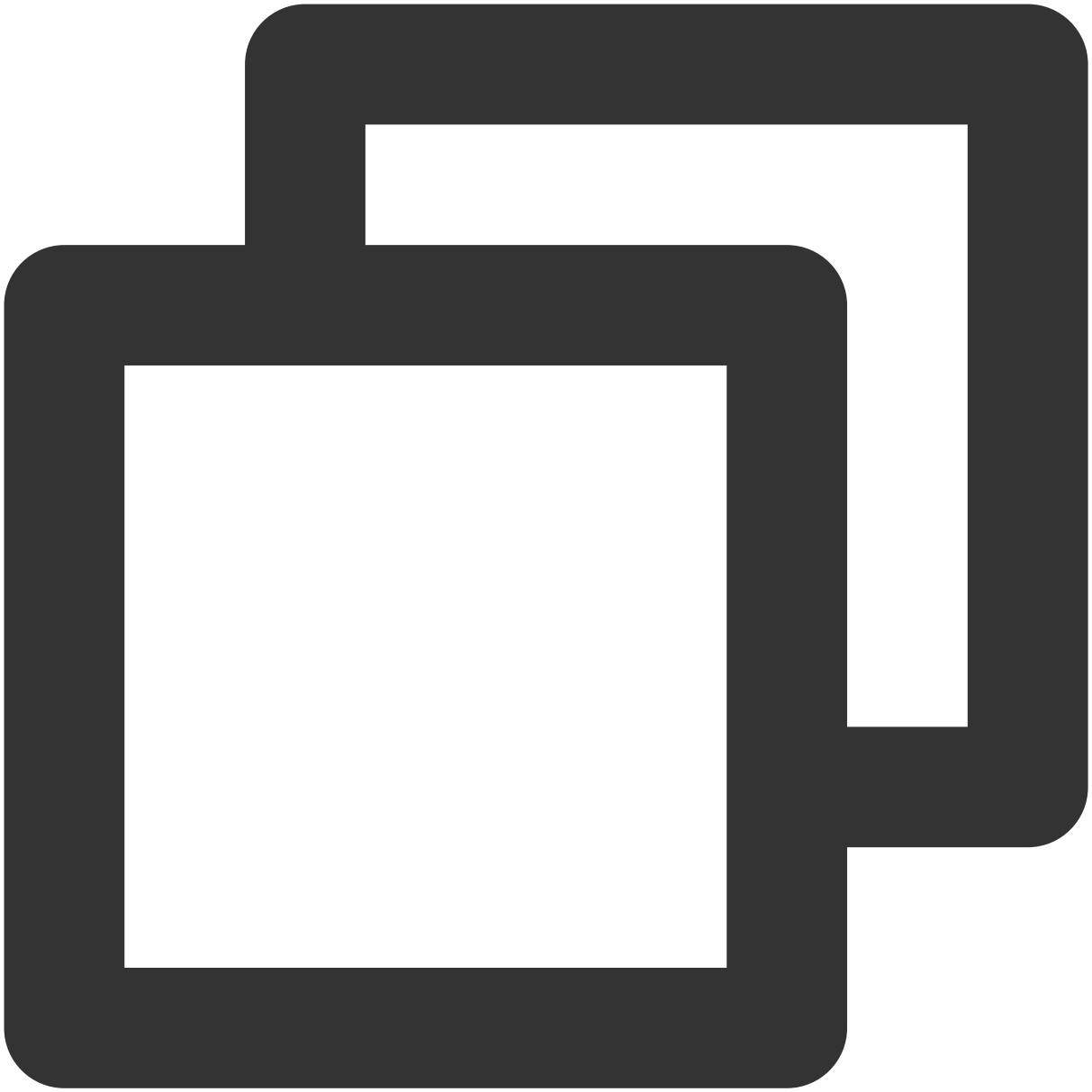
```
x-ci-request-id: NjJmMWZmMzRfOTBmYTUwNjRfNjVmOF8x

<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>
```

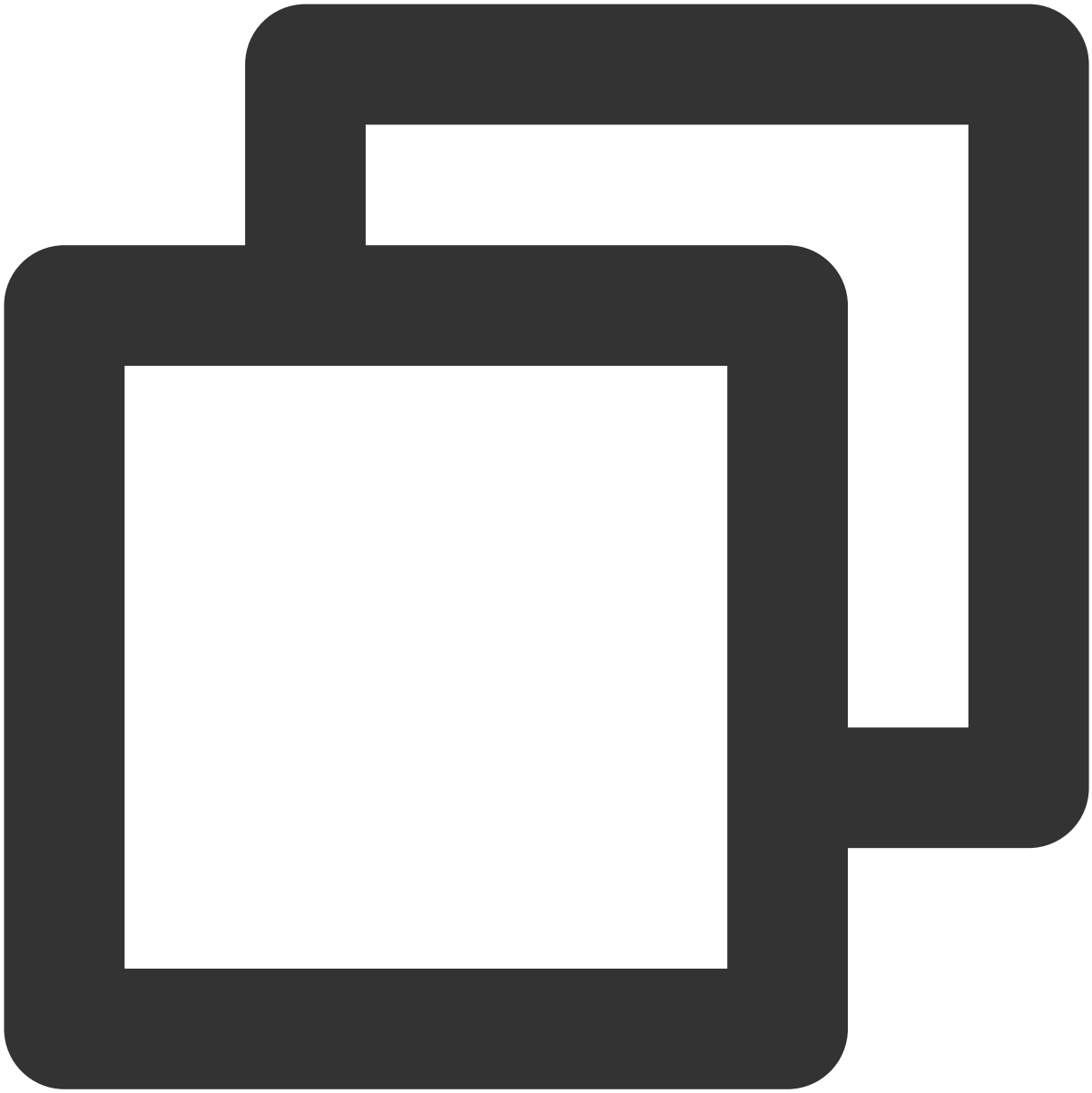
```
                    <Operation>
                        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
                        <Output>
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <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

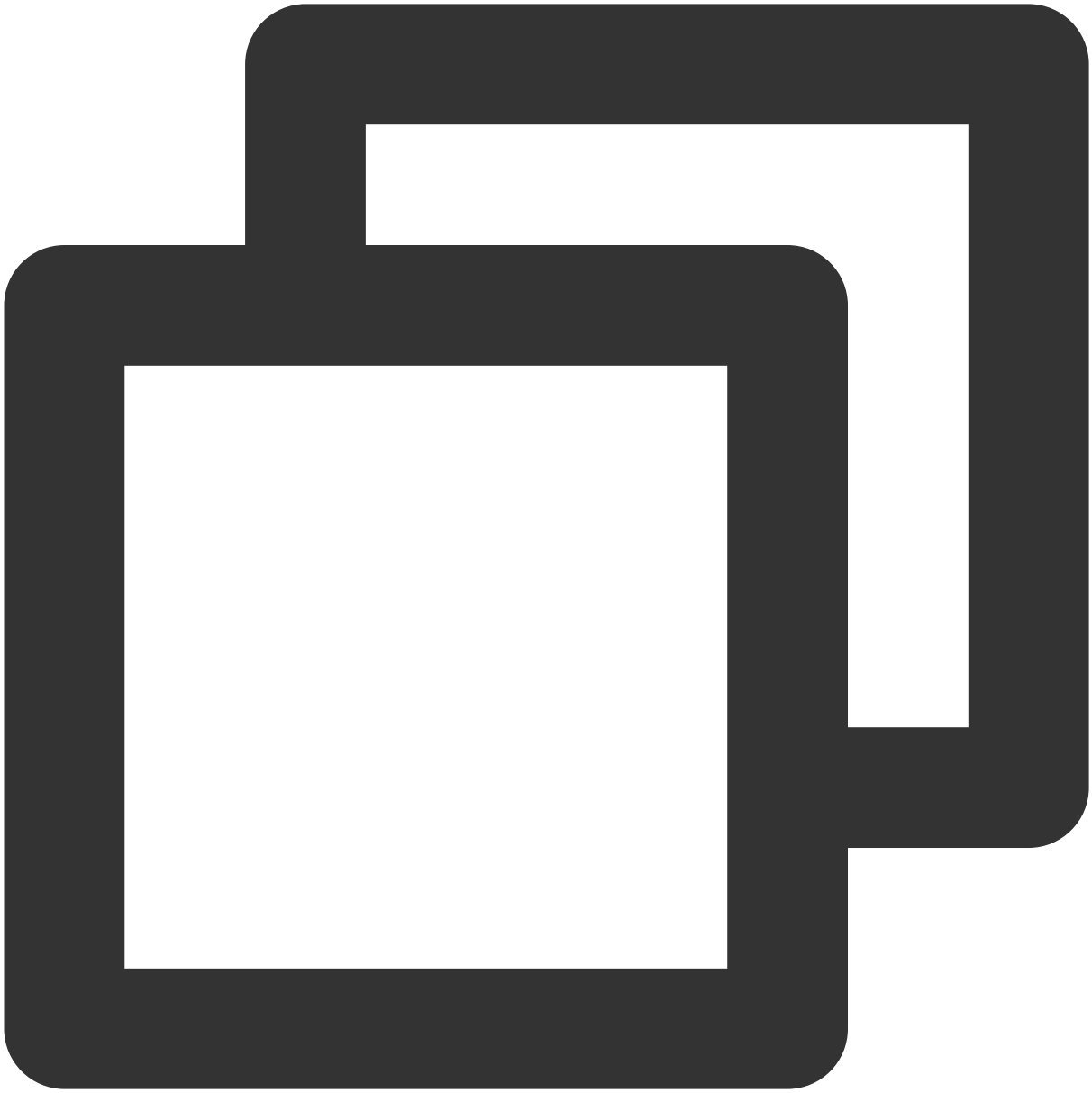
最近更新时间：2024-01-20 16:26:09

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。

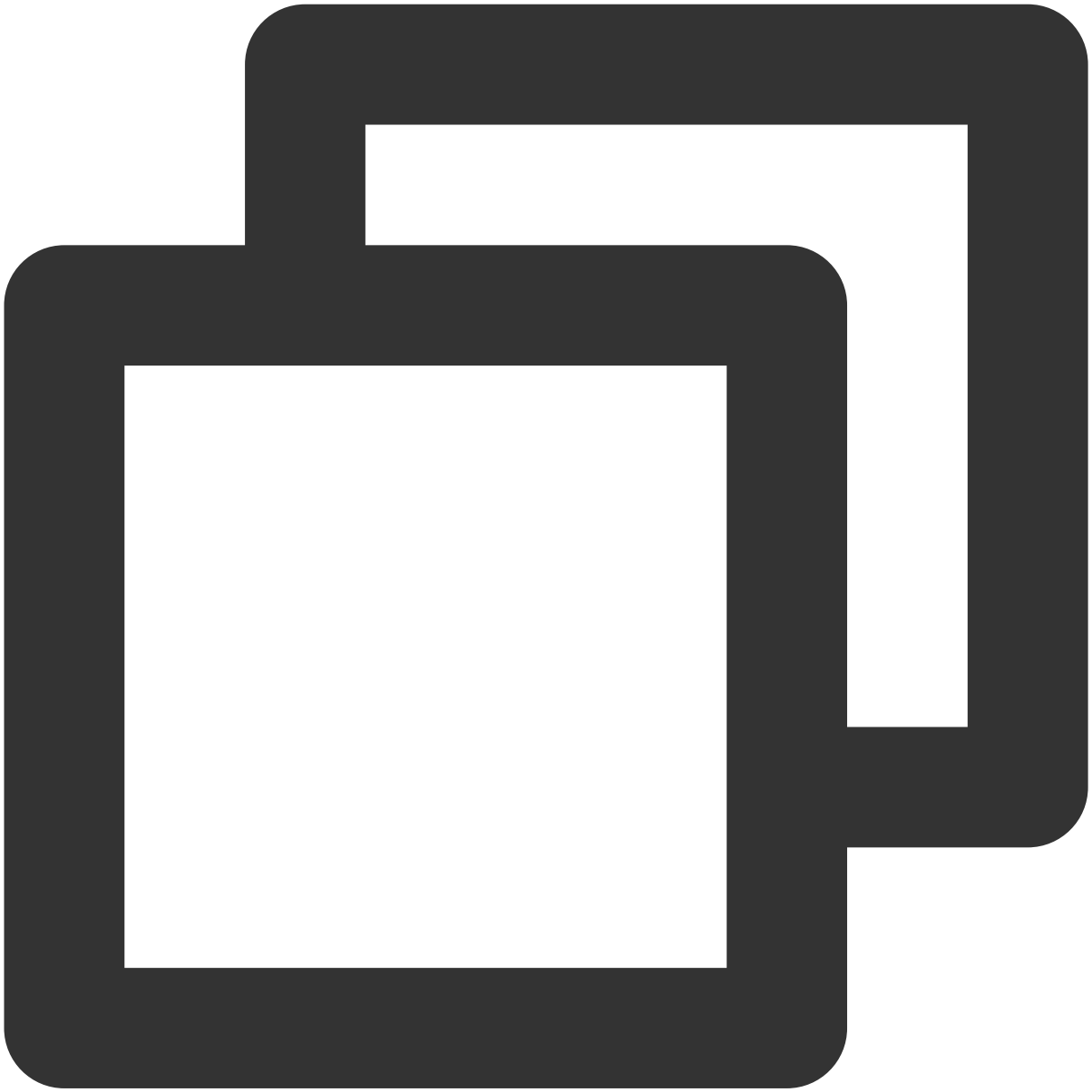
响应

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：

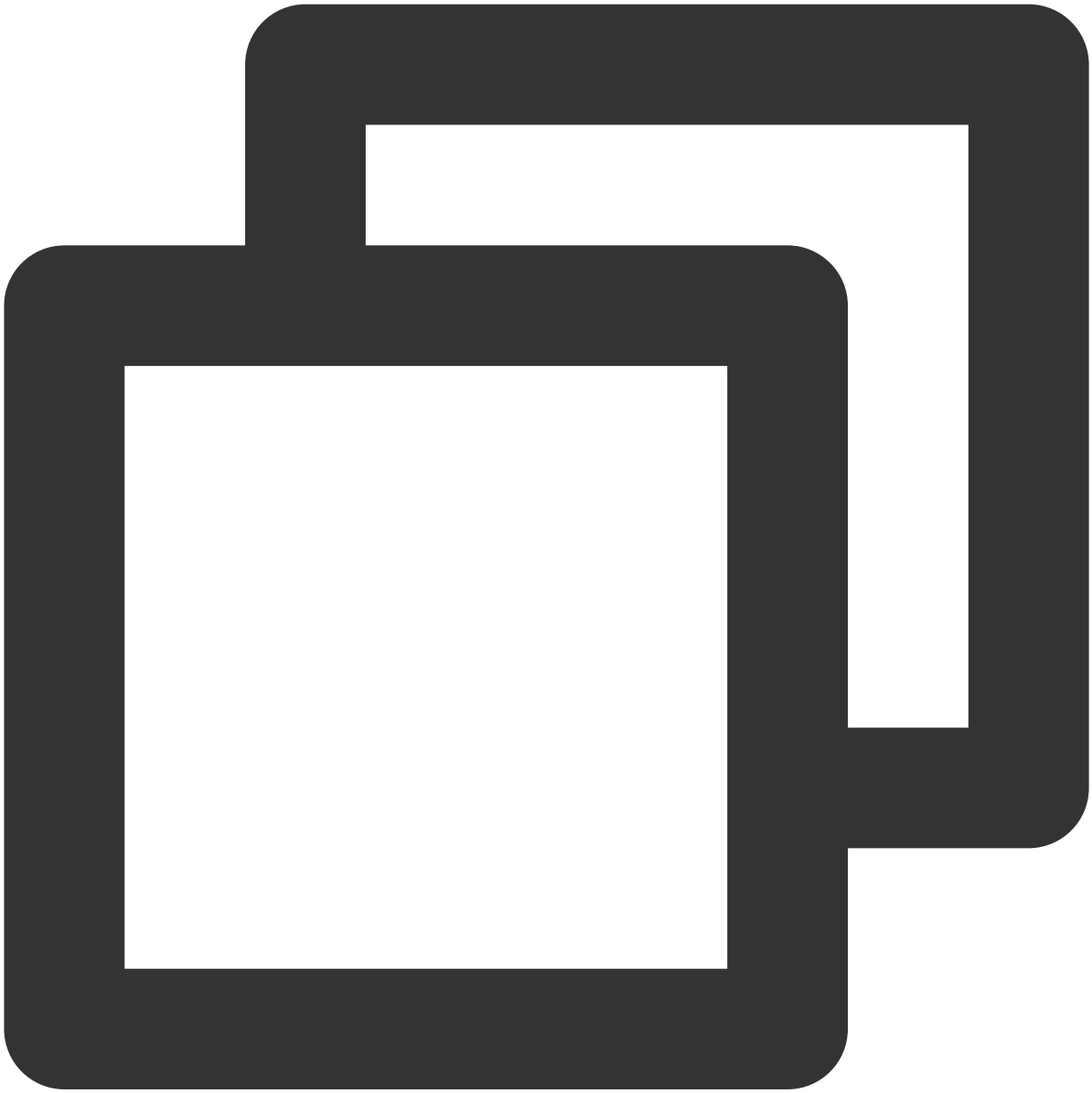
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
WorkflowId	Response	被删除的工作流 ID	String

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。

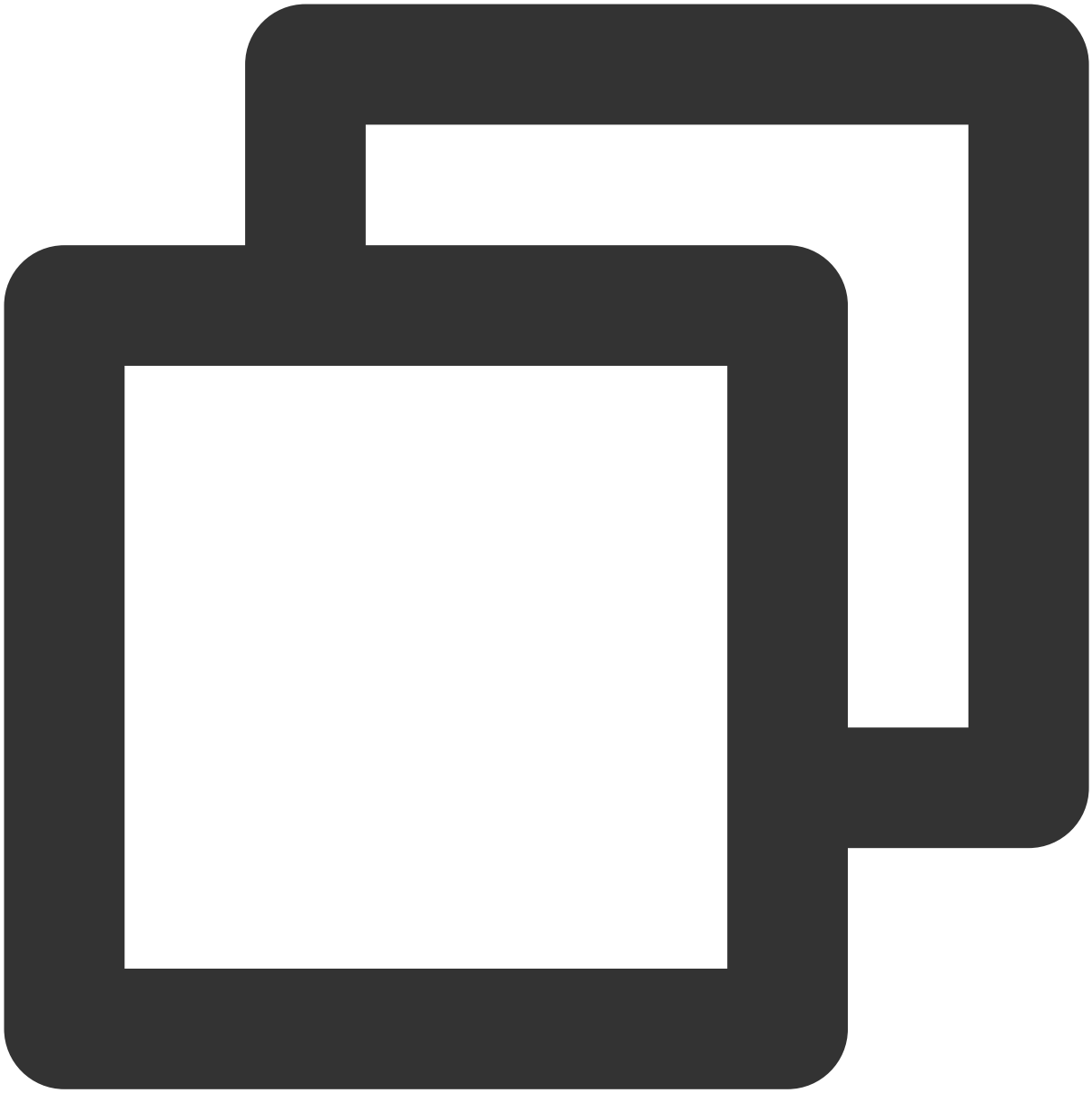
实际案例

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

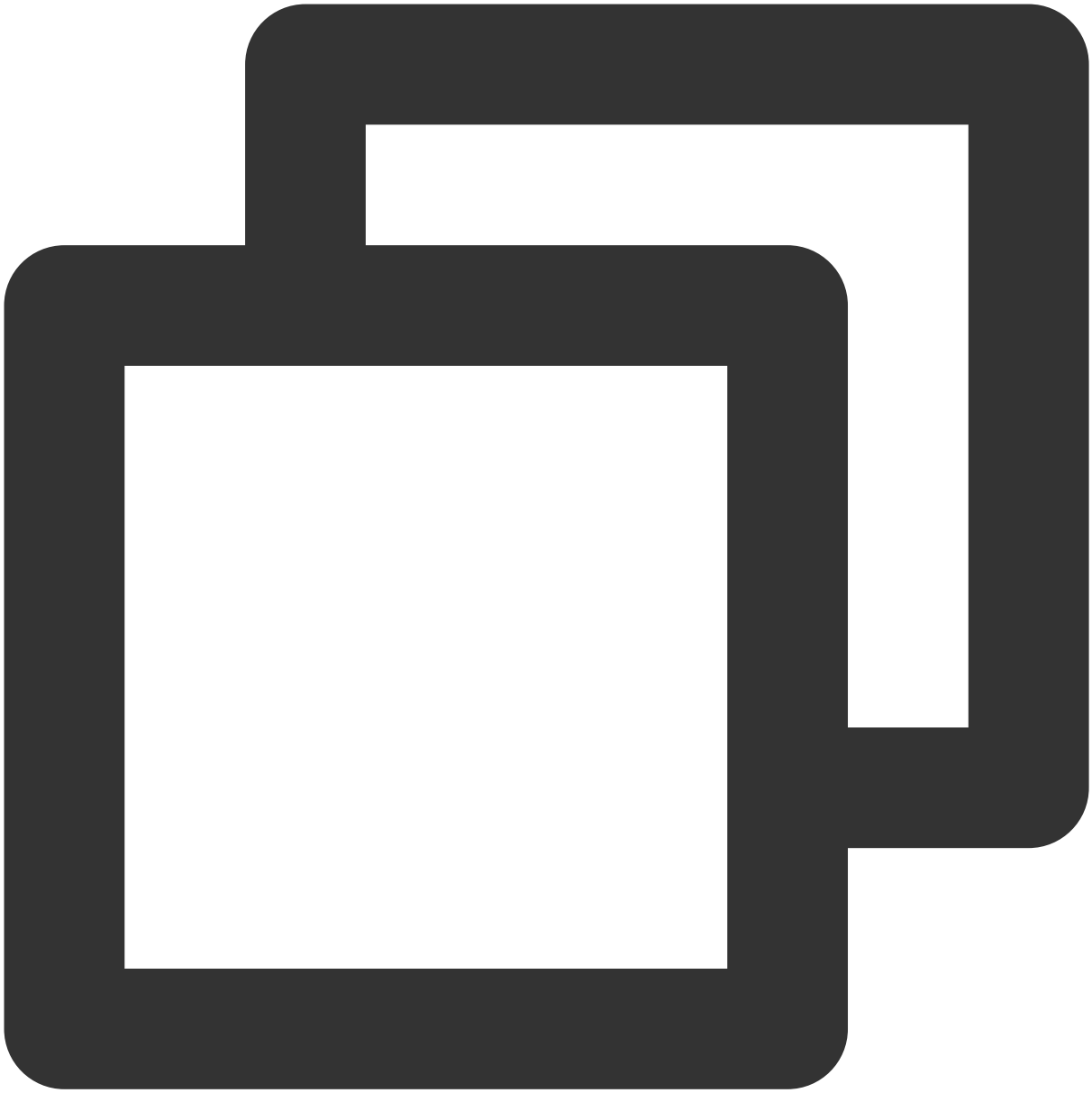
最近更新时间：2024-01-20 16:26:09

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：

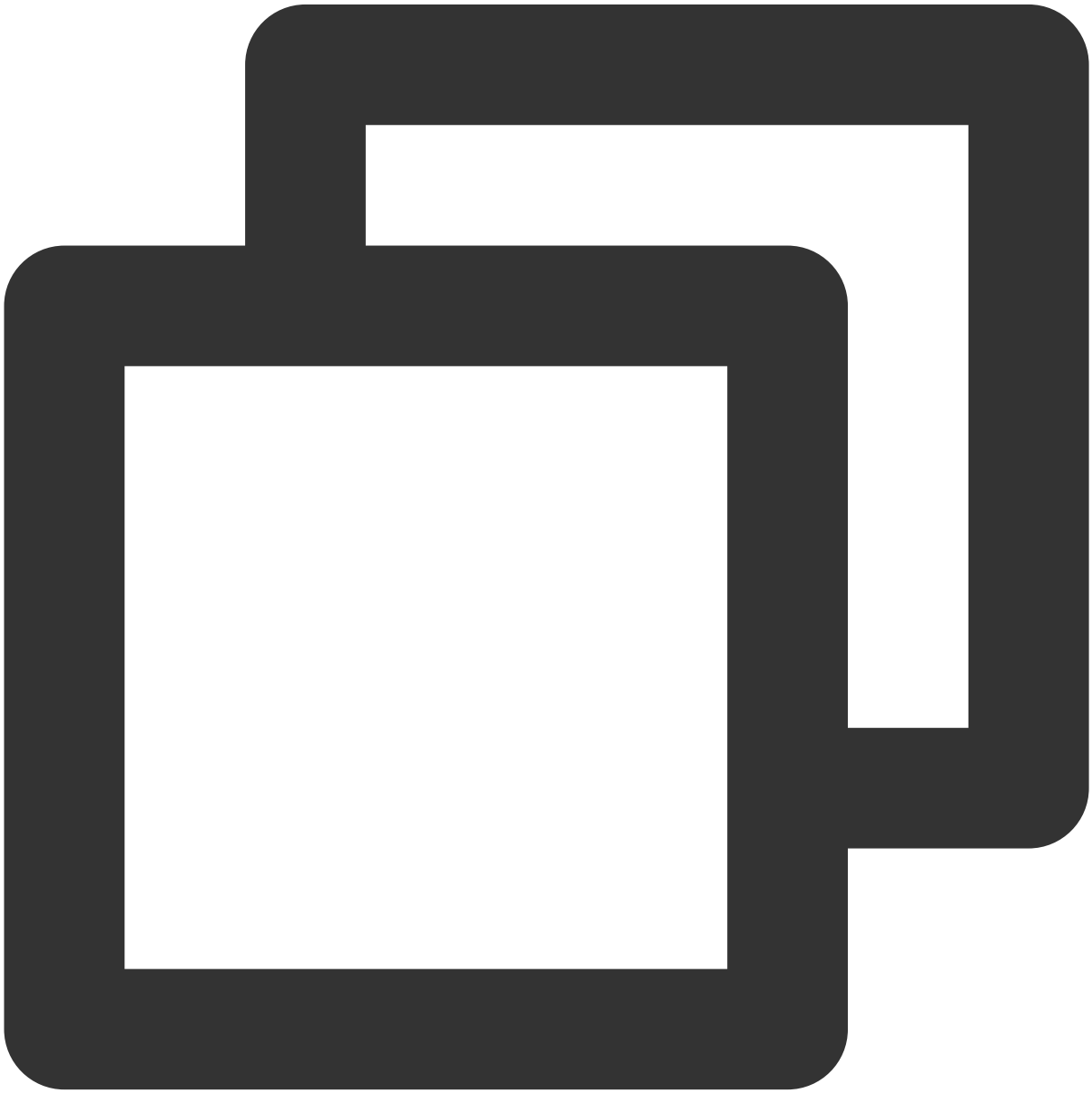
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
                    <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

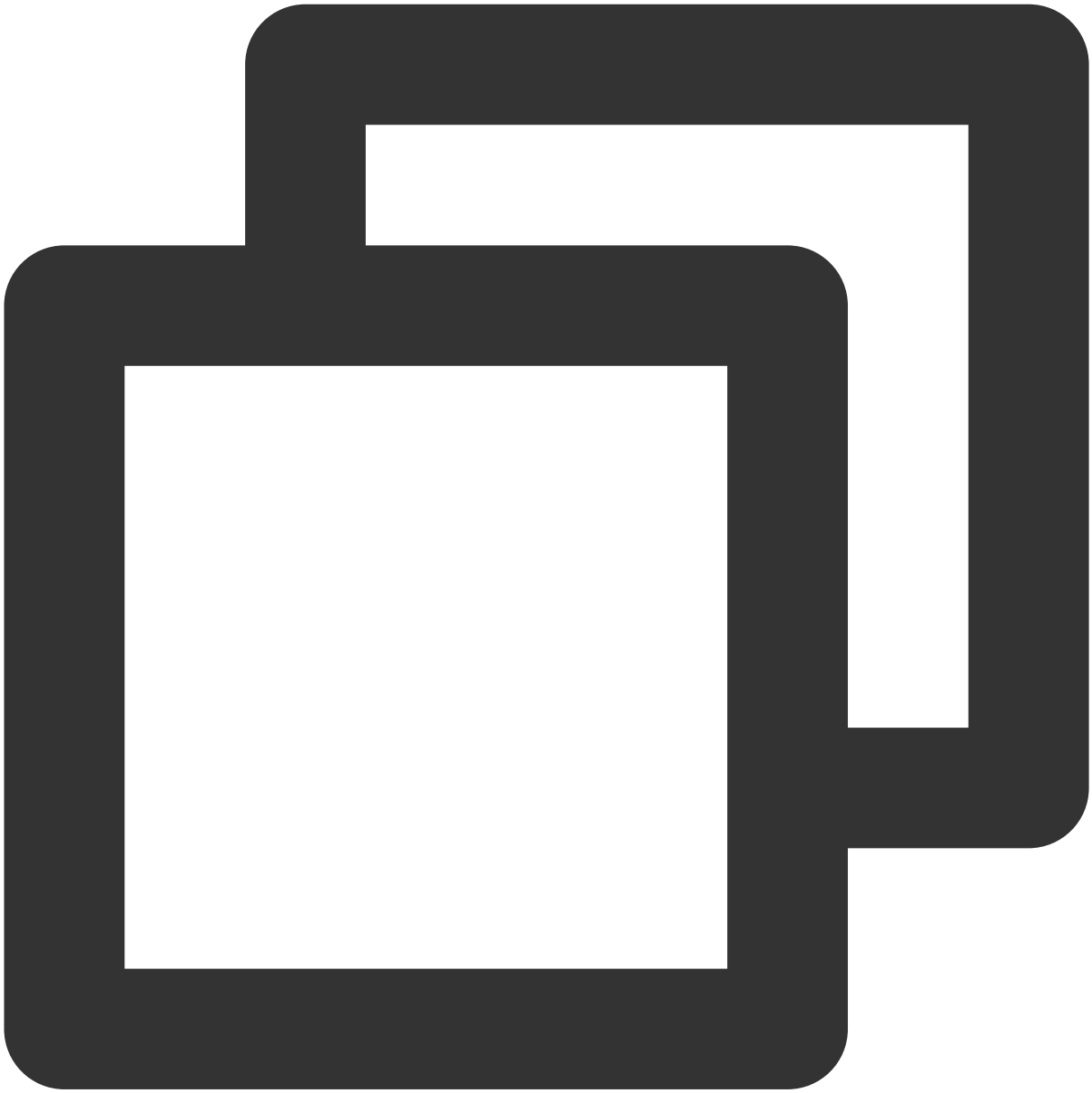
响应

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：

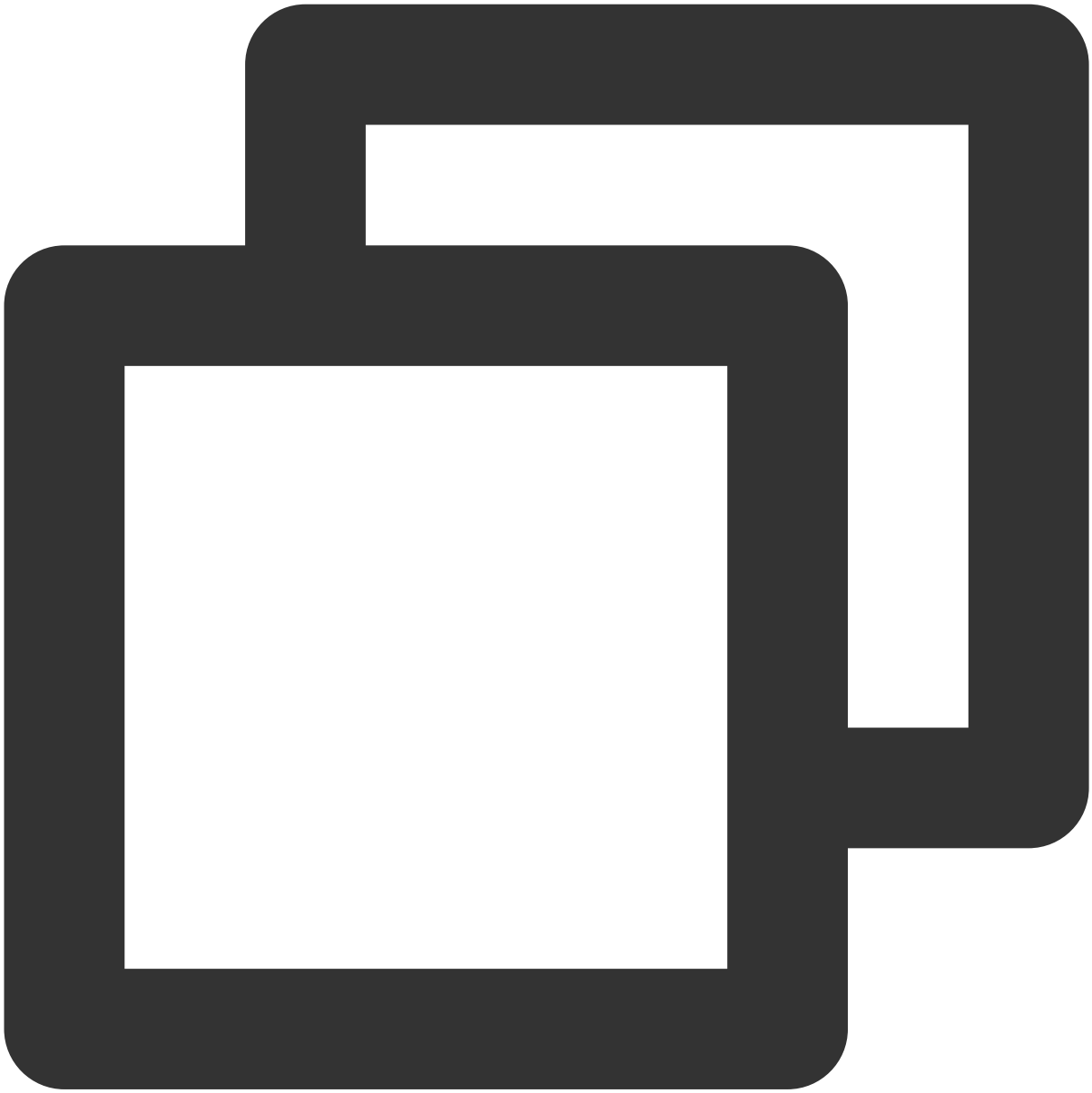
节点名称（关键字）	父节点	描述	类型	是否必选
Response	无	同创建工作流接口的 Response	Container	是

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



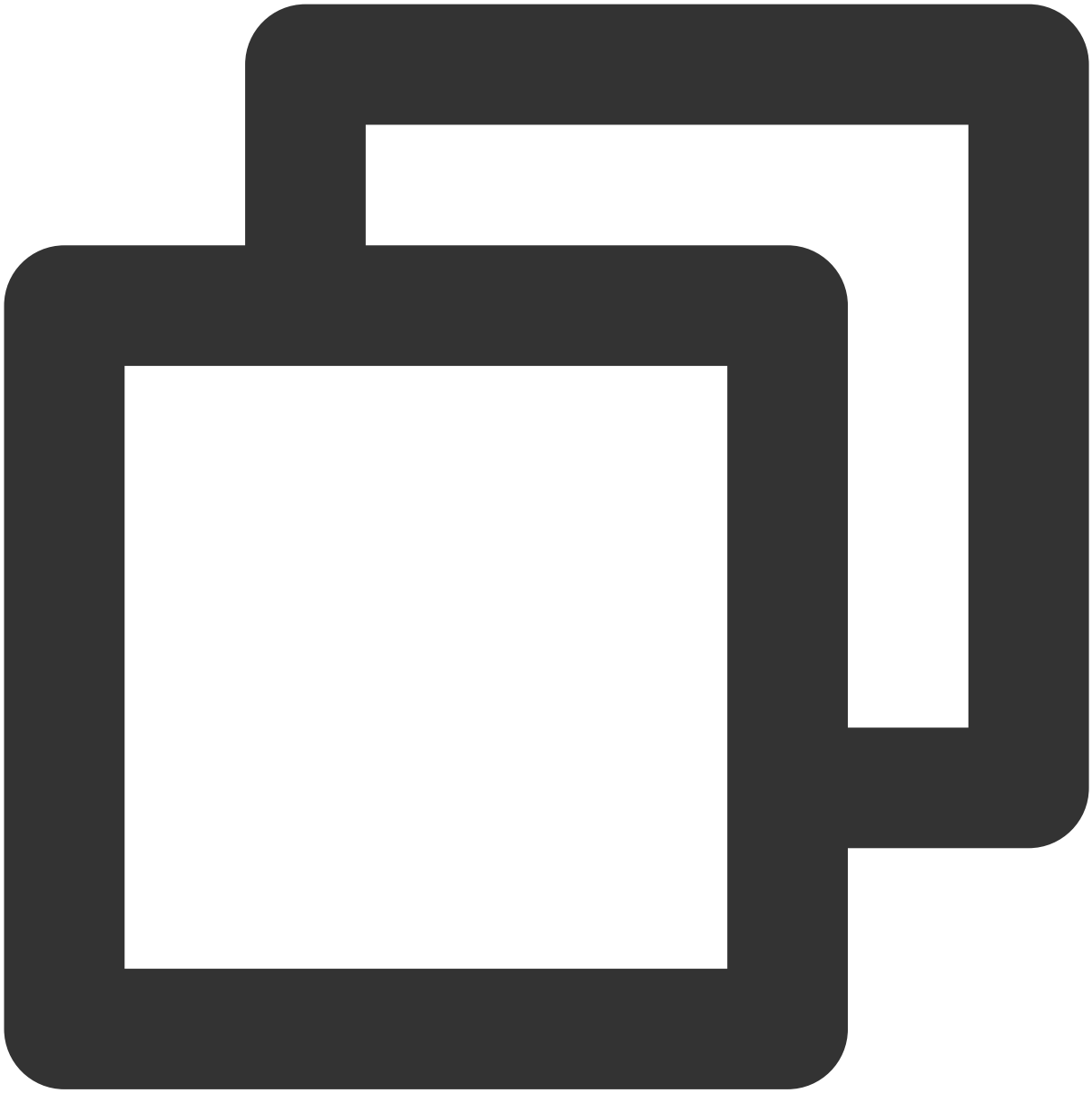
```
PUT /workflow/wc666d0b9f9dd47ae9137a096252d49f7 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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>
            <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

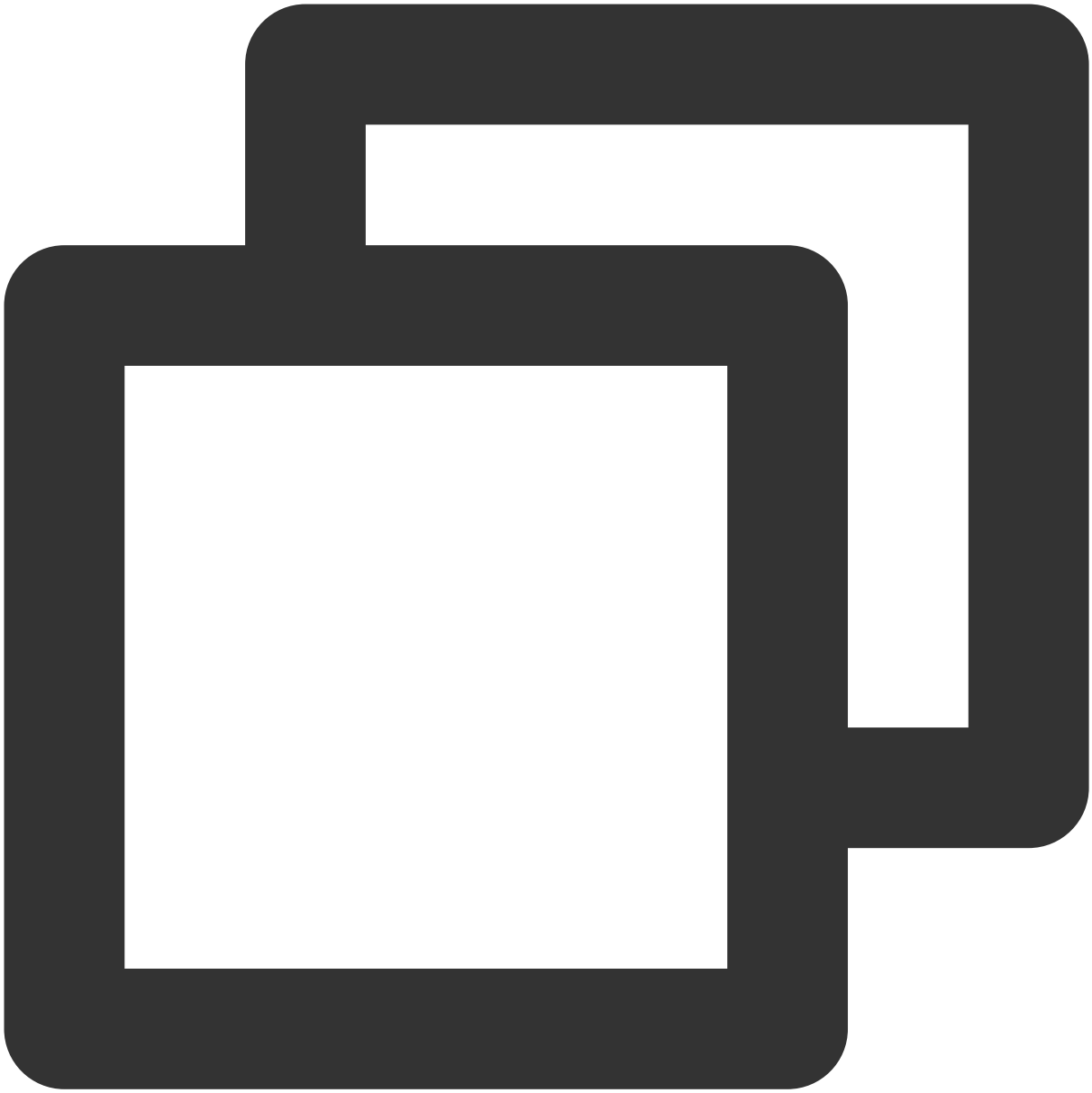
<Response>
```

```
<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>
```

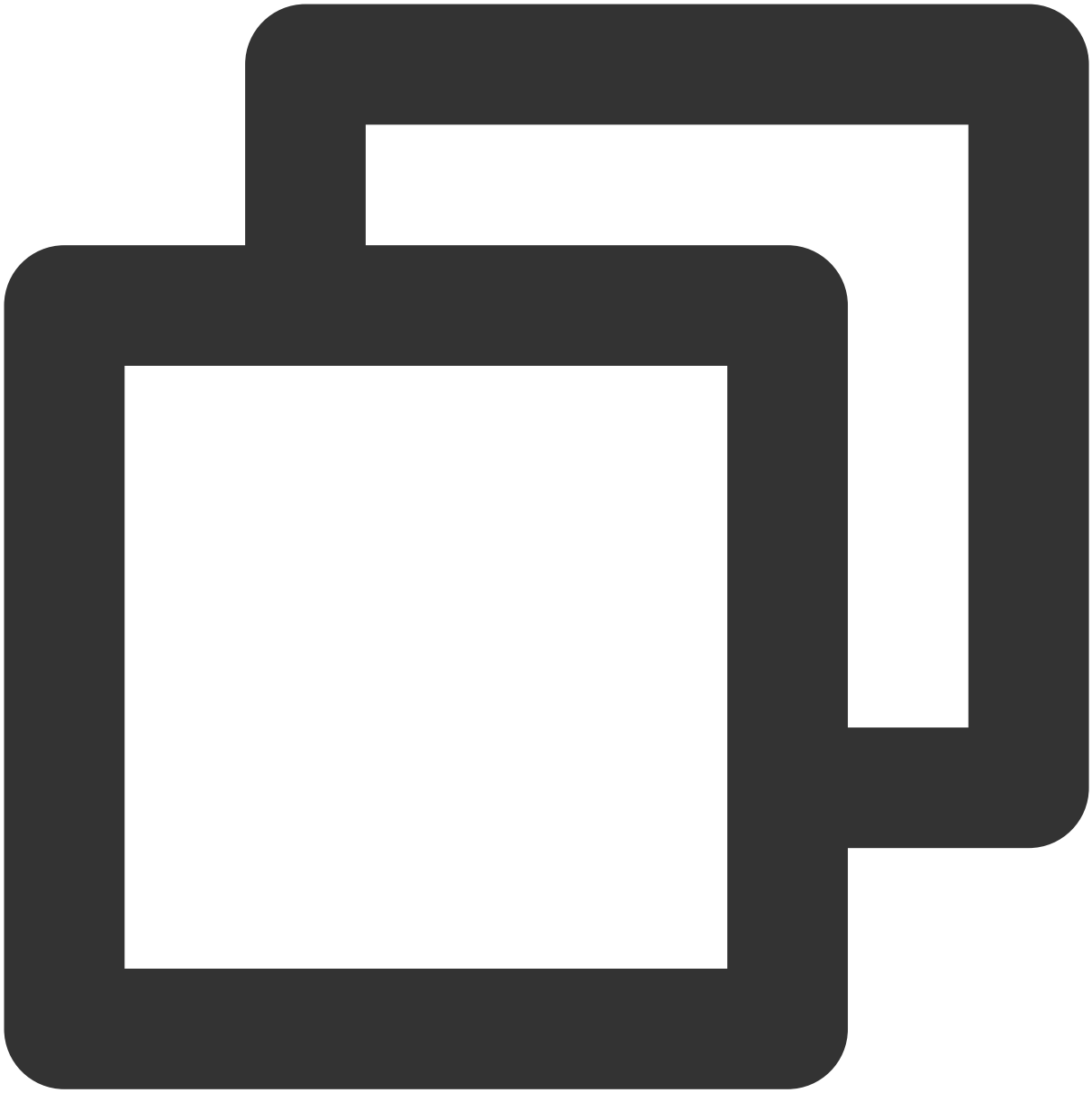
```
                            <Region>ap-chongqing</Region>
                            <Bucket>test-1234567890</Bucket>
                            <Object>bcd/${RunId}/pic.{Ext}</Object>
                        </Output>
                    </Operation>
                </PicProcess_15816659605668>
            </Nodes>
        </Topology>
    </MediaWorkflow>
</Response>
```

请求2：停用工作流



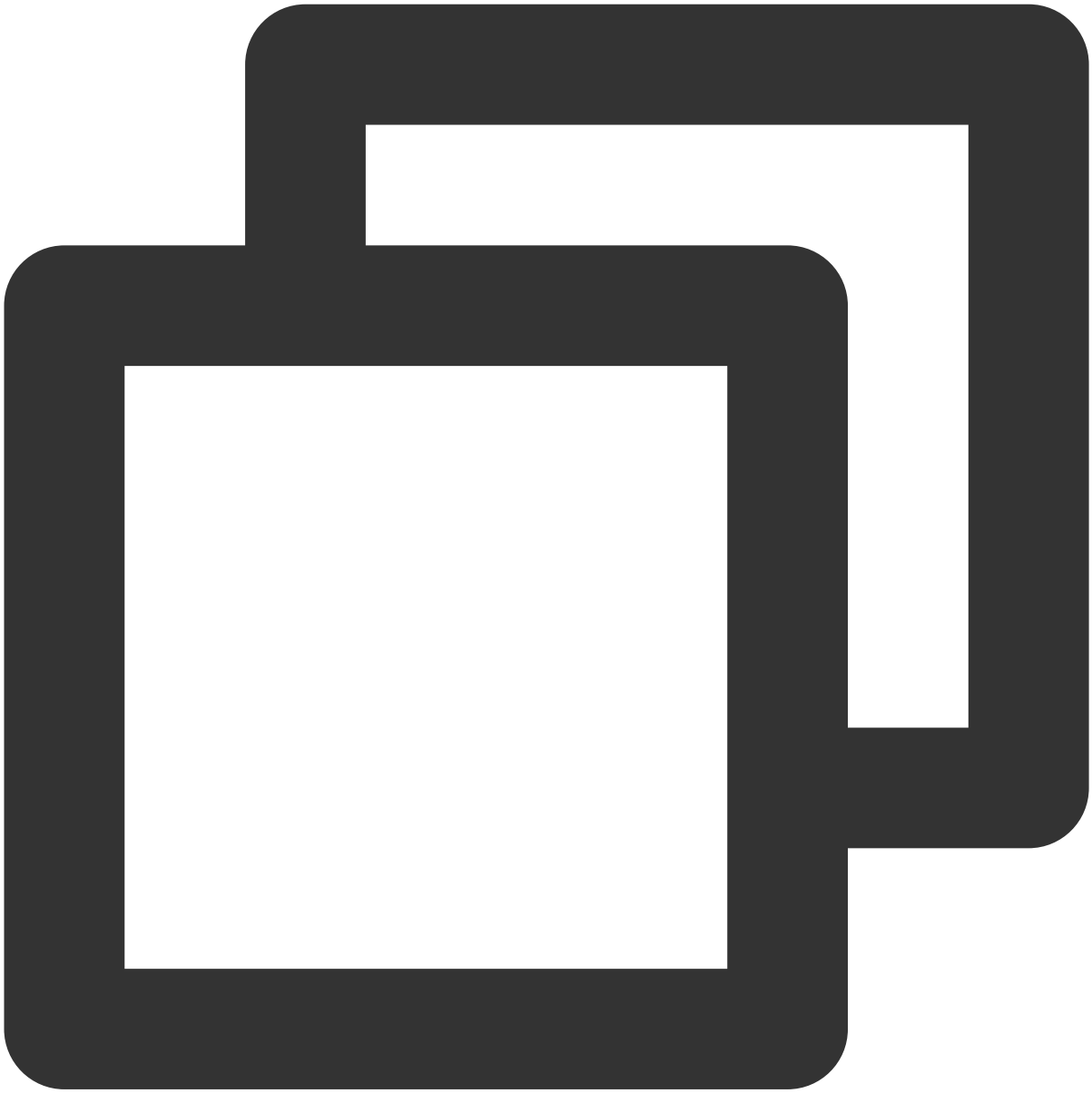
```
PUT /workflow/wc666d0b9f9dd47ae9137a096252d49f7?paused HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
```

响应2



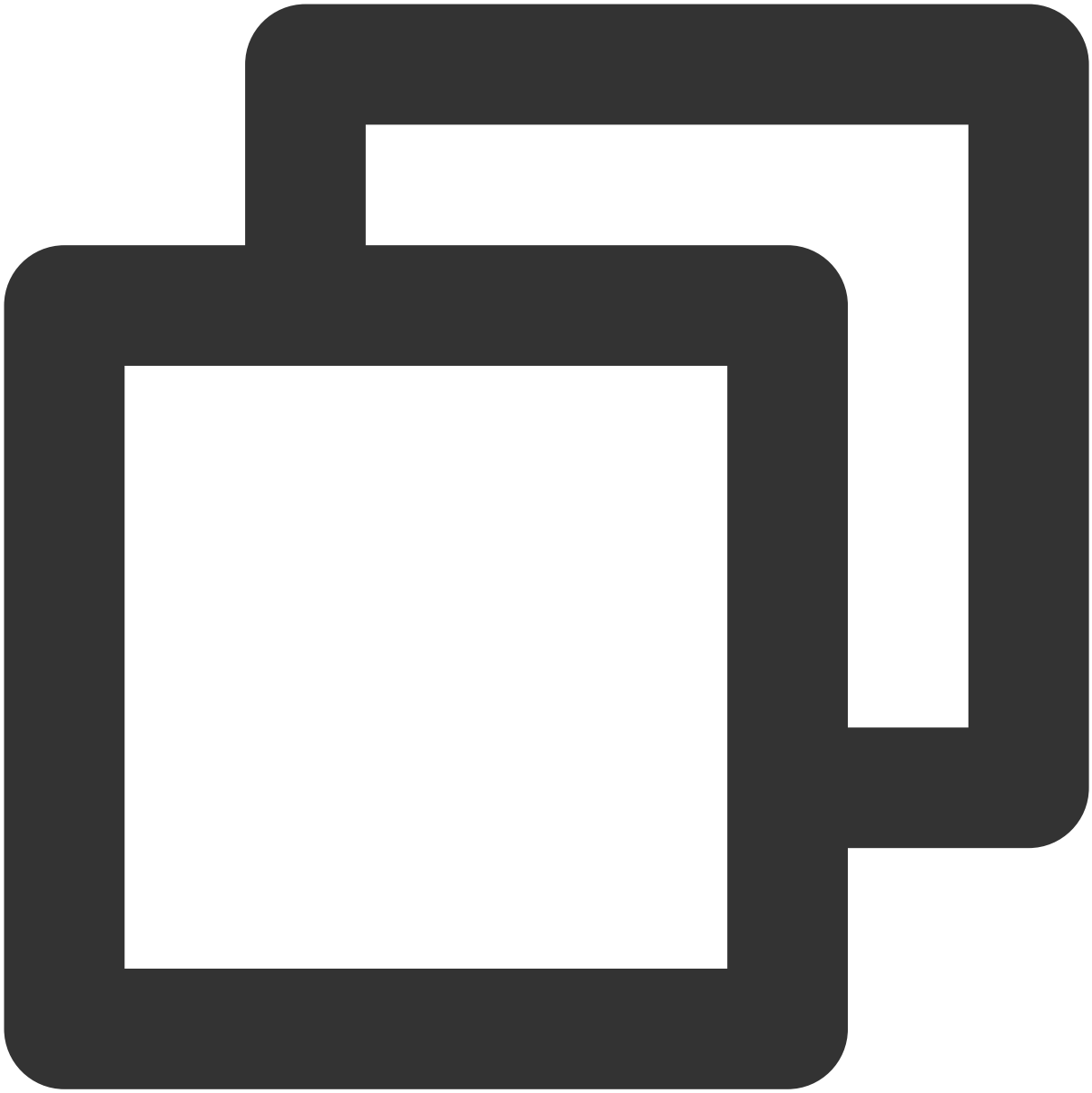
```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjJmMWRiMDlfOTBmYTUwNjRfNWYzMl80
```

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

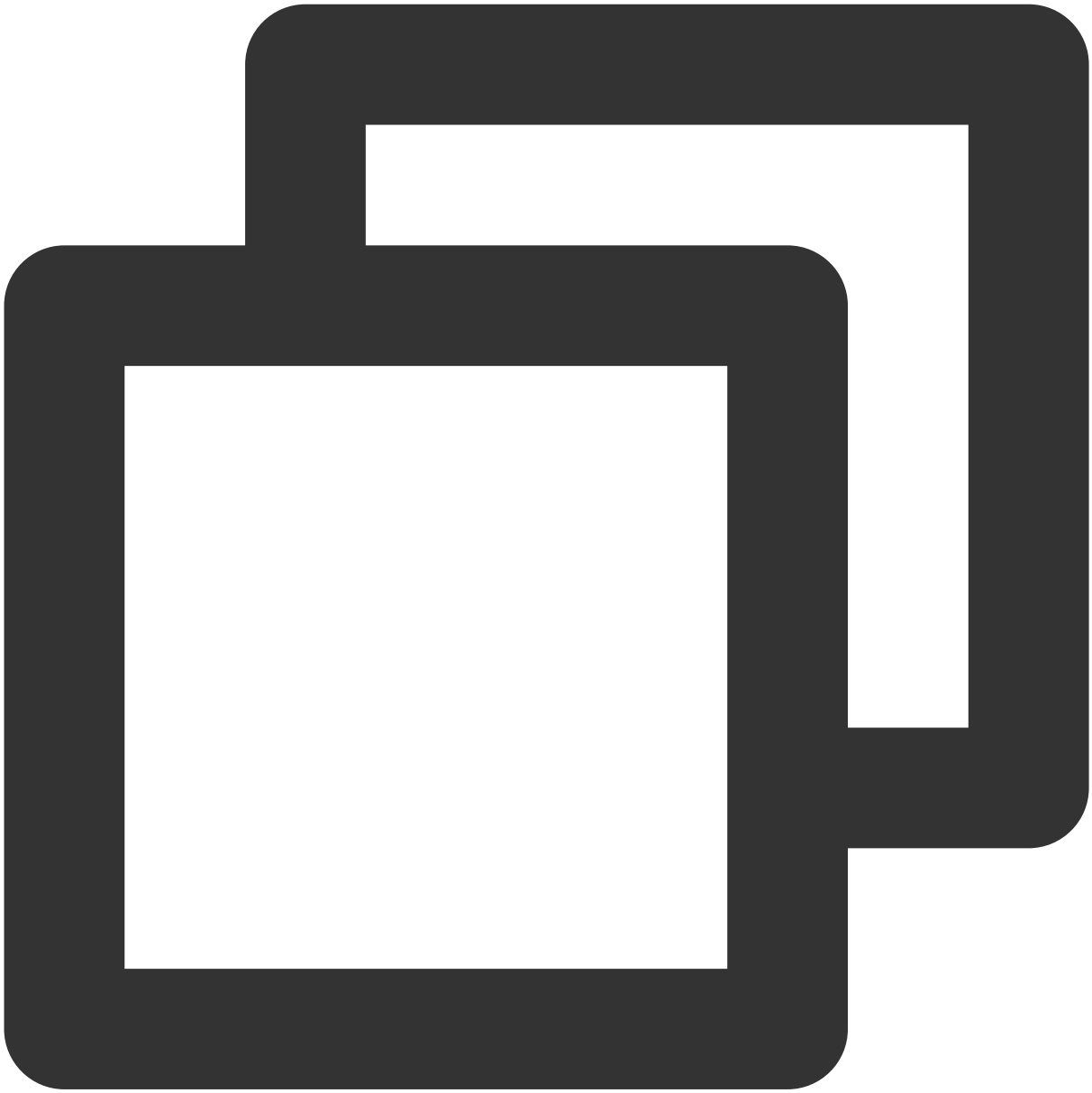
最近更新时间：2024-01-20 16:26:09

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。

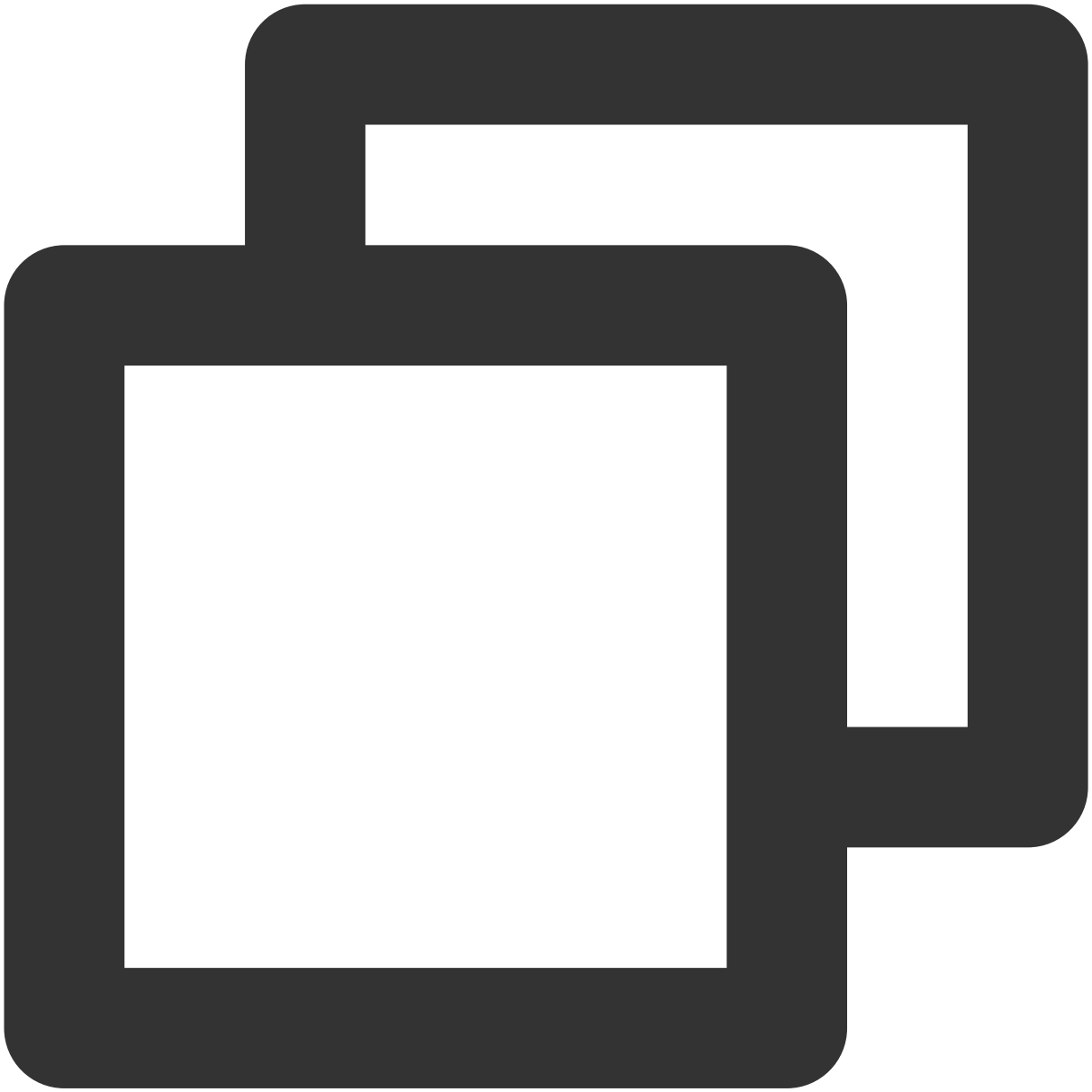
响应

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：

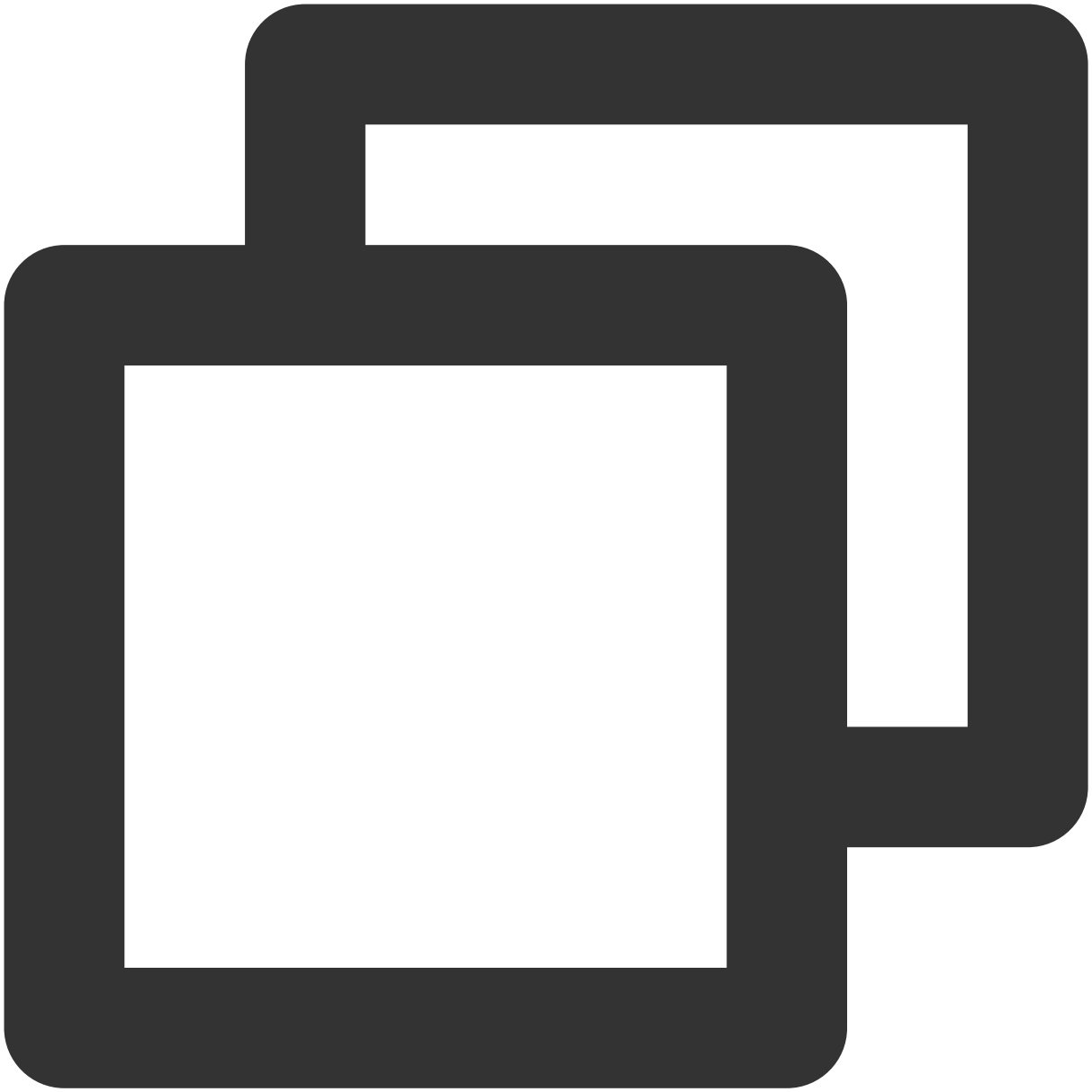
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
InstanceId	Response	实例 ID	String

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。

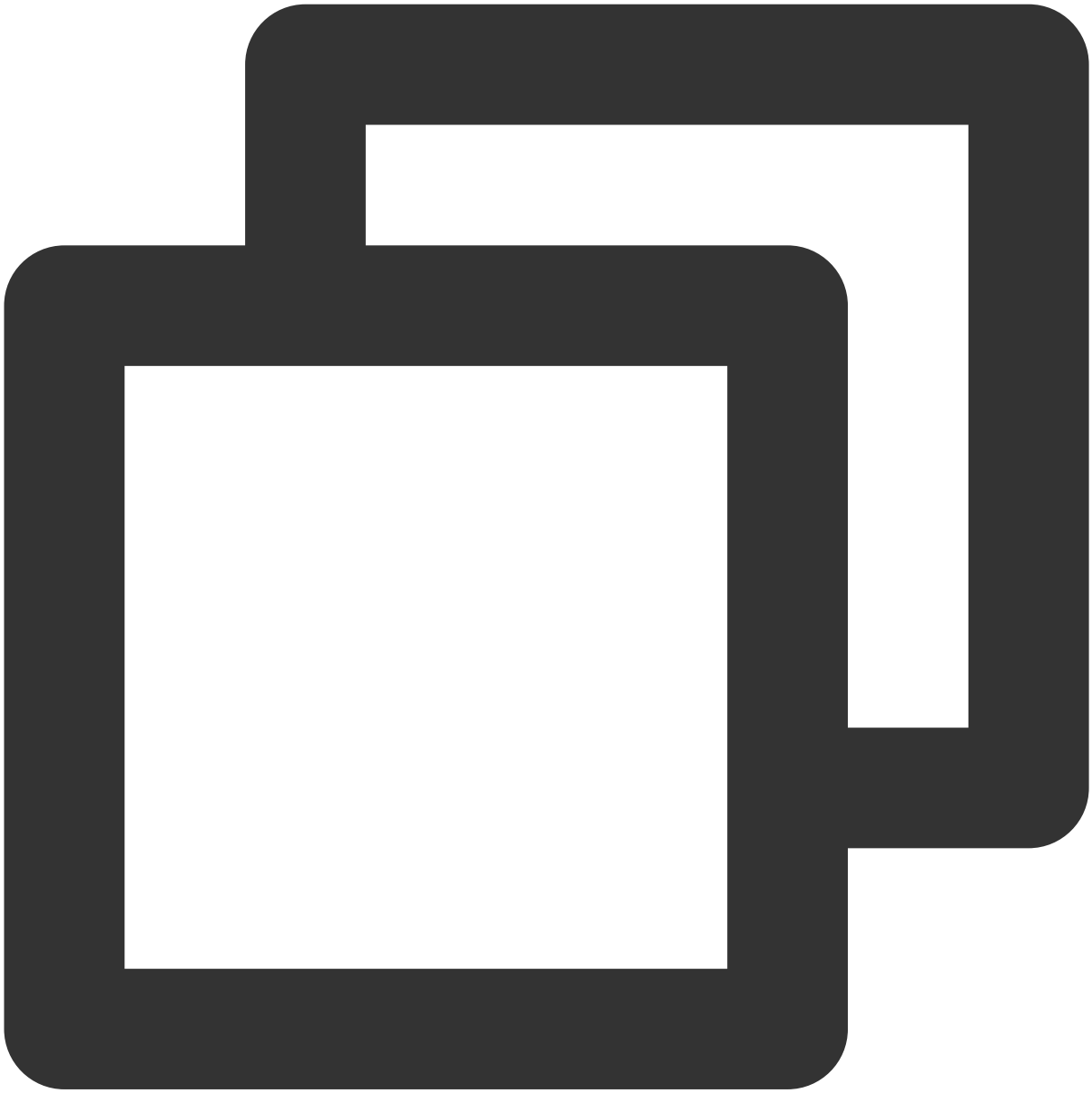
实际案例

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

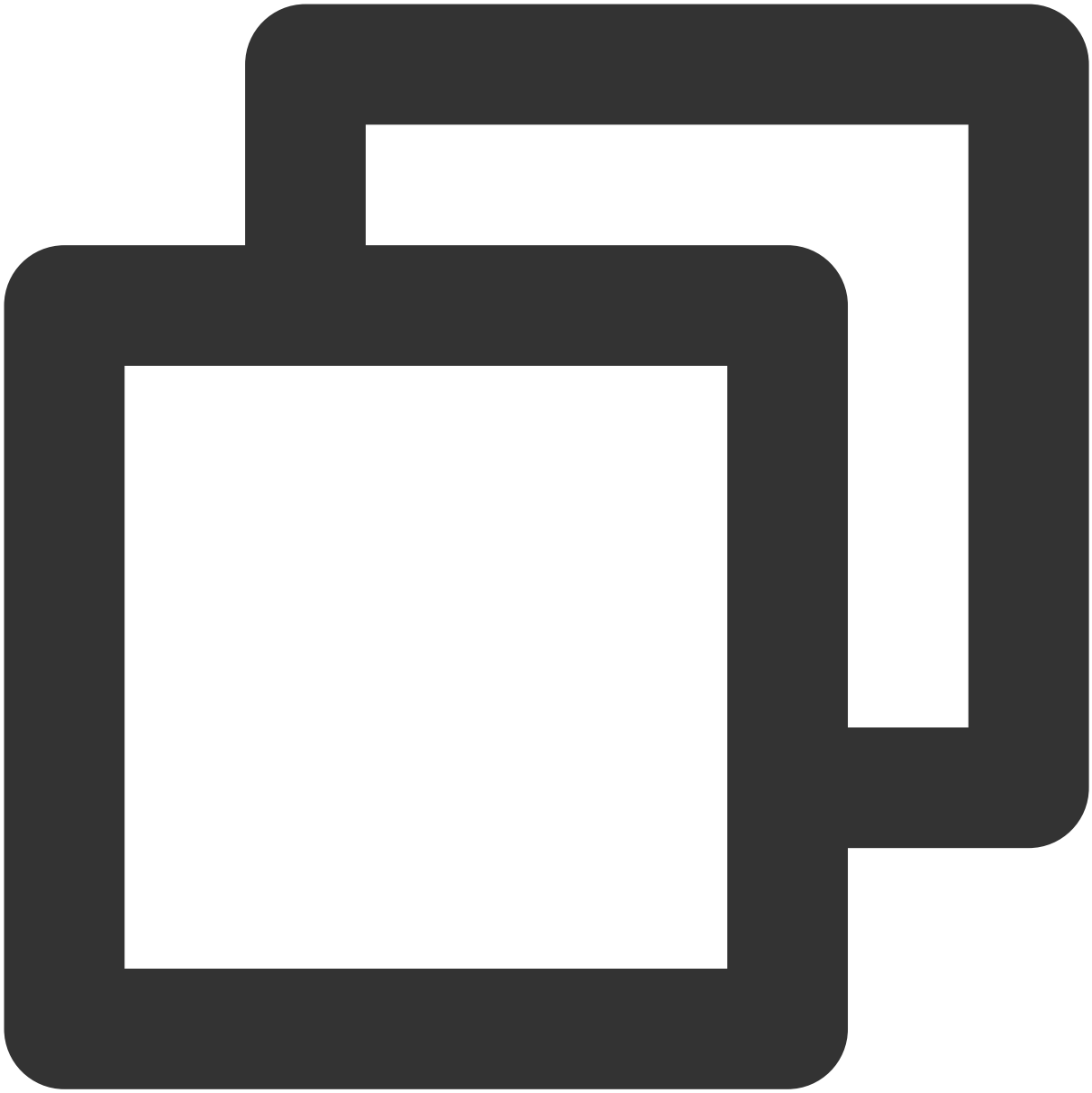
最近更新时间：2024-01-20 16:26:09

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>
```

```
            <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（关键字）	父节点	描述	类型
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：

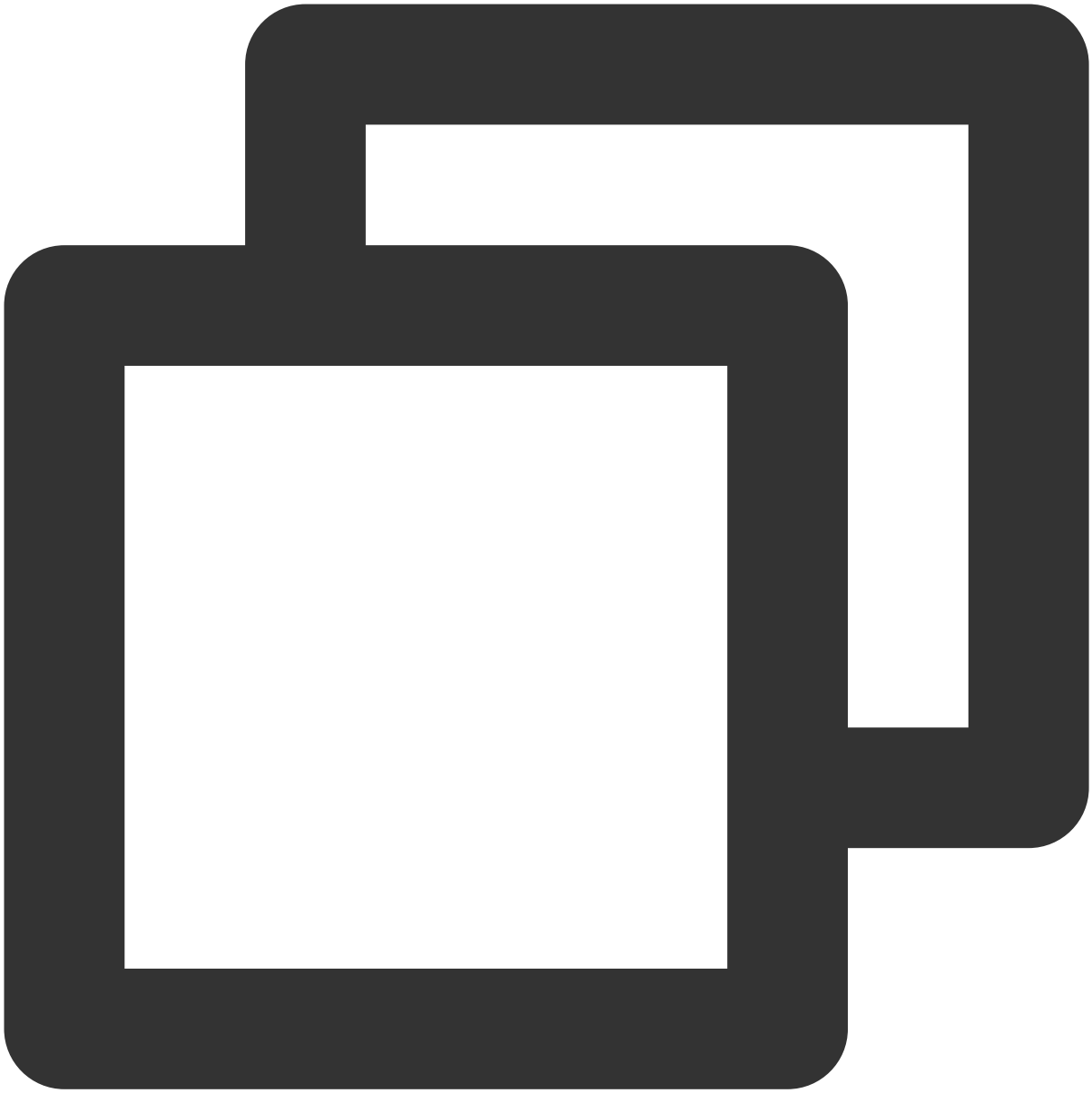
节点名称（关键字）	父节点	描述	类型
Type	Response.WorkflowExecution.Tasks	任务类型	String
Createtime	Response.WorkflowExecution.Tasks	任务的创建时间	String
StartTime	Response.WorkflowExecution.Tasks	任务开始时间	String
EndTime	Response.WorkflowExecution.Tasks	任务的完成时间	String
State	Response.WorkflowExecution.Tasks	任务的状态，值为 Success、	String

版权所有：腾讯云计算（北京）有限责任公司

		Failed 等	
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

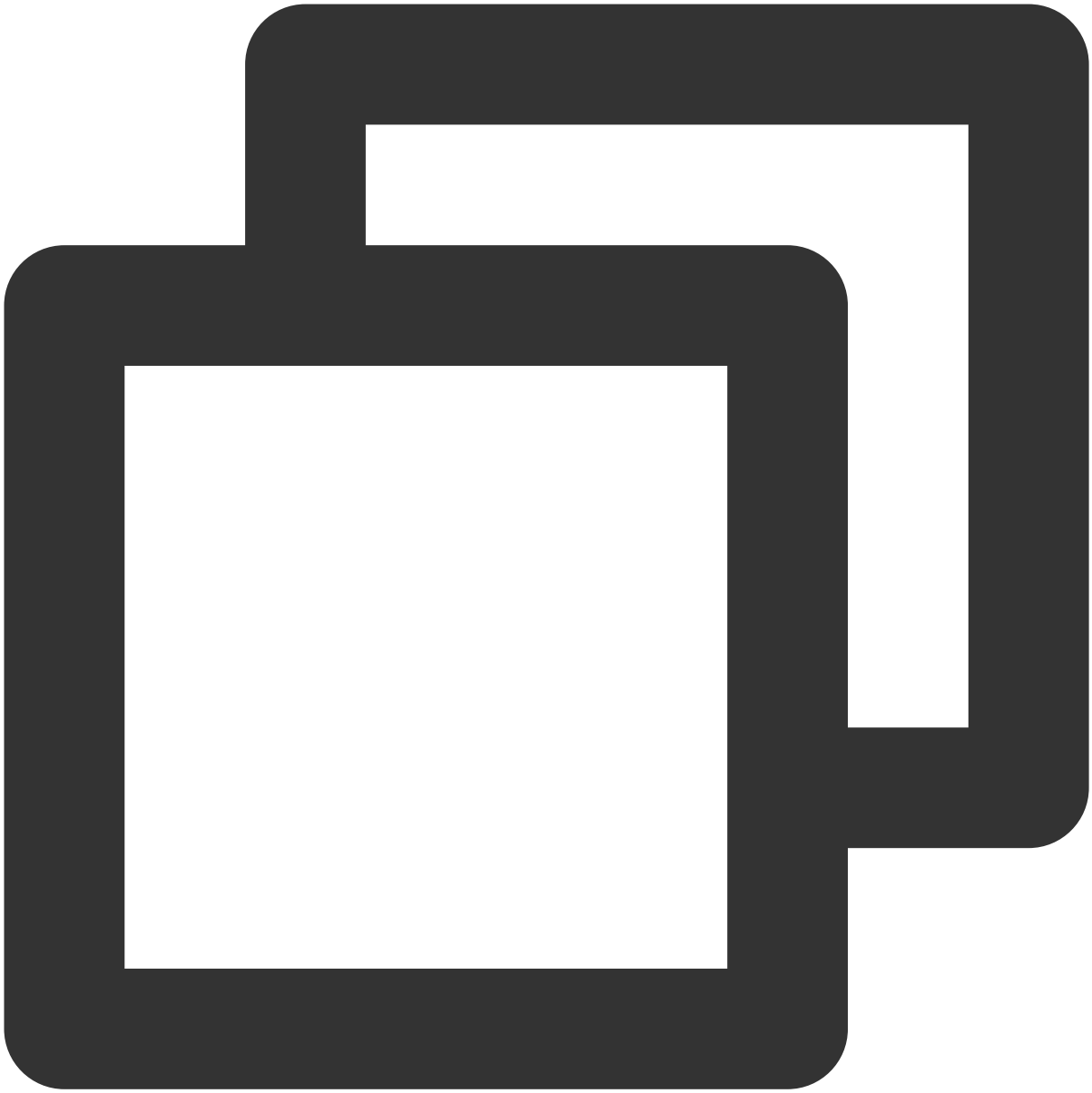
案例 1 (XML)



```
<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

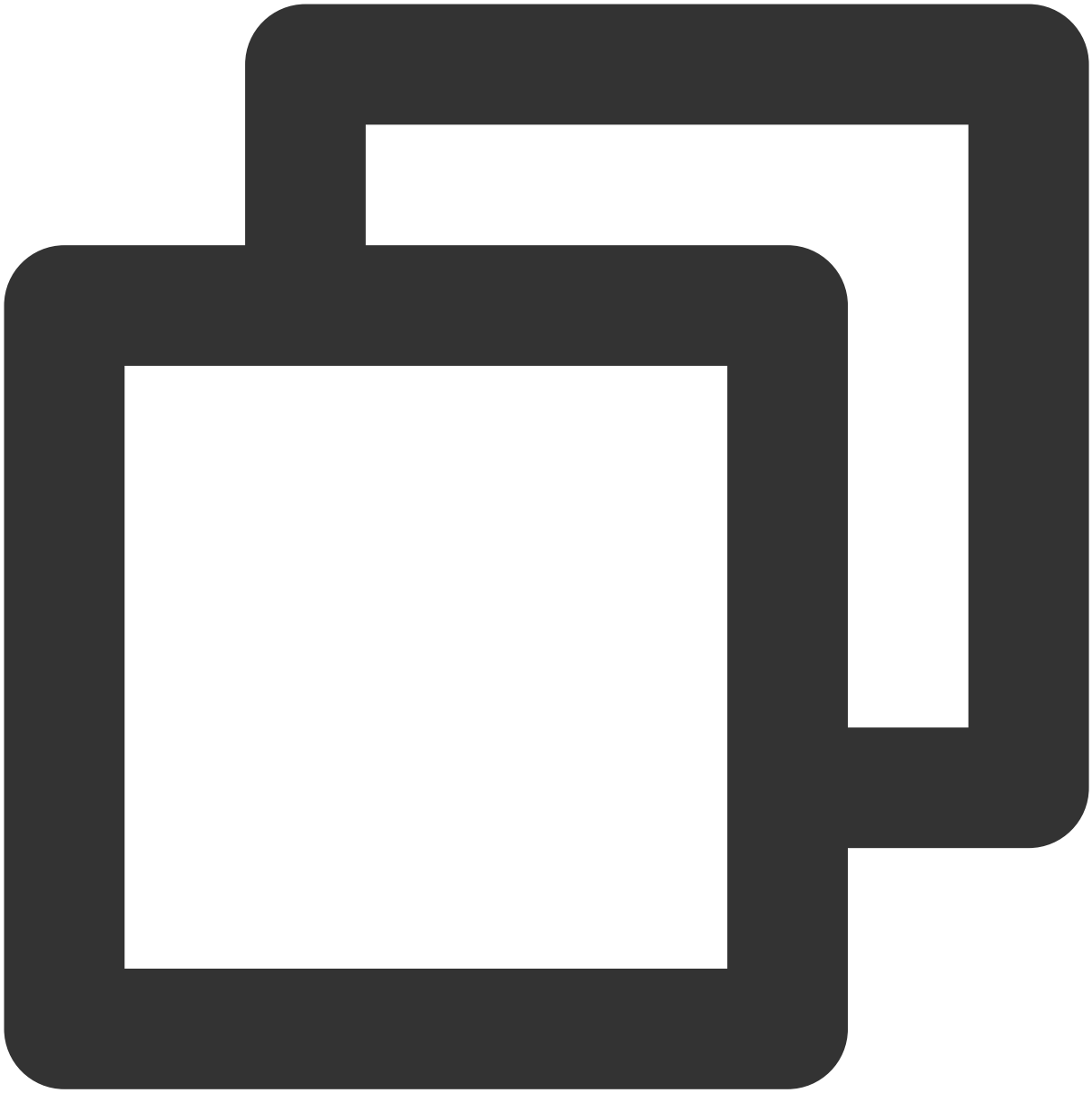
最近更新时间：2024-01-20 16:26:09

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。

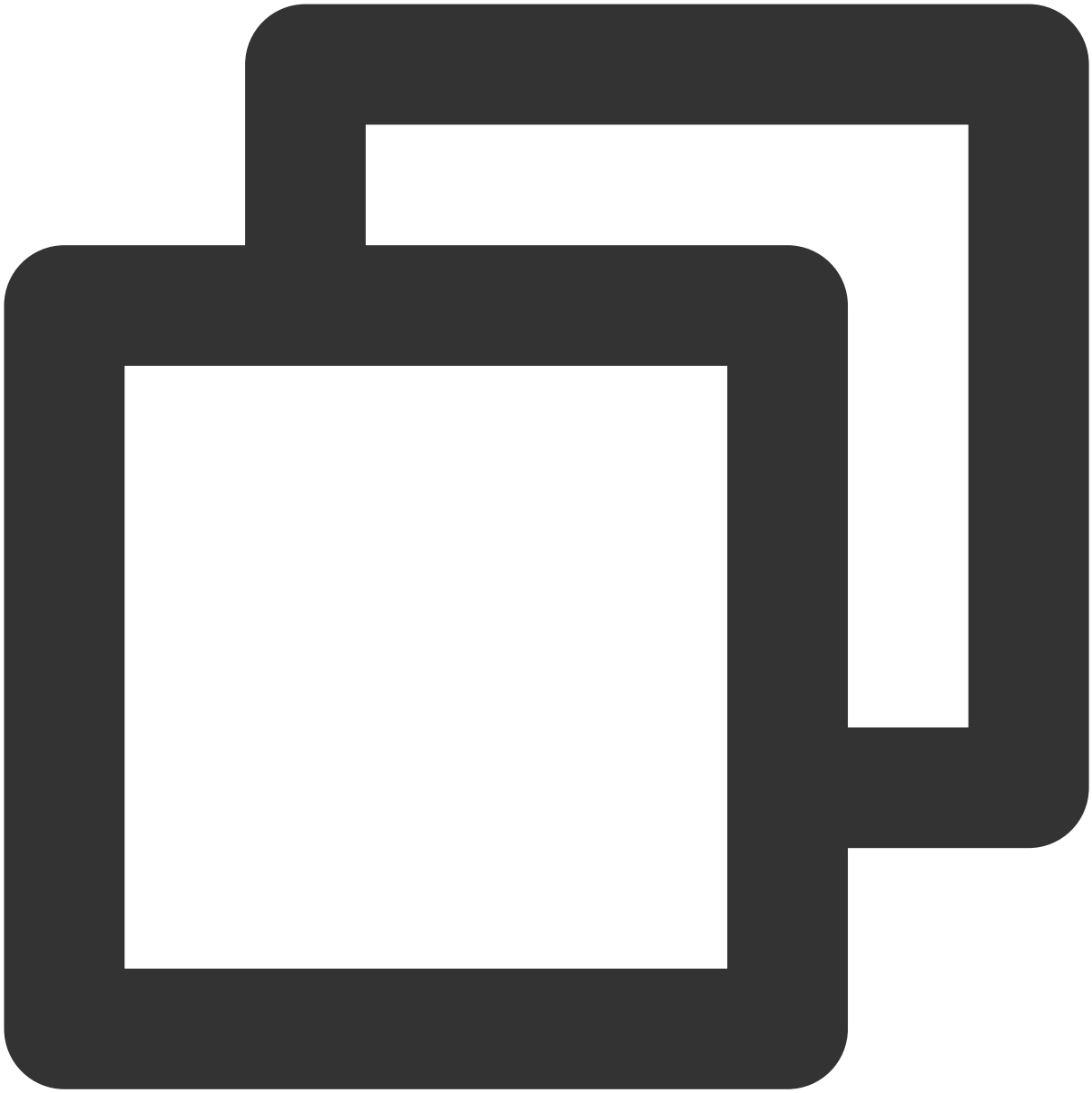
响应

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：

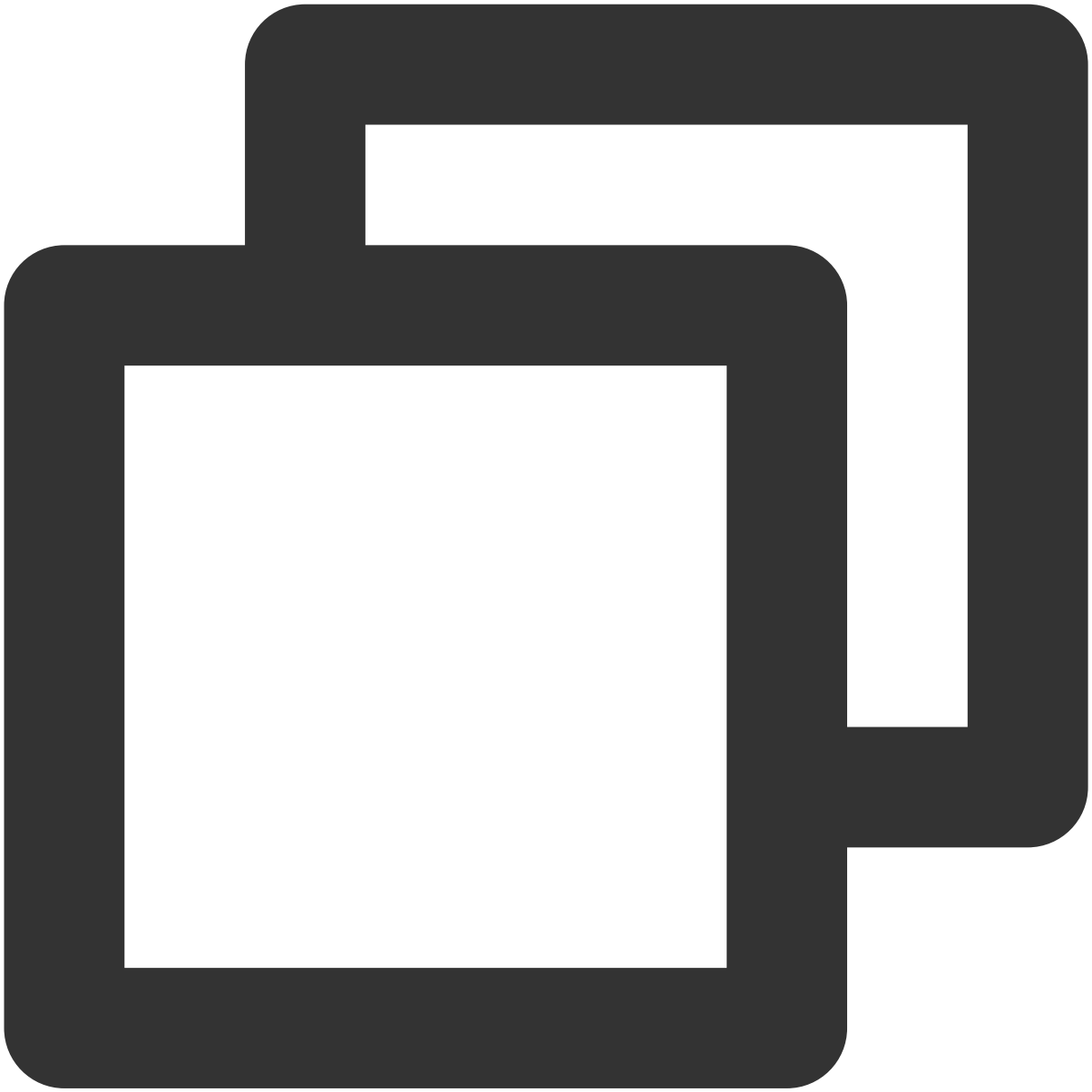
节点名称（关键字）	父节点	描述	类型
Type	Response.WorkflowExecution.Tasks	任务所属节点类型	String
State	Response.WorkflowExecution.Tasks	任务状态	String
JobId	Response.WorkflowExecution.Tasks	任务 ID	String
CreateTime	Response.WorkflowExecution.Tasks	任务创建时间	String
StartTime	Response.WorkflowExecution.Tasks	任务开始时间	String
EndTime	Response.WorkflowExecution.Tasks	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。

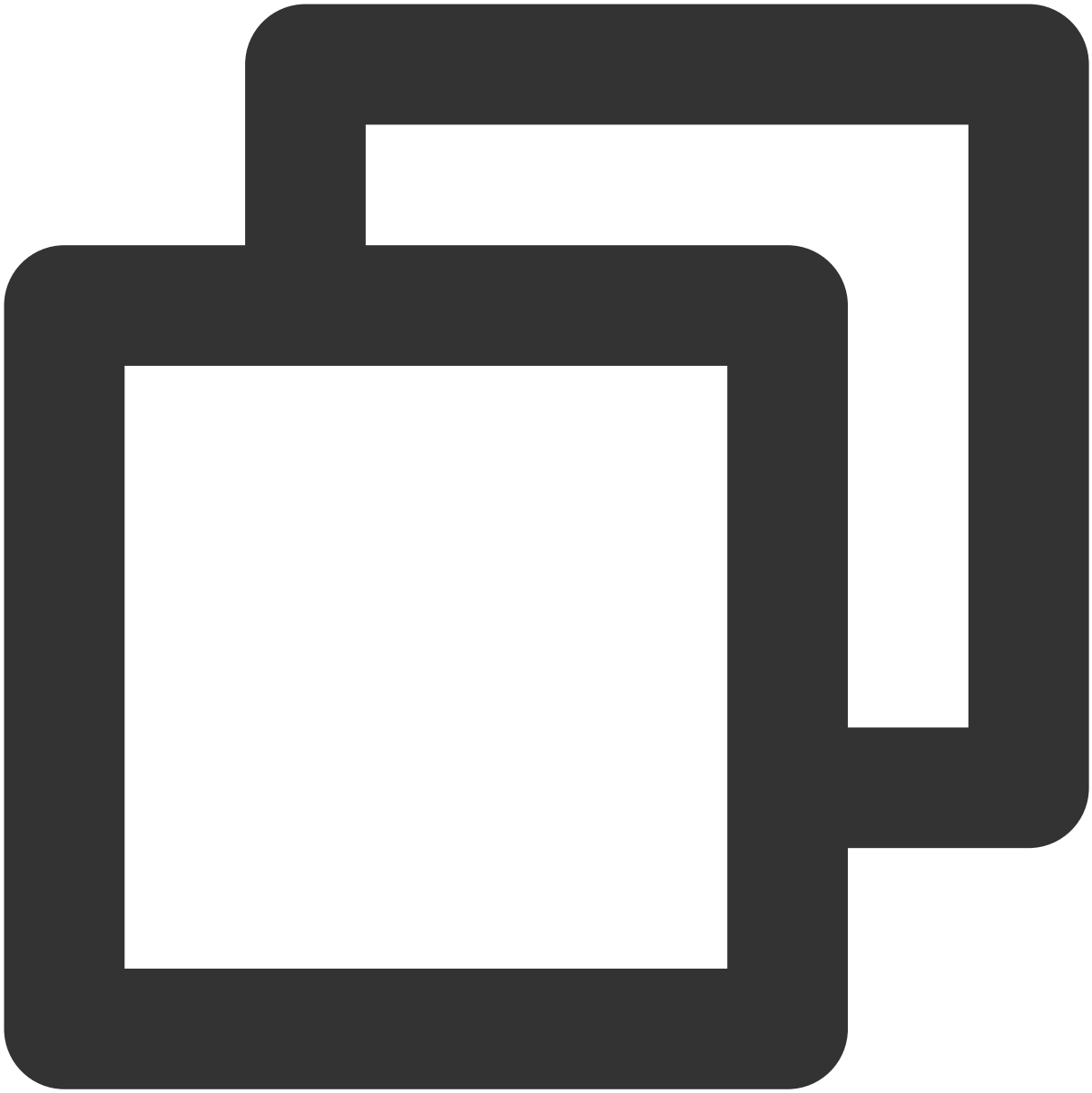
实际案例

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>
```


版权所有：腾讯云计算（北京）有限责任公司

获取工作流实例详情列表

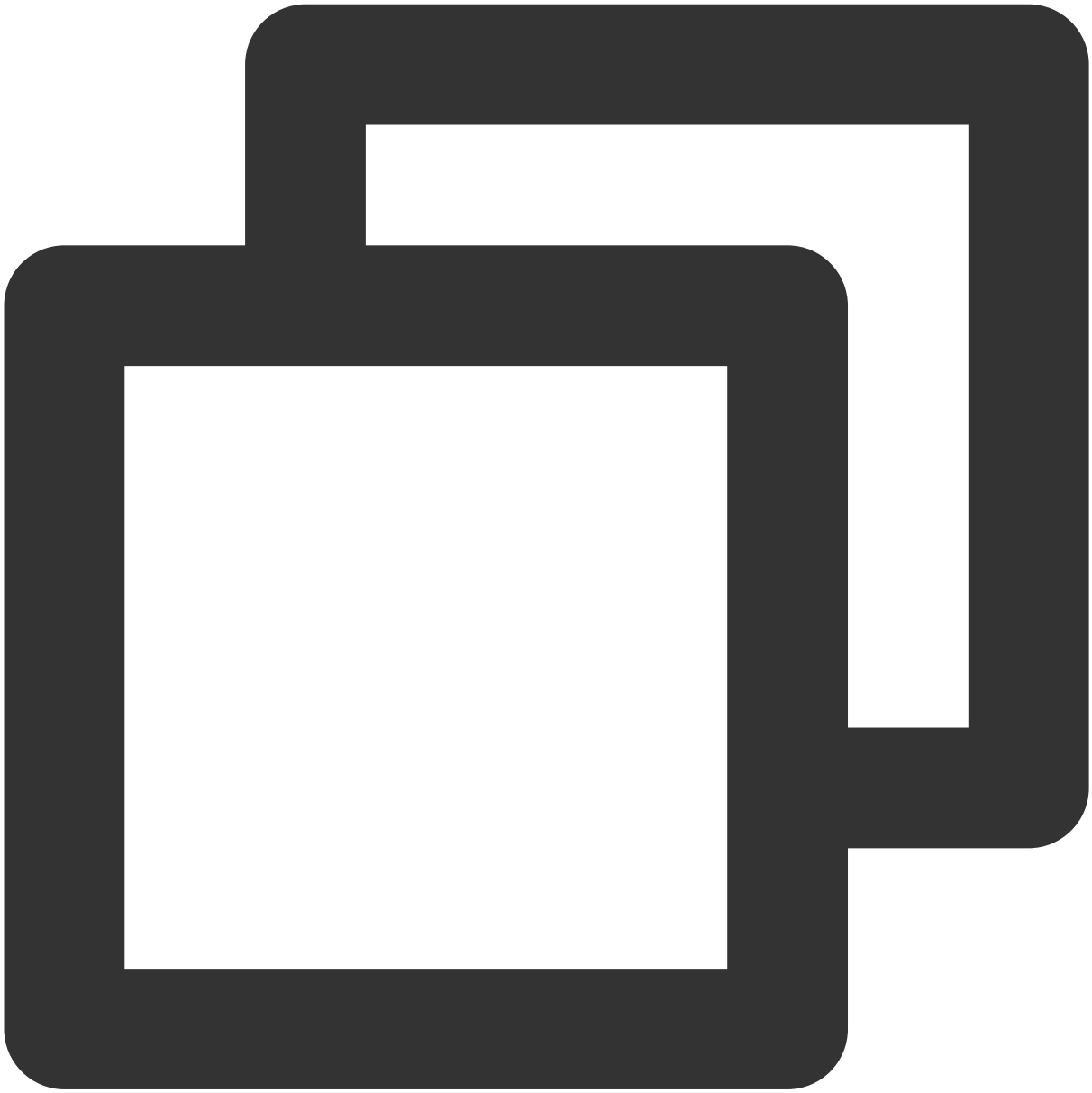
最近更新时间：2024-01-20 16:26:09

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

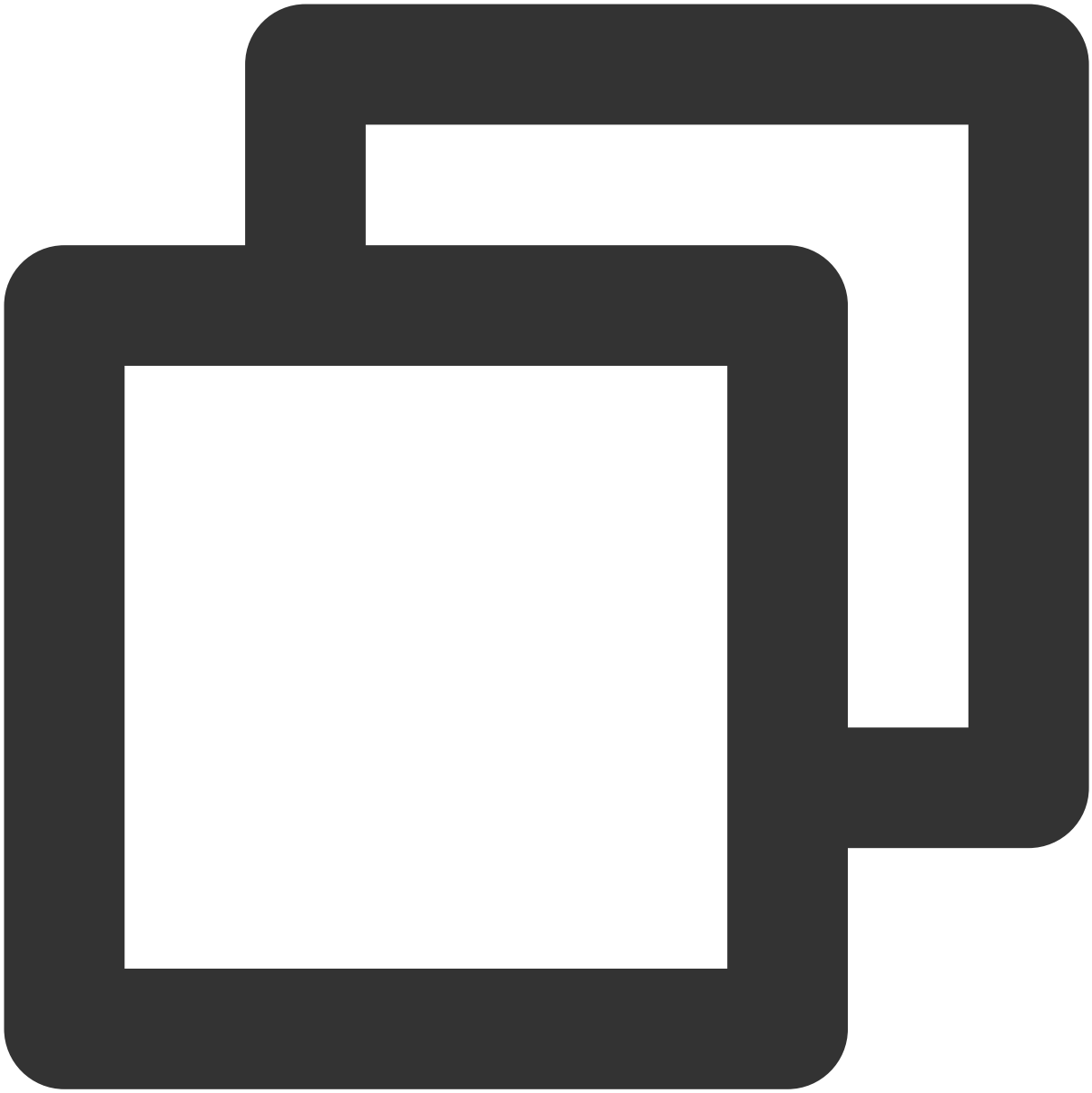
响应

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：

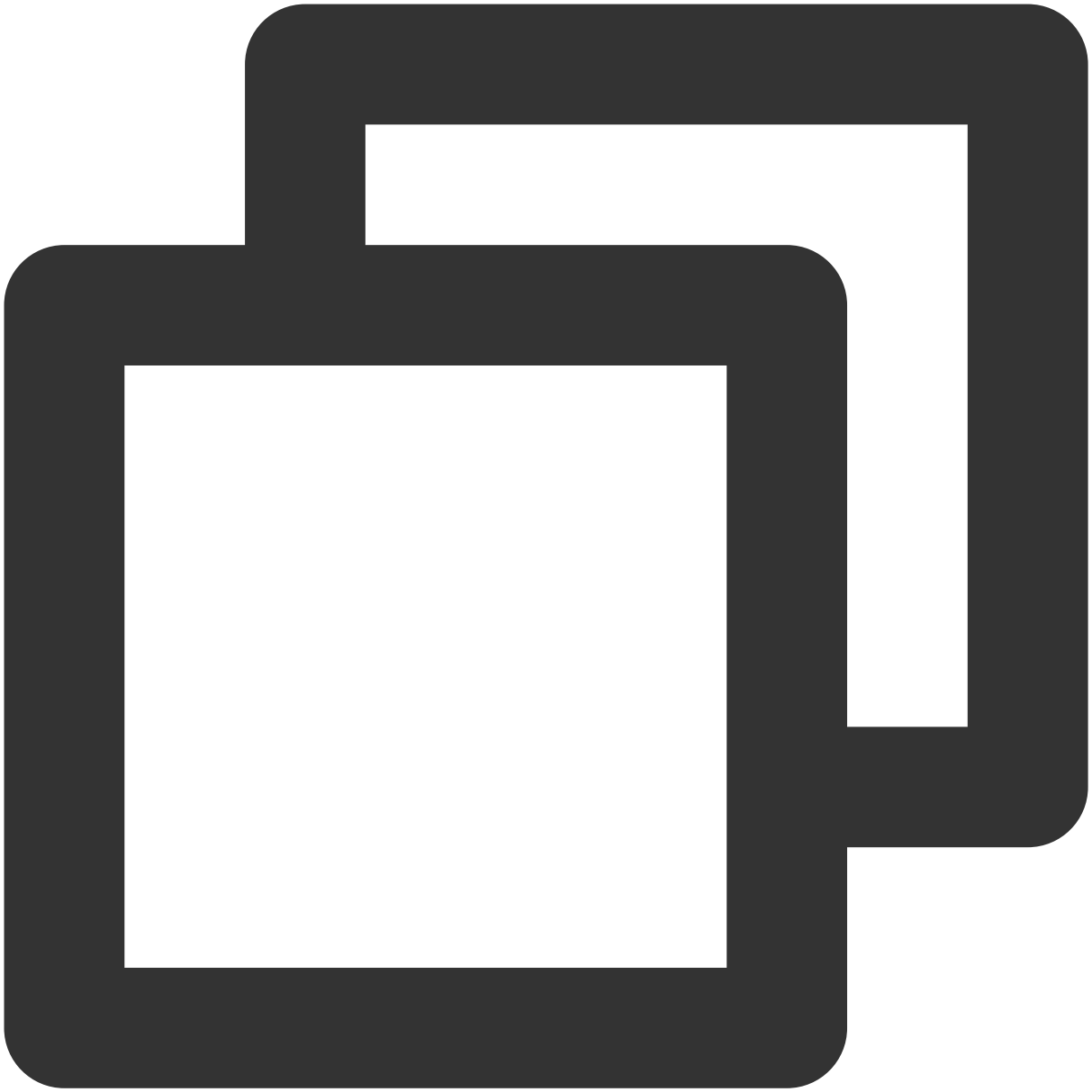
节点名称（关键字）	父节点	描述	类型
RunId	Response.WorkflowExecutionList	工作流实例 ID	String
WorkflowId	Response.WorkflowExecutionList	工作流 ID	String
State	Response.WorkflowExecutionList	工作流实例状态	String
CreateTime	Response.WorkflowExecutionList	创建时间	String
Object	Response.WorkflowExecutionList	COS 对象地址	String

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。

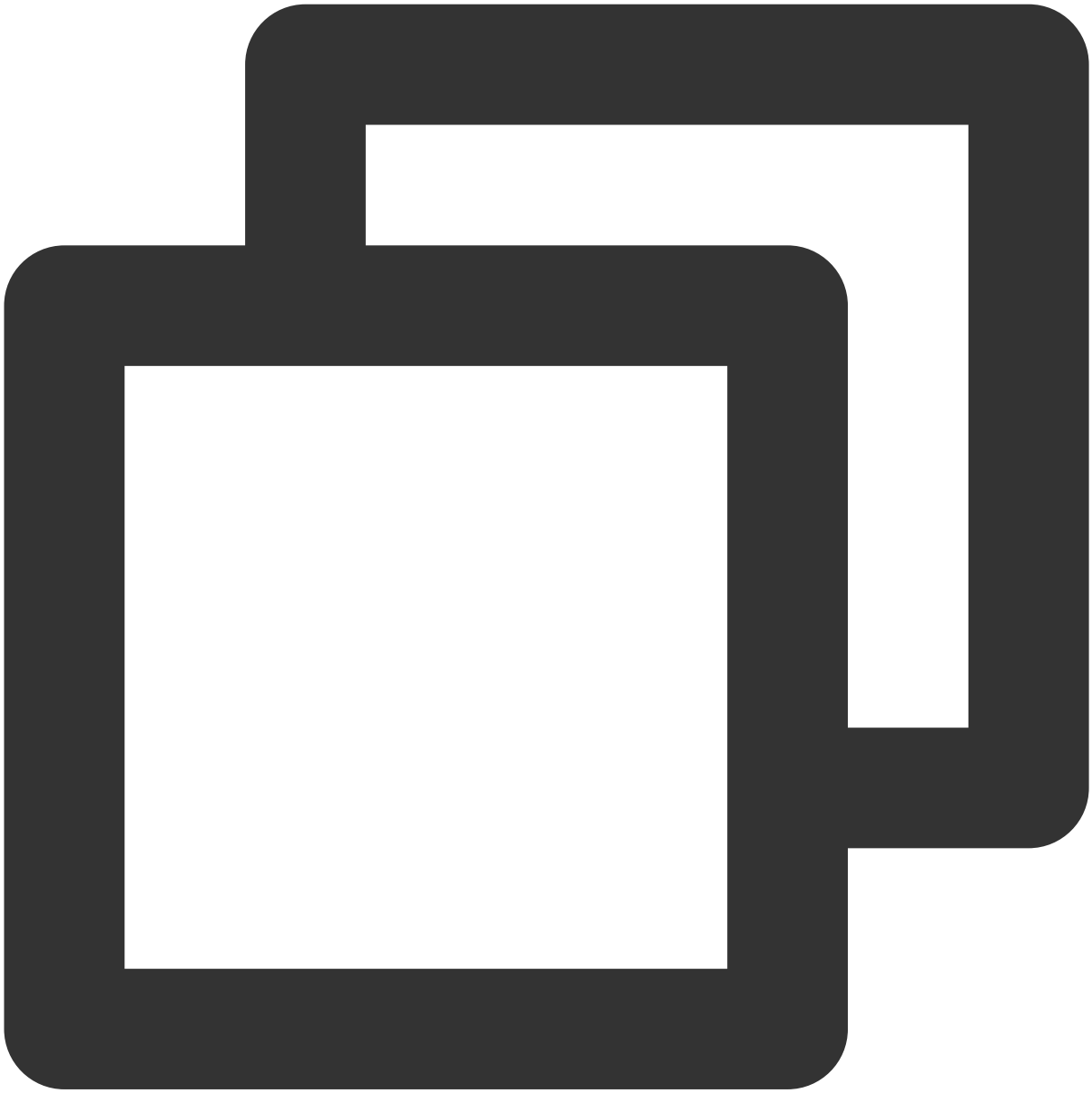
实际案例

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
提交媒体处理任务
提交视频转动图任务

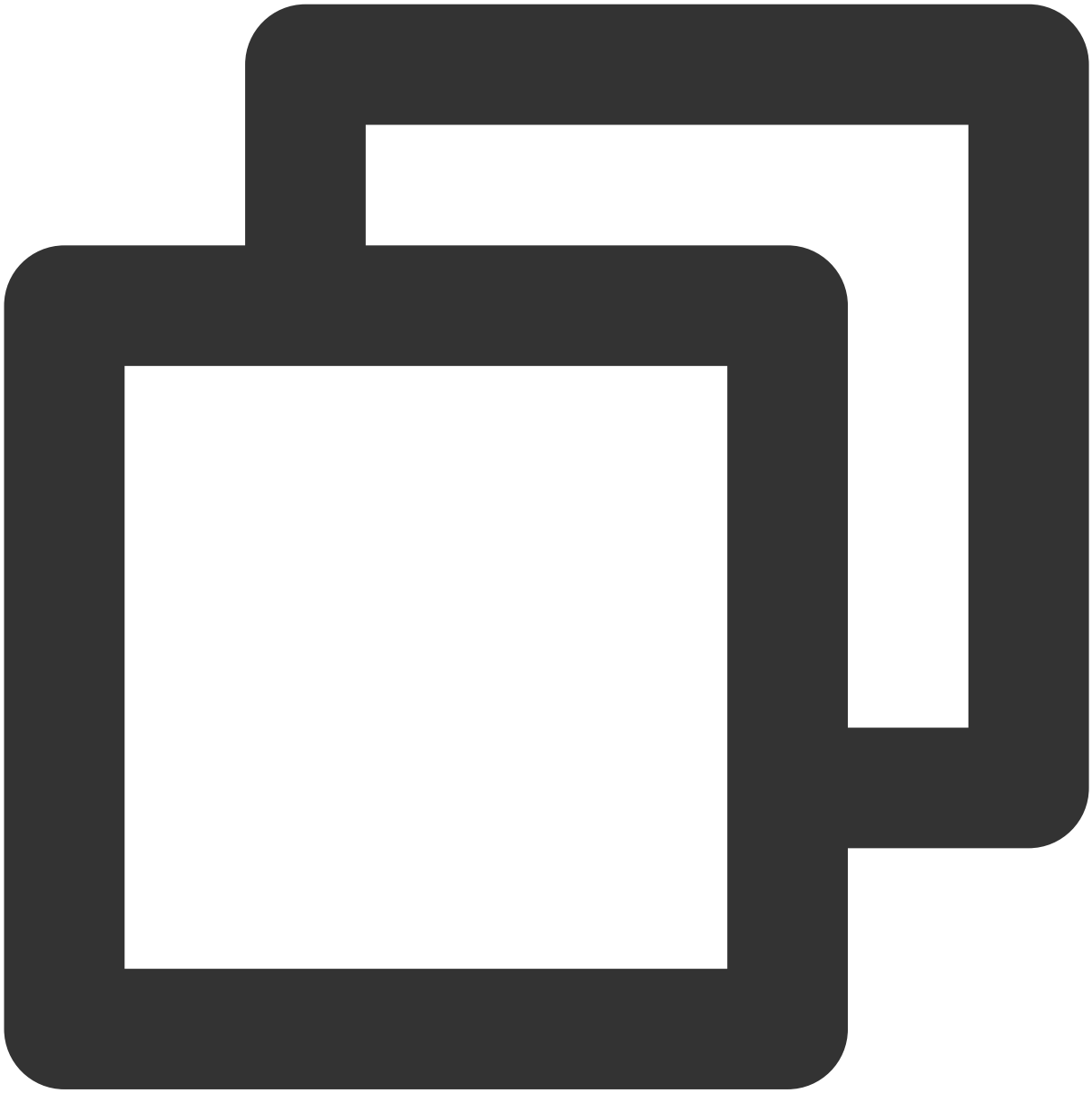
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个动图任务。

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：

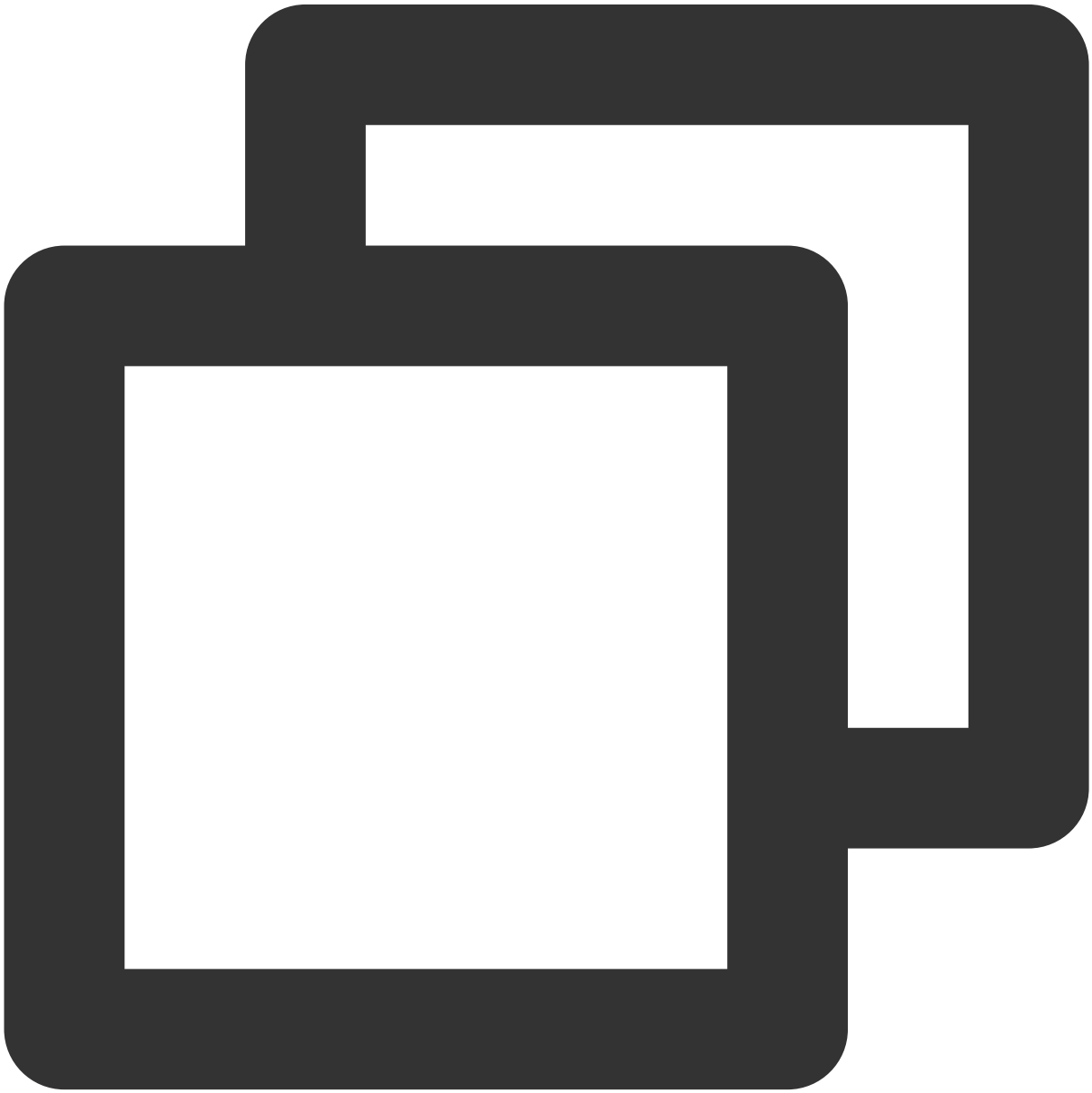
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Animation</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.gif</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：Animation	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
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
Animation	Request.Operation	指定该任务的参数	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
Output	Request.Operation	结果输出地址	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

注意：

优先使用 TemplateId，无 TemplateId 时使用 Animation。

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
Container	Request.Operation.Animation	同动图模板 CreateMediaTemplate 接口中的 Request.Container	Container	否
Video	Request.Operation.Animation	同动图模板 CreateMediaTemplate 接口中的 Request.Video	Container	否
TimeInterval	Request.Operation.Animation		Container	否

版权所有：腾讯云计算（北京）有限责任公司

		同动图模板 CreateMediaTemplate 接口中的 Request.TimeInterval		

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

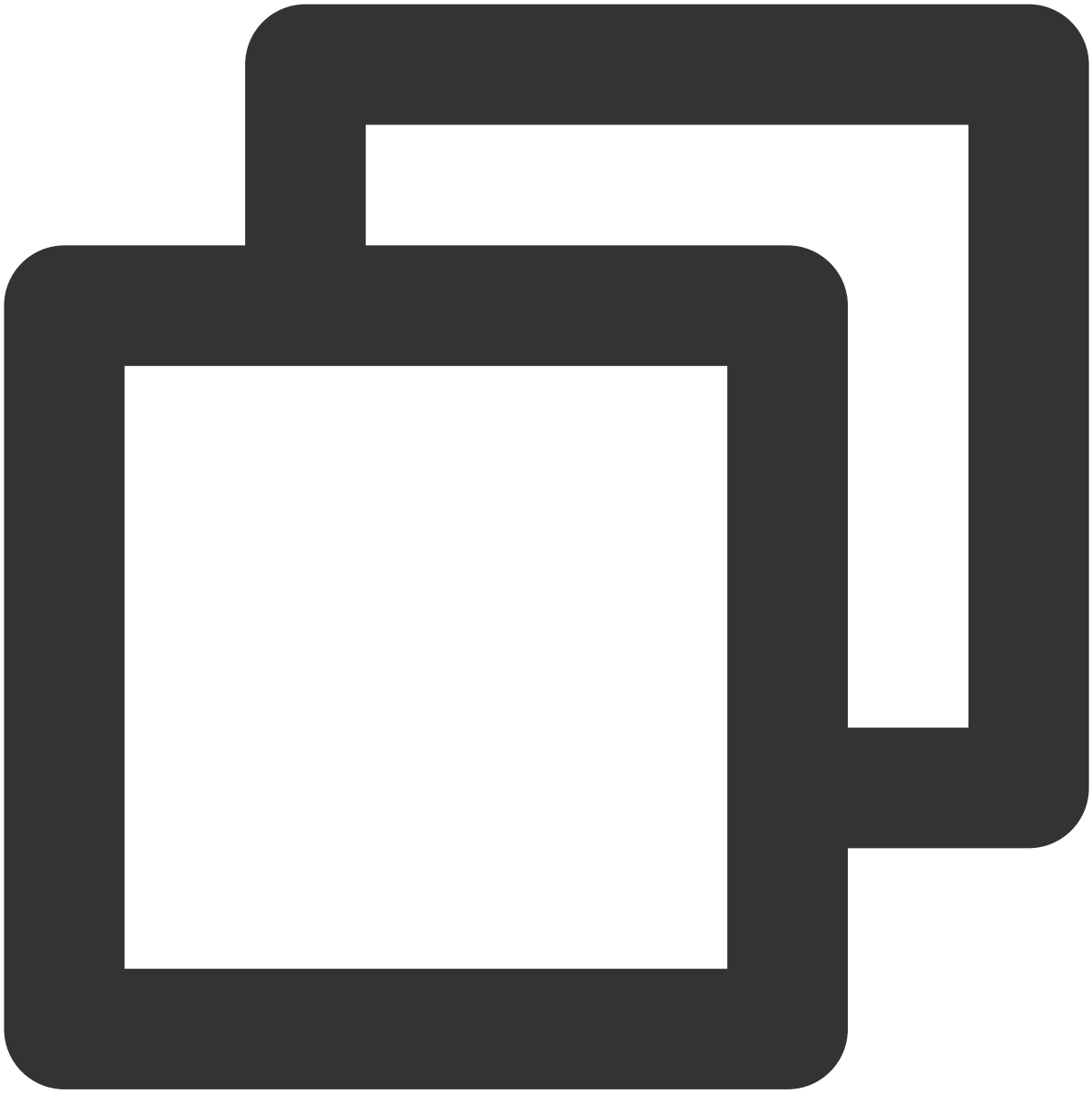
响应

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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Animation</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>animation_demo</TemplateName>
            <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>
```

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：Animation	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
Animation	Response.JobsDetail.Operation	同请求中的 Request.Operation.Animation	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

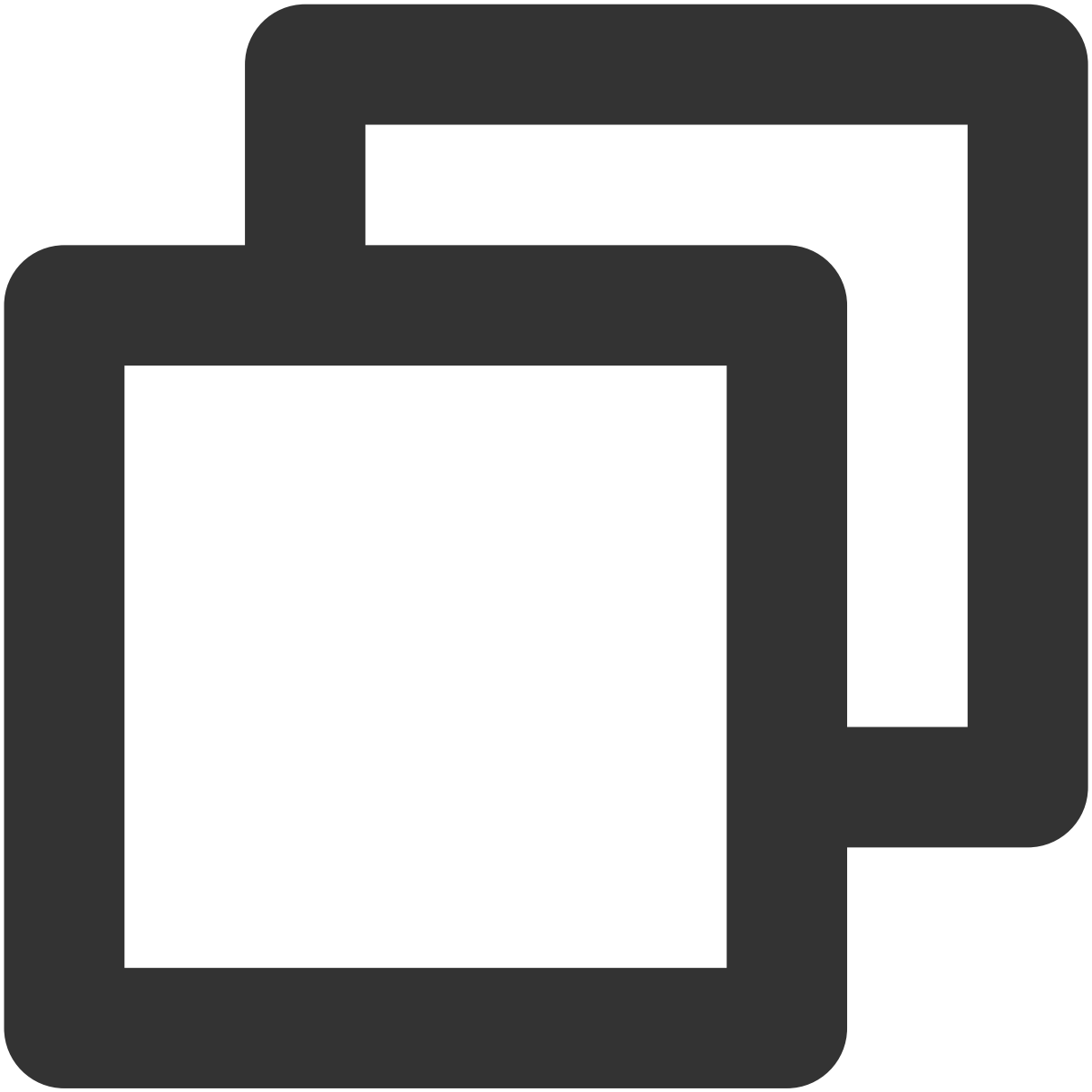
错误码

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

使用动图模板 ID

请求

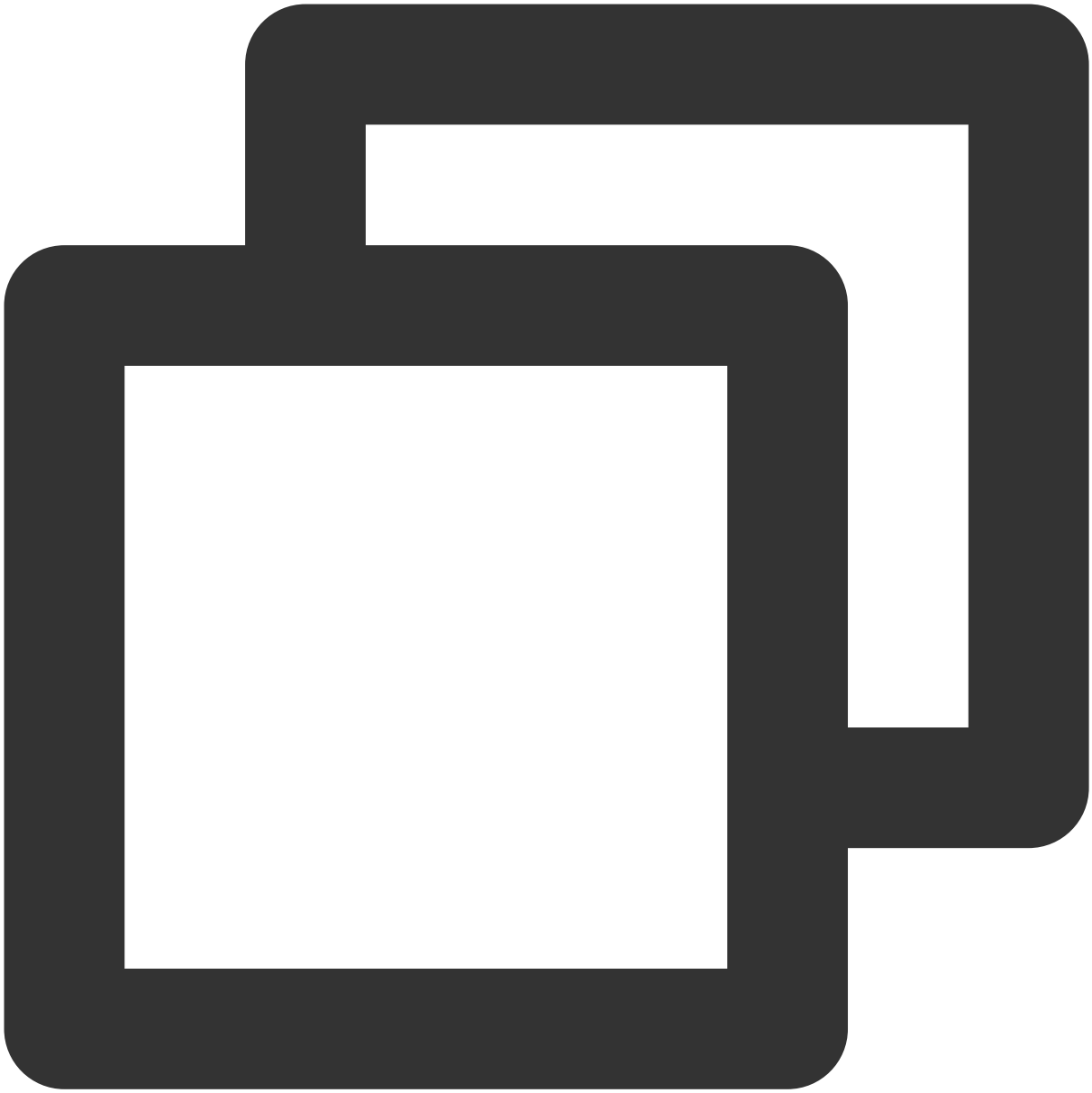


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Animation</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.gif</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



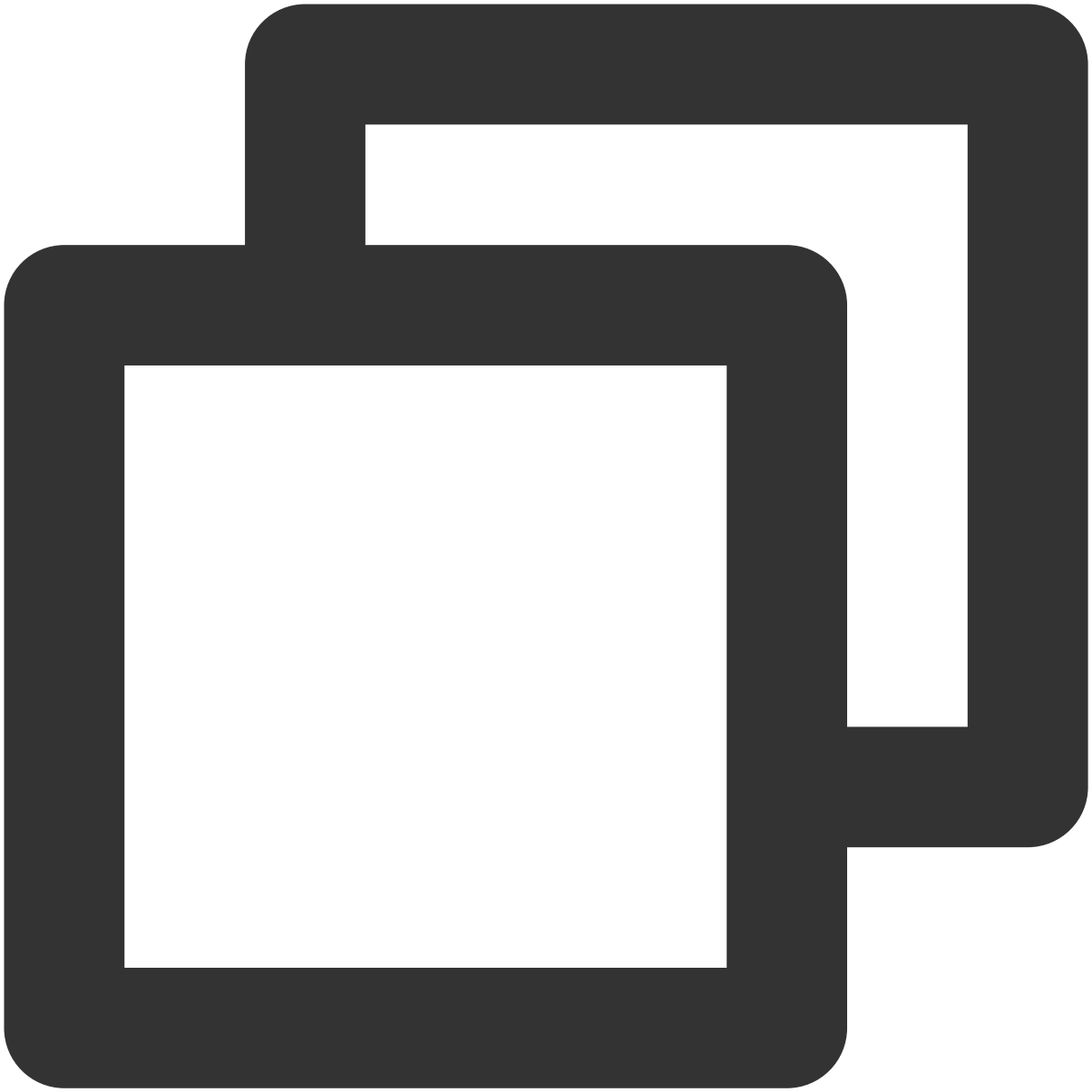
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Animation</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
            <TemplateName>animation_demo</TemplateName>
            <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>
```

使用动图处理参数

请求

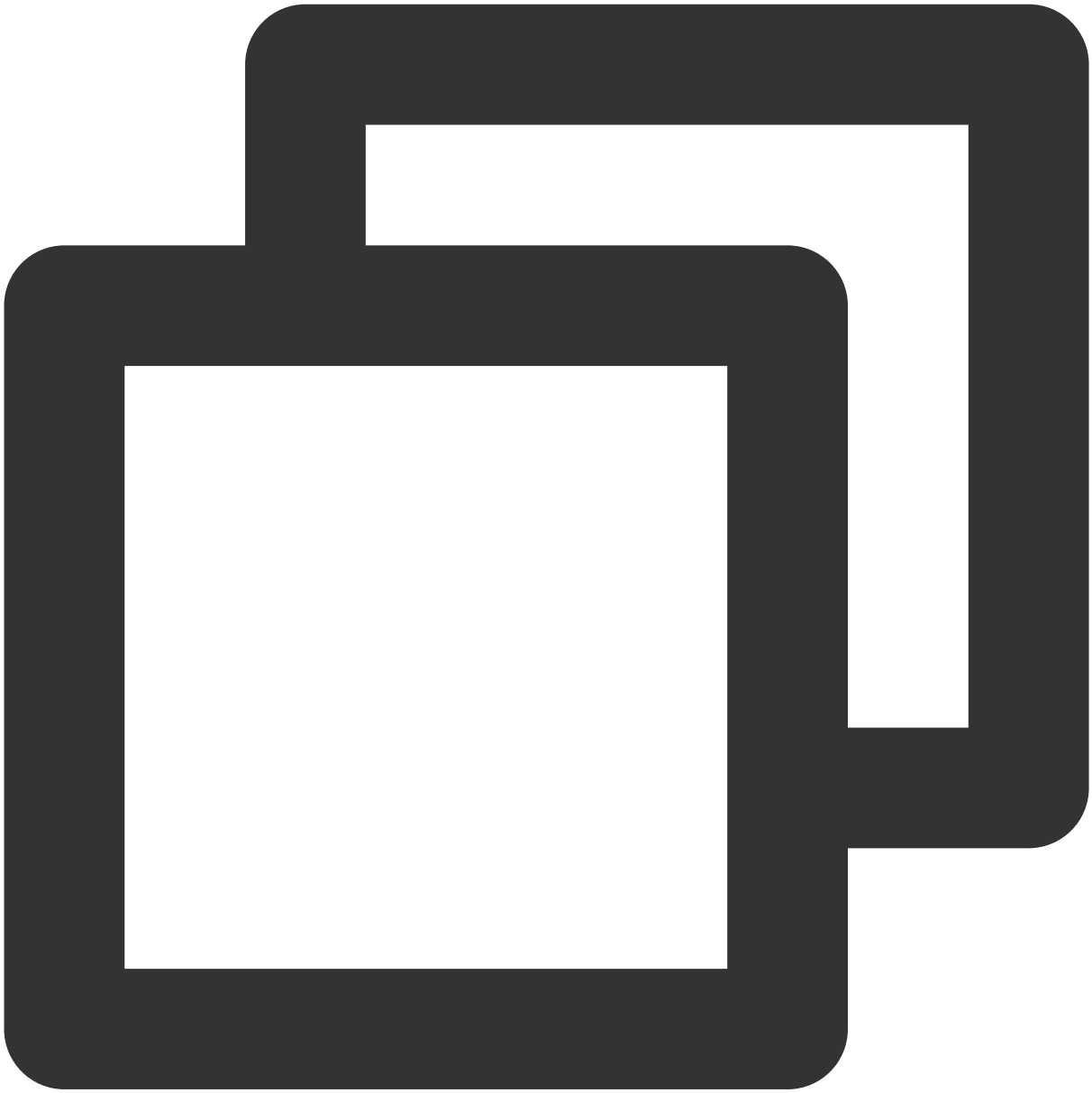


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Animation</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <Animation>
            <Container>
                <Format>gif</Format>
            </Container>
            <Video>
                <Codec>gif</Codec>
                <Width>1280</Width>
                <Height>960</Height>
                <Fps>15</Fps>
                <AnimateOnlyKeepKeyFrame>true</AnimateOnlyKeepKeyFrame>
            </Video>
            <TimeInterval>
                <Start>0</Start>
                <Duration>60</Duration>
            </TimeInterval>
        </Animation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.gif</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:12+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Animation</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Animation>
                <Container>
                    <Format>gif</Format>
                </Container>
                <Video>
                    <Codec>gif</Codec>
                    <Width>1280</Width>
                    <Height>960</Height>
                    <Fps>15</Fps>
                    <AnimateOnlyKeepKeyFrame>true</AnimateOnlyKeepKeyFrame>
                </Video>
                <TimeInterval>
                    <Start>0</Start>
                    <Duration>60</Duration>
                </TimeInterval>
            </Animation>
            <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>
```

提交视频拼接任务

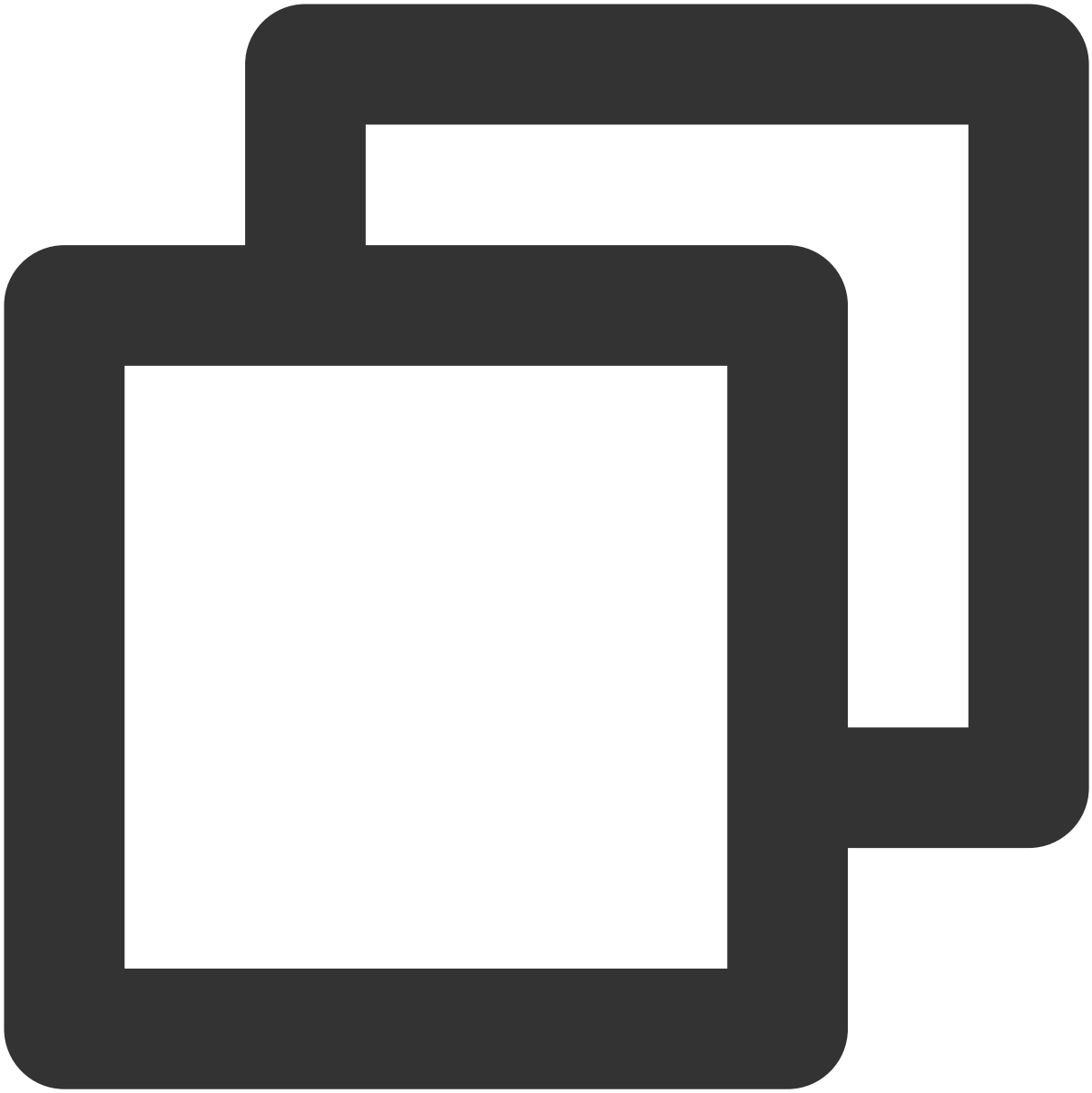
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个拼接任务。

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：

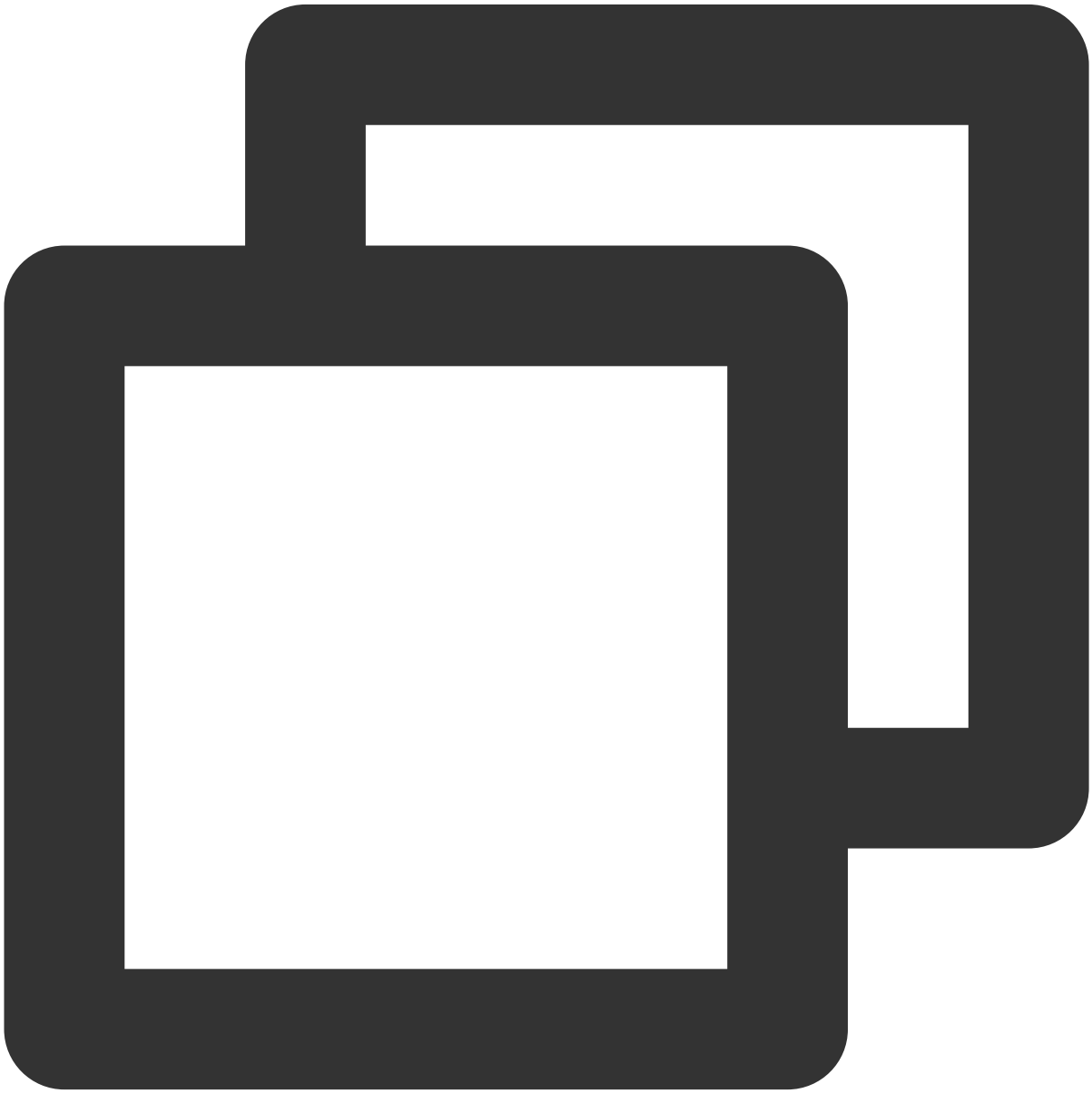
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Concat</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <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>
    <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：Concat	String	是
Input	Request	待操作的媒体信息	Container	否
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，优先级高于队列的回调类型	String	否
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
ConcatTemplate	Request.Operation	指定拼接参数	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
Output	Request.Operation	结果输出地址	Container	是
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

注意：

优先使用 TemplateId，无 TemplateId 时使用 ConcatTemplate。

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
ConcatFragment	Request.Operation.ConcatTemplate	拼接节点	Container 数组
Audio	Request.Operation.ConcatTemplate	音频参数，同创建拼接模板 CreateMediaTemplate 接口中的 Request.ConcatTemplate.Audio	Container
Video	Request.Operation.ConcatTemplate	视频参数，同创建拼接模板 CreateMediaTemplate 接口中的	Container

版权所有：腾讯云计算（北京）有限责任公司

		Request.ConcatTemplate.Video	
Container	Request.Operation.ConcatTemplate	封装格式，同创建拼接模板 CreateMediaTemplate 接口中的 Request.ConcatTemplate.Container	Container
AudioMix	Request.Operation.ConcatTemplate	混音参数, 详情见 AudioMix	Container 数组
Index	Request.Operation.ConcatTemplate	Input 节点位于 ConcatFragment 序列索引	String
DirectConcat	Request.Operation.ConcatTemplate	简单拼接方式（不转码直接拼接），其他的视频和音频参数失效	String

Container 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	是否必选	默认值	限制
Url	Request.Operation.ConcatTemplate.ConcatFragment	拼接对象地址	String	是	无	同 bucket 对象文件
FragmentIndex	Request.Operation.ConcatTemplate.ConcatFragment	拼接对象的索引位置	String	否	0	大于等于0的整数
StartTime	Request.Operation.ConcatTemplate.ConcatFragment	开始	String	否	视频	[0 视频时长]

		时间			开始	单位为秒
EndTime	Request.Operation.ConcatTemplate.ConcatFragment	结束时间	String	否	视频结束	[0 视频时长] 单位为秒

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

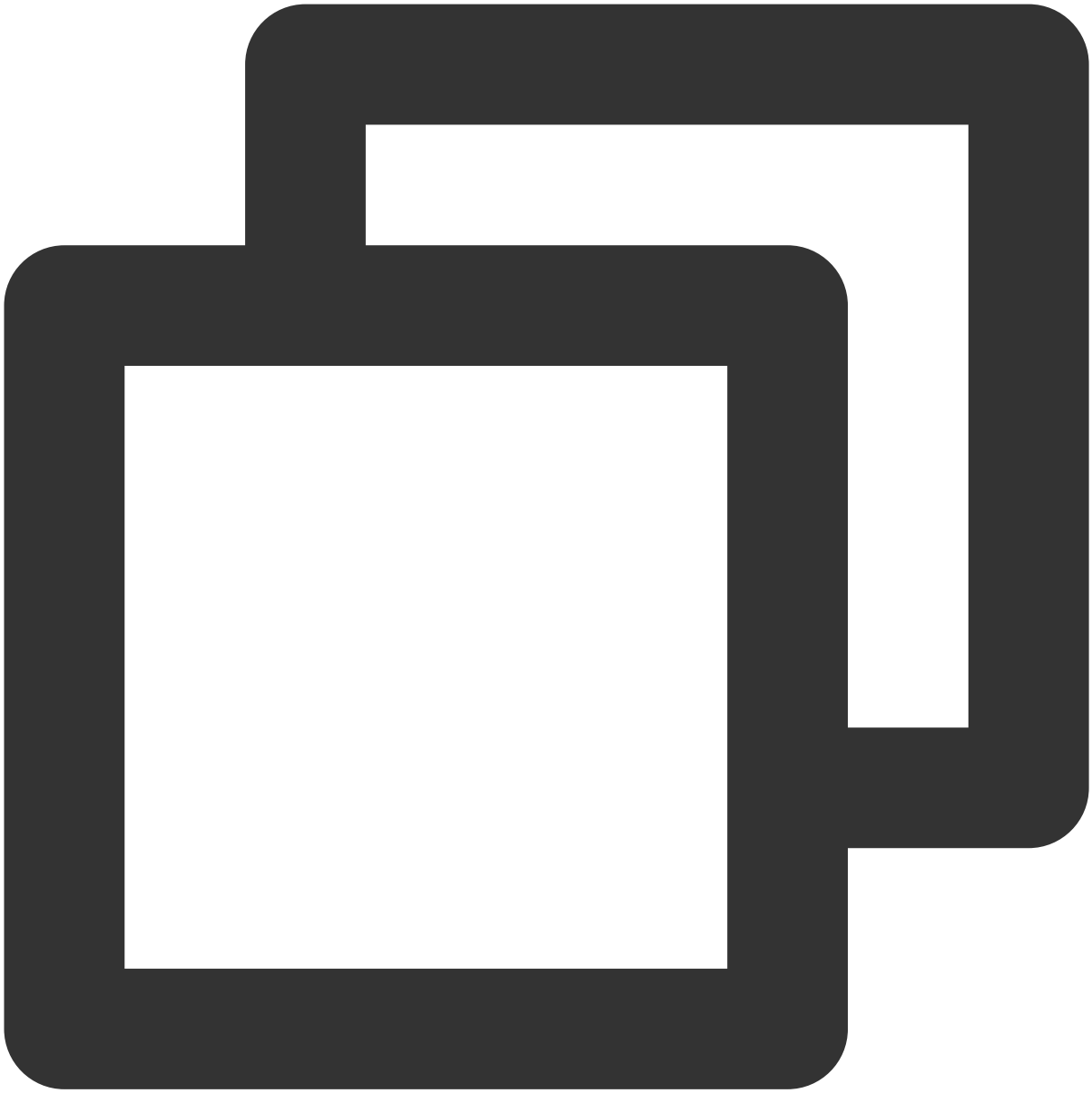
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Concat</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>concat_demo</TemplateName>
            <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>
```

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：Concat	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
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
ConcatTemplate	Response.JobsDetail.Operation	同请求中的 Request.Operation.ConcatTemplate	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

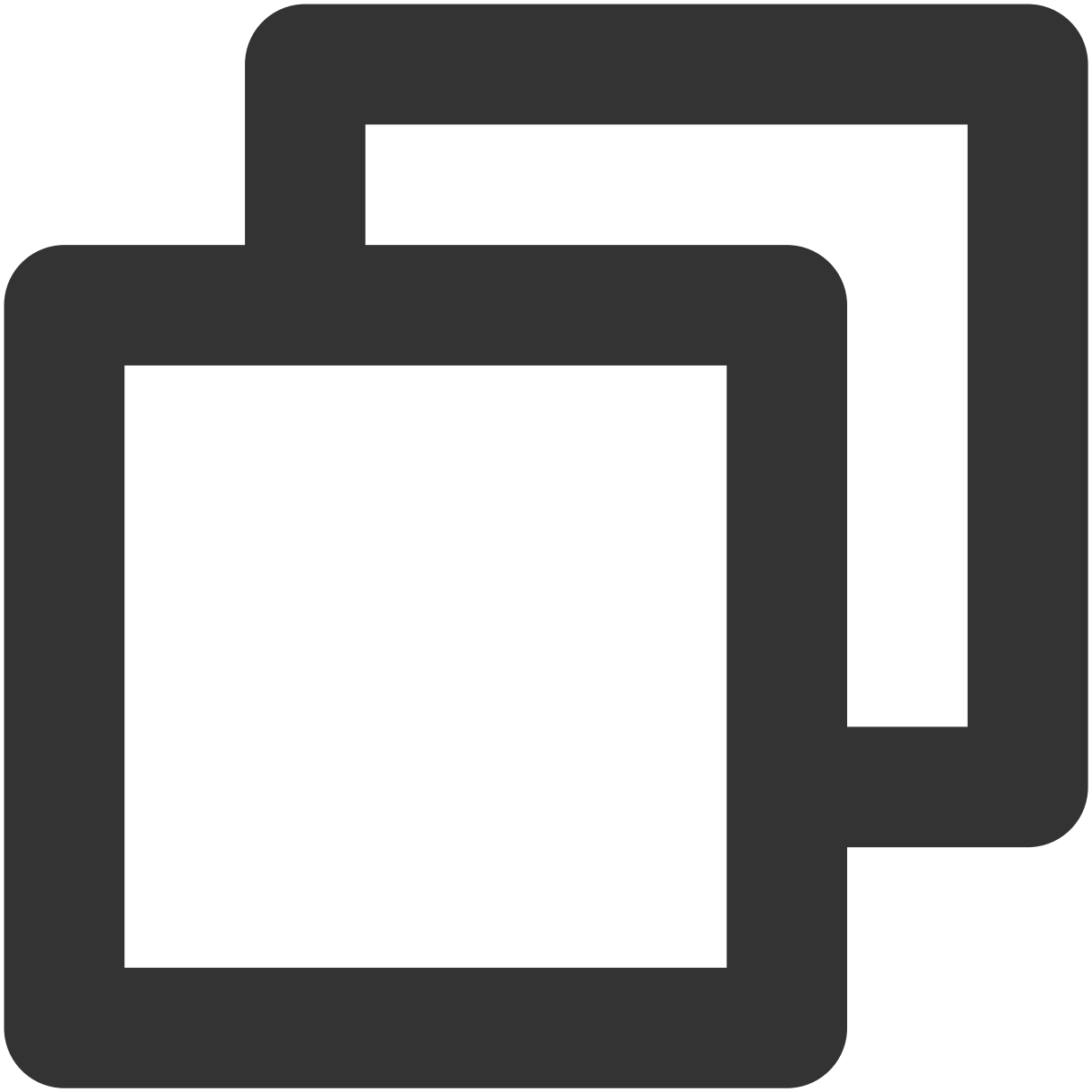
错误码

常见的错误信息请参见 错误码 文档。

实际案例

请求1：使用拼接模板 ID

请求

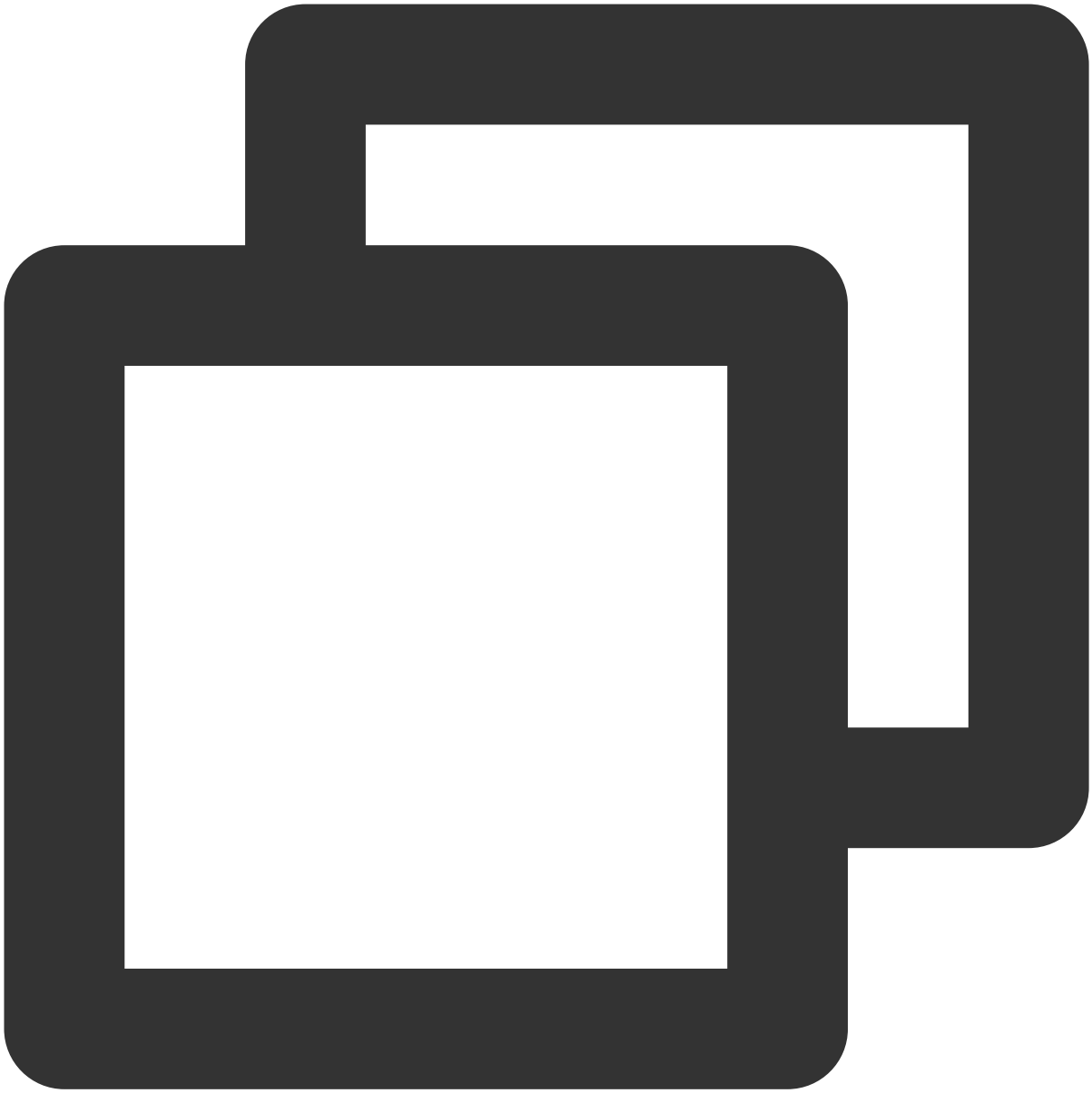


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:test-123456789.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Concat</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <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>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



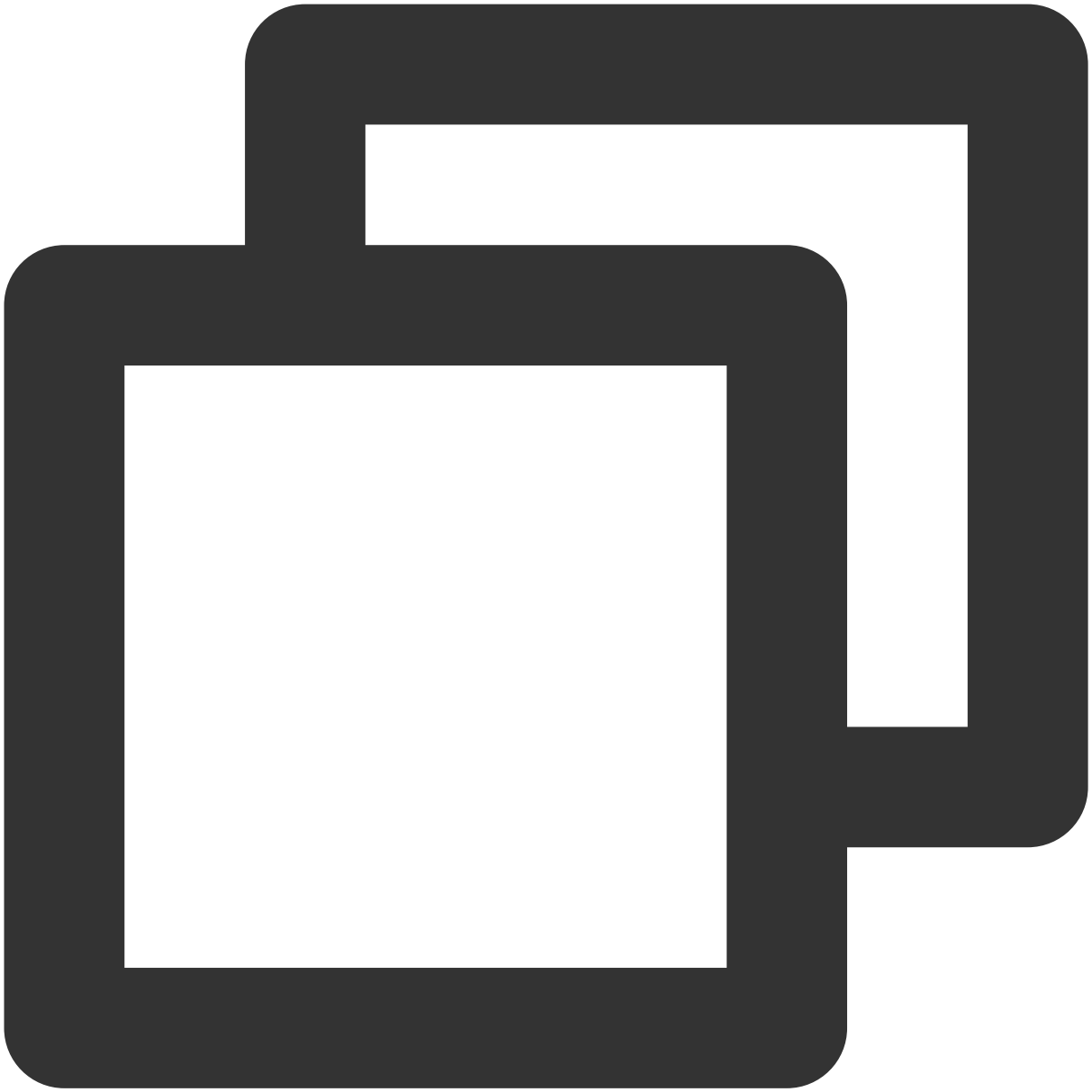
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Concat</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>concat_demo</TemplateName>
            <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：使用拼接参数

请求

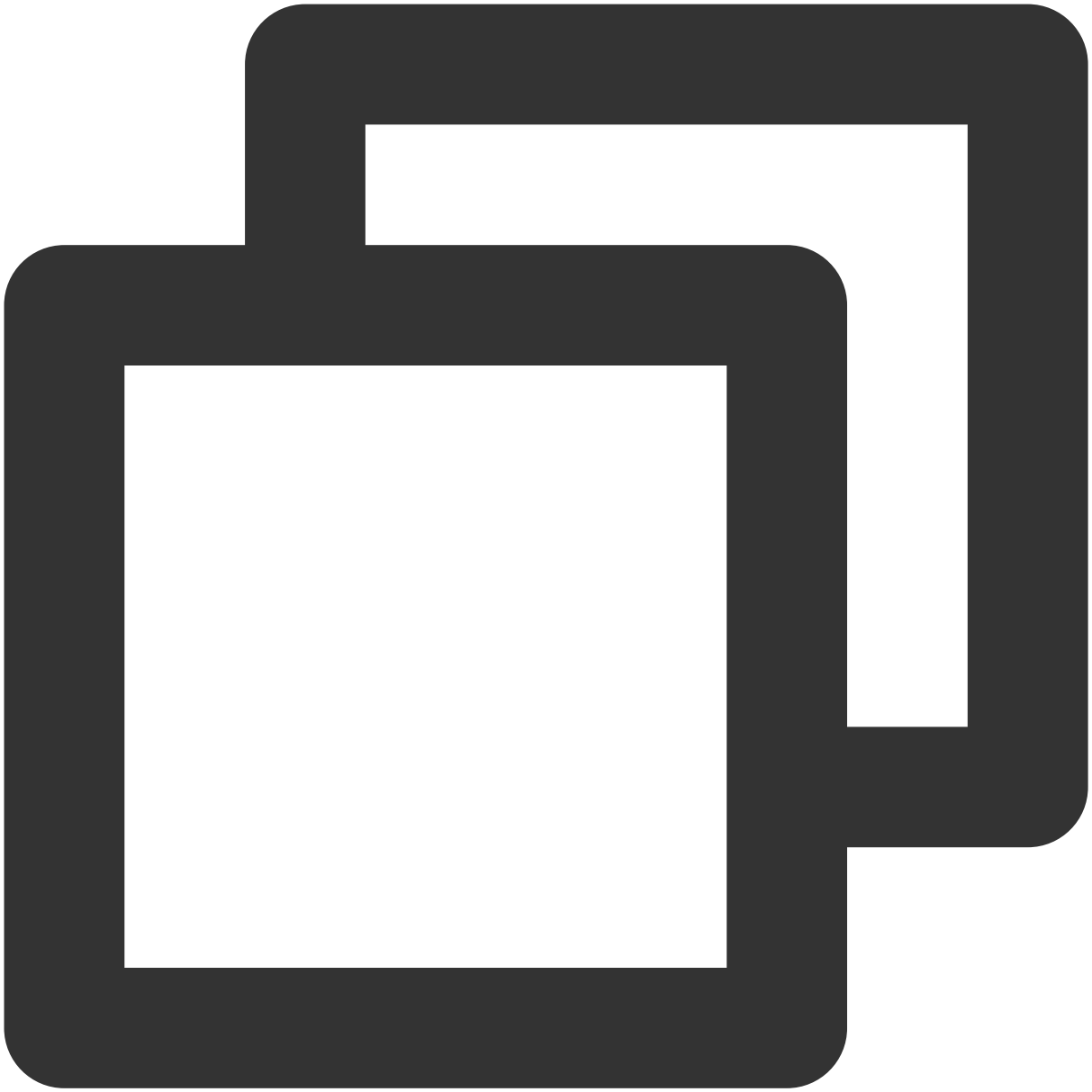


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:test-123456789.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Concat</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <ConcatTemplate>
            <ConcatFragment>
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/start.mp4<
            </ConcatFragment>
            <ConcatFragment>
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp4</U
            </ConcatFragment>
            <Audio>
                <Codec>mp3</Codec>
            </Audio>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Height>720</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>
        <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>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



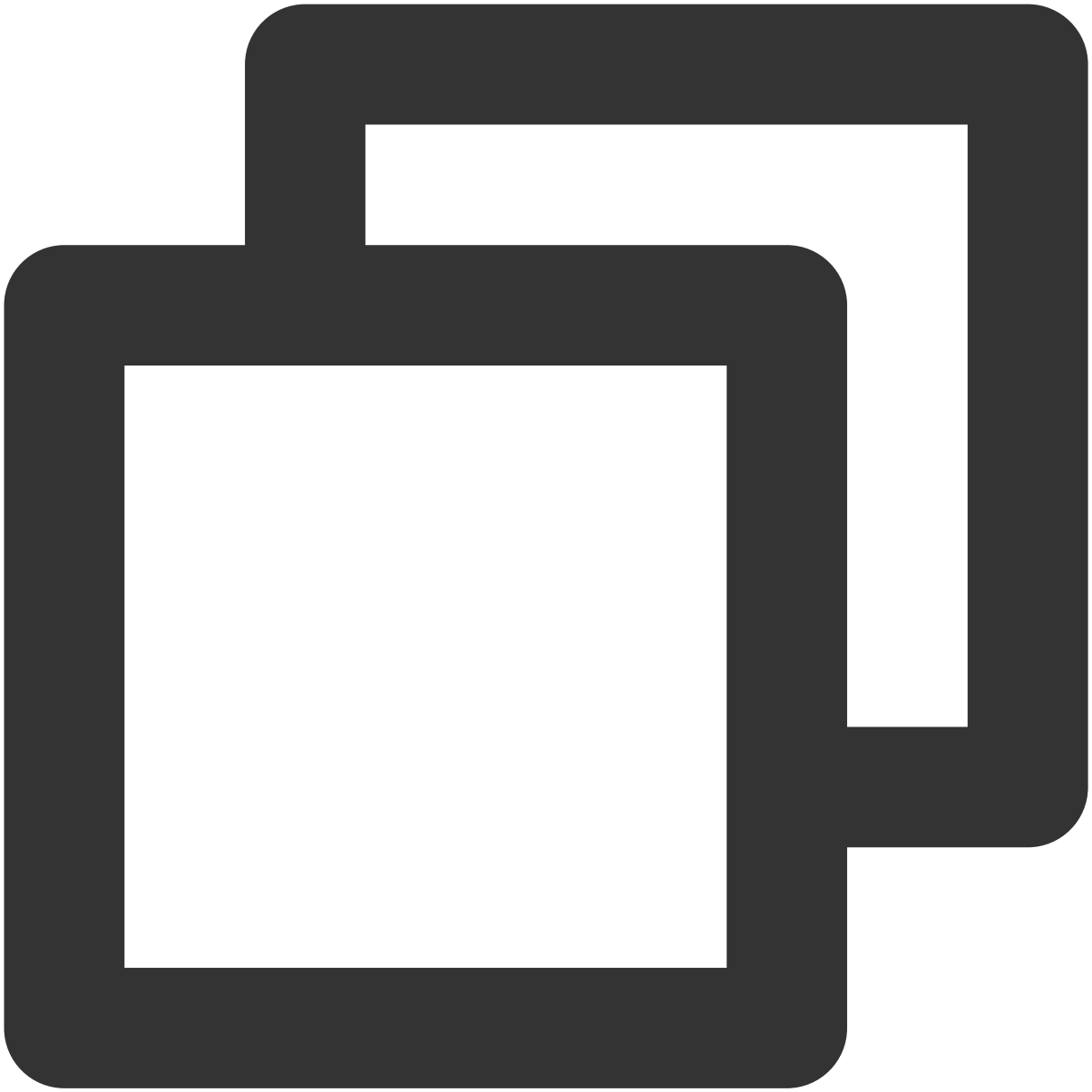
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=
```

```
<Response>
    <JobsDetail>
        <Code>Success</Code>
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c5d</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Concat</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <ConcatTemplate>
                <ConcatFragment>
                    <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/start.
                </ConcatFragment>
                <ConcatFragment>
                    <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp
                </ConcatFragment>
                <Audio>
                    <Codec>mp3</Codec>
                </Audio>
                <Video>
                    <Codec>H.264</Codec>
                    <Bitrate>1000</Bitrate>
                    <Width>1280</Width>
                    <Height>720</Height>
                    <Fps>30</Fps>
                </Video>
                <Container>
                    <Format>mp4</Format>
                </Container>
                <AudioMix>
                    <AudioSource>https://test-123456789.cos.ap-chongqing.myqcloud.c
                    <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>
                <Index>1</Index>
            </ConcatTemplate>
            <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3：多文件拼接

请求

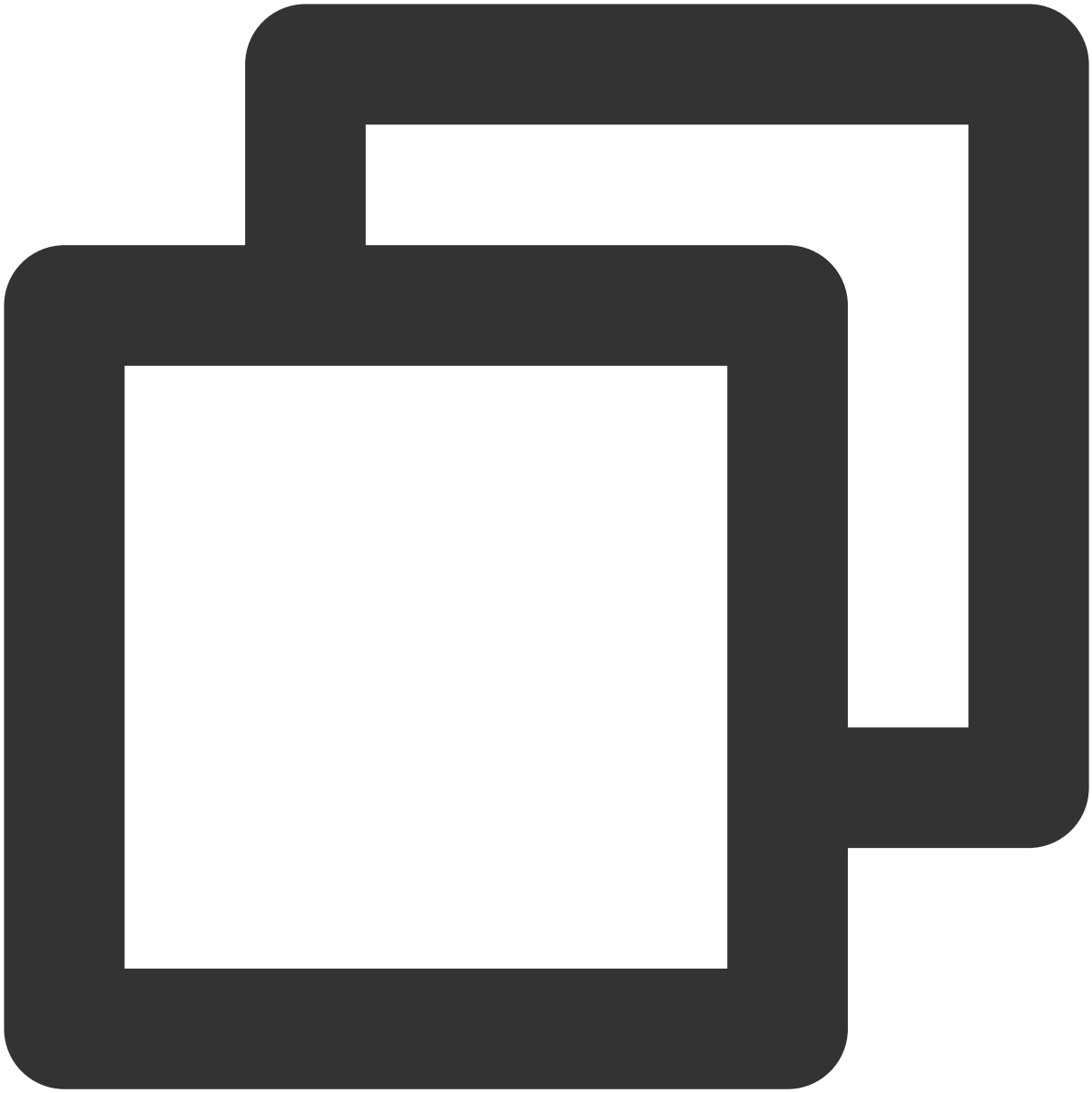


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:test-123456789.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Concat</Tag>
    <Operation>
        <ConcatTemplate>
            <ConcatFragment>
```

```
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/start.mp4<
                <FragmentIndex>0</FragmentIndex>
                <StartTime>0</StartTime>
                <EndTime>6</EndTime>
            </ConcatFragment>
            <ConcatFragment>
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/middle.mp4
                <FragmentIndex>1</FragmentIndex>
            </ConcatFragment>
            <ConcatFragment>
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp4</U
                <FragmentIndex>2</FragmentIndex>
                <StartTime>5</StartTime>
                <EndTime>10</EndTime>
            </ConcatFragment>
            <Container>
                <Format>mp4</Format>
            </Container>
        </ConcatTemplate>
        <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>
    <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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c5d</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Concat</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <ConcatTemplate>
                <ConcatFragment>
                    <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/start.
                    <StartTime>0</StartTime>
                    <EndTime>6</EndTime>
                </ConcatFragment>
                <ConcatFragment>
                    <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/middle
                </ConcatFragment>
                <ConcatFragment>
                    <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp
                    <StartTime>5</StartTime>
                    <EndTime>10</EndTime>
                </ConcatFragment>
                <Container>
                    <Format>mp4</Format>
                </Container>
            </ConcatTemplate>
            <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>
```

提交添加数字水印任务

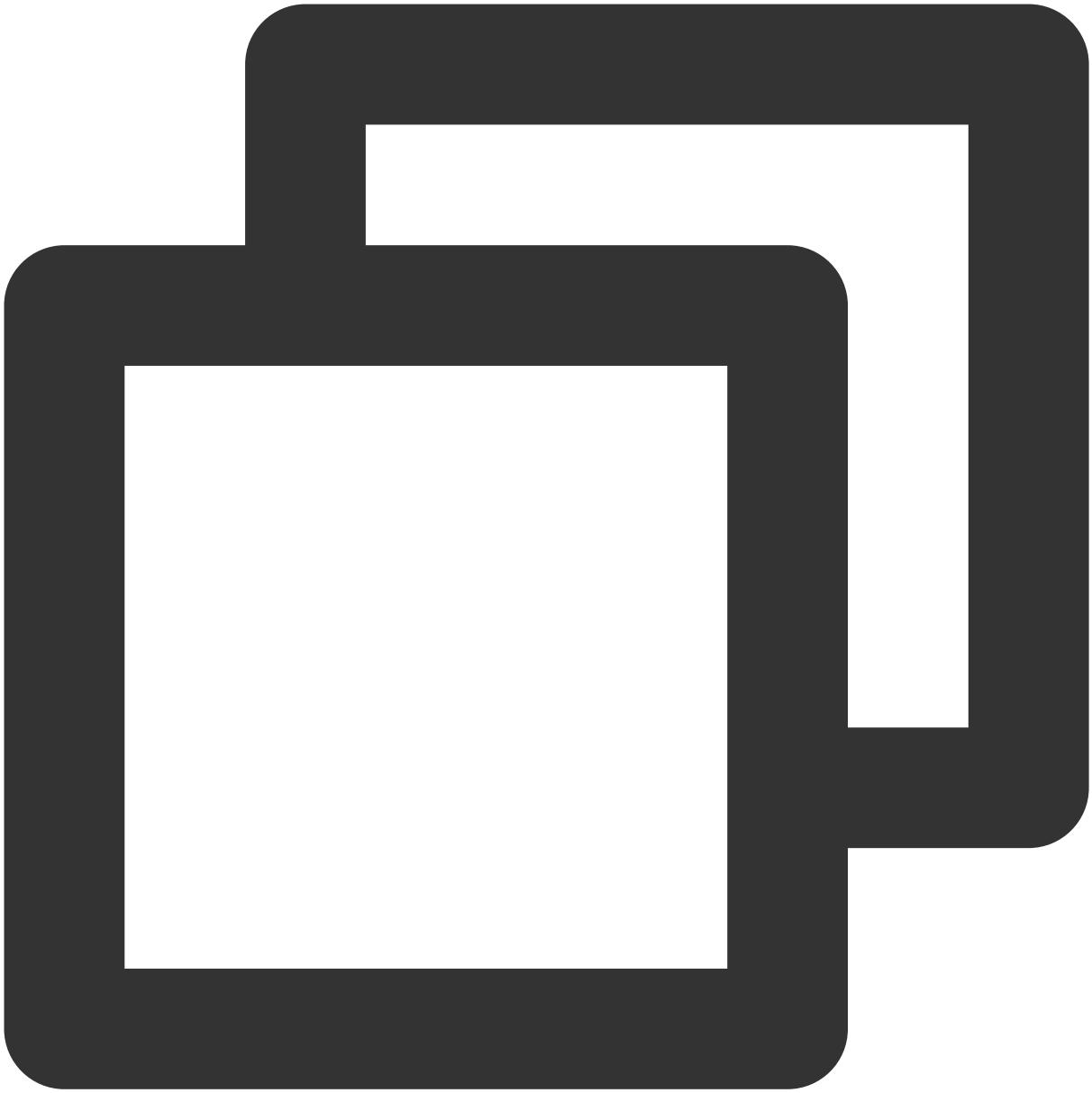
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个添加数字水印任务。

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：

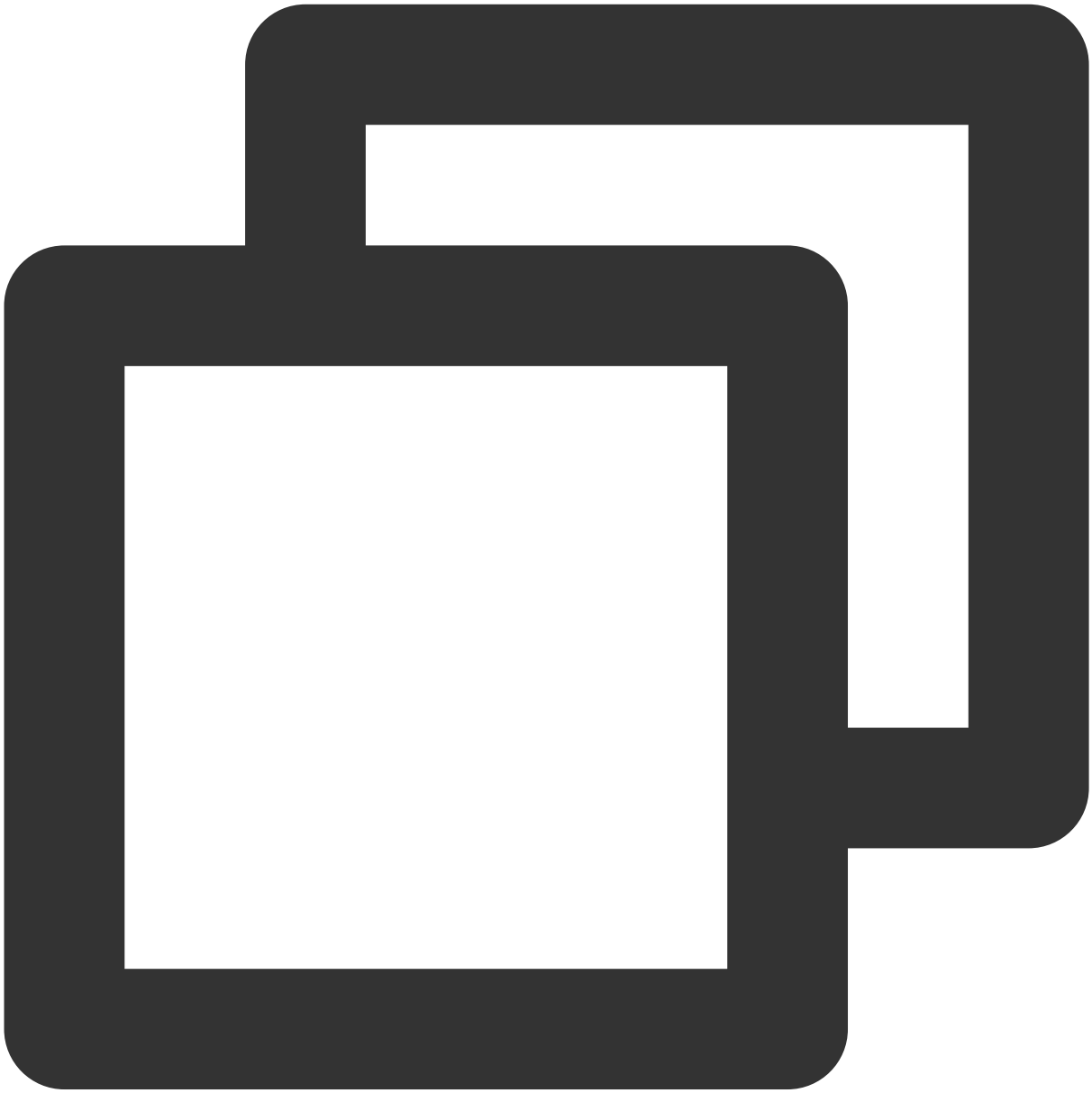
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>DigitalWatermark</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
        </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>
    <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：DigitalWatermark	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
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

版权所有：腾讯云计算（北京）有限责任公司

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
Object	Request.Input	媒体文件名	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
Output	Request.Operation	结果输出地址	Container	是
DigitalWatermark	Request.Operation	数字水印配置	Container	是
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
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 DigitalWatermark 的具体数据类型描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Message	Request.Operation.DigitalWatermark	数字水印嵌入的字符串信息	string	是	长度不超过64个字符，仅支持中文、英文、数字、_、-和*
Type	Request.Operation.DigitalWatermark	水印类型	String	是	Text
Version	Request.Operation.DigitalWatermark	水印版本	String	是	V1

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

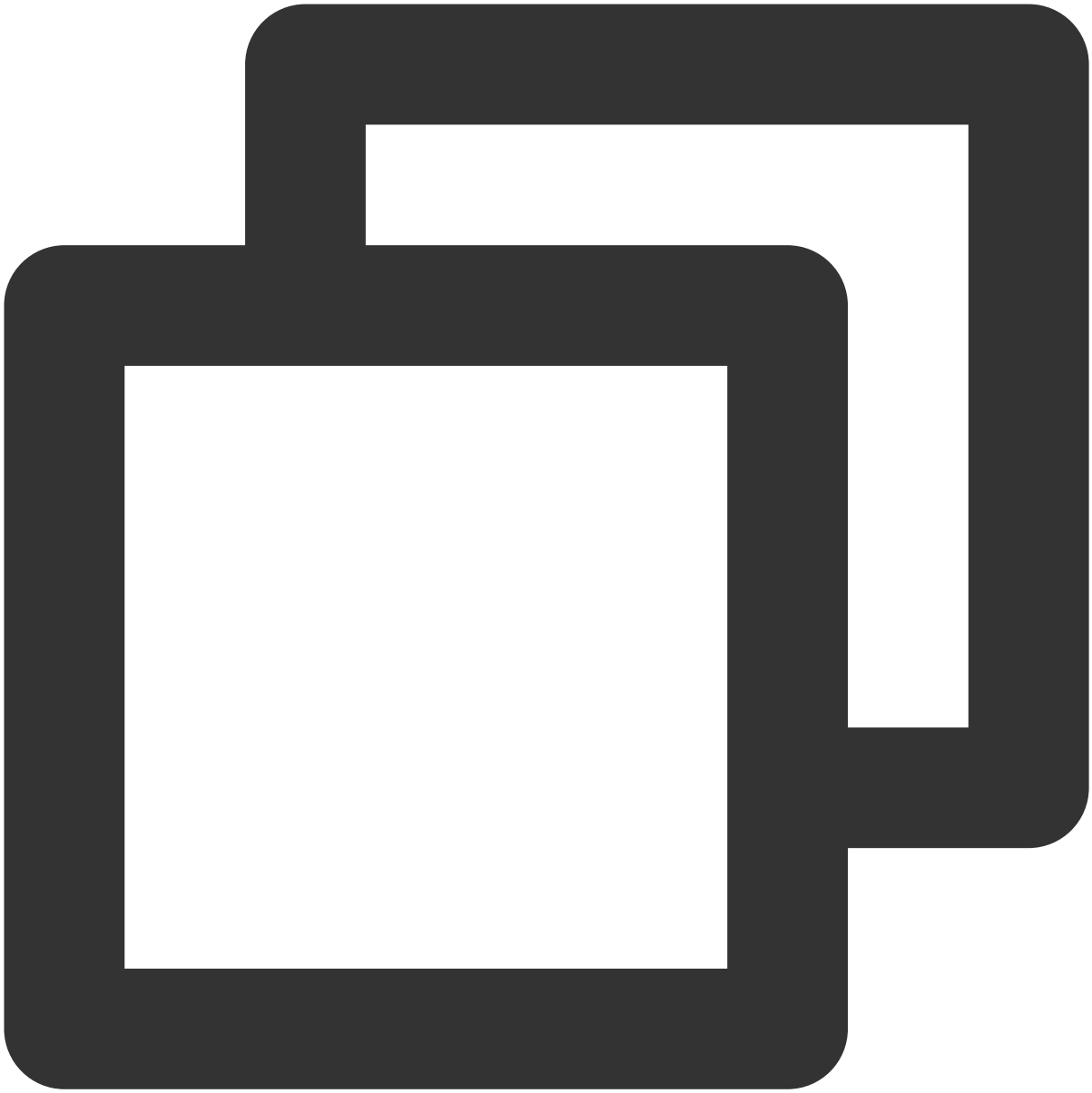
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>DigitalWatermark</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
            </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>
```

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：DigitalWatermark	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、	String

		Success、Failed、Pause、Cancel 其中一个	
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	该任务的操作规则	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
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
DigitalWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.DigitalWatermark	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

Container 节点 MediaResult 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

OutputFile	Response.Operation.MediaResult	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

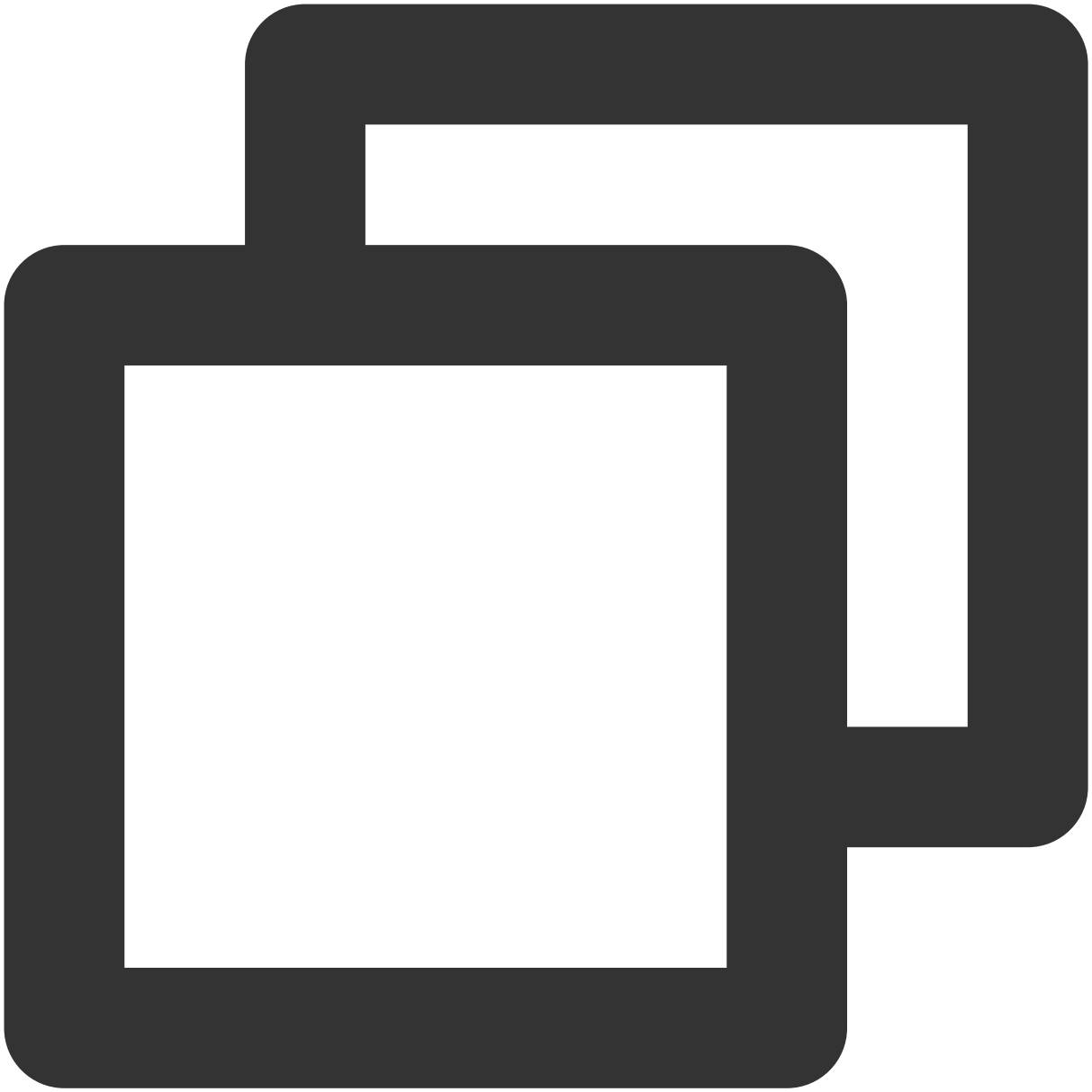
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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

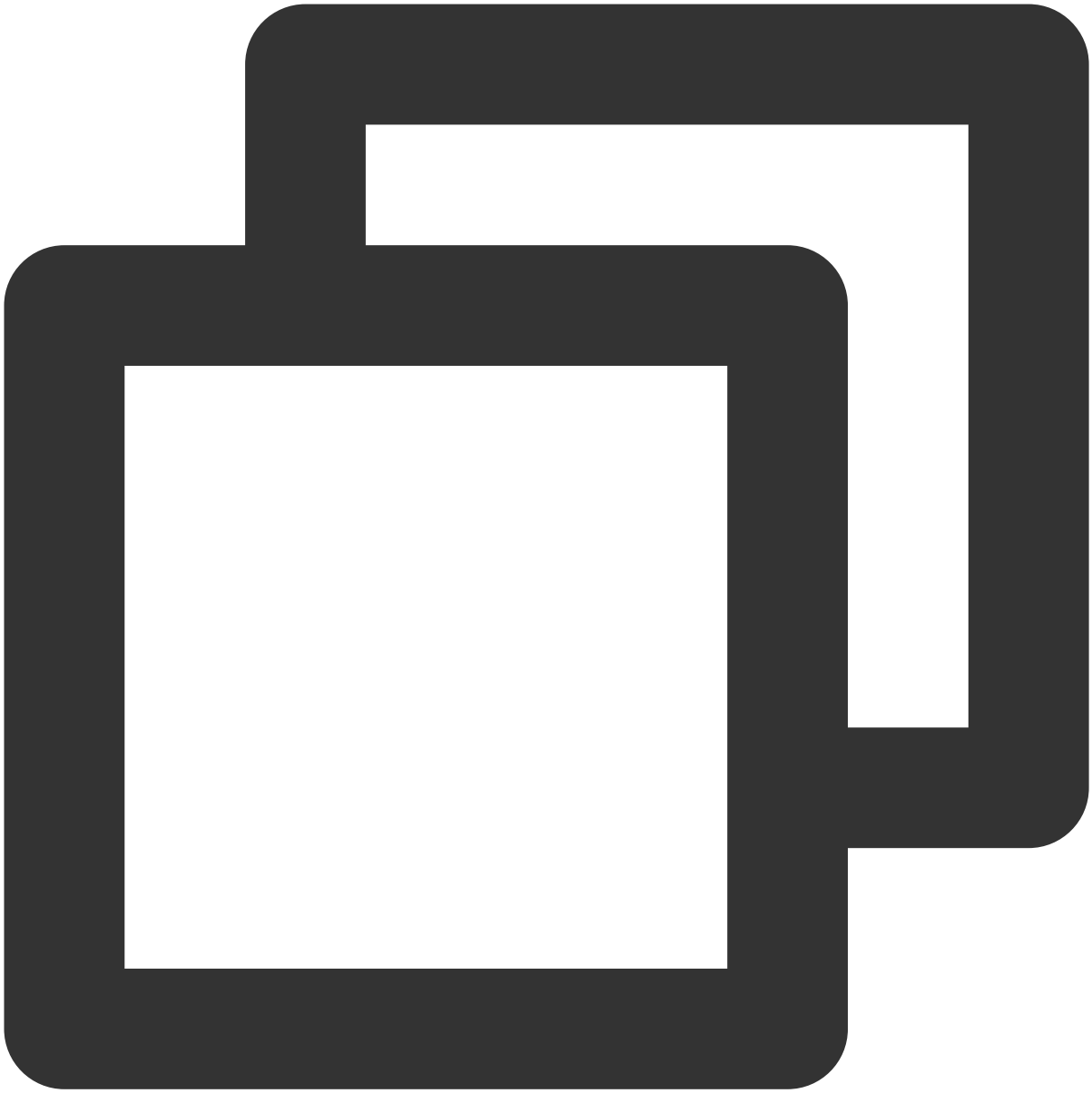


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>DigitalWatermark</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>DigitalWatermark</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
            </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>
```

提交提取数字水印任务

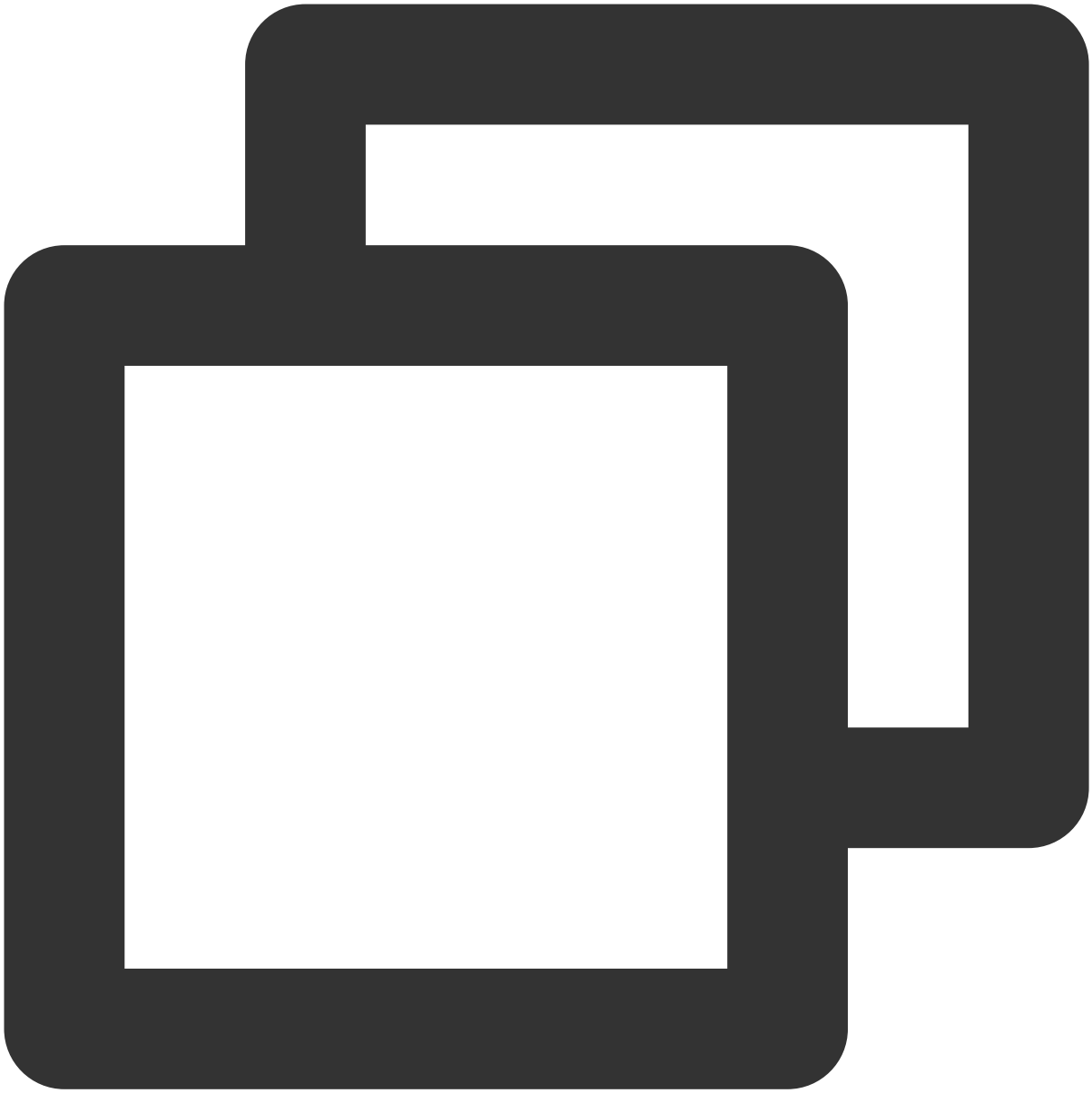
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个提取数字水印任务。

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：

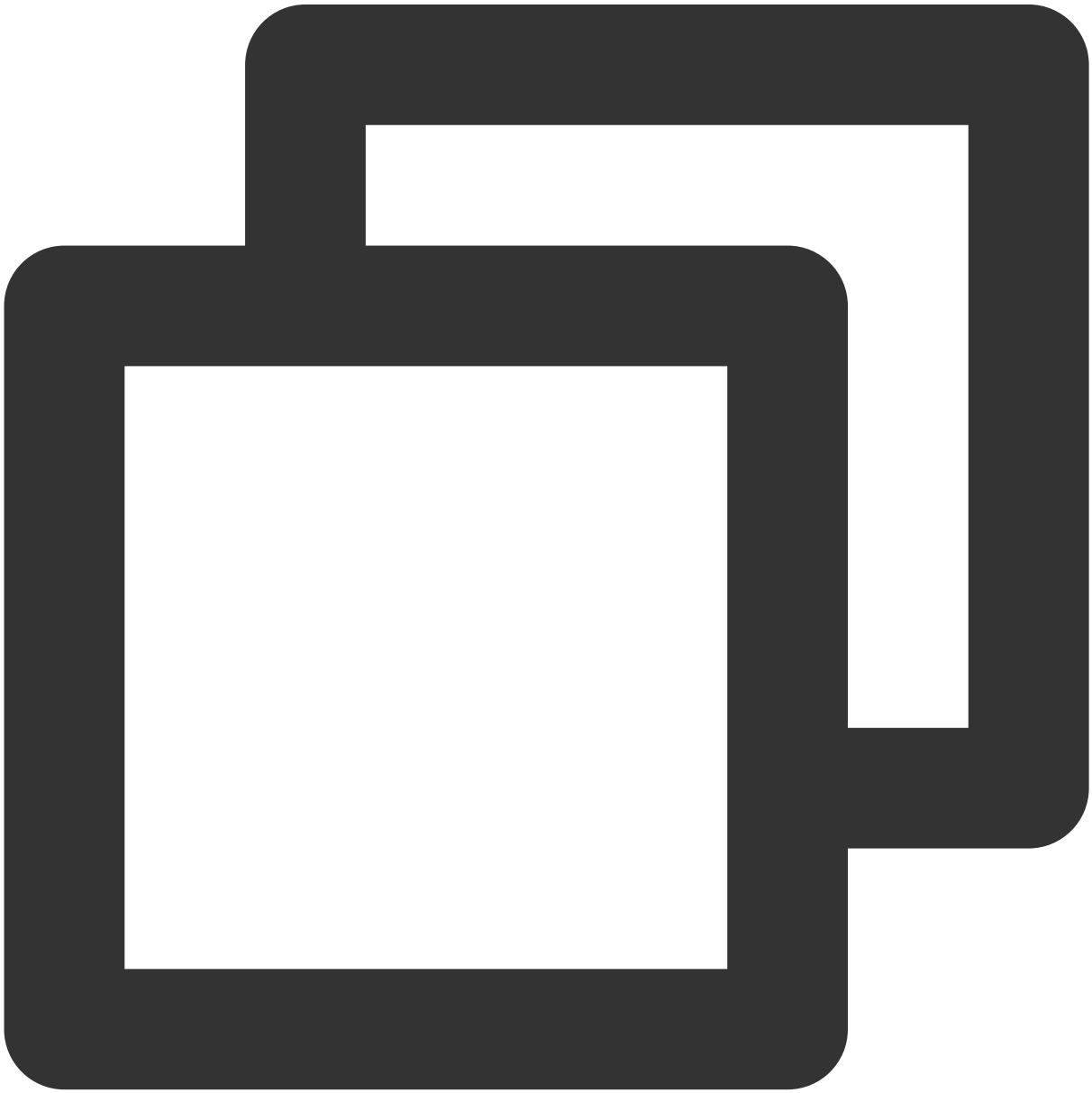
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>ExtractDigitalWatermark</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <ExtractDigitalWatermark>
            <Type>Text</Type>
            <Version>V1</Version>
        </ExtractDigitalWatermark>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：ExtractDigitalWatermark	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
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

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	媒体文件名	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
ExtractDigitalWatermark	Request.Operation	提取数字水印配置	Container	是
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
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 ExtractDigitalWatermark 的具体数据类型描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.Operation.ExtractDigitalWatermark	水印类型	String	是	Text
Version	Request.Operation.ExtractDigitalWatermark	水印版本	String	是	V1

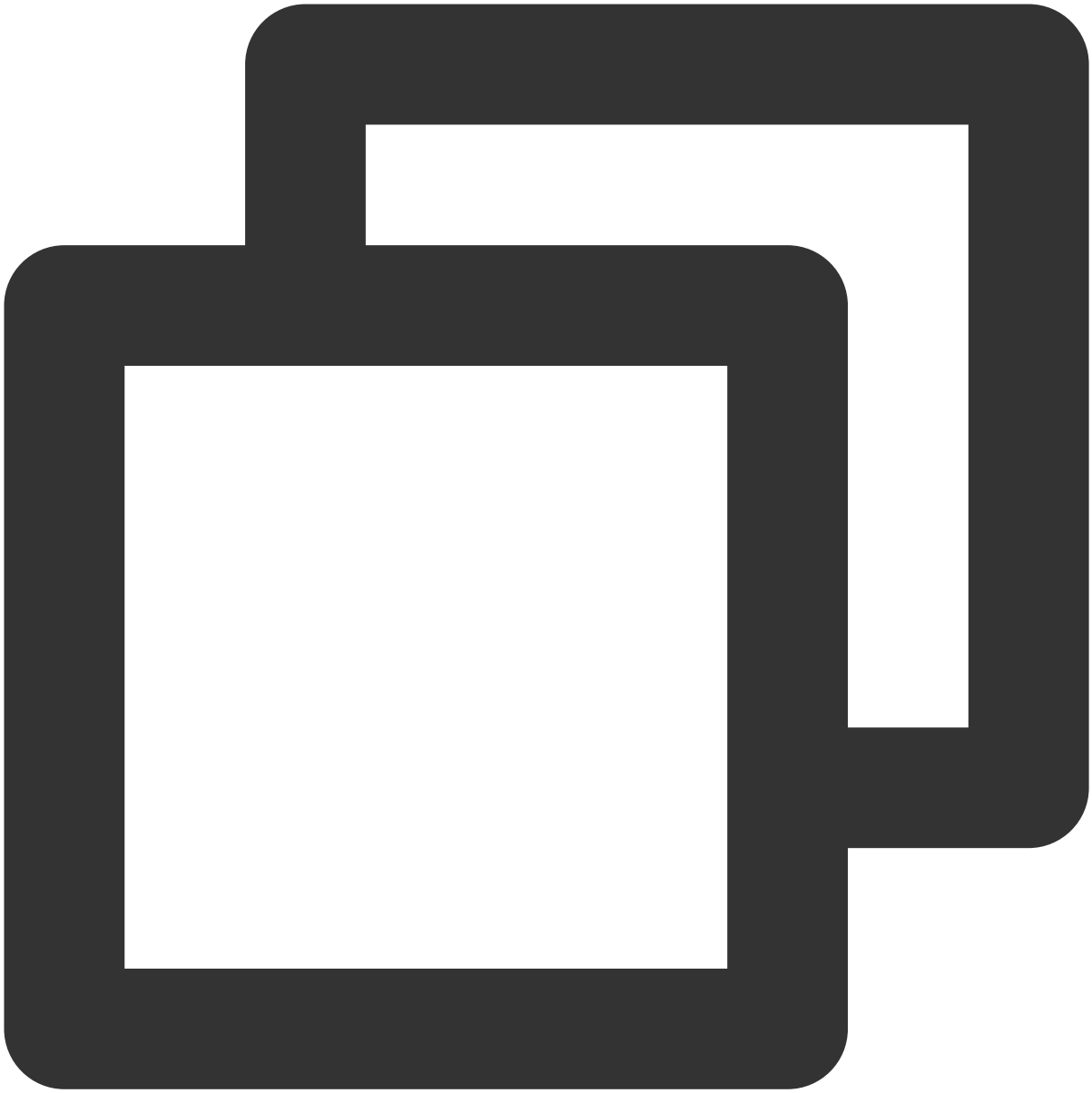
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>ExtractDigitalWatermark</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <ExtractDigitalWatermark>
                <Type>Text</Type>
                <Version>V1</Version>
            </ExtractDigitalWatermark>
            <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：ExtractDigitalWatermark	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
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	该任务的输入资源地址	Container
Operation	Response.JobsDetail	该任务的操作规则	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
ExtractDigitalWatermark	Response.JobsDetail.Operation	数字水印配置	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 类型 ExtractDigitalWatermark 的具体数据类型描述如下：

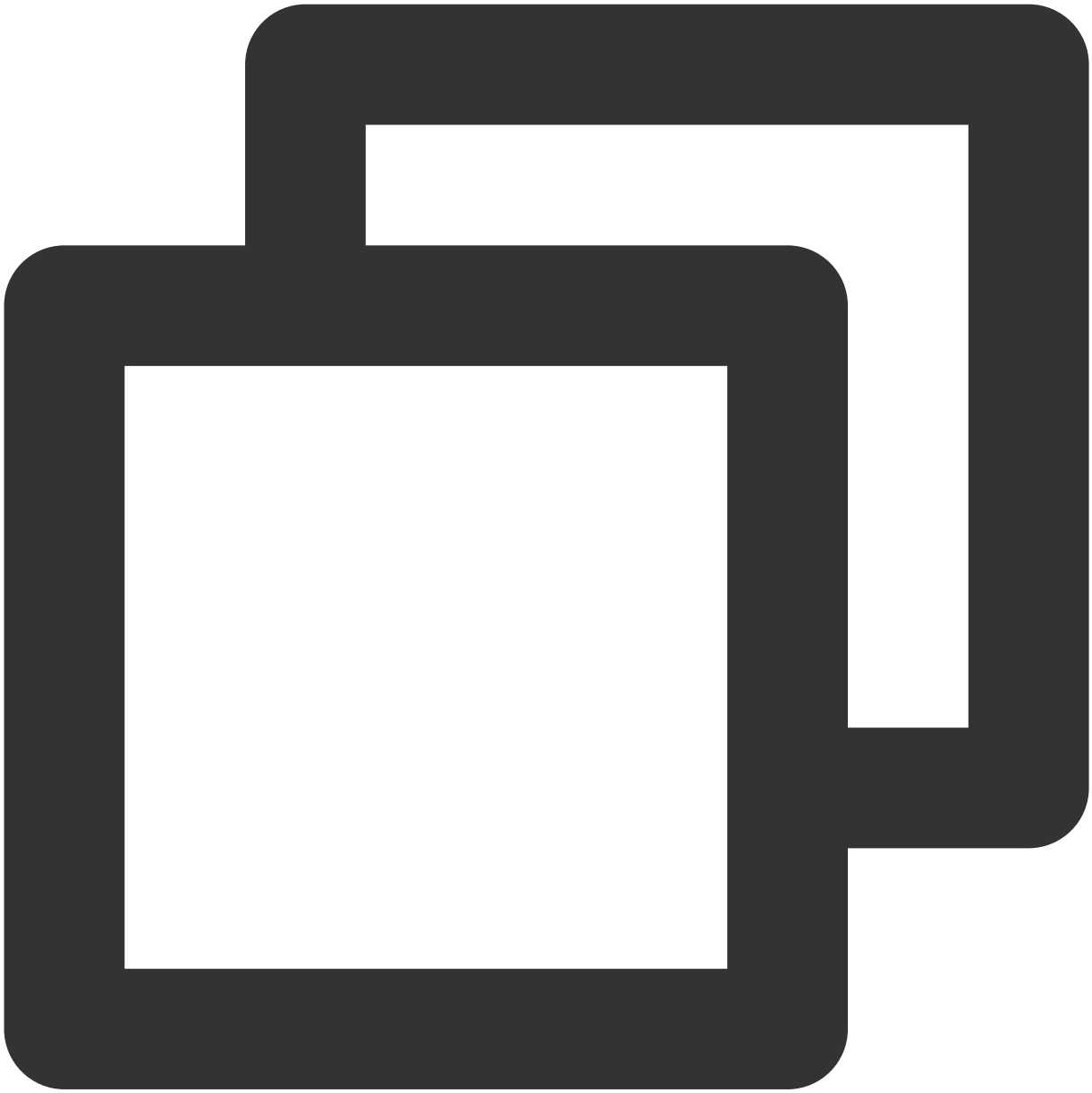
节点名称（关键字）	父节点	描述	类型
Message	Response.JobsDetail.Operation.ExtractDigitalWatermark	提取出的数字水印字符串信息	string
Type	Response.JobsDetail.Operation.ExtractDigitalWatermark	水印类型	String
Version	Response.JobsDetail.Operation.ExtractDigitalWatermark	水印版本	String

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

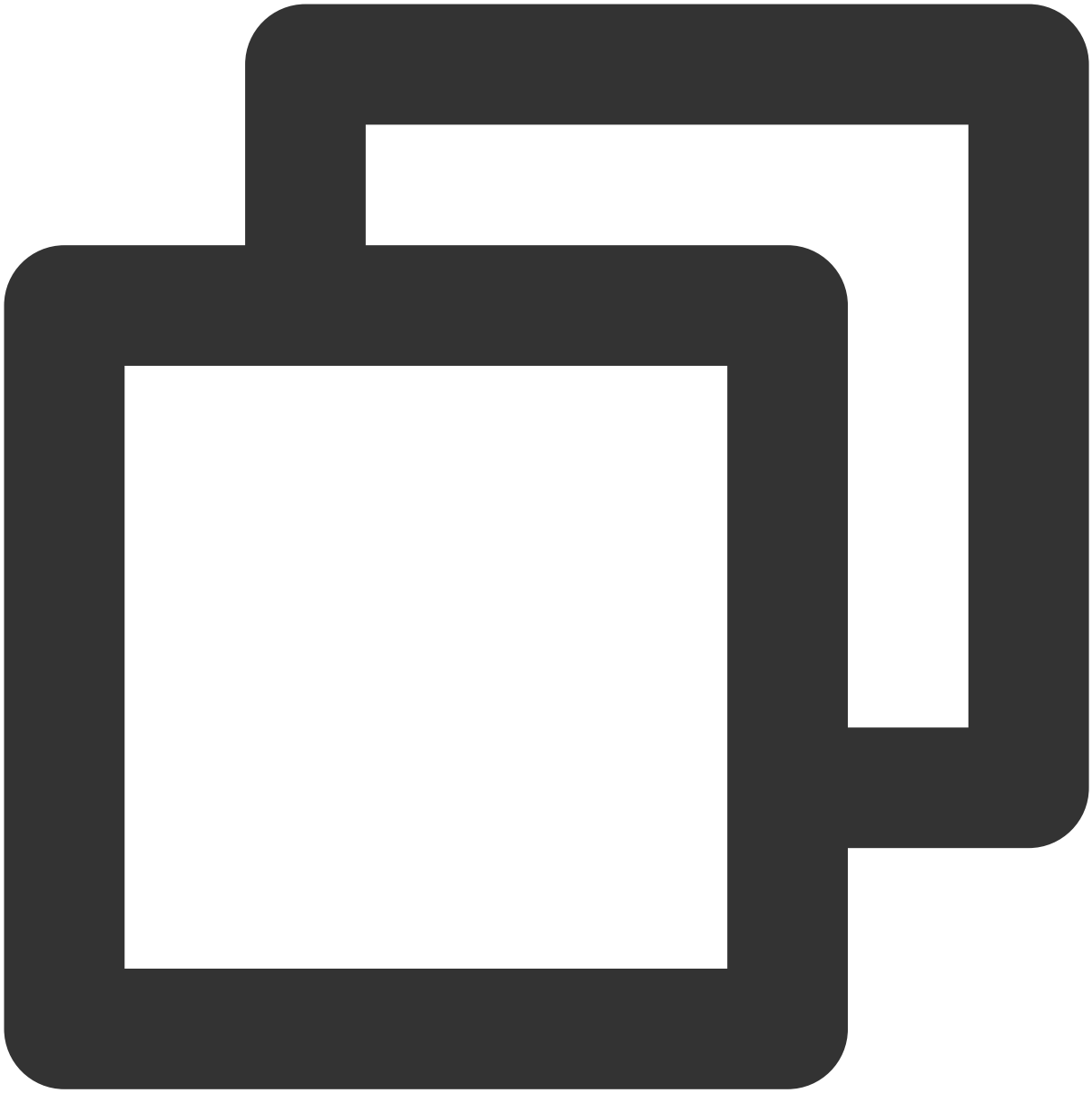


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>ExtractDigitalWatermark</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <ExtractDigitalWatermark>
            <Type>Text</Type>
            <Version>V1</Version>
        </ExtractDigitalWatermark>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>ExtractDigitalWatermark</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <ExtractDigitalWatermark>
                <Type>Text</Type>
                <Version>V1</Version>
            </ExtractDigitalWatermark>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交获取媒体信息任务

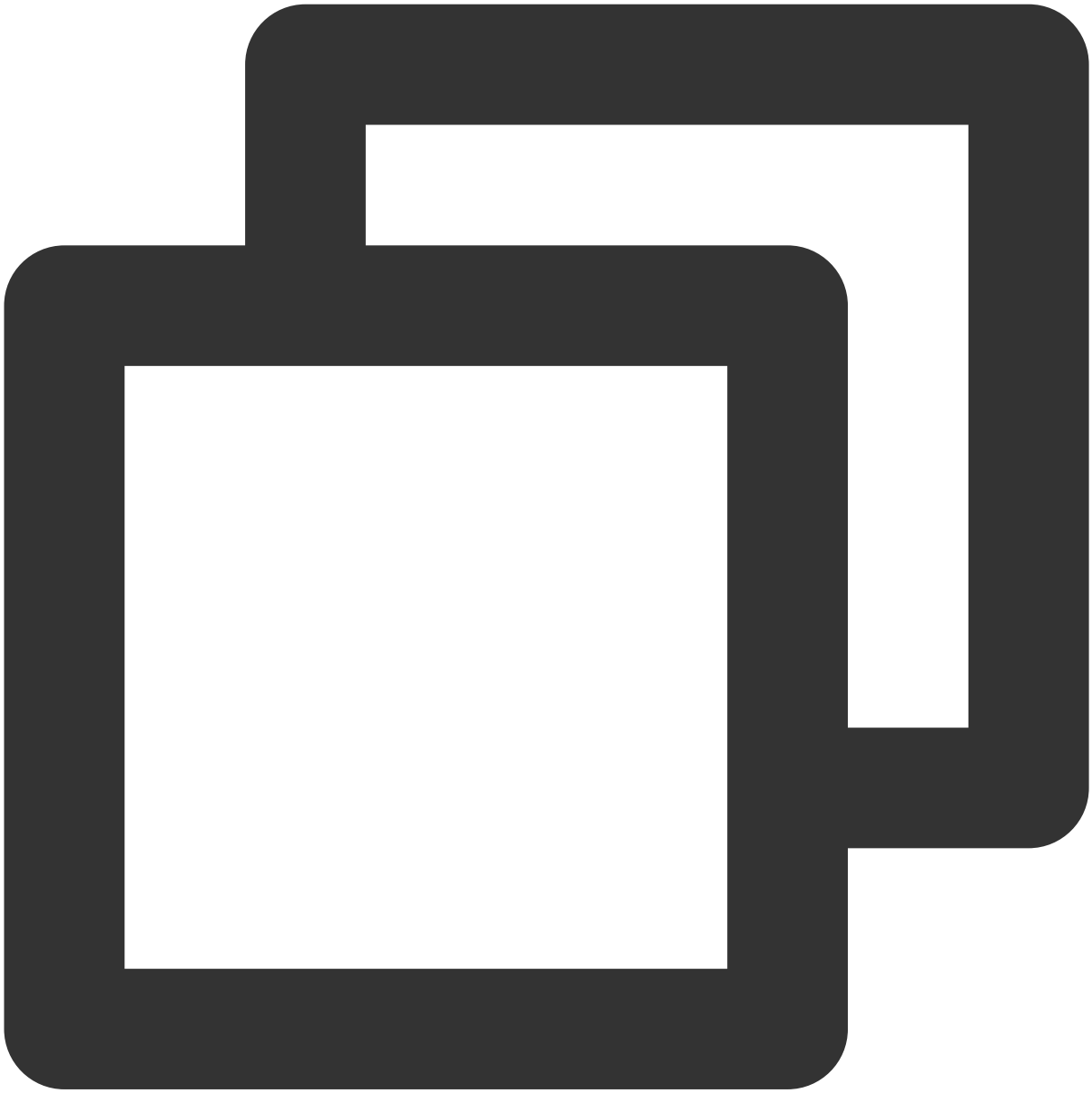
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个获取媒体信息任务。

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：

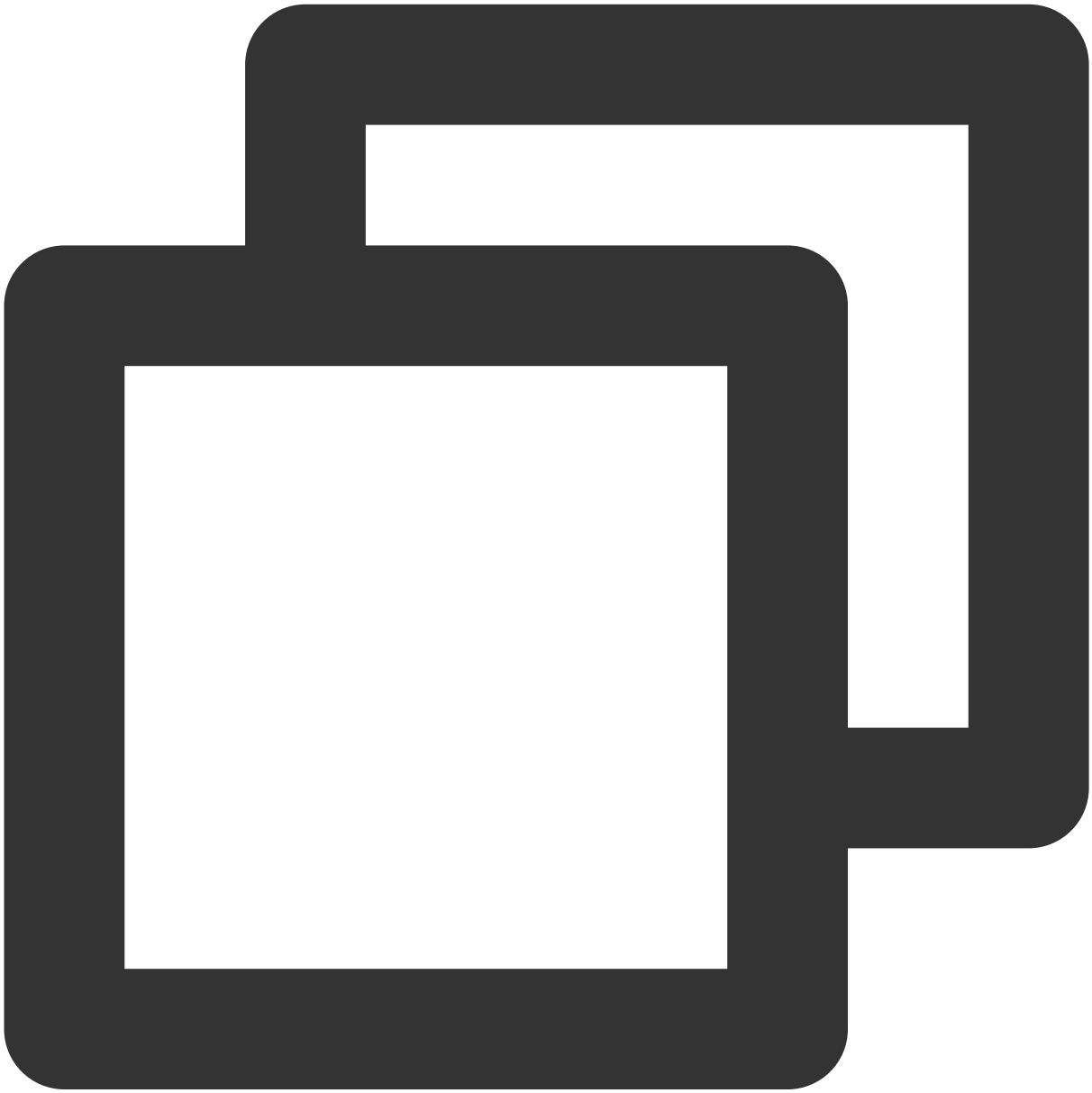
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>MediaInfo</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：MediaInfo	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
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 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
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 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

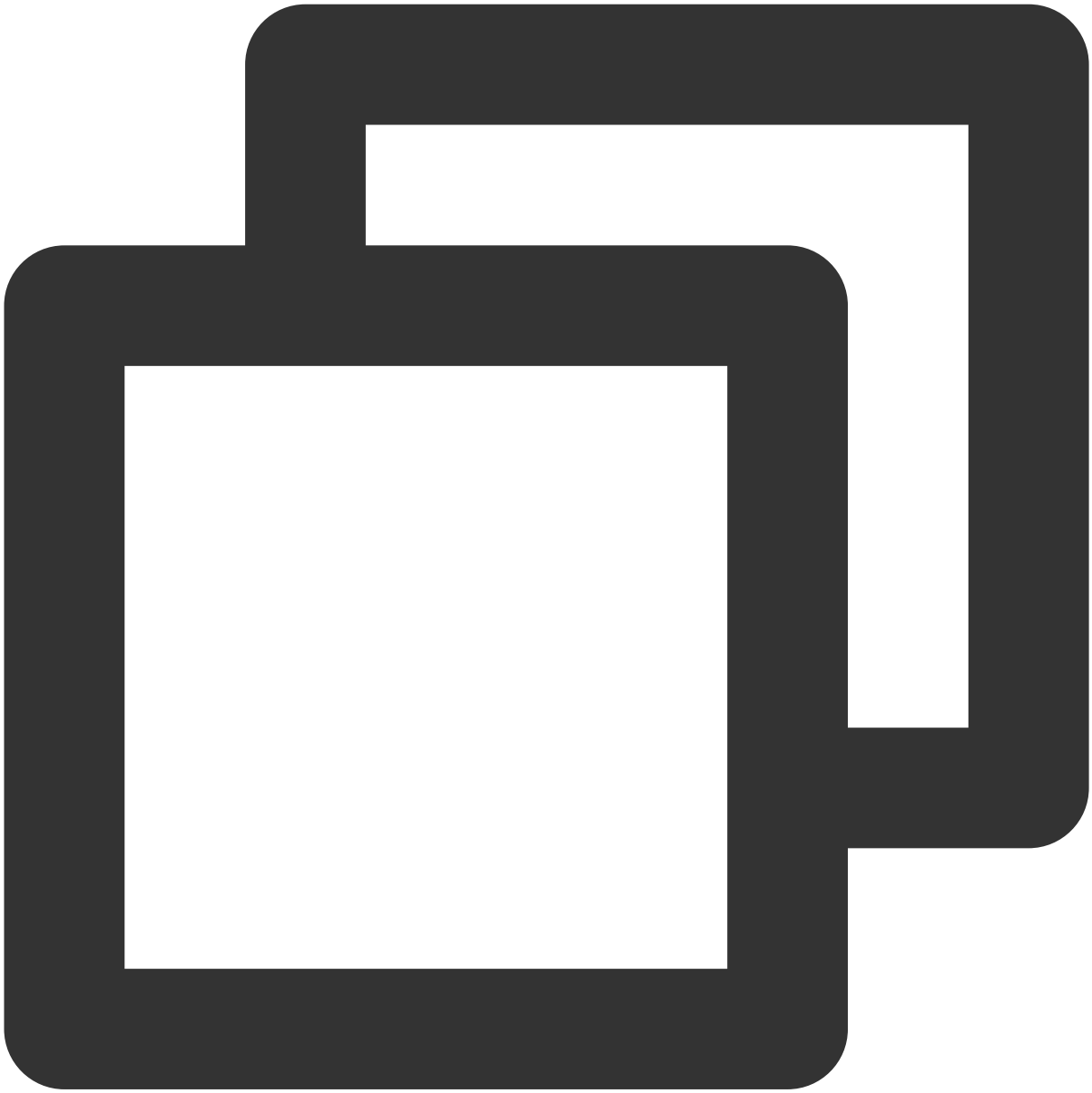
响应

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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>MediaInfo</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <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：MediaInfo	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
QueueId	Response.JobsDetail	任务所属的队列 ID	String

版权所有：腾讯云计算（北京）有限责任公司

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
MediaInfo	Response.JobsDetail.Operation	输出视频的信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

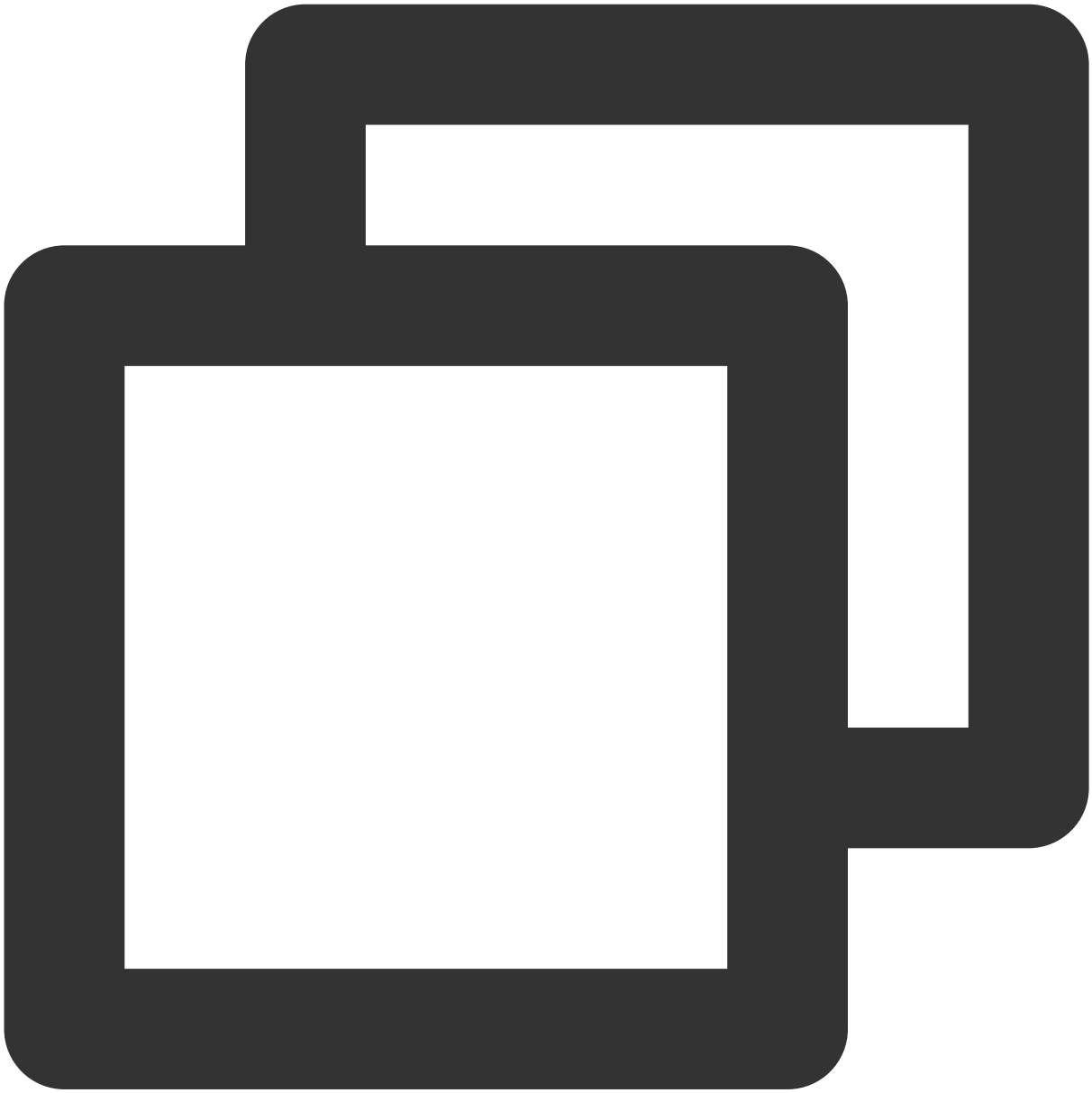
同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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：

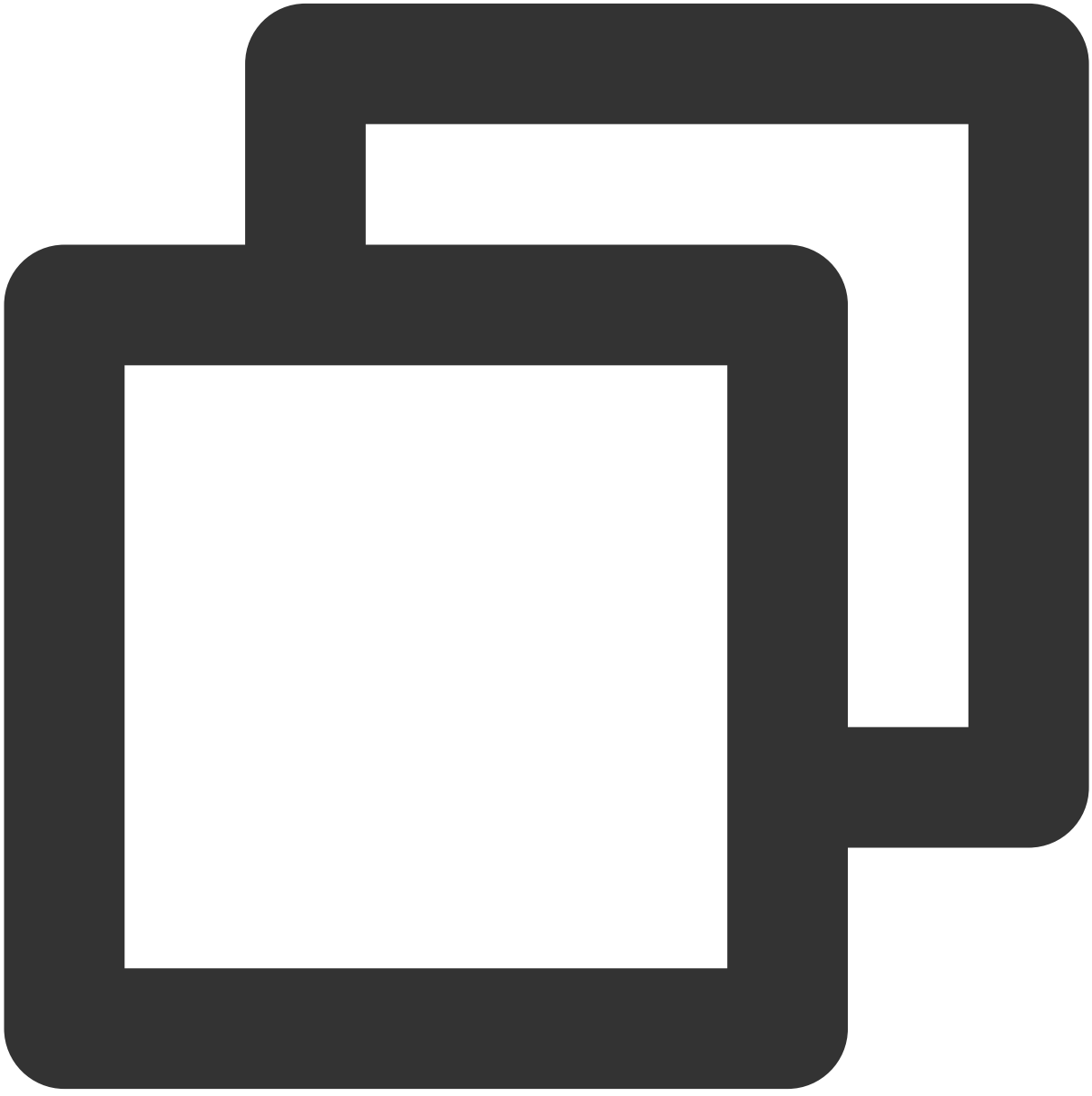


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>MediaInfo</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>MediaInfo</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <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

提交音频降噪任务

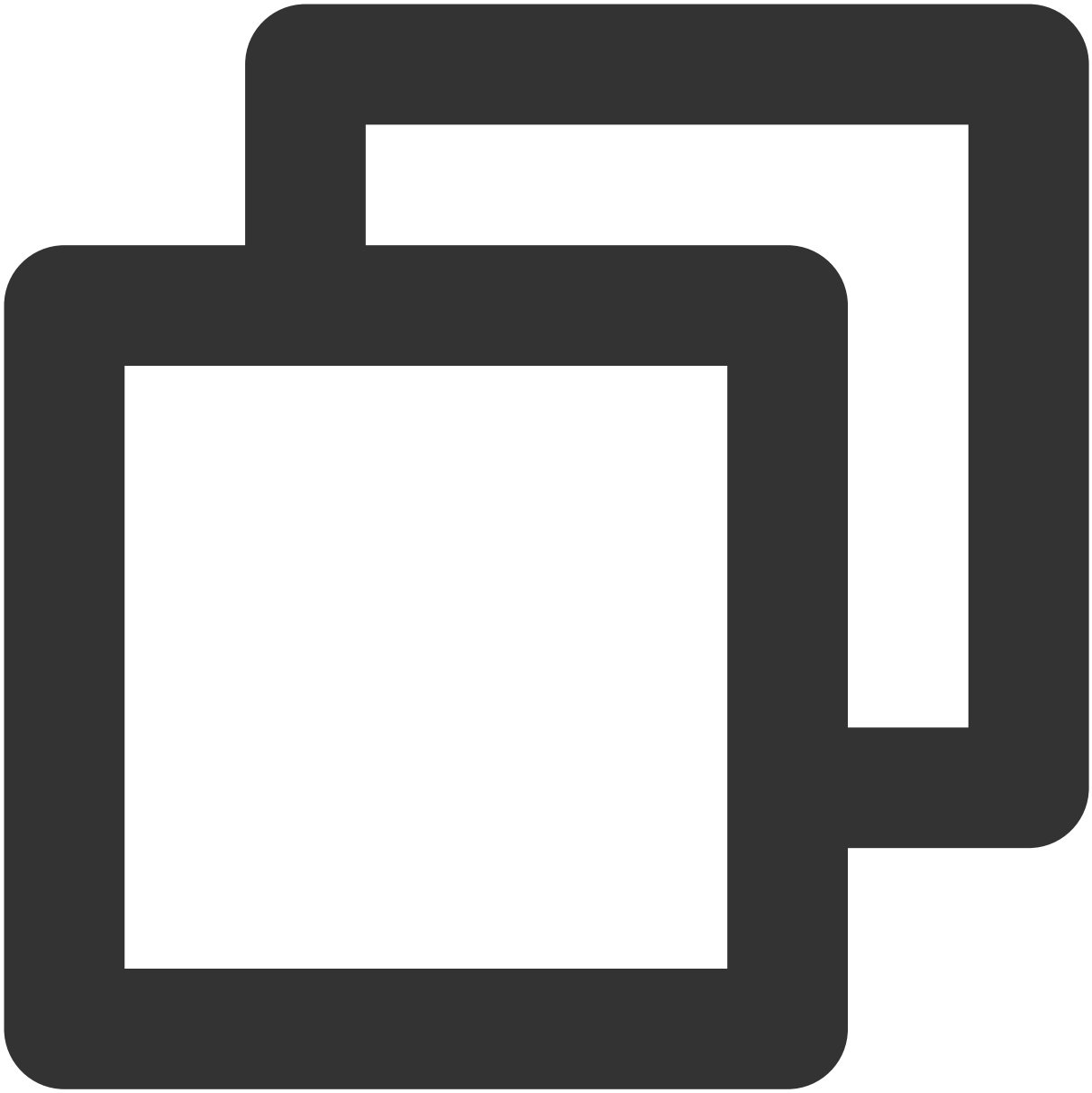
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个音频降噪任务。

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：

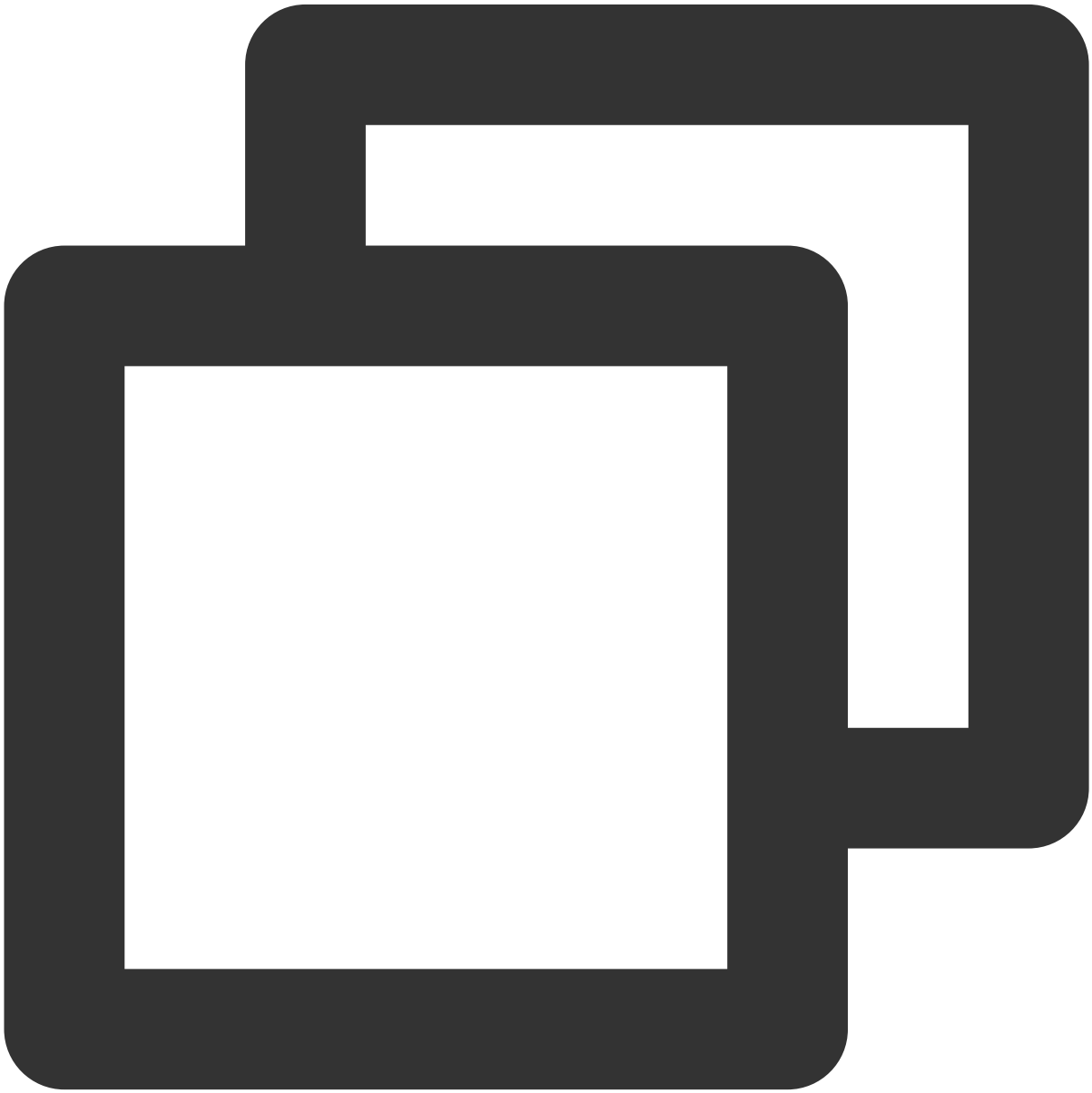
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>NoiseReduction</Tag>
    <Input>
        <Object>input/demo.mp3</Object>
    </Input>
    <Operation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.wav</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：NoiseReduction	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
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	Container	否

版权所有：腾讯云计算（北京）有限责任公司

		TDMQ 时必填。详情见 CallBackMqConfig		

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	执行音频降噪任务的媒体文件名 1. 目前只支持文件大小在10M之内的音频 2. 如果输入为视频文件或者多通道的音频，只会保留单通道的音频流 3. 目前暂不支持m3u8格式输入	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
Output	Request.Operation	结果输出地址	Container	是
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
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 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

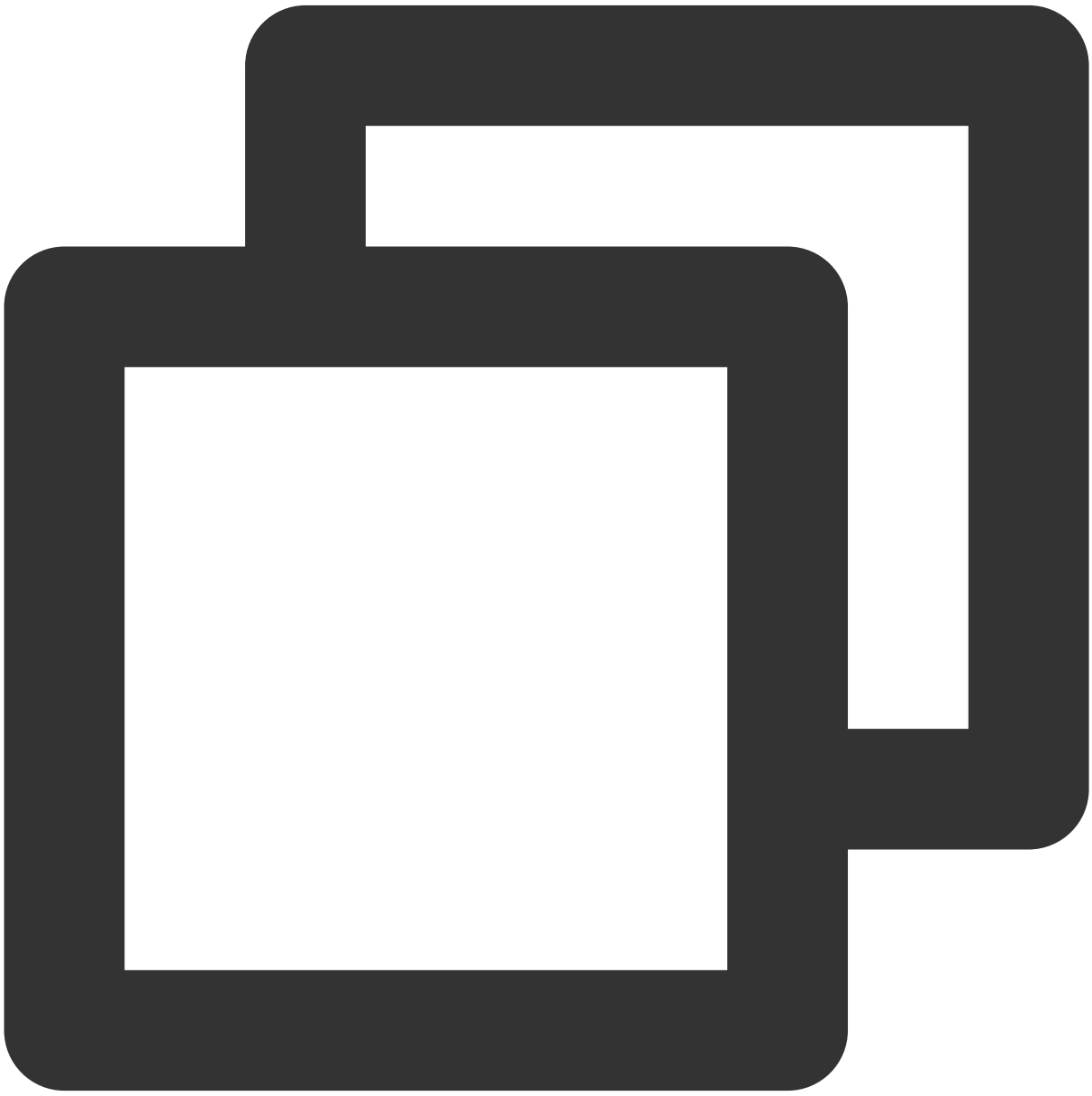
版权所有：腾讯云计算（北京）有限责任公司

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>NoiseReduction</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out.wav</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

版权所有：腾讯云计算（北京）有限责任公司

Tag	Response.JobsDetail	新创建任务的 Tag：NoiseReduction	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
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	该任务的操作规则	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：

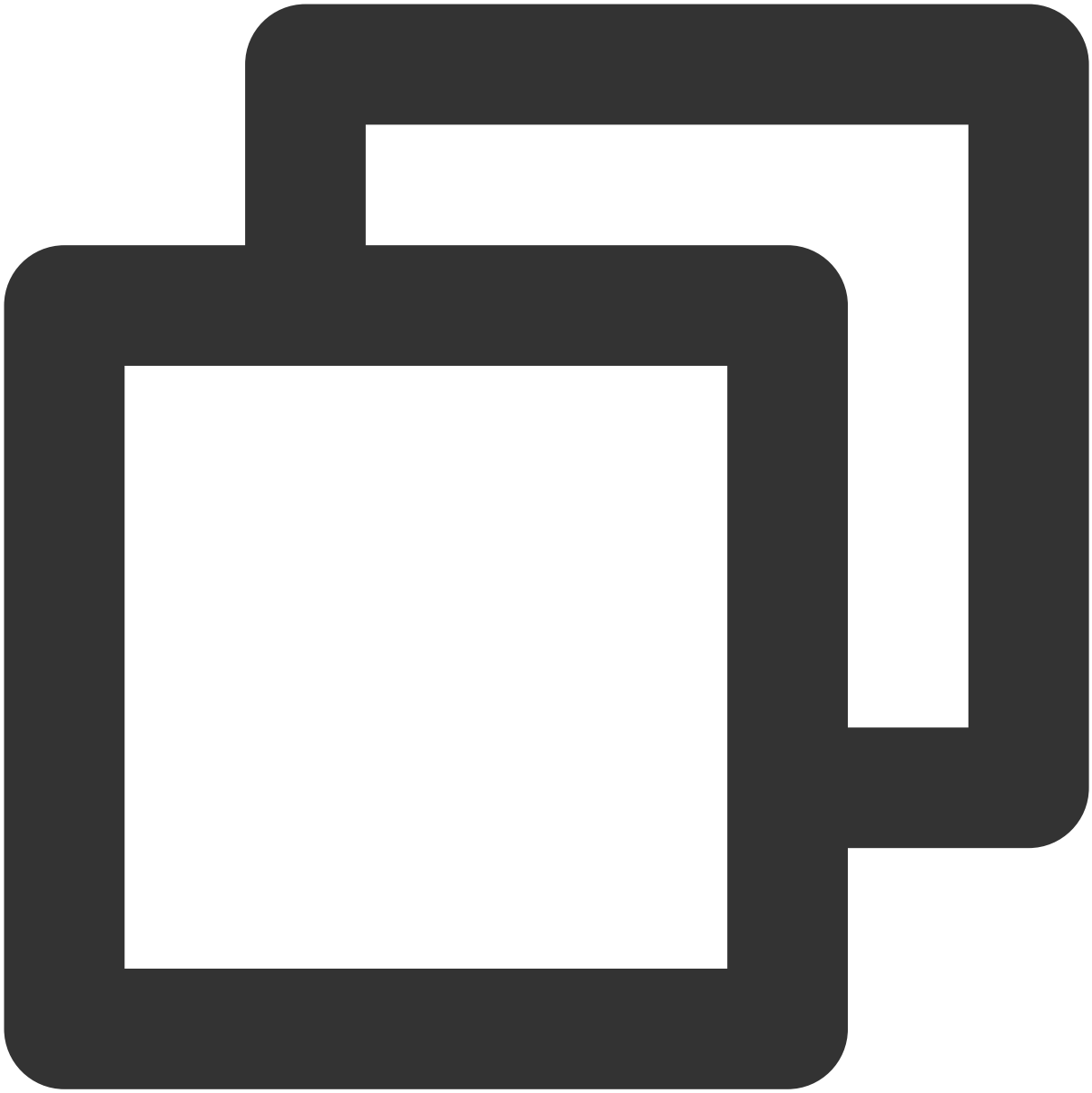
节点名称（关键字）	父节点	描述	类型
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

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

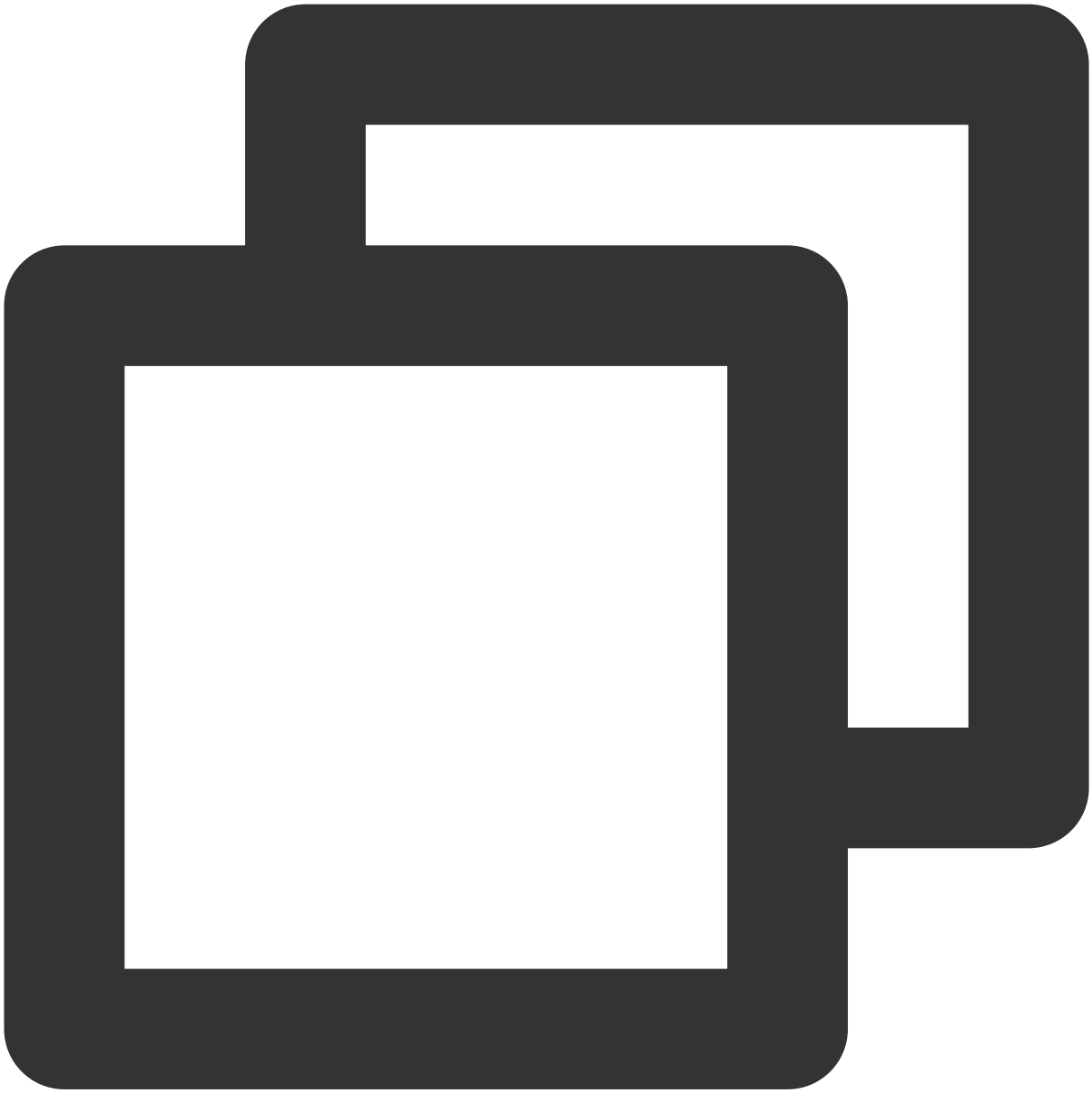


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>NoiseReduction</Tag>
    <Input>
        <Object>input/demo.mp3</Object>
    </Input>
```

```
    <Operation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.wav</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>NoiseReduction</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out.wav</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交视频质量评分任务

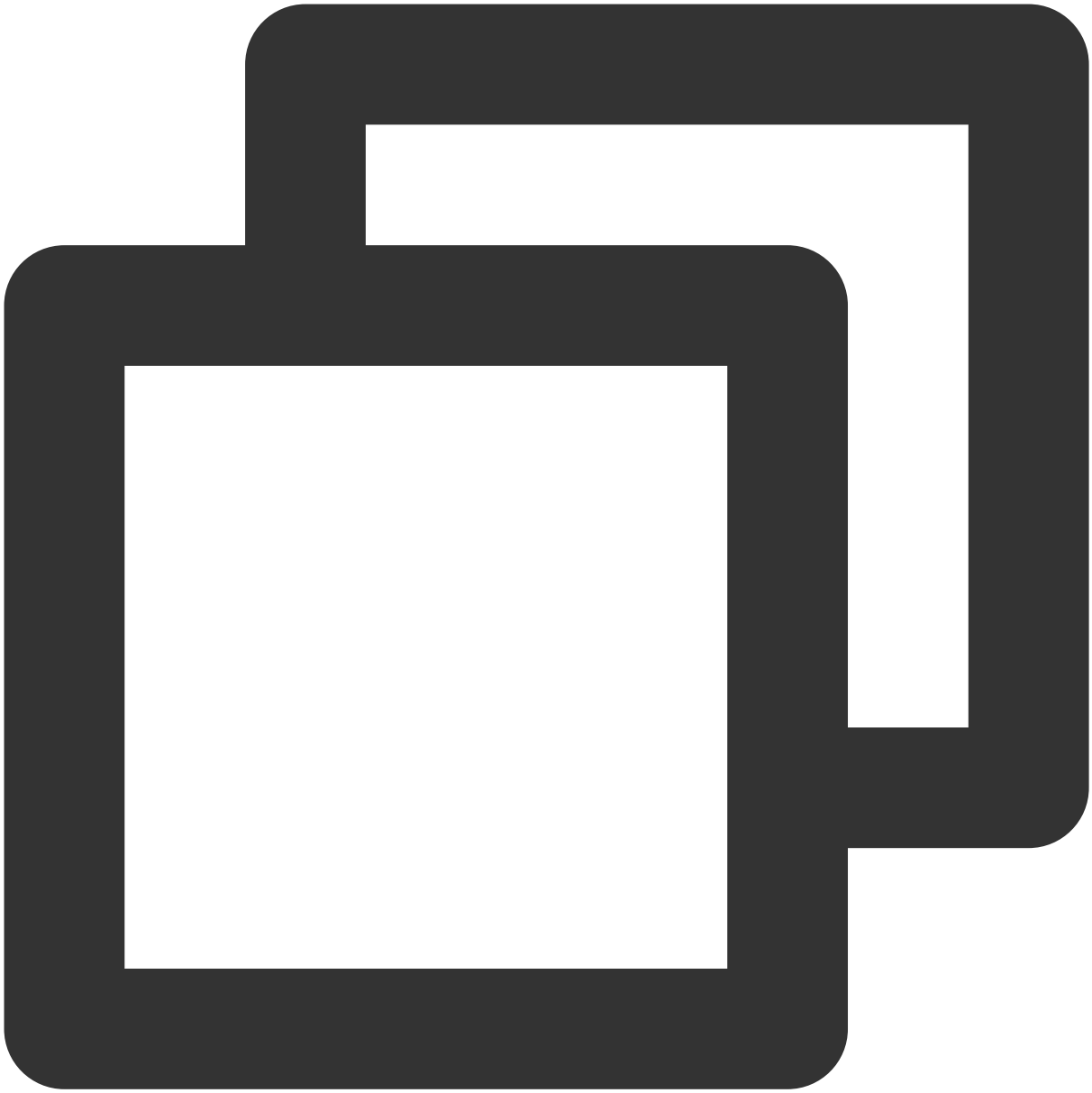
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个视频质量分析任务。

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：

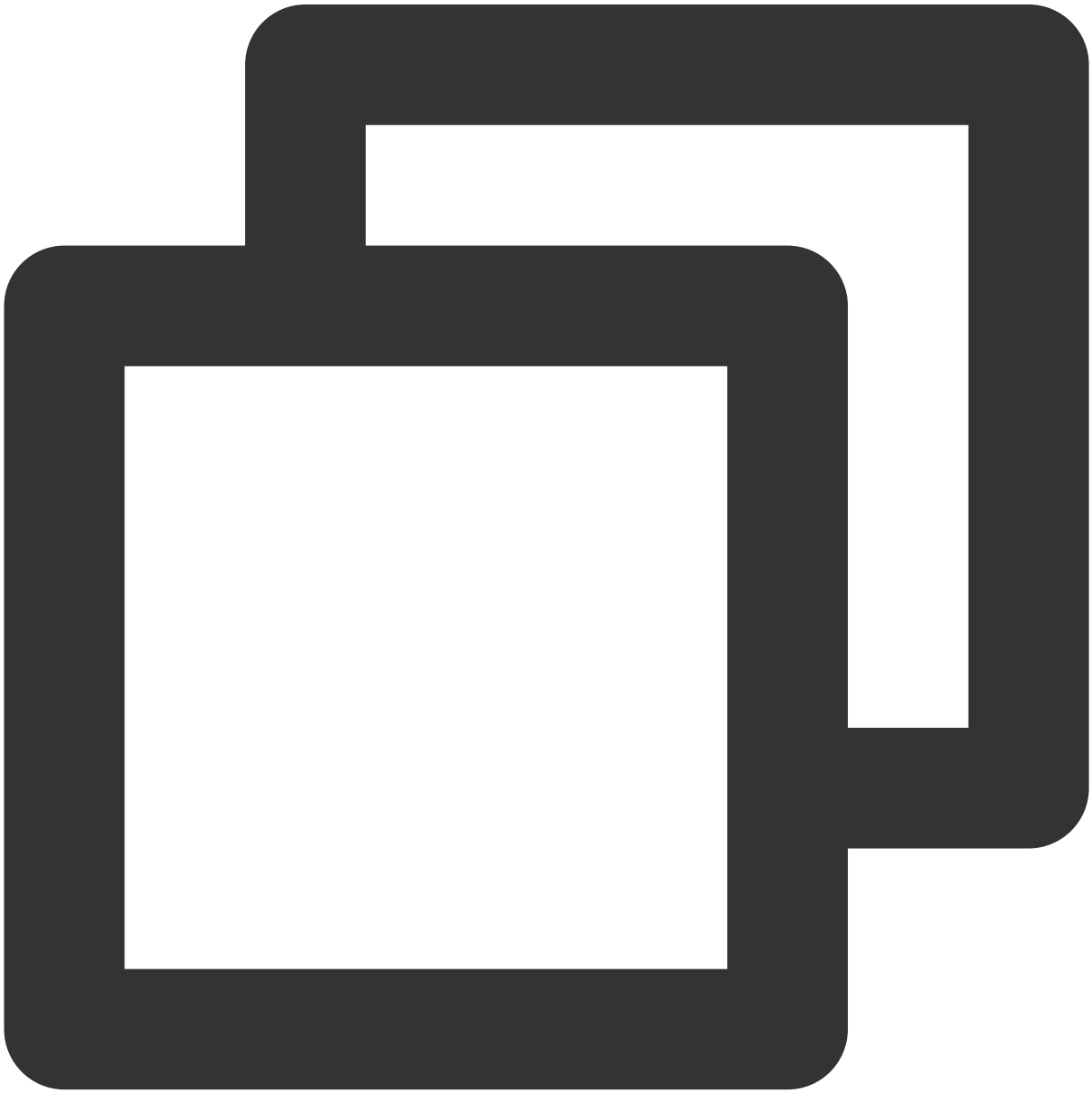
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>QualityEstimate</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：QualityEstimate	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
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 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
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 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

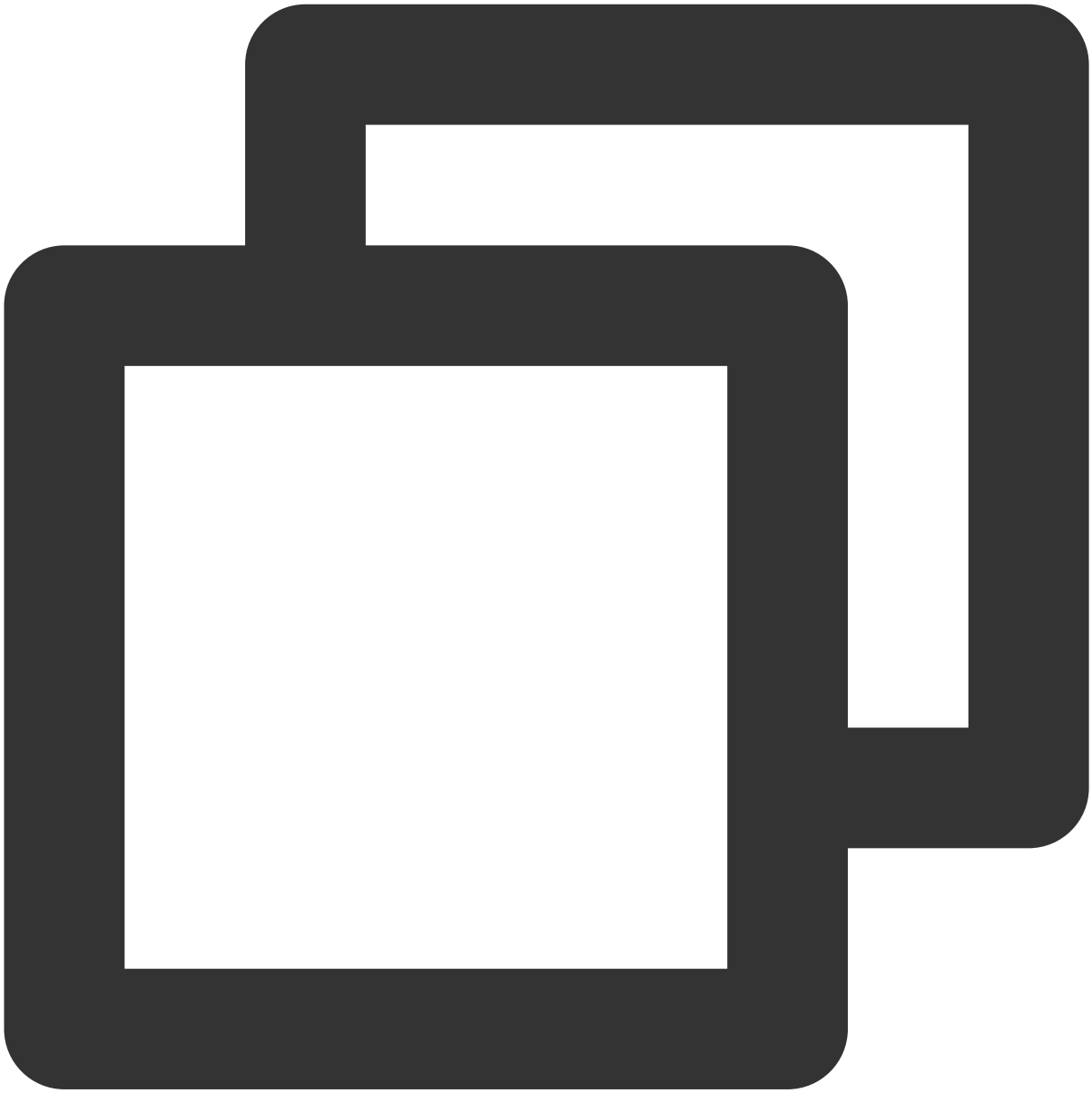
响应

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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>QualityEstimate</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <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：QualityEstimate	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
QueueId	Response.JobsDetail	任务所属的队列 ID	String

版权所有：腾讯云计算（北京）有限责任公司

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
MediaInfo	Response.JobsDetail.Operation	输出视频的信息，任务未完成时不返回	Container
QualityEstimate	Response.JobsDetail.Operation	视频分析结果，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String

Container 节点 QualityEstimate 的内容：

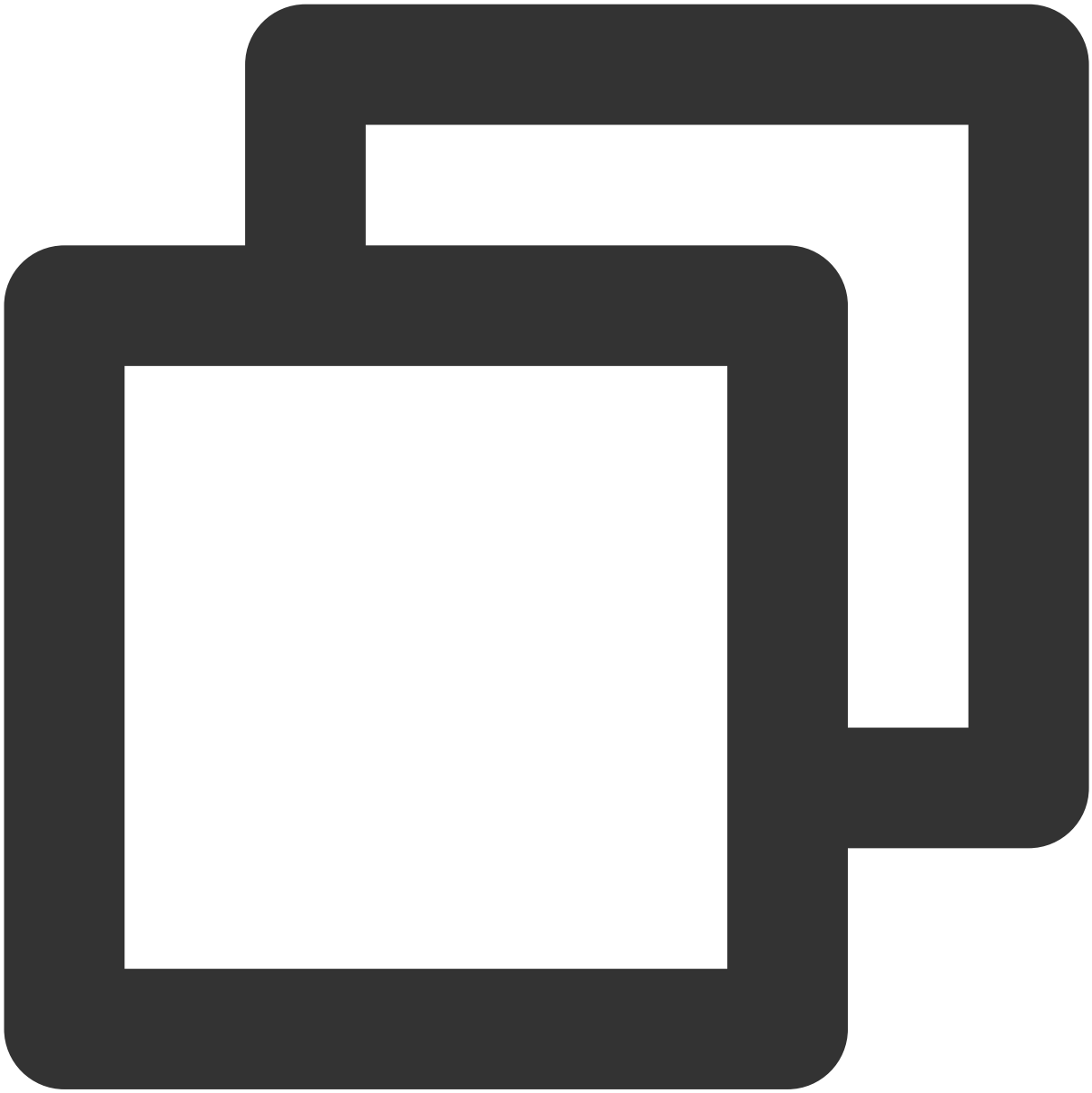
节点名称（关键字）	父节点	描述	类型
Score	Response.JobsDetail.Operation.QualityEstimate	视频质量得分，满分为10分，达到6分就可以认为视频质量良好	String

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

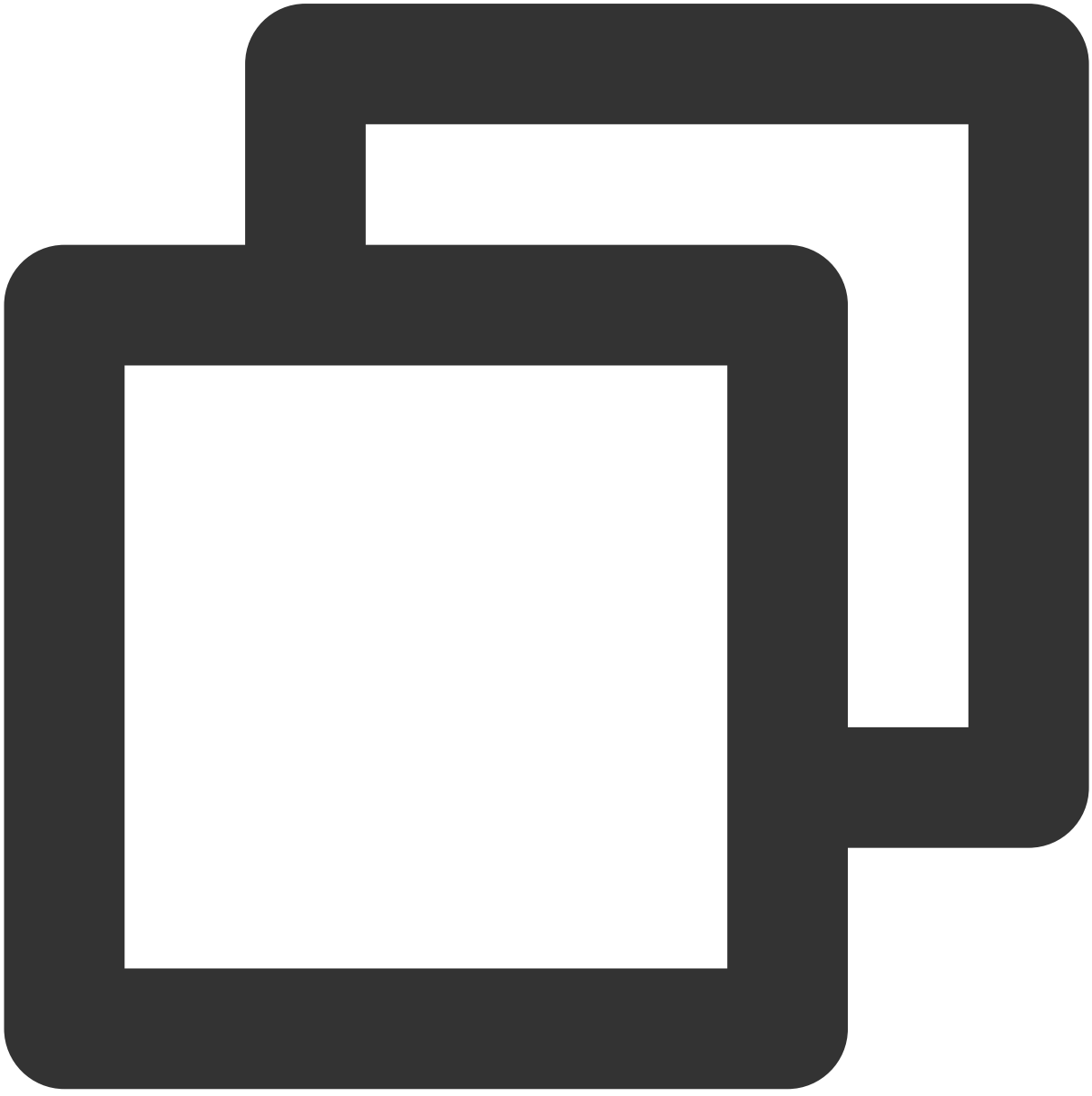


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>QualityEstimate</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>QualityEstimate</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交 SDRtoHDR 任务

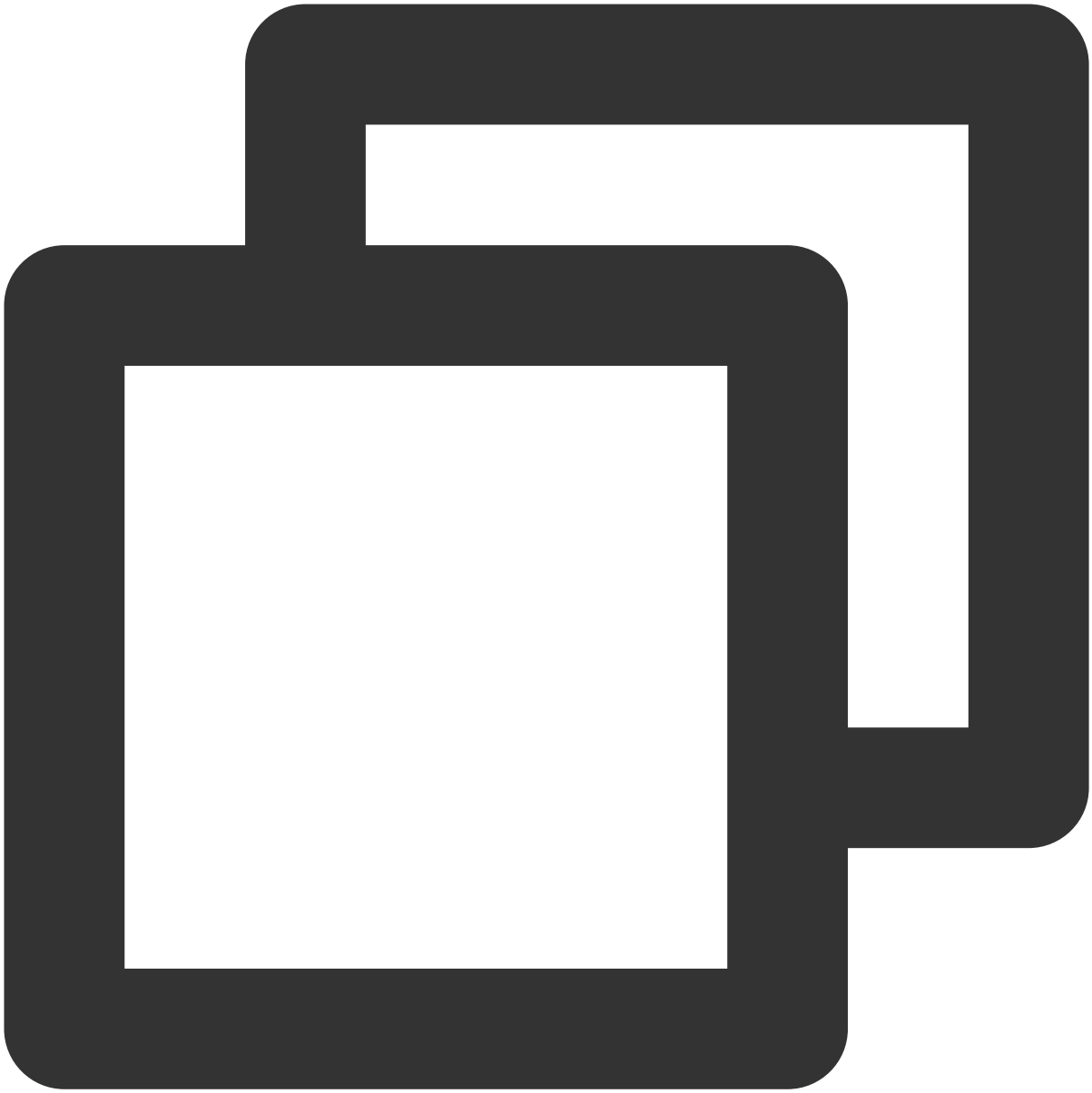
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个 SDR to HDR 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：

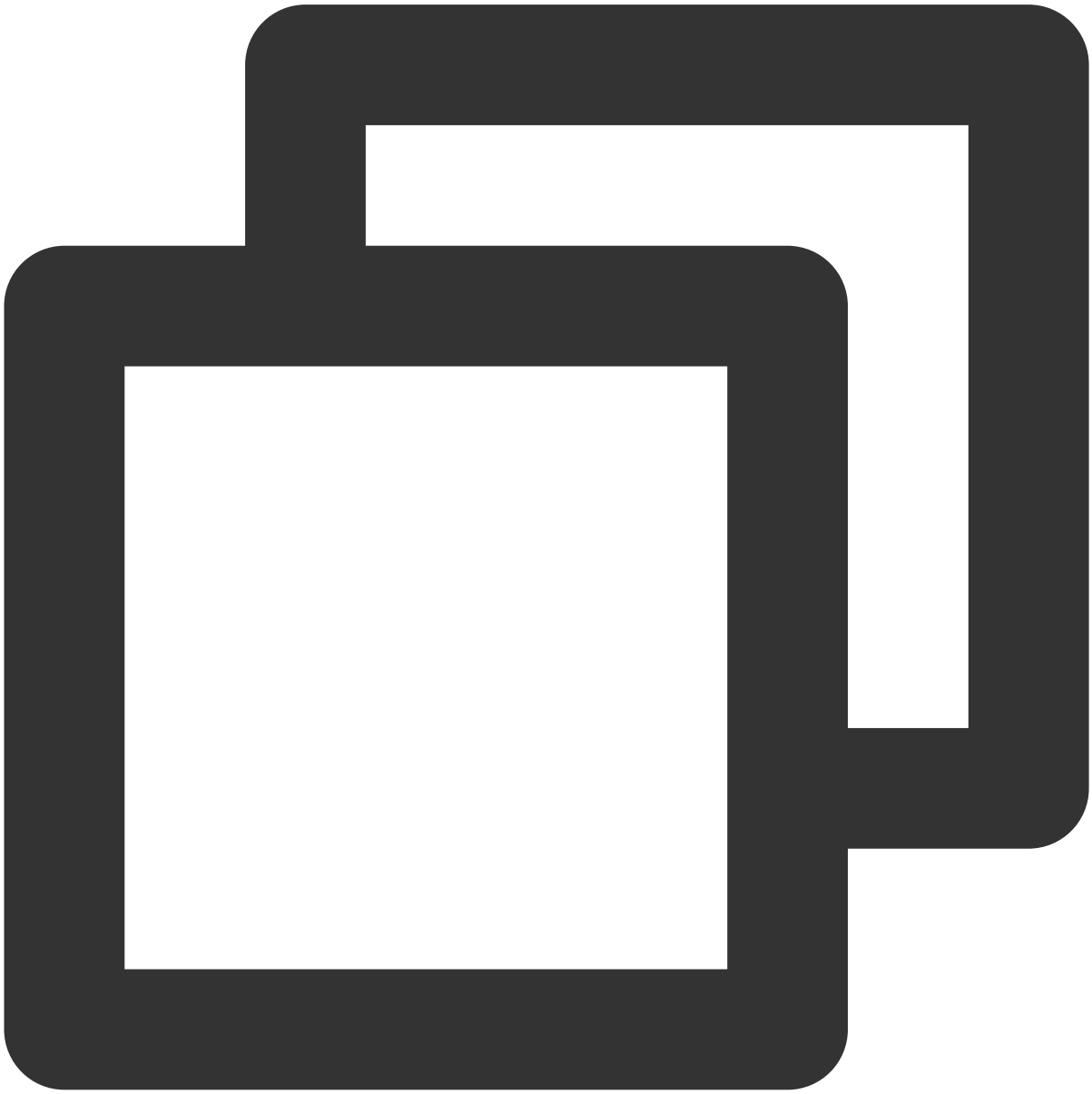
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>SDRtoHDR</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <SDRtoHDR>
            <HdrMode>HLG</HdrMode>
        </SDRtoHDR>
        <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplateI
        <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemplateI
        <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTemplateI
        <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>
    <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：SDRtoHDR	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
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
Object	Request.Input	媒体文件名	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
SDRtoHDR	Request.Operation	指定 SDRtoHDR 参数	Container	是
Transcode	Request.Operation	指定转码模板参数，不能与 TranscodeTemplateId 同时为空	Container	否
TranscodeTemplateId	Request.Operation	指定的转码模板 ID，优先使用模板 ID，不能与 Transcode 同时为空	String	否
Watermark	Request.Operation	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口中的 Request.Watermark, 最多传3个	Container 数组	否
WatermarkTemplateId	Request.Operation	指定的水印模板 ID，可以传多个水印模板 ID，最多传3个，优先使用模板 ID	String 数组	否
Output	Request.Operation	结果输出地址	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

说明：

提交 SDRtoHDR 任务必须传入转码参数。对于转码参数，优先使用 TranscodeTemplateId，无 TranscodeTemplateId 时使用 Transcode；对于水印参数，可以使用 WatermarkTemplateId 或Watermark 设置，WatermarkTemplateId 优先级更高。

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	限制
HdrMode	Request.Operation.SDRtoHDR	HDR 标准	String	是	1. HLG 2. HDR10

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	必选
TimeInterval	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Video	Container	否
Audio	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Audio	Container	否
TransConfig	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TransConfig	Container	否
AudioMix	Request.Operation.Transcode	混音参数, 详情见 AudioMix	Container 数组	否

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

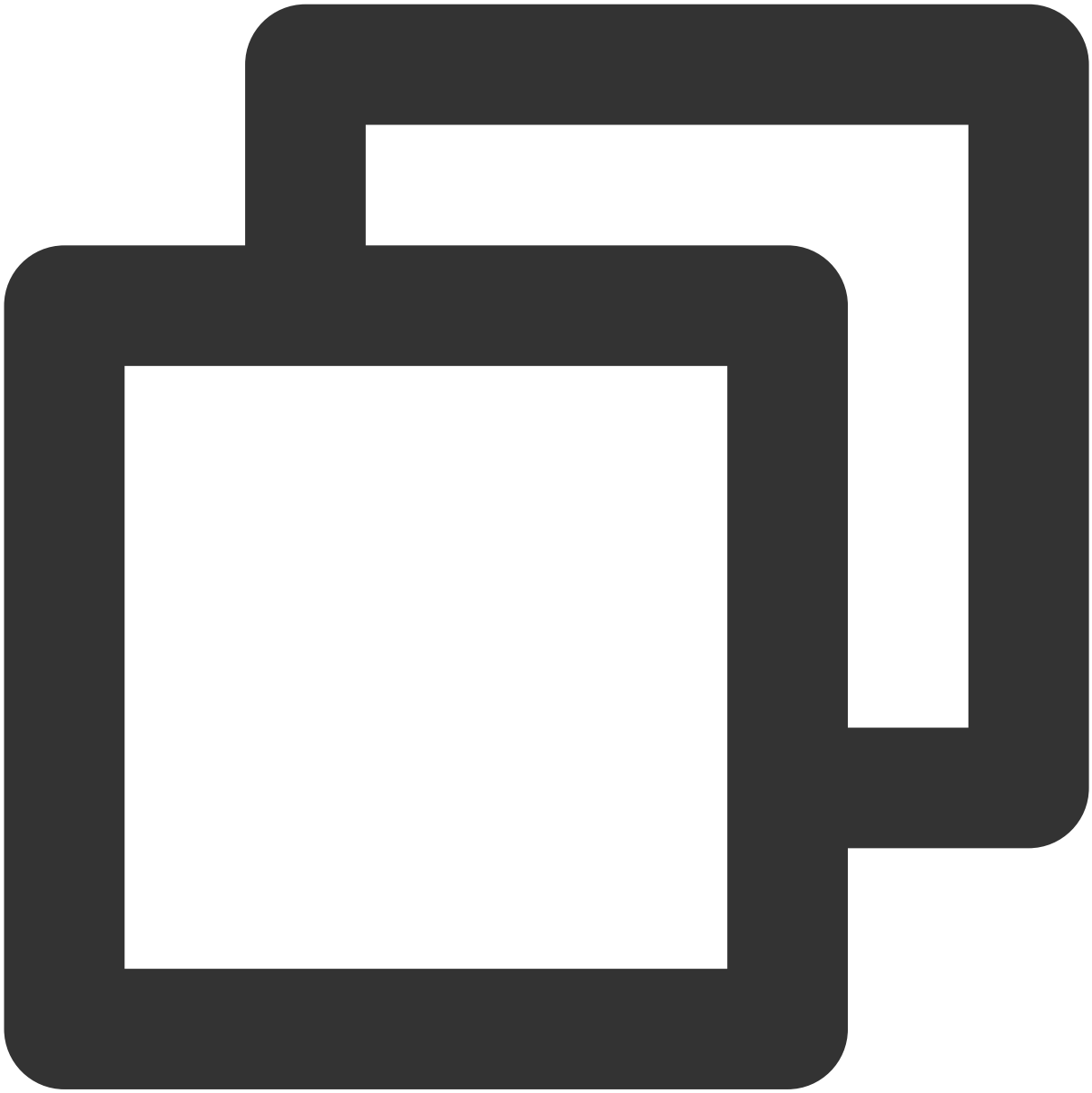
响应

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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SDRtoHDR</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TranscodeTemplateId>t1460606b9752148c4ab182f55163ba7cd</TranscodeTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <SDRtoHDR>
                <HdrMode>HDR10</HdrMode>
            </SDRtoHDR>
            <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>
```

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

版权所有：腾讯云计算（北京）有限责任公司

Tag	Response.JobsDetail	新创建任务的 Tag：SDRtoHDR	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
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
SDRtoHDR	Response.JobsDetail.Operation	同请求中的 Request.Operation.SDRtoHDR	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

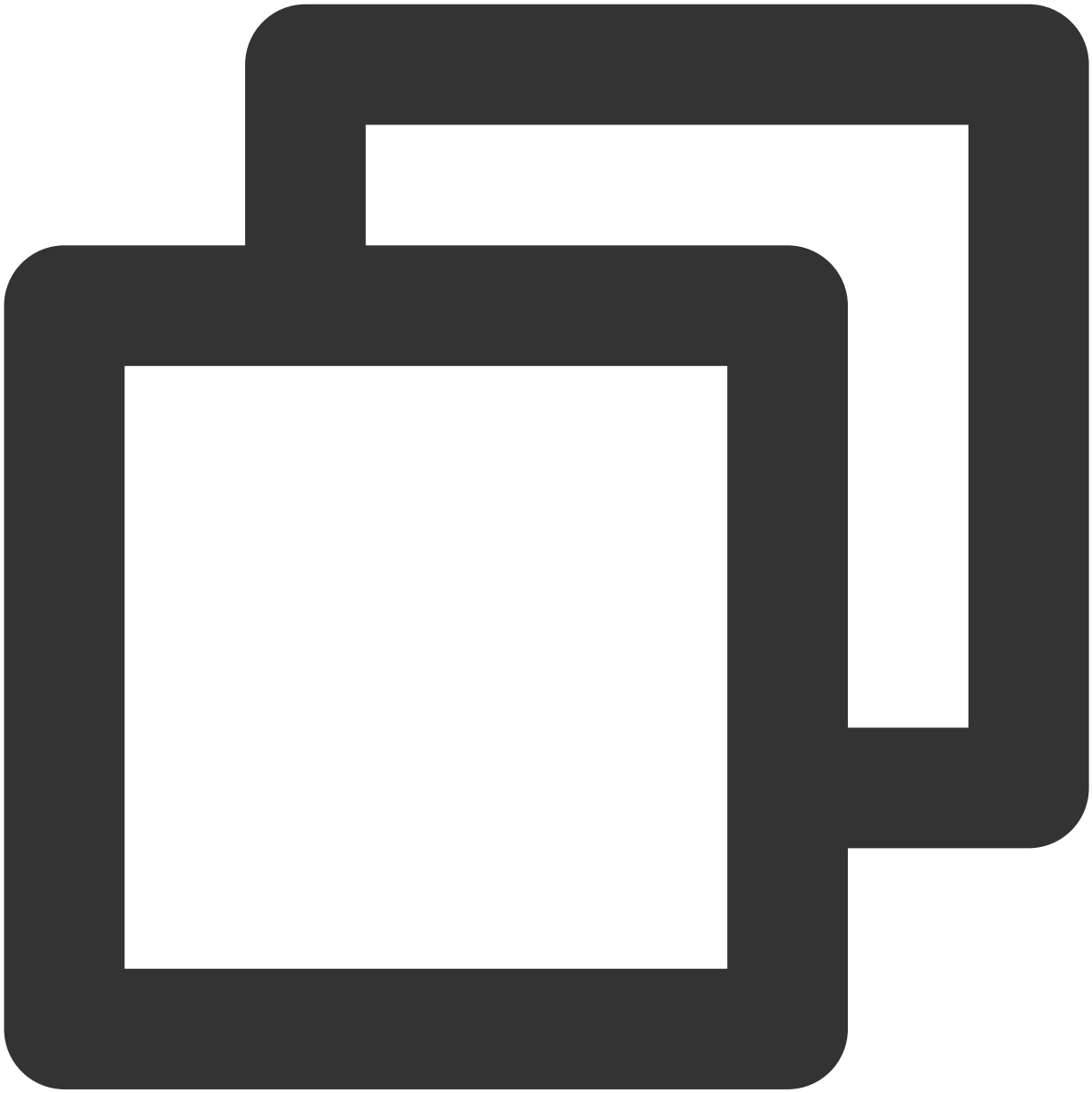
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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

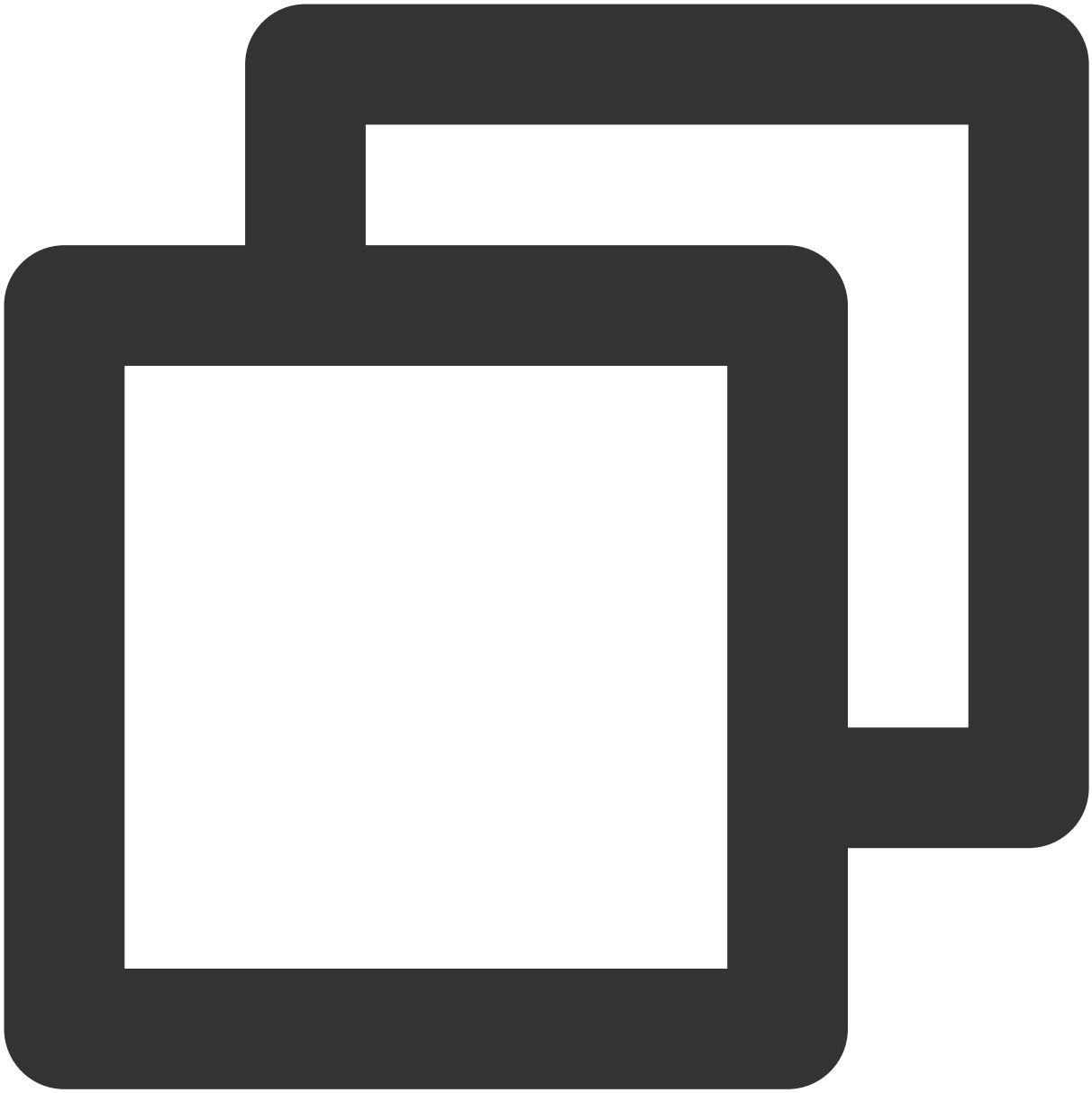


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SDRtoHDR</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <SDRtoHDR>
            <HdrMode>HLG</HdrMode>
        </SDRtoHDR>
        <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplateI
        <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemplateI
        <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTemplateI
        <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>
    <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>j229ed9e2f60c11ec8525e36307395bf9</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SDRtoHDR</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TranscodeTemplateId>t1460606b9752148c4ab182f55163ba7cd</TranscodeTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <SDRtoHDR>
                <HdrMode>HDR10</HdrMode>
            </SDRtoHDR>
            <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>
```

提交转封装任务

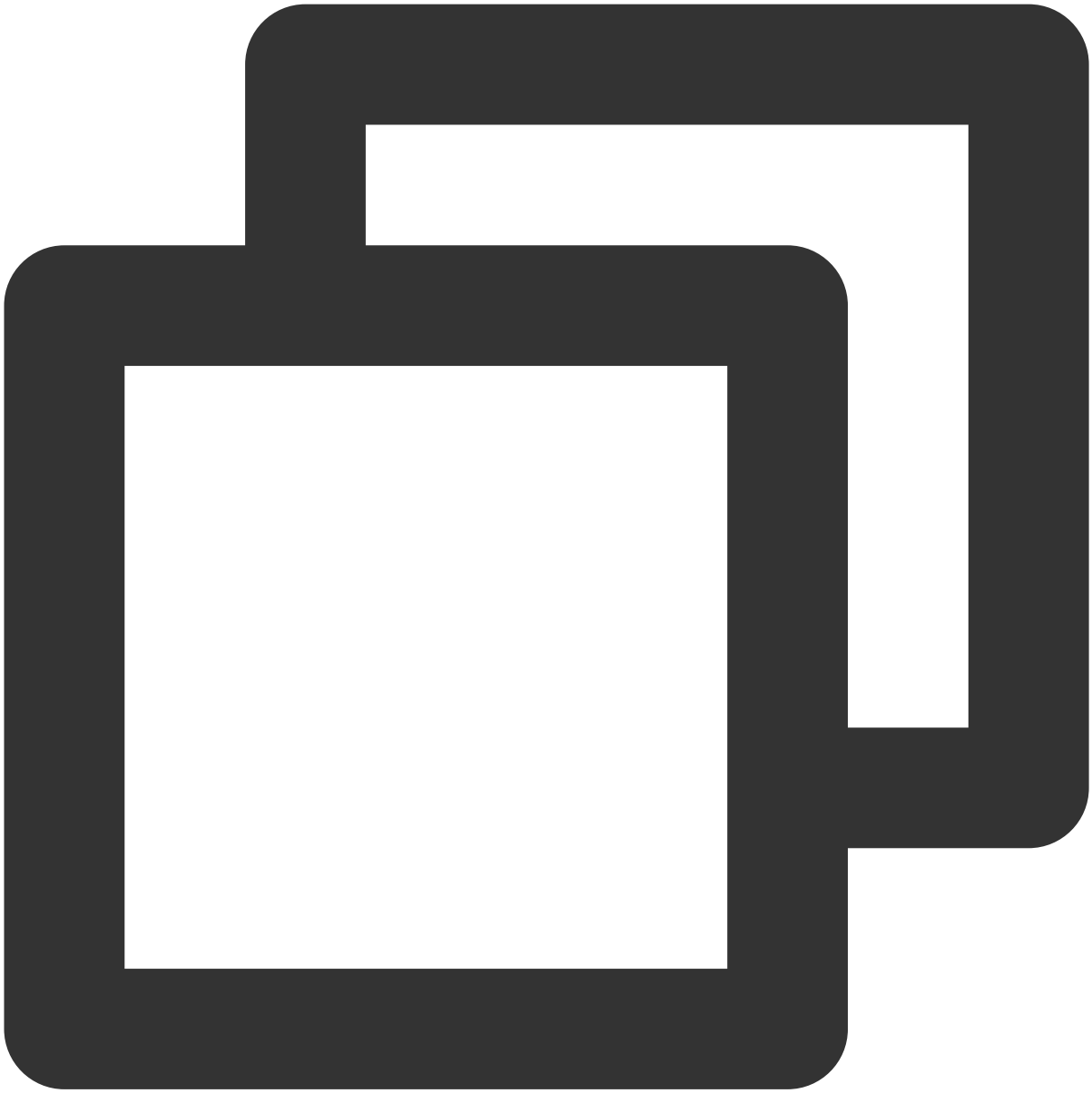
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个转封装任务。

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：

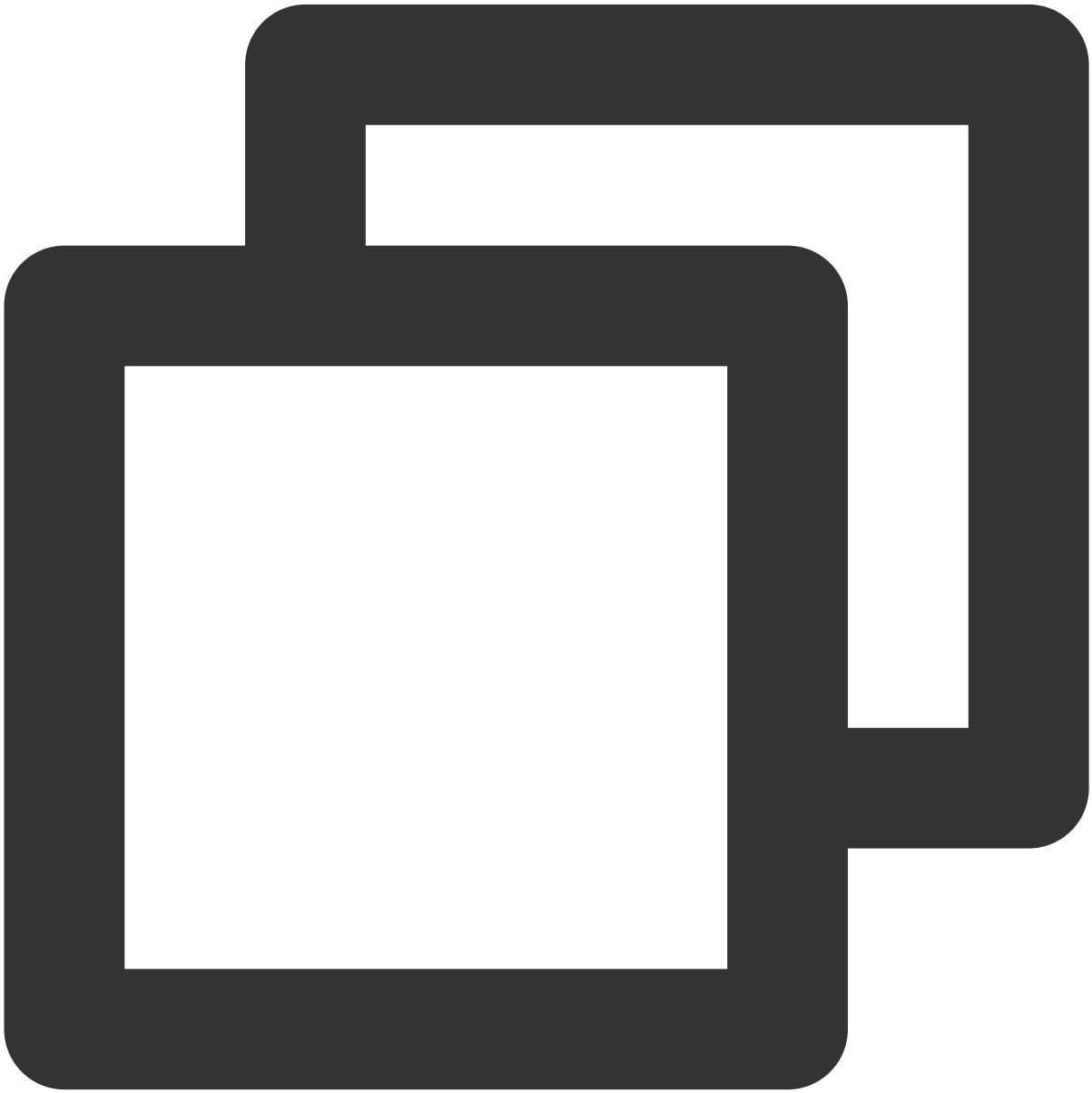
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Segment</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <Segment>
            <Format>hls</Format>
            <Duration>5</Duration>
            <HlsEncrypt>
                <IsHlsEncrypt>true</IsHlsEncrypt>
                <UriKey>test-key</UriKey>
            </HlsEncrypt>
        </Segment>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out-${Number}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：Segment	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
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
Object	Request.Input	媒体文件名	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
Segment	Request.Operation	指定转封装参数	Container	是
Output	Request.Operation	结果输出地址	Container	是
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 Segment 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型
Format	Request.Operation.Segment	封装格式	String

Duration	Request.Operation.Segment	转封装时长，单位：秒	String
TranscodeIndex	Request.Operation.Segment	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.Operation.Segment	hls 加密配置	Container

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.Operation.Segment.HlsEncrypt	是否开启 HLS 加密	String	否	false	1. true/false 2. Segment.Format 为 HLS 时支持加密
UriKey	Request.Operation.Segment.HlsEncrypt	HLS 加密的 key	String	否	无	当 IsHlsEncrypt 为 true 时，该参数才有意义

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，如果设置了 Duration，且 Format 不为 HLS 或 m3u8 时，文件名必须包含${Number}	String	是

版权所有：腾讯云计算（北京）有限责任公司

		参数作为自定义转封装后每一小段音/视频流的输出序号		

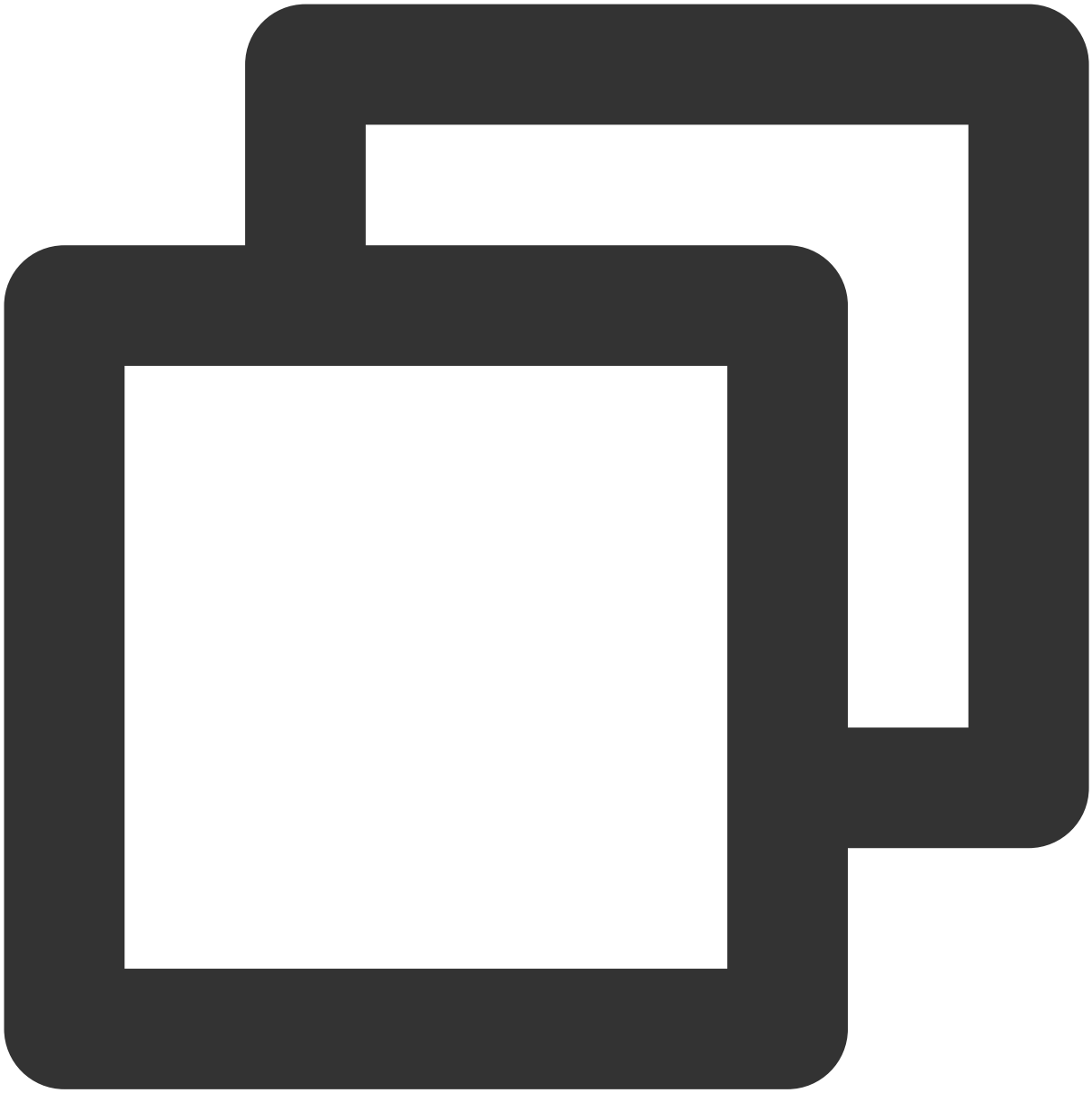
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Segment</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Segment>
                <Format>hls</Format>
                <Duration>5</Duration>
                <HlsEncrypt>
                    <IsHlsEncrypt>true</IsHlsEncrypt>
                    <UriKey>test-key</UriKey>
                </HlsEncrypt>
            </Segment>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out-${Number}</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <CallBack>http://callback.demo.com</CallBack>
        <CallBackFormat>JSON<CallBackFormat>
    </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

版权所有：腾讯云计算（北京）有限责任公司

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：Segment	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
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
Segment	Response.JobsDetail.Operation	同请求中的 Request.Operation.Segment	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	转码输出视频的信息，没有时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String

版权所有：腾讯云计算（北京）有限责任公司

JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

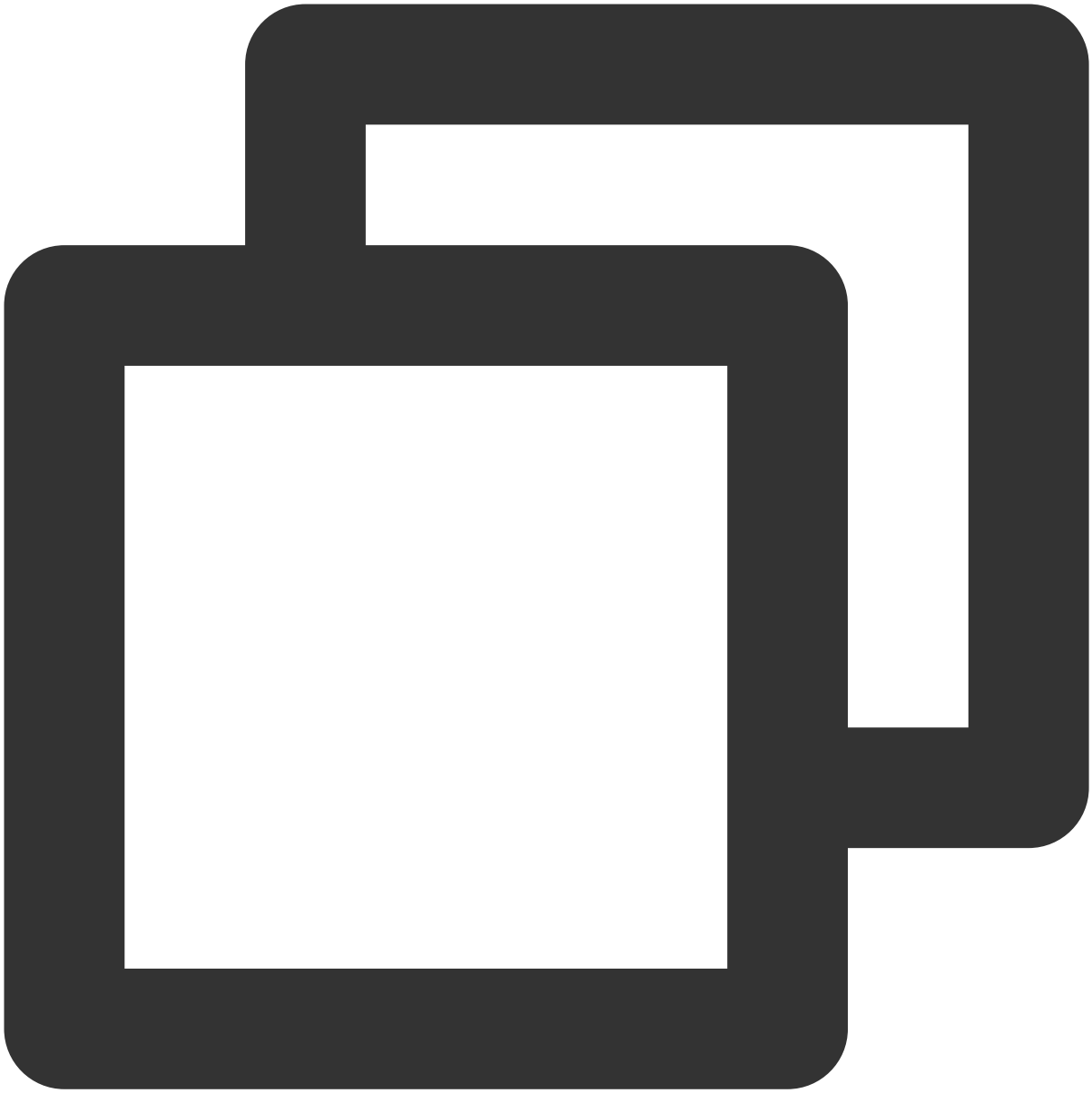
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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：mp4转封装

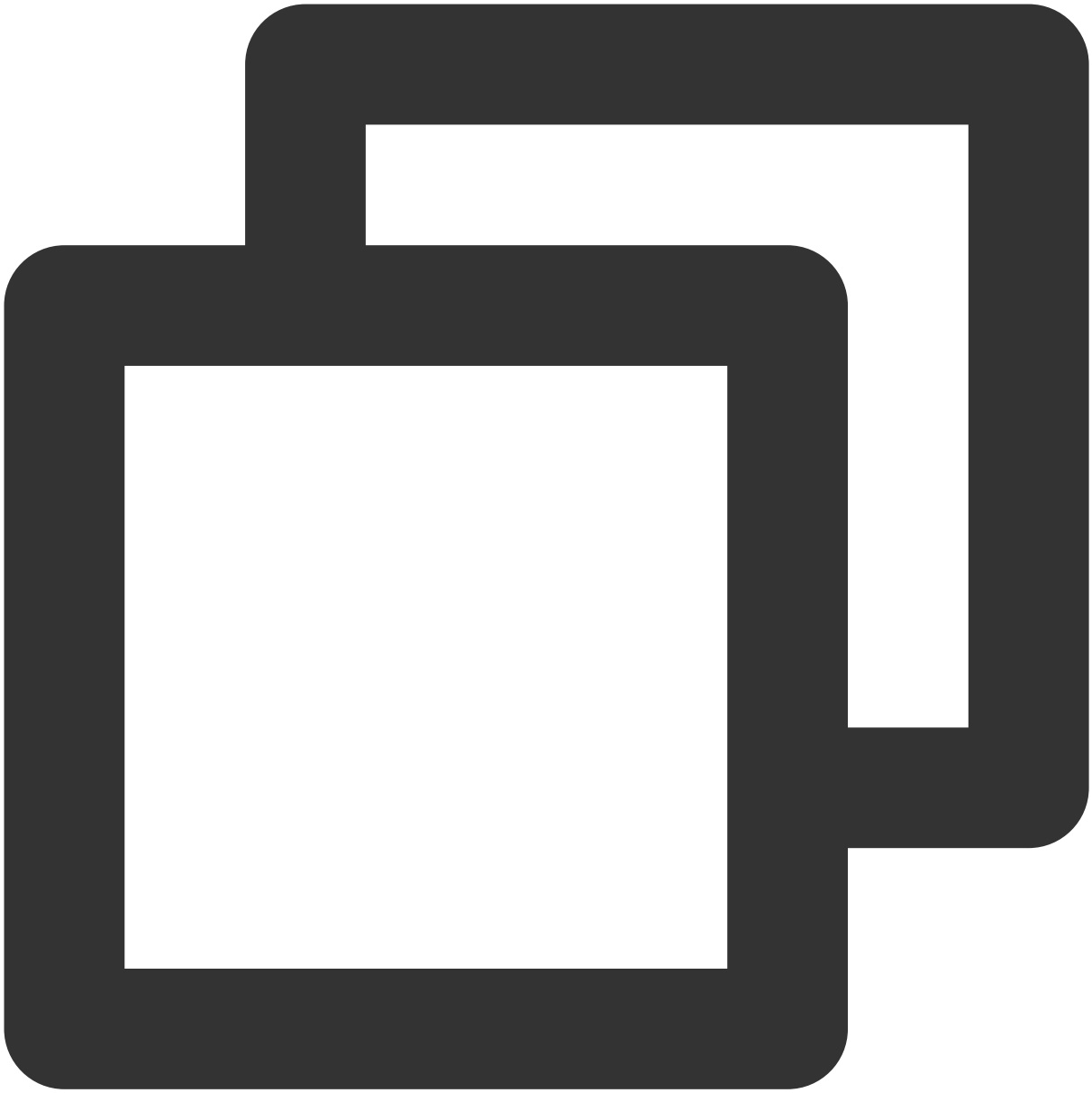


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Segment</Tag>
    <Input>
        <Object>input/demo.mkv</Object>
    </Input>
```

```
    <Operation>
        <Segment>
            <Format>mp4</Format>
            <Duration>15</Duration>
        </Segment>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out-${Number}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应1

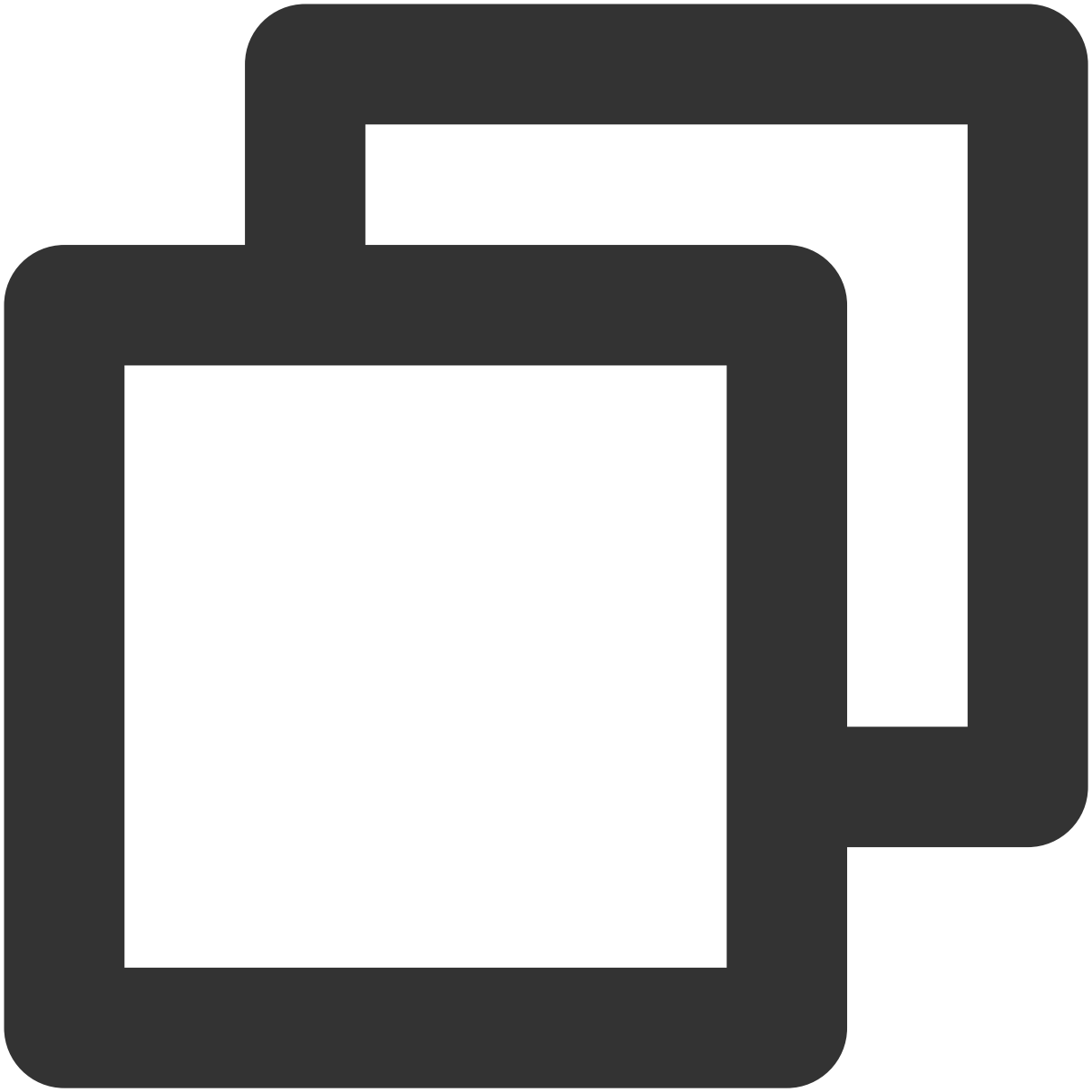


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Segment</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mkv</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Segment>
                <Format>mp4</Format>
                <Duration>15</Duration>
            </Segment>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out-${Number}</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <CallBack>http://callback.demo.com</CallBack>
        <CallBackFormat>JSON<CallBackFormat>
    </JobsDetail>
</Response>
```

请求2：hls转封装并加密

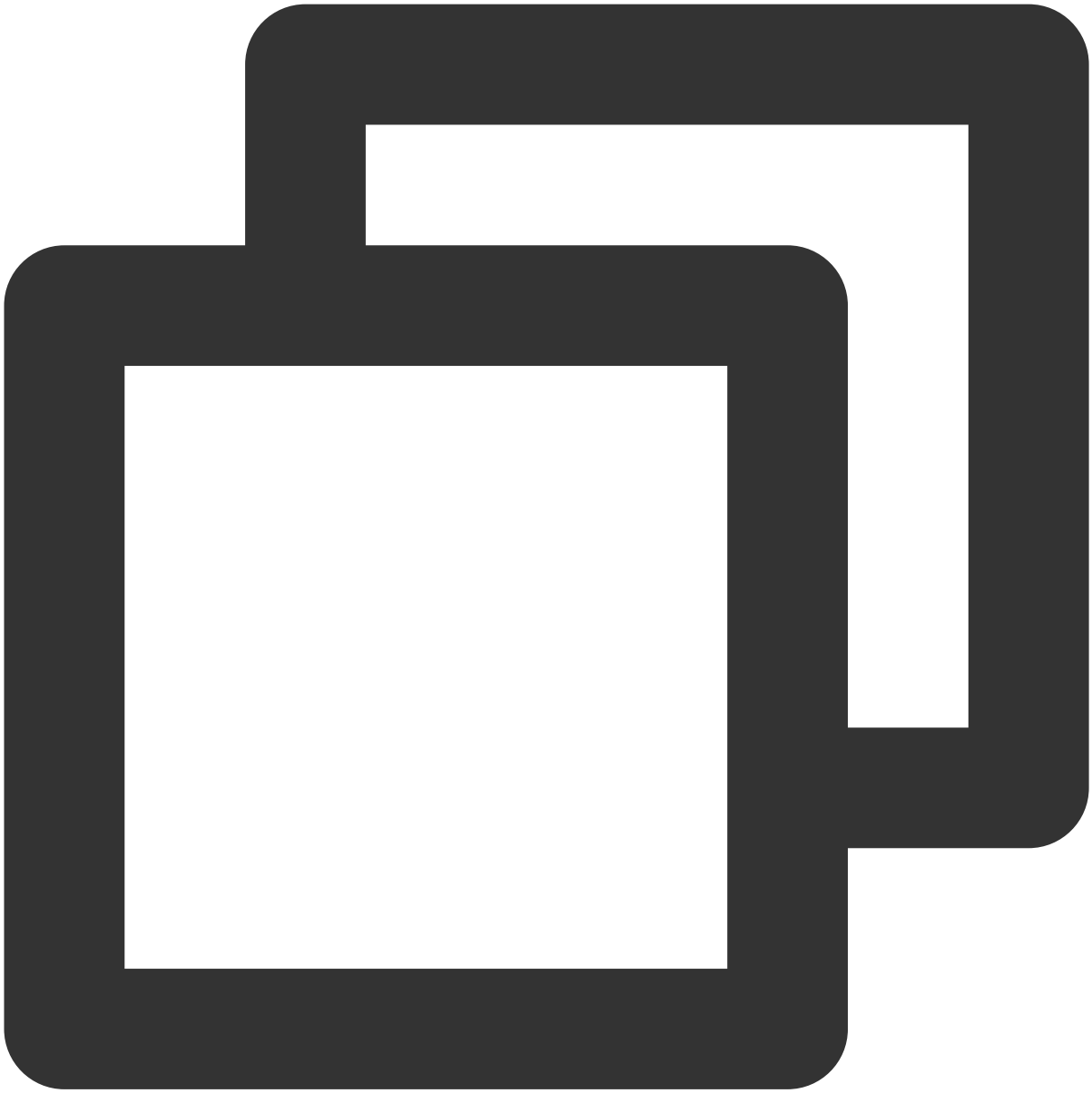


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Segment</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <Segment>
            <Format>hls</Format>
            <Duration>5</Duration>
            <HlsEncrypt>
                <IsHlsEncrypt>true</IsHlsEncrypt>
                <UriKey>test-key</UriKey>
            </HlsEncrypt>
        </Segment>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out-${Number}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应2



```
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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Segment</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Segment>
                <Format>hls</Format>
                <Duration>5</Duration>
                <HlsEncrypt>
                    <IsHlsEncrypt>true</IsHlsEncrypt>
                    <UriKey>test-key</UriKey>
                </HlsEncrypt>
            </Segment>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/out</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <CallBack>http://callback.demo.com</CallBack>
        <CallBackFormat>JSON<CallBackFormat>
    </JobsDetail>
</Response>
```

提交智能封面任务

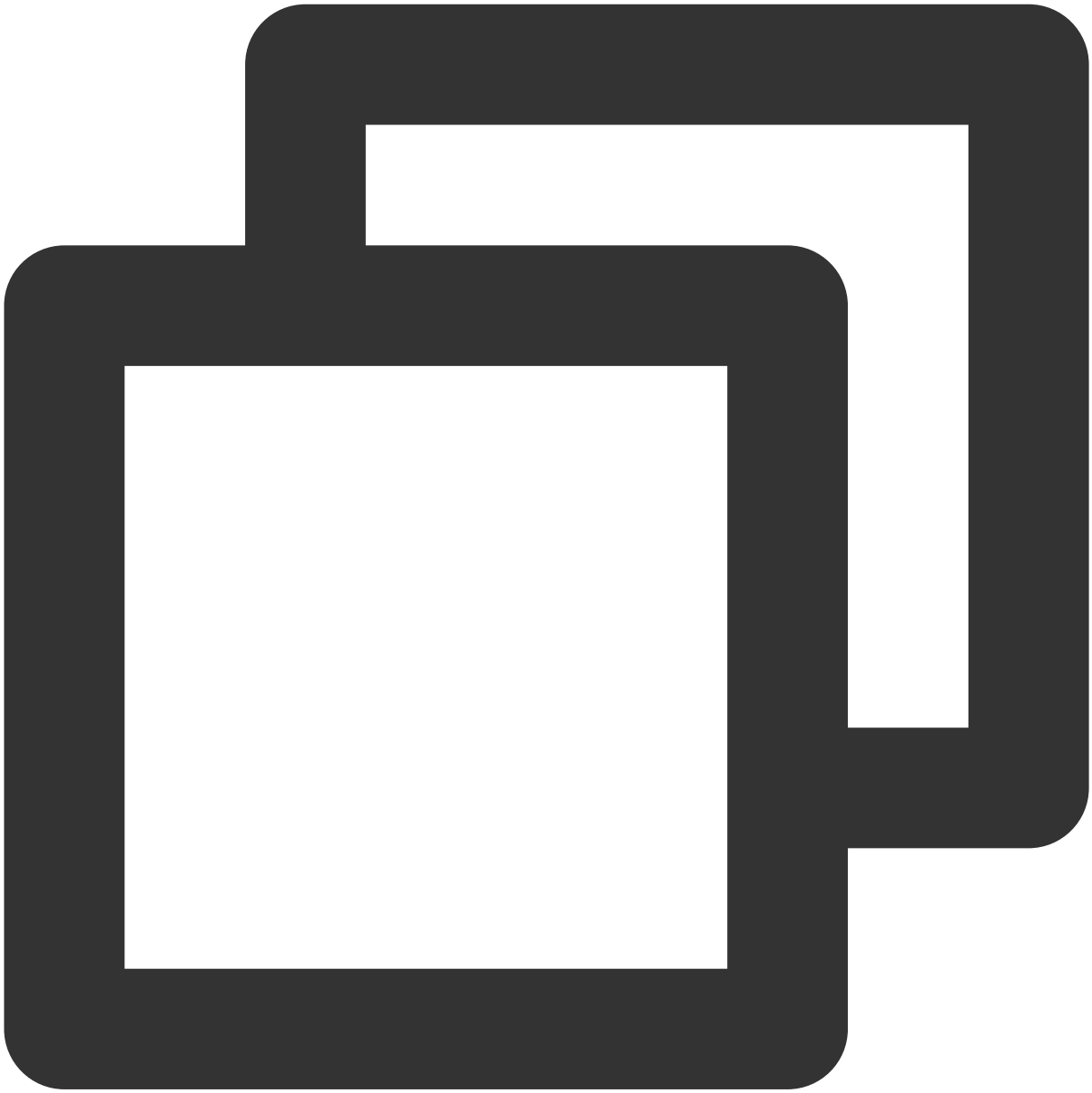
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个智能封面任务。

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：

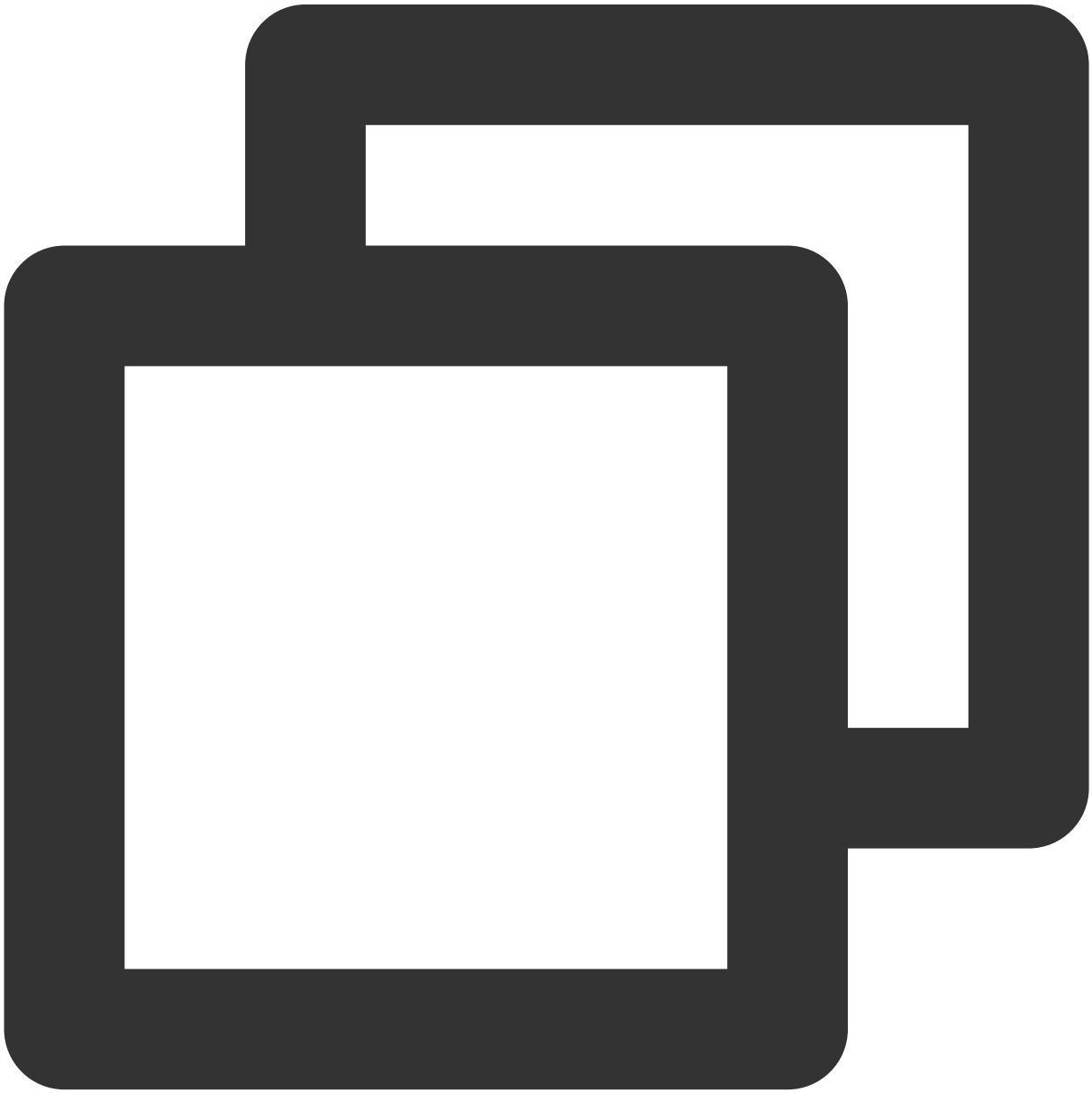
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>SmartCover</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t16567cd58ade9411d952283873123b9b1</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/smartcover-${Number}.jpg</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：SmartCover	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
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
TemplateId	Request.Operation	智能封面模板 ID	String	否
SmartCover	Request.Operation	封面配置	Container	否
Output	Request.Operation	结果输出地址	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

说明：

如果同时设置了 TemplateId 和 SmartCover，则优先使用 TemplateId

Container 类型 SmartCover 的具体数据类型描述如下：

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.Operation.SmartCover	封面图片类型	String	是	无	png、jpg、webp
Width	Request.Operation.SmartCover	封面图片宽度	String	是	无	1. 值范围：[128，4096] 2. 单位：px
Height	Request.Operation.SmartCover	封面图片高度	String	是	无	1. 值范围：[128，4096]

版权所有：腾讯云计算（北京）有限责任公司

						2. 单位：px
Count	Request.Operation.SmartCover	封面数量	String	否	3	1. 值范围：[1，10] 2. 单位：px
DeleteDuplicates	Request.Operation.SmartCover	封面是否去重	String	否	false	true/false

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。 **必须包含 ${Number} 参数。** 如 Object 为 my-new-cover-${Number}.jpg，对应实际3张输出结果时，分别为 my-new-cover-0.jpg my-new-cover-1.jpg my-new-cover-2.jpg	String	是

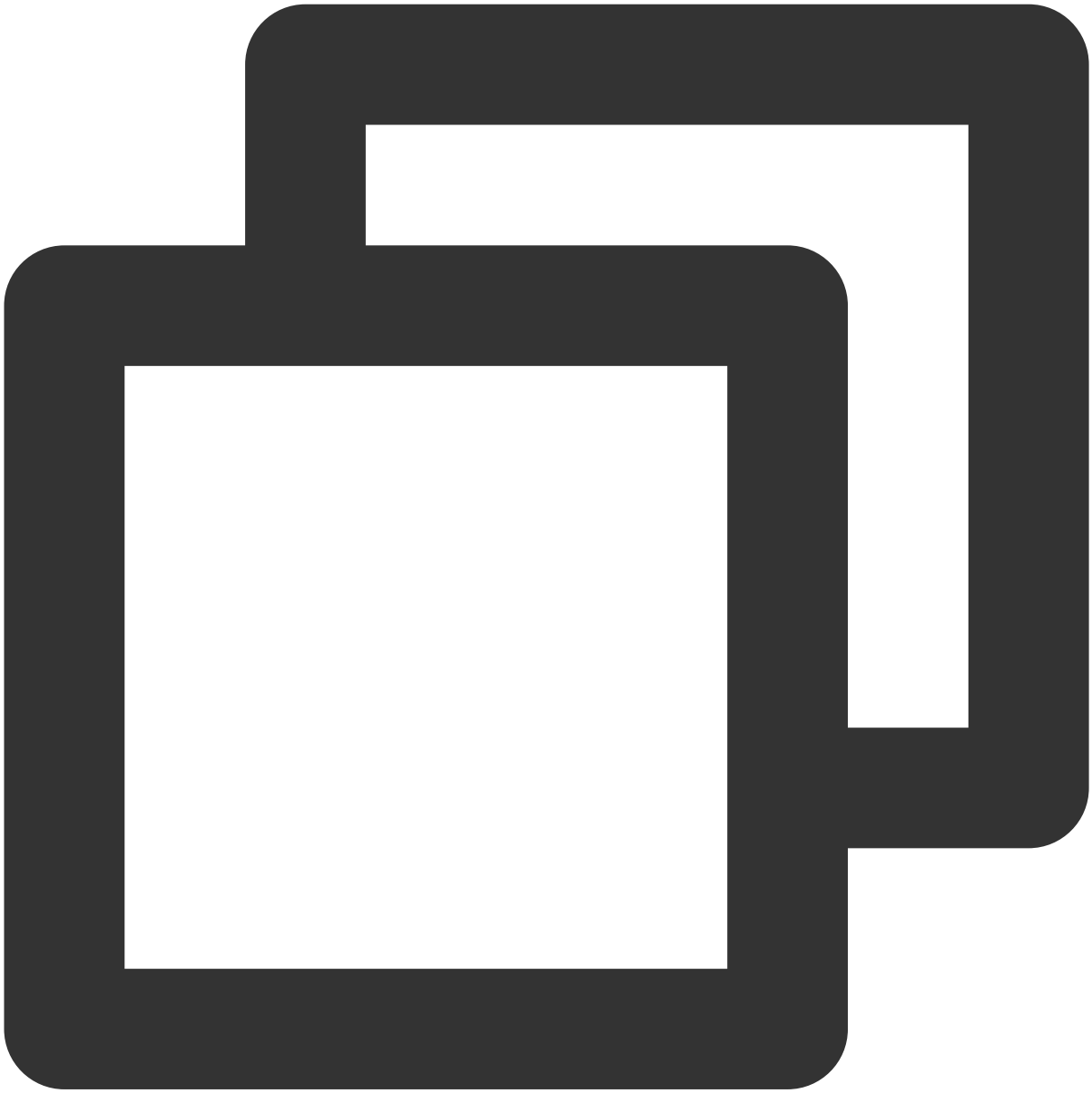
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SmartCover</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t16567cd58ade9411d952283873123b9b1</TemplateId>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</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：SmartCover	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String

版权所有：腾讯云计算（北京）有限责任公司

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

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
SmartCover	Response.JobsDetail.Operation	同请求中的 Request.Operation.SmartCover	String
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	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	输出文件的基本信息	Container

Container 节点 OutputFile 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

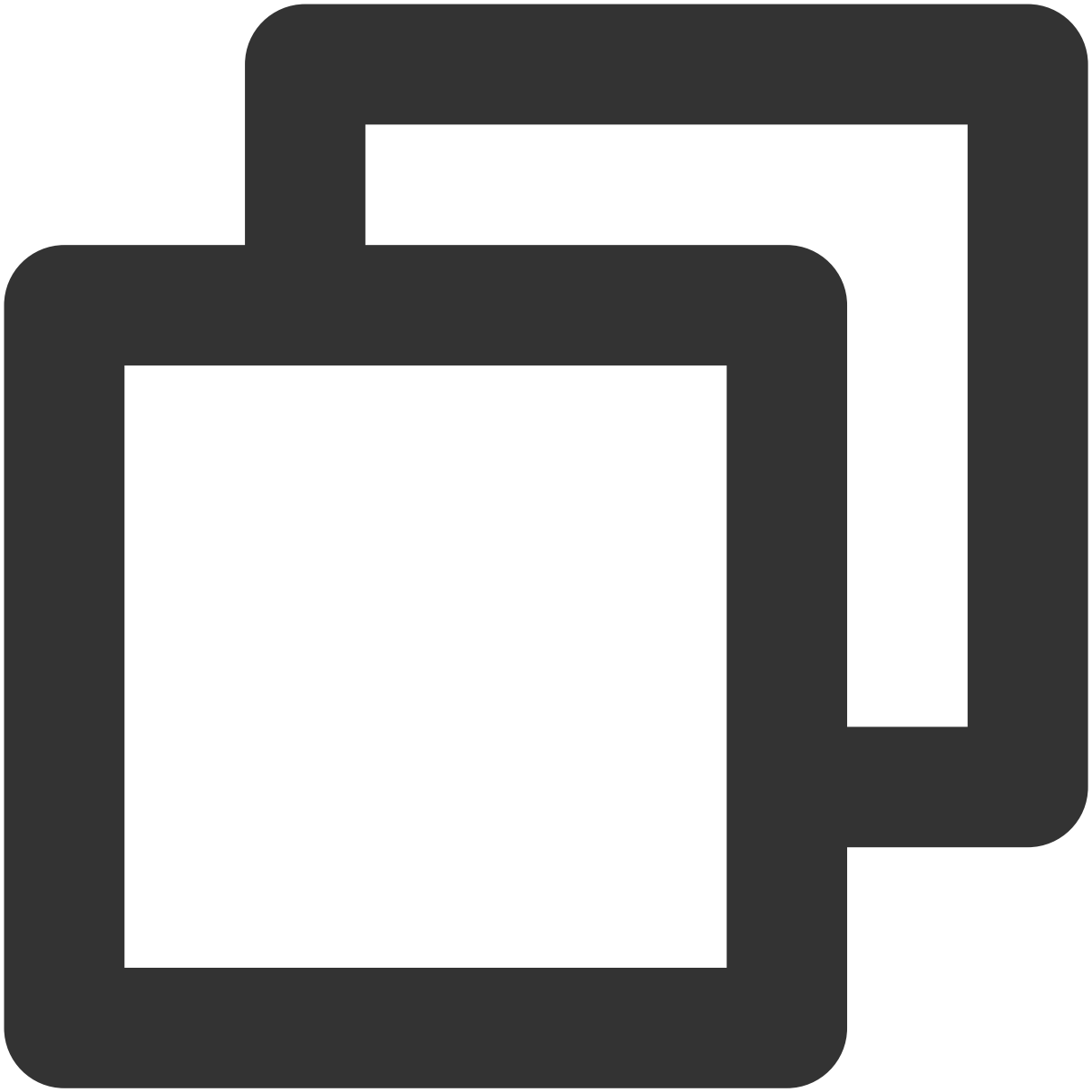
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组

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1：使用智能封面模板 ID

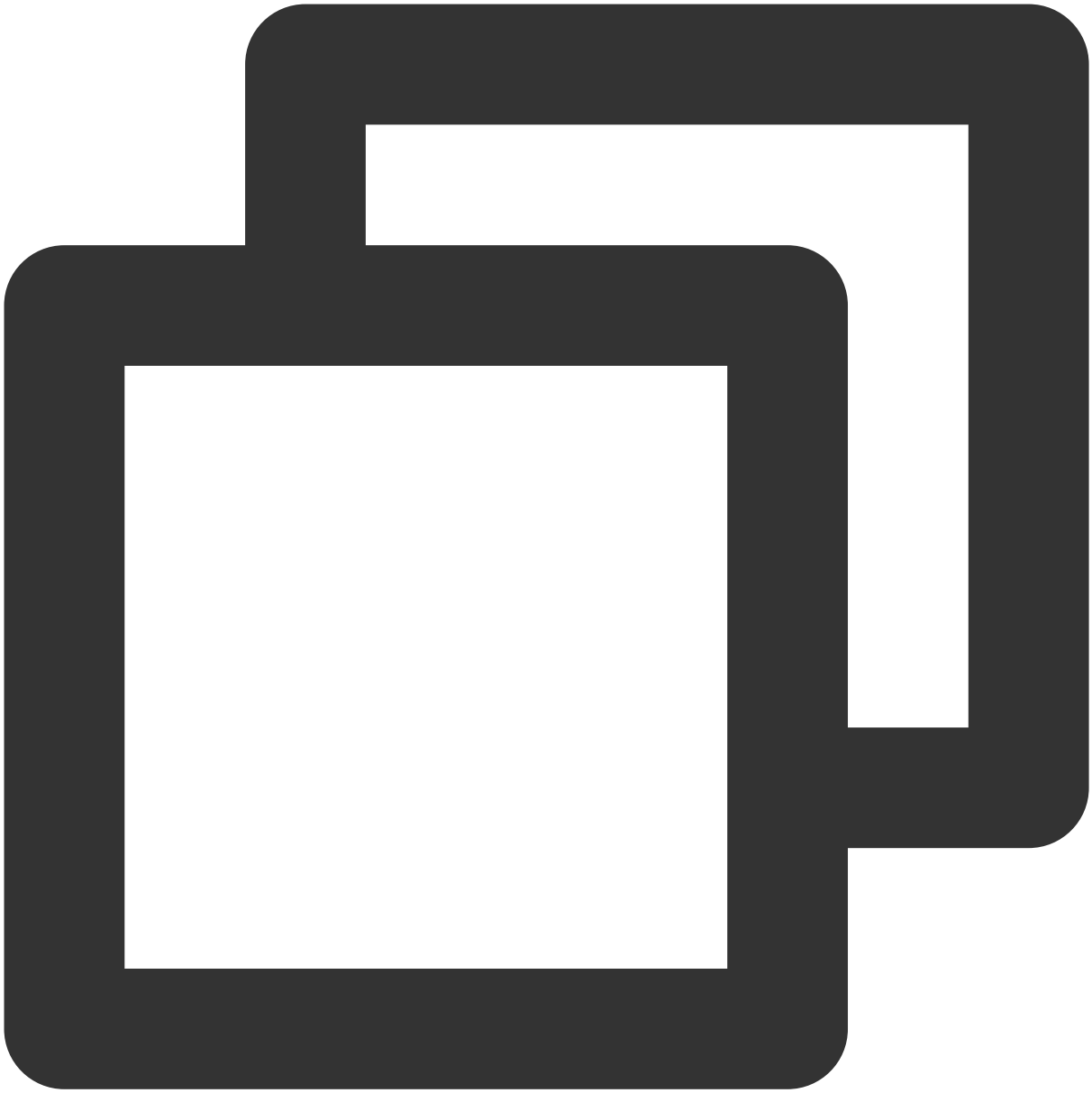


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SmartCover</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t16567cd58ade9411d952283873123b9b1</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/smartcover-${Number}.jpg</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应

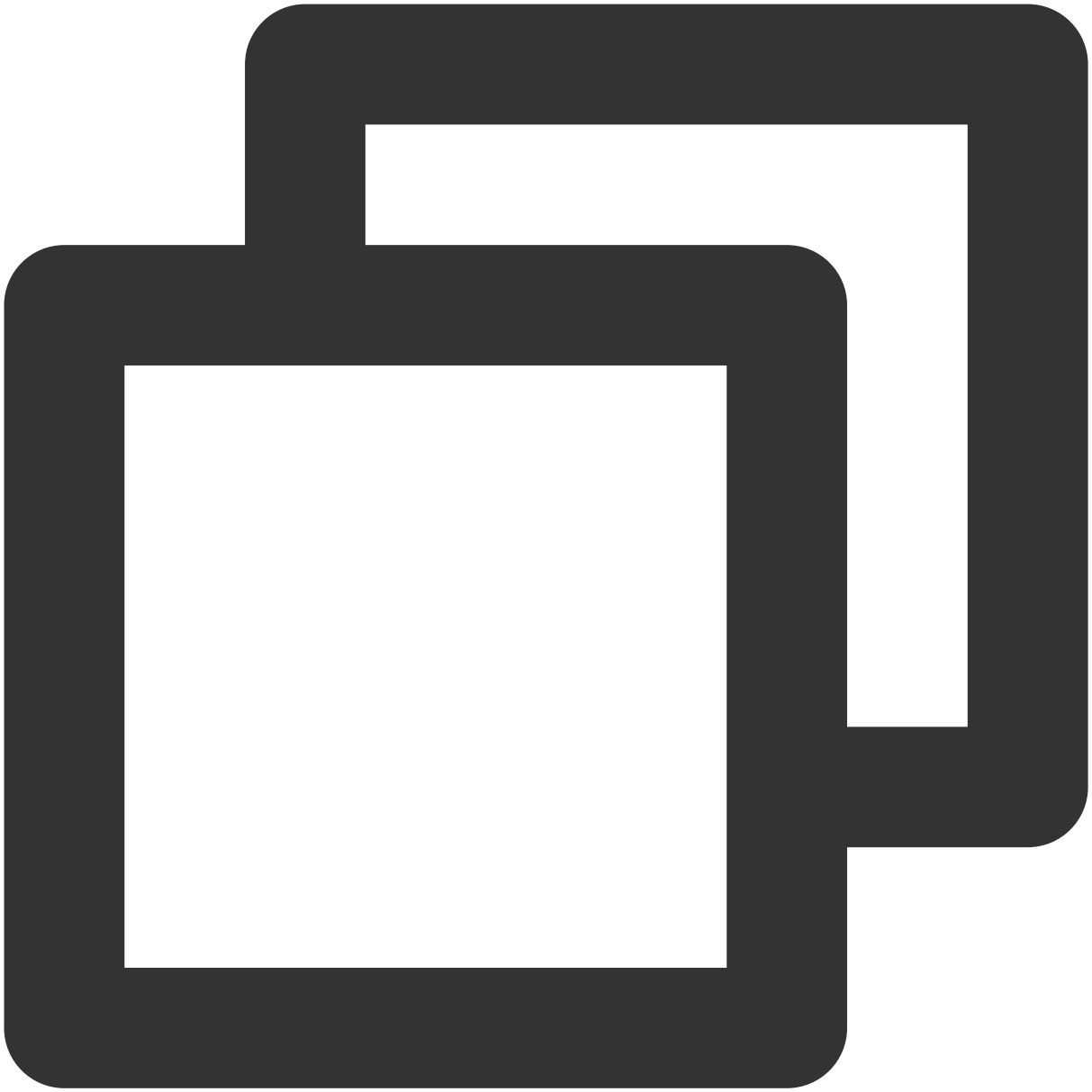


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SmartCover</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t16567cd58ade9411d952283873123b9b1</TemplateId>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/smartcover-${Number}.jpg</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求2：使用智能封面参数

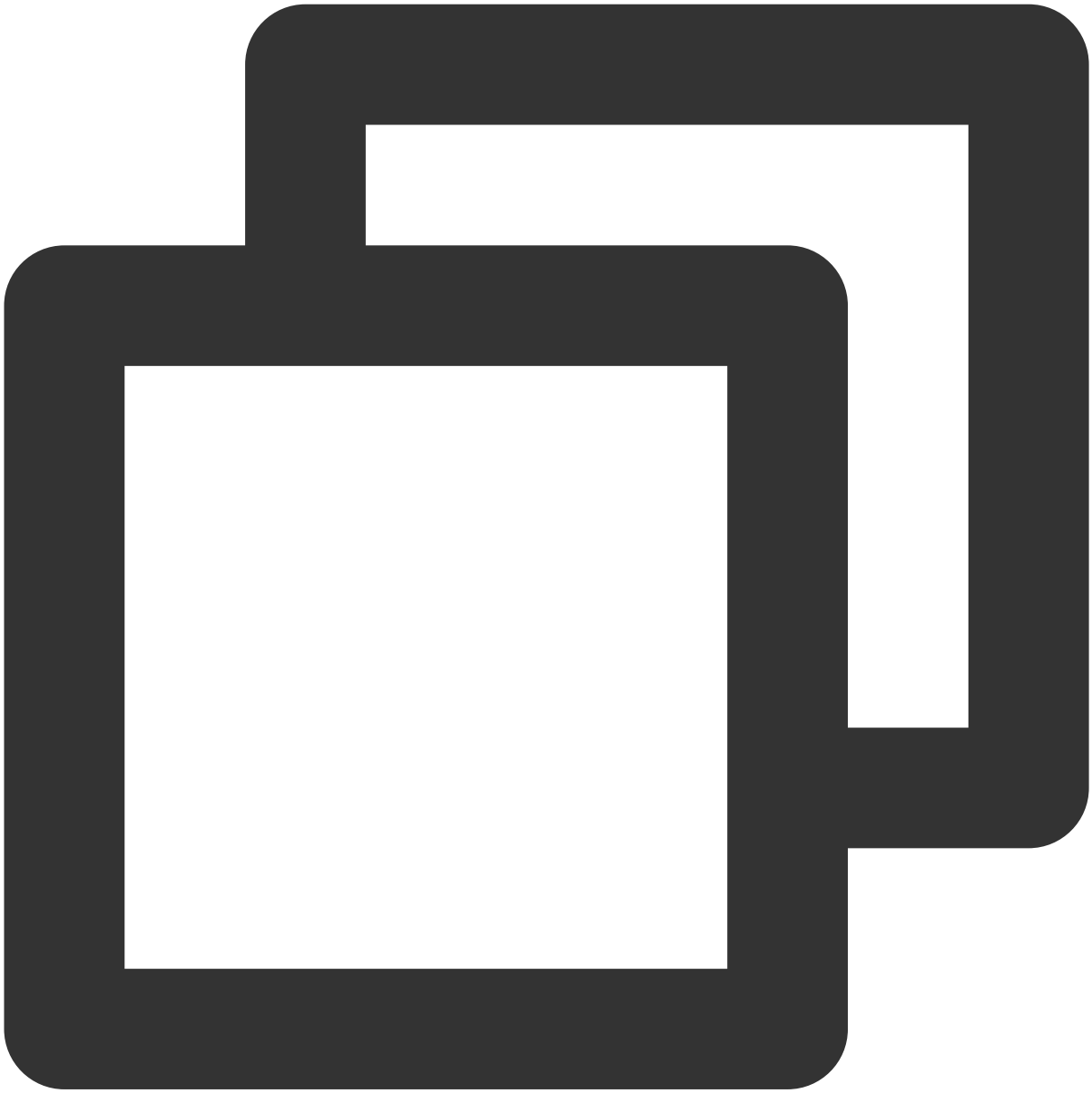


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SmartCover</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <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 data.</UserData>
        <JobLevel>0</JobLevel>
    </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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SmartCover</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <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 data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交视频截帧任务

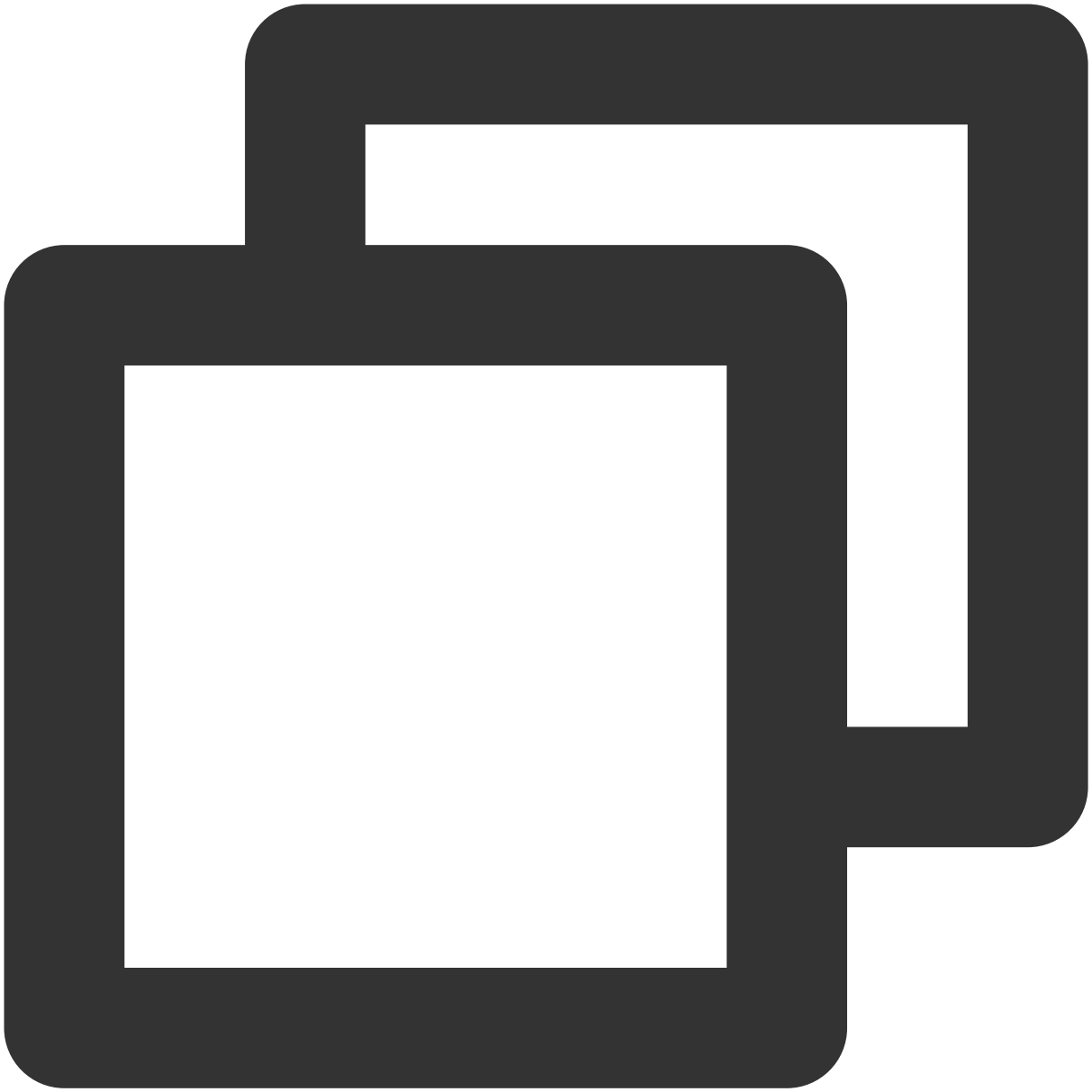
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个截图任务。

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：

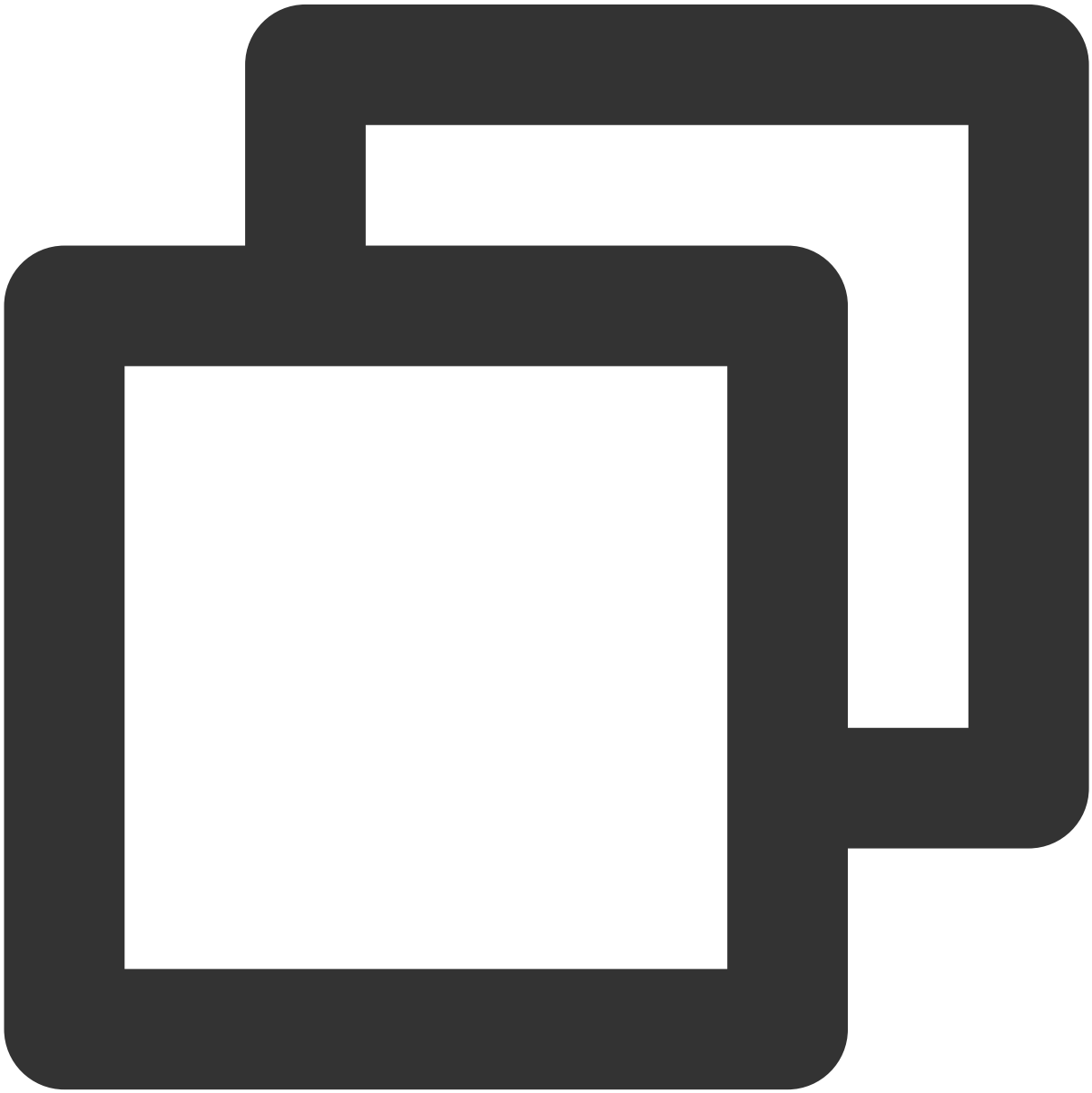
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Snapshot</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/snapshot-${Number}.jpg</Object>
            <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：Snapshot	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
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
Snapshot	Request.Operation	指定该任务的参数，同创建截图模板 CreateMediaTemplate 接口中的 Request.Snapshot	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
Output	Request.Operation	结果输出地址	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

注意：

优先使用 TemplateId，无 TemplateId 时使用 Snapshot。

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。**必须包含 ${Number} 参数。**	String	否

		如 Object 为 snapshot-${Number}.jpg		
SpriteObject	Request.Operation.Output	雪碧图的名字。 **必须包含 ${Number} 参数。** 如 Object 为 sprite-${Number}.jpg 仅支持 jpg 格式	String	否

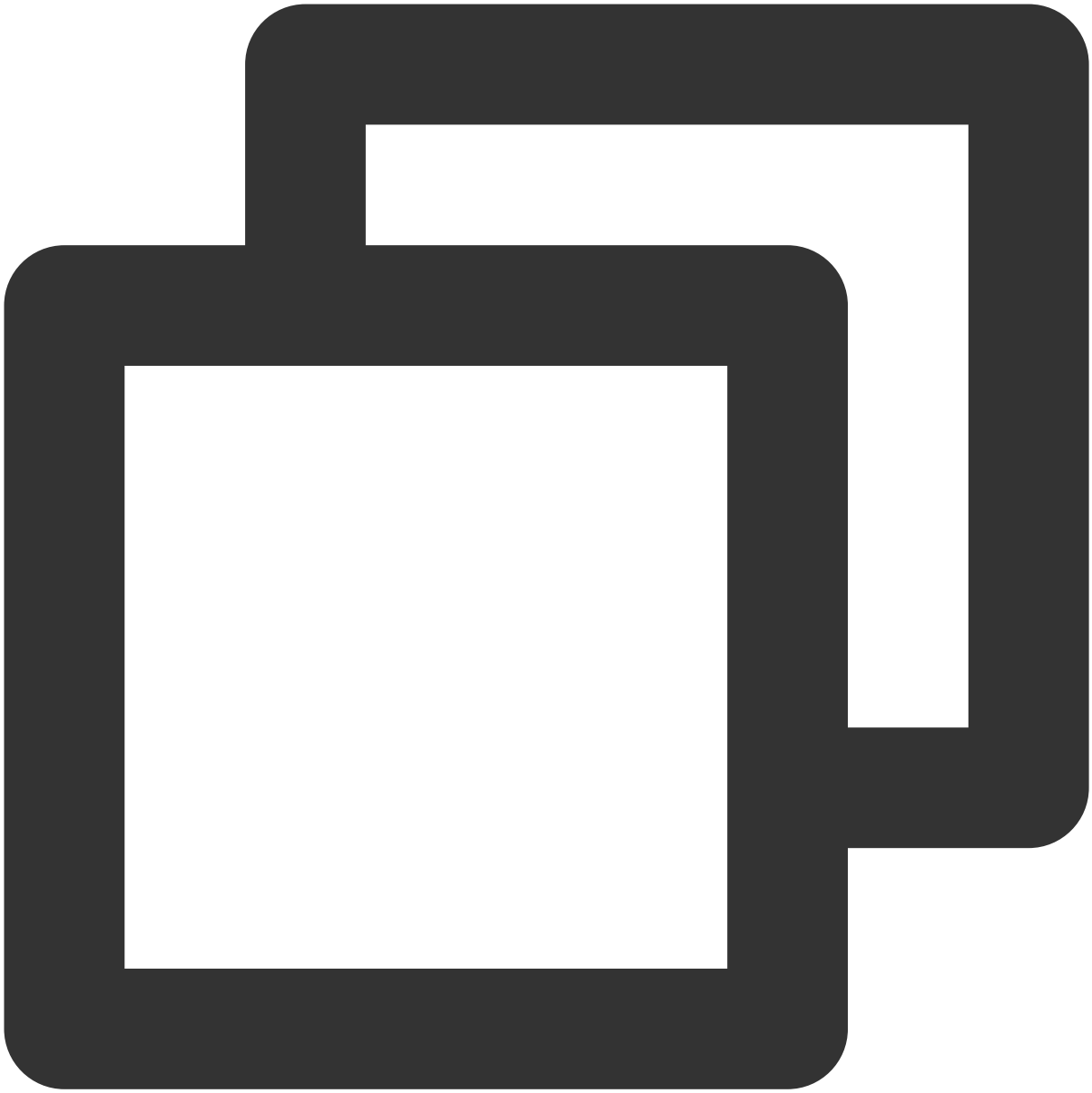
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Snapshot</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>snapshot_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/snapshot-${Number}.jpg</Object>
                <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
            </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：Snapshot	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	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
Snapshot	Response.JobsDetail.Operation	同请求中的 Request.Operation.Snapshot	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

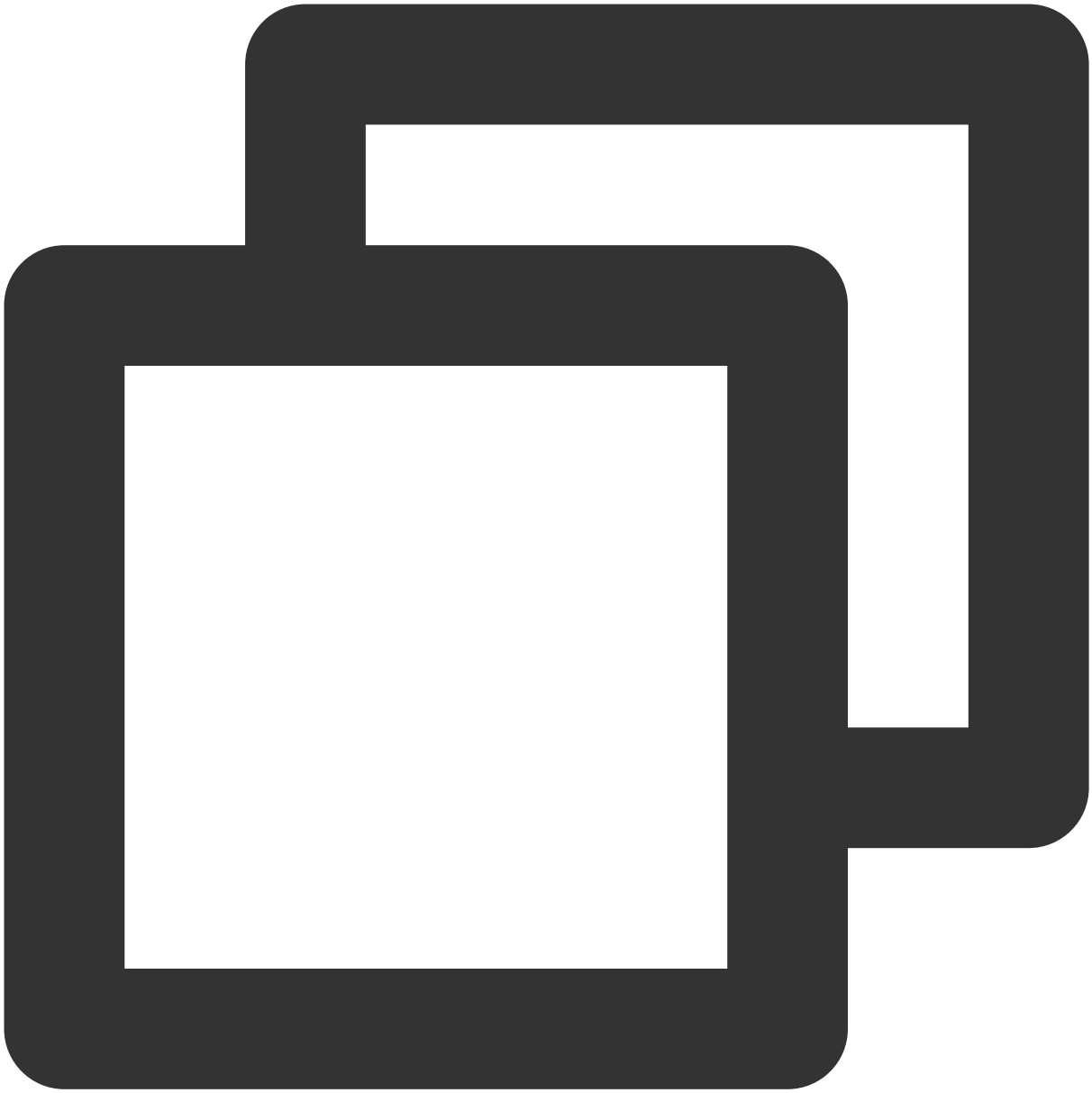
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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：使用截图模板 ID

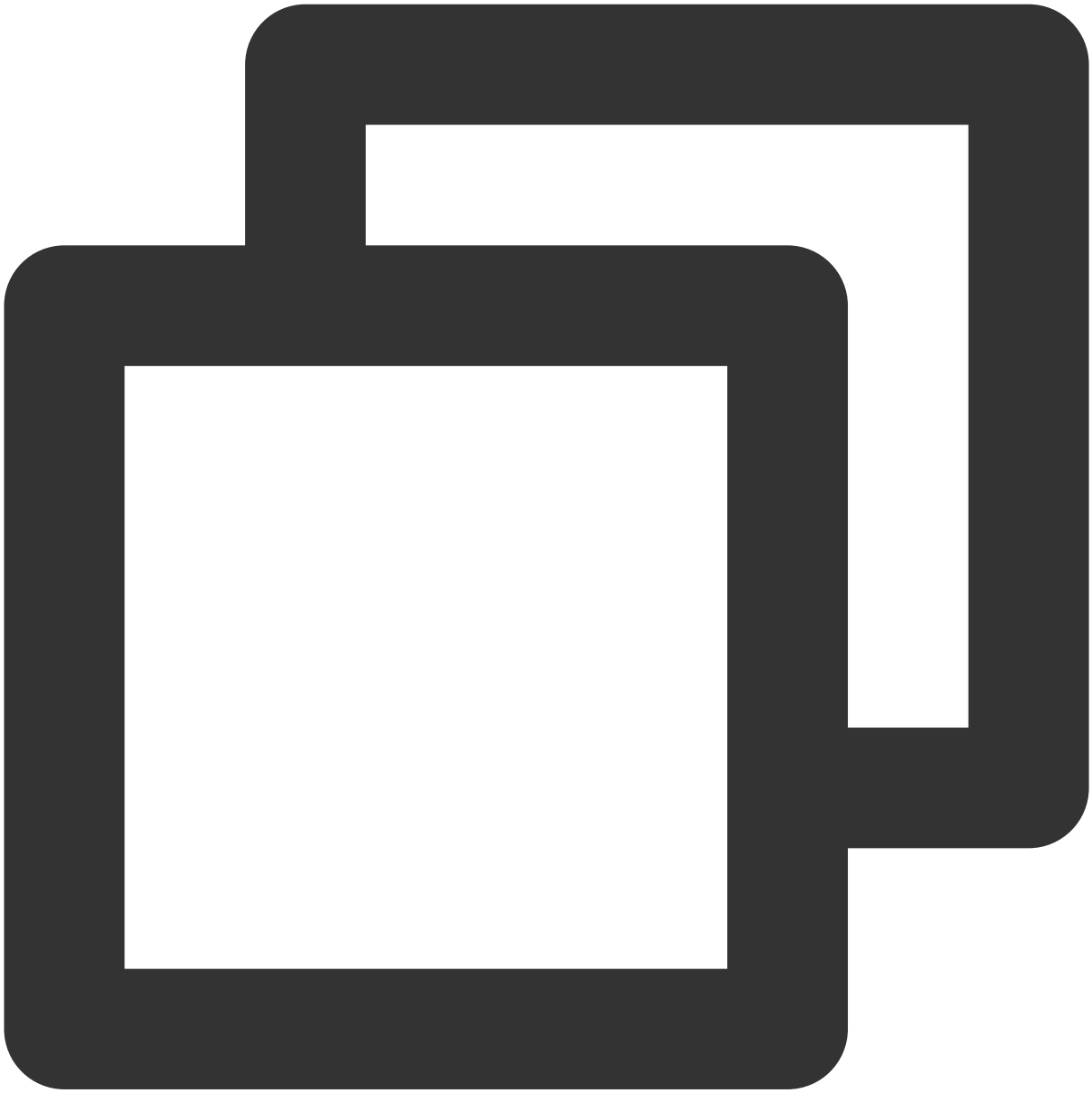


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/snapshot-${Number}.jpg</Object>
            <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应

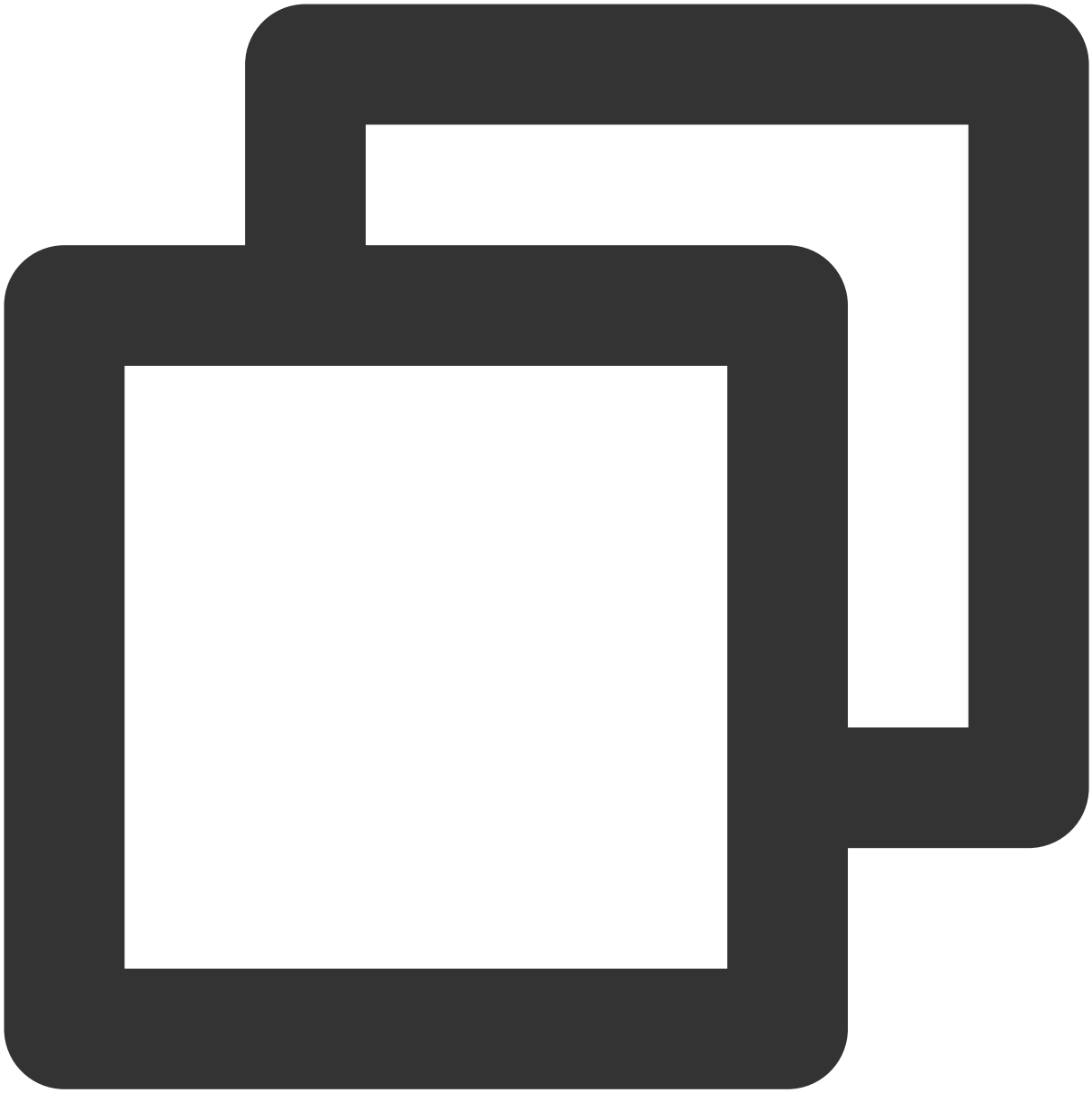


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Snapshot</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>snapshot_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/snapshot-${Number}.jpg</Object>
                <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求2：使用截图处理参数

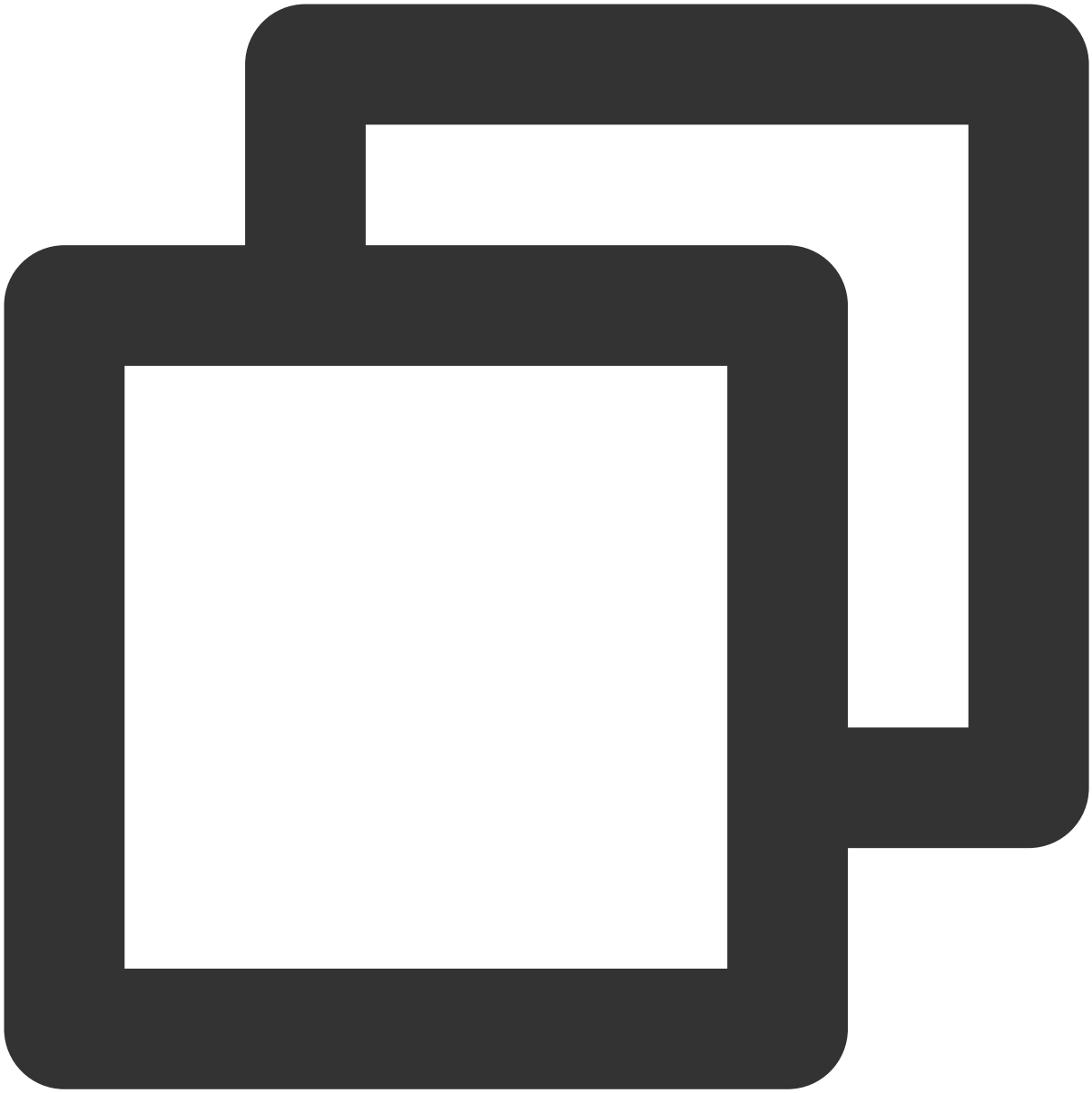


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <Snapshot>
            <BlackLevel>0</BlackLevel>
            <Count>10</Count>
            <IsCheckBlack>false</IsCheckBlack>
            <IsCheckCount>false</IsCheckCount>
            <Mode>Interval</Mode>
            <PixelBlackThreshold>0</PixelBlackThreshold>
            <SnapshotOutMode>SnapshotAndSprite</SnapshotOutMode>
            <SpriteSnapshotConfig>
                <Color>Azure</Color>
                <Columns>3</Columns>
                <Lines>2</Lines>
            </SpriteSnapshotConfig>
            <Start>1</Start>
            <TimeInterval>2</TimeInterval>
        </Snapshot>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/snapshot-${Number}.jpg</Object>
            <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Snapshot</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <Snapshot>
                <BlackLevel>0</BlackLevel>
                <Count>10</Count>
                <IsCheckBlack>false</IsCheckBlack>
                <IsCheckCount>false</IsCheckCount>
                <Mode>Interval</Mode>
                <PixelBlackThreshold>0</PixelBlackThreshold>
                <SnapshotOutMode>SnapshotAndSprite</SnapshotOutMode>
                <SpriteSnapshotConfig>
                    <Color>Azure</Color>
                    <Columns>3</Columns>
                    <Lines>2</Lines>
                </SpriteSnapshotConfig>
                <Start>1</Start>
                <TimeInterval>2</TimeInterval>
            </Snapshot>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/snapshot-${Number}.jpg</Object>
                <SpriteObject>output/sprite-${Number}.jpg</SpriteObject>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交流分离任务

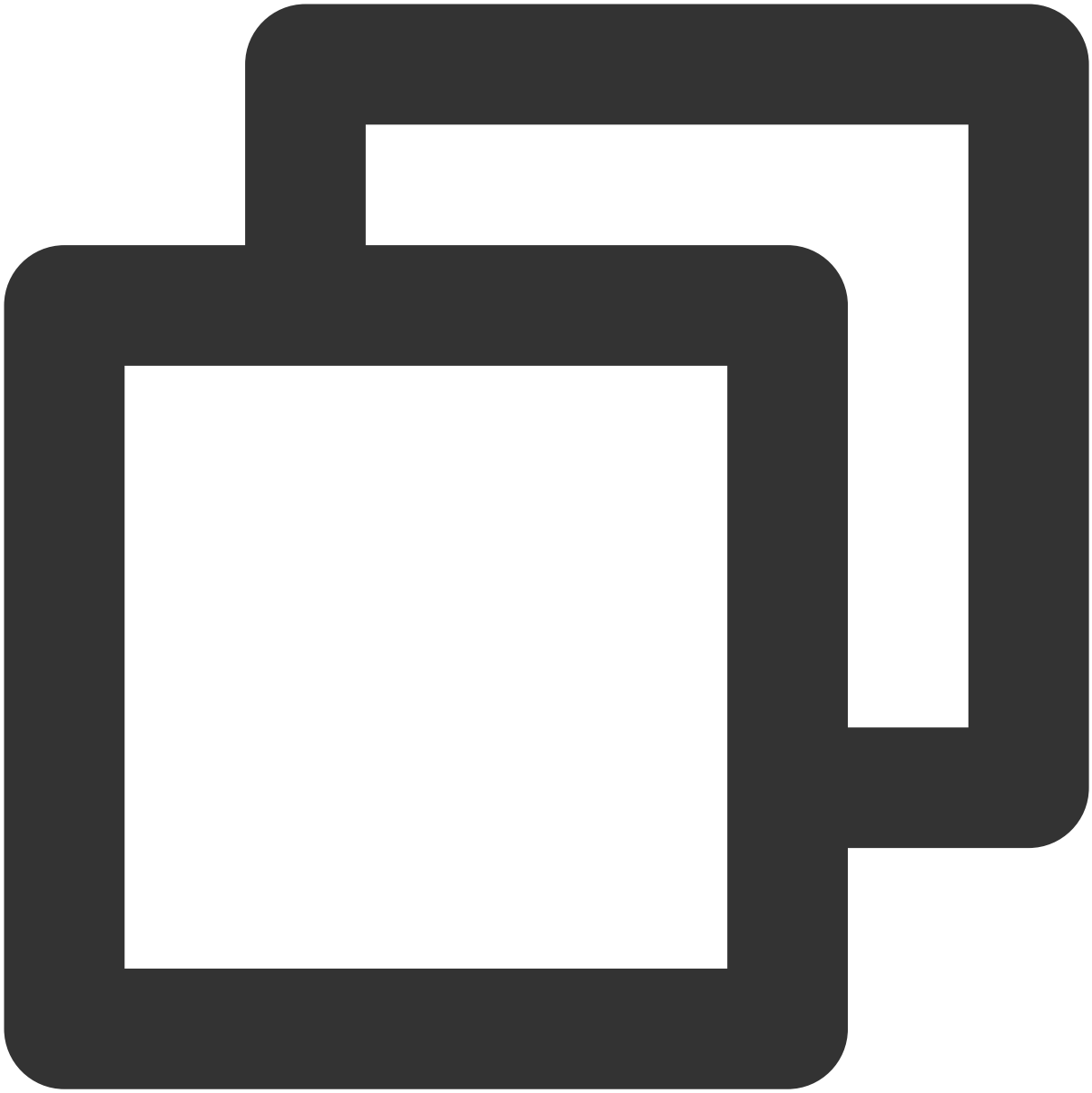
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个音视频流分离任务。

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：

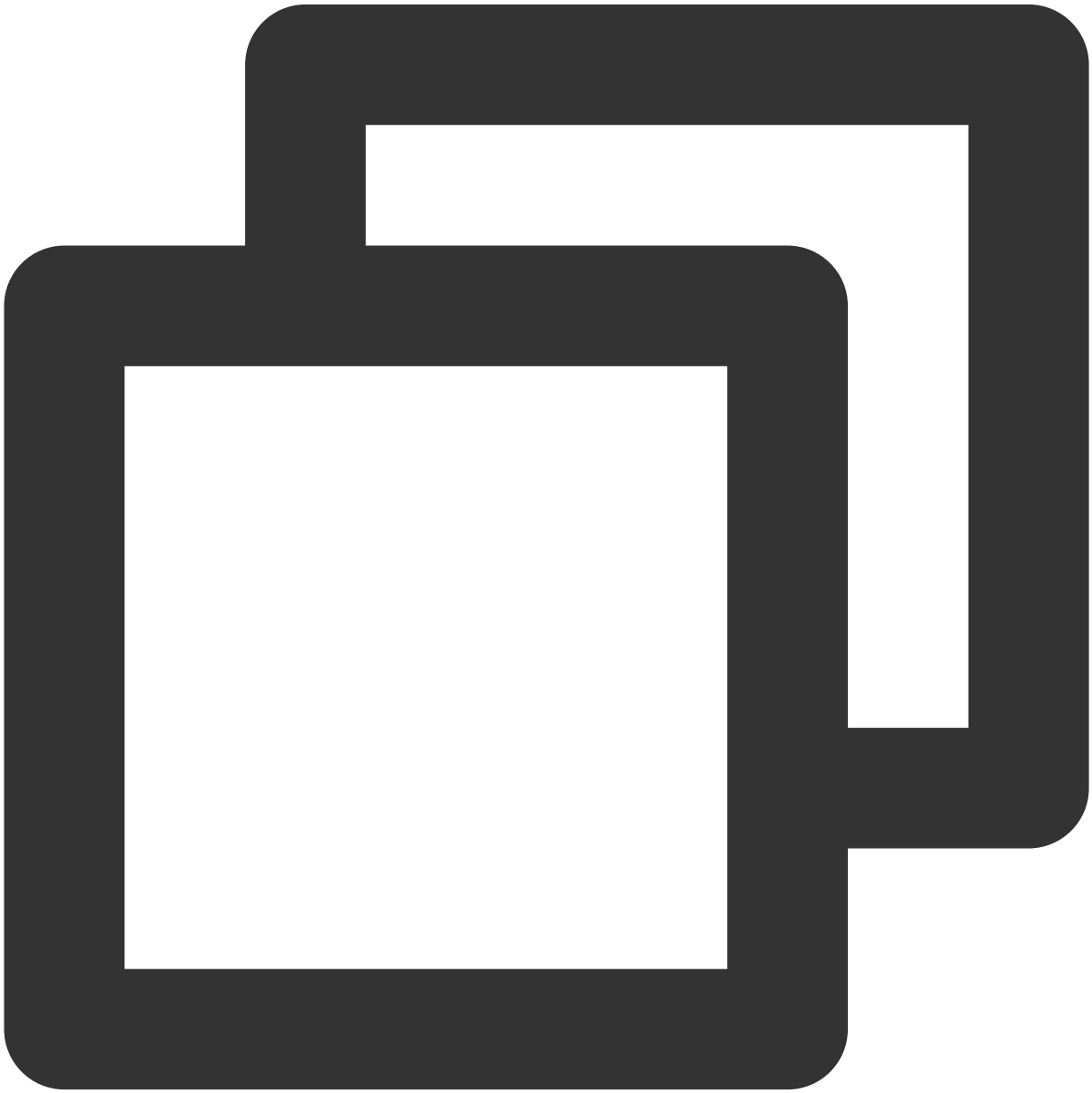
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>StreamExtract</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <StreamExtract>
                <Index>0</Index>
                <Object>output/out0.mp4</Object>
            </StreamExtract>
            <StreamExtract>
                <Index>1</Index>
                <Object>output/out1.mp4</Object>
            </StreamExtract>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Output>
    </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：StreamExtract	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
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
Object	Request.Input	媒体文件名	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
Output	Request.Operation	结果输出地址	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 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
StreamExtract	Request.Operation.Output	结果文件配置信息，最多支持10个流配置	Container	是

Container 类型 StreamExtract 的具体数据描述如下：

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）			
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

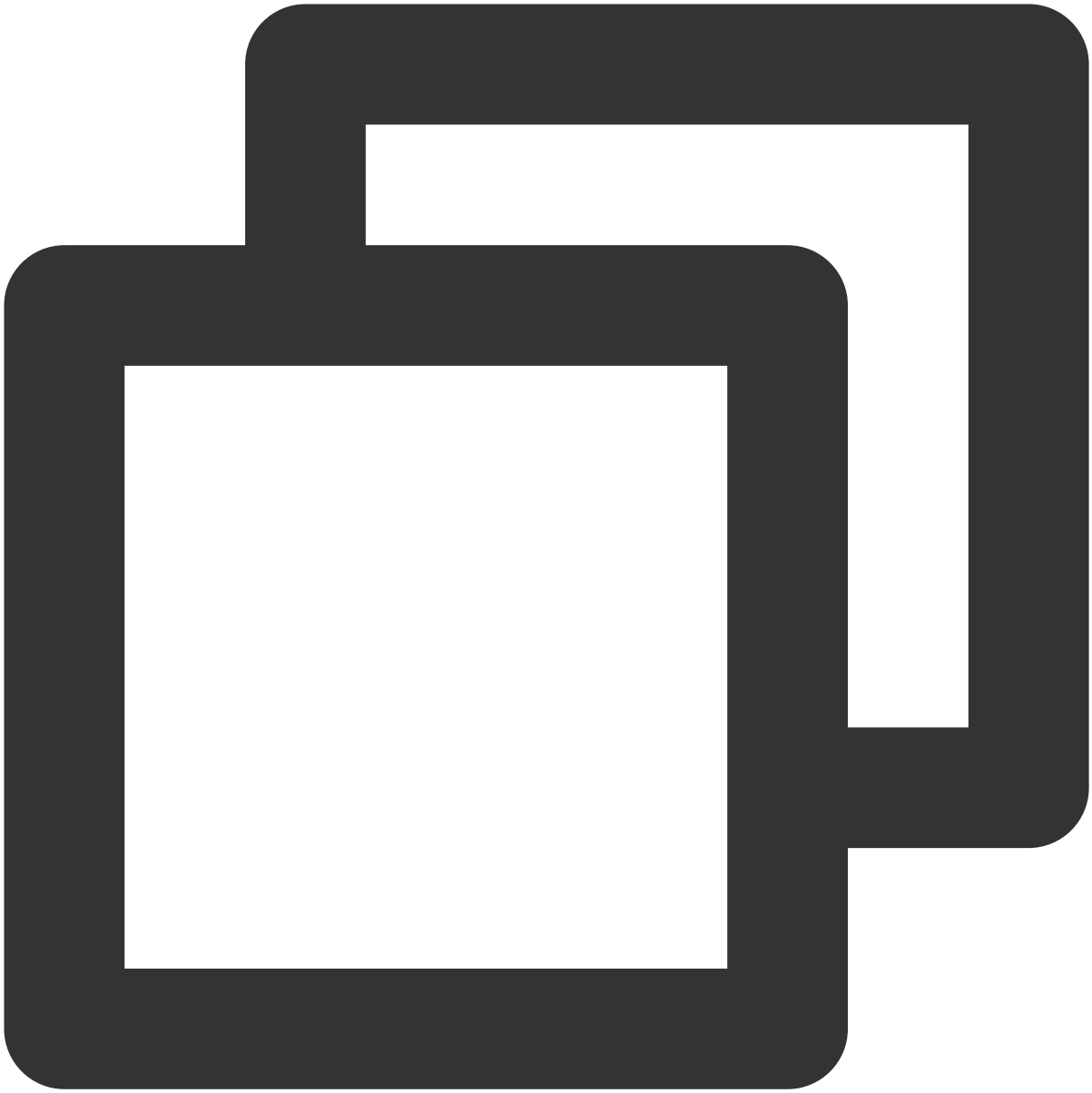
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:11+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>StreamExtract</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <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>
            </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：StreamExtract	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
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
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	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	输出文件的基本信息	Container

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

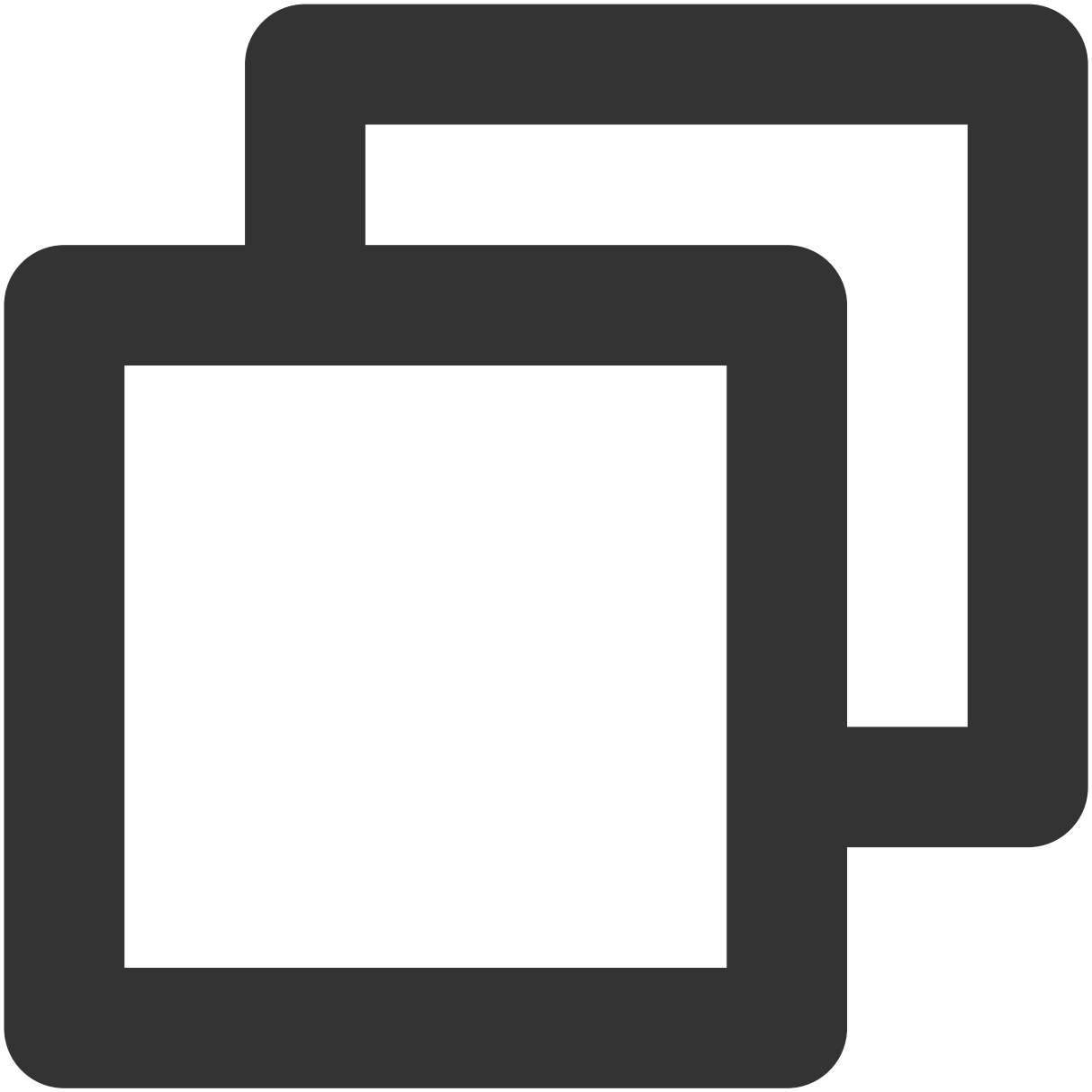
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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

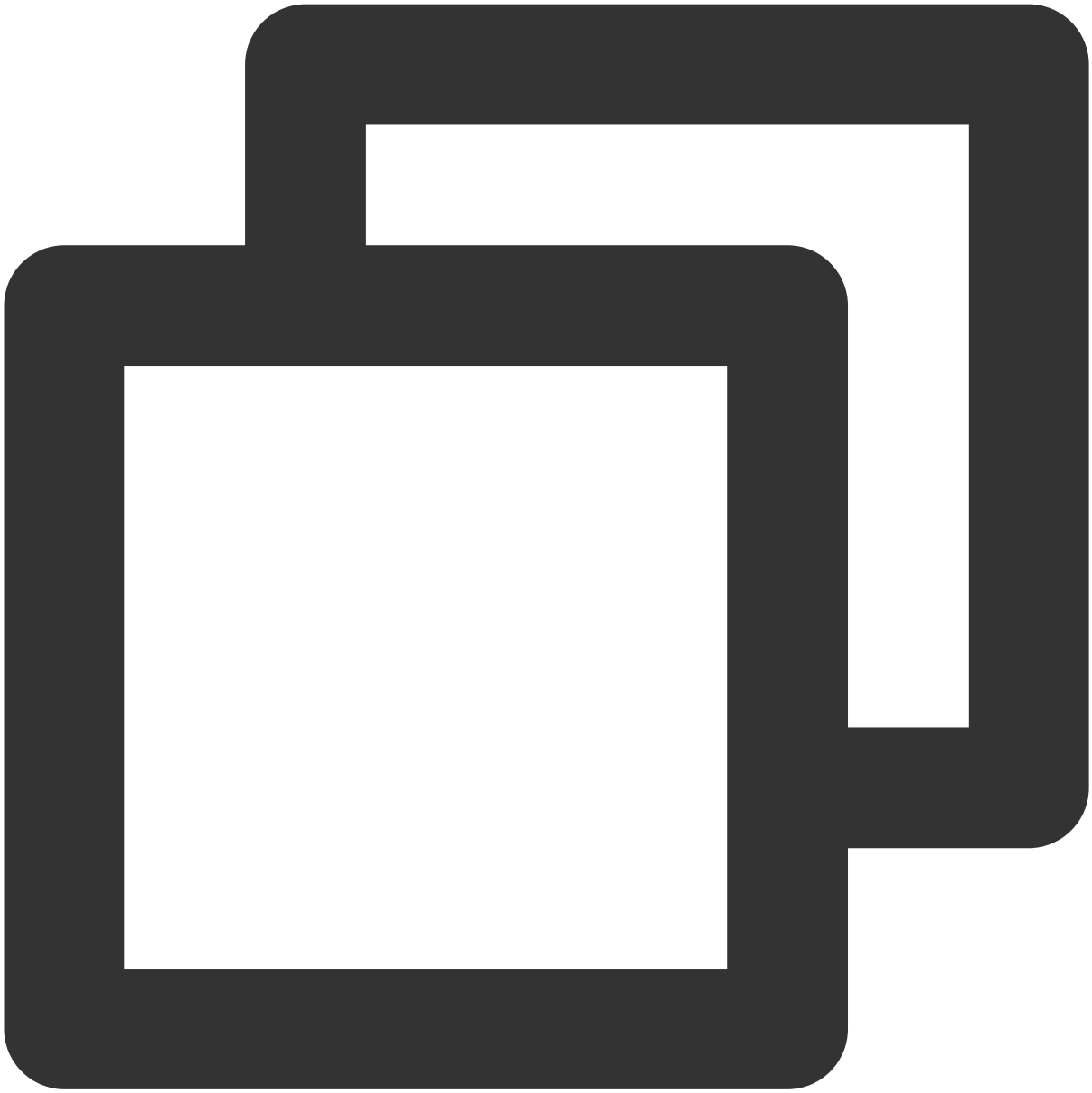


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>StreamExtract</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <StreamExtract>
                <Index>0</Index>
                <Object>output/out0.mp4</Object>
            </StreamExtract>
            <StreamExtract>
                <Index>1</Index>
                <Object>output/out1.mp4</Object>
            </StreamExtract>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Output>
    </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: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:11+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>StreamExtract</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <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>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交超分辨率任务

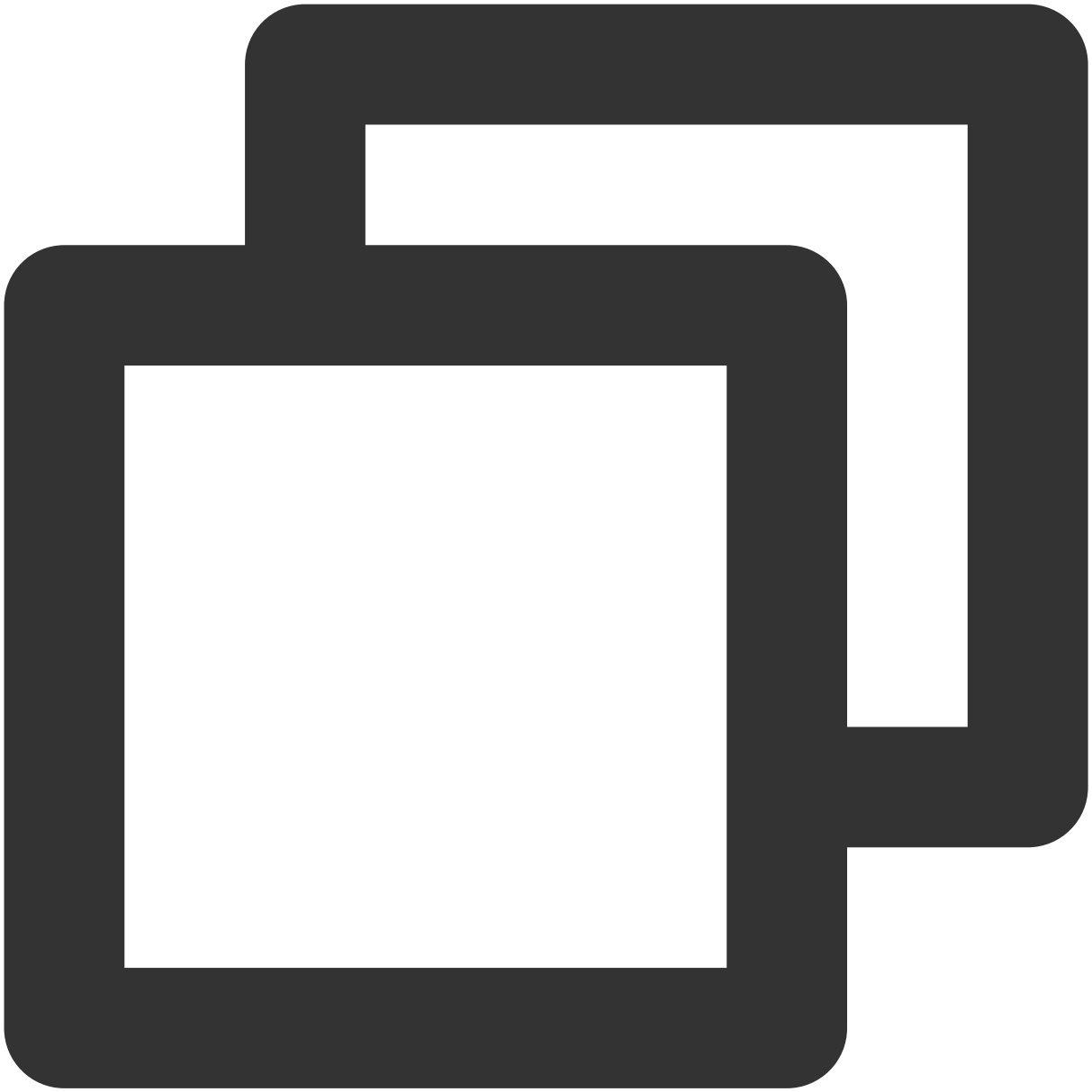
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个超分辨率任务。

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：

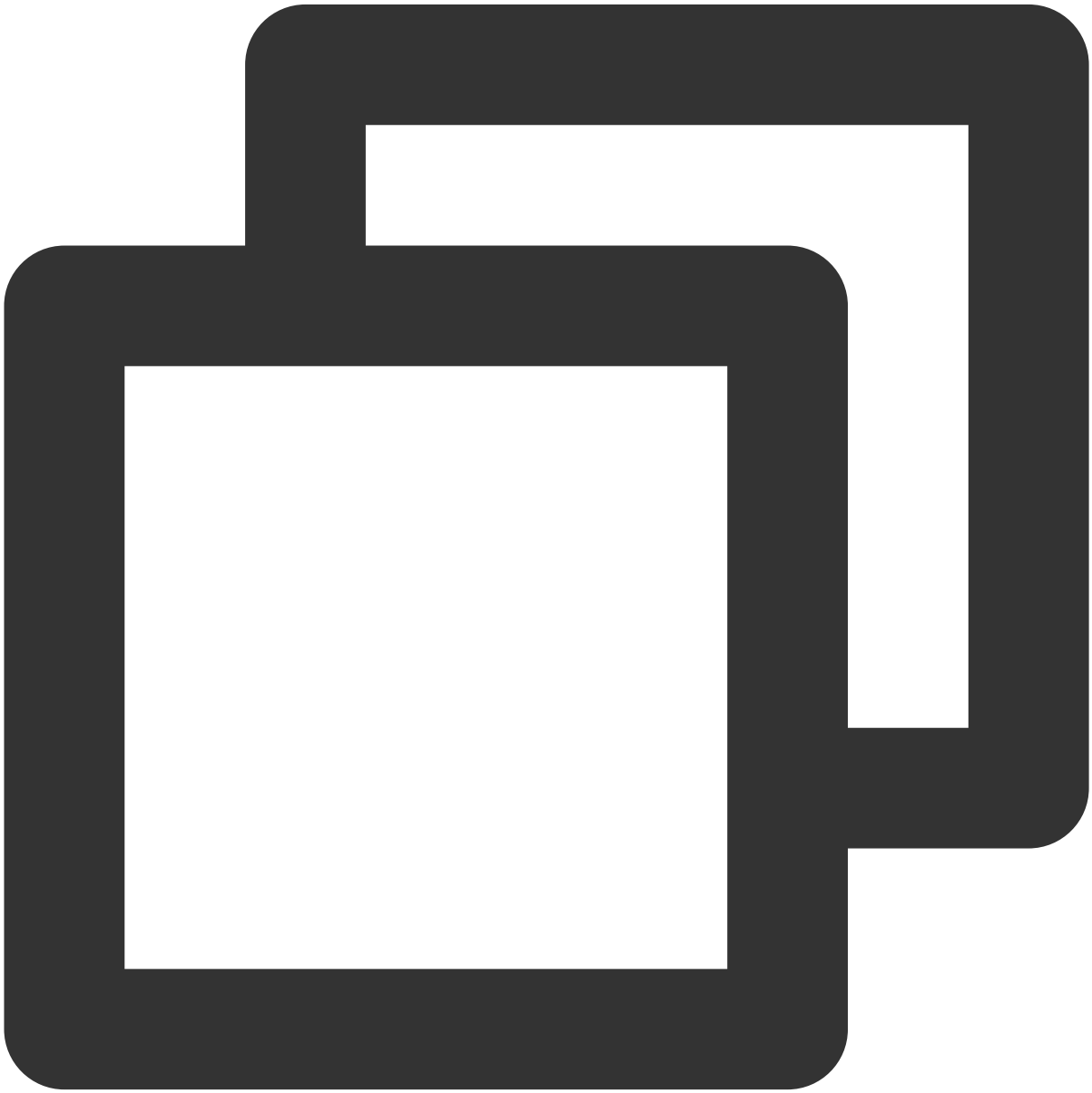
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>SuperResolution</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplateI
        <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemplateI
        <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTemplateI
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <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：SuperResolution	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

版权所有：腾讯云计算（北京）有限责任公司

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
Object	Request.Input	媒体文件名	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
SuperResolution	Request.Operation	指定超分辨率模板参数，不能与 TemplateId 同时为空	Container	否
TemplateId	Request.Operation	指定的超分辨率模 ID，优先使用模板 ID，不能与 SuperResolution 同时为空	String	否
Transcode	Request.Operation	指定转码模板参数，不能与 TranscodeTemplateId 同时为空	Container	否
TranscodeTemplateId	Request.Operation	指定的转码模板 ID，优先使用模板 ID，不能与 Transcode 同时为空	String	否
Watermark	Request.Operation	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口中的 Request.Watermark，最多传3个	Container 数组	否
WatermarkTemplateId	Request.Operation	指定的水印模板 ID，可以传多个水印模板 ID，最多传3个，优先使用模板 ID	String 数组	否
DigitalWatermark	Request.Operation	指定数字水印参数	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Output	Request.Operation	结果输出地址	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

说明：

提交超分辨率任务必须传入转码参数。对于超分辨率参数，优先使用 TemplateId，无 TemplateId 时使用 SuperResolution；对于转码参数，优先使用 TranscodeTemplateId，无 TranscodeTemplateId 时使用 Transcode；对于水印参数，可以使用 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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	必选
Resolution	Request.Operation.SuperResolution	同创建超分辨率模板 CreateMediaTemplate 接口中的 Request.Resolution	String	是
EnableScaleUp	Request.Operation.SuperResolution	同创建超分辨率模板 CreateMediaTemplate 接口中的 Request.EnableScaleUp	String	否
Version	Request.Operation.SuperResolution	同创建超分辨率模板 CreateMediaTemplate 接口中的 Request.Version	String	否

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	必选
TimeInterval	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Video	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Audio	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Audio	Container	否
TransConfig	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TransConfig	Container	否
AudioMix	Request.Operation.Transcode	混音参数，详情见 AudioMix	Container 数组	否

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.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.DigitalWatermark	水印类型，当前仅可设置为 Text	String	是
Version	Request.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String	是
IgnoreError	Request.Operation.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 false	string	否

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

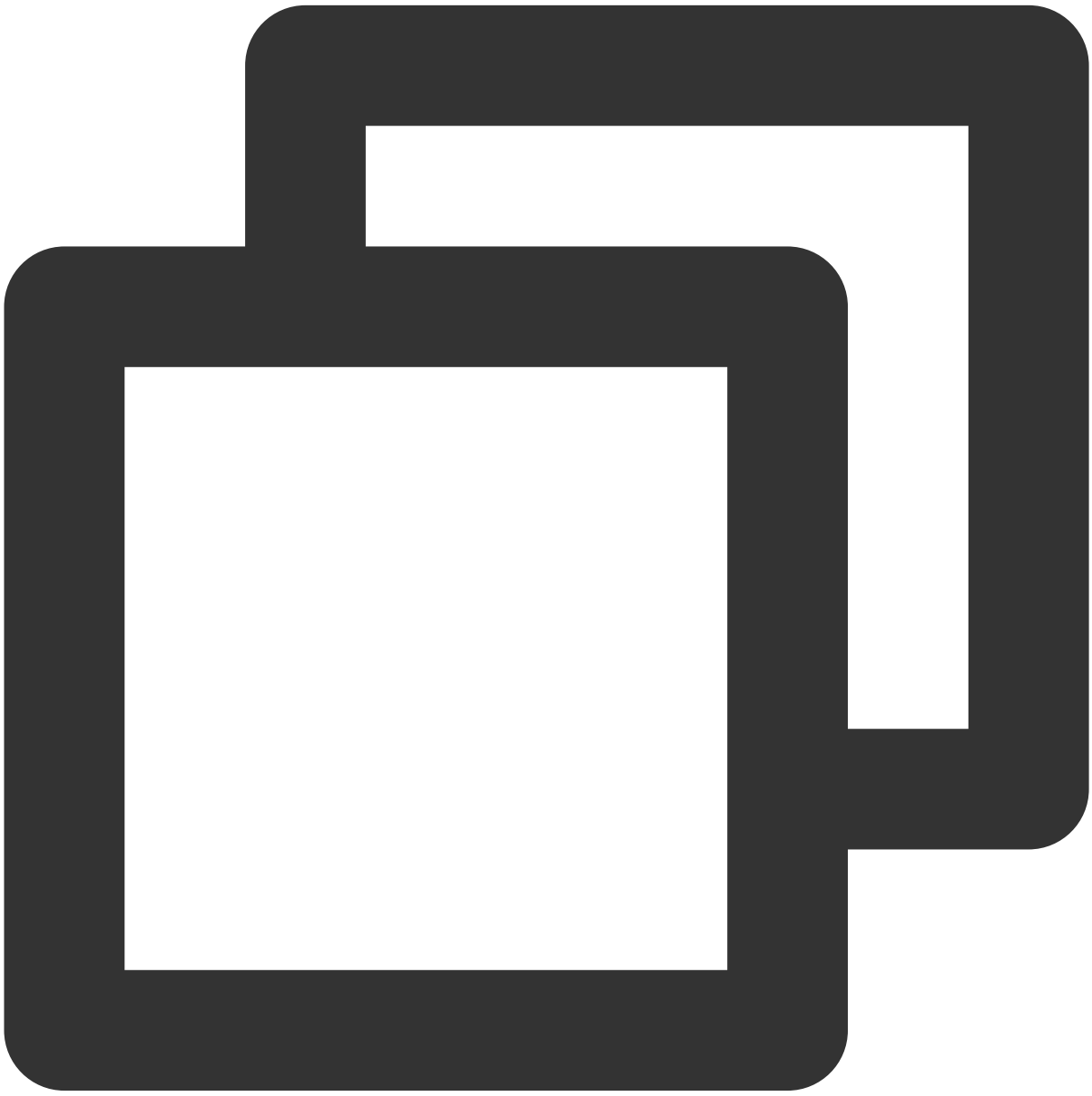
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SuperResolution</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
              <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
              <TemplateName>superresolution_demo</TemplateName>
              <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTem
              <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTem
              <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTem
              <DigitalWatermark>
                  <Type>Text</Type>
                  <Message>123456789ab</Message>
                  <State>Running</State>
                  <Version>V1</Version>
                  <IgnoreError>false</IgnoreError>
              </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>
```

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：SuperResolution	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
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
SuperResolution	Response.JobsDetail.Operation	同请求中的 Request.Operation.SuperResolution	Container
TemplateId	Response.JobsDetail.Operation	任务的模板 ID	String

版权所有：腾讯云计算（北京）有限责任公司

TemplateName	Response.JobsDetail.Operation	任务的模板名称, 当 TemplateId 存在时返回	String
Transcode	Response.JobsDetail.Operation	同请求中的 Request.Operation.Transcode	Container
TranscodeTemplateId	Response.JobsDetail.Operation	指定的转码模板 ID	String
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组
WatermarkTemplateId	Response.JobsDetail.Operation	指定的水印模板 ID	String 数组
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
DigitalWatermark	Response.JobsDetail.Operation	指定数字水印参数	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String

版权所有：腾讯云计算（北京）有限责任公司

ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

Container 类型 DigitalWatermark 的具体数据描述如下：

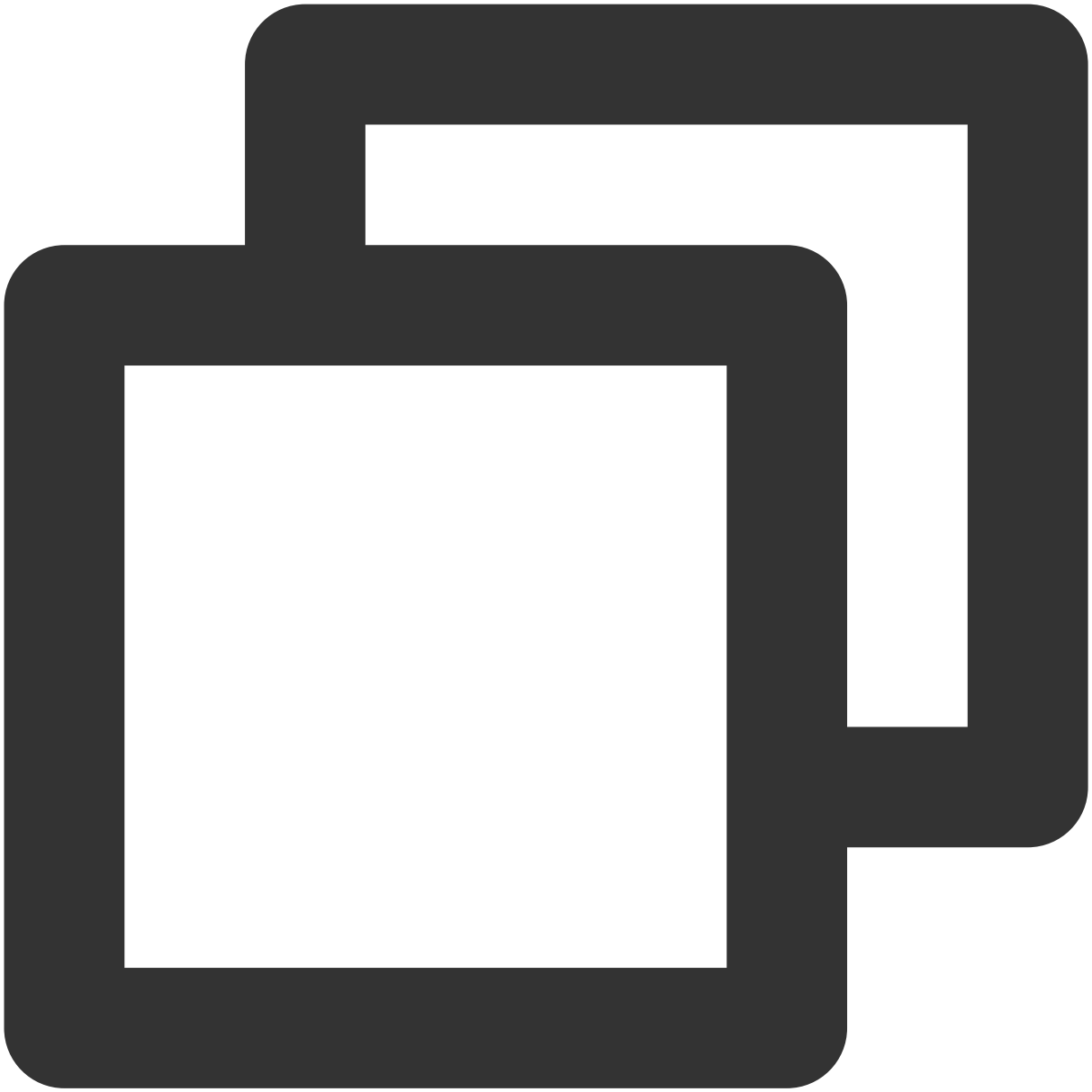
节点名称（关键字）	父节点	描述	类型
Message	Response.Operation.DigitalWatermark	成功嵌入视频的数字水印的字符串信息，长度不超过64个字符，仅支持中文、英文、数字、_、-和*	string
Type	Response.Operation.DigitalWatermark	水印类型，当前仅可设置为 Text	String
Version	Response.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String
IgnoreError	Response.Operation.DigitalWatermark	当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 false	string
State	Response.Operation.DigitalWatermark	添加水印是否成功，执行中为 Running，成功为 Success，失败为 Failed	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：使用超分辨率模板 ID

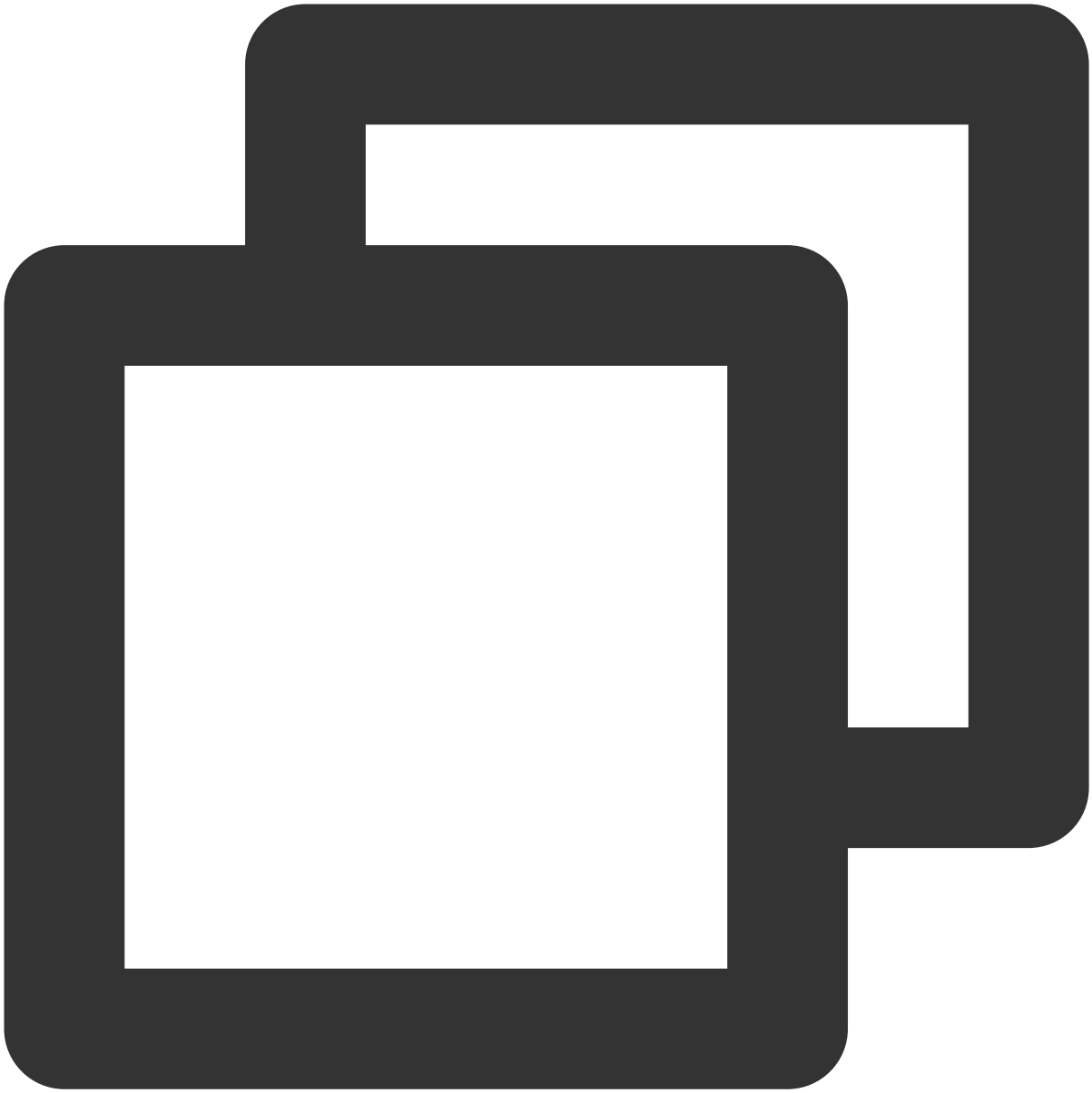


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SuperResolution</Tag>
    <Input>
        <Object>test.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplateI
        <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemplateI
        <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTemplateI
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </DigitalWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/sr.mp4</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应1

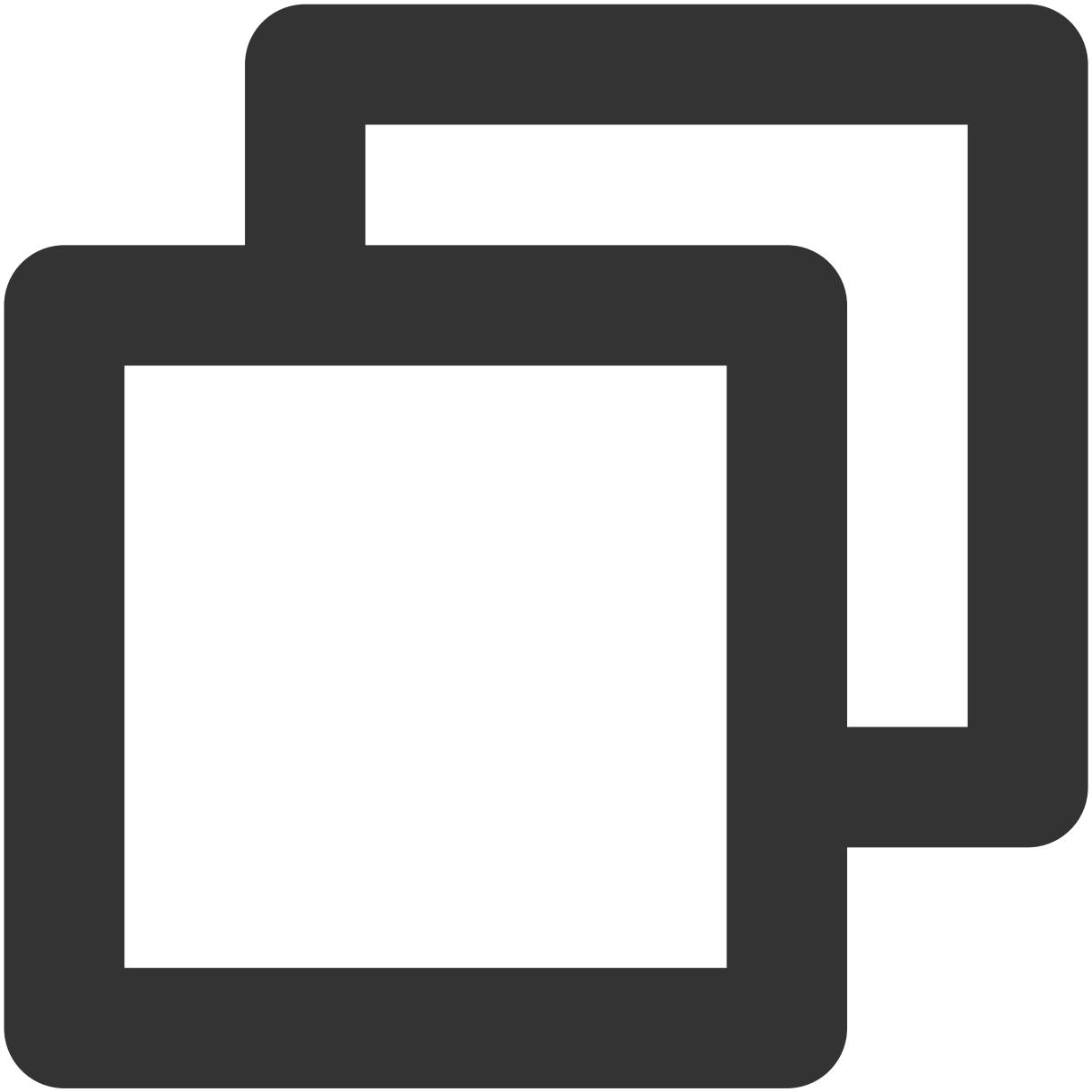


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>je8f65004eb8511eaaed4f377124a303c</JobId>
        <State>Submitted</State>
        <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SuperResolution</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>superresolution_demo</TemplateName>
            <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTempl
            <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTempl
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTempl
            <DigitalWatermark>
                <State>Running</State>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
            </DigitalWatermark>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/sr.mp4</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求2：使用超分辨率处理参数



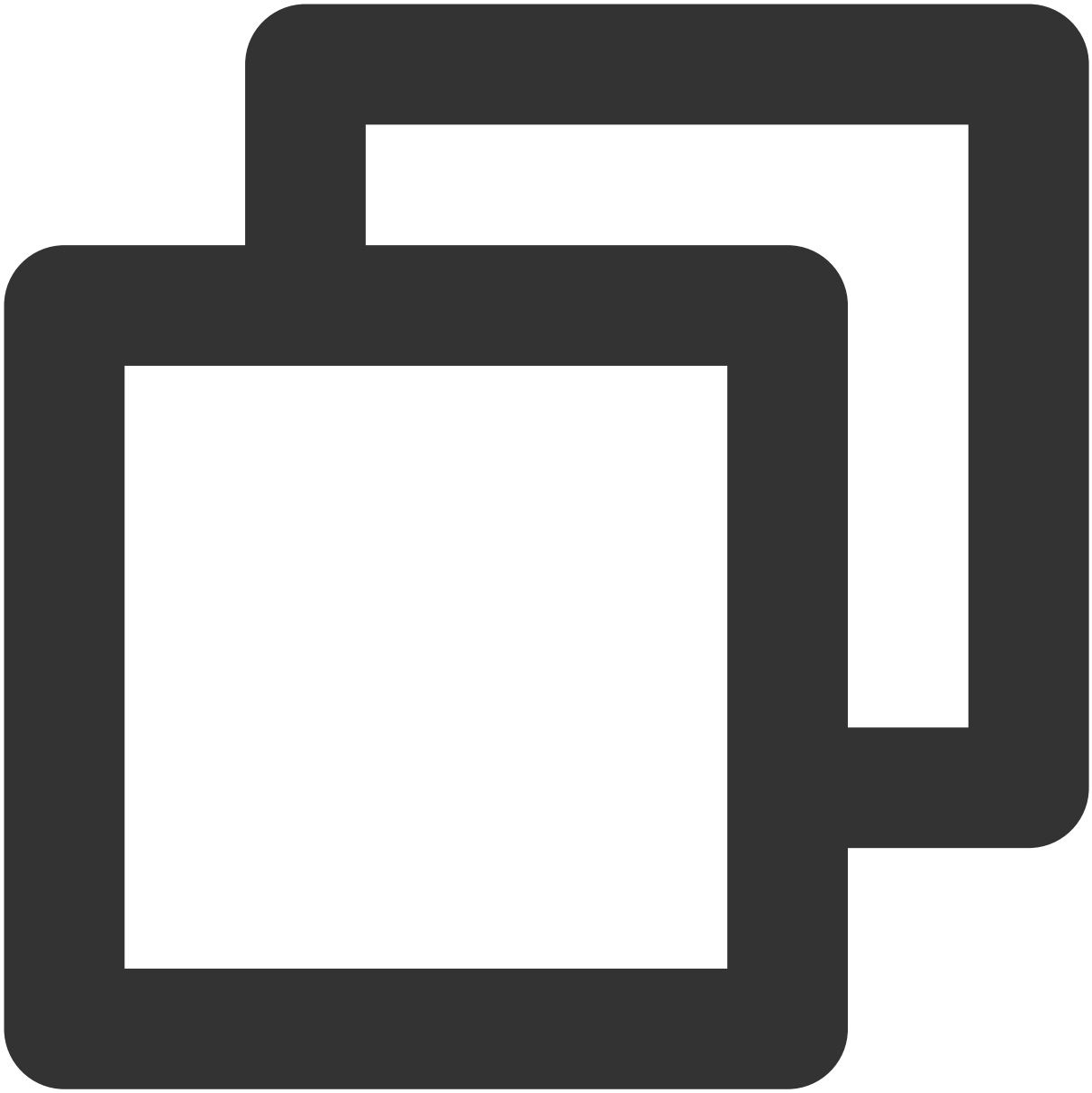
```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SuperResolution</Tag>
    <Input>
        <Object>test.mp4</Object>
    </Input>
```

```
    <Operation>
        <SuperResolution>
            <Resolution>sdtohd</Resolution>
            <EnableScaleUp>true</EnableScaleUp>
            <Version>Enhance</Version>
        </SuperResolution>
        <Transcode>
            <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>
        </Transcode>
        <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>simfang.ttf</FontType>
                <FontColor>0xRRGGBB</FontColor>
```

```
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </DigitalWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/sr.mp4</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应2



```
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>je8f65004eb8511eaaed4f377124a303c</JobId>
        <State>Submitted</State>
        <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>SuperResolution</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <SuperResolution>
                <Resolution>sdtohd</Resolution>
                <EnableScaleUp>true</EnableScaleUp>
                <Version>Enhance</Version>
            </SuperResolution>
            <Transcode>
                <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>
            </Transcode>
```

```
            <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>simfang.ttf</FontType>
                    <FontColor>0xRRGGBB</FontColor>
                    <Transparency>30</Transparency>
                </Text>
            </Watermark>
            <DigitalWatermark>
                <State>Running</State>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
            </DigitalWatermark>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/sr.mp4</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交音视频转码任务

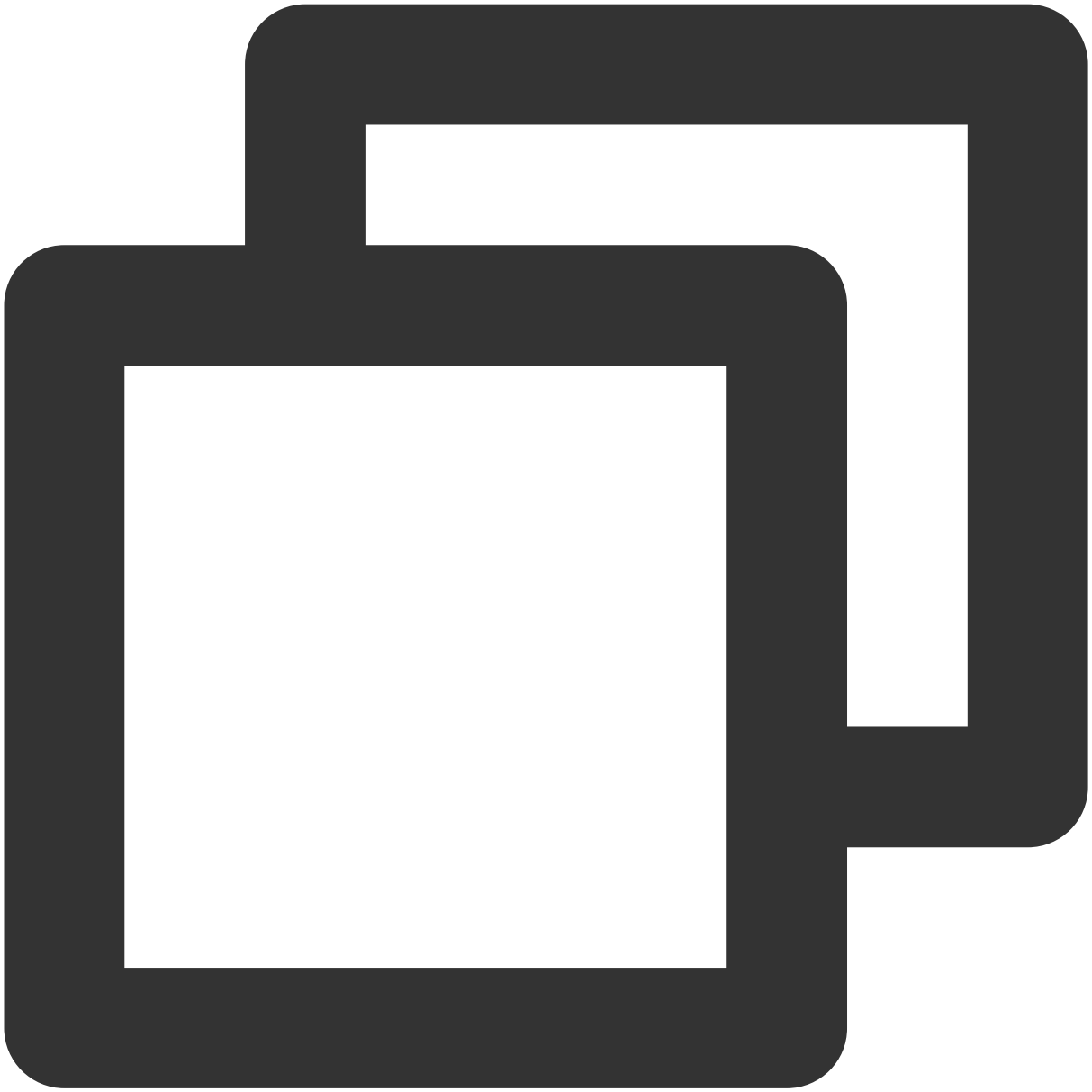
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个转码任务

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：

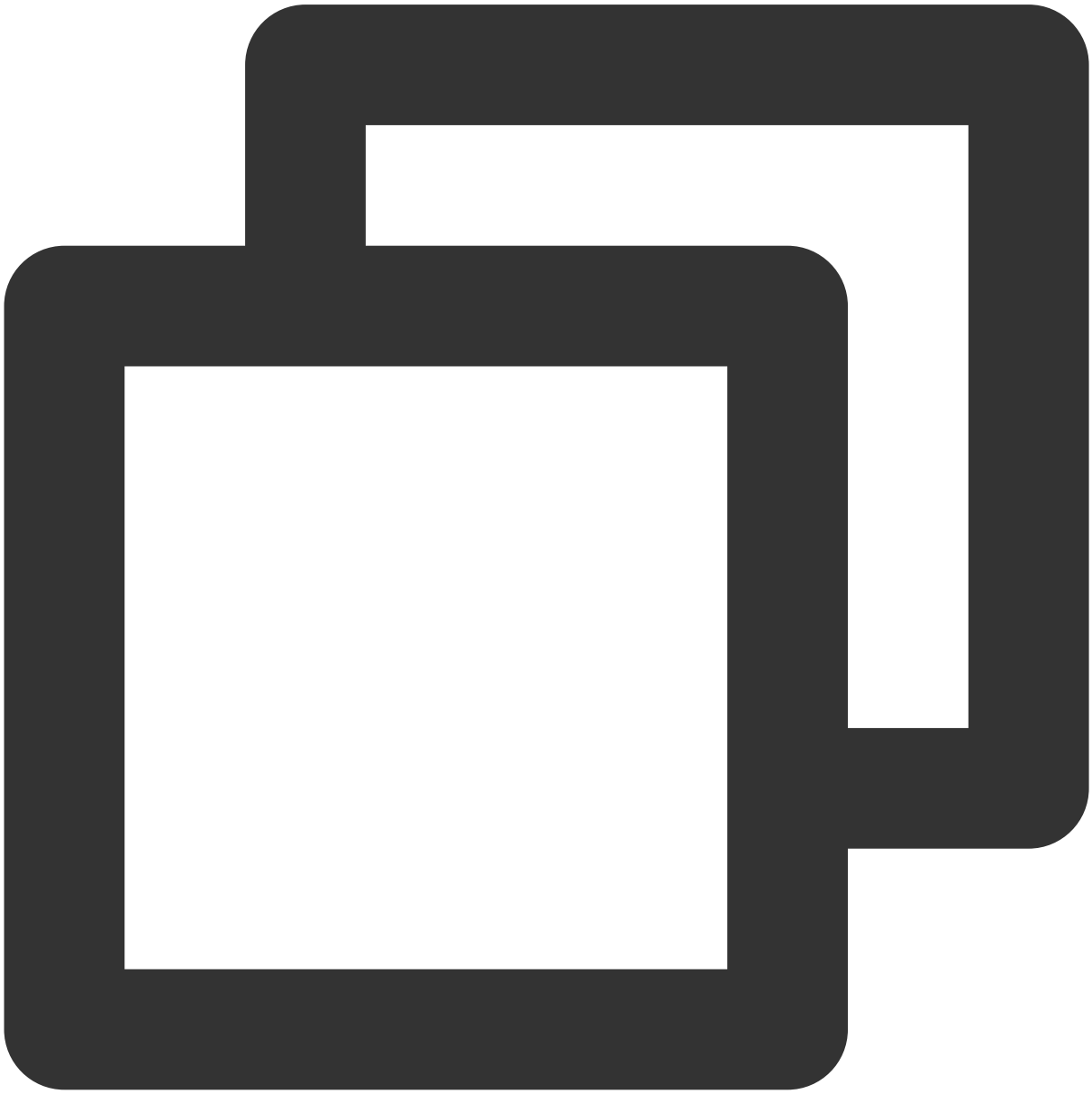
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Transcode</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemplateI
        <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>
        </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>
    <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：Transcode	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
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
Object	Request.Input	媒体文件名	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
TemplateId	Request.Operation	指定的模板 ID	String	否
Transcode	Request.Operation	指定转码模板参数	Container	否
WatermarkTemplateId	Request.Operation	指定的水印模板 ID，可以传多个水印模板 ID	String	否
Watermark	Request.Operation	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口中的 Request.Watermark	Container	否
RemoveWatermark	Request.Operation	指定去除水印参数, H265暂不支持次参数	Container	否
DigitalWatermark	Request.Operation	指定数字水印参数	Container	否

版权所有：腾讯云计算（北京）有限责任公司

Output	Request.Operation	结果输出地址	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

说明：
对于转码参数，优先使用优先使用 TemplateId，无 TemplateId 时使用 Transcode；对于水印参数，可以使用 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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	必选
TimeInterval	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Video	Container	否
Audio	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Audio	Container	否
TransConfig	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TransConfig	Container	否
AudioMix	Request.Operation.Transcode	混音参数，详情见AudioMix	Container 数组	否

Container 类型 RemoveWatermark 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

Dx	Request.Operation.RemoveWatermark	距离左上角原点 x 偏移，范围为[1, 4096]	string	是
Dy	Request.Operation.RemoveWatermark	距离左上角原点 y 偏移，范围为[1, 4096]	string	是
Width	Request.Operation.RemoveWatermark	宽，范围为[1, 4096]	string	是
Height	Request.Operation.RemoveWatermark	高，范围为[1, 4096]	string	是

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.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.DigitalWatermark	水印类型，当前仅可设置为 Text	String	是
Version	Request.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String	是
IgnoreError	Request.Operation.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	string	是

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

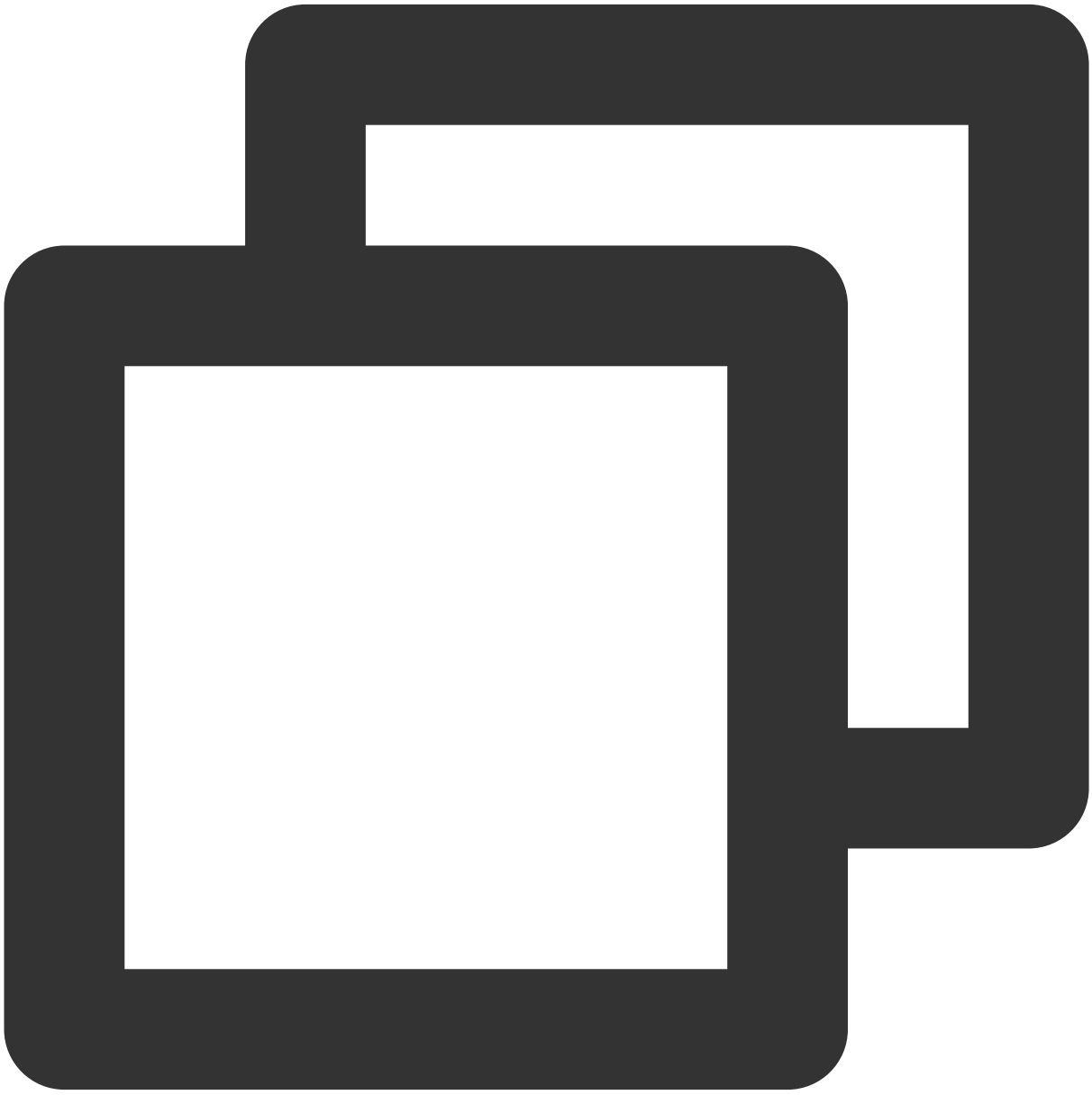
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <Progress>0</Progress>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
```

```
        <StartTime>-</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>
            <TemplateName>trans_993874</TemplateName>
            <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>
```

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：Transcode	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
Progress	Response.JobsDetail	任务进度百分比，只有在State为 Submitted、Running、Success、Pause 时有意义，范围为[0, 100]	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
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

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
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
Transcode	Response.JobsDetail.Operation	同请求中的 Request.Operation.Transcode	Containe
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Containe 数组
RemoveWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.RemoveWatermark	Containe
WatermarkTemplateId	Response.JobsDetail.Operation	指定的水印模板 ID	String 数组
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Containe
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Containe
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Containe
DigitalWatermark	Response.JobsDetail.Operation	指定数字水印参数	Containe
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 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

Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

Container 类型 DigitalWatermark 的具体数据描述如下：

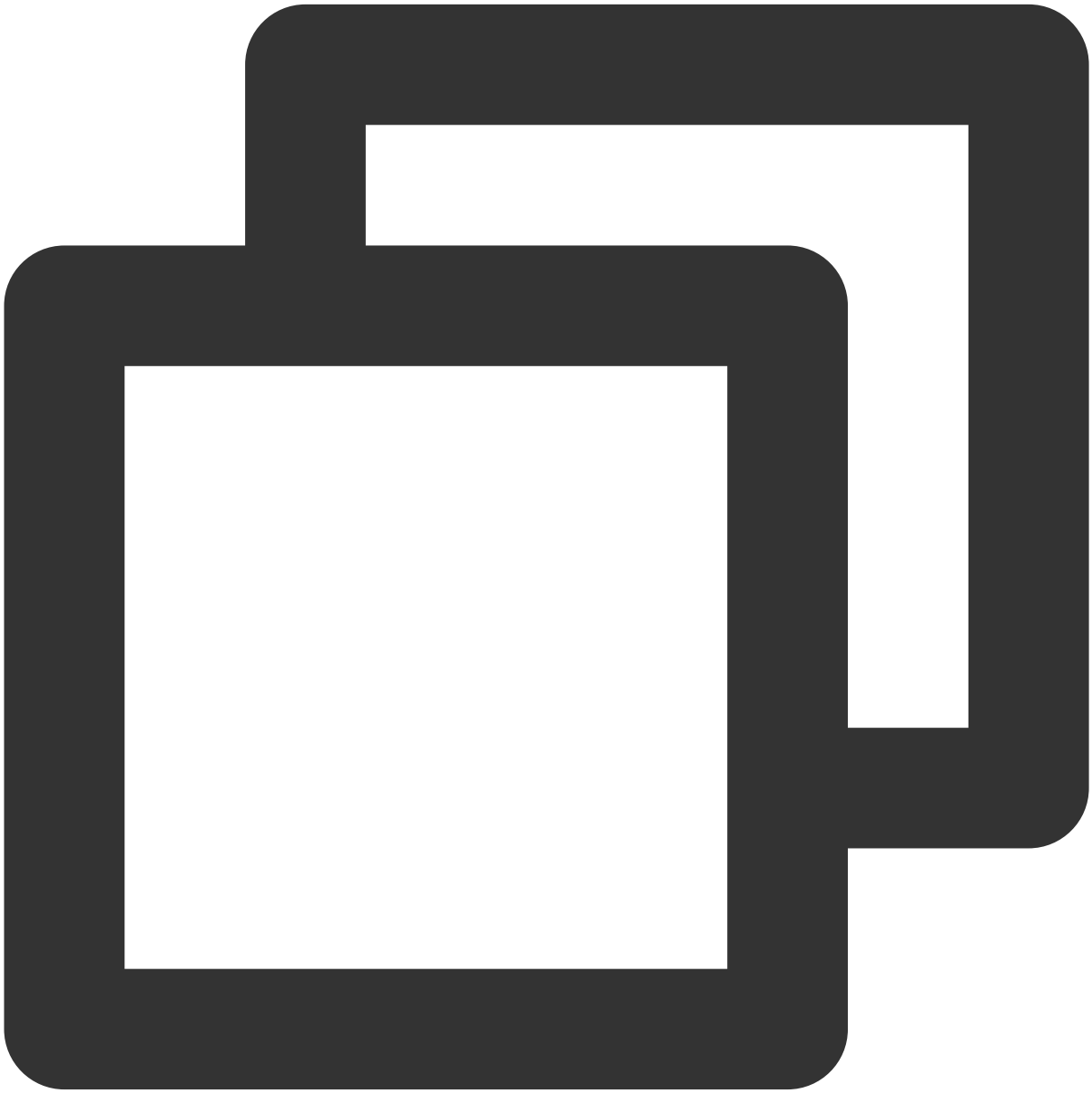
节点名称（关键字）	父节点	描述	类型
Message	Response.Operation.DigitalWatermark	成功嵌入视频的数字水印的字符串信息，长度不超过64个字符，仅支持中文、英文、数字、_、-和*	string
Type	Response.Operation.DigitalWatermark	水印类型，当前仅可设置为 Text	String
Version	Response.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String
IgnoreError	Response.Operation.DigitalWatermark	当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false	string
State	Response.Operation.DigitalWatermark	添加水印是否成功，执行中为 Running，成功为 Success，失败为 Failed	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：使用转码模板 ID

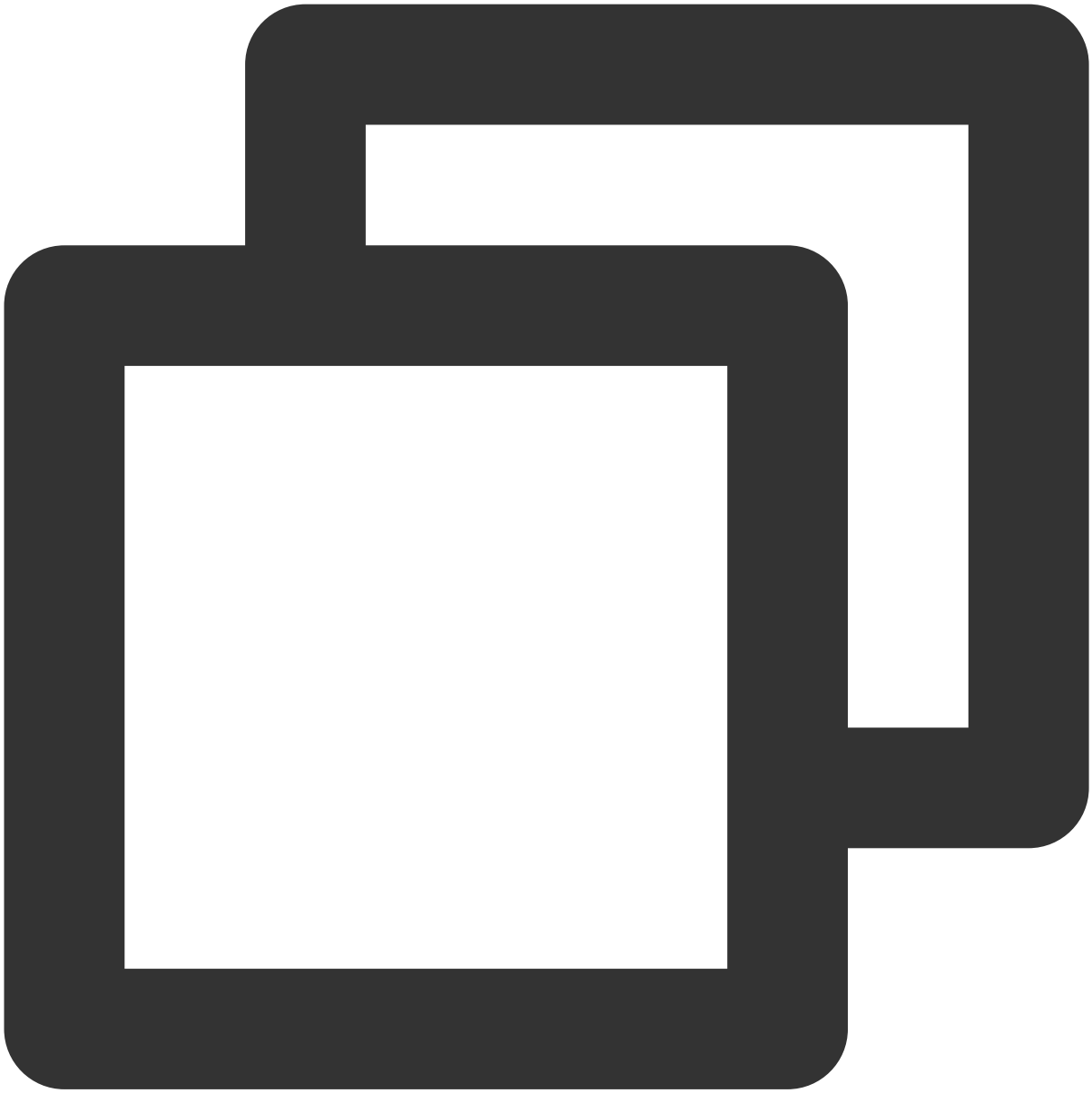


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Transcode</Tag>
    <Input>
```

```
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemplateI
        <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>
        </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>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应1

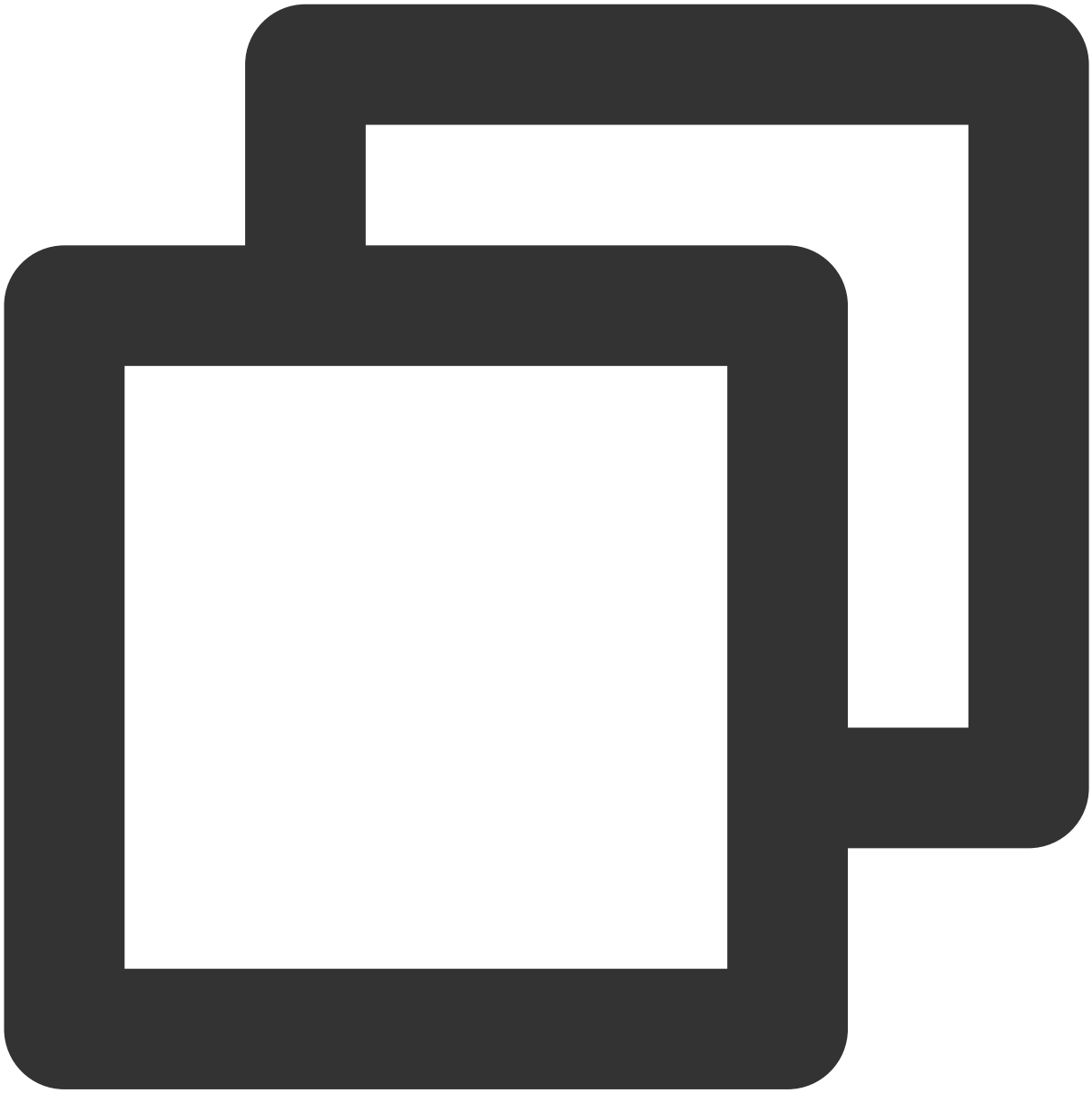


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <Progress>0</Progress>
        <CreationTime>2022-06-27T15:23:11+0800</CreationTime>
        <StartTime>-</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>
            <TemplateName>trans_993874</TemplateName>
            <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：使用转码处理参数



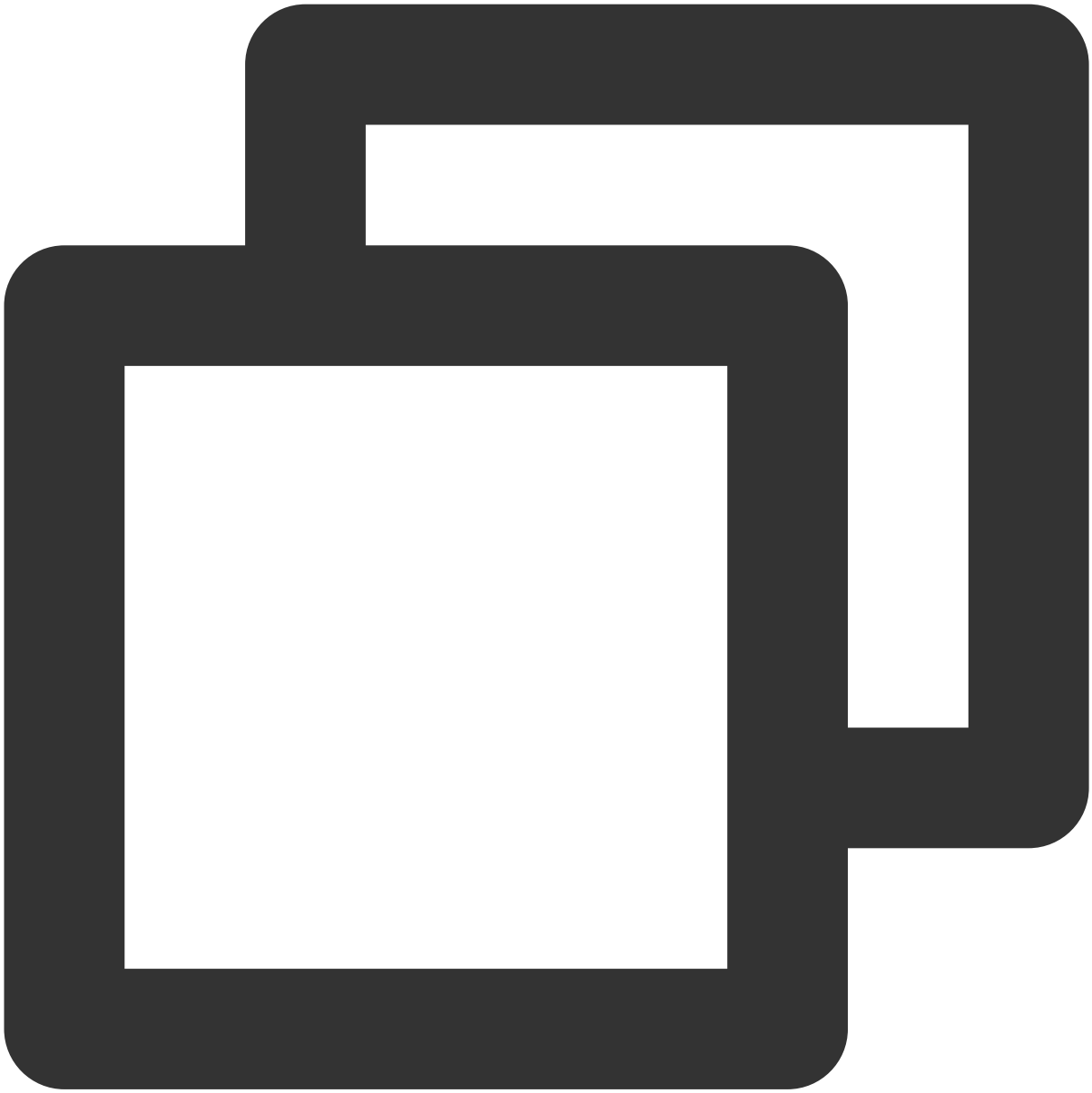
```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Transcode</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <Transcode>
            <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>
            <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>
        </Transcode>
        <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>simfang.ttf</FontType>
                <FontColor>0xRRGGBB</FontColor>
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应2



```
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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <Progress>0</Progress>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</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>
            <Transcode>
                <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>
                <AudioMix>
                    <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix
                    <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>
        </Transcode>
        <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>
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

提交语音合成任务

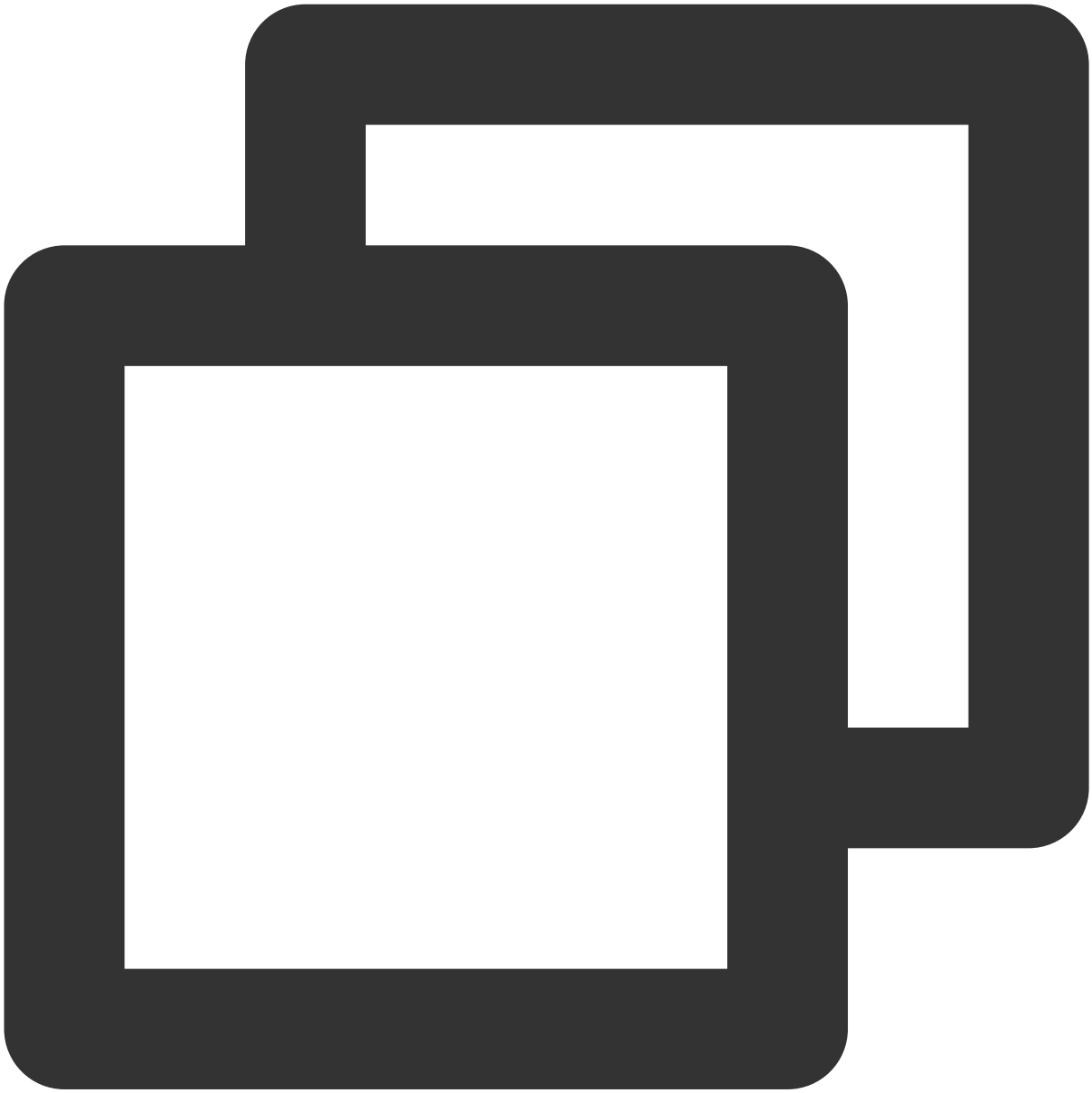
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个语音合成任务。

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：

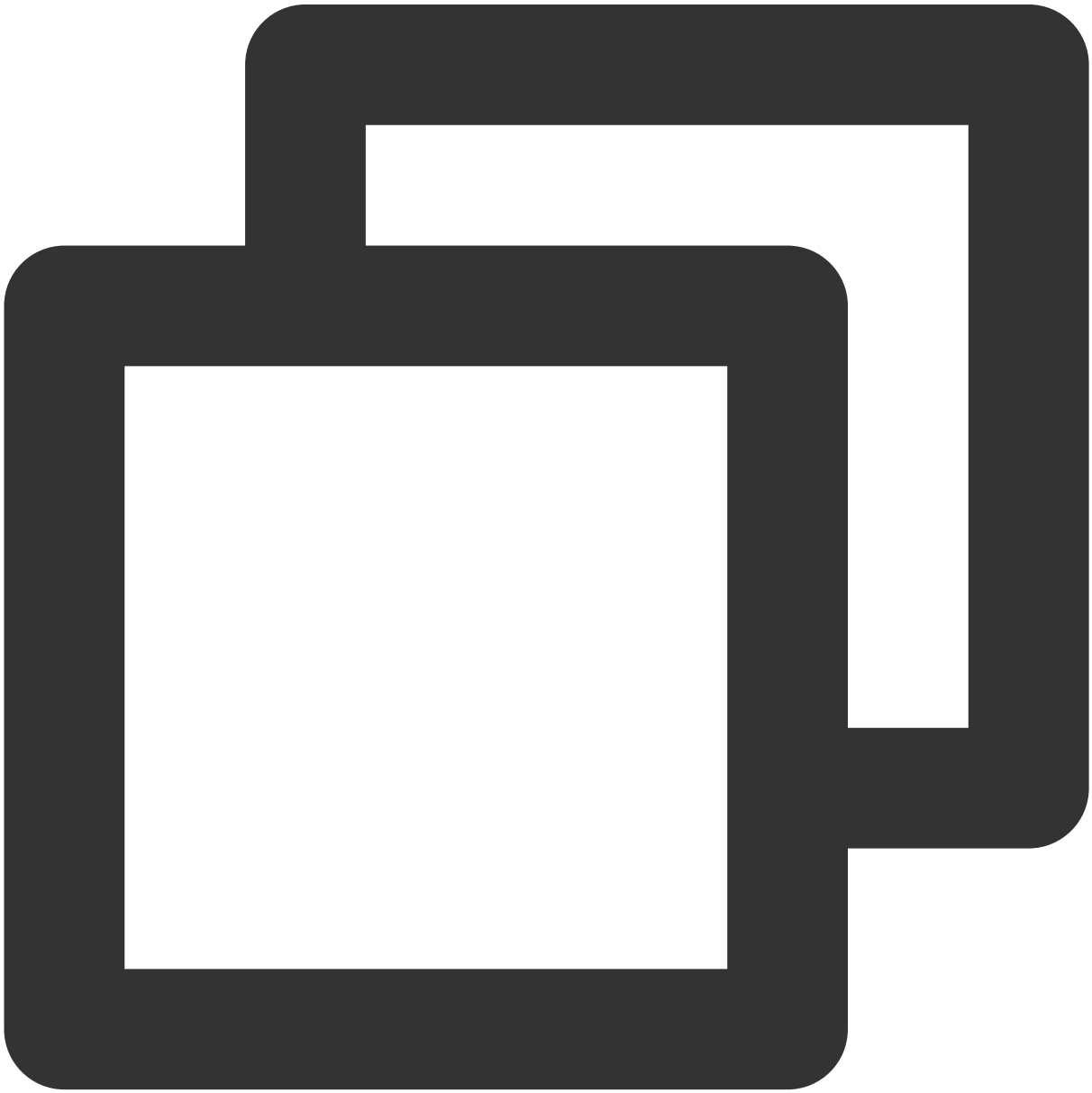
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Tts</Tag>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <TtsConfig>
            <Input>床前明月光，疑是地上霜</Input>
            <InputType>Text</InputType>
        </TtsConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>demo.mp3</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：Tts	String	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，优先级高于队列的回调类型	String	否
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	Container	否

版权所有：腾讯云计算（北京）有限责任公司

		TDMQ 时必填。详情见 CallBackMqConfig		

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
TtsTpl	Request.Operation	语音合成参数	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
TtsConfig	Request.Operation	语音合成任务参数	Container	是
Output	Request.Operation	结果输出地址	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

注意：

优先使用 TemplateId，无 TemplateId 时使用 TtsTpl。

Container 类型 TtsTpl 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.Operation.TtsTpl	创建语音合成模板 CreateMediaTemplate 中 Request.Mode	String	否
Codec	Request.Operation.TtsTpl	创建语音合成模板 CreateMediaTemplate 中 Request.Codec	String	否
VoiceType	Request.Operation.TtsTpl	创建语音合成模板 CreateMediaTemplate 中 Request.VoiceType	String	否
Volume	Request.Operation.TtsTpl	创建语音合成模板 CreateMediaTemplate 中 Request.Volume	String	否
Speed	Request.Operation.TtsTpl	创建语音合成模板 CreateMediaTemplate 中 Request.Speed	String	否

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 TtsConfig 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
InputType	Request.Operation.TtsConfig	输入类型，Url/Text	String	是
Input	Request.Operation.TtsConfig	1. 当 InputType 为 Url 时， 必须是合法的 COS 地址，文件必须是utf-8编码，且大小不超过10M。如果模板中指定的合成方式为同步处理，则文件内容不超过300个utf-8字符；如果模板中指定的合成方式为异步处理，则文件内容不超过10000个utf-8字符。 2. 当 InputType为Text 时, 输入必须是utf-8字符, 且不超过300个字符。	String	是

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

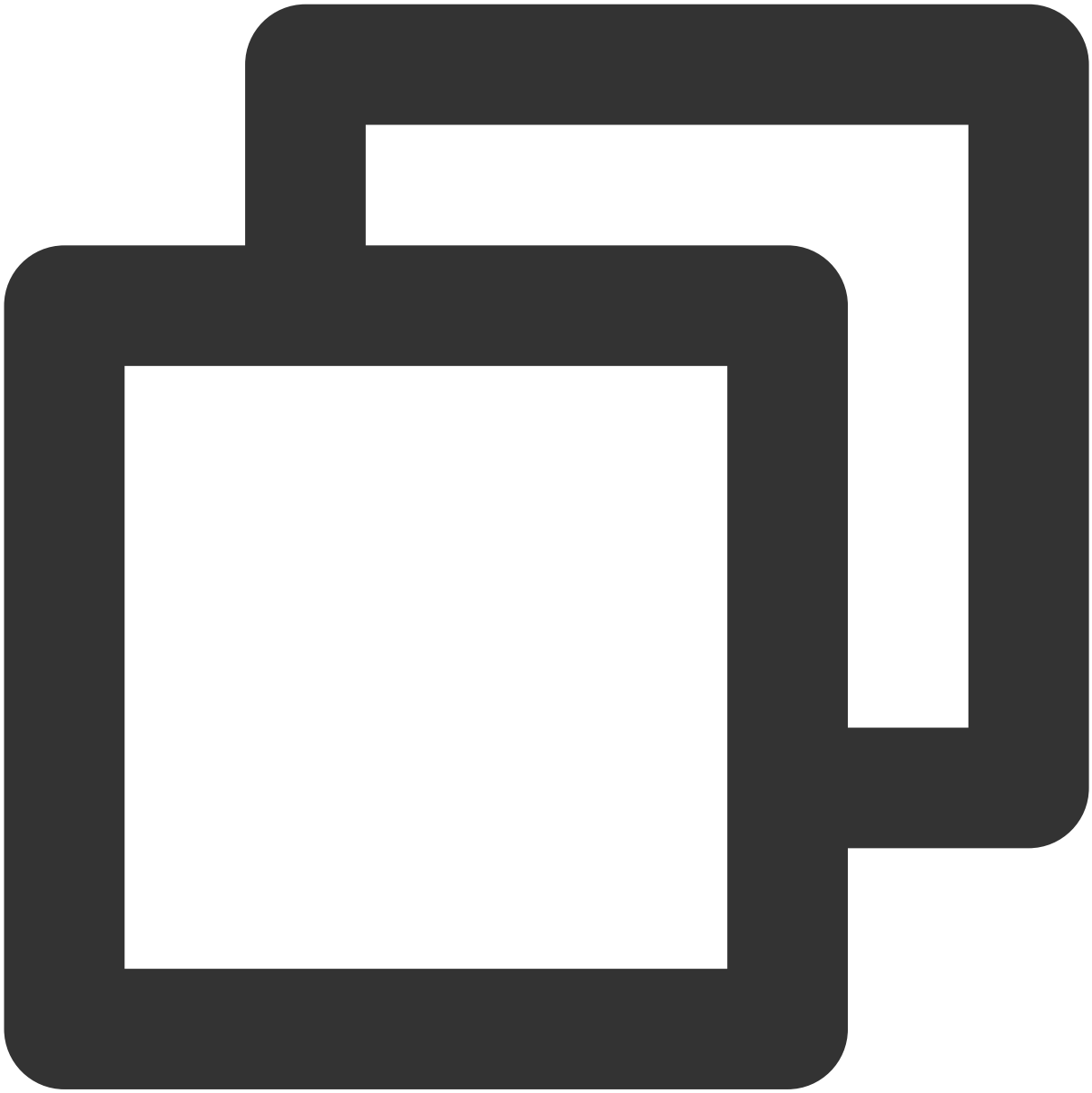
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Tts</Tag>
```

```
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>tts_demo</TemplateName>
            <TtsConfig>
                <Input>床前明月光，疑是地上霜</Input>
                <InputType>Text</InputType>
            </TtsConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/tts.mp3</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：Tts	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String

版权所有：腾讯云计算（北京）有限责任公司

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	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
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
TtsTpl	Response.JobsDetail.Operation	请求中的 Request.Operation.TtsTpl	Container
TtsConfig	Response.JobsDetail.Operation	请求中的 Request.Operation.TtsConfig	Container
Output	Response.JobsDetail.Operation	请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

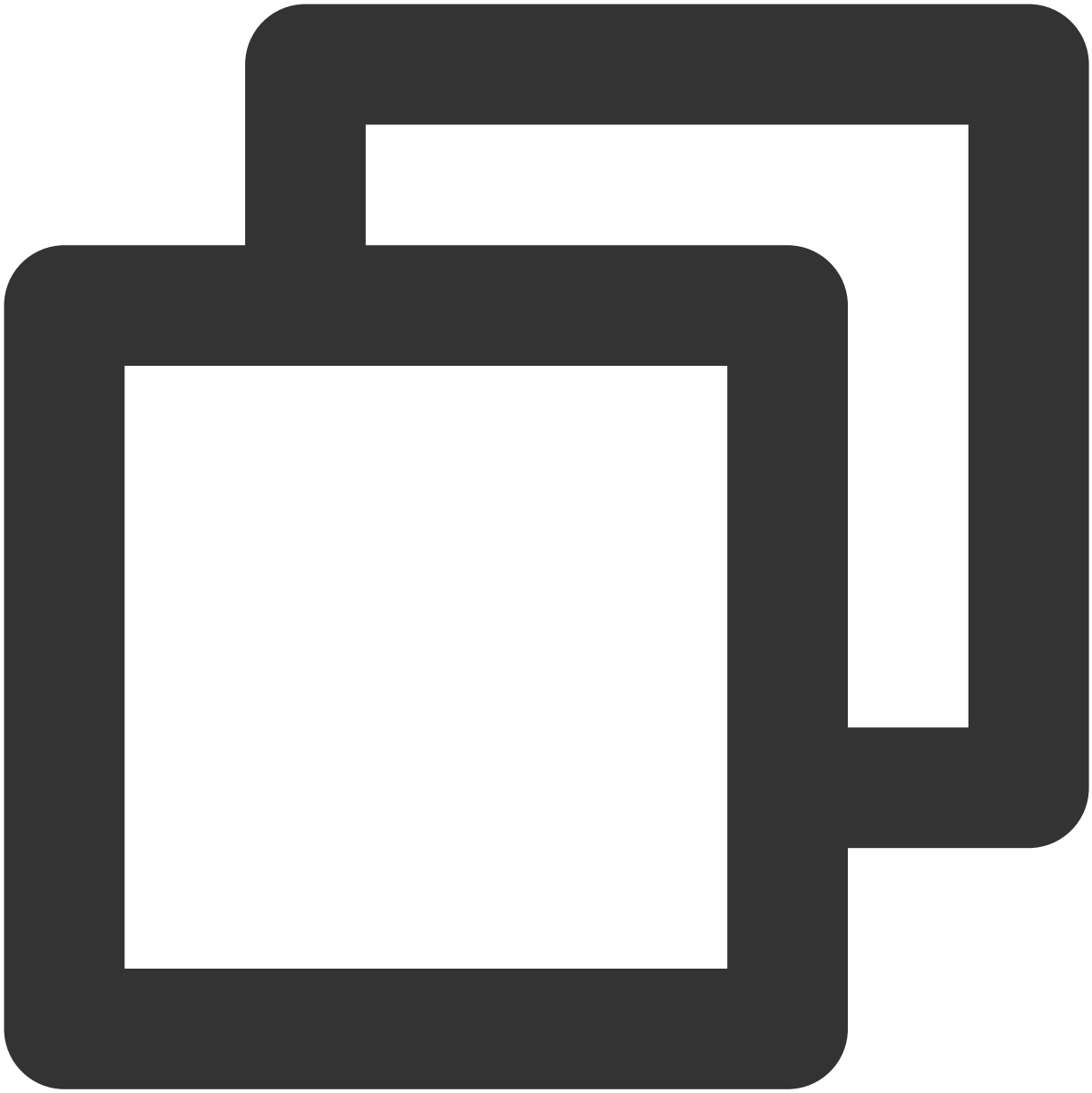
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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：使用语音合成模板 ID

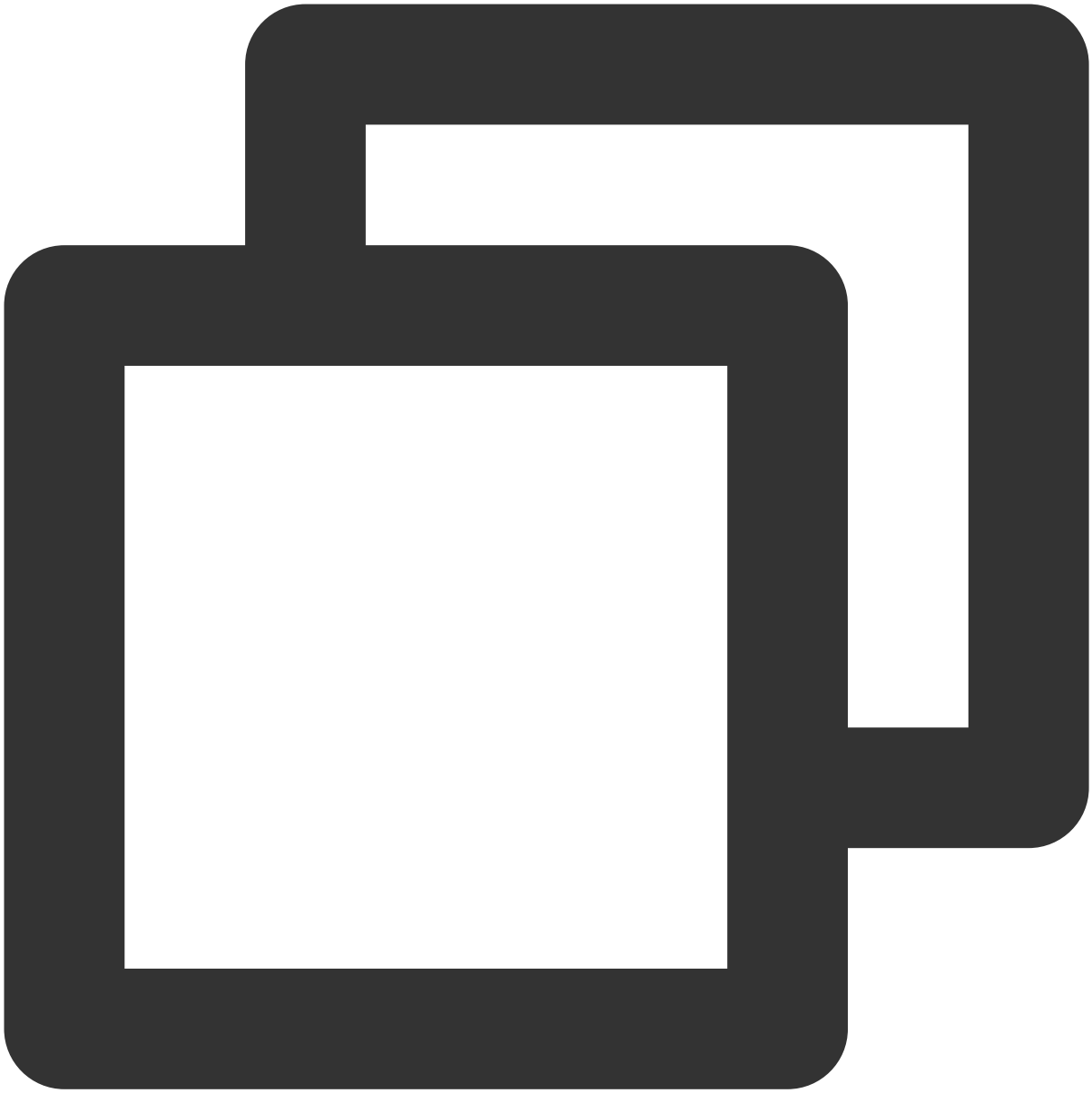


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Tts</Tag>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <TtsConfig>
```

```
            <Input>床前明月光，疑是地上霜</Input>
            <InputType>Text</InputType>
        </TtsConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>demo.mp3</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应

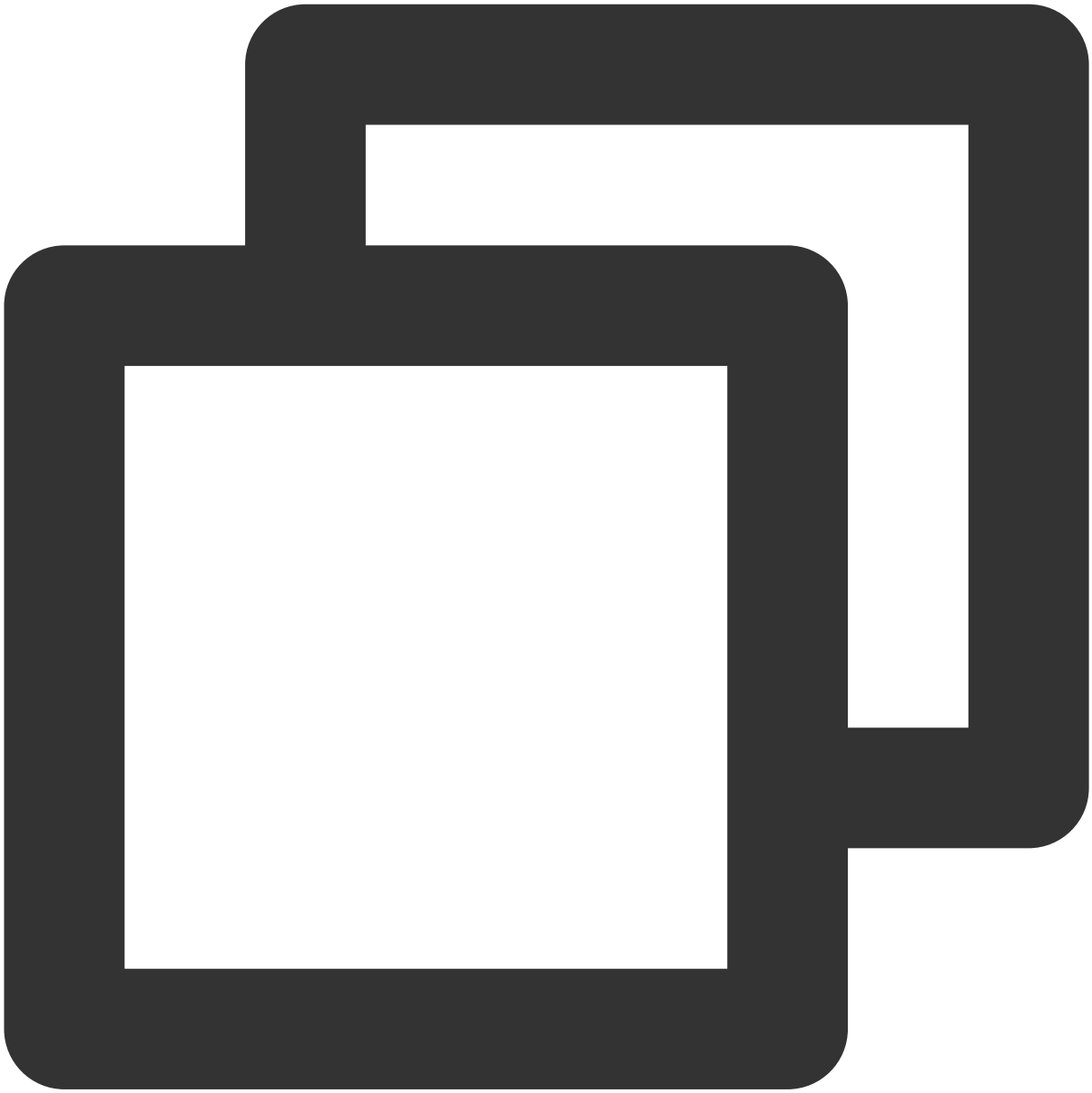


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Tts</Tag>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>tts_demo</TemplateName>
            <TtsConfig>
                <Input>床前明月光，疑是地上霜</Input>
                <InputType>Text</InputType>
            </TtsConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/tts.mp3</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求1：使用语音合成参数

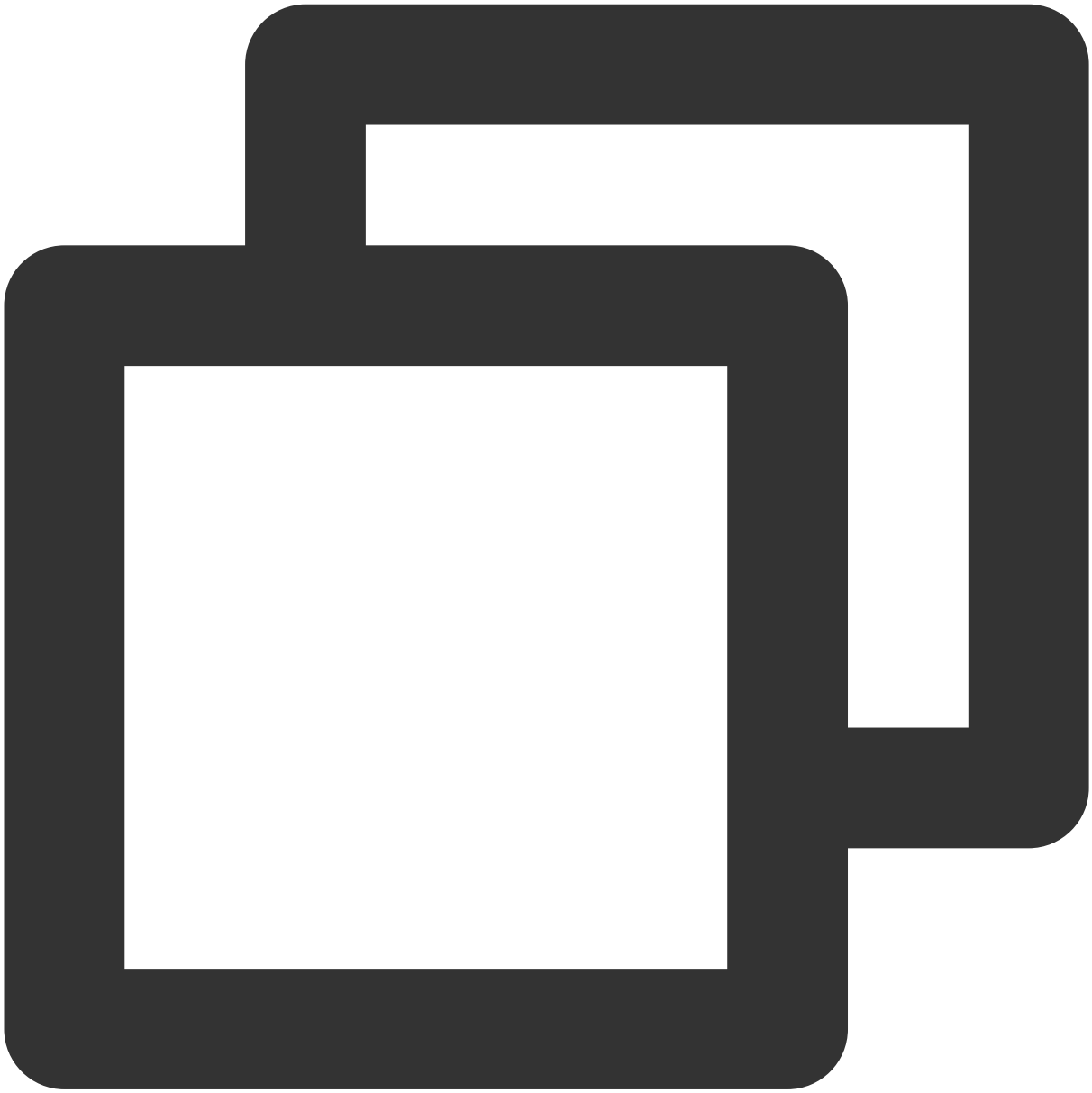


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Tts</Tag>
    <Operation>
        <TtsTpl>
            <Codec>mp3</Codec>
```

```
            <Mode>Sync</Mode>
            <Speed>100</Speed>
            <VoiceType>aixiaonan</VoiceType>
            <Volume>0</Volume>
        </TtsTpl>
        <TtsConfig>
            <Input>床前明月光，疑是地上霜</Input>
            <InputType>Text</InputType>
        </TtsConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>demo.mp3</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>Tts</Tag>
        <Operation>
            <TtsTpl>
                <Codec>mp3</Codec>
                <Mode>Sync</Mode>
                <Speed>100</Speed>
                <VoiceType>aixiaonan</VoiceType>
                <Volume>0</Volume>
            </TtsTpl>
            <TtsConfig>
                <Input>床前明月光，疑是地上霜</Input>
                <InputType>Text</InputType>
            </TtsConfig>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/tts.mp3</Object>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交精彩集锦任务

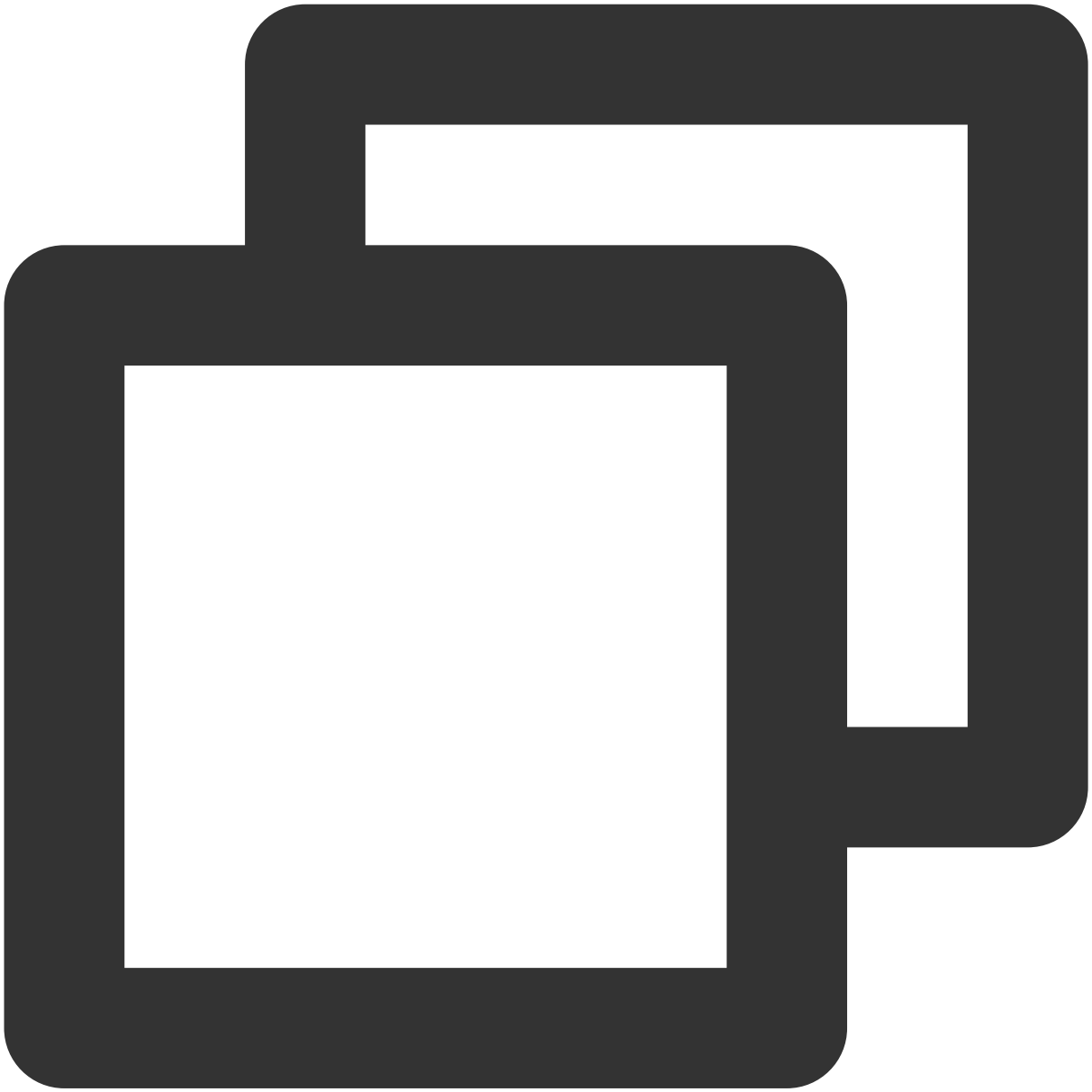
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个精彩集锦任务。

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：

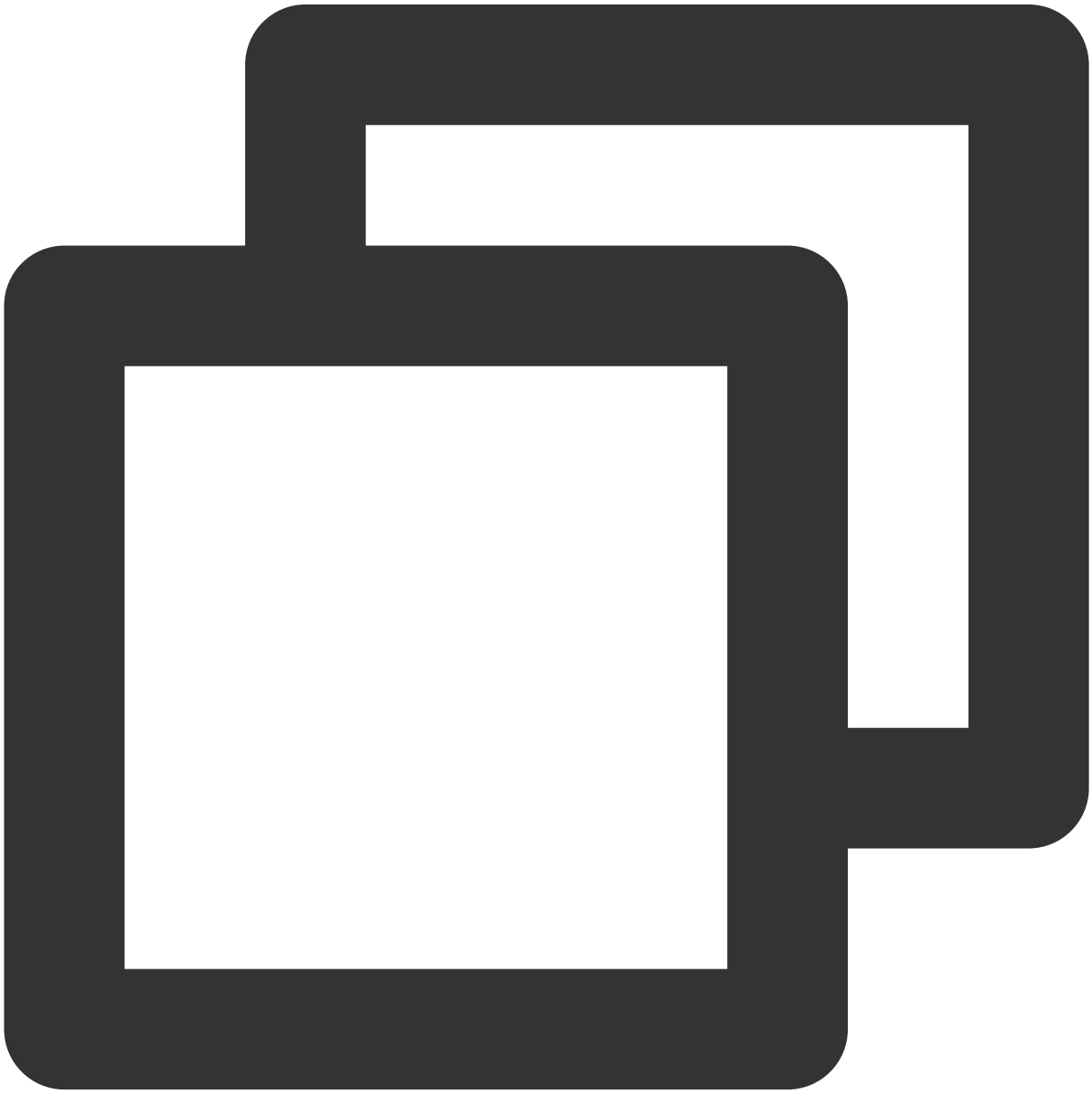
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>VideoMontage</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <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>
    <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：VideoMontage	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
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
VideoMontage	Request.Operation	指定精彩集锦模板参数	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
Output	Request.Operation	结果输出地址	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

注意：

优先使用 TemplateId，无 TemplateId 时使用 VideoMontage。

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	必选
TimeInterval	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的	Container	否

		Request.Video		
Audio	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Audio	Container	否
TransConfig	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TransConfig	Container	否
AudioMix	Request.Operation.Transcode	混音参数，详情见 AudioMix	Container 数组	否

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

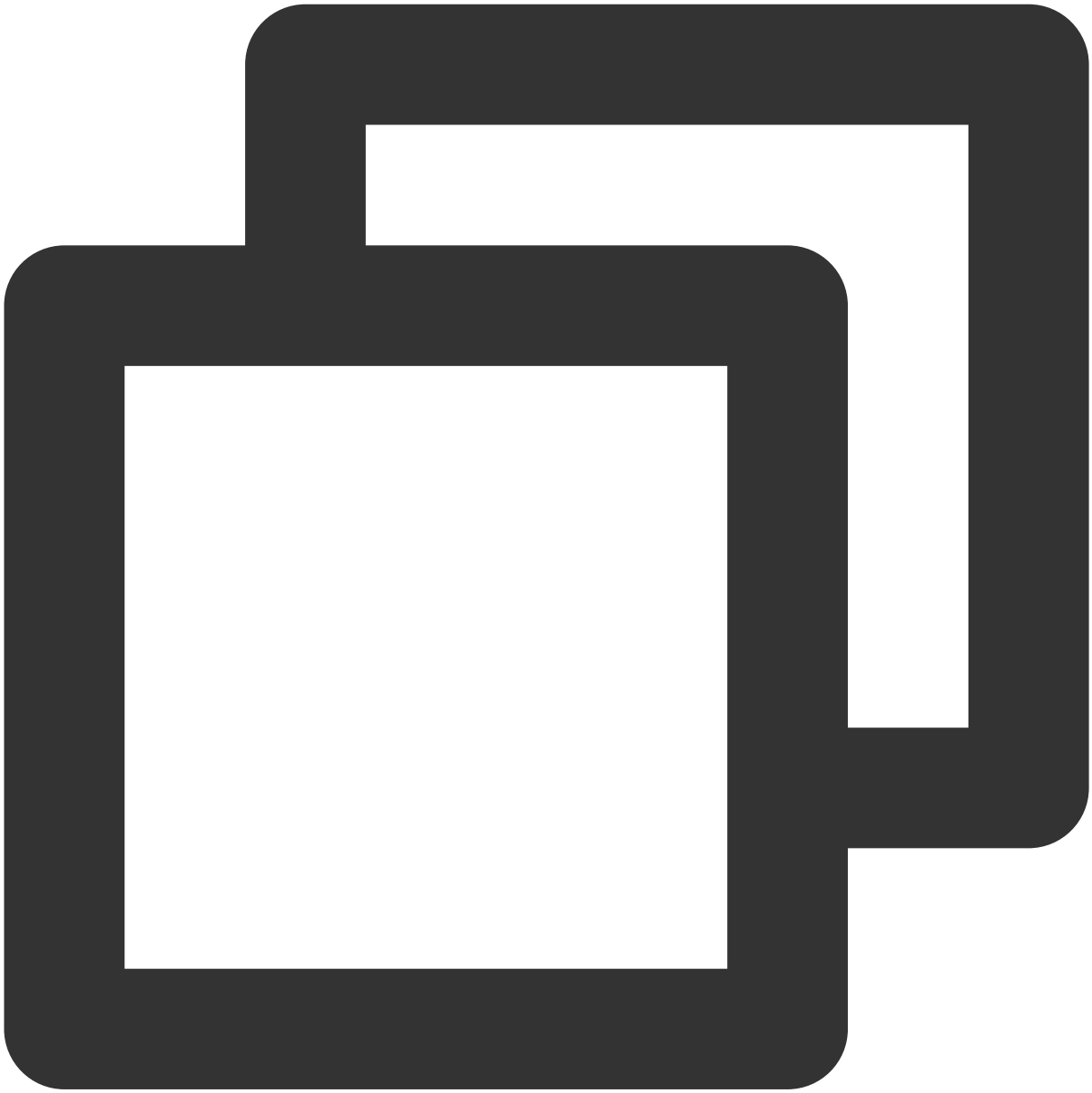
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoMontage</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>videomontage_demo</TemplateName>
            <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>
```

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：VideoMontage	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String

版权所有：腾讯云计算（北京）有限责任公司

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	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
VideoMontage	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoMontage	Container
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
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	转码输出视频的信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

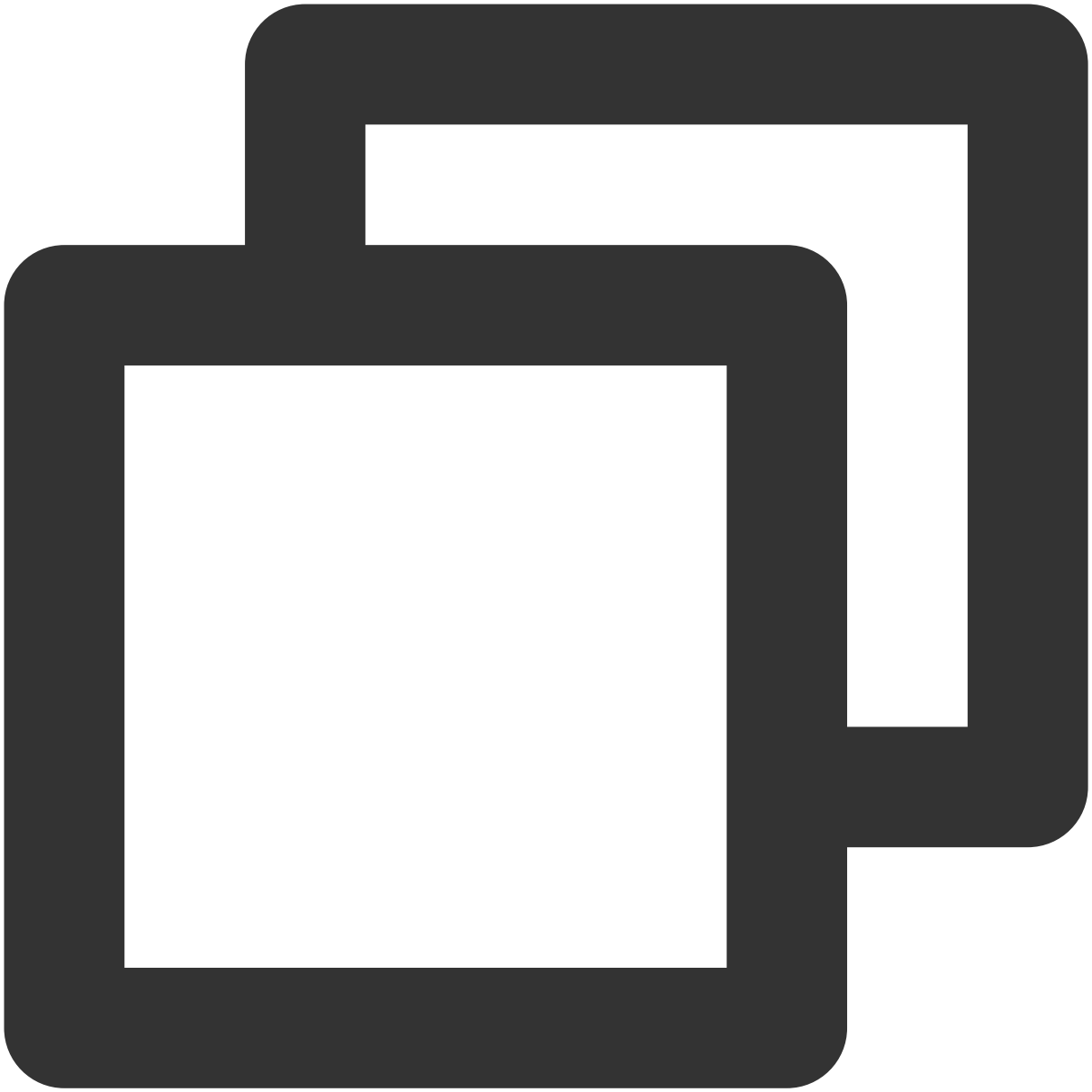
错误码

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

案例一：使用精彩集锦模板 ID

请求

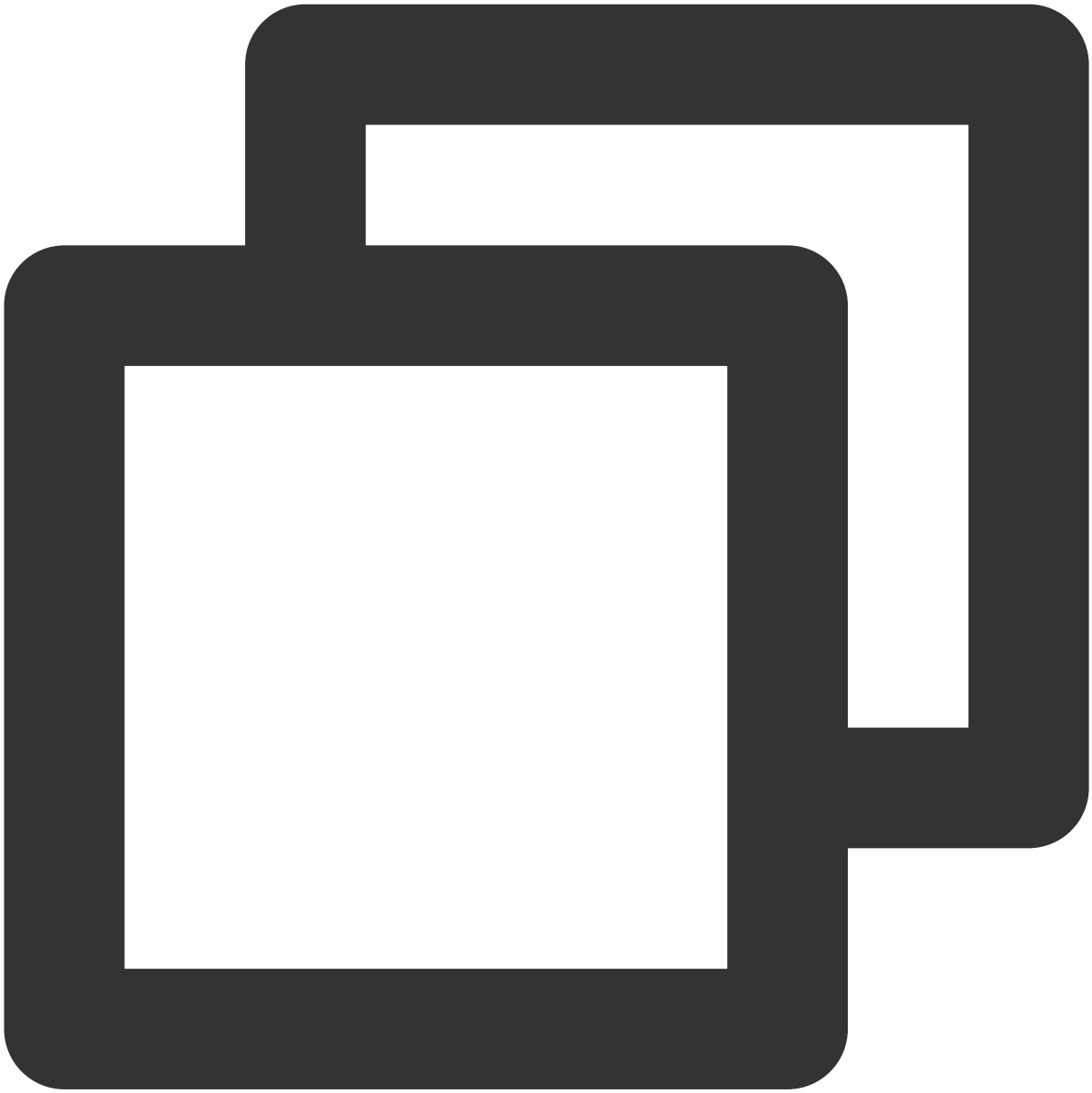


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoMontage</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <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>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应



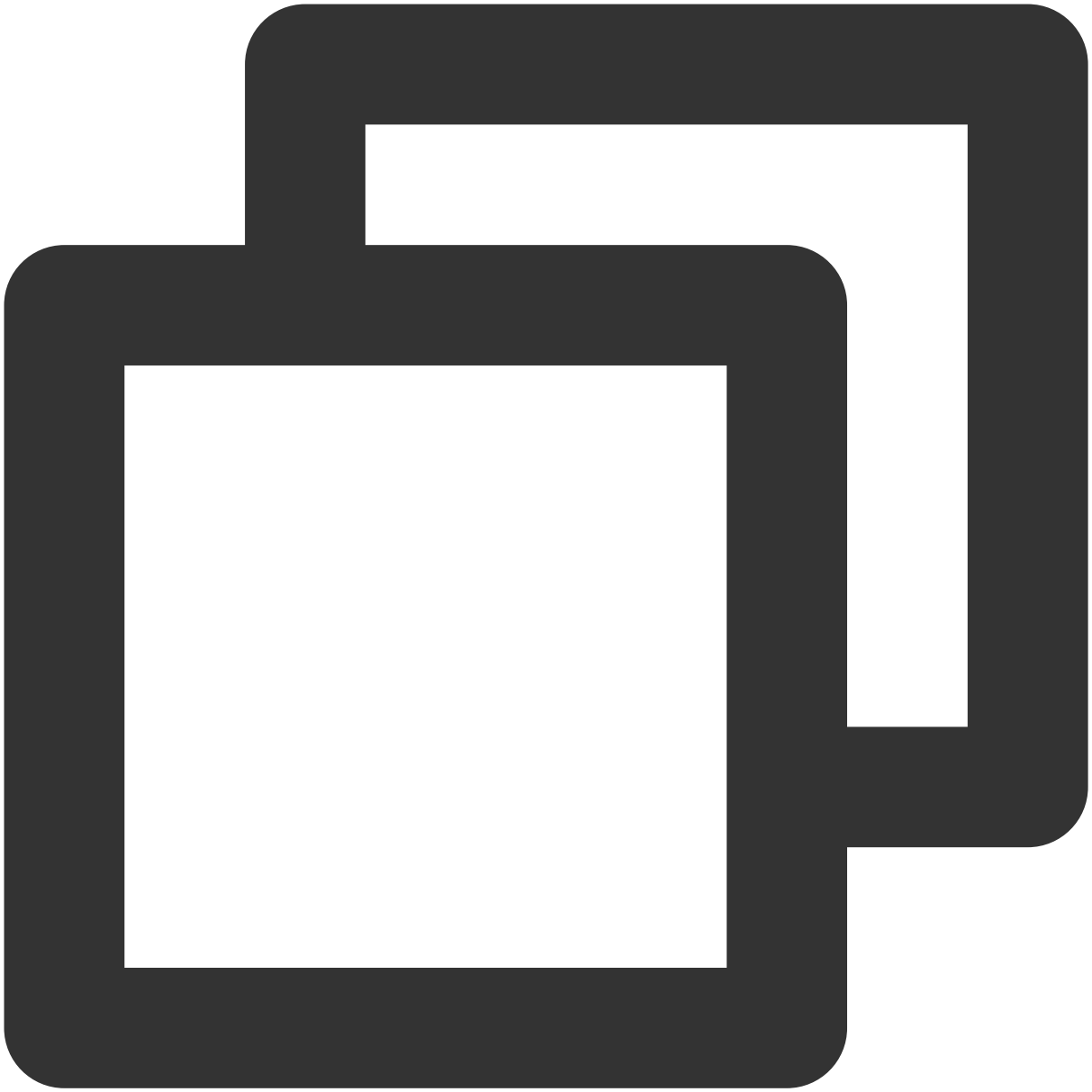
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoMontage</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>videomontage_demo</TemplateName>
            <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>
```

案例二：使用精彩集锦处理参数

请求

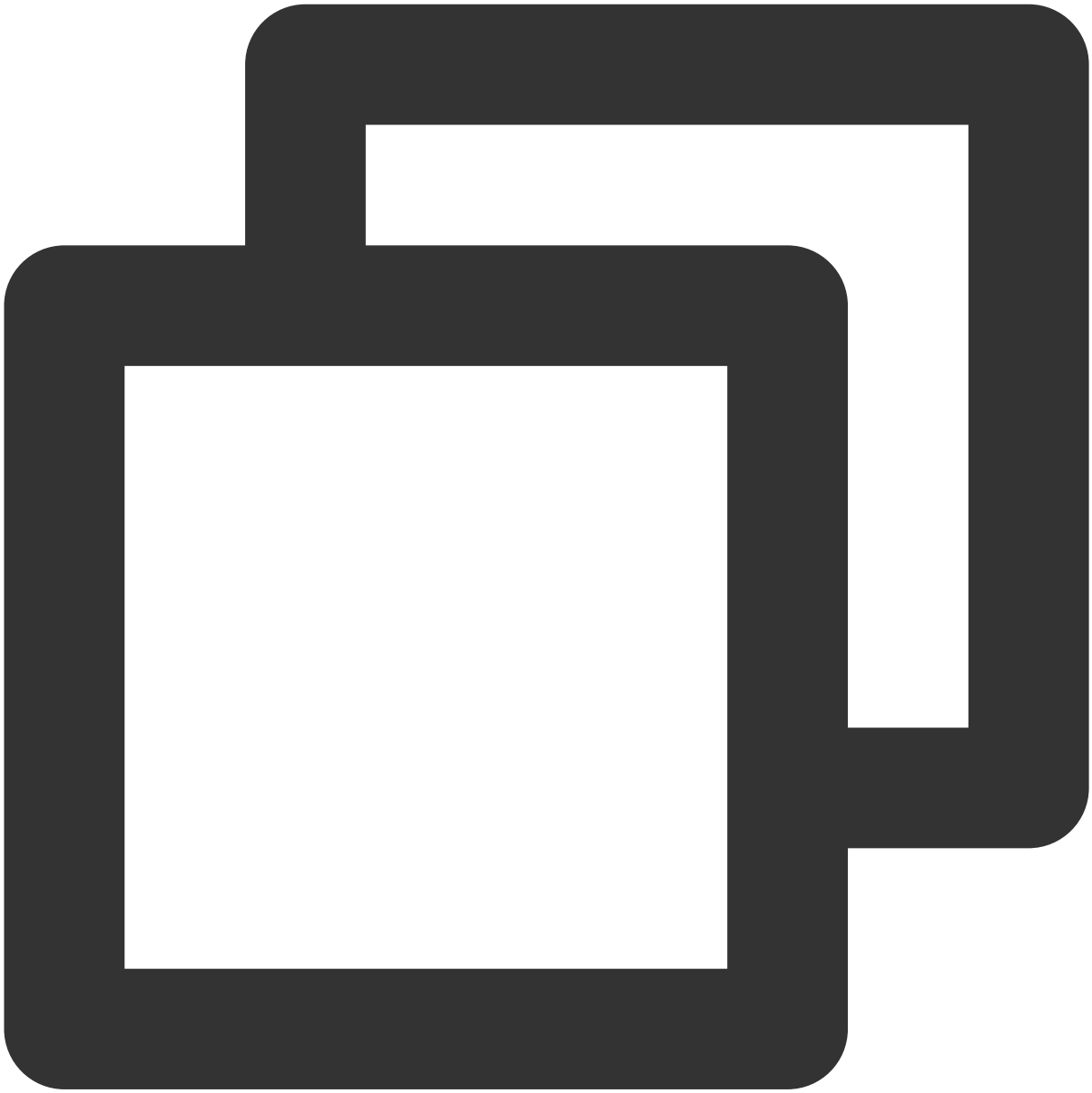


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoMontage</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <VideoMontage>
            <Container>
                <Format>mp4</Format>
            </Container>
            <Video>
                <Codec>H.264</Codec>
                <Bitrate>1000</Bitrate>
                <Width>1280</Width>
                <Height></Height>
            </Video>
            <Audio>
                <Codec>aac</Codec>
                <Samplerate>44100</Samplerate>
                <Bitrate>128</Bitrate>
                <Channels>4</Channels>
            </Audio>
            <AudioMix>
                <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix.mp3
                <MixMode>Once</MixMode>
                <Replace>true</Replace>
            </AudioMix>
            <Duration>15</Duration>
        </VideoMontage>
        <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>
    <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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoMontage</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <VideoMontage>
                <Container>
                    <Format>mp4</Format>
                </Container>
                <Video>
                    <Codec>H.264</Codec>
                    <Bitrate>1000</Bitrate>
                    <Width>1280</Width>
                    <Height></Height>
                </Video>
                <Audio>
                    <Codec>aac</Codec>
                    <Samplerate>44100</Samplerate>
                    <Bitrate>128</Bitrate>
                    <Channels>4</Channels>
                </Audio>
                <AudioMix>
                    <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix
                    <MixMode>Once</MixMode>
                    <Replace>true</Replace>
                </AudioMix>
                <Duration>15</Duration>
            </VideoMontage>
            <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>
```

版权所有：腾讯云计算（北京）有限责任公司

提交视频增强任务

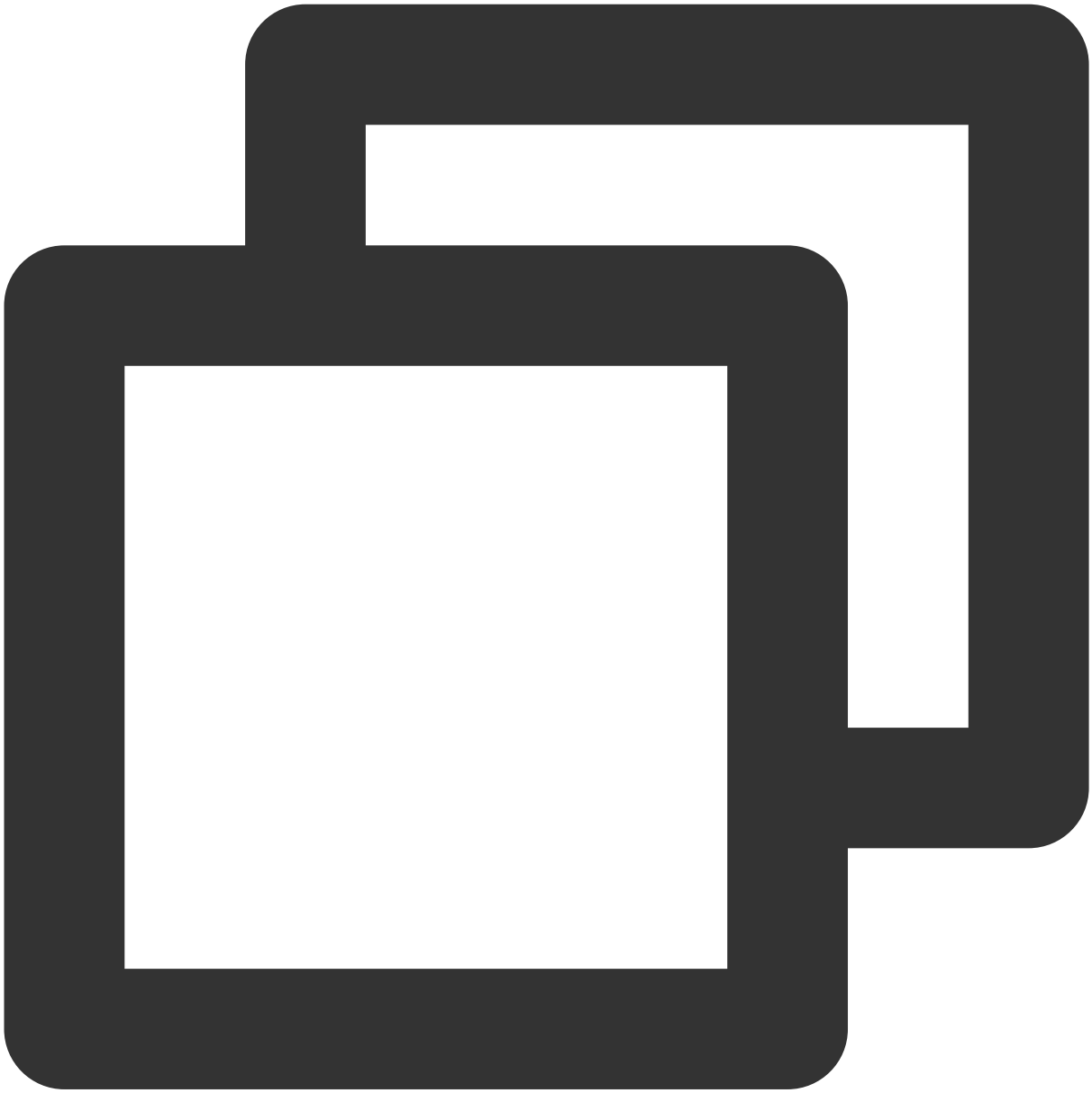
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个视频增强任务。

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：

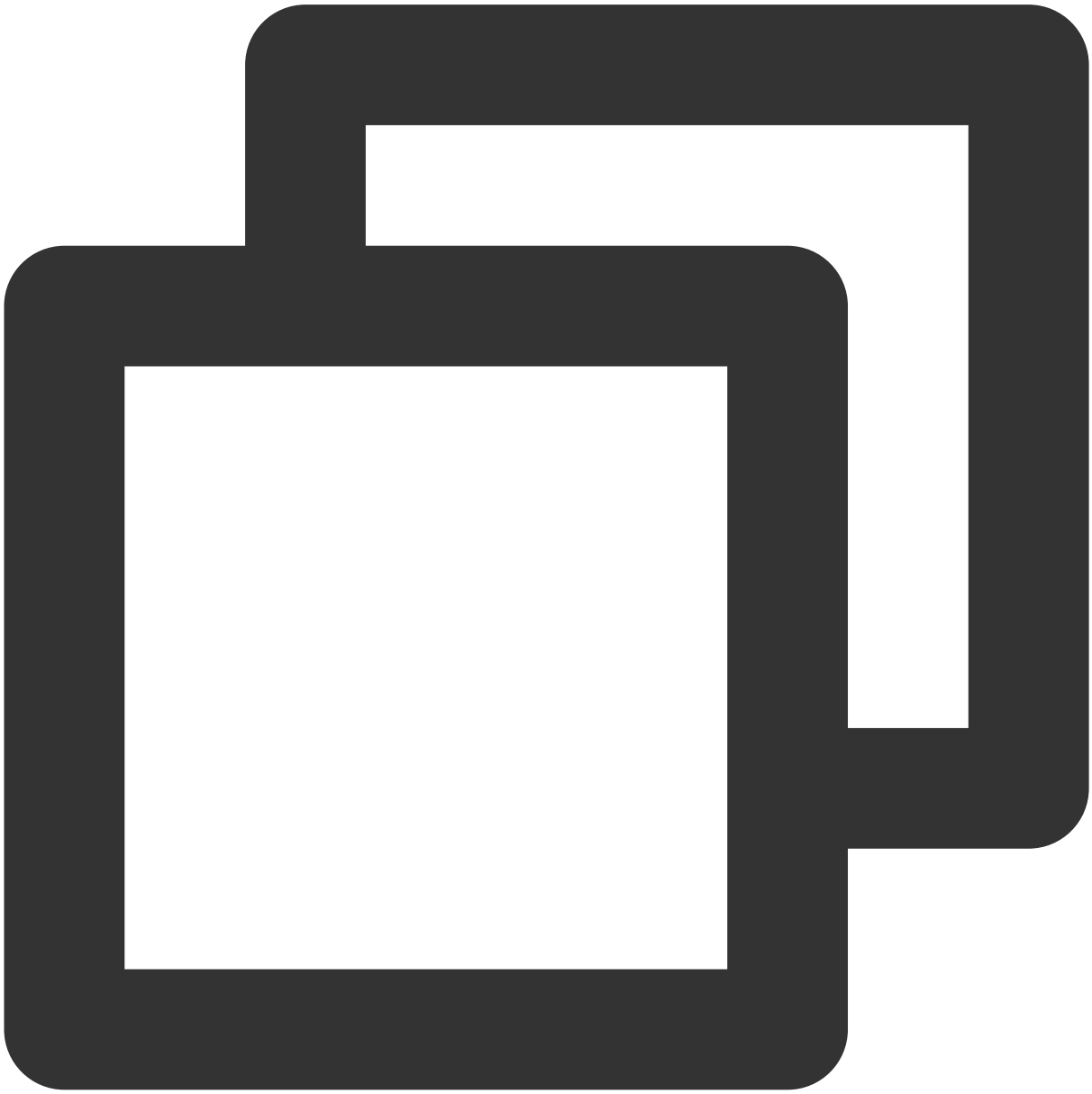
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>VideoProcess</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t156c107210e7243c5817354565d81b578</TemplateId>
        <TranscodeTemplateId>t1460606b9752148c4ab182f55163ba7cd</TranscodeTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemplateI
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <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：VideoProcess	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

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
Object	Request.Input	媒体文件名	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
VideoProcess	Request.Operation	指定视频增强模板参数，不能与 TemplateId 同时为空	Container	否
TemplateId	Request.Operation	指定的模板 ID，不能与 VideoProcess 同时为空	String	否
Transcode	Request.Operation	指定转码模板参数，不能与 TranscodeTemplateId 同时为空	Container	否
TranscodeTemplateId	Request.Operation	指定的转码模板 ID，优先使用模板 ID，不能与 Transcode 同时为空	String 数组	否
Watermark	Request.Operation	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口的 Request.Watermark, 最多传3个	Container 数组	否
WatermarkTemplateId	Request.Operation	指定的水印模板 ID，可以传多个水印模板 ID，最多传3个，优先使用模板 ID	String	否
DigitalWatermark	Request.Operation	指定数字水印参数	Container	否

Output	Request.Operation	结果输出地址	Container	是
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

说明：
提交视频增强任务必须传入转码参数。对于视频增强参数，优先使用 TemplateId，无 TemplateId 时使用 VideoProcess；对于转码参数，优先使用 TranscodeTemplateId，无 TranscodeTemplateId时使用 Transcode；对于水印参数，可以使用 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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
ColorEnhance	Request.Operation.VideoProcess	同创建视频增强模板 CreateMediaTemplate 接口中的 Request.ColorEnhance
MsSharpen	Request.Operation.VideoProcess	同创建视频增强模板 CreateMediaTemplate 接口中的 Request.MsSharpen

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	必选
TimeInterval	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Video	Container	否
Audio	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.Audio	Container	否
TransConfig	Request.Operation.Transcode	同创建转码模板 CreateMediaTemplate 接口中的 Request.TransConfig	Container	否

版权所有：腾讯云计算（北京）有限责任公司

AudioMix	Request.Operation.Transcode	混音参数，详情见 AudioMix	Container 数组	否

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.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.DigitalWatermark	水印类型，当前仅可设置为 Text	String	是
Version	Request.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String	是
IgnoreError	Request.Operation.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	string	是

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

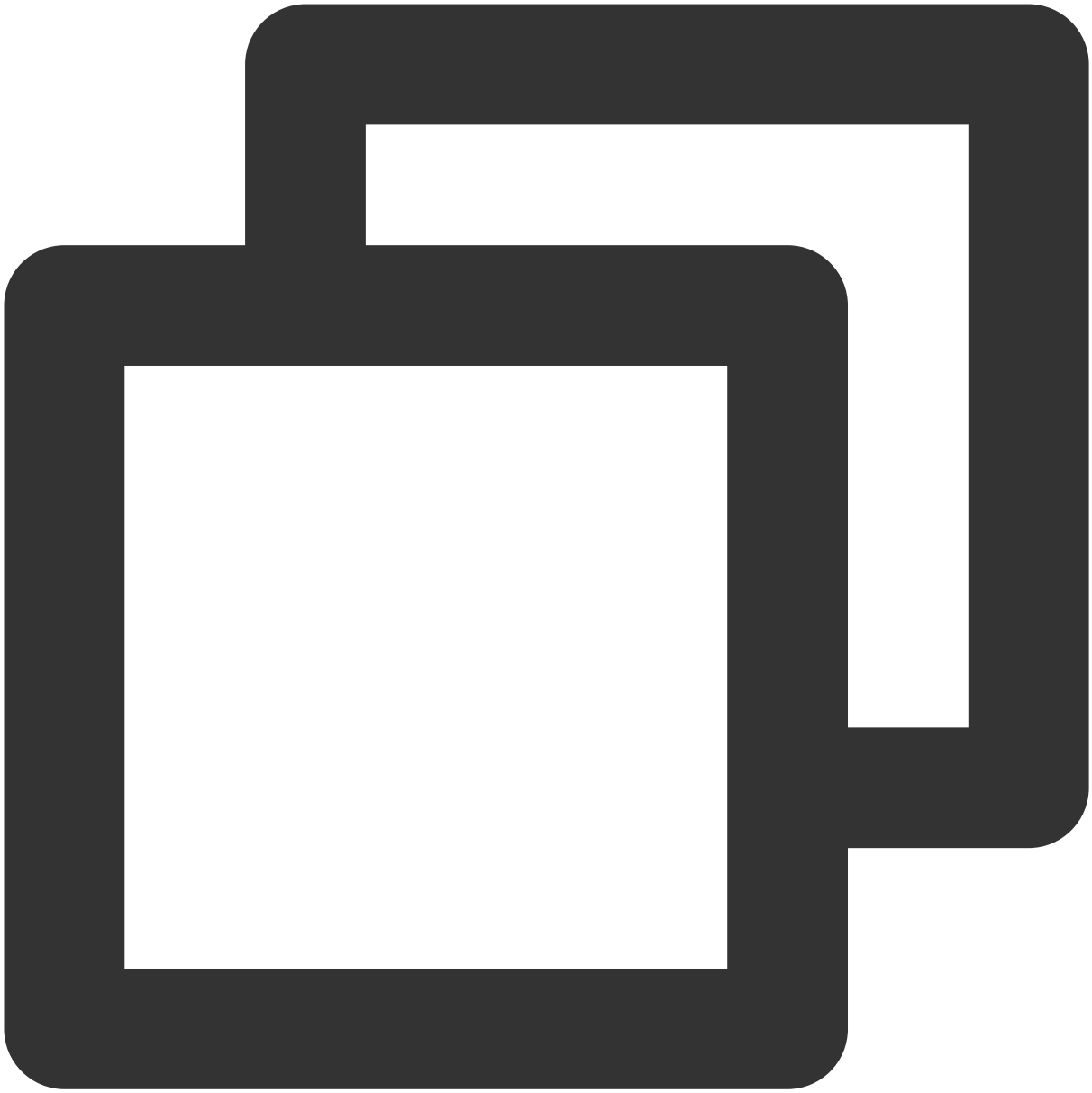
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoProcess</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t156c107210e7243c5817354565d81b578</TemplateId>
            <TemplateName>videoprocess_demo</TemplateName>
            <TranscodeTemplateId>t1460606b9752148c4ab182f55163ba7cd</TranscodeTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <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>
```

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：VideoProcess	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
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
VideoProcess	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoProcess	String
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
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container

版权所有：腾讯云计算（北京）有限责任公司

DigitalWatermark	Response.JobsDetail.Operation	指定数字水印参数	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
DigitalWatermark	Response.JobsDetail.Operation	指定数字水印参数	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 DigitalWatermark 的具体数据描述如下：

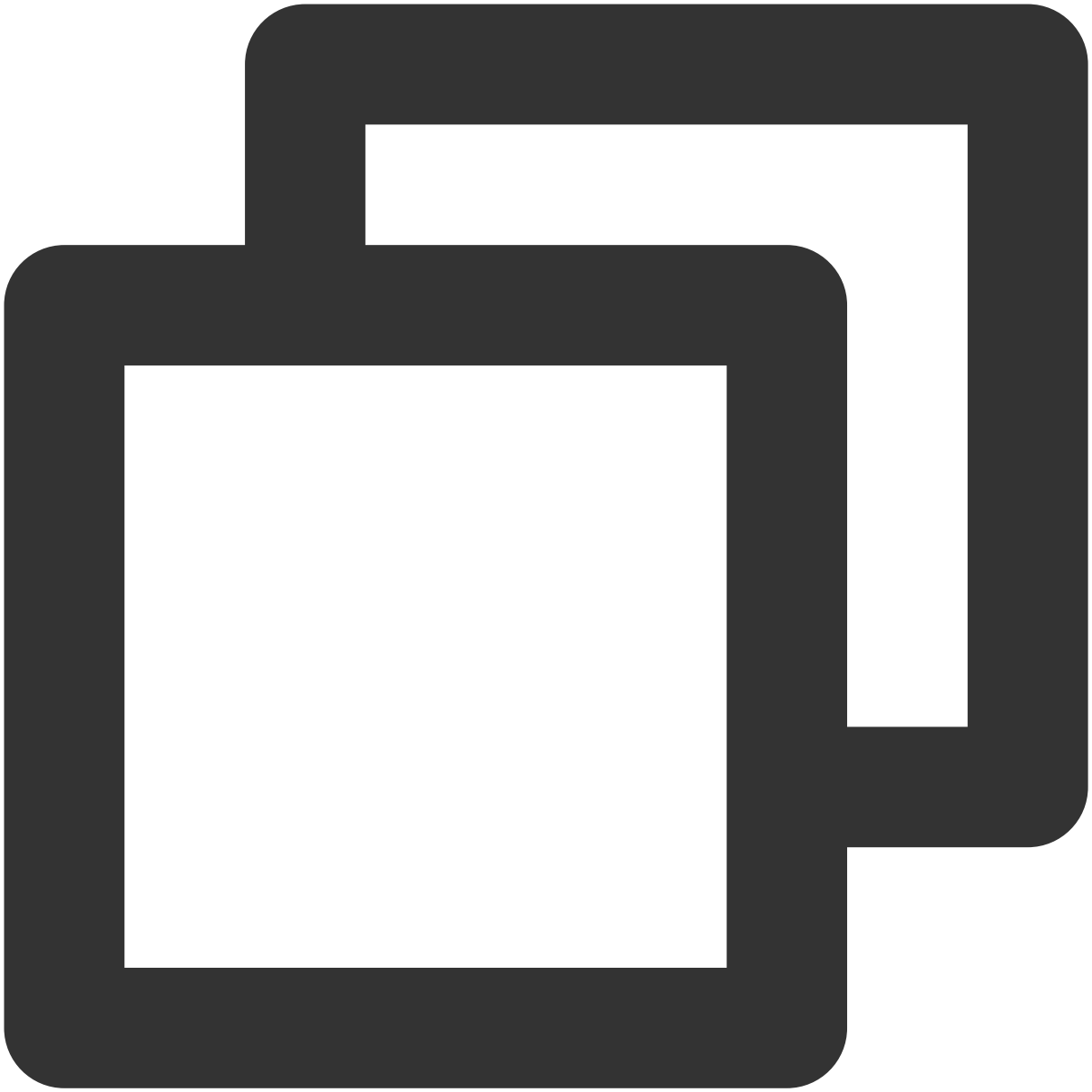
节点名称（关键字）	父节点	描述	类型
Message	Response.Operation.DigitalWatermark	成功嵌入视频的数字水印的字符串信息，长度不超过64个字符，仅支持中文、英文、数字、_、-和*	string
Type	Response.Operation.DigitalWatermark	水印类型，当前仅可设置为 Text	String
Version	Response.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String
IgnoreError	Response.Operation.DigitalWatermark	当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 false	string
State	Response.Operation.DigitalWatermark	添加水印是否成功，执行中为 Running，成功为 Success，失败为 Failed	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：使用视频增强模板 ID

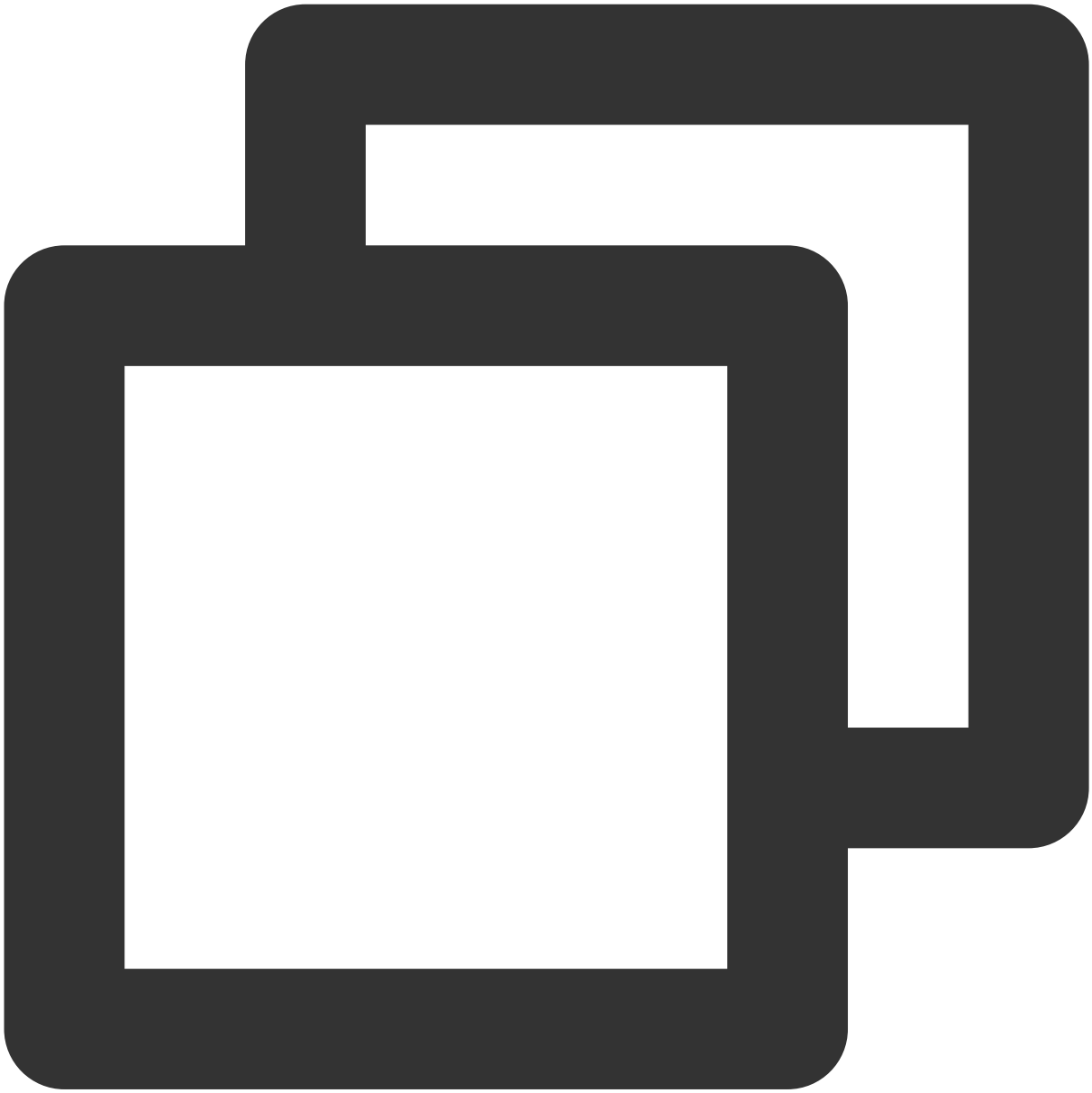


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoProcess</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t156c107210e7243c5817354565d81b578</TemplateId>
        <TranscodeTemplateId>t1460606b9752148c4ab182f55163ba7cd</TranscodeTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemplateI
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应

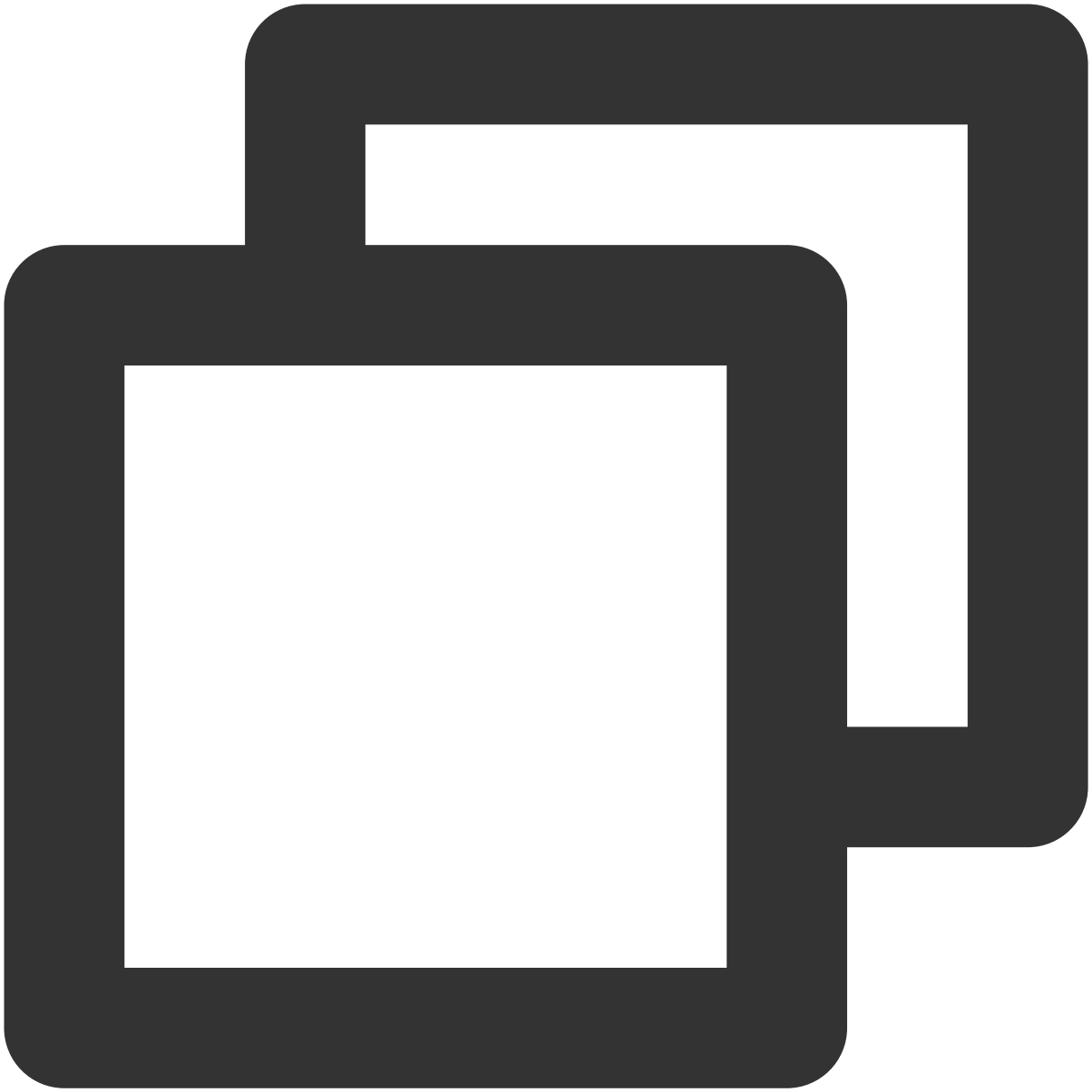


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoProcess</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t156c107210e7243c5817354565d81b578</TemplateId>
            <TemplateName>videoprocess_demo</TemplateName>
            <TranscodeTemplateId>t1460606b9752148c4ab182f55163ba7cd</TranscodeTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <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：使用视频增强处理参数



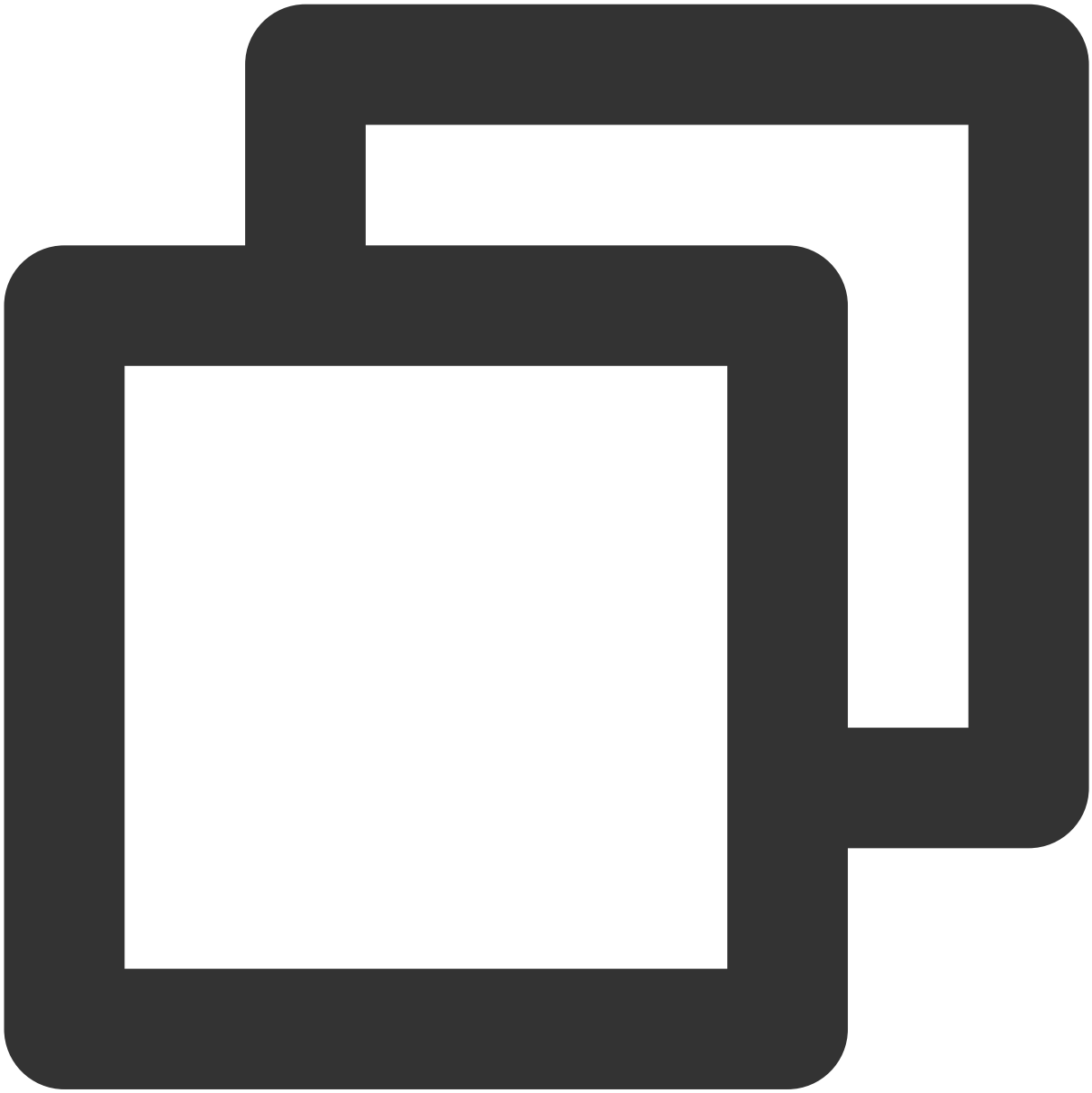
```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoProcess</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <VideoProcess>
            <ColorEnhance>
                <Contrast>50</Contrast>
                <Correction>100</Correction>
                <Enable>true</Enable>
                <Saturation>100</Saturation>
            </ColorEnhance>
            <MsSharpen>
                <Enable>true</Enable>
                <SharpenLevel>5</SharpenLevel>
            </MsSharpen>
        </VideoProcess>
        <Transcode>
            <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>
        </Transcode>
        <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>simfang.ttf</FontType>
                <FontColor>0xRRGGBB</FontColor>
                <Transparency>30</Transparency>
            </Text>
        </Watermark>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T14:44:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoProcess</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <VideoProcess>
                <ColorEnhance>
                    <Contrast>50</Contrast>
                    <Correction>100</Correction>
                    <Enable>true</Enable>
                    <Saturation>100</Saturation>
                </ColorEnhance>
                <MsSharpen>
                    <Enable>true</Enable>
                    <SharpenLevel>5</SharpenLevel>
                </MsSharpen>
            </VideoProcess>
            <Transcode>
                <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>
            </Transcode>
            <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>simfang.ttf</FontType>
                    <FontColor>0xRRGGBB</FontColor>
                    <Transparency>30</Transparency>
                </Text>
            </Watermark>
            <DigitalWatermark>
                <Type>Text</Type>
                <Message>123456789ab</Message>
                <Version>V1</Version>
                <IgnoreError>false</IgnoreError>
            </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>
```

提交视频标签任务

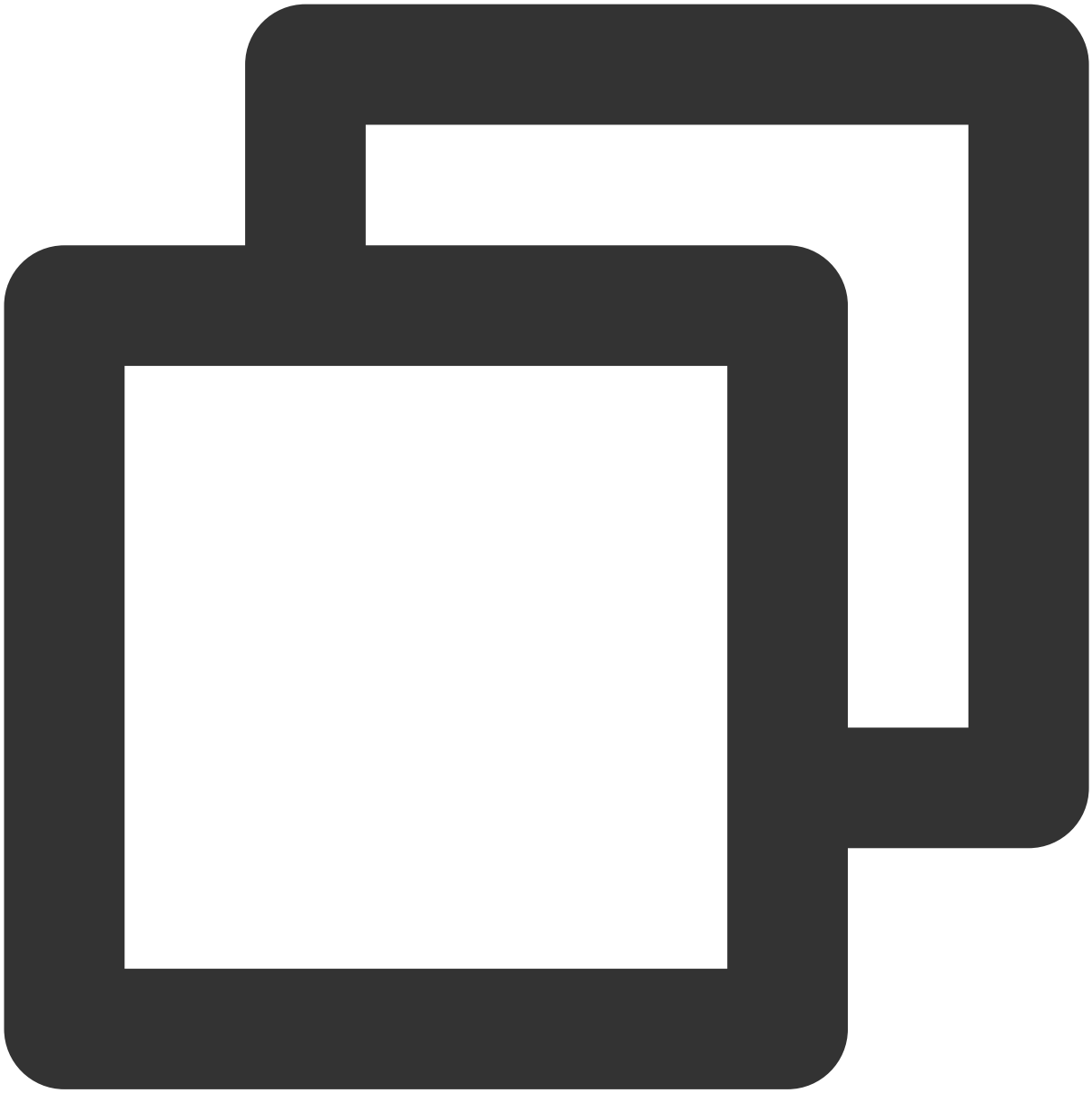
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个视频标签任务。

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：

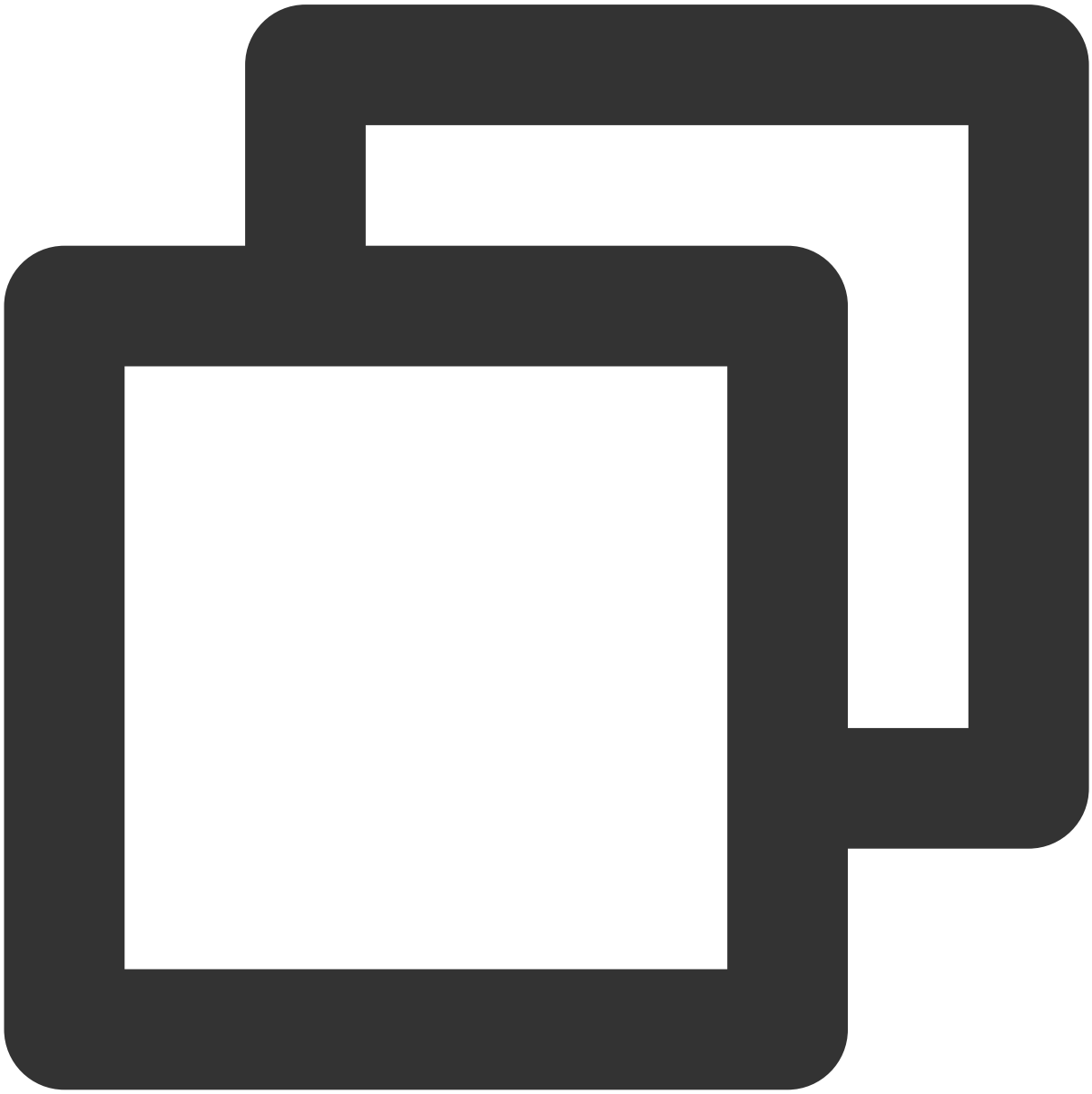
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>VideoTag</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <VideoTag>
            <Scenario>Stream</Scenario>
        </VideoTag>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：VideoTag	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
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

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	执行视频标签任务的媒体文件名，目前支持 mp4、avi、mkv、wmv、rmvb、flv、mov 封装格式，视频时长超过30min的视频请 提交工单 处理	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
VideoTag	Request.Operation	指定该 VideoTag 任务参数	Container	是
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
UserData	Response.JobsDetail.Operation	透传用户信息	String	否

Container 类型 VideoTag 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
Scenario	Request.Operation.VideoTag	场景类型，可选择视频标签的运用场景，不同的运用场景使用的算法、输入输出等都会有所差异	string	是	当前版本只适配 Stream 场景

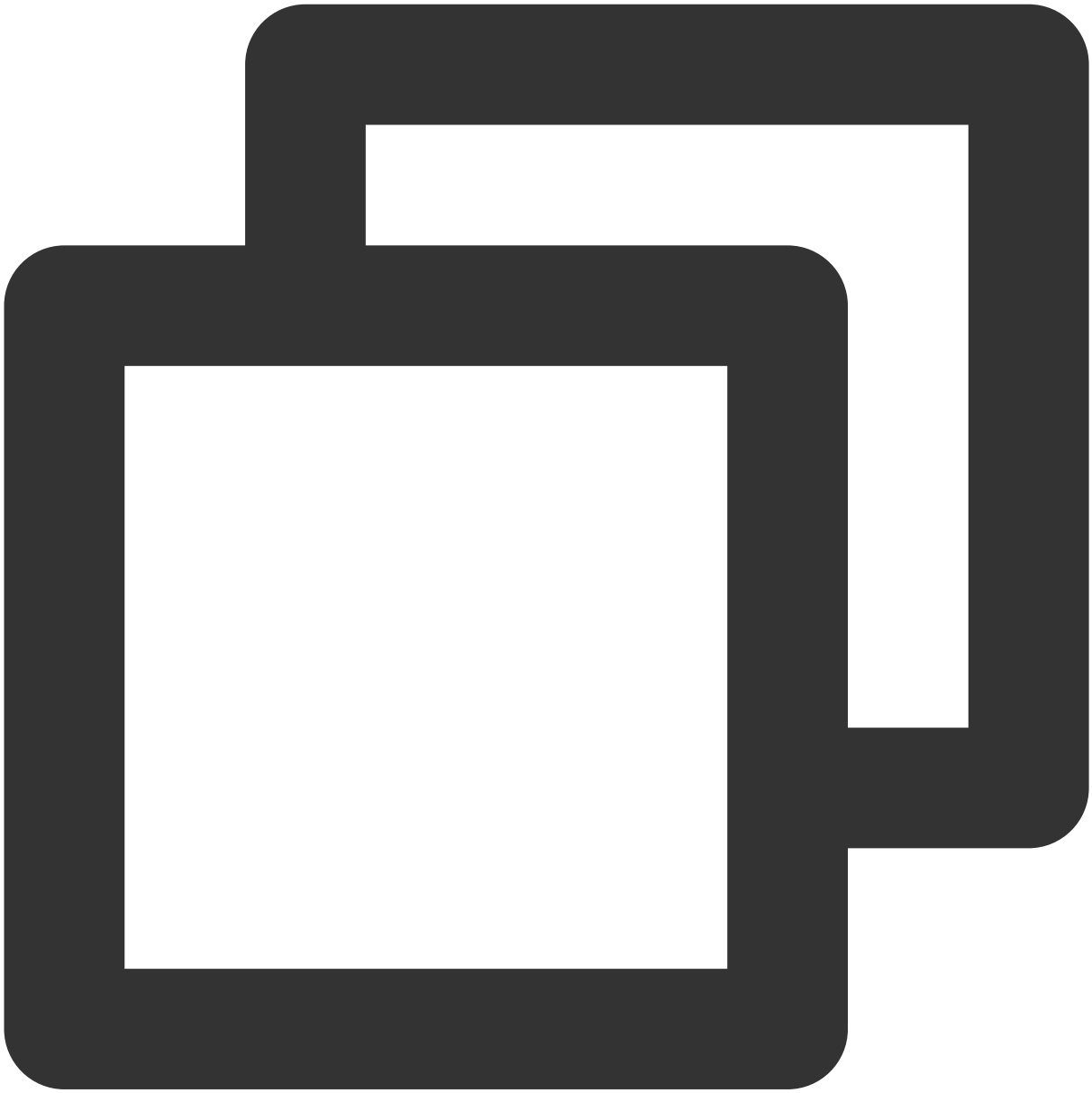
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoTag</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <VideoTag>
                <Scenario>Stream</Scenario>
            </VideoTag>
            <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：VideoTag	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
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
VideoTag	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoTag	Container
VideoTagResult	Response.JobsDetail.Operation	视频标签分析结果,任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 VideoTagResult 的内容：

节点名称（关键字）	父节点	描述	类型
StreamData	Response.JobsDetail.Operation.VideoTagResult	Stream 场景下视频标签任务的结果	Container

Container 节点 StreamData 的内容：

节点名称（关键字）	父节点	描述	类型
Data	Response.JobsDetail.Operation.VideoTagResult.StreamData	Stream 场景	Container

版权所有：腾讯云计算（北京）有限责任公司

		视频标签任务的结果列表	
SubErrCode	Response.JobsDetail.Operation.VideoTagResult.StreamData	算法状态码，成功为0，非0异常	String
SubErrMsg	Response.JobsDetail.Operation.VideoTagResult.StreamData	算法错误描述，成功为ok，不成功返回对应错误	String

Container 节点 Data 的内容：

节点名称（关键字）	父节点	描述	类型
Tags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data	视频标签、视频分类信息	Container
PersonTags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data	人物标签信息	Container数组
PlaceTags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data	场景标签信息	Container数组
ActionTags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data	动作标签信息	Container数组
ObjectTags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data	物体标签信息	Container数组

Container 节点 Tags （视频标签）的内容：

节点名称（关键字）	父节点	描述	类型
Tag	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	标签名称	String
TagCls	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	标签分类名称	String
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	标签模	Float

版权所有：腾讯云计算（北京）有限责任公司

		型预测分数，取值范围[0,1]	

Container 节点 Tags （视频分类）的内容：

节点名称（关键字）	父节点	描述	类型
Tag	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	视频分类名称	String
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	标签模型预测分数，取值范围[0,1]	Float

Container 节点 PersonTags 的内容：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	人物名字	Stri
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	标签模型预测分数	Flo
Count	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	任务出现频次	Stri
DetailPerSecond	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	具	Co

版权所有：腾讯云计算（北京）有限责任公司

		体识别到人物出现的位置和时间信息	数组

Container 节点 DetailPerSecond 的内容：

节点名称（关键字）	父节点
TimeStamp	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond
BBox	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond

Container 节点 BBox 的内容：

节点	父节点

版权所有：腾讯云计算（北京）有限责任公司

名称（关键字）	
X1	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox
Y1	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox
X2	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox
Y2	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox

Container 节点 PlaceTags 的内容：

节点名称（关键字）	父节点	描述

Tags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	视频场景标 Response.J
StartTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段起始时
EndTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段结束时
StartIndex	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段起始帧
EndIndex	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段结束帧
ClipFrameResult	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	单帧识别结

Container 节点 ActionTags 的内容：

节点名称（关键字）	父节点	描述
Tags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ActionTags	视频动作标签信息 Response.JobsDe
StartTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ActionTags	片段起始时间，单
EndTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ActionTags	片段结束时间，单

Container 节点 ObjectTags 的内容：

节点名称（关键字）	父节点	描述
Objects	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags	视频物体标签信 Response.Jobs
TimeStamp	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags	识别物体时间戳

Container 节点 Object 的内容：

节点名称（关键字）	父节点	描述
Name	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags.Objects	物体名称
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags.Objects	标签模型
BBox	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags.Objects	左上角坐

版权所有：腾讯云计算（北京）有限责任公司

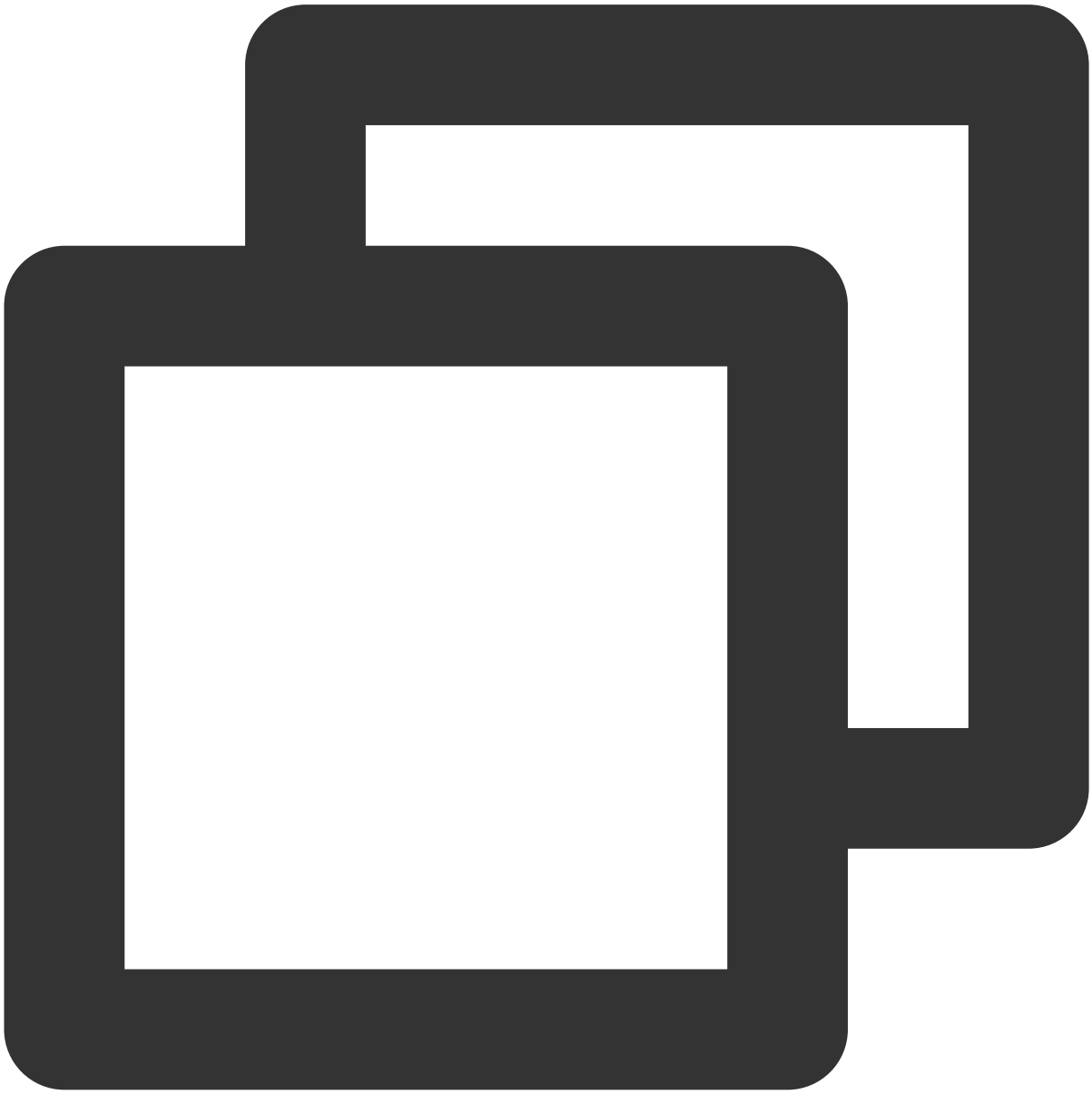
		Respon

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

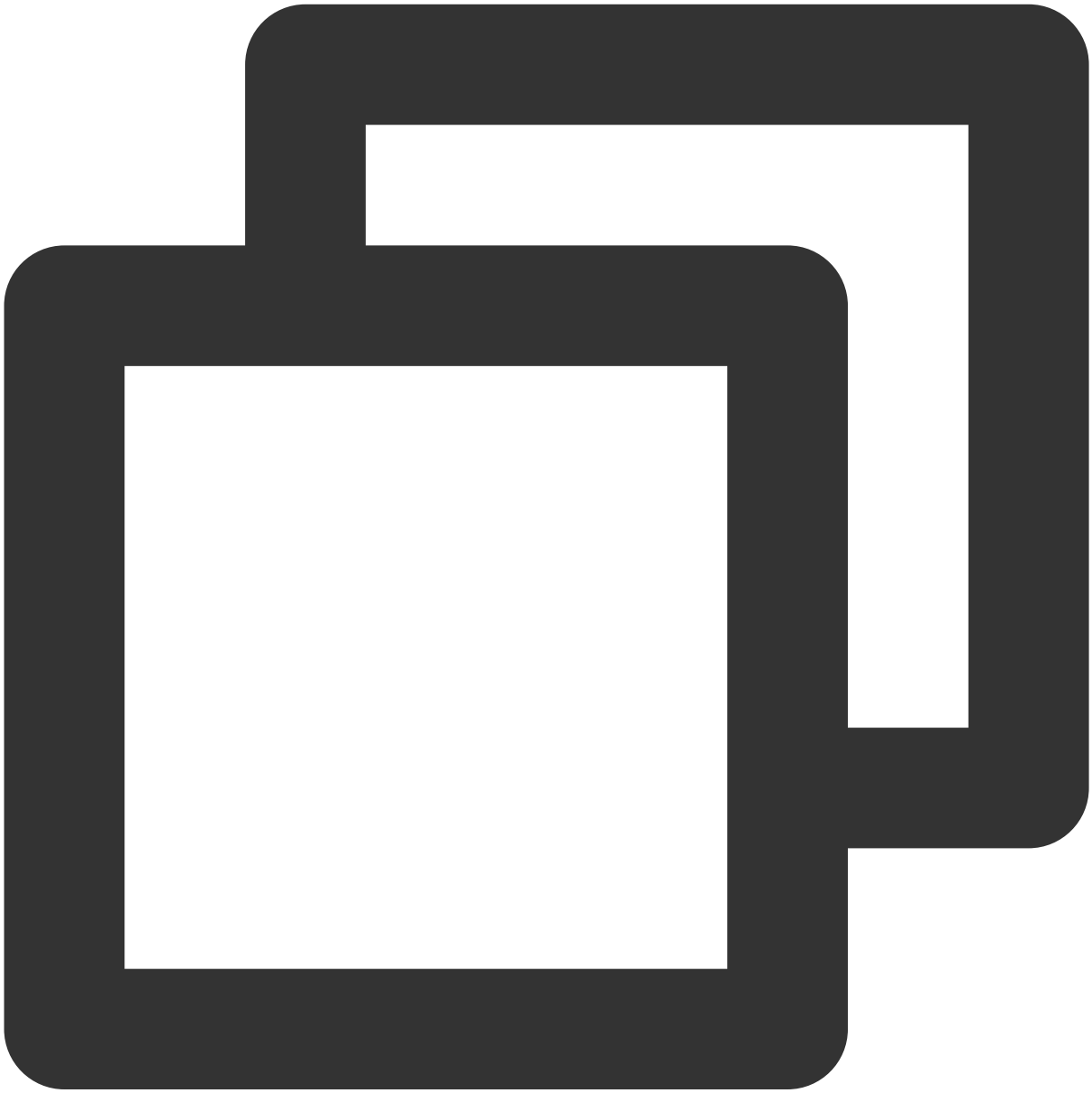
请求



```
POST /jobs 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>
    <Tag>VideoTag</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <VideoTag>
            <Scenario>Stream</Scenario>
        </VideoTag>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoTag</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <VideoTag>
                <Scenario>Stream</Scenario>
            </VideoTag>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交人声分离任务

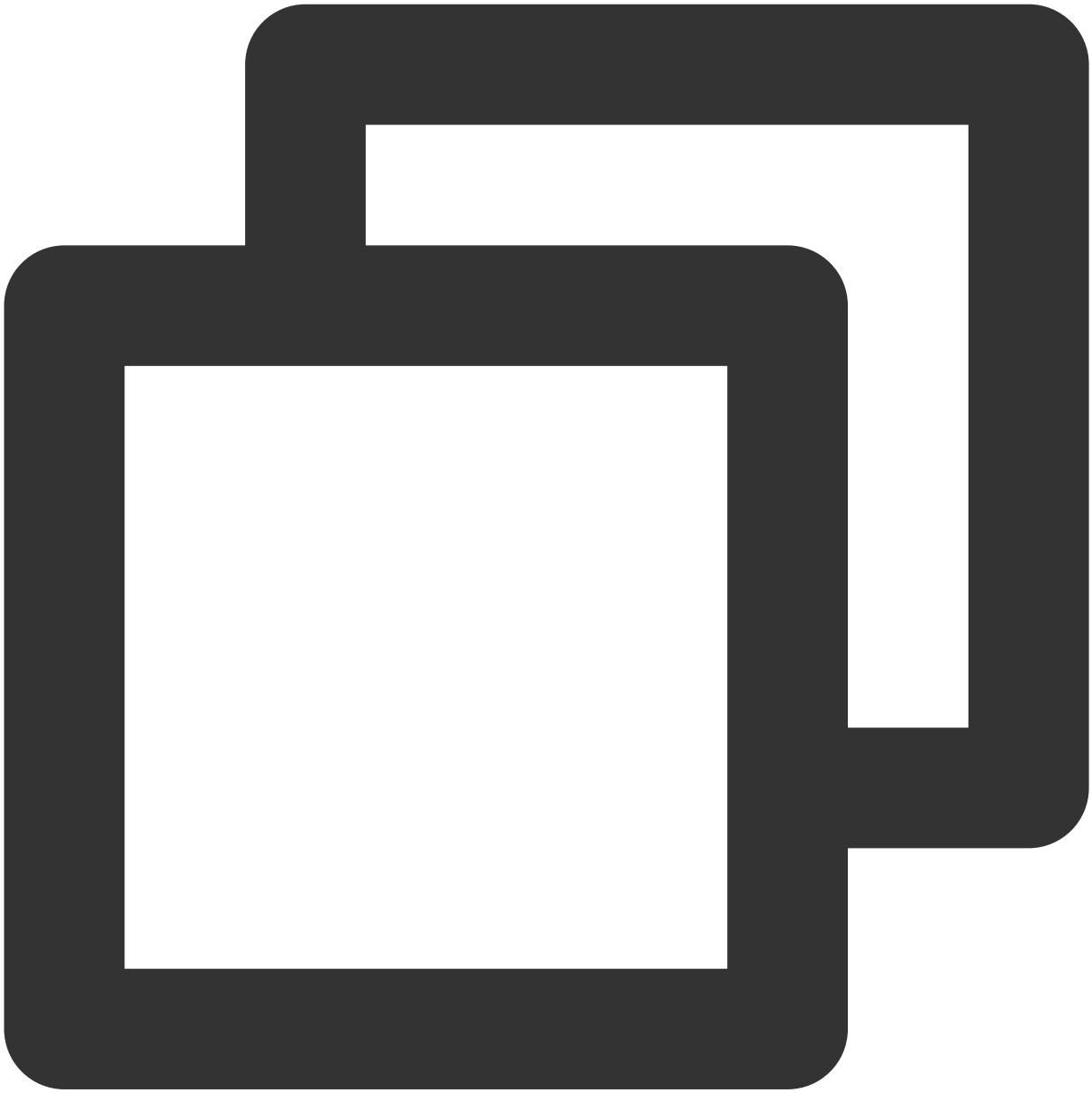
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个人声分离任务。

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：

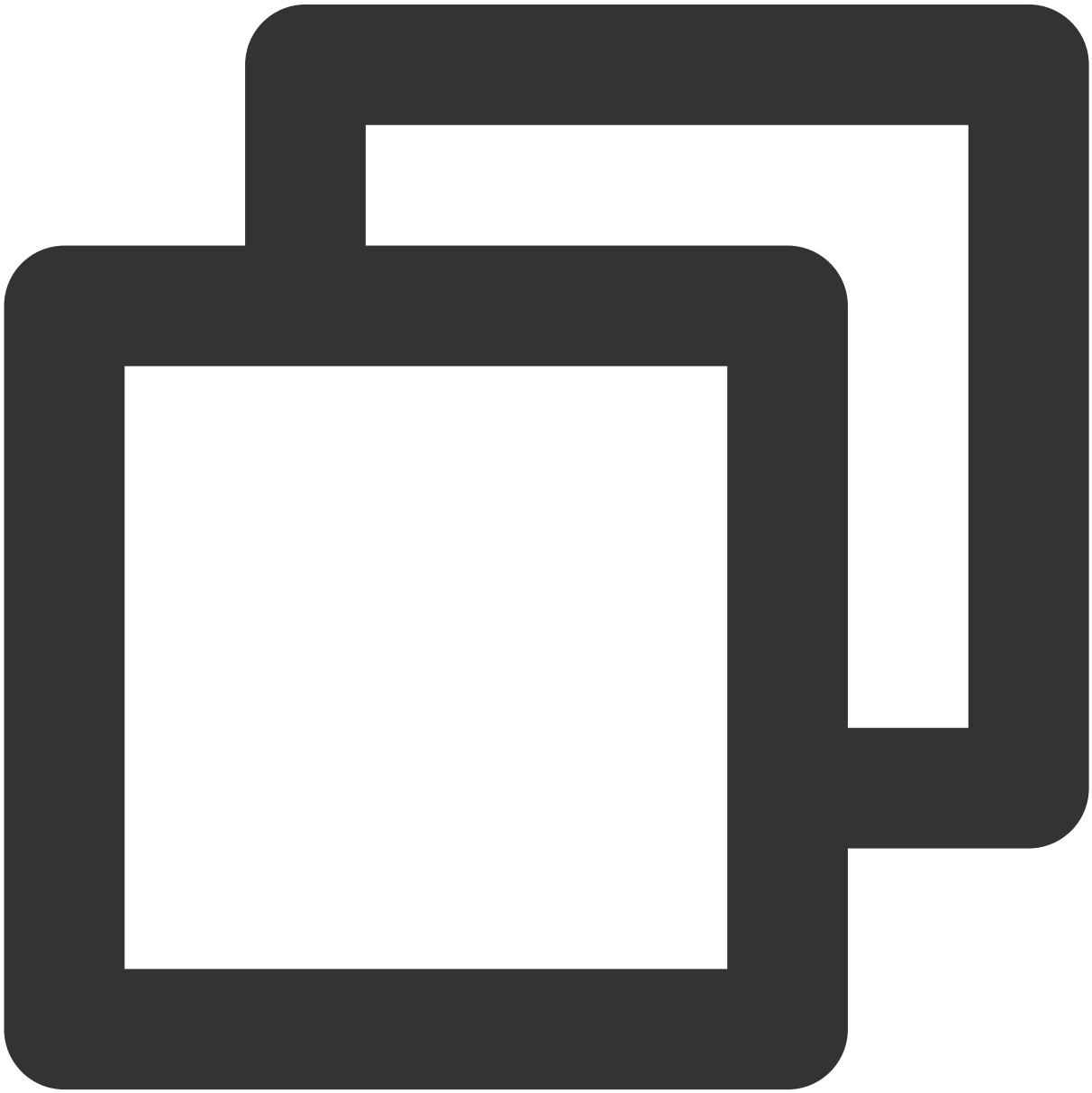
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>VoiceSeparate</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/backgroud.mp3</Object>
            <AuObject>output/audio.mp3</AuObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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：VoiceSeparate	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
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
VoiceSeparate	Request.Operation	指定转码模板参数	Container	否
TemplateId	Request.Operation	指定的模板 ID	String	否
Output	Request.Operation	结果输出地址	Container	是
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

注意：

优先使用 TemplateId，无 TemplateId 时使用 VoiceSeparate。

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
AudioMode	Request.Operation.VoiceSeparate	同创建人声分离模板 CreateMediaTemplate 接口中的 Request.AudioMode	Container	否
AudioConfig	Request.Operation.VoiceSeparate	同创建人声分离模板 CreateMediaTemplate 接口中的 Request.AudioConfig	Container	否

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

版权所有：腾讯云计算（北京）有限责任公司

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	背景音结果文件名，不能与 AuObject 同时为空	String	否
AuObject	Request.Operation.AuObject	人声结果文件名，不能与 Object 同时为空	String	否

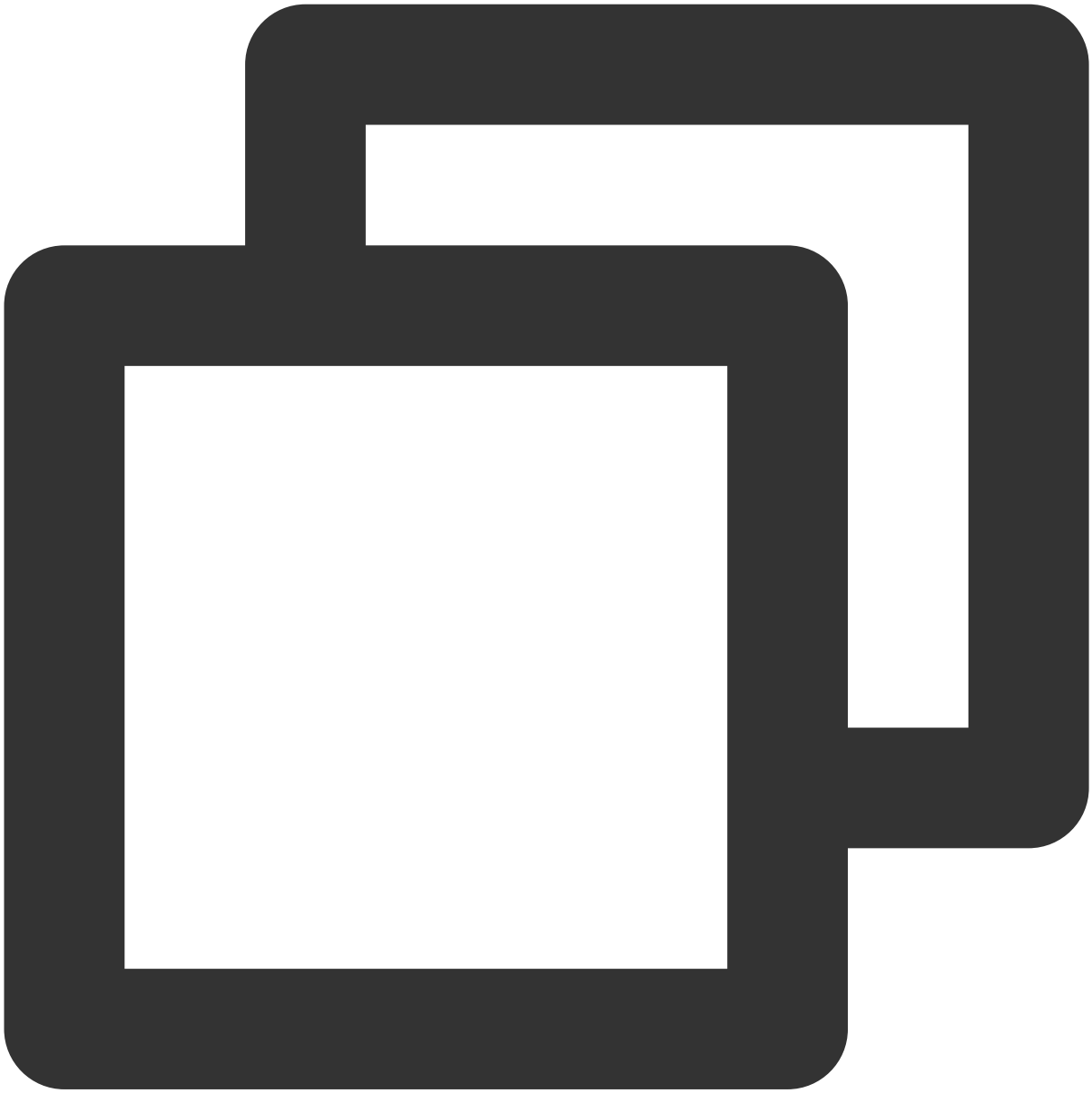
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VoiceSeparate</Tag>
```

```
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>voiceseparate_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/backgroud.mp3</Object>
                <AuObject>output/audio.mp3</AuObject>
            </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：VoiceSeparate	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	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
VoiceSeparate	Response.JobsDetail.Operation	同请求中的 Request.Operation.VoiceSeparate	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	转码输出视频的信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

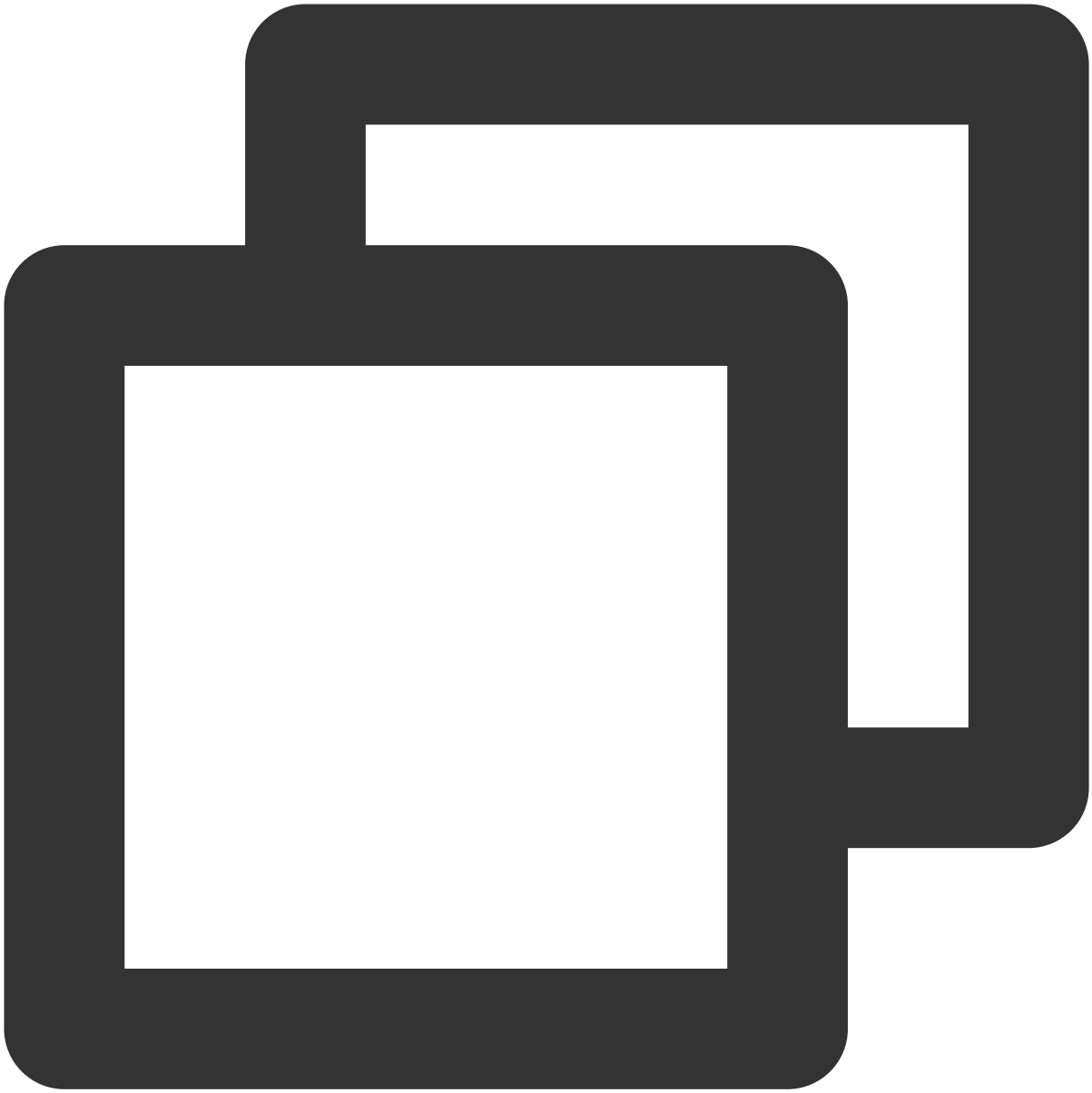
节点名称（关键字）	父节点	描述	类型
ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

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：使用人声分离模板 ID

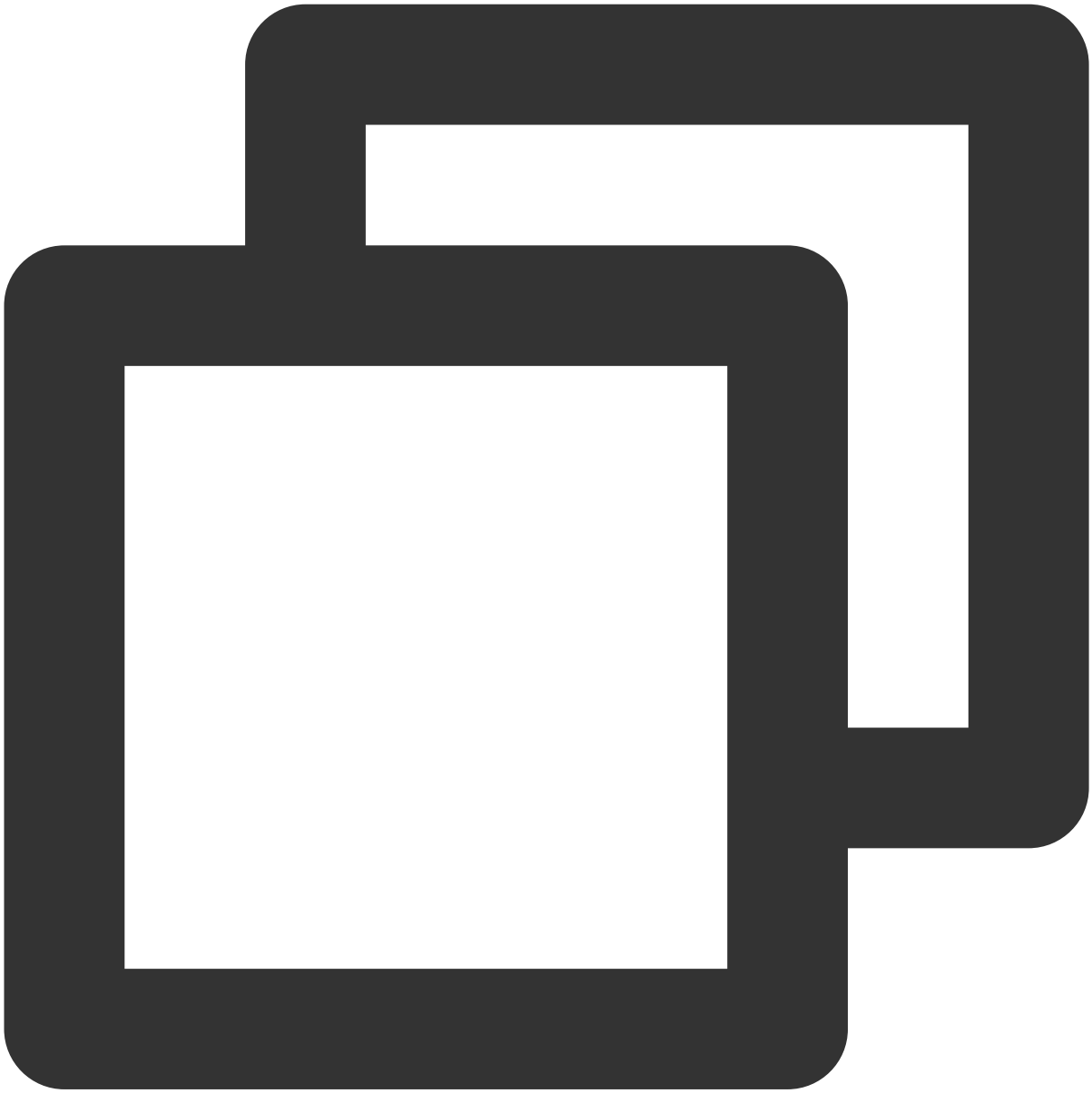


```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VoiceSeparate</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/backgroud.mp3</Object>
            <AuObject>output/audio.mp3</AuObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应1

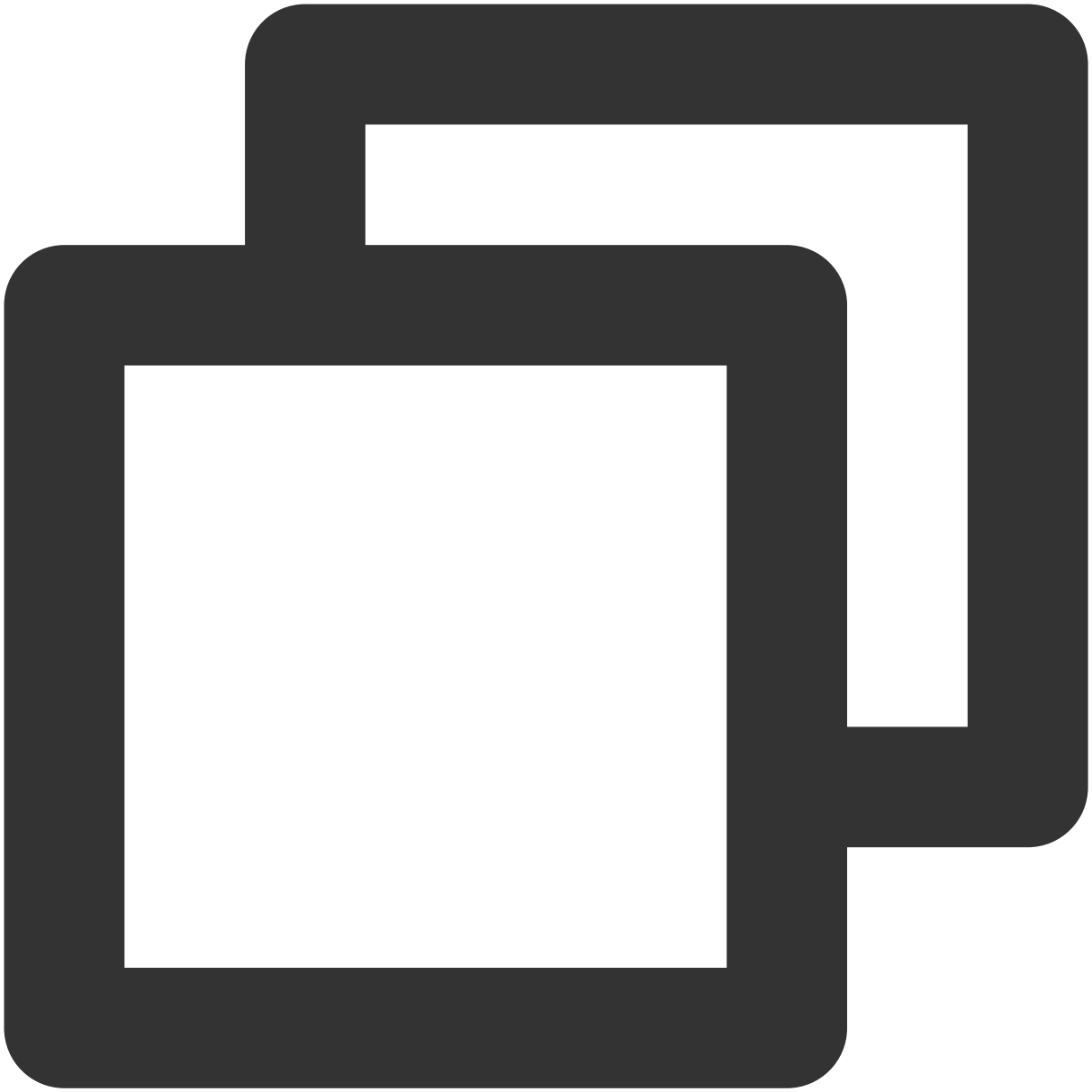


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VoiceSeparate</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>voiceseparate_demo</TemplateName>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/backgroud.mp3</Object>
                <AuObject>output/audio.mp3</AuObject>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求2：使用人声分离参数

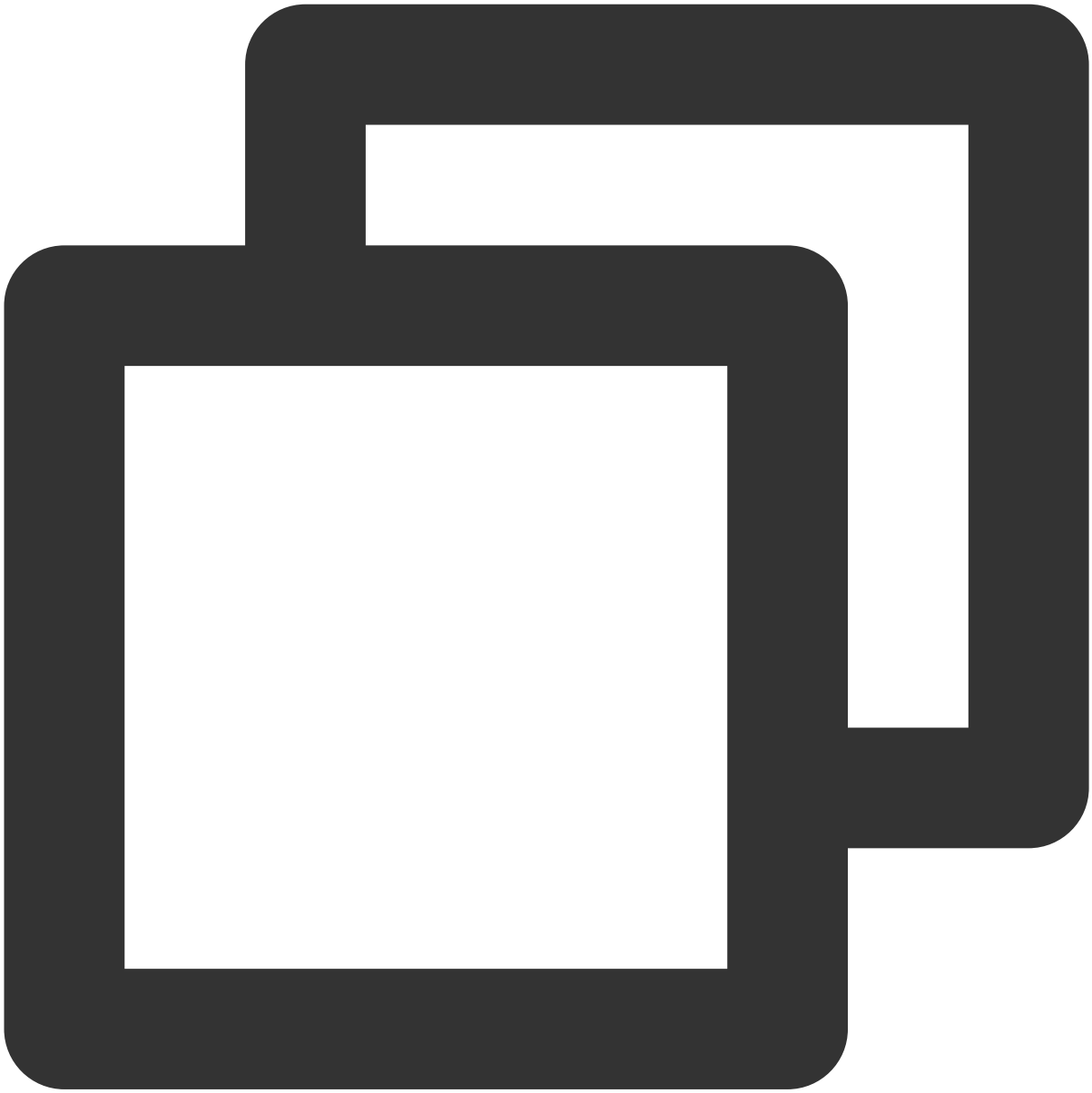


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VoiceSeparate</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <VoiceSeparate>
            <AudioConfig>
                    <Bitrate>500</Bitrate>
                    <Channels>2</Channels>
                    <Codec>mp3</Codec>
                    <Samplerate>44100</Samplerate>
            </AudioConfig>
            <AudioMode>AudioAndBackground</AudioMode>
        </VoiceSeparate>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/backgroud.mp3</Object>
            <AuObject>output/audio.mp3</AuObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VoiceSeparate</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <VoiceSeparate>
                <AudioConfig>
                    <Bitrate>500</Bitrate>
                    <Channels>2</Channels>
                    <Codec>mp3</Codec>
                    <Samplerate>44100</Samplerate>
                </AudioConfig>
                <AudioMode>AudioAndBackground</AudioMode>
            </VoiceSeparate>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/backgroud.mp3</Object>
                <AuObject>output/audio.mp3</AuObject>
            </Output>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

提交极速高清任务

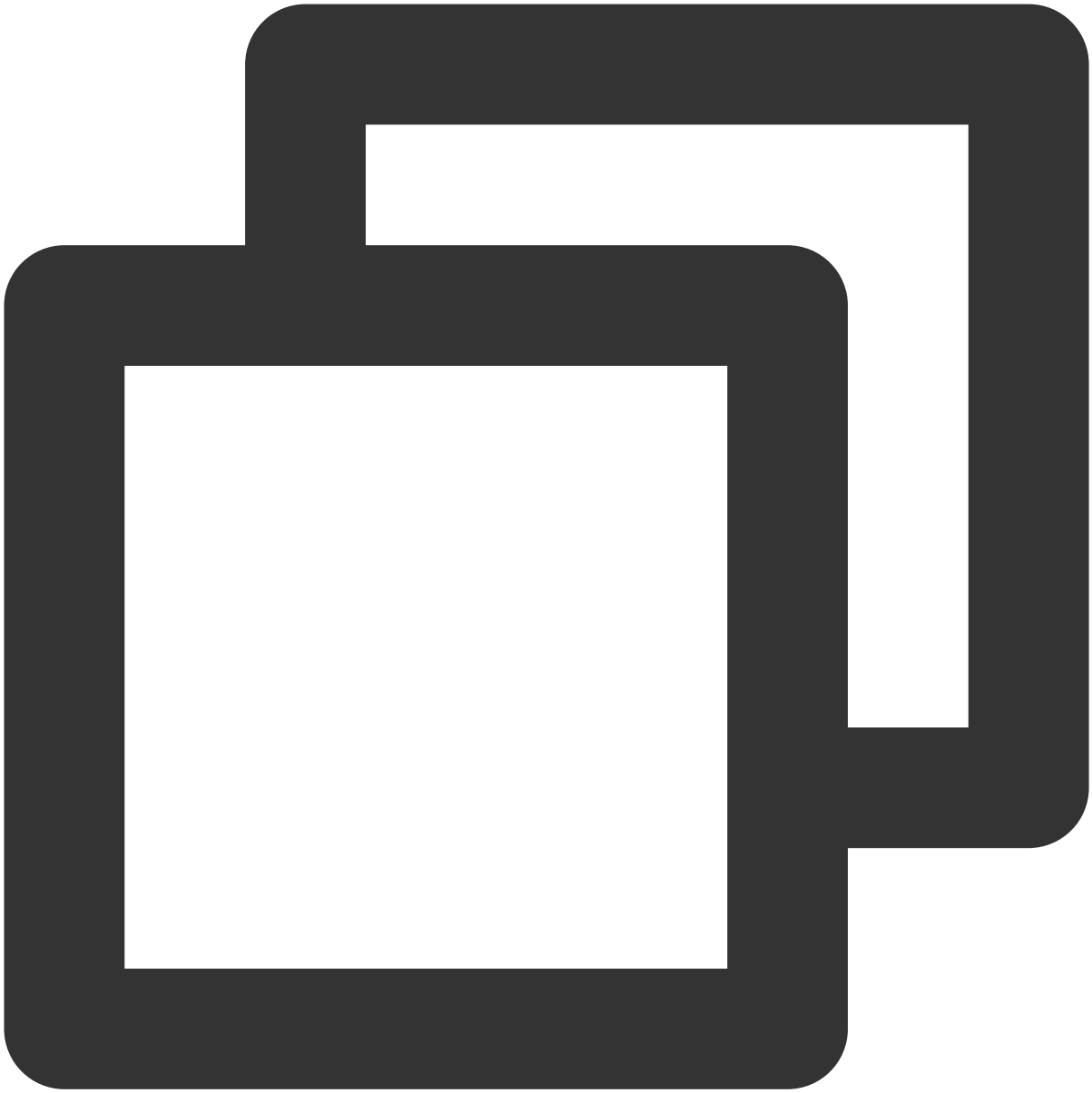
最近更新时间：2024-01-20 16:30:33

功能描述

提交一个极速高清转码任务。

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：

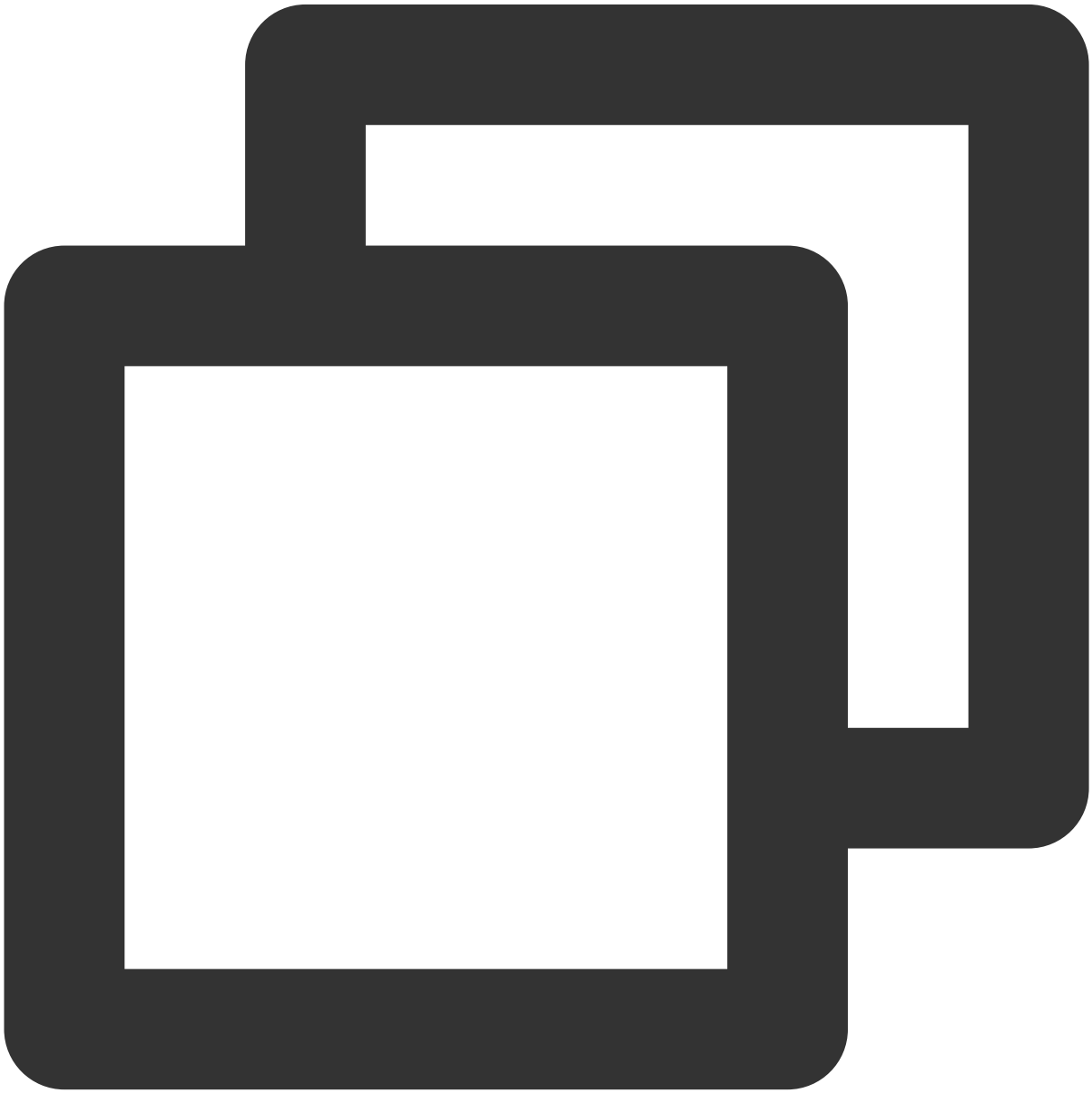
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
```

```
    <Tag>Transcode</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemplateI
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <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：Transcode	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

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
Object	Request.Input	媒体文件名	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
TemplateId	Request.Operation	指定的模板 ID, 此处需要传入极速高清模板	String	否
WatermarkTemplateId	Request.Operation	指定的水印模板 ID，可以传多个水印模板 ID	String	否
Watermark	Request.Operation	指定水印模板参数，同创建水印模板 CreateMediaTemplate 接口中的 Request.Watermark	Container	否
DigitalWatermark	Request.Operation	指定数字水印参数	Container	否
Output	Request.Operation	结果输出地址	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

说明：

对于水印参数，可以使用 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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.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.DigitalWatermark	水印类型，当前仅可设置为 Text	String	是
Version	Request.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String	是
IgnoreError	Request.Operation.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	string	是

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

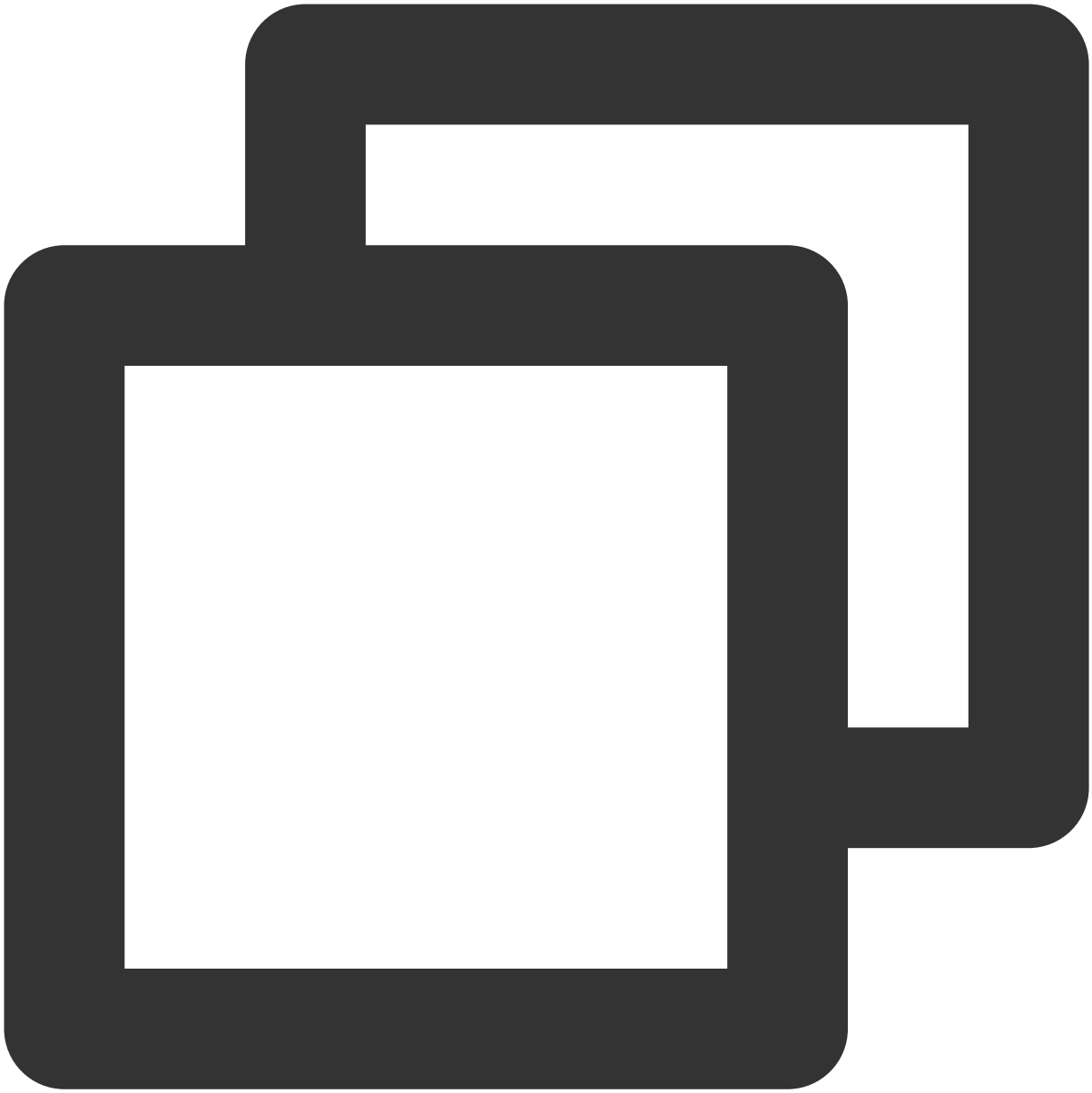
响应

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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <Progress>0</Progress>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
        <StartTime>-</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>
            <TemplateName>high_993874</TemplateName>
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <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>
```

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

版权所有：腾讯云计算（北京）有限责任公司

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：Transcode	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
Progress	Response.JobsDetail	任务进度百分比，只有在State为 Submitted、Running、Success、Pause 时有意义，范围为[0, 100]	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
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
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组

版权所有：腾讯云计算（北京）有限责任公司

WatermarkTemplateId	Response.JobsDetail.Operation	指定的水印模板 ID	String 数组
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回	Container
DigitalWatermark	Response.JobsDetail.Operation	指定数字水印参数	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String

Container 节点 MediaInfo 的内容：

同 GenerateMediaInfo 接口中的 Response.MediaInfo 节点。

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	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	父节点	描述	类型
Bucket	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶	String
Region	Response.Operation.MediaResult.OutputFile	输出文件所在的存储桶所在的园区	String
ObjectName	Response.Operation.MediaResult.OutputFile	输出文件名，可能有多个	String 数组
Md5Info	Response.Operation.MediaResult.OutputFile	输出文件的 MD5 信息	Container 数组

Container 节点 Md5Info 的内容：

节点名称（关键字）	父节点	描述	类型

ObjectName	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件名	String
Md5	Response.Operation.MediaResult.OutputFile.Md5Info	输出文件的 MD5 值	Container

Container 类型 DigitalWatermark 的具体数据描述如下：

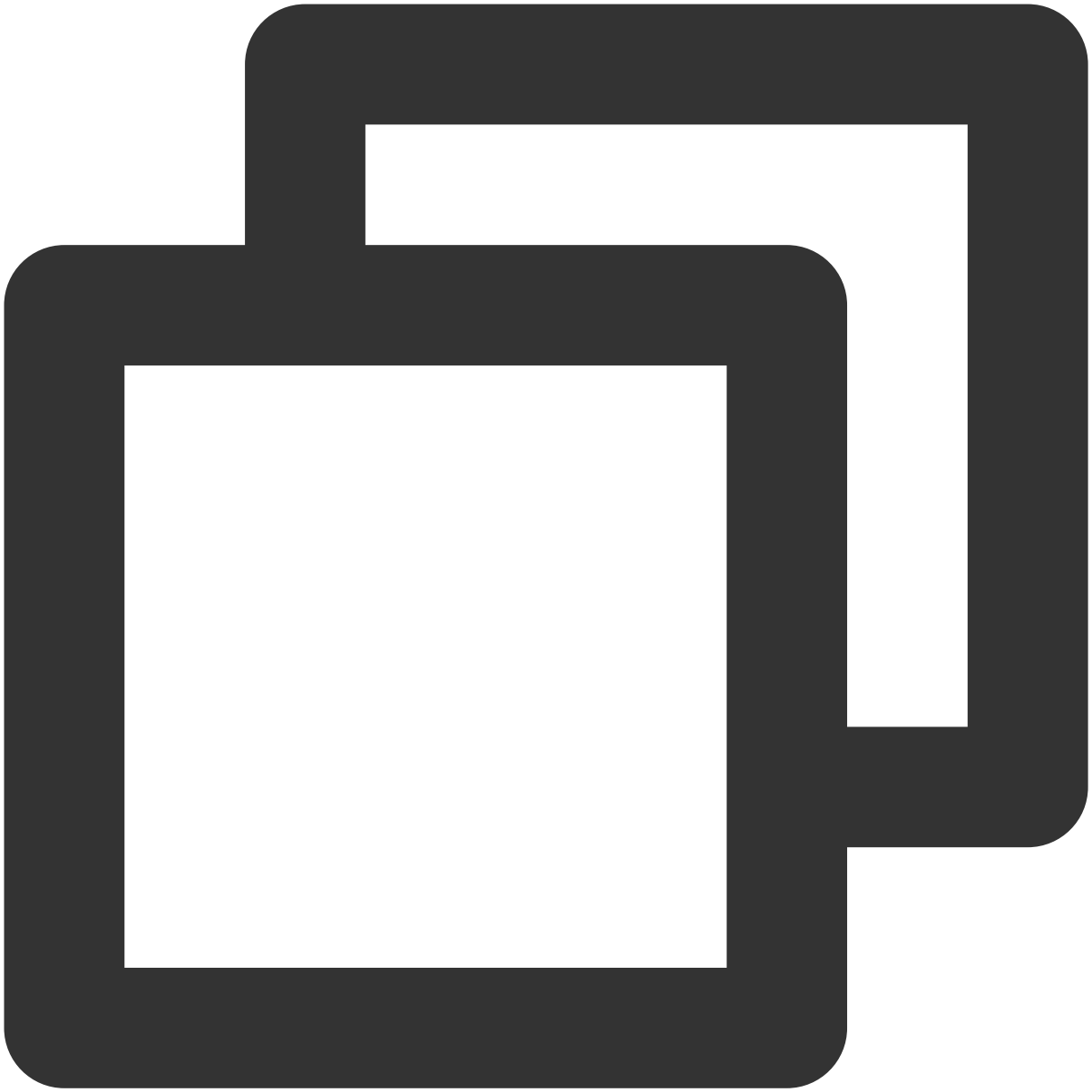
节点名称（关键字）	父节点	描述	类型
Message	Response.Operation.DigitalWatermark	成功嵌入视频的数字水印的字符串信息，长度不超过64个字符，仅支持中文、英文、数字、_、-和*	string
Type	Response.Operation.DigitalWatermark	水印类型，当前仅可设置为 Text	String
Version	Response.Operation.DigitalWatermark	水印版本，当前仅可设置为 V1	String
IgnoreError	Response.Operation.DigitalWatermark	当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false	string
State	Response.Operation.DigitalWatermark	添加水印是否成功，执行中为 Running，成功为 Success，失败为 Failed	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

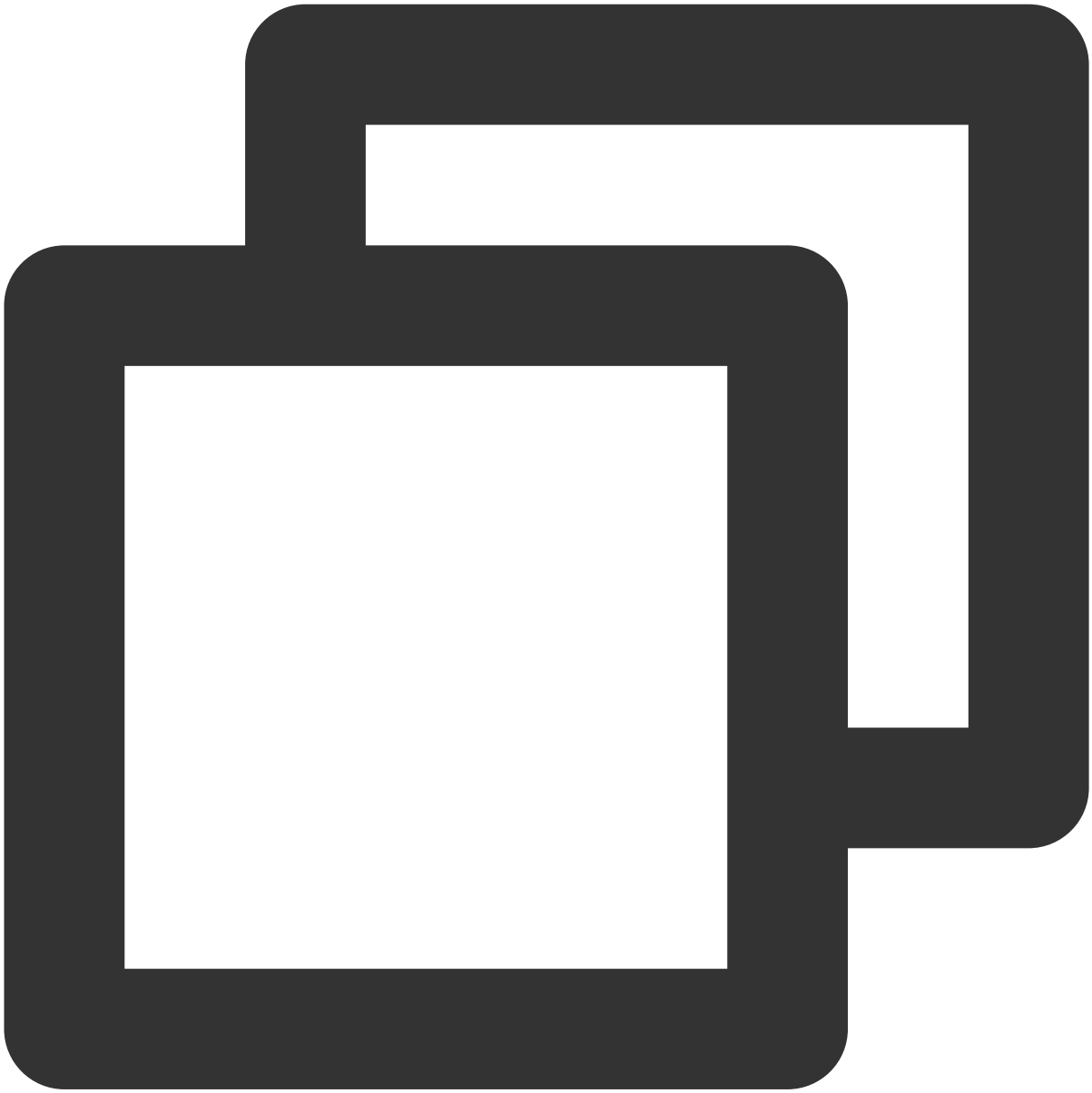


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Transcode</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
```

```
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemplateI
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemplateI
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </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>
    <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>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Submitted</State>
        <Progress>0</Progress>
        <CreationTime>2022-06-27T15:23:11+0800</CreationTime>
        <StartTime>-</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>
            <TemplateName>high_993874</TemplateName>
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTempl
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTempl
            <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>
```

取消媒体处理任务

CancelMediaJob

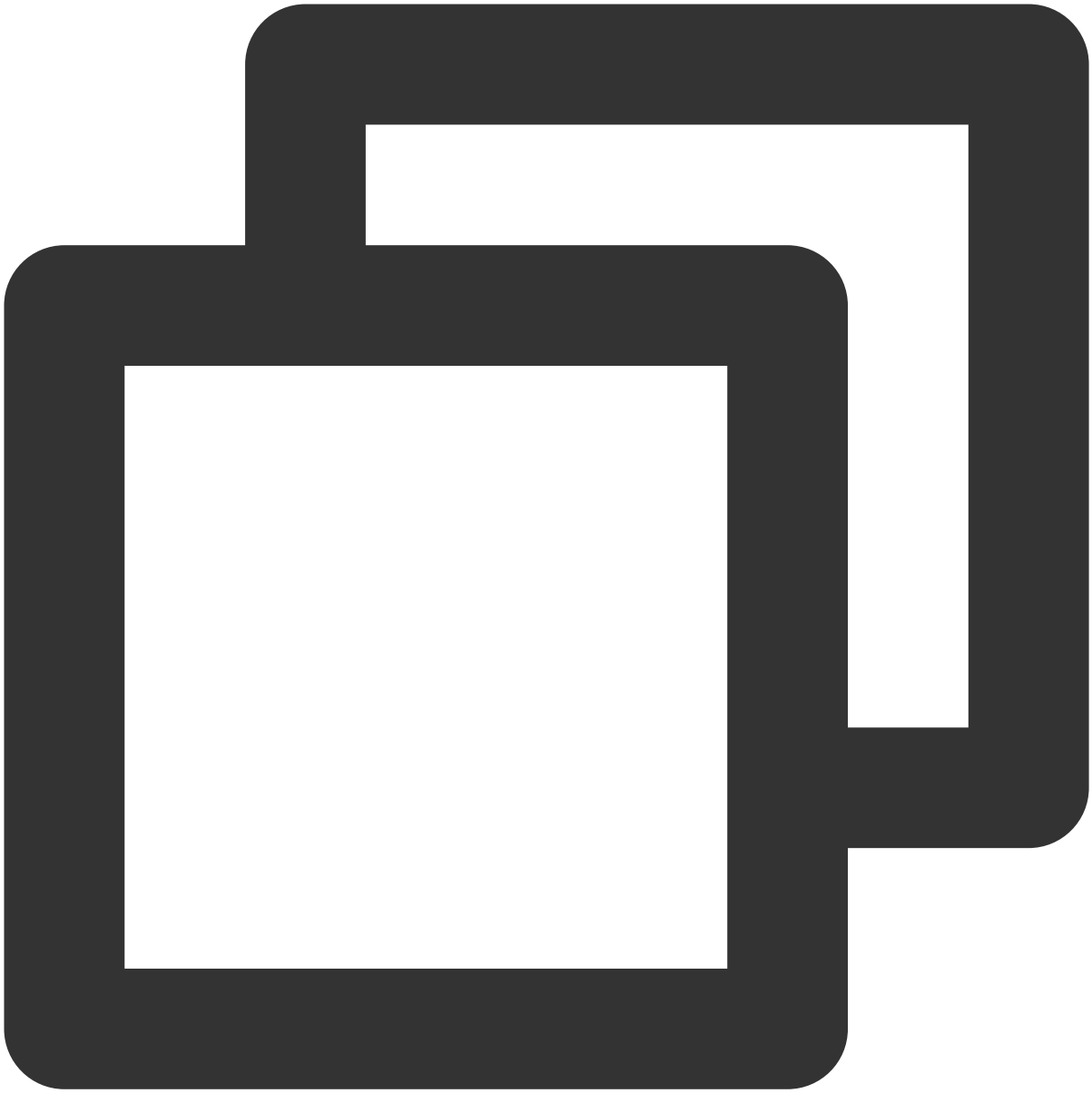
最近更新时间：2024-01-20 16:30:33

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

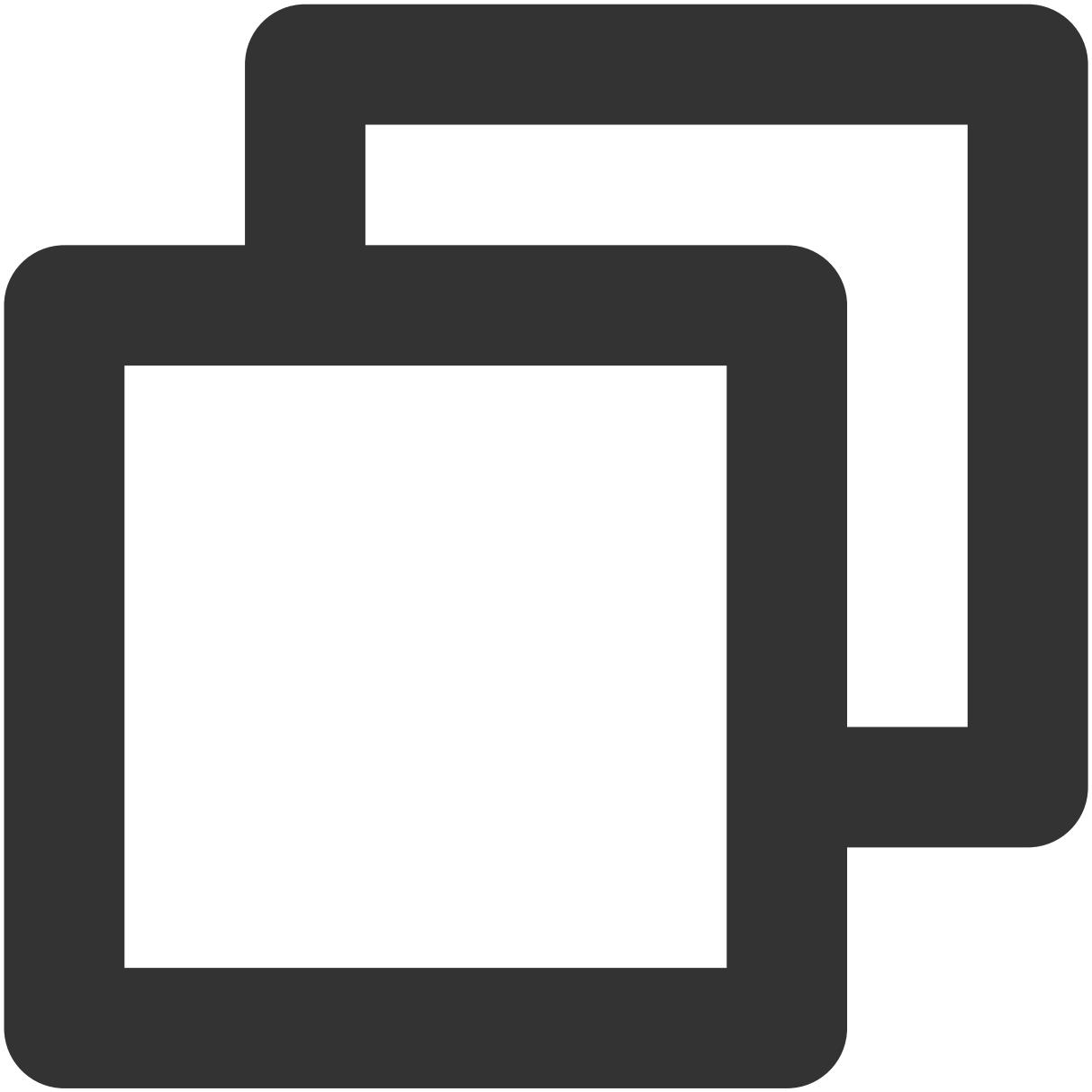
该请求无响应体。

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。

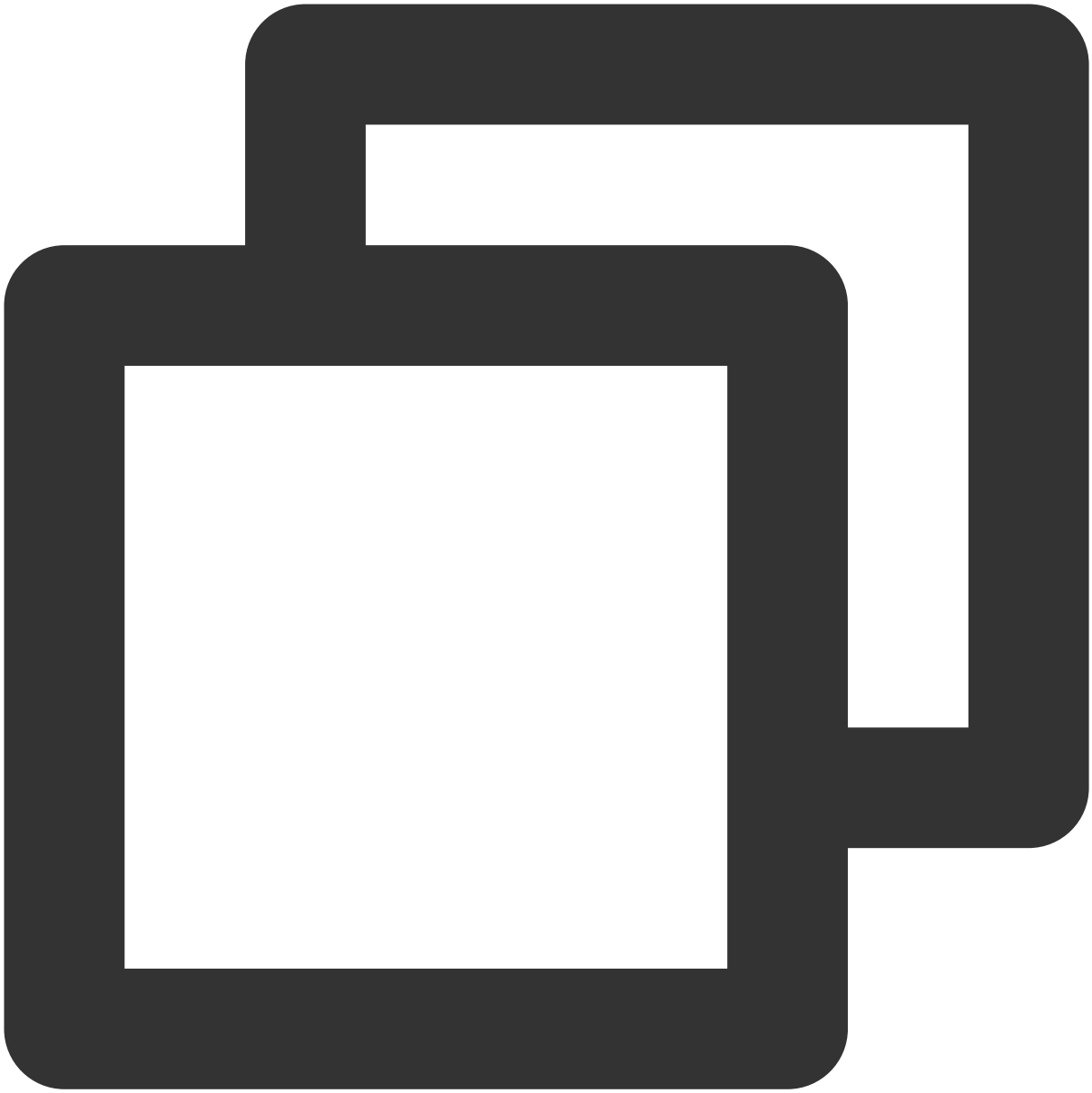
实际案例

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

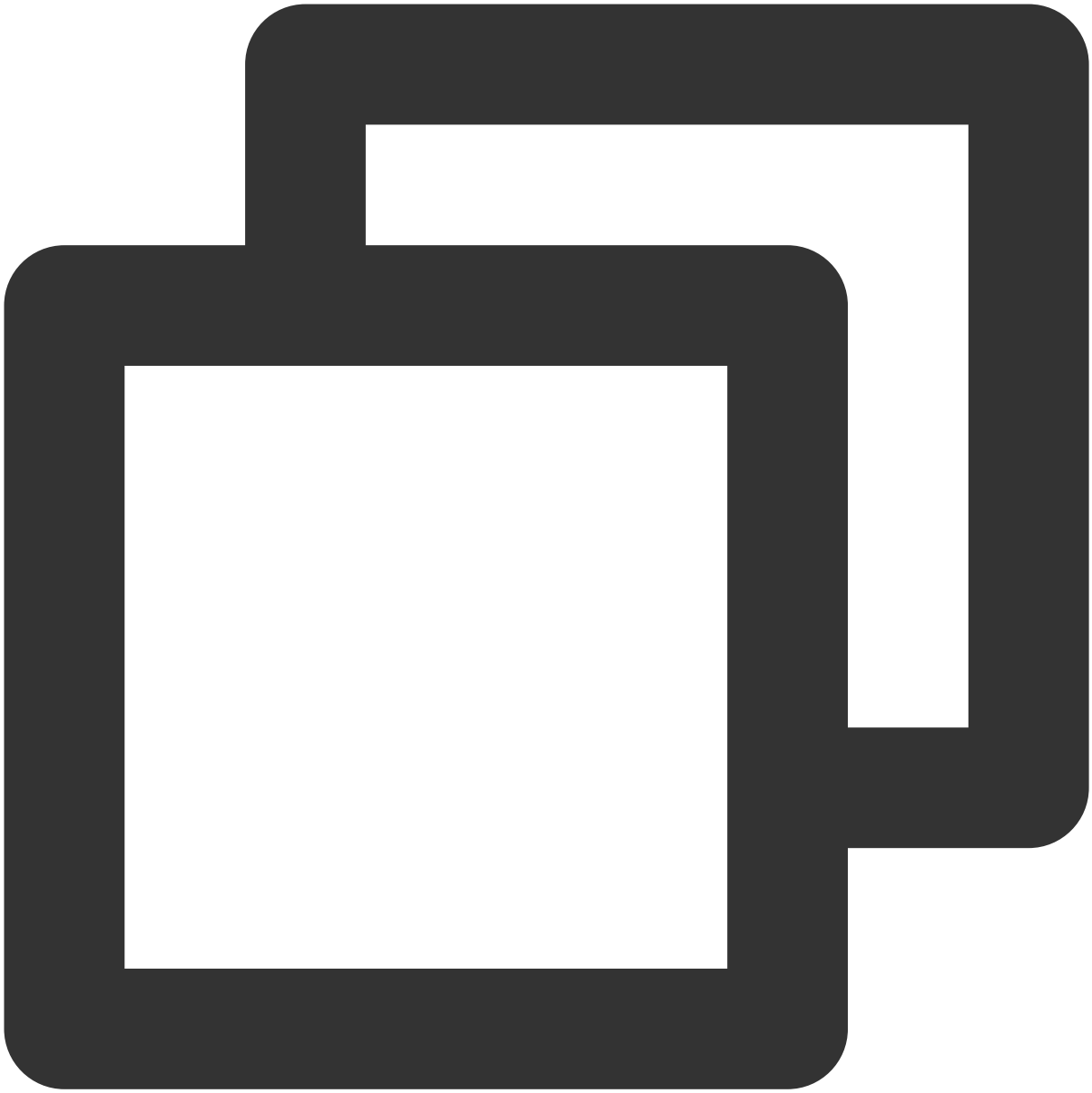
最近更新时间：2024-01-20 16:30:33

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。

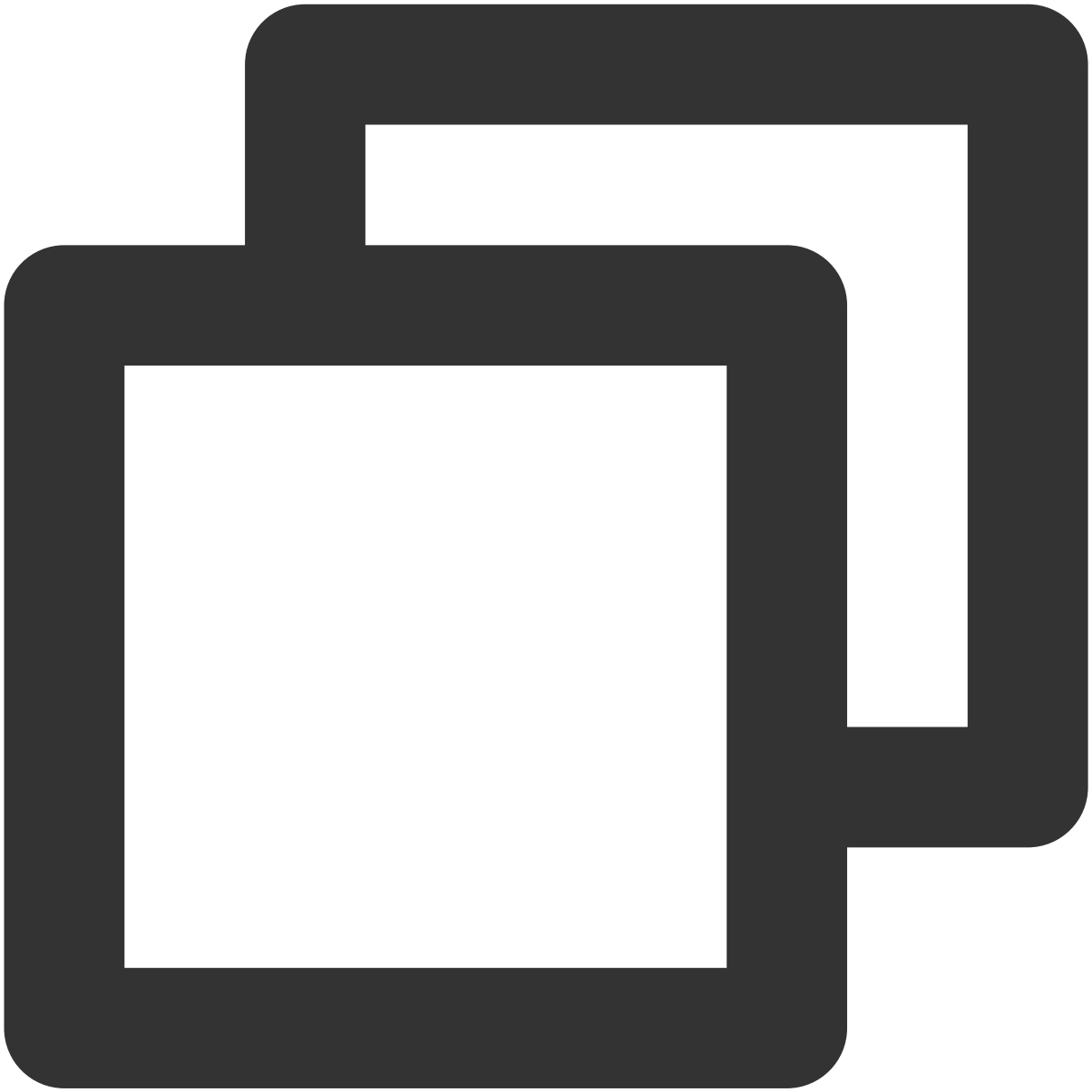
响应

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 数组
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：

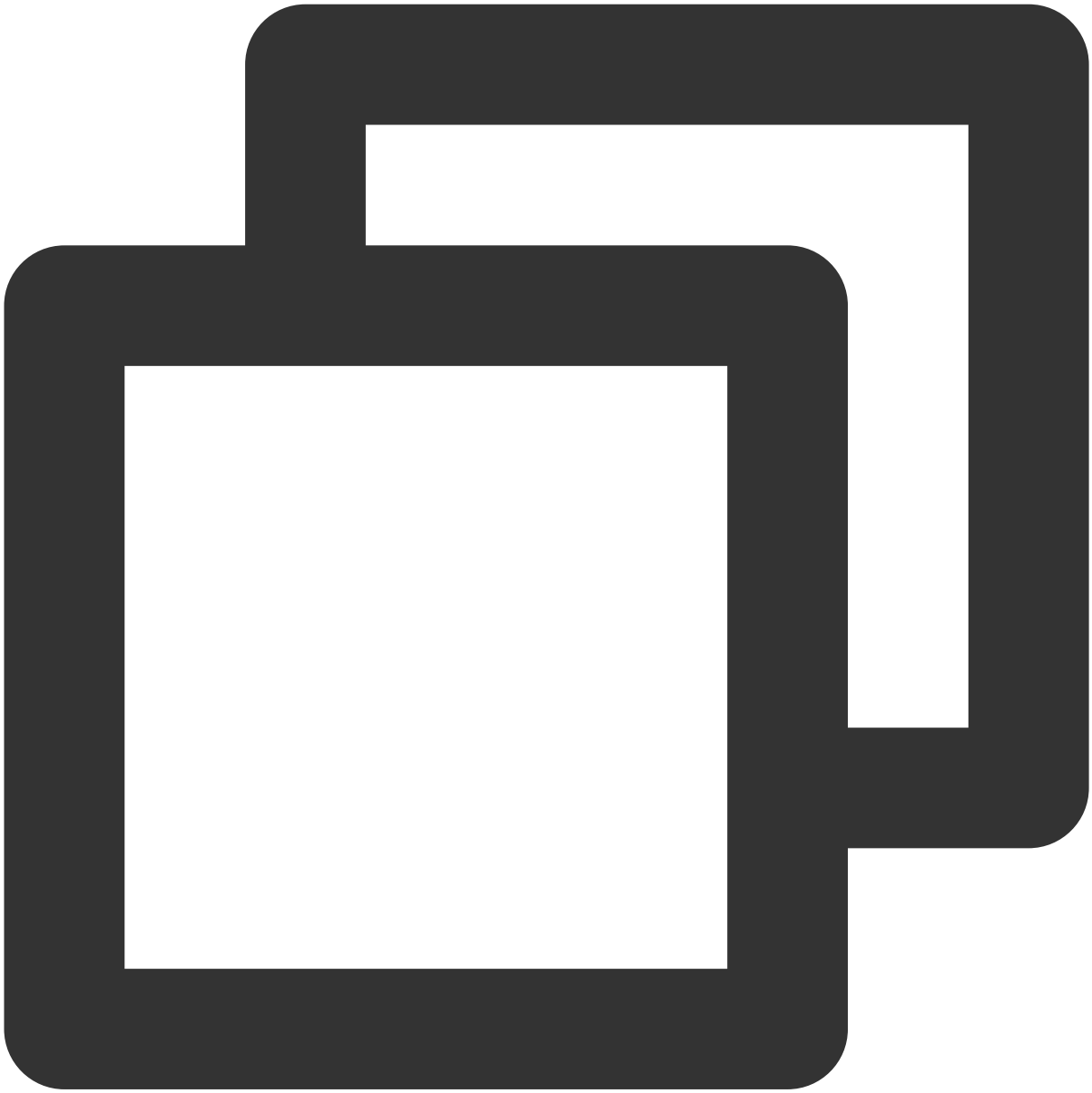
音视频转码
极速高清转码
视频转动图
视频截帧
智能封面
音视频拼接
人声分离
精彩集锦
SDR to HDR
视频增强
超分辨率
音视频转封装
数字水印
提取数字水印
视频标签
获取媒体信息
音视频流分离
视频质量分析
语音合成
音频降噪

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。

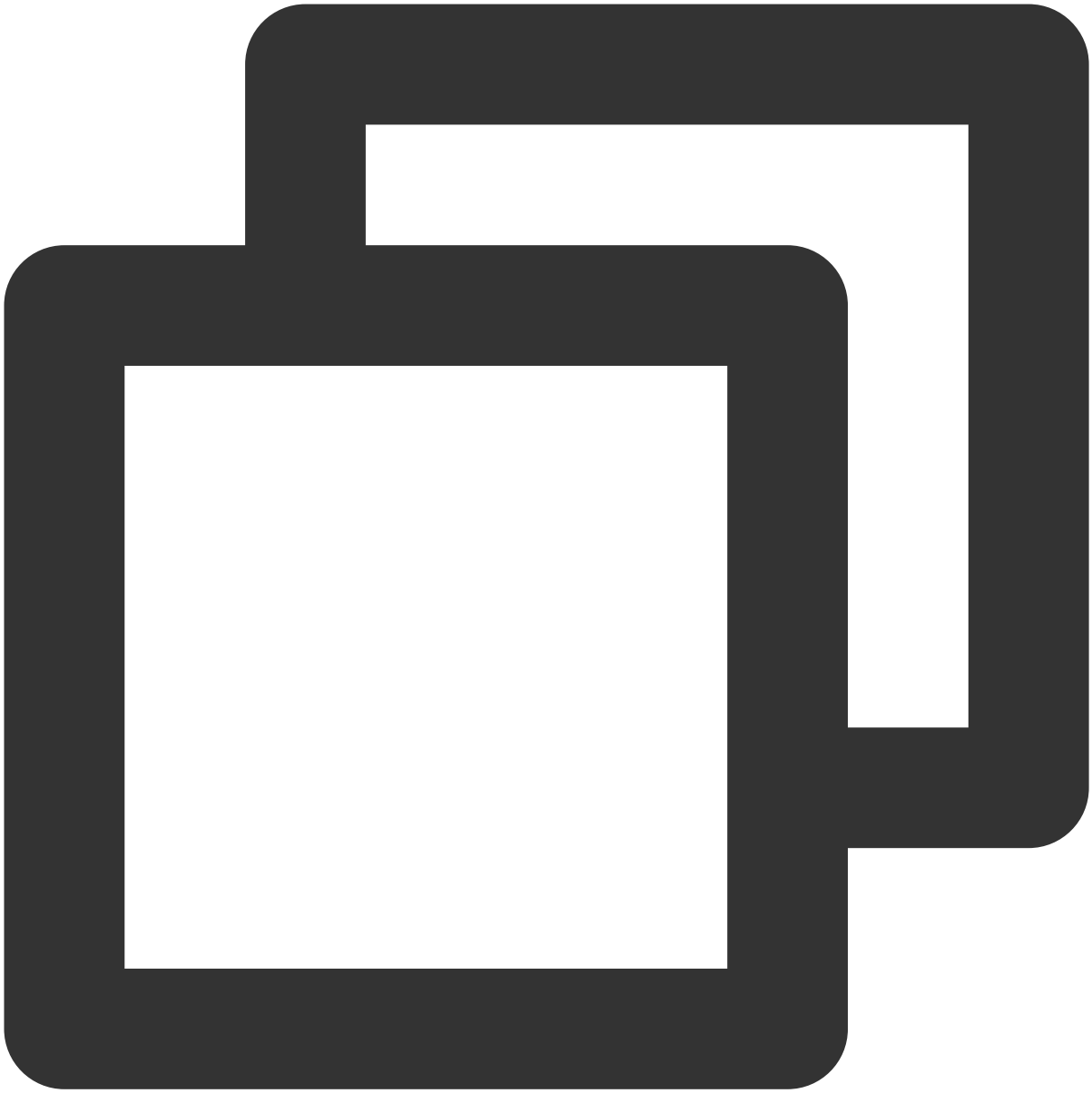
实际案例

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

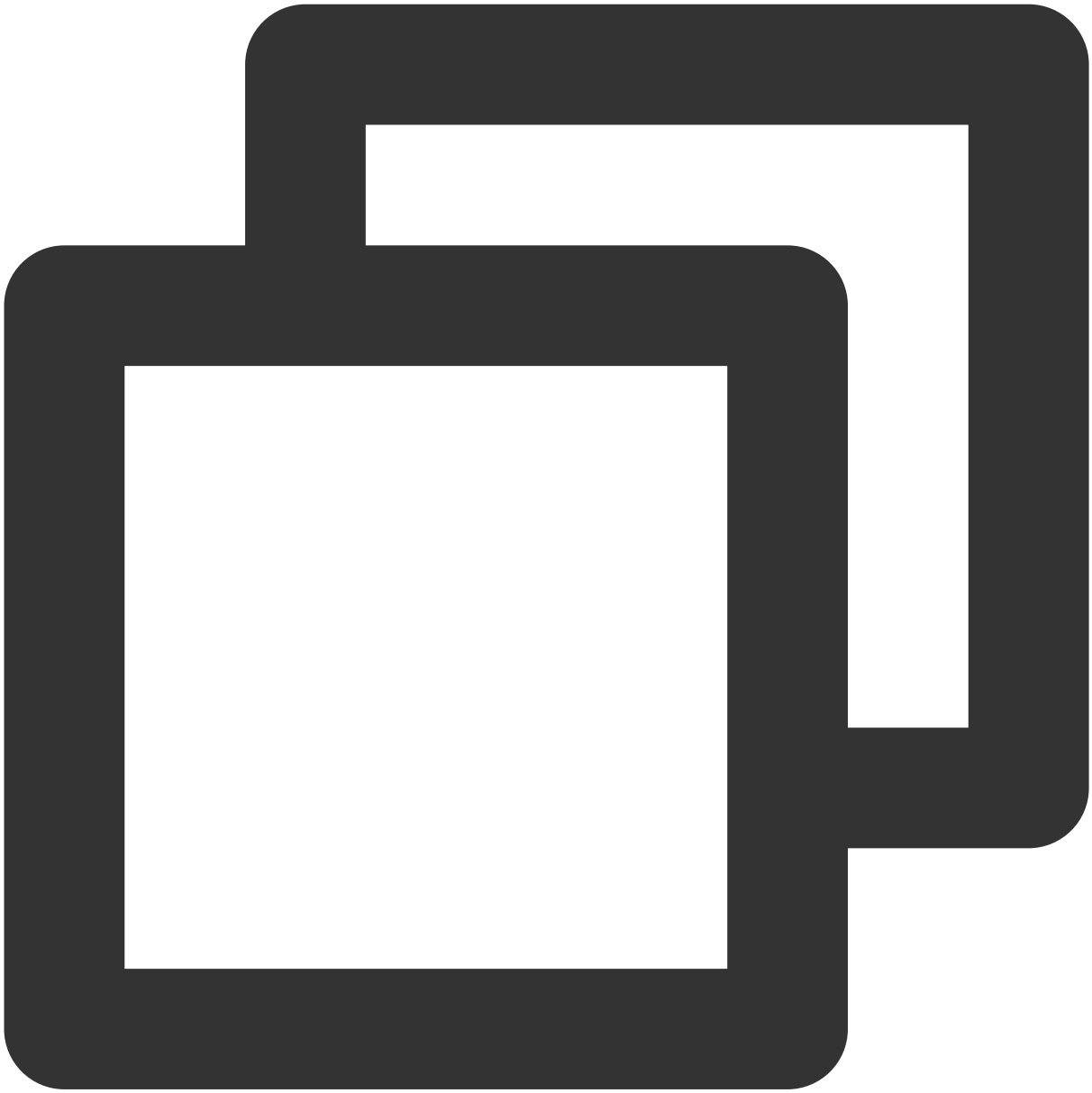


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

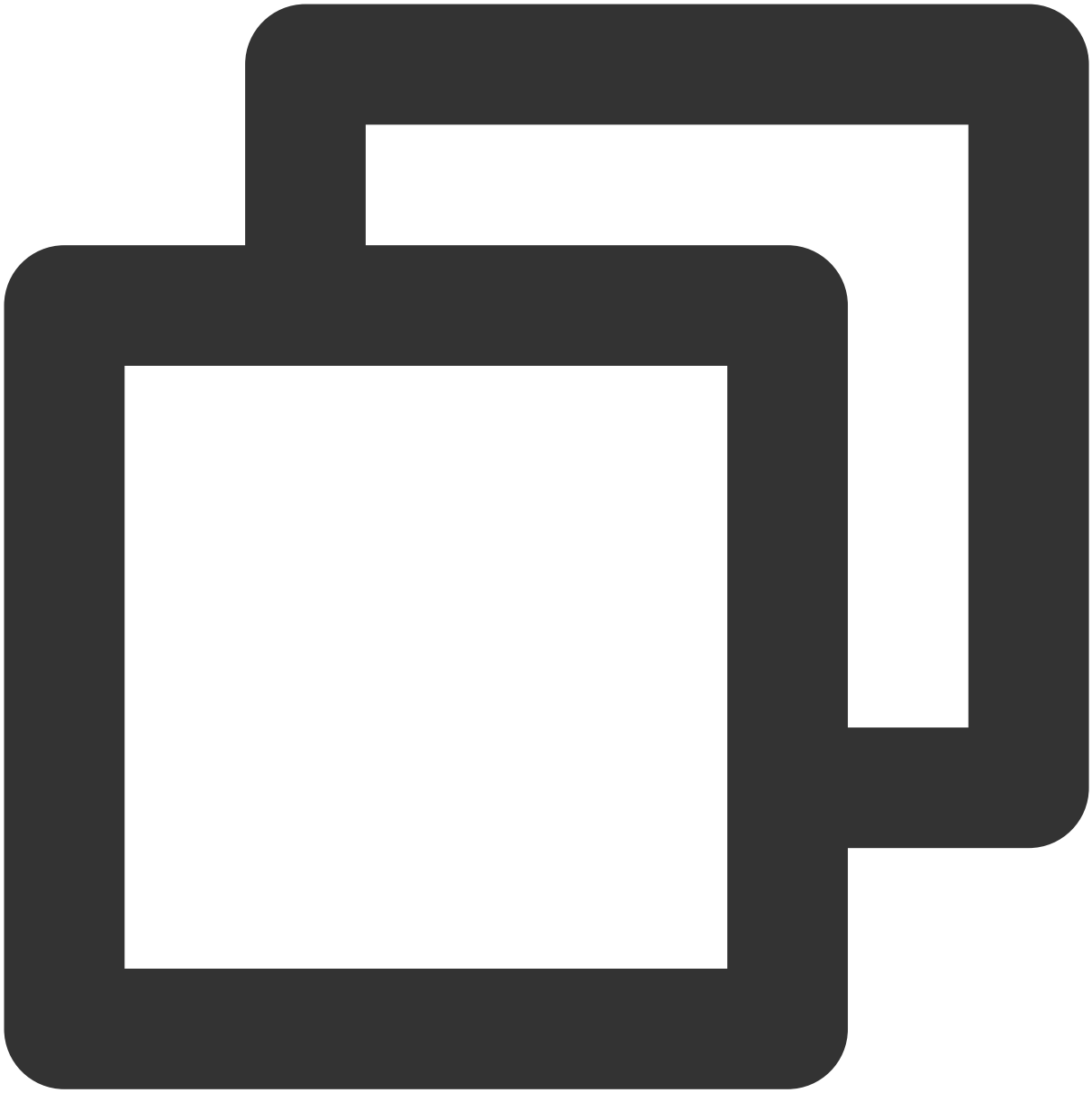
```
        </Message>
        <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
        <State>Running</State>
        <Progress>30</Progress>
        <CreationTime>2022-06-27T15:23:12+0800</CreationTime>
        <StartTime>2022-06-27T15:23:13+0800</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</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>
```

```
                        <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

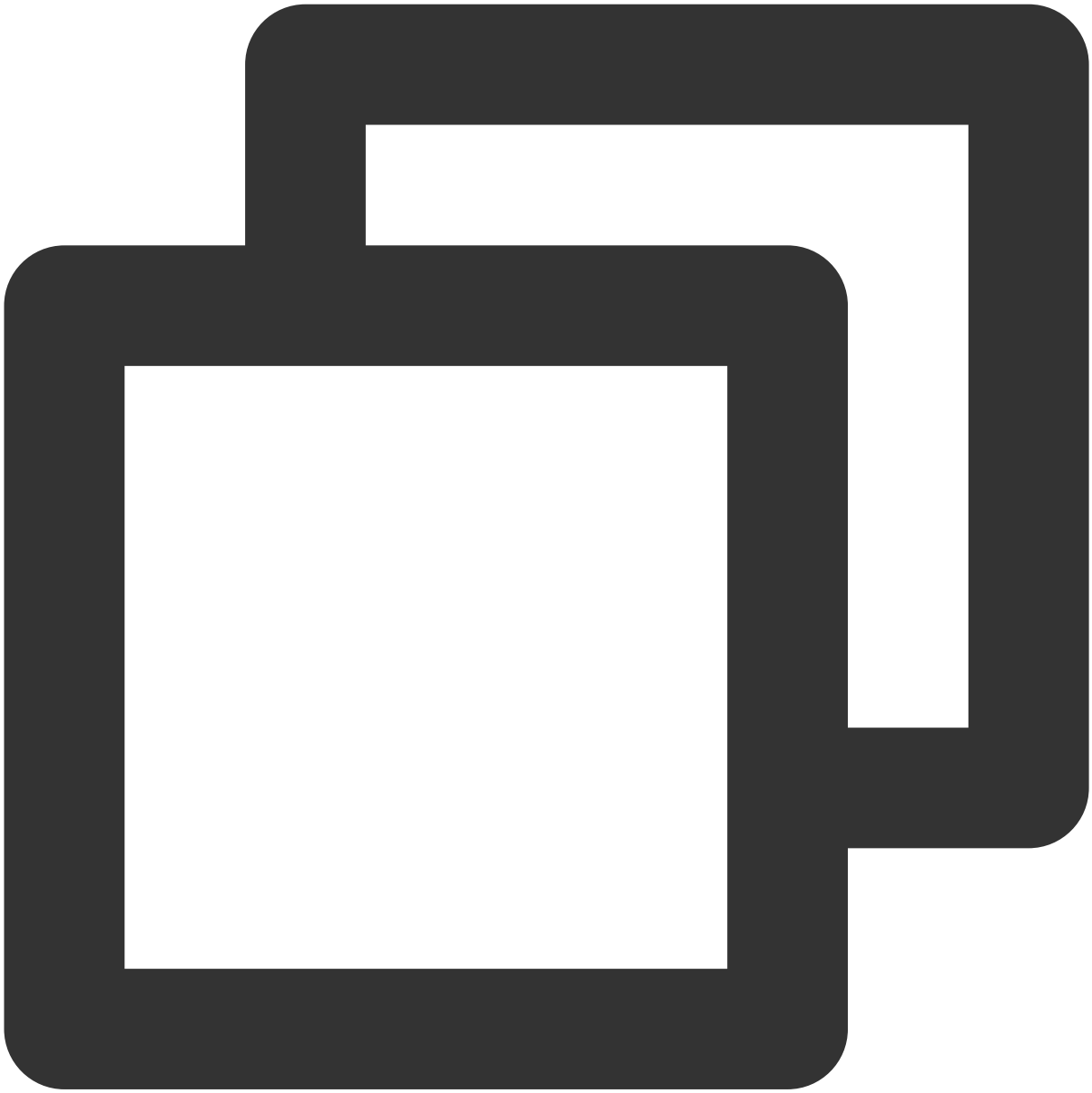
最近更新时间：2024-03-01 17:15:28

功能描述

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。

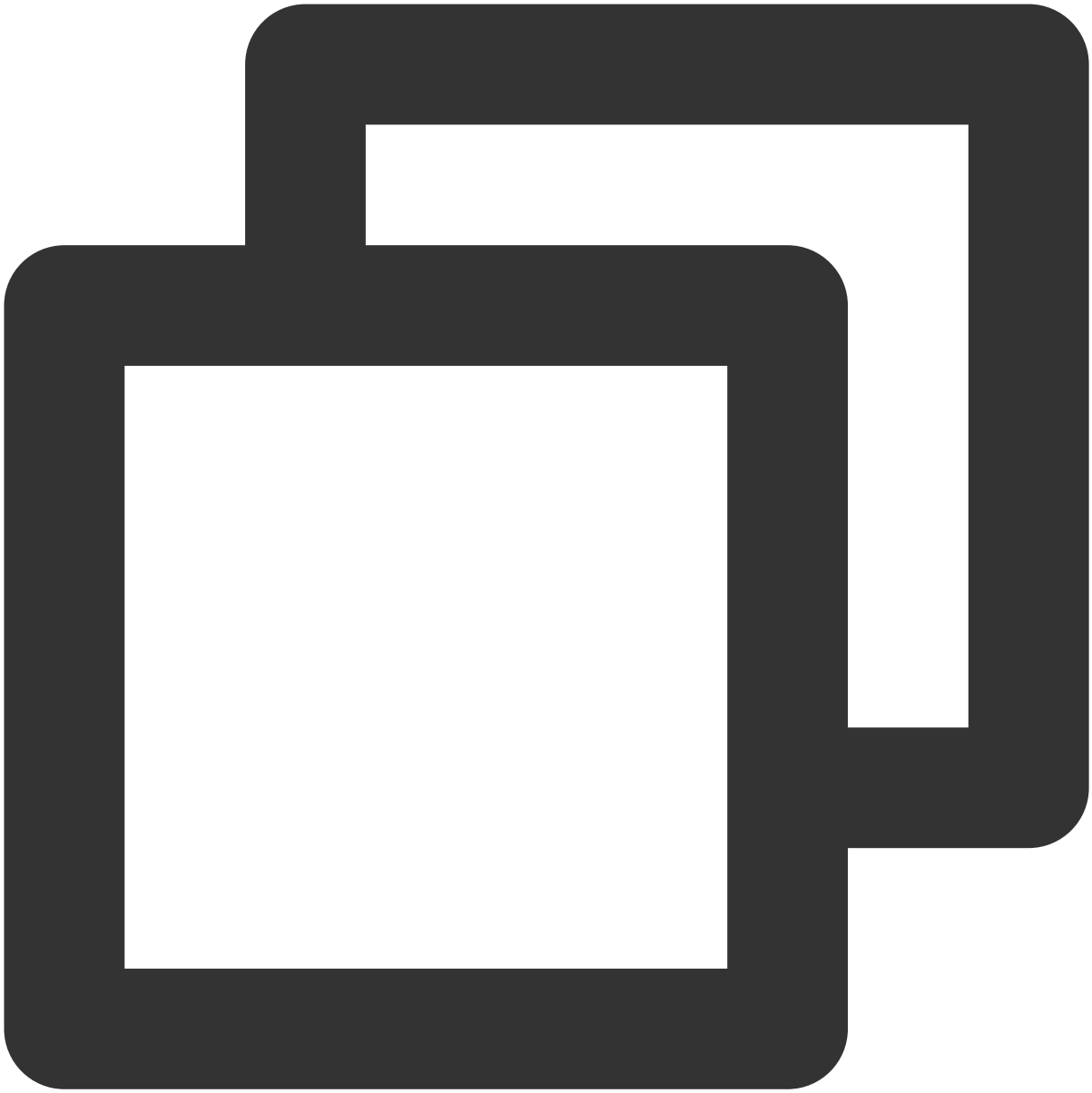
响应

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：

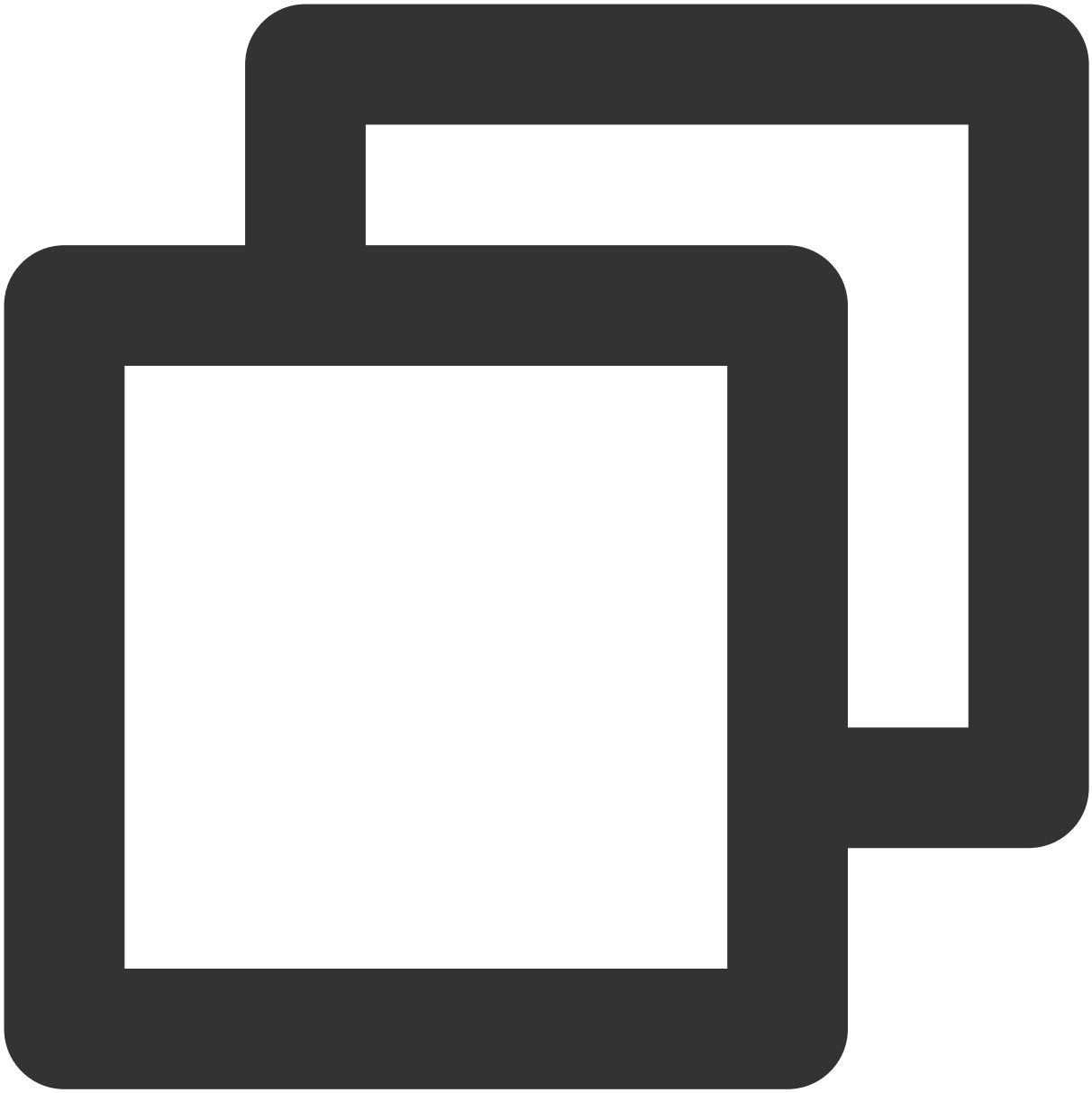
音视频转码
极速高清转码
视频转动图
视频截帧
智能封面
音视频拼接
人声分离
精彩集锦
SDR to HDR
视频增强
超分辨率
音视频转封装
数字水印
提取数字水印
视频标签
获取媒体信息
音视频流分离
视频质量分析
语音合成
音频降噪

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。

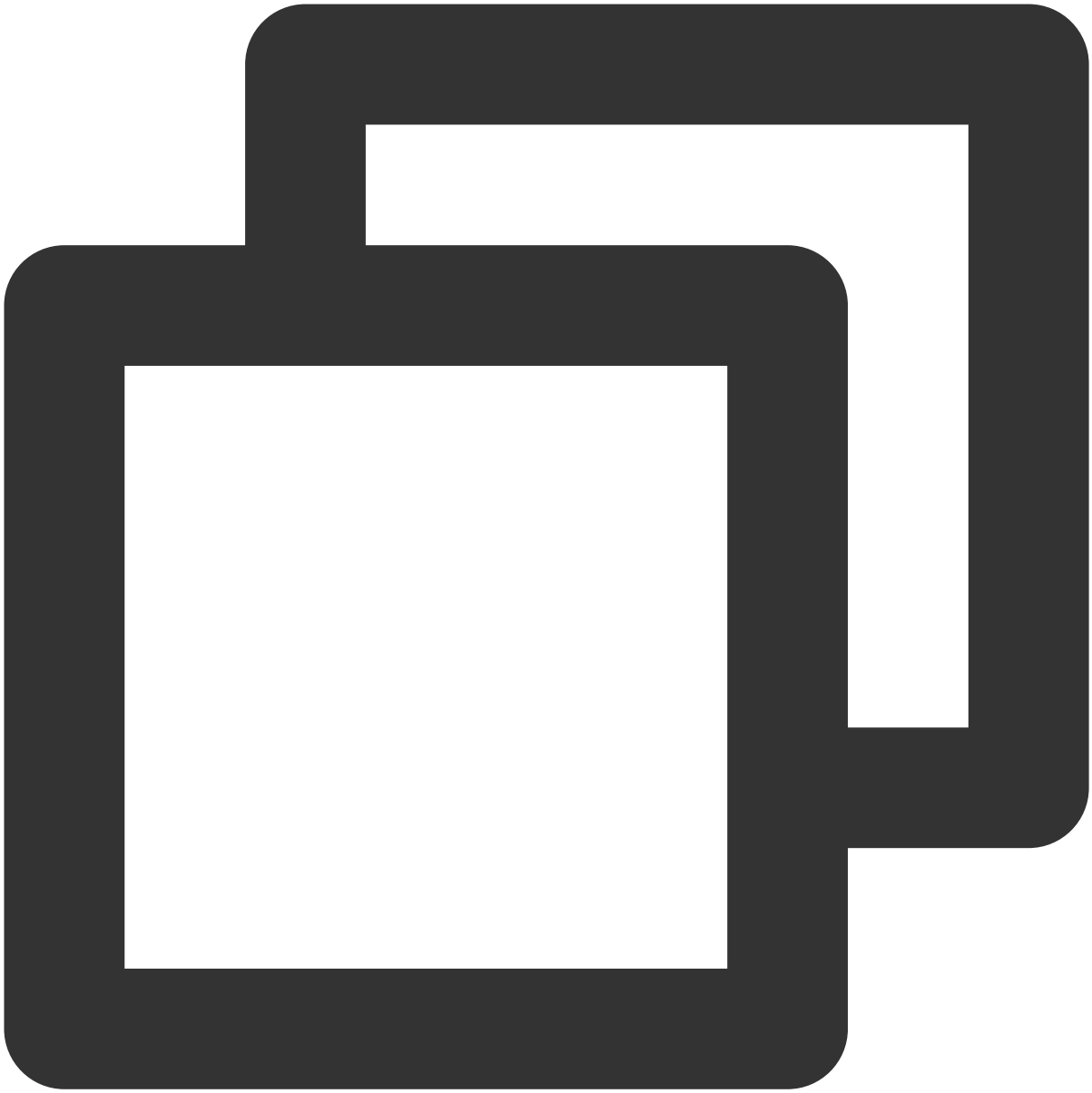
实际案例

请求



```
GET /jobs?queueId=p2242ab62c7c94486915508540933a2c6&tag=Transcode HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&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>
    <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

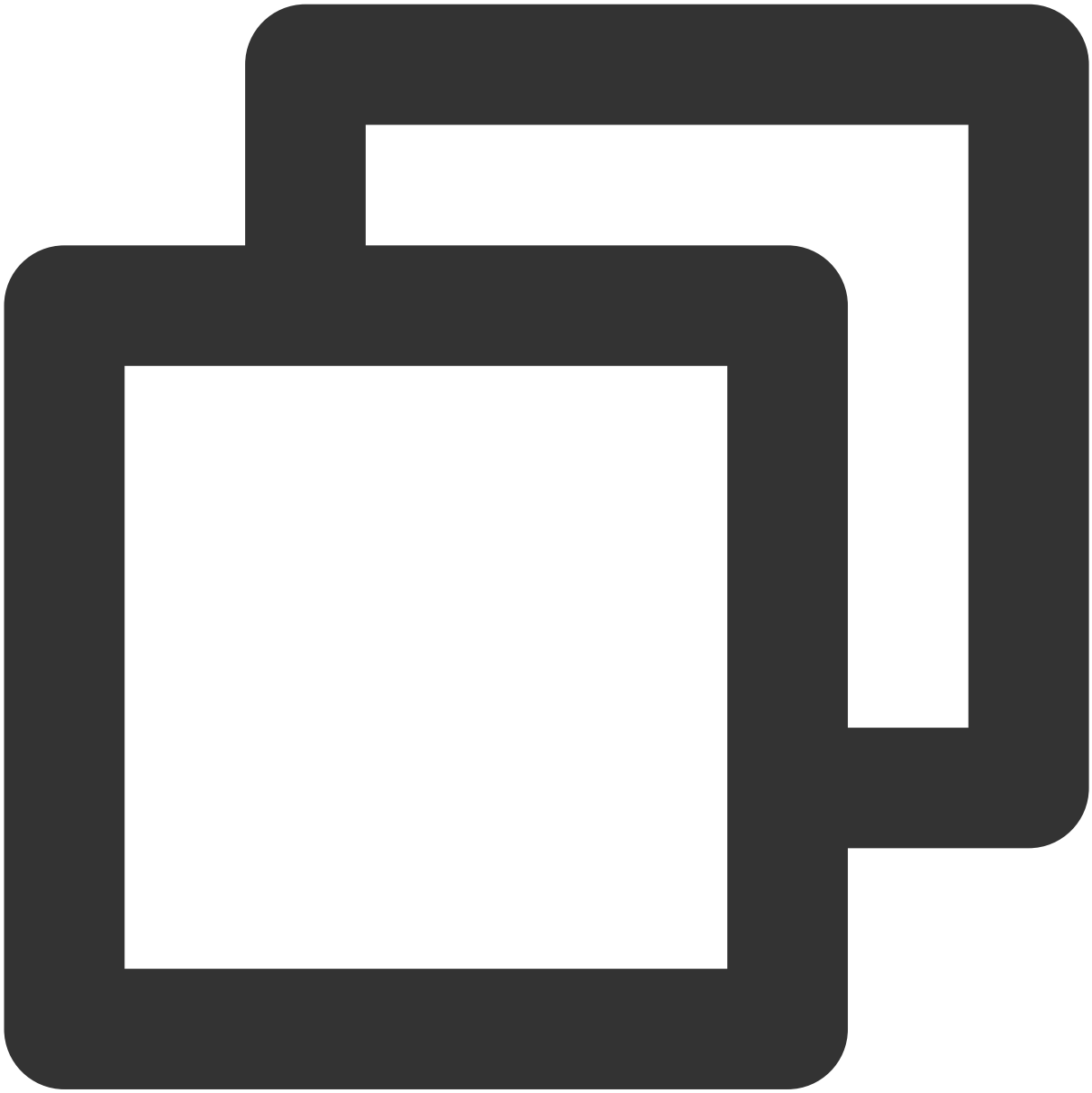
最近更新时间：2024-01-20 16:30:33

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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
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

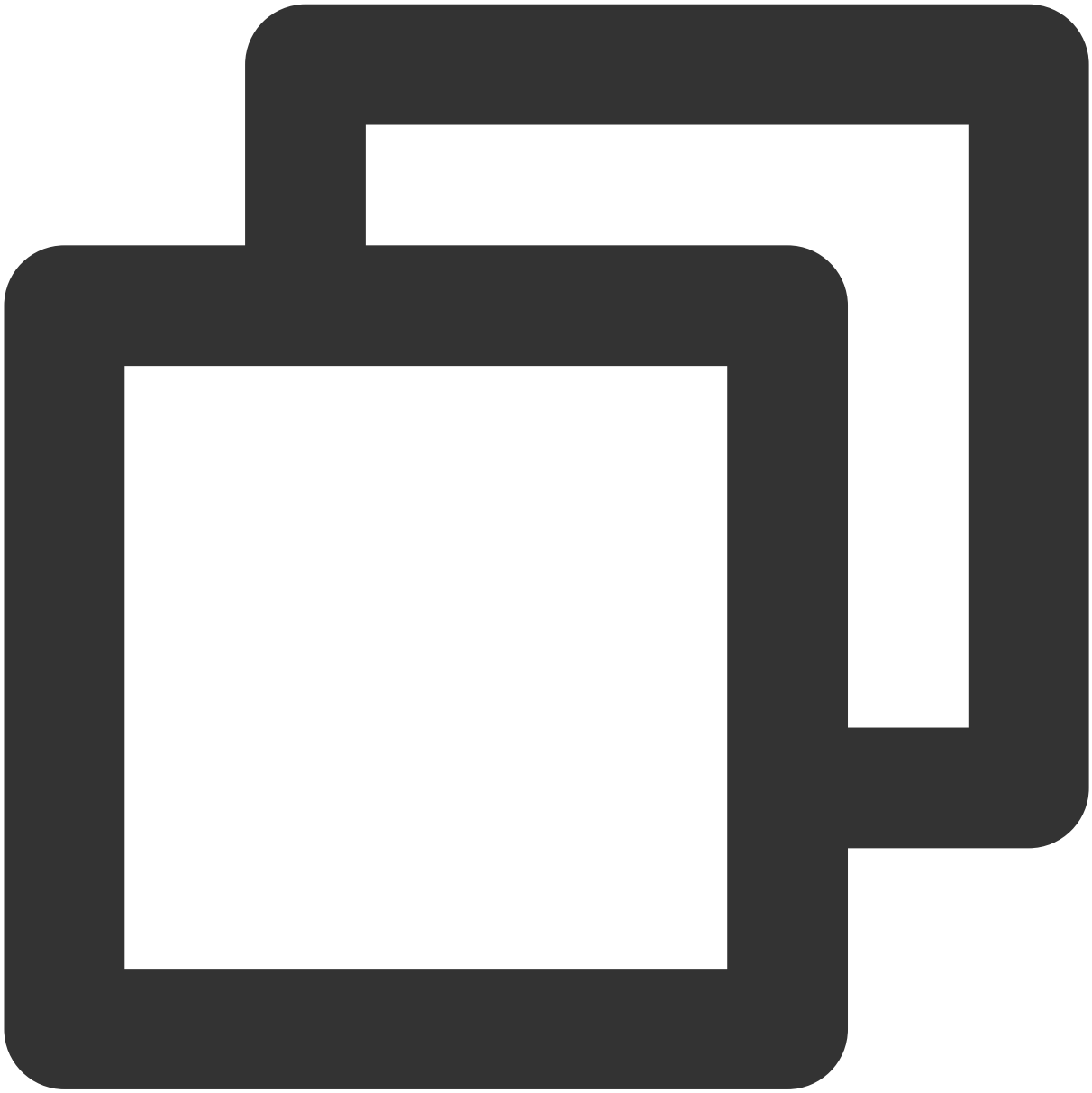
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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



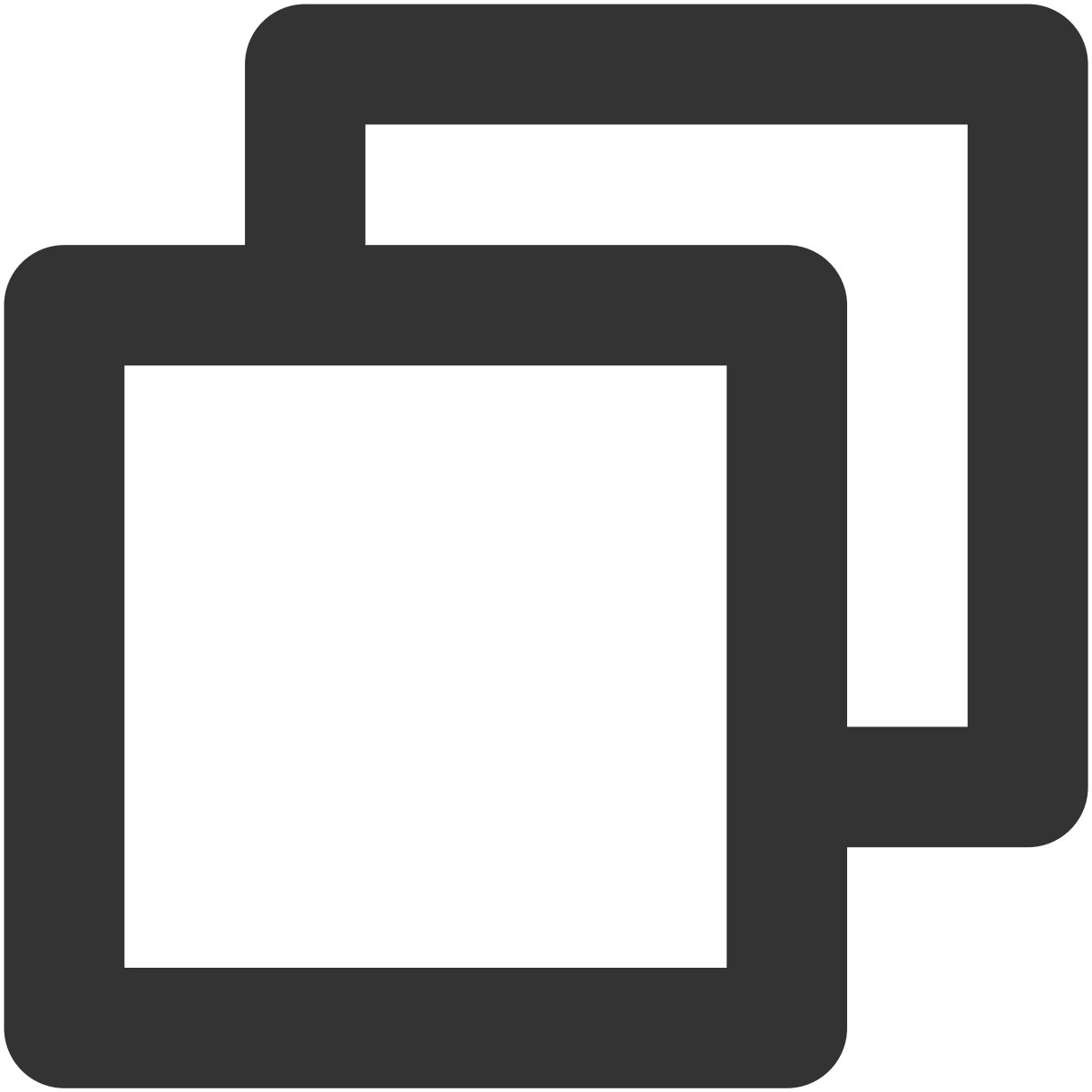
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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

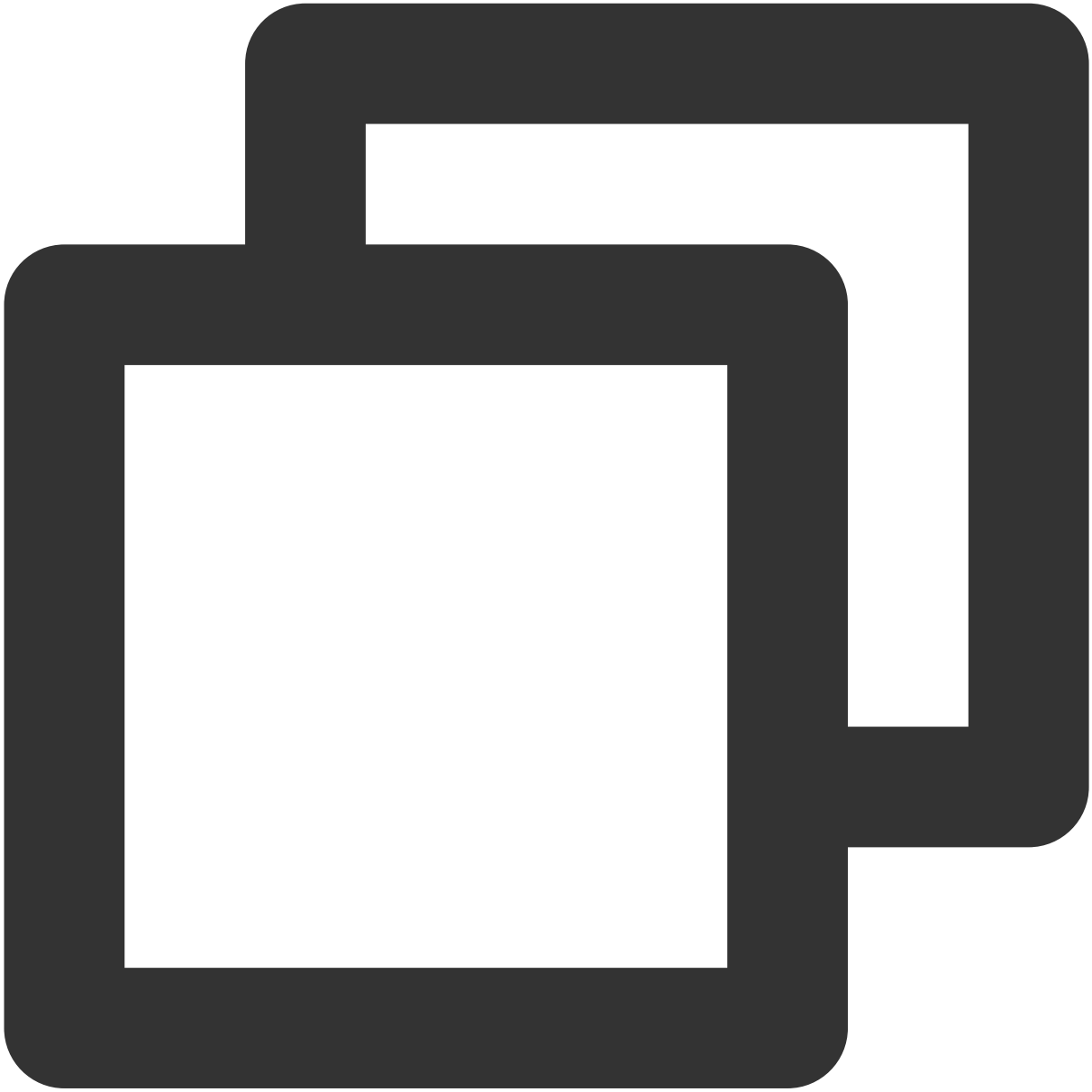


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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

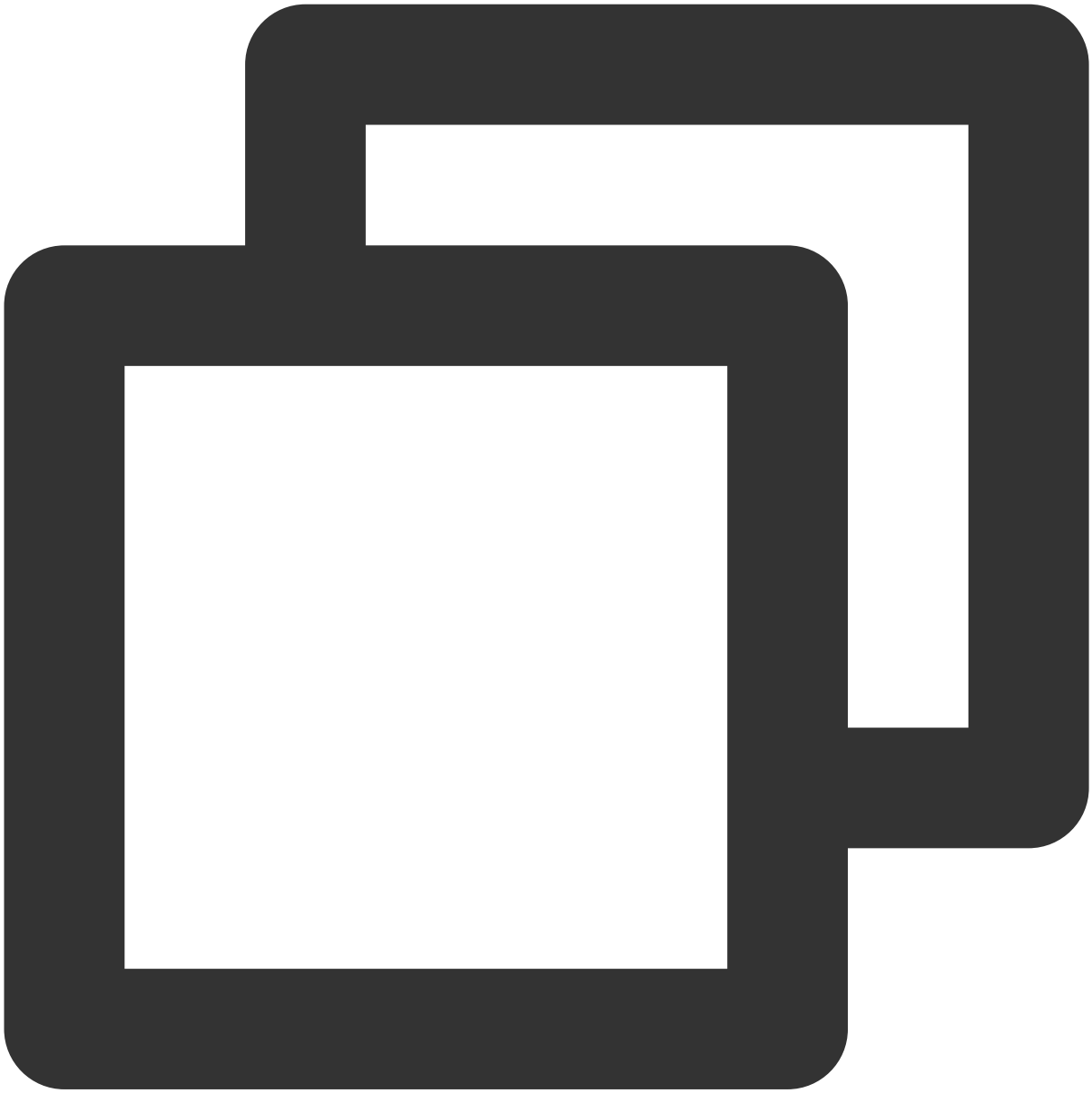
最近更新时间：2024-01-20 16:30:33

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

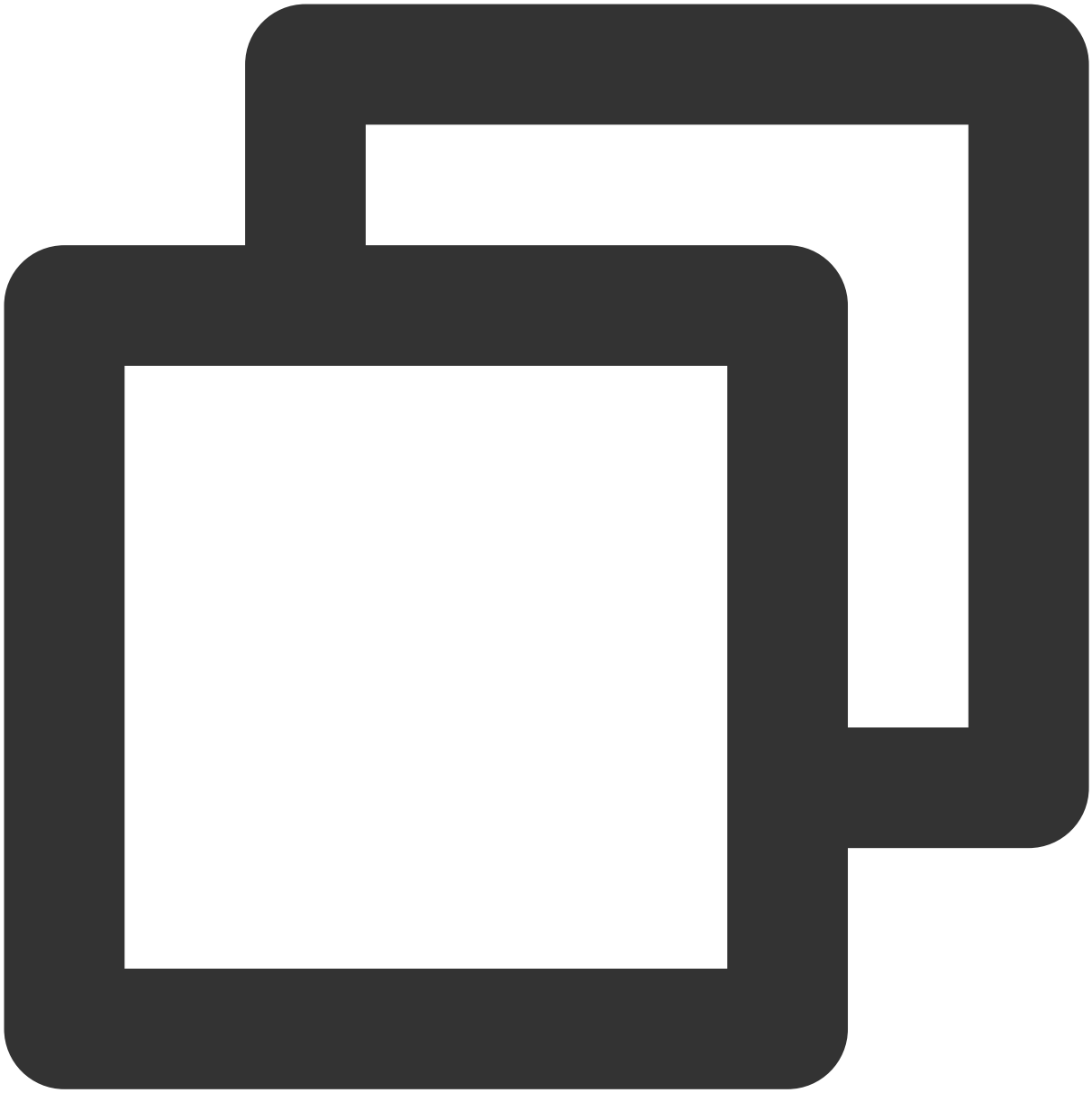
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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



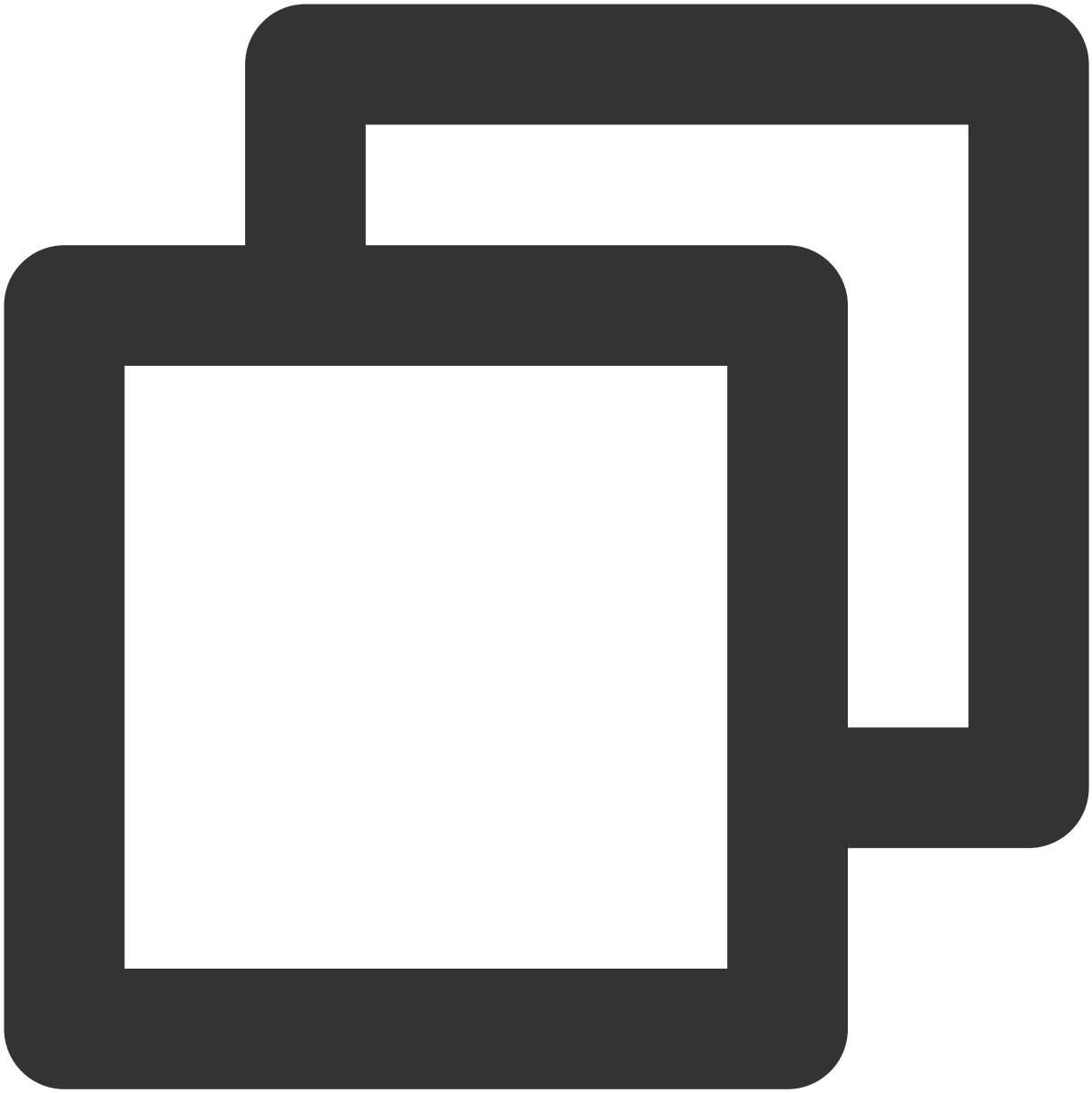
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:33:07+0800</CreationTime>
        <EndTime>2022-06-30T18:33:20+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
                        <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



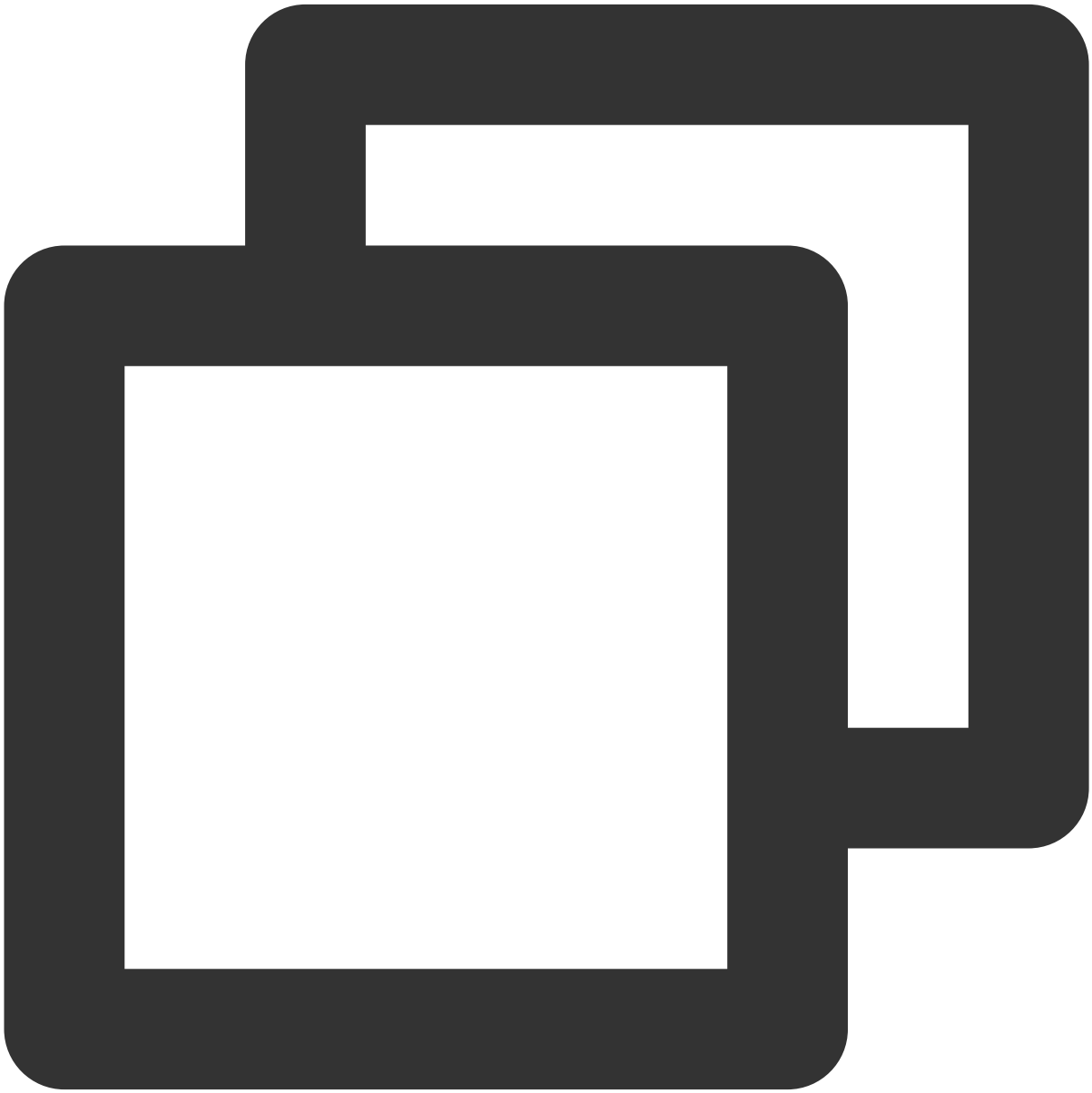
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:33:07+0800</CreationTime>
        <EndTime>2022-06-30T18:33:20+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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>
```

```
                <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

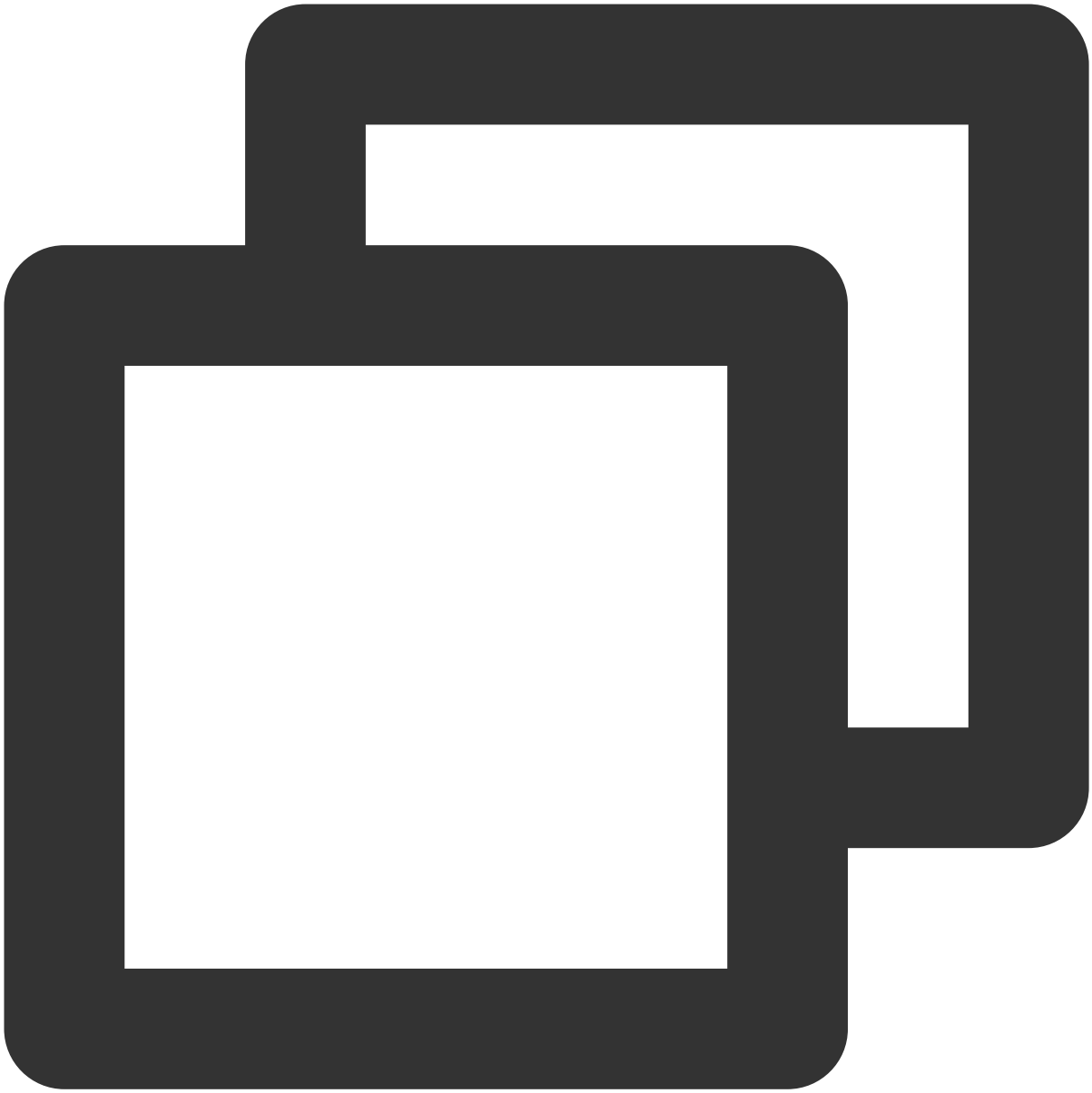
最近更新时间：2024-01-20 16:30:33

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

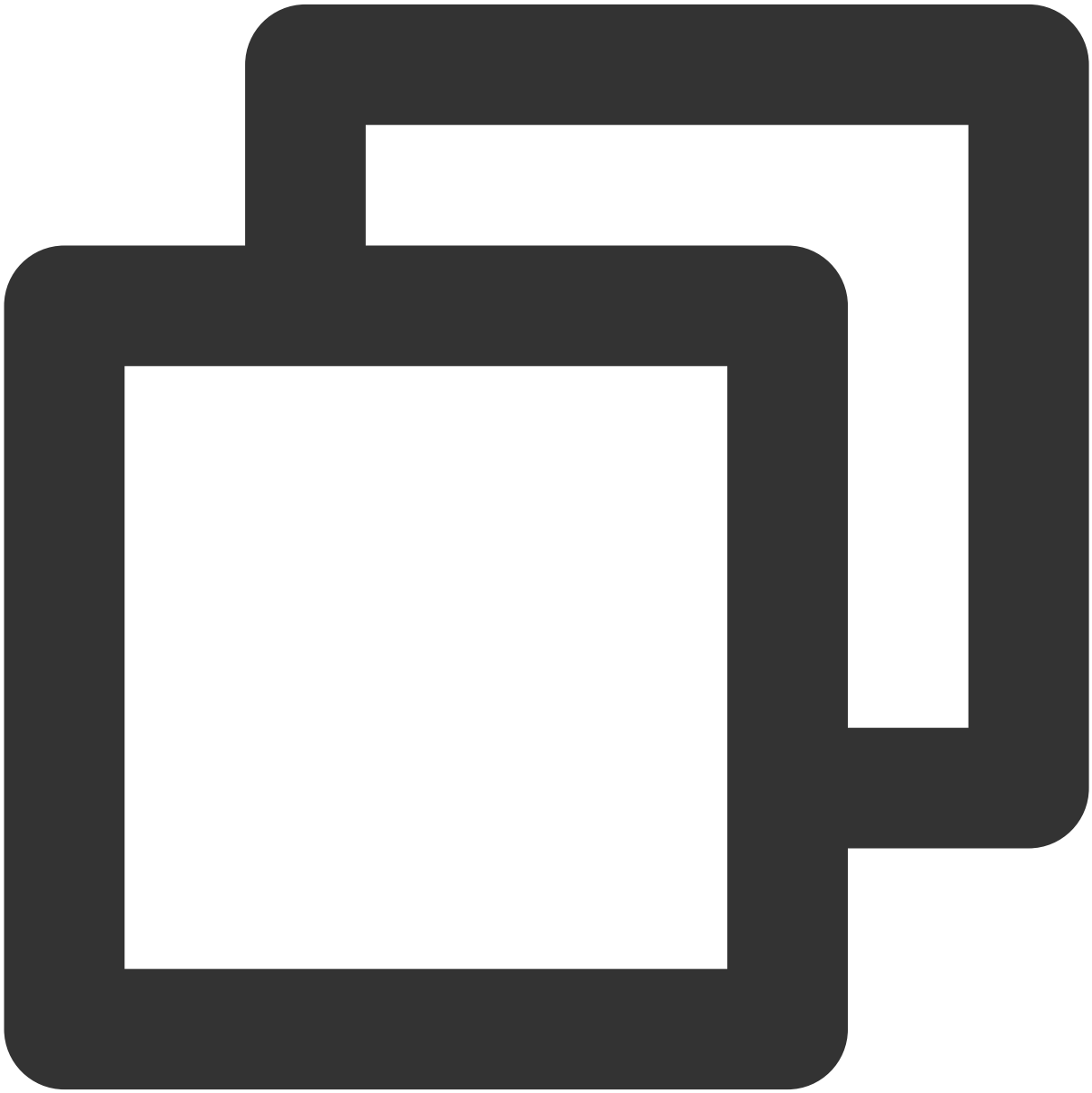
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

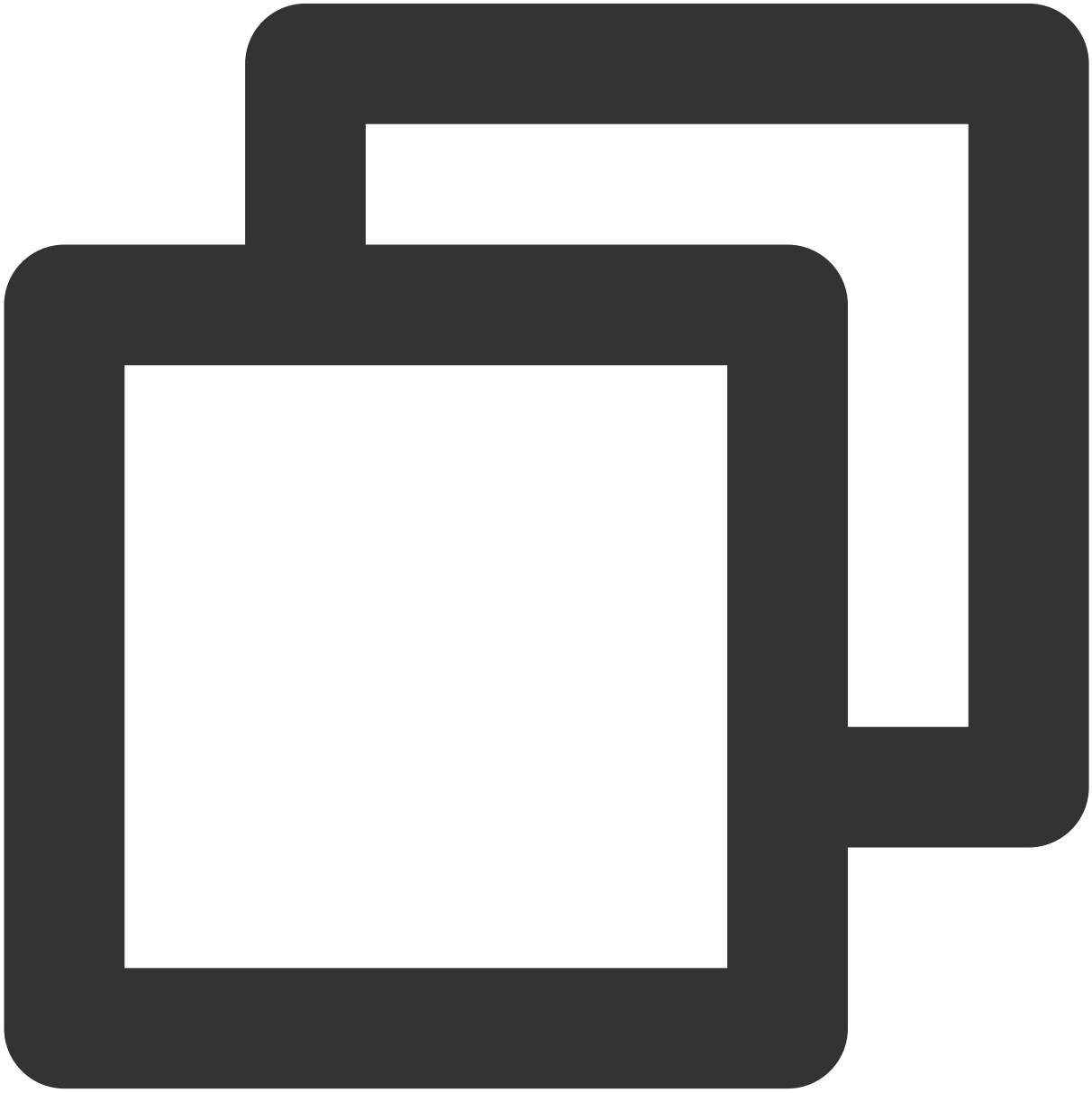


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:37:37+0800</CreationTime>
        <EndTime>2022-06-30T18:39:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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

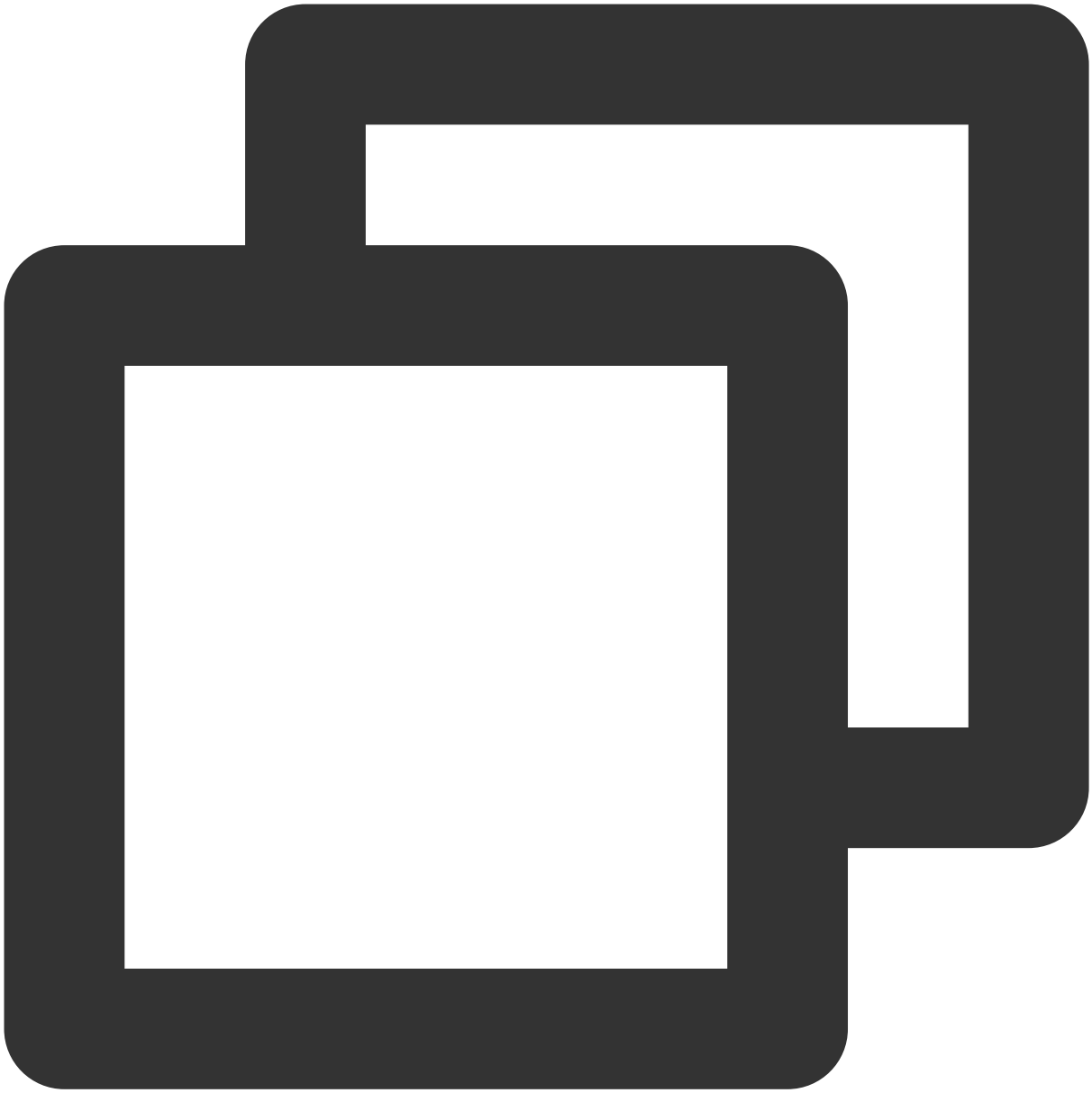
最近更新时间：2024-01-20 16:30:33

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。

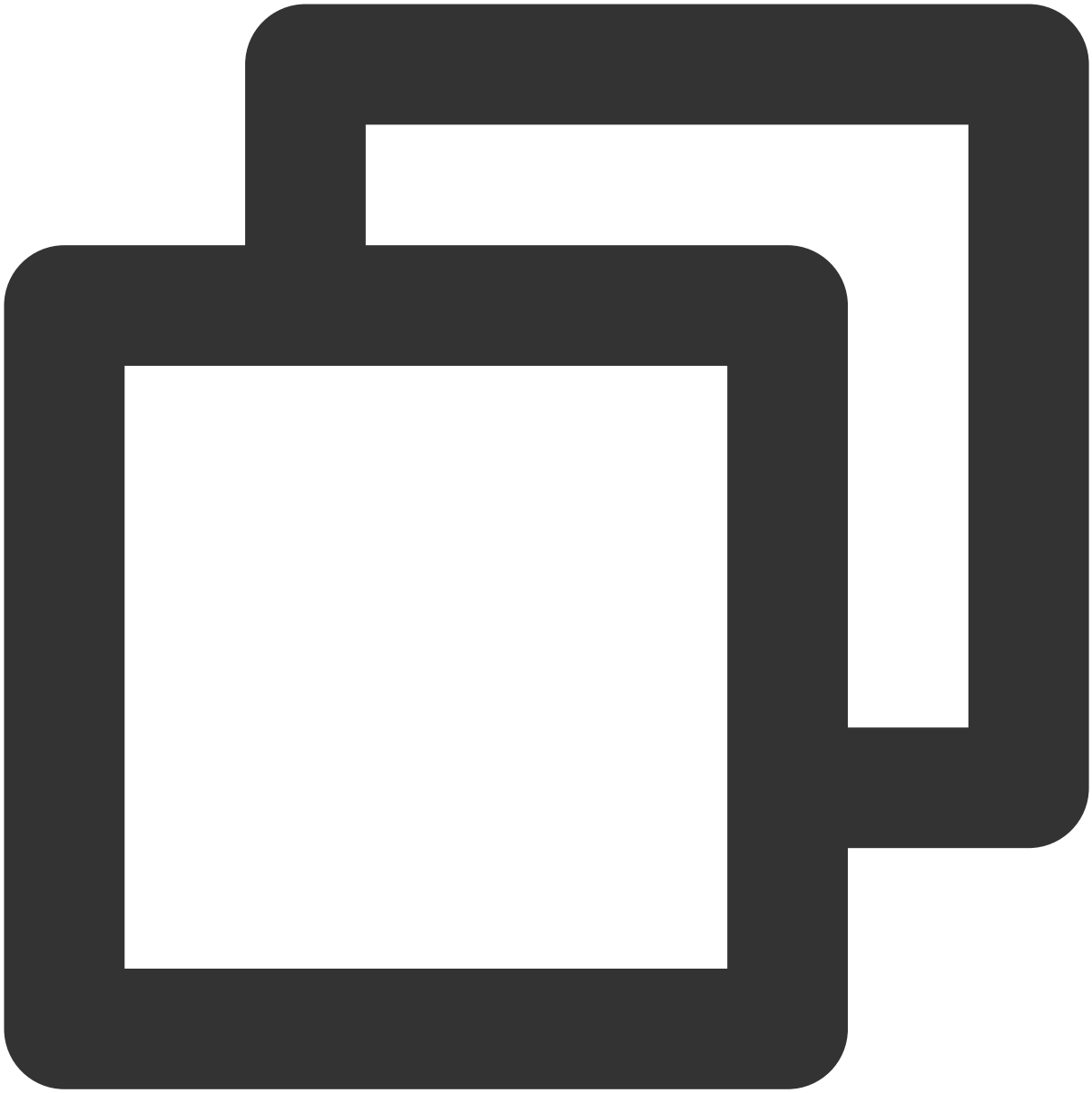
Container 节点 Workflow 的内容：

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

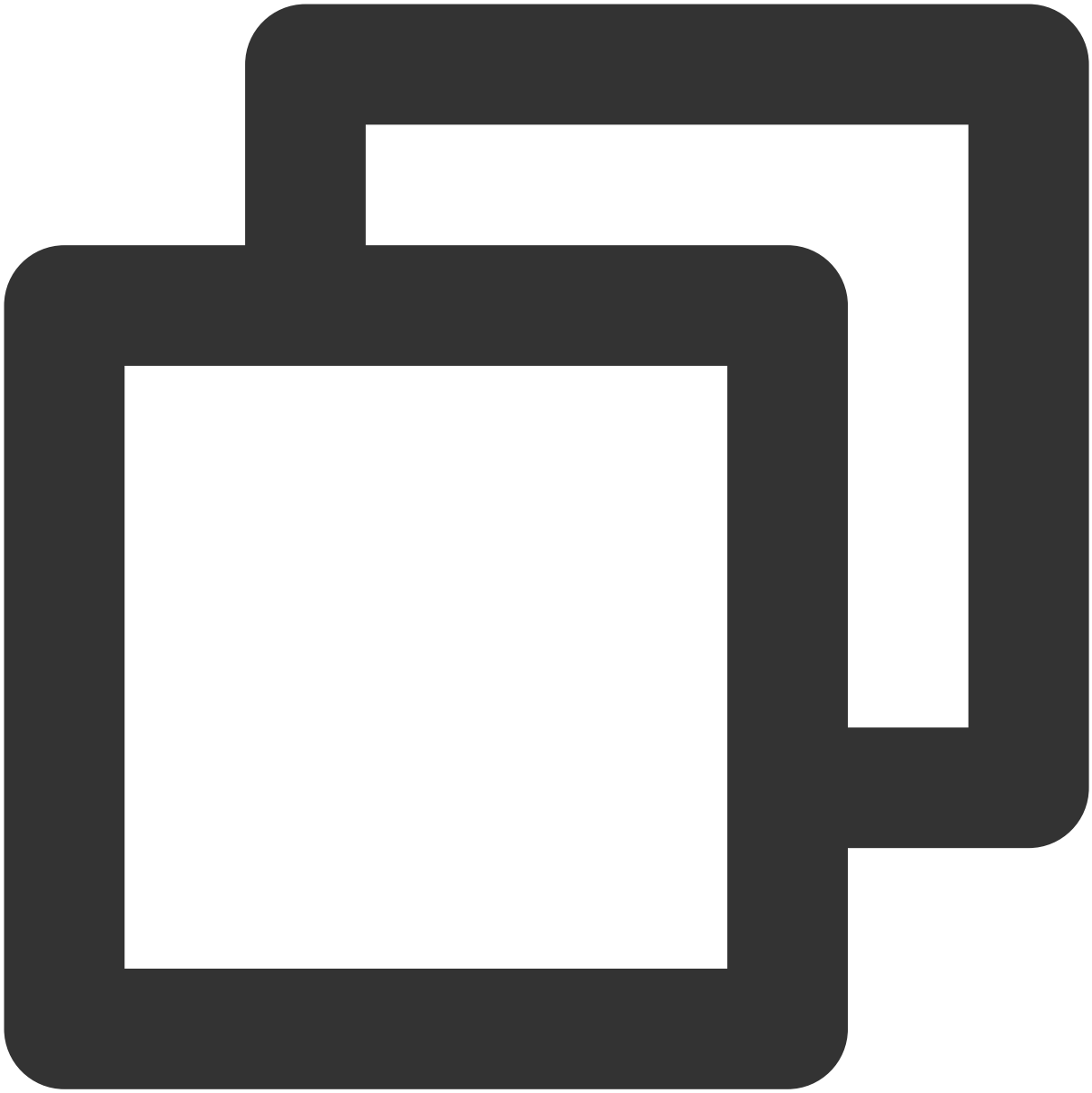
案例 1：通过任务接口触发的任务回调



```
<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

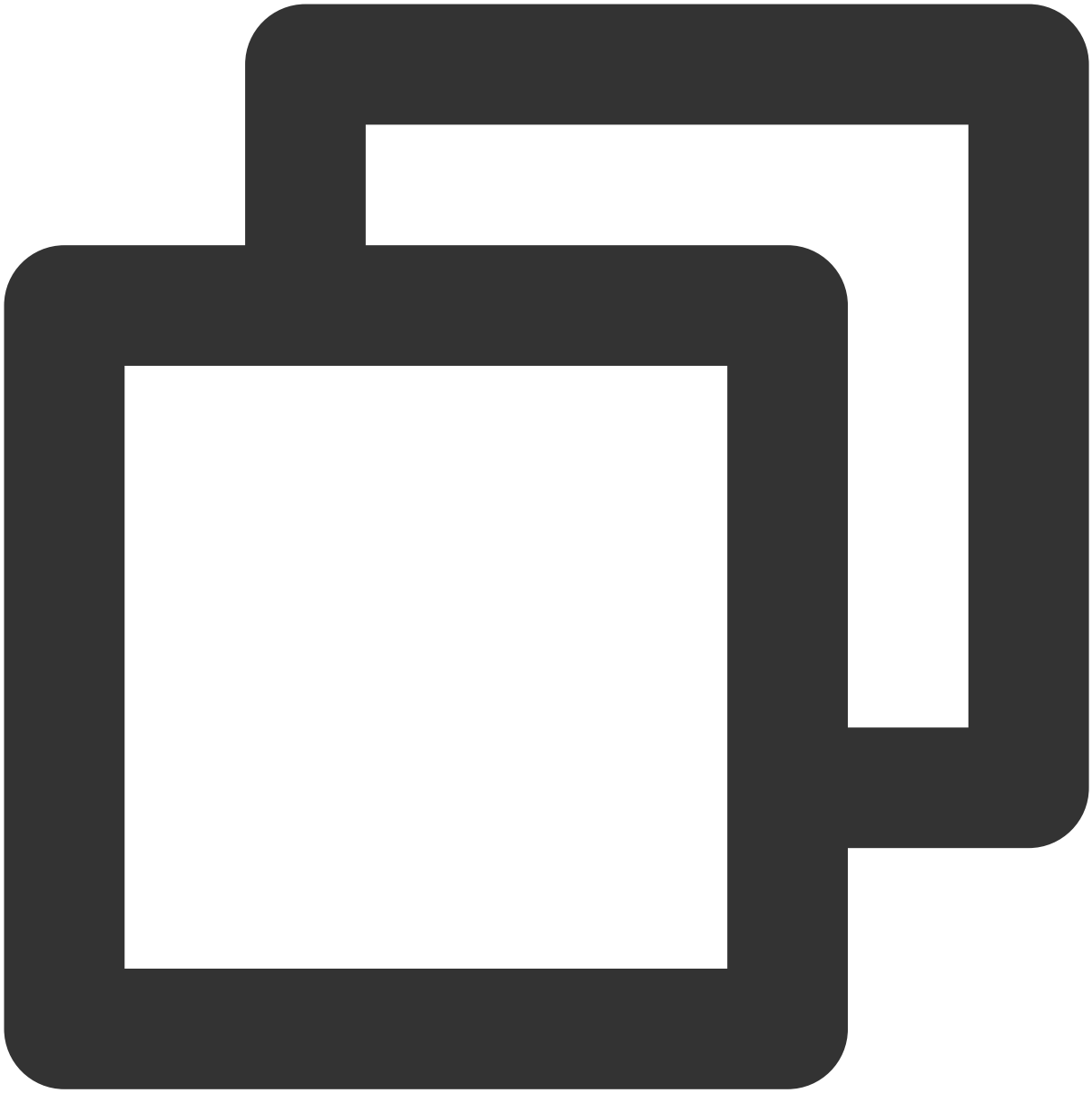
最近更新时间：2024-01-20 16:30:33

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>2022-06-30T18:45:14+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>
```

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

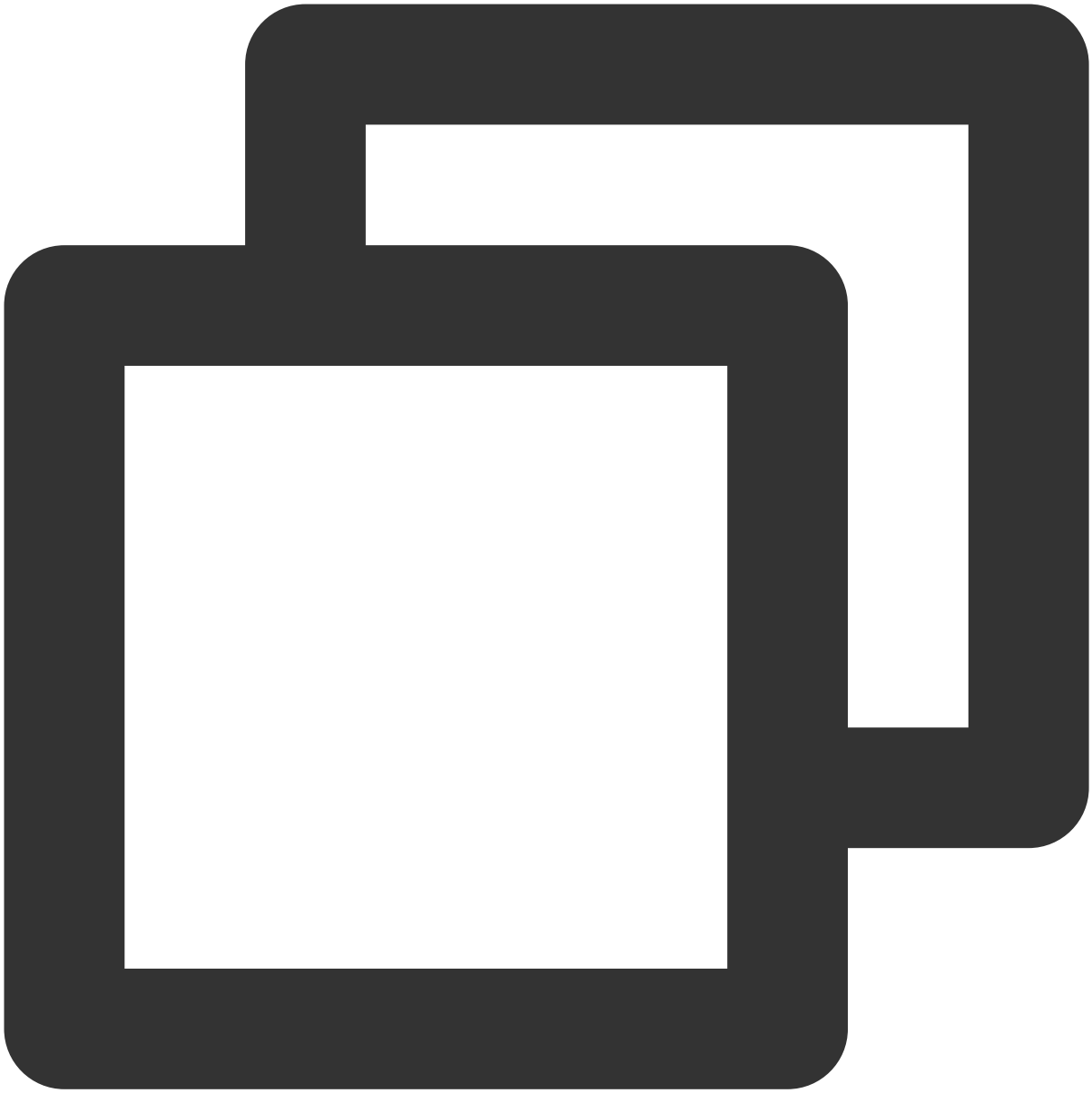
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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



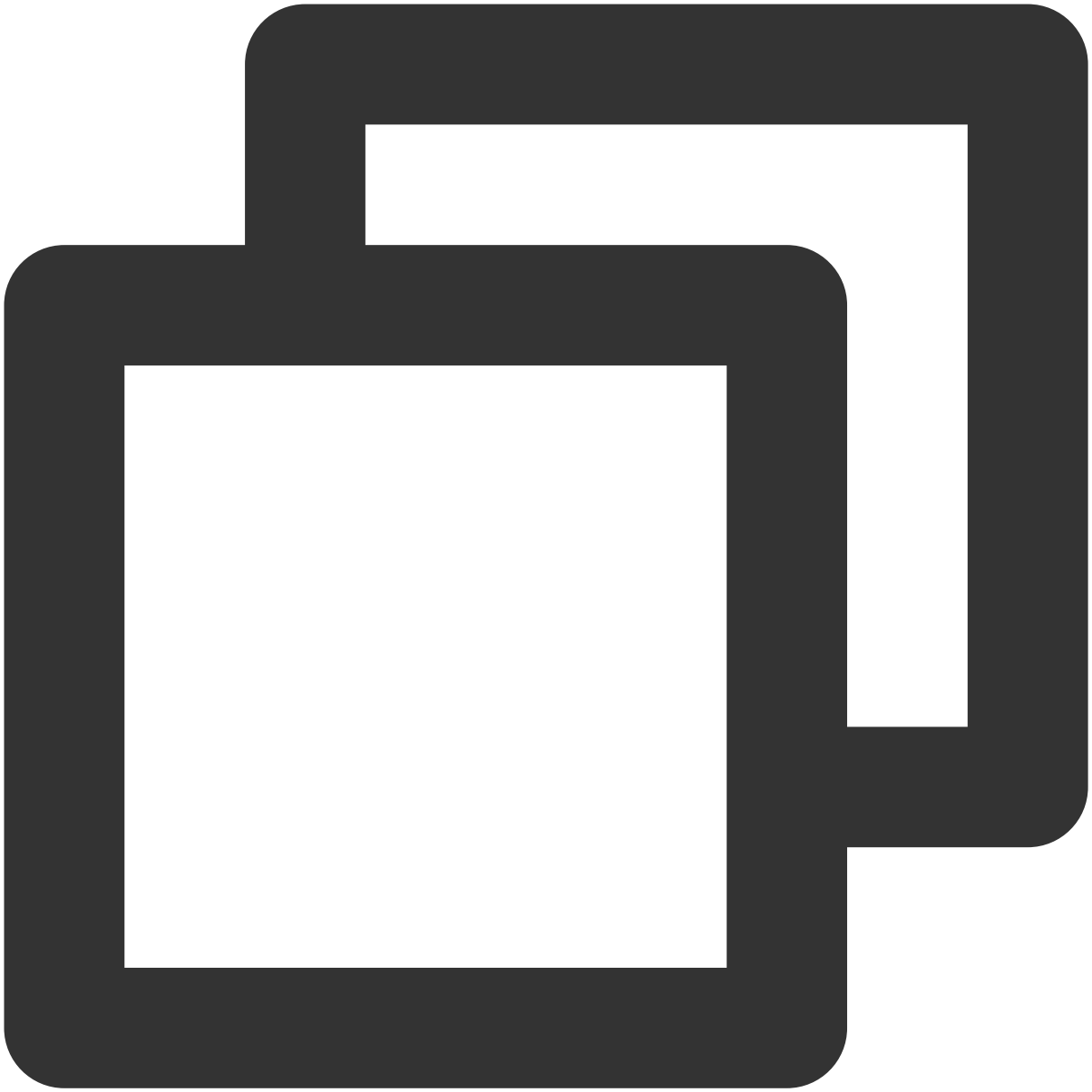
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T18:45:14+0800</CreationTime>
        <EndTime>2022-06-30T18:45:15+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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"
        },
```


版权所有：腾讯云计算（北京）有限责任公司

```
        "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

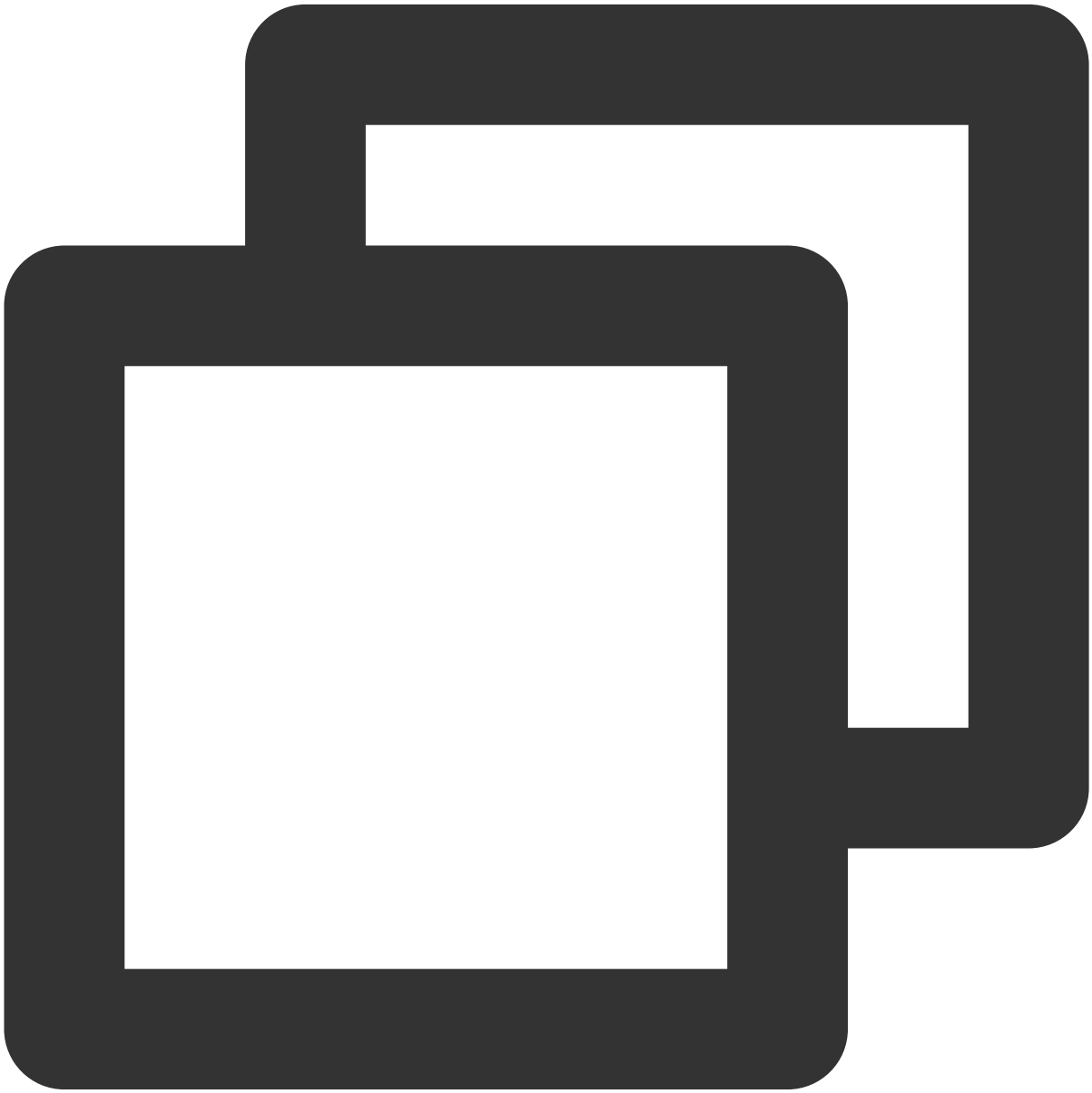
最近更新时间：2024-01-20 16:30:33

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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

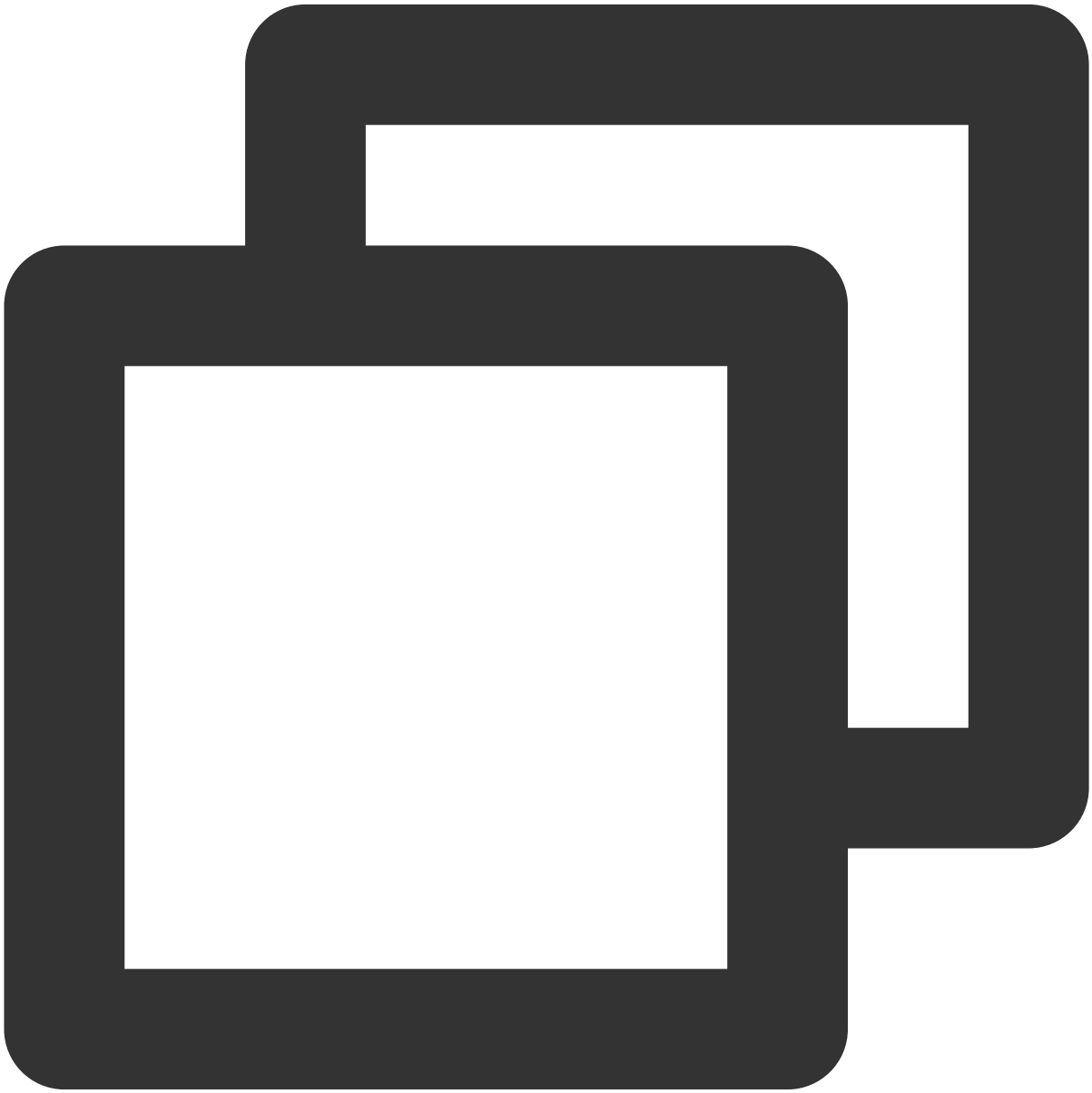
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

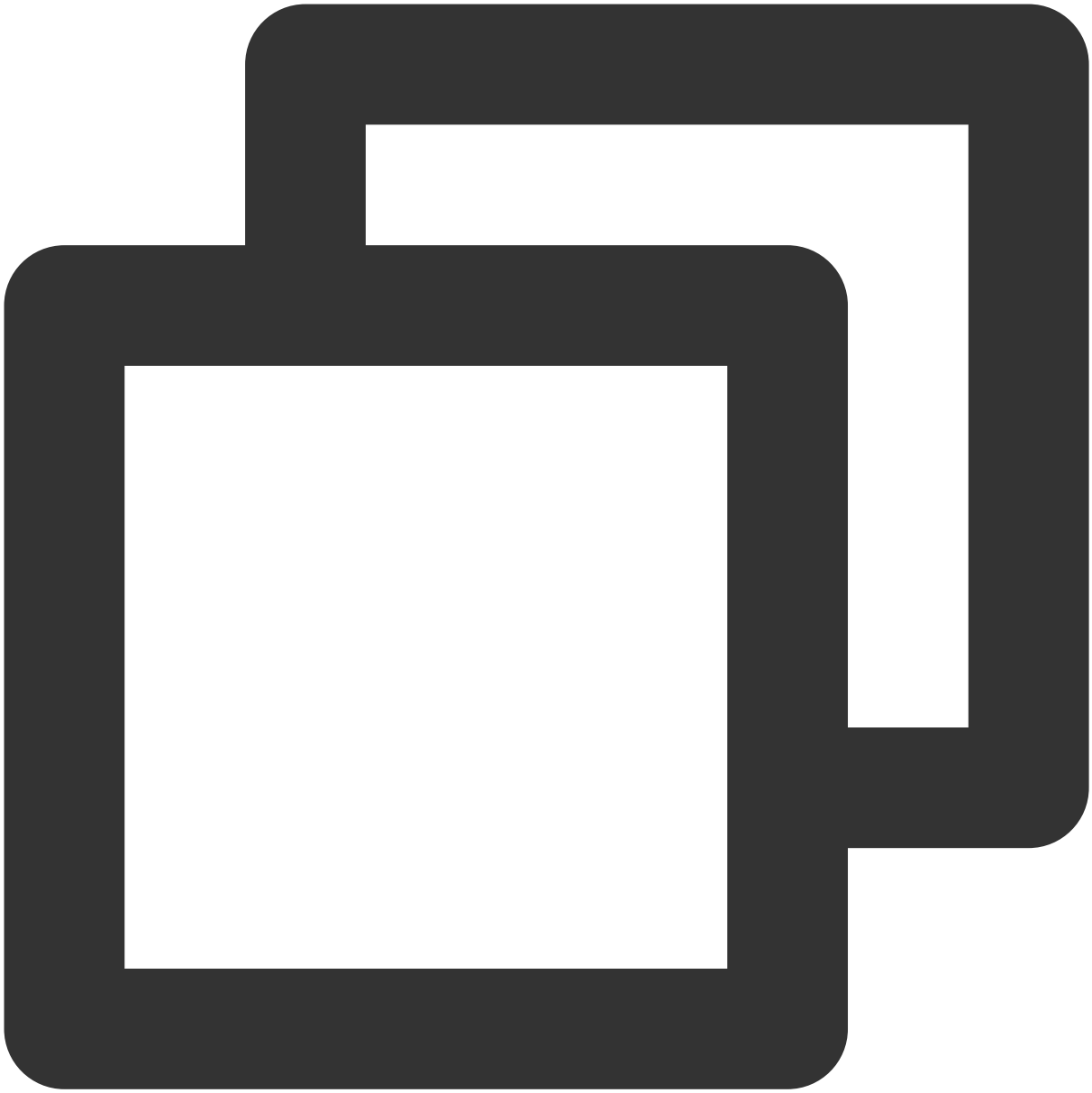
案例 1：通过任务接口触发的任务回调



```
<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"
        },
```


版权所有：腾讯云计算（北京）有限责任公司

```
        "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

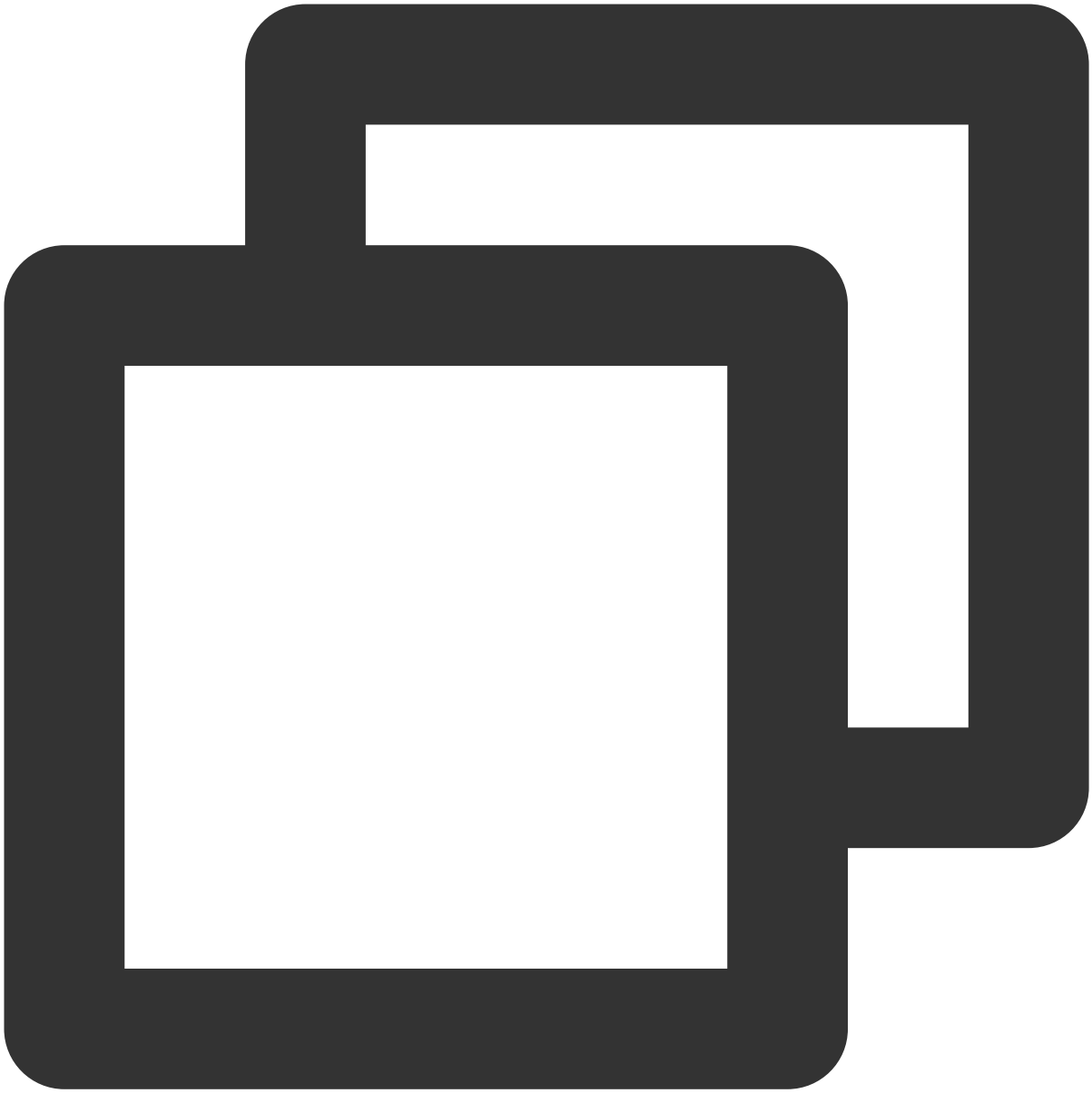
最近更新时间：2024-01-20 16:30:33

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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：

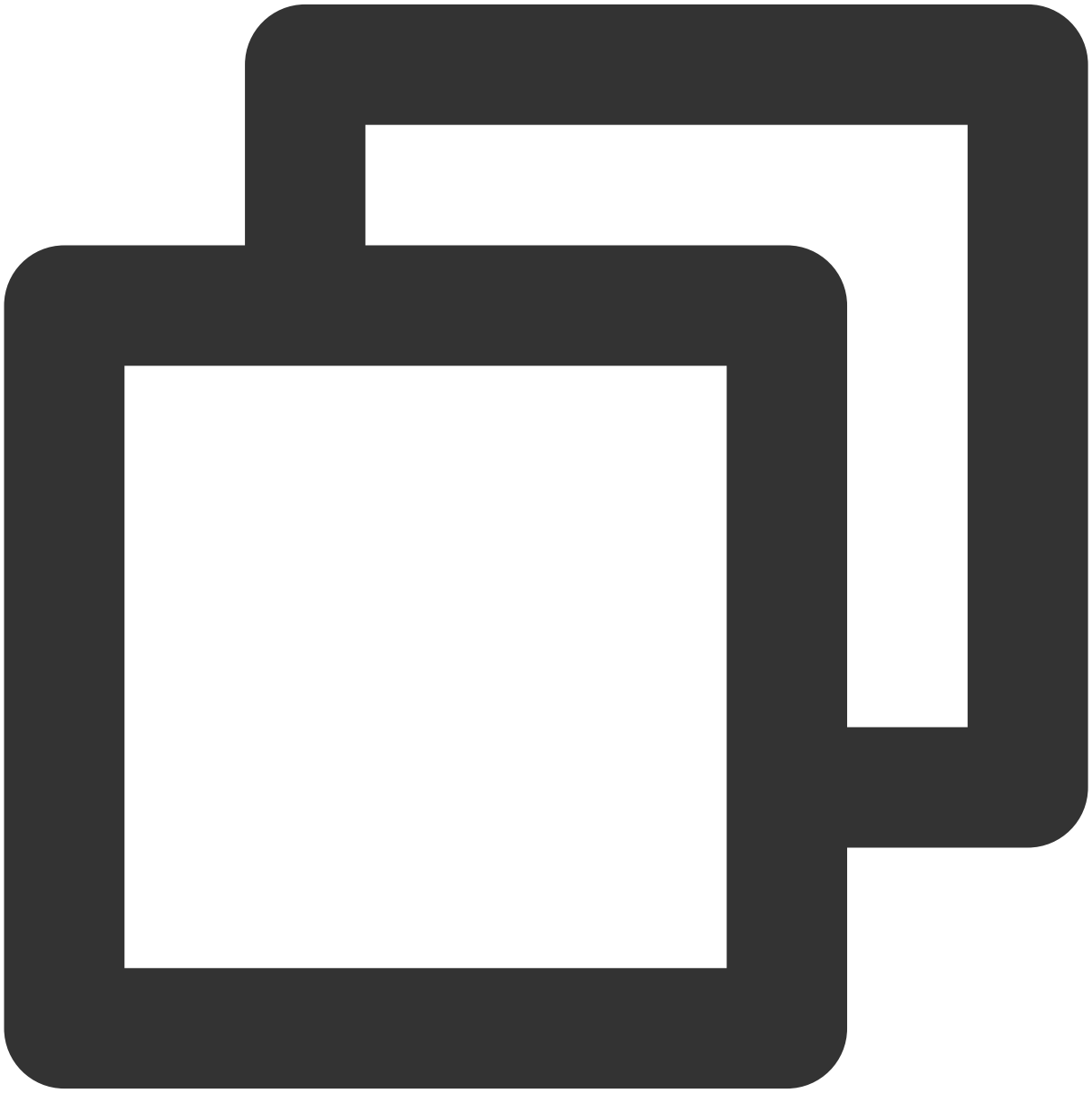
节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String

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

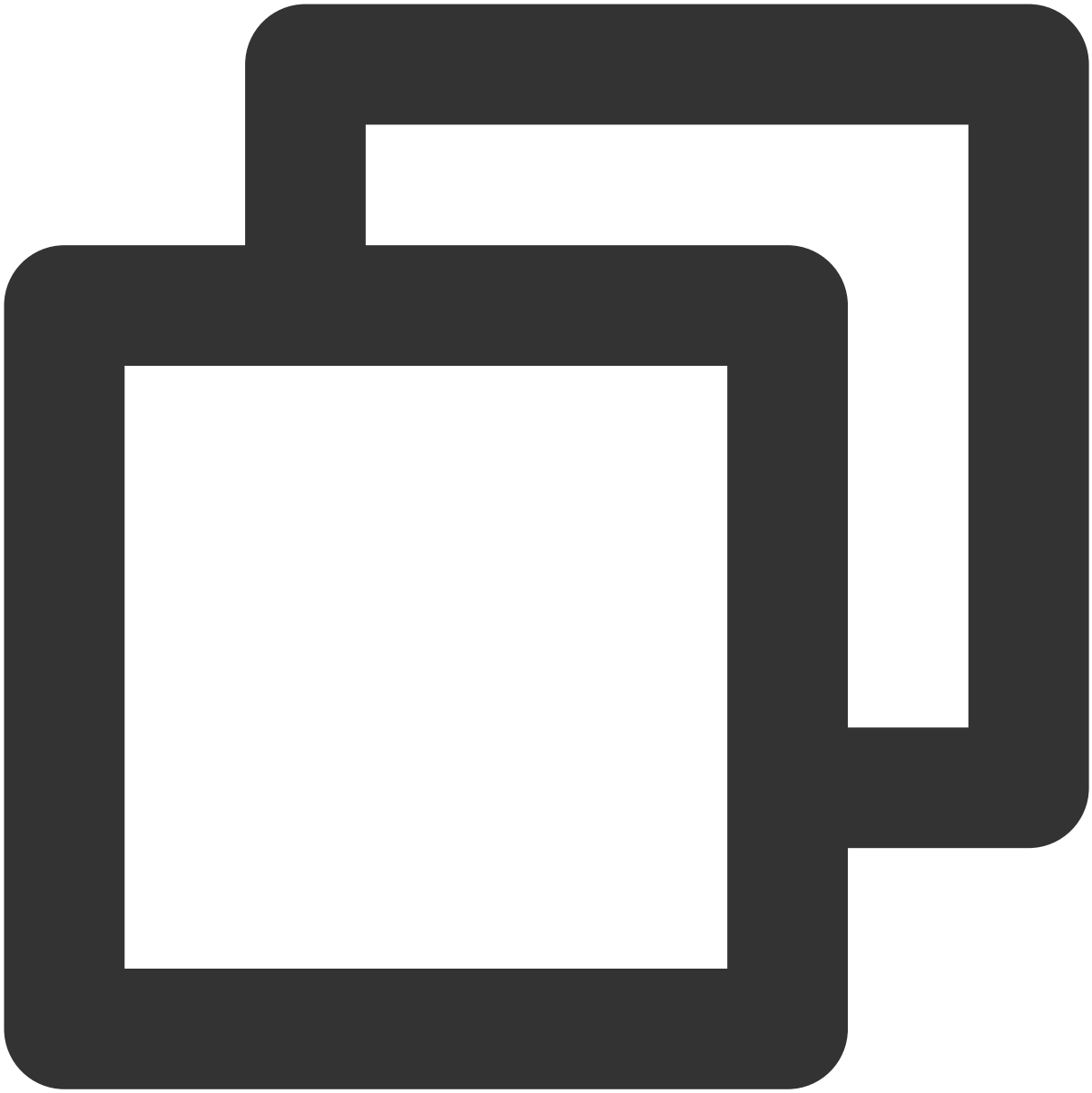


```
<Response>
```

```
    <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"
        },
```


版权所有：腾讯云计算（北京）有限责任公司

```
        "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

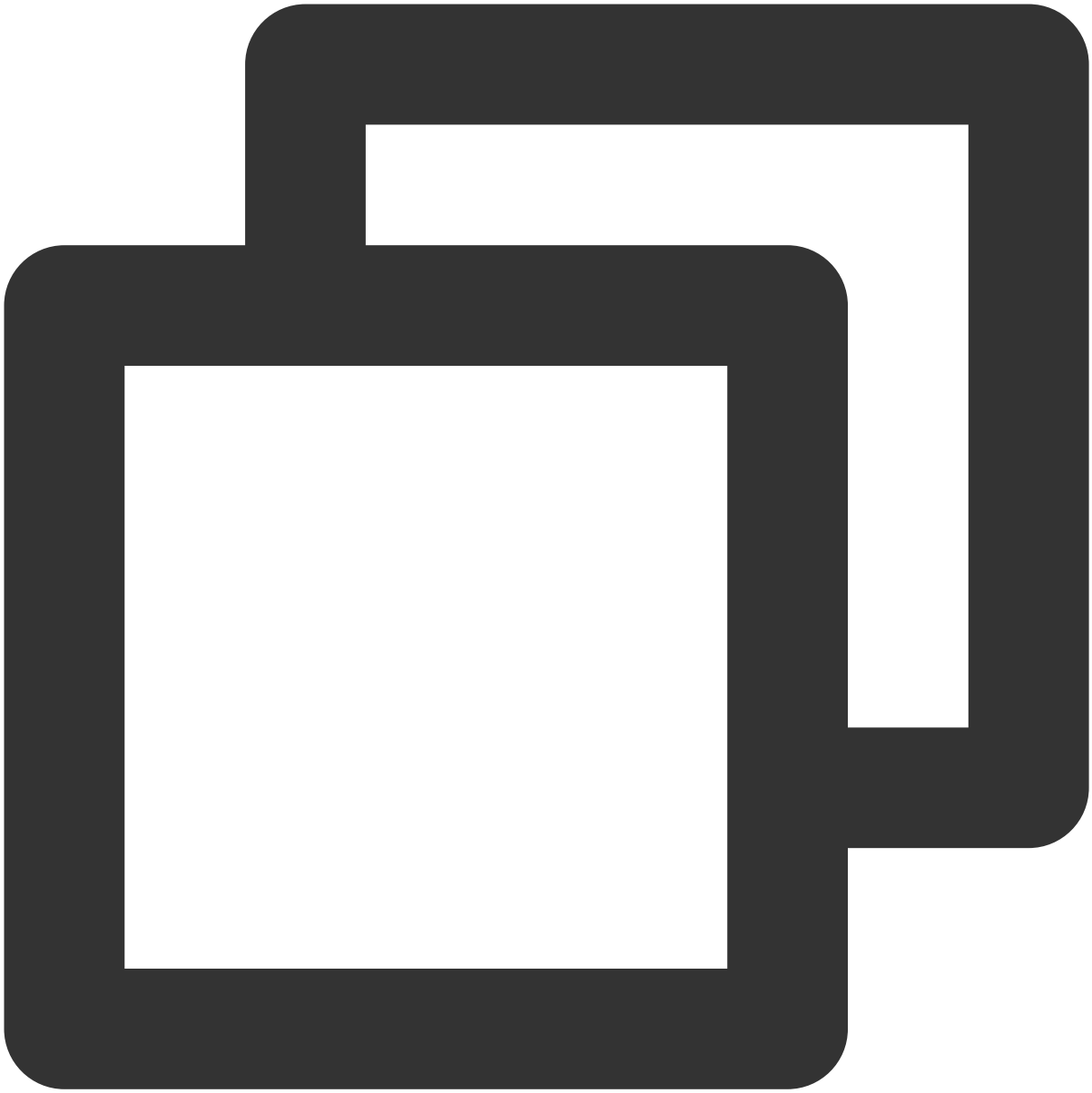
最近更新时间：2024-01-20 16:30:33

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 数组

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

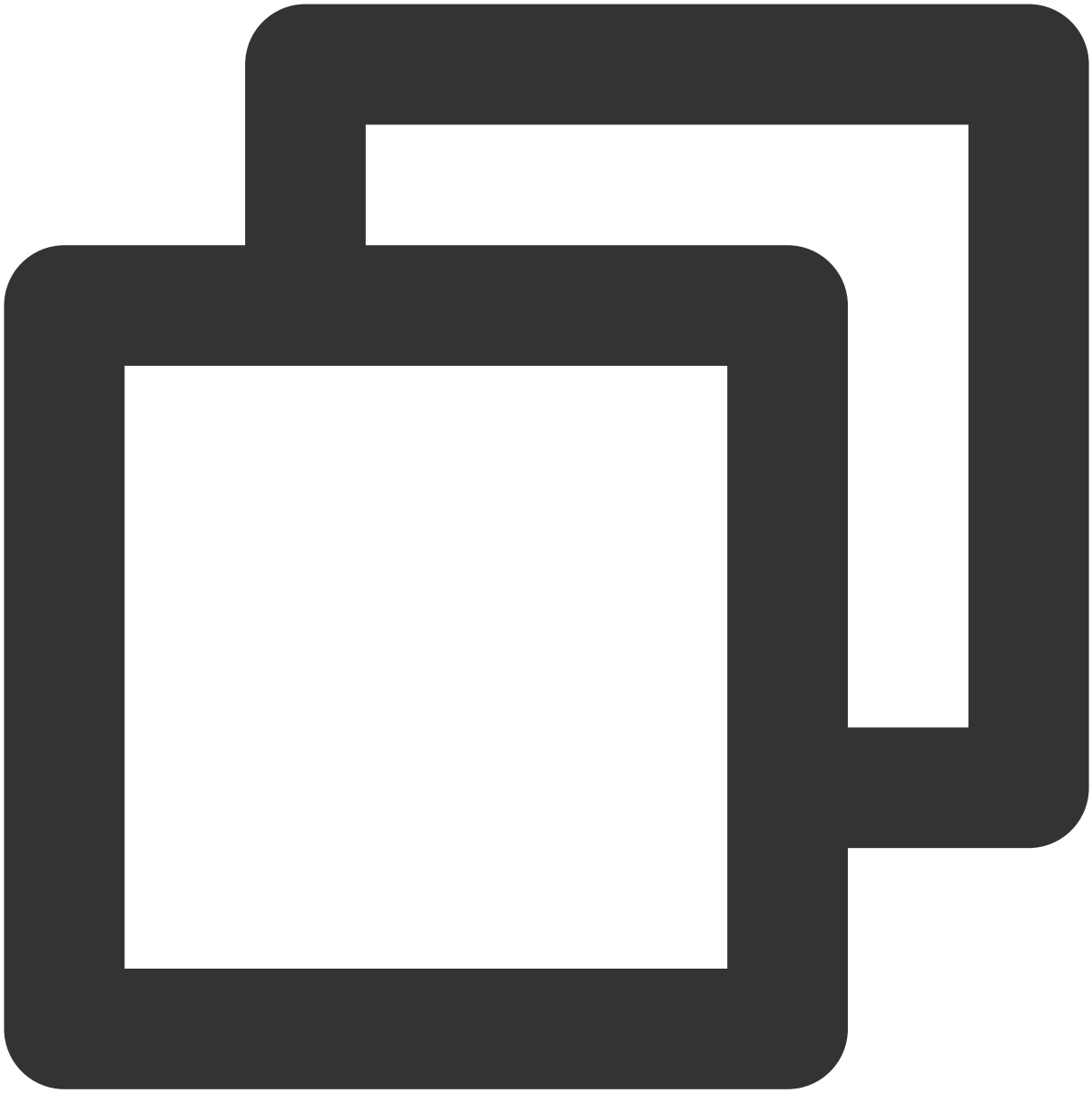
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

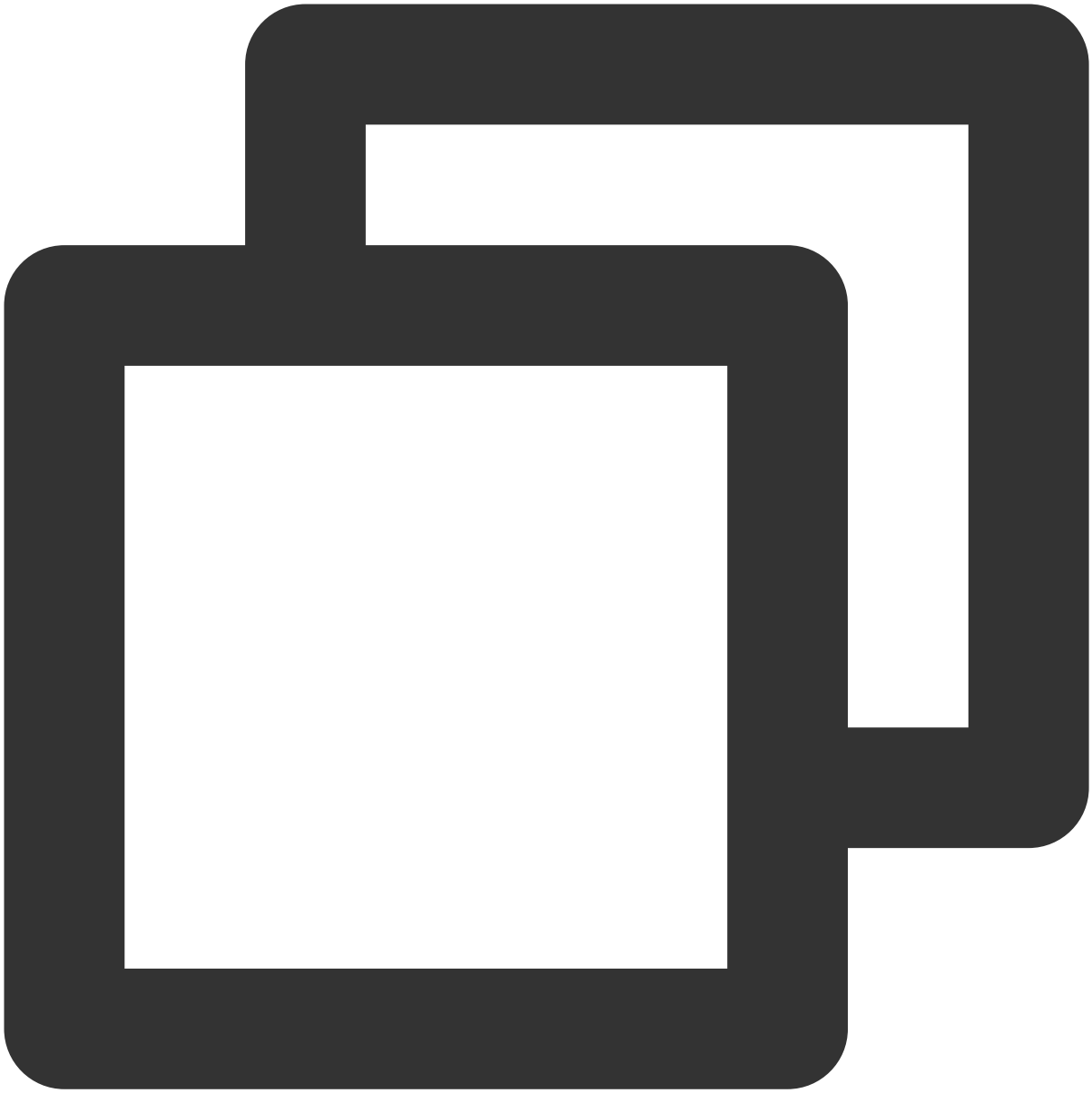


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:20:45+0800</CreationTime>
        <EndTime>2022-06-30T19:21:06+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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



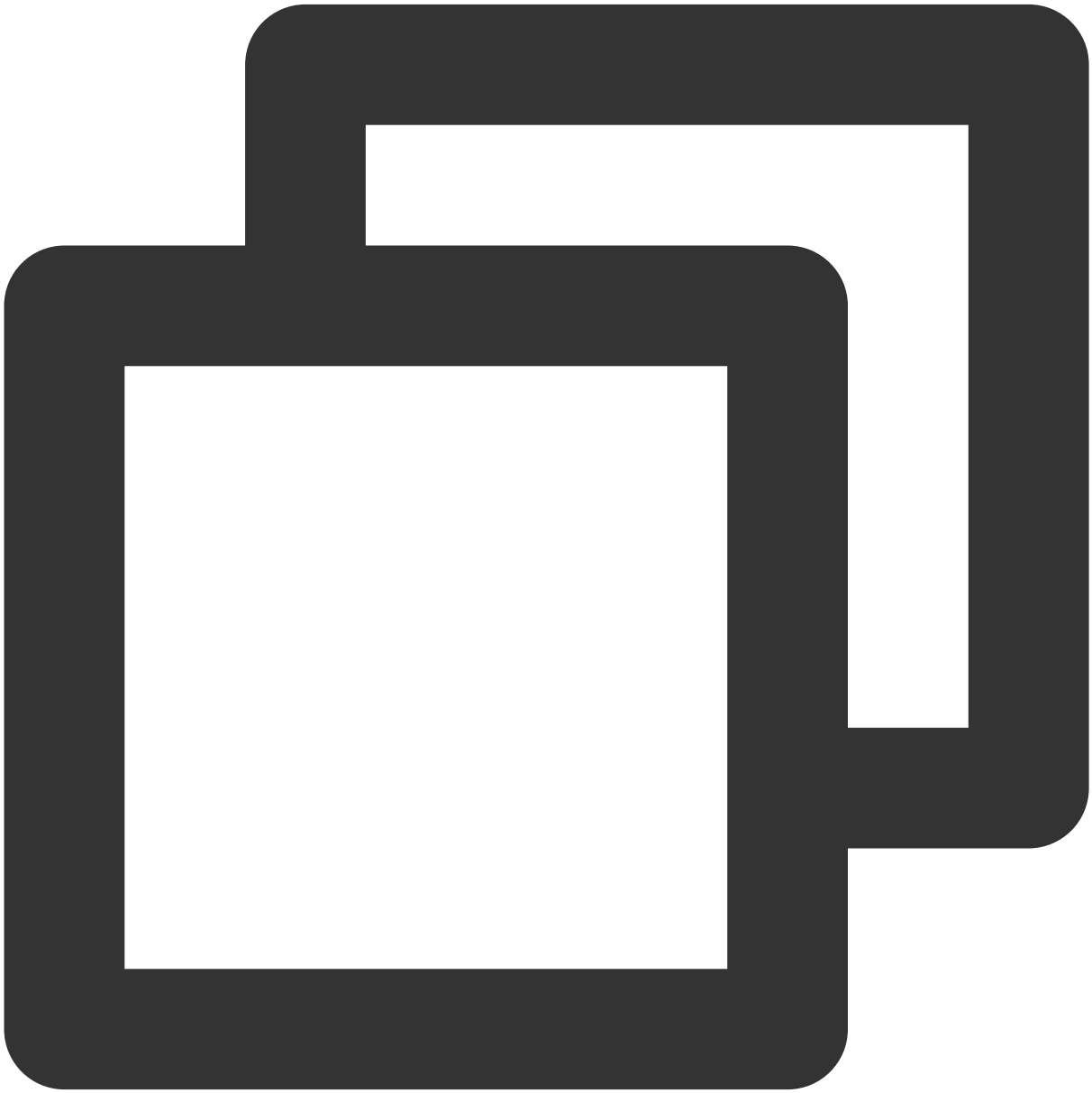
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:20:45+0800</CreationTime>
        <EndTime>2022-06-30T19:21:06+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
```

```
            <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）

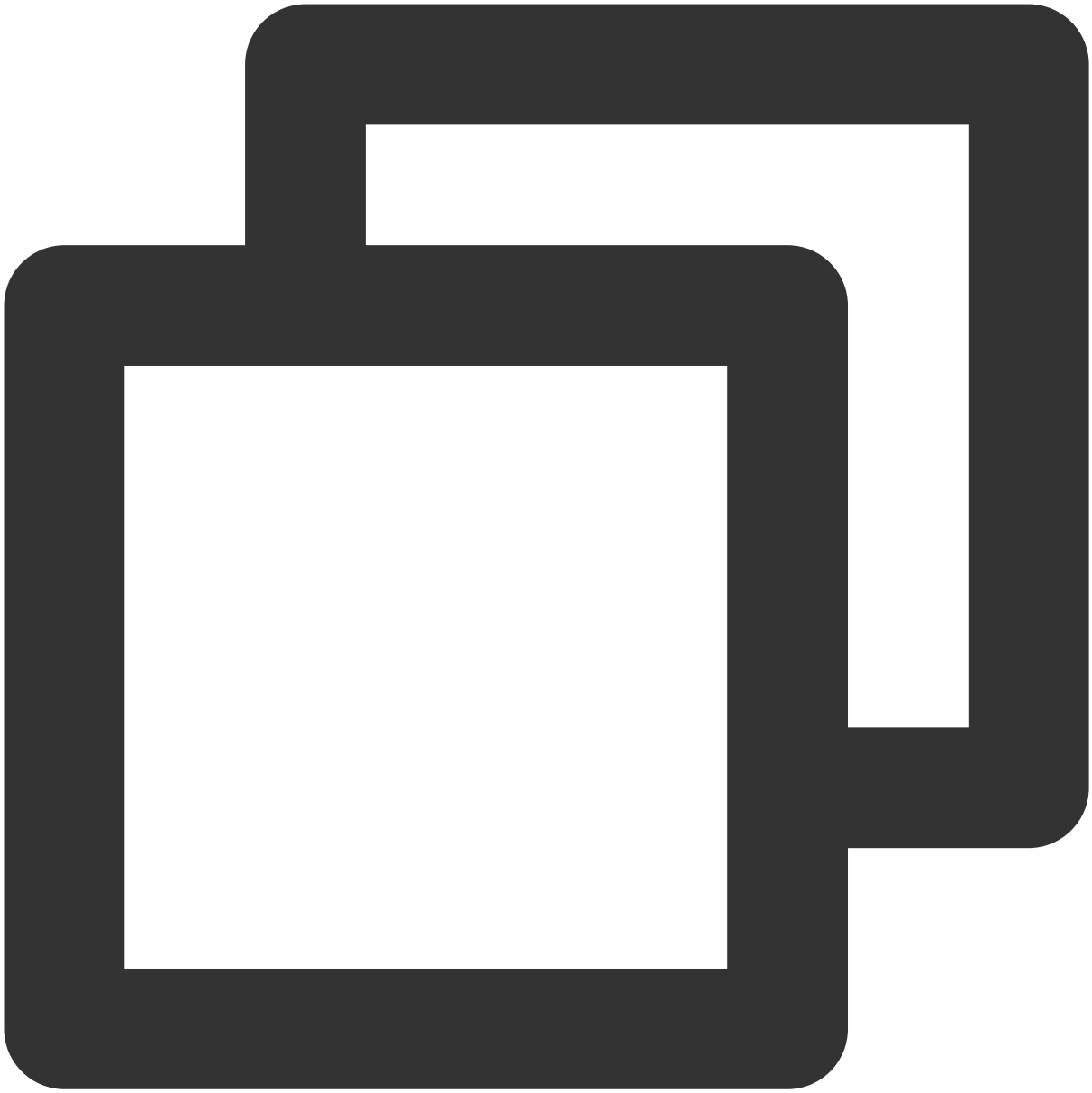
最近更新时间：2024-01-20 16:30:33

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>2022-06-30T19:14:31+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 数组

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

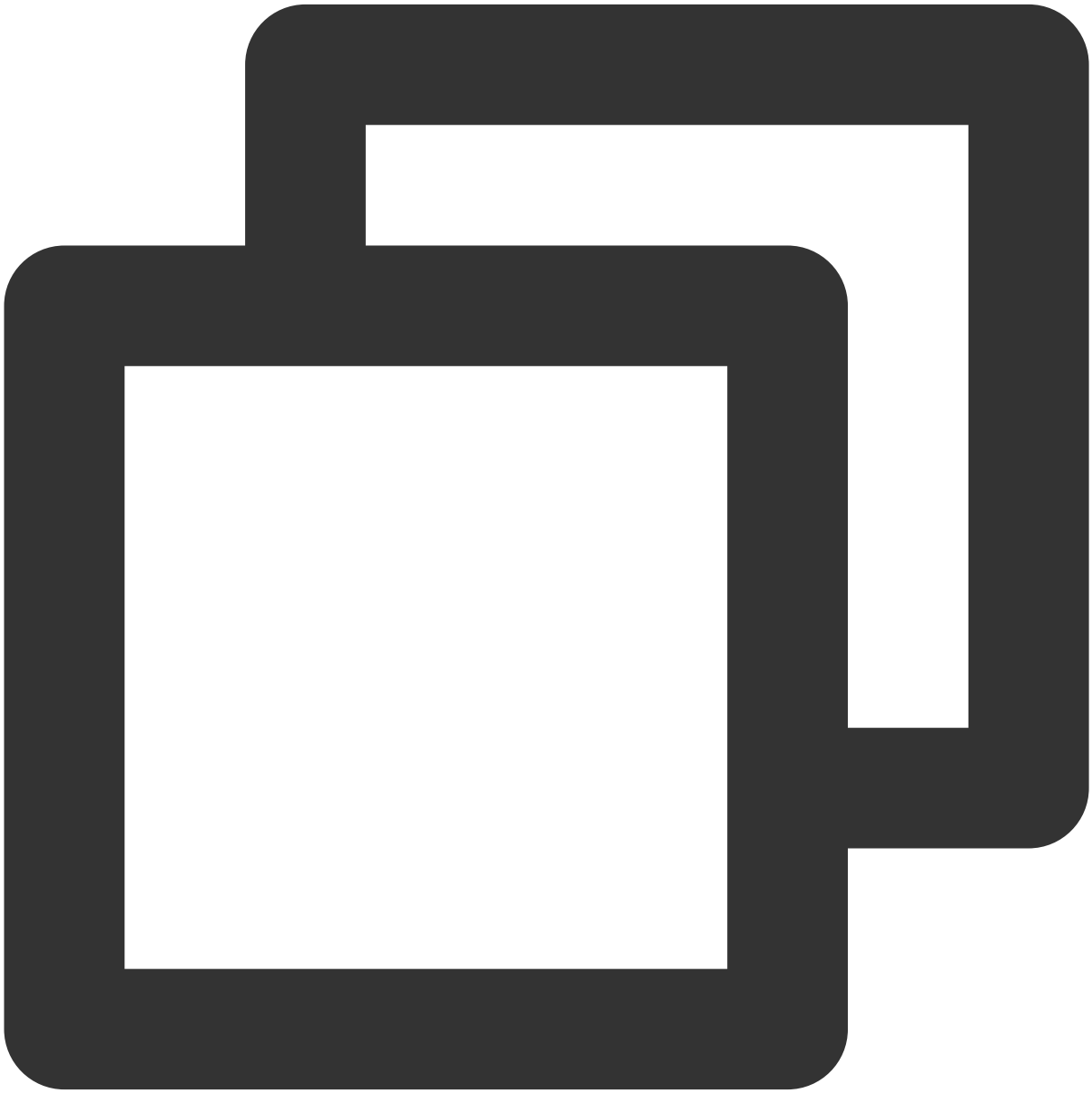
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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>
```

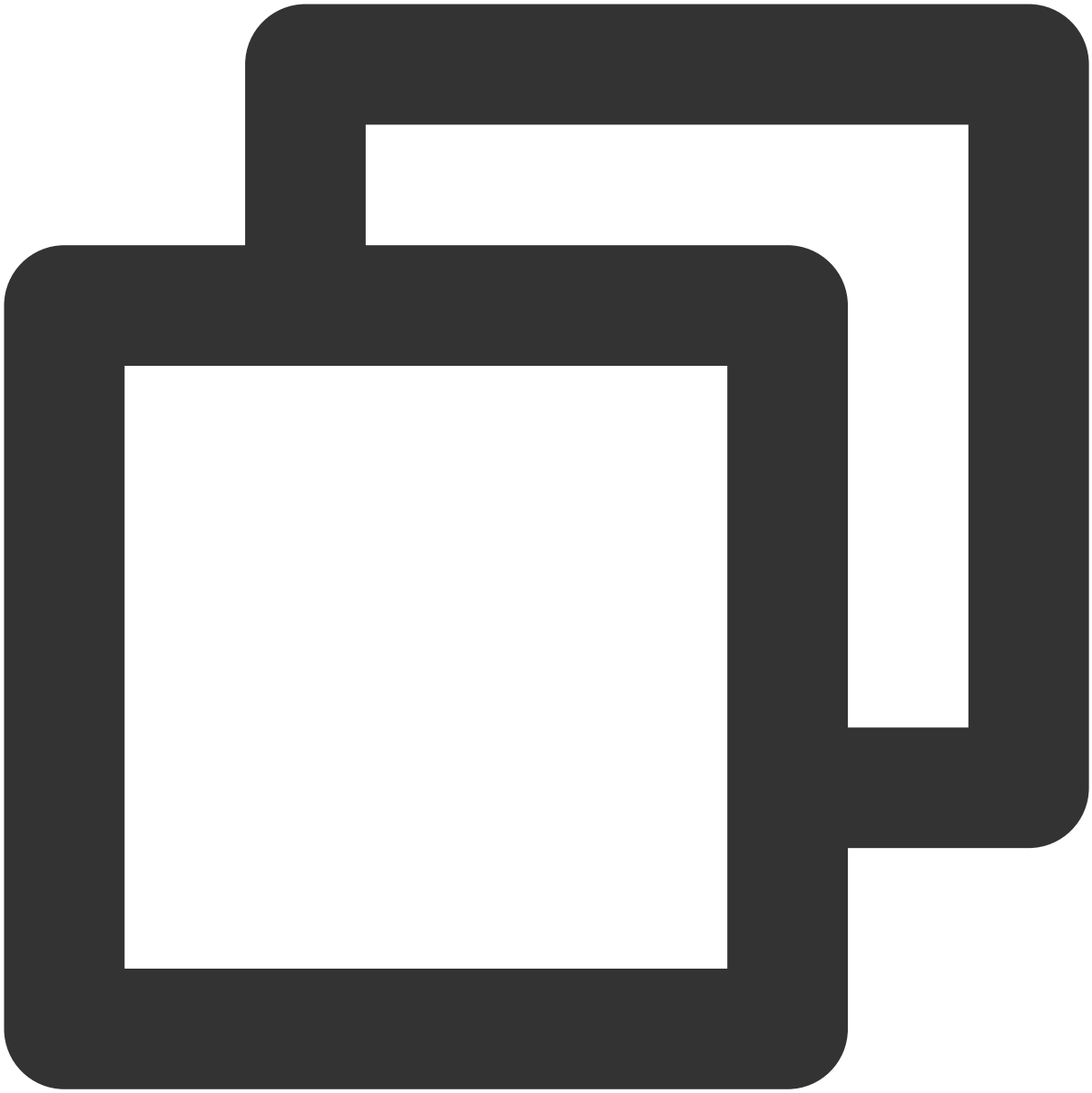
版权所有：腾讯云计算（北京）有限责任公司

```
        <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>
```

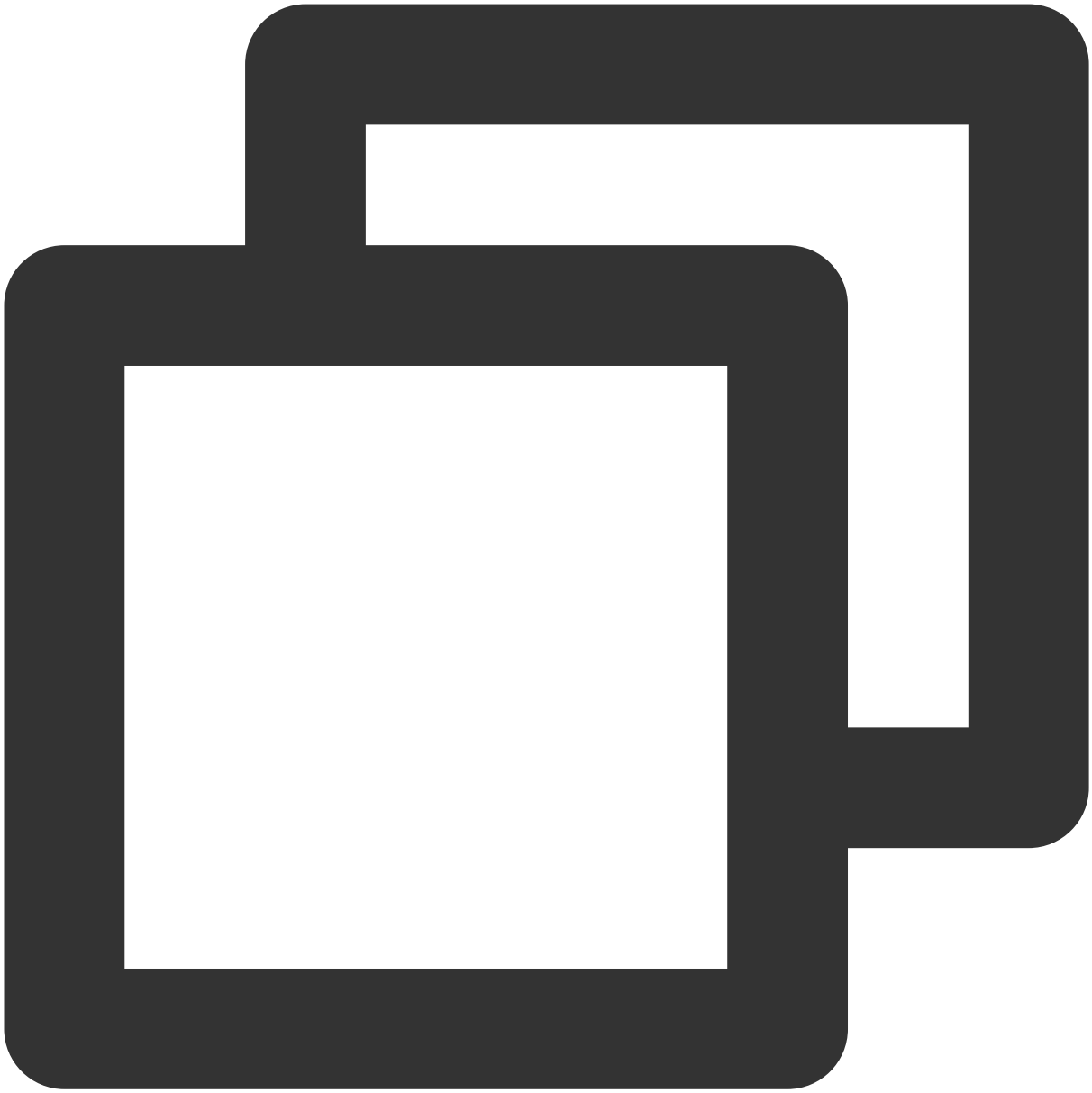
```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <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

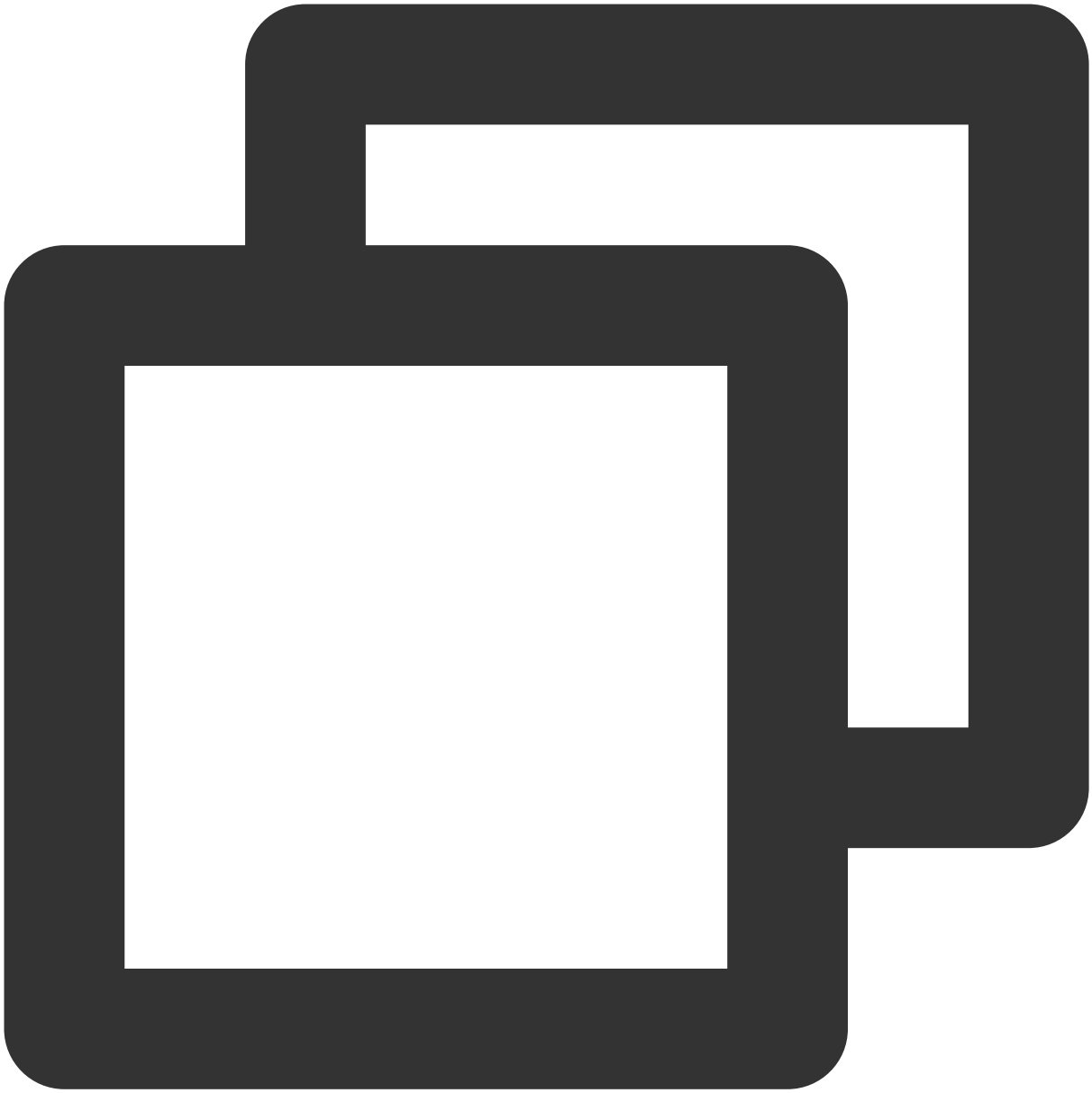
最近更新时间：2024-01-20 16:30:33

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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 数组

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

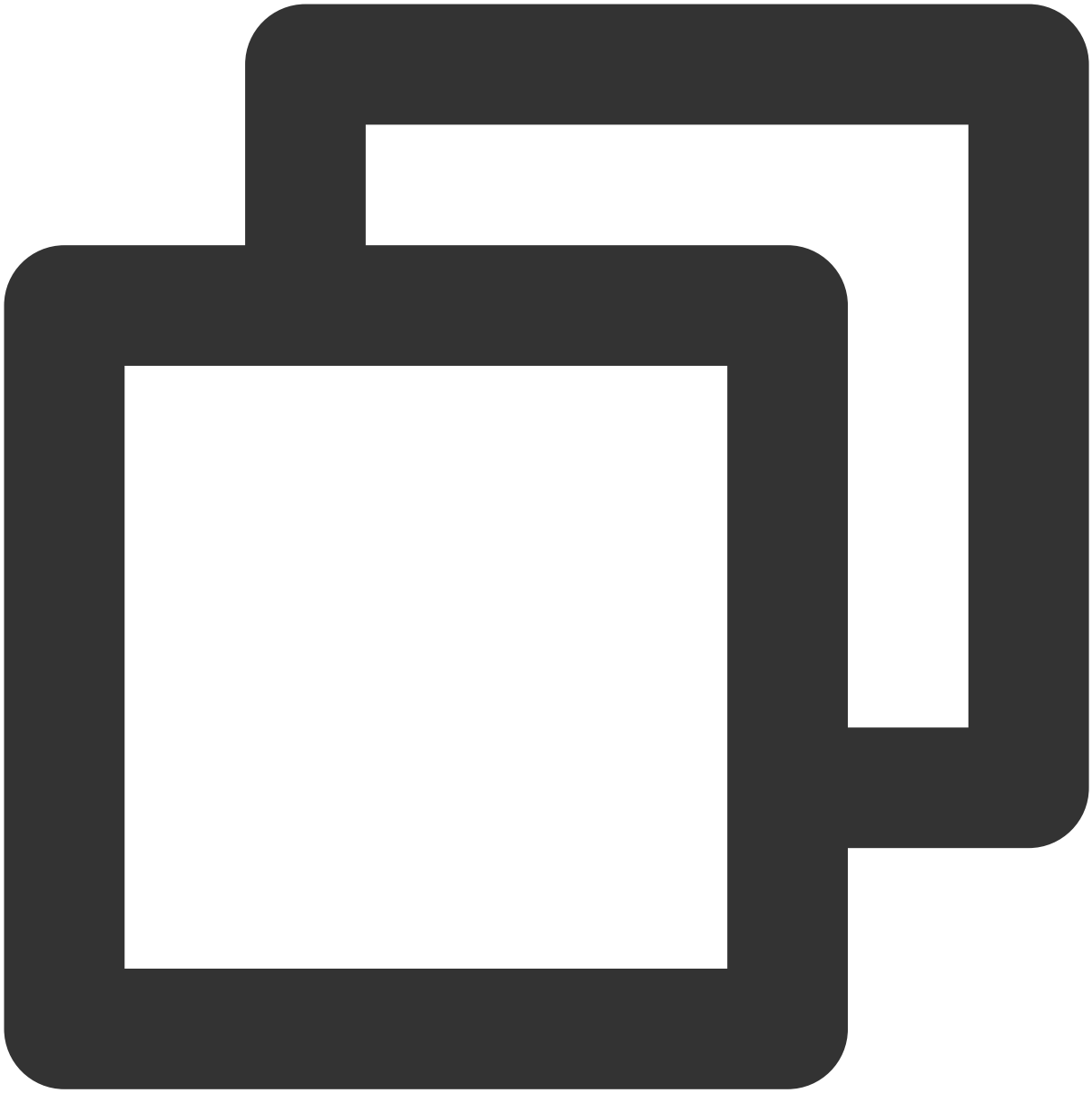
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

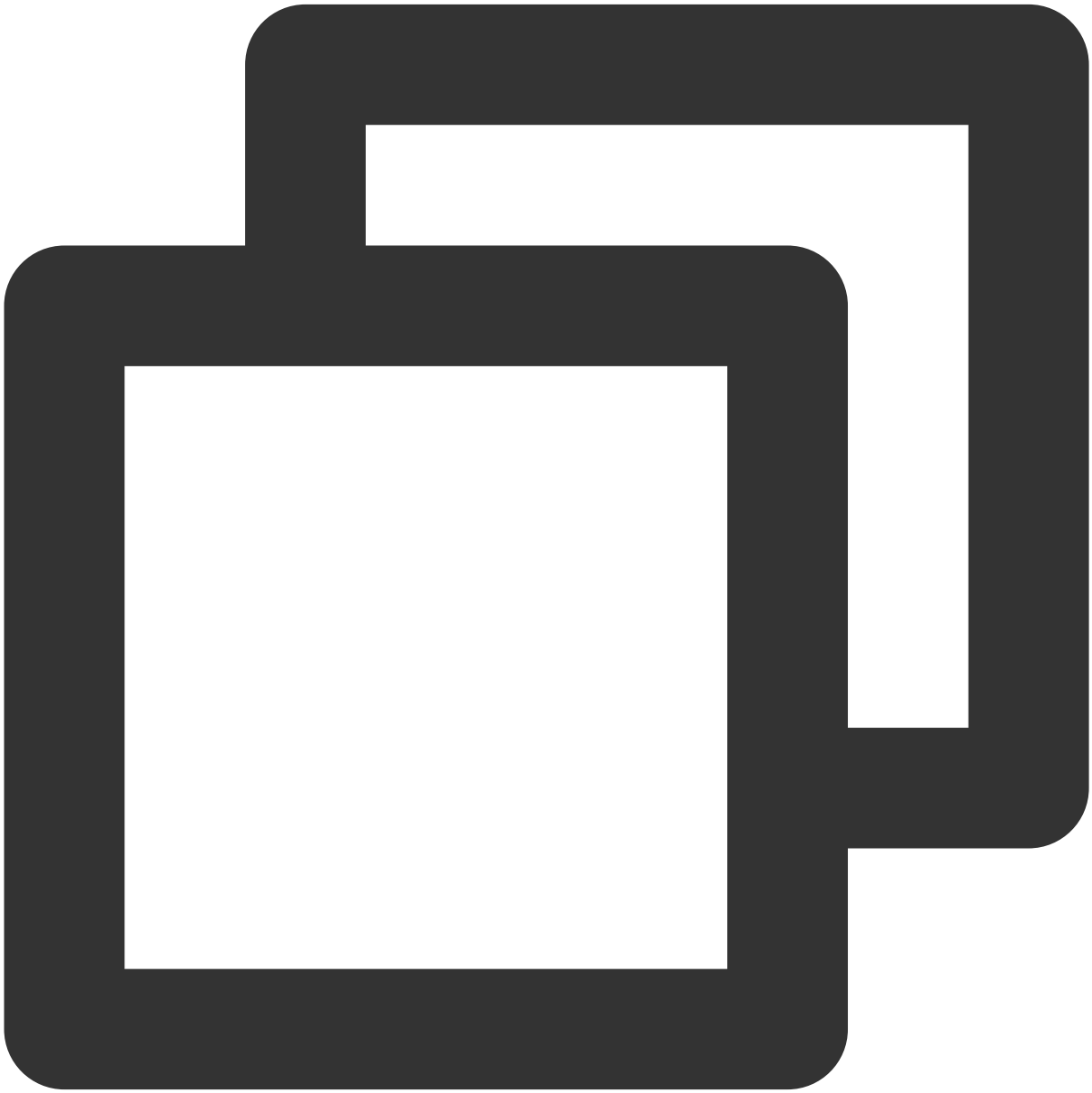


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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

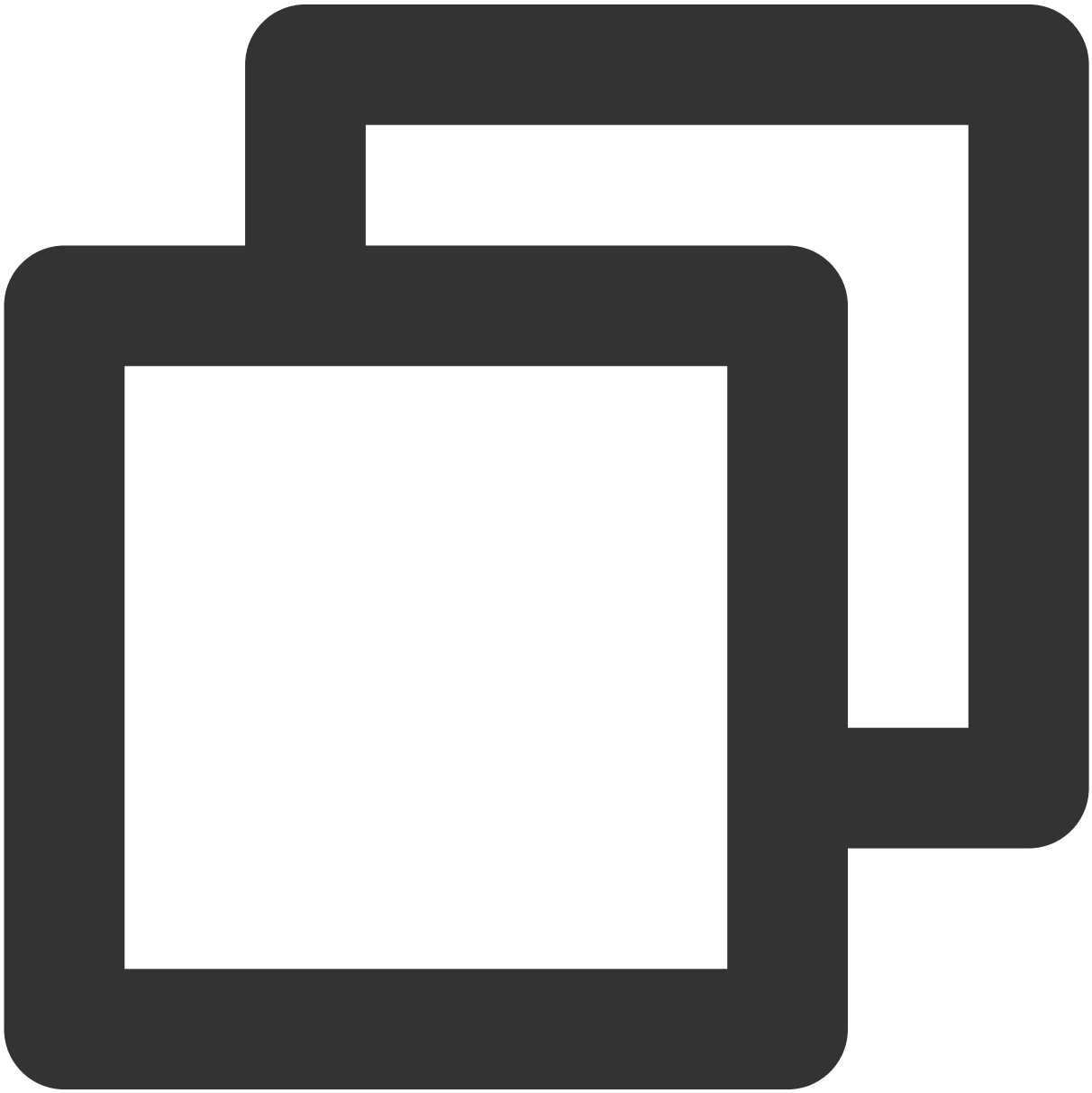


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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",
```


版权所有：腾讯云计算（北京）有限责任公司

```
                "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

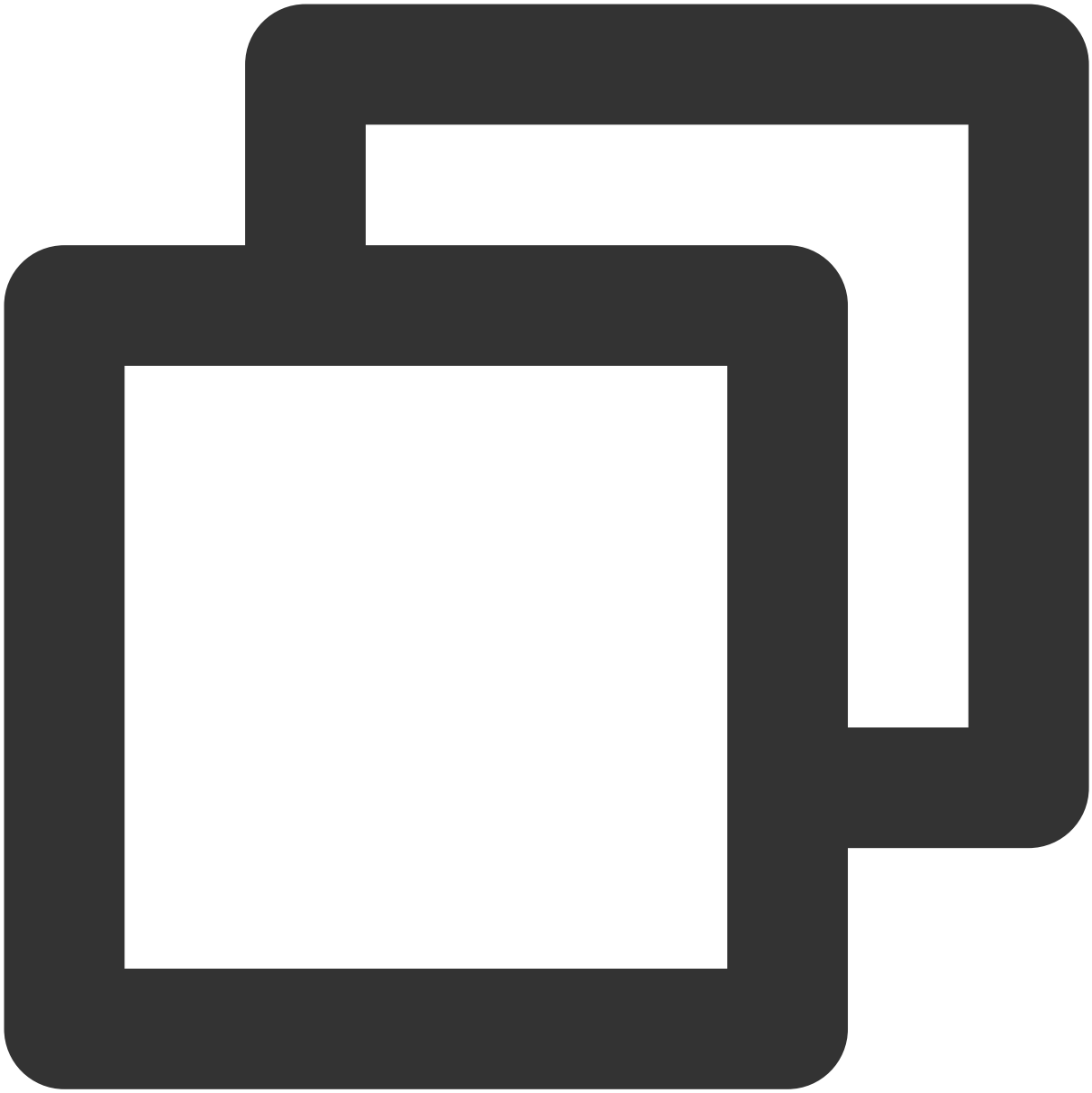
最近更新时间：2024-01-20 16:30:33

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 数组

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

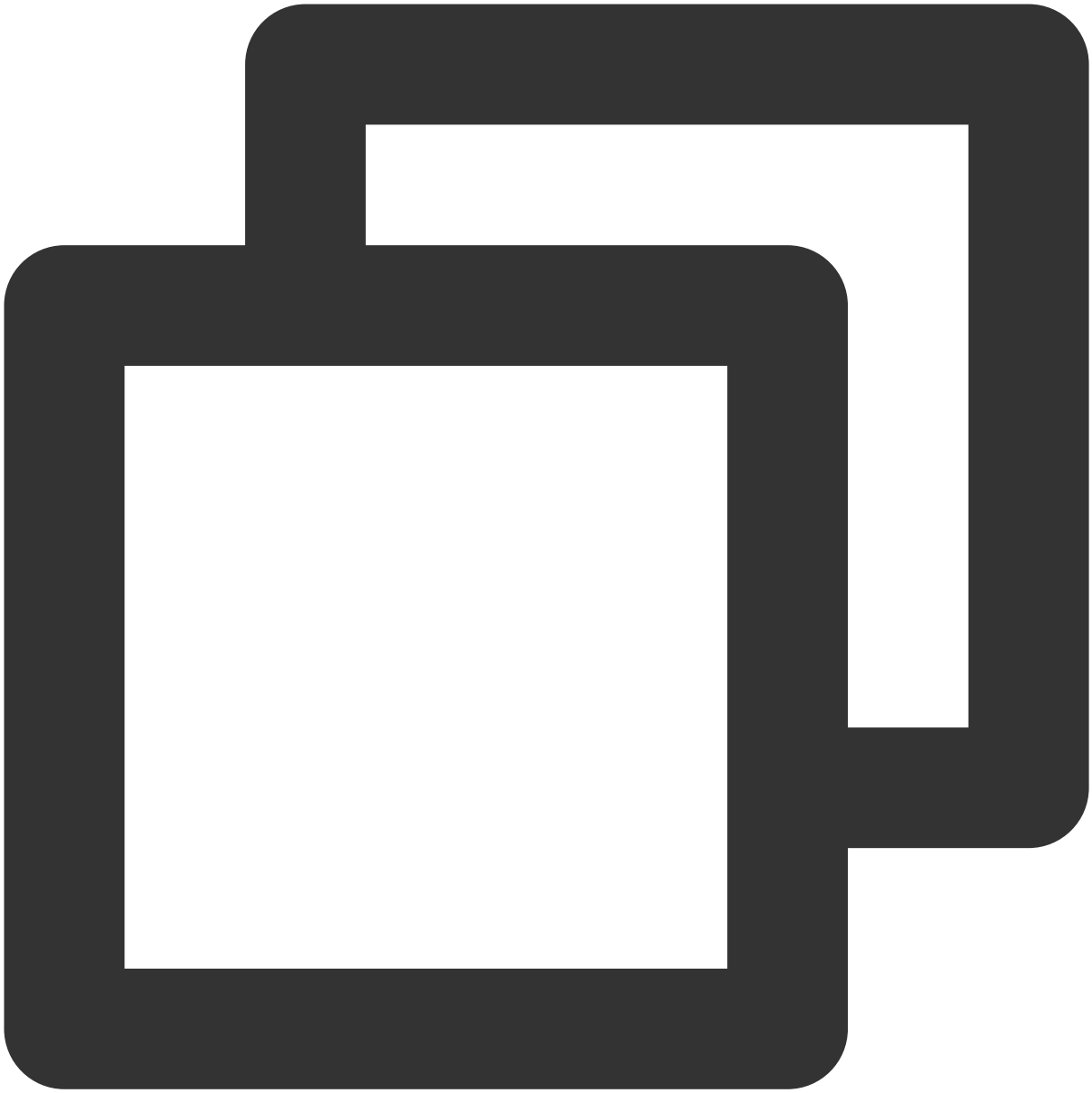
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

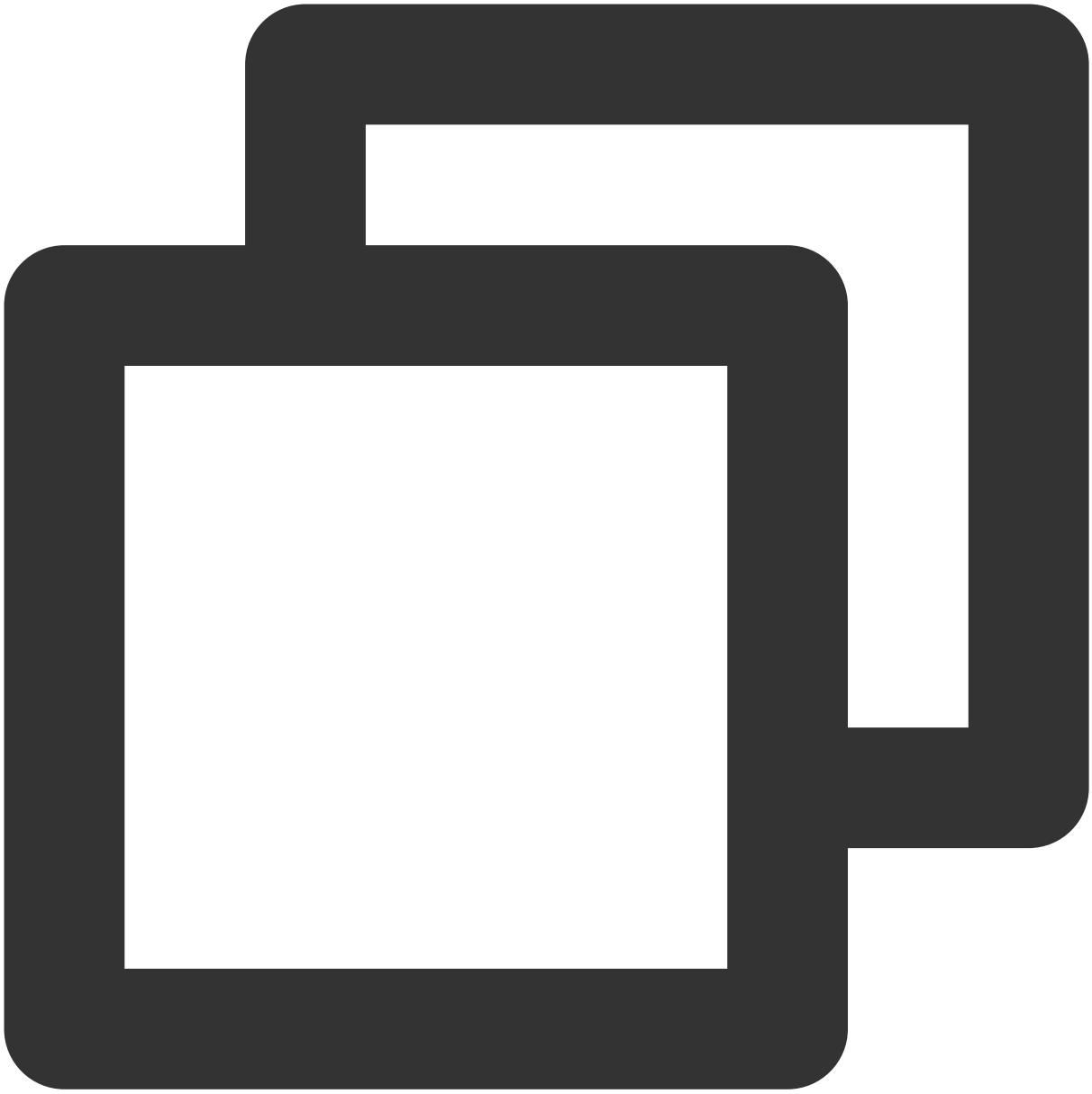


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:25:11+0800</CreationTime>
        <EndTime>2022-06-30T19:25:14+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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

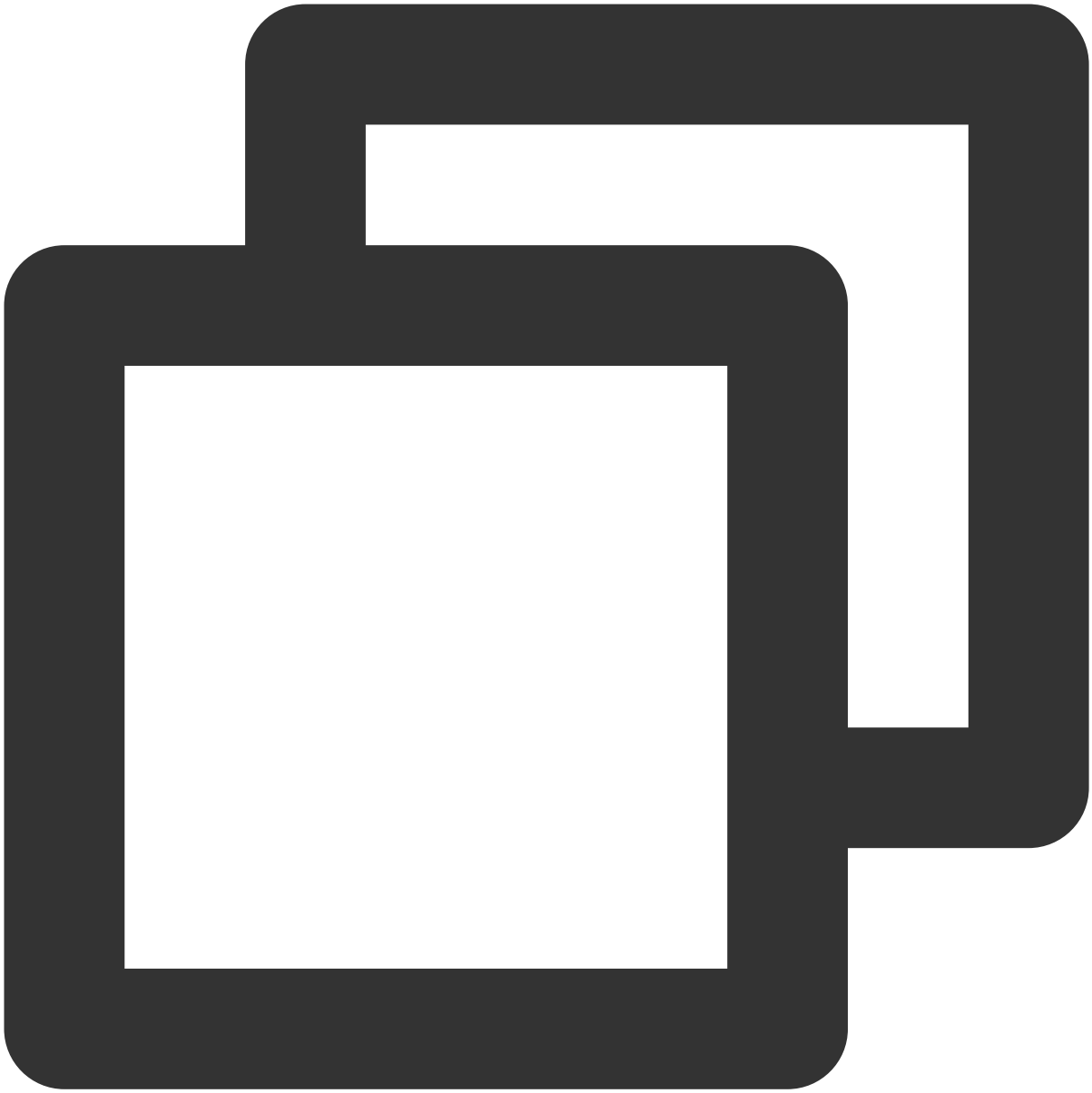


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:25:11+0800</CreationTime>
        <EndTime>2022-06-30T19:25:14+0800</EndTime>
```

```
        <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": [{
```

```
                    "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

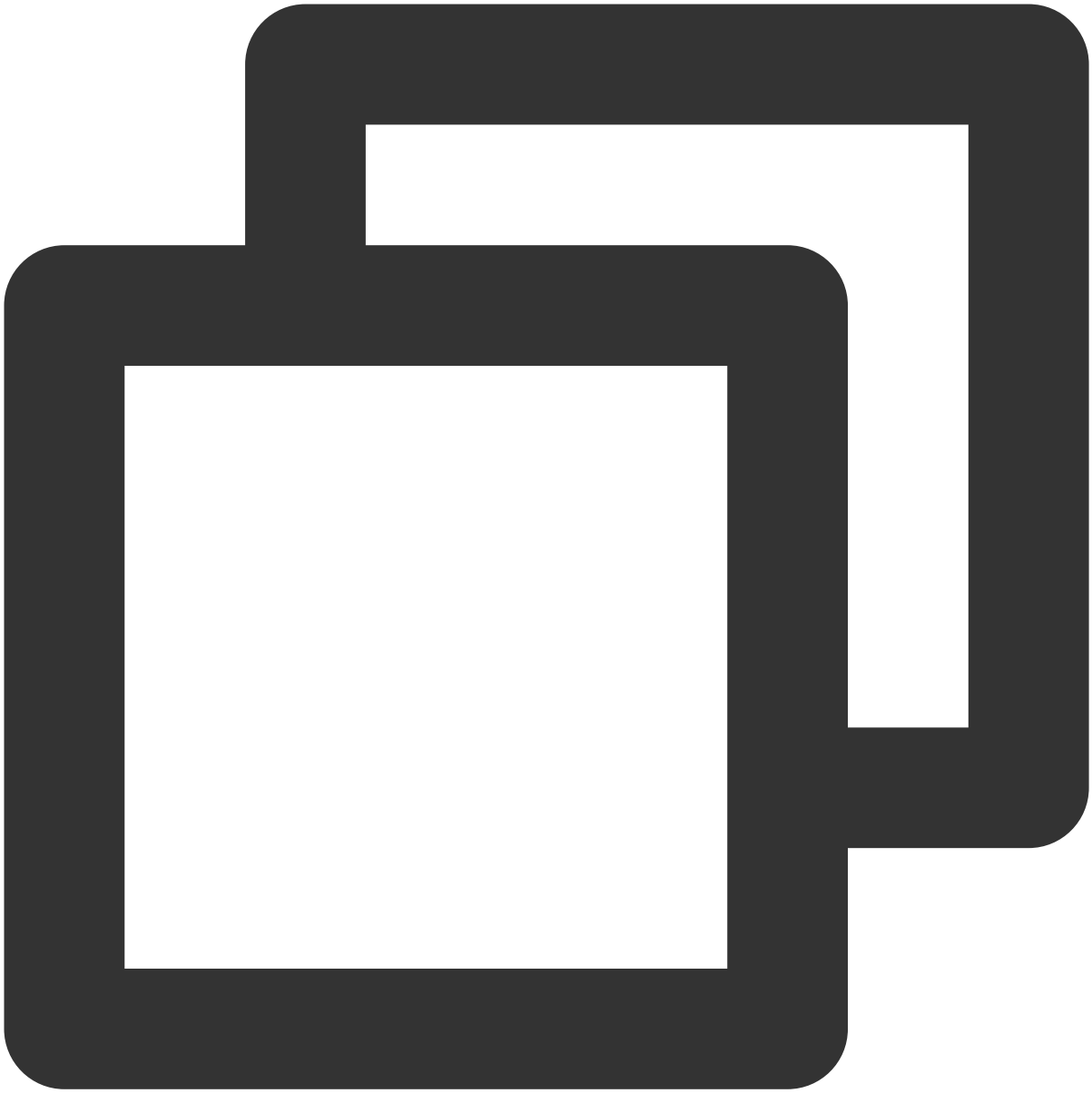
最近更新时间：2024-01-20 16:30:33

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 数组

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

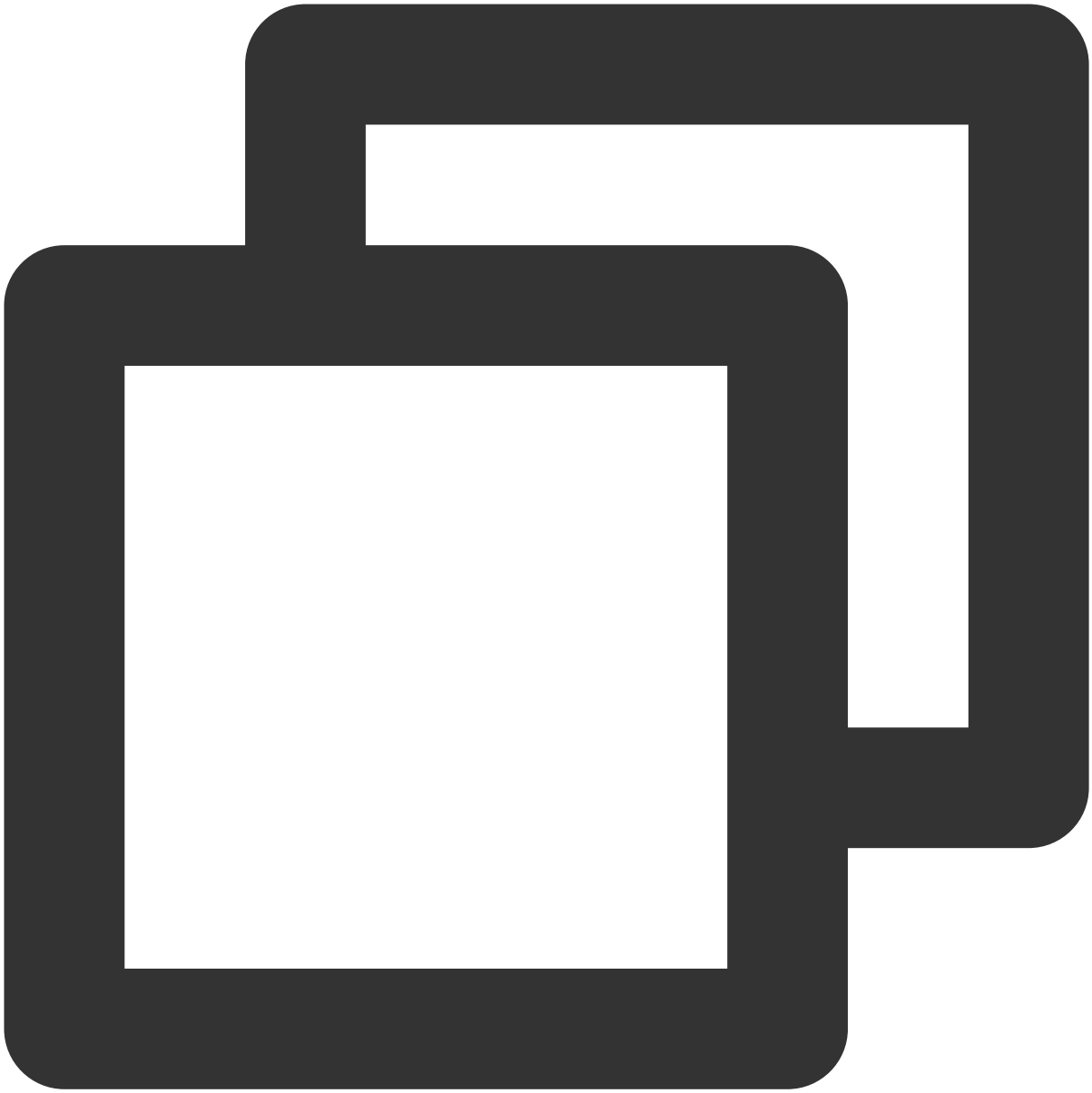
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

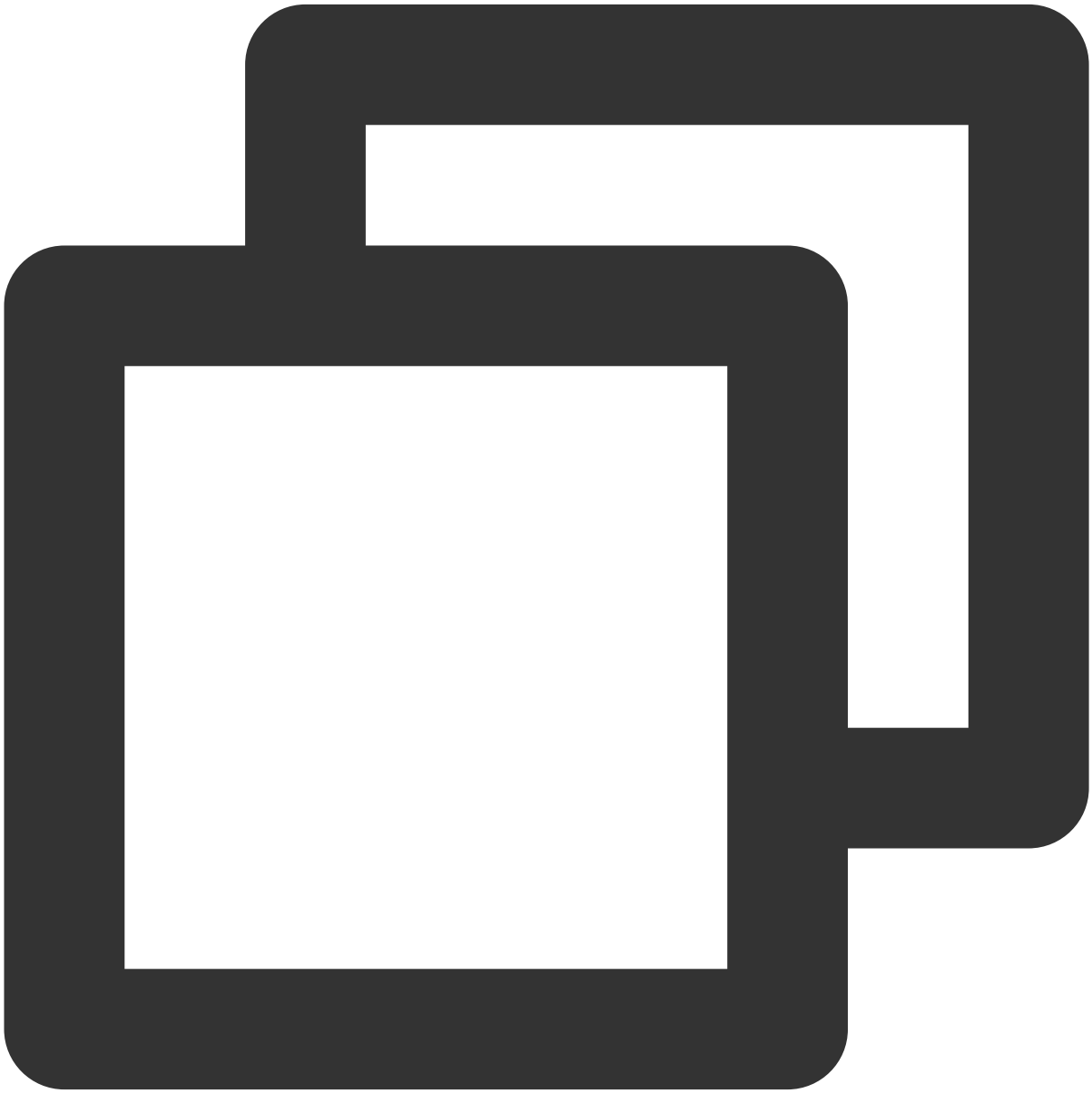
案例 1：通过任务接口触发的任务回调



```
<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"
        },
```

```
        "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

超分辨率

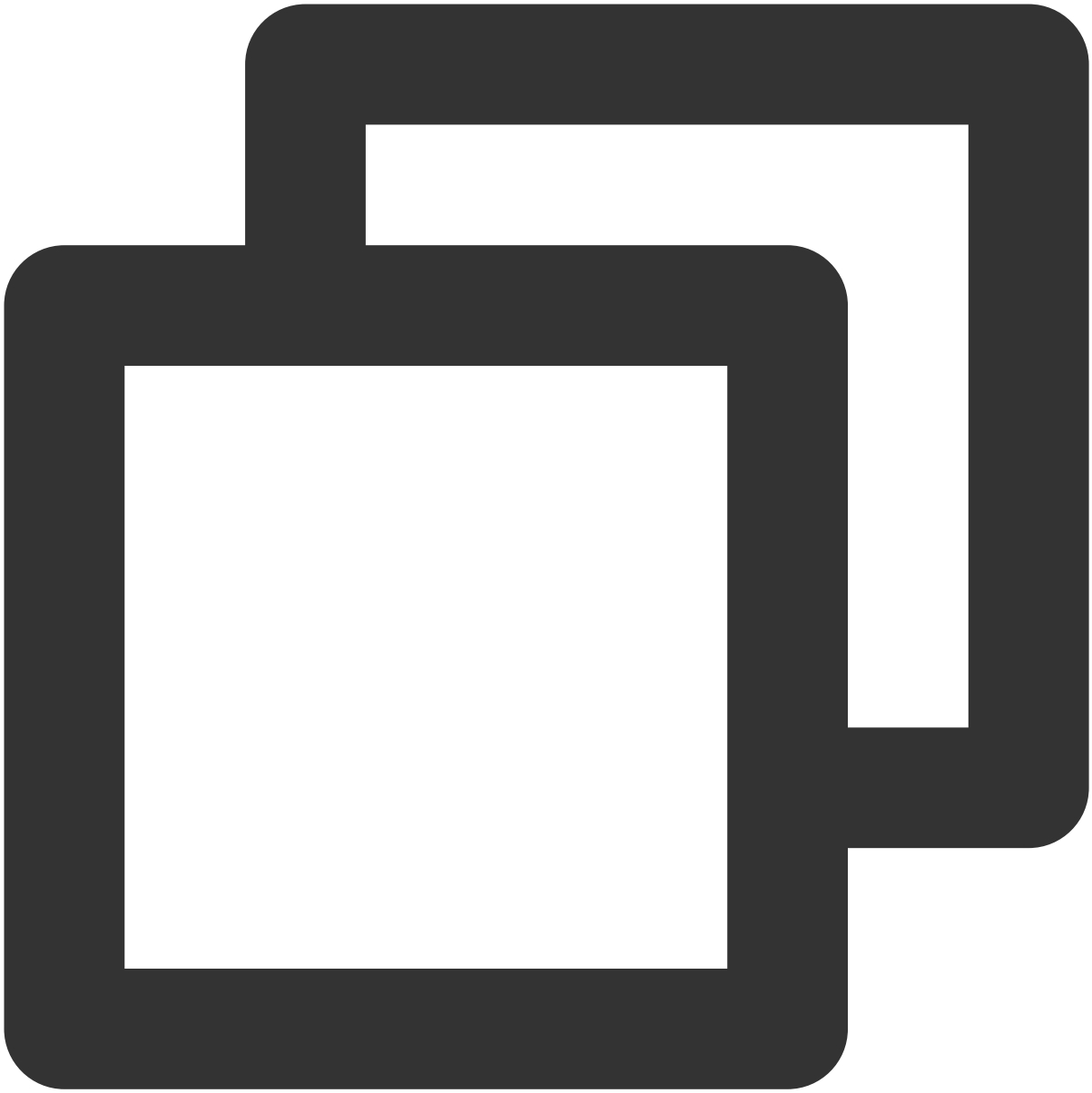
最近更新时间：2024-01-20 16:30:33

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 数组

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。

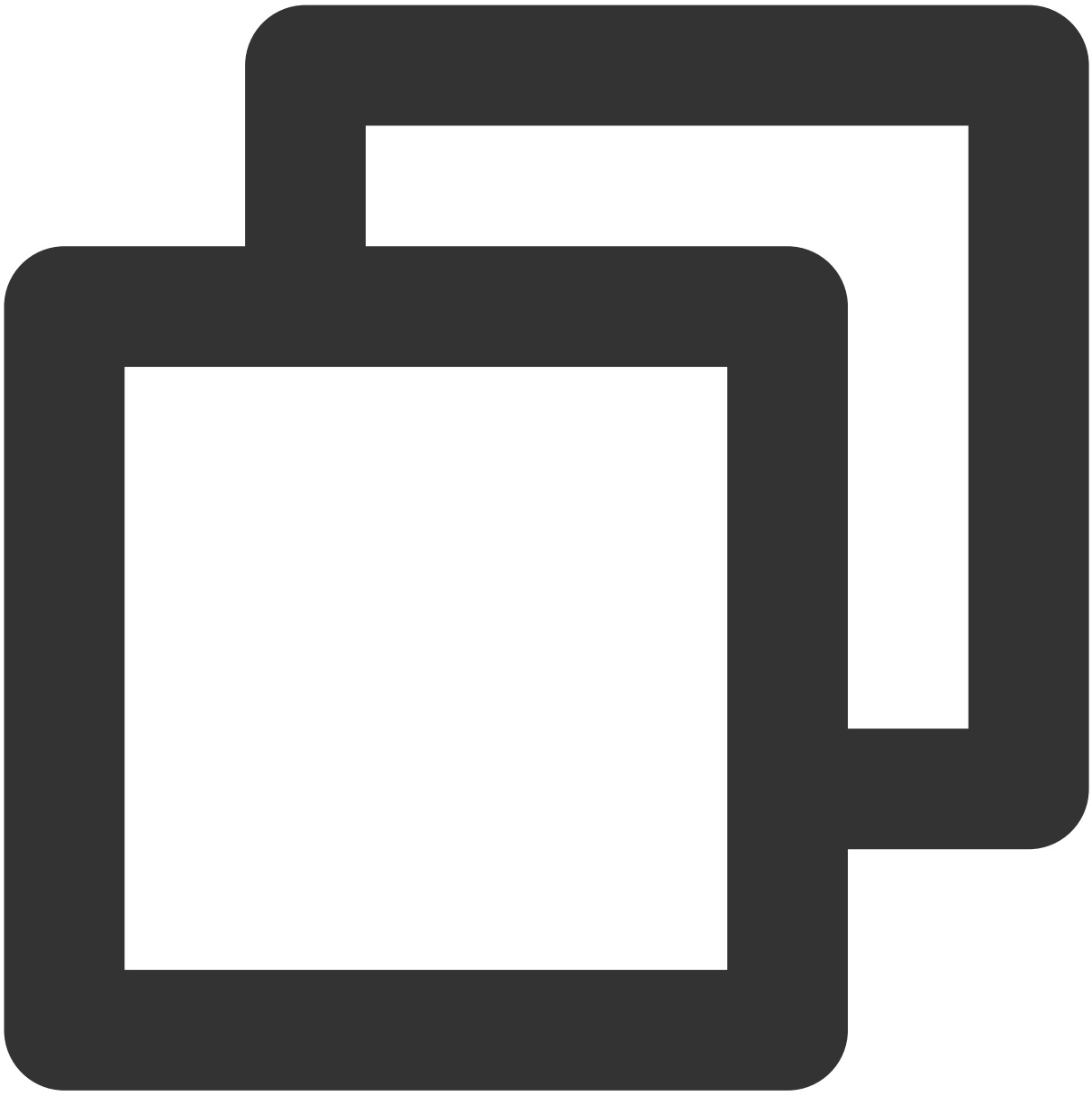
Container 节点 Workflow 的内容：

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



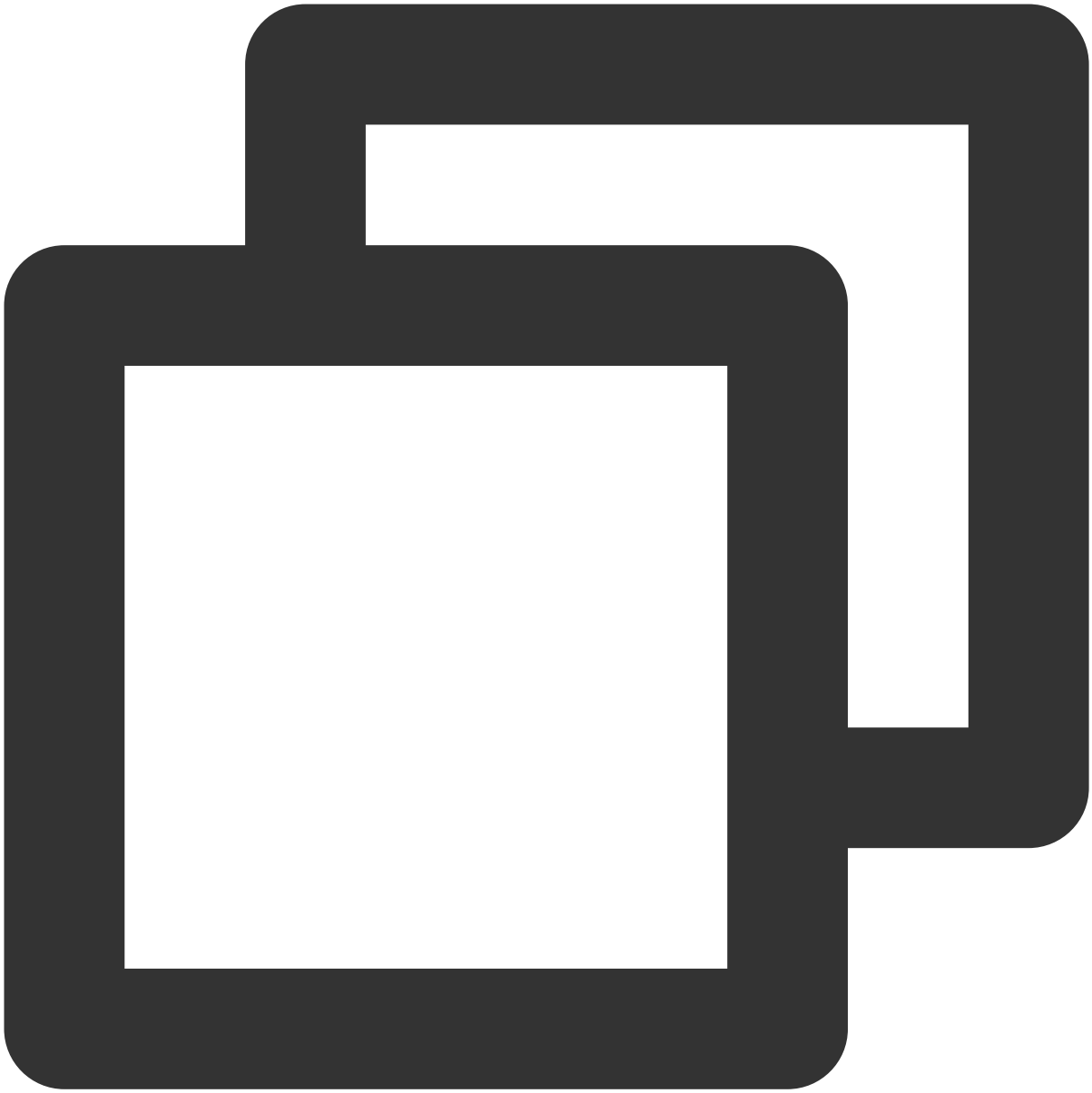
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:30:20+0800</CreationTime>
        <EndTime>2022-06-30T19:31:56+0800</EndTime>
```

```
        <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



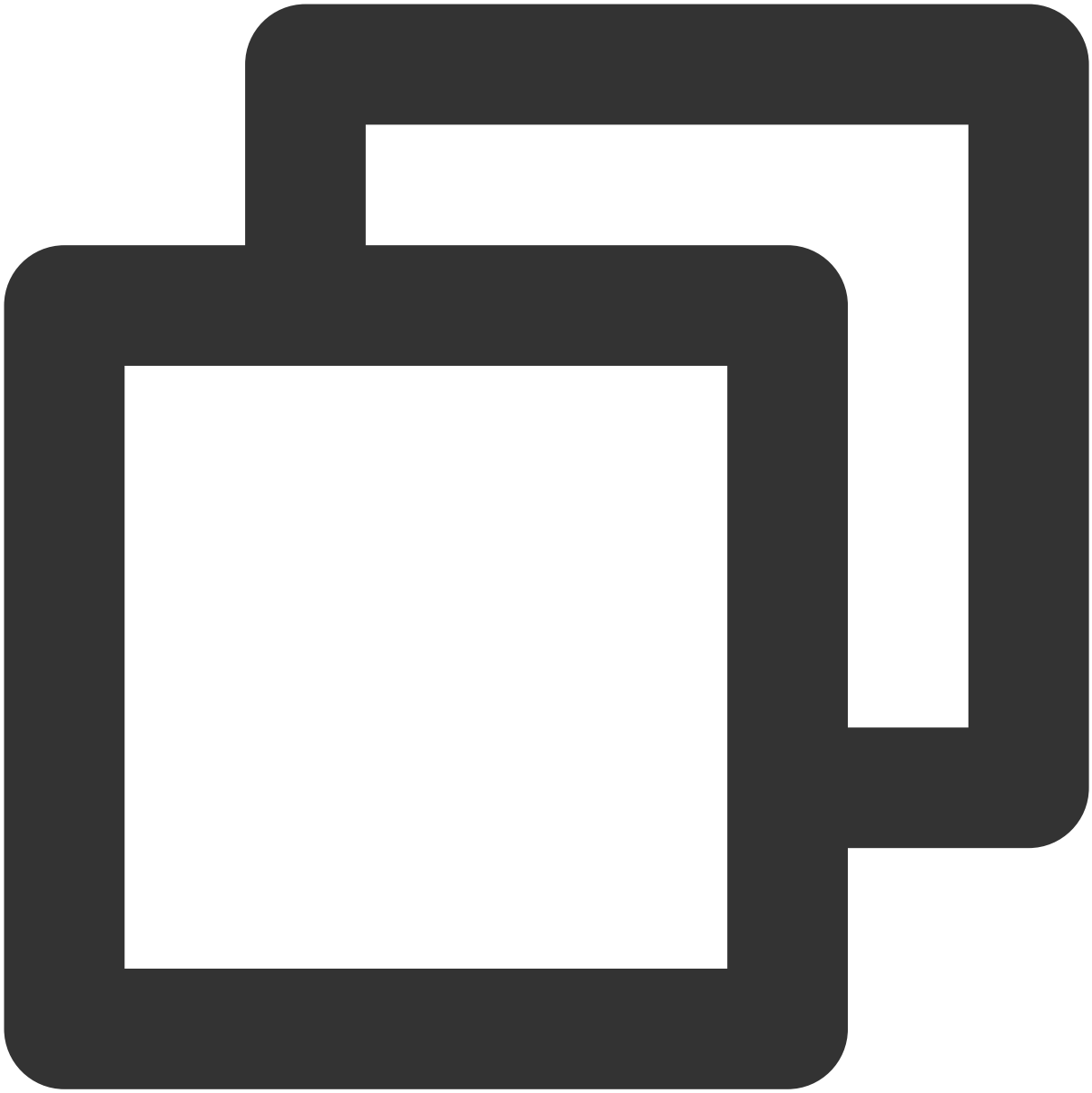
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:30:20+0800</CreationTime>
        <EndTime>2022-06-30T19:31:56+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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

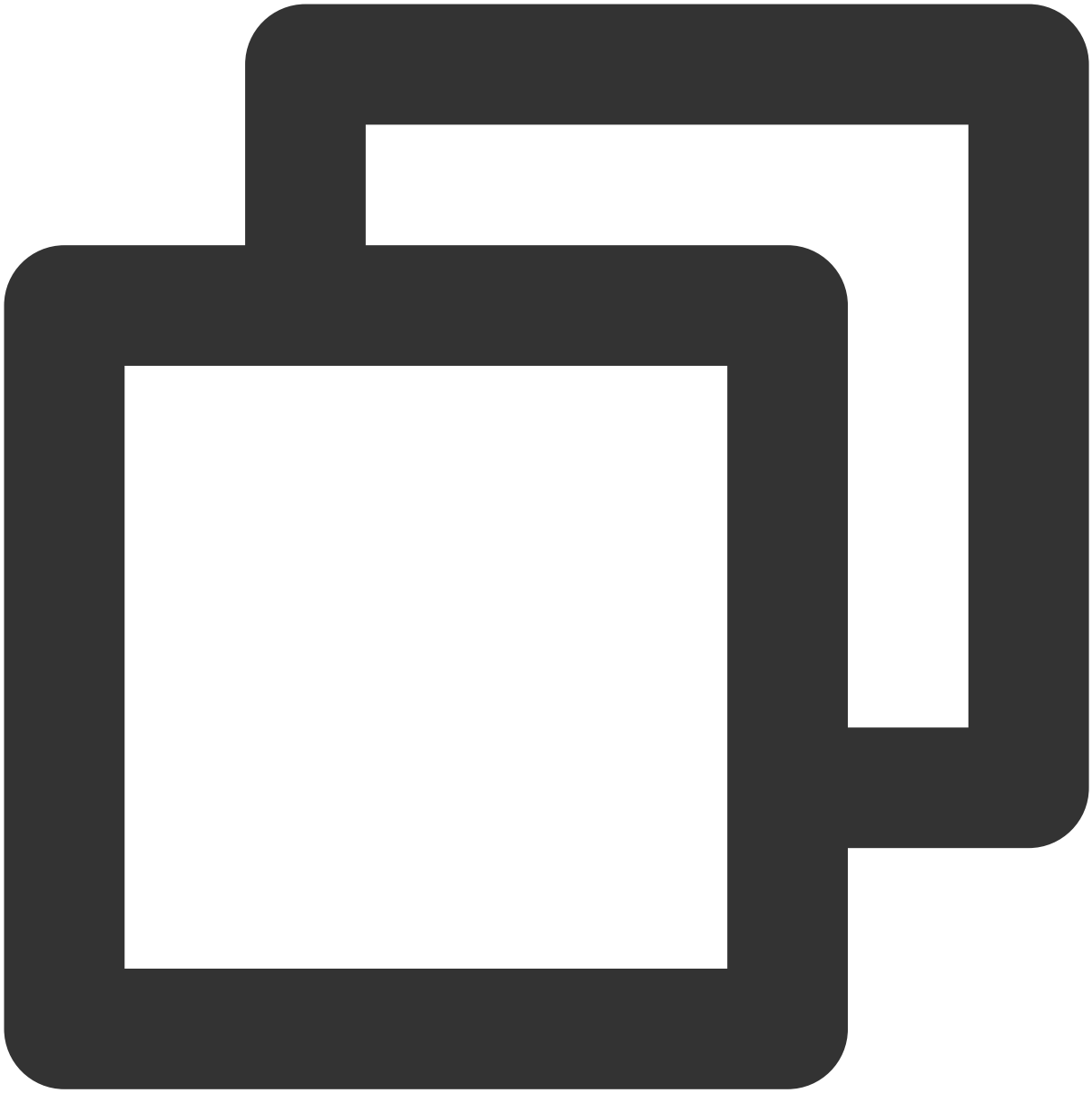
最近更新时间：2024-01-20 16:30:33

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>2022-06-30T15:30:08+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 数组

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

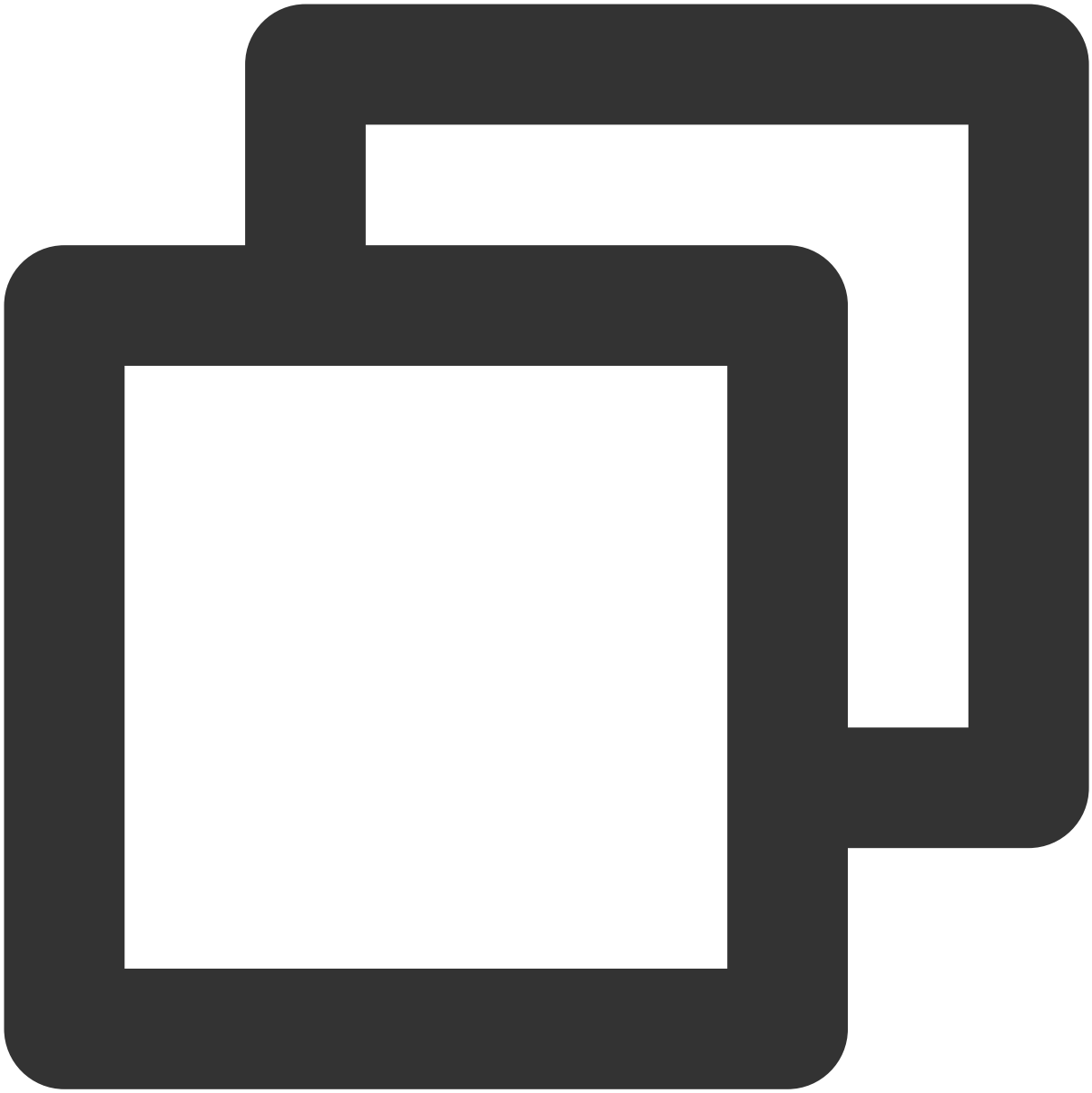
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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



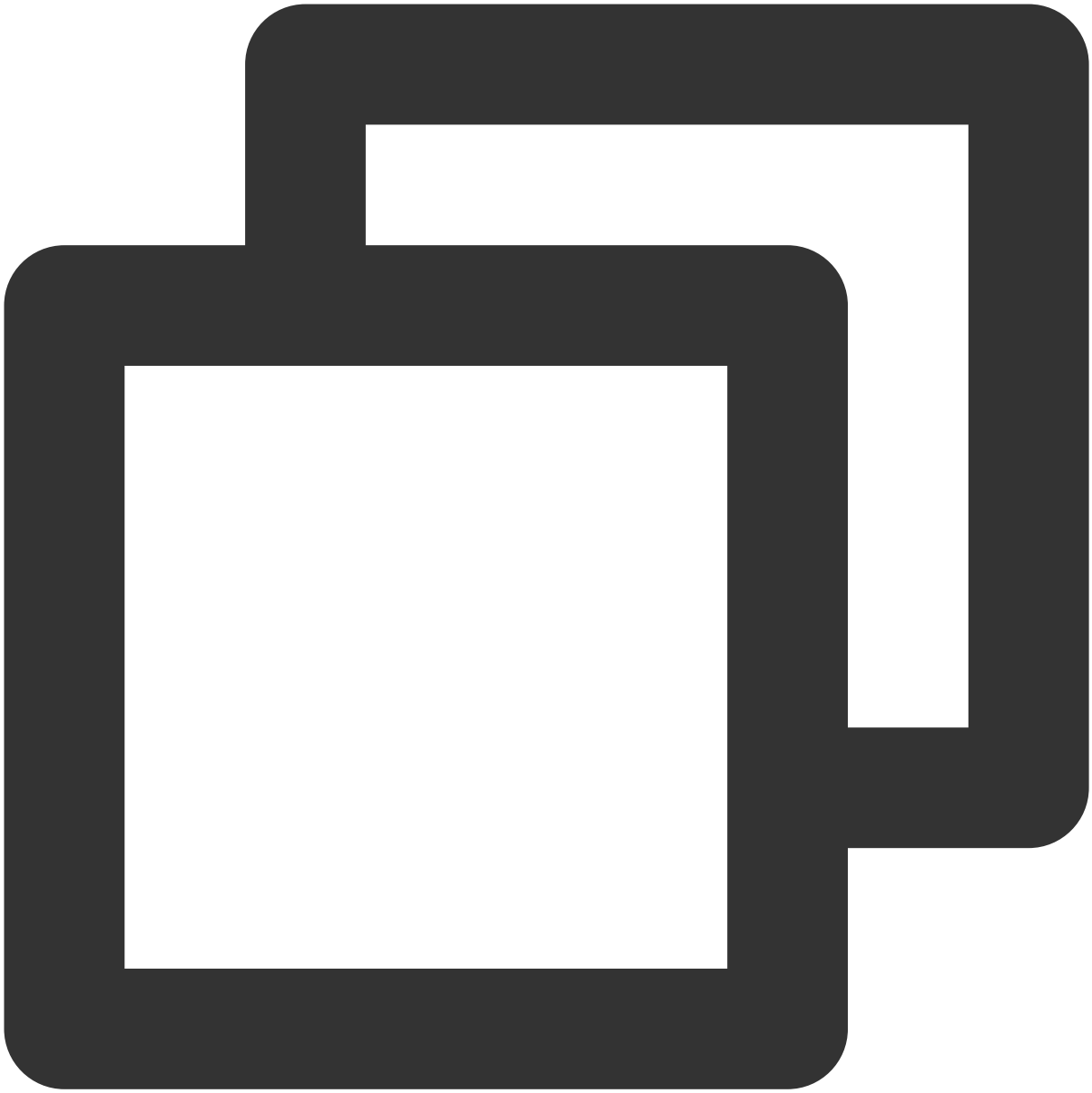
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:29:48+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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



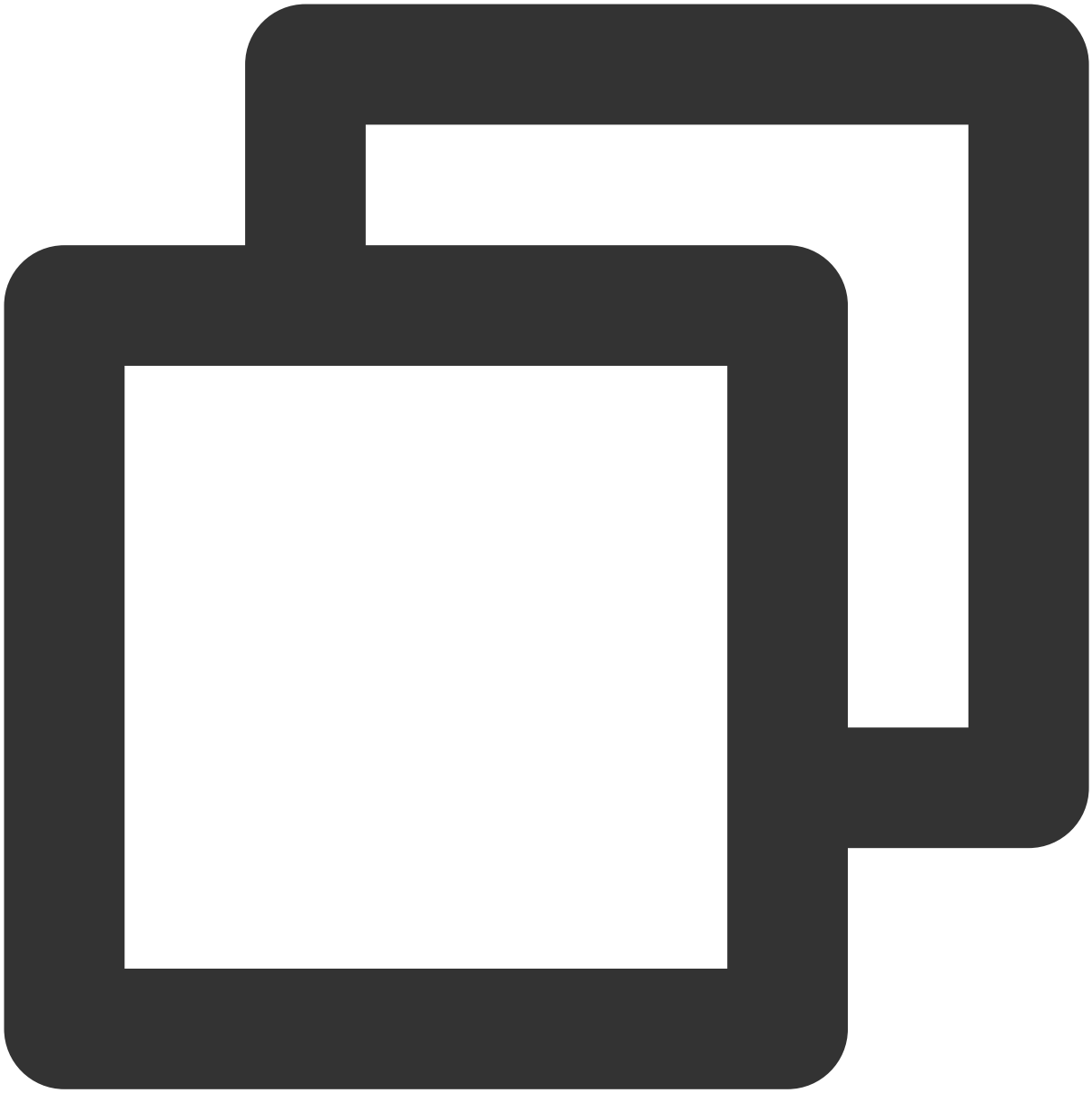
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:30:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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

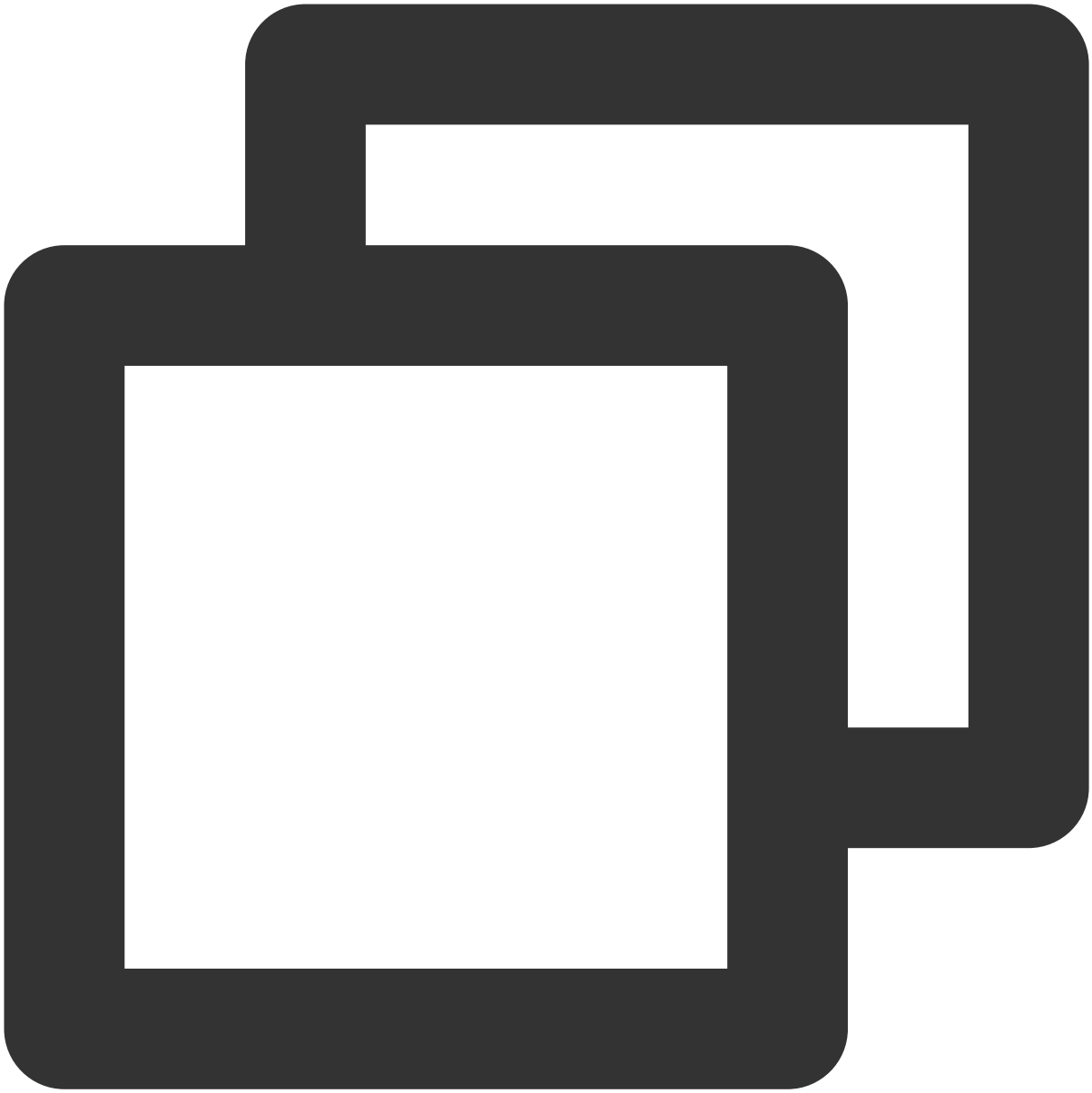
最近更新时间：2024-01-20 16:30:33

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 数组

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

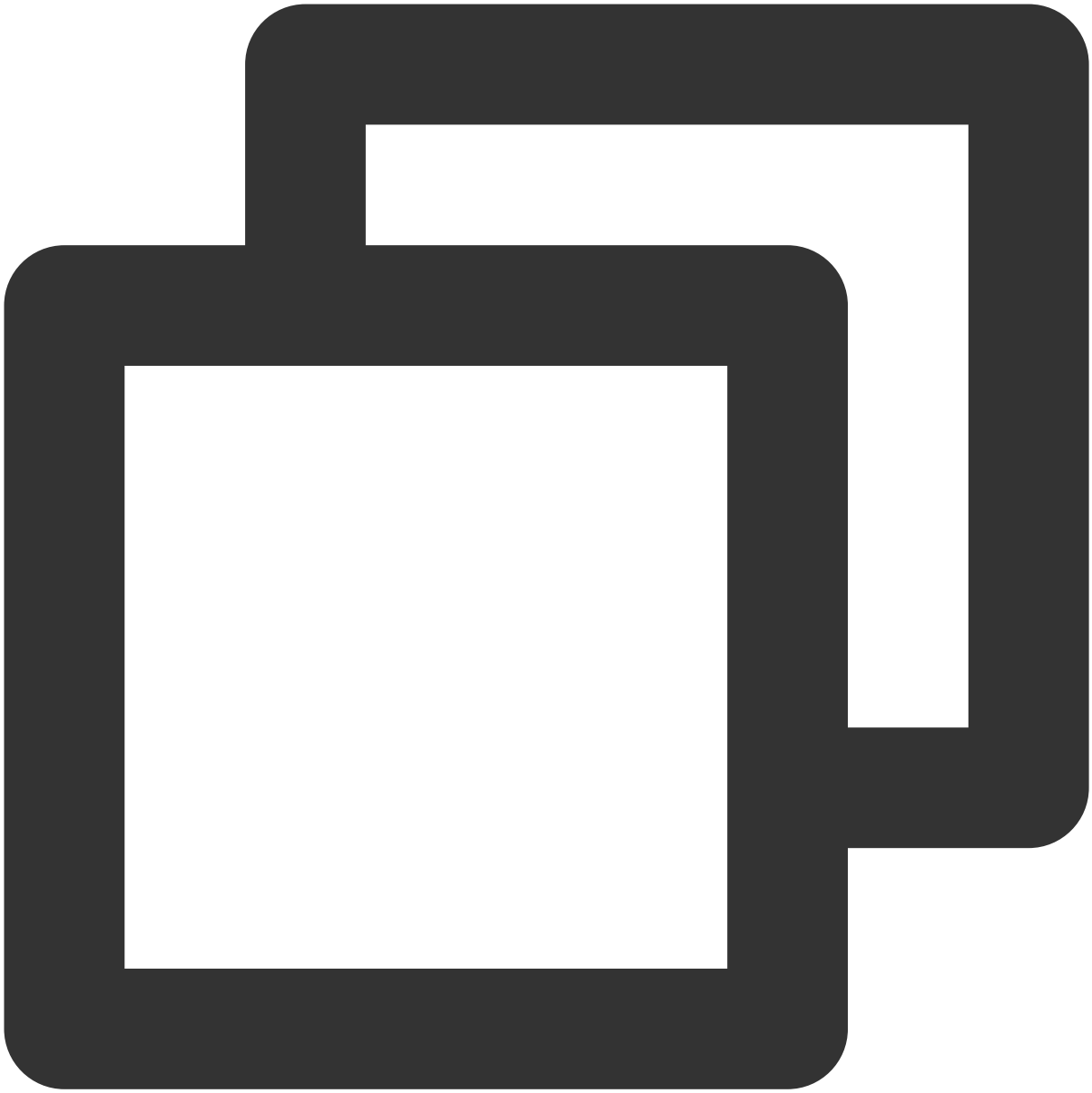
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

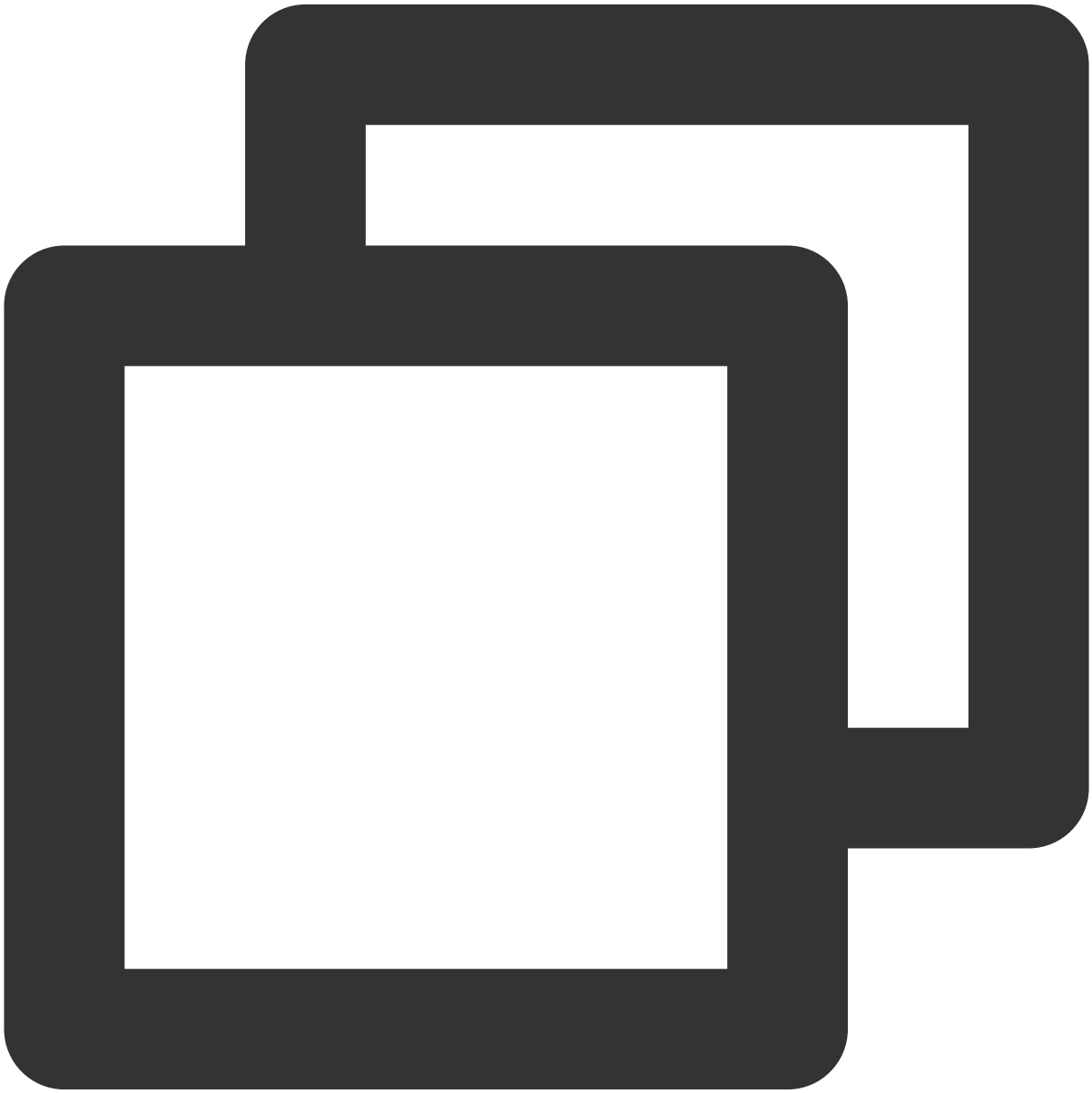
案例 1：通过任务接口触发的任务回调



```
<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

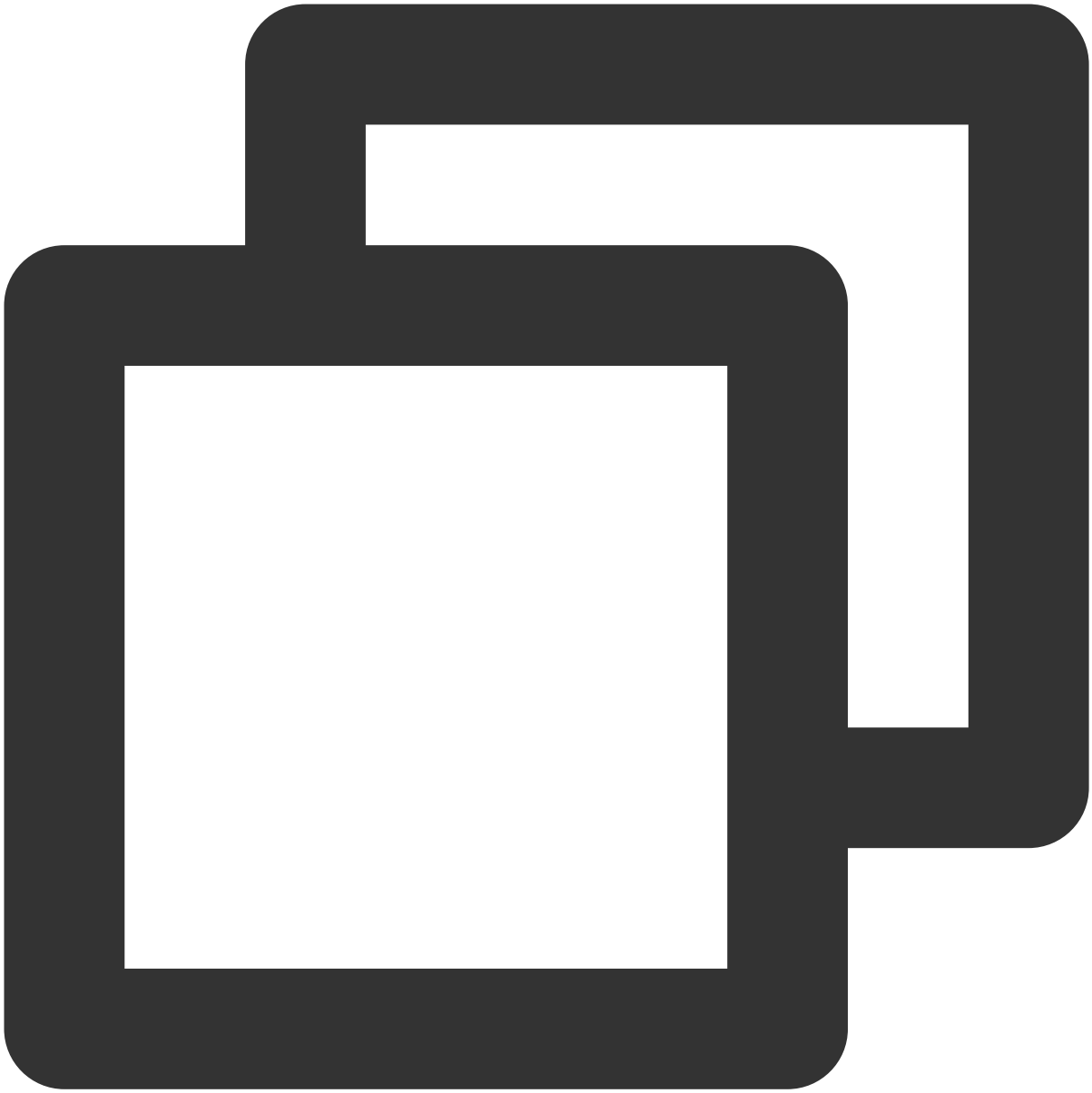


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:33:18+0800</CreationTime>
        <EndTime>2022-06-30T19:33:24+0800</EndTime>
        <JobId>j707f4864f86811ecb90d0b03267ce0e5</JobId>
        <Message/>
        <Operation>
            <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

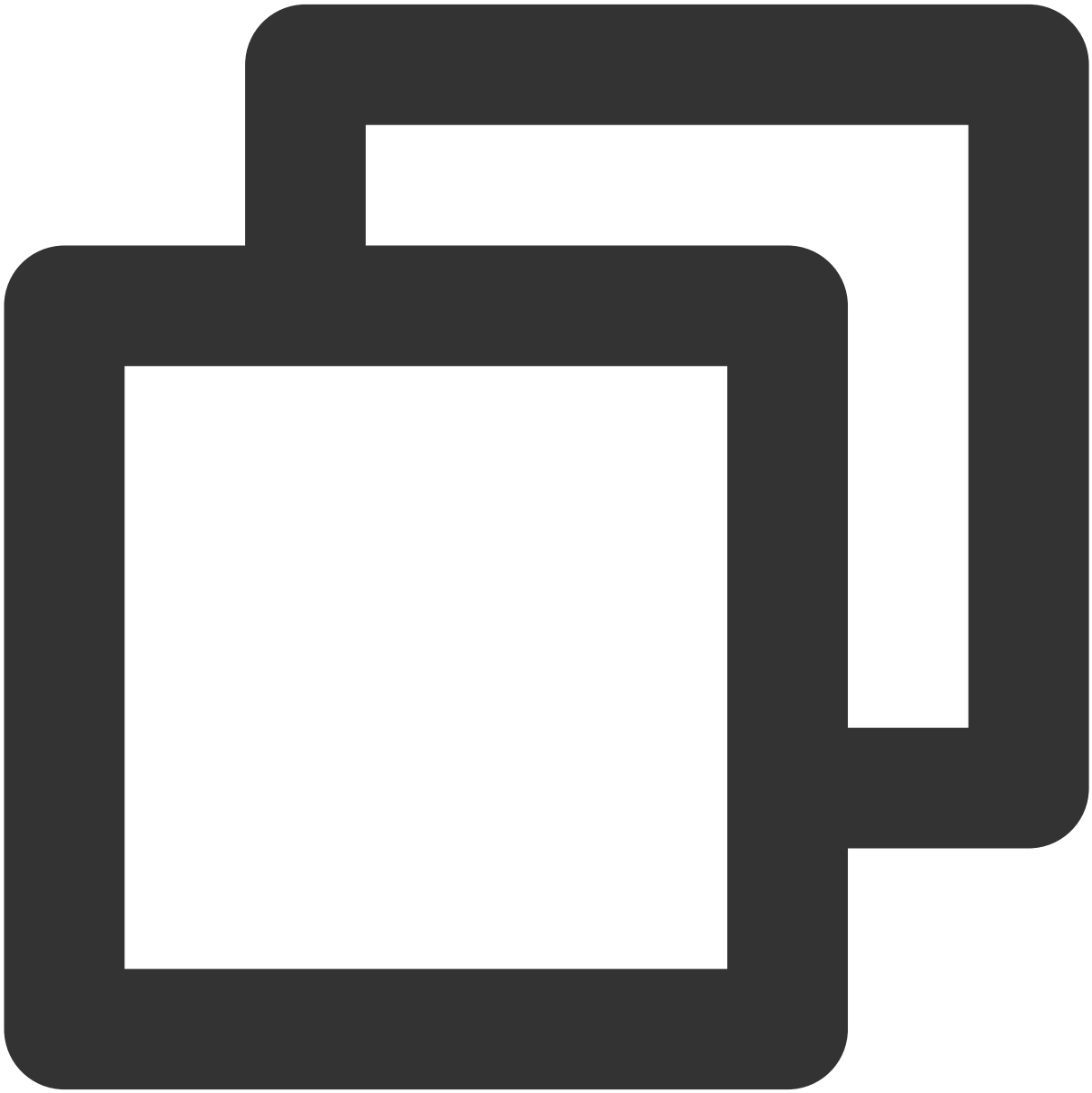
最近更新时间：2024-01-20 16:30:33

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>
```


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>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 数组

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

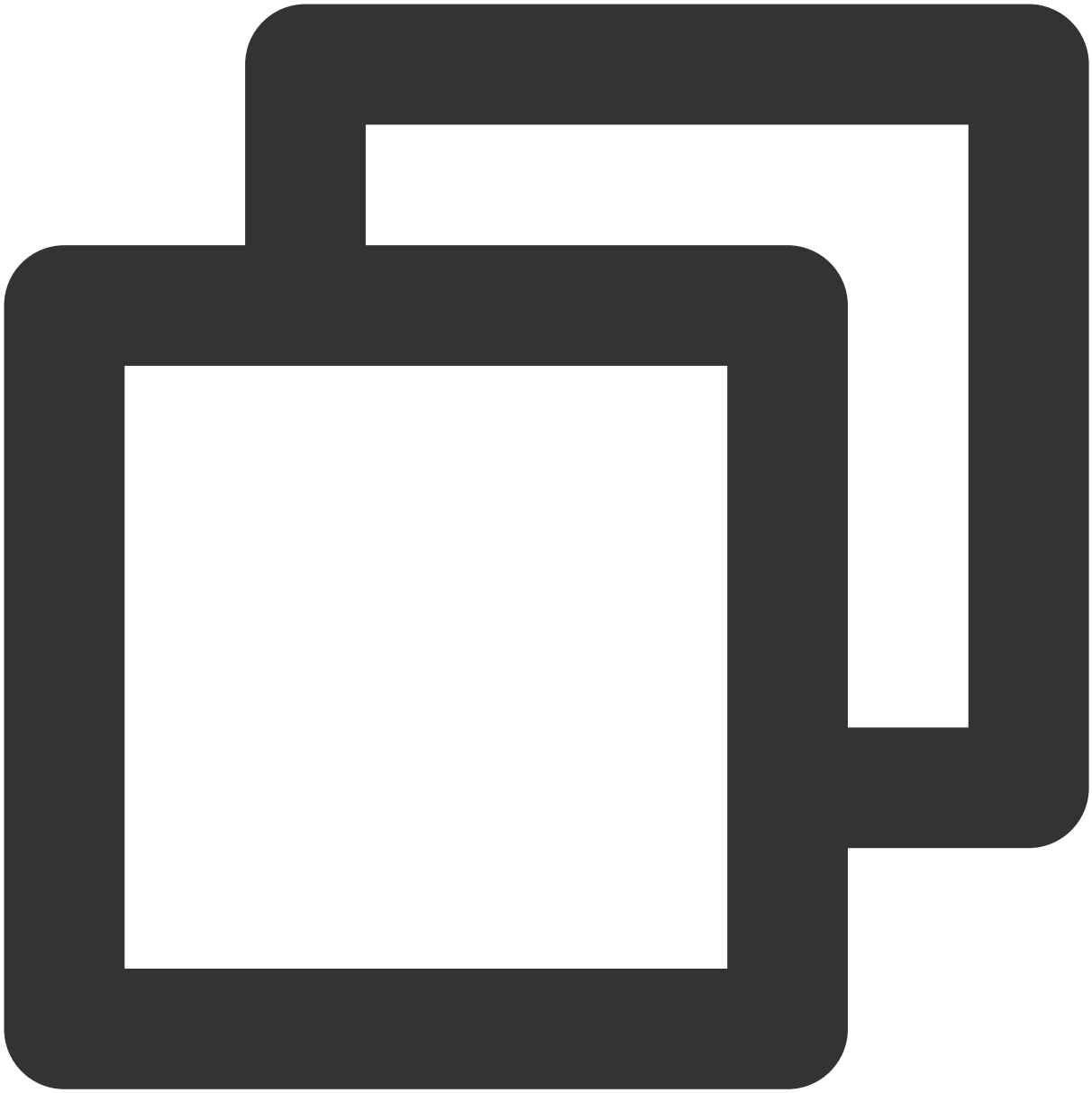
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

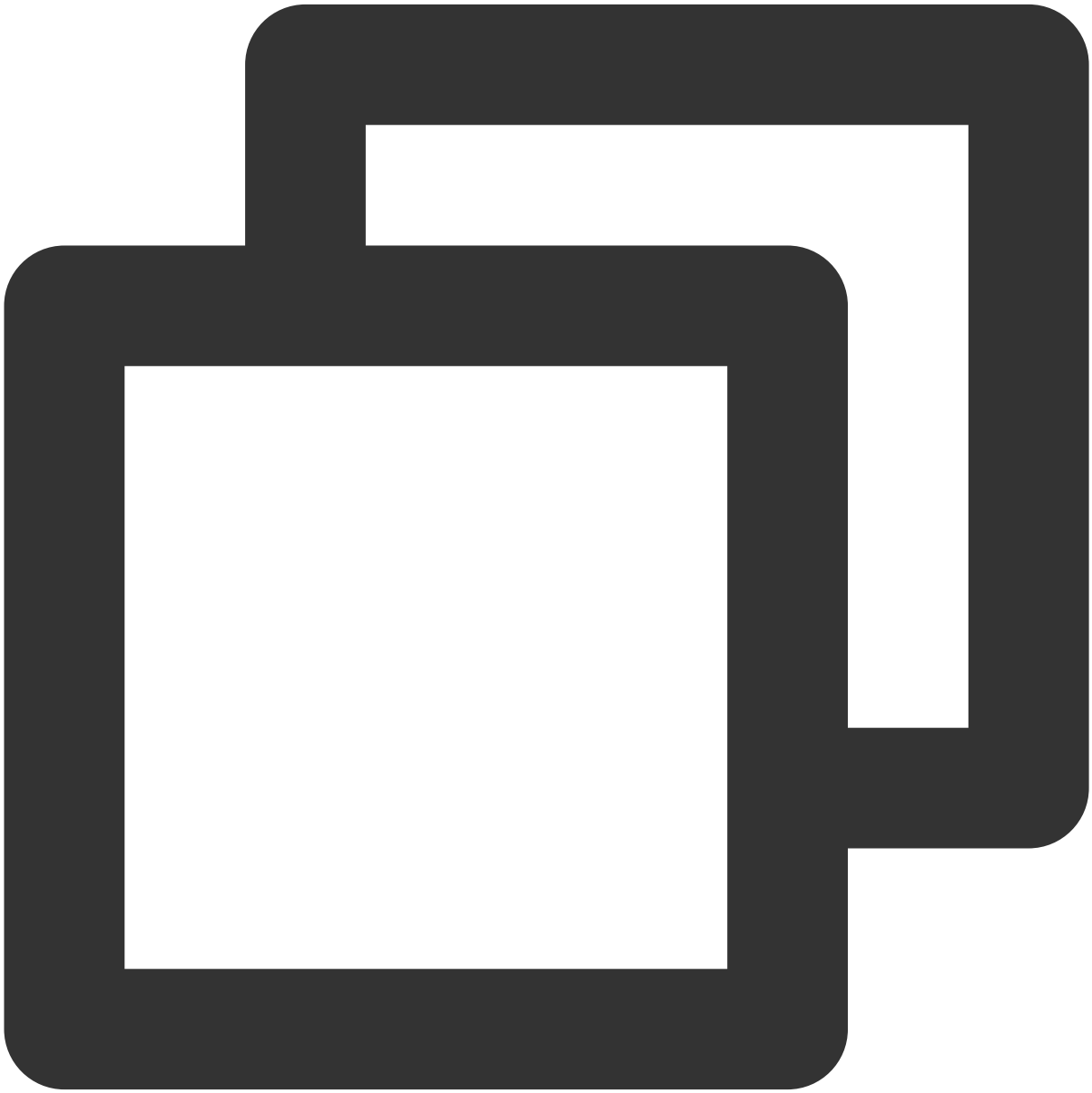


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T19:55:05+0800</CreationTime>
        <EndTime>2022-06-30T19:55:19+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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>
```

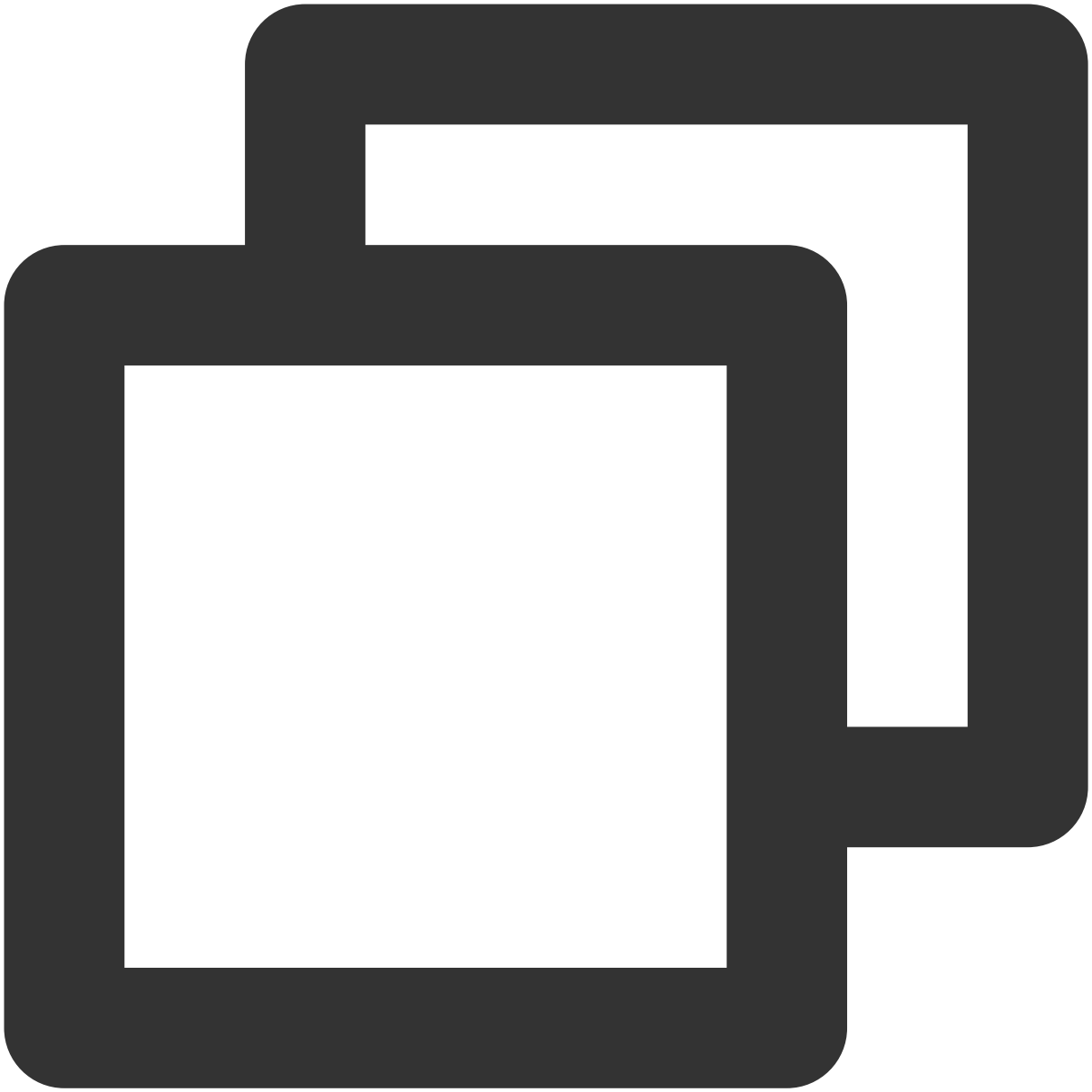
```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
                </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

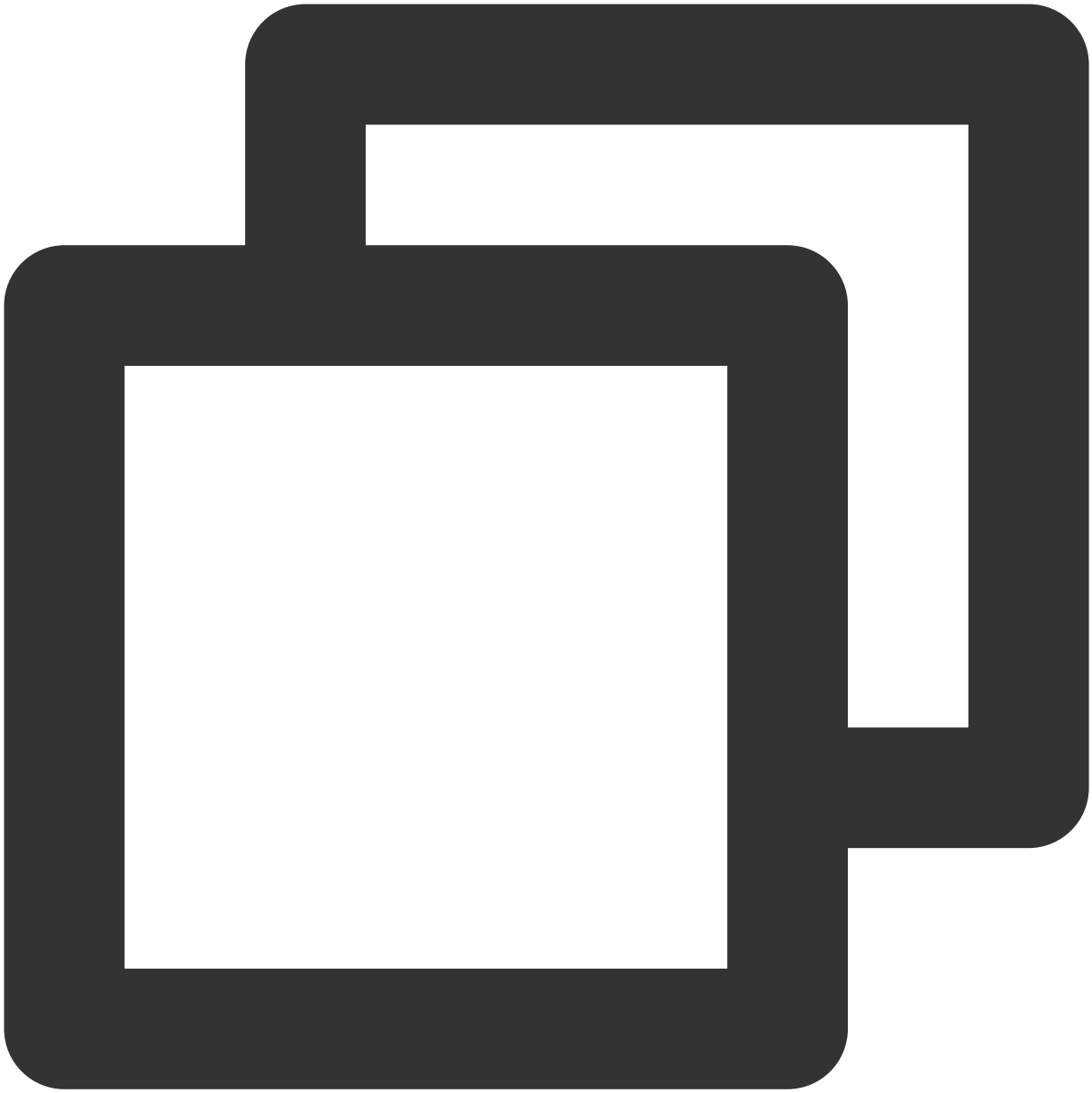
最近更新时间：2024-01-20 16:30:33

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>2022-06-30T20:02:17+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 数组

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

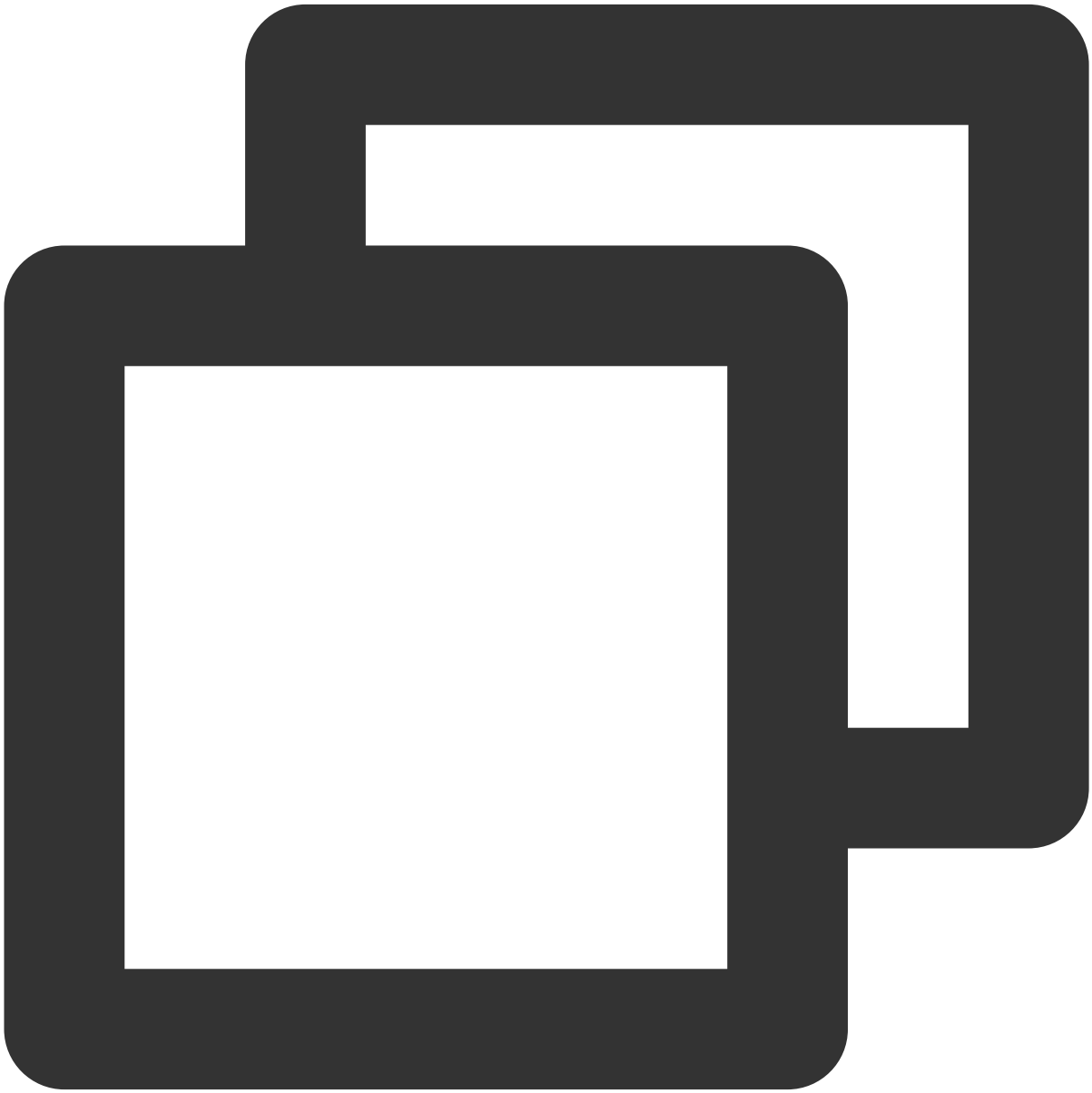
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

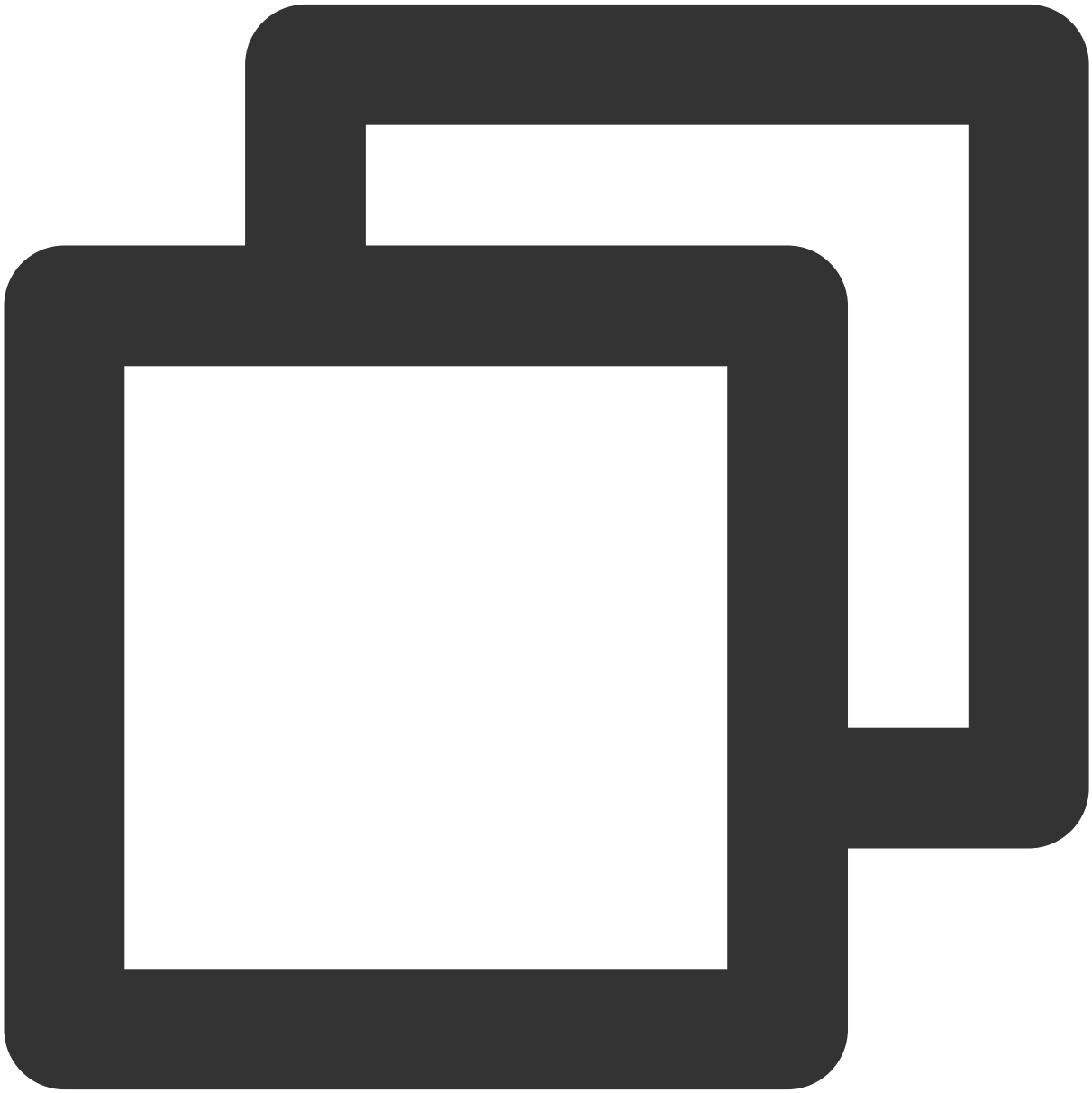


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T20:02:17+0800</CreationTime>
        <EndTime>2022-06-30T20:02:34+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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>
```

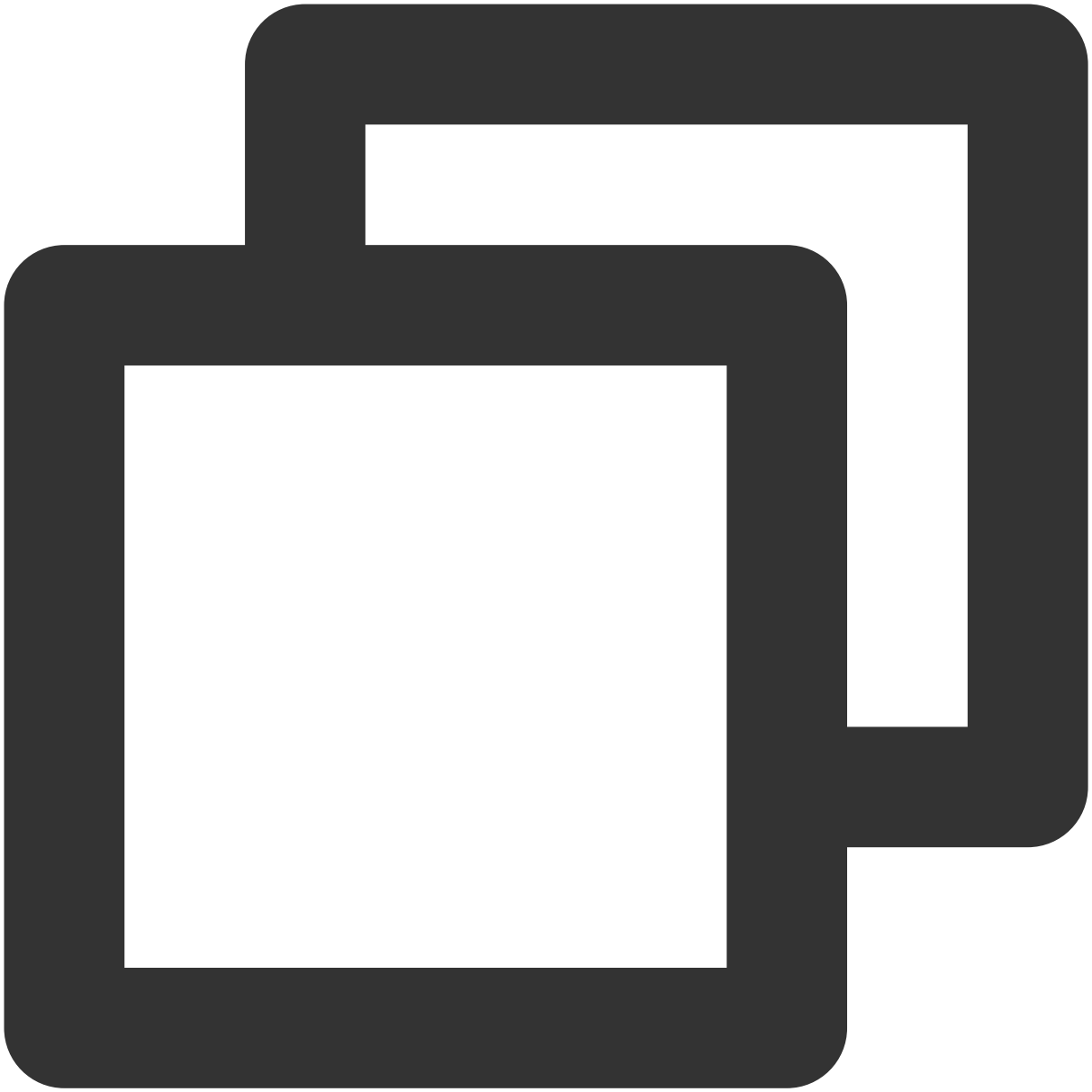
```
                <Key>Content-Type</Key>
                <Value>video/mp4</Value>
            </CosHeaders>
            <CosHeaders>
                <Key>x-cos-request-id</Key>
                <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
            </CosHeaders>
            <CosHeaders>
                <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
                </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

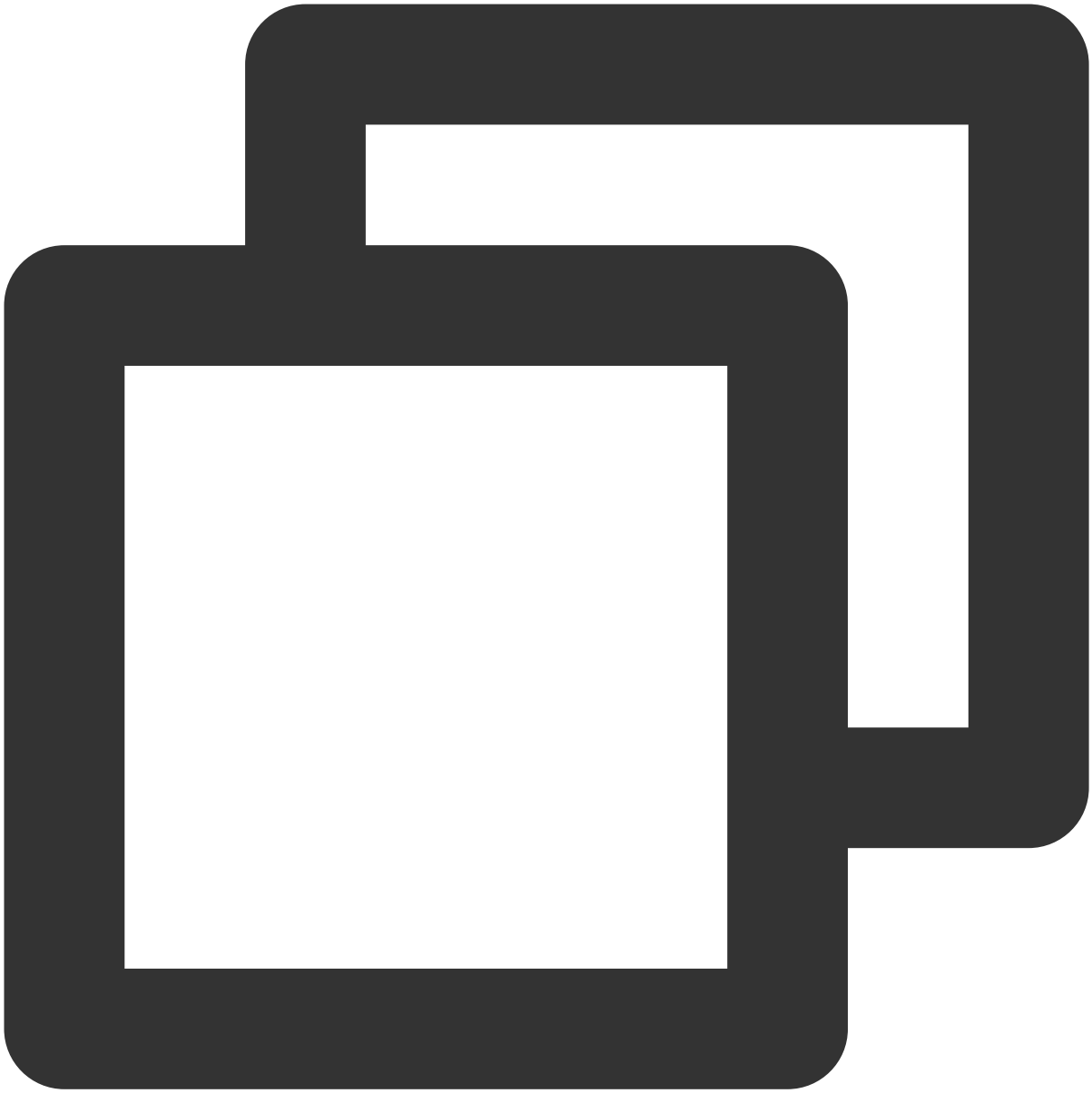
最近更新时间：2024-01-20 16:30:33

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 数组

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

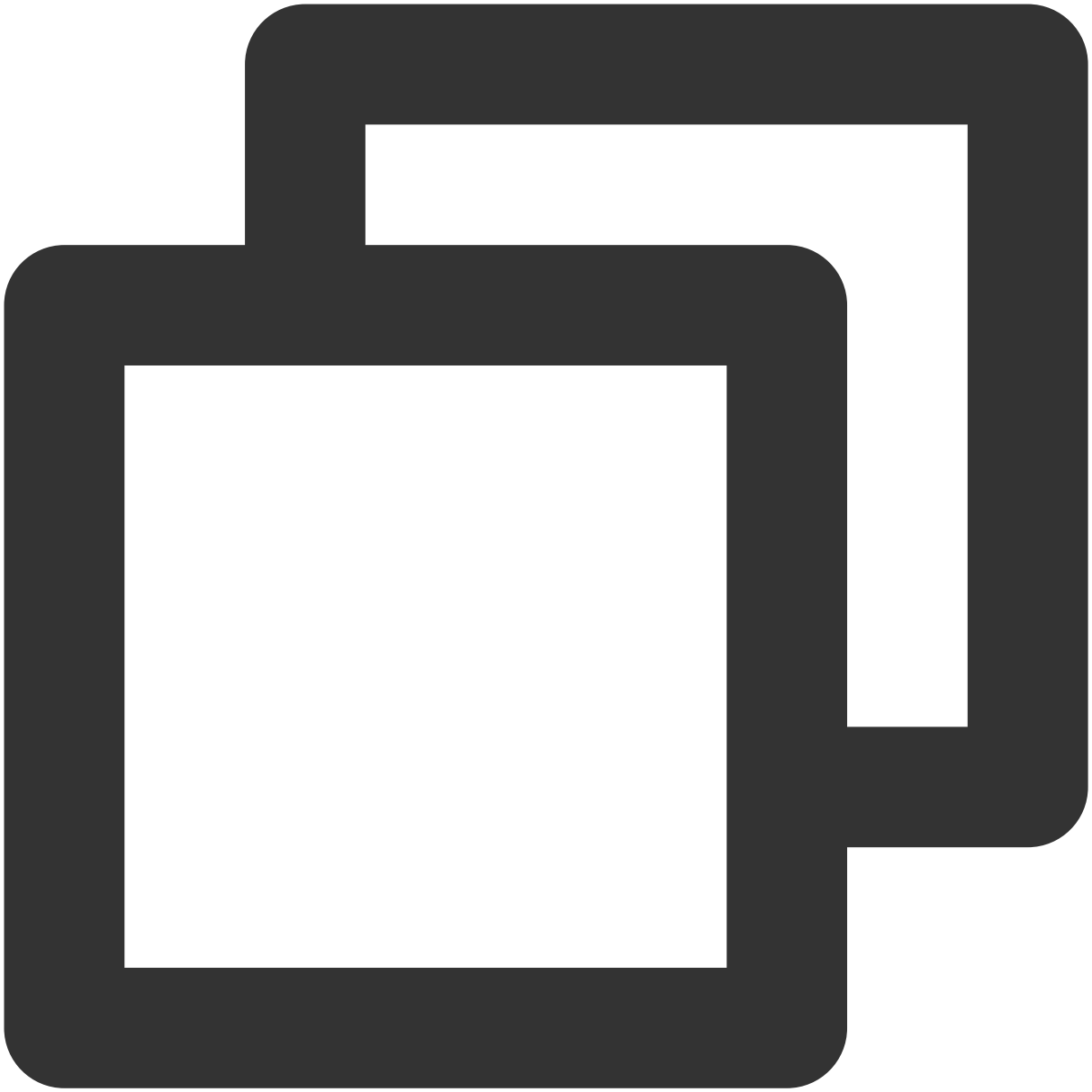
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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>
```

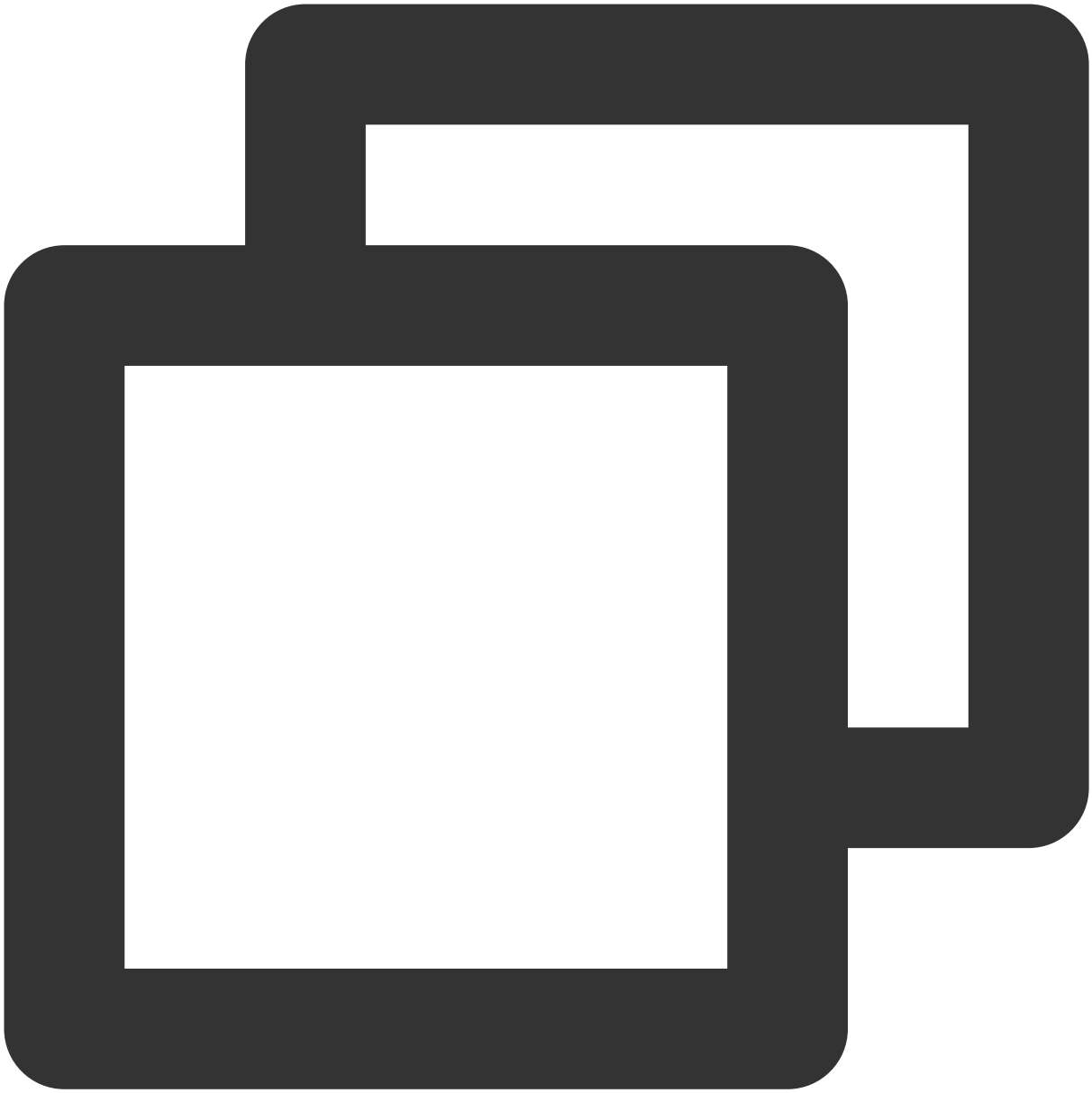
```
        <JobId>j535b10a4f8e211ec9452c1c27f56a659</JobId>
        <Message>Success</Message>
        <Operation>
            <UserData>This is my VideoTag job.</UserData>
            <VideoTag>
                <Scenario>Stream</Scenario>
            </VideoTag>
            <VideoTagResult>
                <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"
        },
```

```
        "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

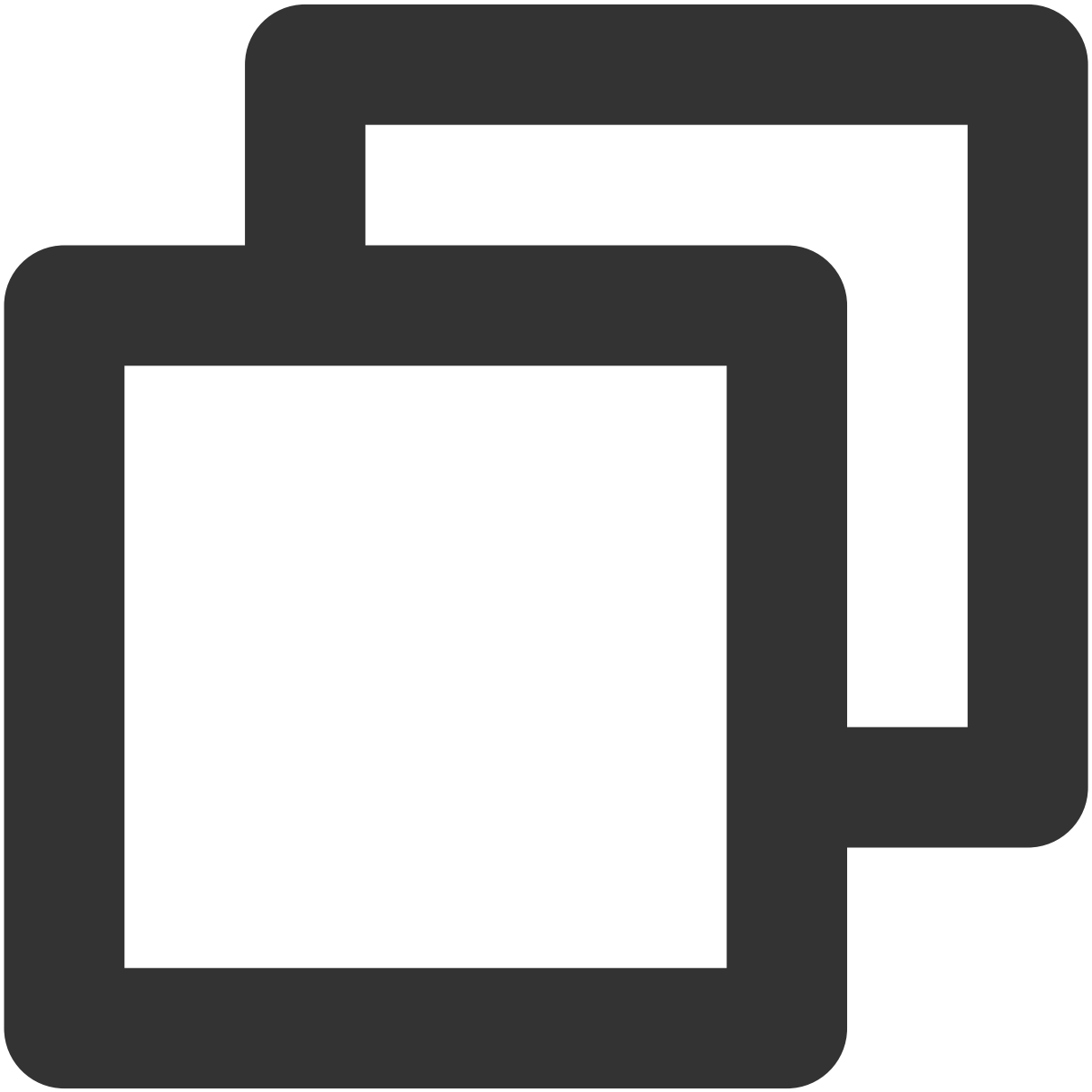
最近更新时间：2024-01-20 16:30:33

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>2022-07-01T10:12:26+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 数组

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

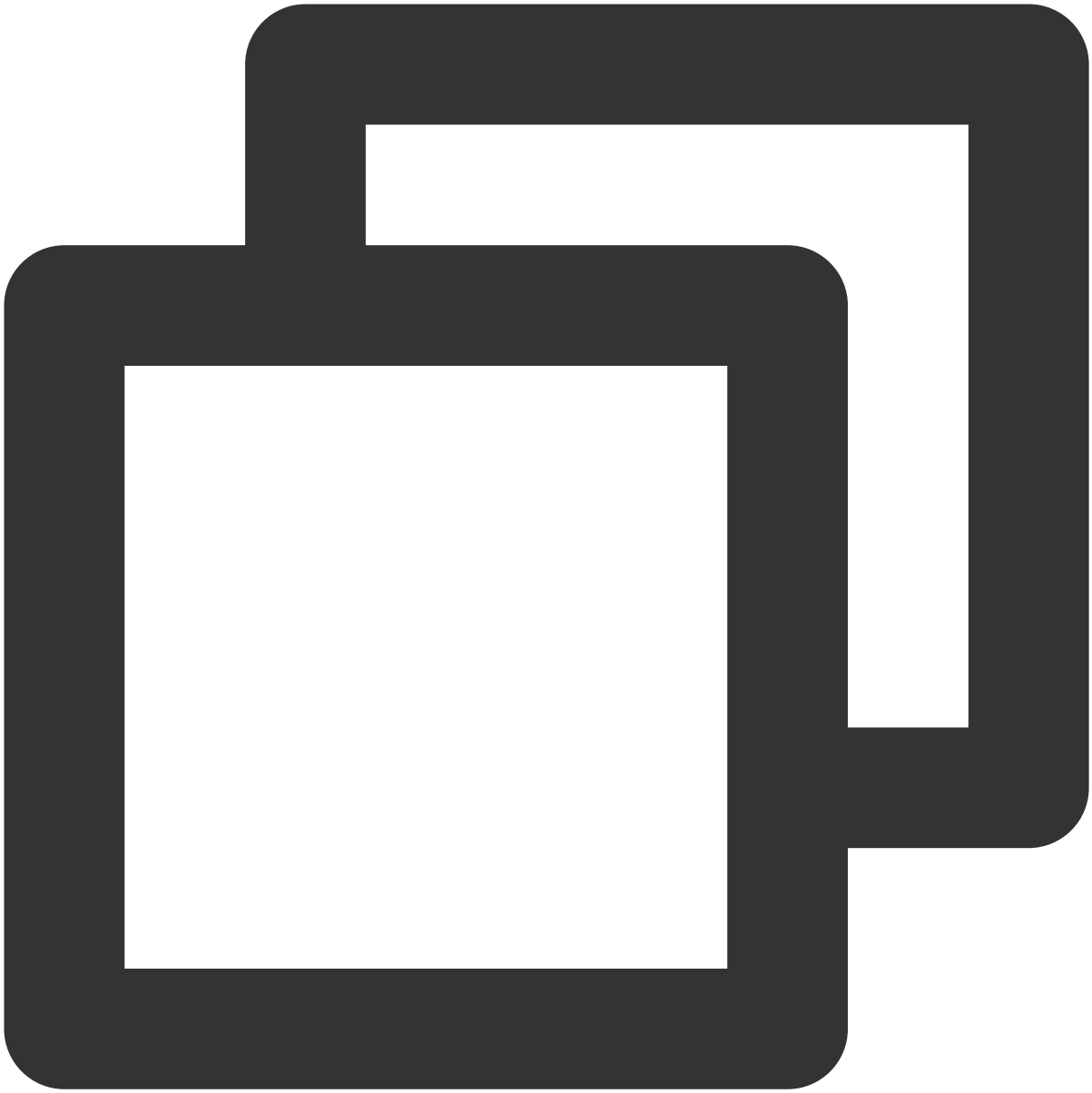
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

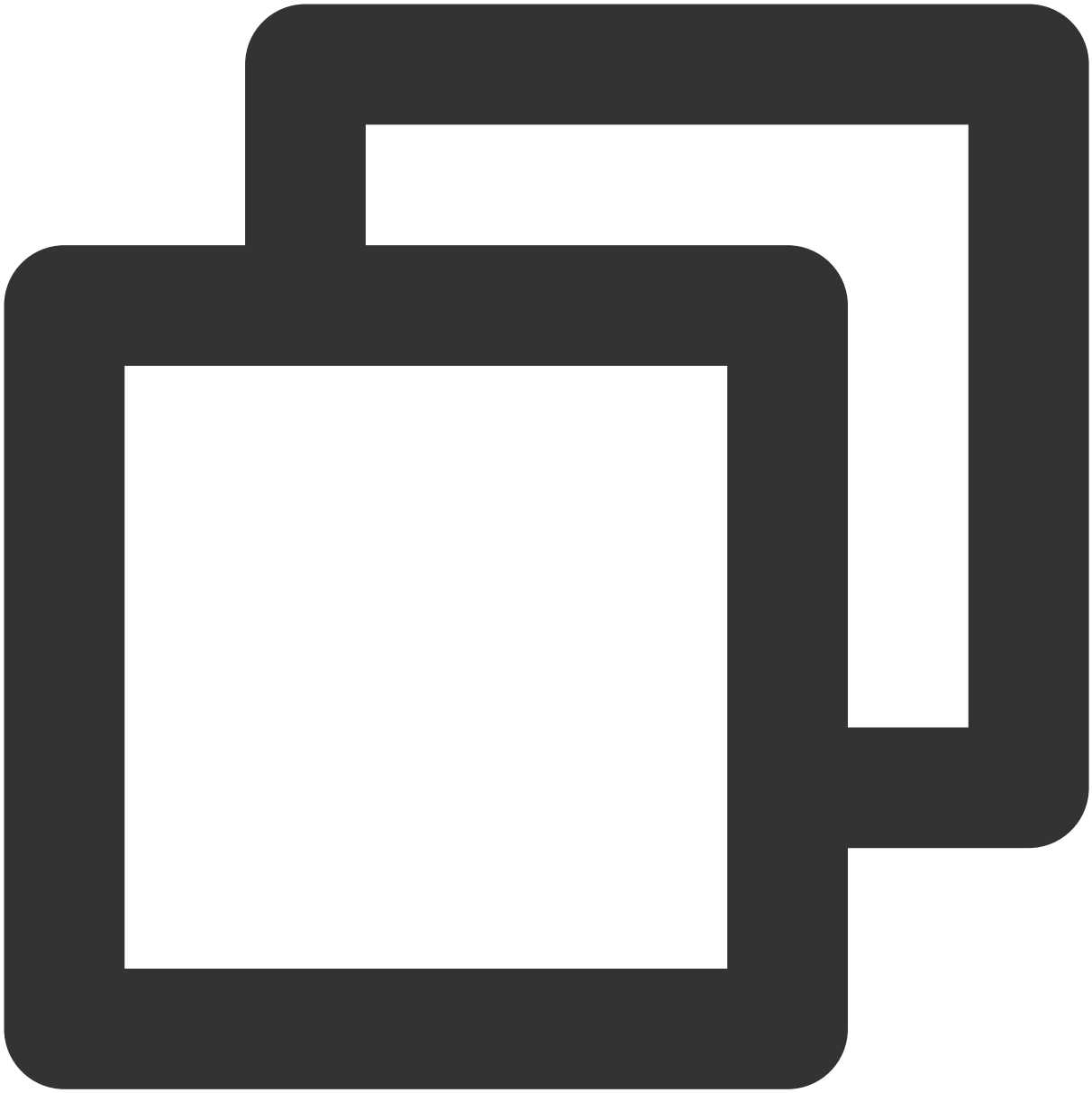


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:12:26+0800</CreationTime>
        <EndTime>2022-07-01T10:12:32+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</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



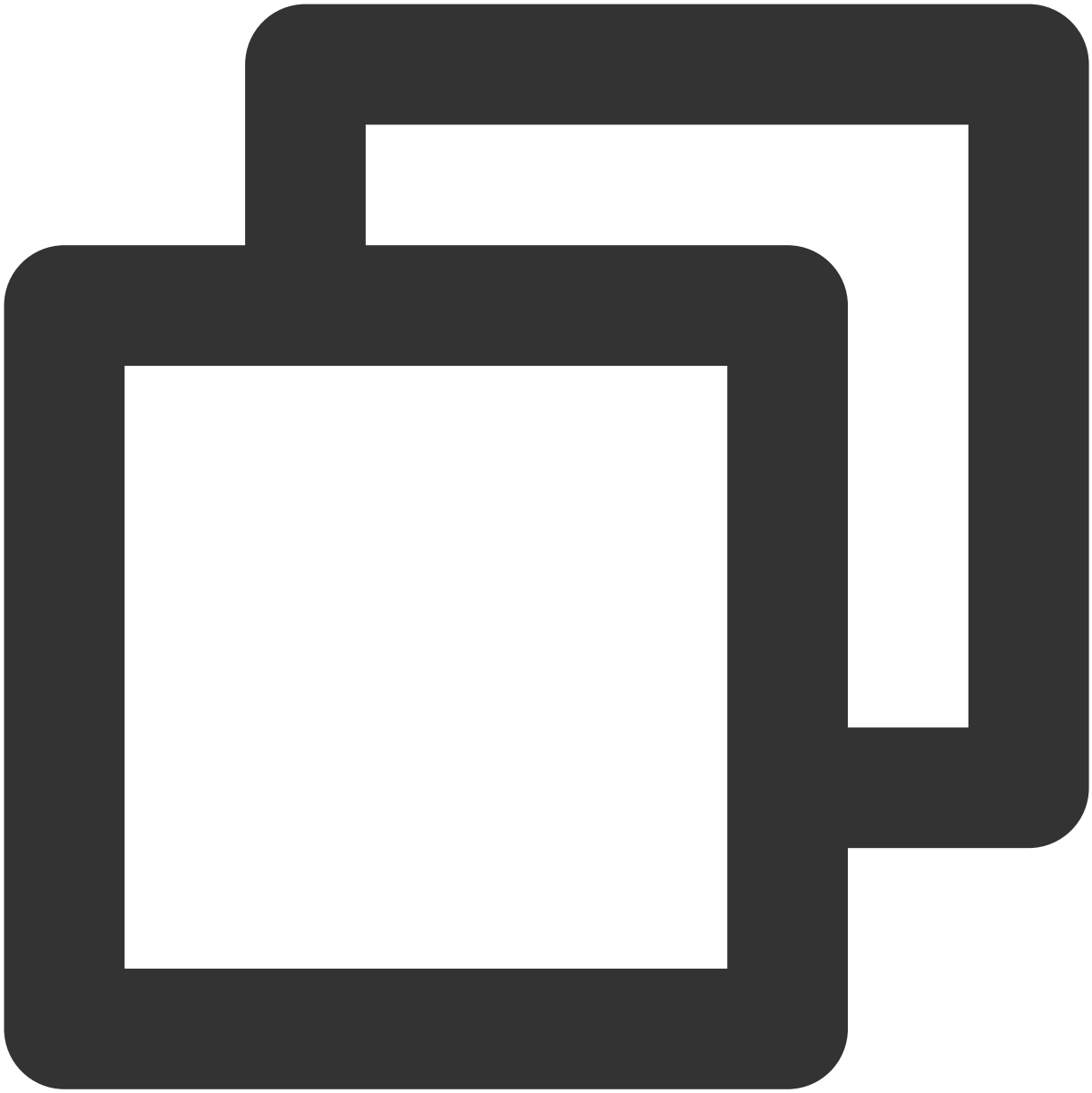
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-01T10:12:26+0800</CreationTime>
        <EndTime>2022-07-01T10:12:32+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp3</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"
        }
    }]
}
```

极速高清转码

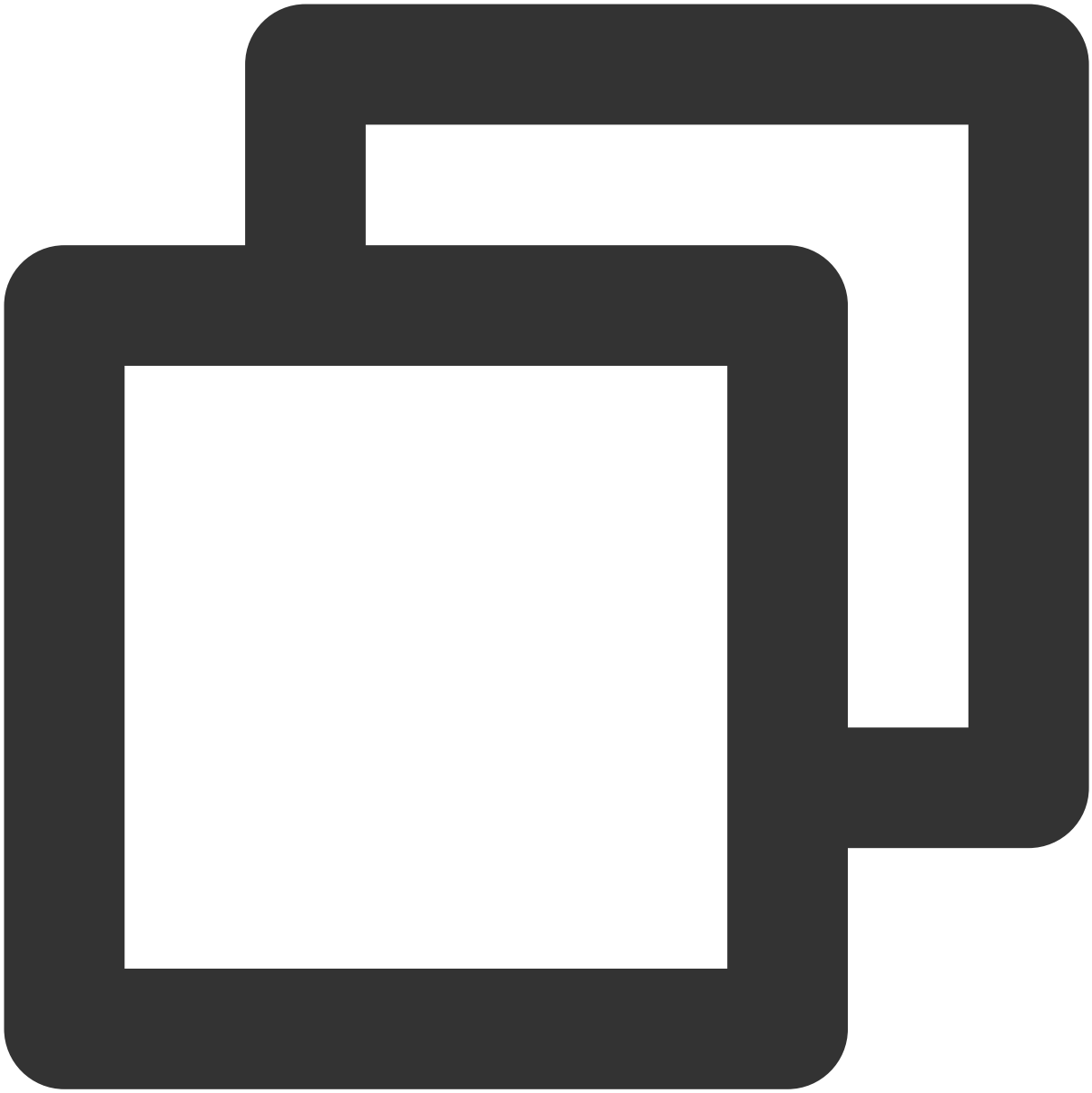
最近更新时间：2024-01-20 16:30:33

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>2022-06-30T15:30:08+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/high_speed.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/high_speed.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/high_speed.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
            <TemplateName>high_993874</TemplateName>
            <UserData>This is my HighSpeedHd job.</UserData>
            <JobLevel>0</JobLevel>
        </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 数组

Container 节点 JobsDetail 的内容：

同提交极速高清转码任务接口中的 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

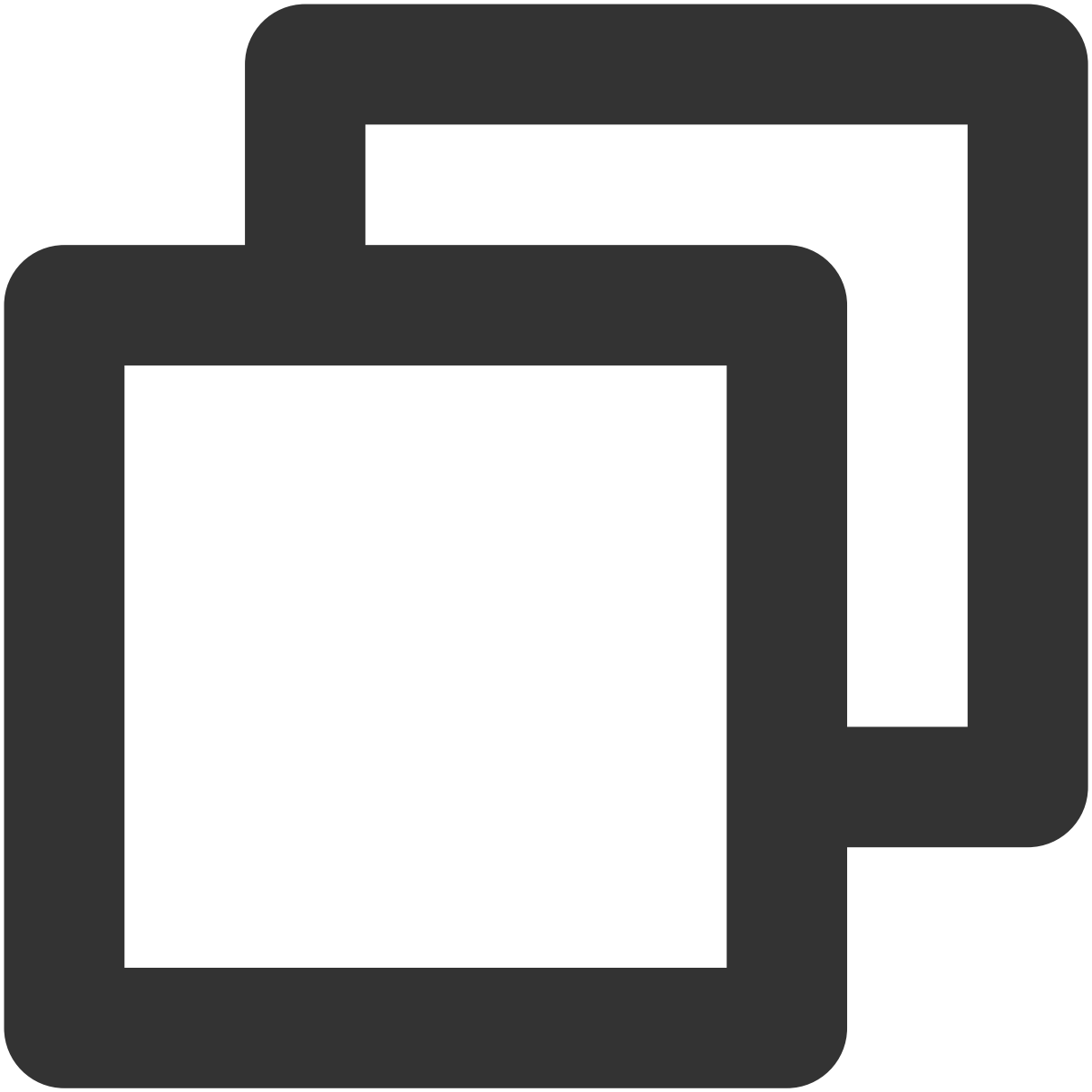
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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



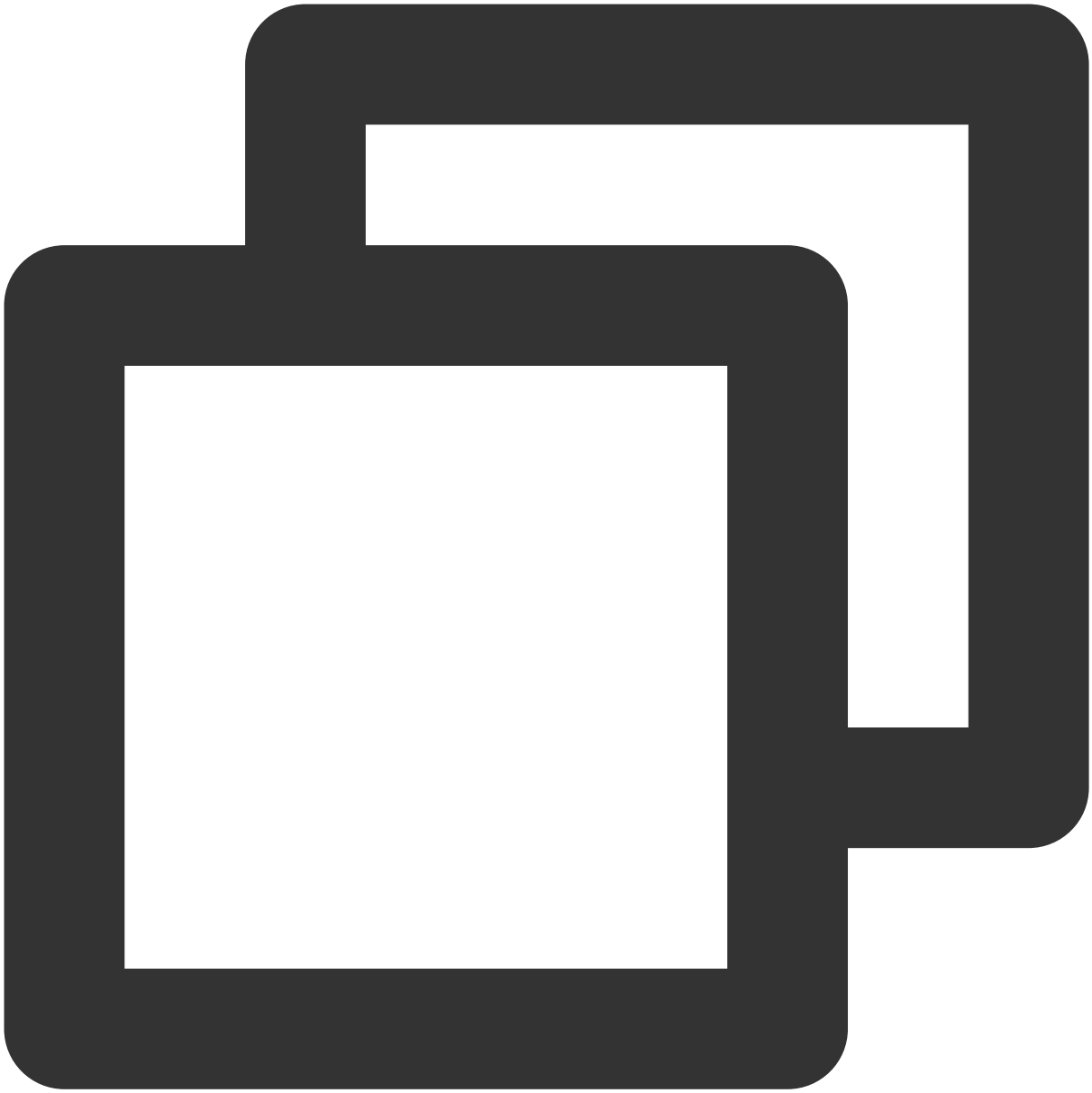
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:30:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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/high_speed.mp4</ObjectName>
                    </Md5Info>
                    <ObjectName>output/high_speed.mp4</ObjectName>
                    <ObjectPrefix/>
                    <Region>ap-chongqing</Region>
                </OutputFile>
            </MediaResult>
            <Output>
                <Bucket>test-123456789</Bucket>
                <Object>output/high_speed.mp4</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
            <TemplateName>high_993874</TemplateName>
            <UserData>This is my HighSpeedHd job.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <Progress>100</Progress>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T15:29:42+0800</StartTime>
        <State>Success</State>
        <Tag>Transcode</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调



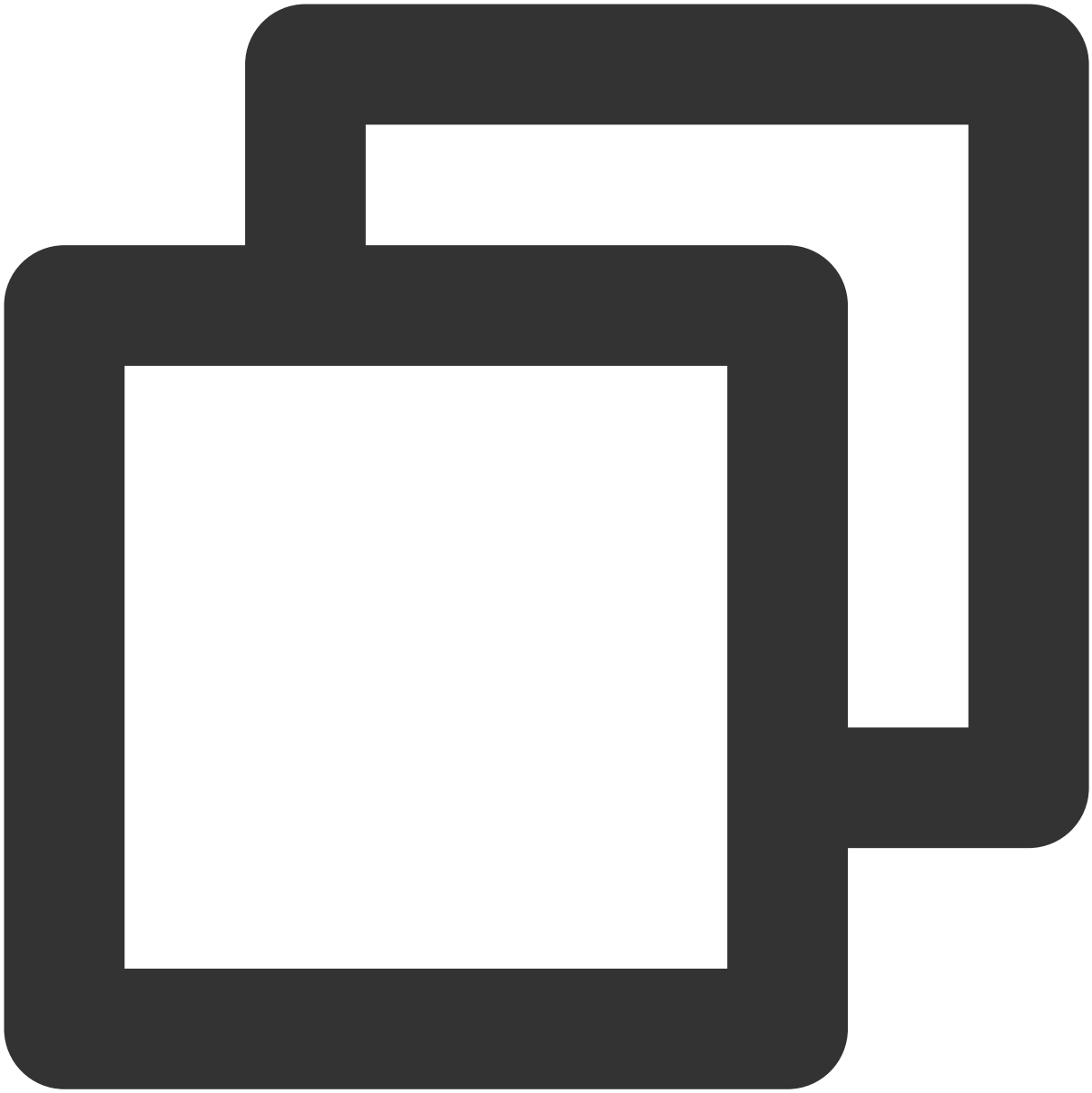
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
        <EndTime>2022-06-30T15:30:08+0800</EndTime>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</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/high_speed.mp4</ObjectName>
                </Md5Info>
                <ObjectName>output/high_speed.mp4</ObjectName>
                <ObjectPrefix/>
                <Region>ap-chongqing</Region>
            </OutputFile>
        </MediaResult>
        <Output>
            <Bucket>test-123456789</Bucket>
            <Object>output/high_speed.mp4</Object>
            <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
        <TemplateName>high_993874</TemplateName>
        <UserData>This is my HighSpeedHd job.</UserData>
```

```
            <JobLevel>0</JobLevel>
        </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/high_speed.mp4"
                },
                "ObjectName": "output/high_speed.mp4",
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/high_speed.mp4",
            "Region": "ap-chongqing"
        },
        "TemplateId": "t1995d523e42df4c5e858f244b4174360c",
        "TemplateName": "high_993874",
        "UserData": "This is my HighSpeedHd job.",
        "JobLevel": 0
```

```
        },
        "Progress": "100",
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T15:29:42+0800",
        "State": "Success",
        "Tag": "Transcode",
        "Workflow": {
            "Name": "HighSpeedHd_1581665960537",
            "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
            "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
            "WorkflowName": "workflow-test"
        }
    }]
}
```

图片处理
提交图片处理任务
CreatePicJobs

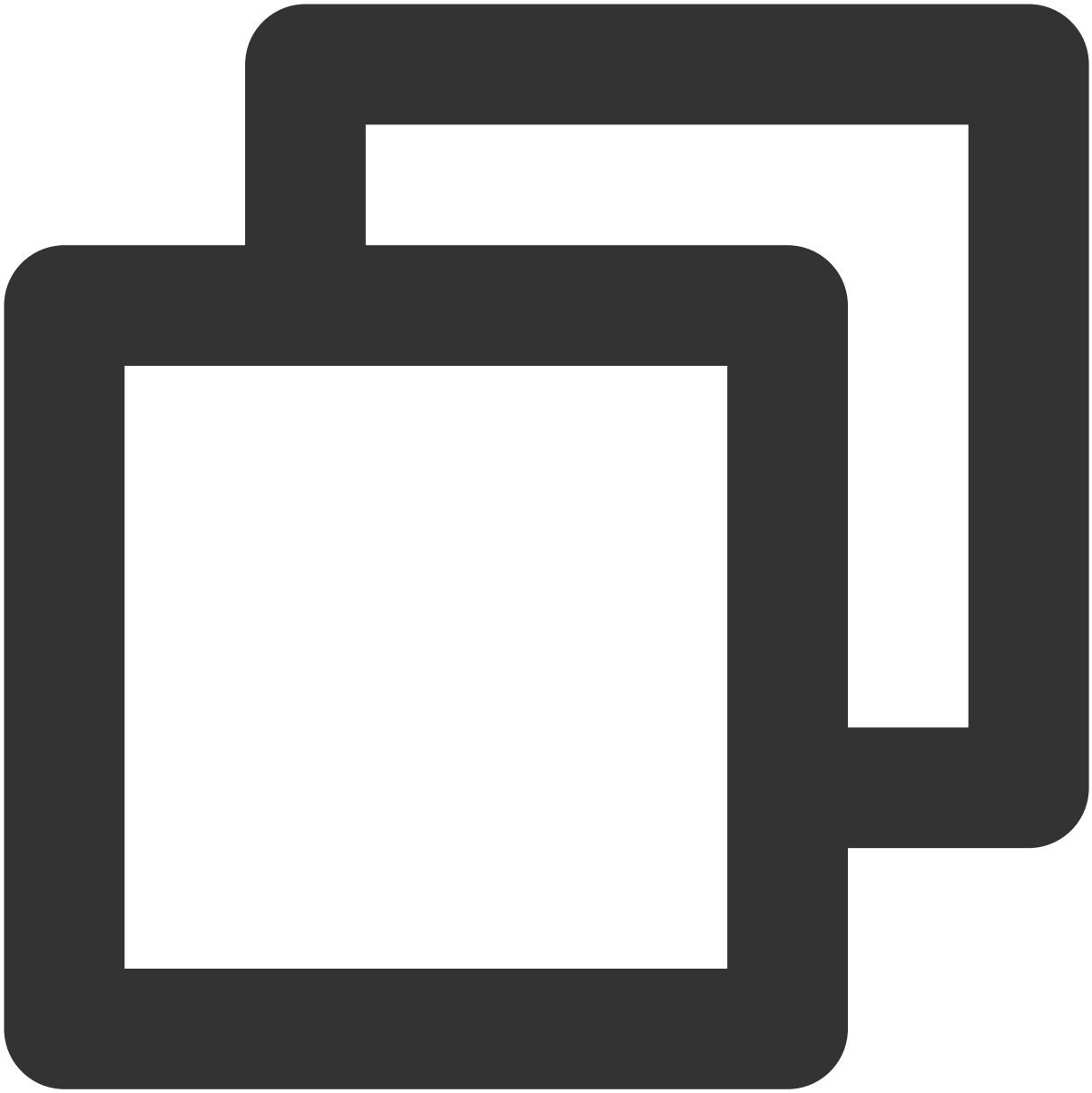
最近更新时间：2024-01-20 16:30:33

功能描述

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：

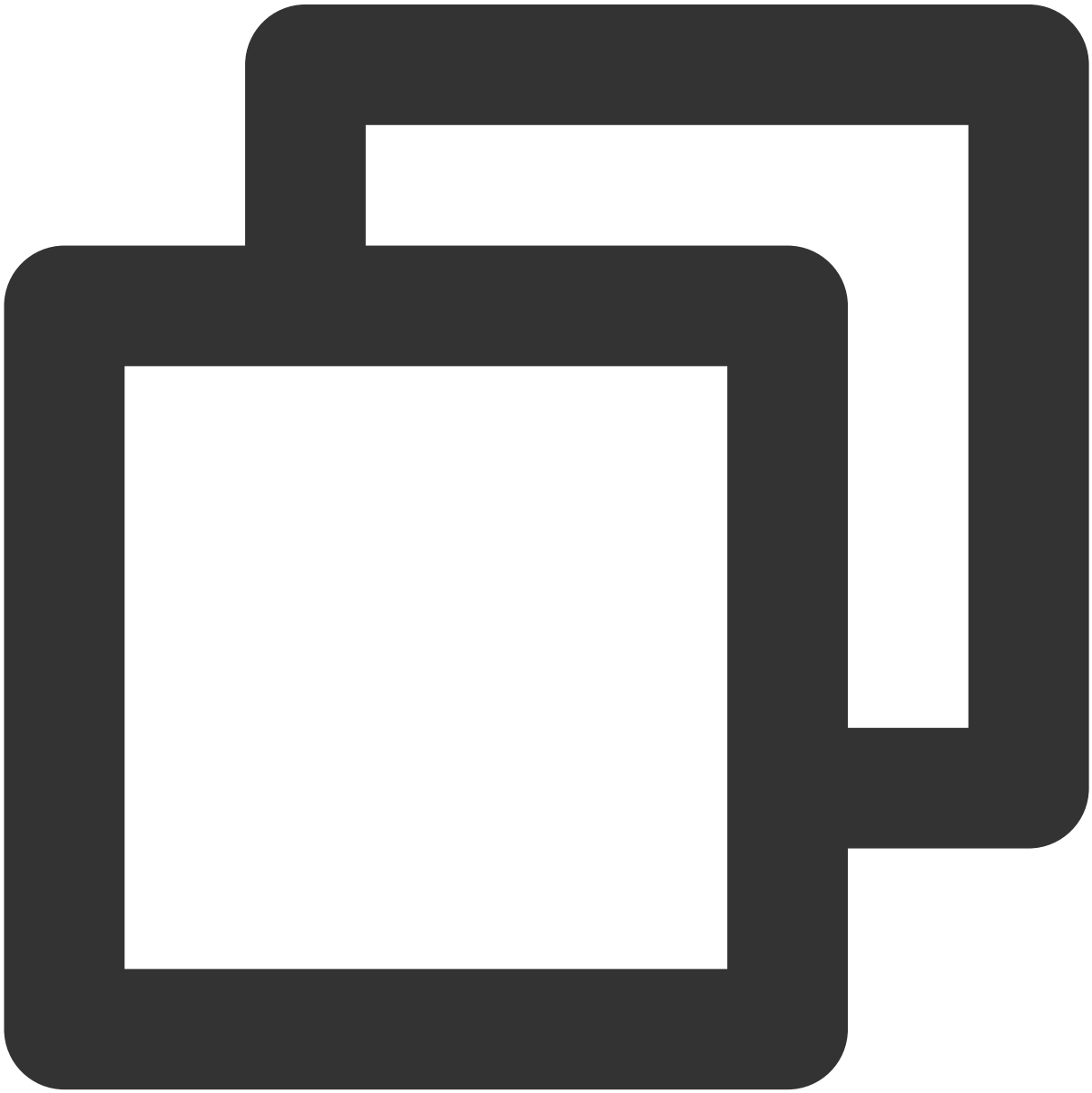
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>
        <JobLevel>0</JobLevel>
    </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
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

版权所有：腾讯云计算（北京）有限责任公司

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	媒体文件名	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
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
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

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

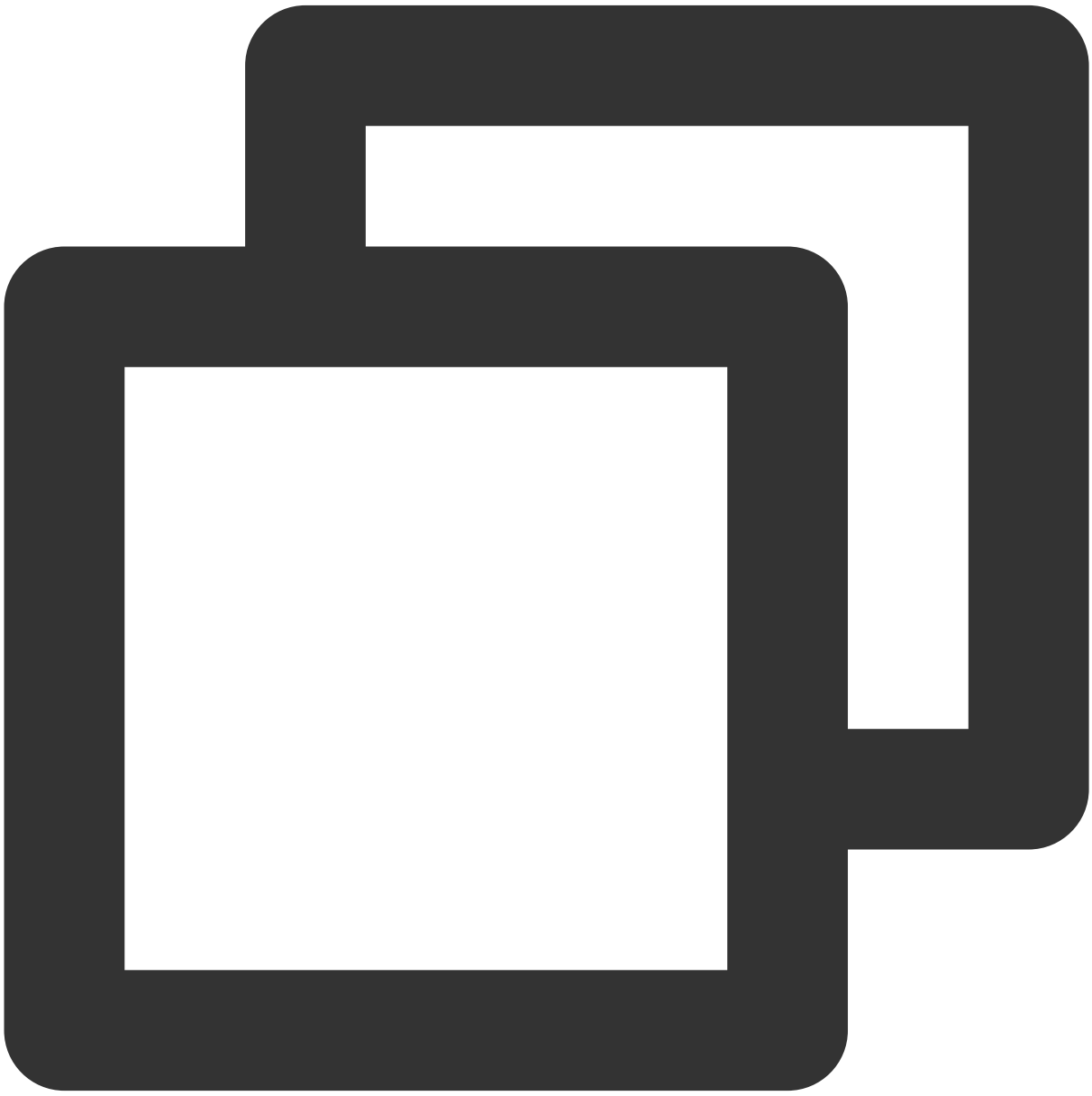
响应

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>
            <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

版权所有：腾讯云计算（北京）有限责任公司

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
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

版权所有：腾讯云计算（北京）有限责任公司

UserData	Response.JobsDetail.Operation	透传用户信息	String

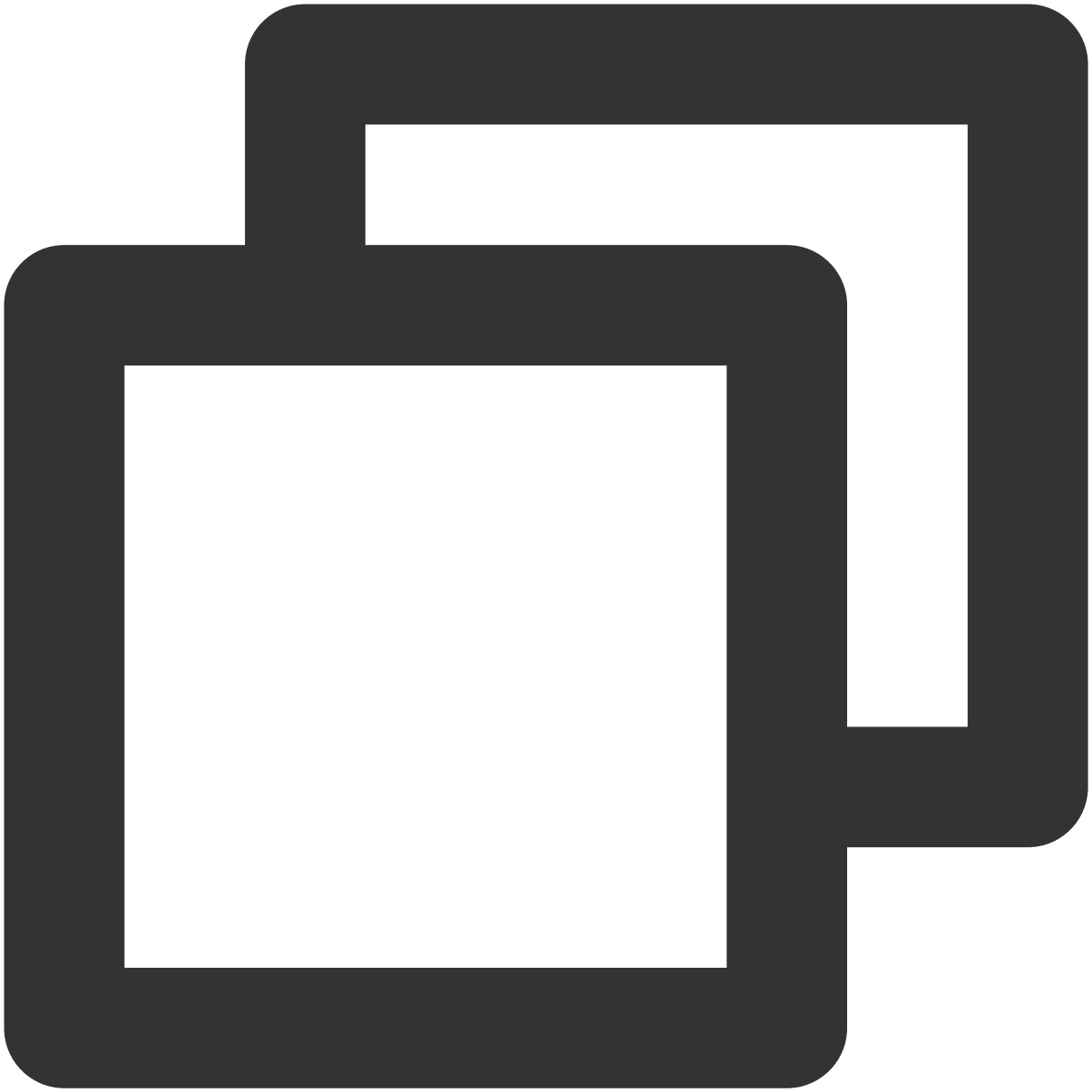
Container 节点 PicProcessResult 的内容：
同 图片持久化处理 接口中的 UploadResult 节点。

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

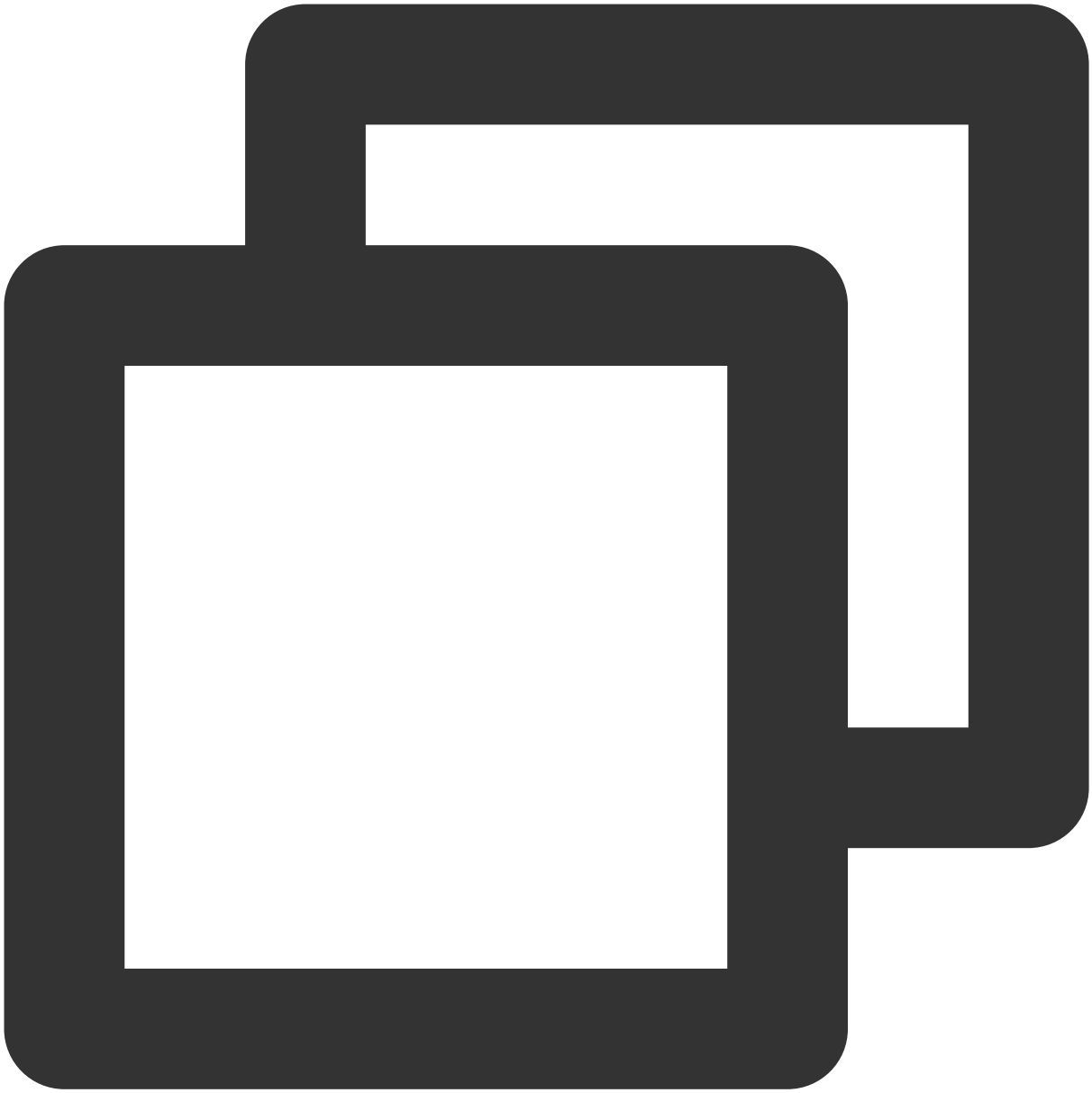


```
POST /pic_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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>
        <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应

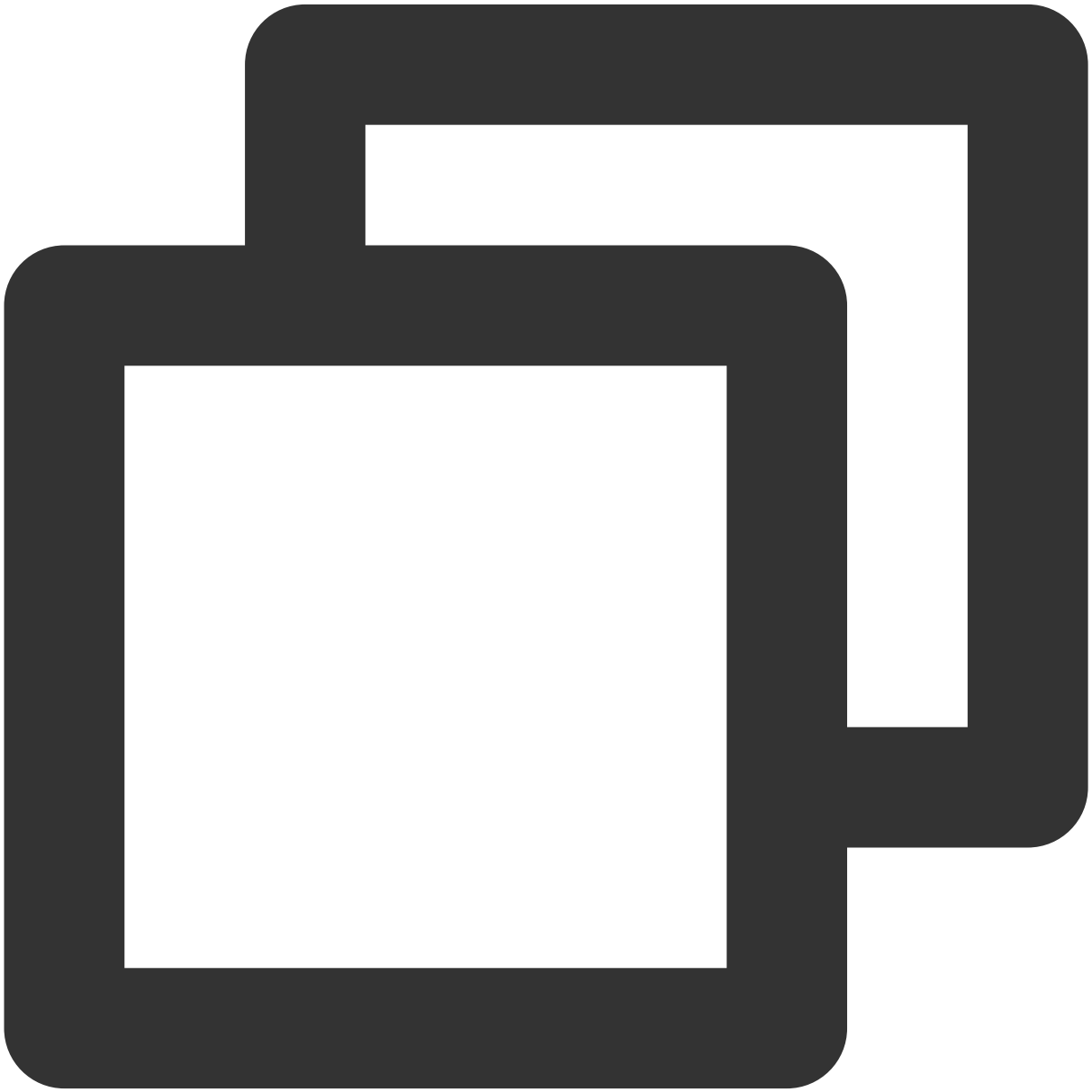


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

请求2：使用图片处理处理参数

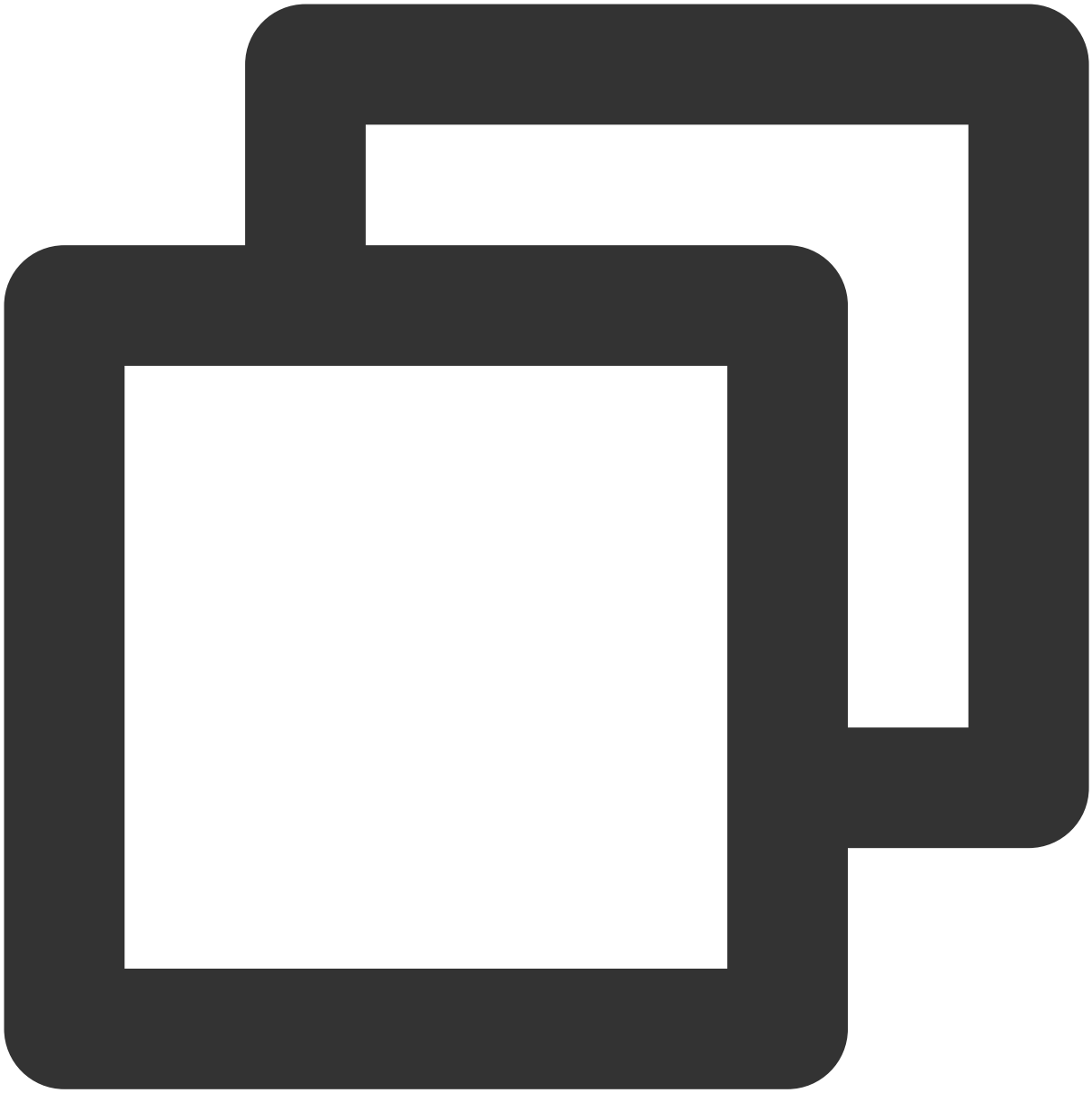


```
POST /pic_jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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>
        <JobLevel>0</JobLevel>
    </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>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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

取消图片处理任务

CancelPicJob

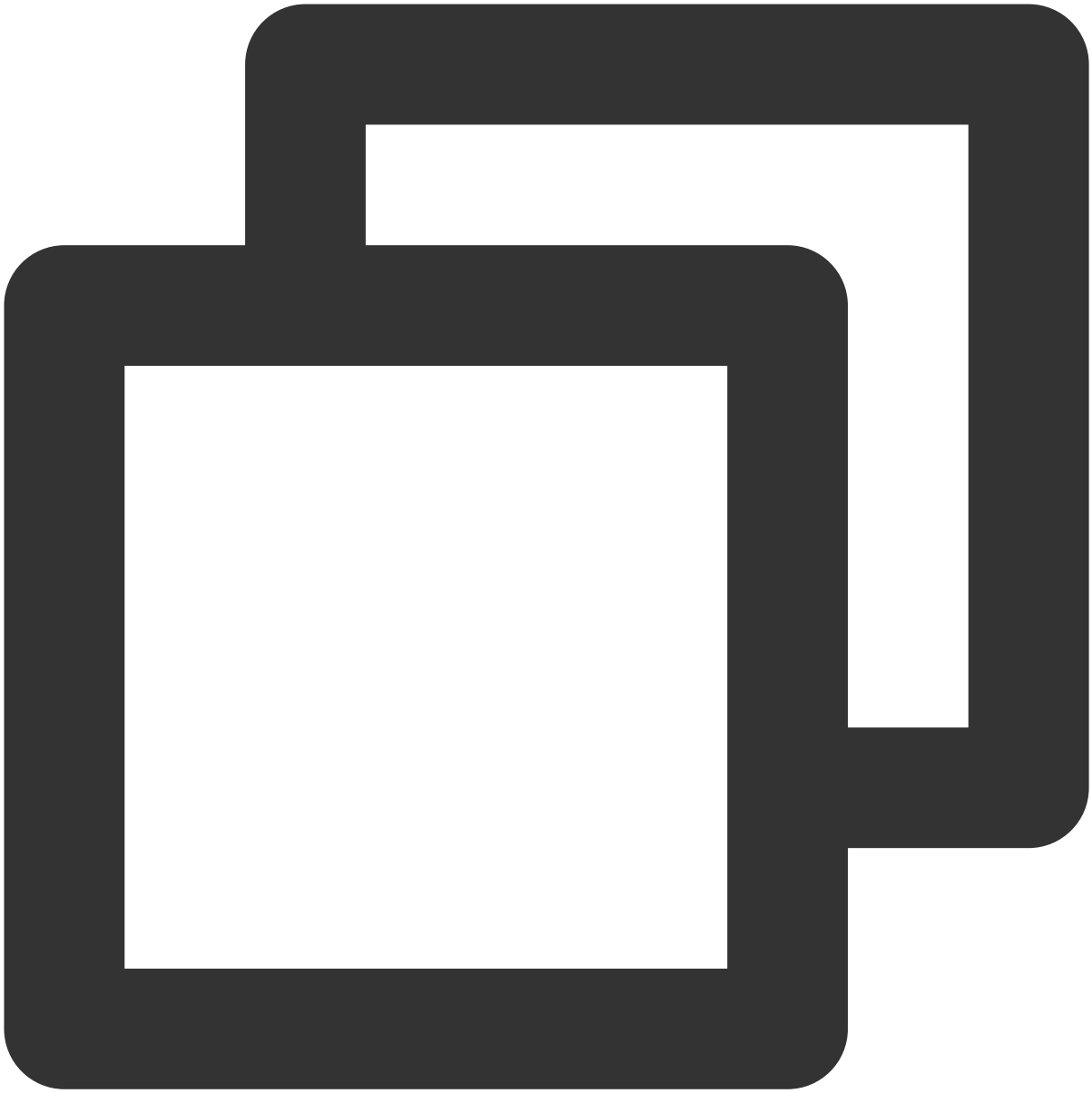
最近更新时间：2024-01-20 16:30:33

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

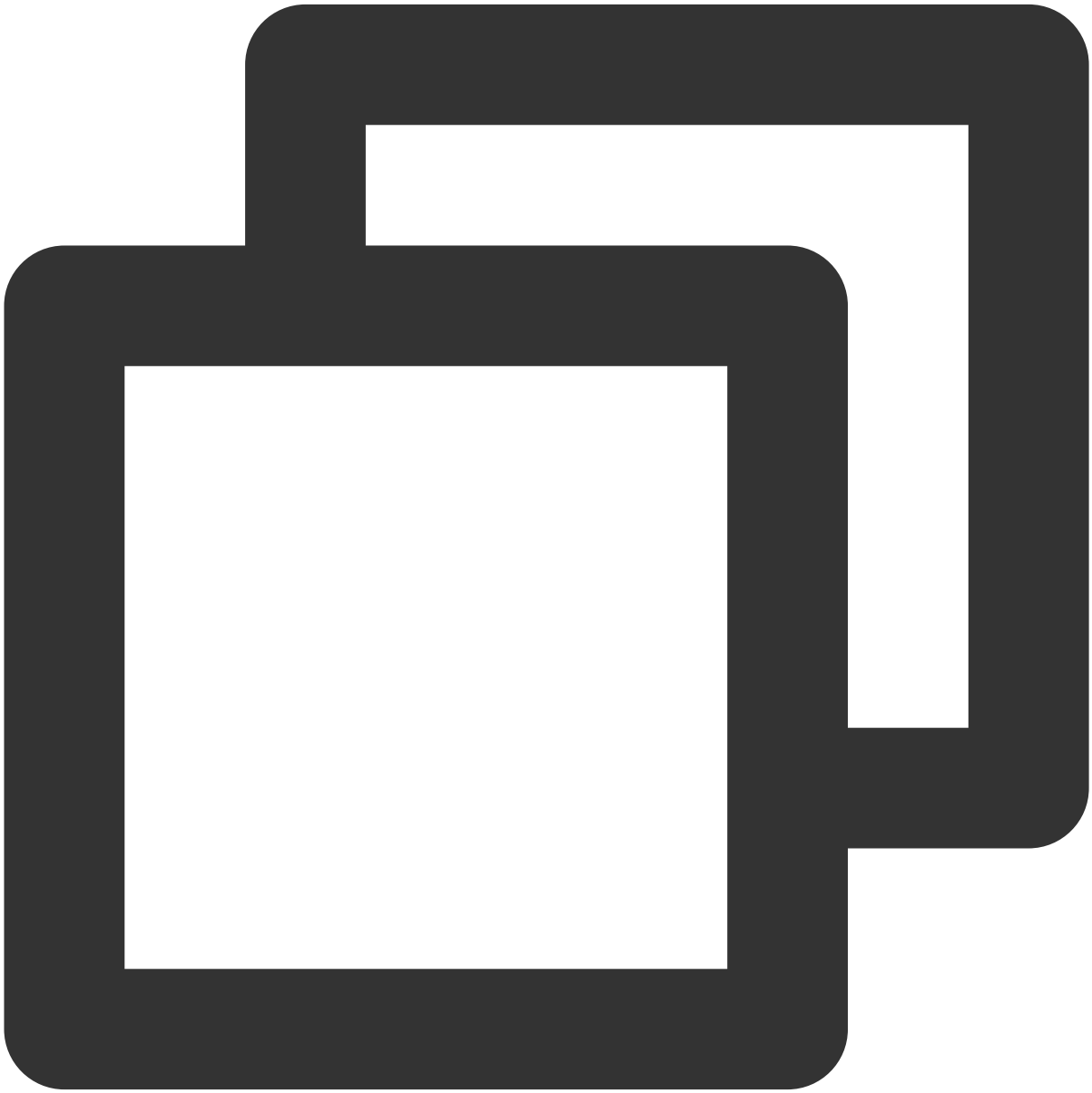
该请求无响应体。

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。

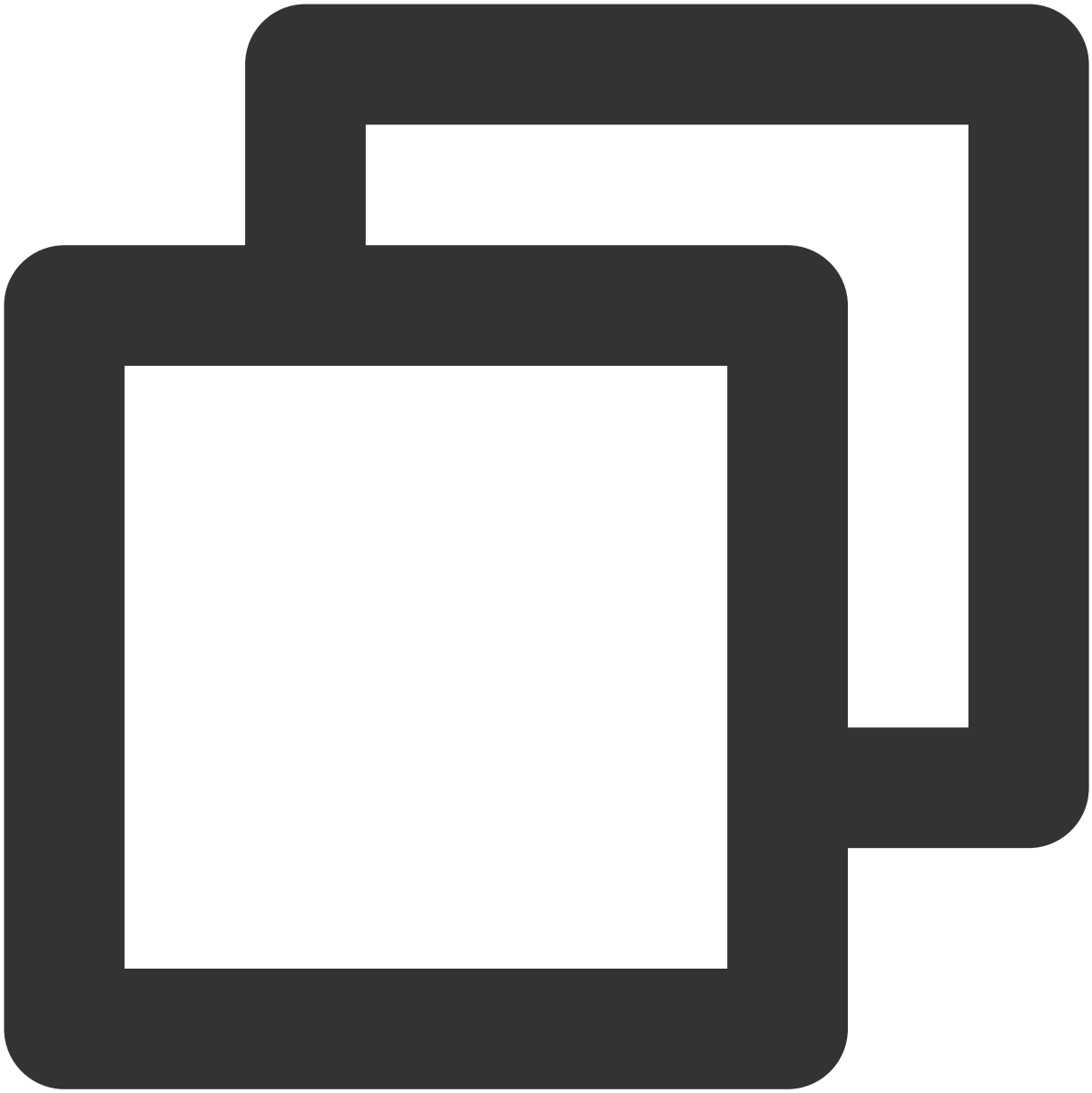
实际案例

请求



```
PUT /jobs/c93984788066911ed89ed352d4d9d2084?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, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=
```

查询图片处理任务

DescribePicJob

最近更新时间：2024-01-20 16:30:33

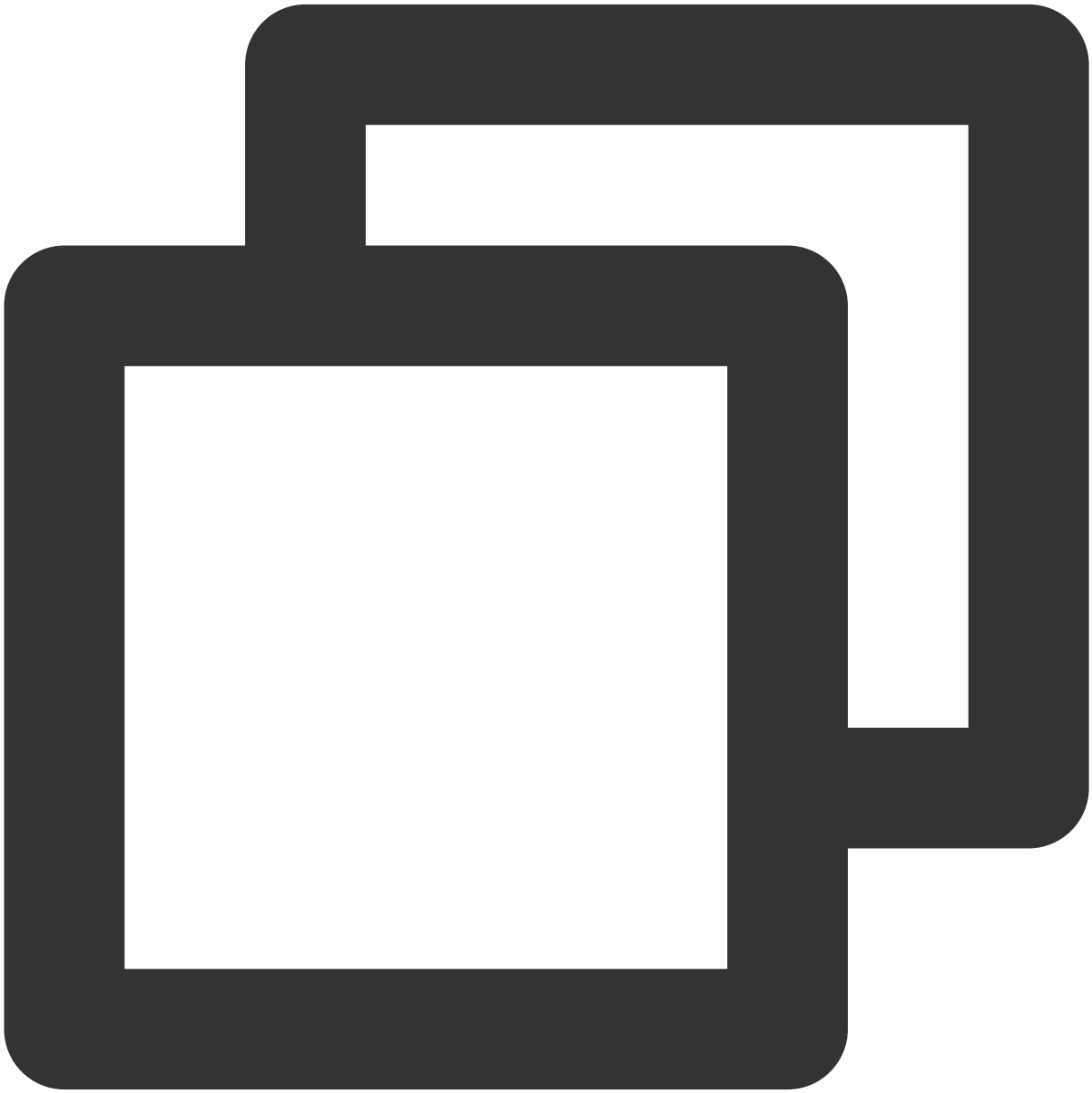
功能描述

查询指定的任务。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
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。

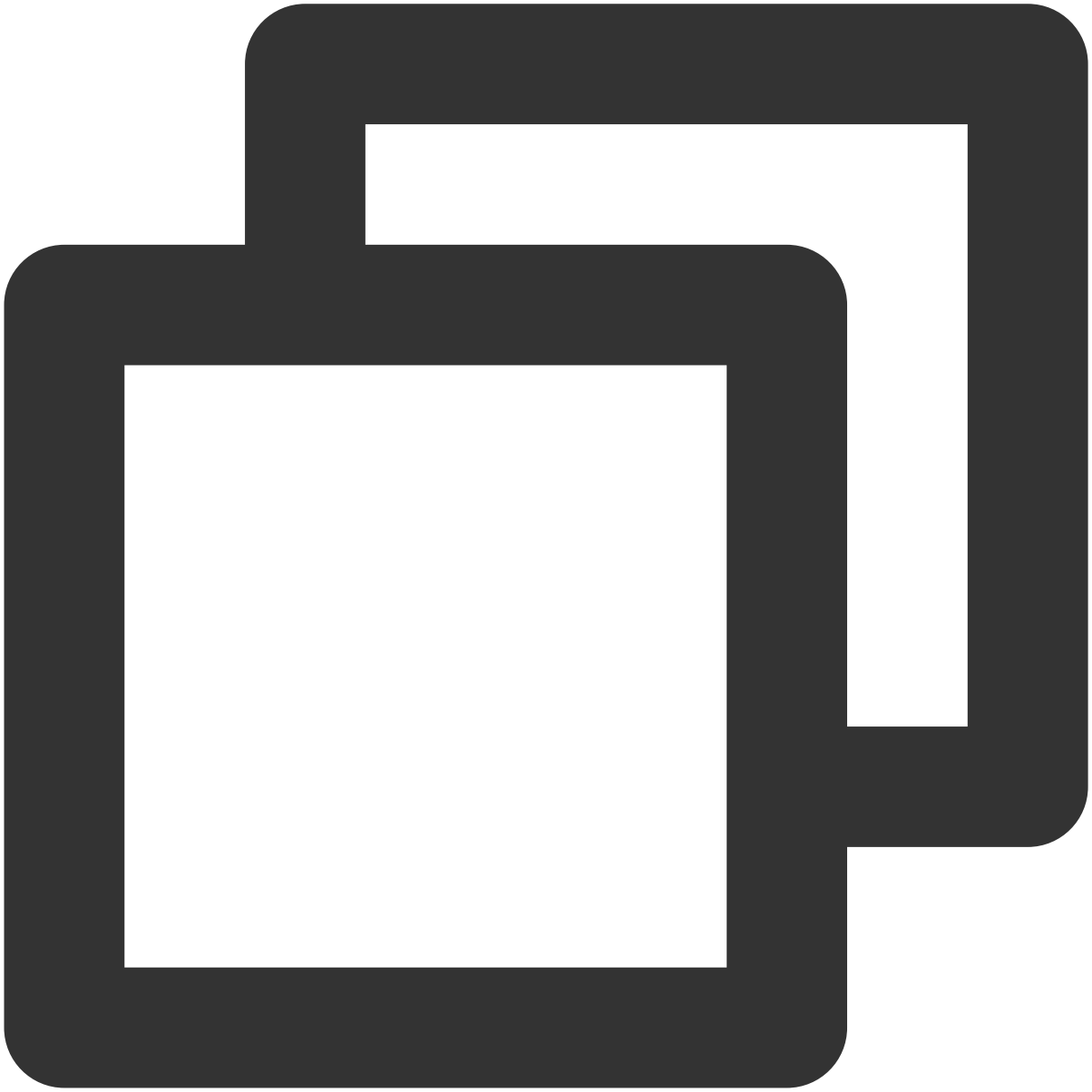
响应

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

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
NonExistJobIds	Response	查询的 ID 中不存在任务，所有任务都存在时不返回	String

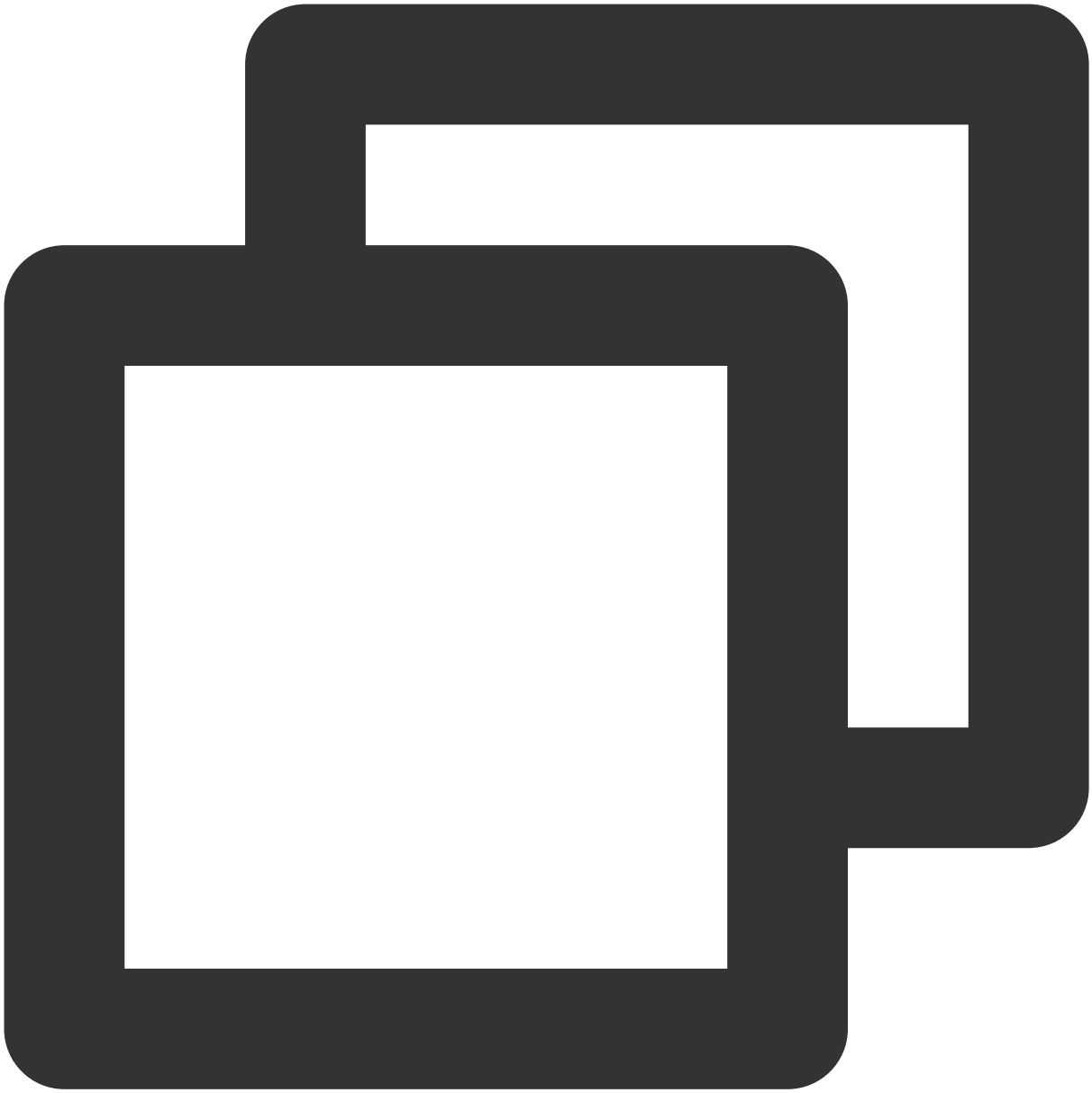
对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：
图片处理

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。

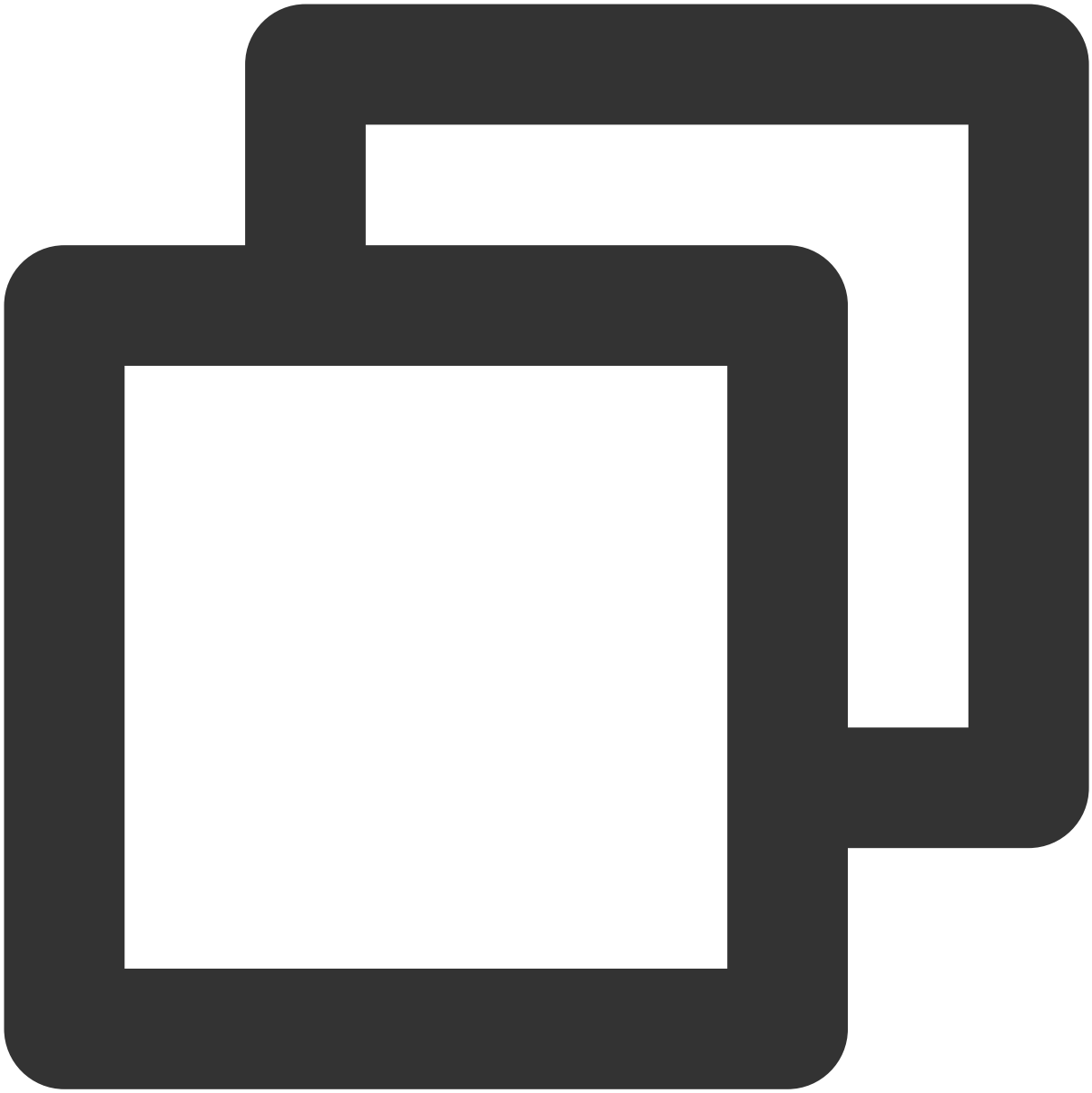
实际案例

请求



```
GET /pic_jobs/c93984788066911ed89ed352d4d9d2084 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, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

DescribePicJobs

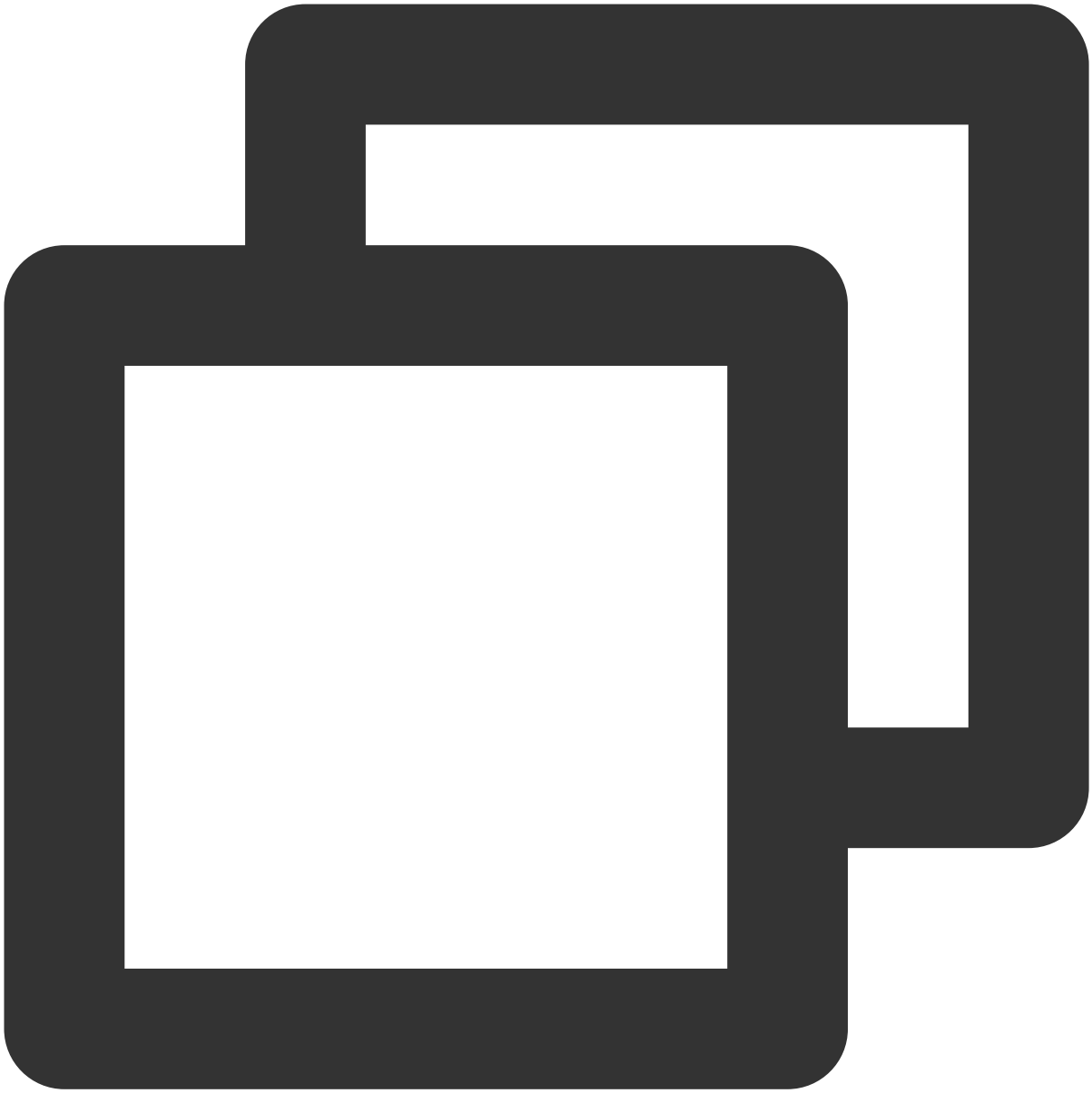
最近更新时间：2024-01-20 16:30:33

功能描述

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	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
图片处理	PicProcess

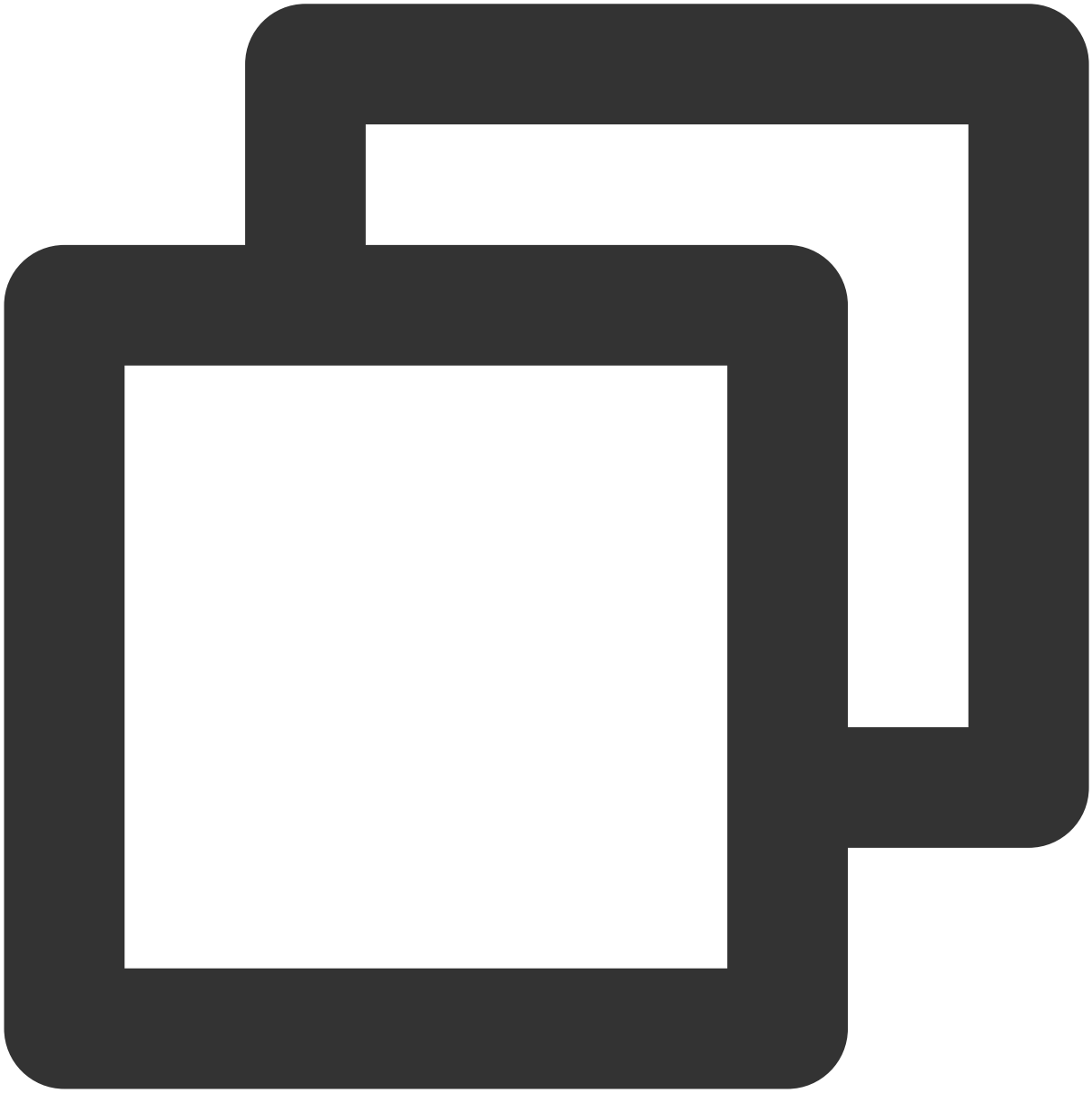
响应

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

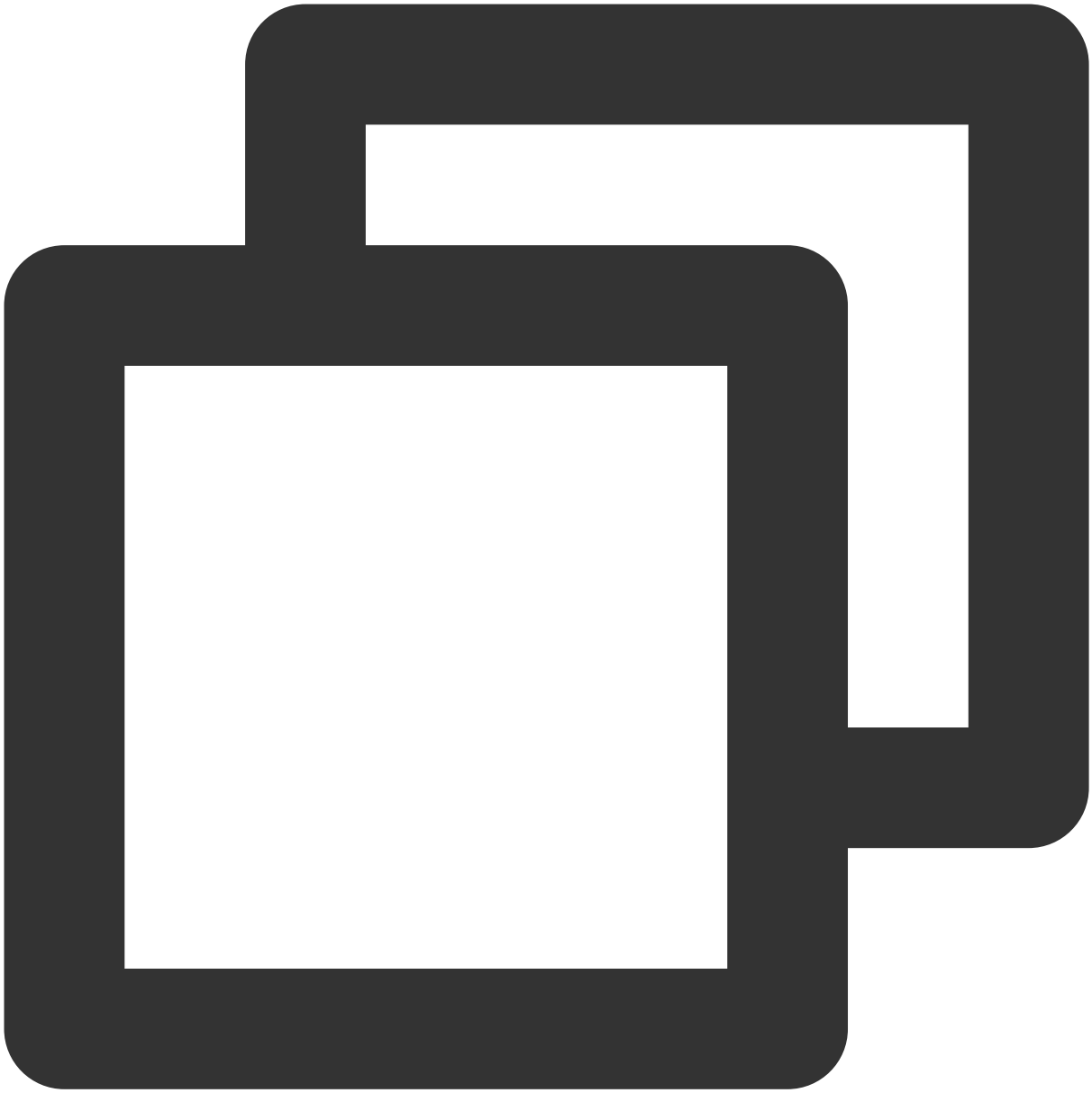
对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：
图片处理

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。

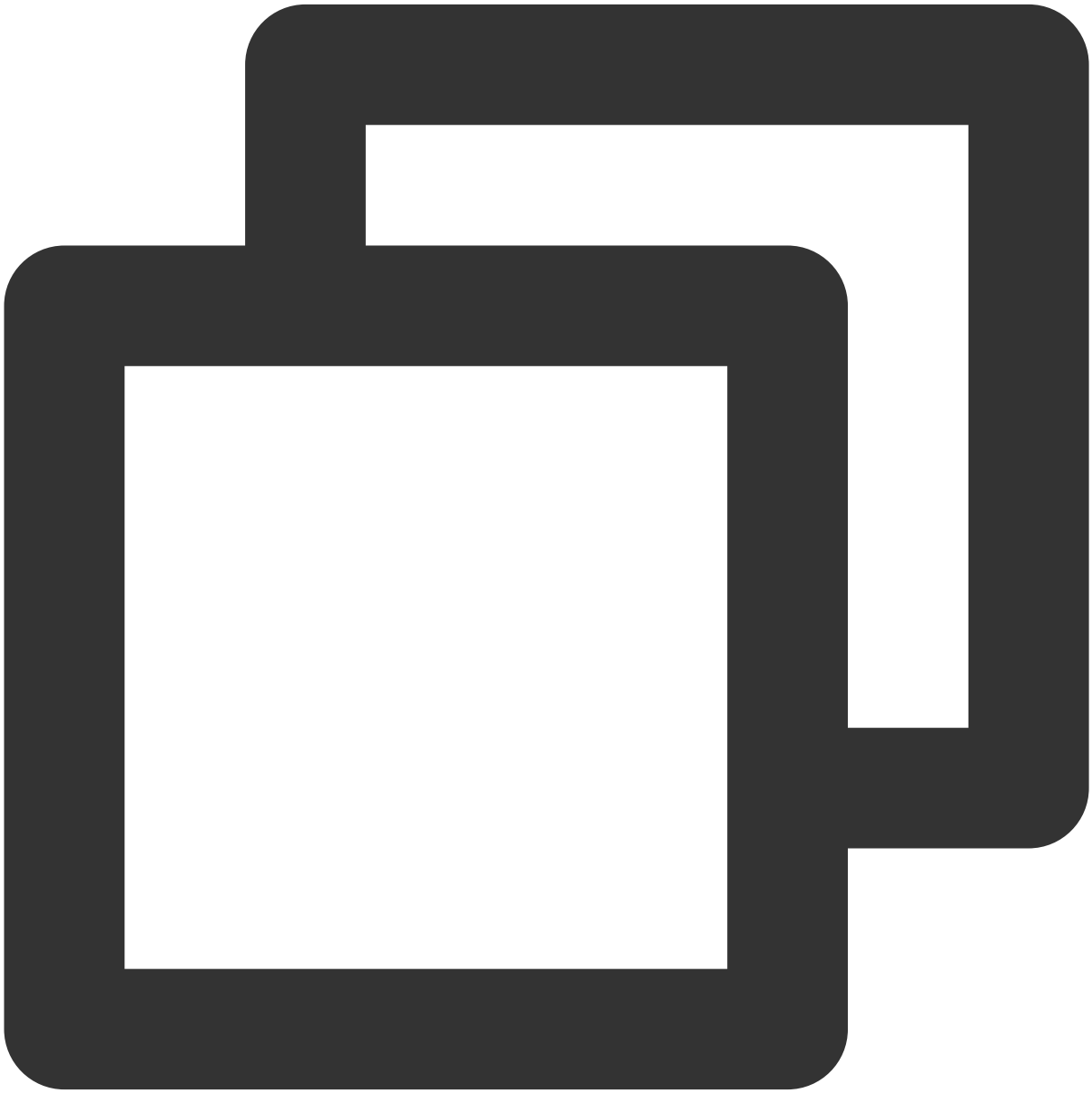
实际案例

请求



```
GET /pic_jobs?queueId=p2911917386e148639319e13c285cc774&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: Mon, 18 Jul 2022 19:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <JobsDetail>
        <Code>Success</Code>
```

```
        <Message/>
        <JobId>c93984788066911e689ed352d4d9d2ghd4</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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
    <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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

图片处理回调

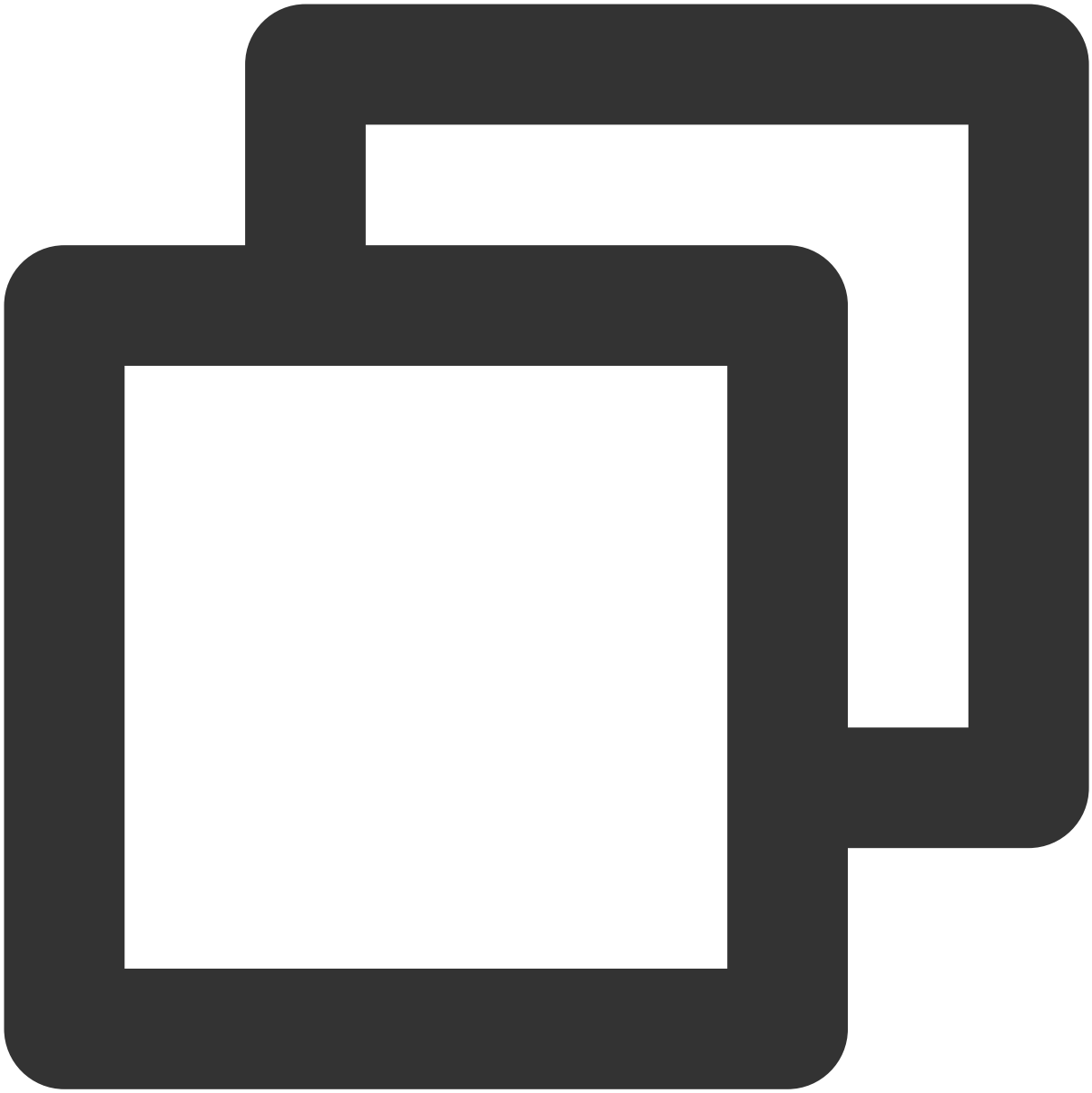
最近更新时间：2024-01-20 16:30:33

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>
            <JobLevel>0</JobLevel>
        </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 数组

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

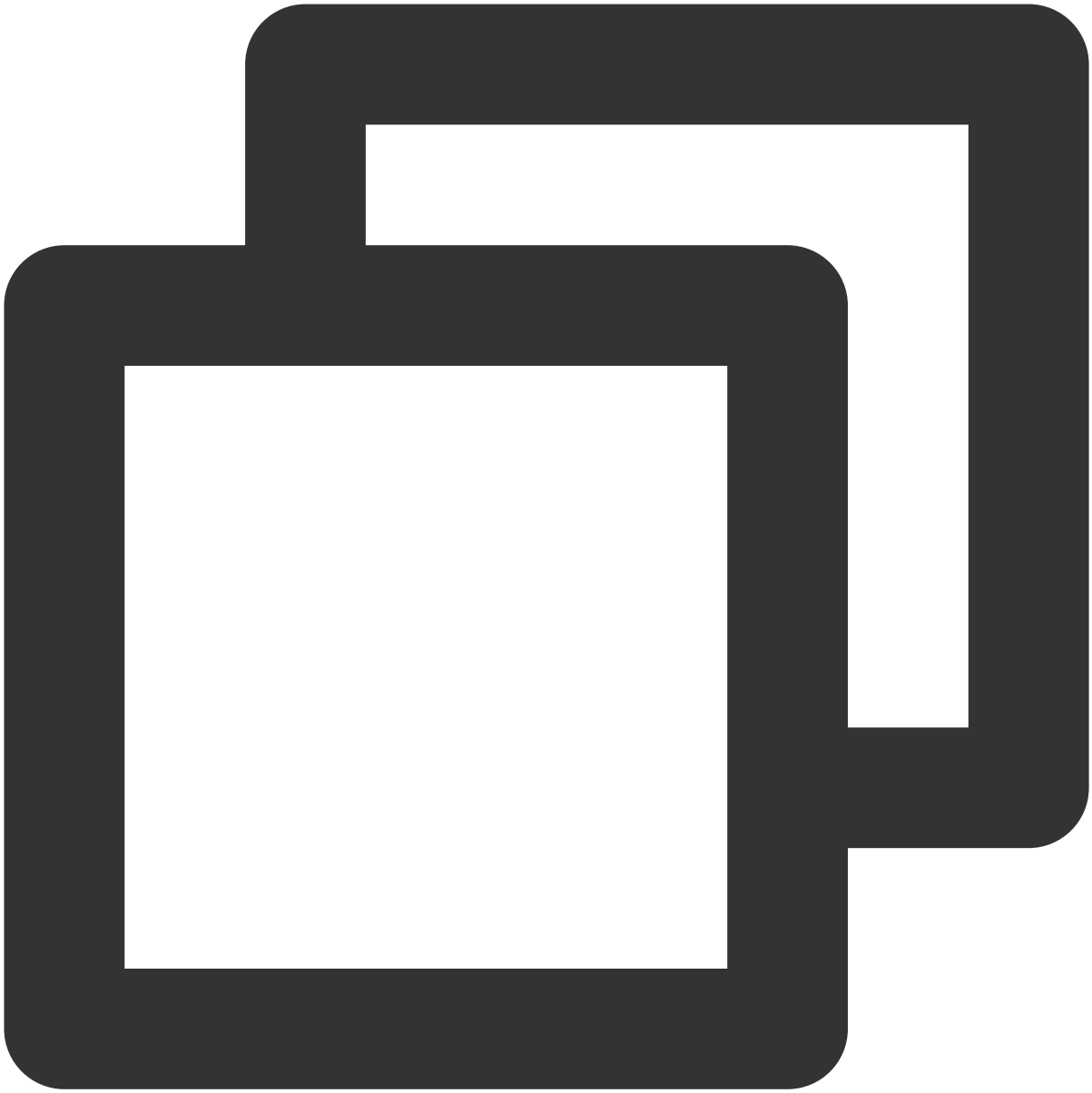
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

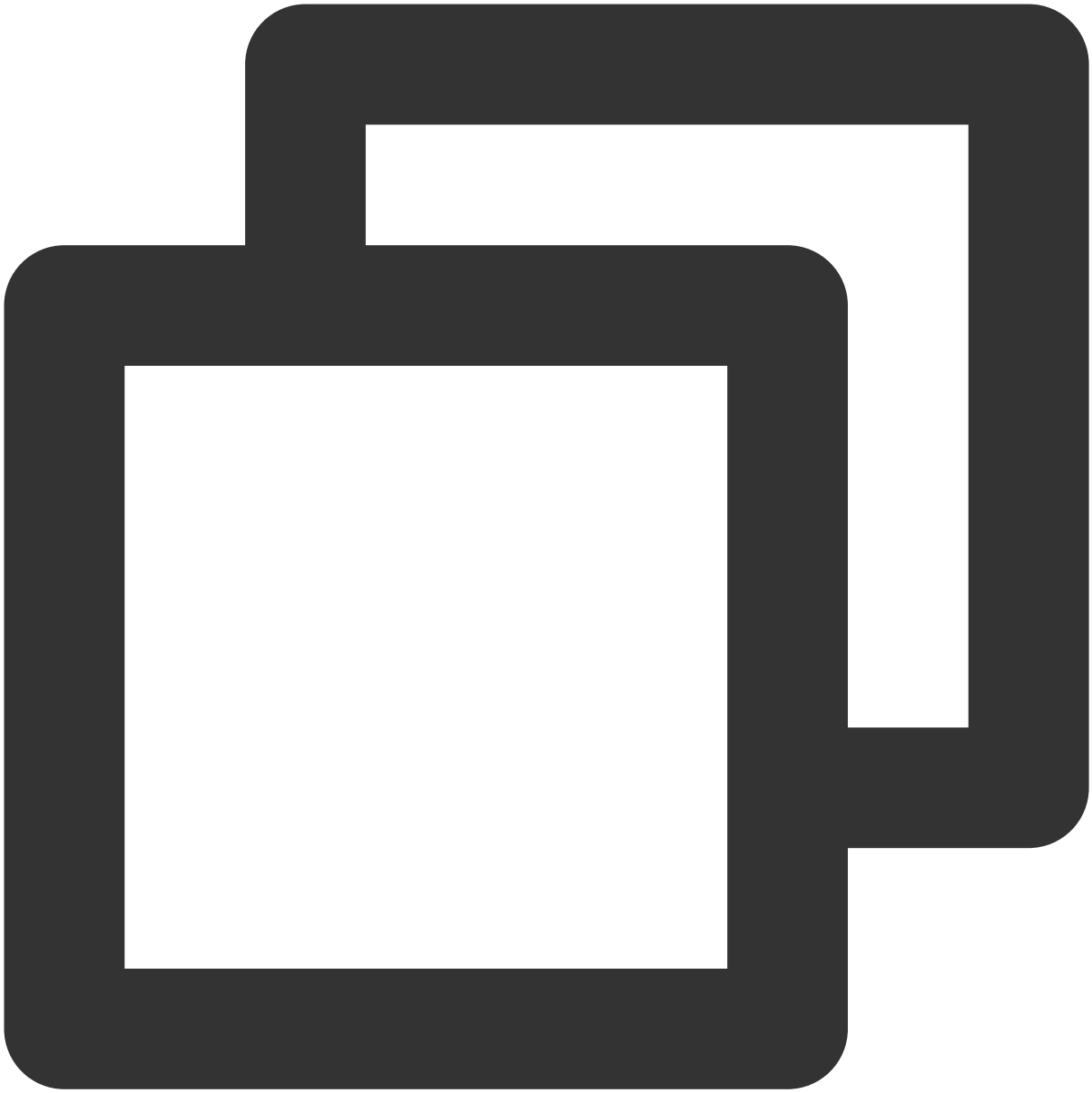
案例 1：通过任务接口触发的任务回调



```
<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>
            <JobLevel>0</JobLevel>
        </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

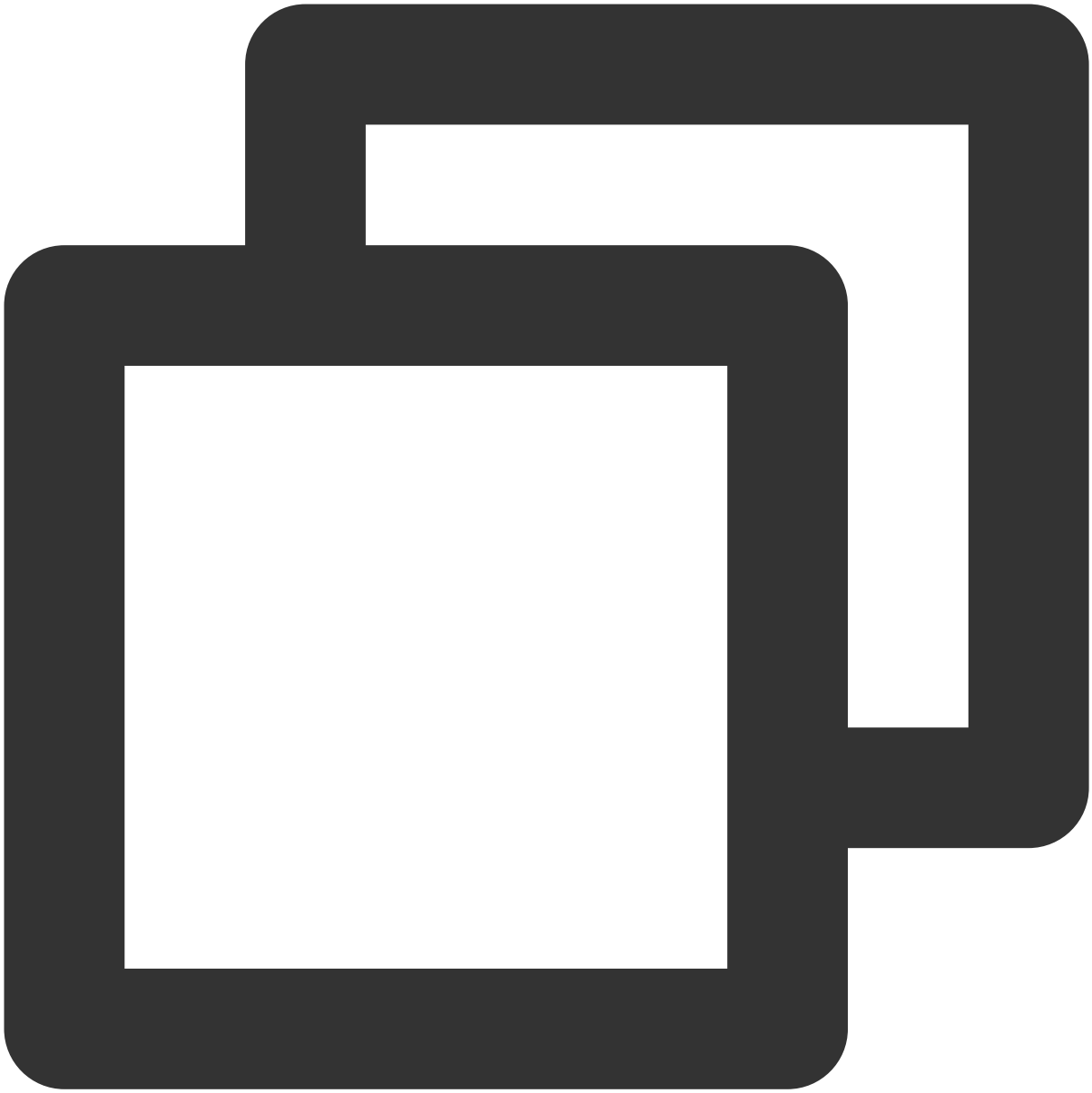


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>2022-07-18T15:16:44+0800</EndTime>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/deer.jpg</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>
            <JobLevel>0</JobLevel>
        </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.",
            "JobLevel": 0,
        },
        "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

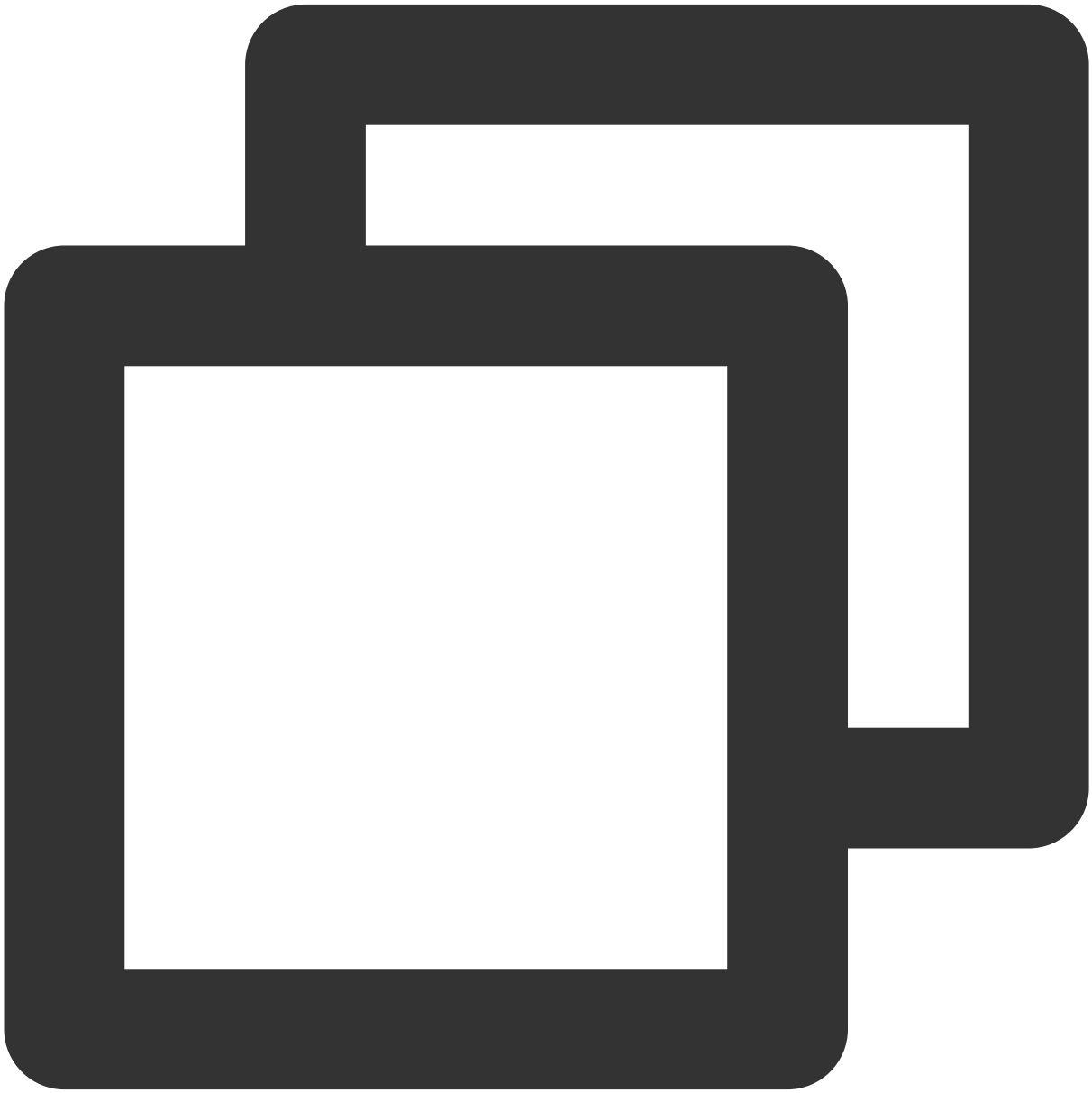
最近更新时间：2024-03-01 17:21:38

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：

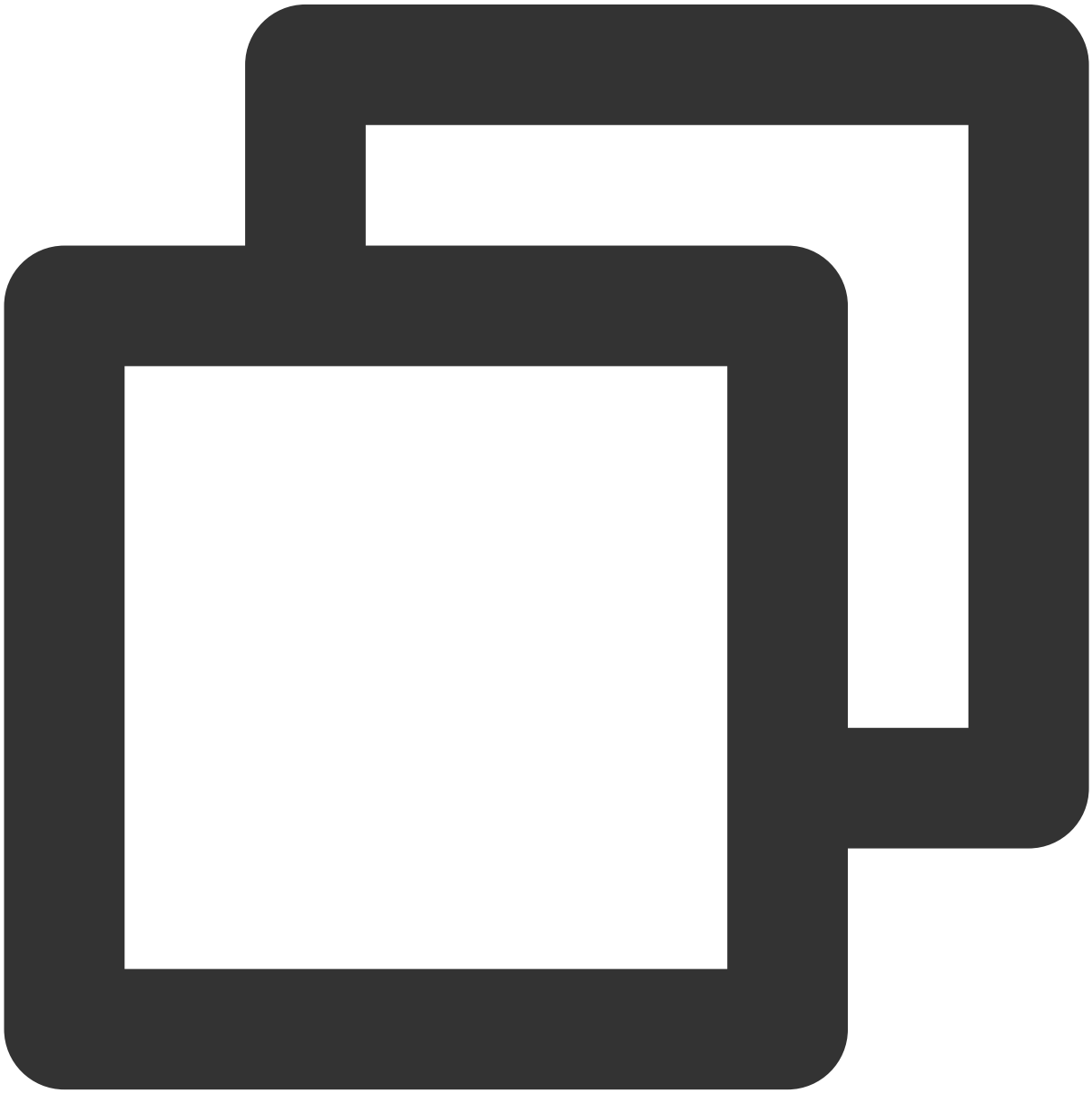
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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
视频质量分析
语音合成

版权所有：腾讯云计算（北京）有限责任公司

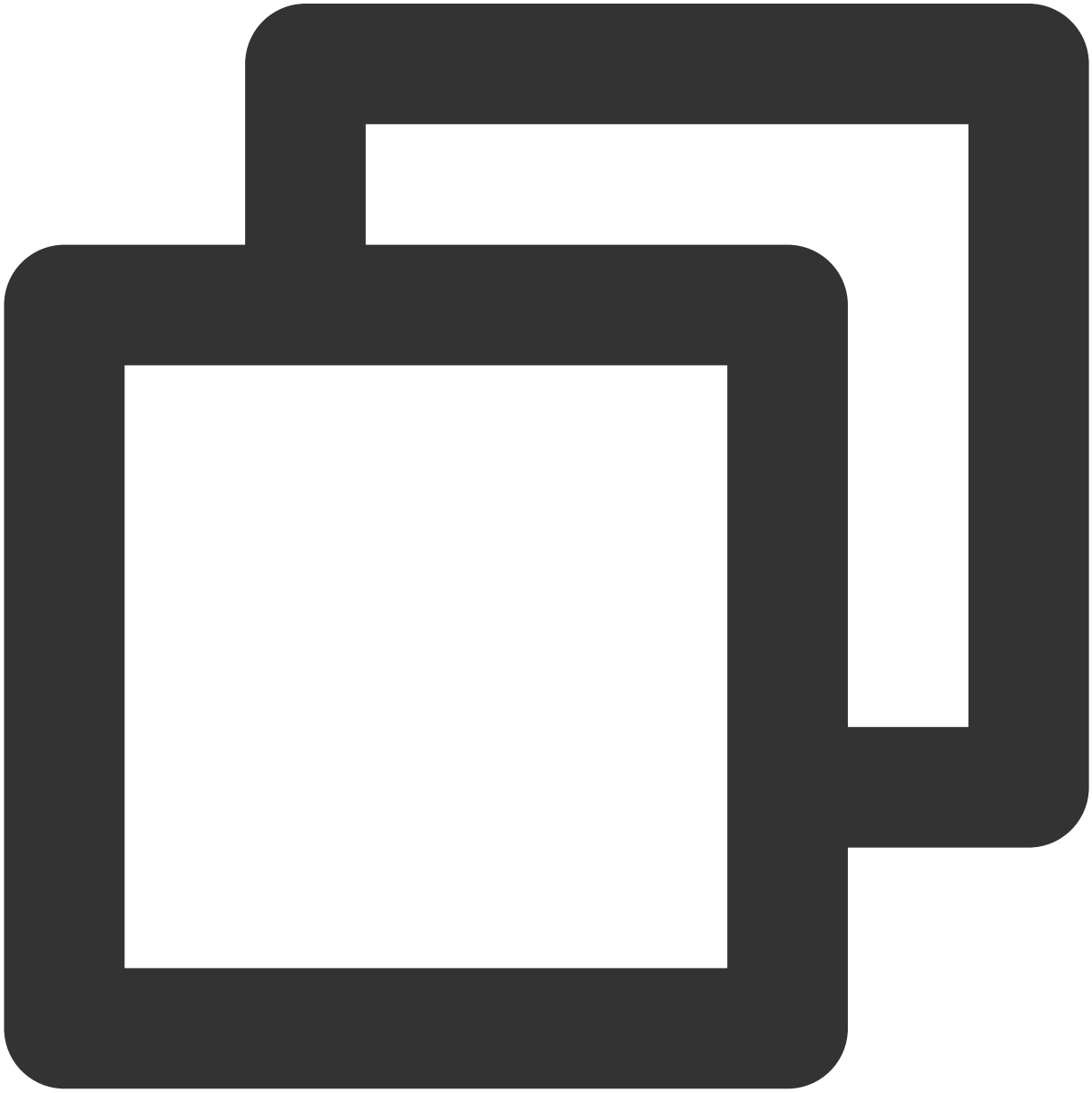
音频降噪

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
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

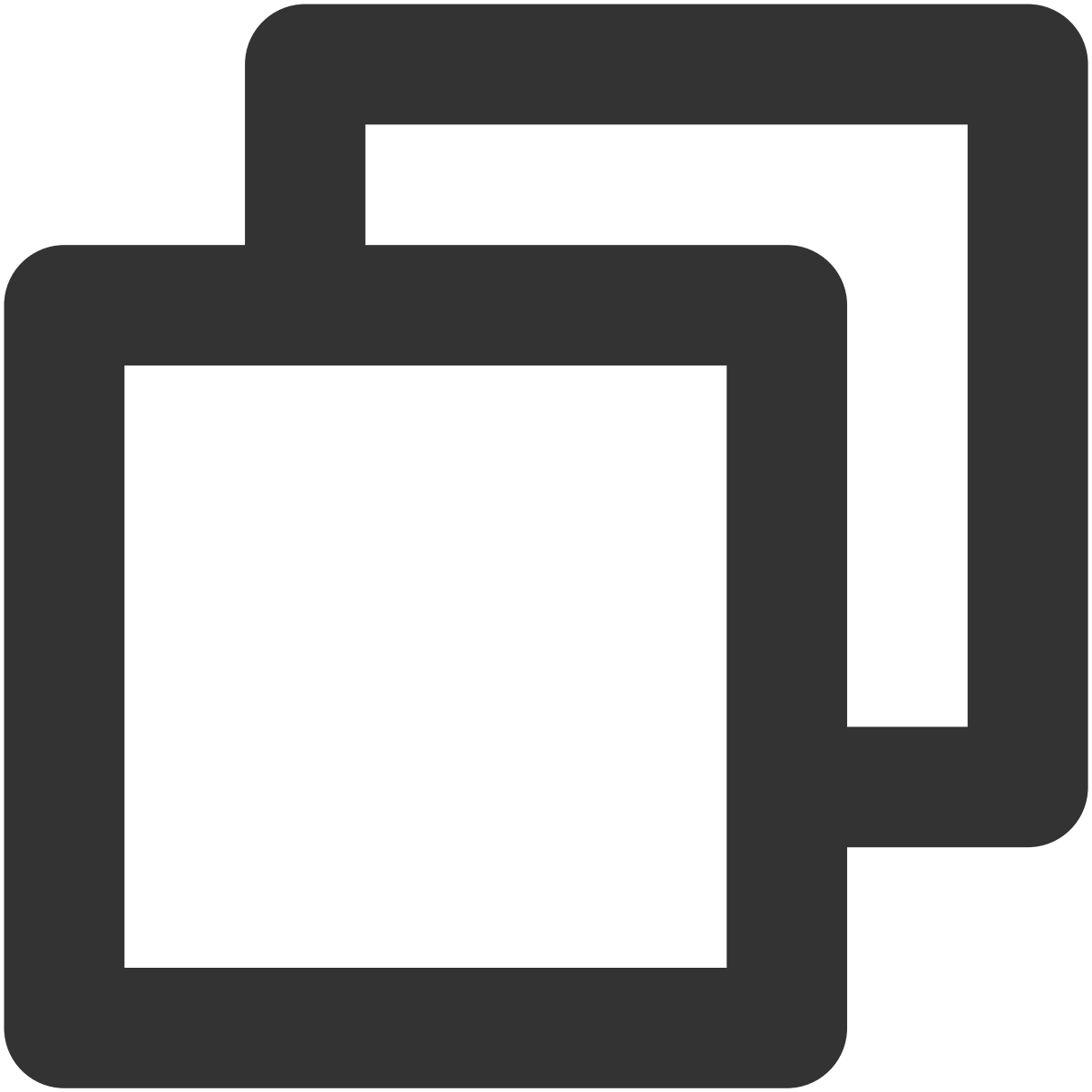
错误码

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

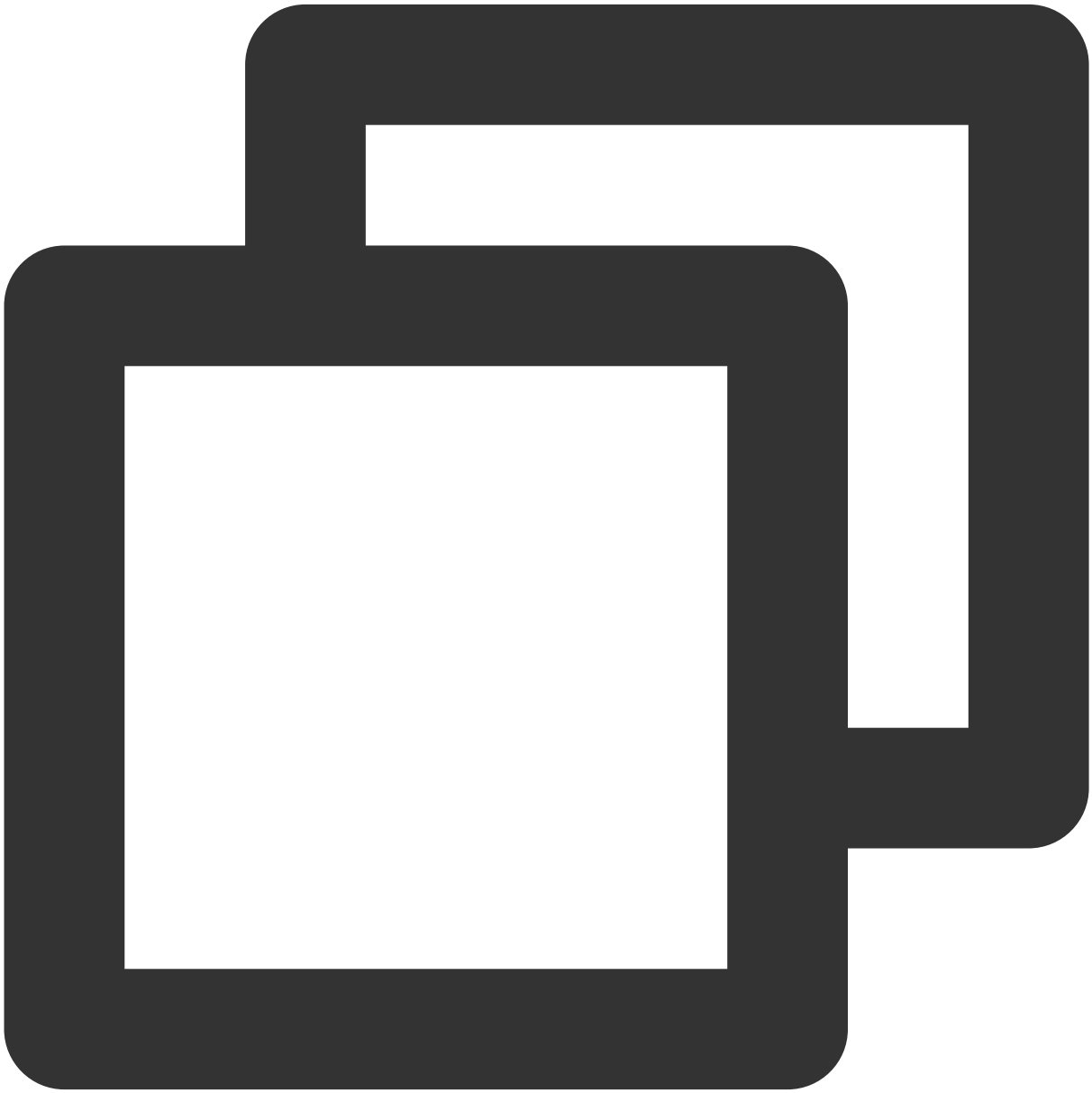


```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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

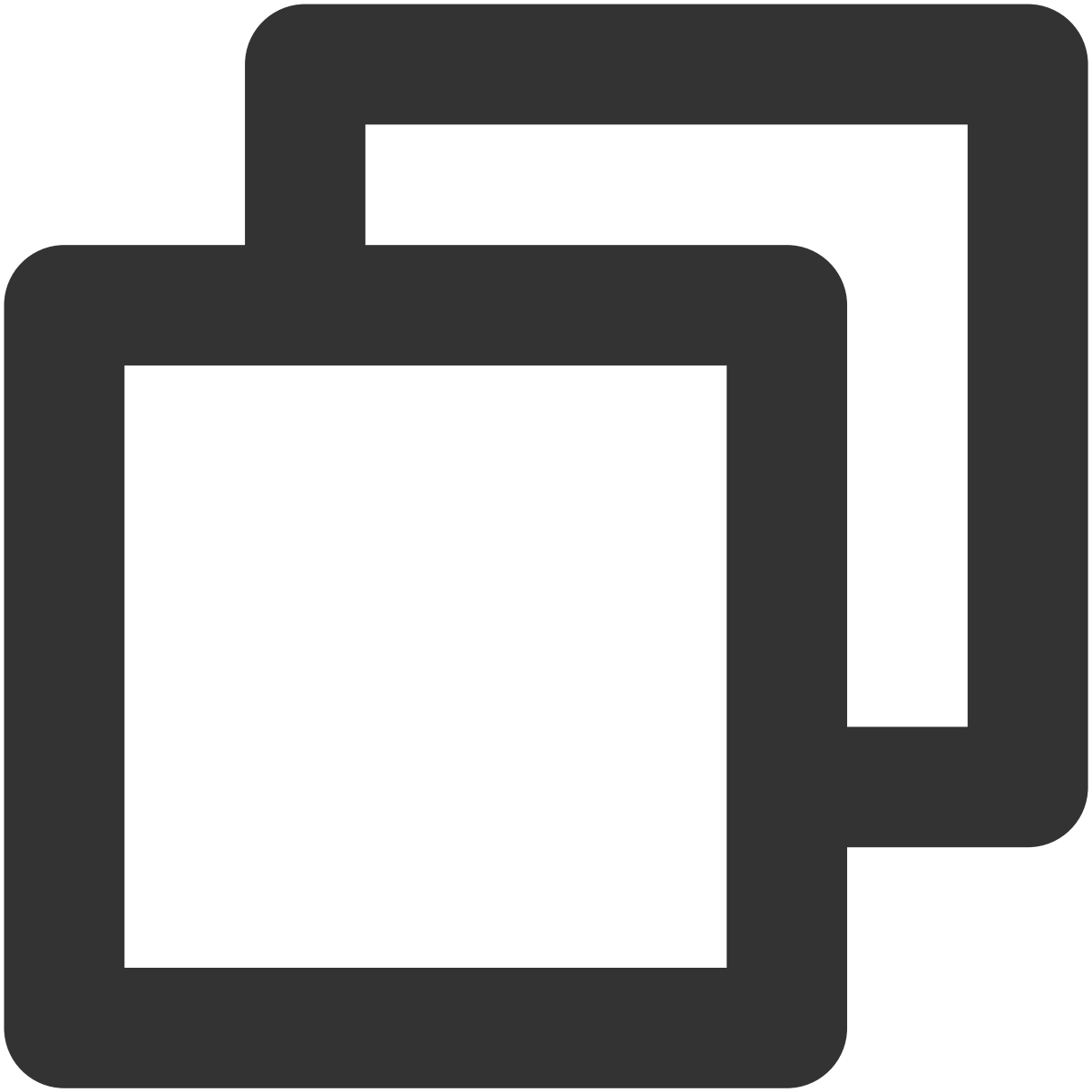
最近更新时间：2024-01-20 16:35:12

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：

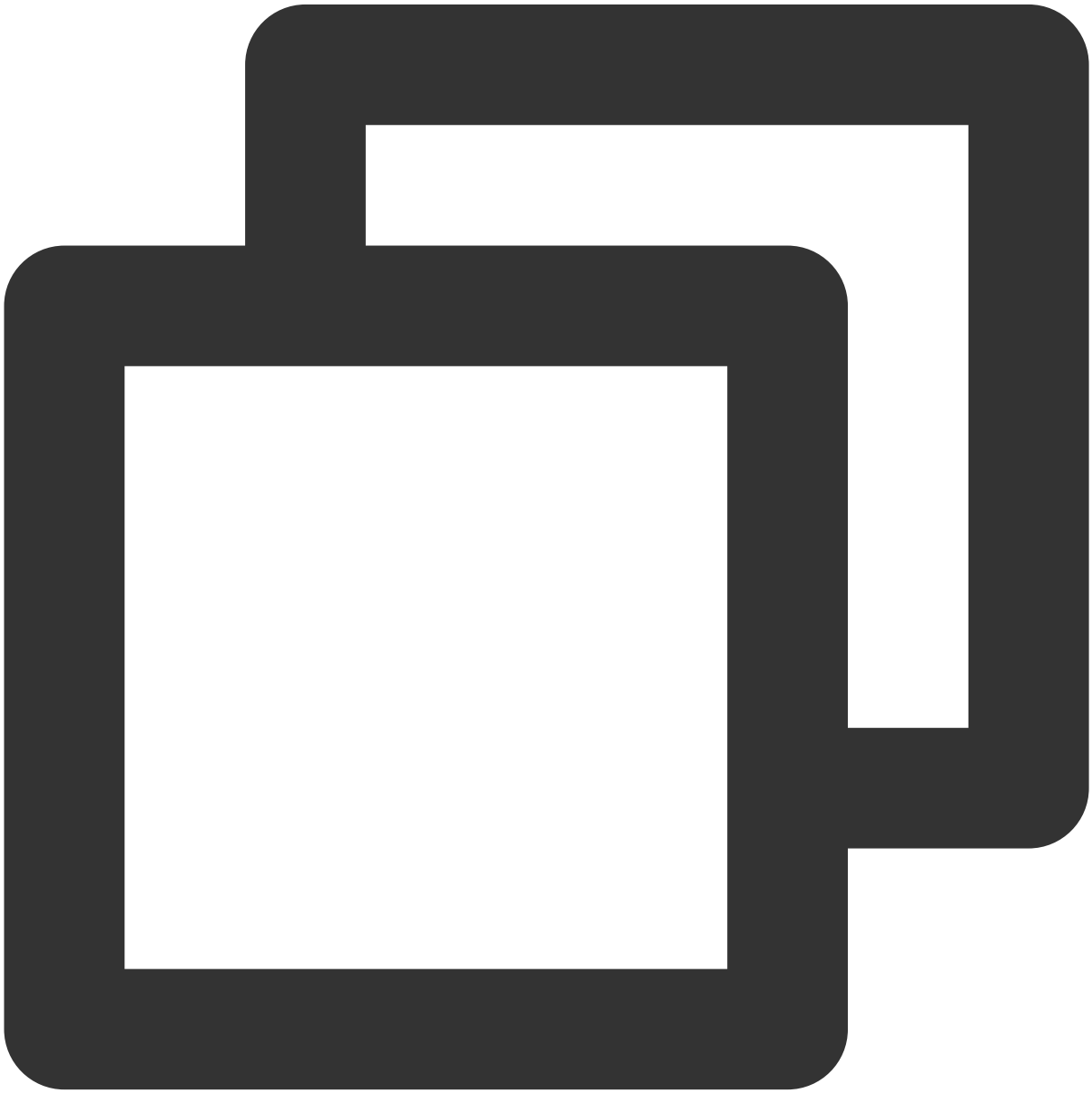
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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，优先级高于队列的回调类型	String	否
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

版权所有：腾讯云计算（北京）有限责任公司

					txt 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 zh/zh-hk/zh-tw/zh-tr 时支持：en、ar、de、es、fr、id、it、ja、it、ru、ko、

版权所有：腾讯云计算（北京）有限责任公司

					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

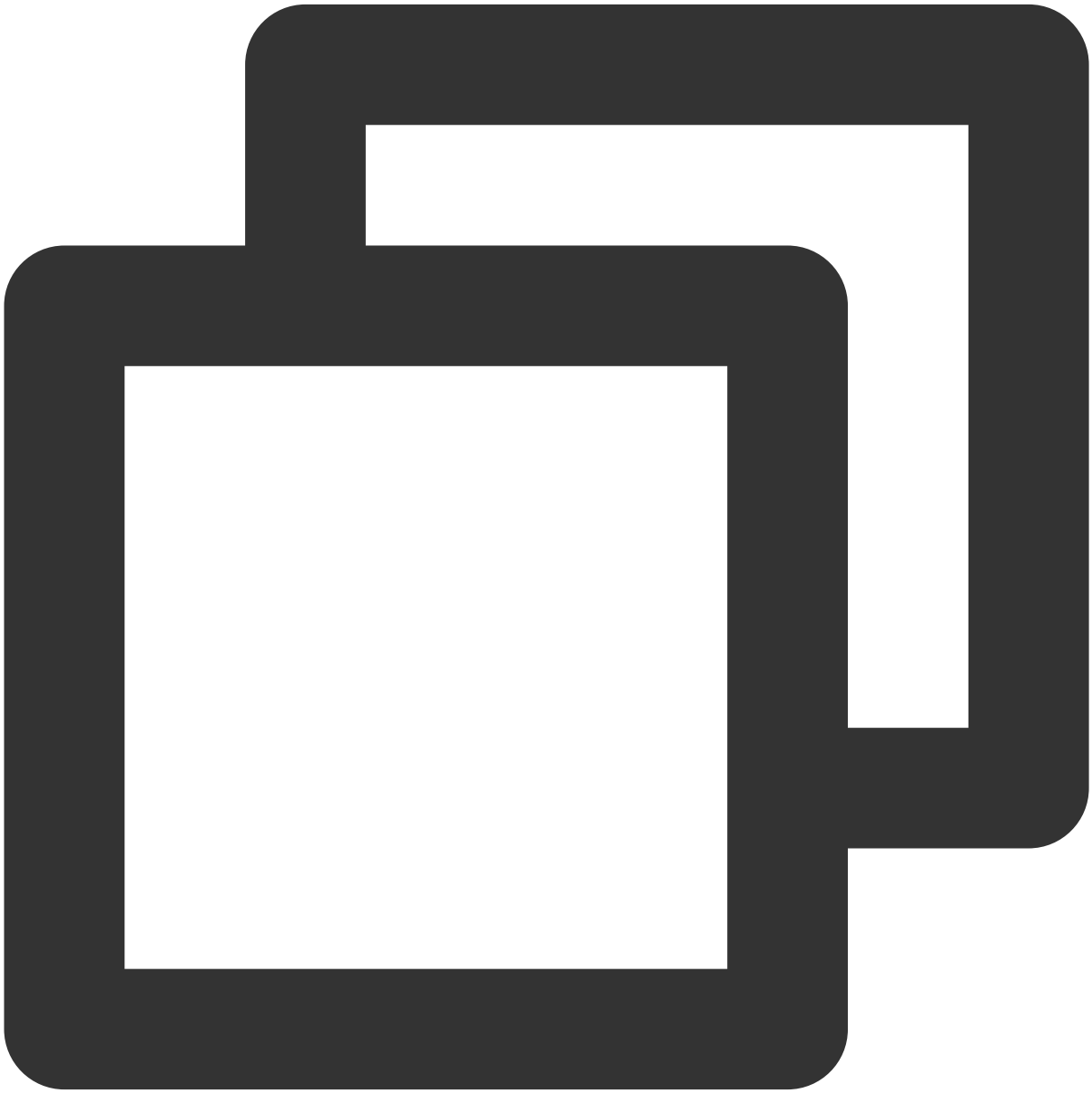
响应

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、	String

版权所有：腾讯云计算（北京）有限责任公司

		Success、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：

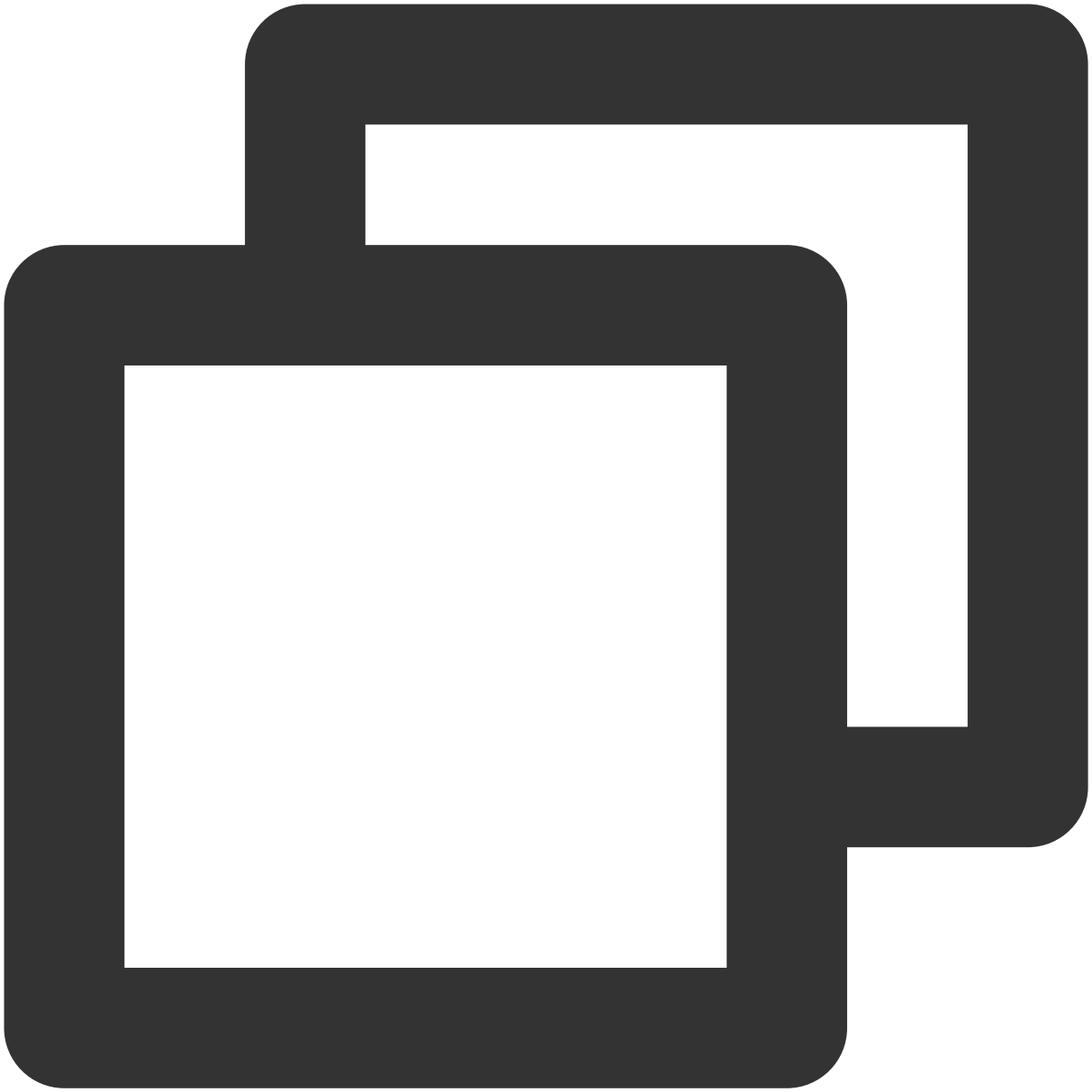
节点名称（关键字）	父节点	描述	类型
Result	Response.JobsDetail.Operation	翻译结果内容	String

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

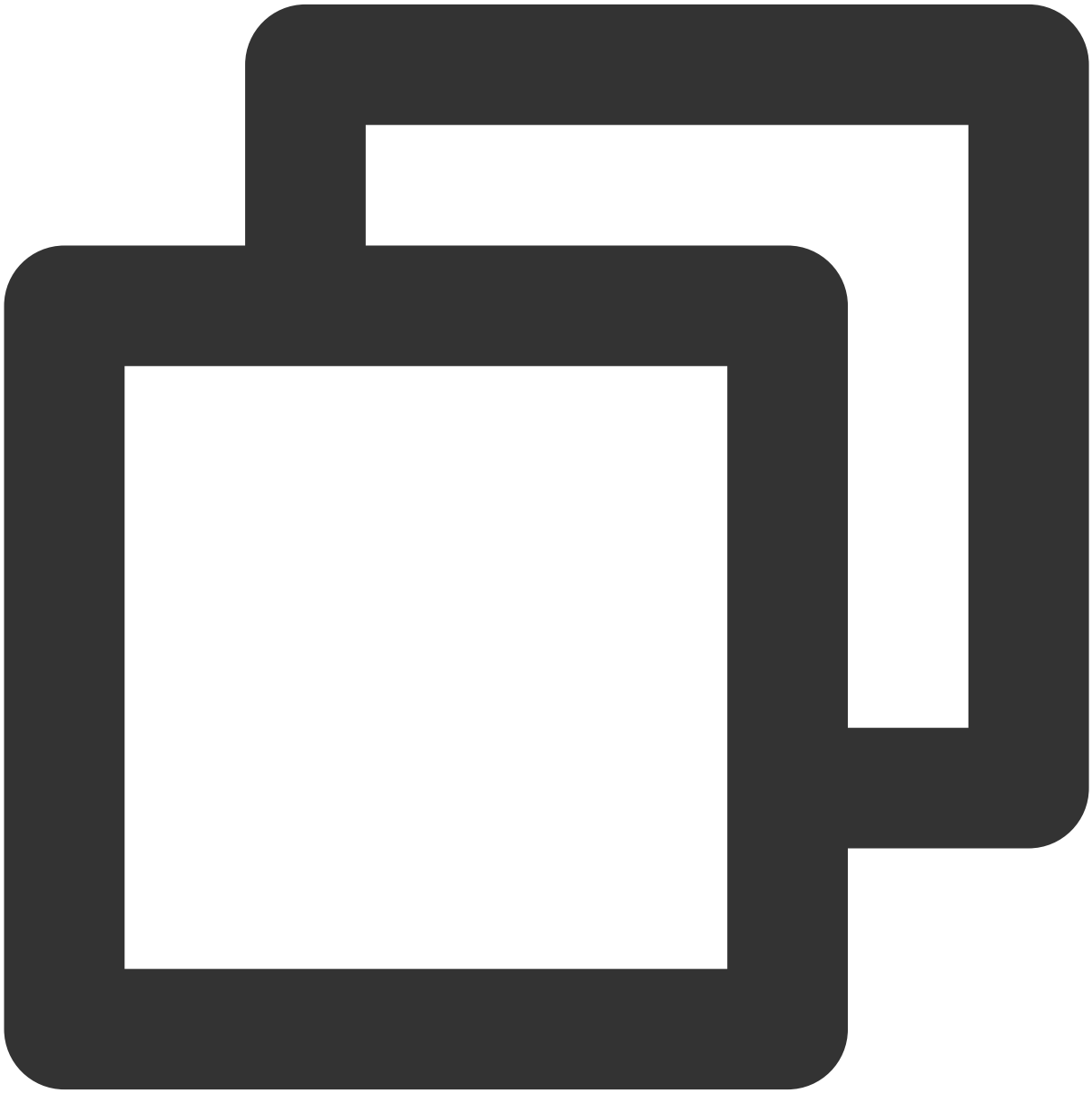


```
POST /ai_jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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

提交语音识别任务

最近更新时间：2024-01-20 16:35:12

功能描述

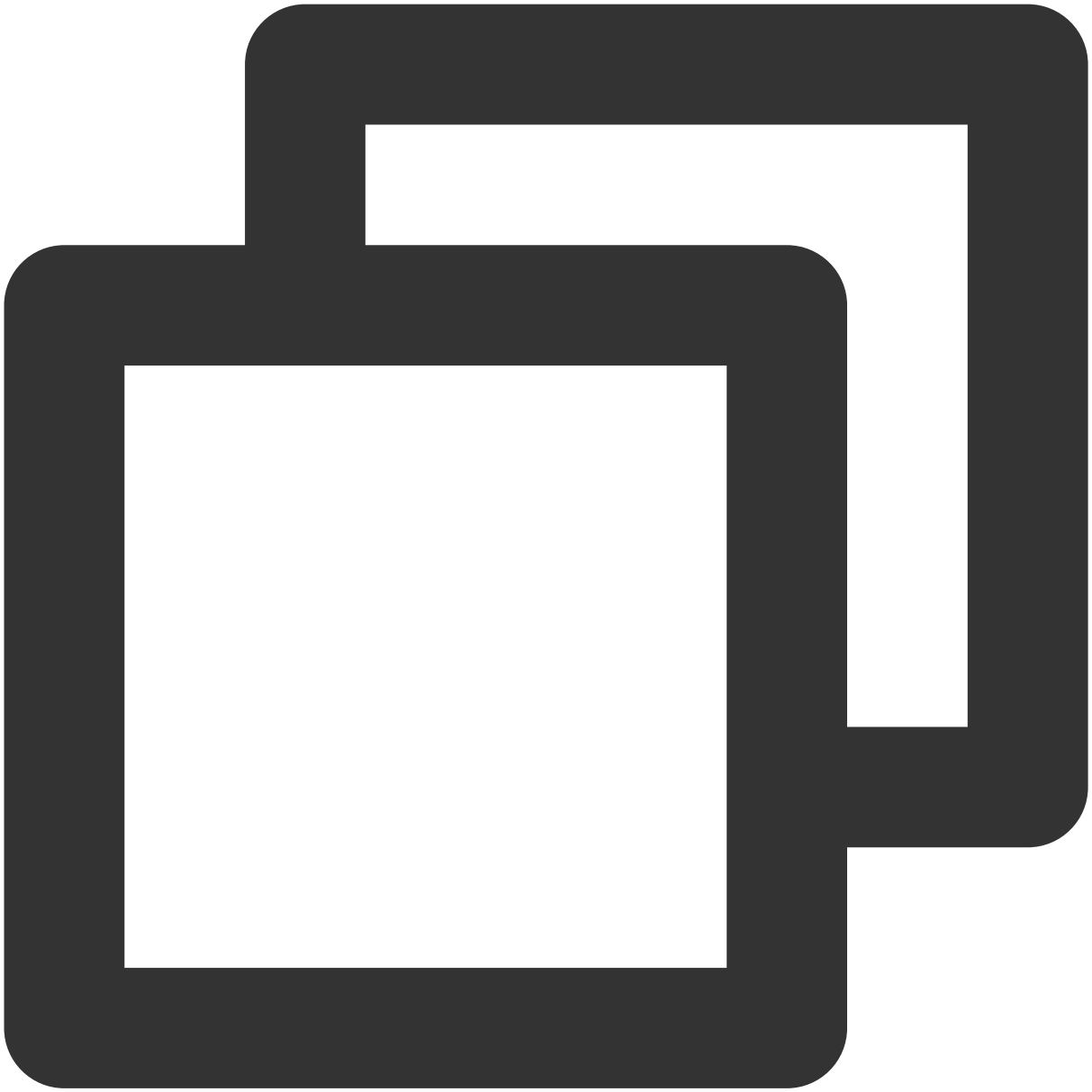
提交一个语音识别任务。

说明：

关于语音识别的限制说明，请参见 规则与限制 文档。

请求

请求示例



```
POST /asr_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：

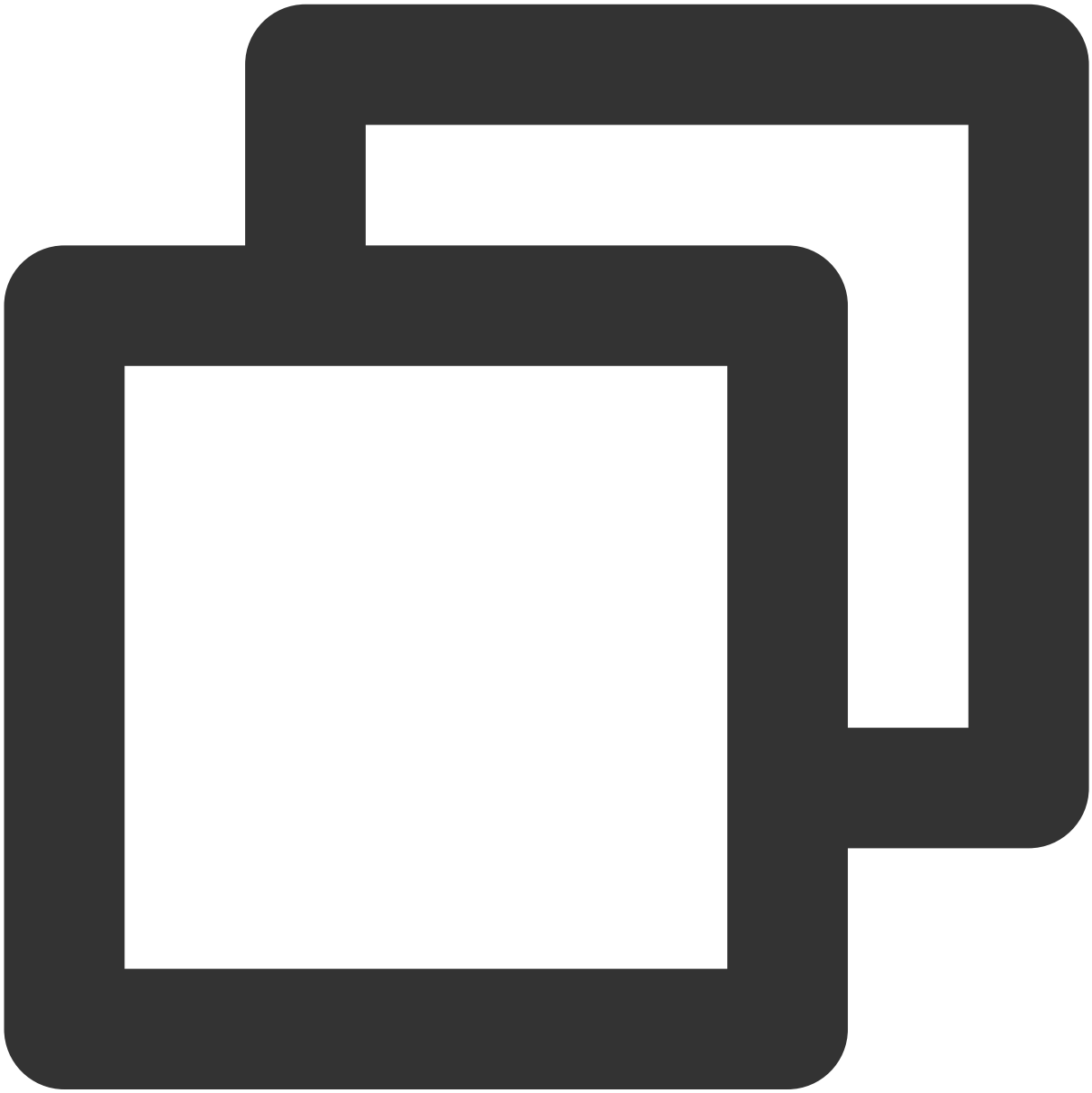
Authorization：Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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>SpeechRecognition</Tag>
    <Input>
        <Object>input/test.mp3</Object>
        <Url></Url>
    </Input>
    <Operation>
        <SpeechRecognition>
            <EngineModelType>16k_zh_video</EngineModelType>
            <ChannelNum>1</ChannelNum>
            <FilterDirty>1</ChannelNum>
            <FilterModal>1</ChannelNum>
        </SpeechRecognition>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/asr.txt</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：SpeechRecognition	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，优先级高于队列的回调类型	String	否
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
Object	Request.Input	媒体文件名	String	否
Url	Request.Input	支持公网下载的 URL，与 Object 必须有其中一个，且当两者都传入时，优先使用 Object	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
SpeechRecognition	Request.Operation	指定 语音识别 参数	Container	是
Output	Request.Operation	结果输出地址	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 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.Operation.SpeechRecognition	引擎模型类型。 电话场景： 8k_zh：电话 8k 中文普通话通用（可用于双声道音频）。 8k_zh_s：电话 8k 中文普通话话者分离（仅适用于单声道音频）。	String	是

版权所有：腾讯云计算（北京）有限责任公司

		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。		
ChannelNum	Request.Operation.SpeechRecognition	语音声道数。 1：单声道。 2：双声道（仅支持 8k_zh 引擎模型）。	Integer	是
ResTextFormat	Request.Operation.SpeechRecognition	识别结果返回形式。 0： 识别结果文本(含分段时间戳)。 1：仅支持16k 中文引擎，含识别结果详情(词时间戳列表，一般用于生成字幕场景)。	Integer	是
FilterDirty	Request.Operation.SpeechRecognition	是否过滤脏词（目前支持中文普通话引擎）。 0：不过滤脏词。 1：过滤脏词。 2：将脏词替换为 * 。默认值为0。	Integer	否
FilterModal	Request.Operation.SpeechRecognition	是否过语气词（目前支持中文普通话引擎）。 0：不过滤语气词。 1：部分过滤。 2：严格过滤 。默认值为 0。	Integer	否
ConvertNumMode	Request.Operation.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：不转换，直接输出中文数字。 1：根据场景智能转换为阿拉伯数字。默认值为1。	Integer	否

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	结果文件的名称	String	是

版权所有：腾讯云计算（北京）有限责任公司

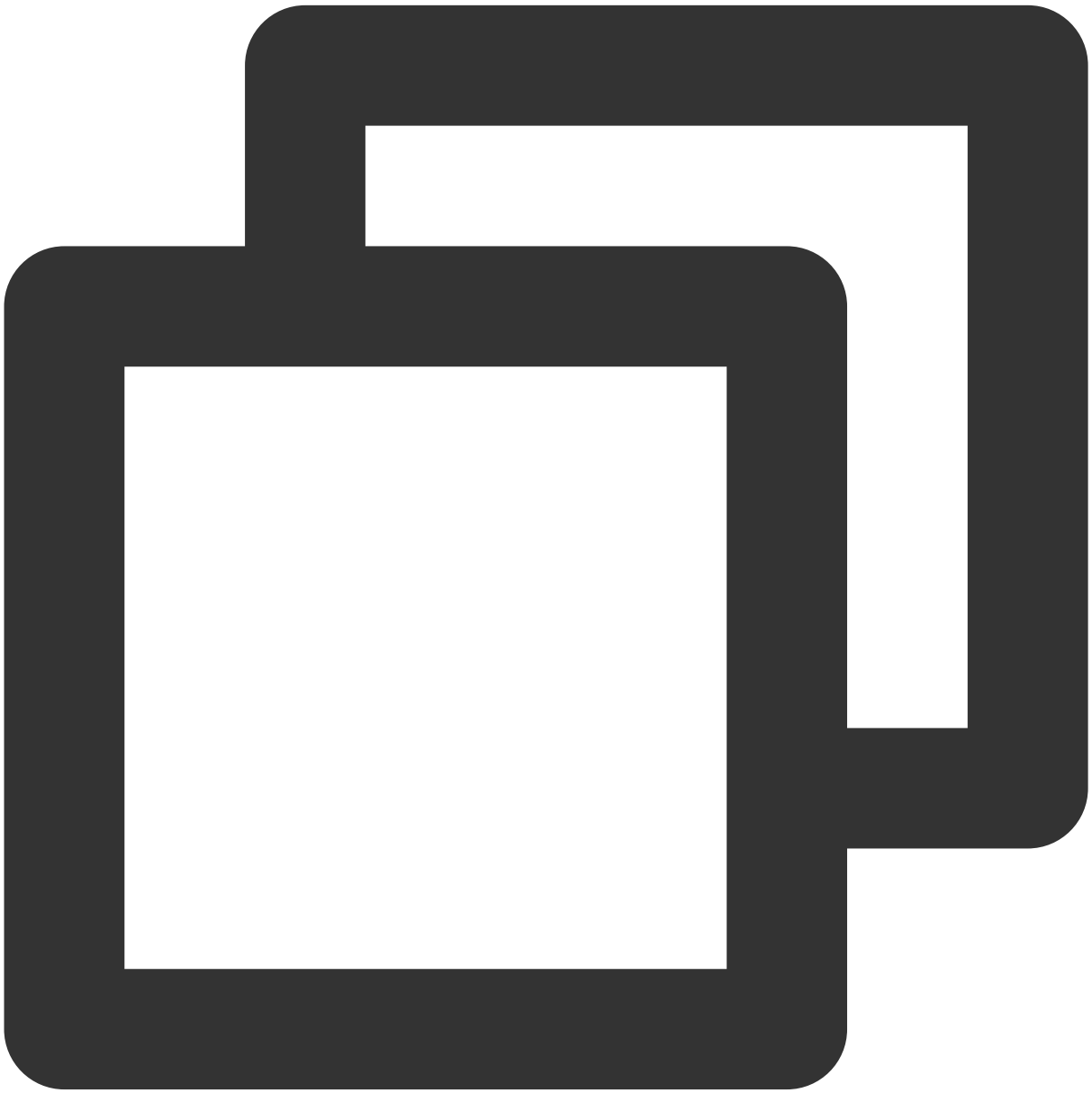
响应

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>
    <JobsDetail>
```

```
        <Code>Success</Code>
        <CreationTime>2021-08-05T15:43:50+0800</CreationTime>
        <EndTime>-</EndTime>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.mp3</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <JobId>s58ccb634149211ed84ce2b1cd7fbb14a</JobId>
        <Message/>
        <Operation>
            <JobLevel>0</JobLevel>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/asr.txt</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SpeechRecognition>
                <ChannelNum>1</ChannelNum>
                <ConvertNumMode>0</ConvertNumMode>
                <EngineModelType>16k_zh_video</EngineModelType>
                <FilterDirty>0</FilterDirty>
                <FilterModal>0</FilterModal>
                <FilterPunc>0</FilterPunc>
                <OutputFileType>txt</OutputFileType>
                <ResTextFormat>0</ResTextFormat>
                <SpeakerDiarization>0</SpeakerDiarization>
                <SpeakerNumber>0</SpeakerNumber>
            </SpeechRecognition>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <StartTime>-</StartTime>
        <State>Submitted</State>
        <Tag>SpeechRecognition</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：SpeechRecognition	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
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
SpeechRecognition	Response.JobsDetail.Operation	同请求中的 Request.Operation.SpeechRecognition	Cont
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Cont
UserData	Response.JobsDetail.Operation	透传用户信息	Strin
JobLevel	Response.JobsDetail.Operation	任务优先级	Strin
SpeechRecognitionResult	Response.JobsDetail.Operation	语音识别任务结果，没有时不返回	Cont

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 SpeechRecognitionResult 的内容：

节点名称（关键字）	父节点	描述	类型
AudioTime	Response.JobsDetail.Operation.SpeechRecognitionResult	音频时长(秒)	String
Result	Response.JobsDetail.Operation.SpeechRecognitionResult	语音识别结果	String
FlashResult	Response.JobsDetail.Operation.SpeechRecognitionResult	极速语音识别结果	Container 数组
ResultDetail	Response.JobsDetail.Operation.SpeechRecognitionResult	识别结果详情，包含每个句子中的词时间偏移，一般用于生成字幕的场景。(语音识别请求中 ResTextFormat=1 时该字段不为空) 注意：此字段可能为空，表示取不到有效值。	Container 数组

Container 节点 FlashResult 的内容：

节点名称（关键字）	父节点	描述	类型
channel_id	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult	声道标识，从0开始，对应音频声道数	Integer
text	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult	声道音频完整识别结果	String
sentence_list	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult	句	Container

		子/段落级别的识别结果列表	数组

Container 节点 sentence_list 的内容：

节点名称（关键字）	父节点	描述
text	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	句子/段落
start_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	开始时间
end_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	结束时间
speaker_id	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	说话人 ID 果设置了 speaker_d 可以按照 s 来区分说话
word_list	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	词级别的词 表

Container 节点 word_list 的内容：

节点名称（关键字）	父节点	描述
word	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list.word_list	词级别文本
start_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list.word_list	开始时间

end_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list.word_list	结束时间

Container 节点 ResultDetail 的内容：

节点名称（关键字）	父节点	描述
FinalSentence	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句最终识别结果
SliceSentence	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句中间识别结果，使用空格拆分为多个词
StartMs	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句开始时间（毫秒）
EndMs	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句结束时间（毫秒）
WordsNum	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句中词个数
SpeechSpeed	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句语速，单位：字数/秒
SpeakerId	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	声道或说话人 ID（请求中如果设置了 speaker_diarization 或者ChannelNum 为双声道，可区分说话人或声道）
Words	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句中词详情

Container 节点 Words 的内容：

节点名称（关键字）	父节点	描述	类型
Word	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail.Words	词文本	String

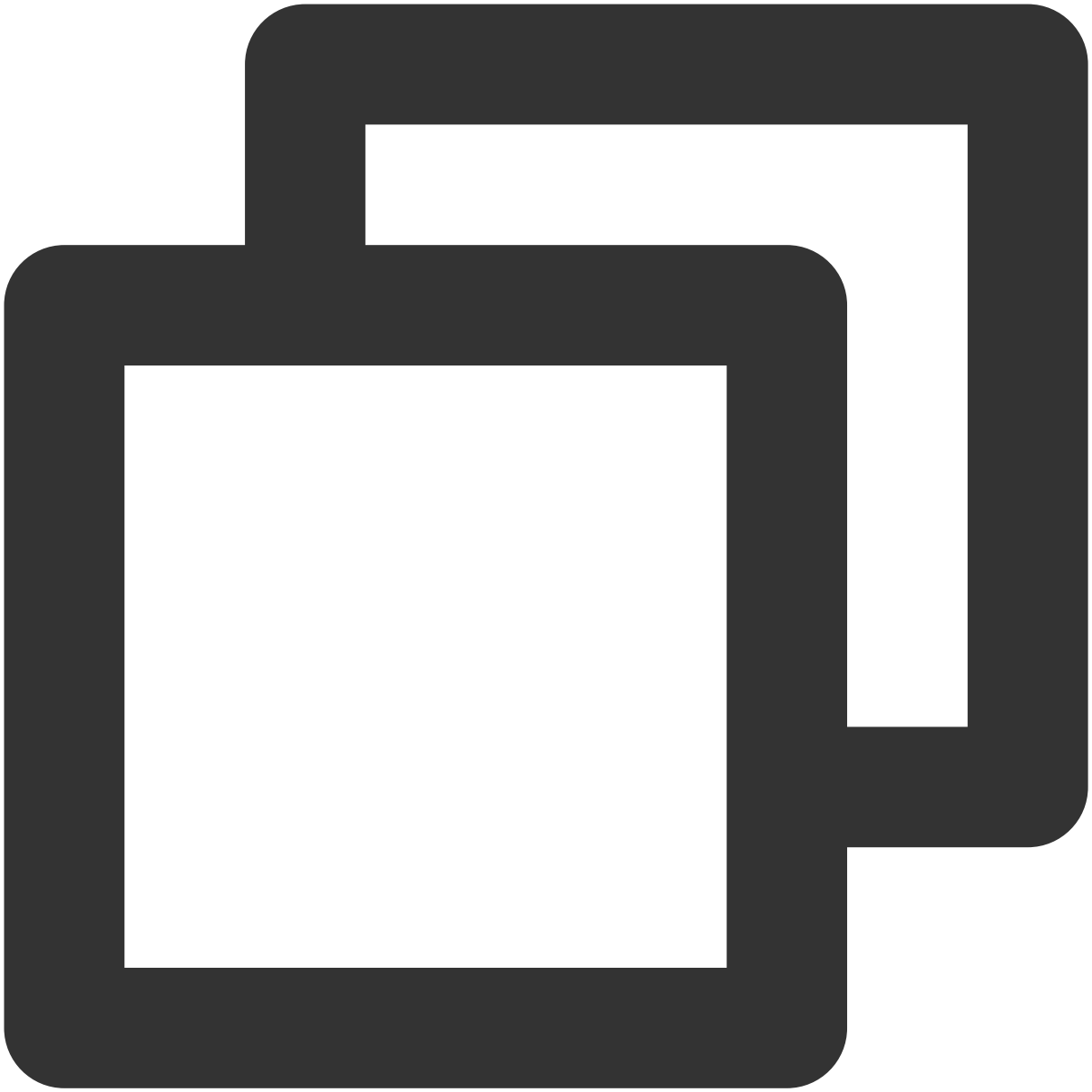
OffsetStartMs	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail.Words	在句子中的开始时间偏移量	String
OffsetEndMs	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail.Words	在句子中的结束时间偏移量	String

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

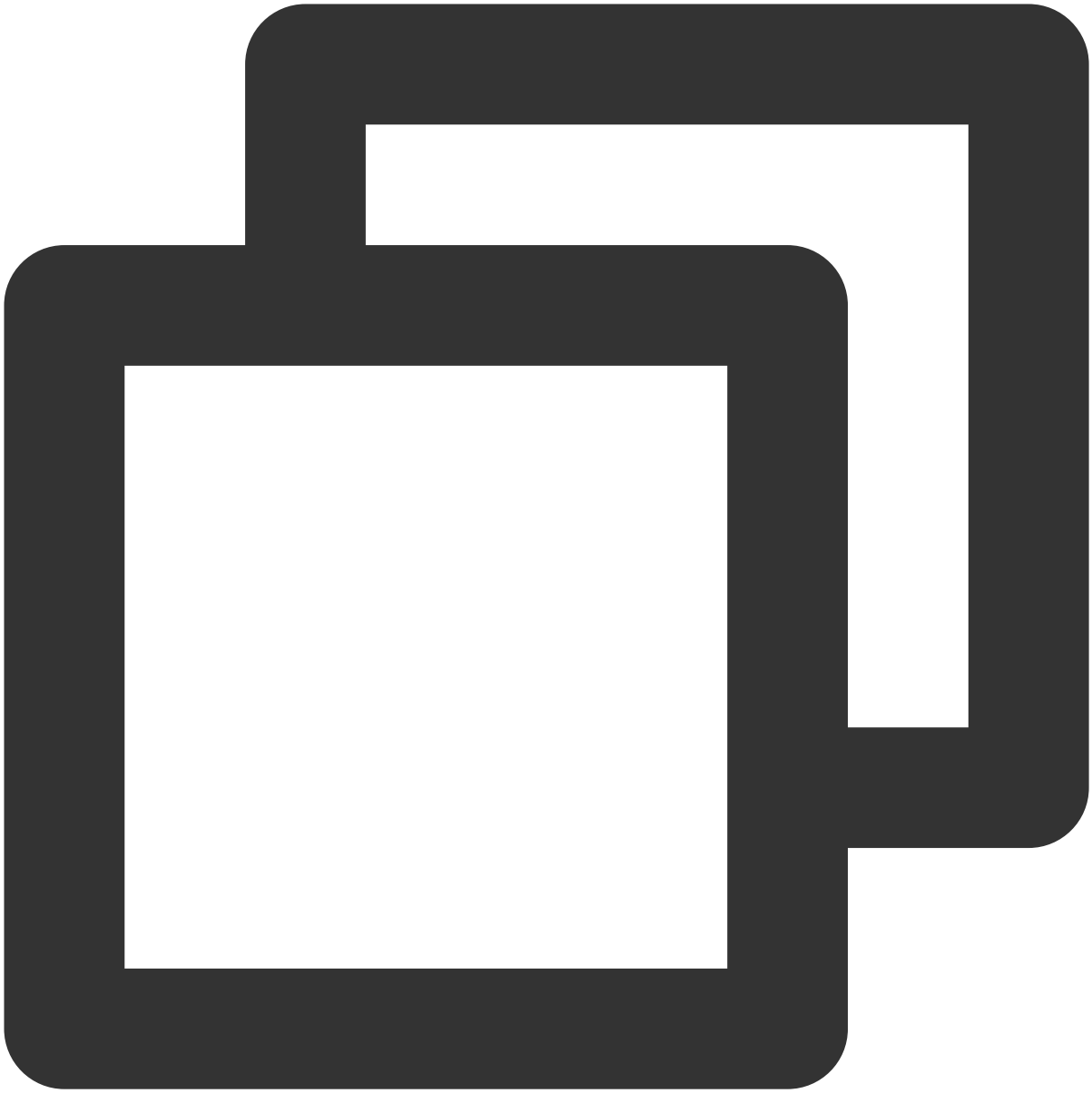


```
POST /asr_jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SpeechRecognition</Tag>
    <Input>
        <Object>input/test.mp3</Object>
    </Input>
```

```
    <Operation>
        <SpeechRecognition>
            <EngineModelType>16k_zh_video</EngineModelType>
            <ChannelNum>1</ChannelNum>
            <FilterDirty>1</ChannelNum>
            <FilterModal>1</ChannelNum>
        </SpeechRecognition>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/asr.txt</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>
```

```
        <CreationTime>2021-08-05T15:43:50+0800</CreationTime>
        <EndTime>-</EndTime>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.mp3</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <JobId>s58ccb634149211ed84ce2b1cd7fbb14a</JobId>
        <Message/>
        <Operation>
            <JobLevel>0</JobLevel>
            <Output>
                <Bucket>test-1234567890</Bucket>
                <Object>output/asr.txt</Object>
                <Region>ap-chongqing</Region>
            </Output>
            <SpeechRecognition>
                <ChannelNum>1</ChannelNum>
                <ConvertNumMode>0</ConvertNumMode>
                <EngineModelType>16k_zh_video</EngineModelType>
                <FilterDirty>0</FilterDirty>
                <FilterModal>0</FilterModal>
                <FilterPunc>0</FilterPunc>
                <OutputFileType>txt</OutputFileType>
                <ResTextFormat>0</ResTextFormat>
                <SpeakerDiarization>0</SpeakerDiarization>
                <SpeakerNumber>0</SpeakerNumber>
            </SpeechRecognition>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <StartTime>-</StartTime>
        <State>Submitted</State>
        <Tag>SpeechRecognition</Tag>
    </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

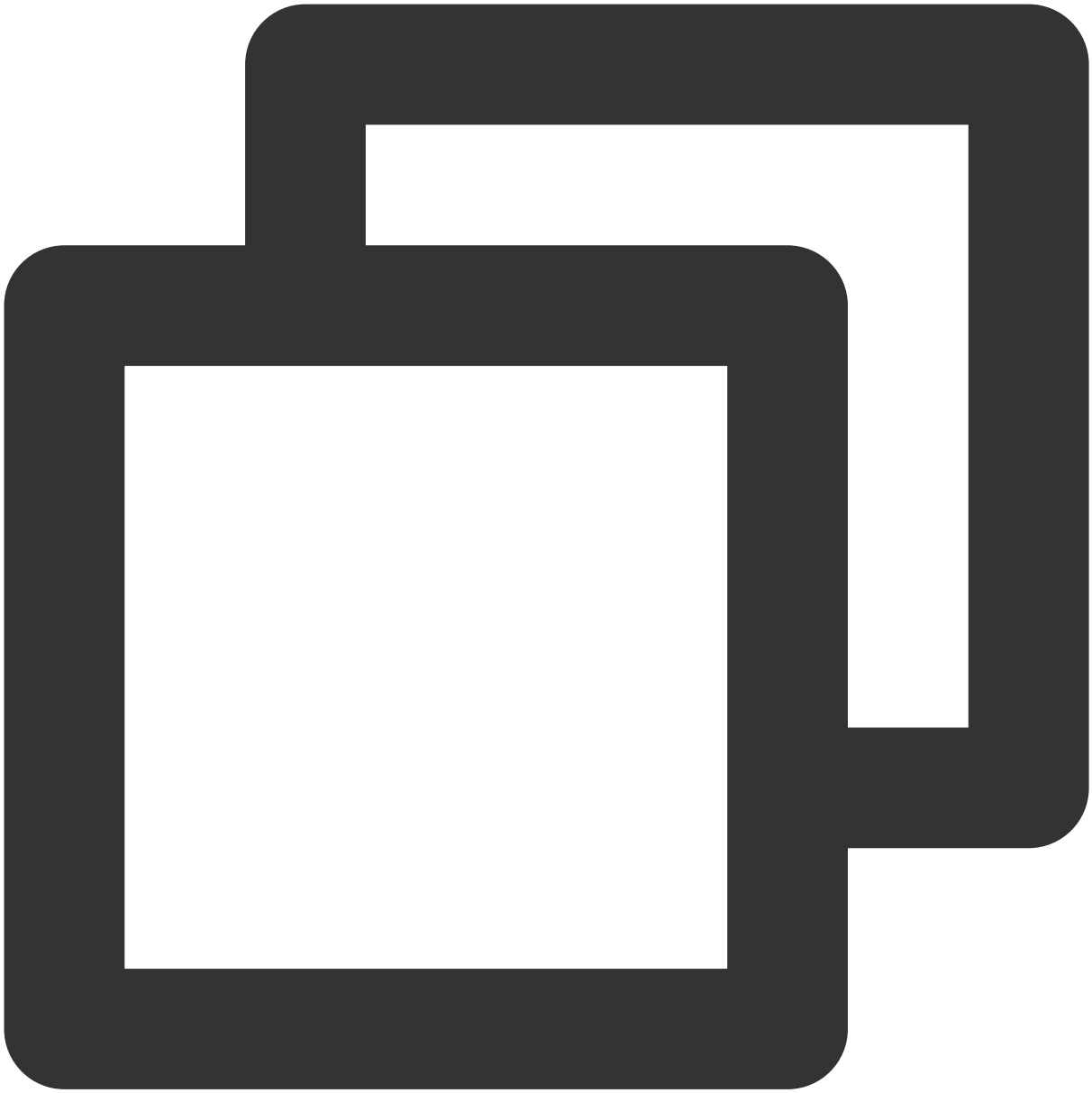
最近更新时间：2024-01-20 16:35:12

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：

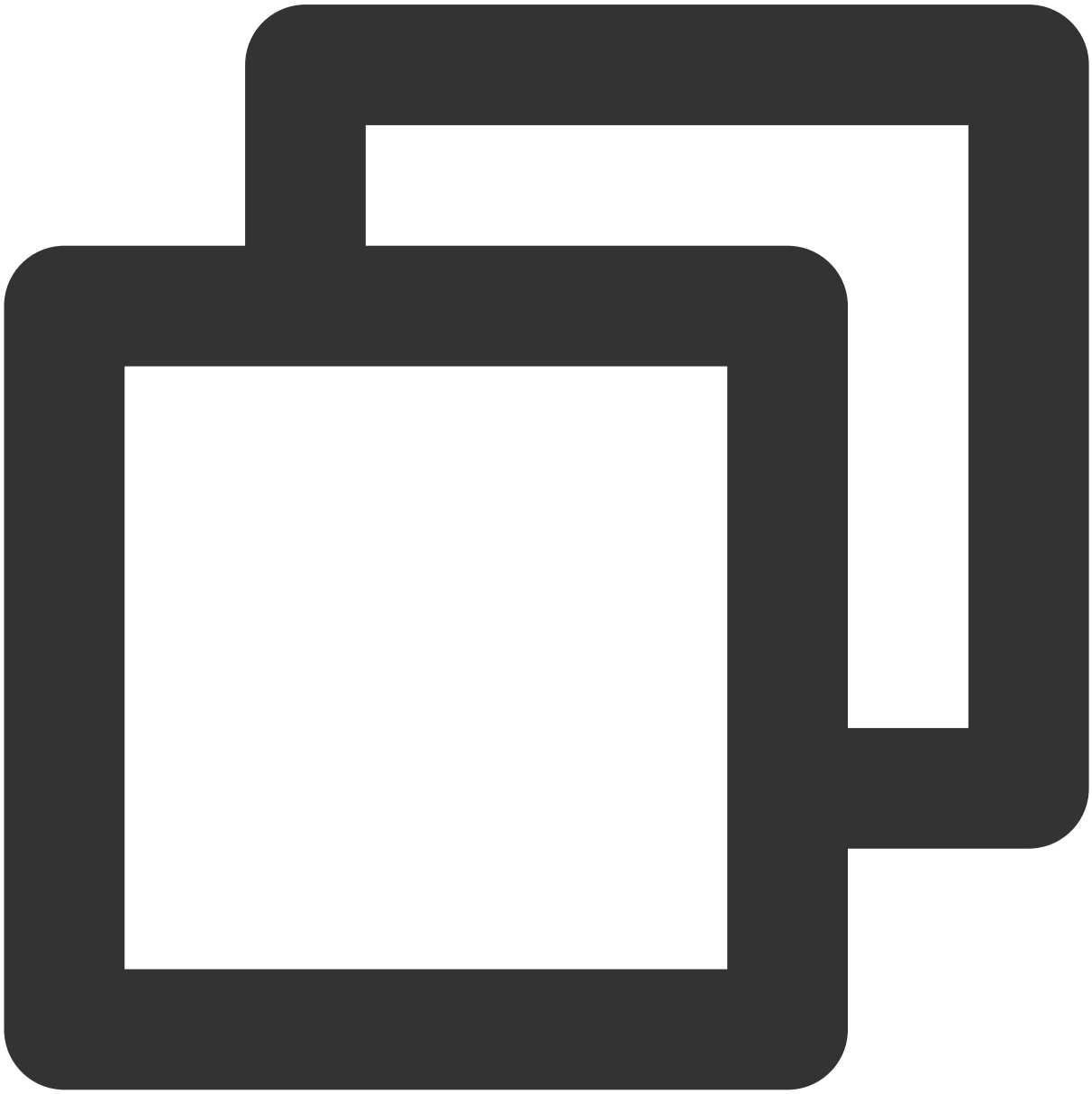
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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
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

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	媒体文件名	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
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

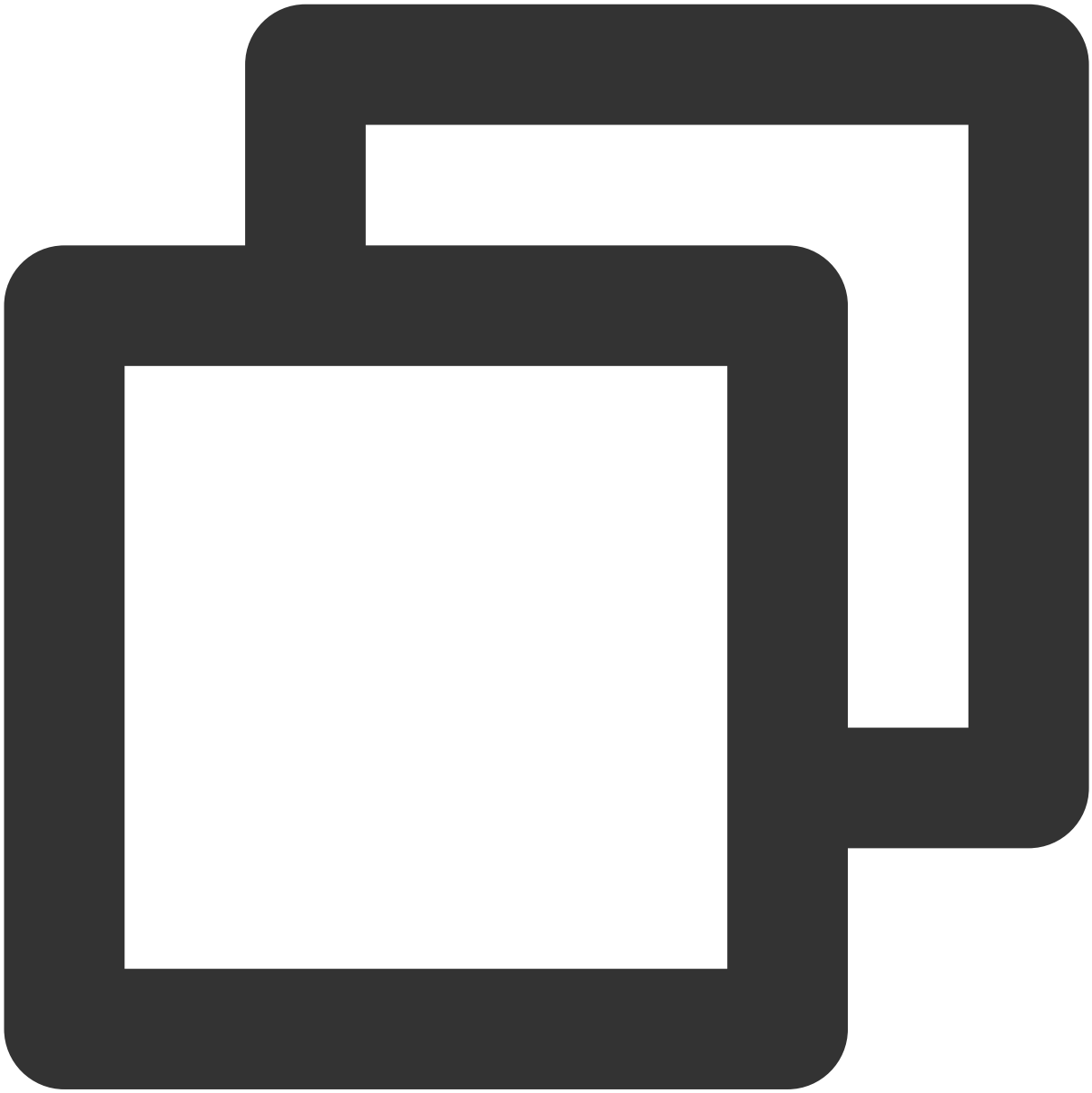
响应

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

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

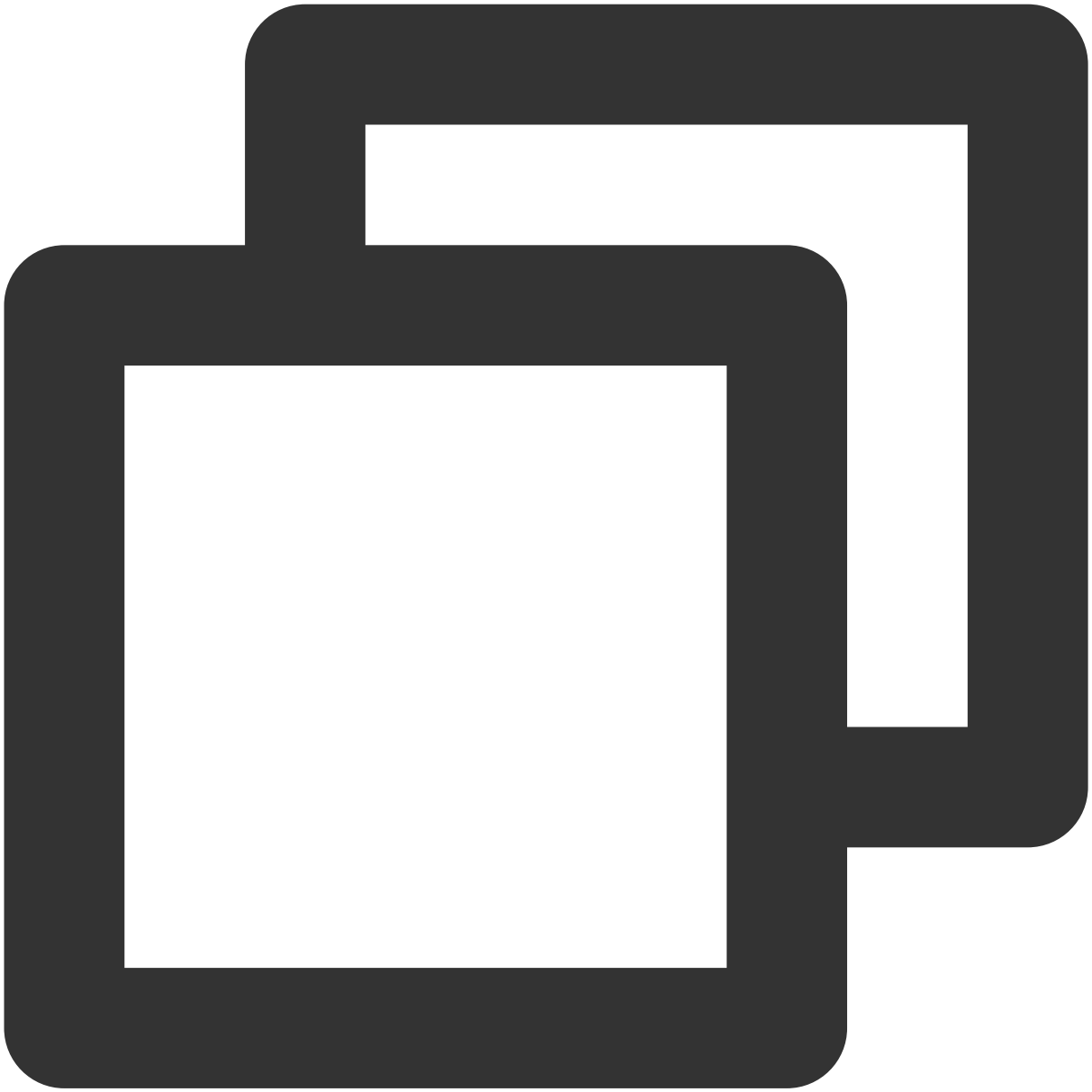
DG	副语素
MG	数词性语素
NG	名语素
QG	量语素
RG	代语素
TG	动语素
VG	量语素
YG	语气词语素
ZG	状态词语素
UNK	未知词性

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

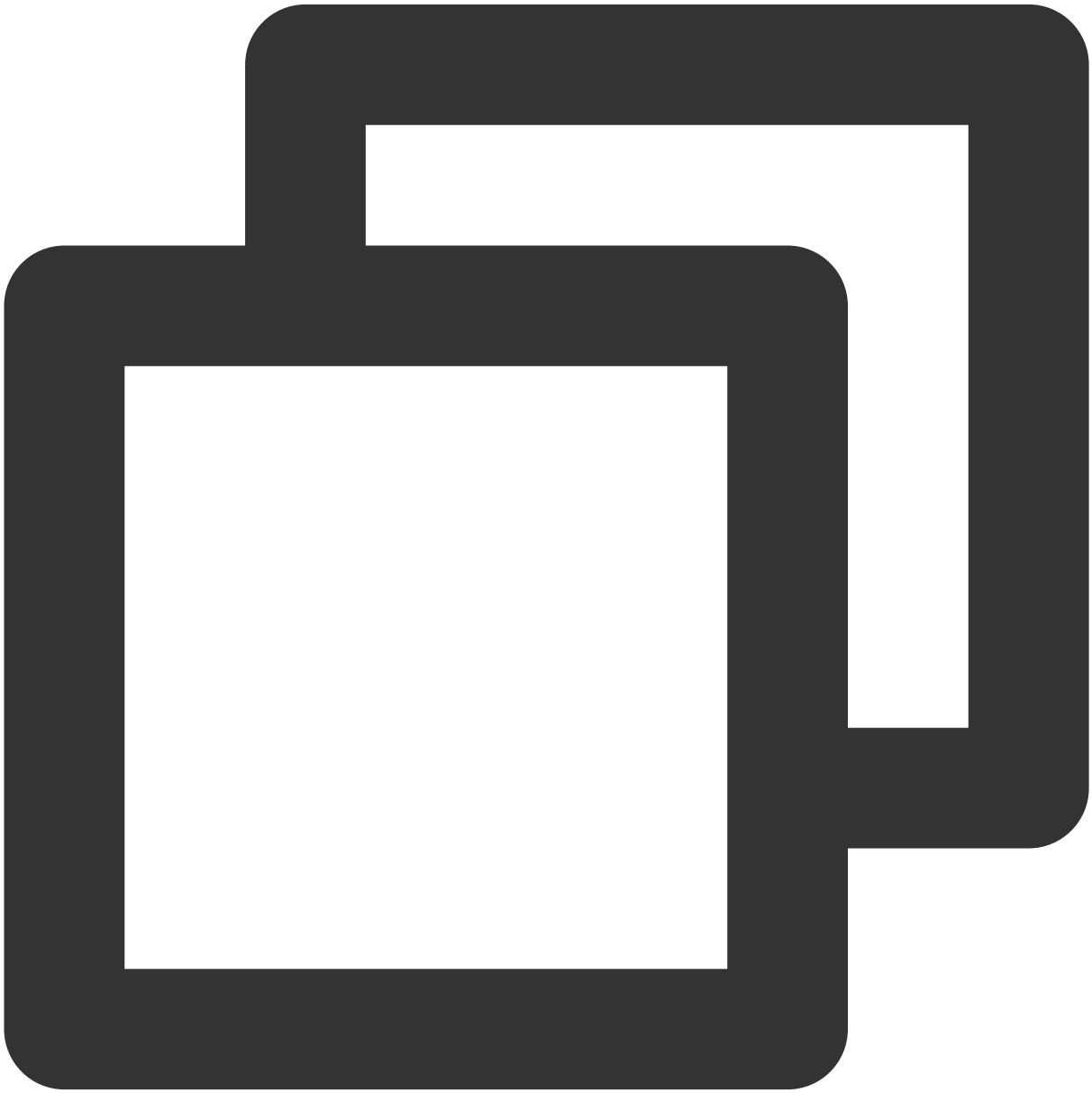


```
POST /ai_jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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

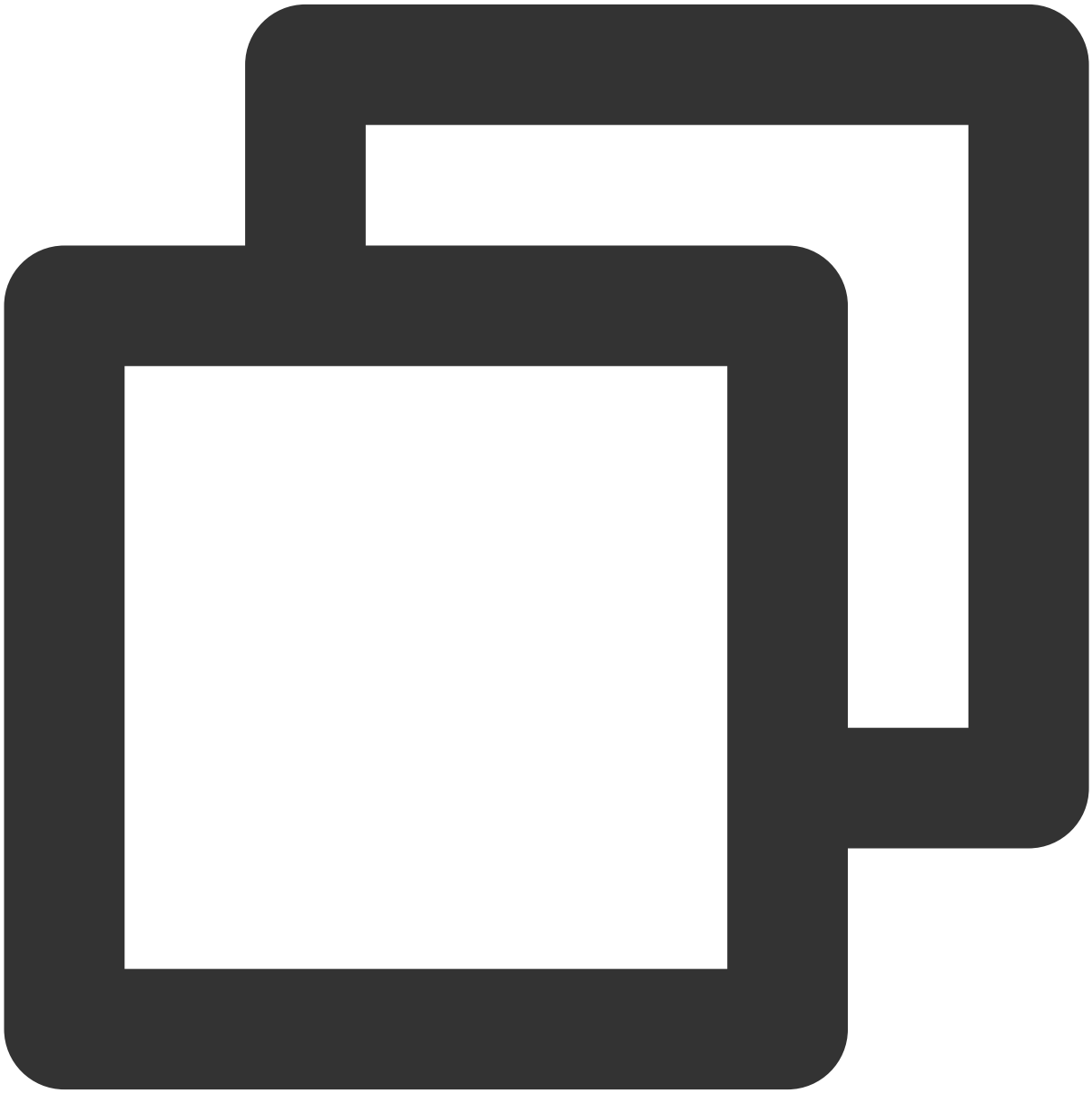
最近更新时间：2024-01-20 16:35:12

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 /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

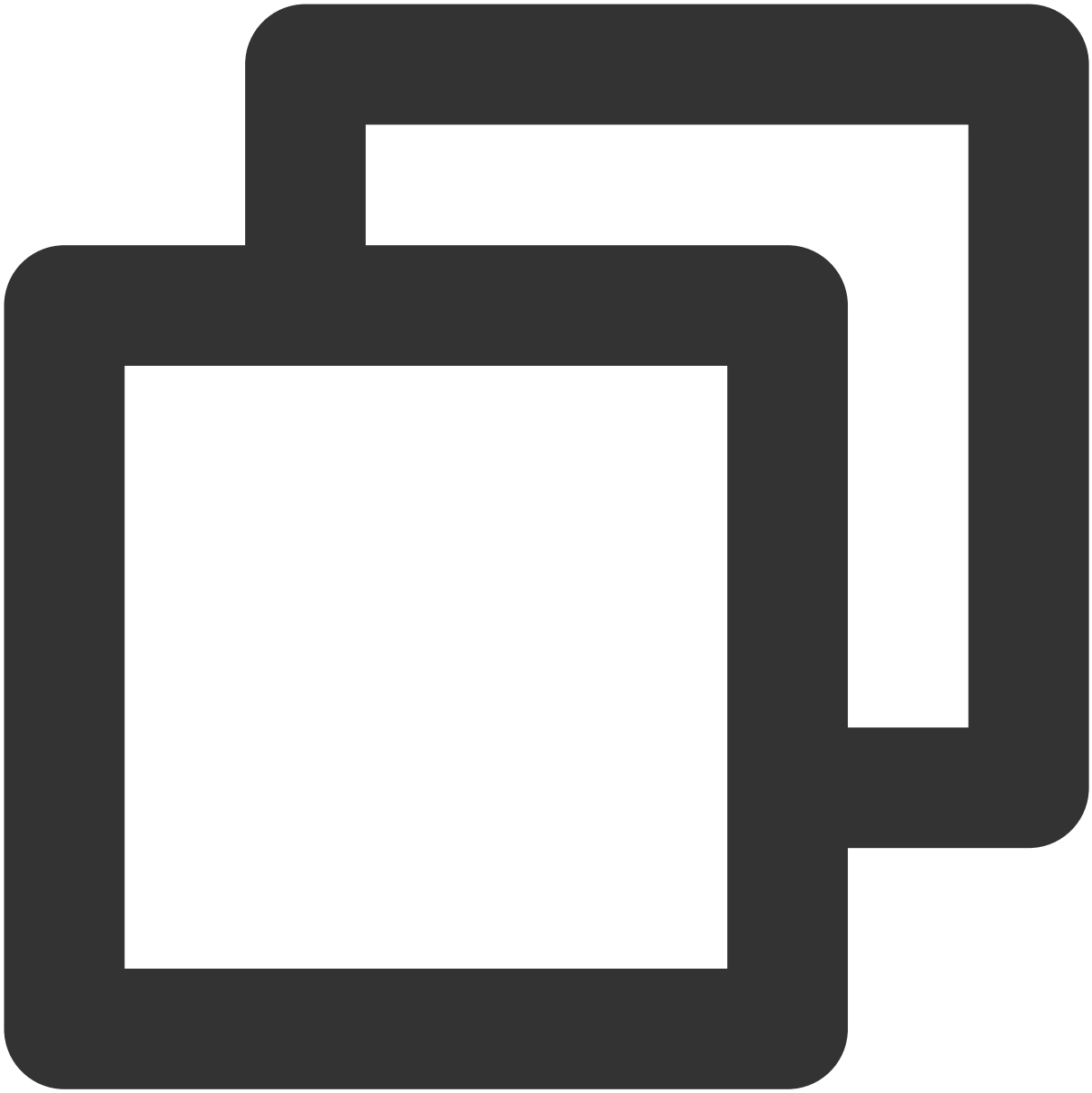
该请求无响应体。

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。

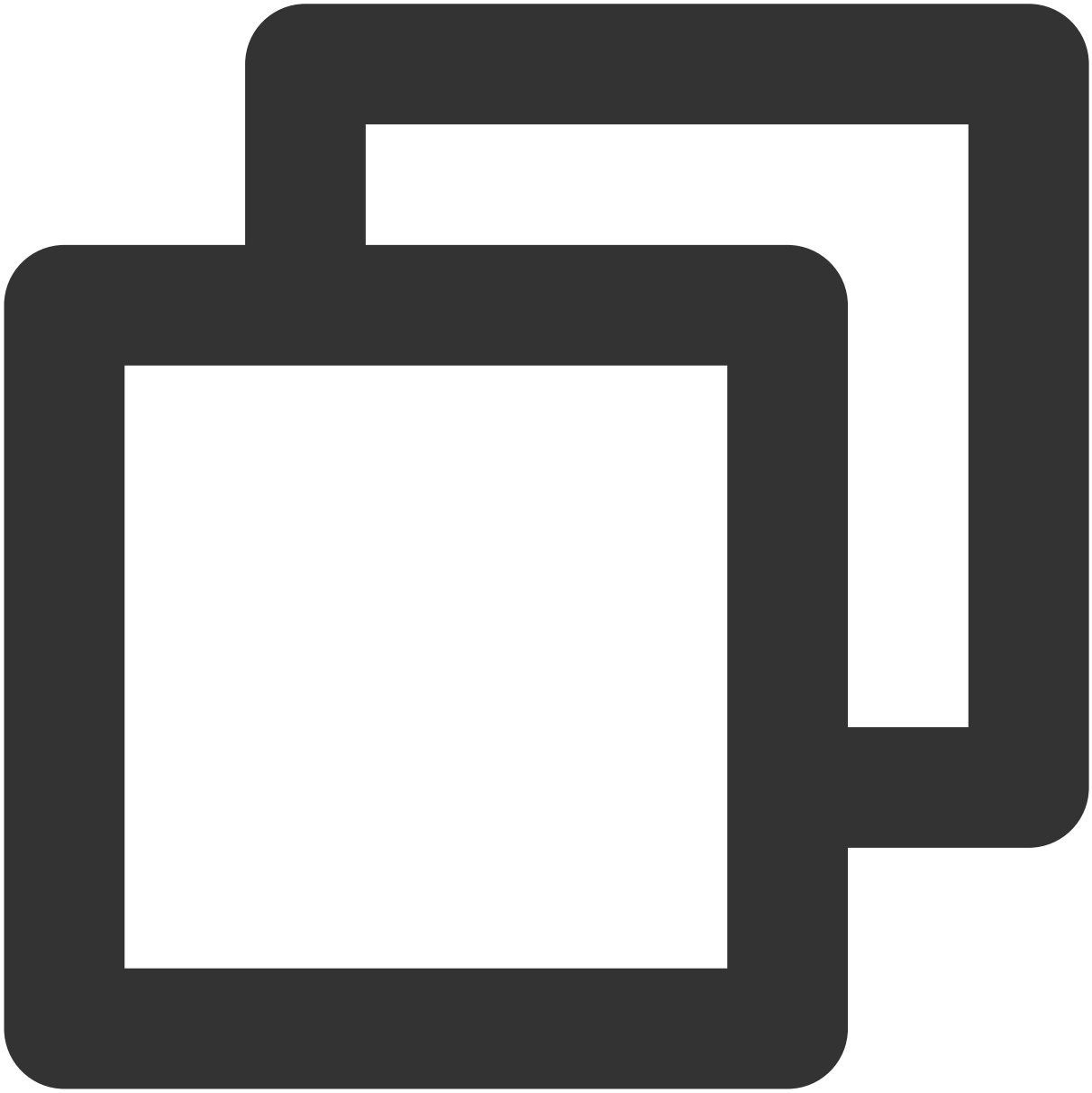
实际案例

请求



```
PUT /ai_jobs/a632d4d3429a811ed8a5fa3d8277c400f?cancel HTTP/1.1
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-Length: 0
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=
```

查询内容识别任务

查询内容识别任务

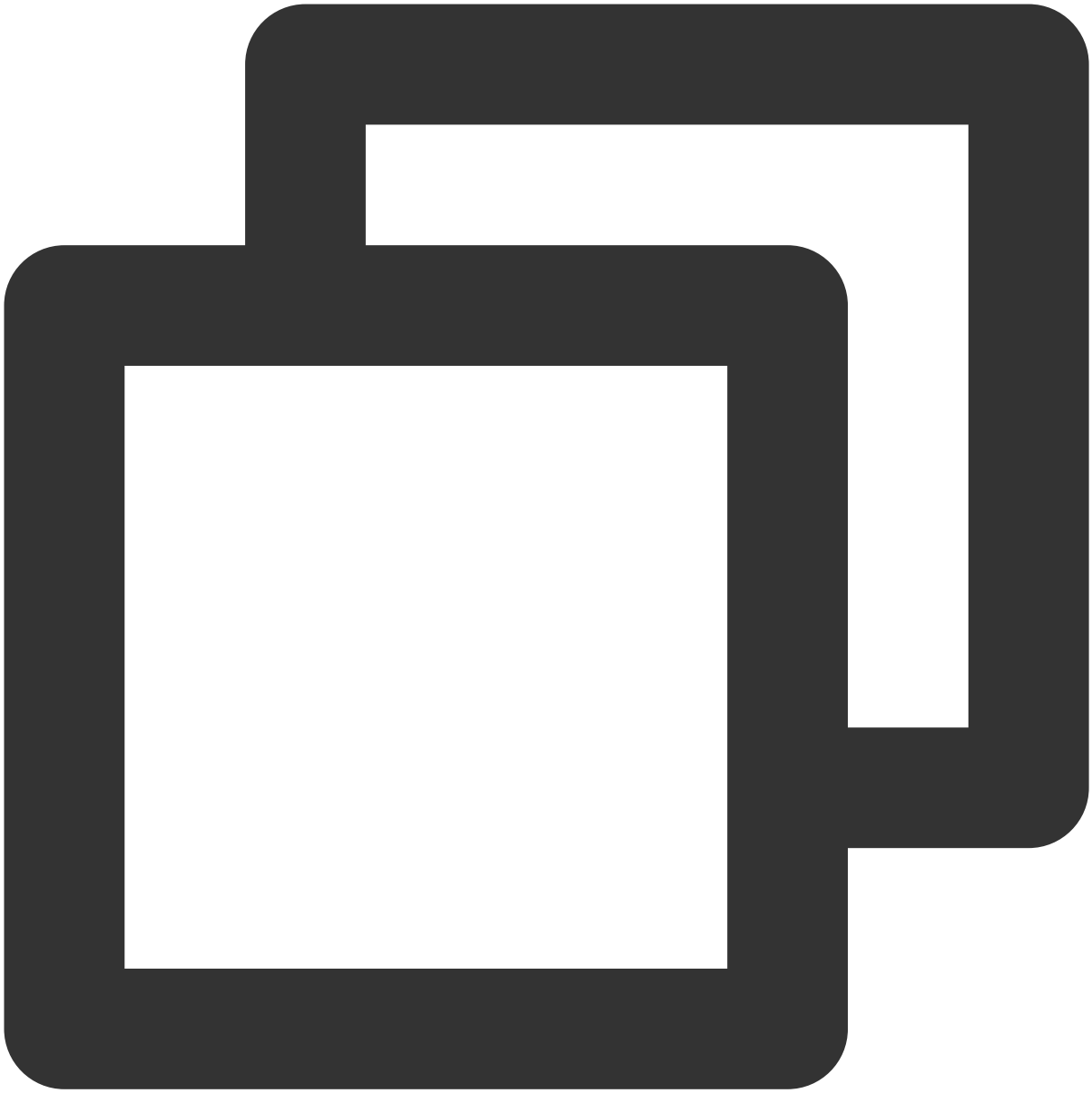
最近更新时间：2024-01-20 16:35:12

功能描述

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。

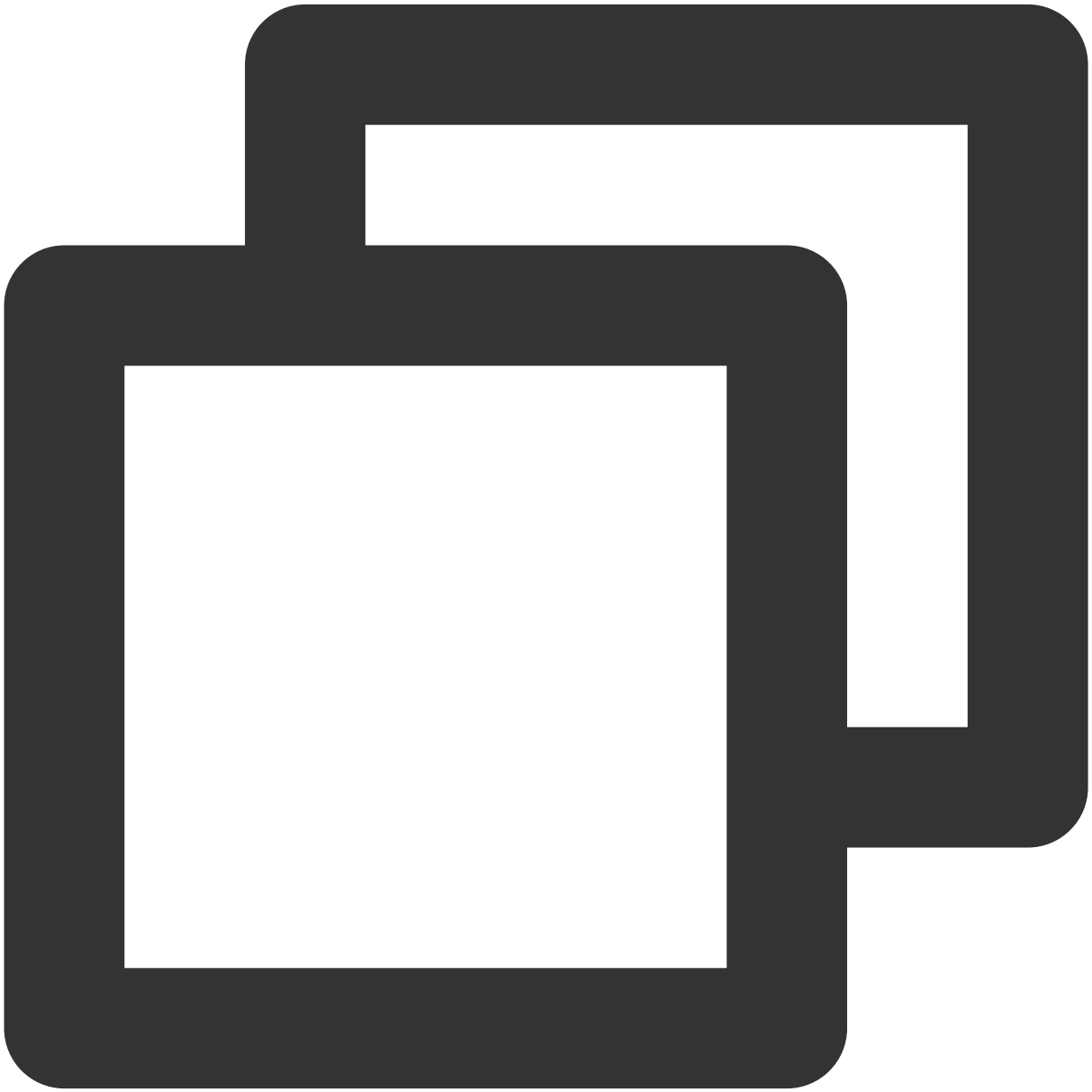
响应

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

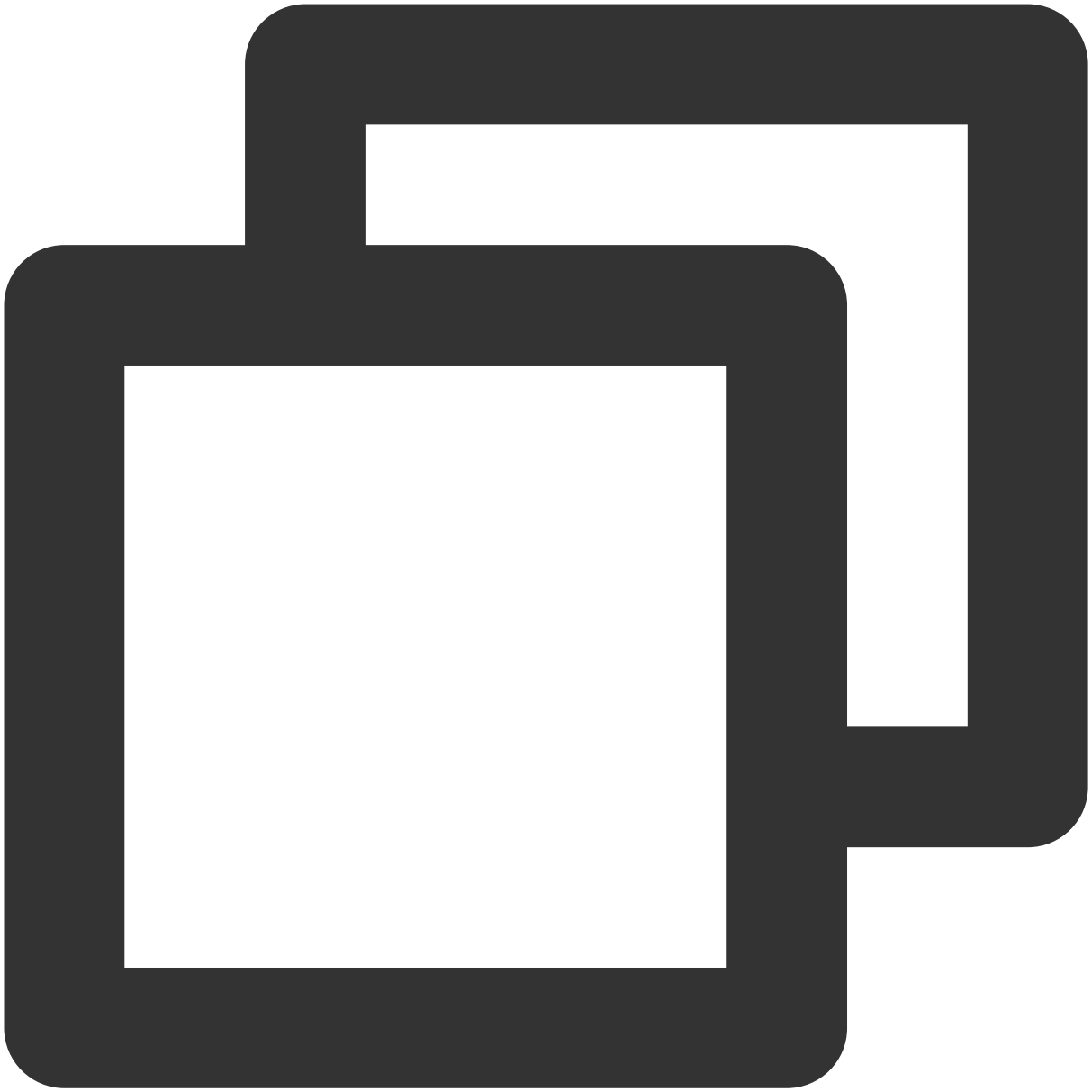
分词

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。

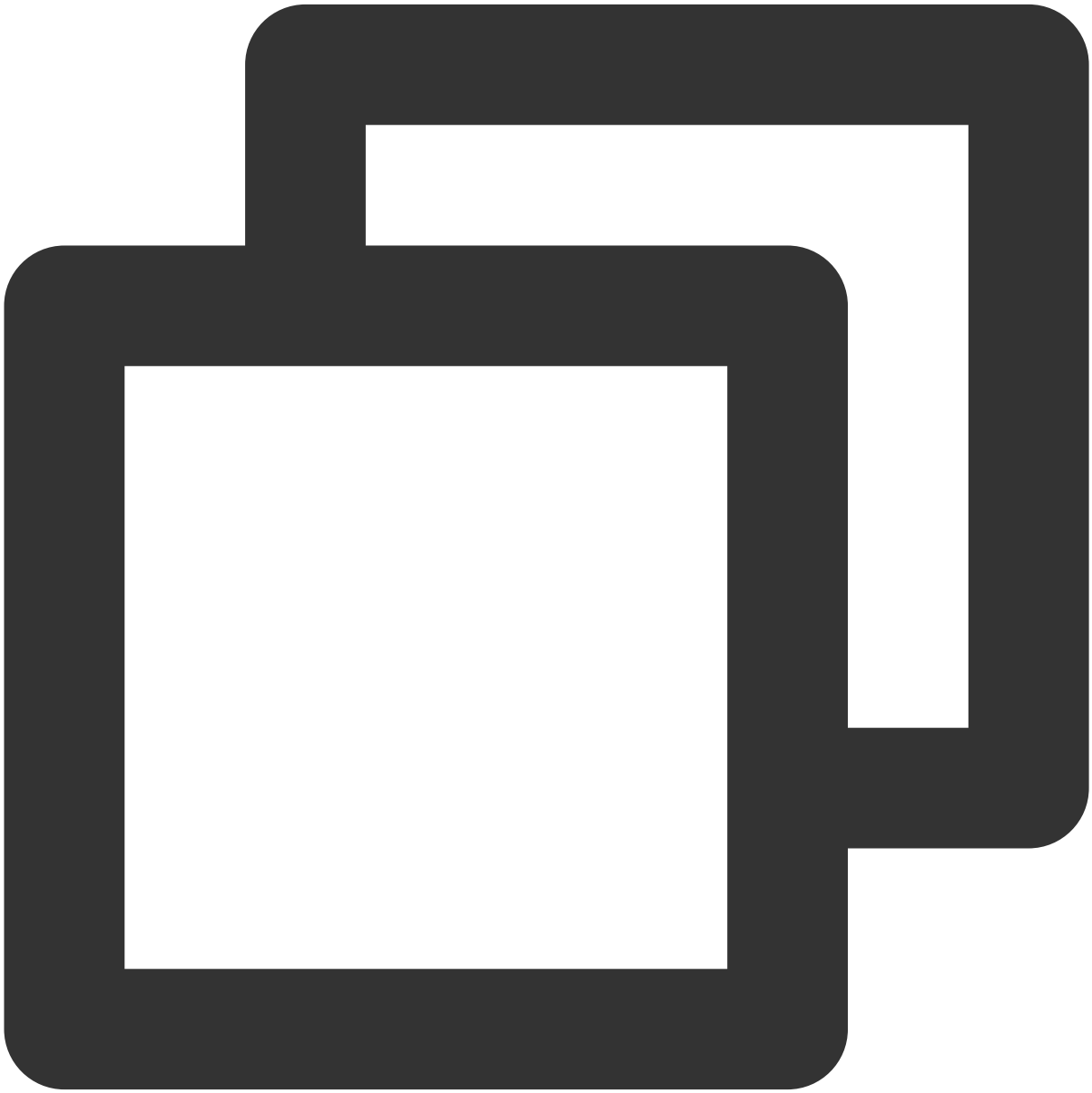
实际案例

请求



```
GET /ai_jobs/a8d121820f5e411ec926ef19d53ba9c6f HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.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: NjMxMDJhYTRfMThhYTk0MGFfYmU1Nl8xNTk=

<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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

批量拉取内容识别任务

最近更新时间：2024-01-20 16:35:12

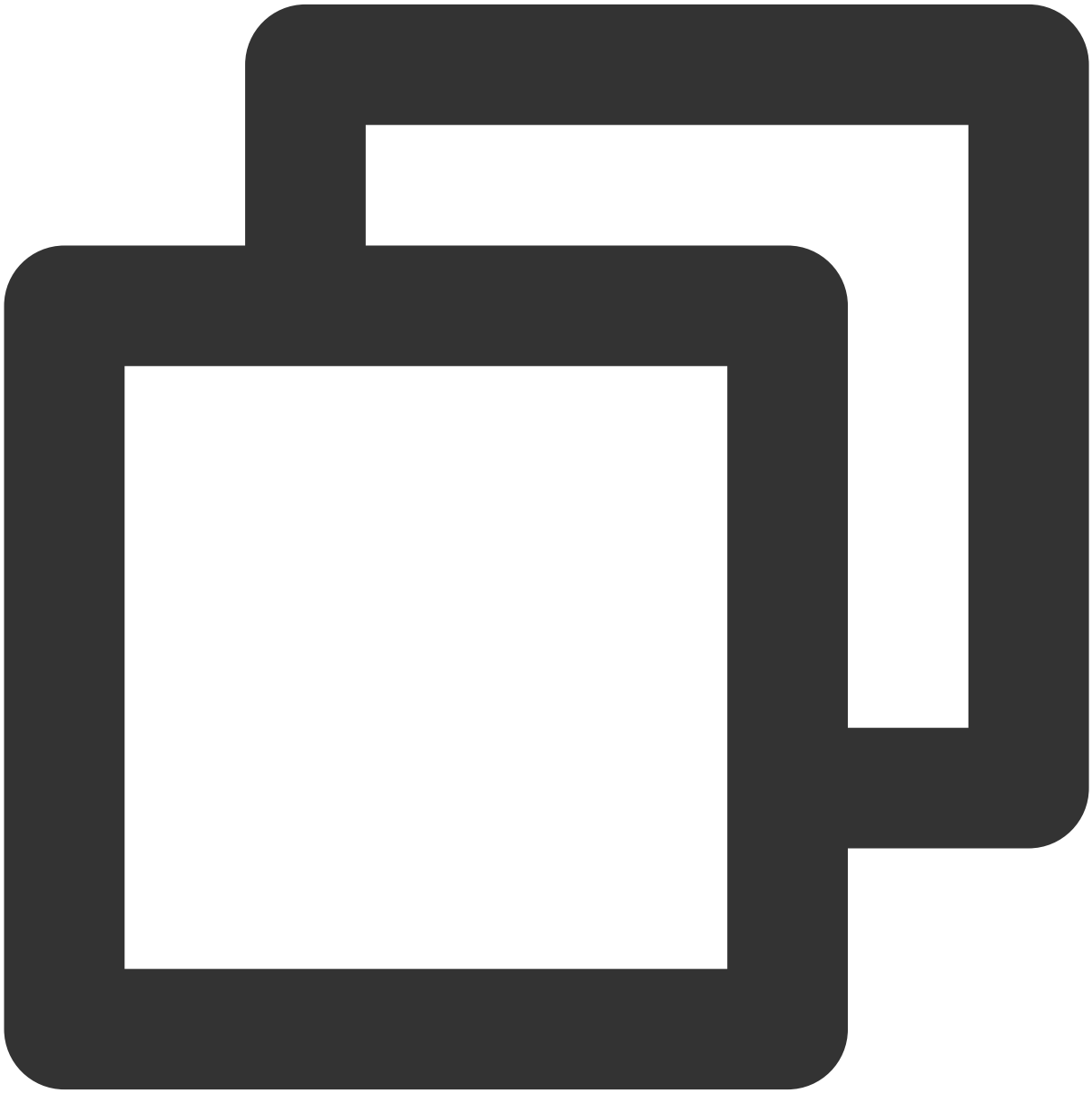
功能描述

拉取符合条件的任务。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
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
翻译	Translation
语音识别	SpeechRecognition
分词	WordsGeneralize

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。

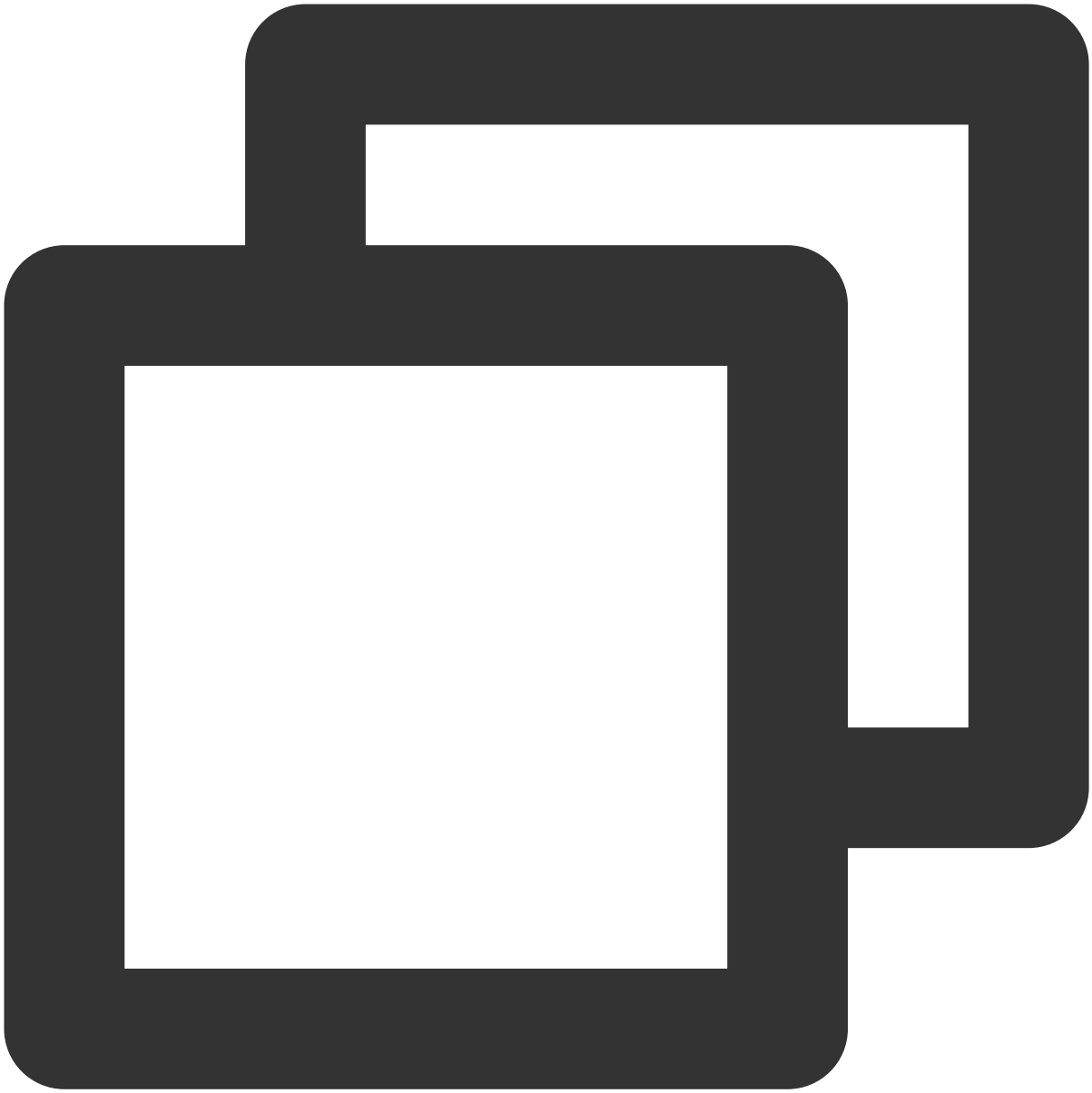
响应

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：

翻译

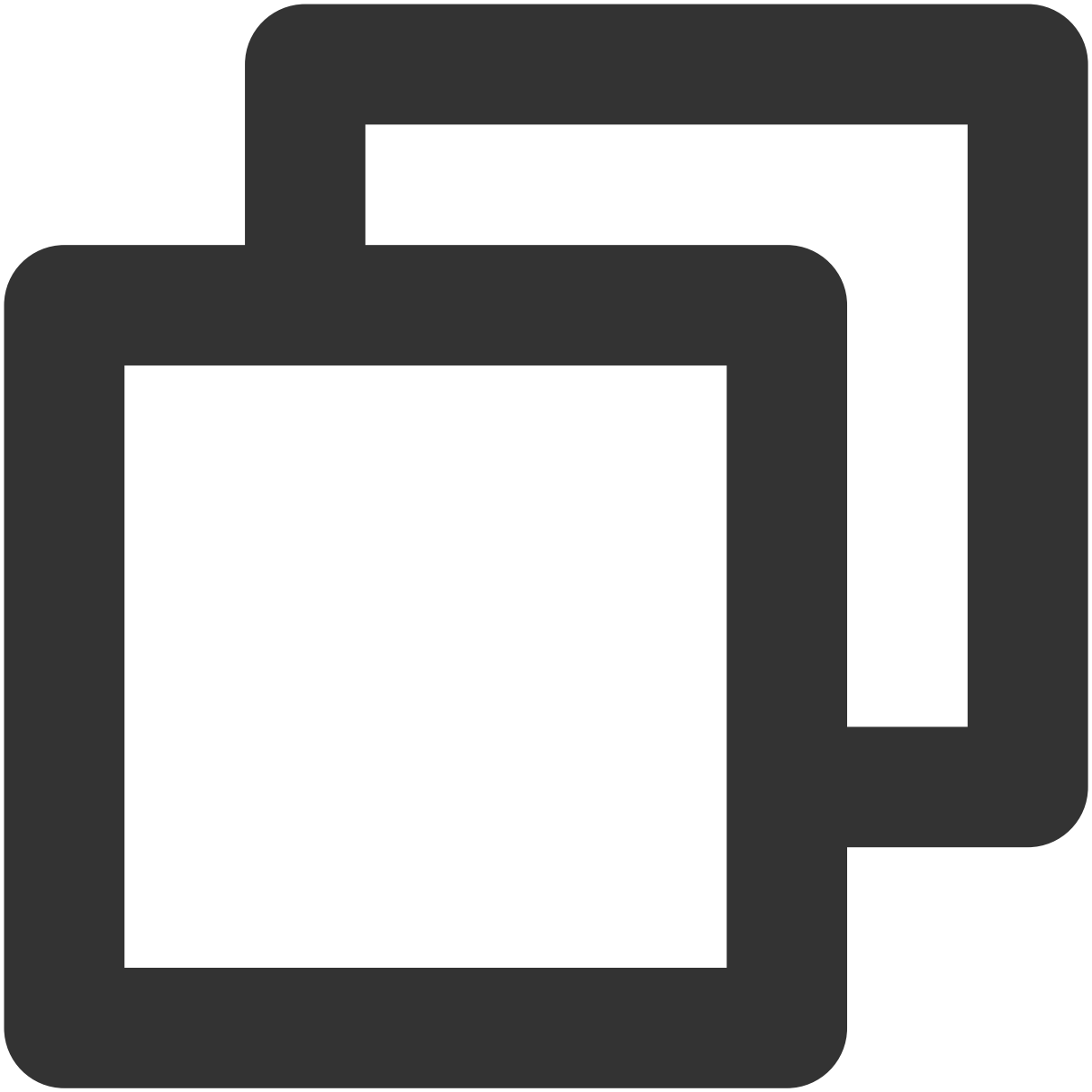
分词

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。

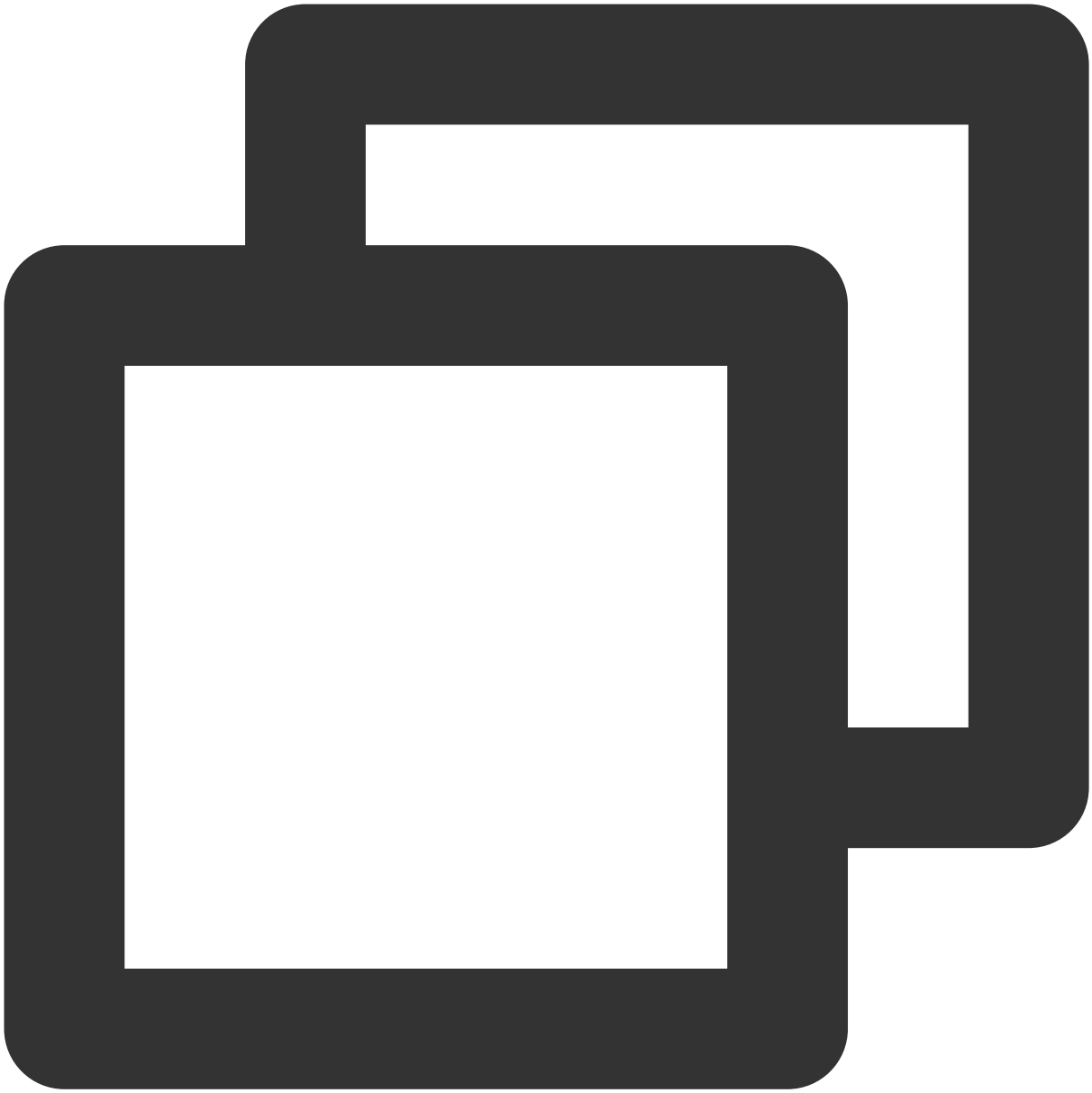
实际案例

请求



```
GET /jobs?queueId=p893bcda225bf4945a378da6662e81a89&tag=Translation HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: test-1234567890.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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
    <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>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
    <NextToken>225508</NextToken>
</Response>
```

内容识别任务回调
翻译任务回调

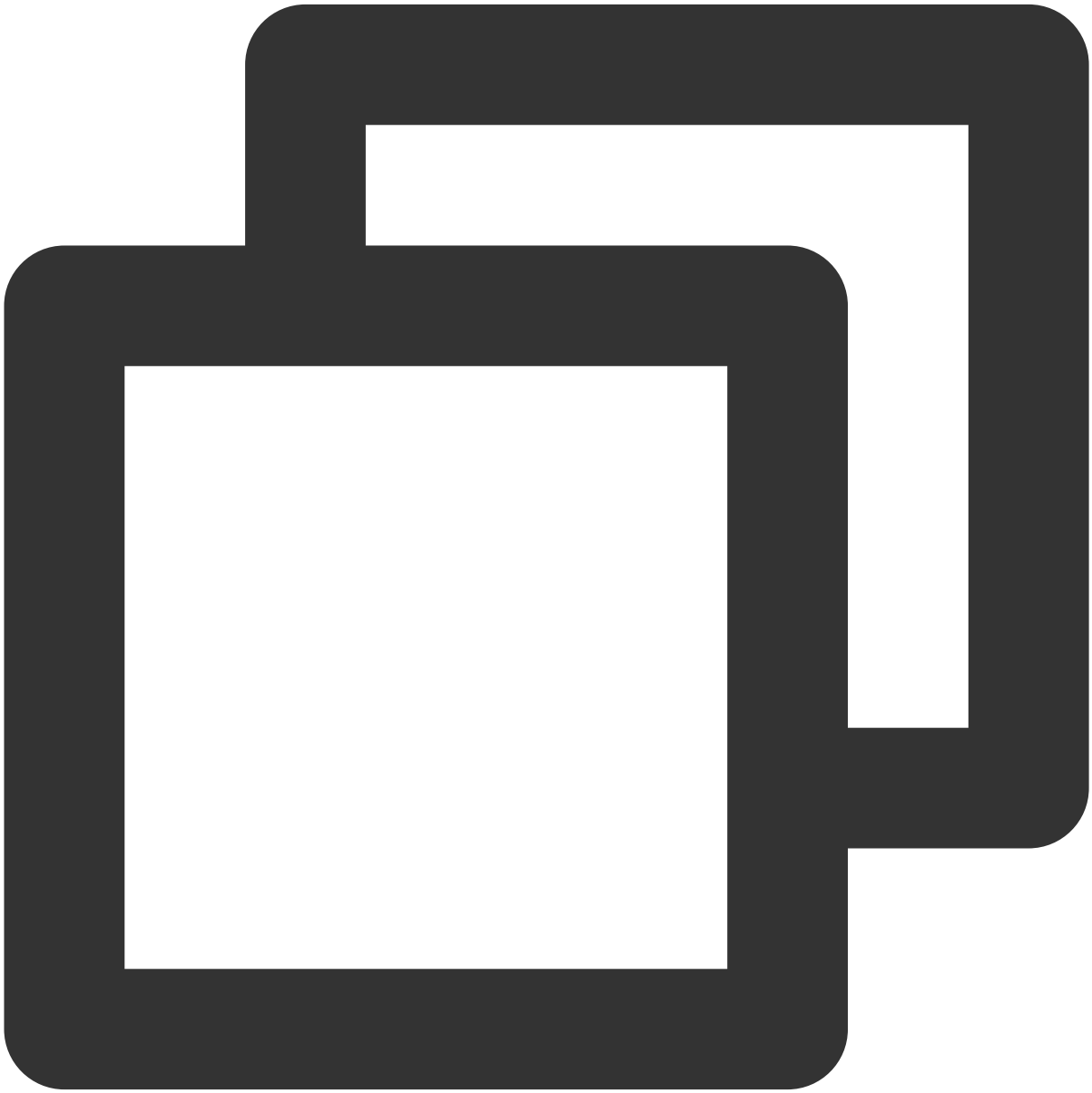
最近更新时间：2024-01-20 16:35:12

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>input/en.pdf</Object>
            <Lang>en</Lang>
            <Type>pdf</Type>
            <BasicType>pptx</BasicType>
```

```
        </Input>
        <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</JobId>
        <Message>Success</Message>
        <Operation>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/zh.pdf</Object>
            </Output>
            <Translation>
                <Lang>en</Lang>
                <Type>txt</Type>
            </Translation>
            <UserData>This is my Translation job.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>pbde29b842b1b4e2ea4f0b20d264fcec2</QueueId>
        <StartTime>2022-07-25T16:35:39+0800</StartTime>
        <State>Success</State>
        <Tag>Translation</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：

同提交翻译任务接口中的 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

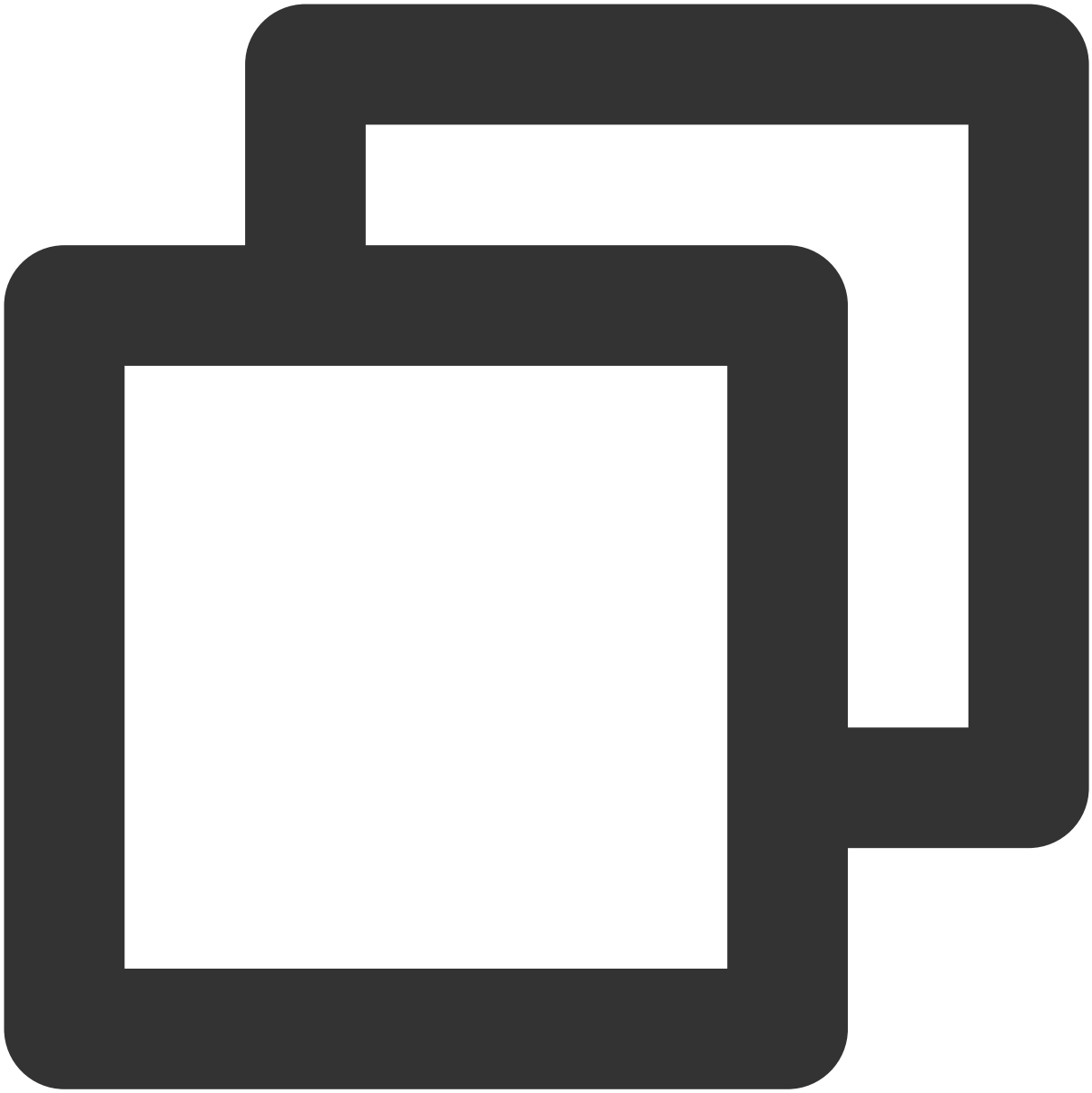
如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

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

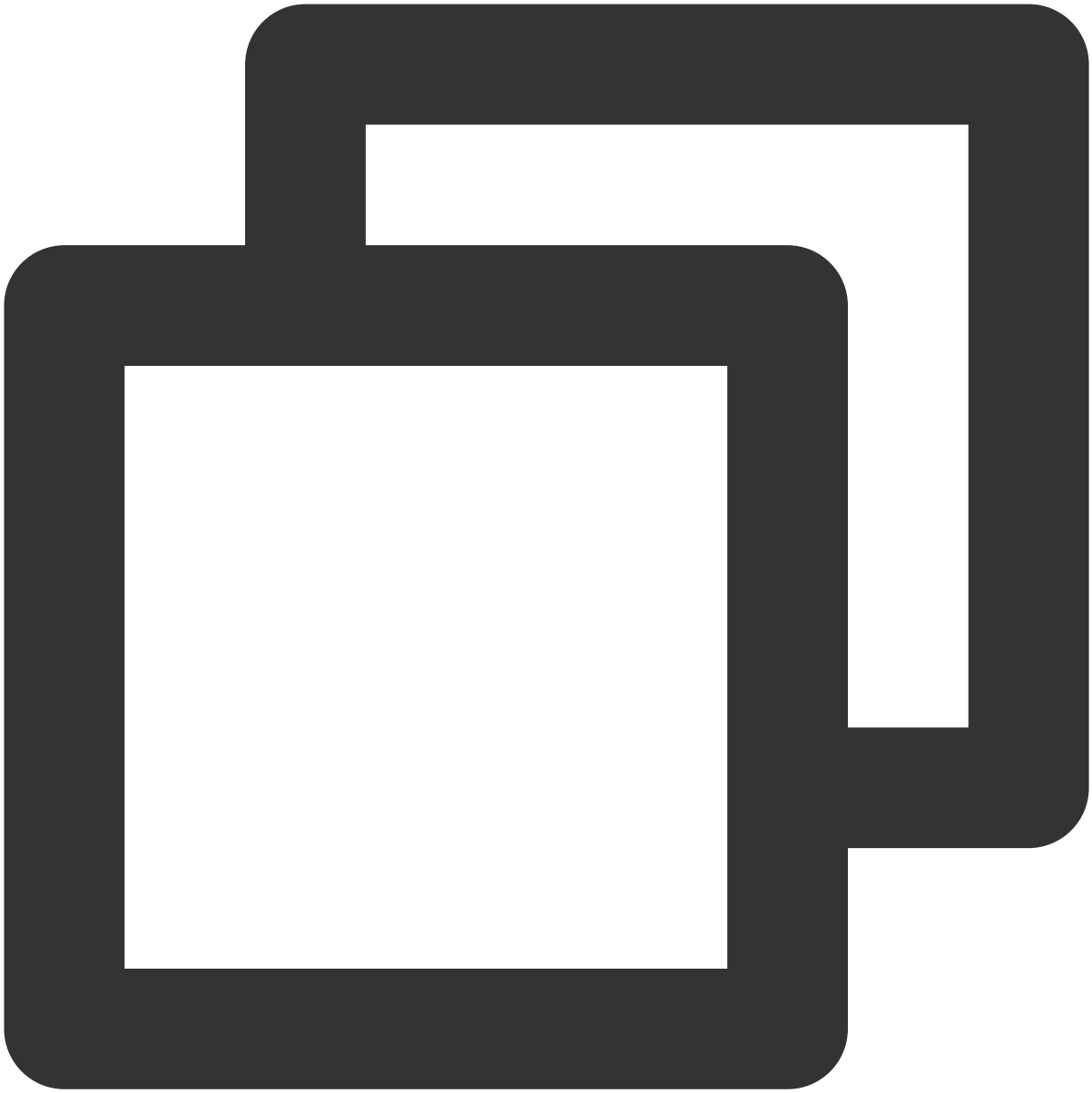


```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Code>Success</Code>
        <CreationTime>2022-07-25T16:35:39+0800</CreationTime>
        <EndTime>2022-07-25T16:35:43+0800</EndTime>
        <Input>
            <Object>input/en.pdf</Object>
            <Lang>en</Lang>
            <Type>pdf</Type>
            <BasicType>pptx</BasicType>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </Input>
        <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</JobId>
        <Message>Success</Message>
        <Operation>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-123456789</Bucket>
                <Object>output/zh.pdf</Object>
            </Output>
            <Translation>
                <Lang>en</Lang>
                <Type>txt</Type>
            </Translation>
            <UserData>This is my Translation job.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>pbde29b842b1b4e2ea4f0b20d264fcec2</QueueId>
        <StartTime>2022-07-25T16:35:39+0800</StartTime>
        <State>Success</State>
        <Tag>Translation</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务触发的任务回调，格式为 JSON




```
{
    "EventName": "TaskFinish",
    "JobsDetail": {
        "Code": "Success",
        "CreationTime": "2022-07-25T16:35:39+0800",
        "EndTime": "2022-07-25T16:35:43+0800",
        "Input": {
            "Object": "input/en.pdf",
            "Lang": "en",
            "Type": "pdf",
            "BasicType": "pptx"
```

```
        },
        "JobId": "ac34be3aa0bf411ed9ce39d7cc972e427",
        "Message": "Success",
        "Operation": {
            "Output": {
                "Region": "ap-chongqing",
                "Bucket": "test-123456789",
                "Object": "output/zh.pdf"
            },
            "Translation": {
                "Lang": "en",
                "Type": "txt"
            },
            "UserData": "This is my Translation job.",
            "JobLevel": 0,
        },
        "QueueId": "pbde29b842b1b4e2ea4f0b20d264fcec2",
        "StartTime": "2022-07-25T16:35:39+0800",
        "State": "Success",
        "Tag": "Translation"
    }
}
```

分词任务回调

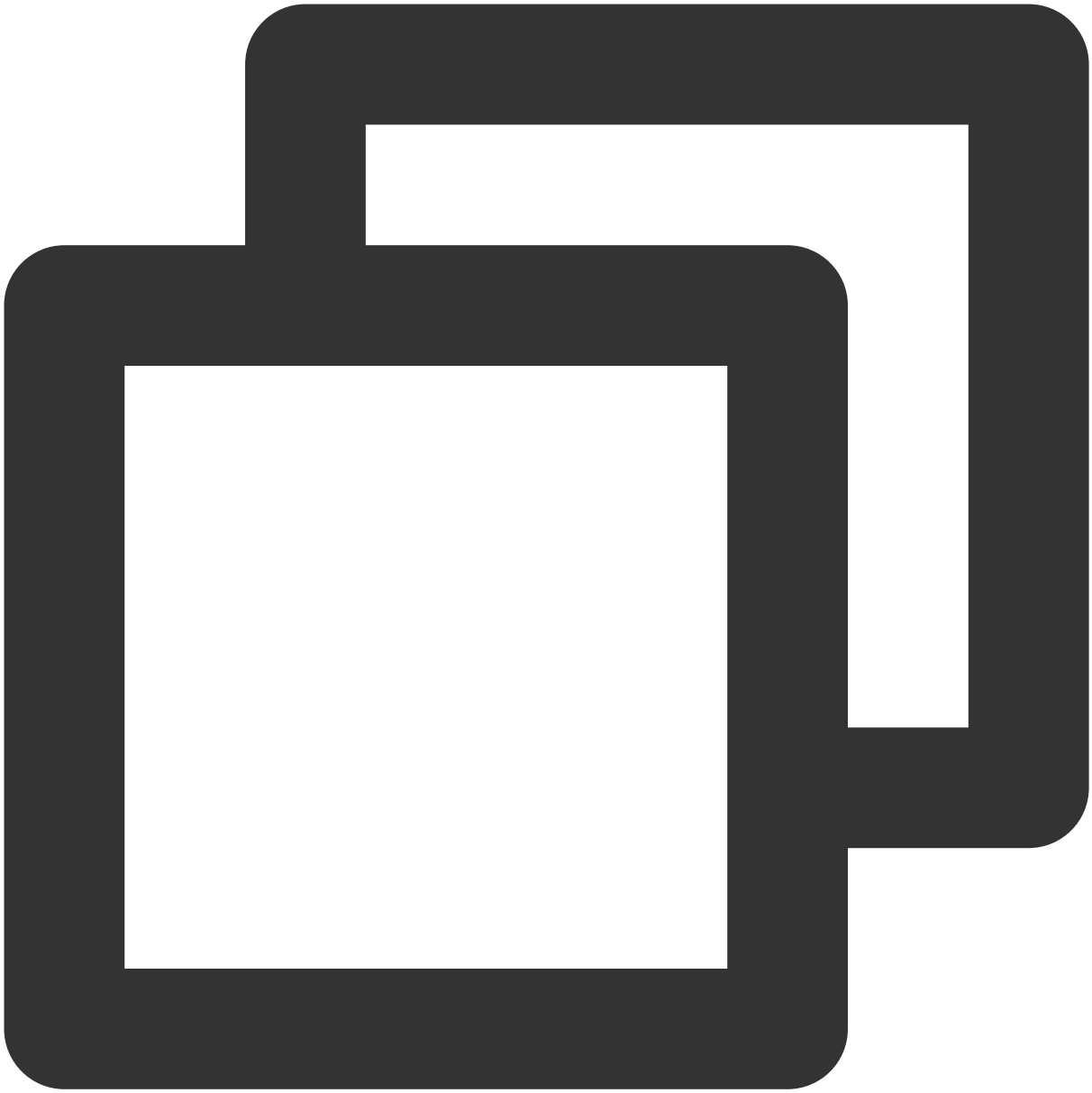
最近更新时间：2024-01-20 16:35:12

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>
        <Tag>WordsGeneralize</Tag>
        <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>
            <UserData>This is my WordsGeneralize job.</UserData>
            <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。

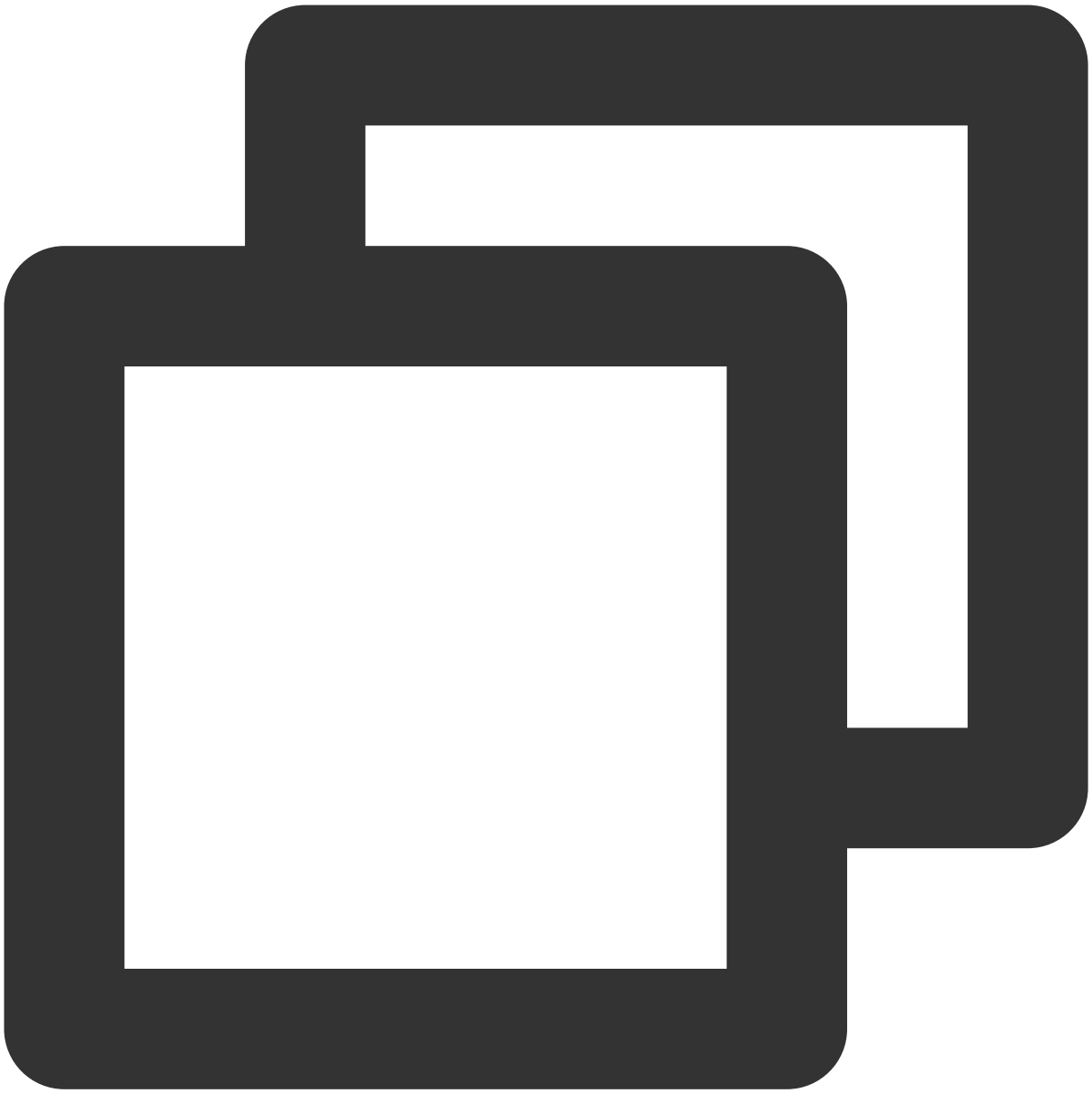
Container 节点 Workflow 的内容：

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

版权所有：腾讯云计算（北京）有限责任公司

案例 1：通过任务接口触发的任务回调



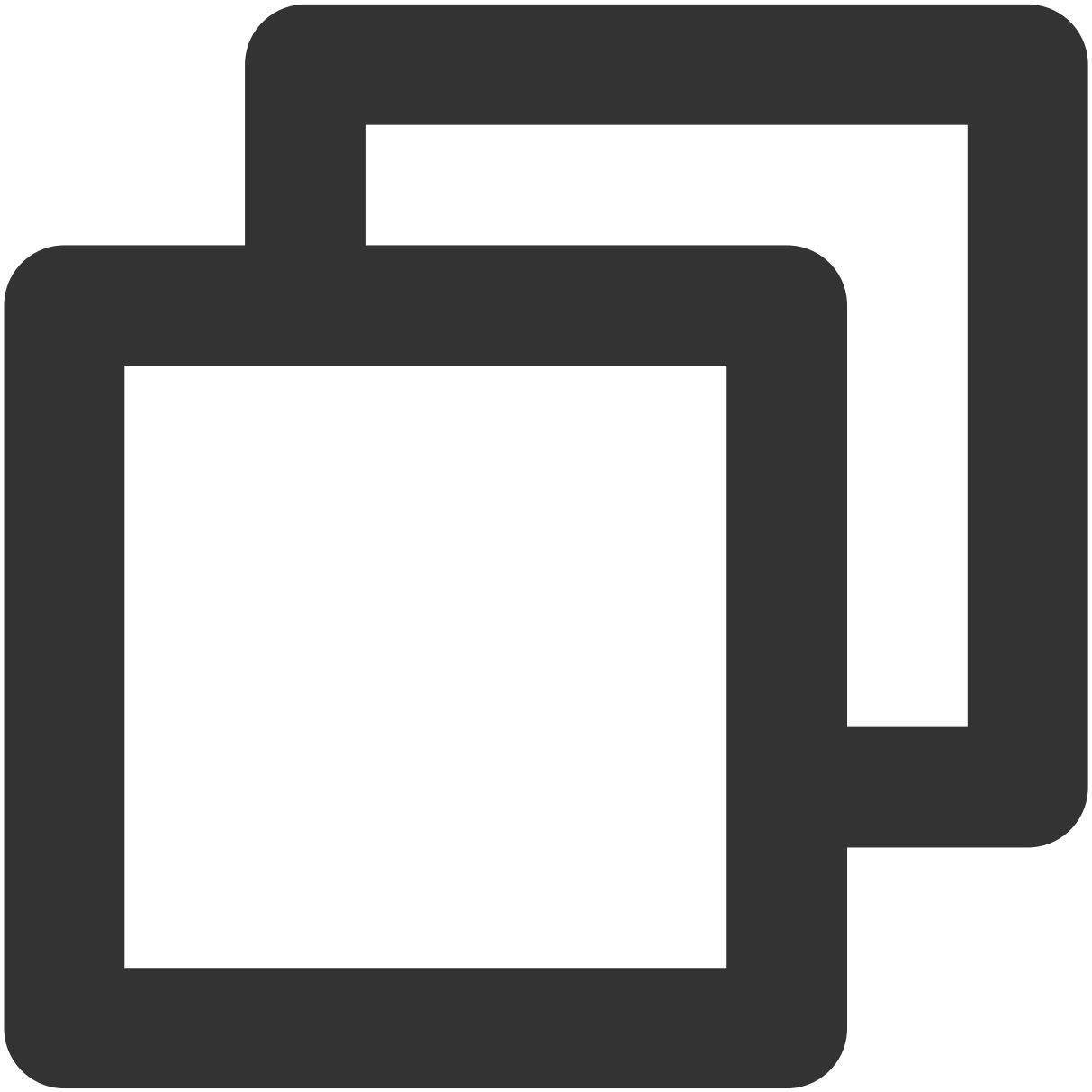
```
<Response>
    <EventName>TaskFinish</EventName>
    <JobsDetail>
        <Tag>WordsGeneralize</Tag>
        <Code>Success</Code>
        <CreationTime>2022-07-25T16:35:39+0800</CreationTime>
        <EndTime>2022-07-25T16:35:43+0800</EndTime>
        <Input>
            <Object>text.txt</Object>
```

```
            <BucketId>test-123456789</BucketId>
            <Region>ap-chongqing</Region>
        </Input>
        <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</JobId>
        <Message>Success</Message>
        <Operation>
            <UserData>This is my WordsGeneralize job.</UserData>
            <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            </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"
                            }
                        ]
                    }
                }
            }
        }
    }
```

模板接口
媒体处理
创建媒体处理模板
创建动图模板

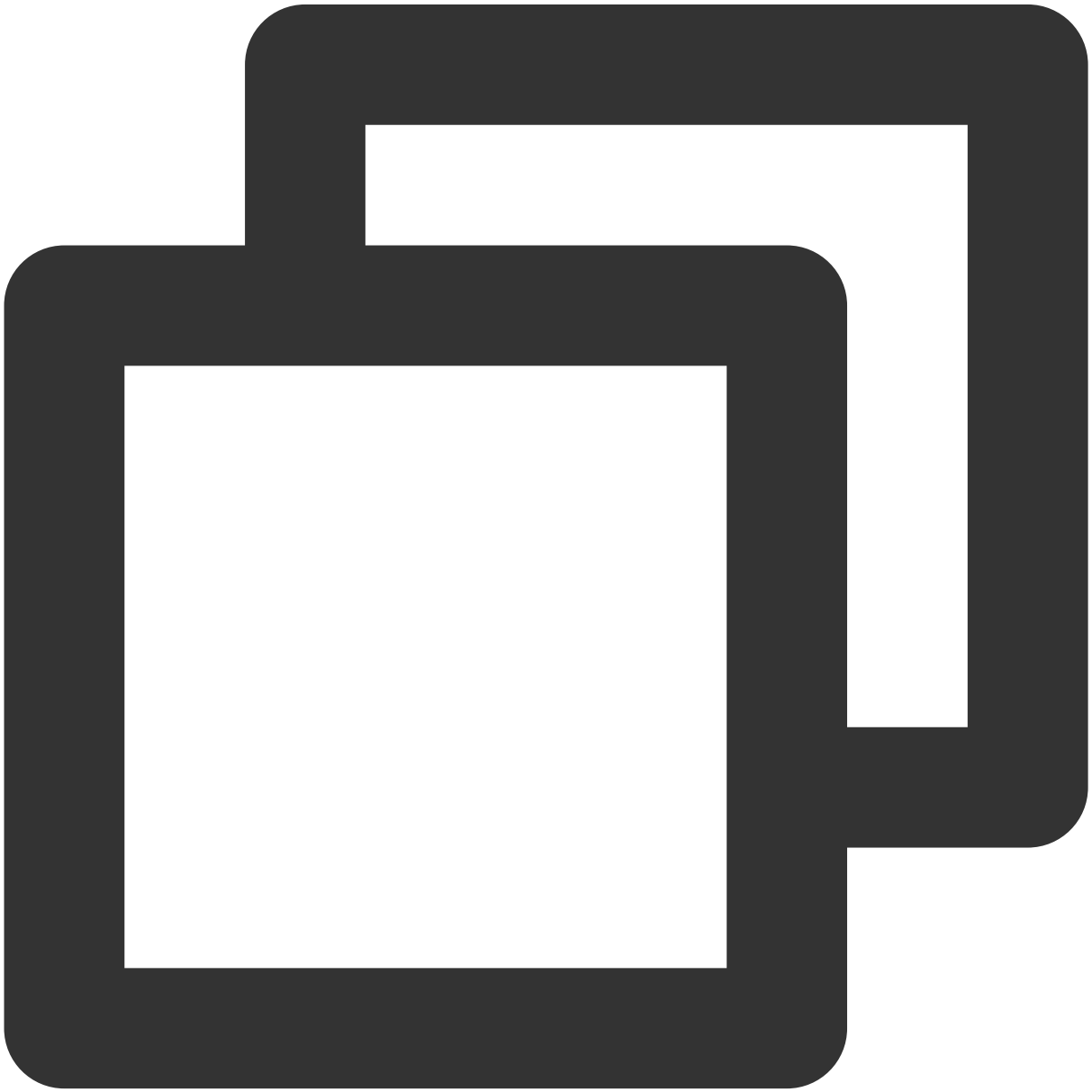
最近更新时间：2024-01-23 10:00:02

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：

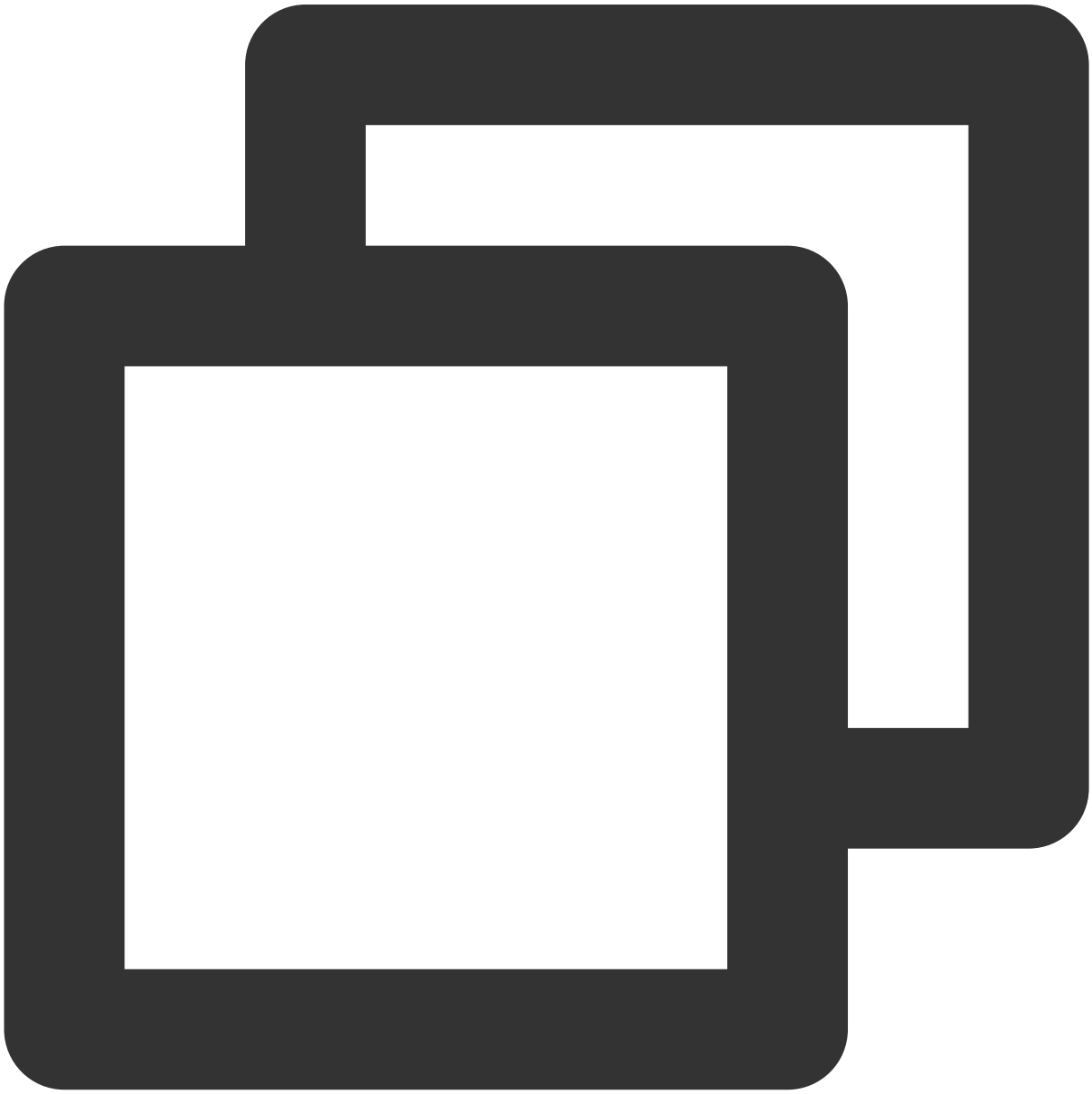
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	限制
Codec	Request.Video	编解码格式	String	是	无	gif, webp
Width	Request.Video	宽	String	否	视频原始宽度	值范围：[12 单位：px 若只设置 W 频原始比例
Height	Request.Video	高	String	否	视频原始高度	值范围：[12 单位：px 若只设置 He 频原始比例
Fps	Request.Video	帧率	String	否	视频原始帧率	值范围：(0, 单位：fps 如果不设置， 按照原来的 置 fps 为动
AnimateOnly KeepKeyFrame	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	true、false 动图保留关 优先级： AnimateFra AnimateOnl > AnimateTim
AnimateTime IntervalOfFrame	Request.Video	动图抽帧间隔时间	String	否	无	（0，视频时 动图抽帧时 若设置 TimeInterva 于该值
AnimateFrames	Request.Video	Animation 每秒	String	否	无	（0，视频帧

PerSecond		抽帧帧数				动图抽帧频 优先级： AnimateFra AnimateOnl > AnimateTim
Quality	Request.Video	设置相对质量	String	否	无	[1, 100) webp 图像质 没有质量参

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

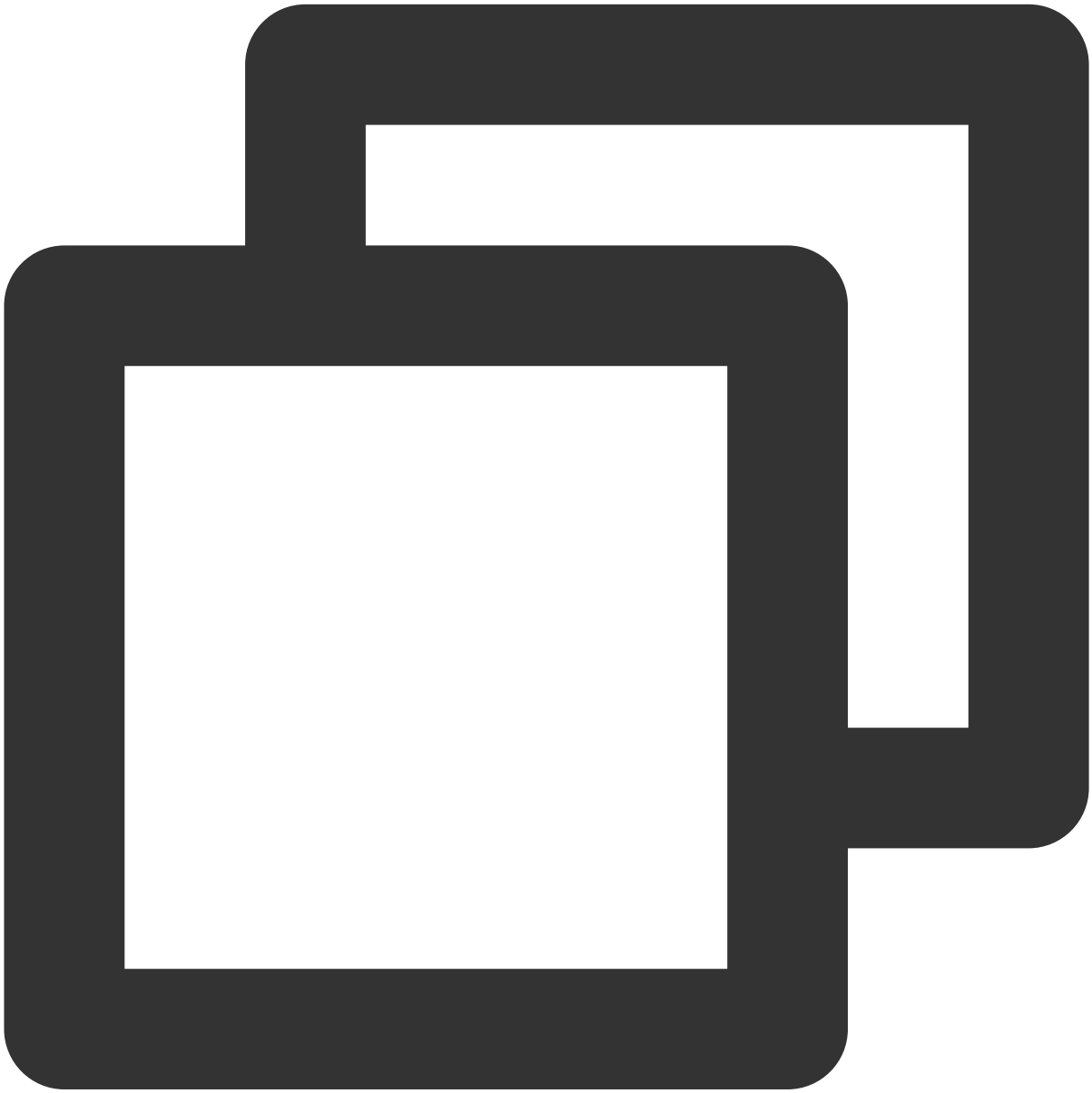
响应

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：

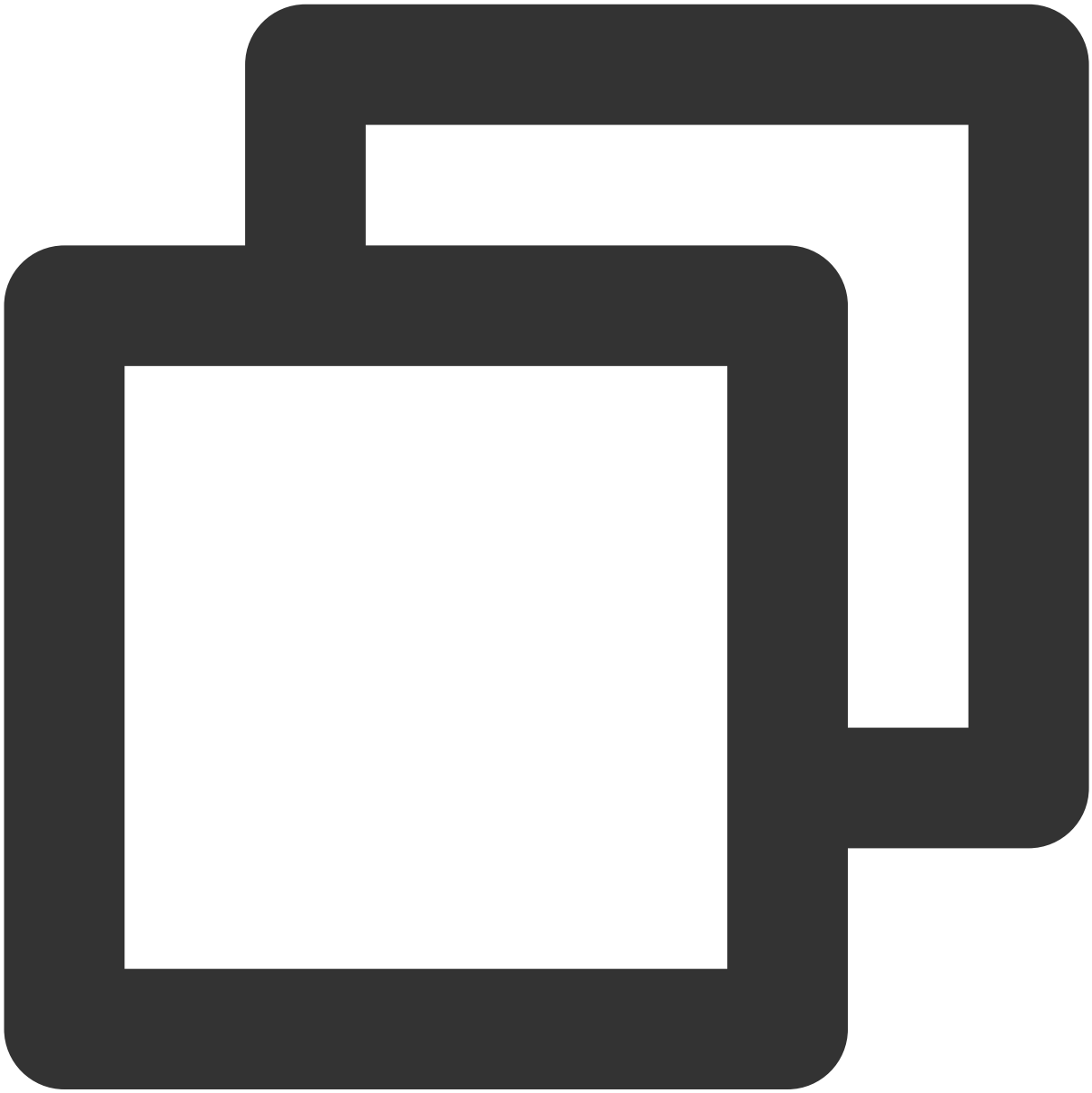
节点名称（关键字）	父节点	描述	类型
Start	Response.TransTpl.TimeInterval	开始时间	String
Duration	Response.TransTpl.TimeInterval	持续时间	String

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

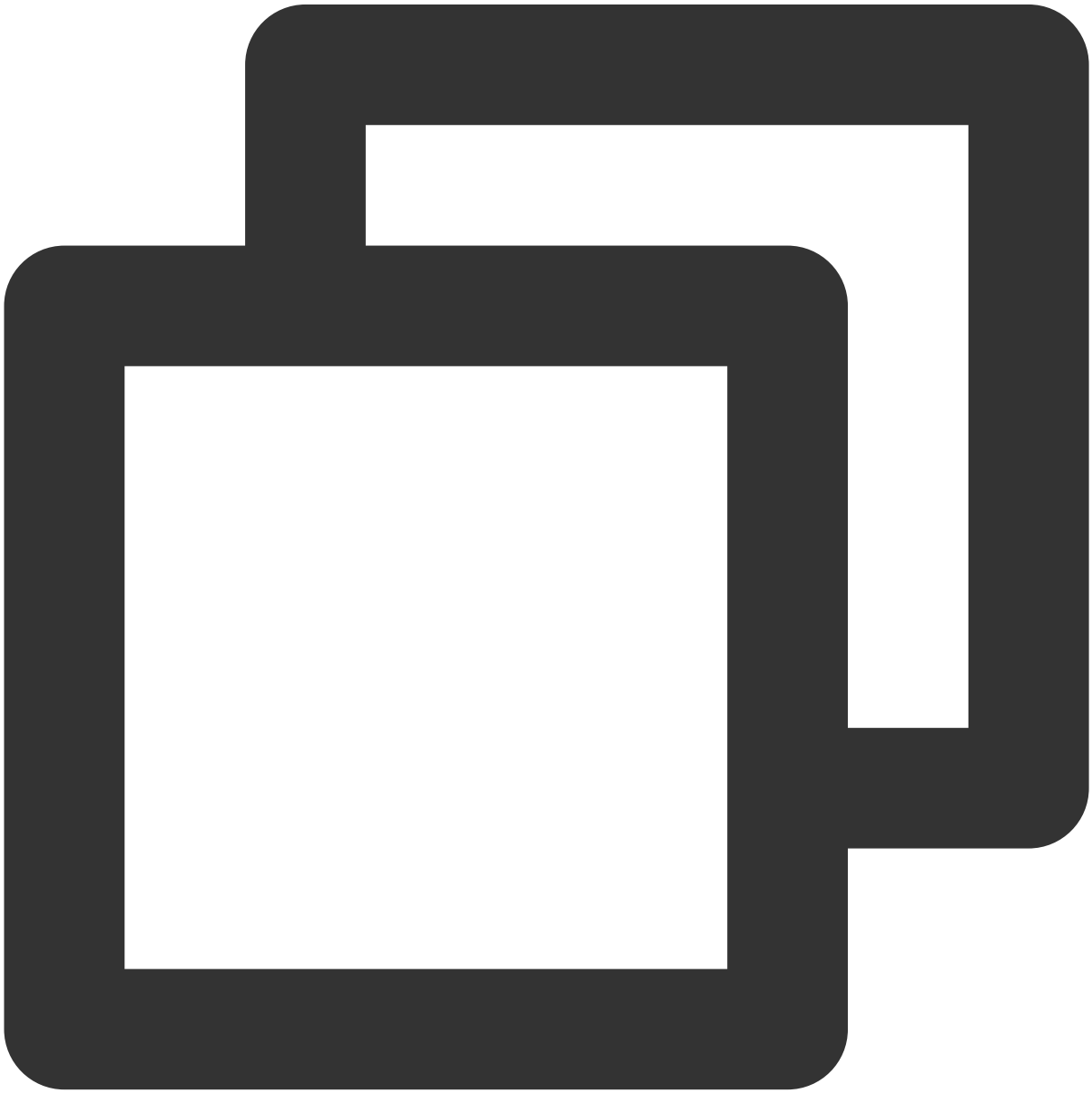


```
POST /template HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

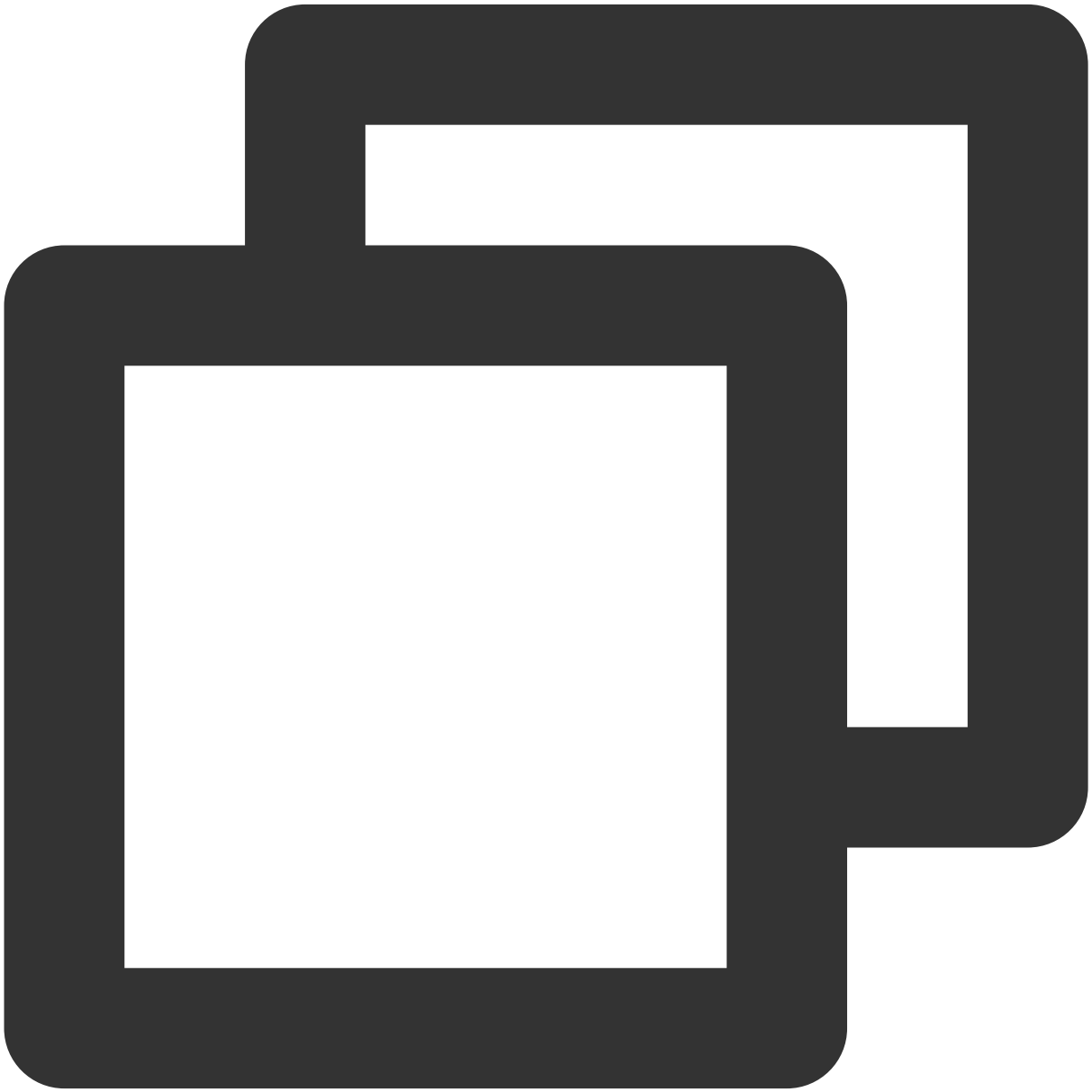
最近更新时间：2024-01-23 10:00:02

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

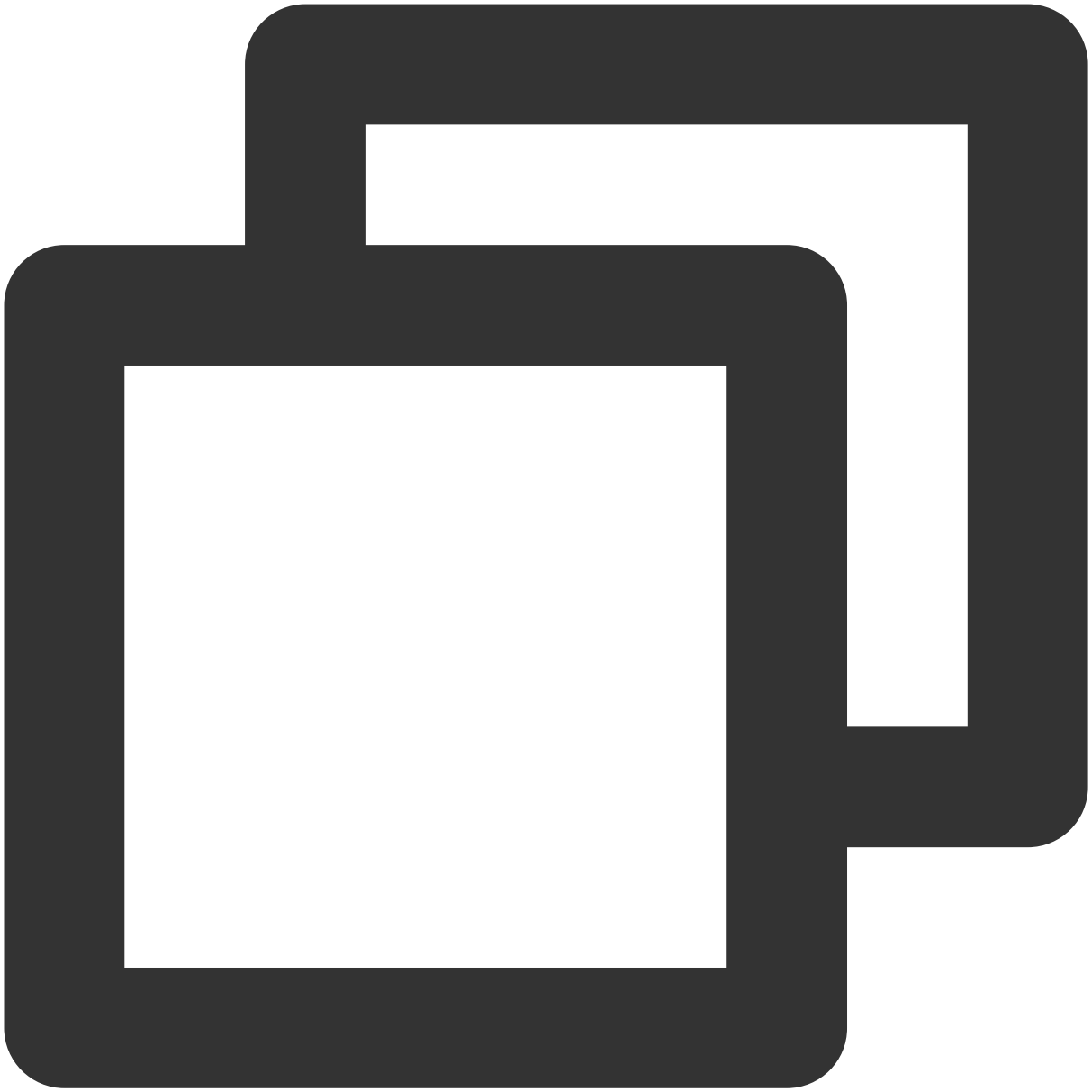
该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

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

版权所有：腾讯云计算（北京）有限责任公司

						End：结尾

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：[10，50000] 单位：Kbps
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	String	否	false	取值 true、

		除视频流				false
Rotate	Request.ConcatTemplate.Video	旋转角度	String	否	无	值范围：[0, 360) 单位：度

响应

响应头

公共响应头

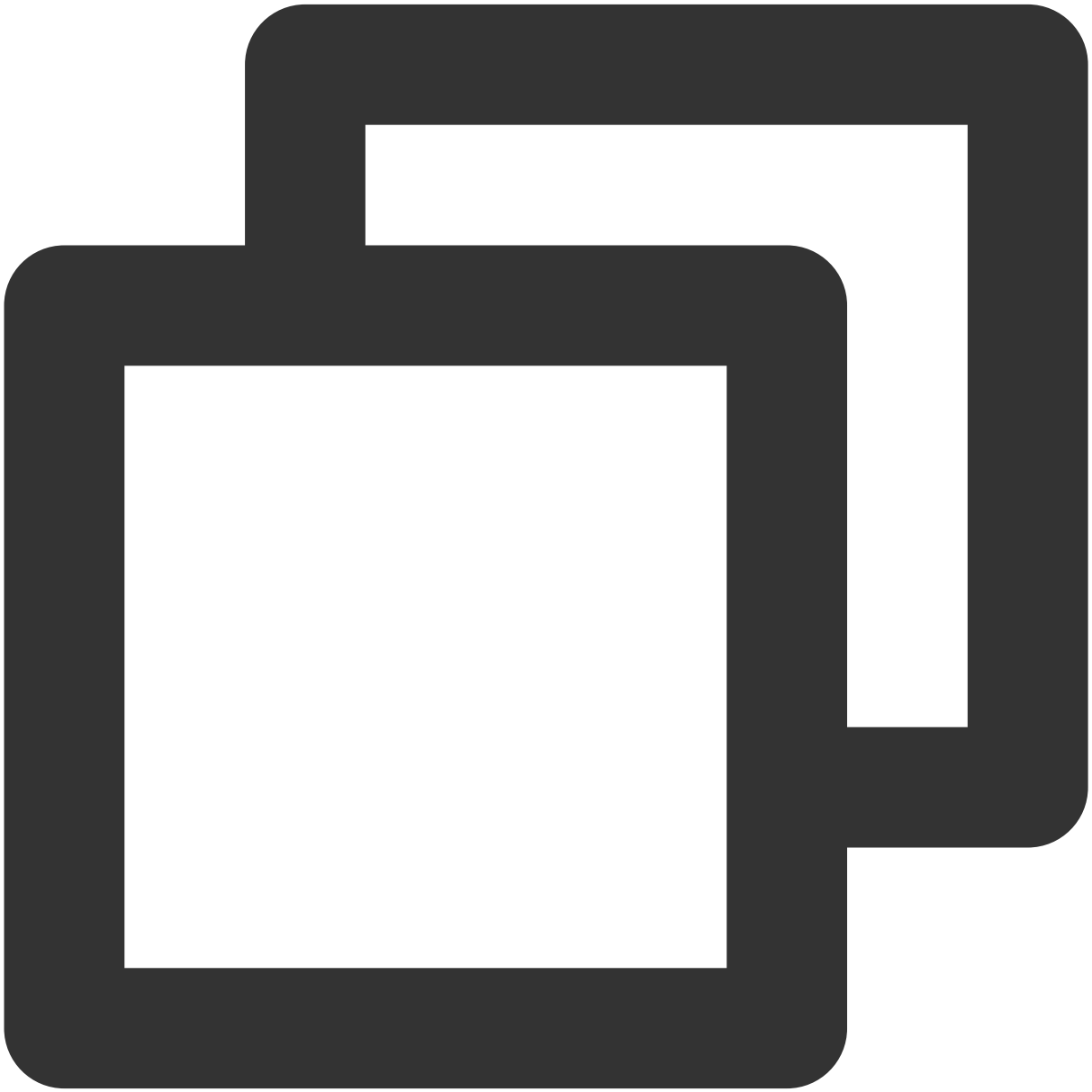
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

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：

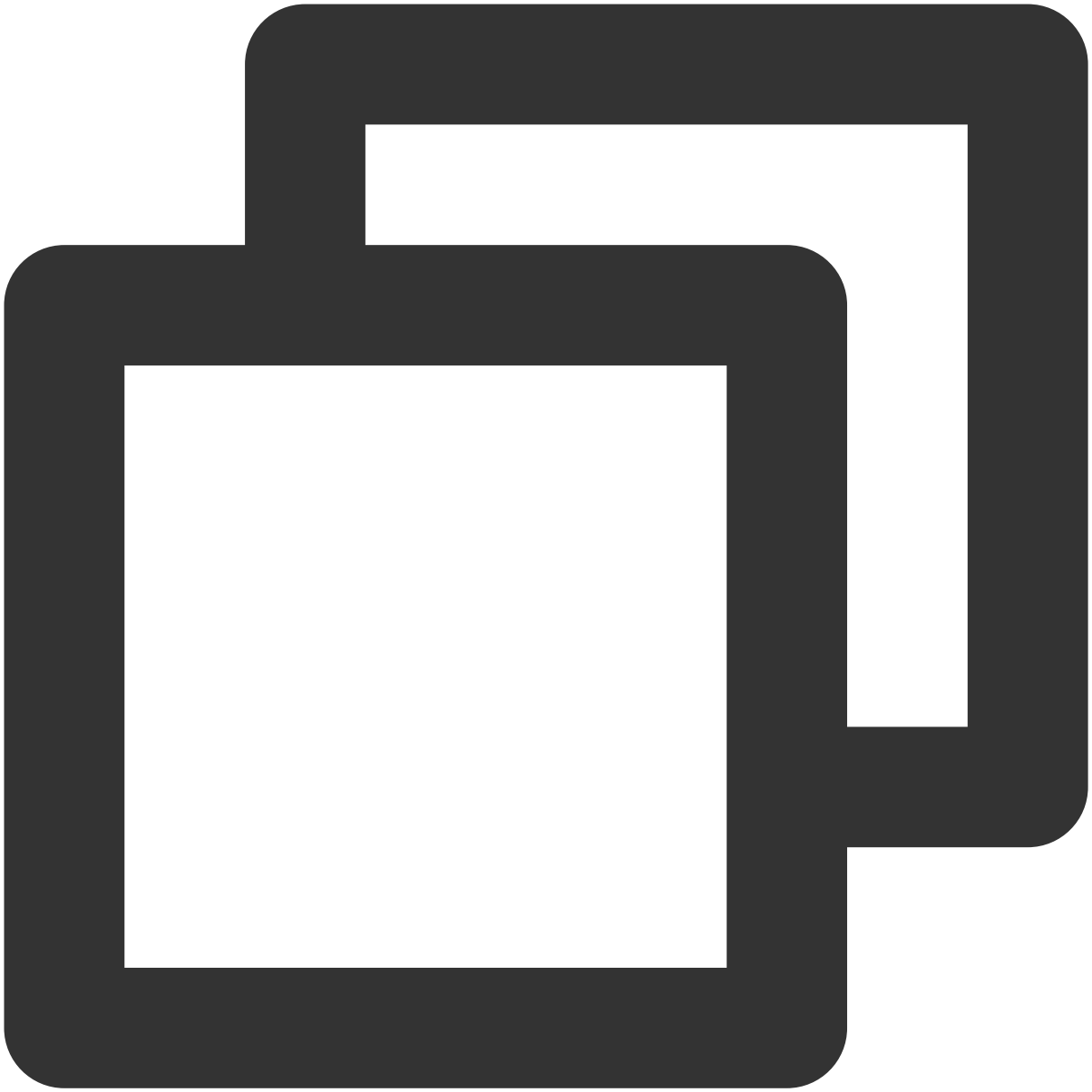
节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，Concat	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	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

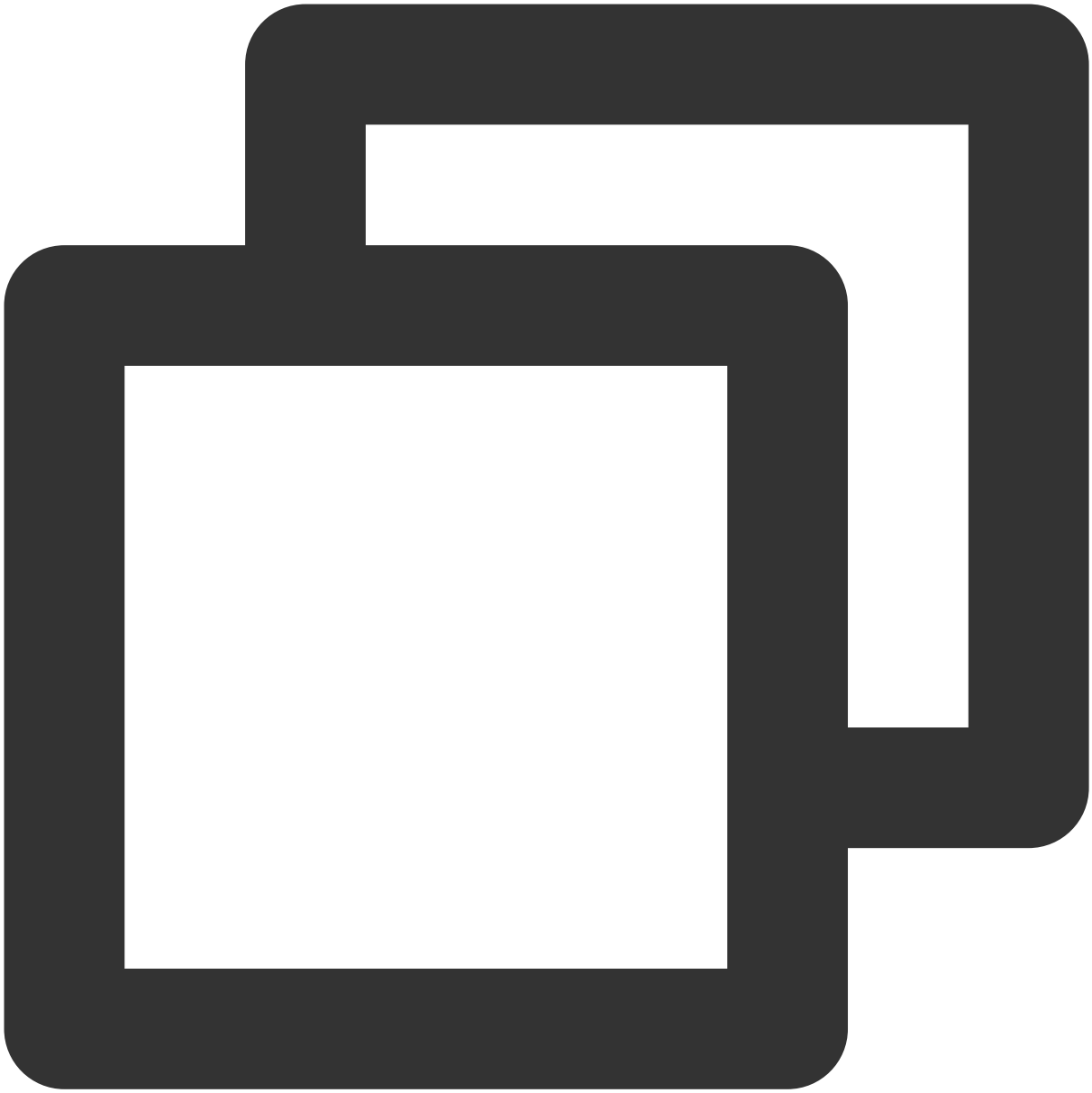


```
POST /template HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

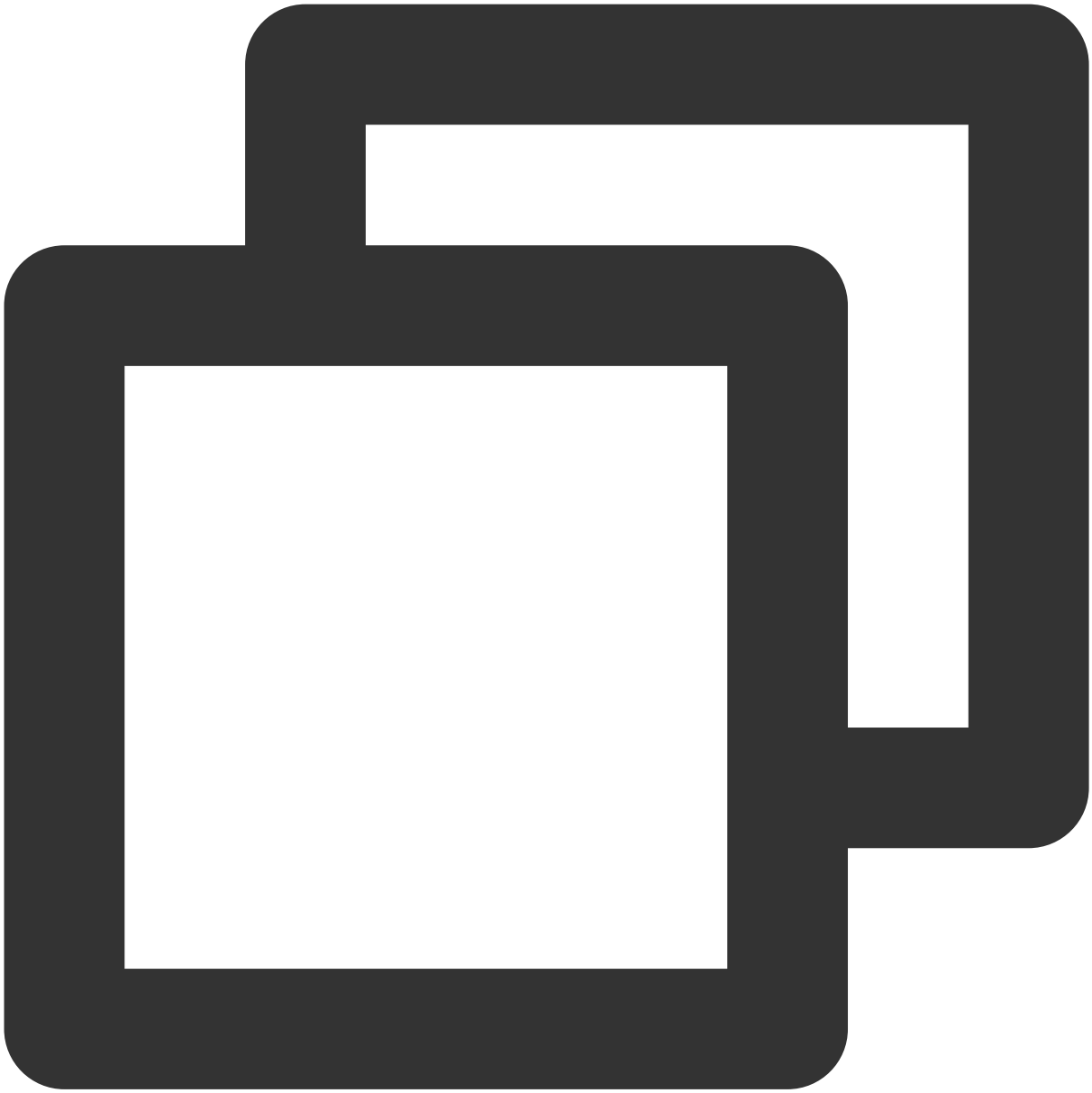
最近更新时间：2024-01-23 10:00:02

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：

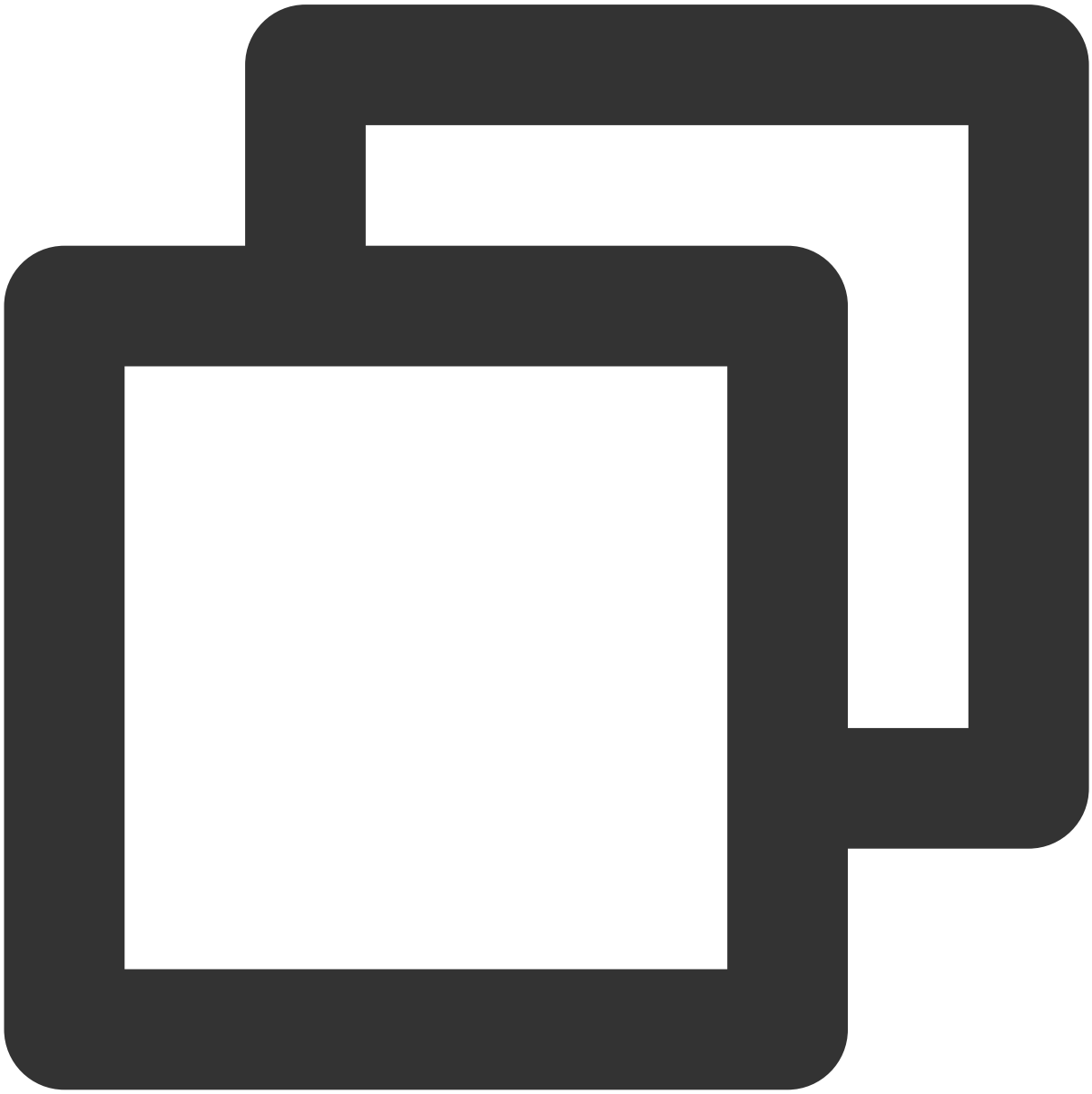
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	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	视频原始宽度	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 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 4. 必须为偶数
Fps	Request.Video	帧率	String	否	无	值范围：(0，60] 单位：fps
Profile	Request.Video	编码级别	String	否	high	支持 baseline、main、high 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
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：[0, 100000]
Preset	Request.Video	视频算法器预置	String	否	medium	仅H.264支持该参数 取值 veryfast、fast、medium、slow、slower
Bufsize	Request.Video	缓冲区大小	String	否	无	值范围：[1000, 128000] 单位：Kb
Maxrate	Request.Video	视频码率峰值	String	否	无	值范围：[10，50000] 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	H.264支持：yuv420p、auto H.265支持：yuv420p、yuv420p10le、auto
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	无	[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	无	[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	否	aac	取值 aac、mp3
Samplerate	Request.Audio	采样率	String	否	44100	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.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，支持1、2、4、5、6、8

版权所有：腾讯云计算（北京）有限责任公司

						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

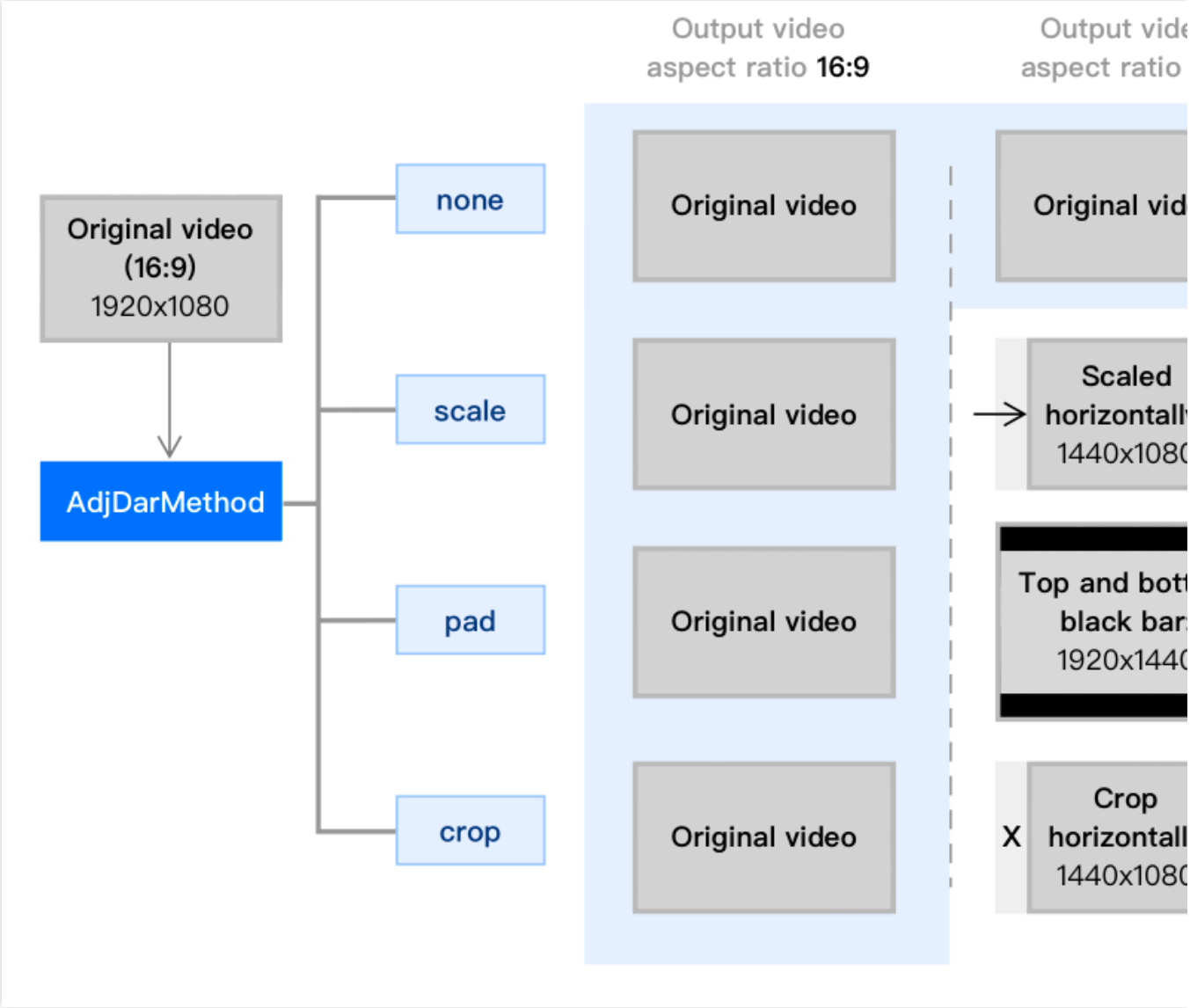
Container 类型 TransConfig 的具体数据描述如下：

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	true、false 当为 false 时，按配置参数转码
ResoAdjMethod	Request.TransConfig	分辨率调整方式	String	否	0	取值0、1；0 表示用原视频分辨率 表示返回转码失败 当 IsCheckReso 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	true、false 当为 false 时，保源文件信息
IsHdr2Sdr	Request.TransConfig	是否开启HDR 转SDR	String	否	false	true/false
HlsEncrypt	Request.TransConfig	hls 加密配置	Container	否	无	无

版权所有：腾讯云计算（北京）有限责任公司

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	true/false 当Container.Format为hls时支持加密
UriKey	Request.TransConfig.HlsEncrypt	HLS加密	String	否	无	当IsHlsEncrypt为true时, 该参数才有意义

		的 key				

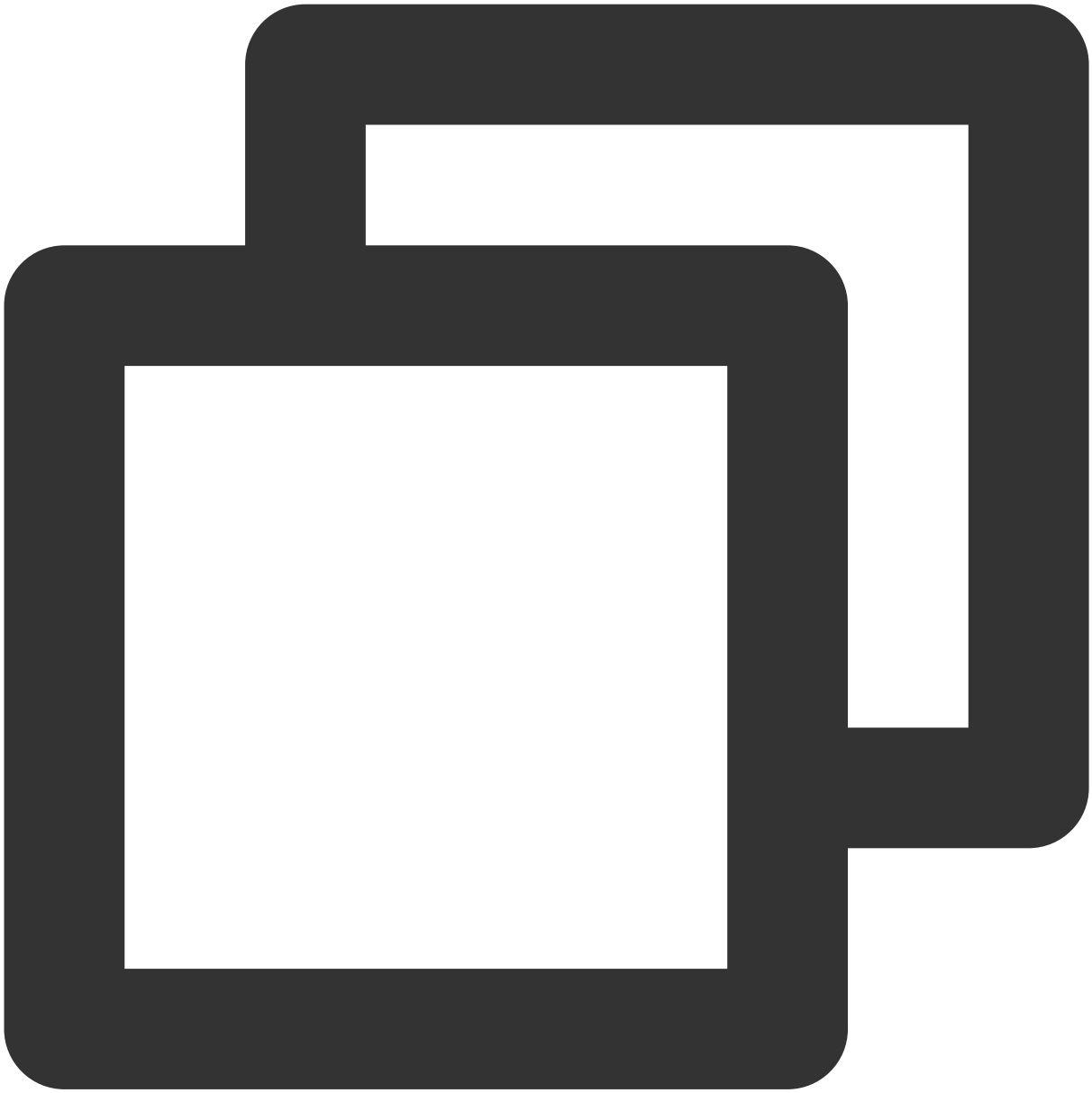
响应

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：

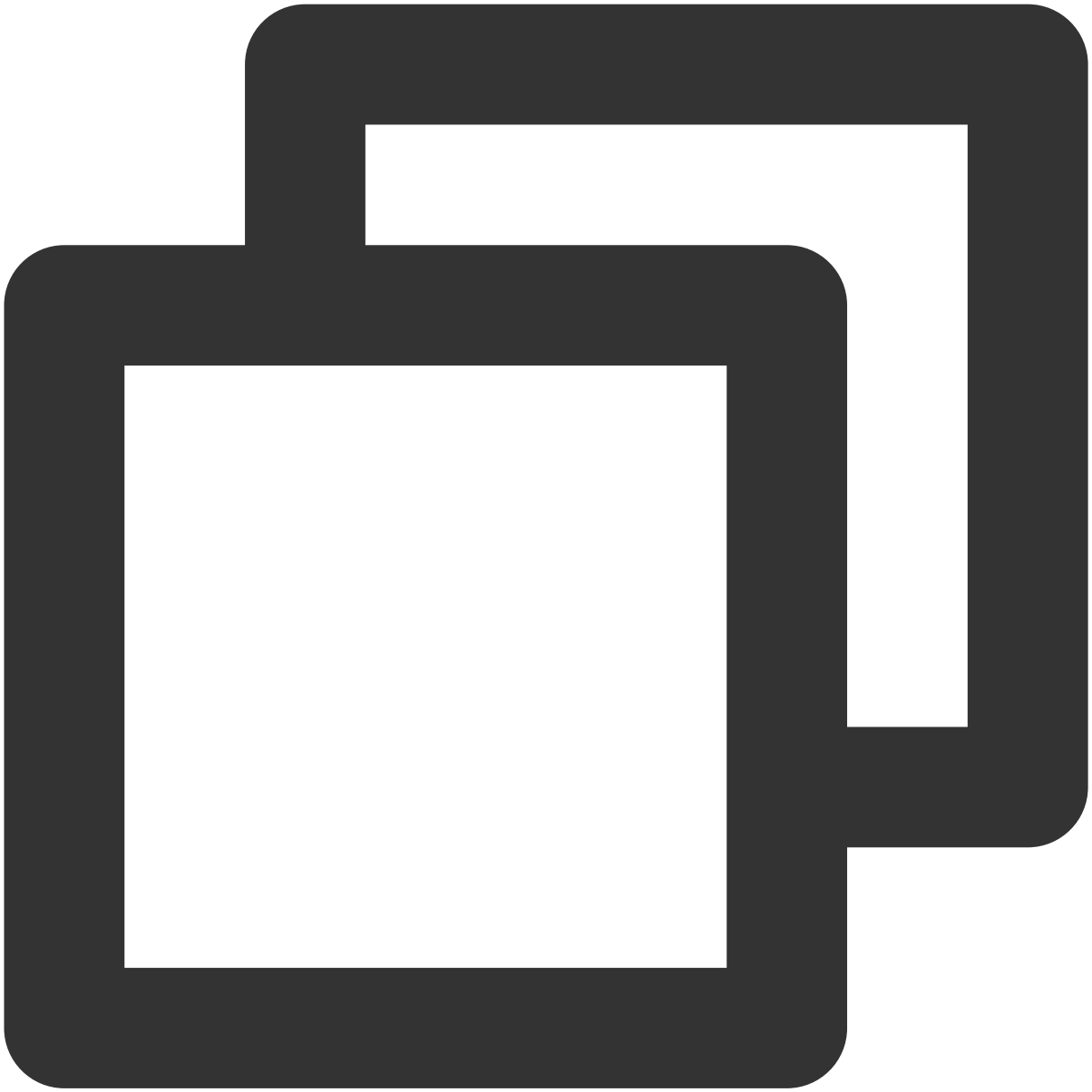
节点名称（关键字）	父节点	描述	类型
TimeInterval	Response.Template.HighSpeedHd	同请求体中的 Request.TimeInterval	Container
Container	Response.Template.HighSpeedHd	同请求体中的 Request.Container	Container
Video	Response.Template.HighSpeedHd	同请求体中的 Request.Video	Container
Audio	Response.Template.HighSpeedHd	同请求体中的 Request.Audio	Container
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

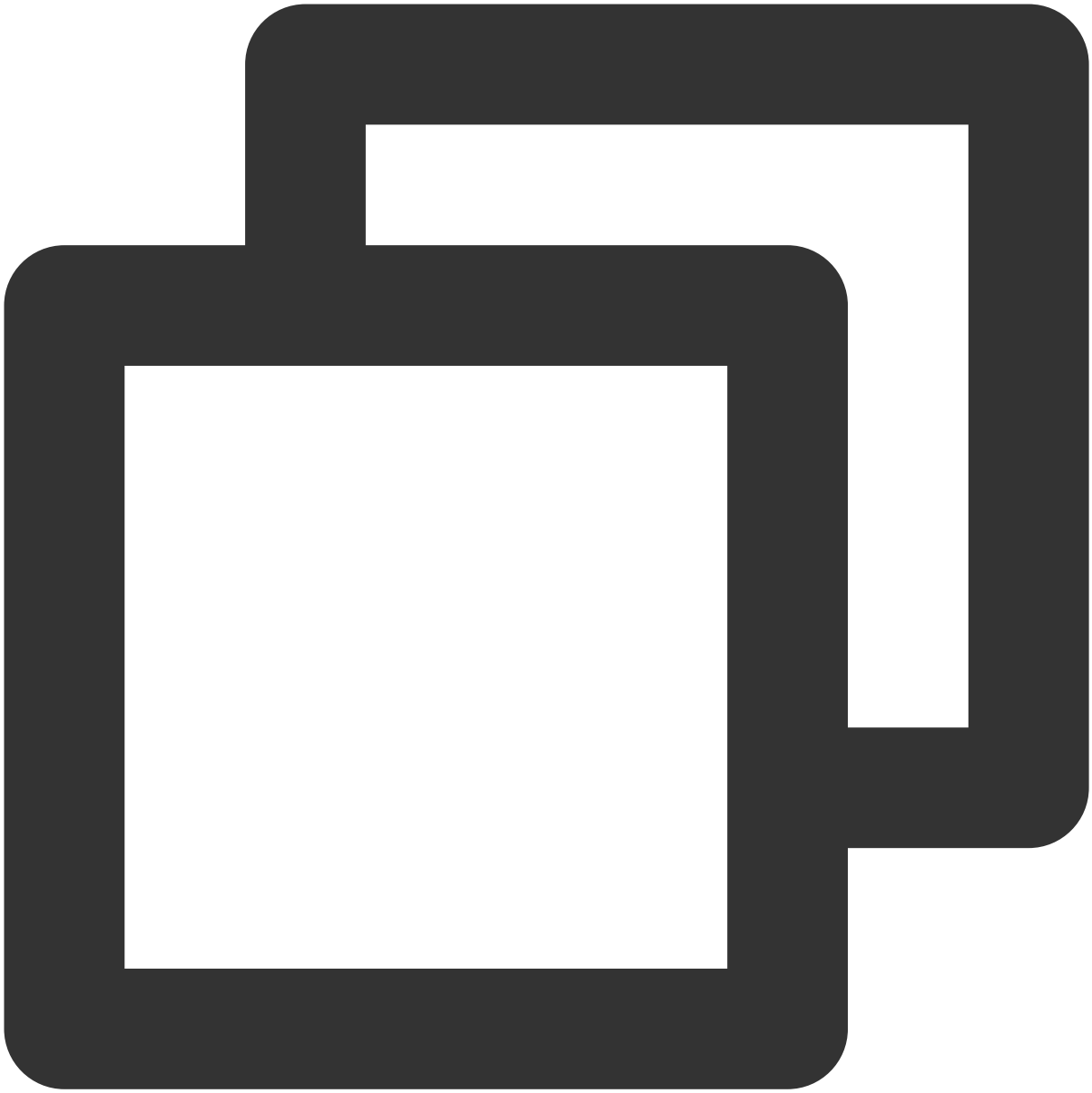


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

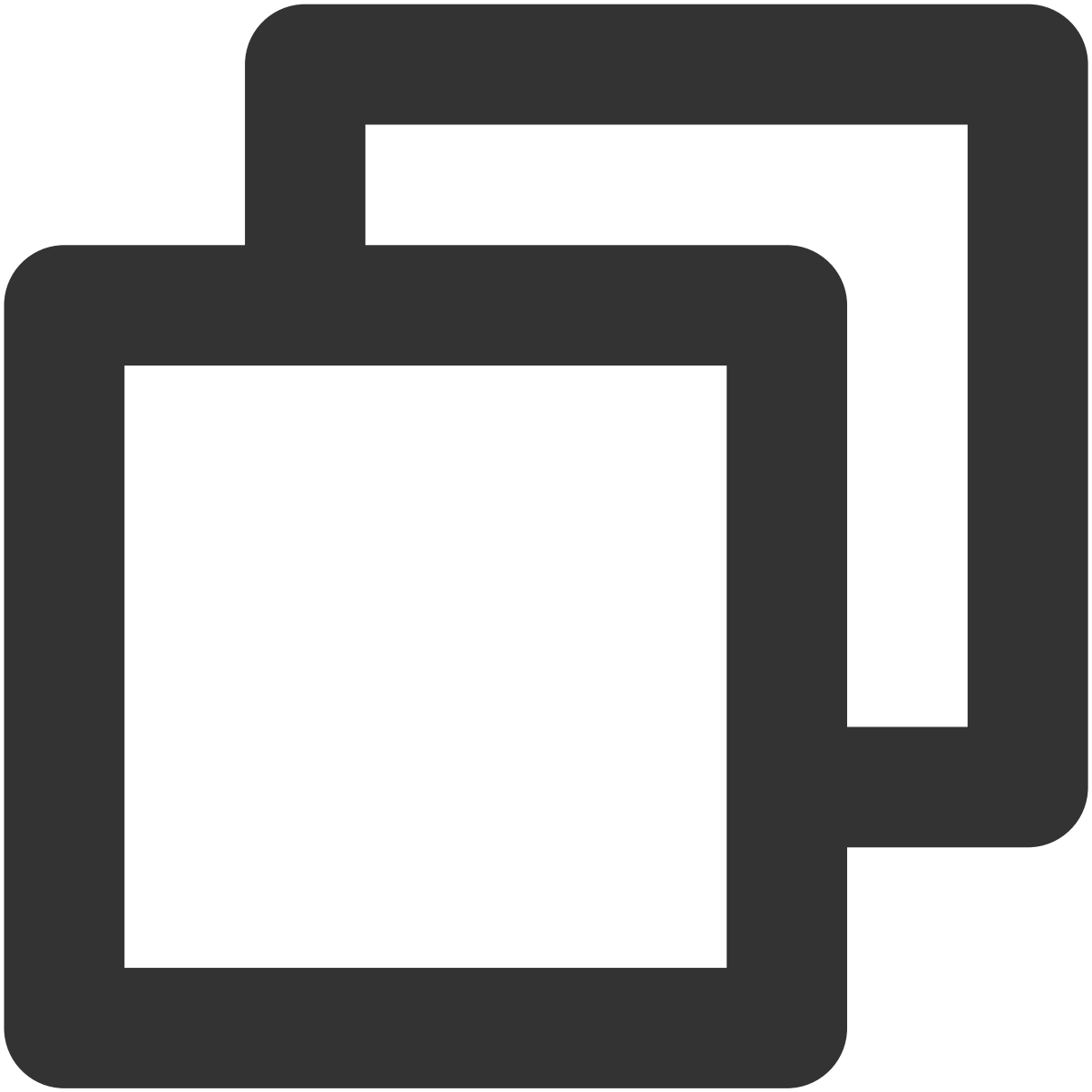
最近更新时间：2024-01-23 10:00:02

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：

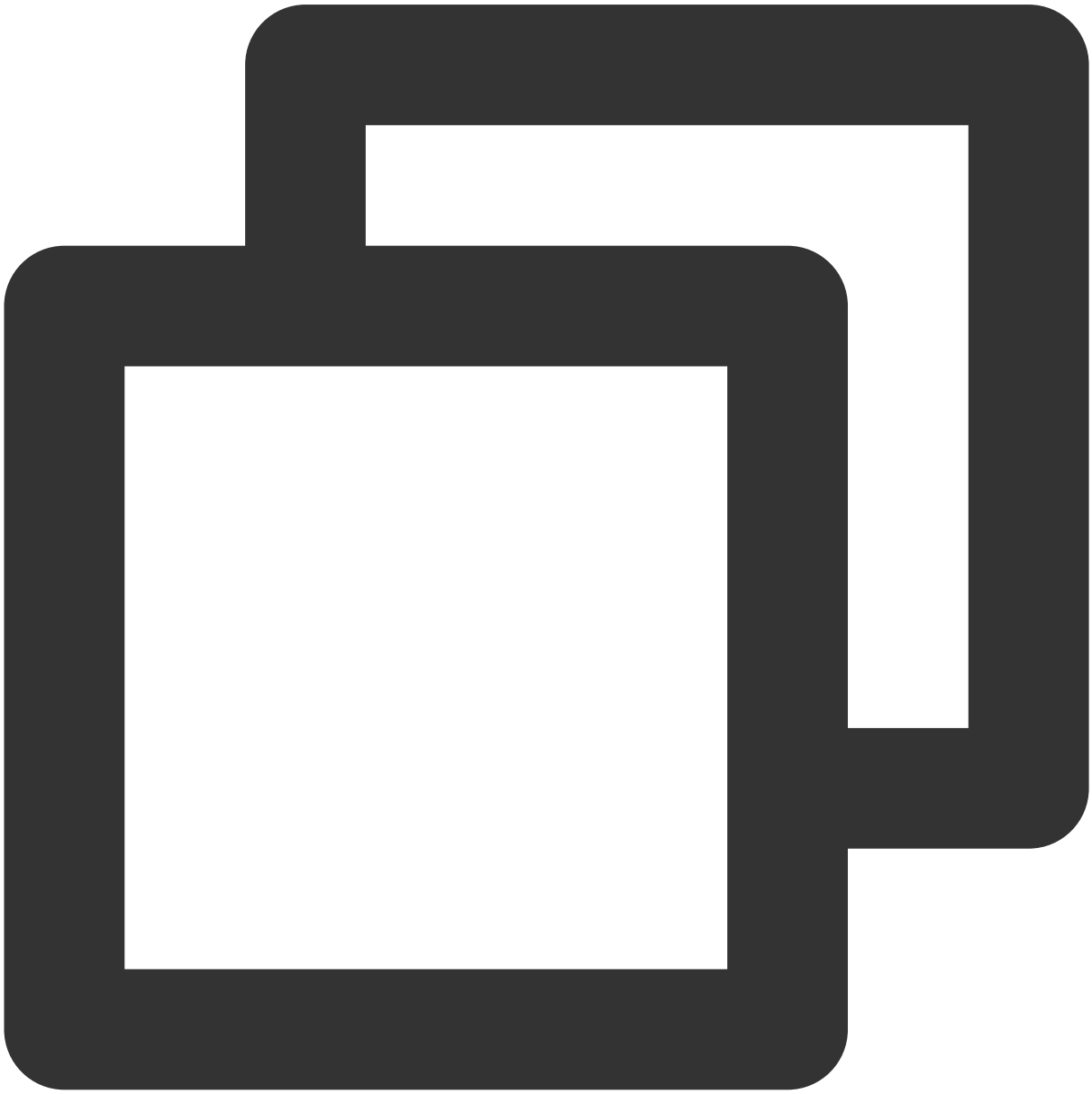
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

版权所有：腾讯云计算（北京）有限责任公司

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]

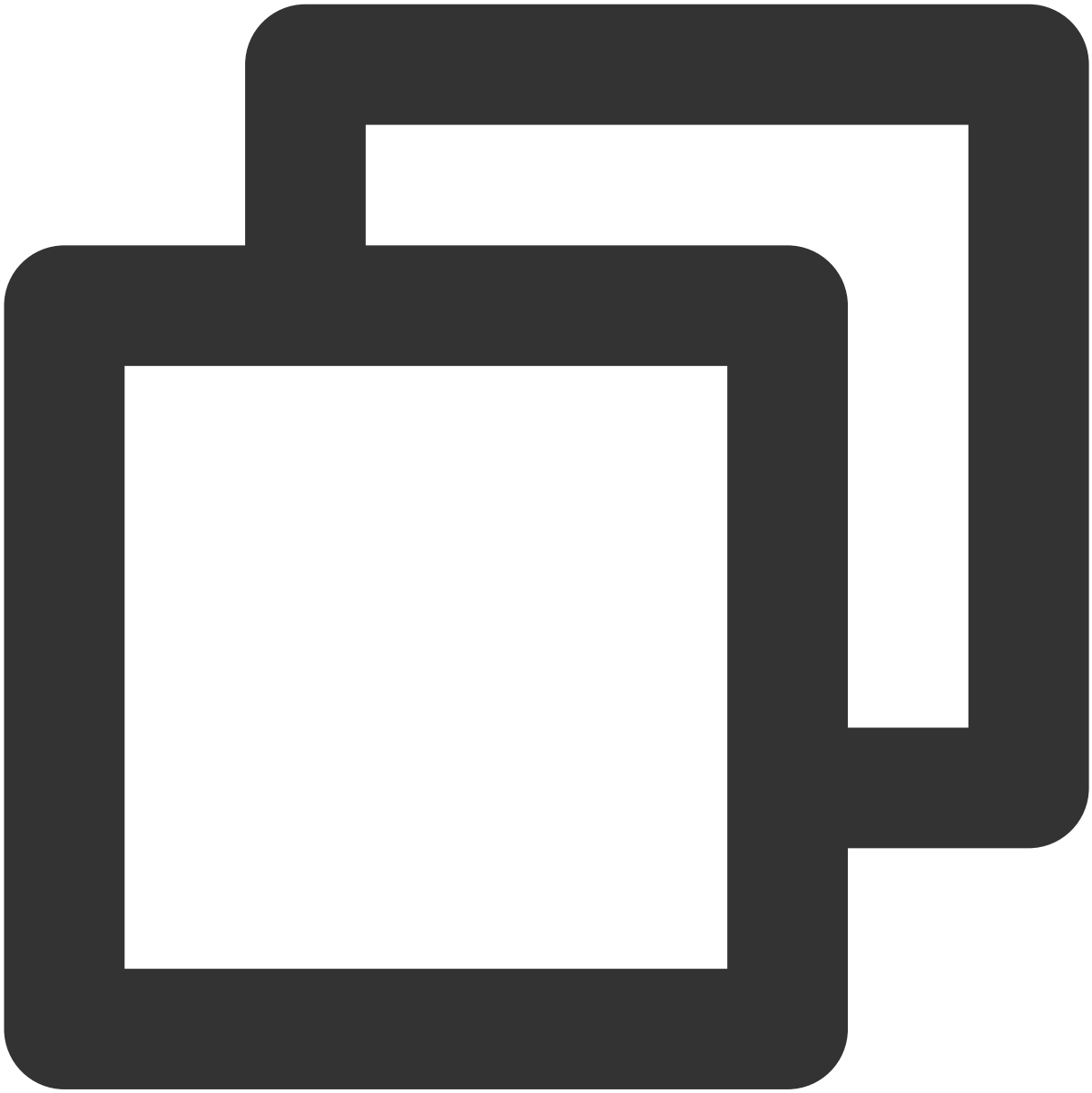
响应

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：

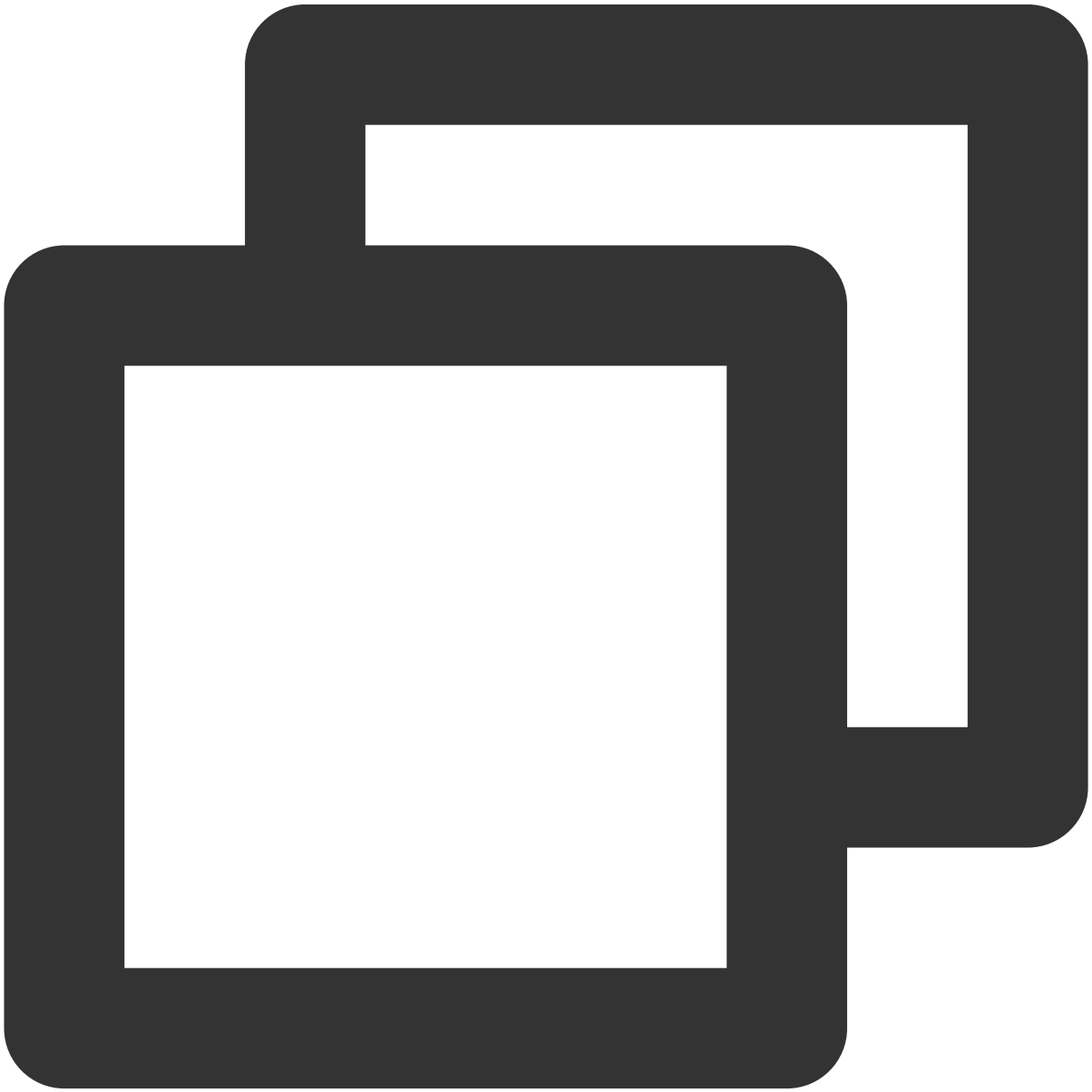
节点名称（关键字）	父节点	描述	类型
Tag	Response	模板类型，Snapshot	String
Name	Response	模板名字	String
TemplateId	Response	模板 ID	String
UpdateTime	Response	更新时间	String
CreateTime	Response	创建时间	String
Snapshot	Response	详细的模板参数，同上述请求体部分 Snapshot 说明	Container

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



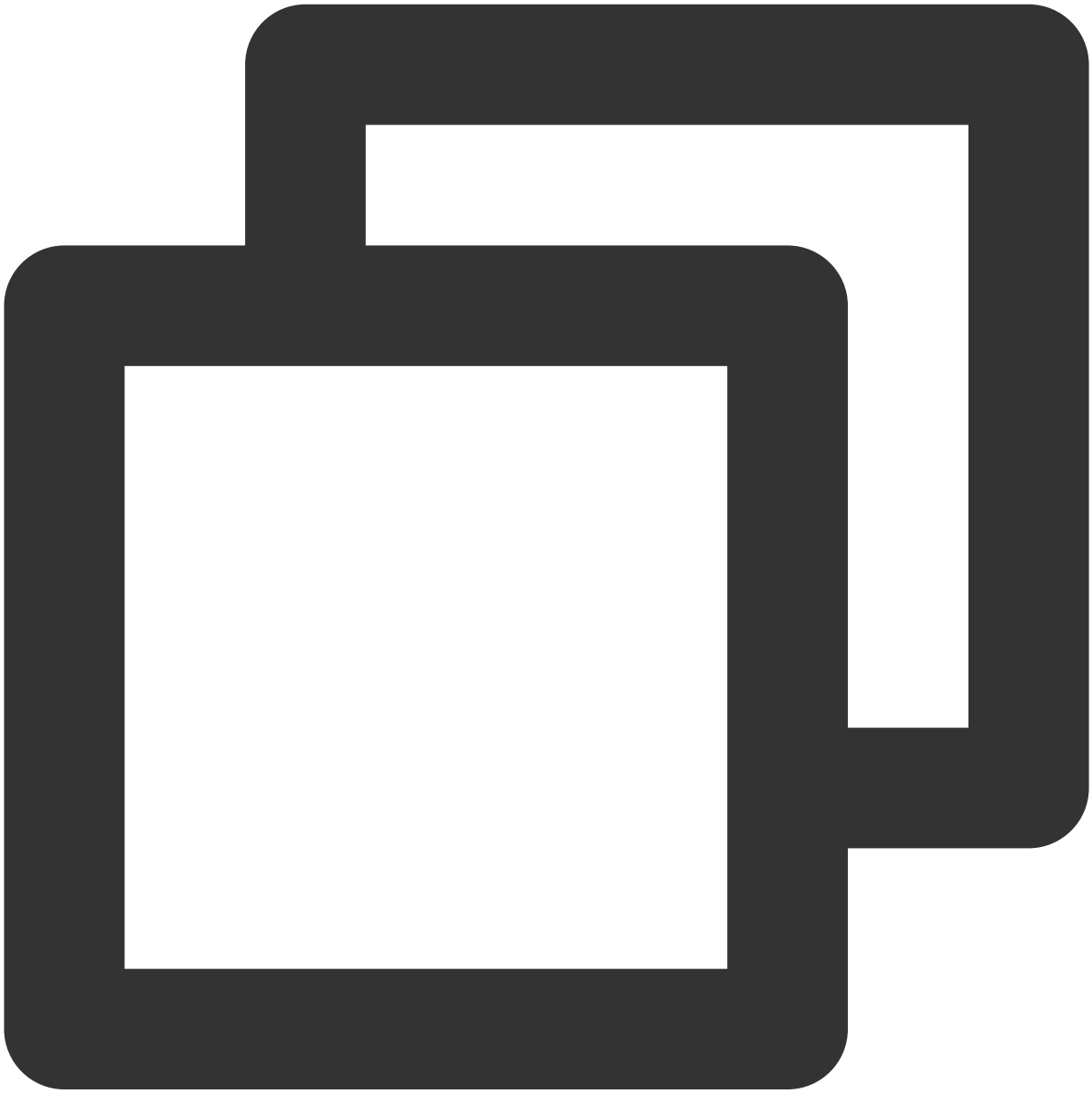
```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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>
```

```
        <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>
```

创建超分辨率模板

最近更新时间：2024-01-23 10:00:02

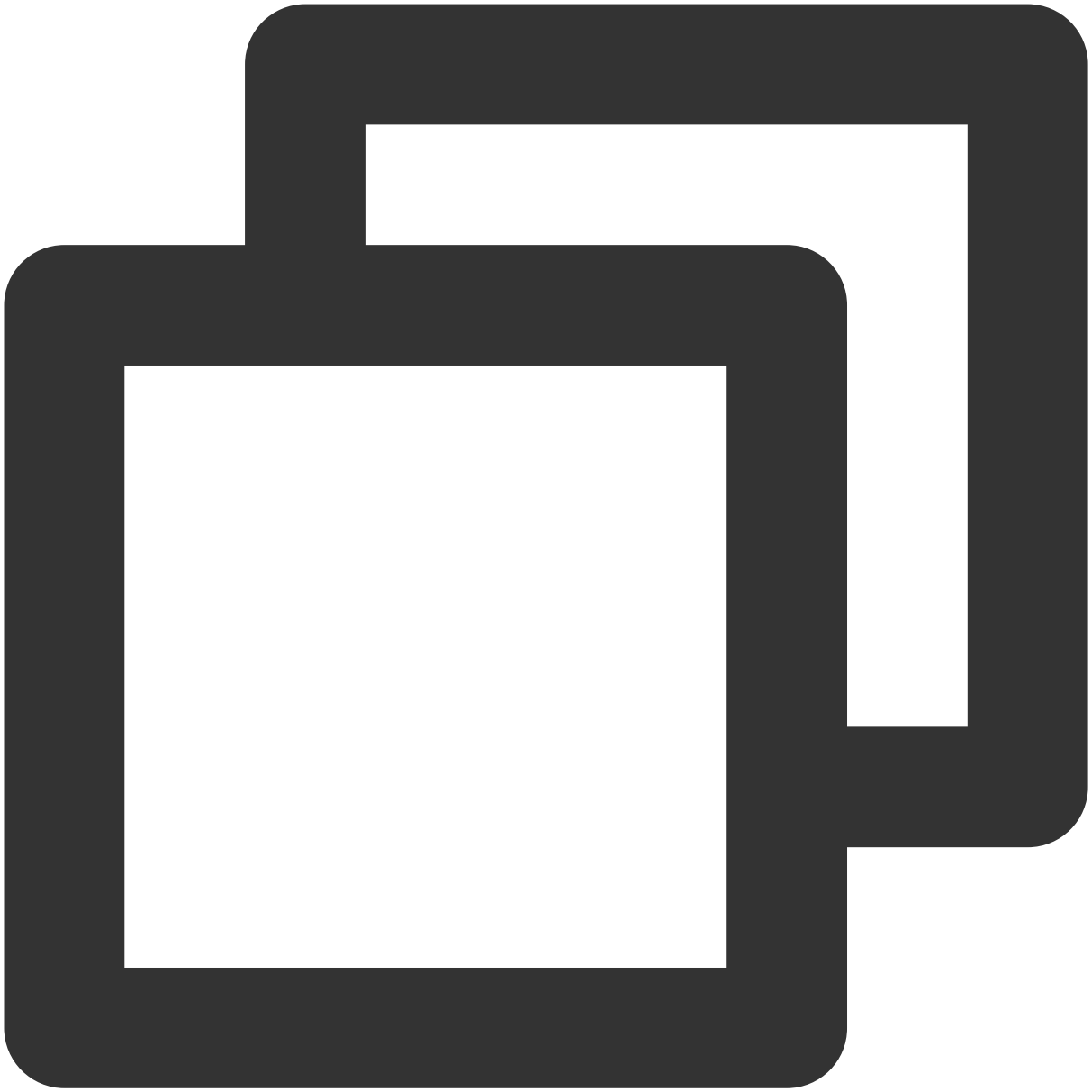
功能描述

创建超分辨率模板。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
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：

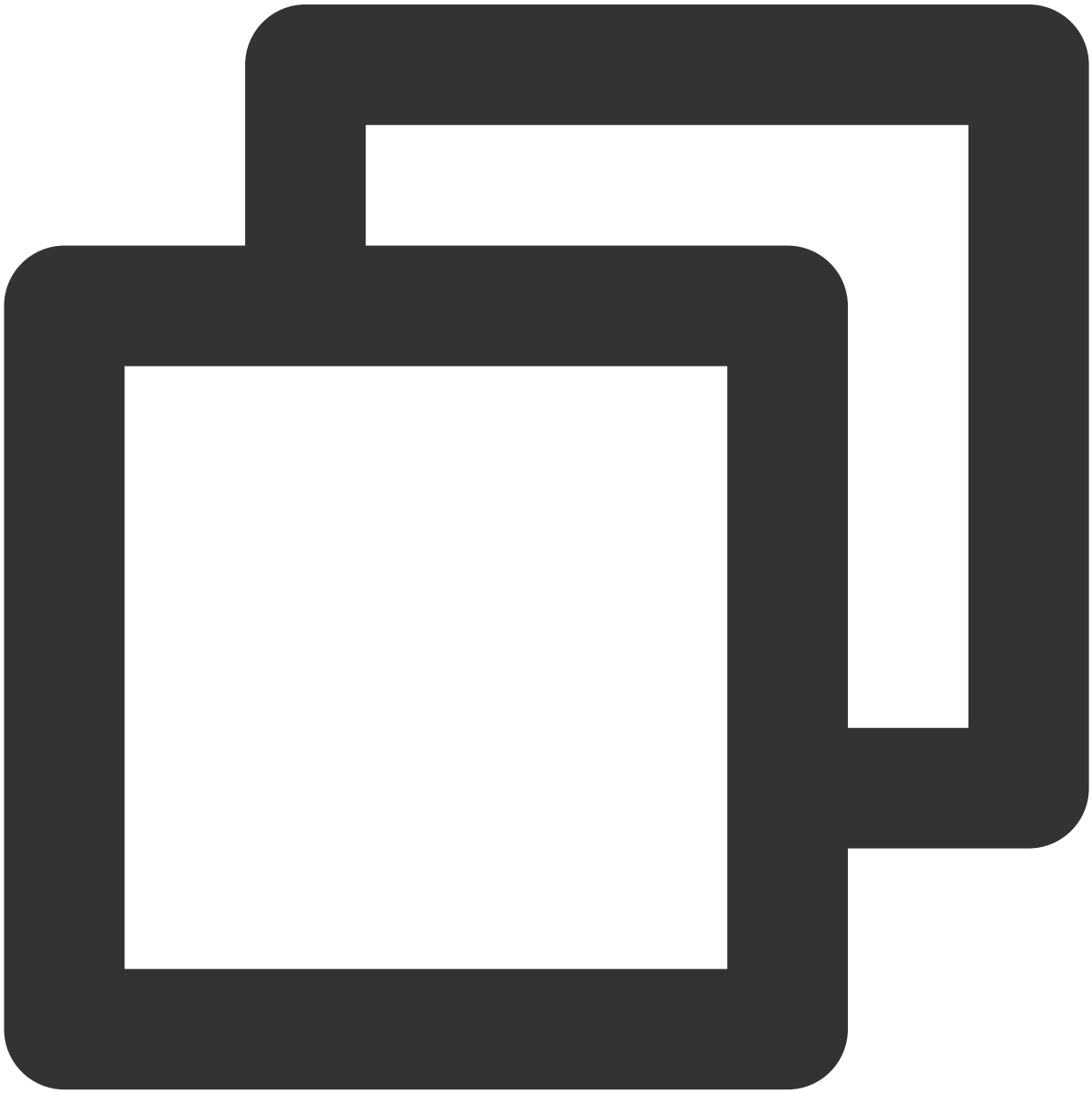
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	sdtohd：标清到超清 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	Base：基础版 Enhance：增强版

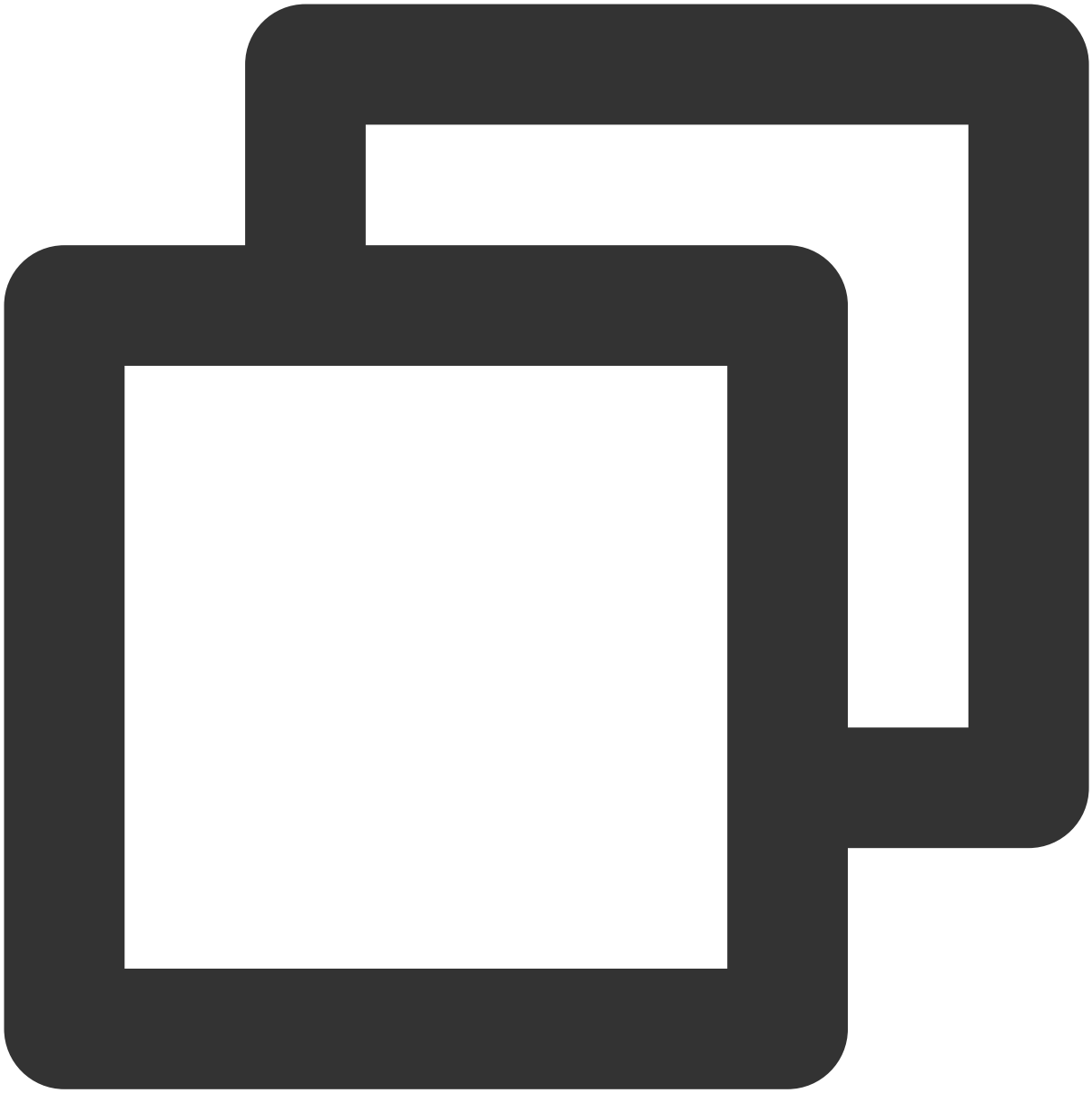
响应

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：

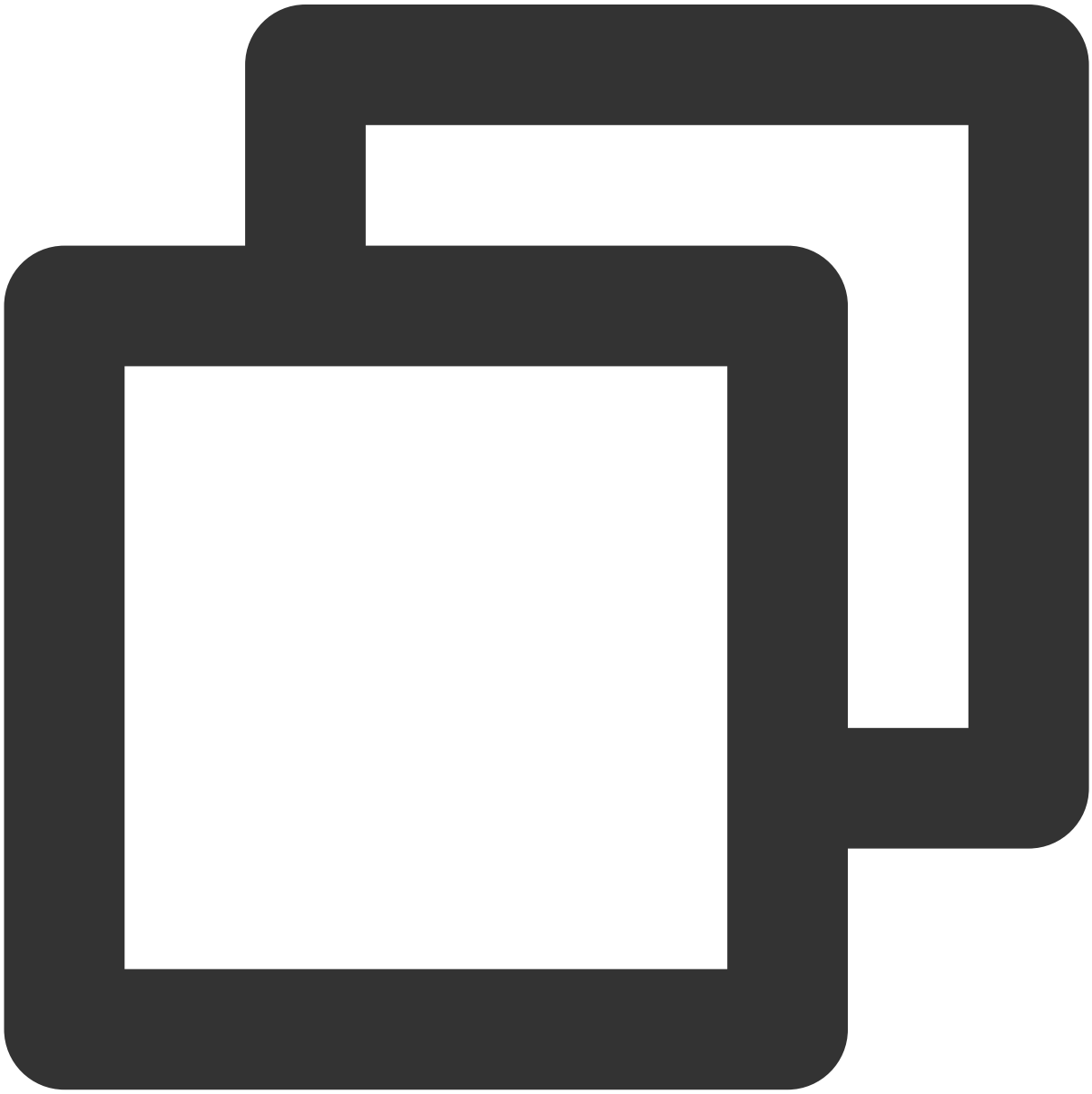
节点名称（关键字）	父节点	描述	类型
Resolution	Response.Template.SuperResolution	同请求体中的 Request.Resolution	String
EnableScaleUp	Response.Template.SuperResolution	同请求体中的 Request.EnableScaleUp	String

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。

实际案例

请求

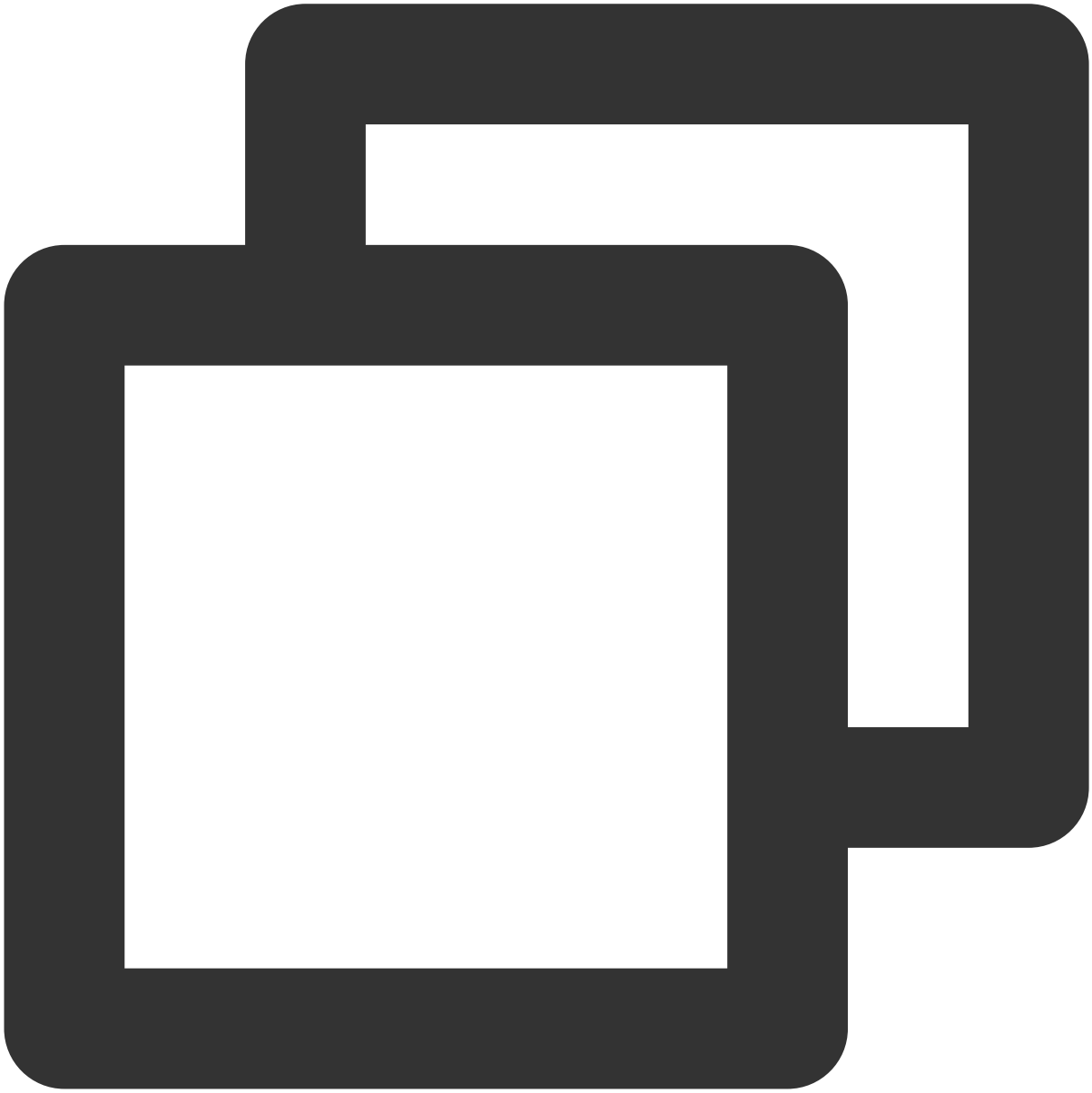


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
```

```
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>
```

创建音视频转码模板

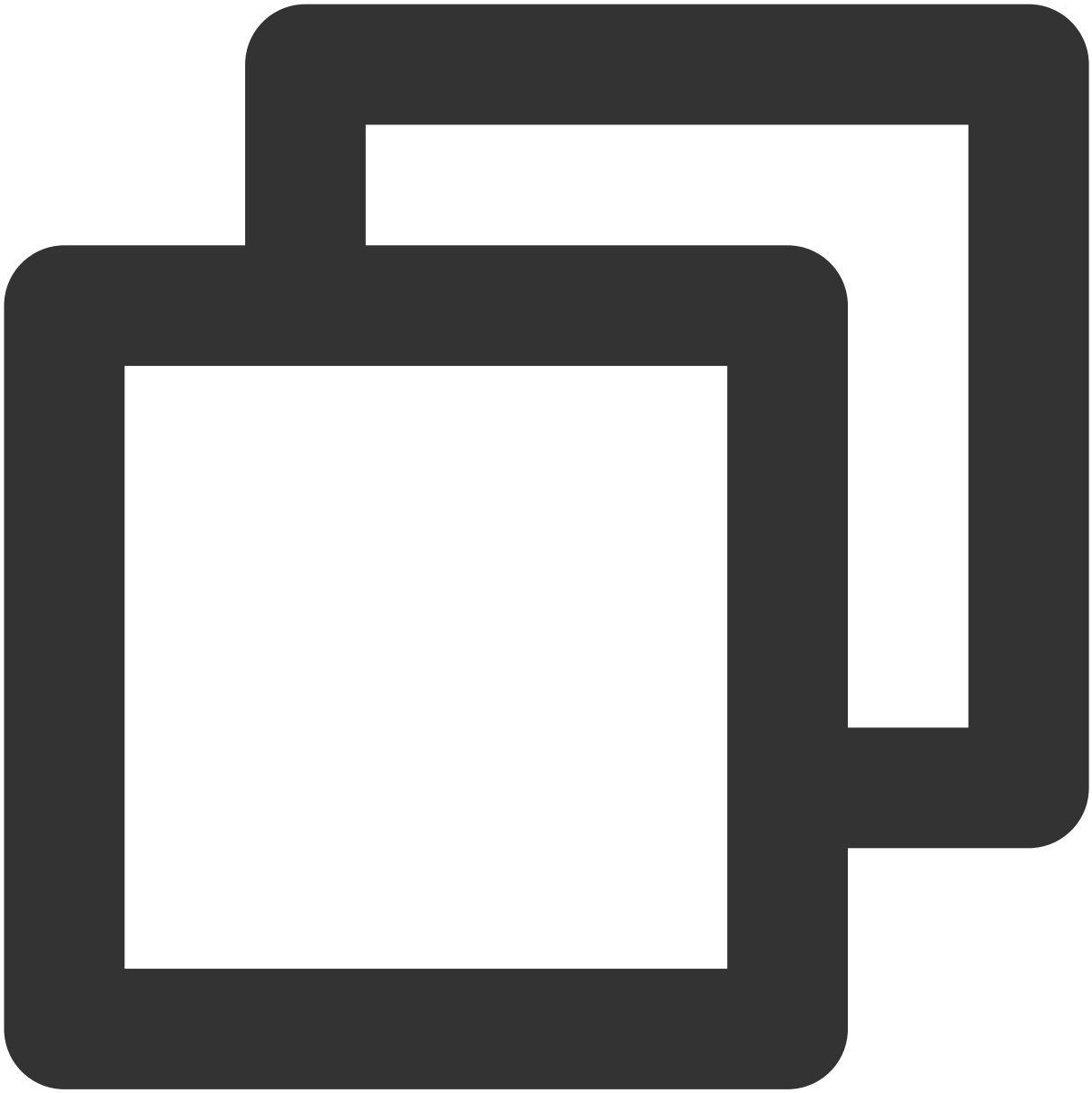
最近更新时间：2024-01-23 10:00:02

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：

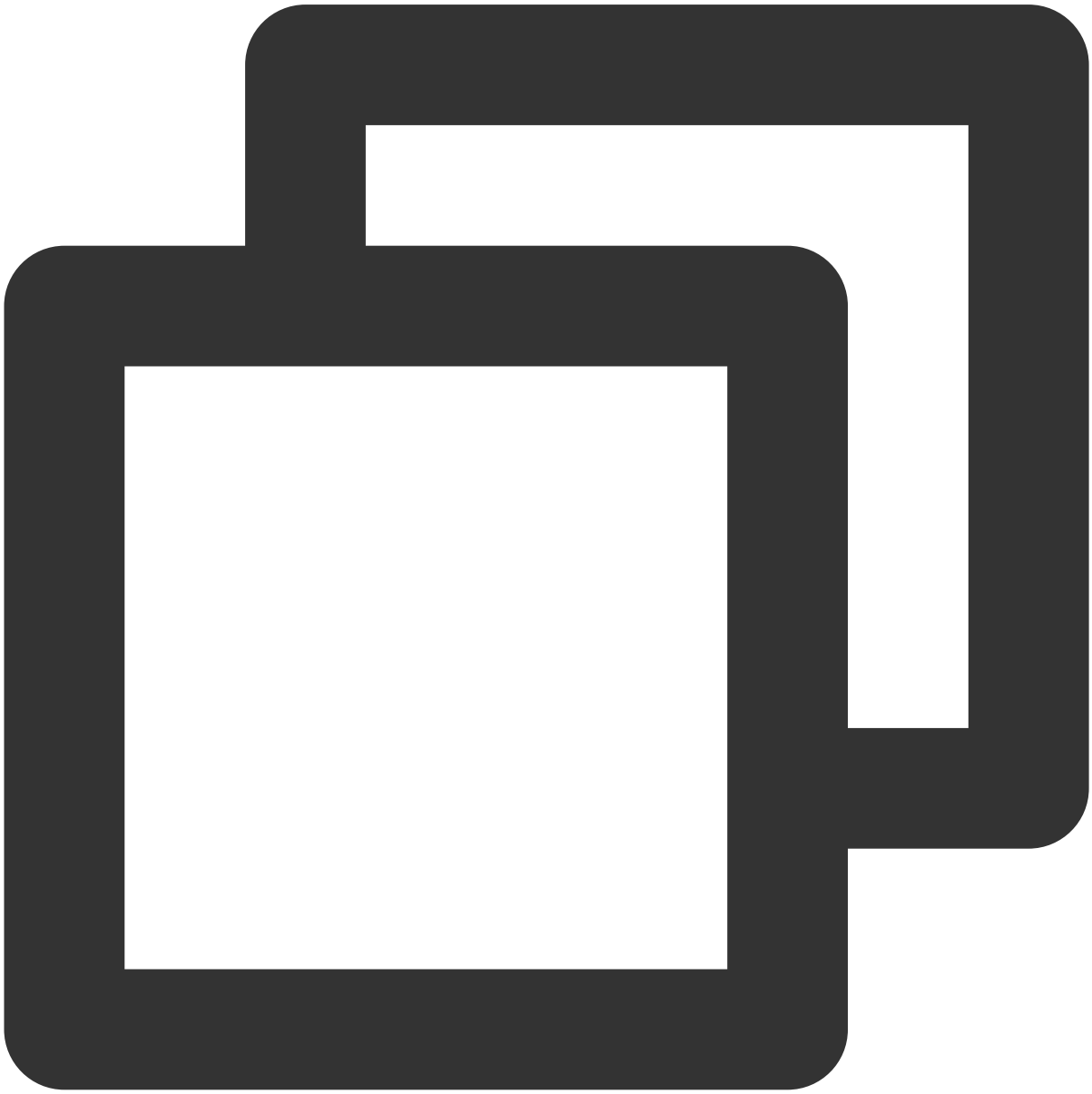
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	Container	否	Audio.Remove

		AudioMixArray	数组		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	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

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	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	String	否	无	值范围：[10, 50000], 单位：Kbps

		文件的码率				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	视频码率峰值	String	否	无	值范围：[10, 50000] 单位：Kbps 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	String	否	false	true、false

		边自适应				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：

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，	单位：Hz 可选 8000、11025、12000、16000、22050、

版权所有：腾讯云计算（北京）有限责任公司

					默认值为48000	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

版权所有：腾讯云计算（北京）有限责任公司

						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

版权所有：腾讯云计算（北京）有限责任公司

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

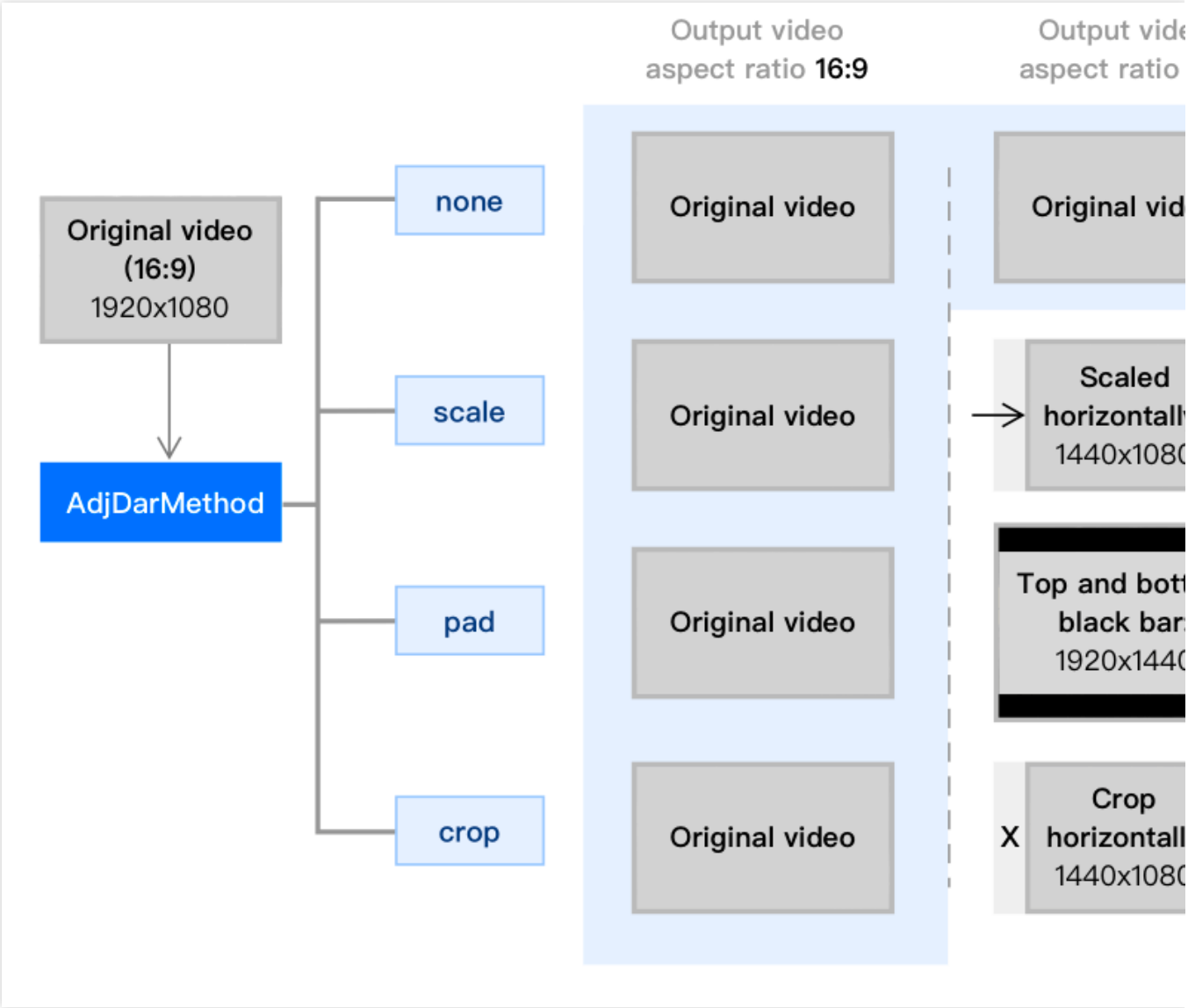
Container 类型 TransConfig 的具体数据描述如下：

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

版权所有：腾讯云计算（北京）有限责任公司

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
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：



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	rue/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（关键字）	父节点	描述	类型	是否必选	默认值	限制
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

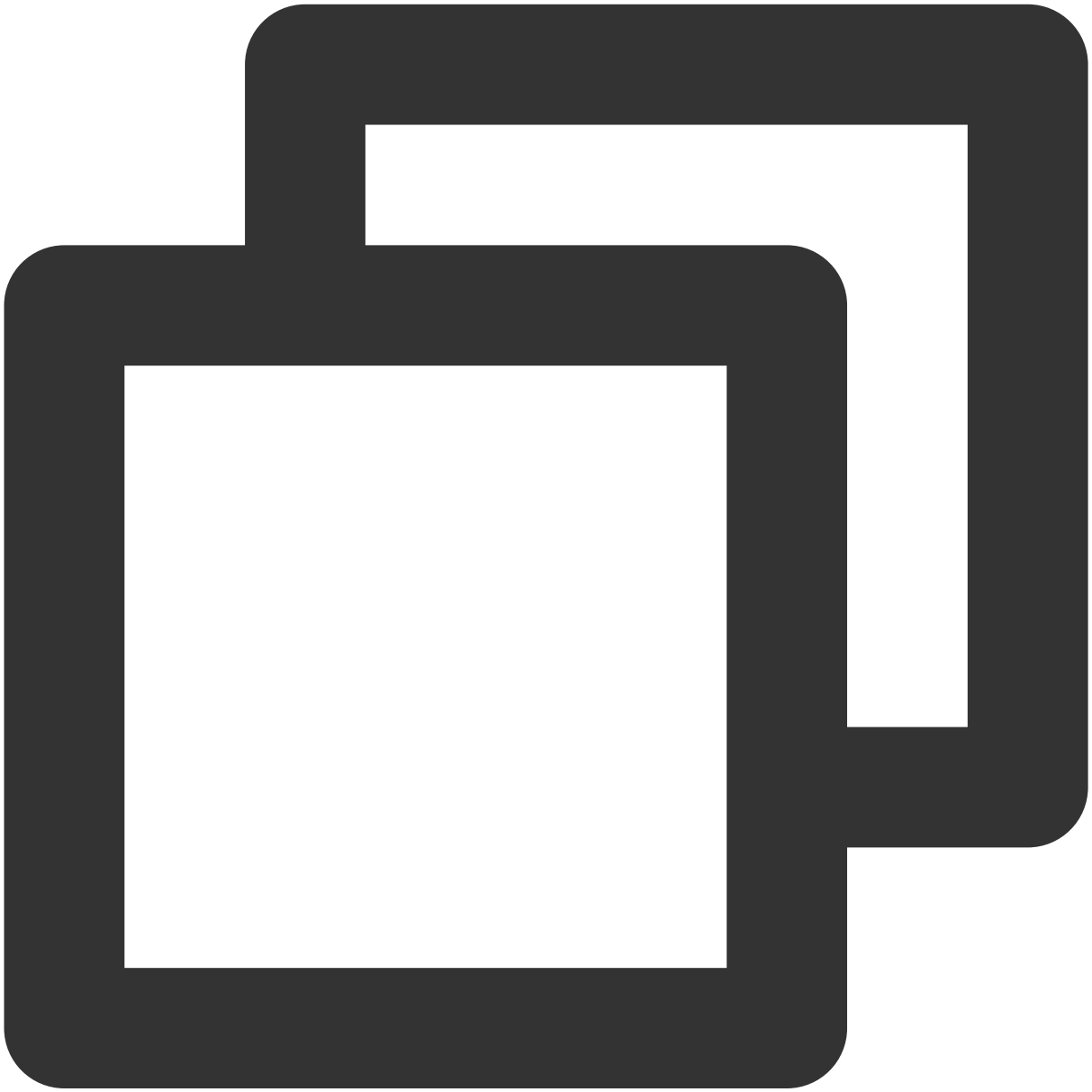
响应

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </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 Request.TransConfig	Container

版权所有：腾讯云计算（北京）有限责任公司

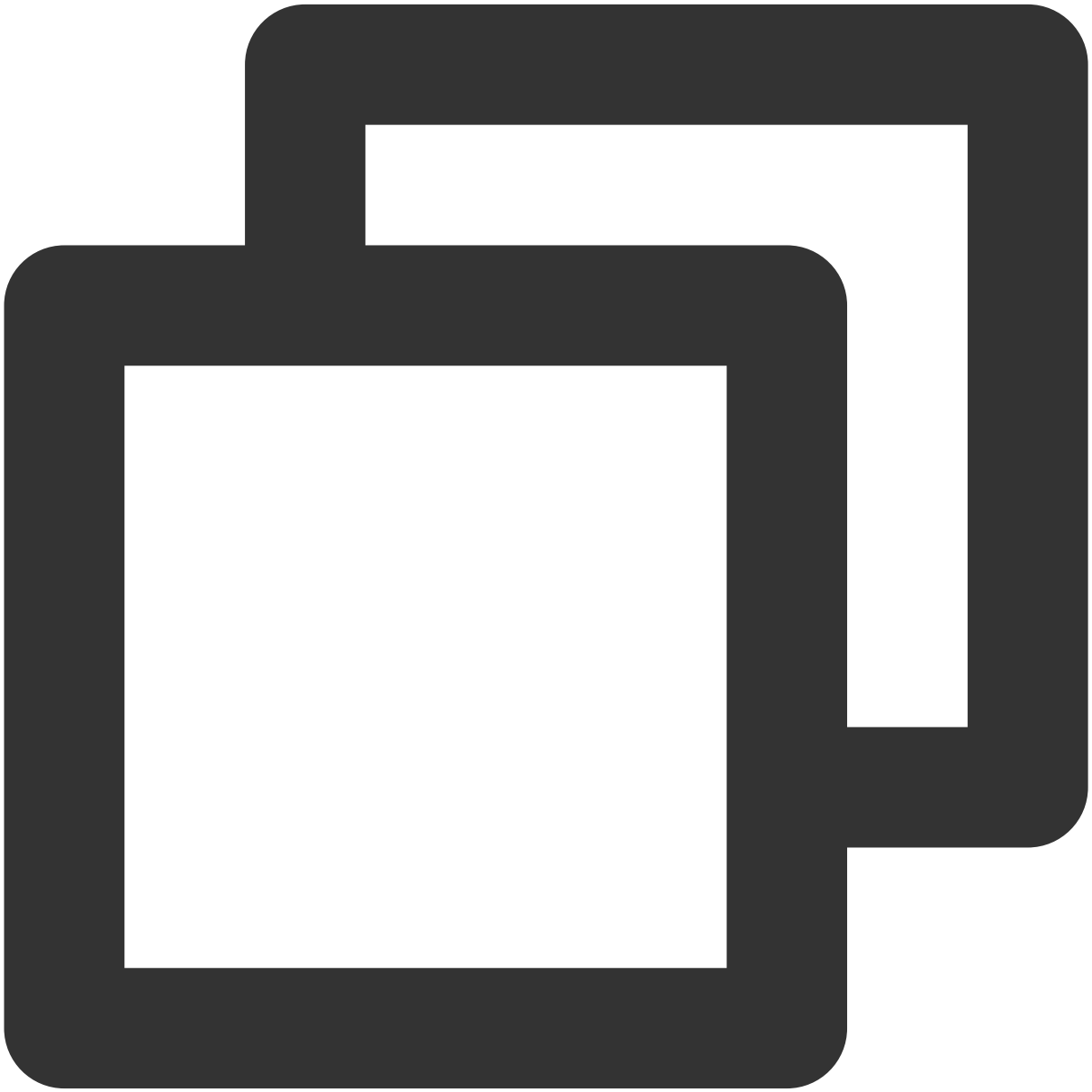
AudioMix	Response.Template.TransTpl	同请求体中的 Request.AudioMix	Container
AudioMixArray	Response.Template.TransTpl	同请求体中的 Request.AudioMixArray	Container 数组

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

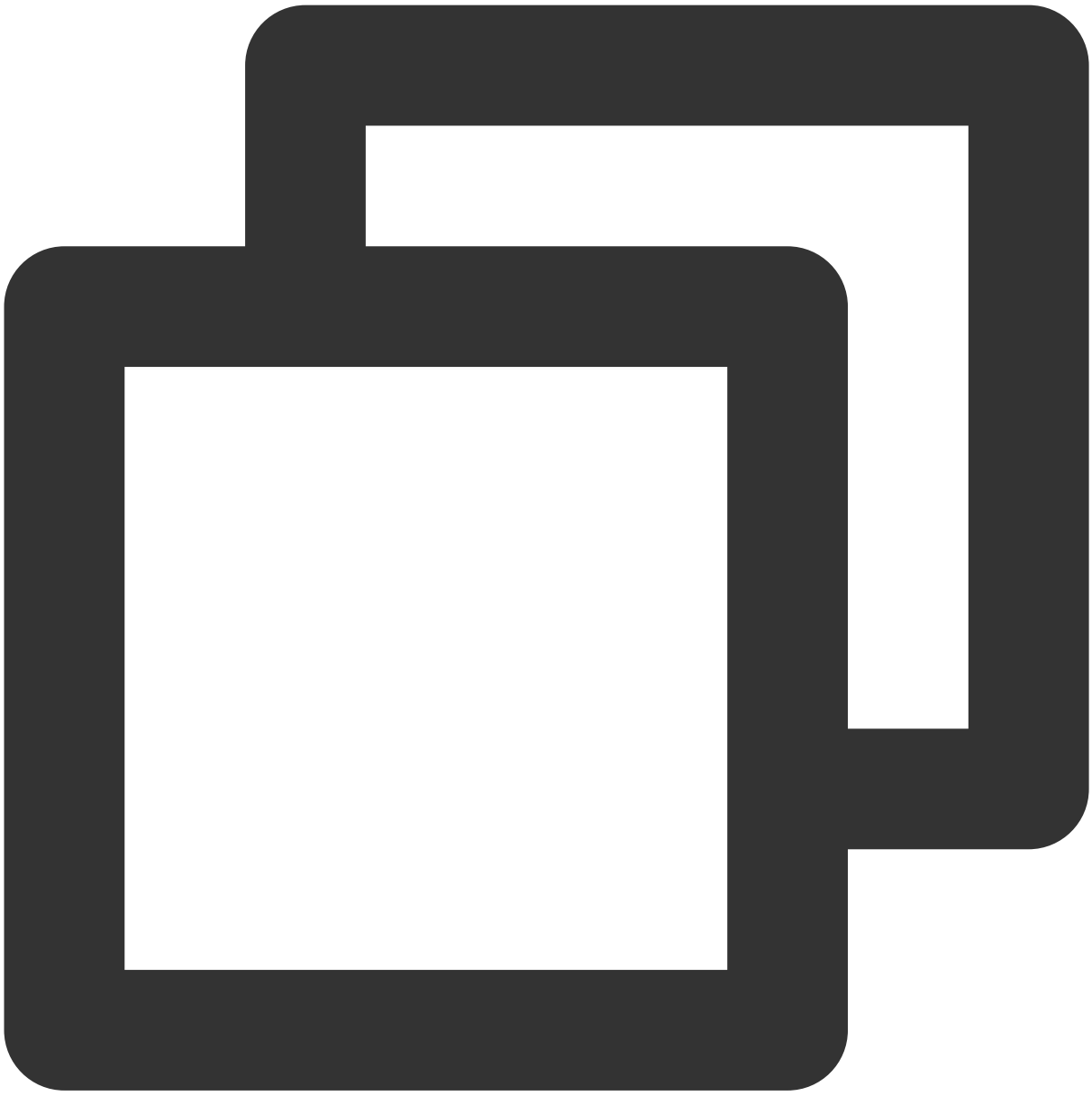
<Request>
    <Tag>Transcode</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mp4</Format>
```

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

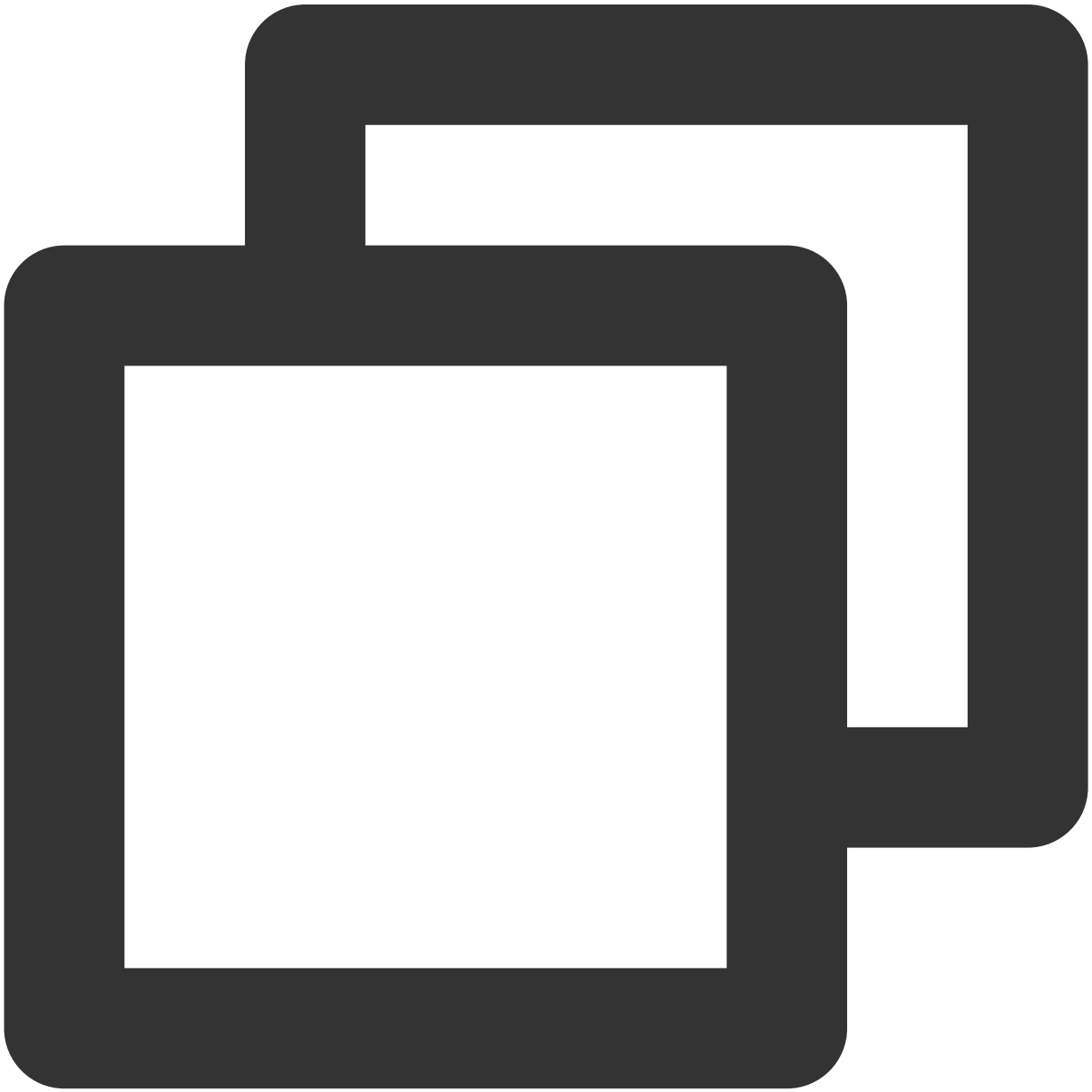
最近更新时间：2024-01-23 10:00:02

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：

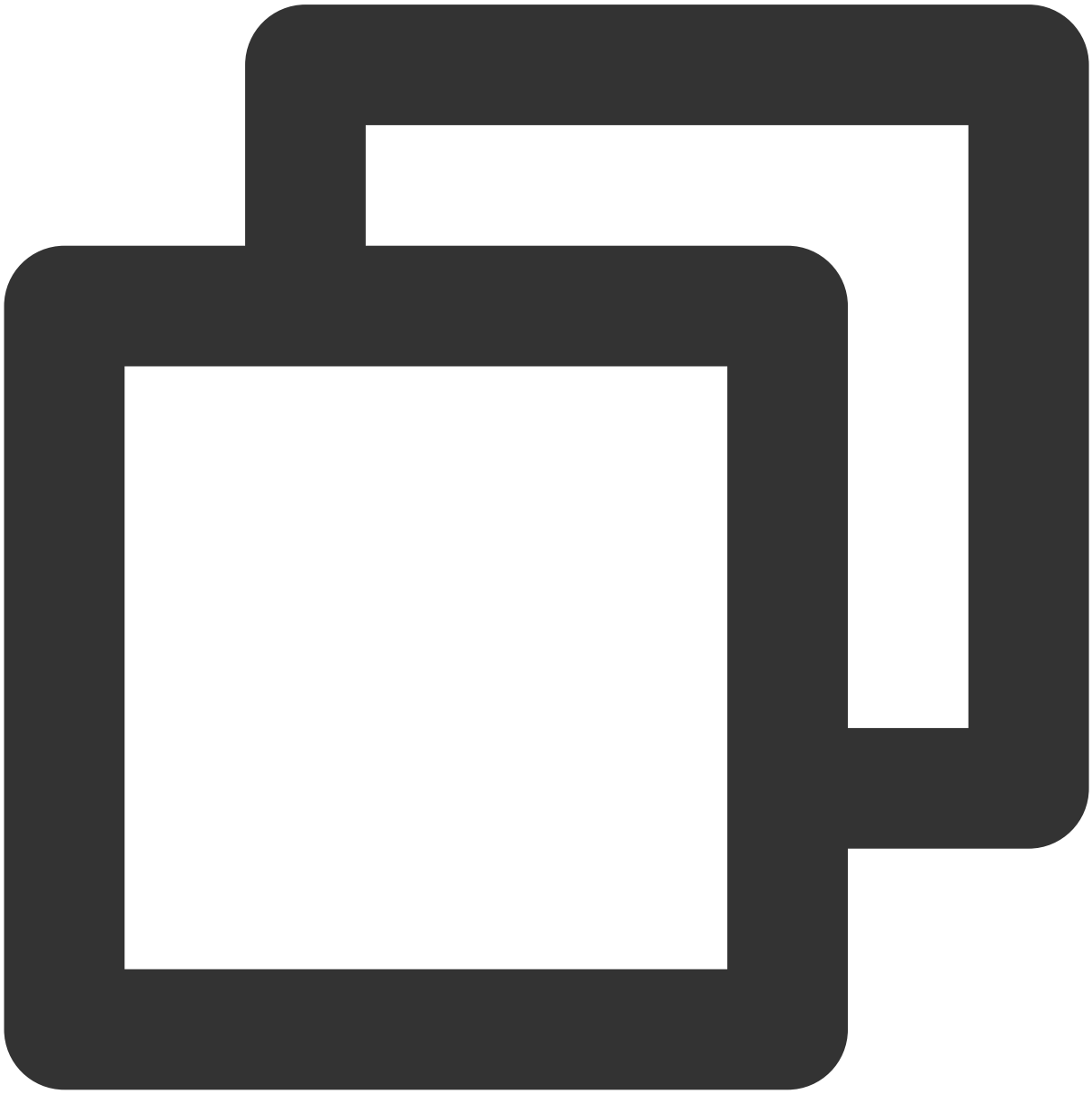
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	-

版权所有：腾讯云计算（北京）有限责任公司

				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 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	-

版权所有：腾讯云计算（北京）有限责任公司

				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 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	1、25000或35000或50000 2、35000或50000 3、25000或35000或50000 4、25000或35000或

版权所有：腾讯云计算（北京）有限责任公司

				5、60000/1001	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 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	取值范围:[2000,3000000]

版权所有：腾讯云计算（北京）有限责任公司

			2、50 3、24000/1001 4、30000/1001 5、60000/1001	
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 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
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	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	
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]

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

版权所有：腾讯云计算（北京）有限责任公司

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：

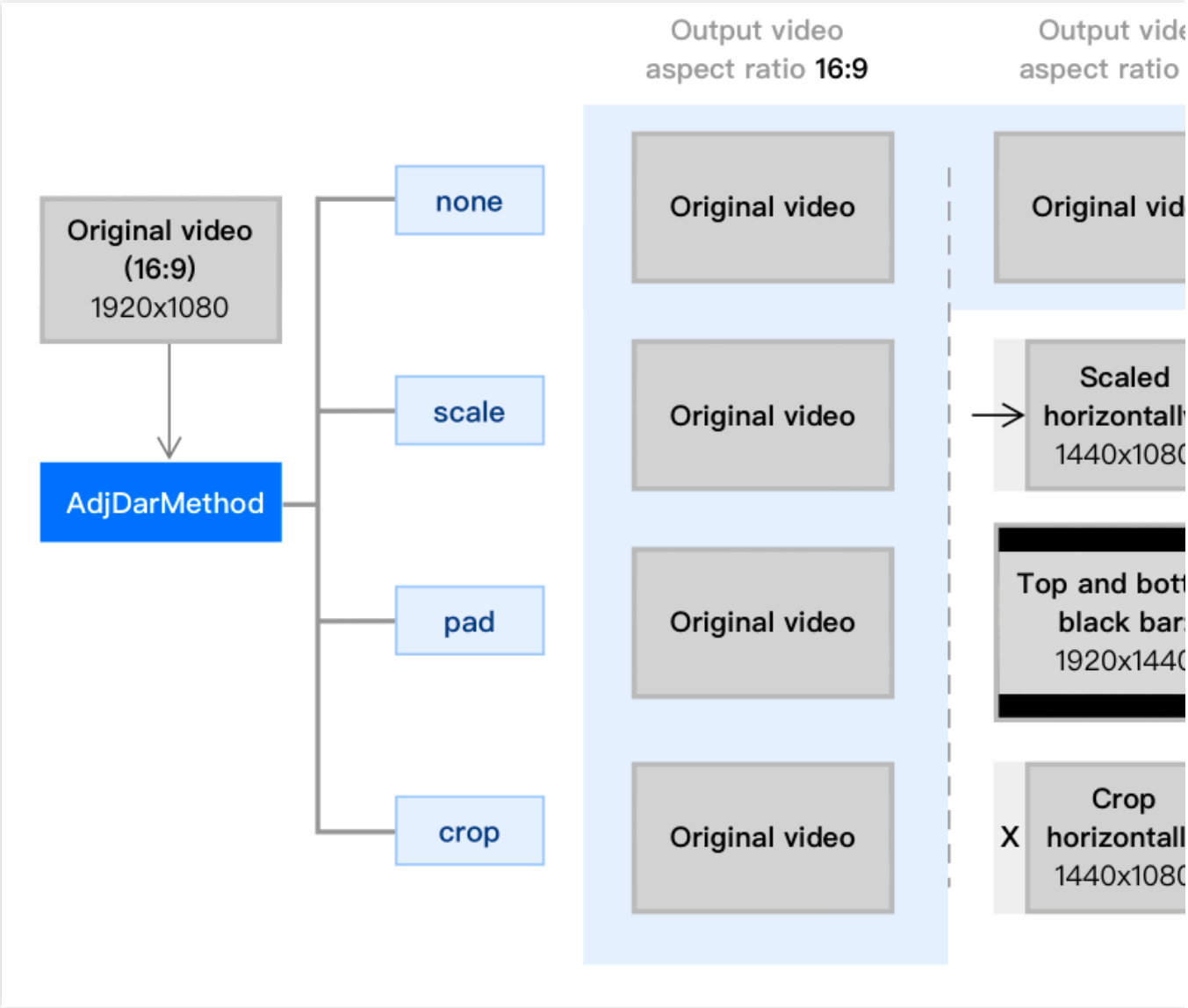
节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
AdjDarMethod	Request.TransConfig	分辨率调整方式	String	否	none	取值 scale、crop、pad、none 当输出视频的宽高比与原视频不等时，根据此参数做分辨率的相应调整
IsCheckReso	Request.TransConfig	是否检查分辨率	String	否	false	true、false 当为 false时，按照配置参数转码
ResoAdjMethod	Request.TransConfig	分辨率调整方式	String	否	0	取值0、1；0 表示使用原视频分辨率；1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

版权所有：腾讯云计算（北京）有限责任公司

						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：

版权所有：腾讯云计算（北京）有限责任公司



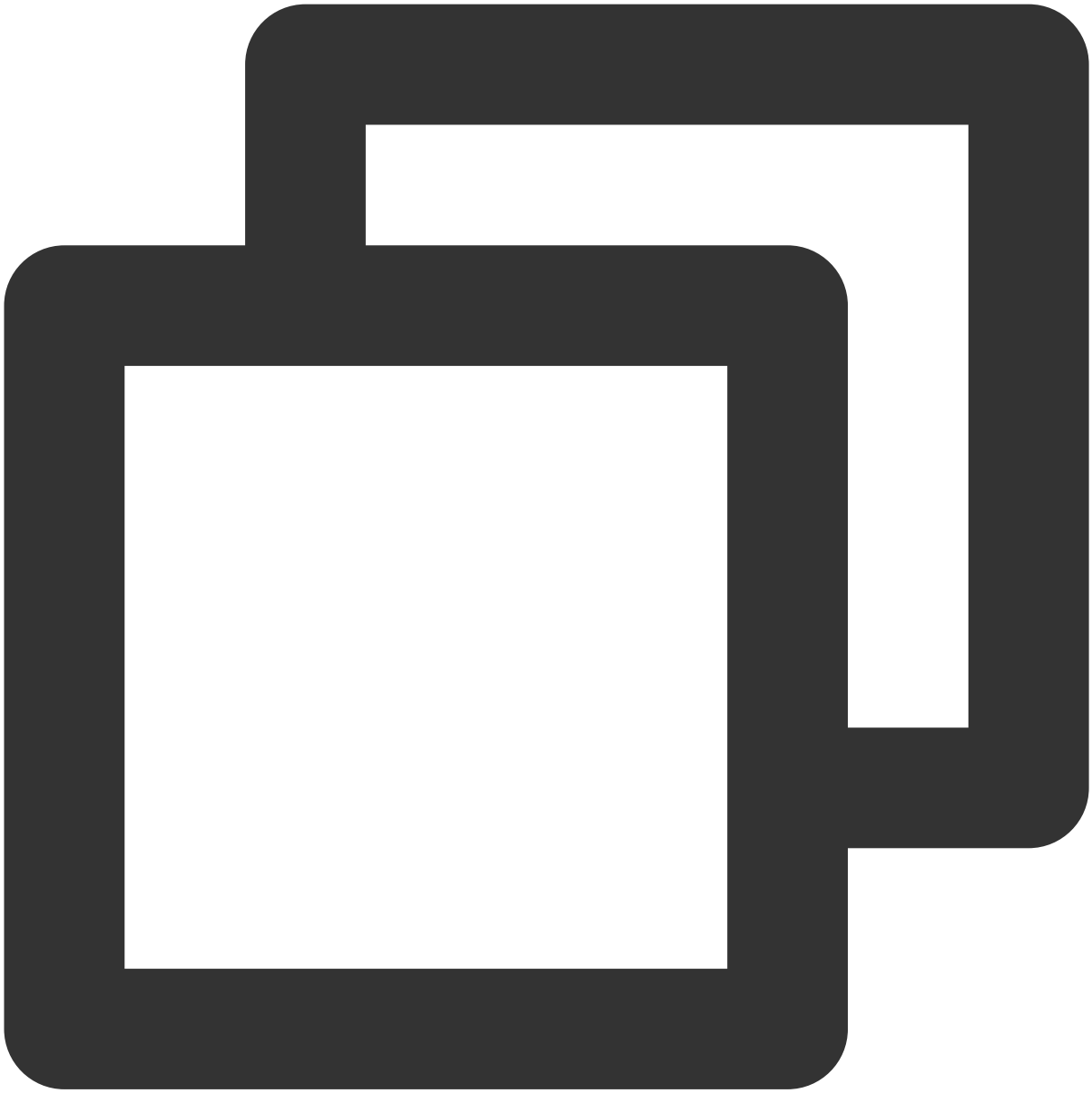
响应

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>
    <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：

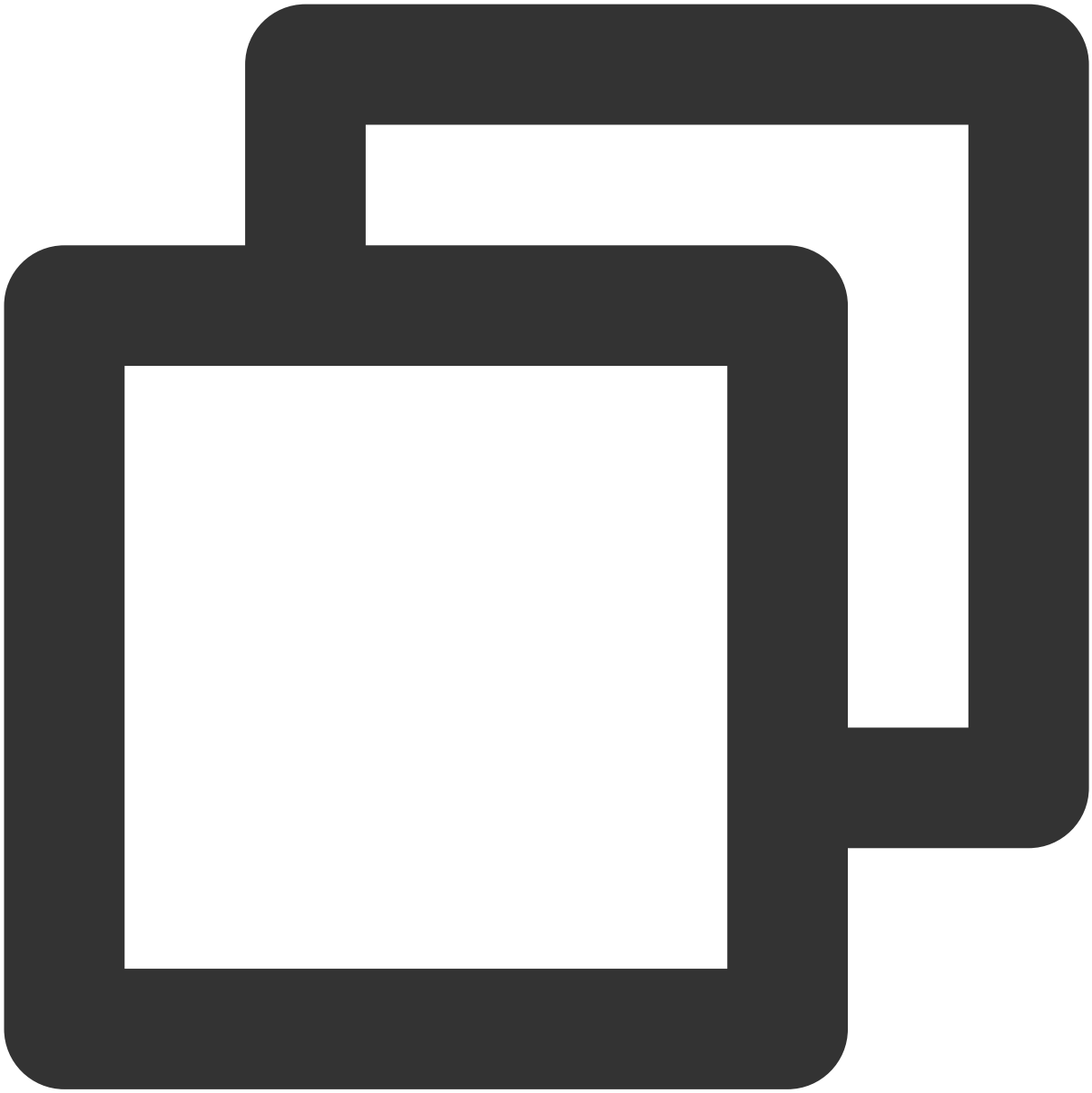
节点名称（关键字）	父节点	描述	类型
TimeInterval	Response.Template.TransProTpl	同请求体中的 Request.TimeInterval	Container
Container	Response.Template.TransProTpl	同请求体中的 Request.Container	Container
Video	Response.Template.TransProTpl	同请求体中的 Request.Video	Container
Audio	Response.Template.TransProTpl	同请求体中的 Request.Audio	Container
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

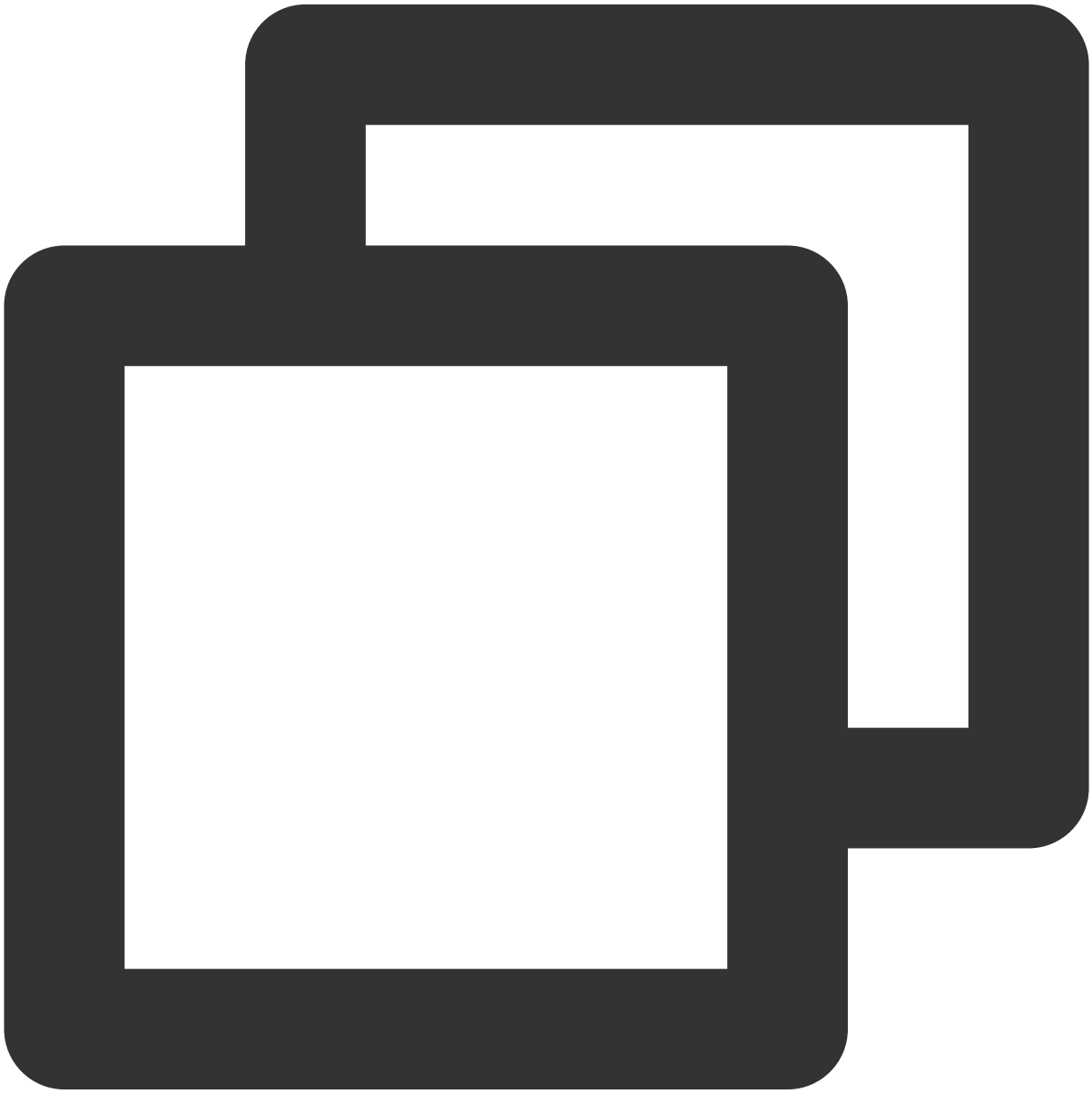


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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>
```

创建语音合成模板

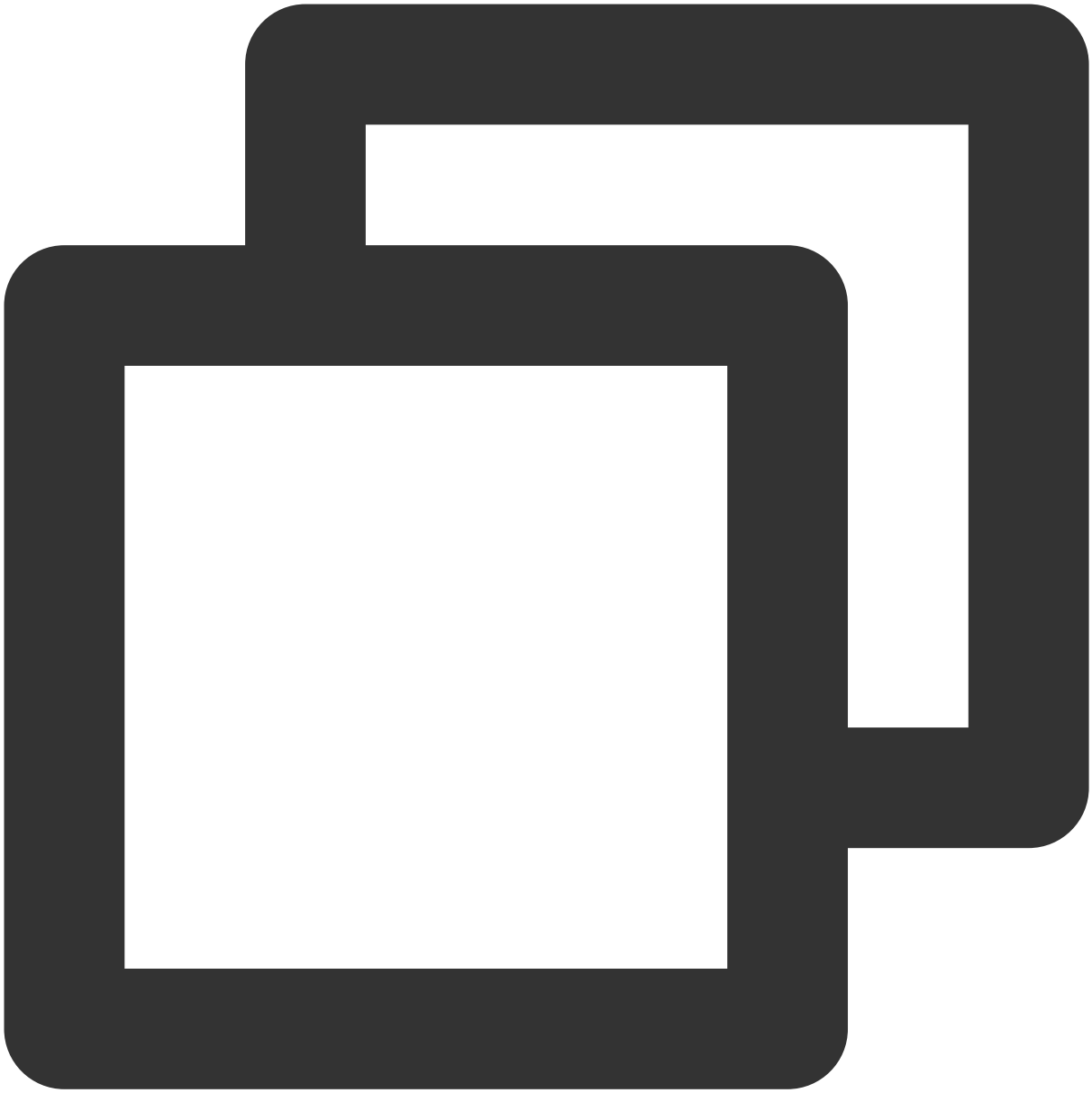
最近更新时间：2024-01-23 10:00:02

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：

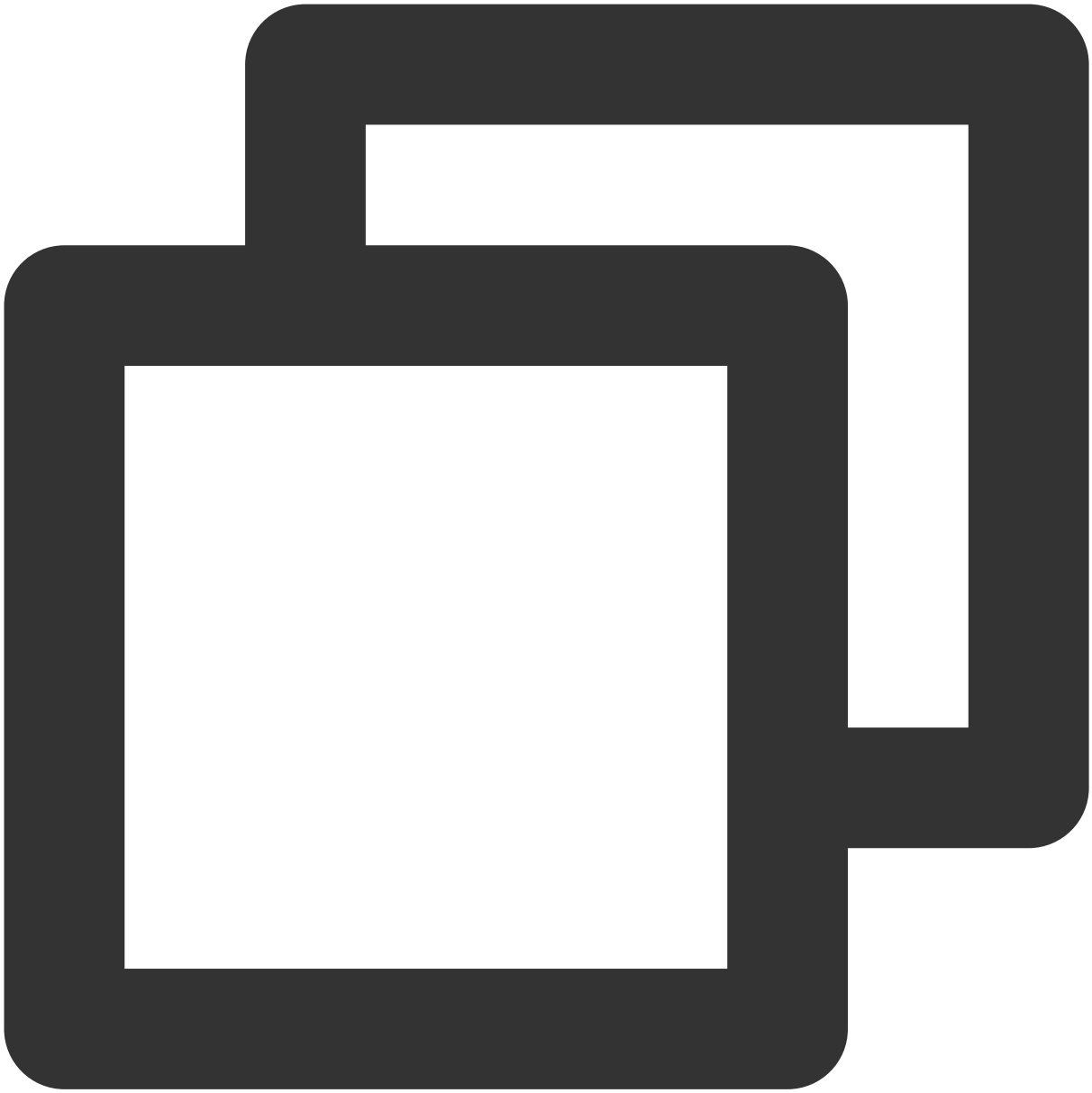
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	详见下表
Volume	Request	音量，默认值0	String	否	[-10,10]
Speed	Request	语速，默认值100	String	否	[50,200]

支持的音色

名称	Voice 参数值	Sync（同步合成）	Asyc（异步合成）
茹雪	ruxue	支持	支持

版权所有：腾讯云计算（北京）有限责任公司

爱小楠	aixiaonan	不支持	支持
爱小星	aixiaoxing	支持	支持
爱小诚	aixiaocheng	支持	支持
爱小雪	aixiaoxue	支持	支持
爱小璐	aixiaolu	支持	支持
爱小栋	aixiaodong	支持	支持
爱小聊	aixiaoliao	不支持	支持
爱小倩	aixiaoqian	支持	支持
爱小阳	aixiaoyang	支持	支持
Alice	alice	支持	支持

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	中文及中英文混合	标准版
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	元气男声	通用/解说场景	中文及中英文混合	精品版
爱小诚	aixiaocheng	标准男声	通用场景	中文及中英文混合	标准版
爱小雪	aixiaoxue	标准女声	通用/客服场景	中文及中英文混合	标准版
爱小璐	aixiaolu	阅读女声	通用/有声书场景	中文及中英文混合	标准版
爱小栋	aixiaodong	新闻男声	通用/新闻播报场景	中文及中英文混合	标准版
爱小聊	aixiaoliao	情绪女声	通用/社交场景	中文及中英文混合	精品版
爱小倩	aixiaoqian	元气女声	通用/社交场景	中文及中英文混合	精品版
爱小阳	aixiaoyang	广播男声	通用/新闻播报场景	中文及中英文混合	标准版
Alice	alice	英文女声	通用场景	英文	标准版

多情感音色说明

名称	Voice 参数值	情绪分类

版权所有：腾讯云计算（北京）有限责任公司

爱小星	aixiaoxing	中性、兴奋
爱小诚	aixiaocheng	中性、广播
爱小雪	aixiaoxue	中性、广播、客服
爱小璐	aixiaolu	中性、故事、客服

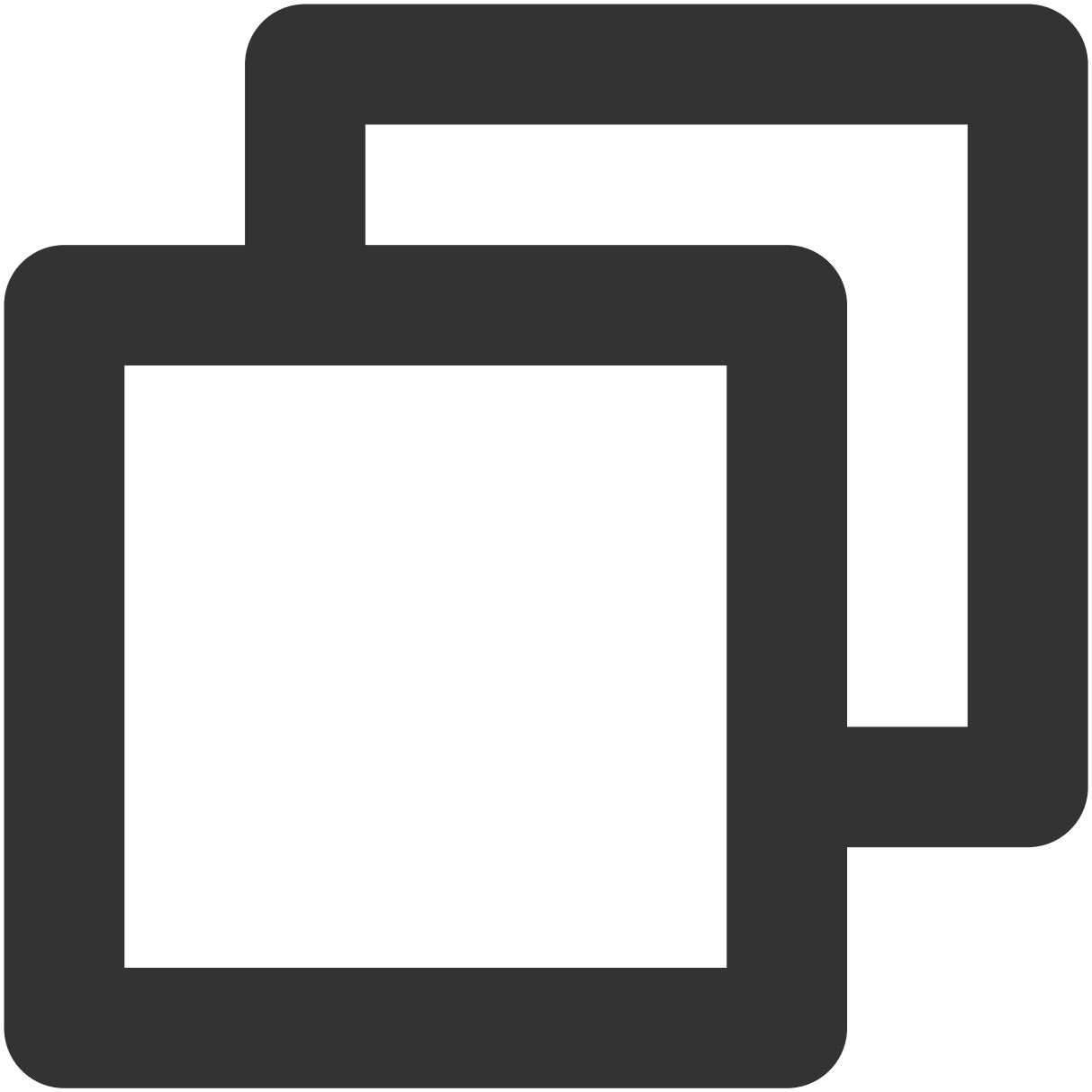
响应

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

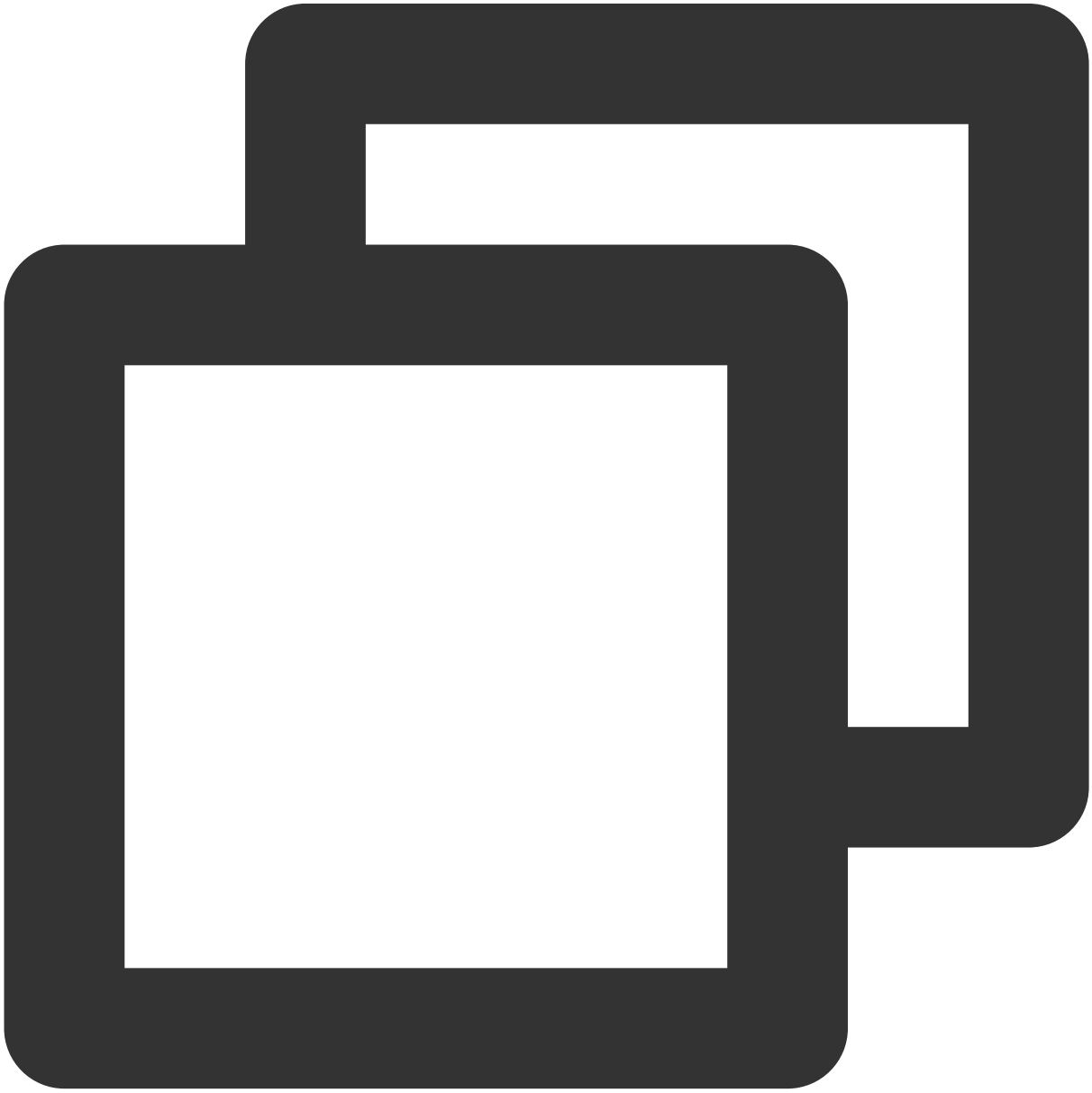
版权所有：腾讯云计算（北京）有限责任公司

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

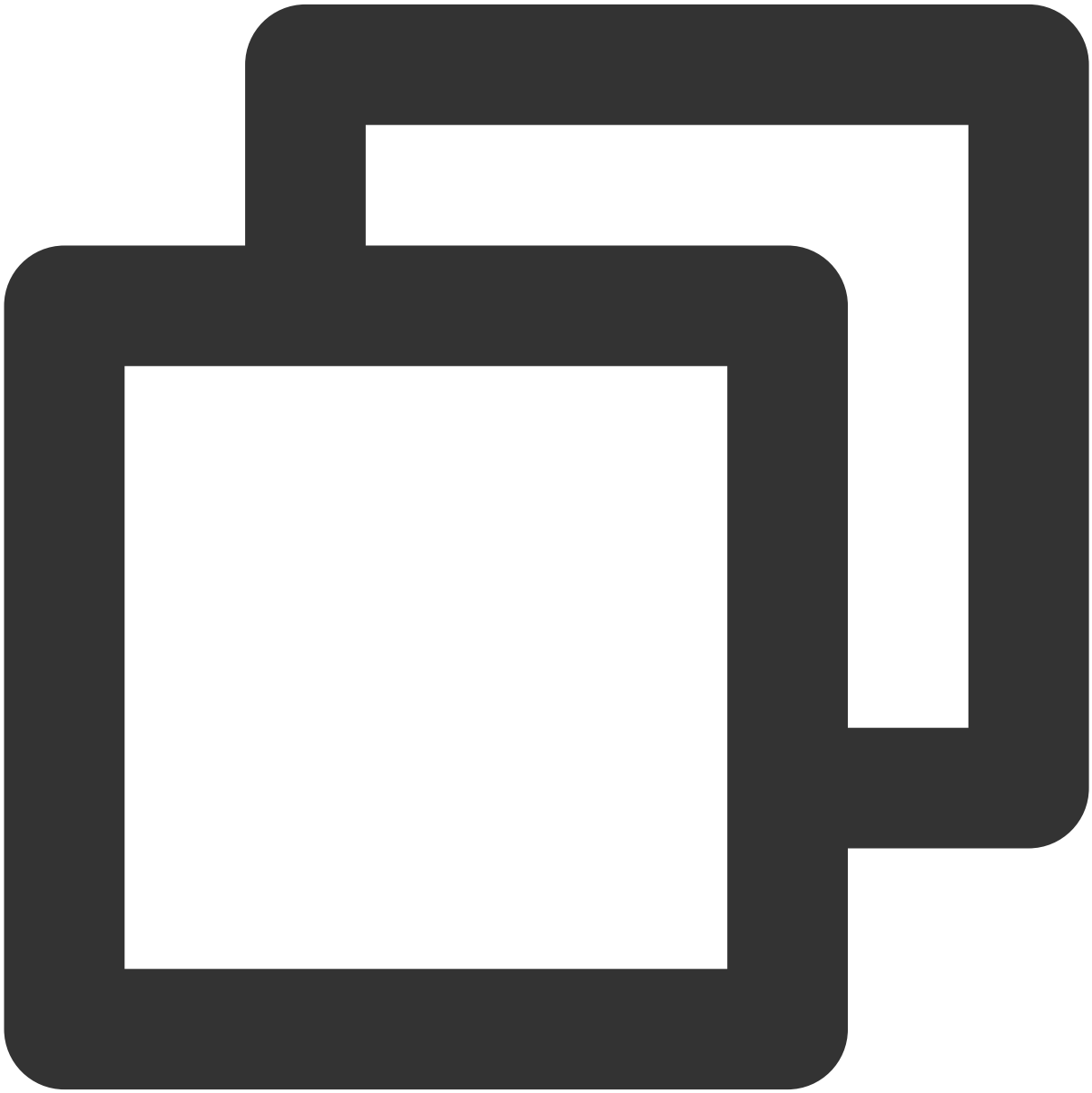


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
```

```
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

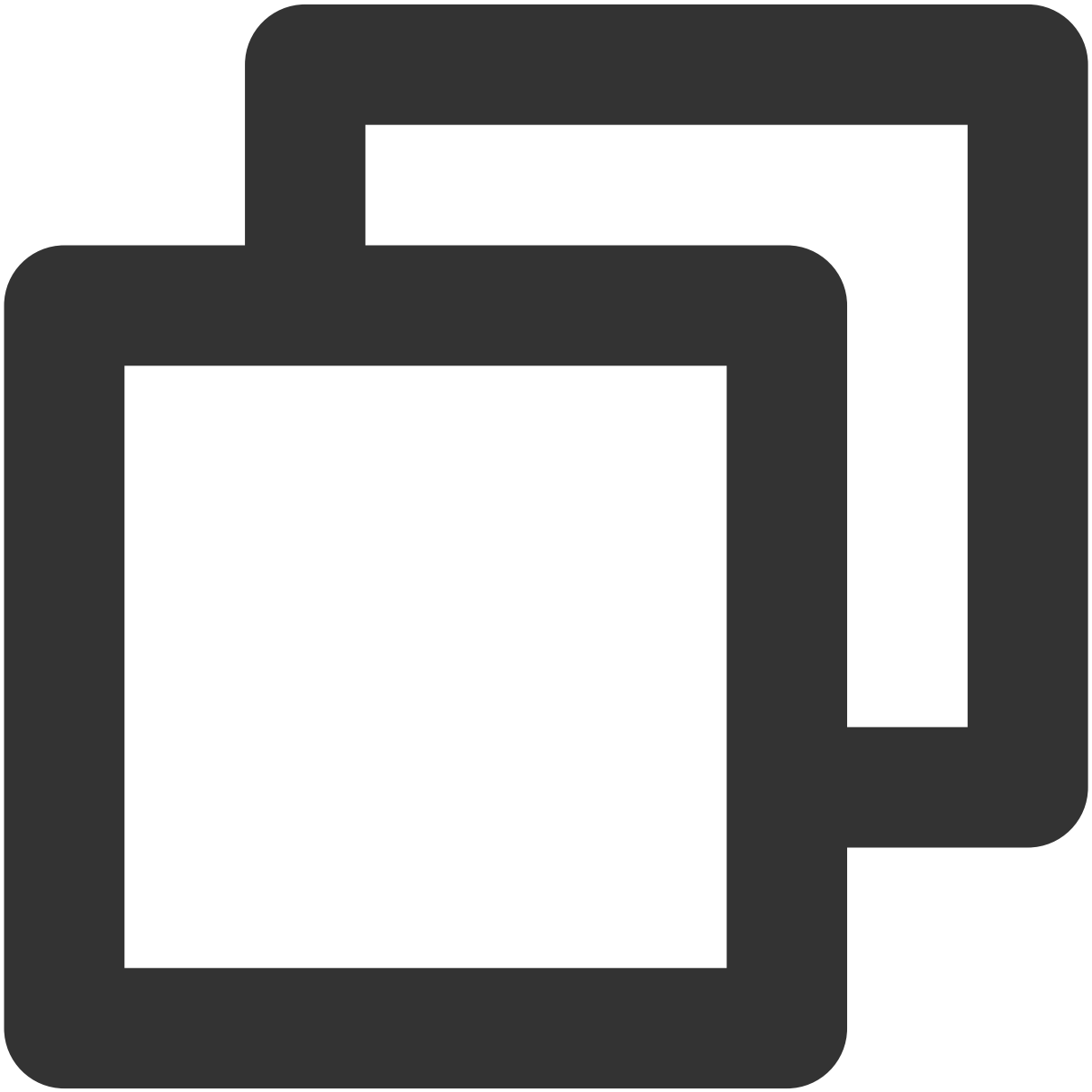
最近更新时间：2024-01-23 10:00:02

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：

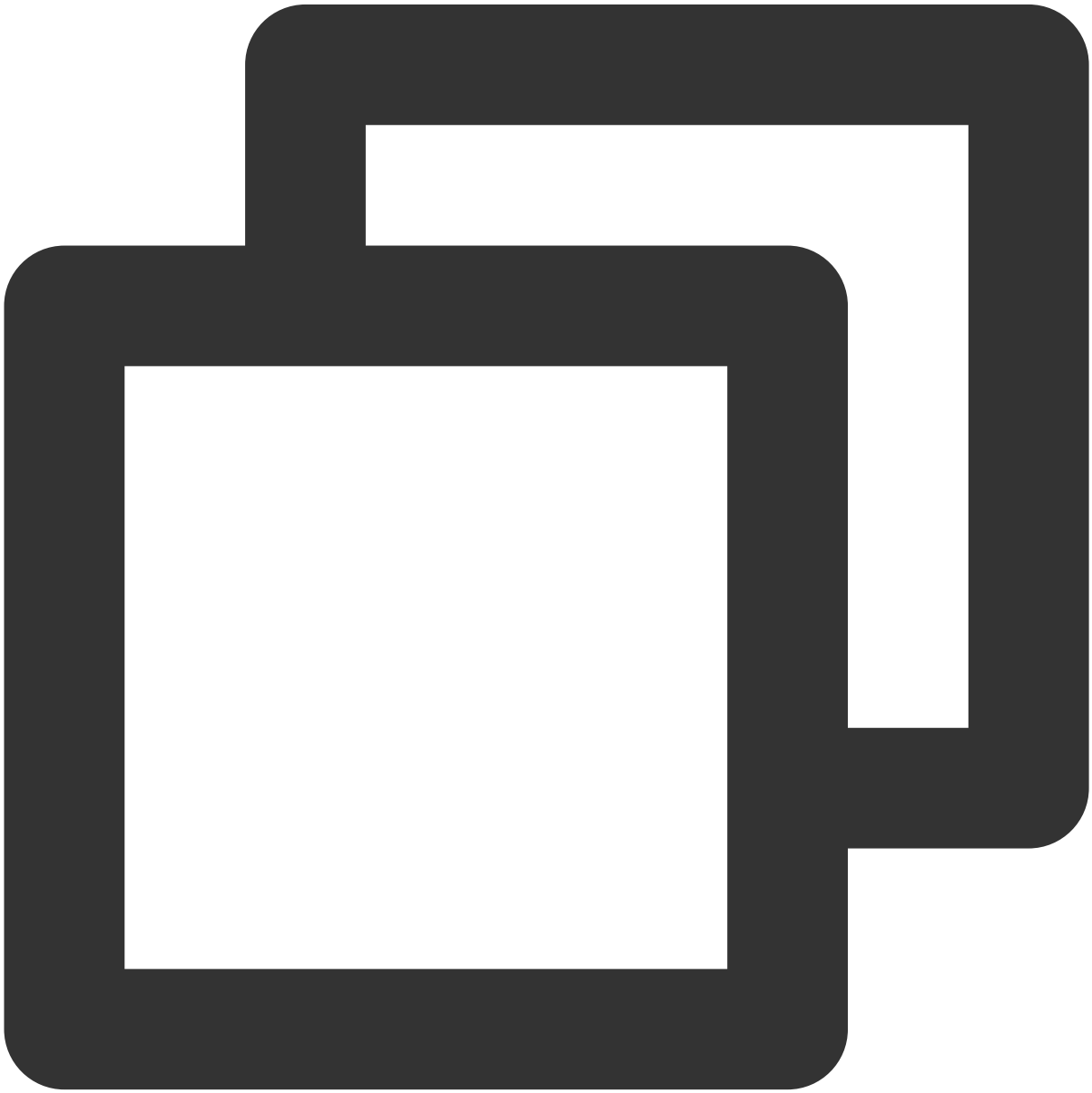
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	H.264 H.265
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
Bitrate	Request.Video	视频输出文件的码率	String	否	无	值范围：[10，50000] 单位：Kbps
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
Rotate	Request.Video	旋转角度	String	否	无	值范围：[0, 360) 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	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.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，支持1、2、4、5、6、8 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

版权所有：腾讯云计算（北京）有限责任公司

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

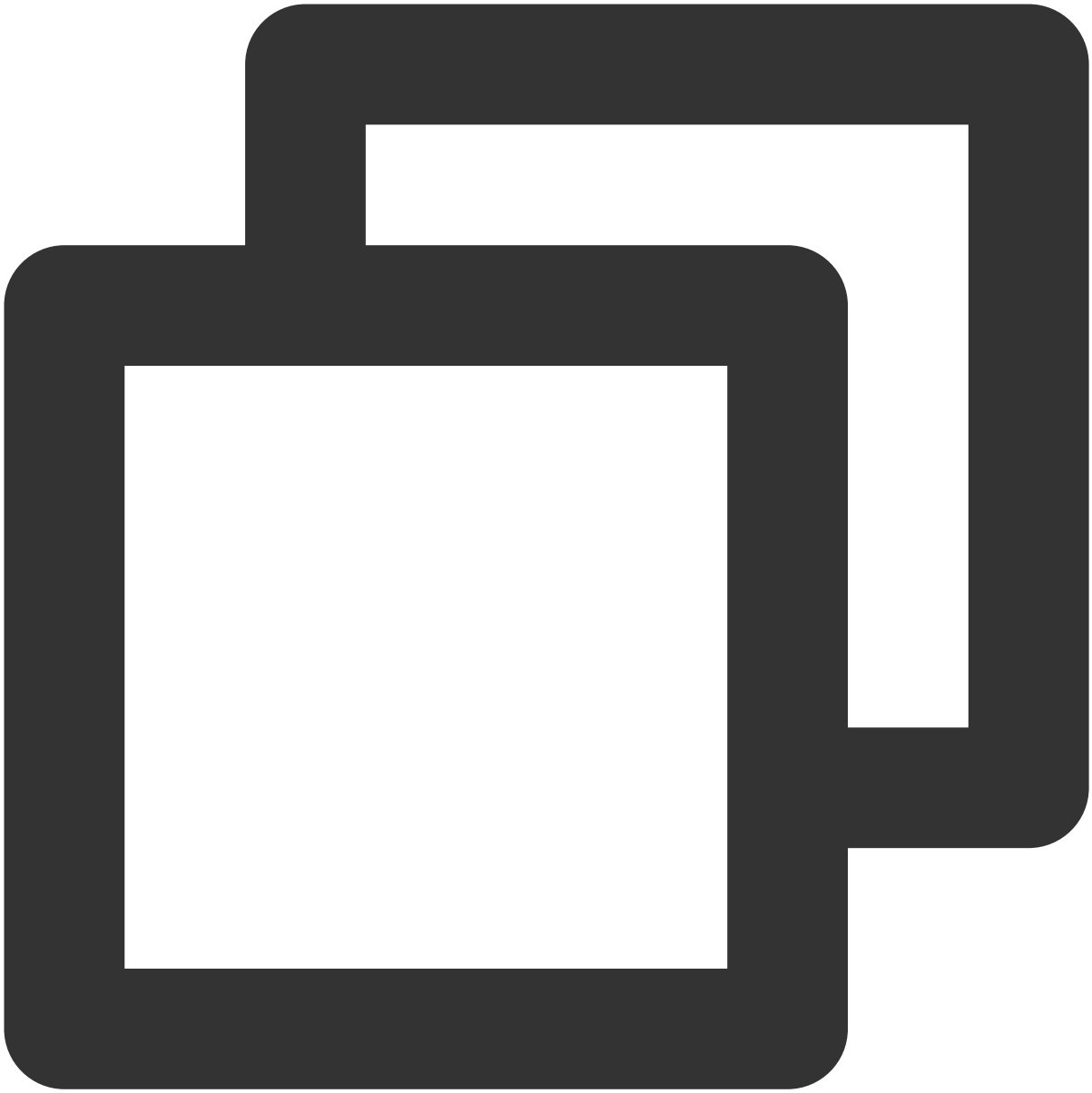
响应

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>
```

```
            </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：

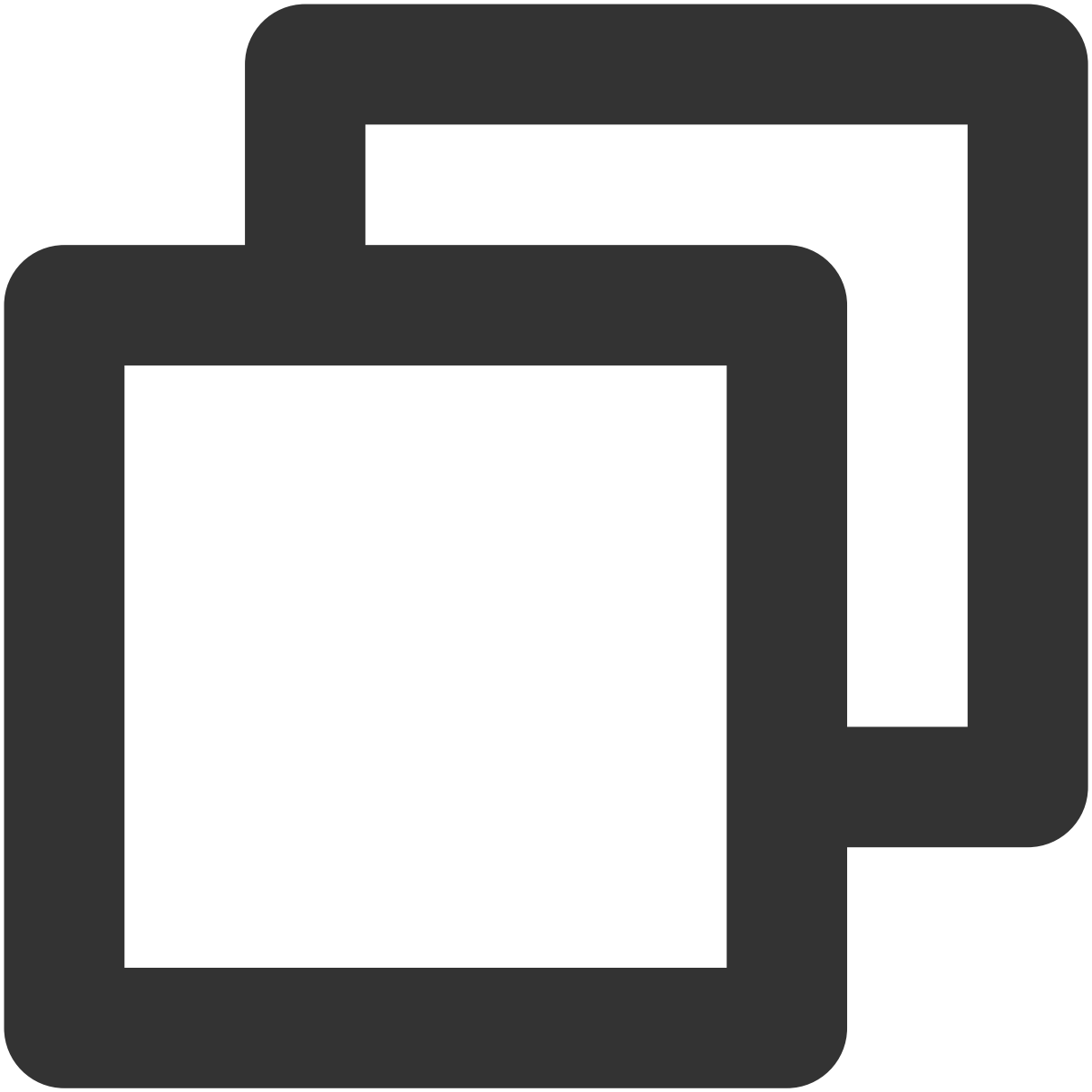
节点名称（关键字）	父节点	描述
Duration	Response.TemplateList.VideoMontage	同请求体中的 Request.Duration
TimeInterval	Response.TemplateList.VideoMontage	同请求体中的 Request.TimeInterval
Container	Response.TemplateList.VideoMontage	同请求体中的 Request.Container
Video	Response.TemplateList.VideoMontage	同请求体中的 Request.Video
Audio	Response.TemplateList.VideoMontage	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

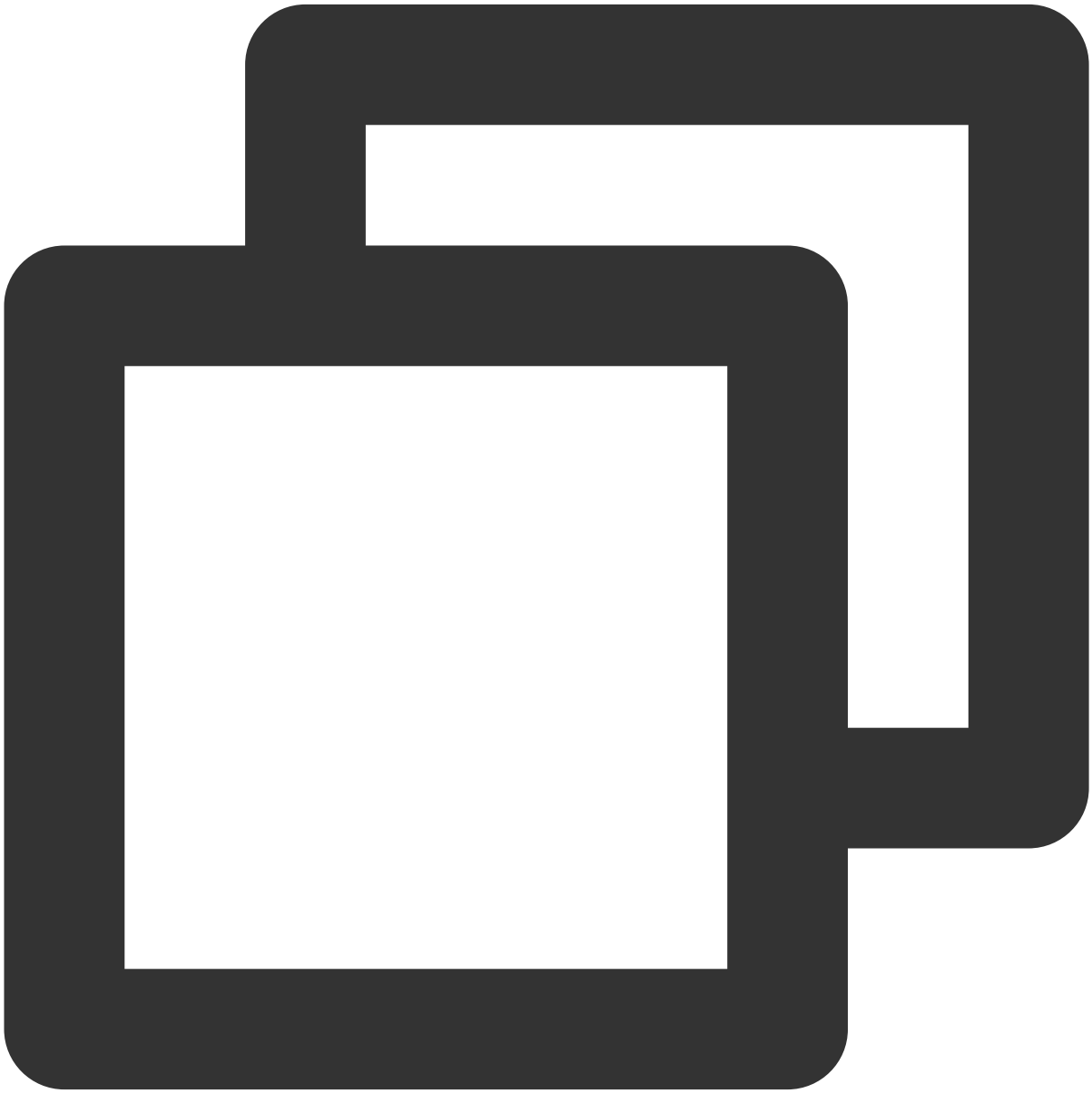


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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>
```

创建视频增强模板

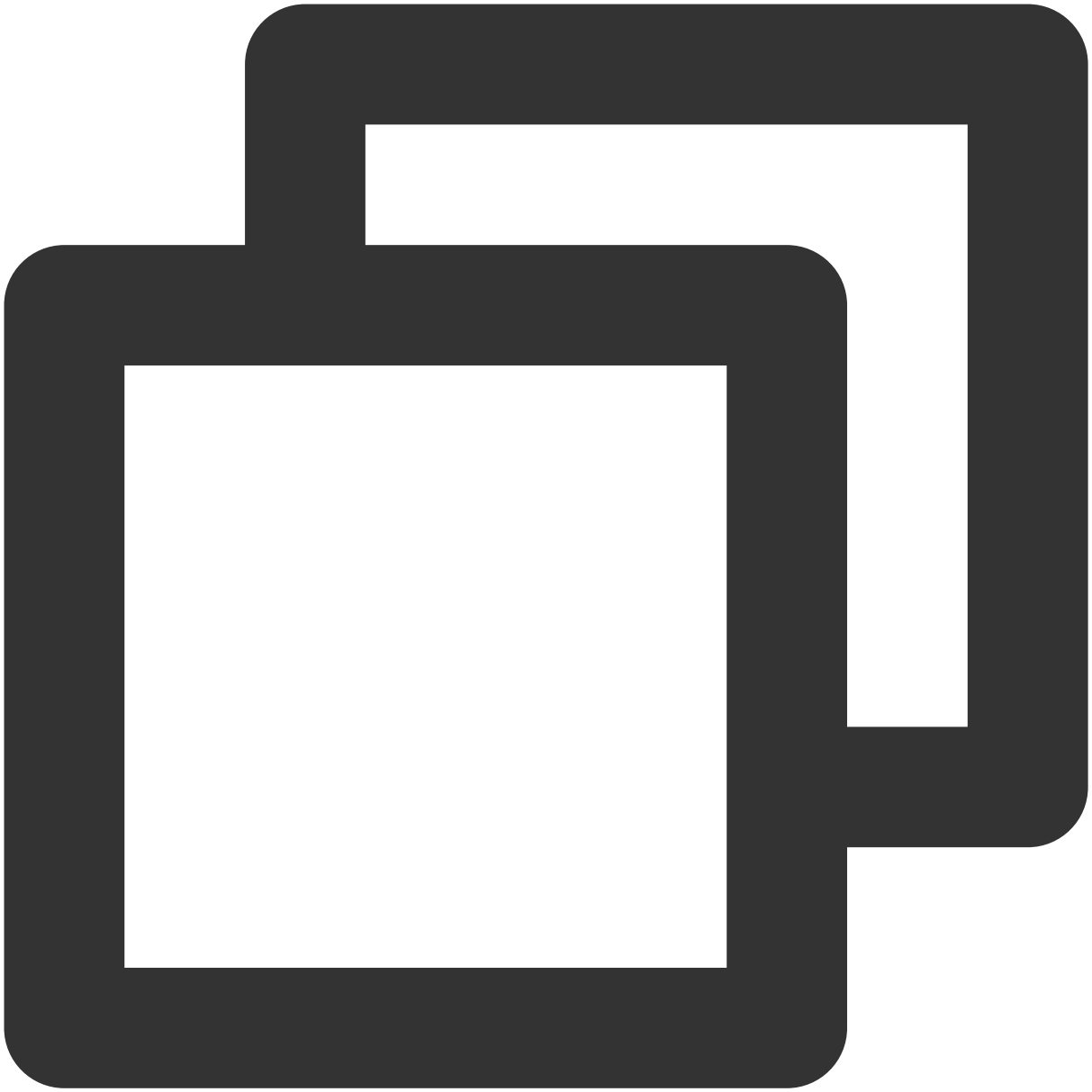
最近更新时间：2024-01-23 10:00:02

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：

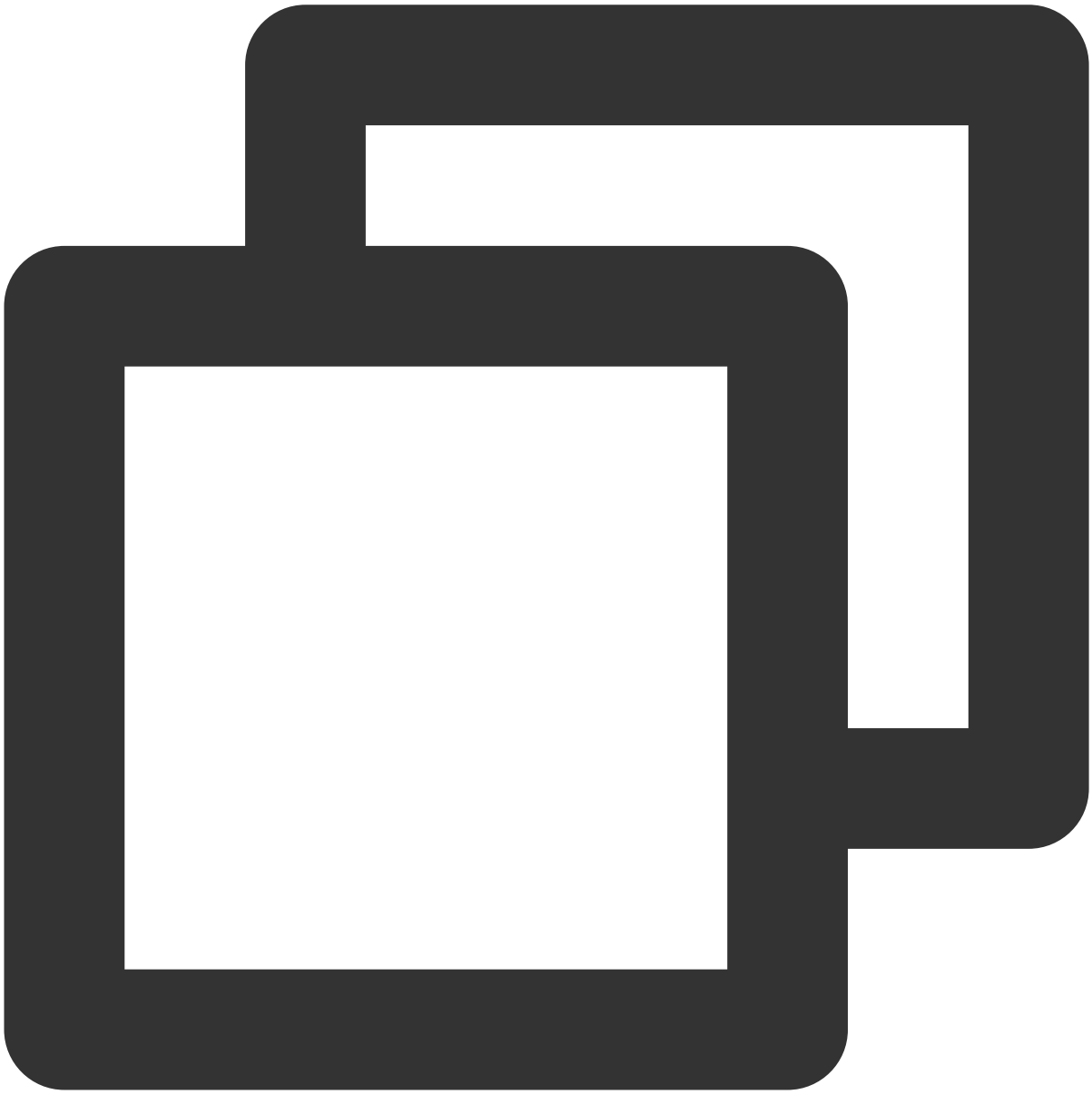
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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）

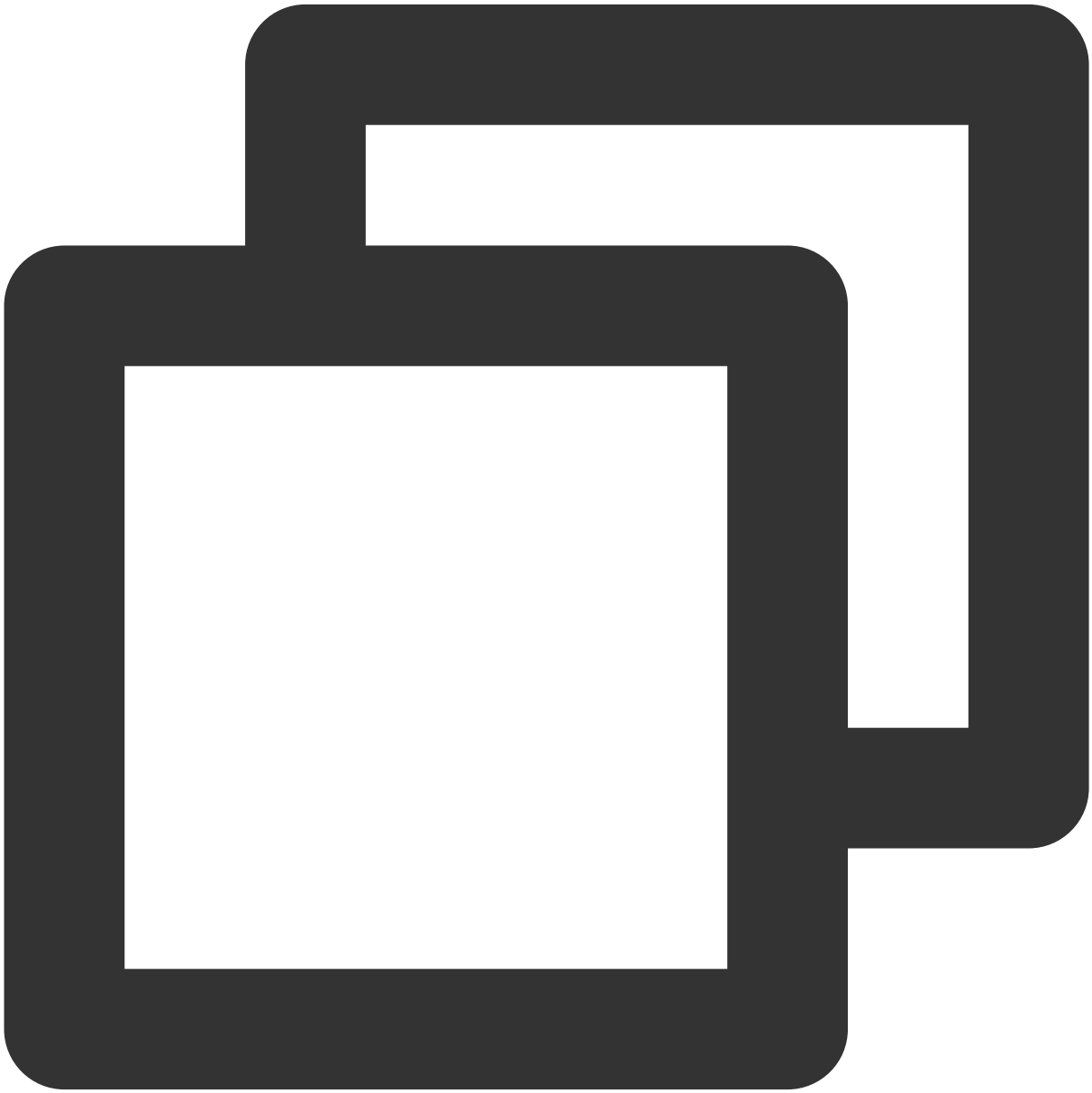
响应

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

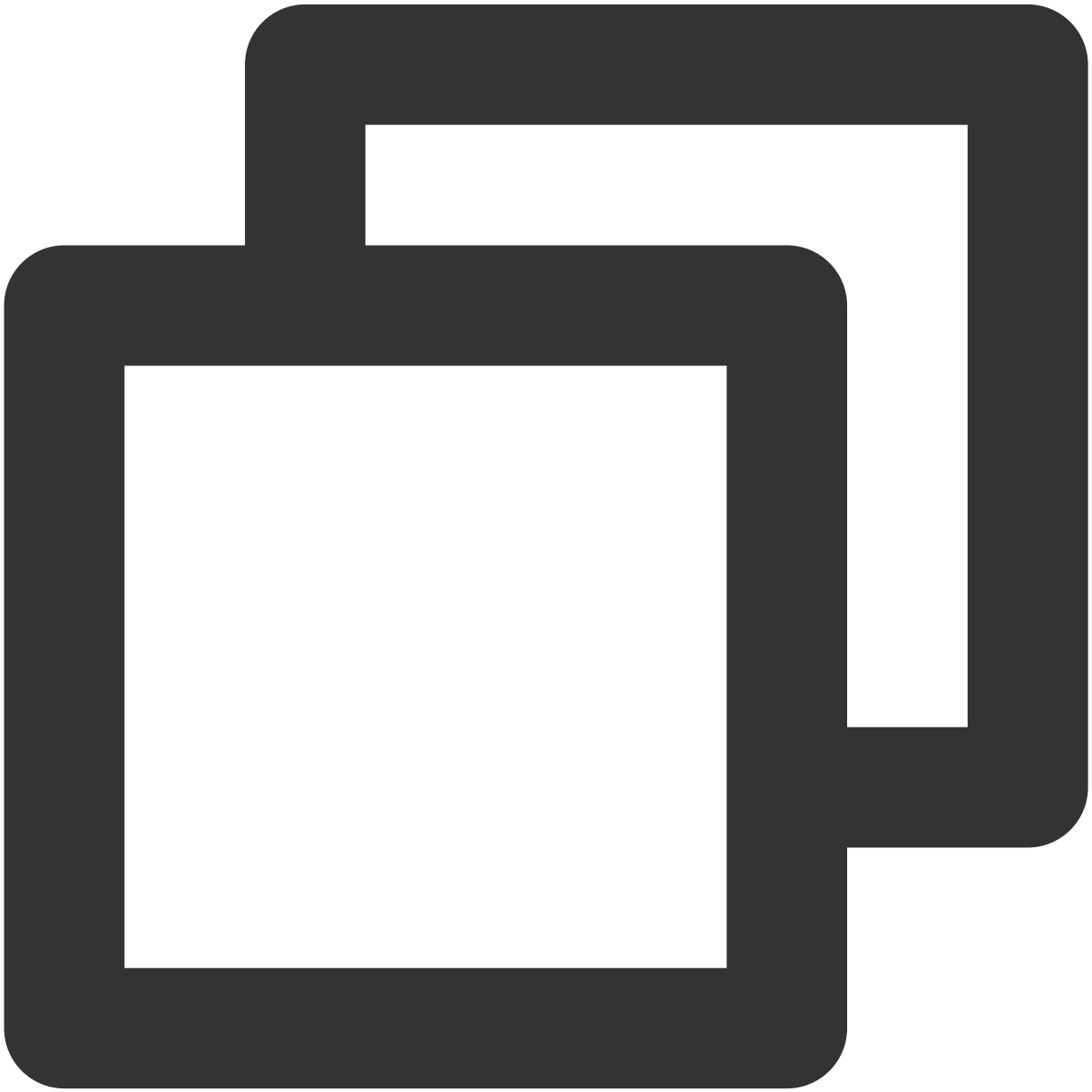
ColorEnhance	Response.Template.VideoProcess	同请求体中的 Request.ColorEnhance
MsSharpen	Response.Template.VideoProcess	同请求体中的 Request.MsSharpen

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

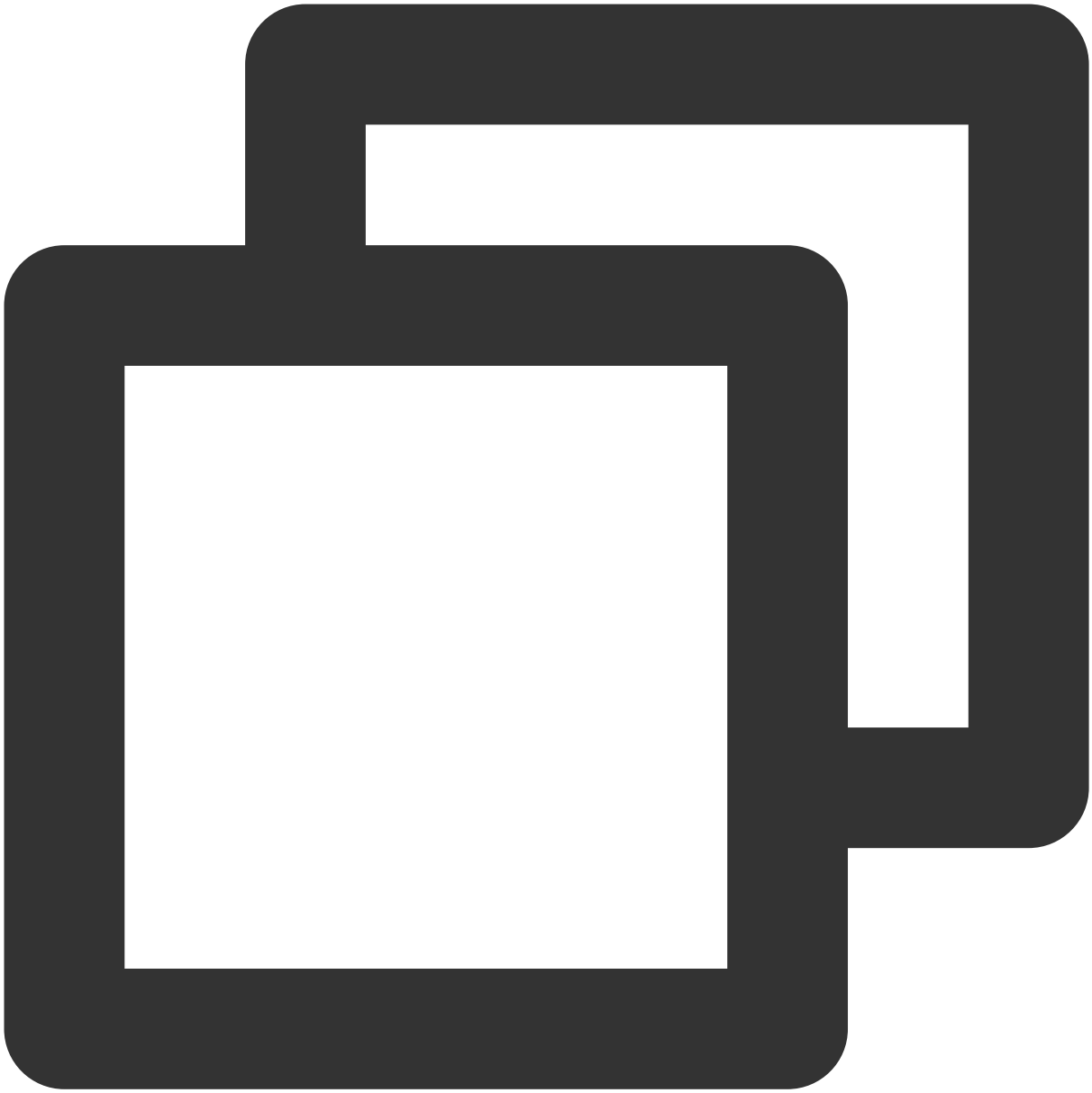


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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>
```

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>
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

创建人声分离模板

最近更新时间：2024-01-23 10:00:02

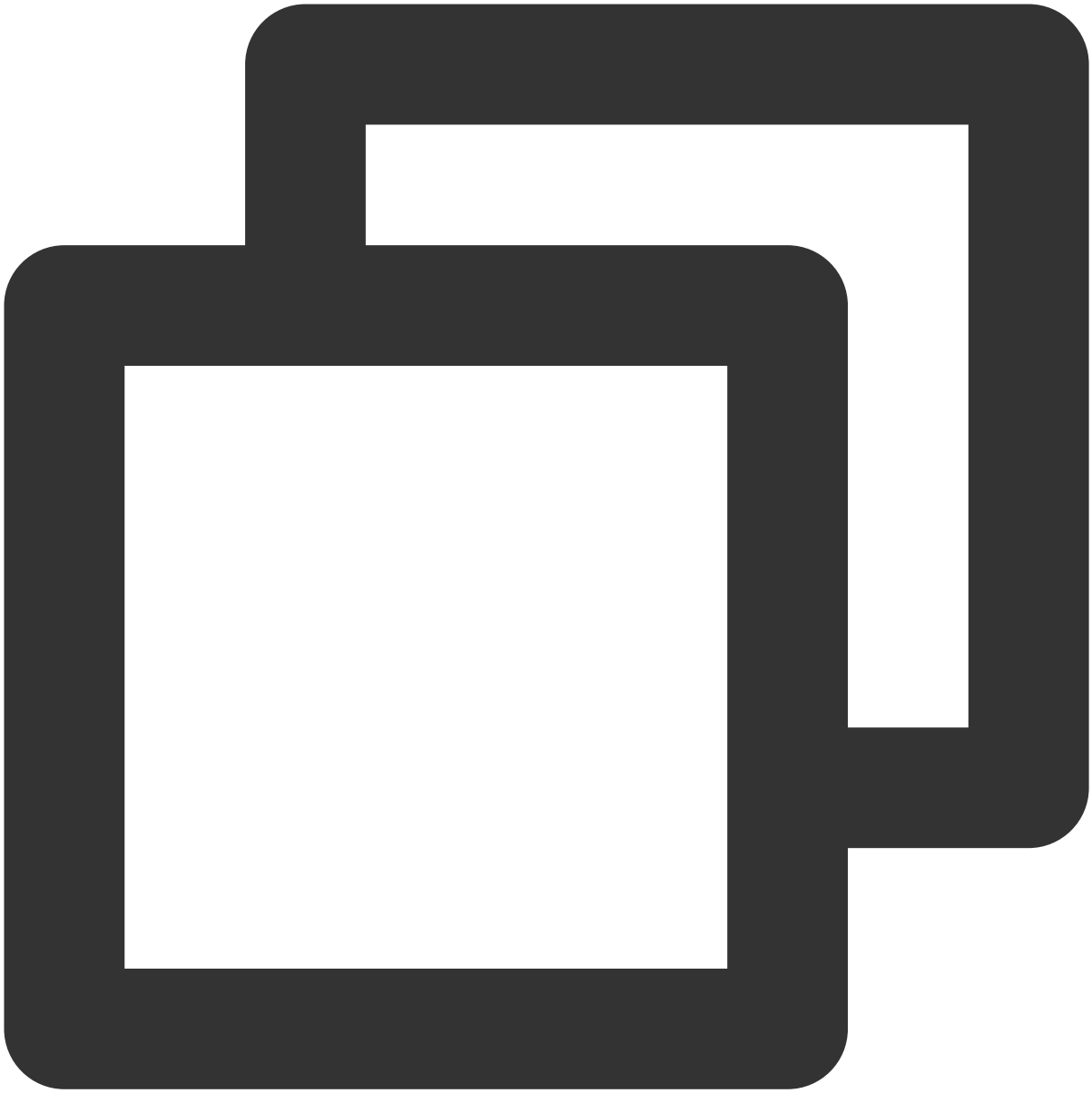
功能描述

创建人声分离模板。

请求

请求示例

版权所有：腾讯云计算（北京）有限责任公司



```
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：

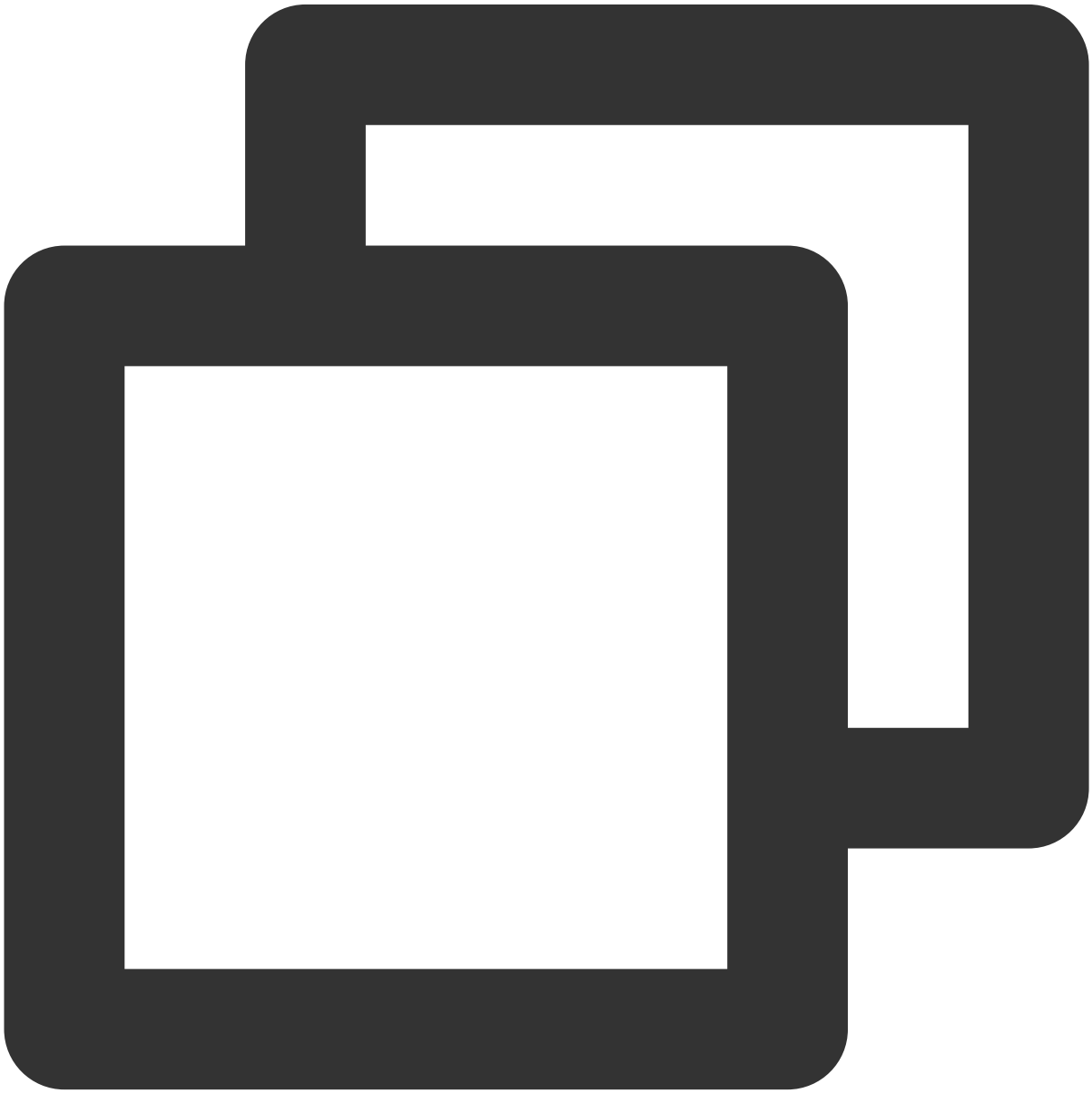
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	是否必选	默认值	限制

版权所有：腾讯云计算（北京）有限责任公司

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	单位：Hz 可选 8000、11025、22050、32000、44100、48000、96000 当 Codec 设置为 aac/flac 时，不支持8000 当 Codec 设置为 mp3 时，不支持8000和96000 当 Codec 设置为 amr 时，只支持8000
Bitrate	Request.Audio	原始音频码率	String	否	无	单位：Kbps 2. 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，支持1、2 当 Codec 设置为 amr，只支持1

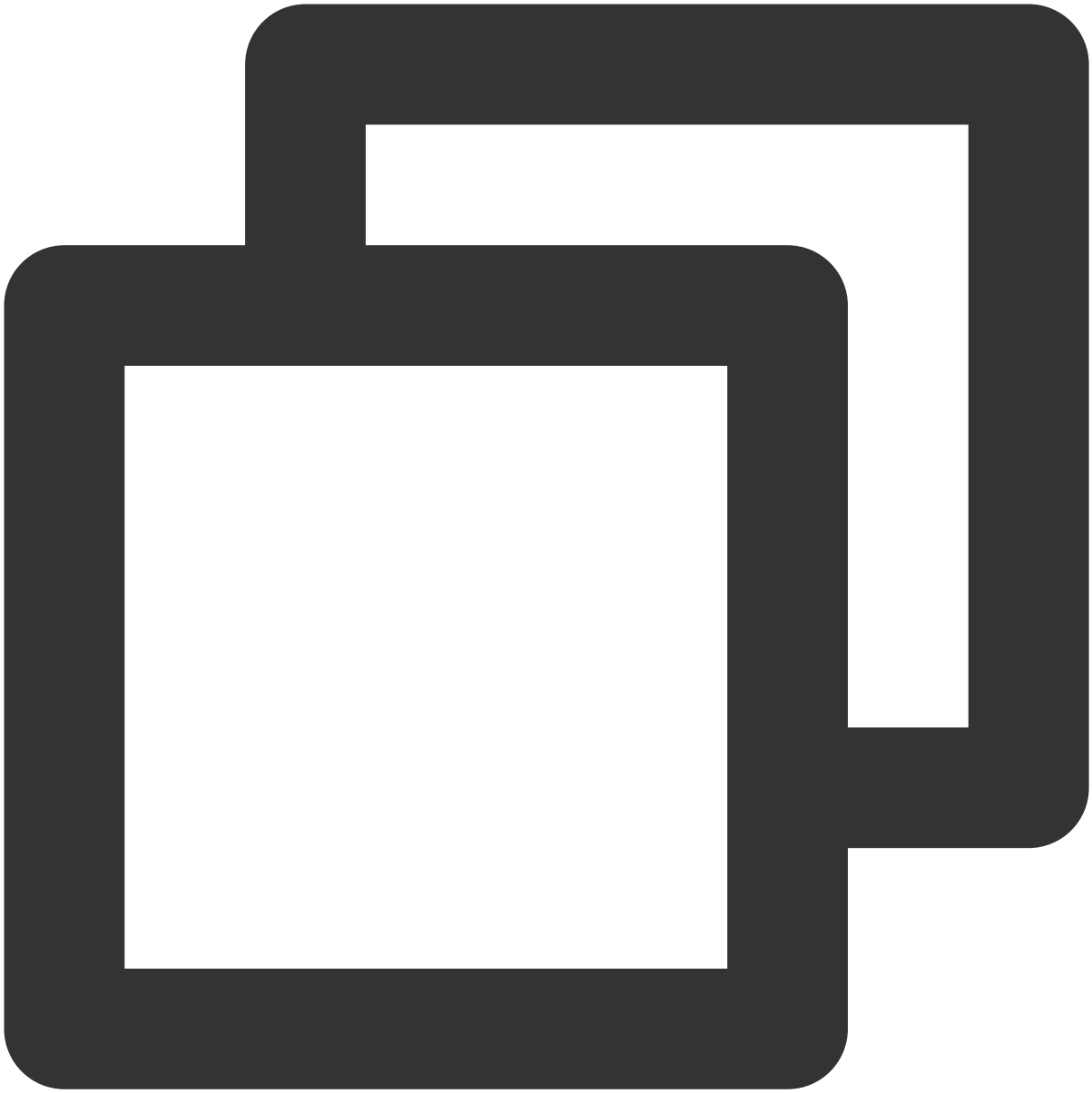
响应

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	Container

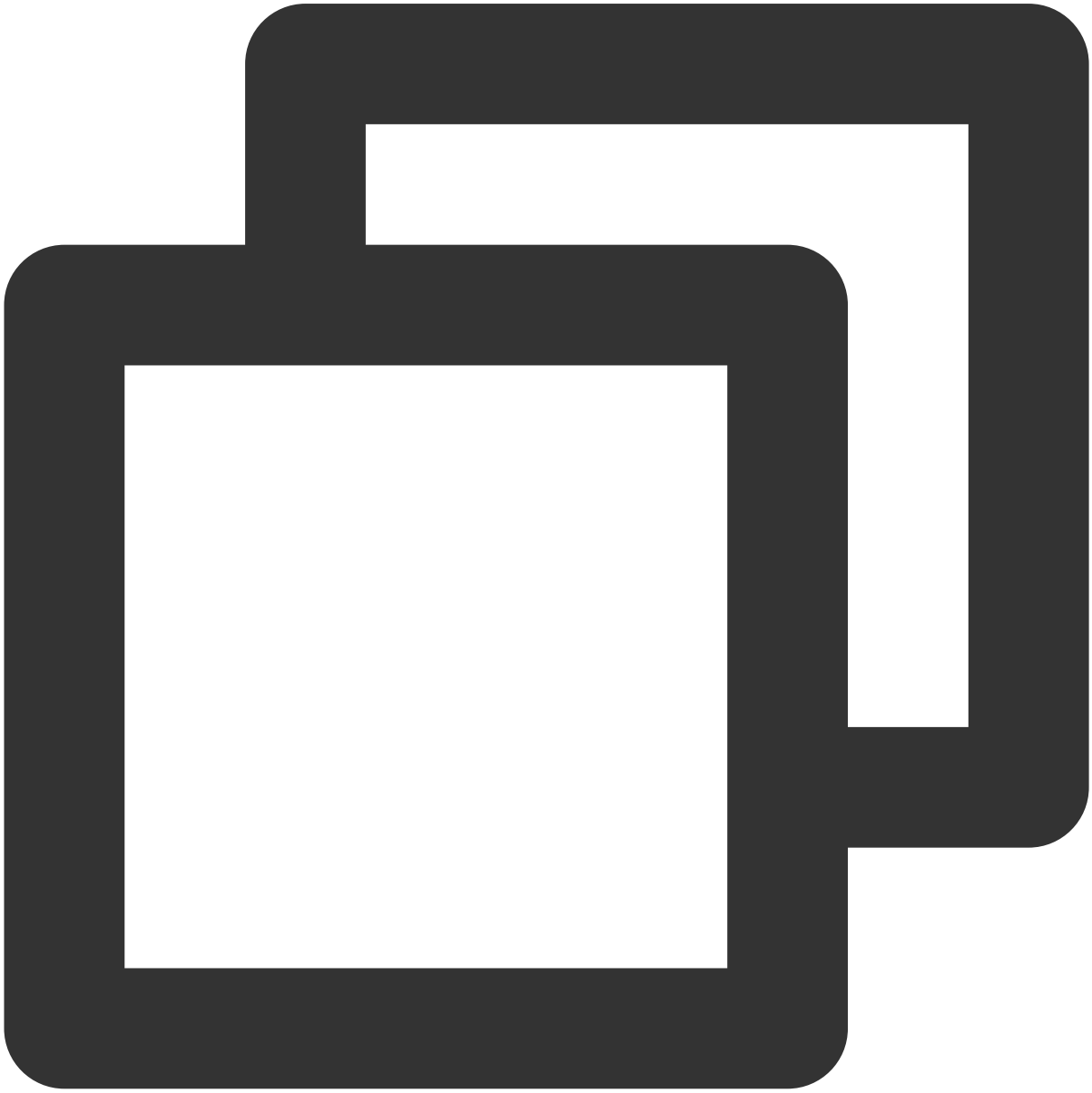
		Request.AudioConfig	

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

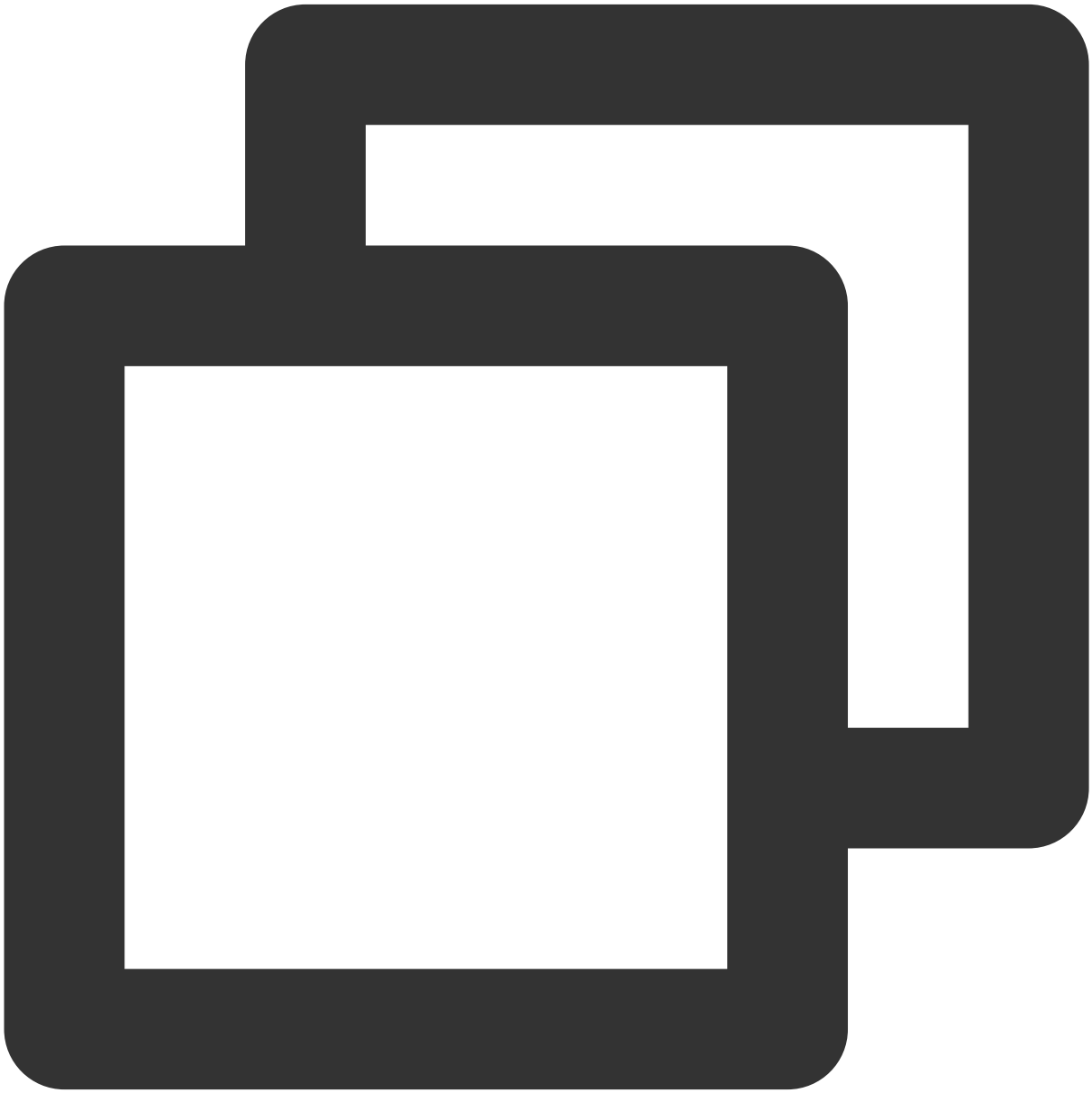
请求



```
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>
```

```
        <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

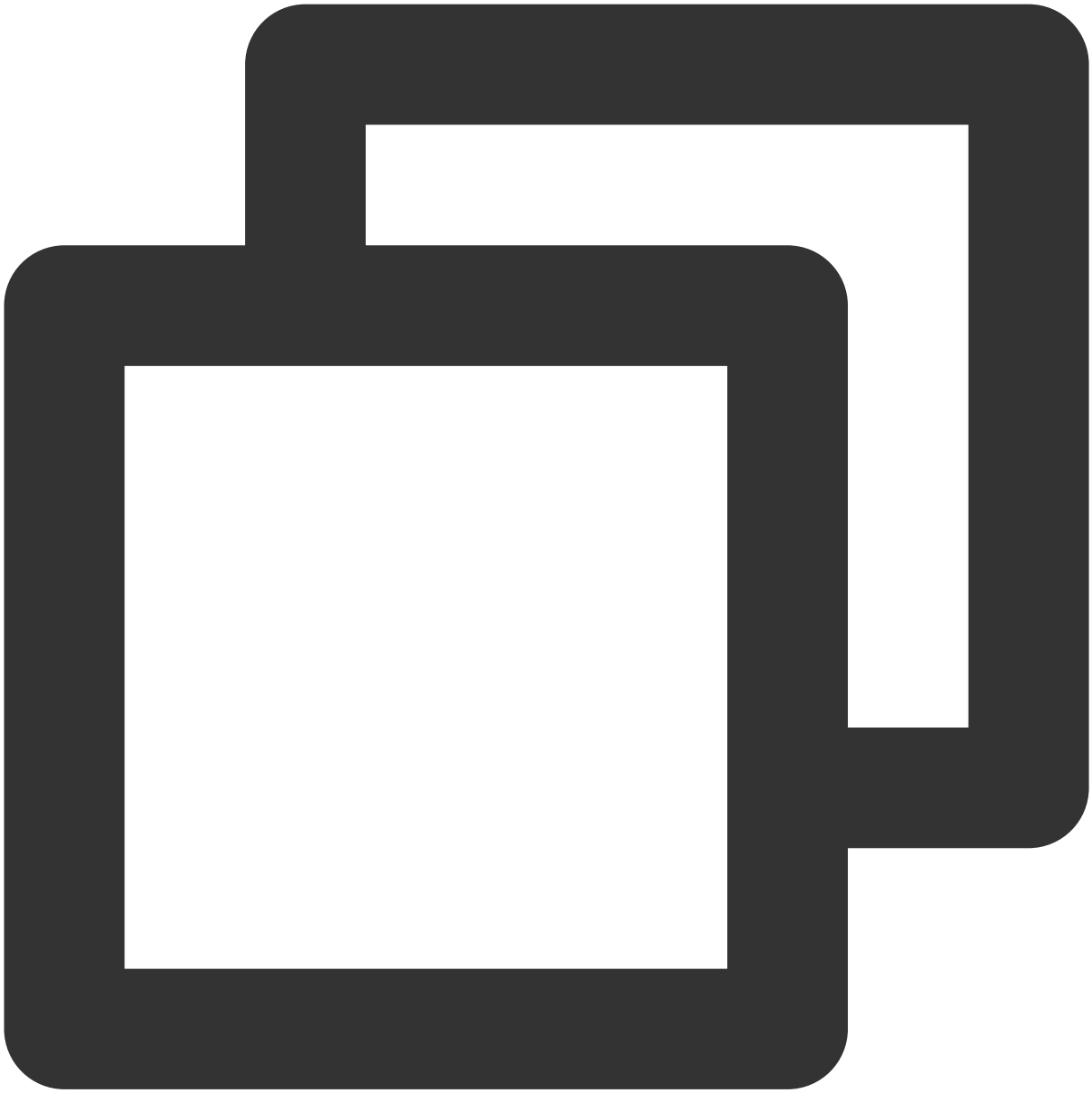
最近更新时间：2024-01-23 10:00:02

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：

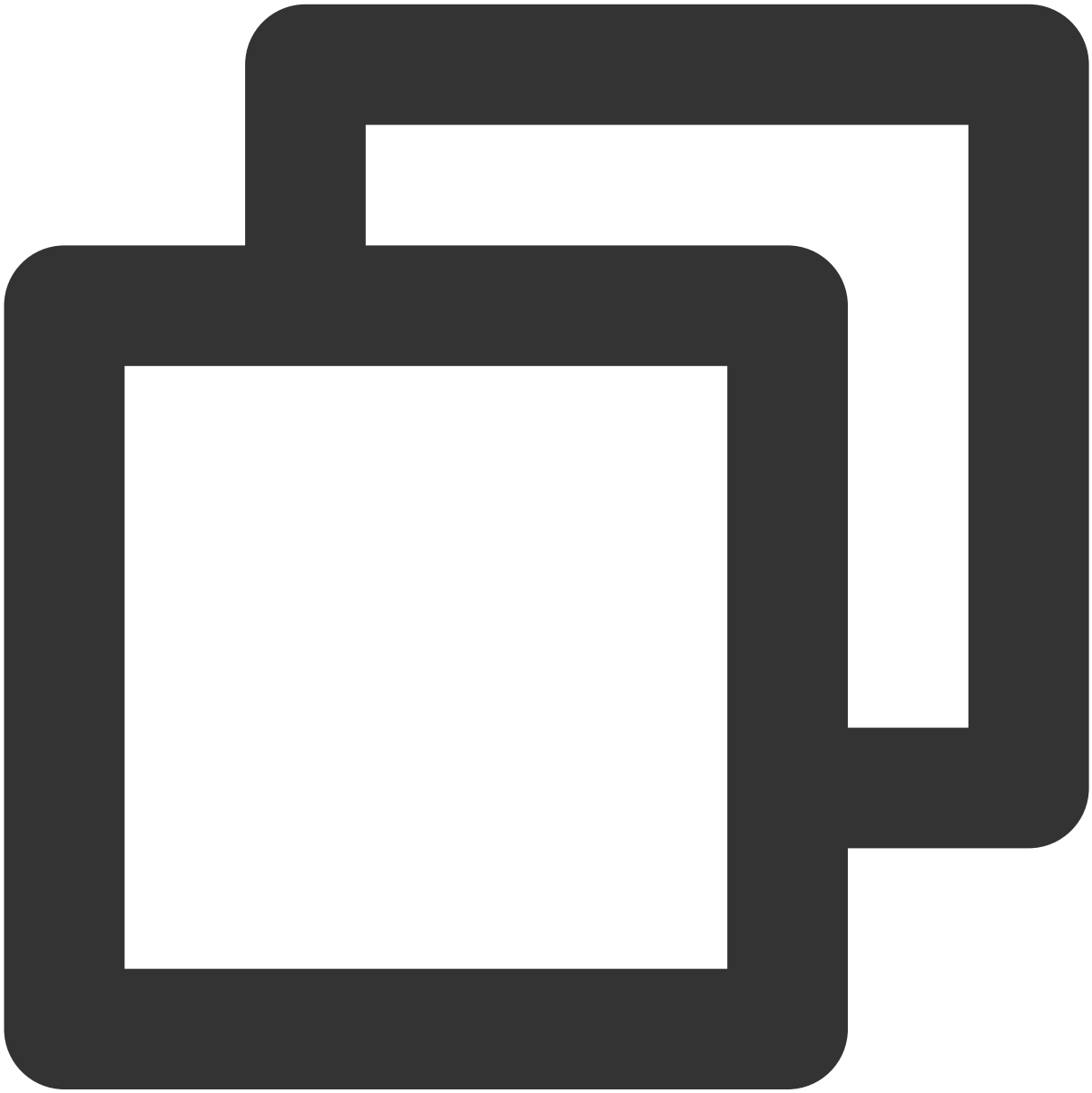
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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	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]。 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]。 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]。 当Pos为Top、Bottom和Center时，该参数无效。
Dy	Request.Watermark	垂直偏移	String	是	无	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]。 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]。 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]。 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	[0 视频时长] 单位为秒 支持 float 格式，执行精度精确到毫秒
EndTime	Request.Watermark	水印结束时	String	否	视频	[0 视频时长]

版权所有：腾讯云计算（北京）有限责任公司

		间			结束时间	单位为秒 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	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

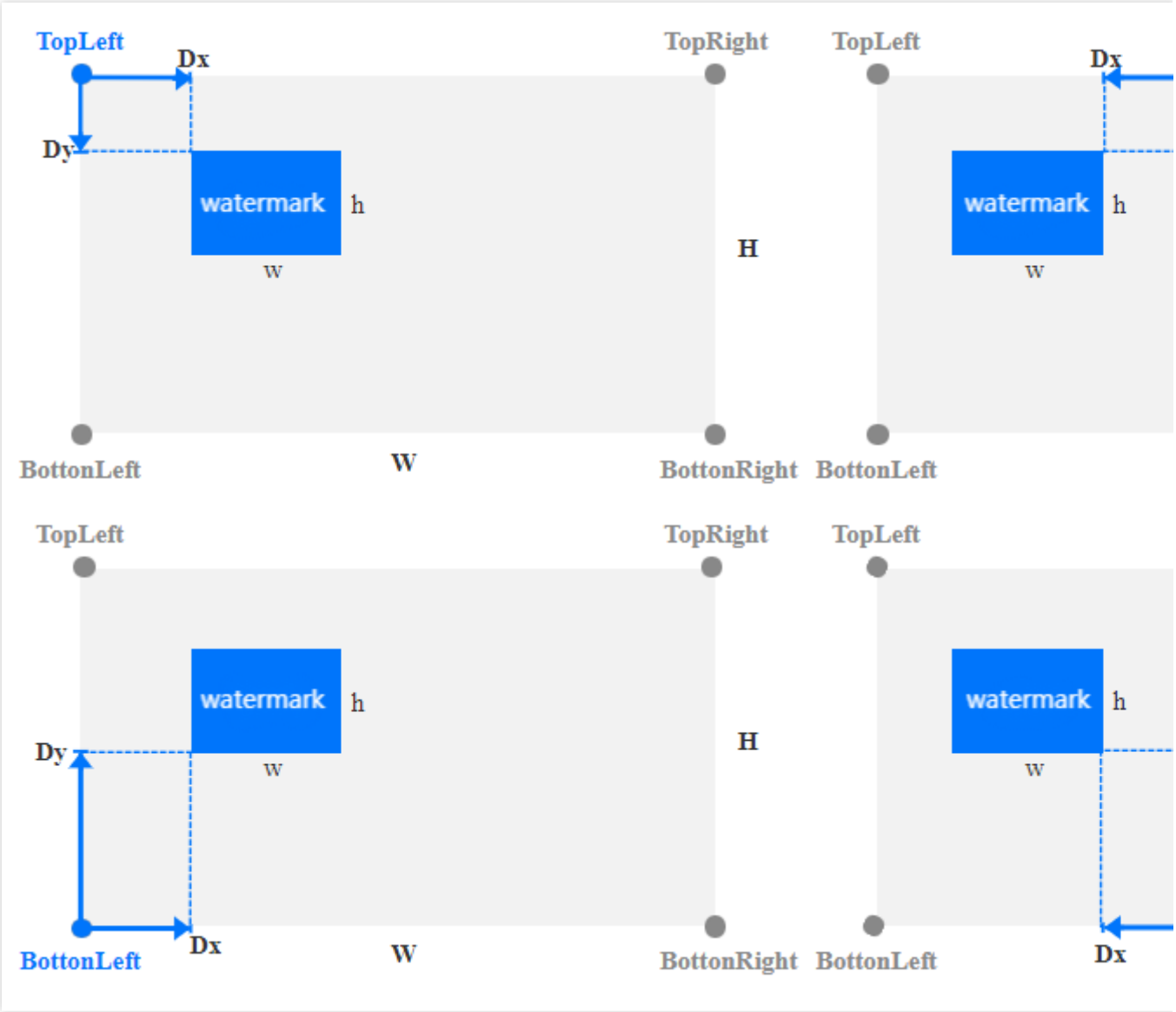
		速度				

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	Original：原有尺寸 Proportion：按比例 Fixed：固定大小
Width	Request.Watermark.Image	宽	String	否	无	当 Mode 为 Original 时，不支持设置水印图宽度 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 当 Mode 为 Fixed，单位为 px，值范围：[8, 4096] 若只设置 Width 时，按照水印图比例计算 Height
Height	Request.Watermark.Image	高	String	否	无	当 Mode 为 Original 时，不支持设置水印图高度 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 当 Mode 为 Fixed，单位为 px，值范围：[8, 4096] 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

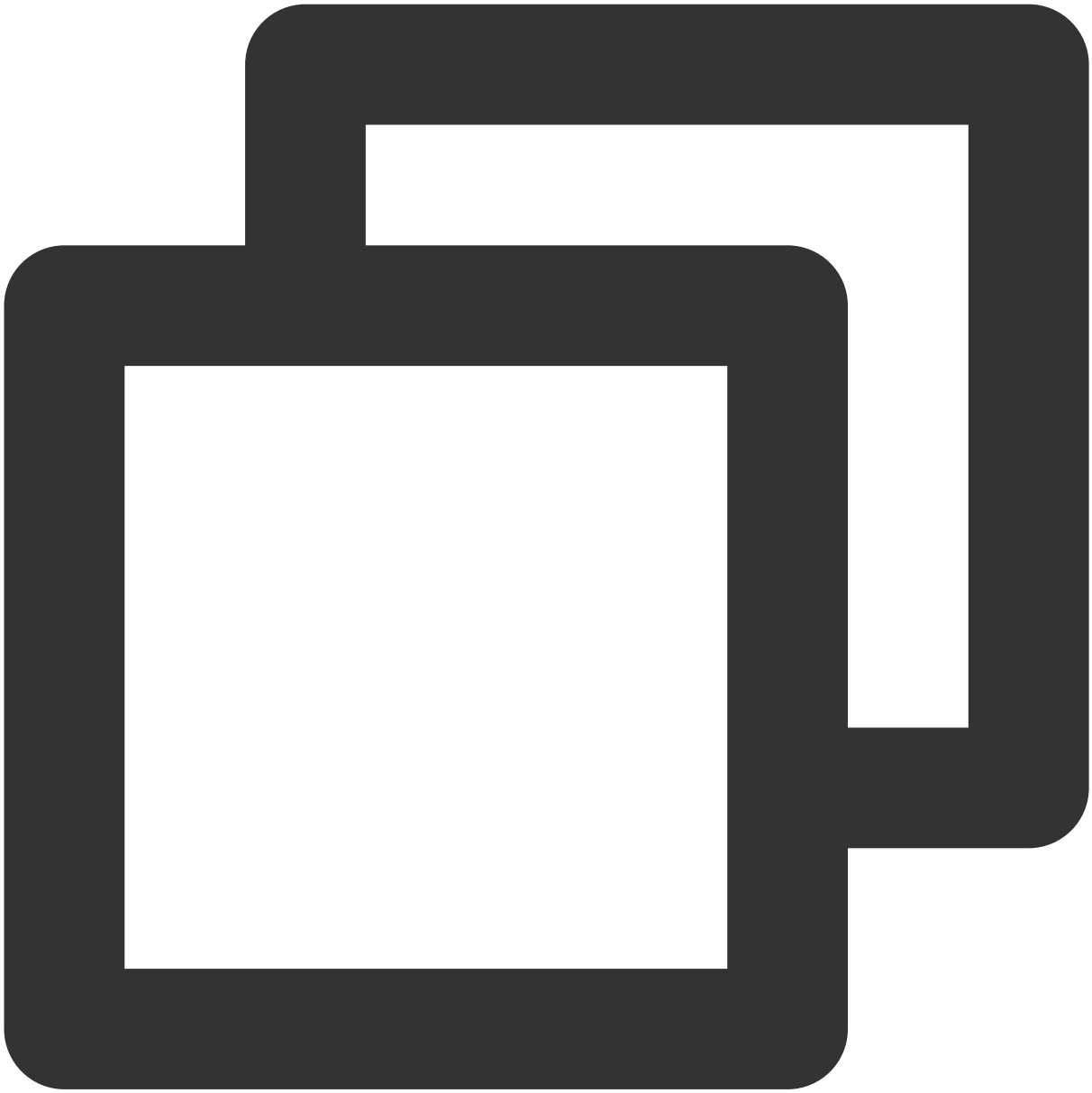
响应

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：

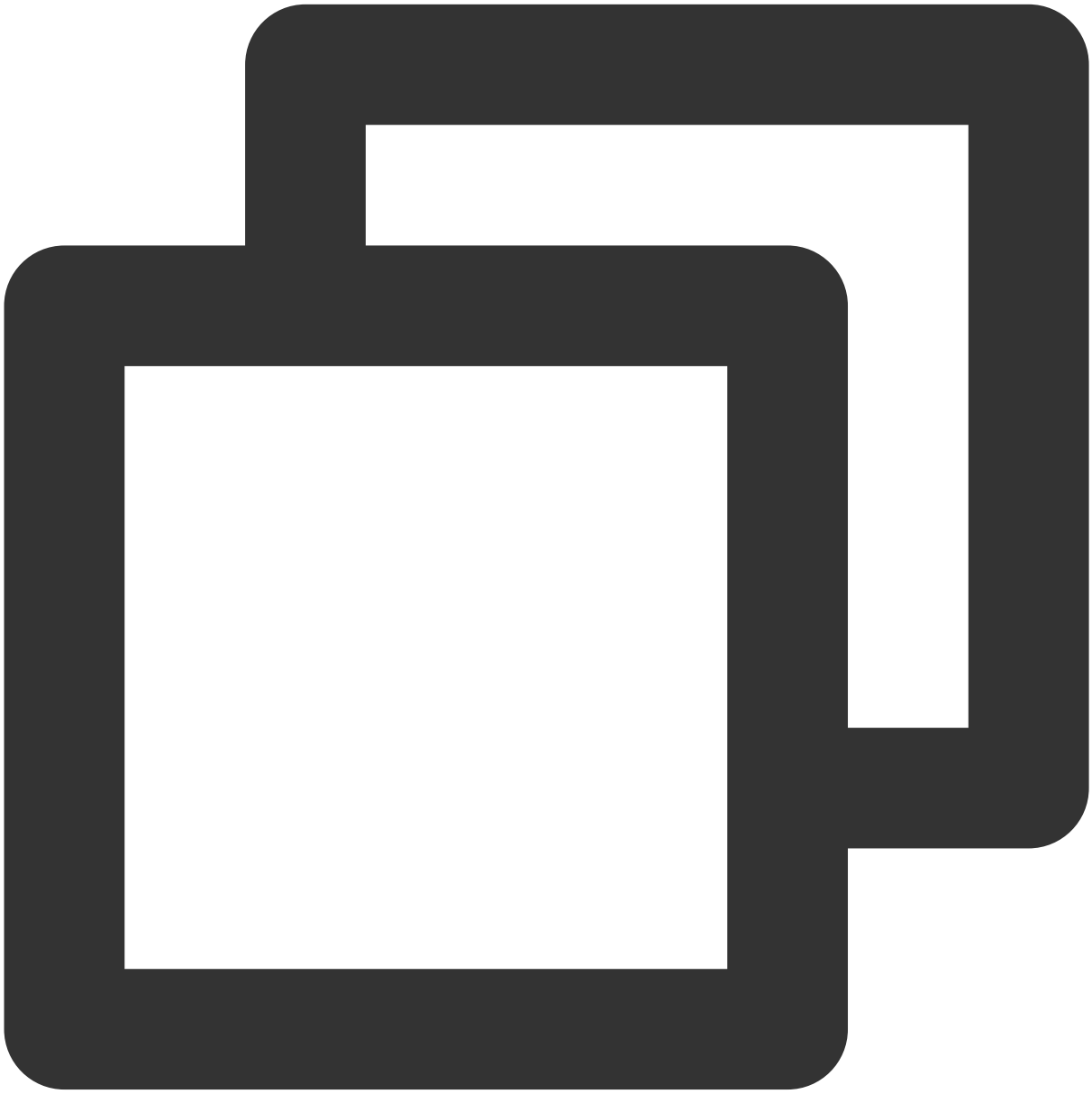
节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，VideoProcess	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	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）

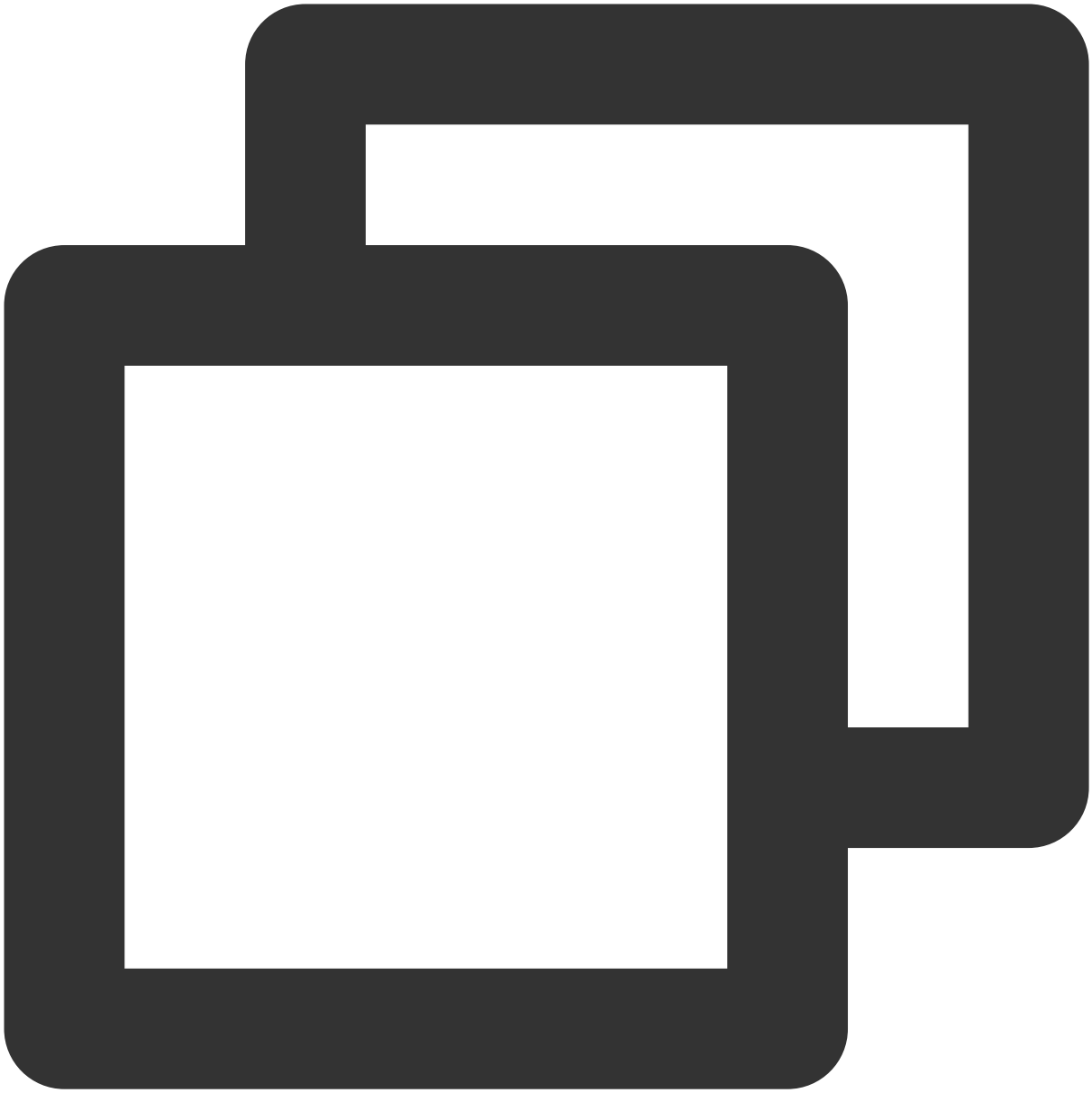


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0****&q-sign-tim
Host: test-1234567890.ci.ap-beijing.myqcloud.com
```

```
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

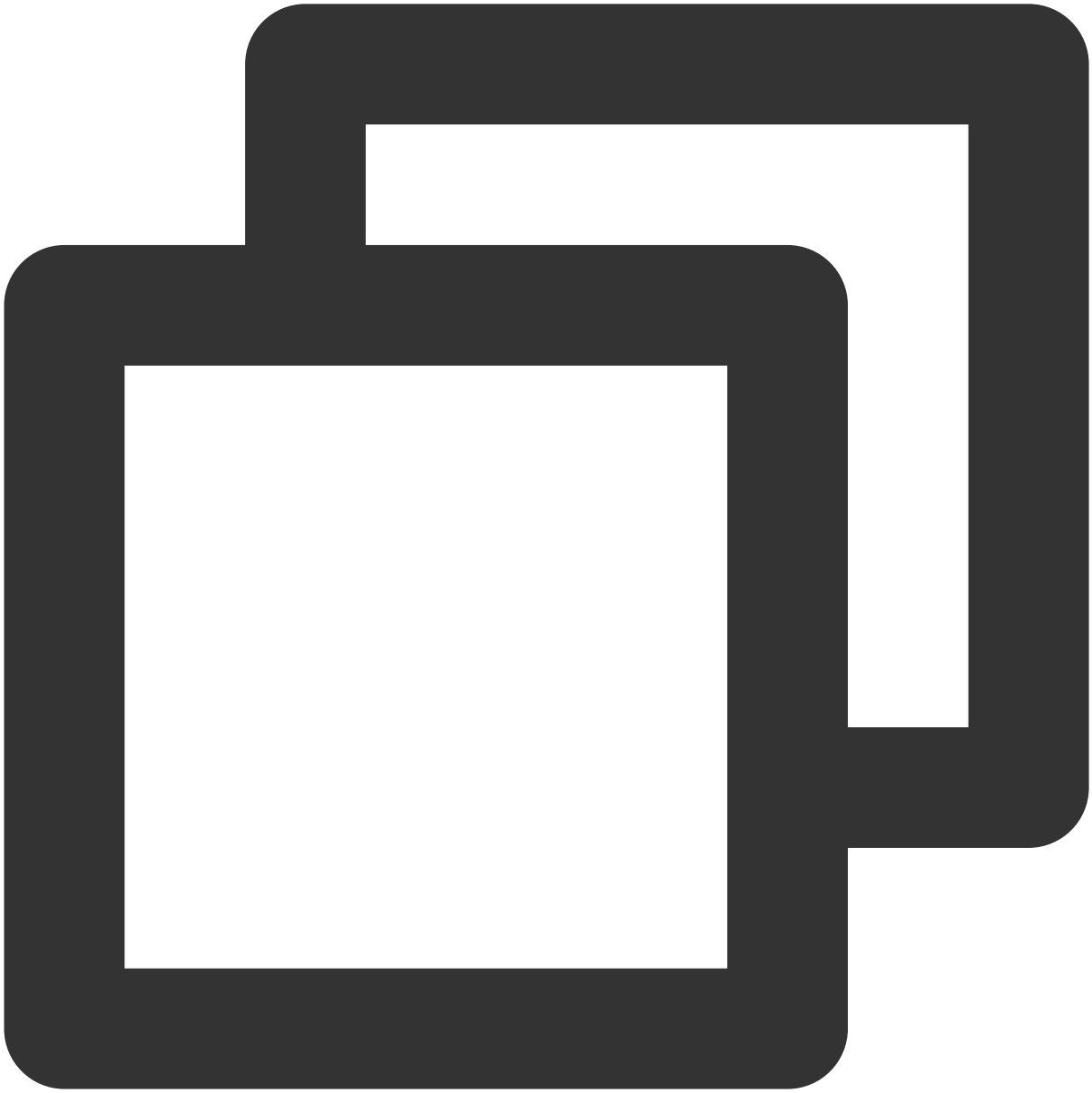


```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<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）

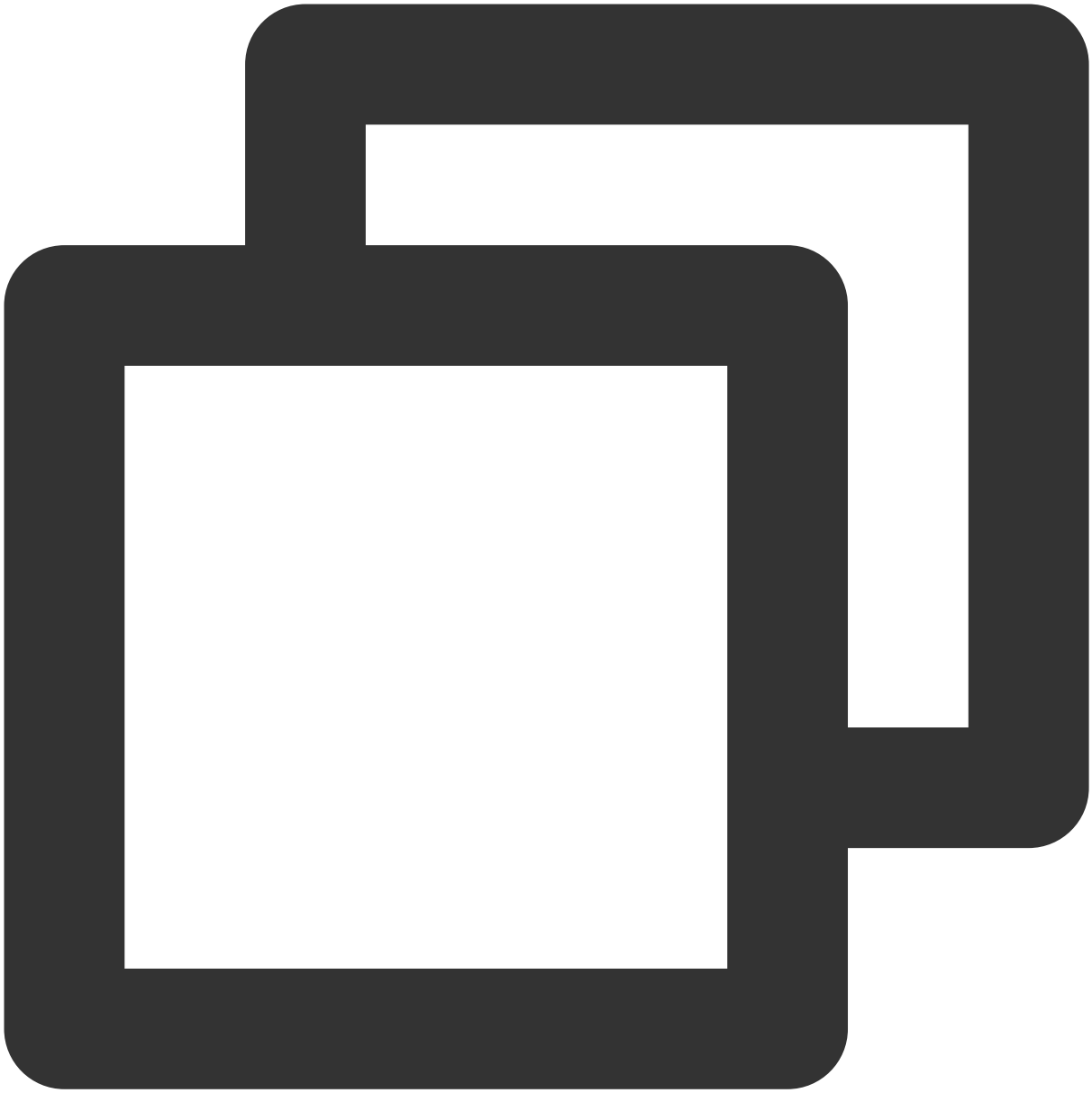


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0****&q-sign-tim
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

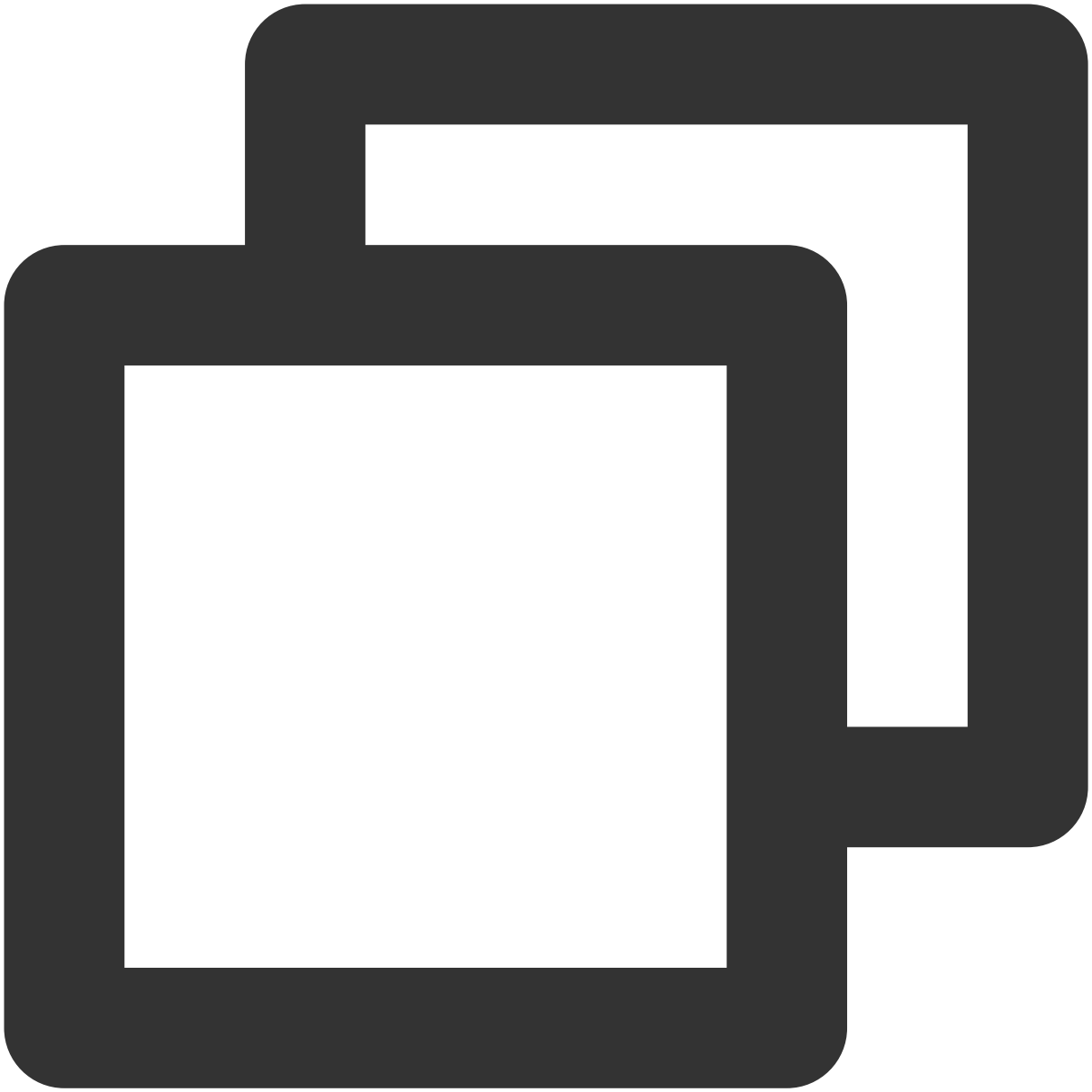
最近更新时间：2024-01-23 10:00:02

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：

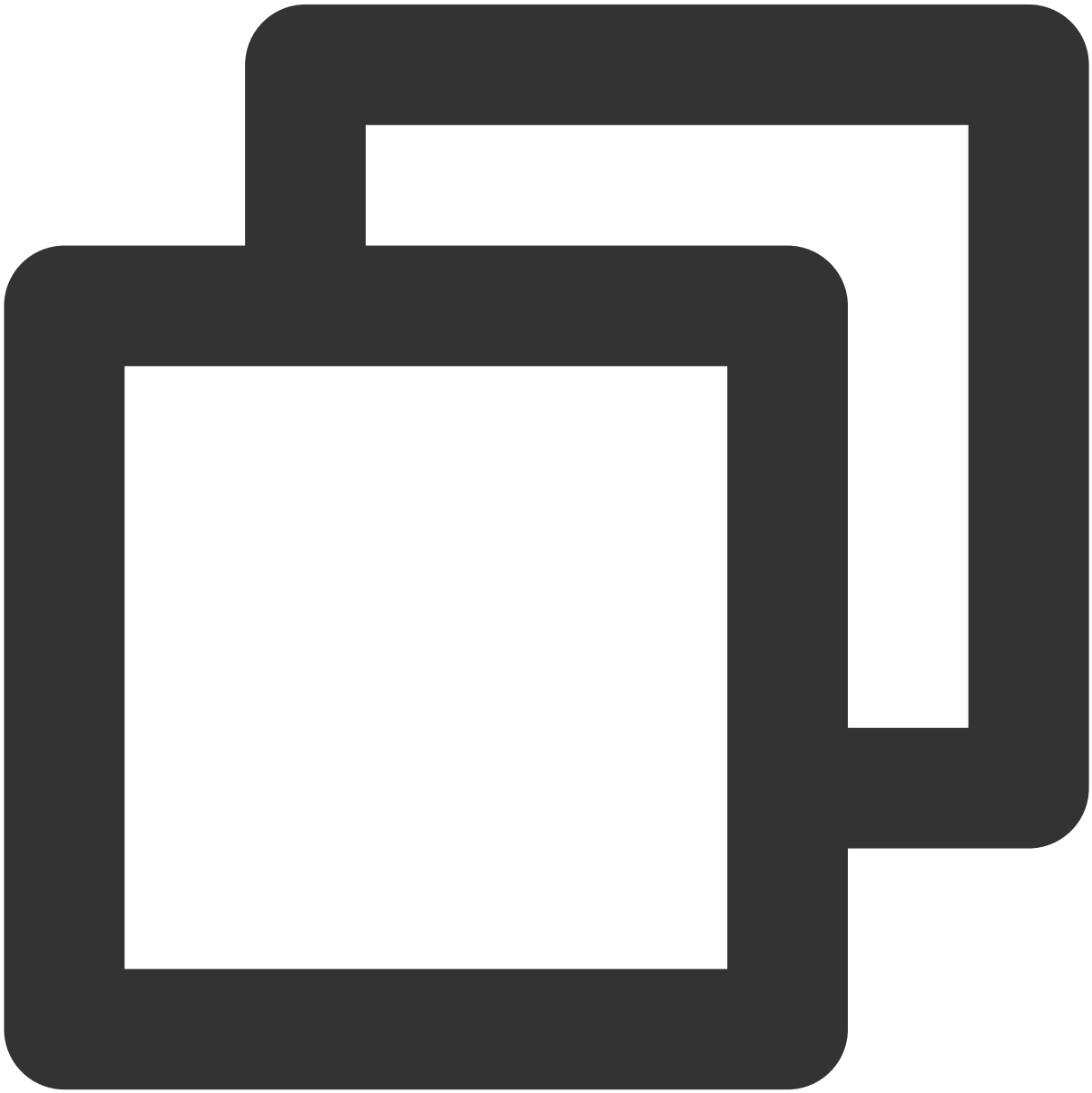
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

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	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 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

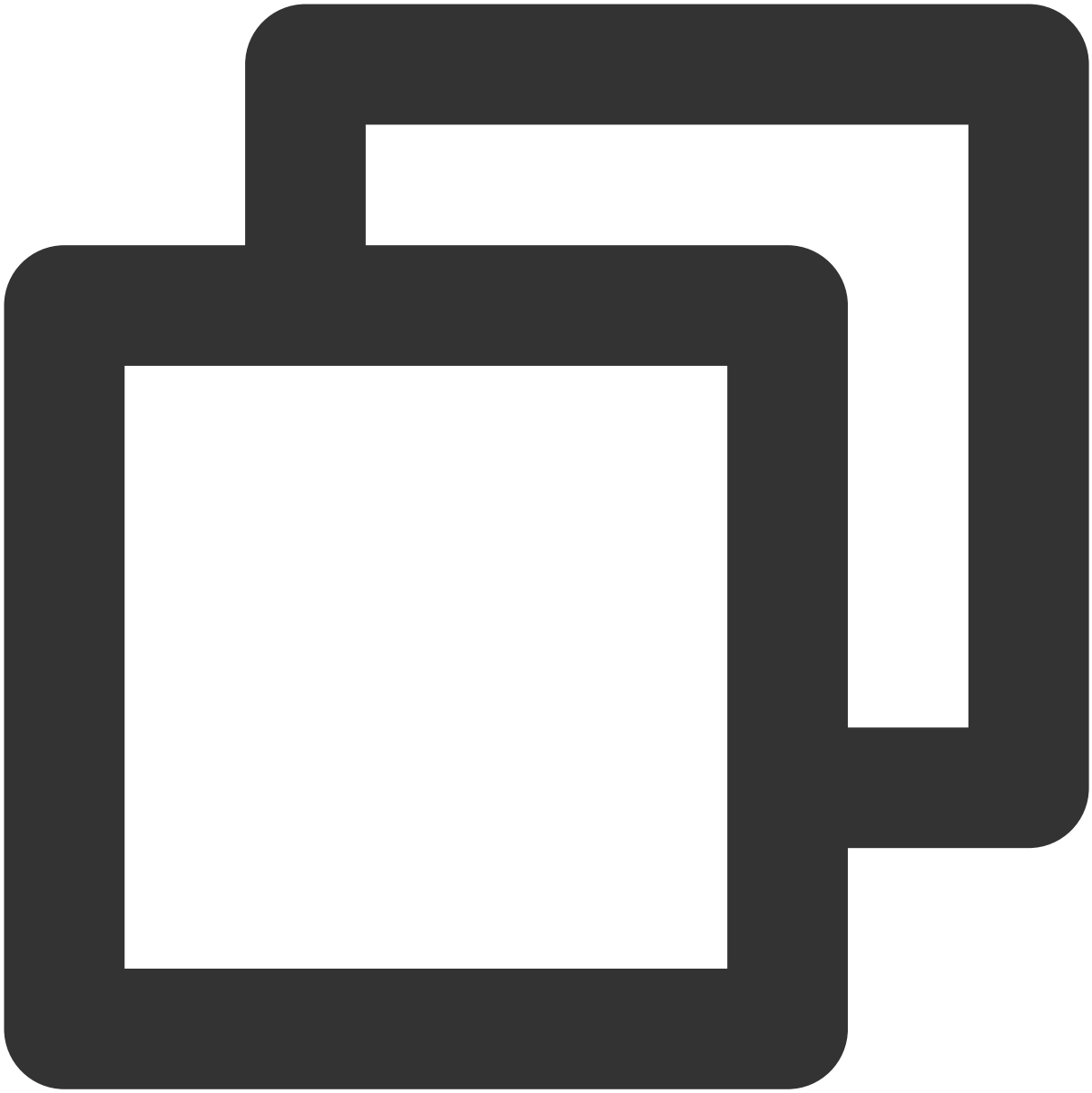
响应

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：

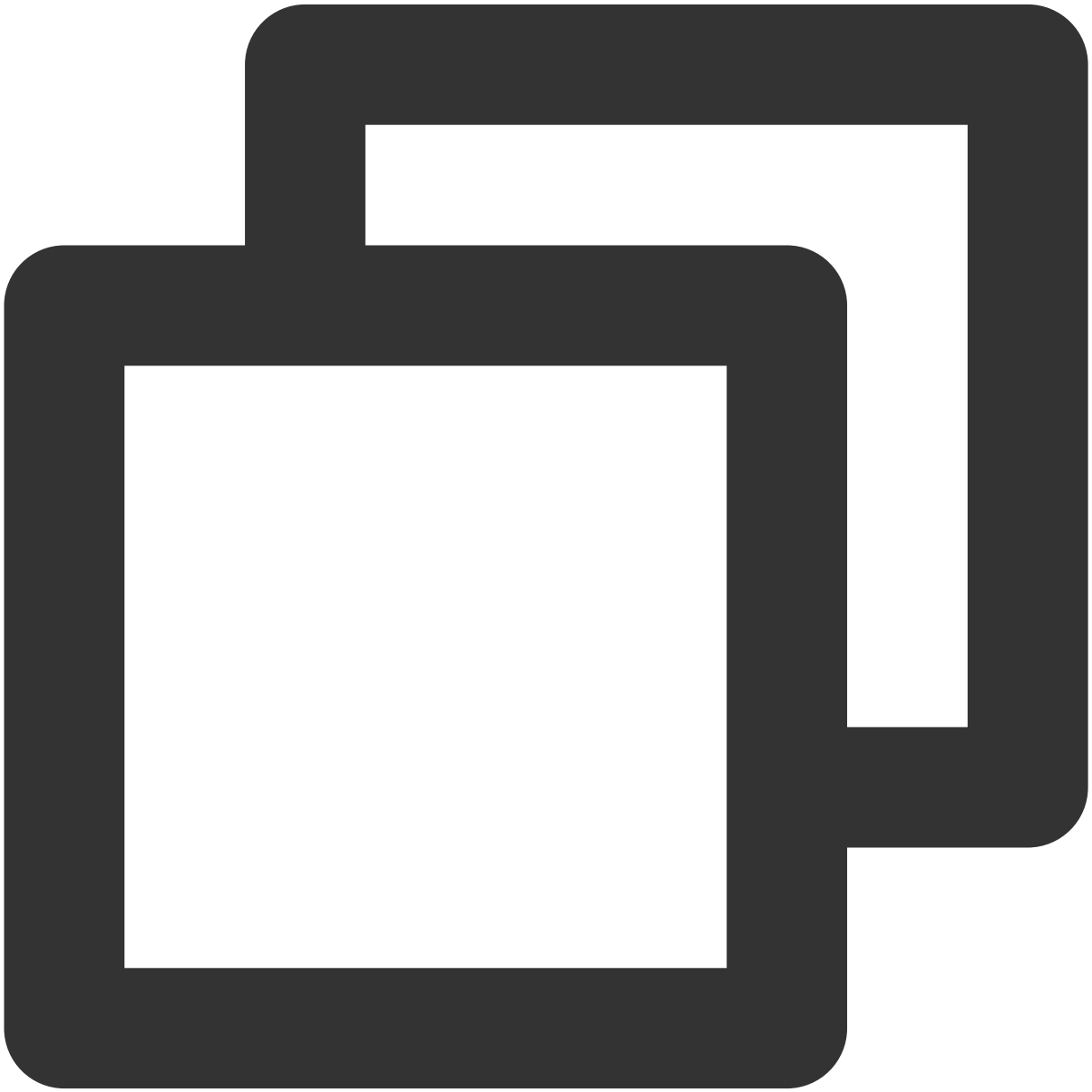
节点名称（关键字）	父节点	描述	类型
Tag	Response.Template	模板类型，SmartCover	String
Name	Response.Template	模板名字	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
TemplateId	Response.Template	模板 ID	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	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

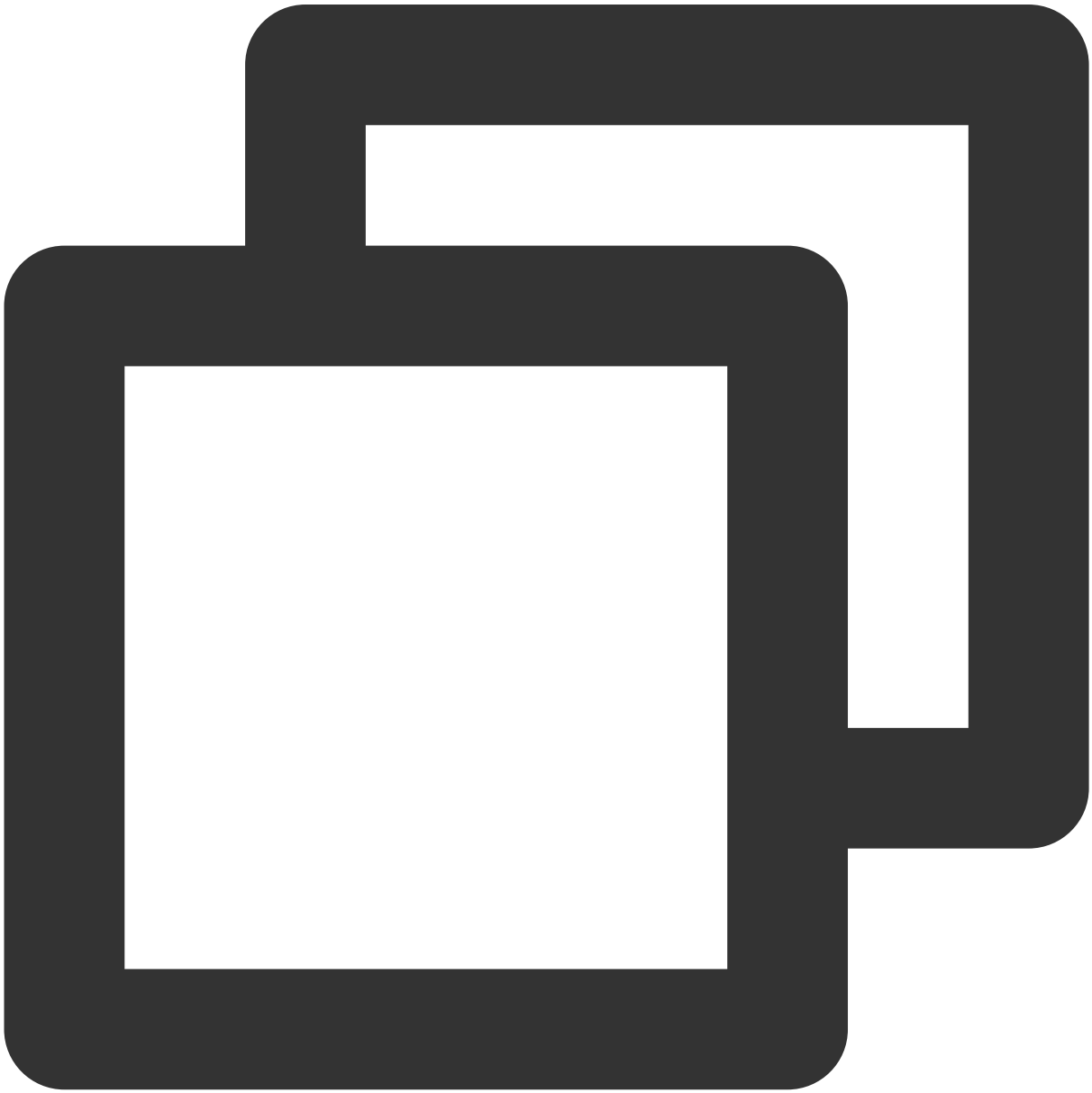


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

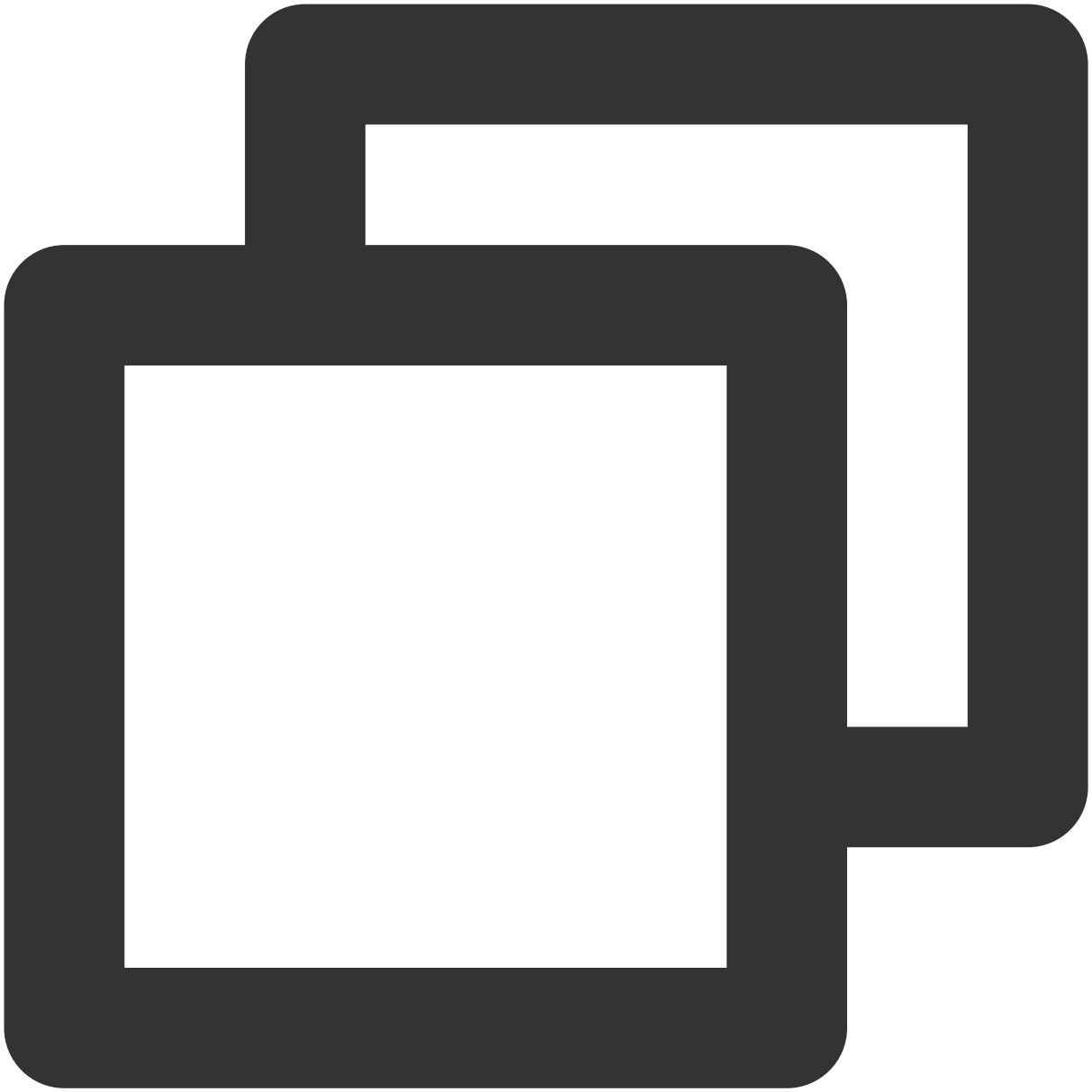
最近更新时间：2024-01-23 10:00:02

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

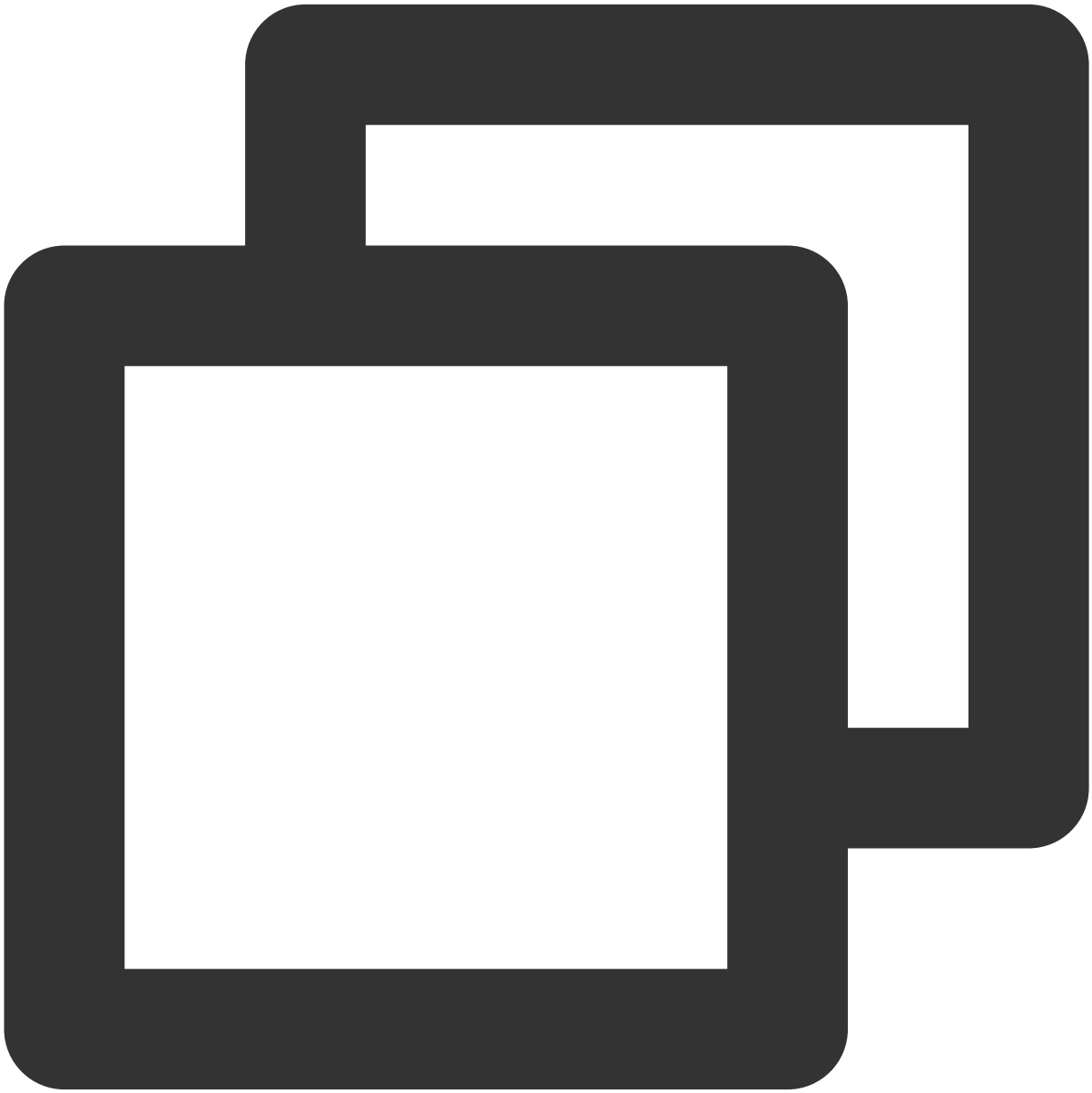
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

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：

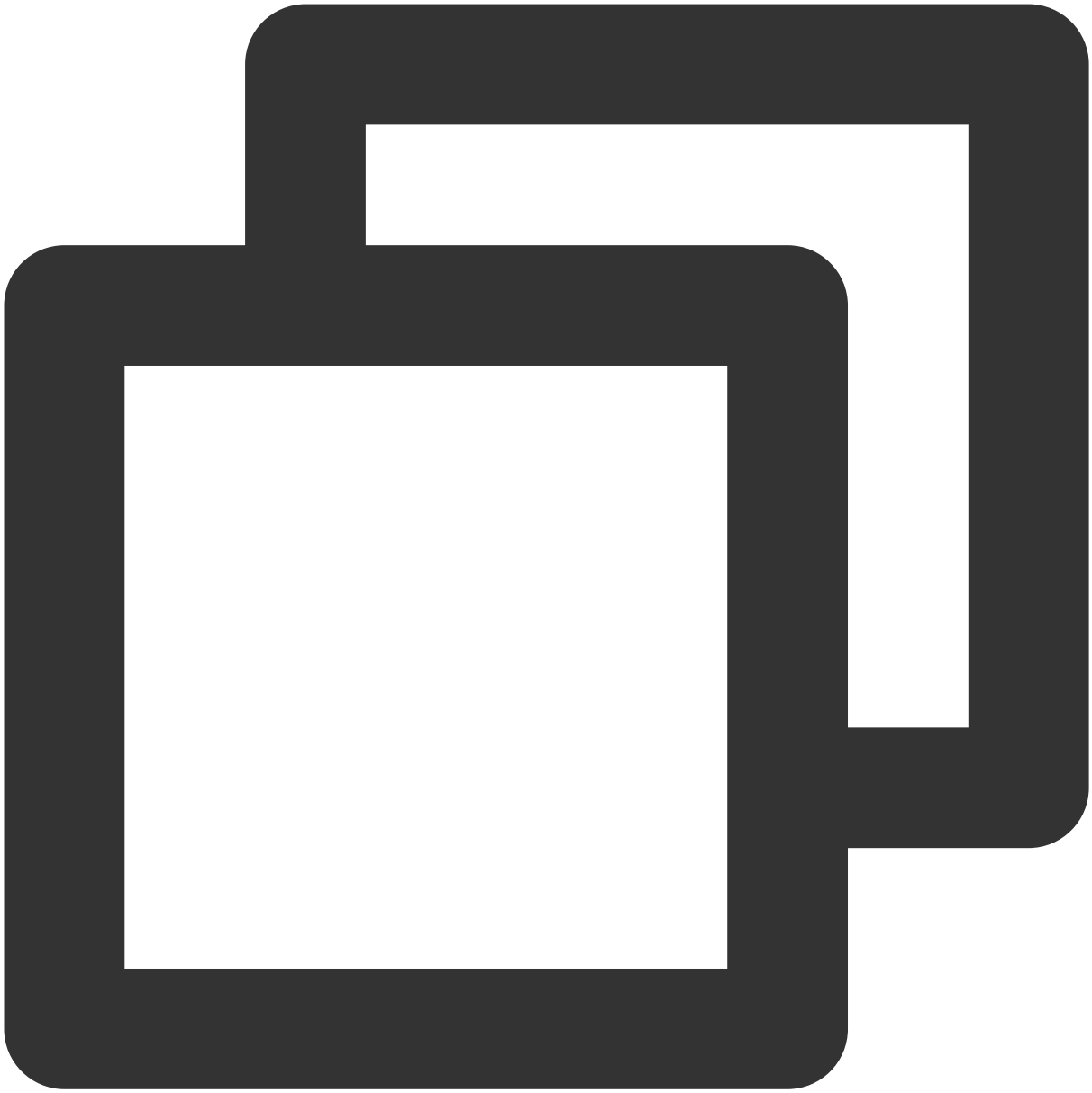
节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TemplateId	Response	被删除的模板的 ID	String

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。

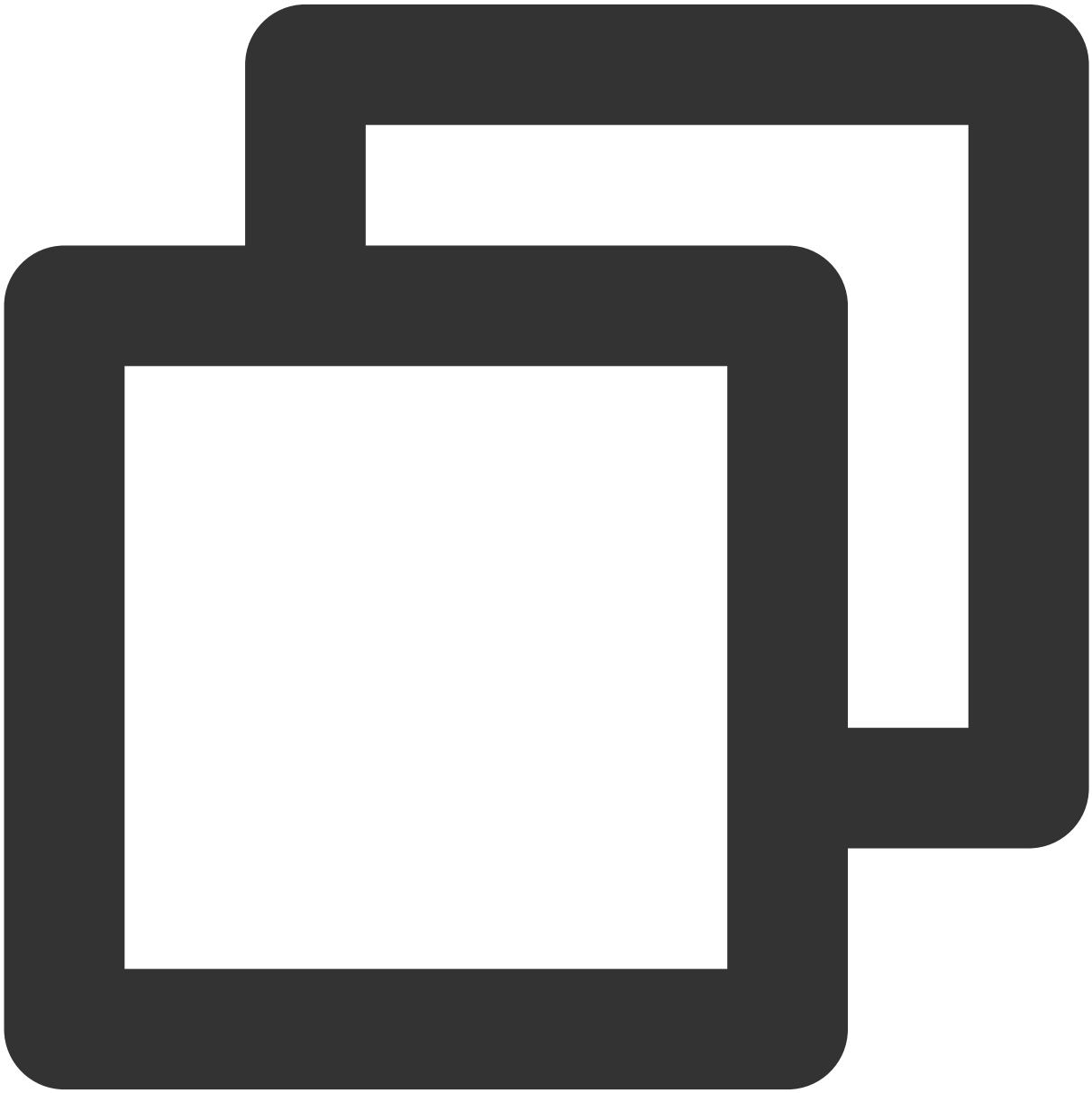
实际案例

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

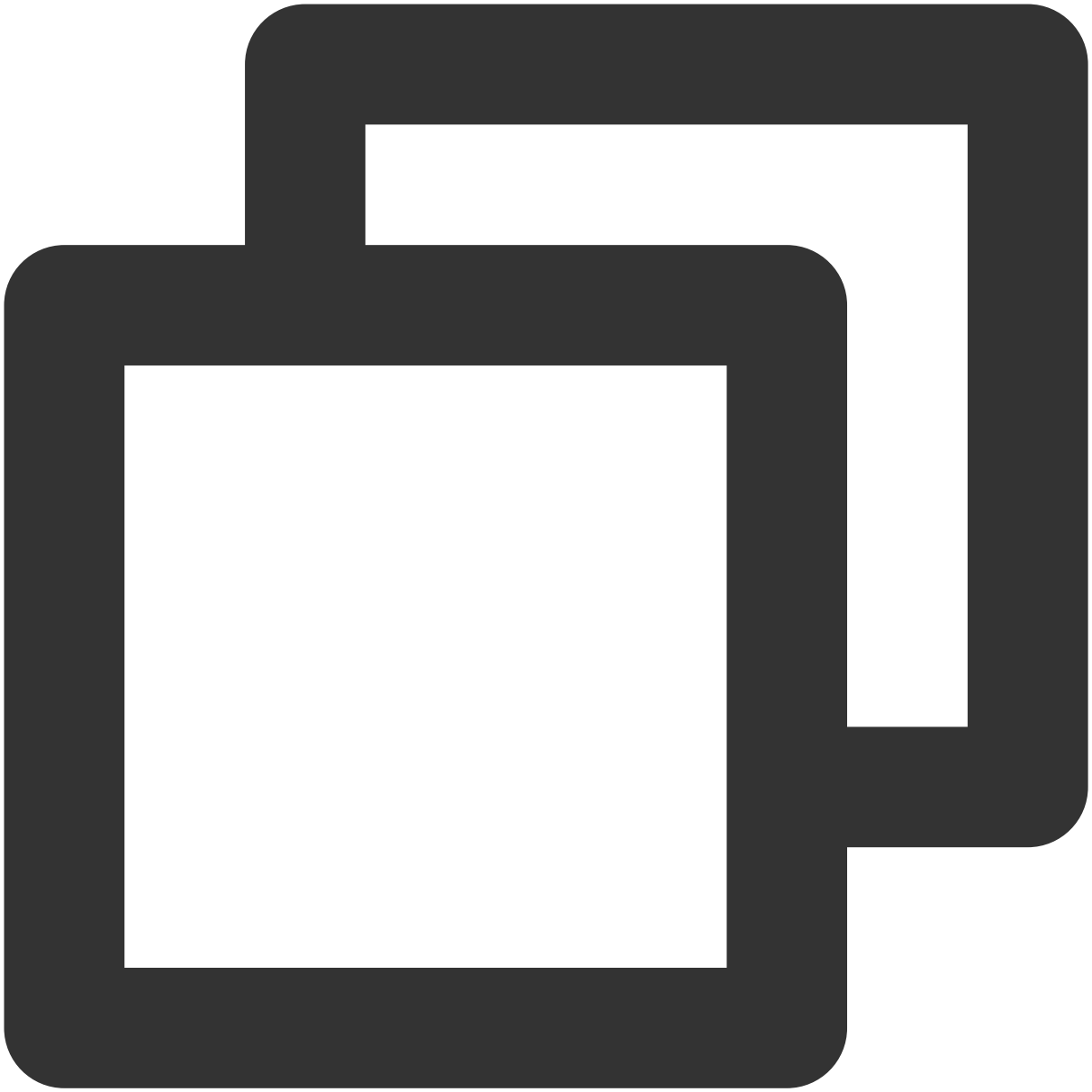
最近更新时间：2024-01-23 10:00:02

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

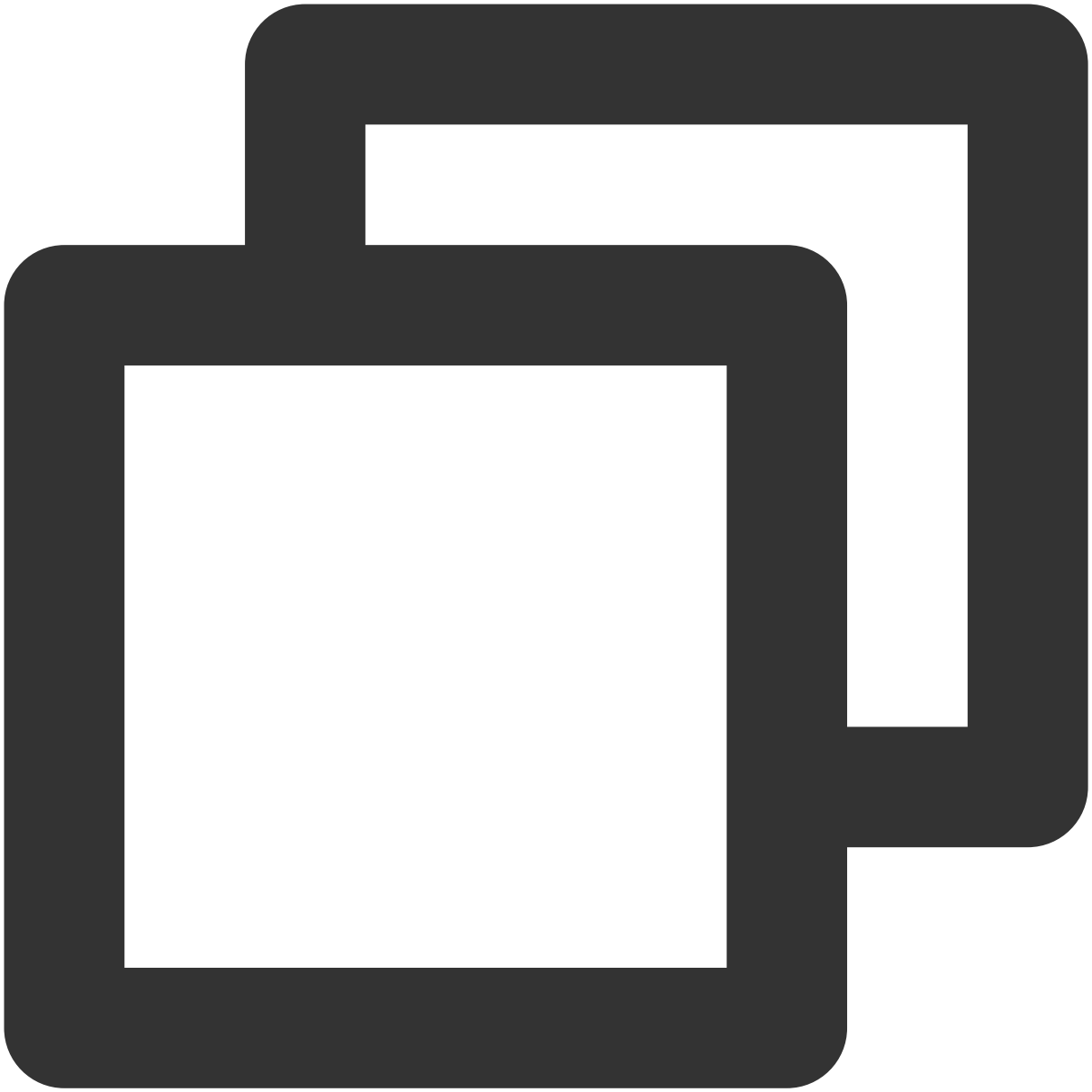
响应

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

版权所有：腾讯云计算（北京）有限责任公司

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

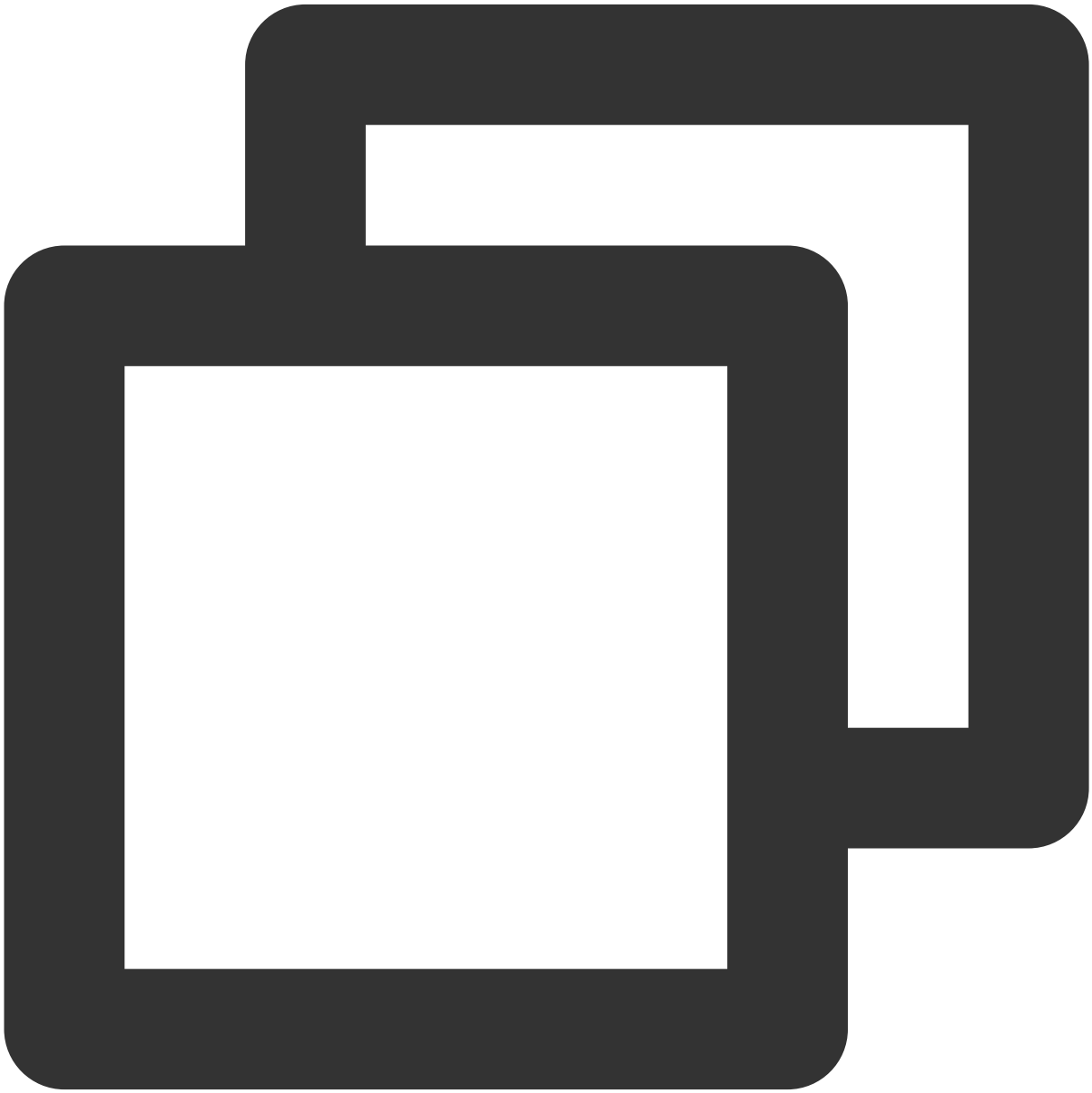
错误码

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

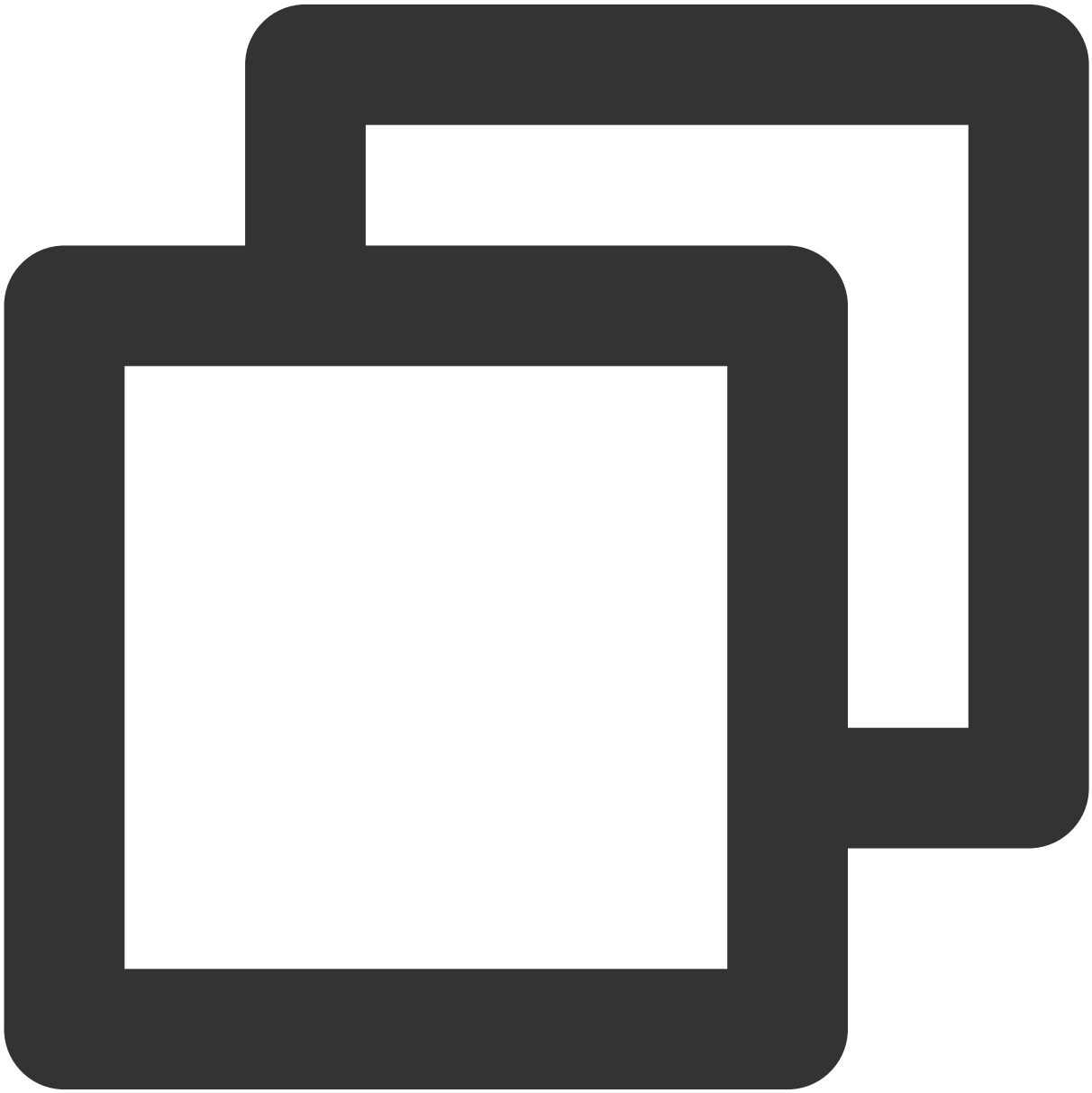
案例一：按照模板 ID 维度查询

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****

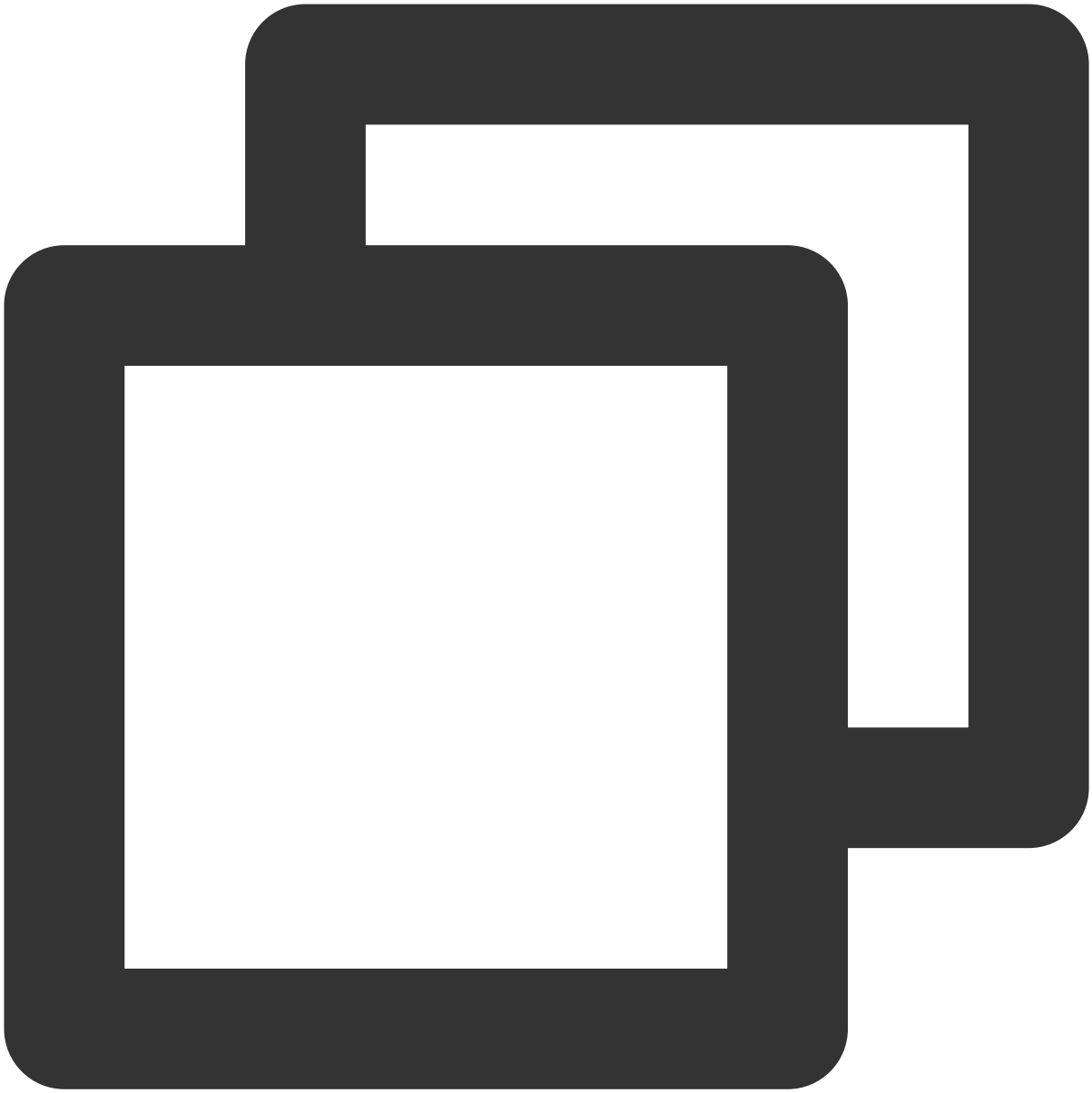
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <TemplateList>
```

```
        <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>
```

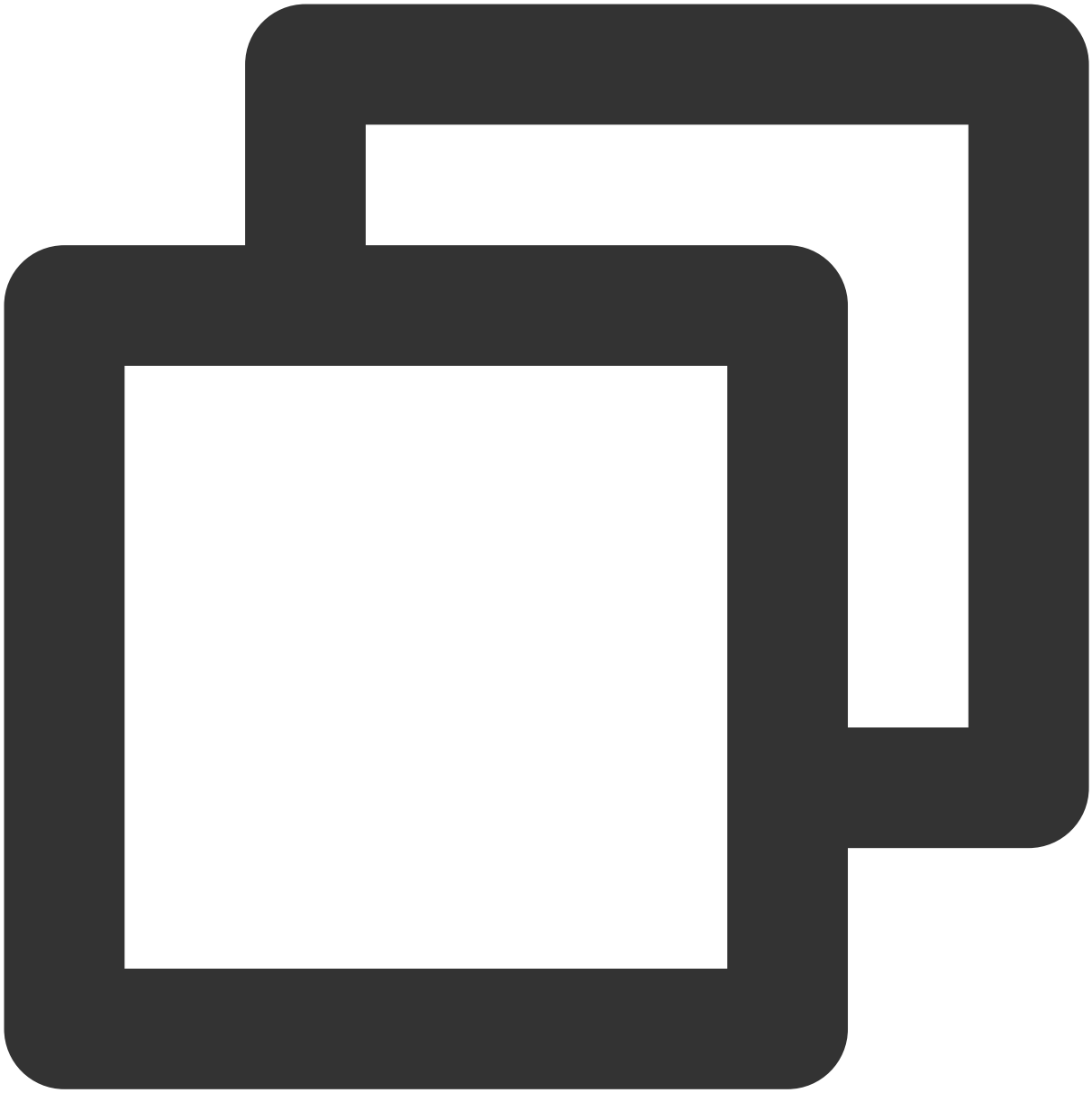
案例二：按照分页列表维度查询

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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>
```

媒体处理
更新动图模板

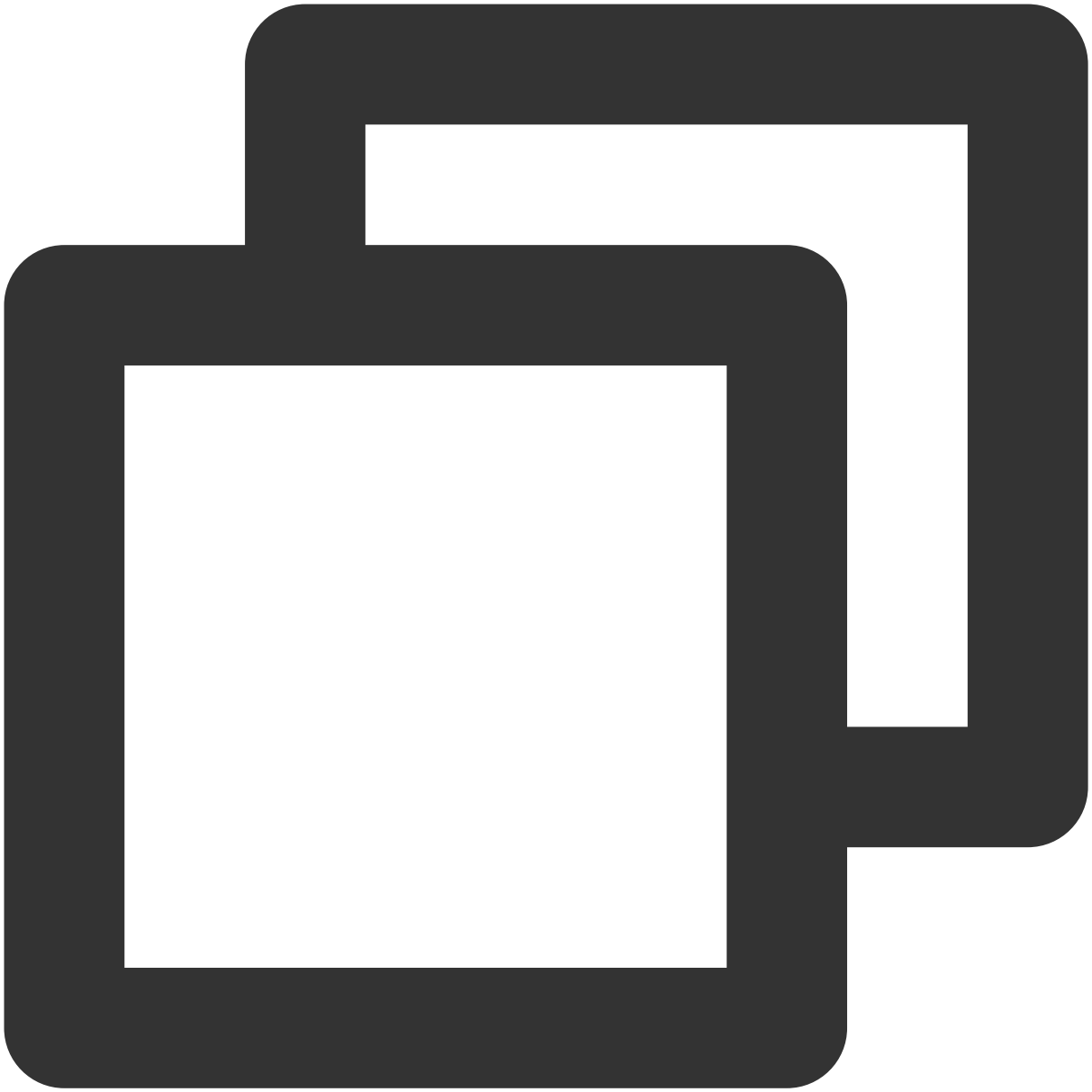
最近更新时间：2024-01-23 10:00:02

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：

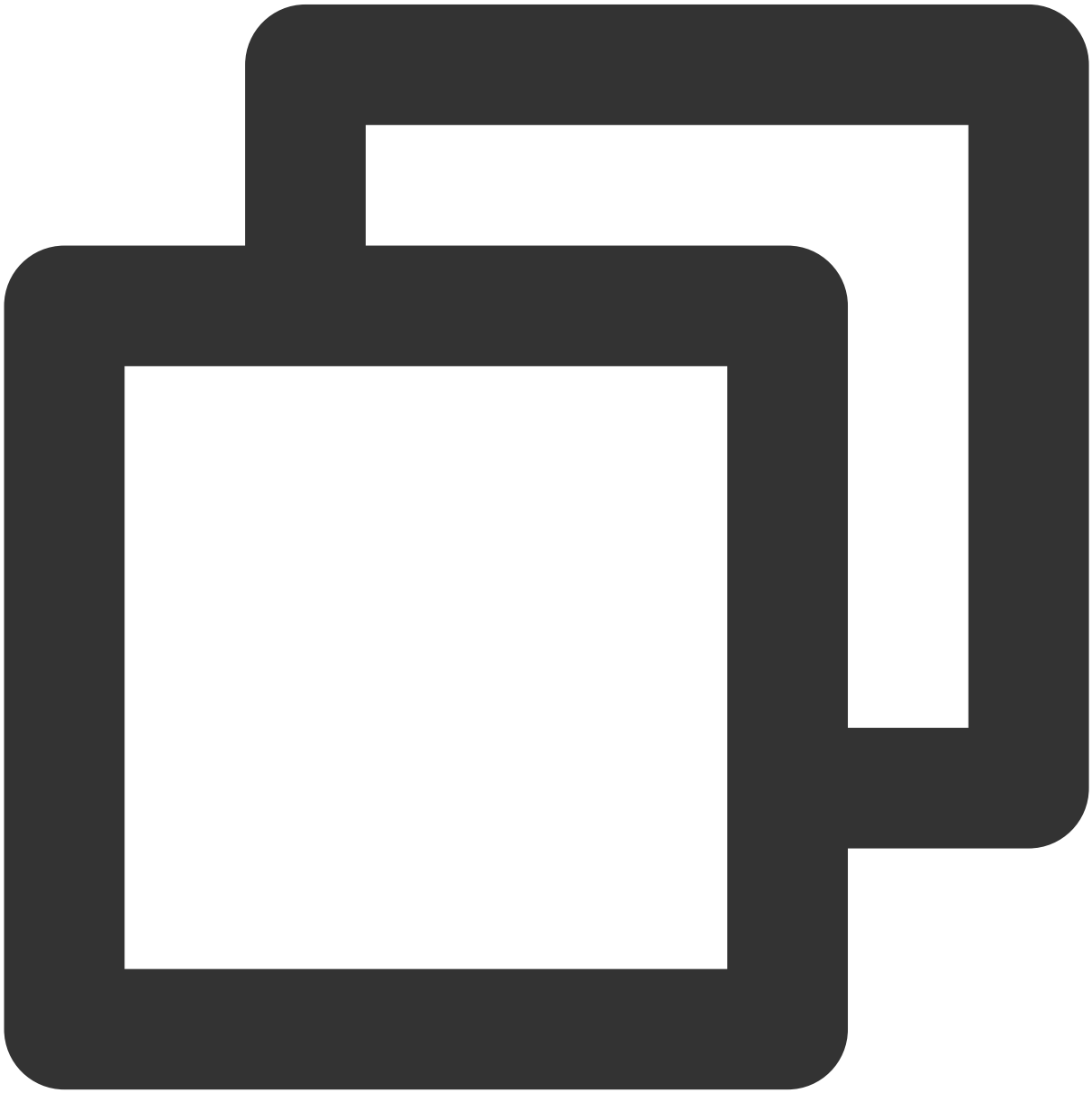
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

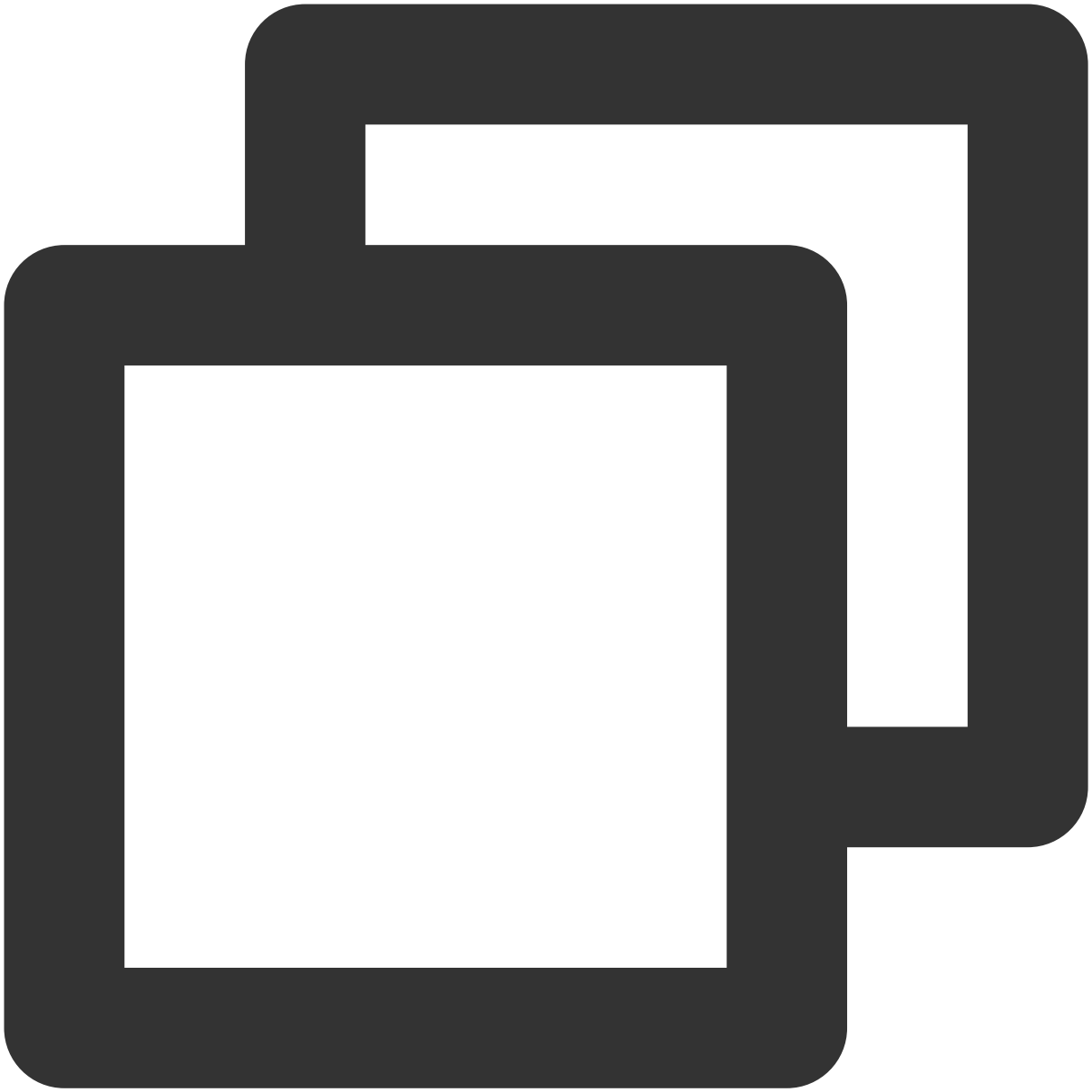
响应

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：

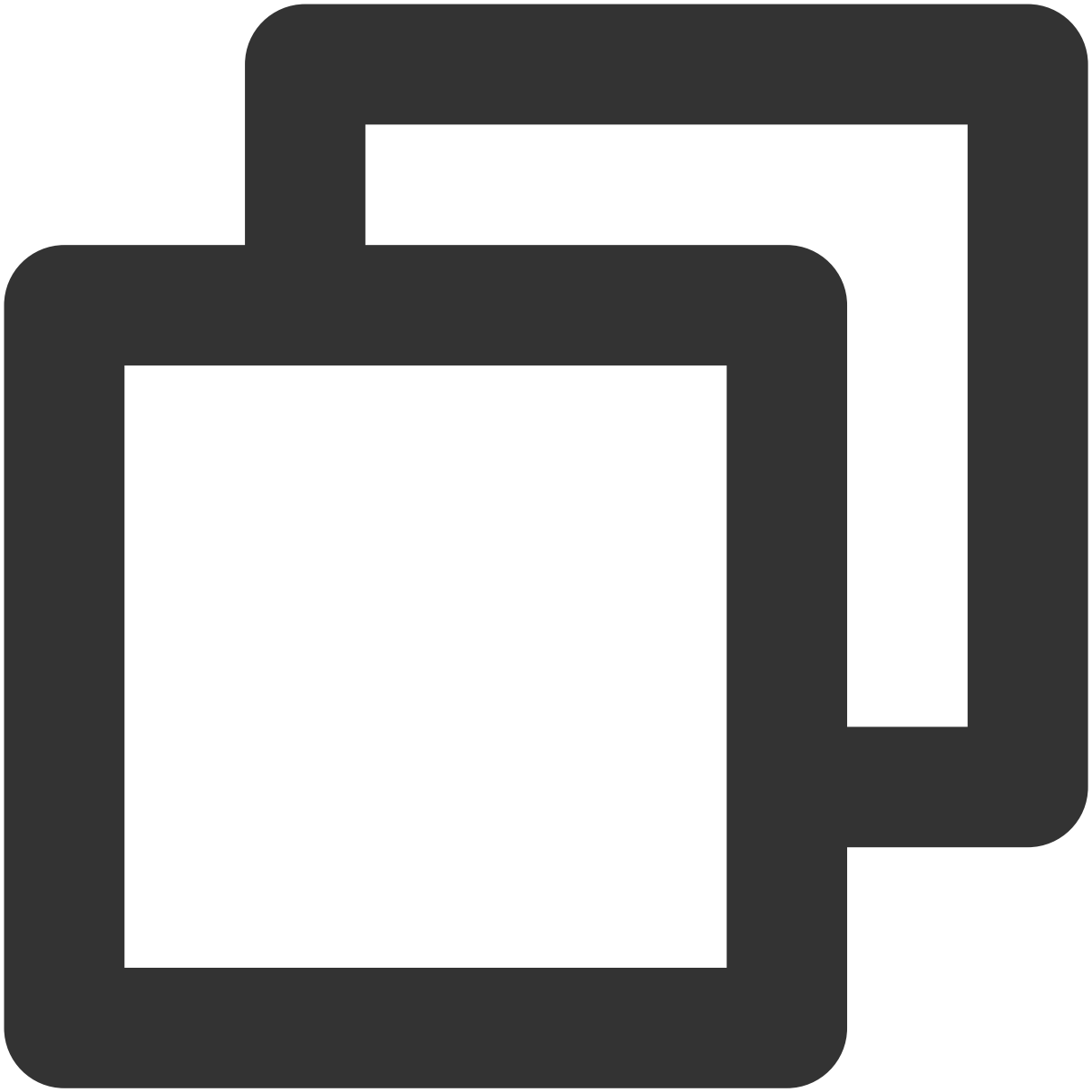
节点名称（关键字）	父节点	描述	类型
Response	无	同创建动图模板接口的 Response	Container

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

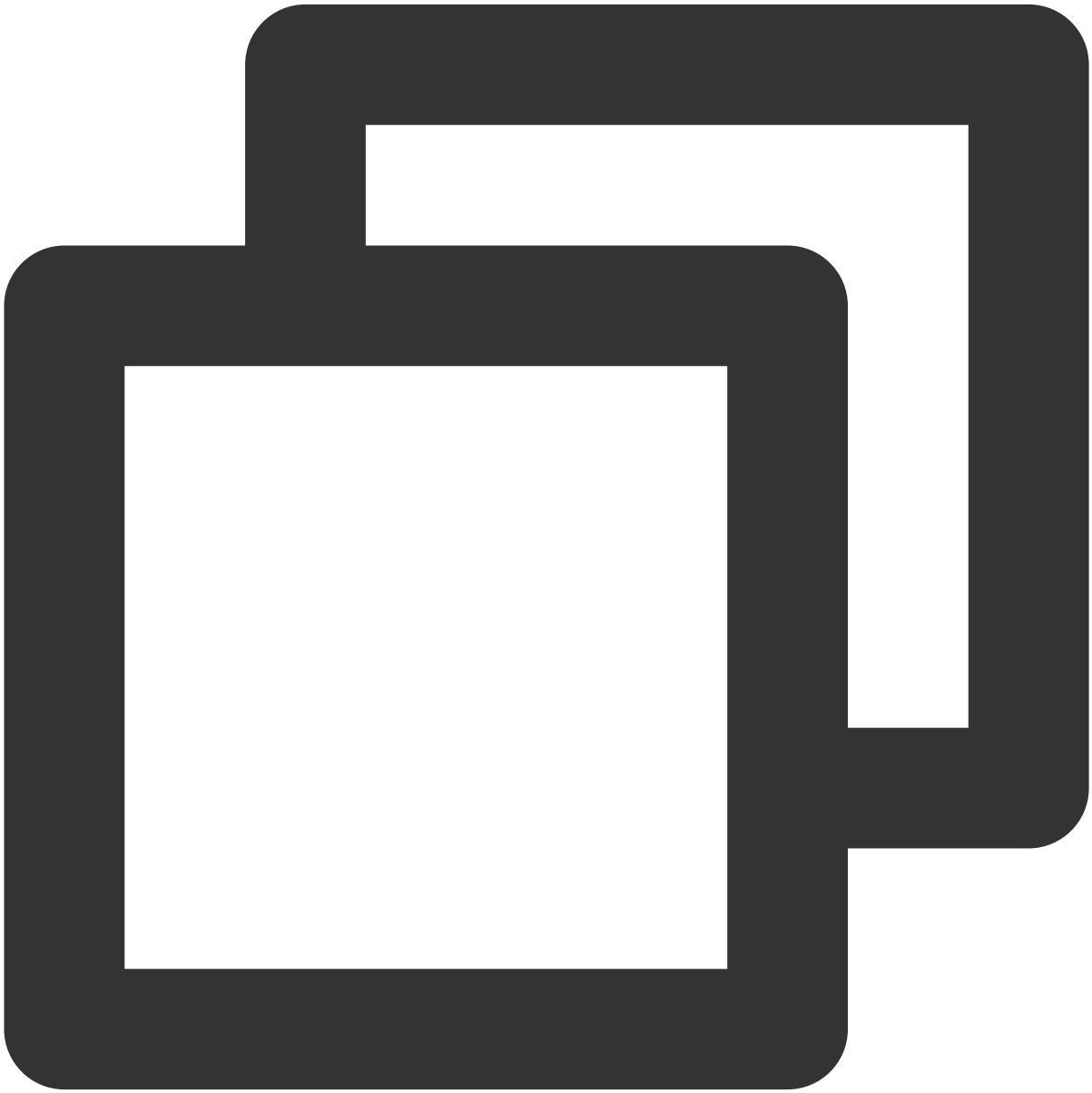


```
PUT /template/t1f16e1dfbdc994105b31292d45710642a HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

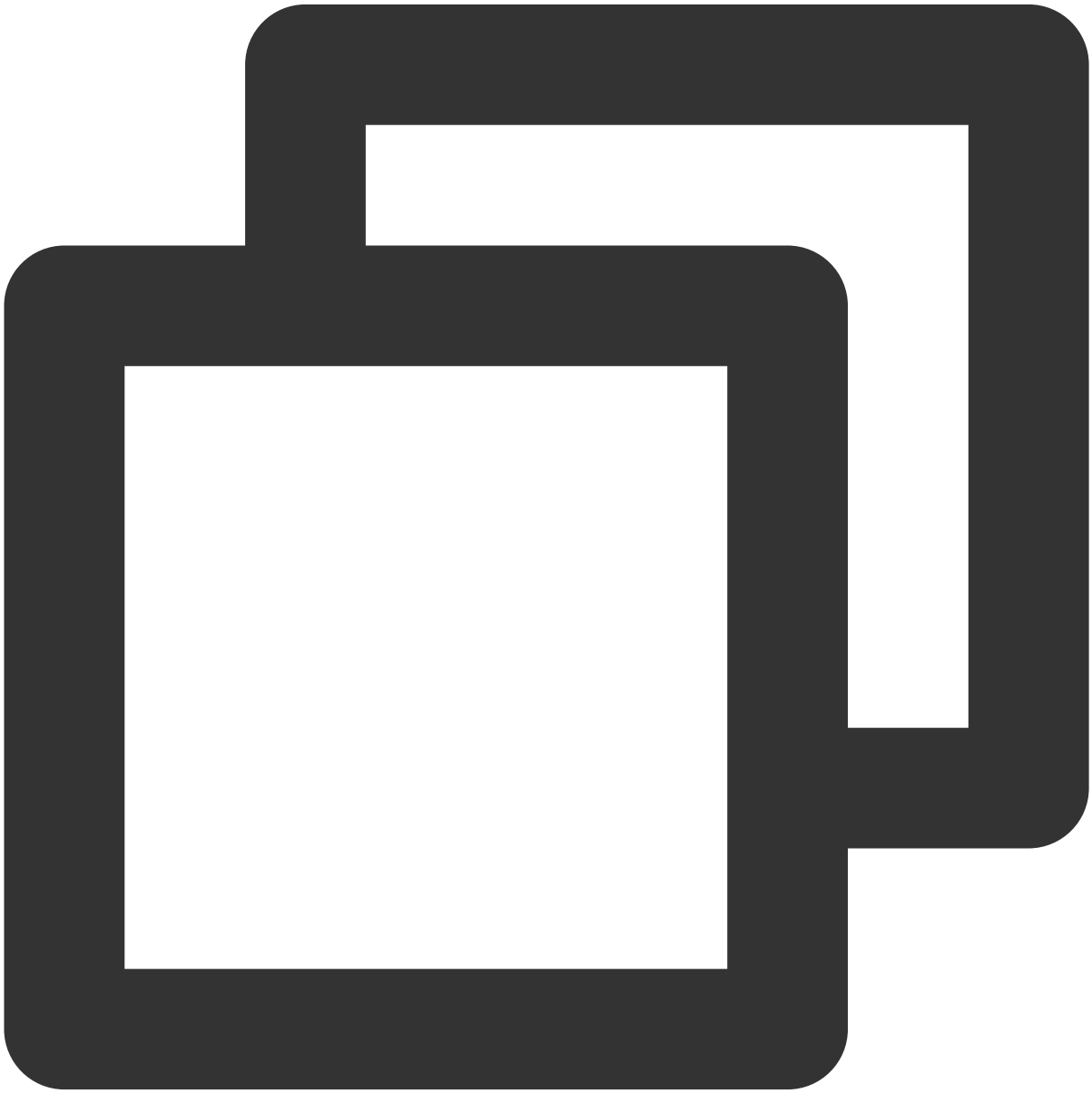
最近更新时间：2024-01-23 10:00:02

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

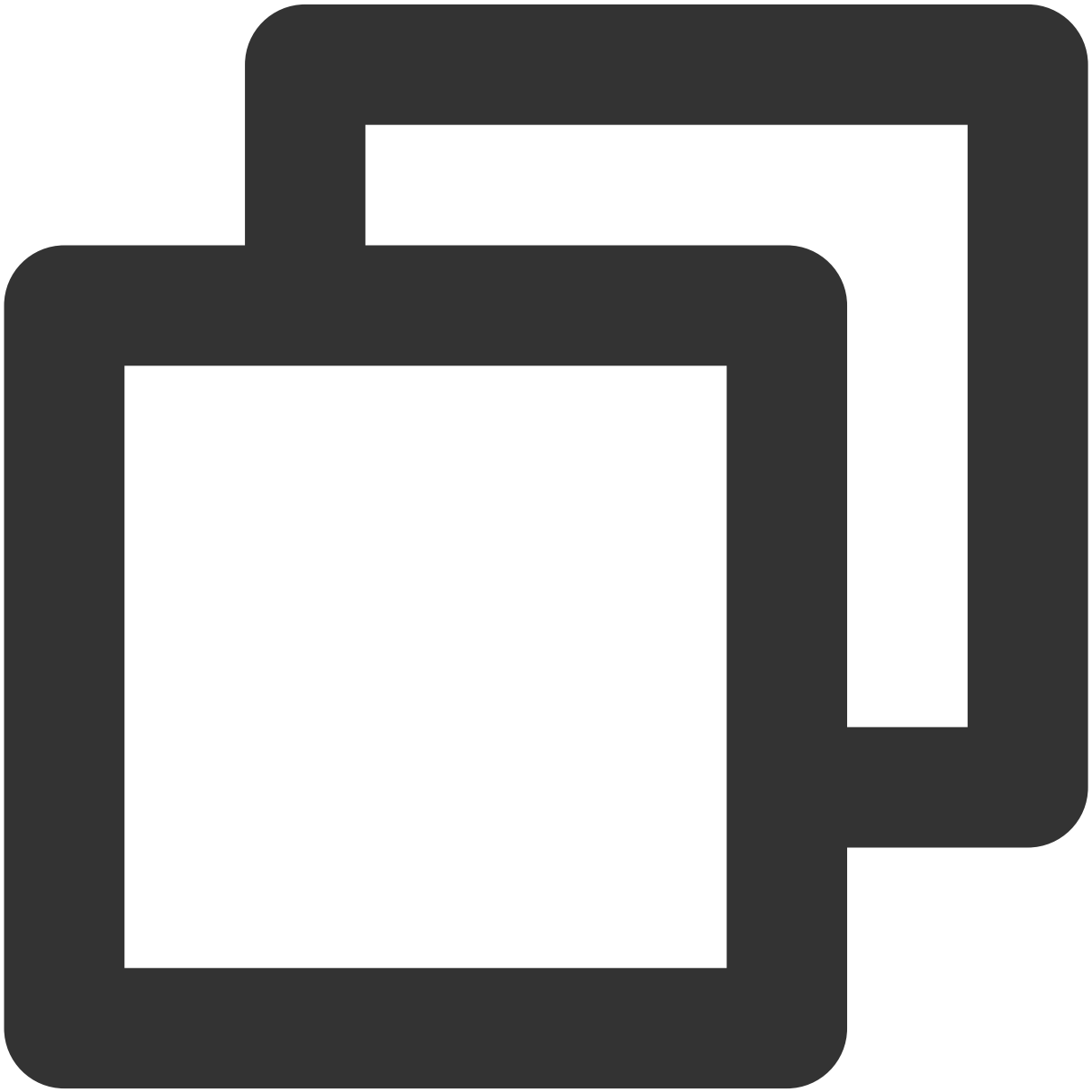
该请求操作的实现使用公共请求头，了解公共请求头详情请参见 公共请求头部 文档。

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

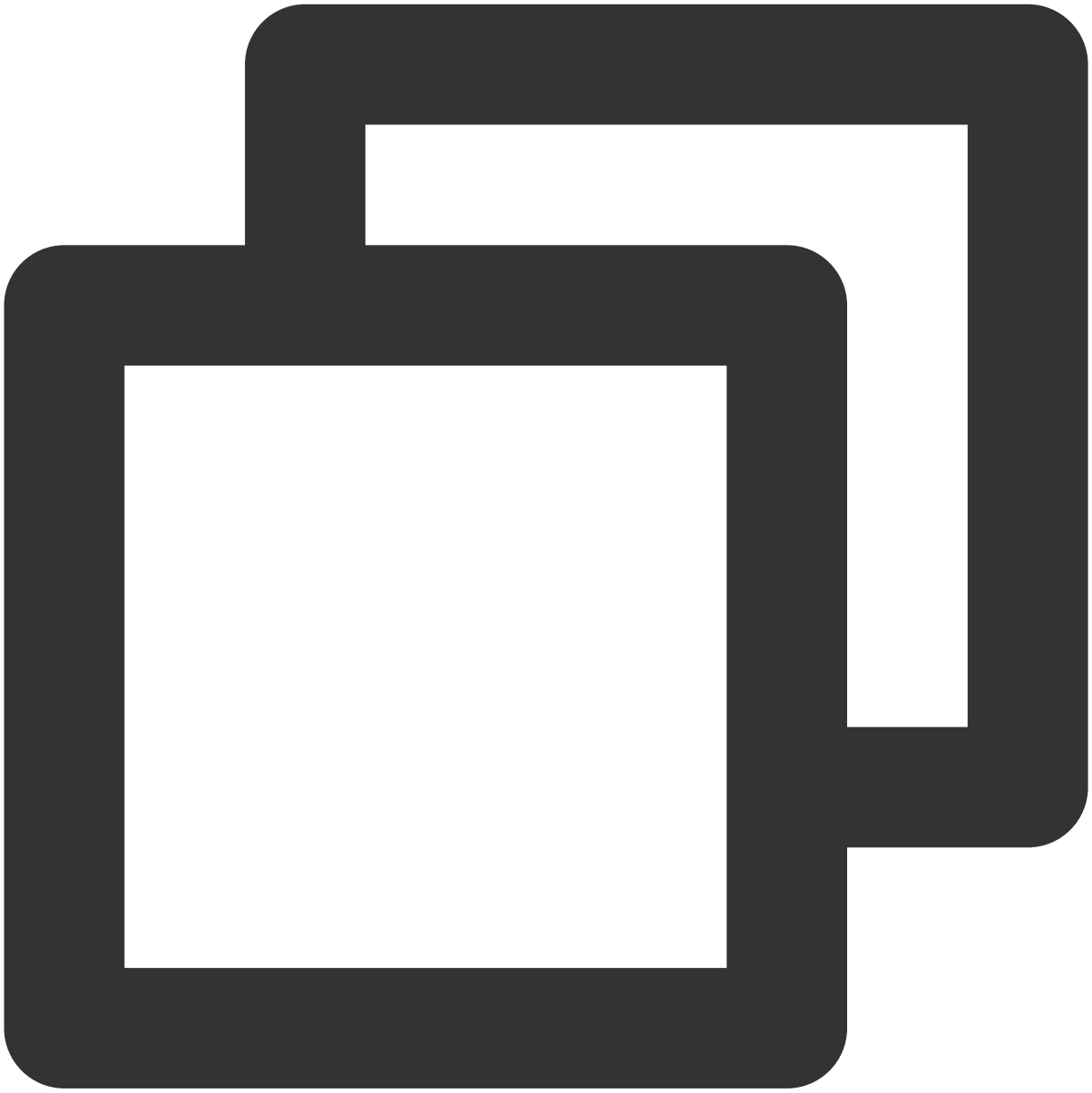
该响应包含公共响应头，了解公共响应头详情请参见 公共响应头部 文档。

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>
```

具体的数据内容如下：

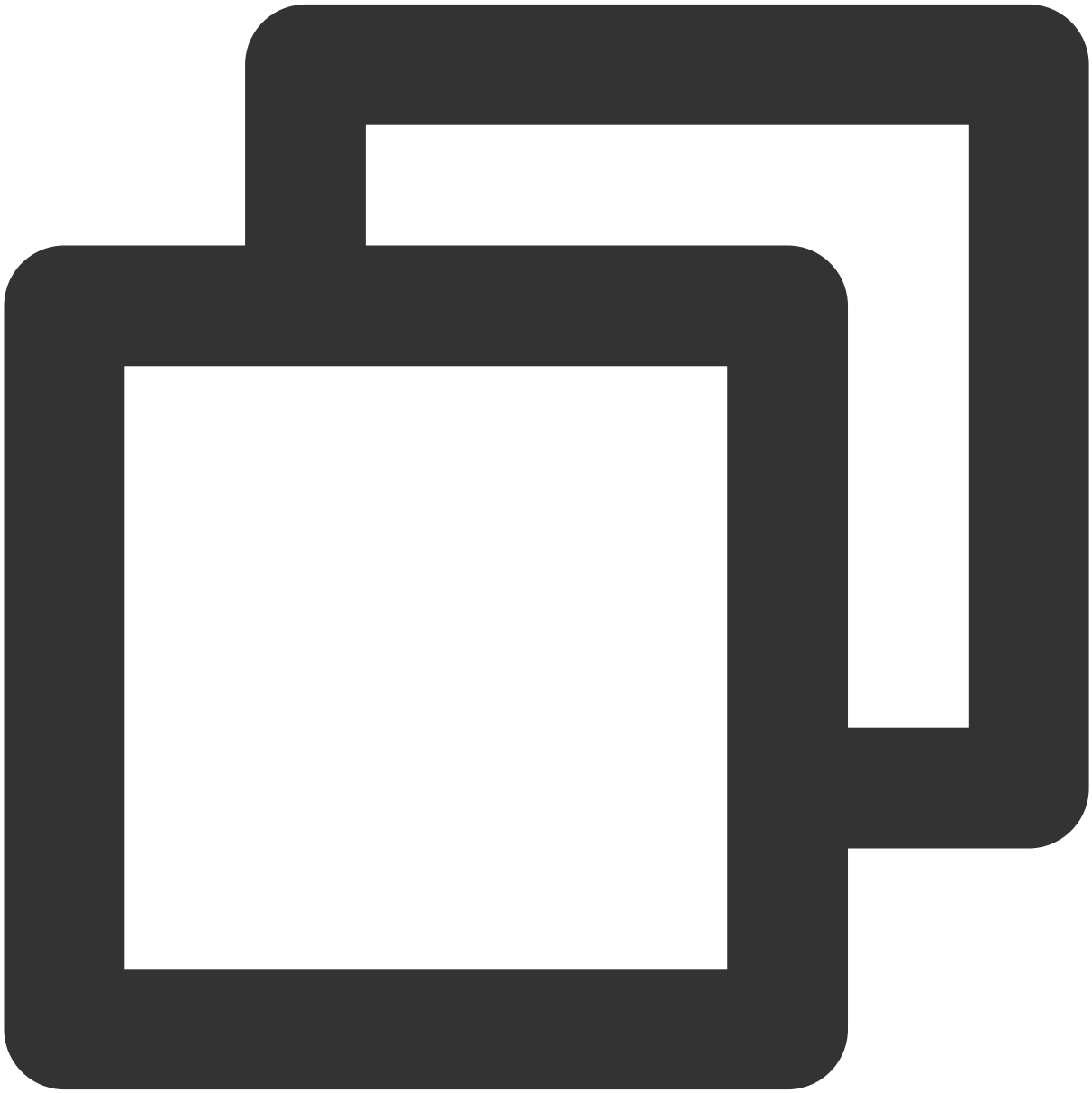
节点名称（关键字）	父节点	描述	类型
Response	无	同创建拼接模板接口的 Response	Container

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

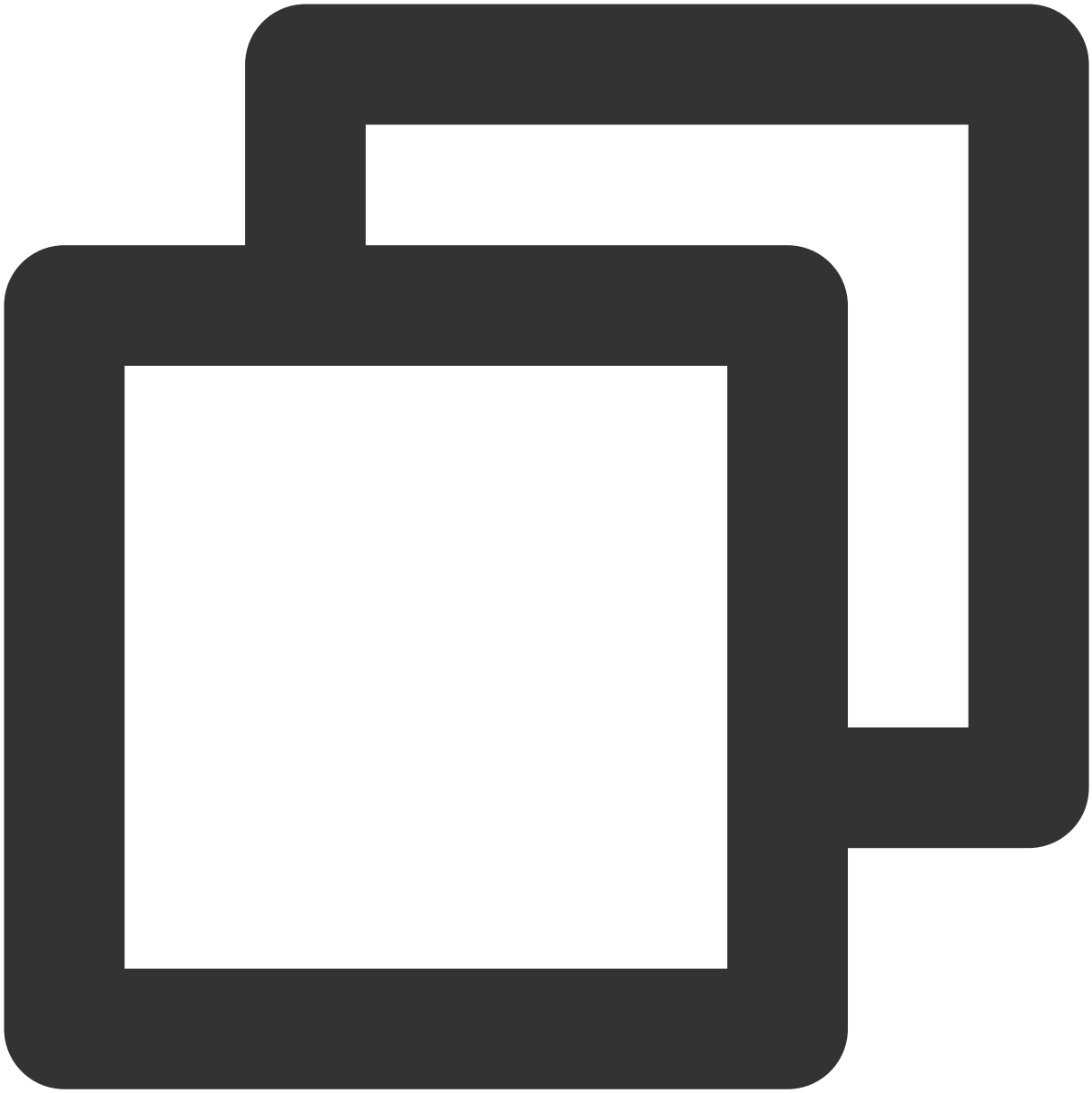


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98****-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

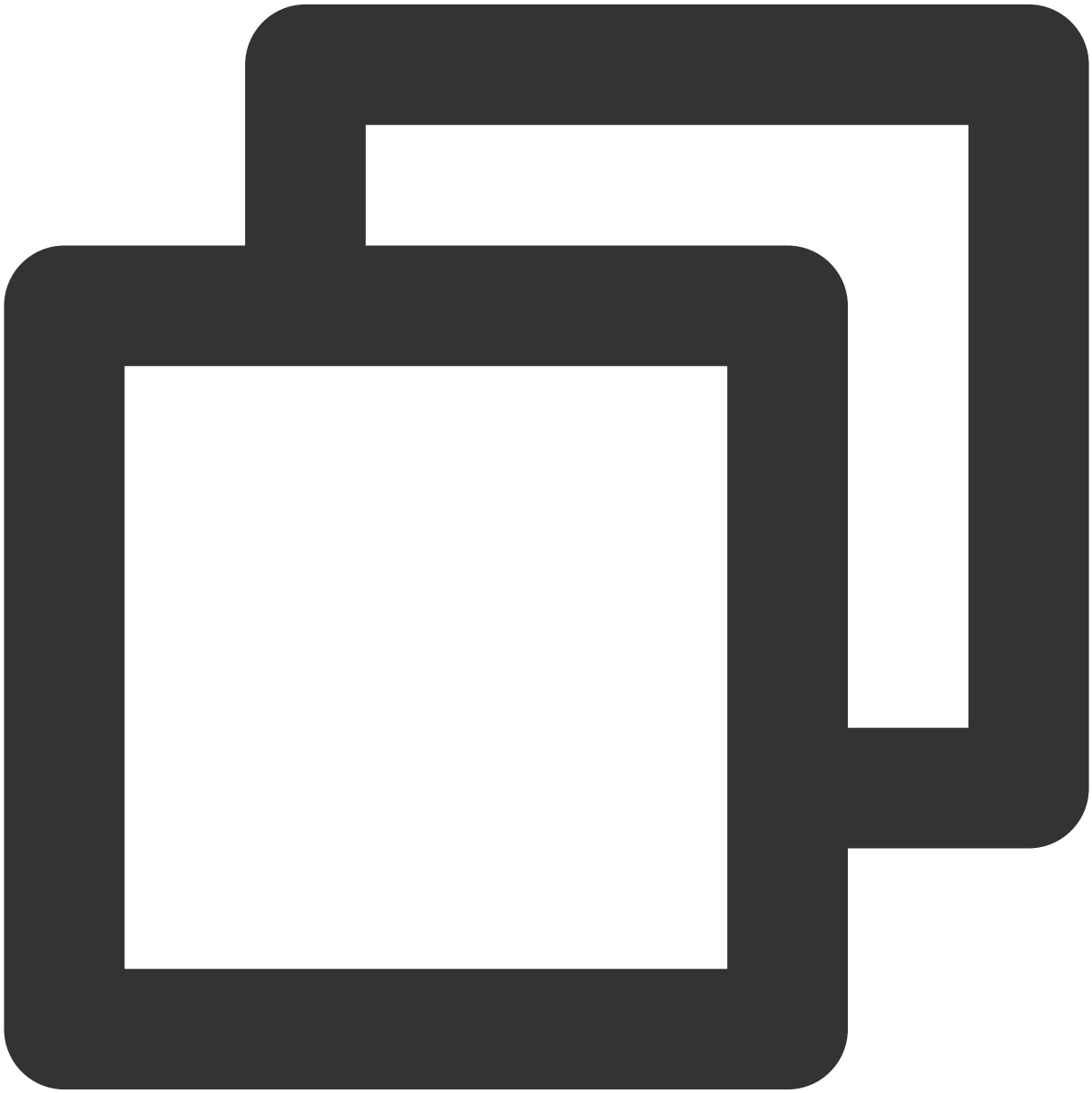
最近更新时间：2024-01-23 10:00:02

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：

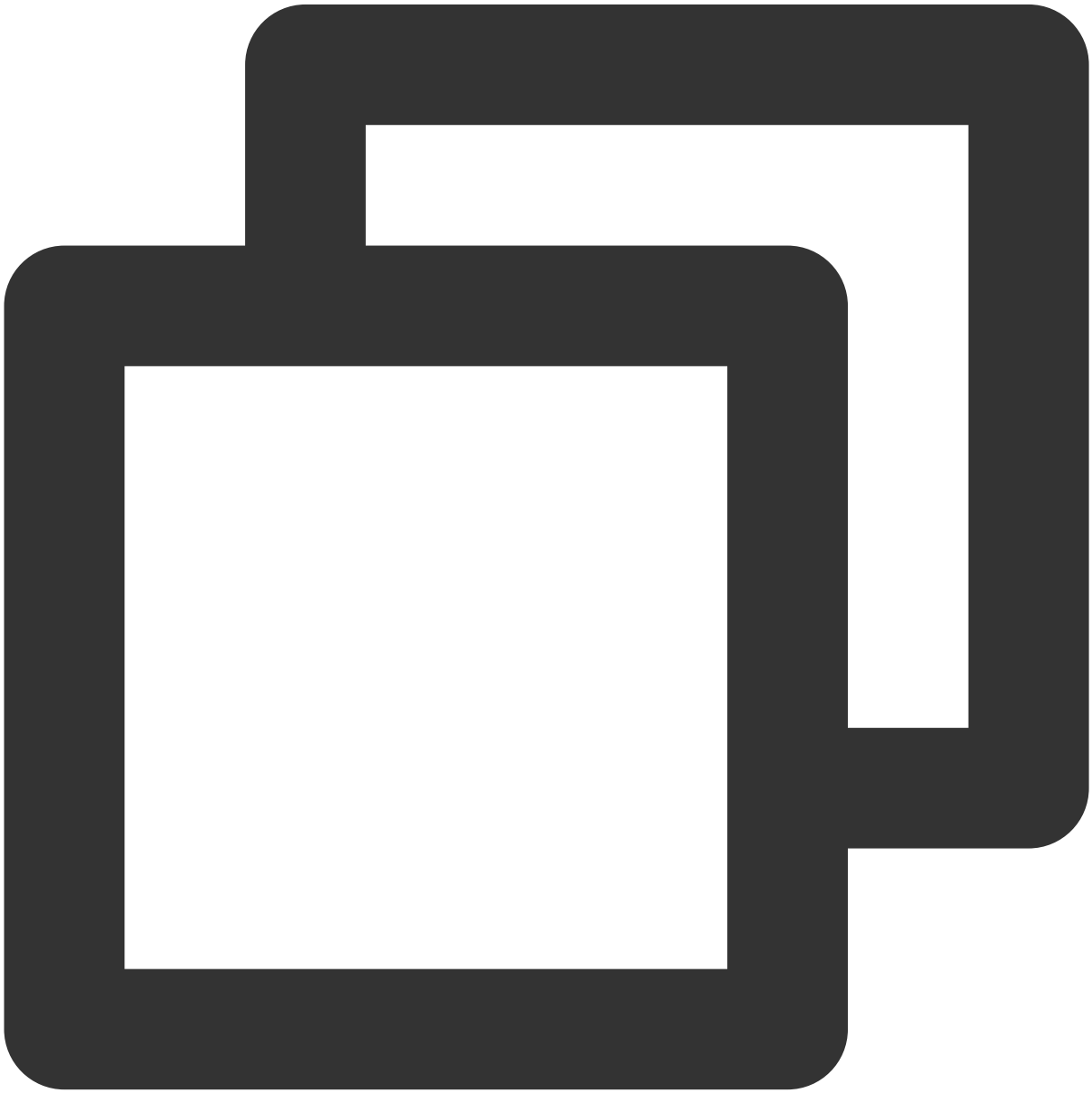
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

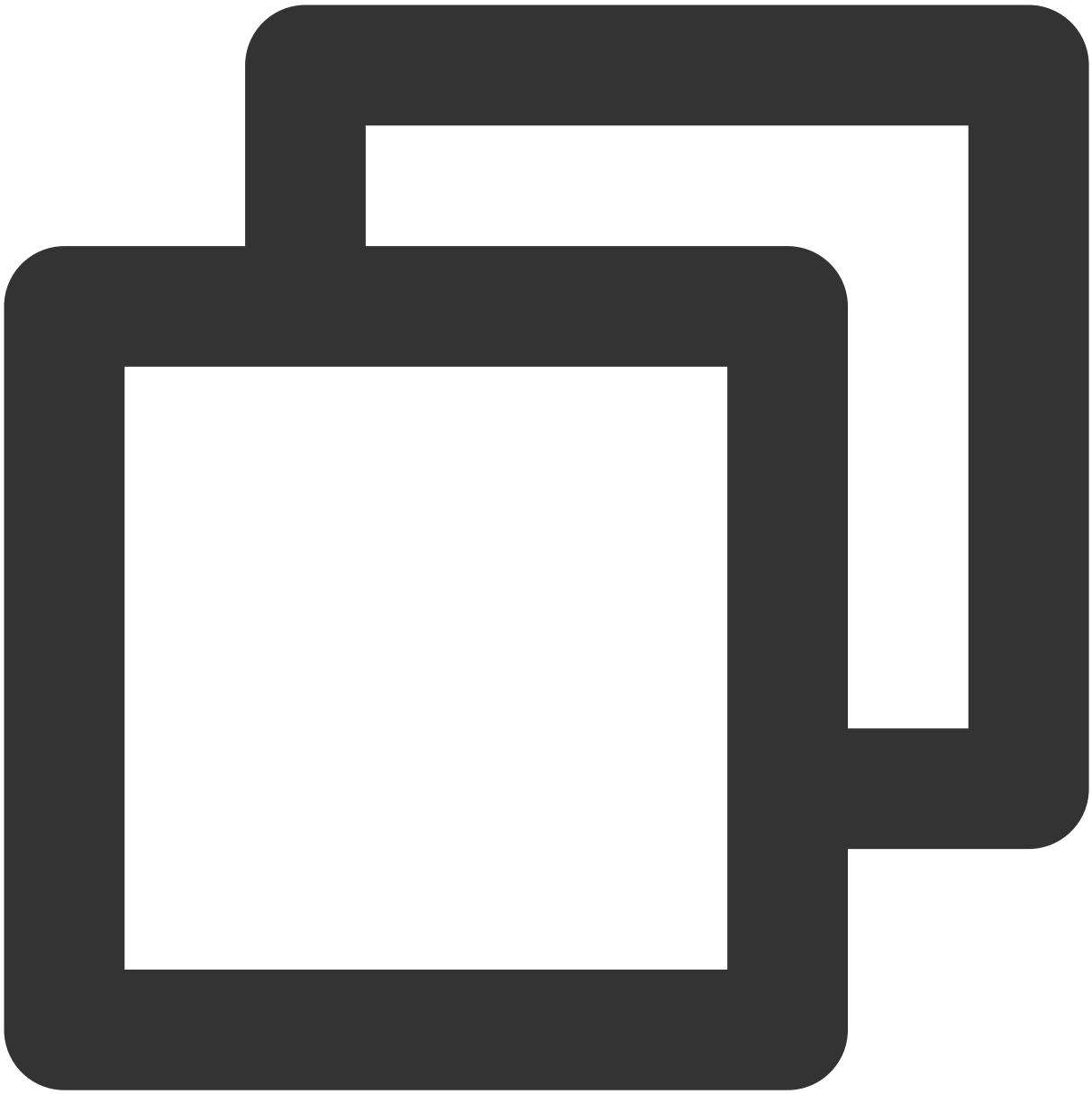
响应

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：

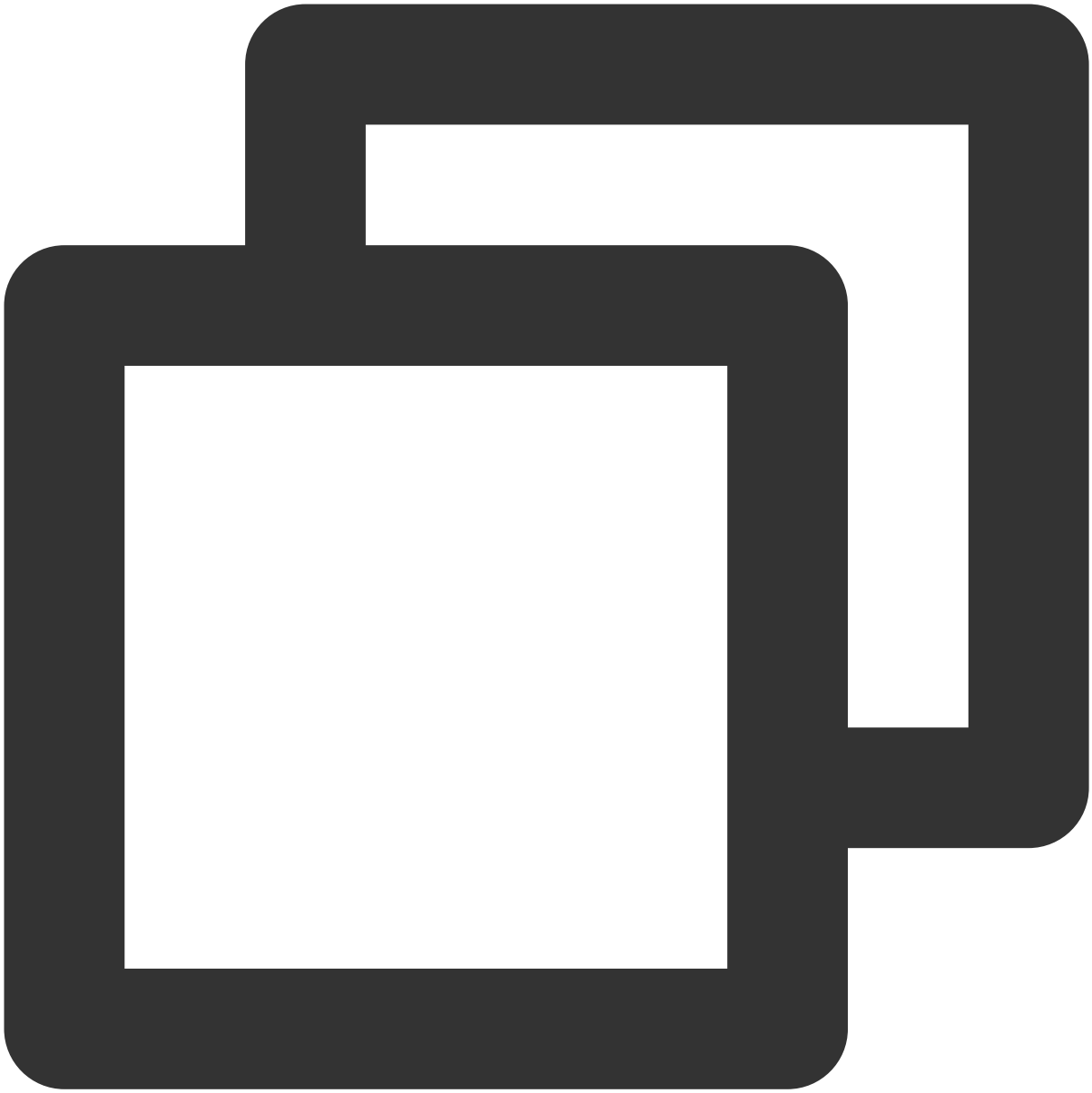
节点名称（关键字）	父节点	描述	类型
Response	无	同创建极速高清模板接口的 Response	Container

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

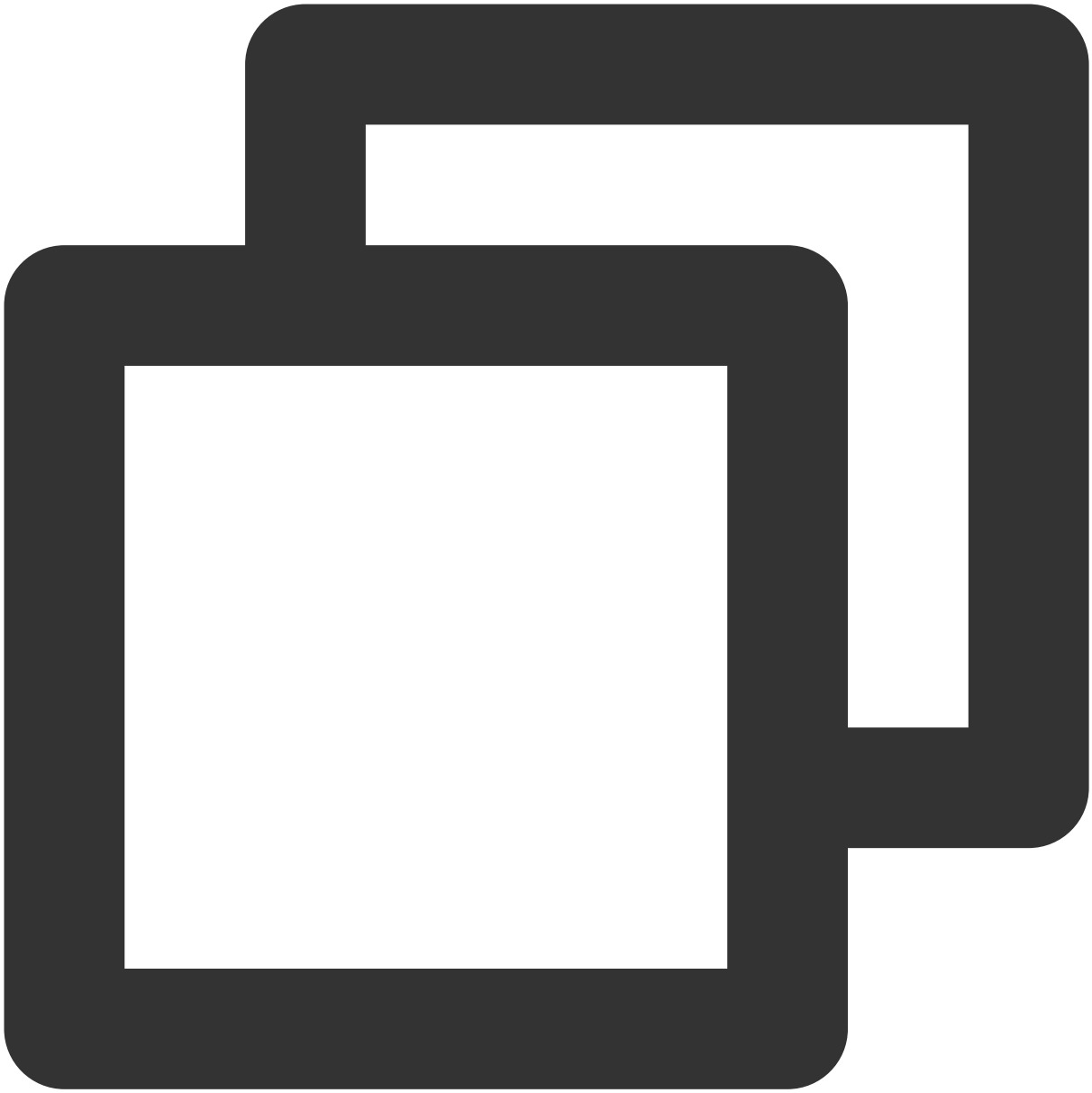


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

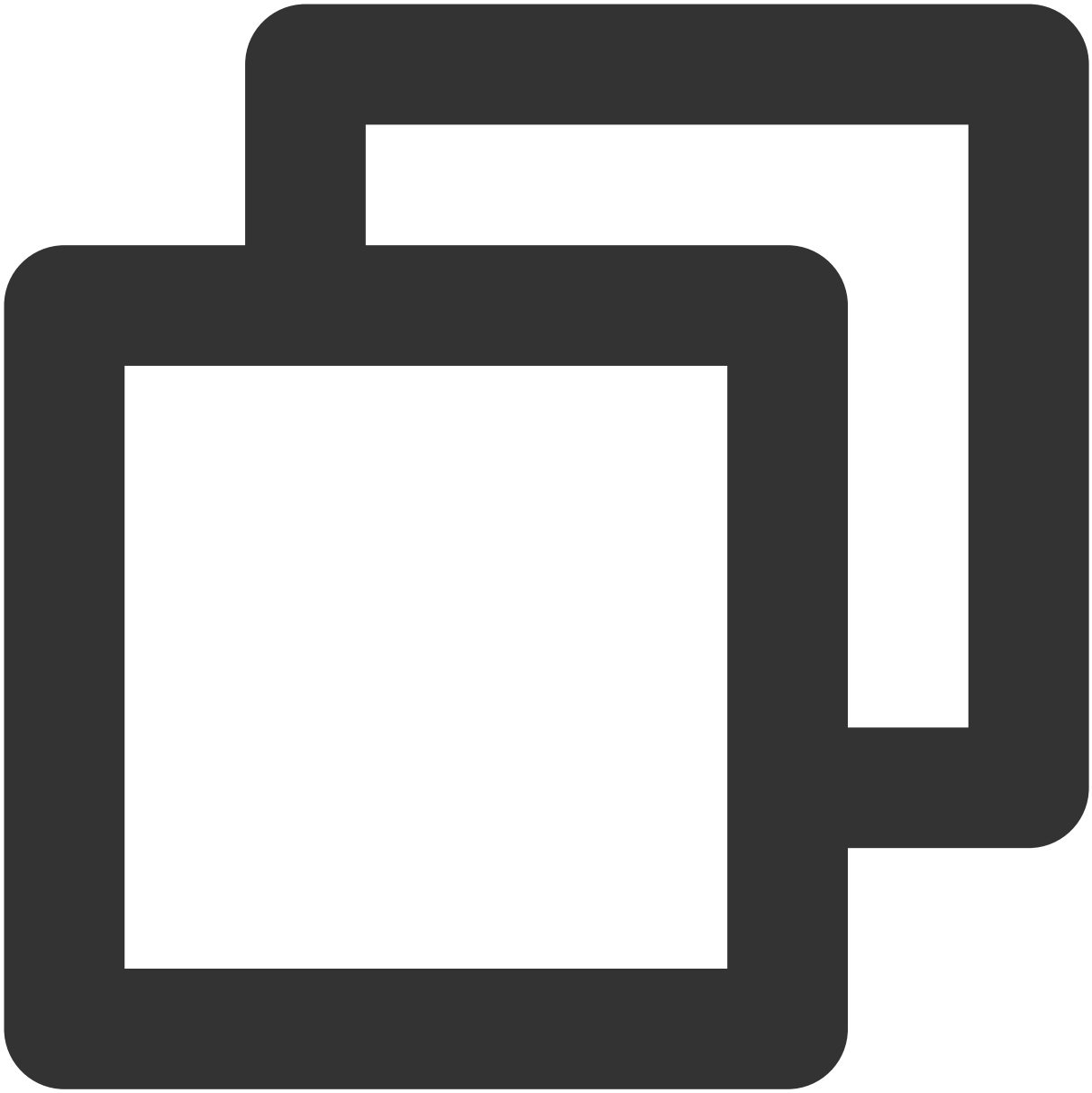
最近更新时间：2024-01-23 10:00:02

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：

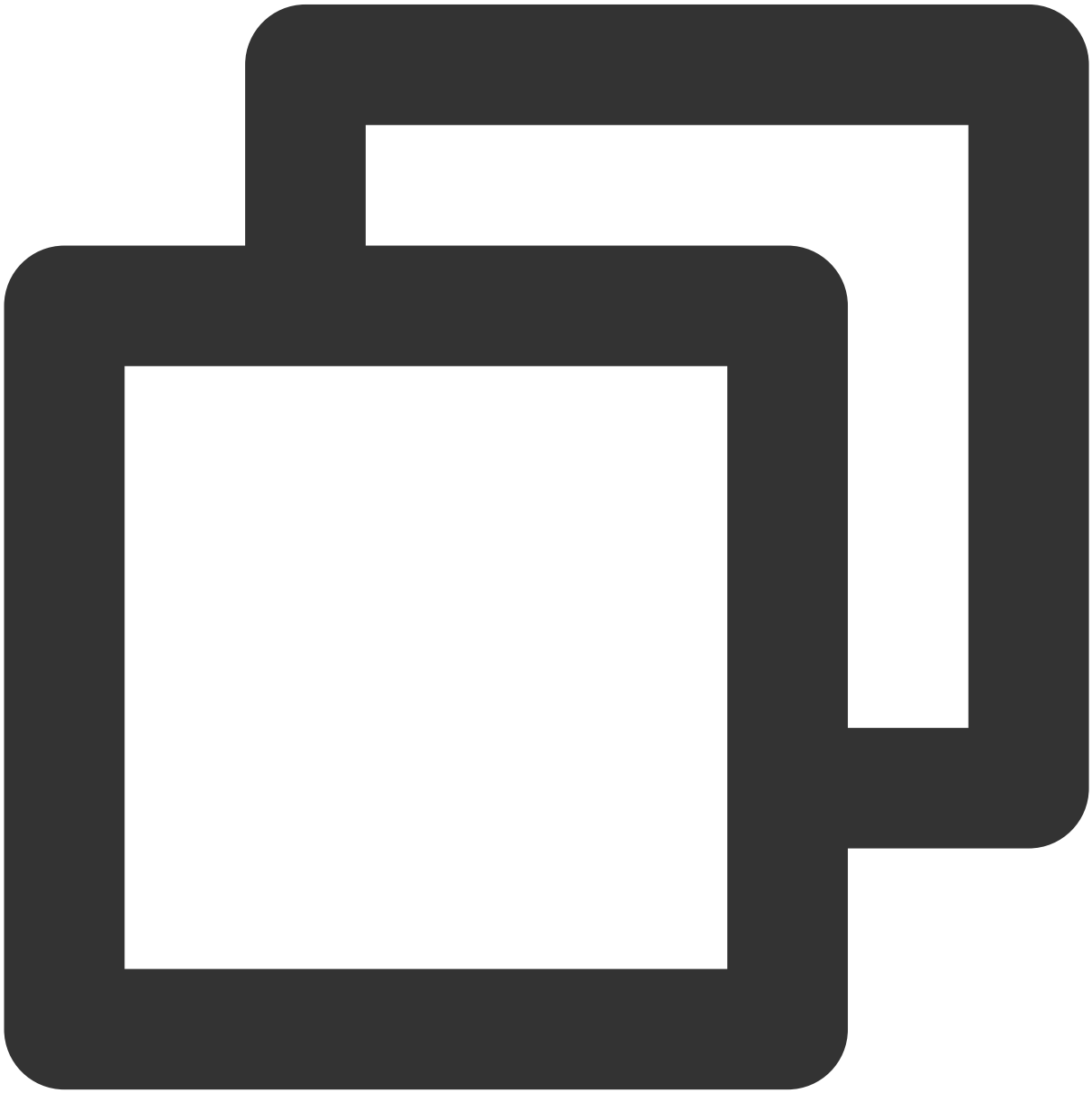
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

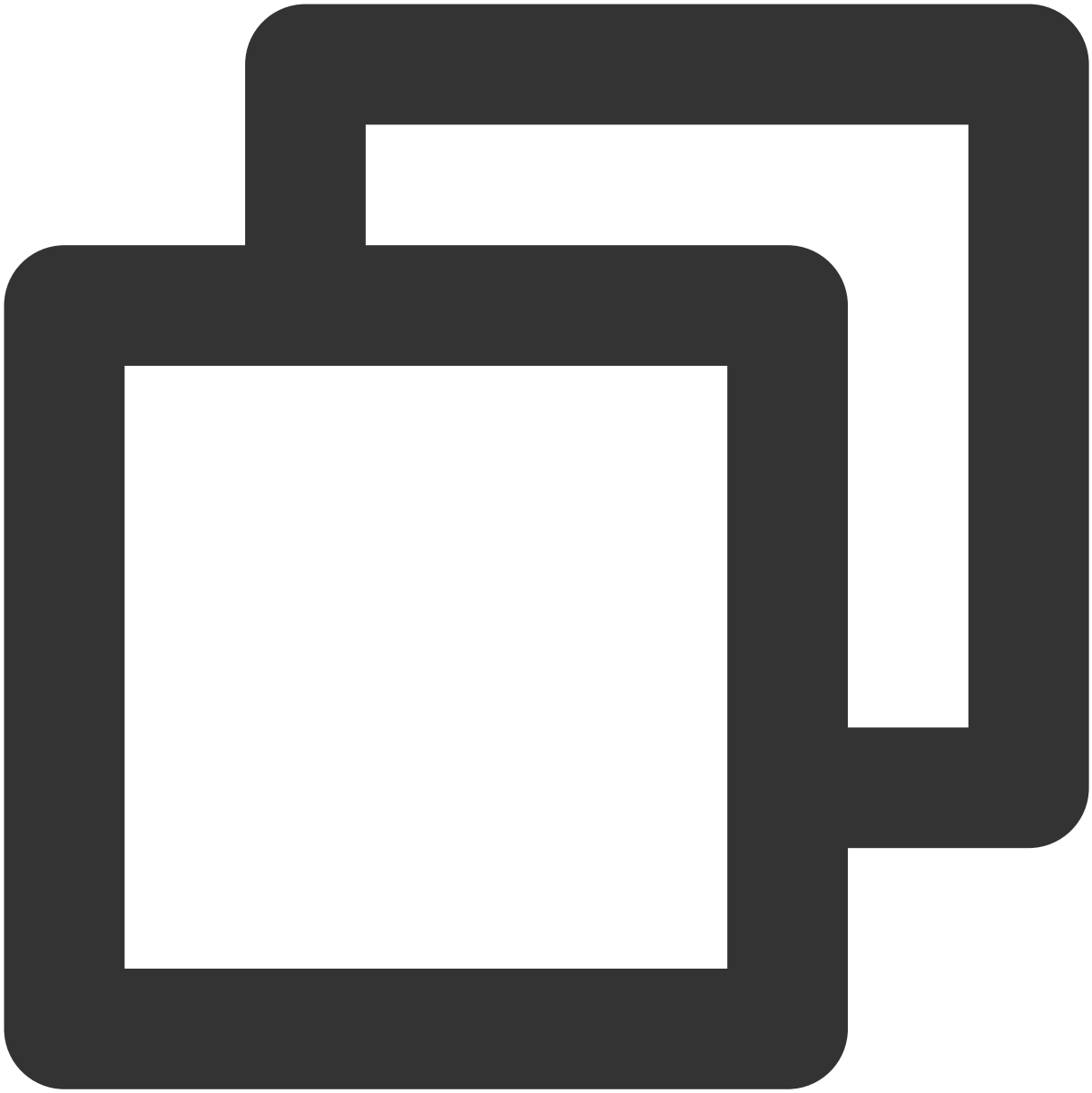
响应

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>
```

```
            <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：

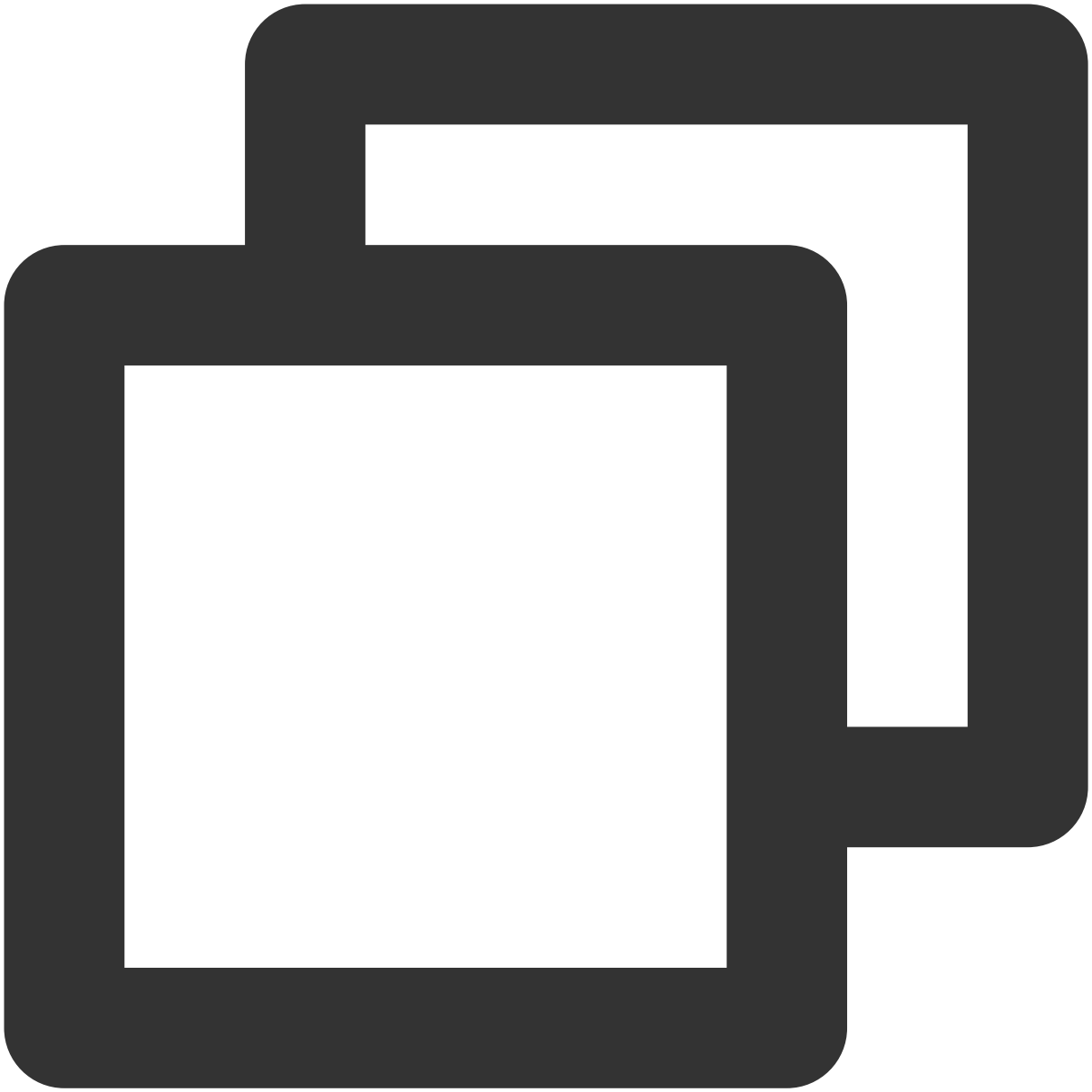
节点名称（关键字）	父节点	描述	类型
Response	无	同创建截图模板接口的 Response	Container

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



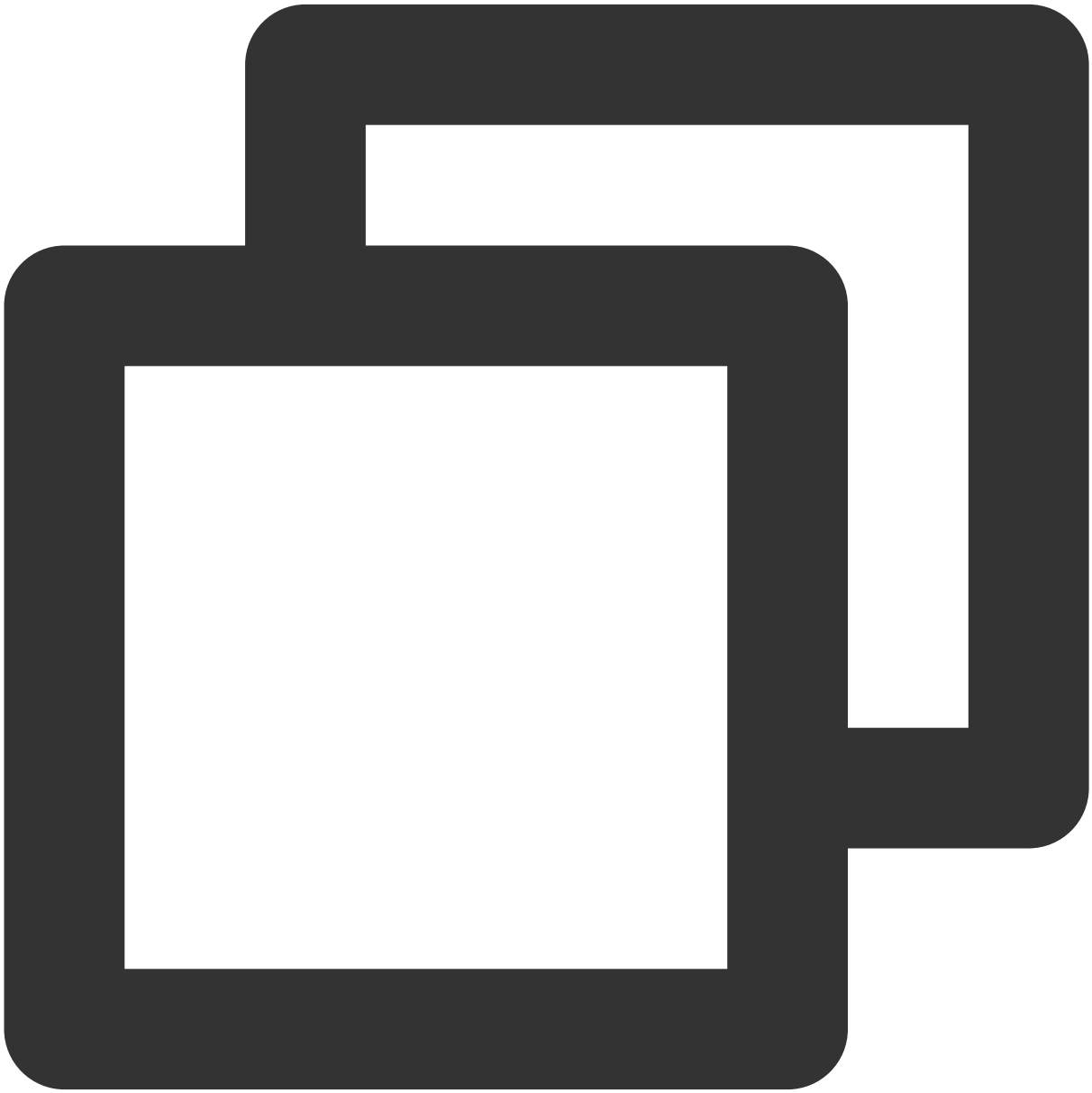
```
PUT /template/t1f16e1dfbdc994105b31292d45710642a HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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

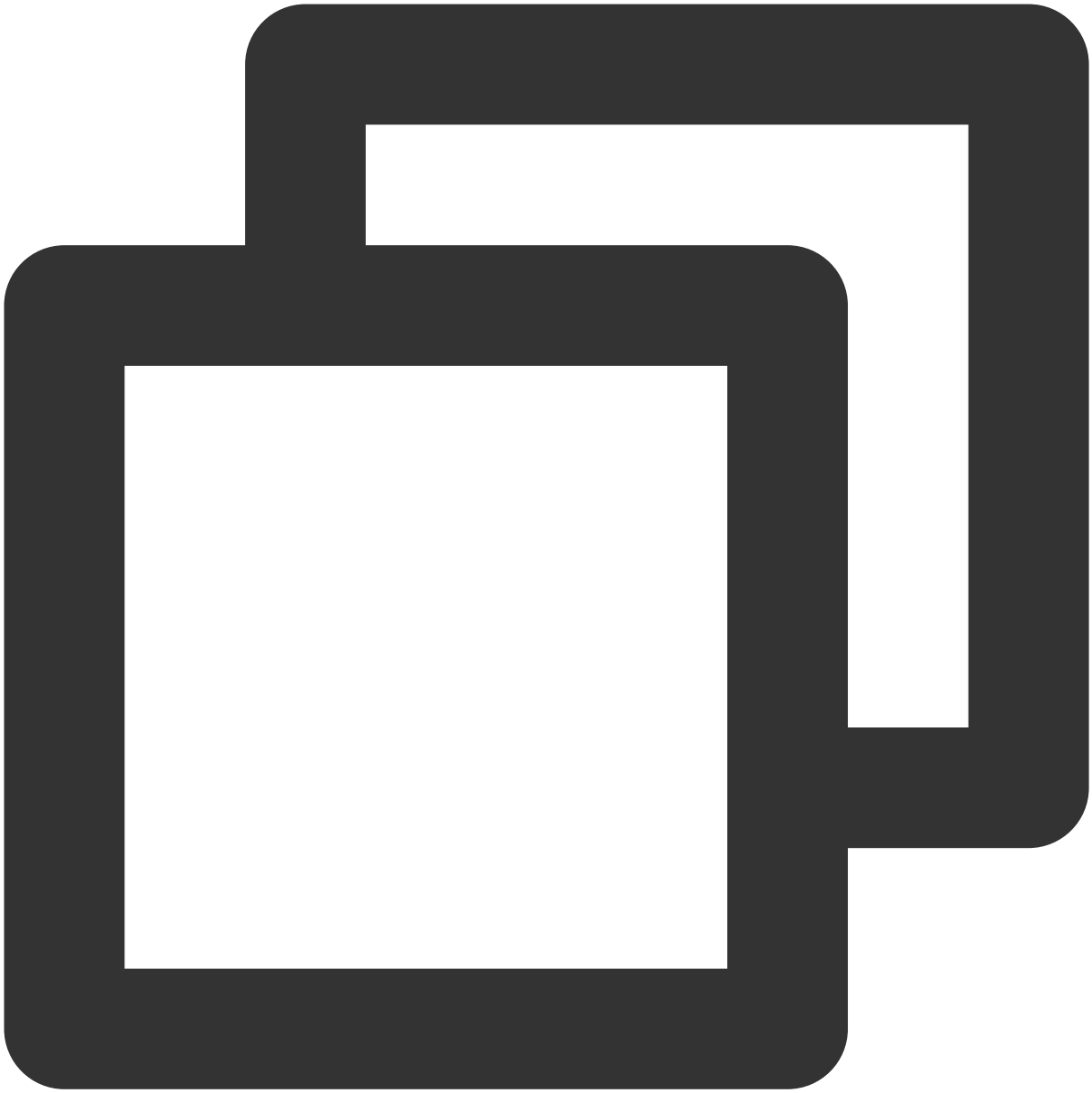
最近更新时间：2024-01-23 10:00:02

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：

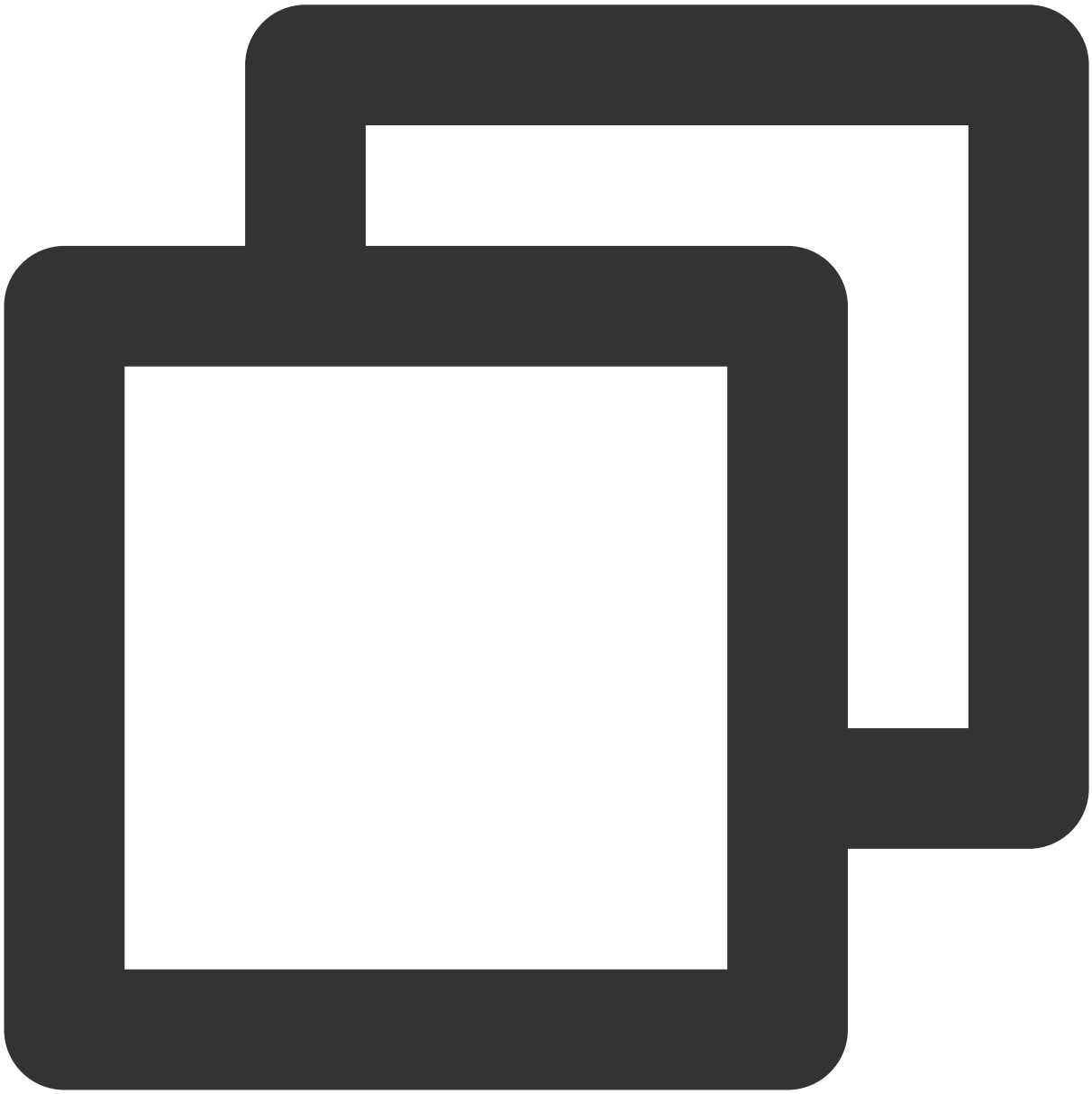
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

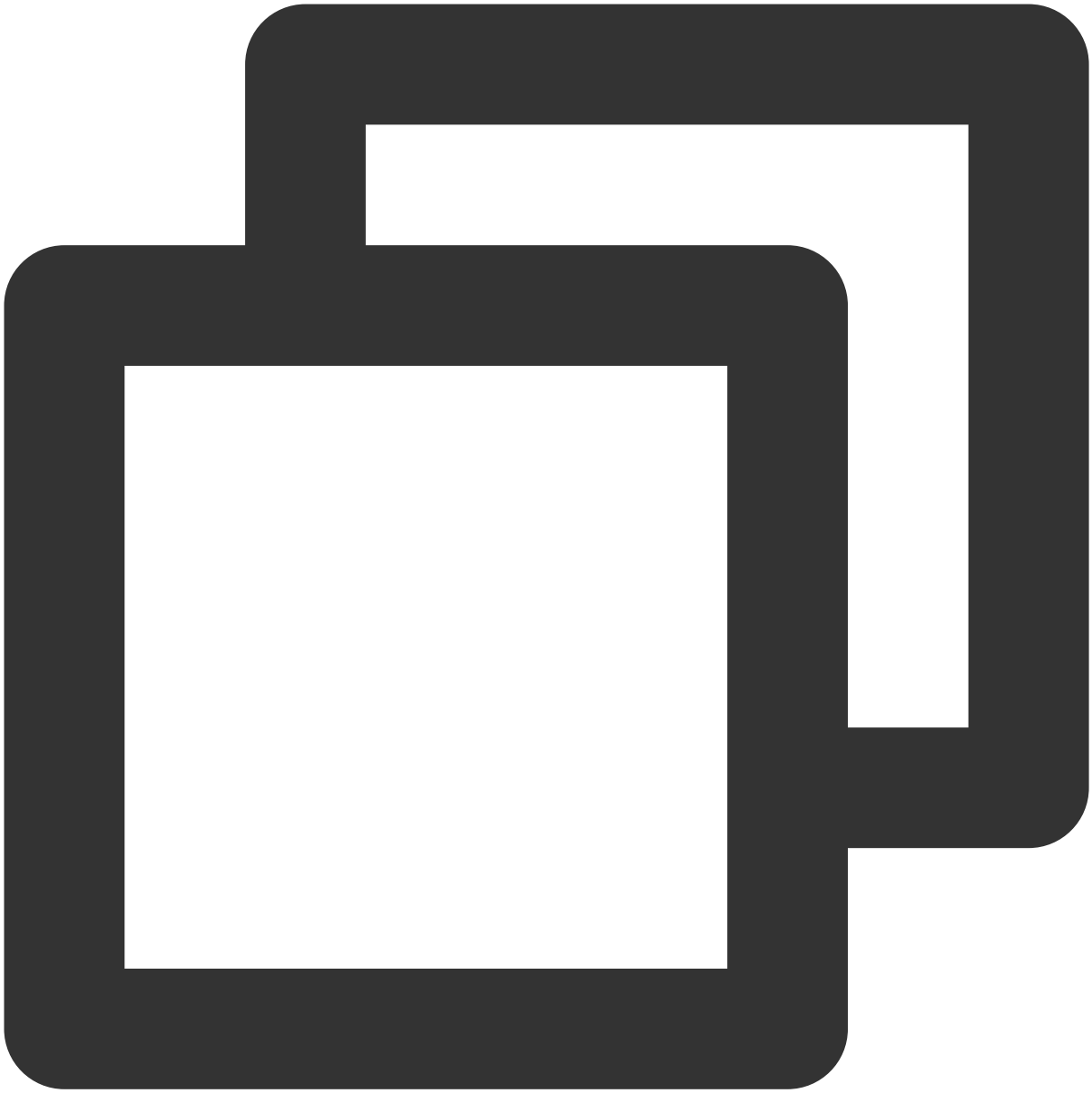
响应

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：

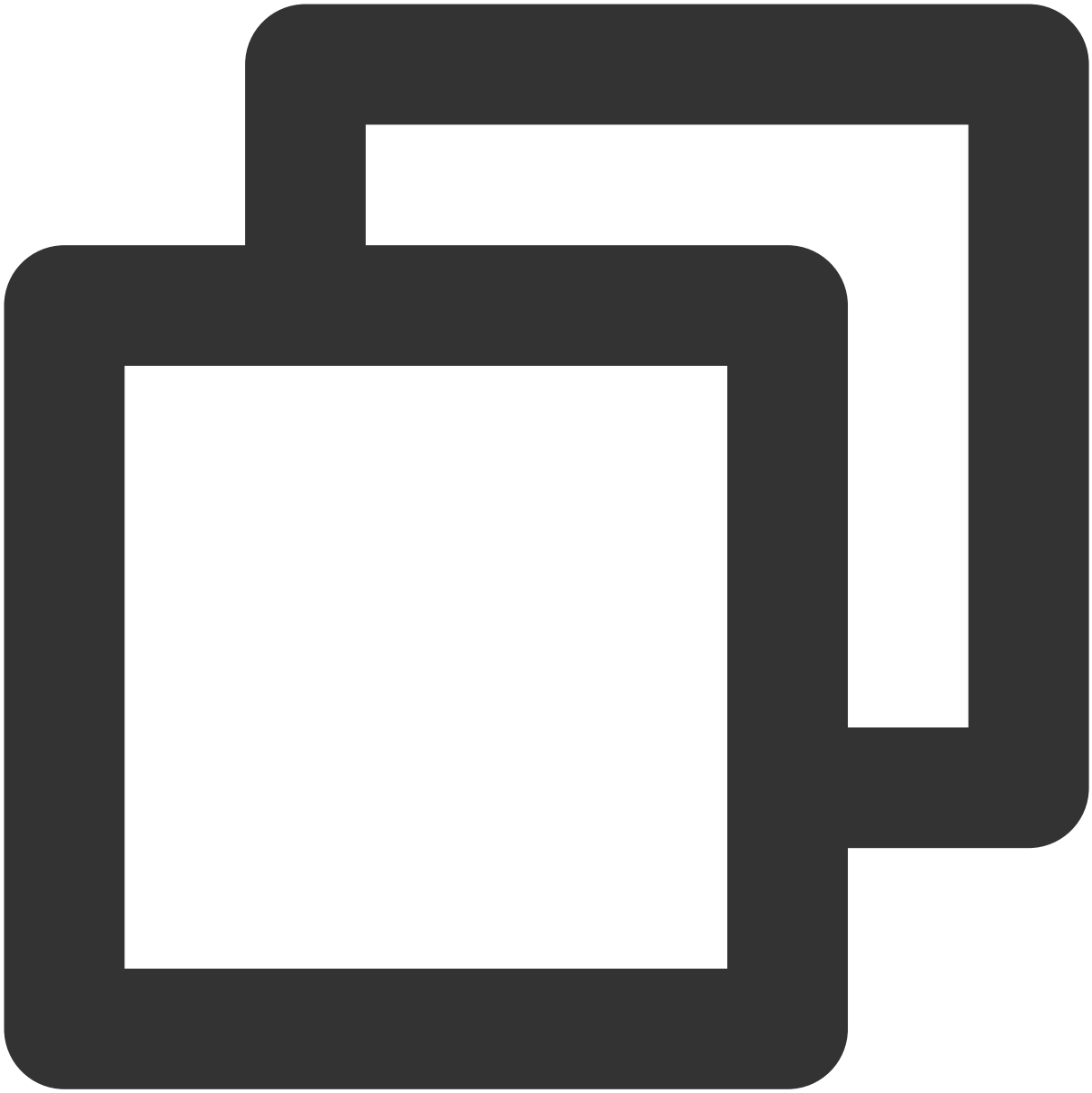
节点名称（关键字）	父节点	描述	类型
Response	无	同创建超分辨率模板接口的 Response	Container

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

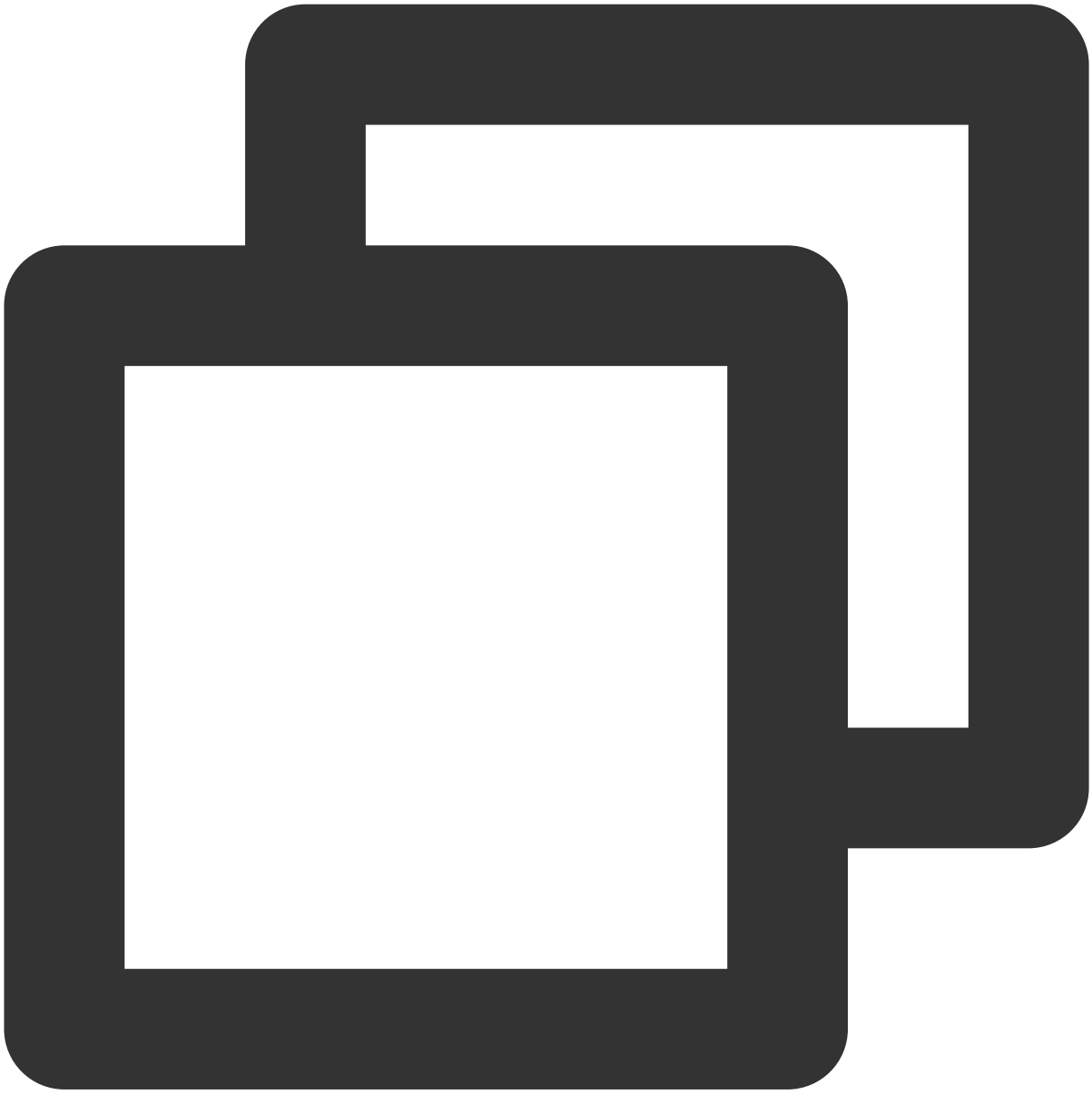


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

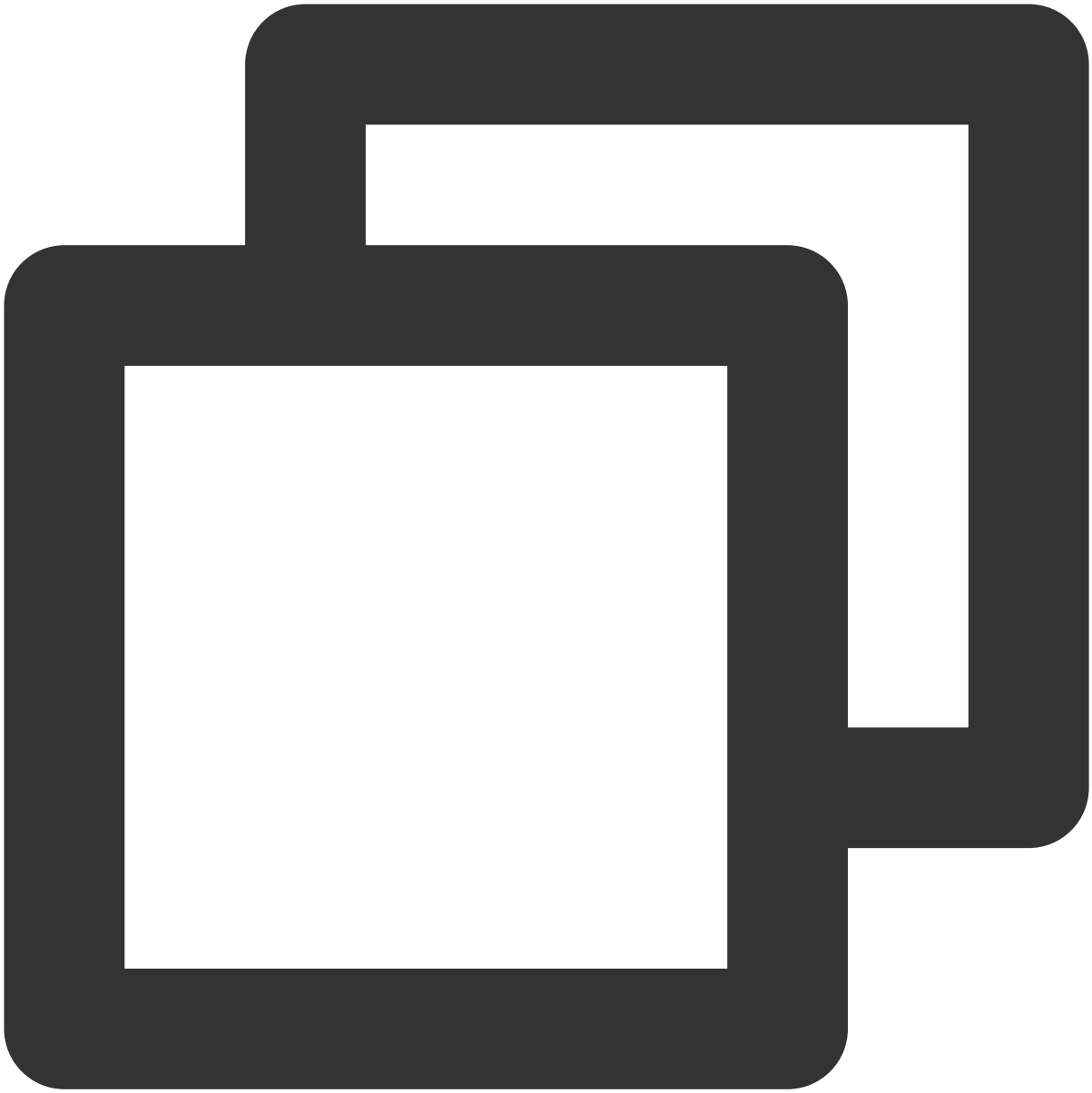
最近更新时间：2024-01-23 10:00:02

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：

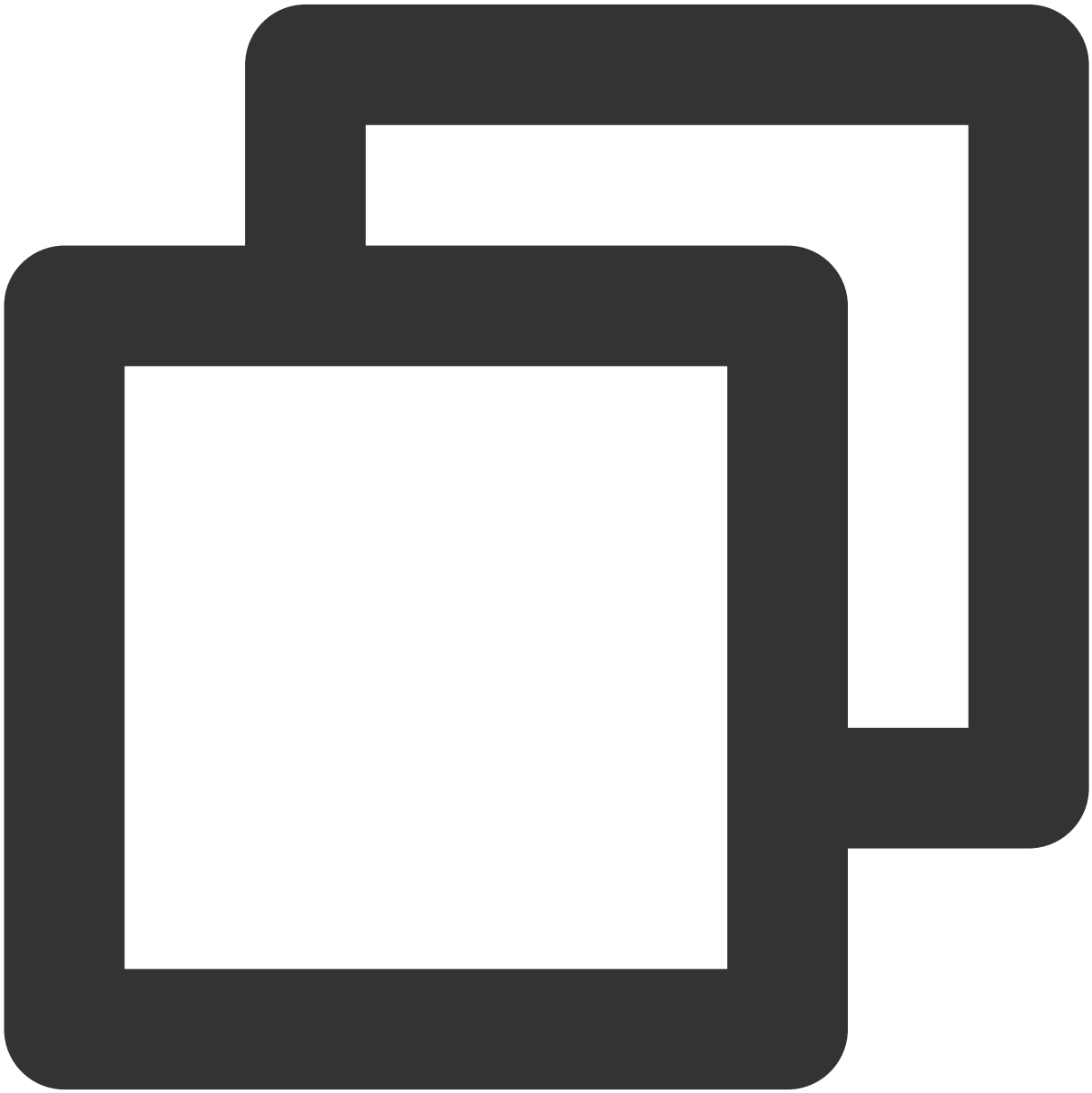
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

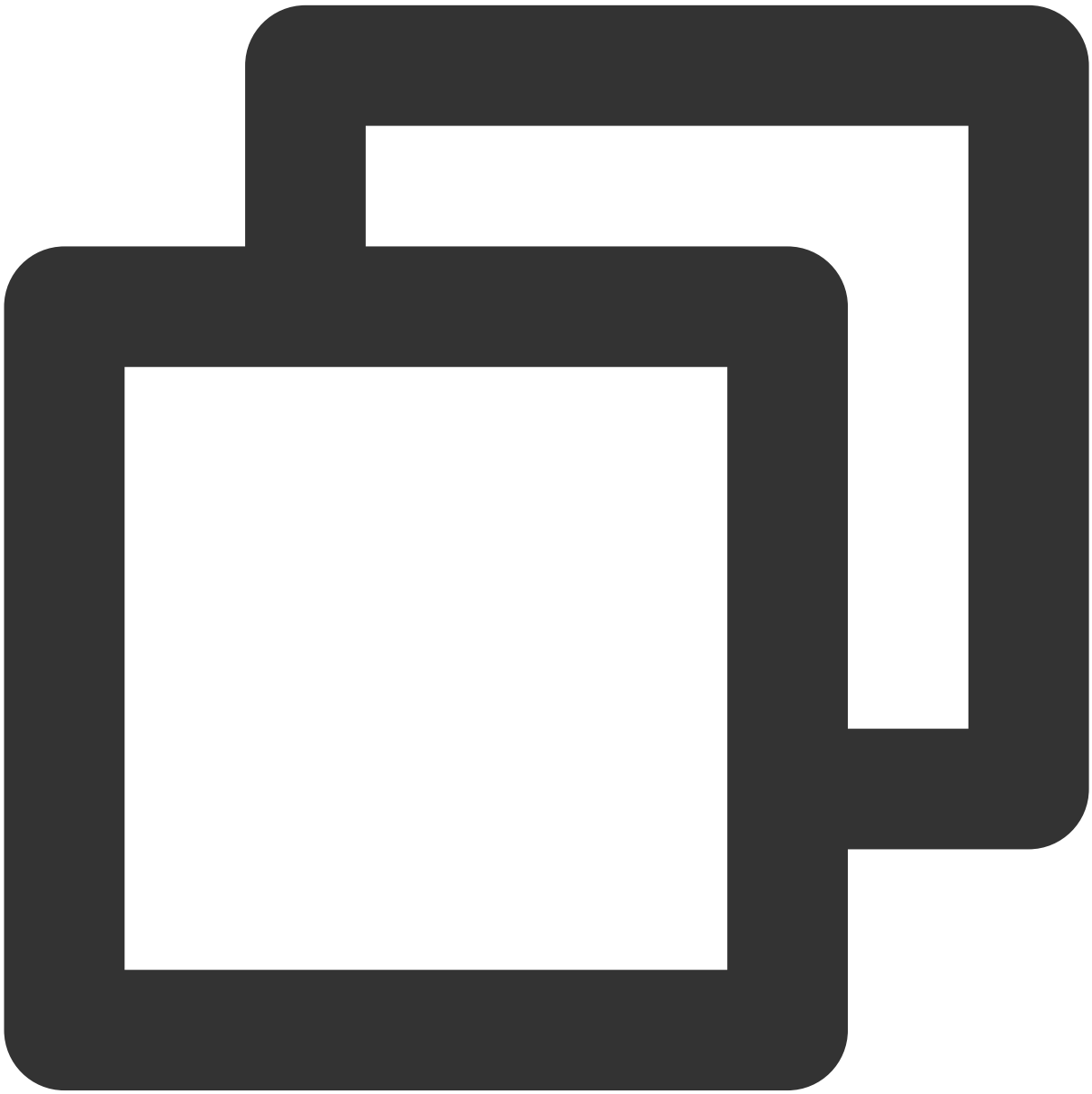
响应

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：

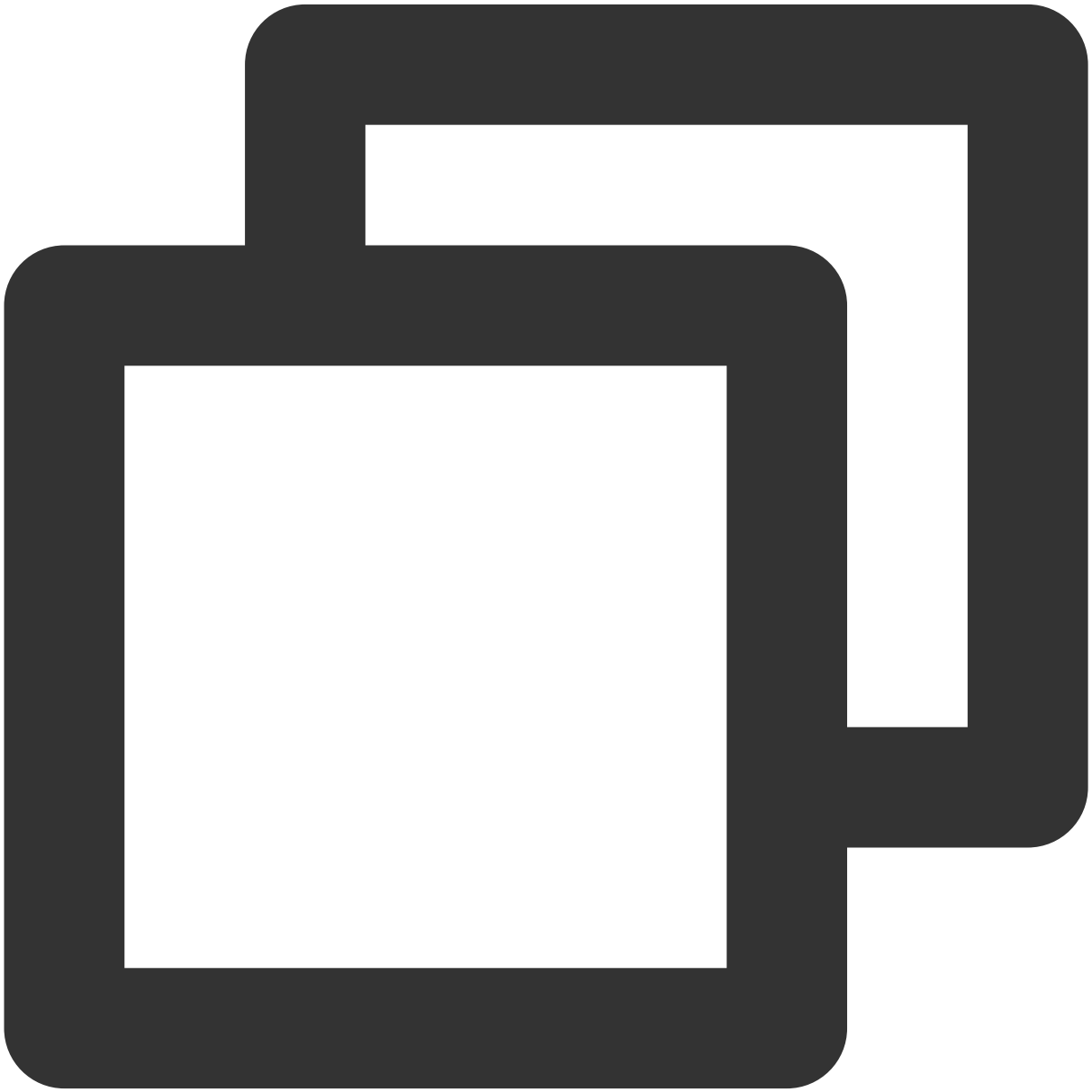
节点名称（关键字）	父节点	描述	类型
Response	无	同创建转码模板接口的 Response	Container

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

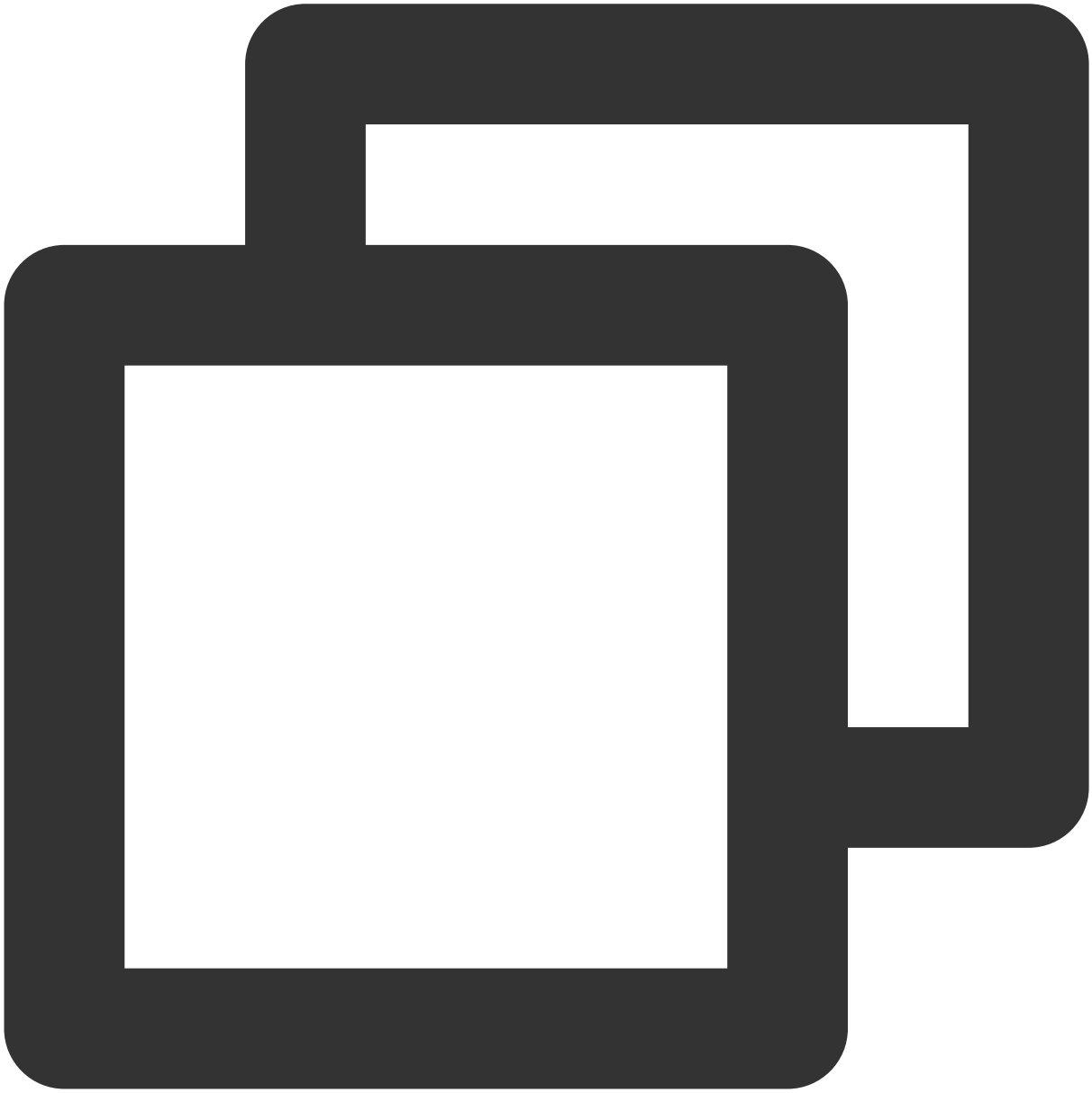


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

音视频转码 pro

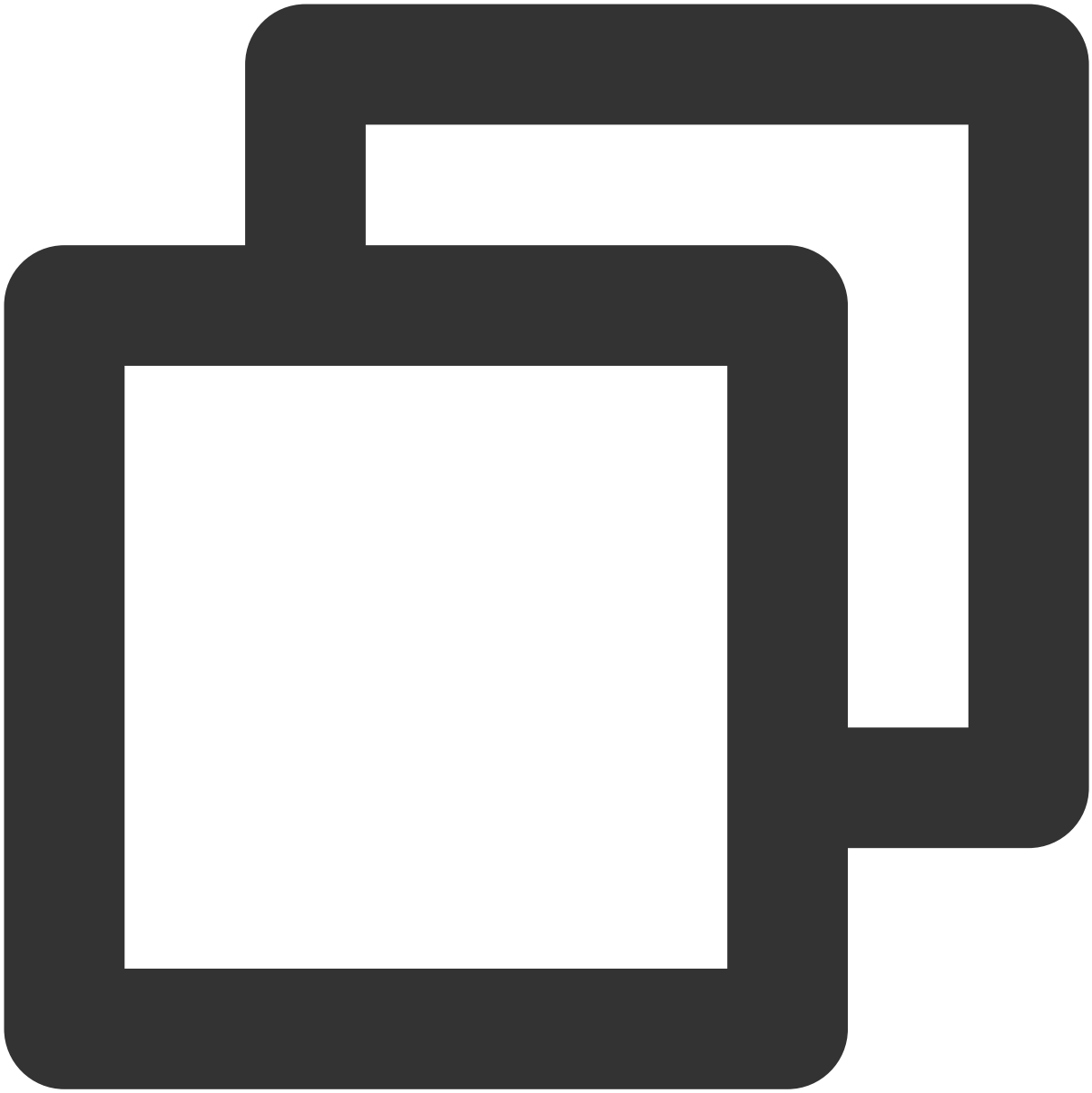
最近更新时间：2024-01-23 10:00:02

功能描述

更新音视频转码 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：

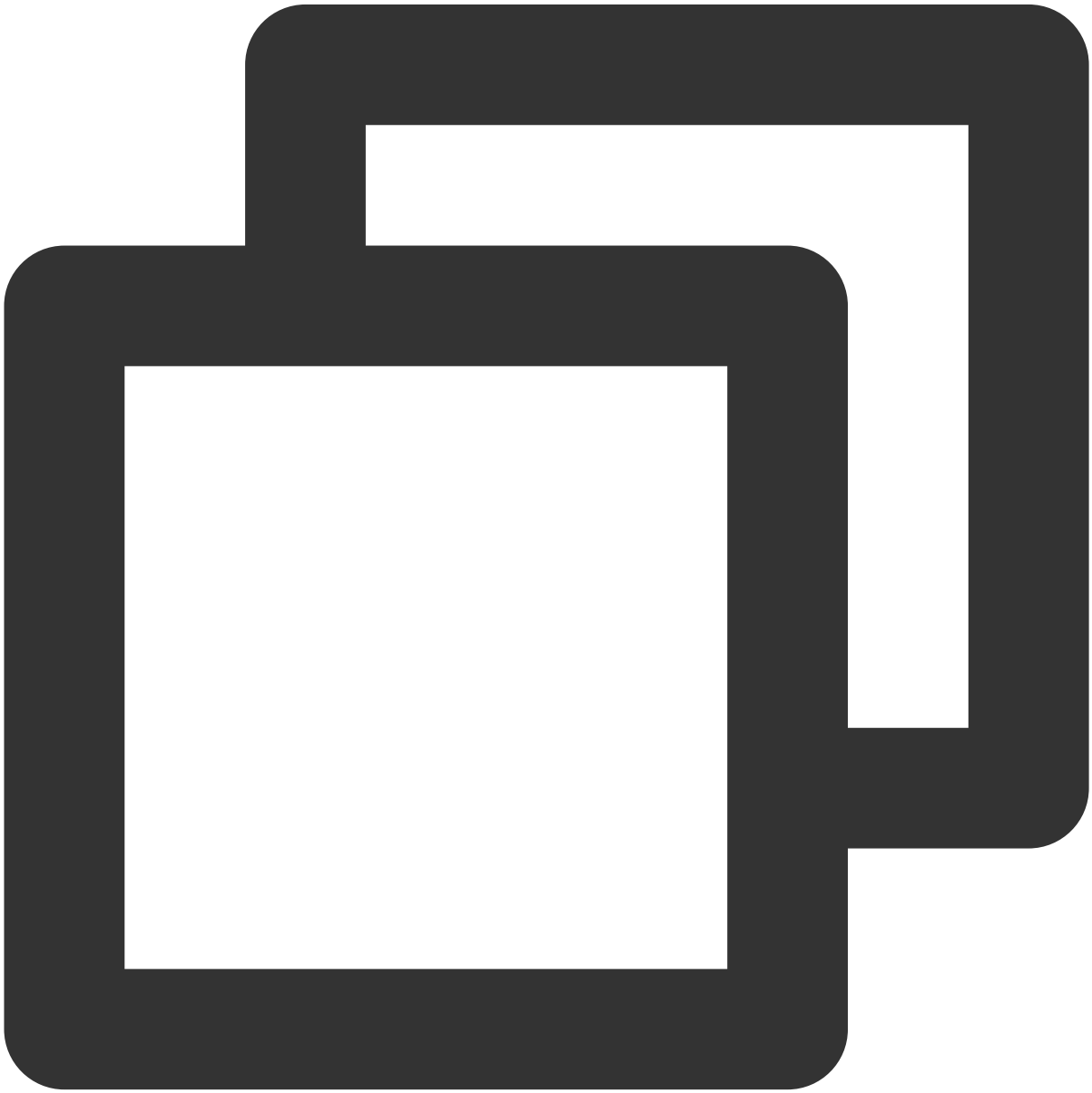
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

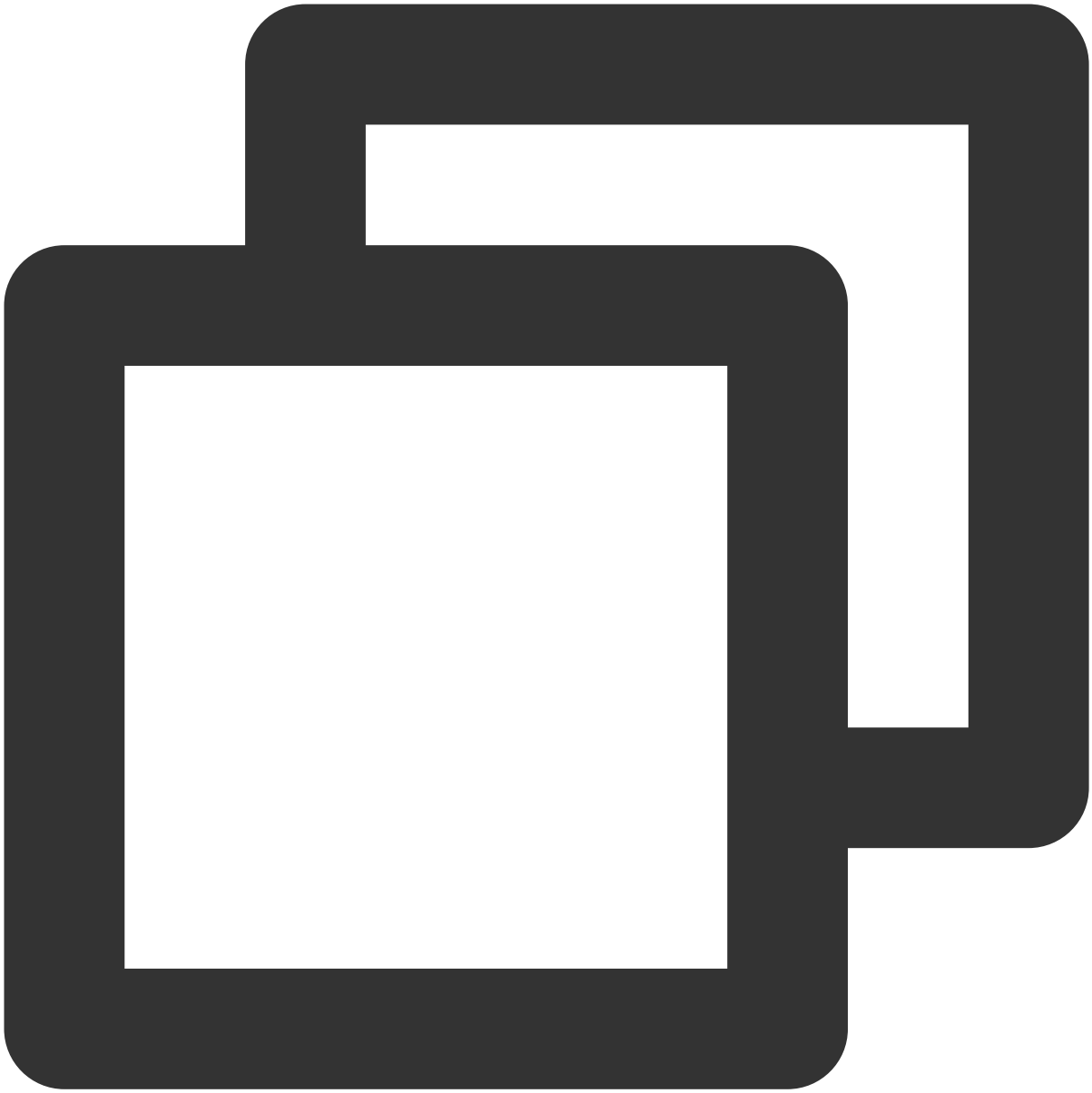
响应

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：

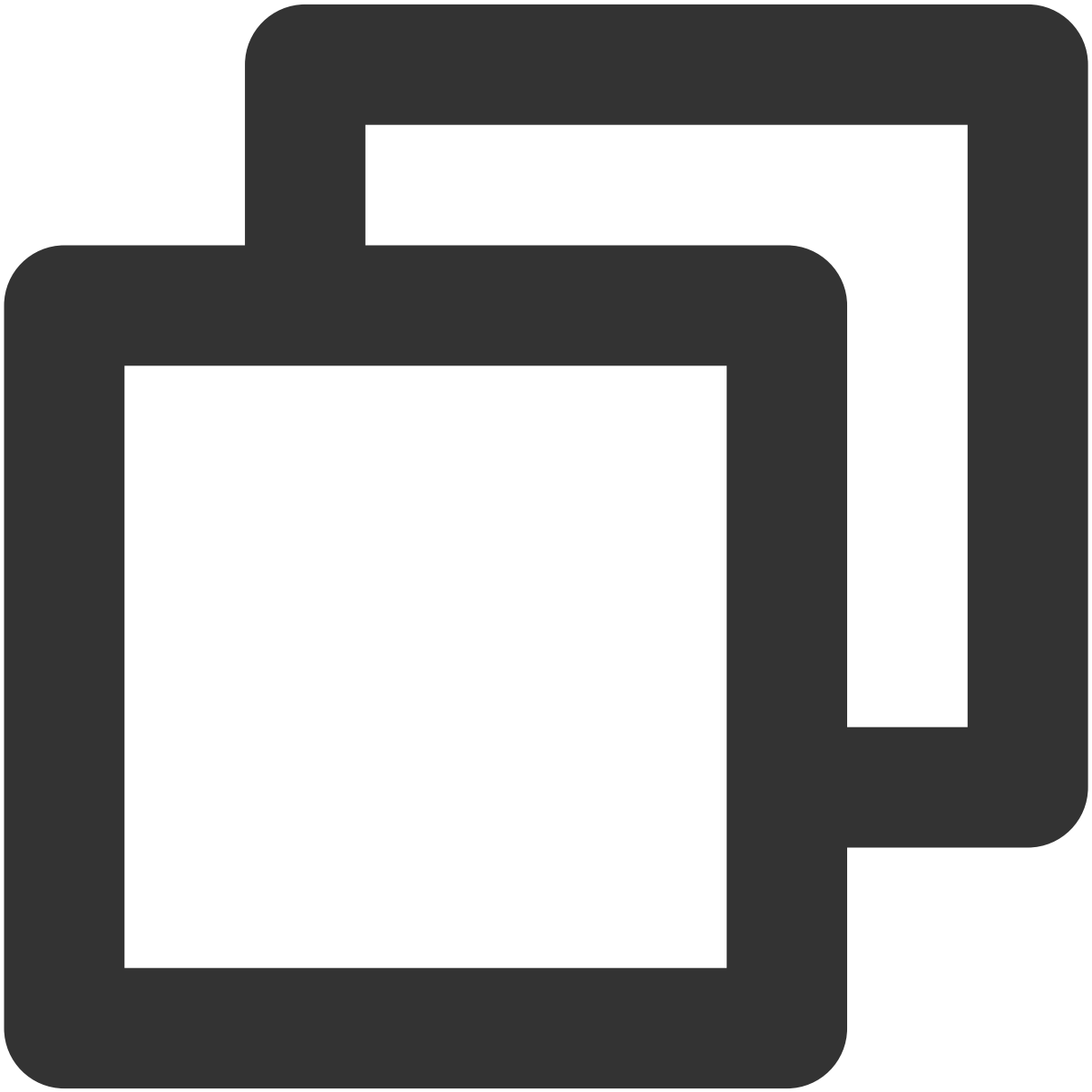
节点名称（关键字）	父节点	描述	类型
Response	无	同创建转码 Pro 模板接口的 Response	Container

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

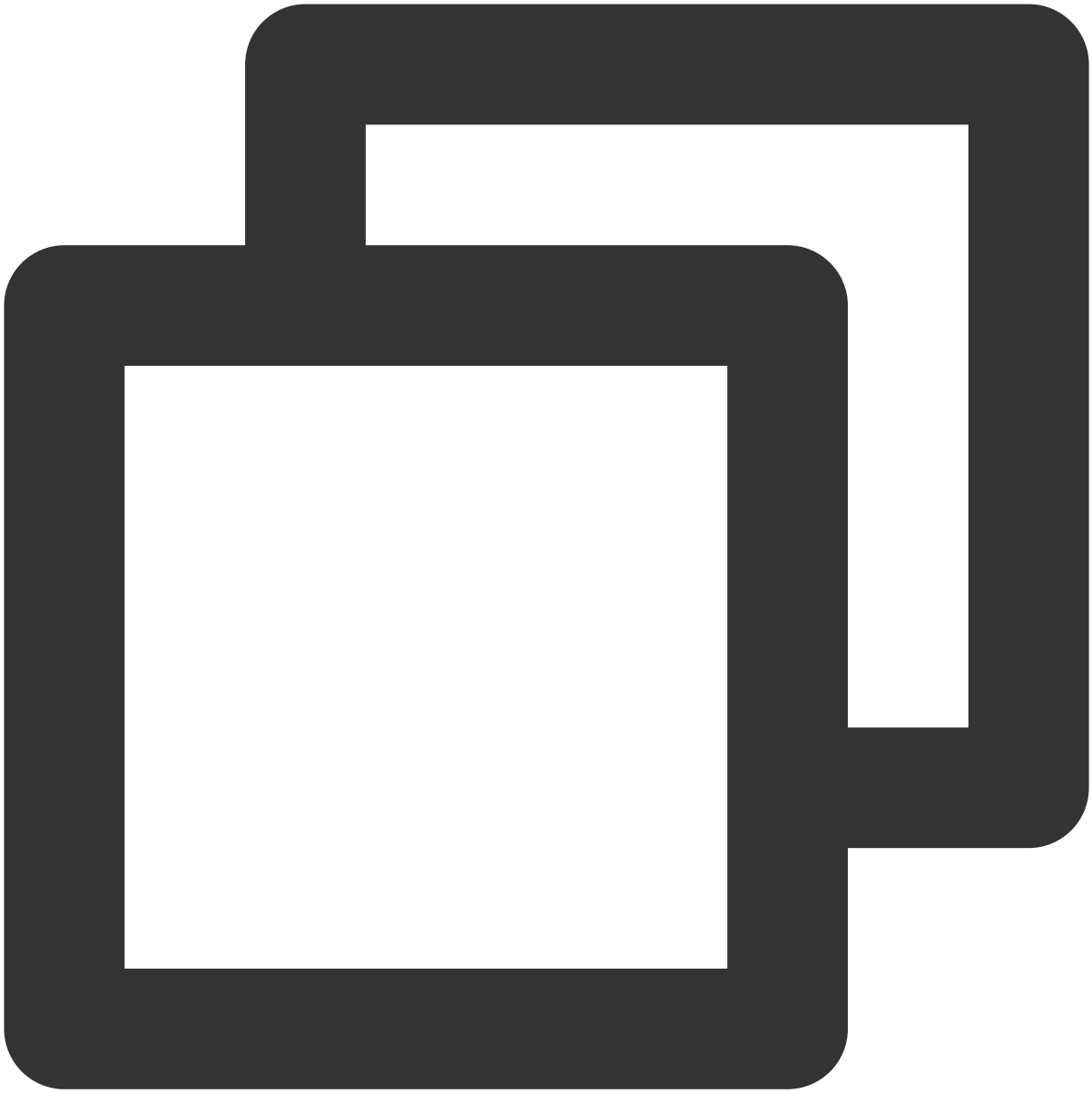


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

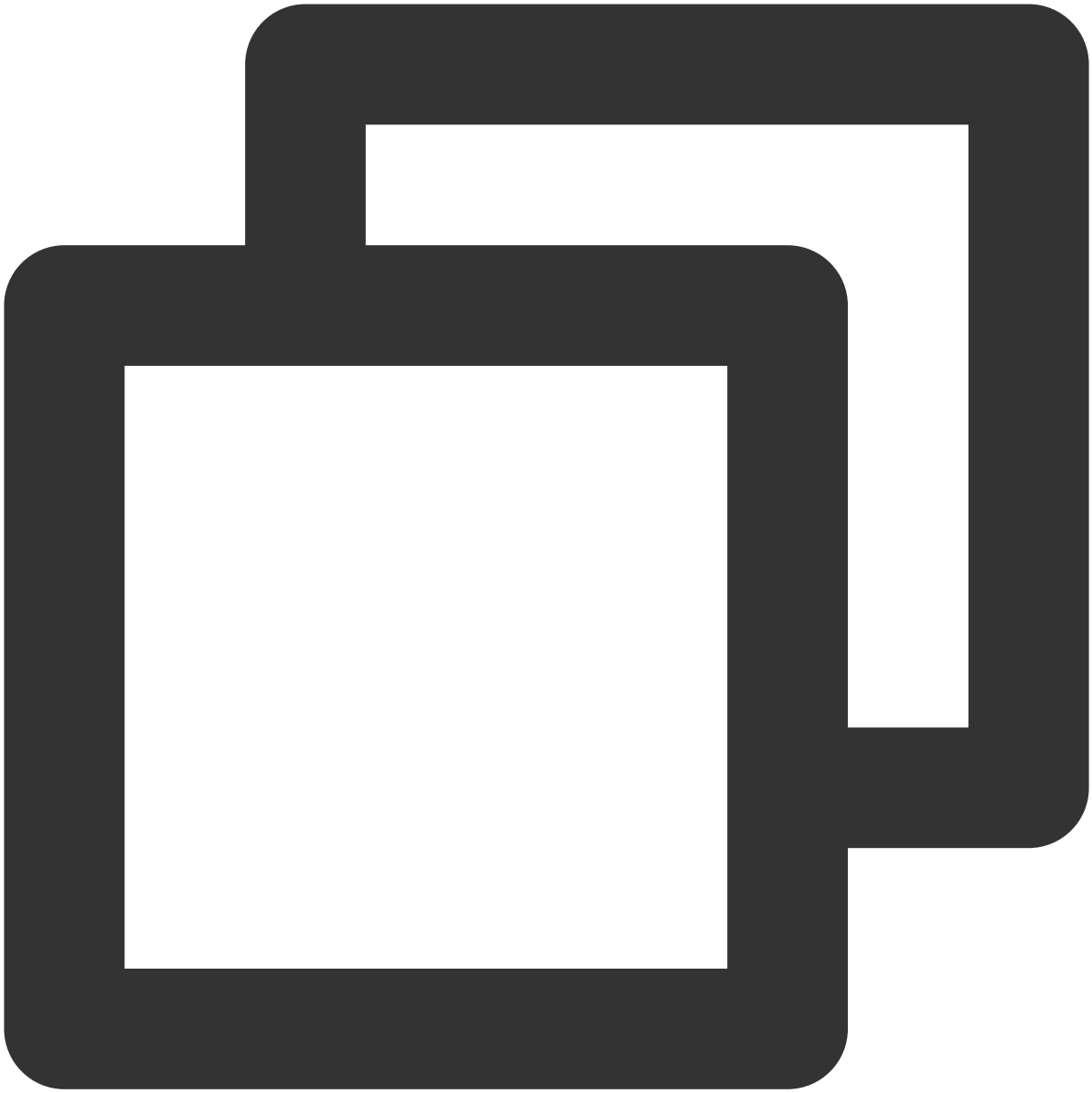
最近更新时间：2024-01-23 10:00:02

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：

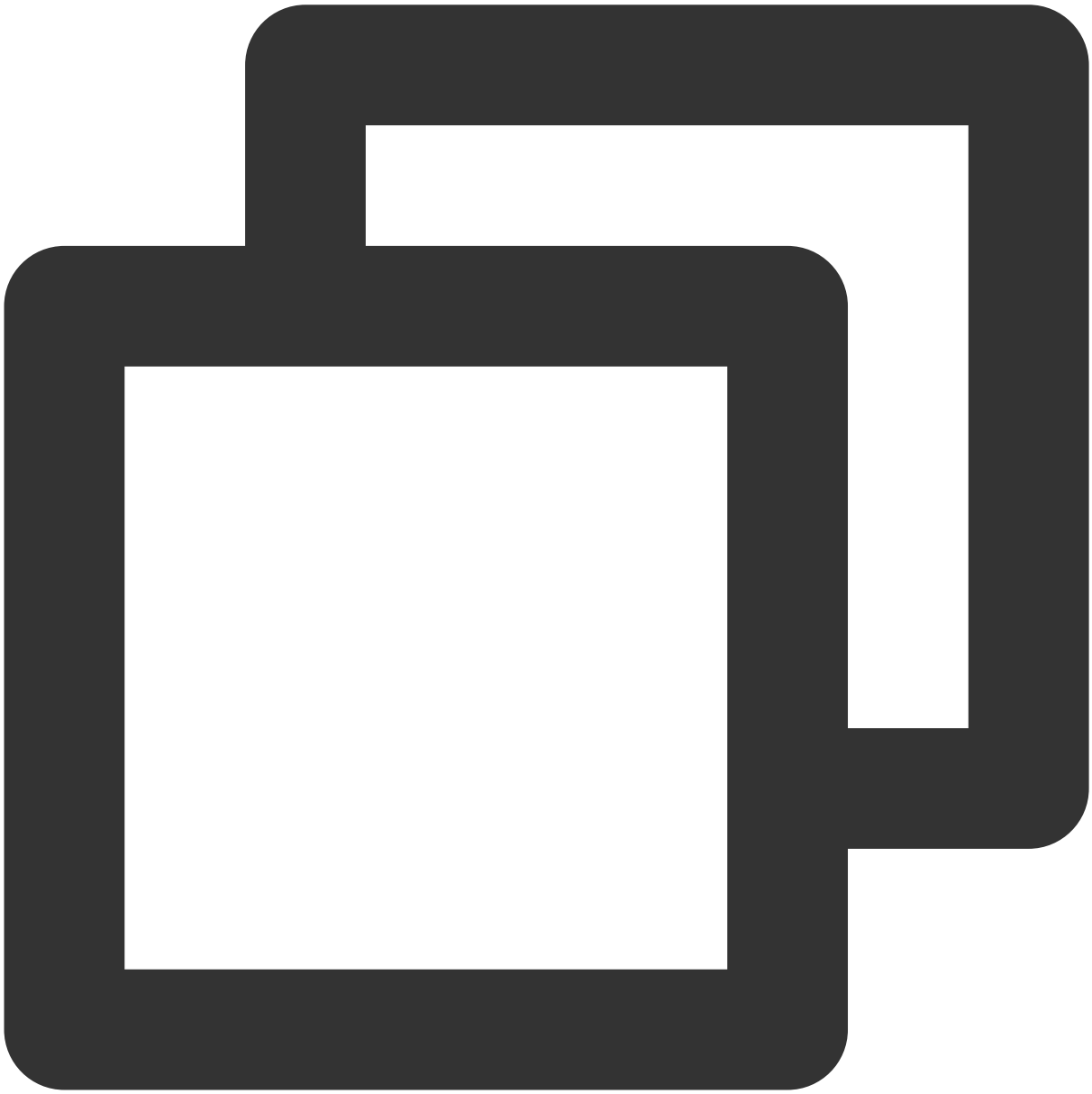
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

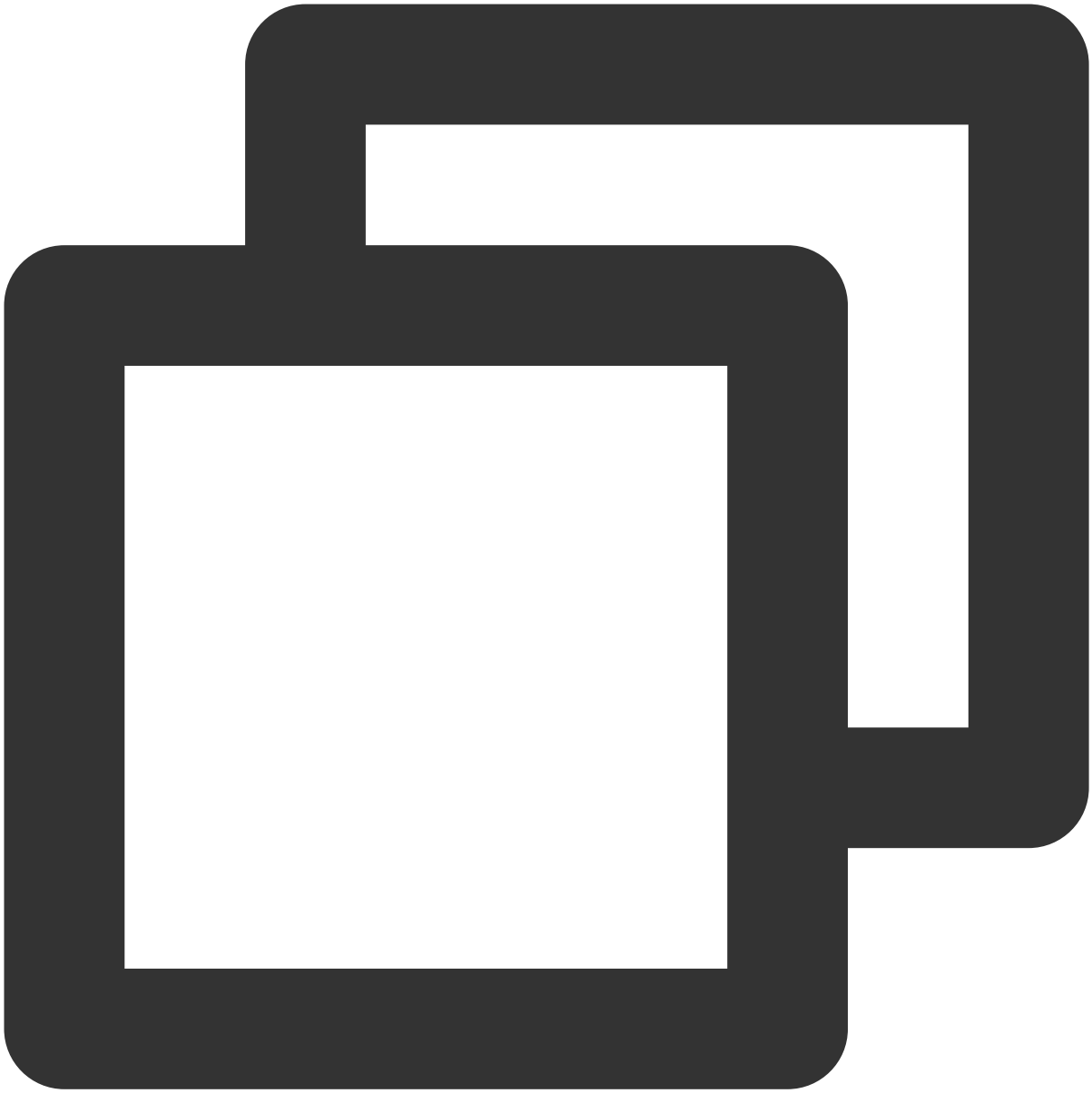
响应

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </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：

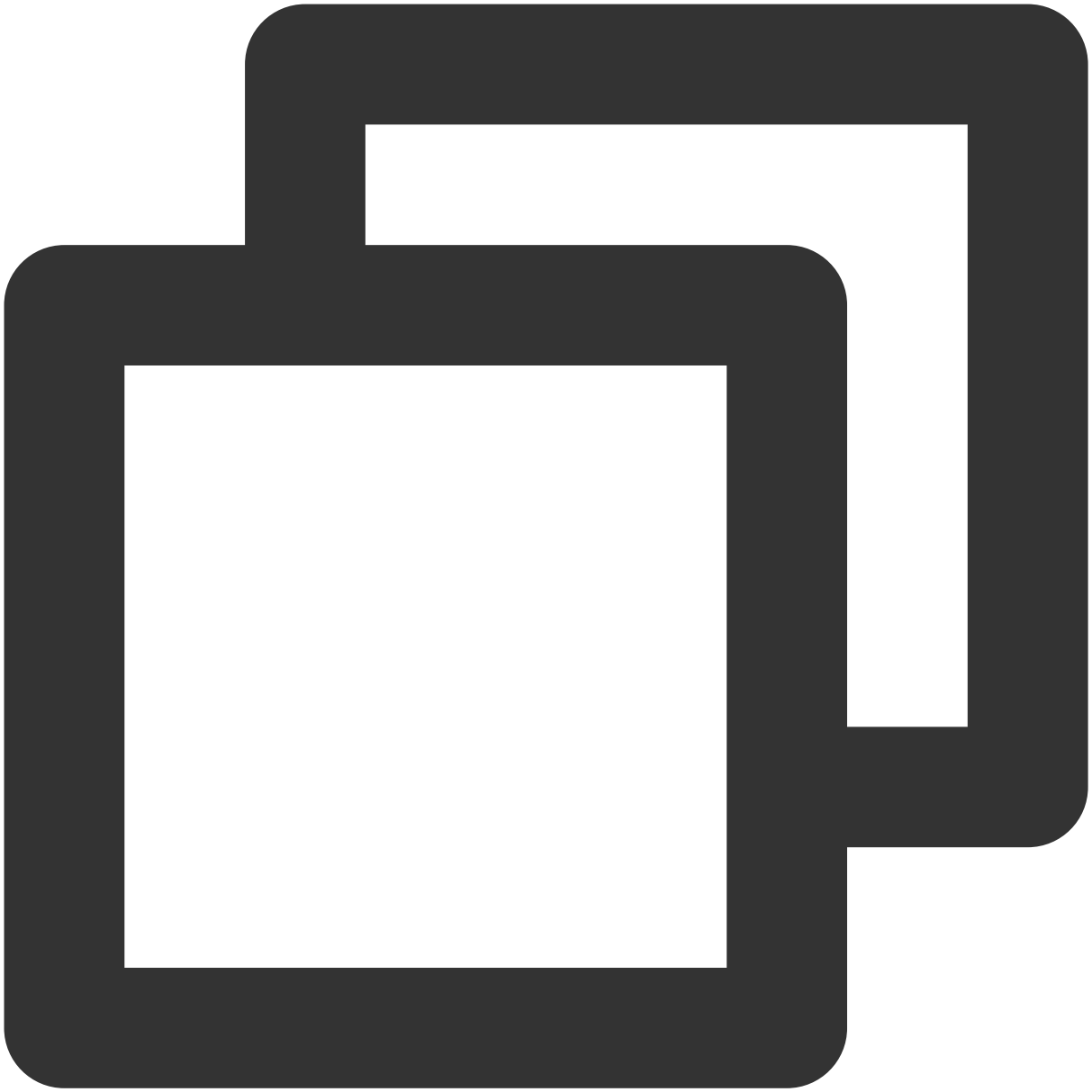
节点名称（关键字）	父节点	描述	类型
Response	无	同创建语音合成模板接口的 Response	Container

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

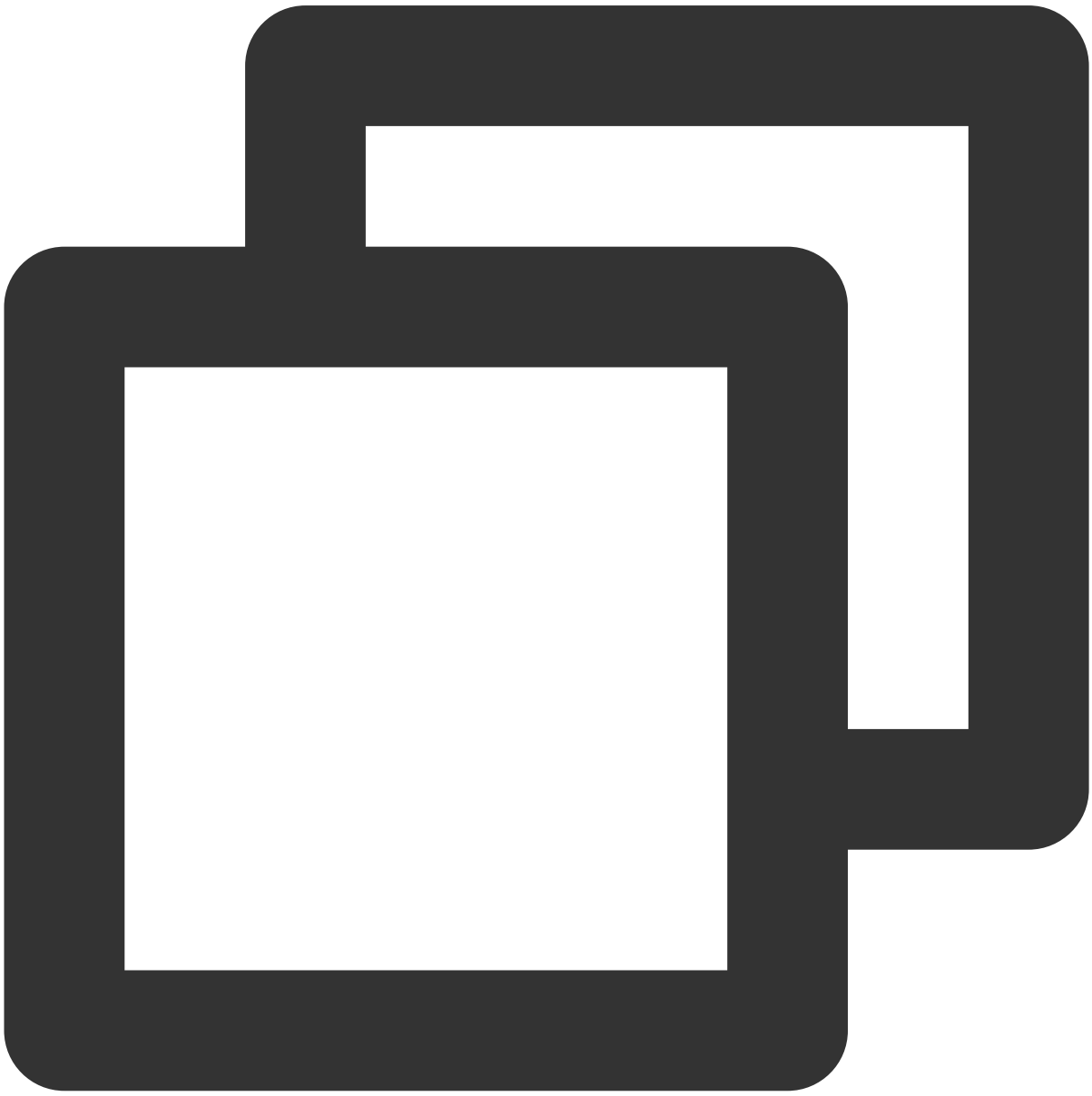


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

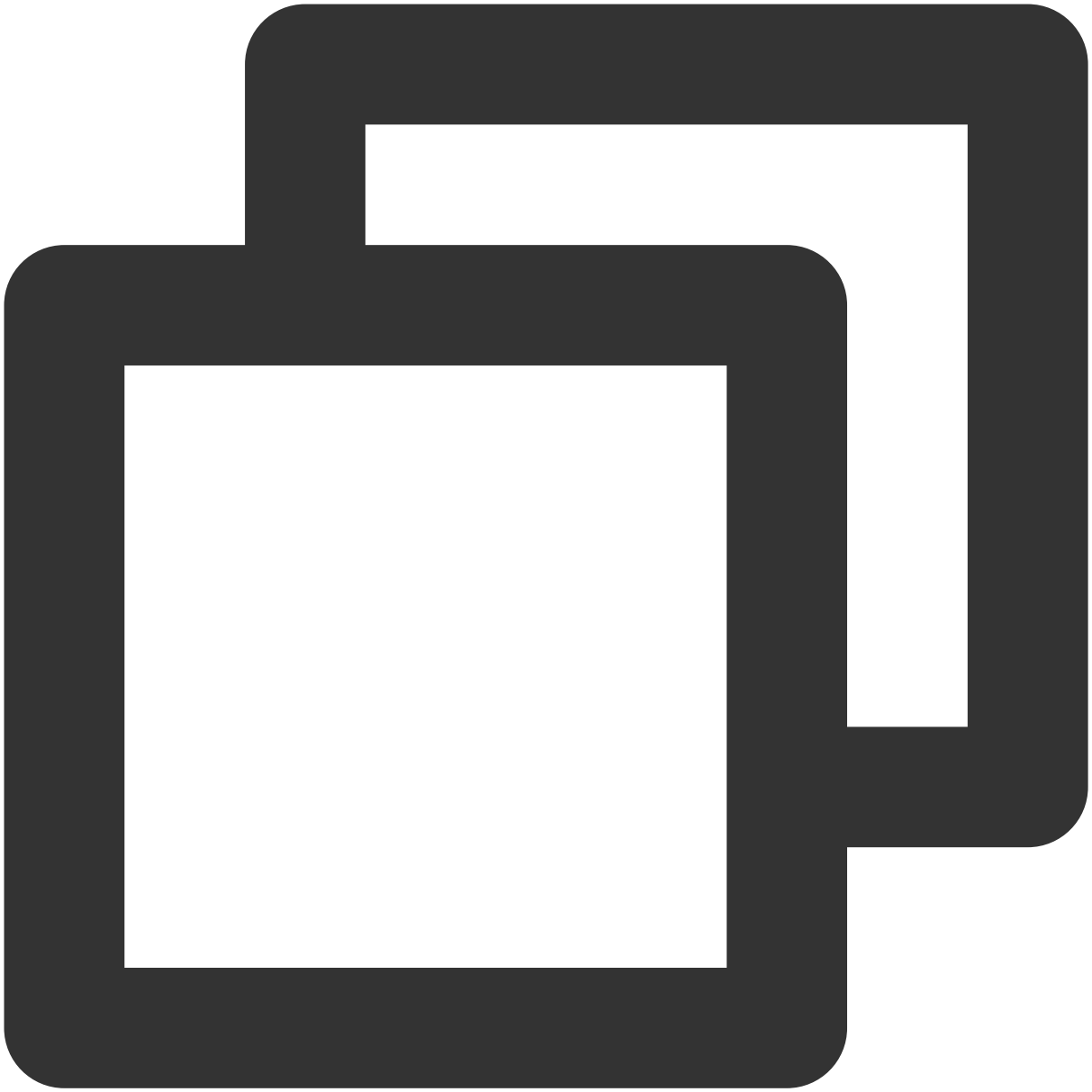
最近更新时间：2024-01-23 10:00:02

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：

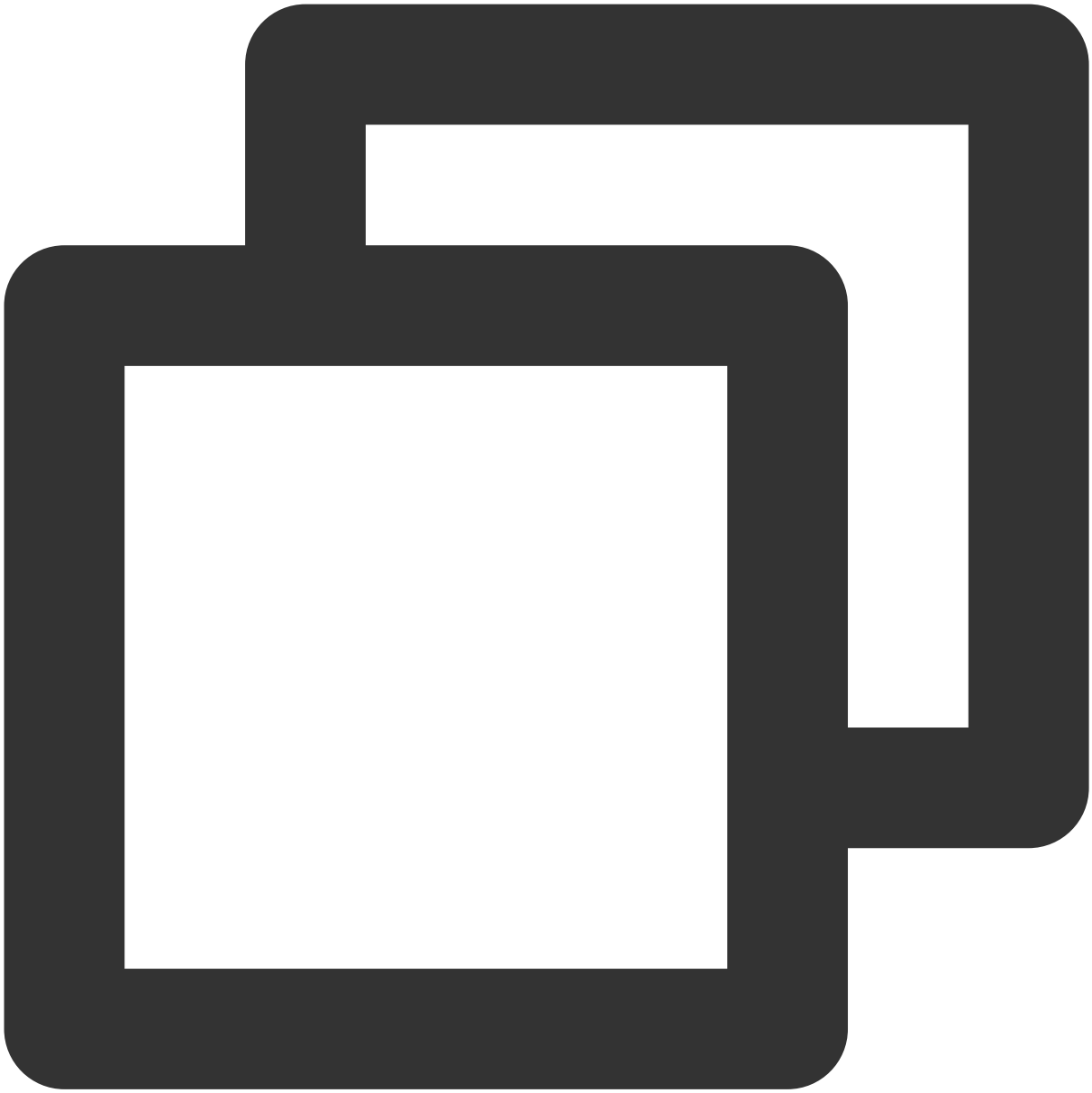
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

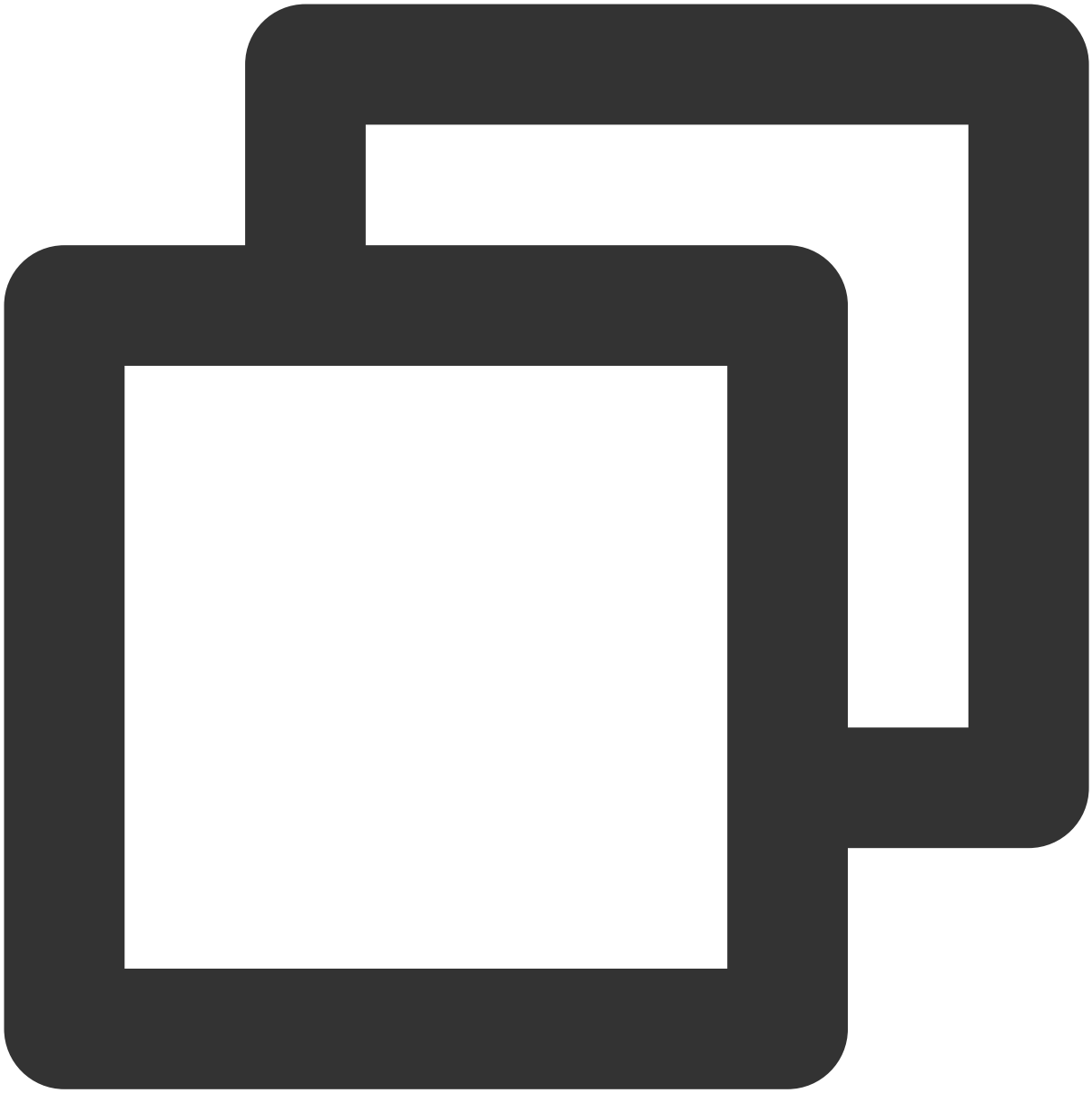
响应

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：

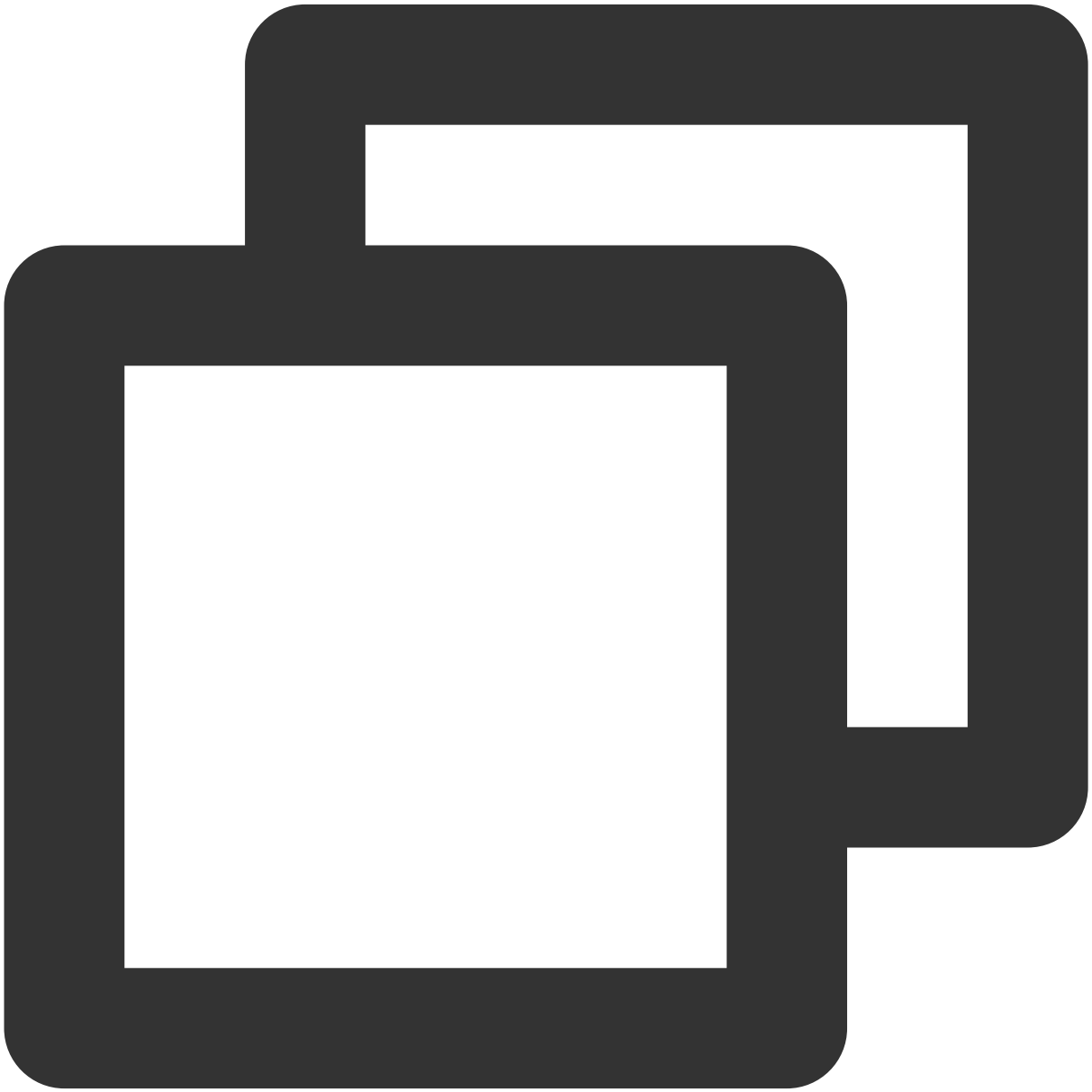
节点名称（关键字）	父节点	描述	类型
Response	无	同创建精彩集锦模板接口的 Response	Container

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

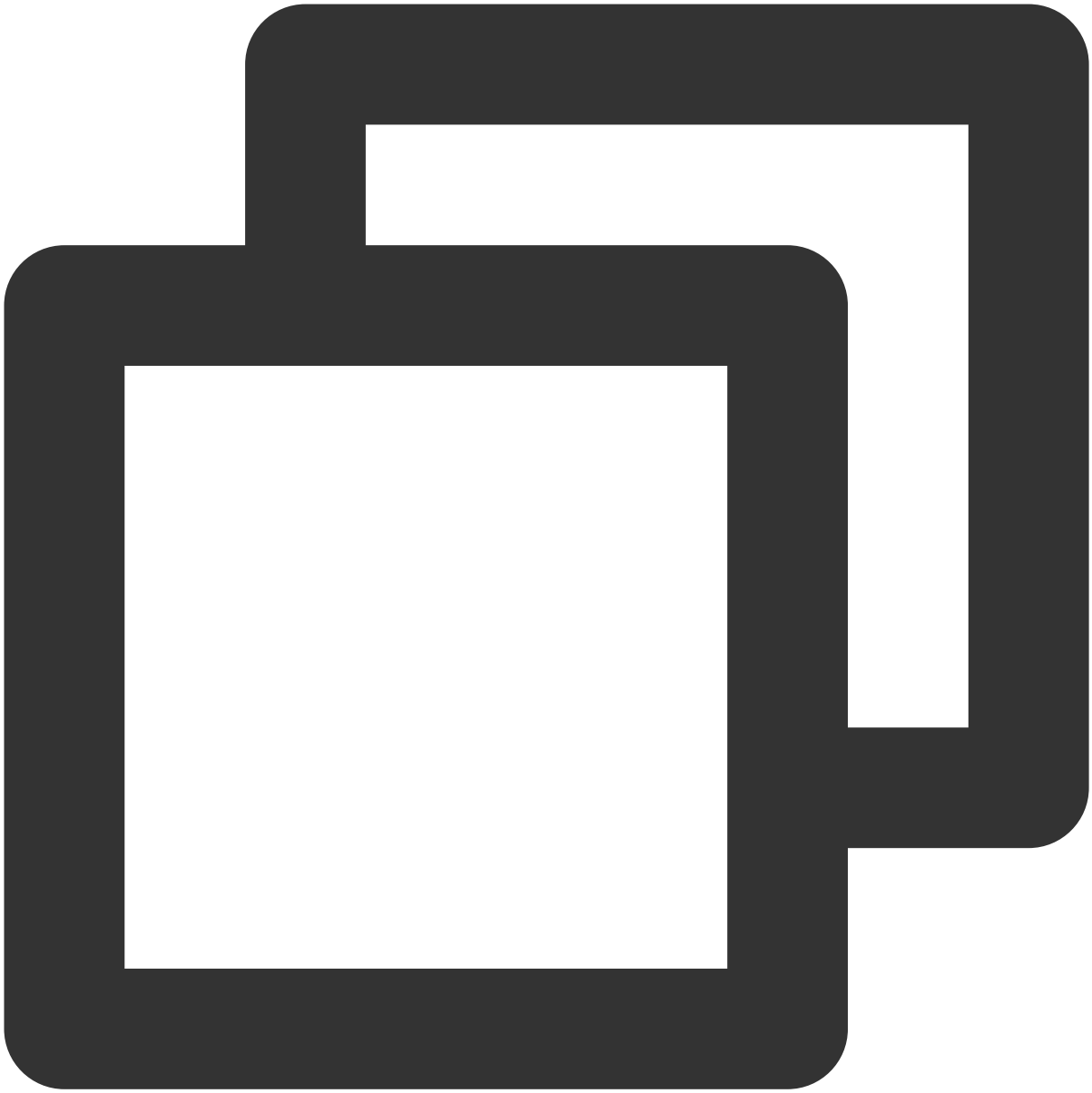


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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>
```

```
        <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

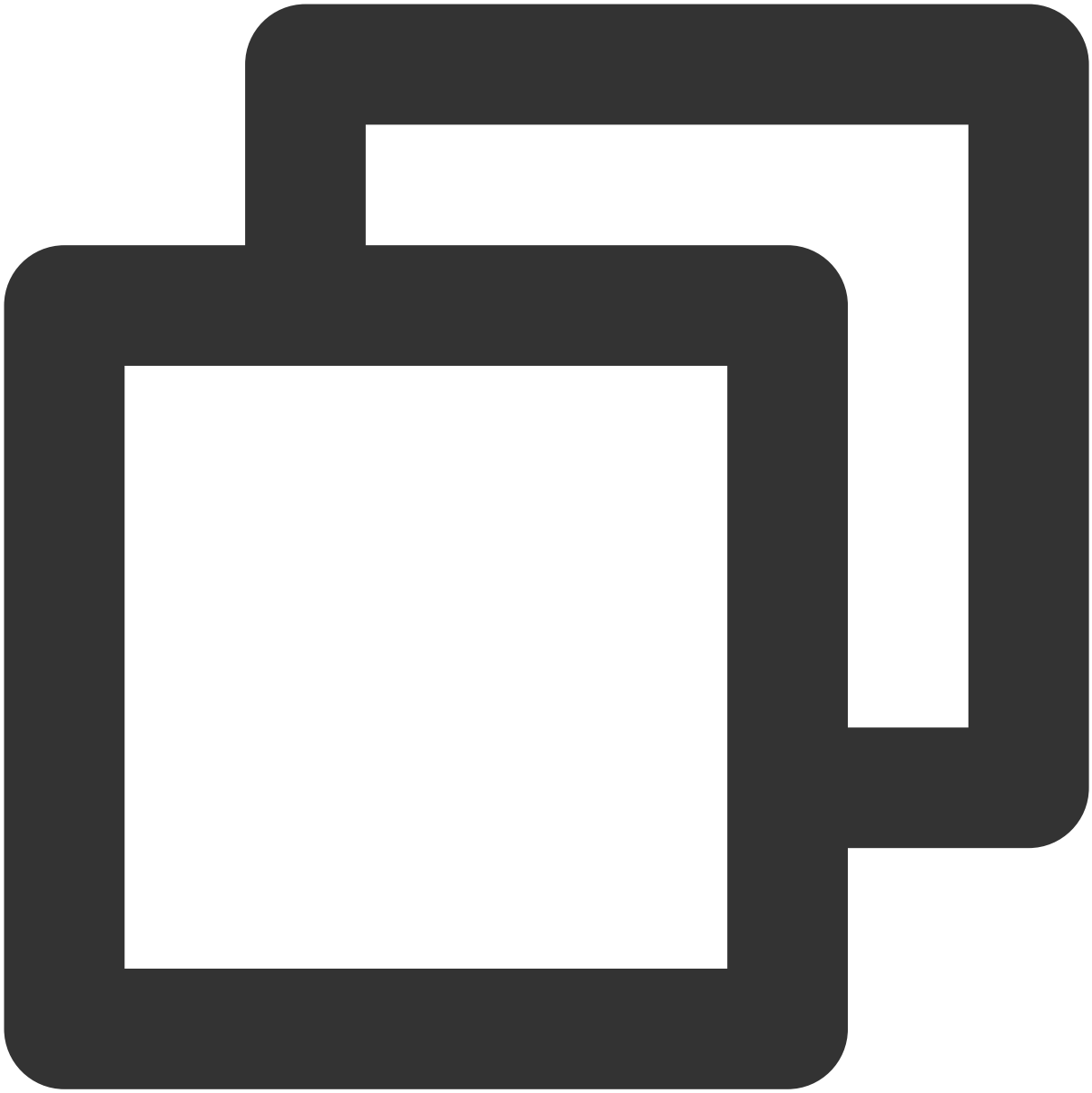
最近更新时间：2024-01-23 10:00:02

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：

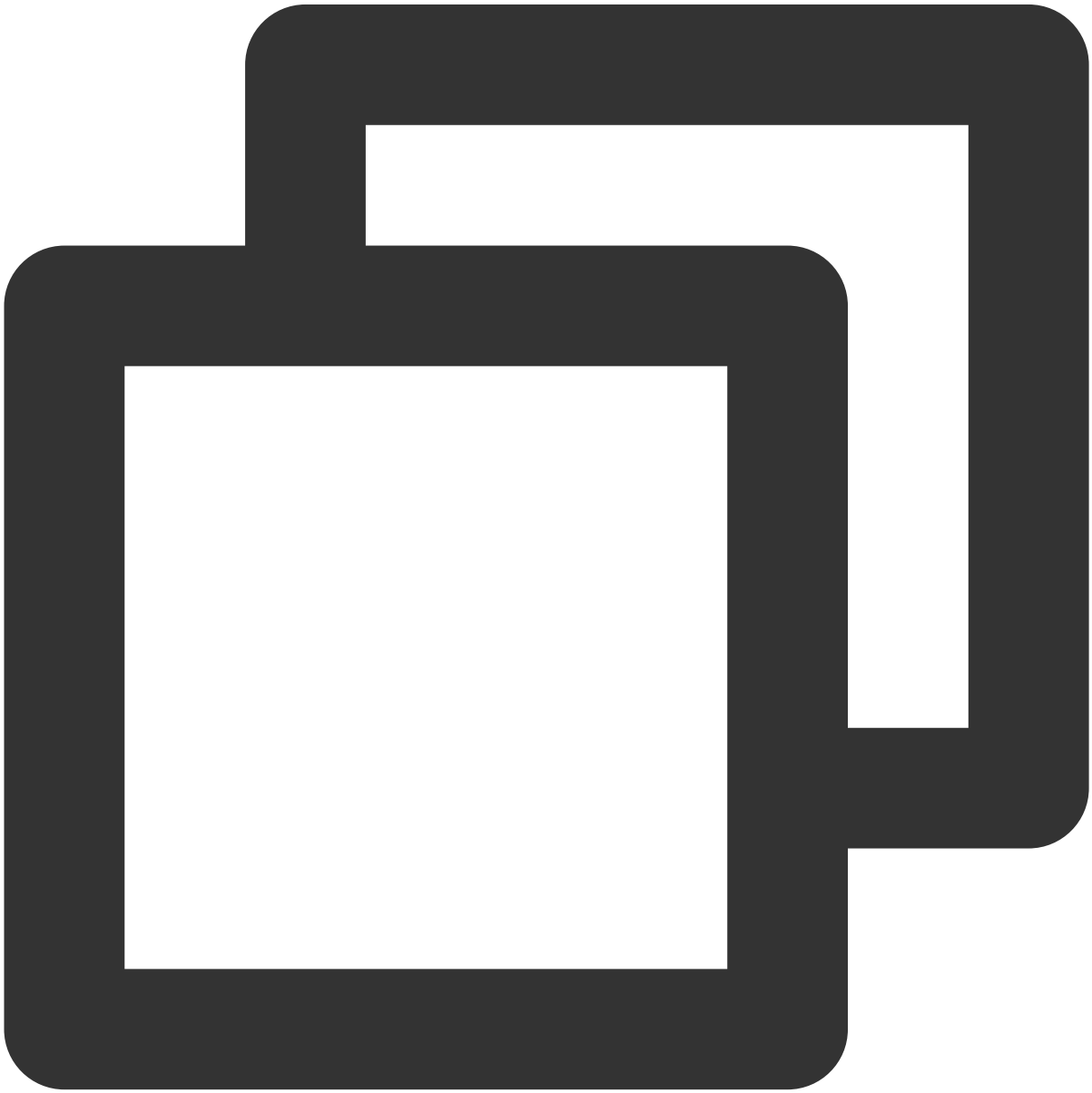
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

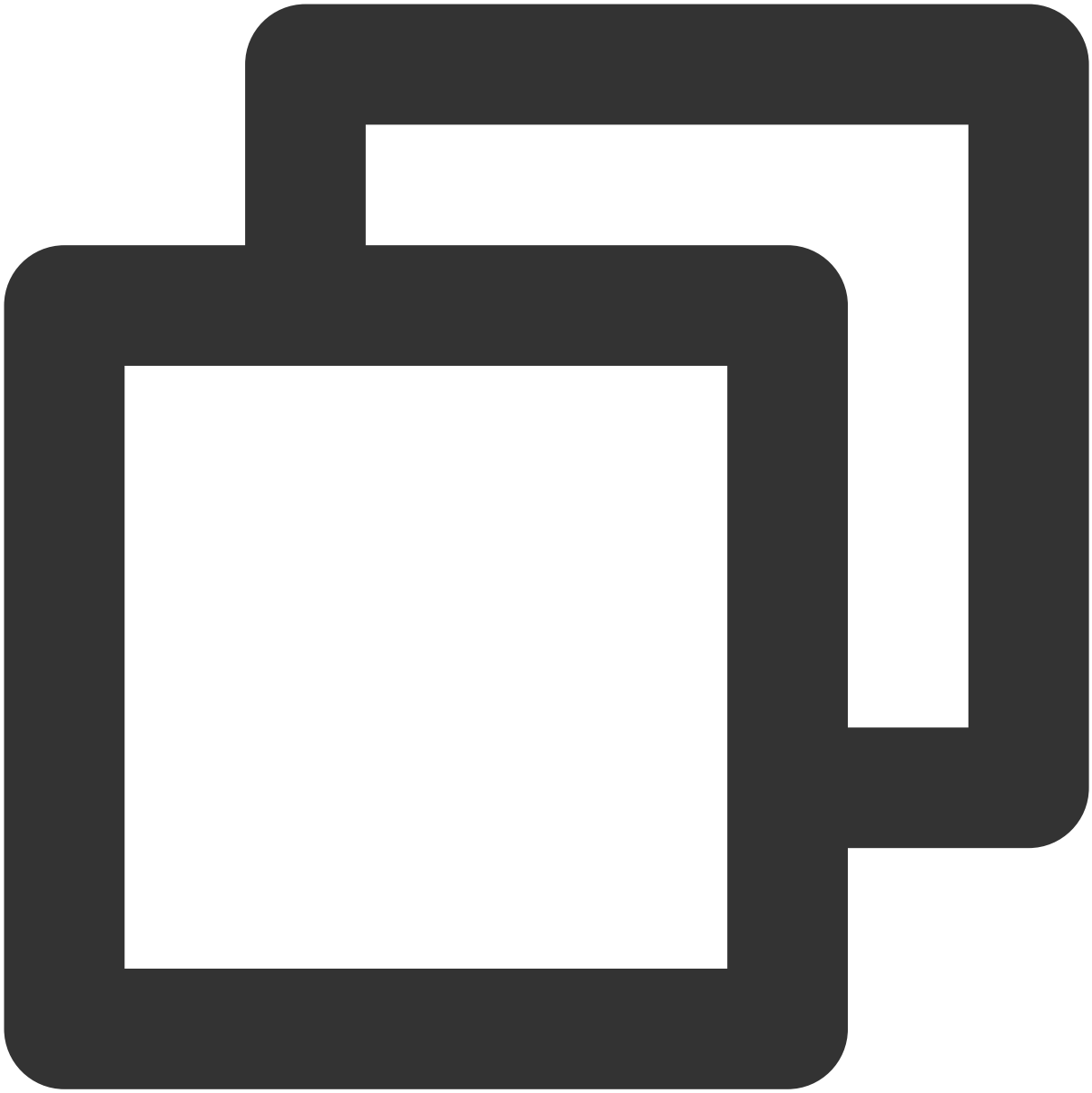
说明：

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：

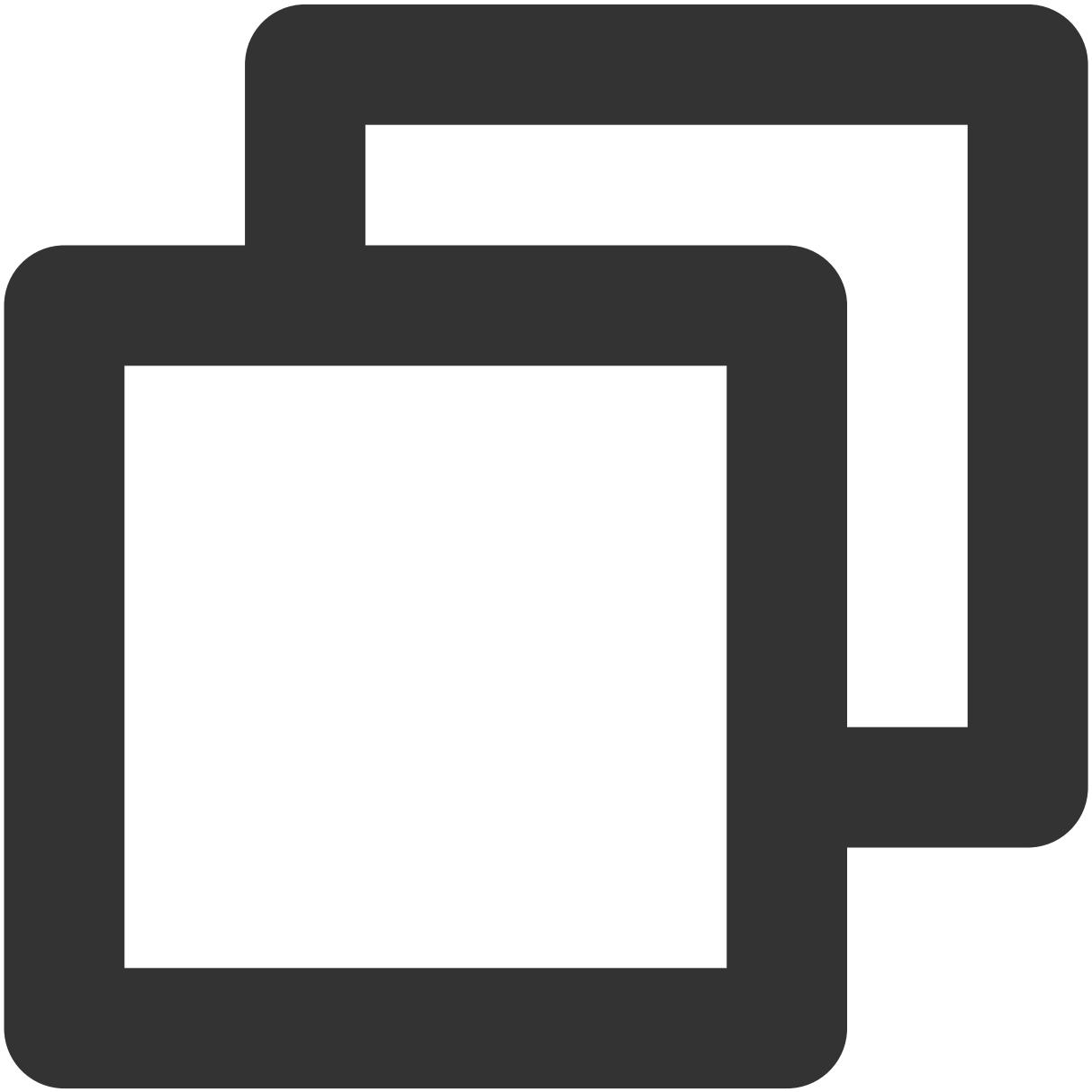
节点名称（关键字）	父节点	描述	类型
Response	无	同创建视频增强模板接口的 Response	Container

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

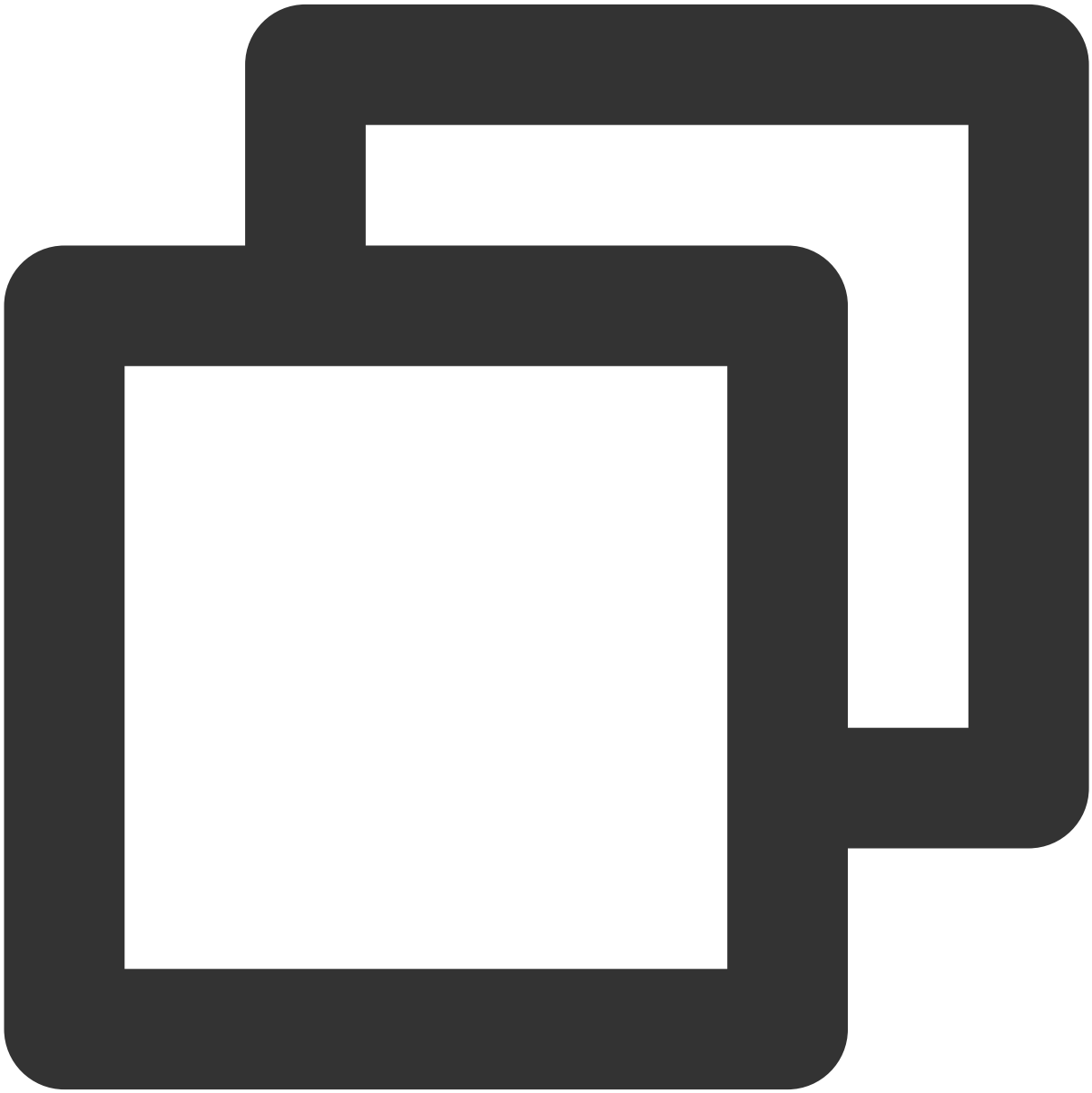


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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>
```

更新人声分离模板

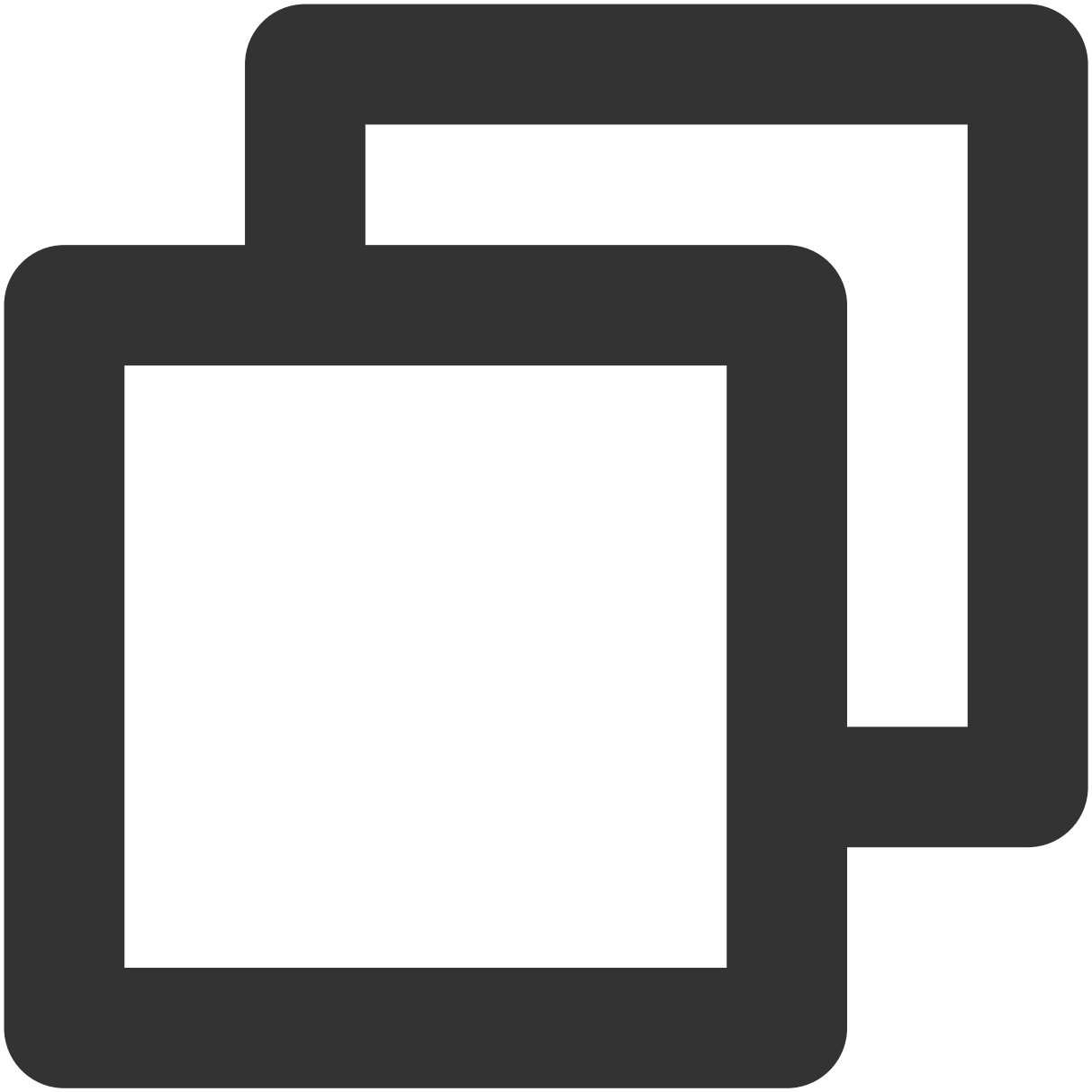
最近更新时间：2024-01-23 10:00:02

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：

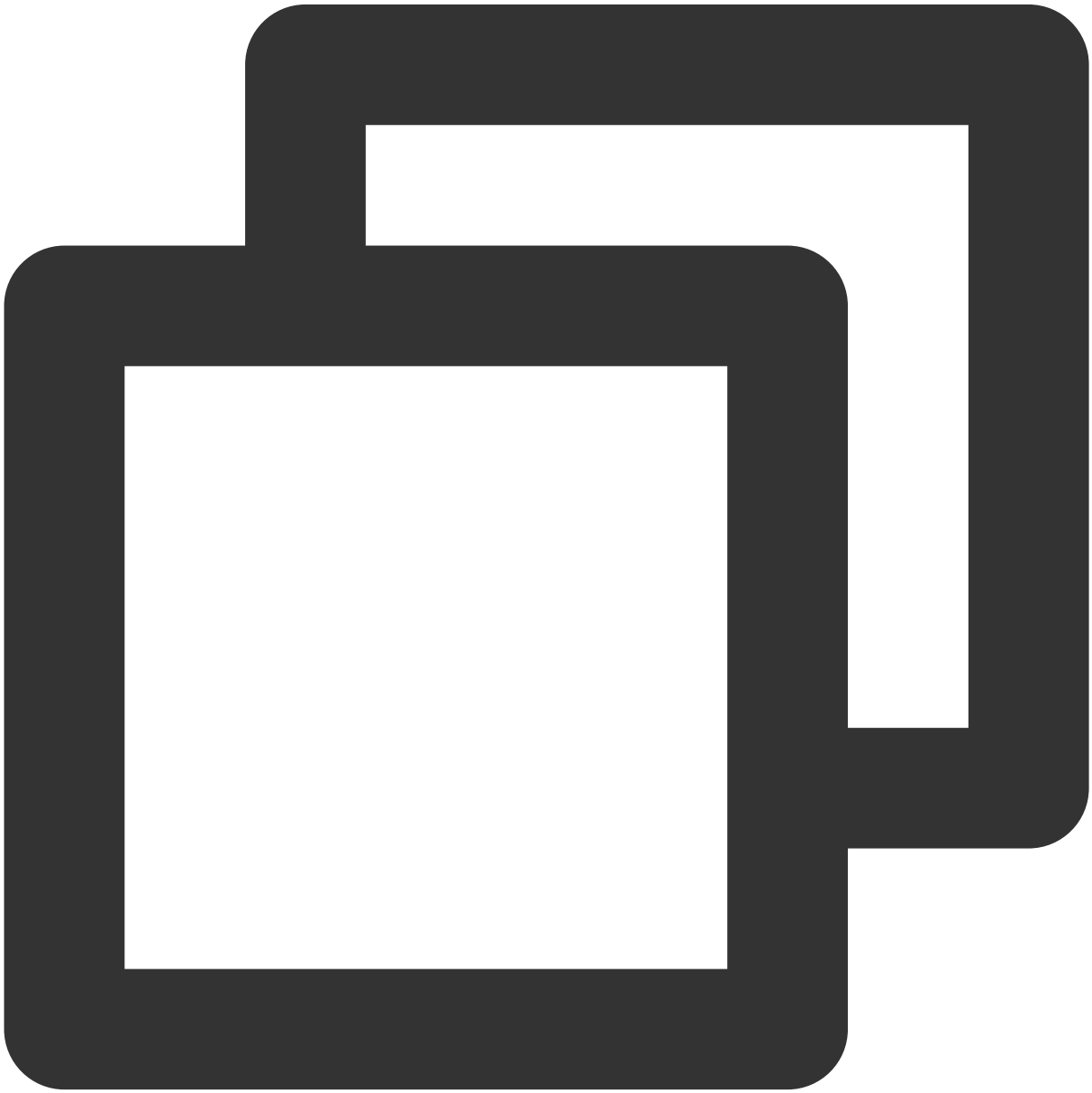
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

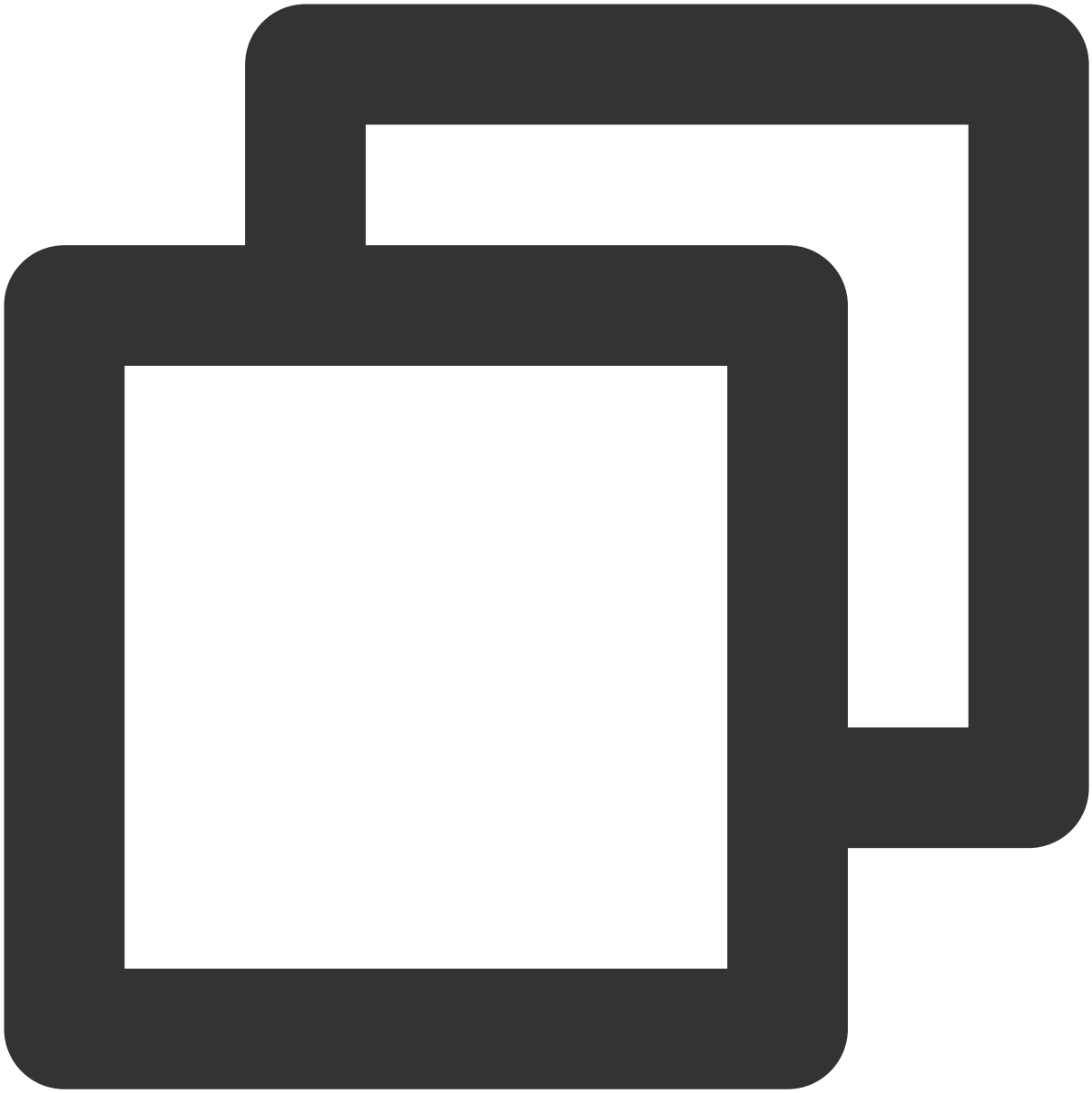
说明：

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：

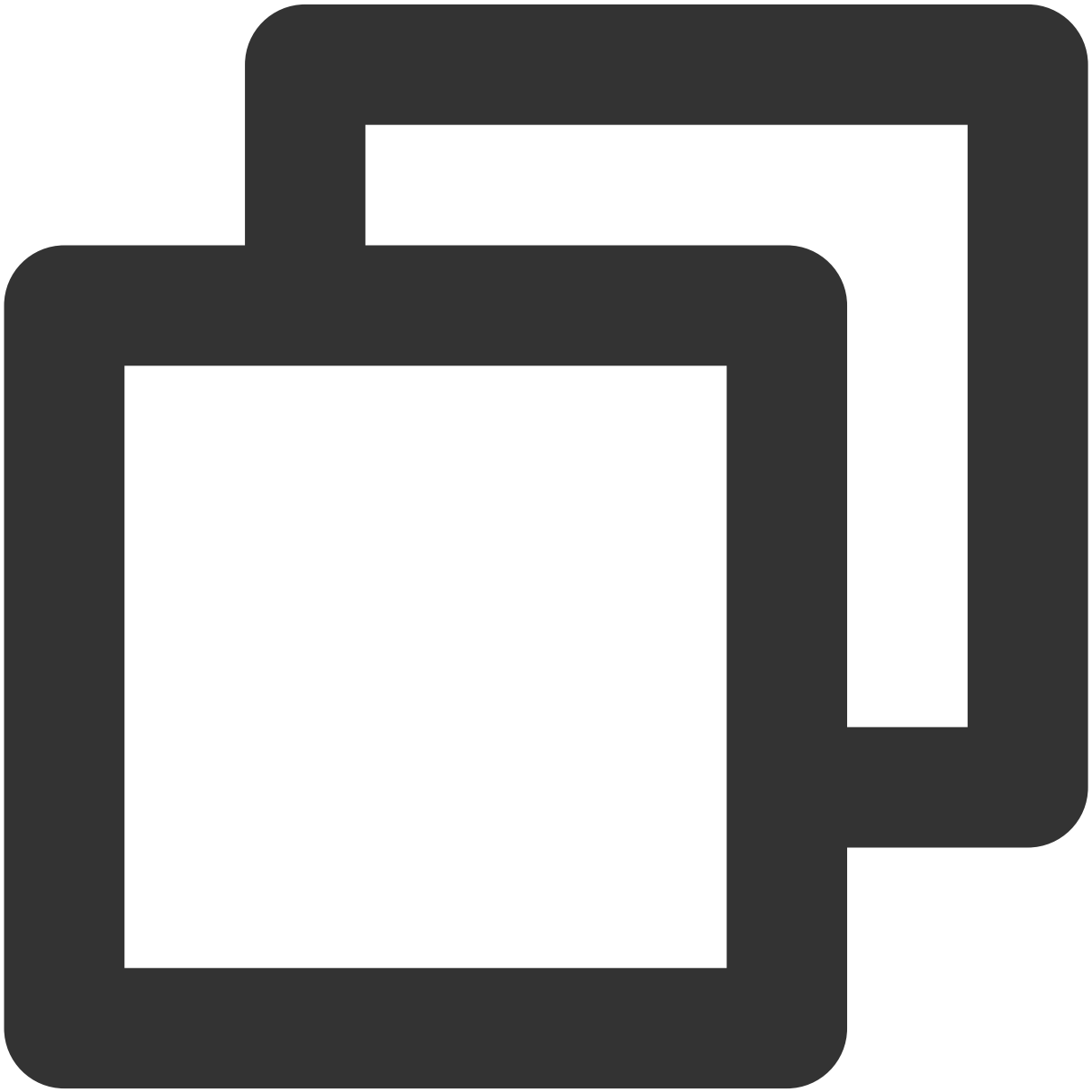
节点名称（关键字）	父节点	描述	类型
Response	无	同创建人声分离模板接口的 Response	Container

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

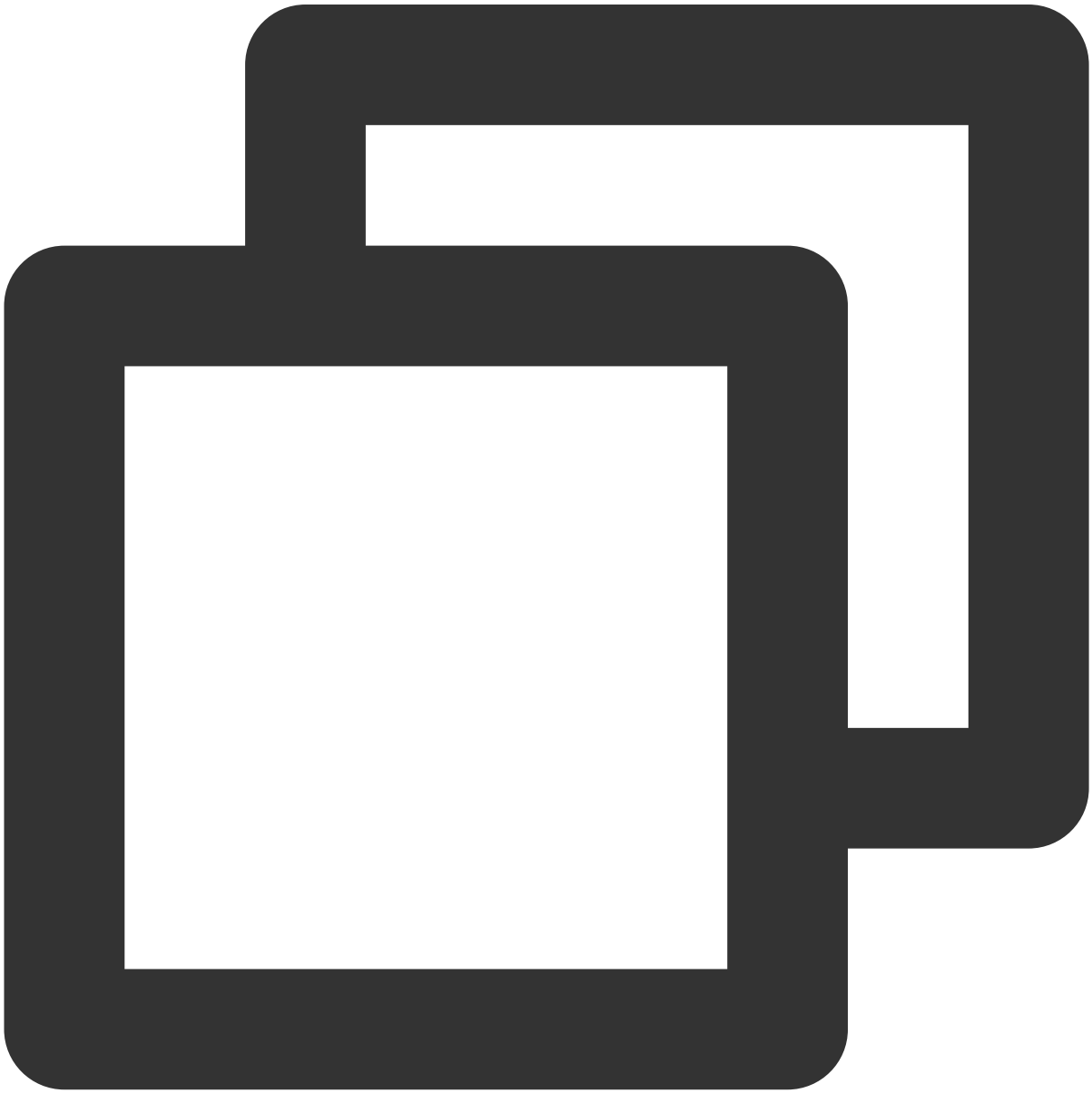


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

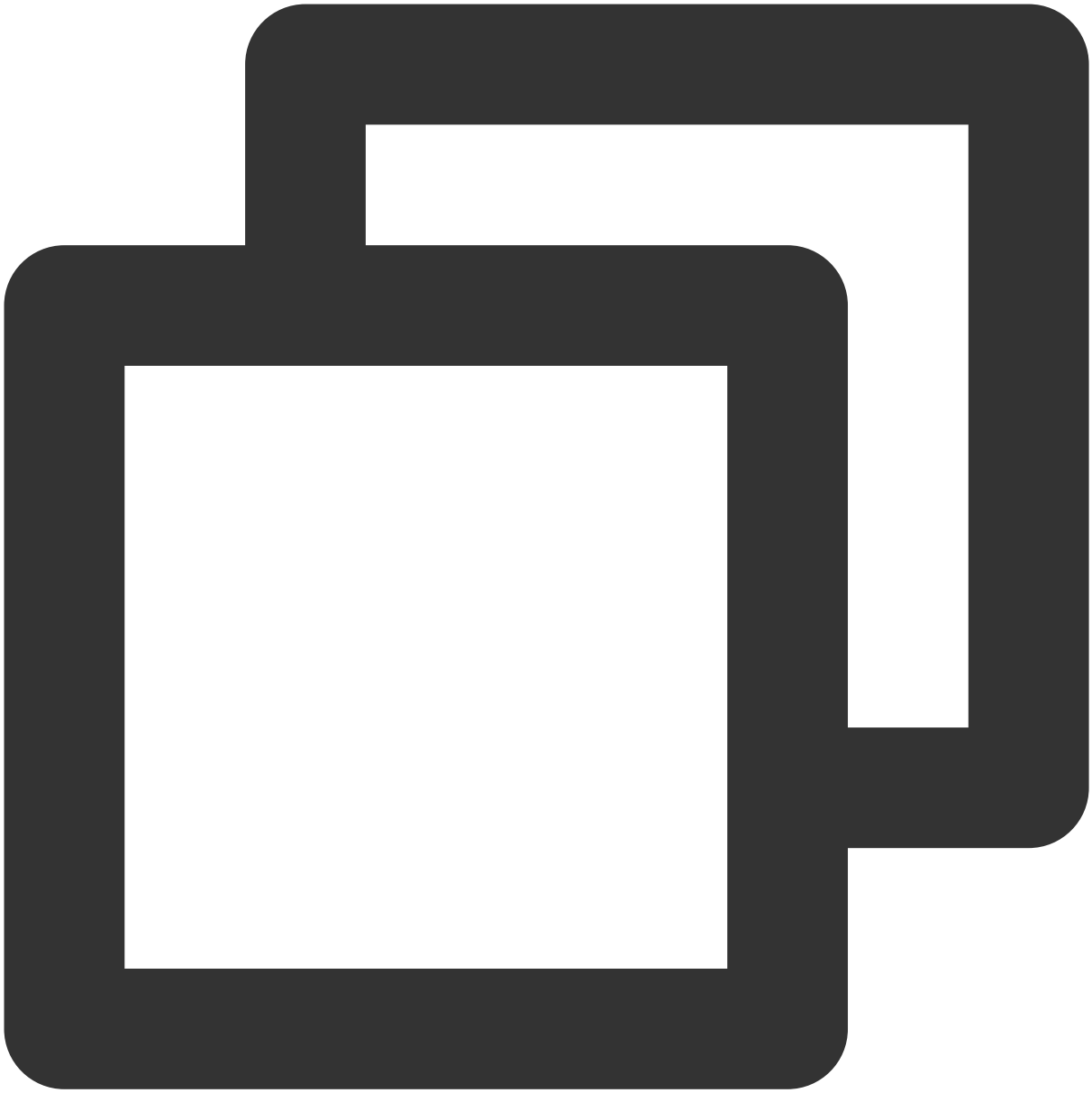
最近更新时间：2024-01-23 10:00:02

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：

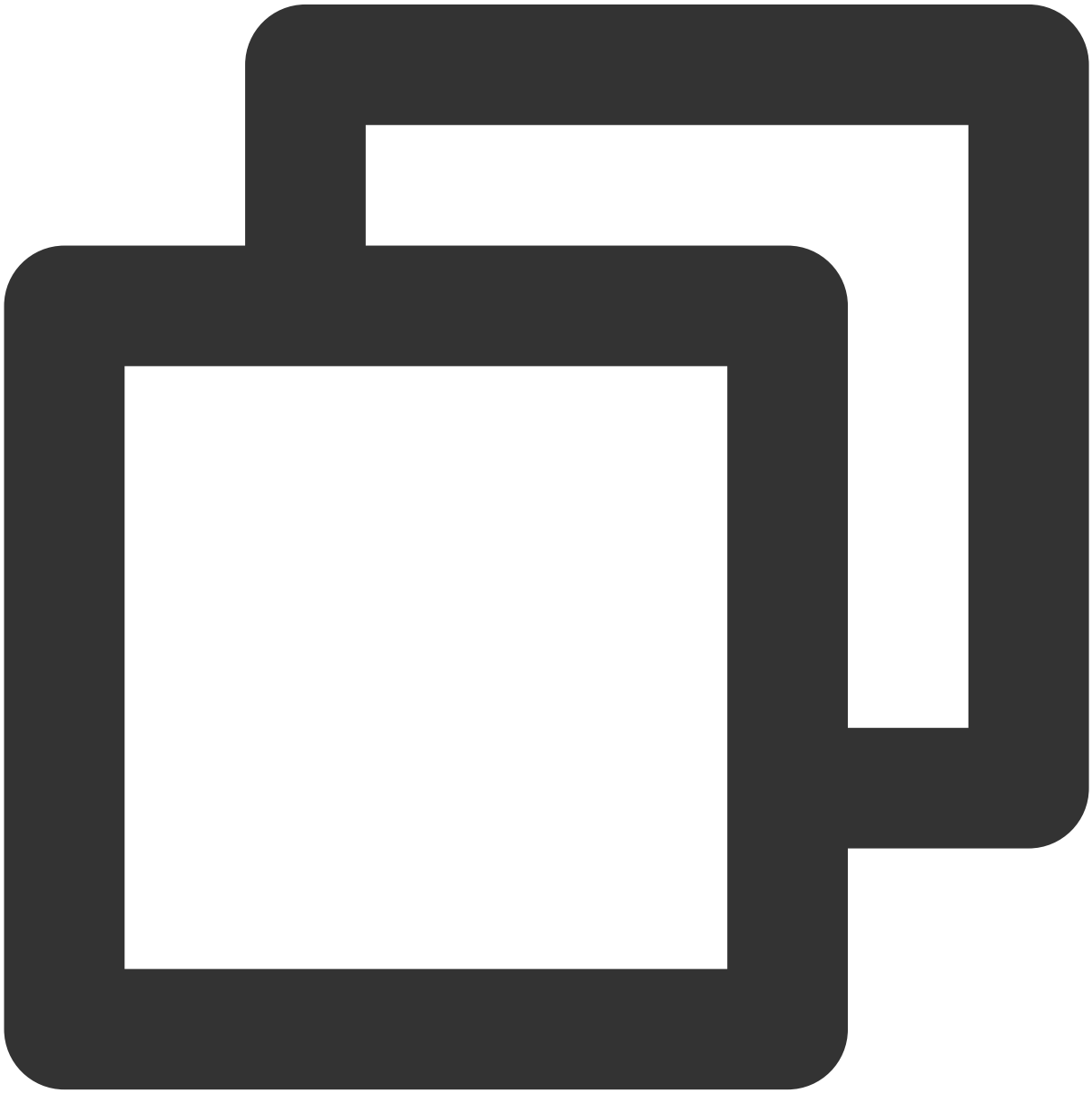
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

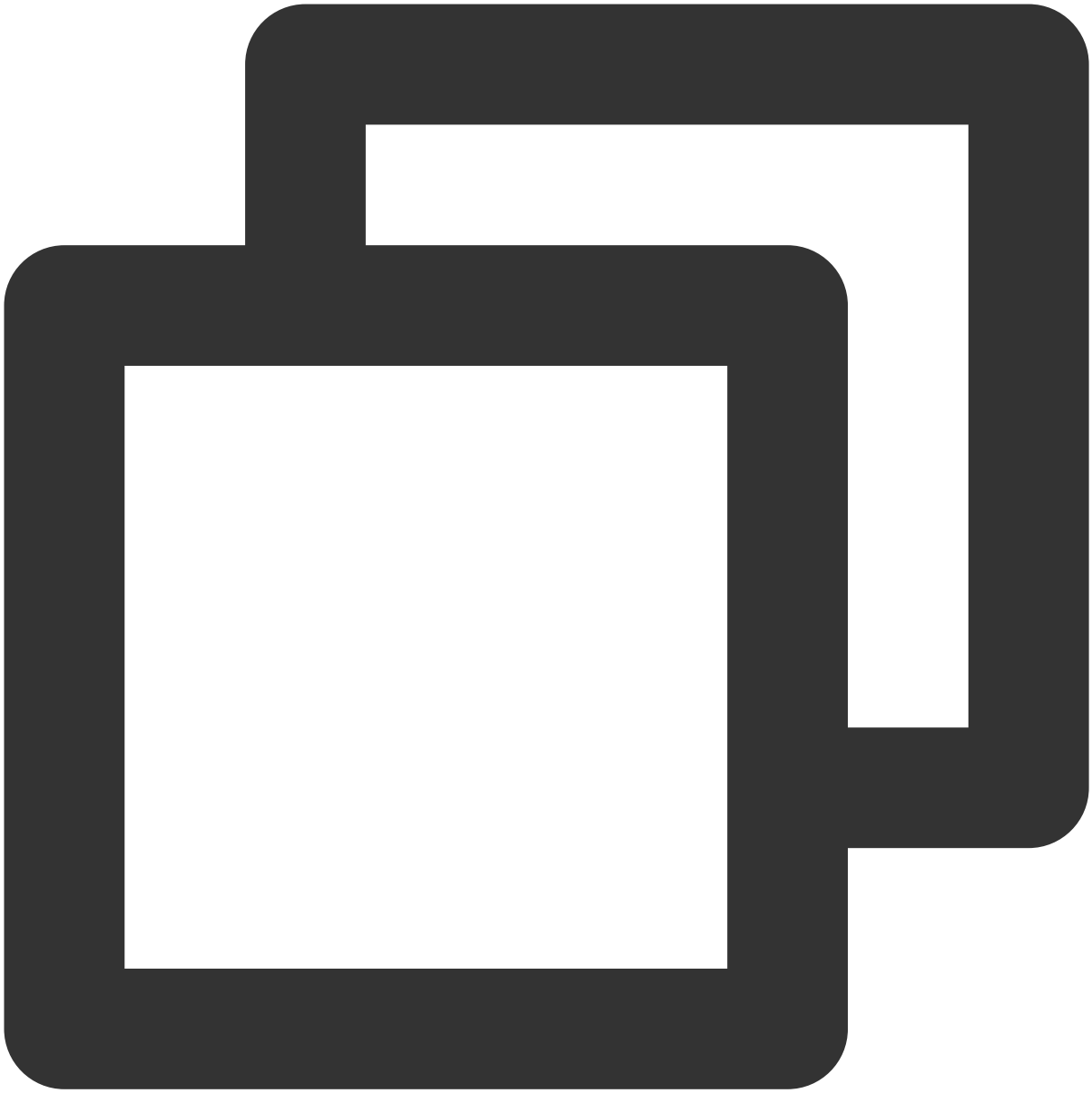
响应

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：

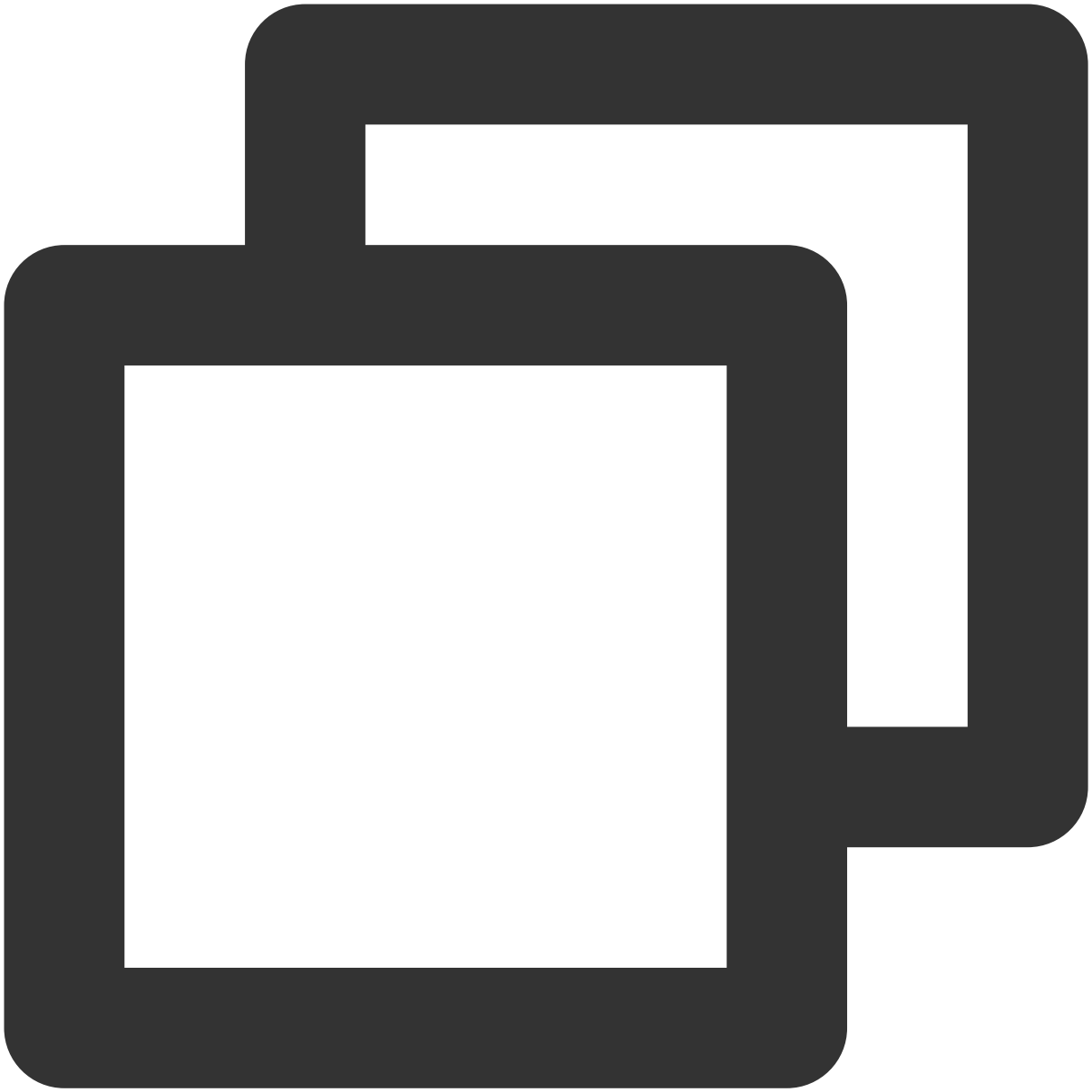
节点名称（关键字）	父节点	描述	类型
Response	无	同创建水印模板接口的 Response	Container

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

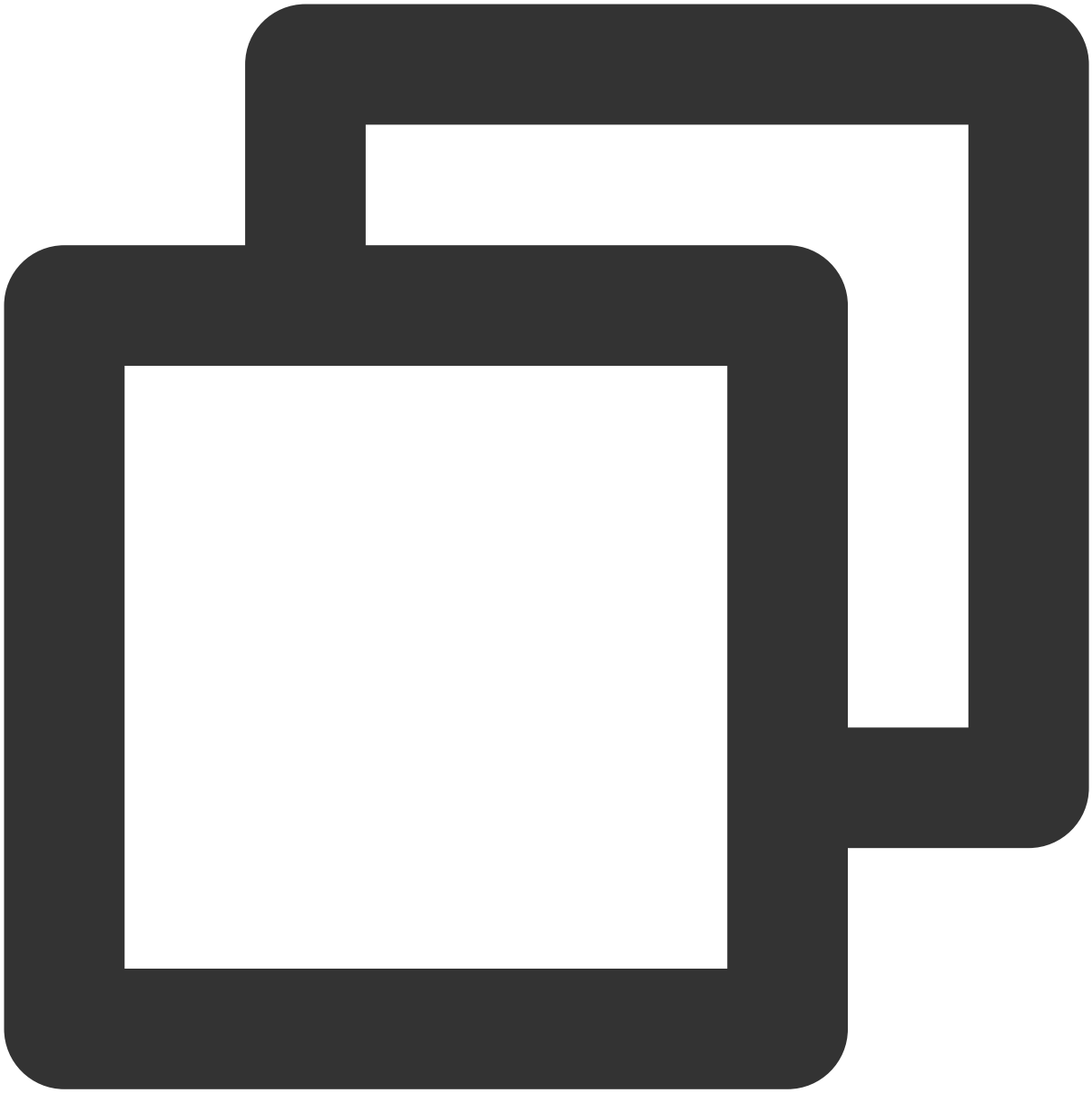


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Watermark</Tag>
    <Name>TemplateName</Name>
    <Watermark>
        <Type>Text</Type>
```

```
        <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

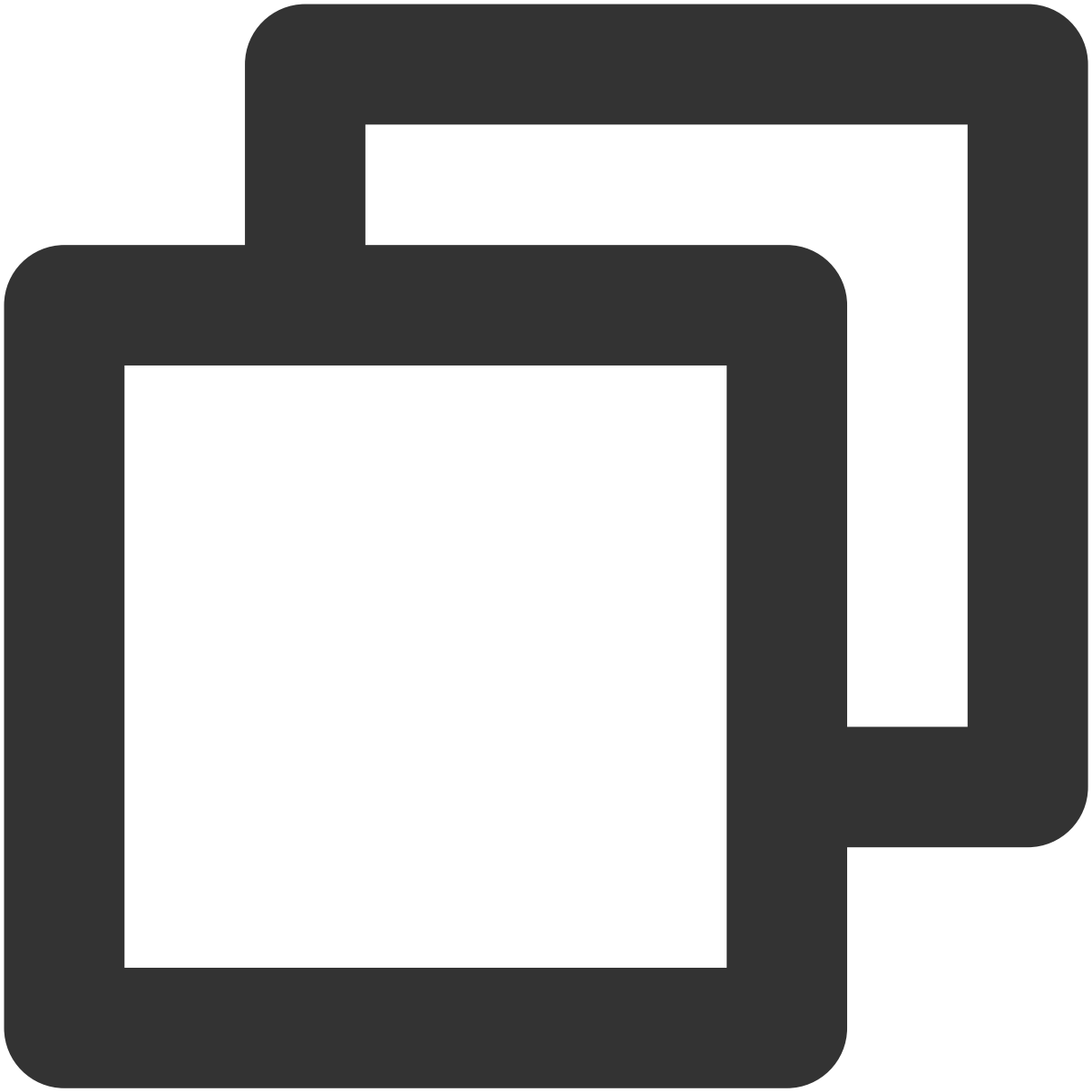
最近更新时间：2024-01-23 10:00:02

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：

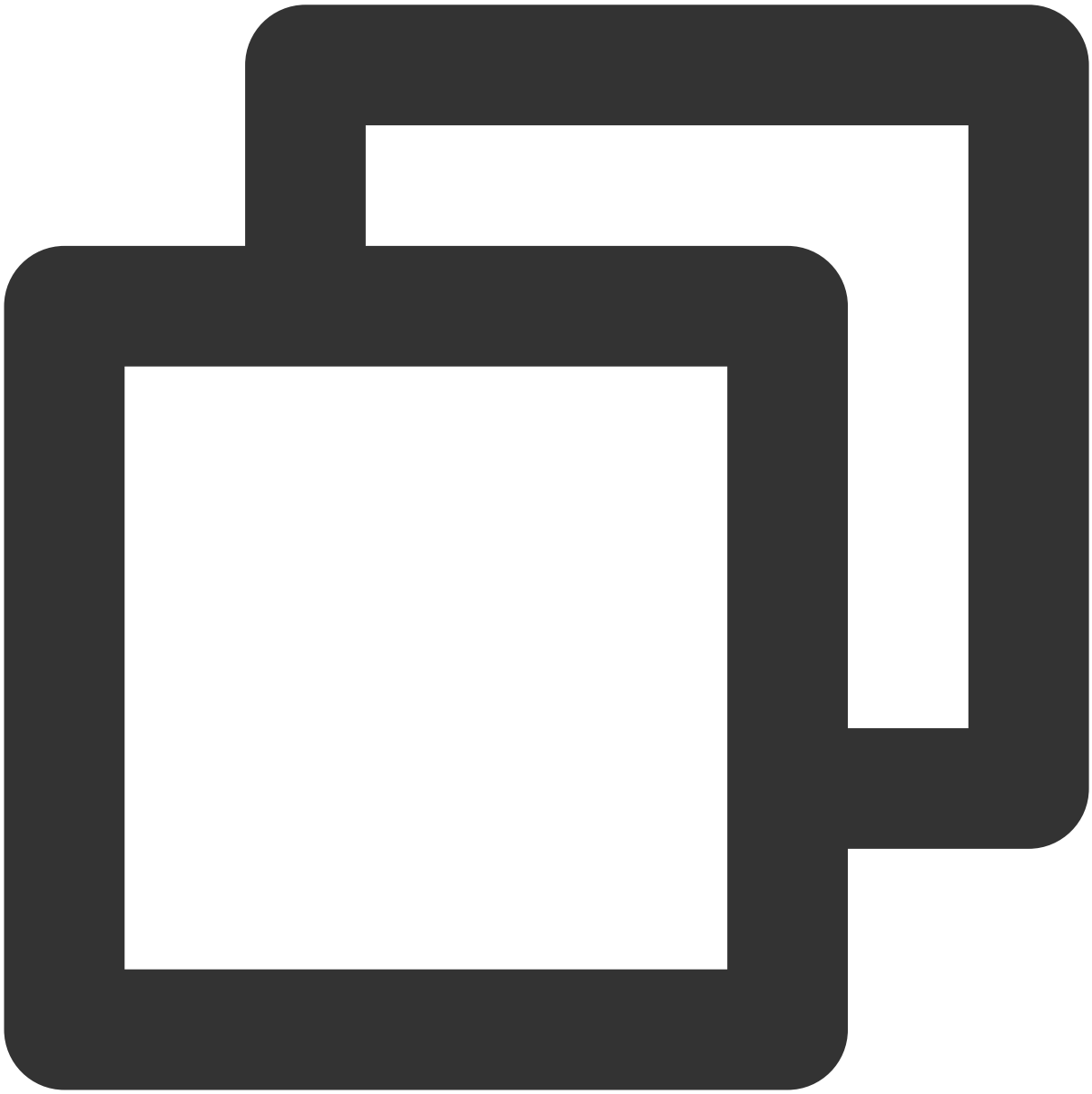
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

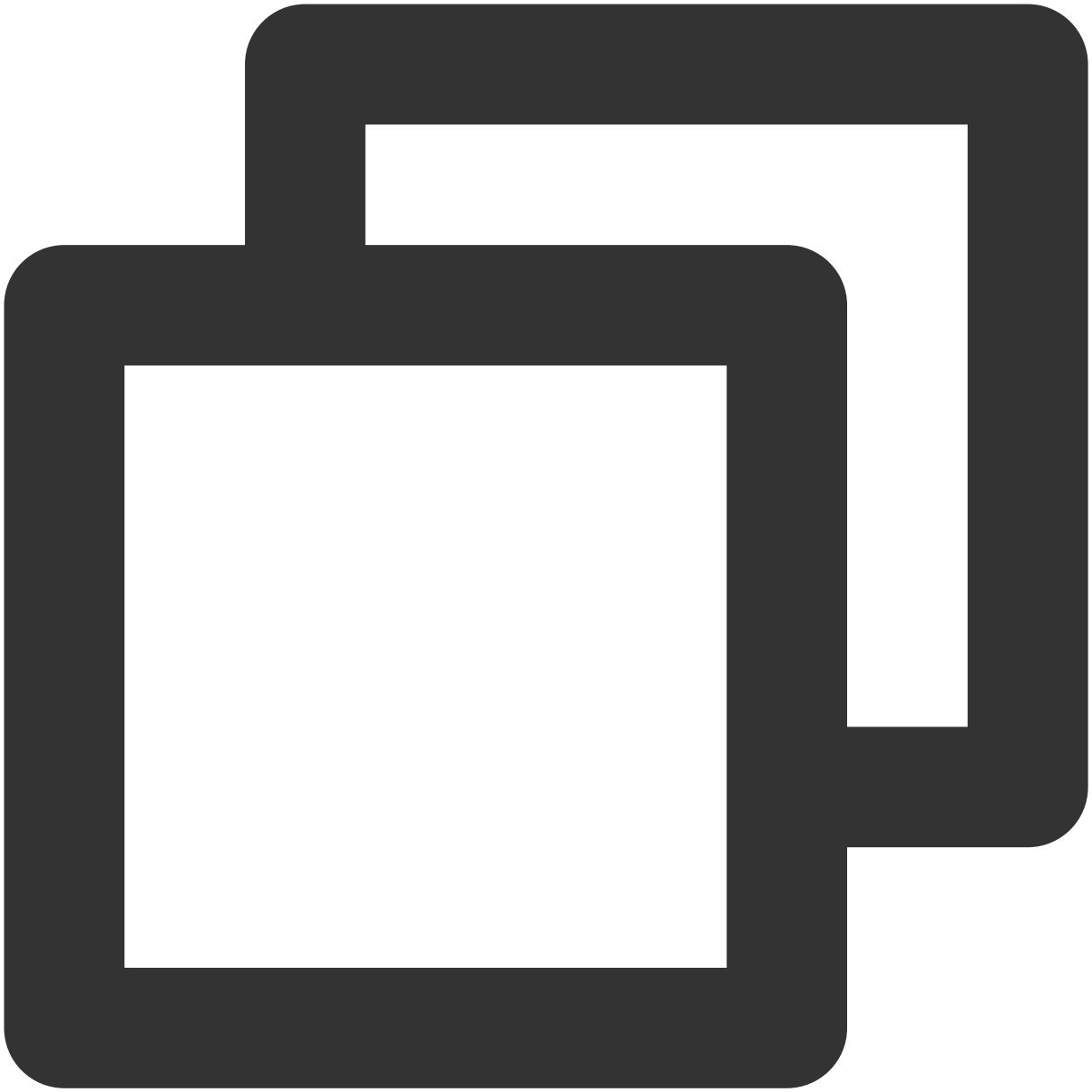
响应

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：

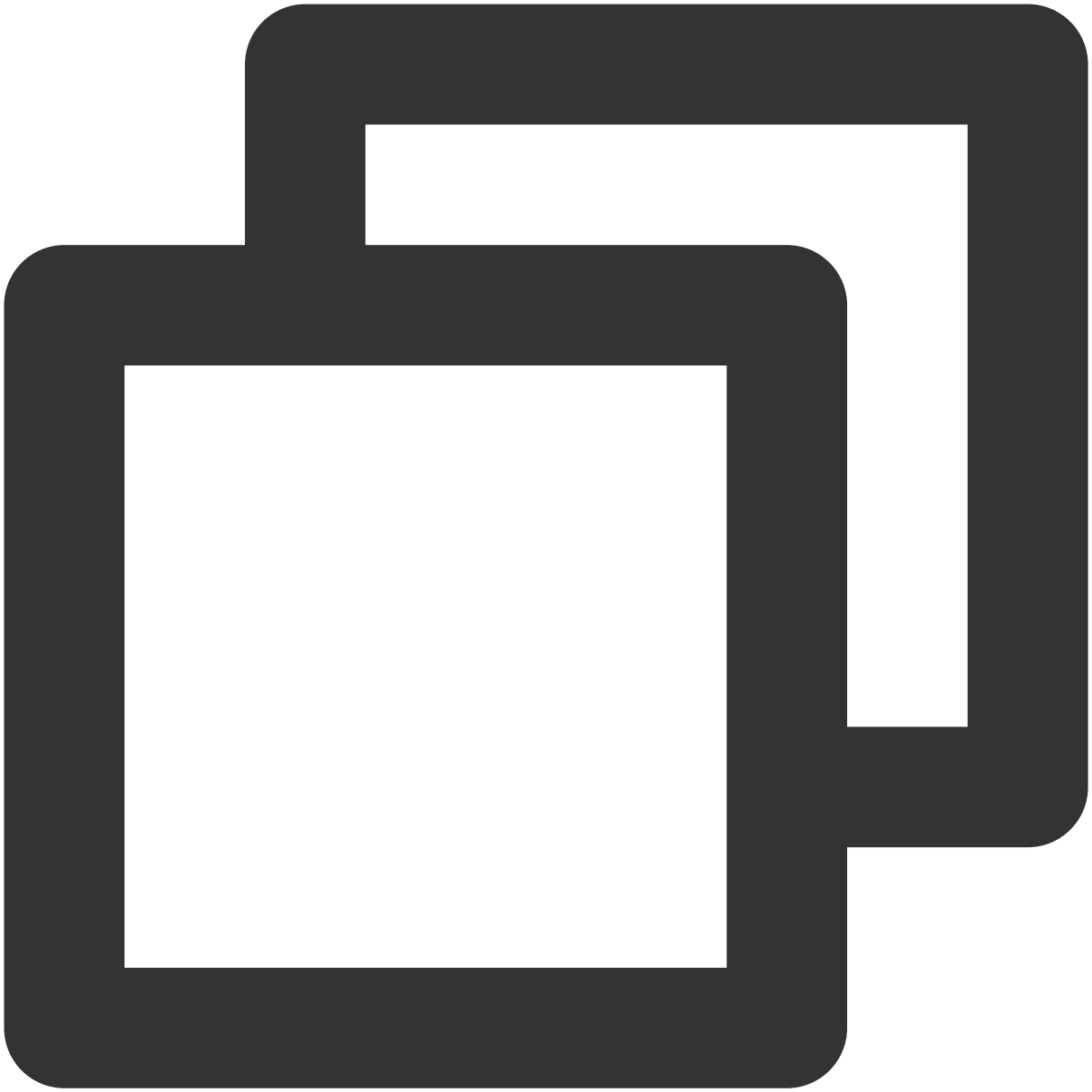
节点名称（关键字）	父节点	描述	类型
Response	无	同创建智能封面模板接口的 Response	Container

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

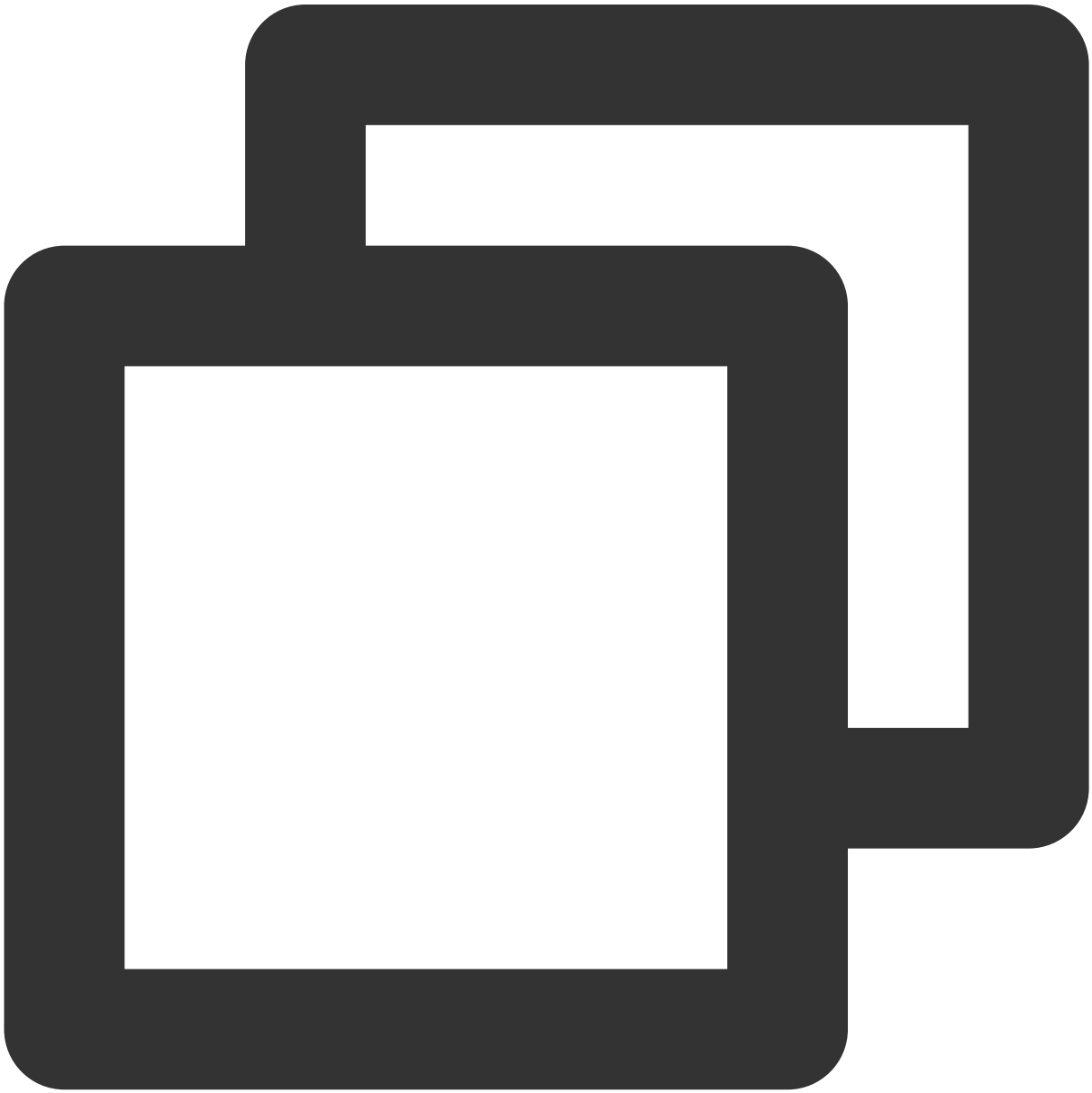


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

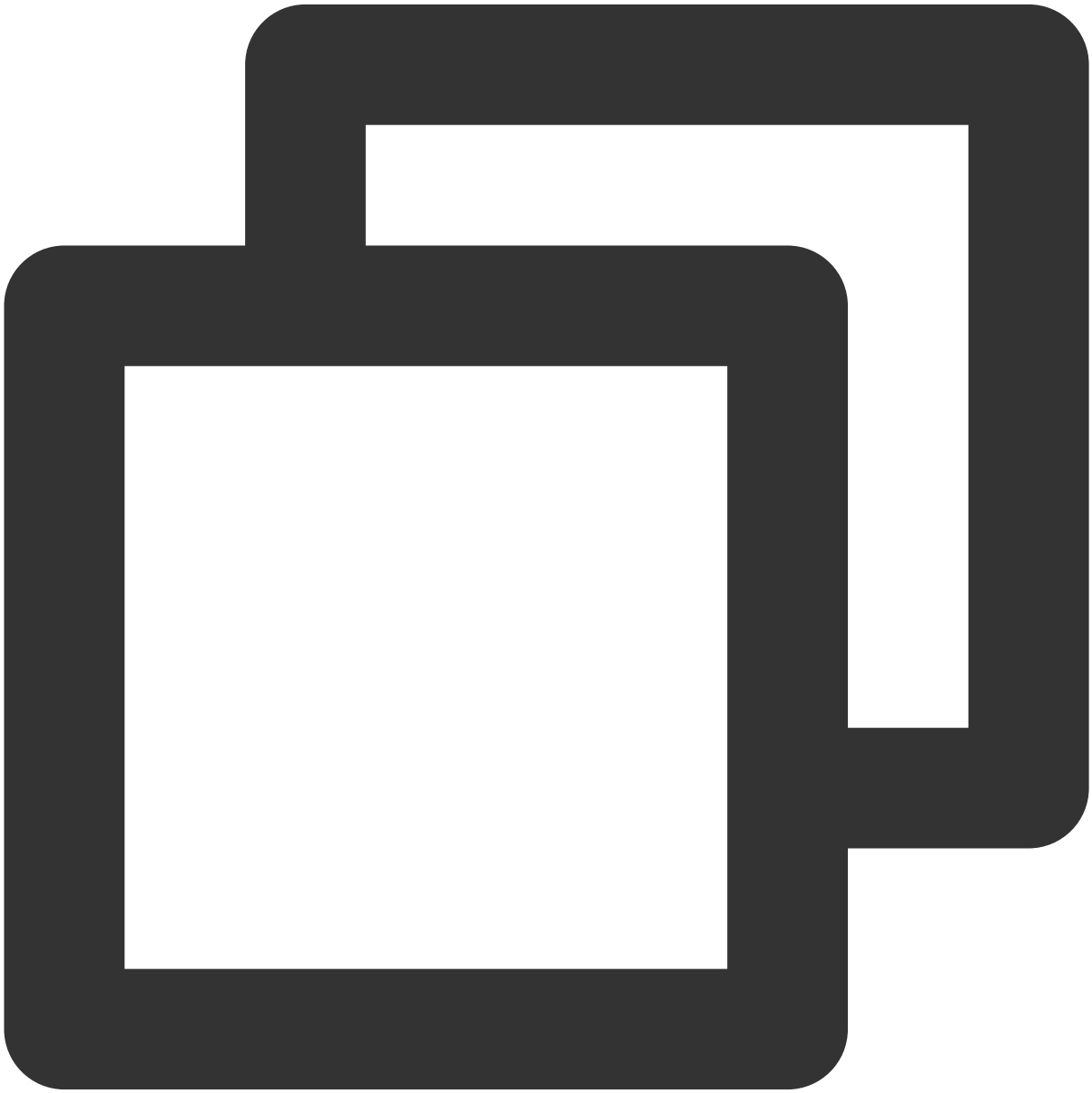
最近更新时间：2024-03-01 17:30:47

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：

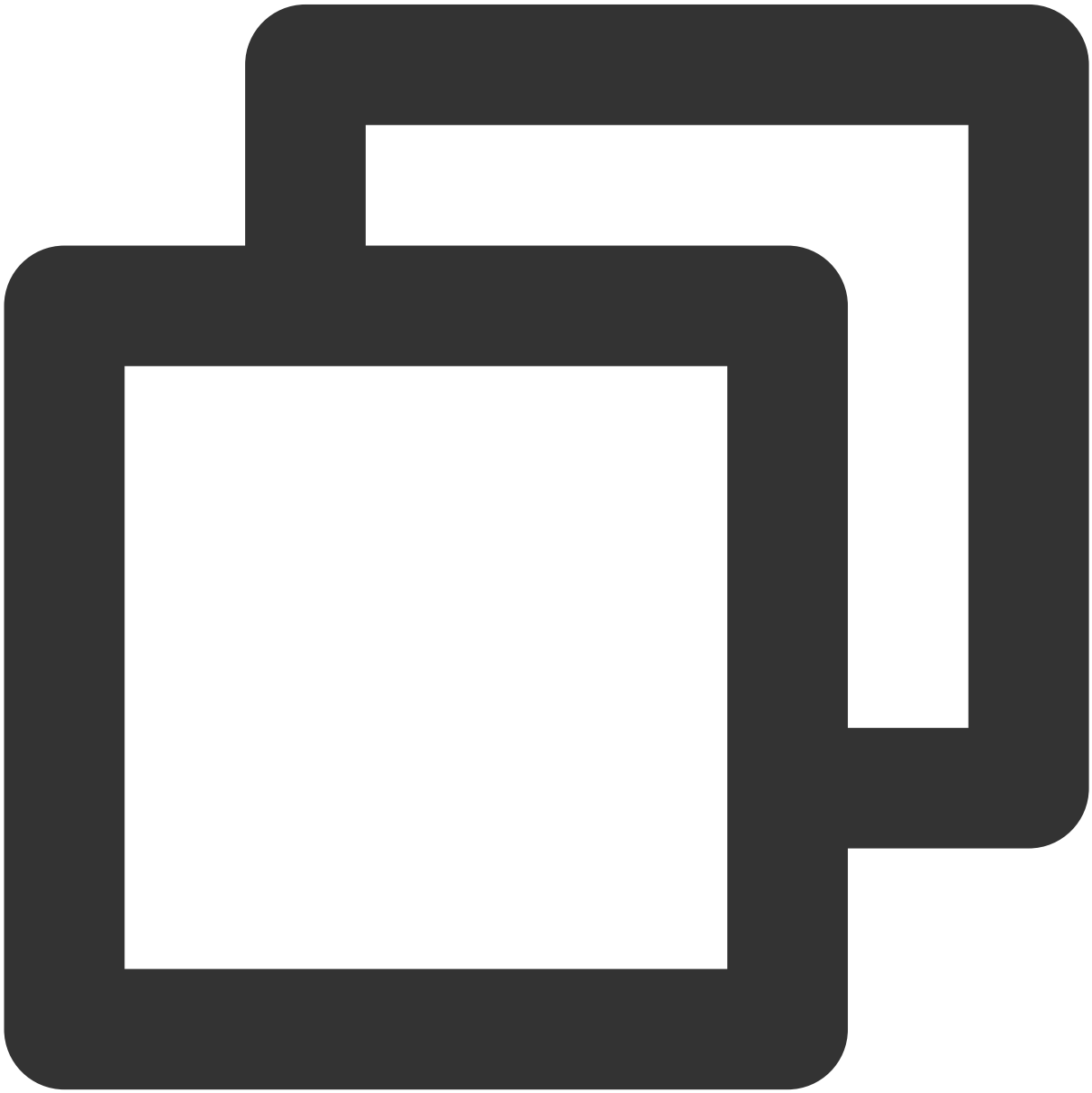
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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）； 8k_zh_s：电话 8k 中文普通话话者分离（仅适用于单声道	String	是

版权所有：腾讯云计算（北京）有限责任公司

		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。 极速 ASR 仅支持8k_zh、16k_zh、16k_en、16k_zh_video		
ChannelNum	Request.SpeechRecognition	仅支持非极速 ASR 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	仅支持非极速 ASR 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 。 默认值为 0。	Integer	否
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。仅非极速 ASR 支持 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	仅支持非极速ASR 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

版权所有：腾讯云计算（北京）有限责任公司

		极速ASR仅支持txt		
FlashAsr	Request.SpeechRecognition	是否开启极速 ASR，可选 true、false。默认为 false	String	否
Format	Request.SpeechRecognition	极速 ASR 音频格式。支持 wav、pcm、ogg-opus、speex、silk、mp3、m4a、aac。	String	当 FlashAsr 为 true 时必选
FirstChannelOnly	Request.SpeechRecognition	极速 ASR 参数。表示是否只识别首个声道，默认为1。0：识别所有声道；1：识别首个声道。	Integer	否
WordInfo	Request.SpeechRecognition	极速 ASR 参数。表示是否显示词级别时间戳，默认为0。0：不显示；1：显示，不包含标点时间戳，2：显示，包含标点时间戳。	Integer	否

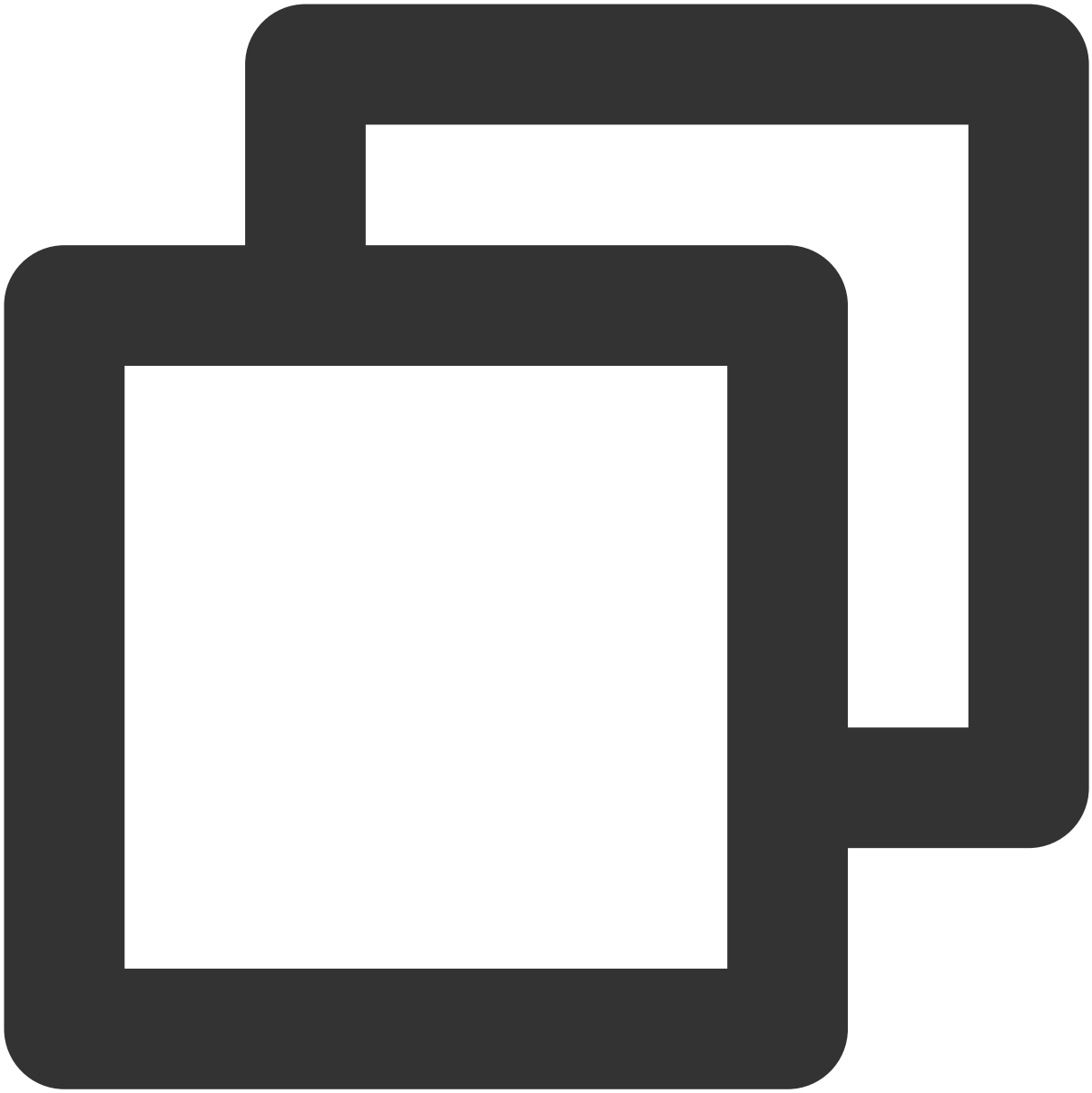
响应

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：

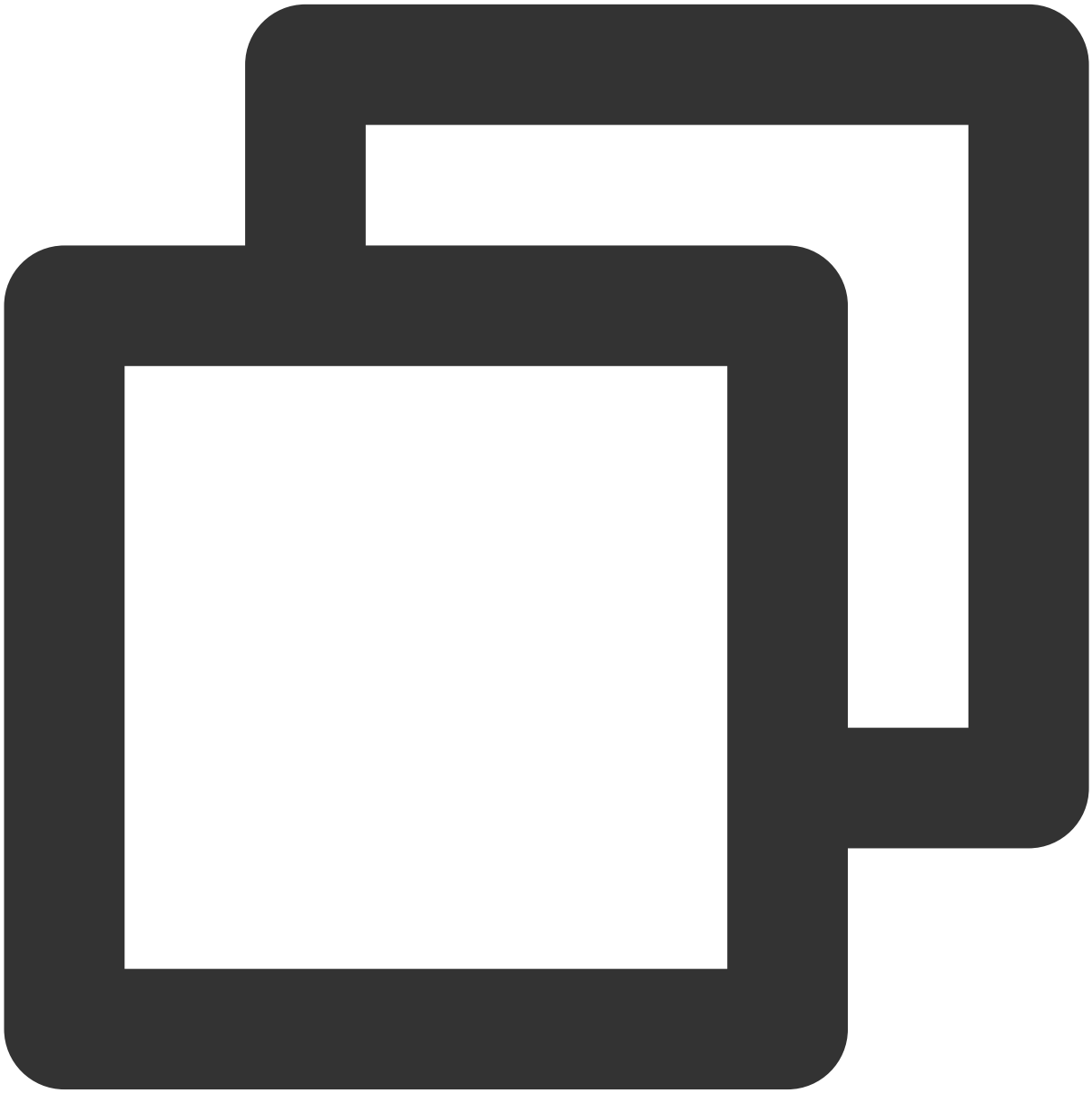
节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，SpeechRecognition	String
UpdateTime	Response.Template	更新时间	String
CreateTime	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

请求

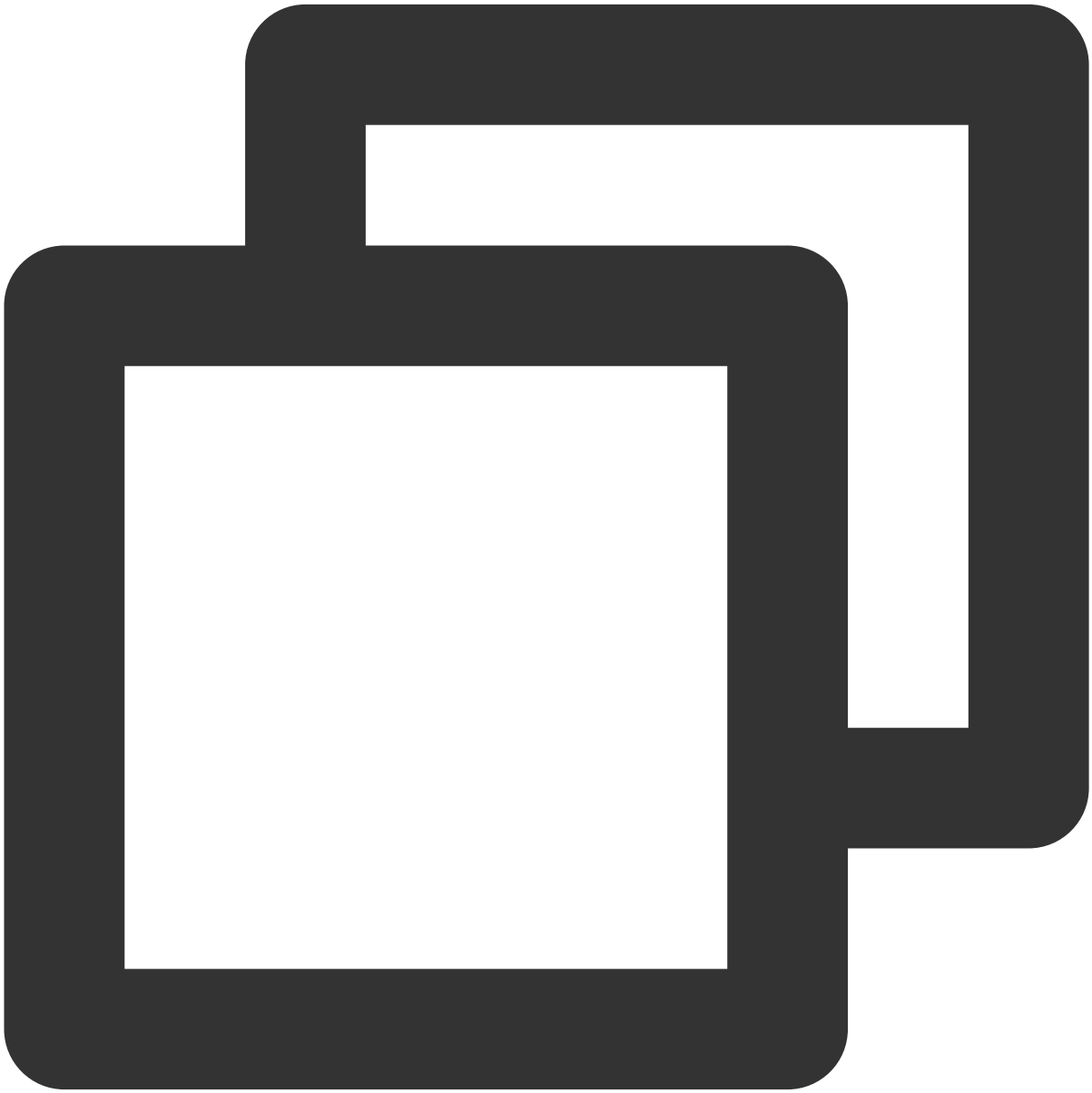


```
POST /template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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

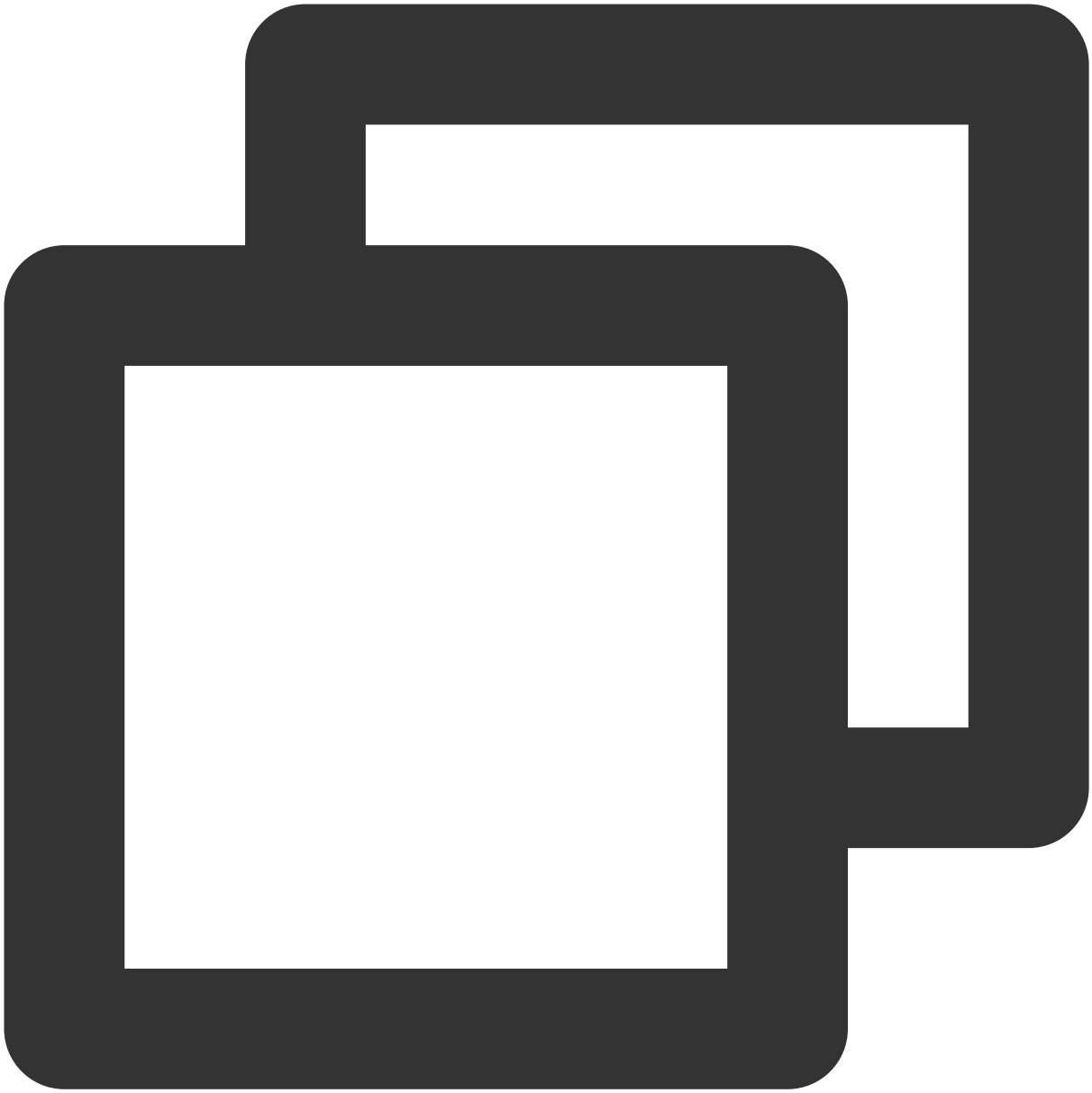
最近更新时间：2024-01-23 10:00:02

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

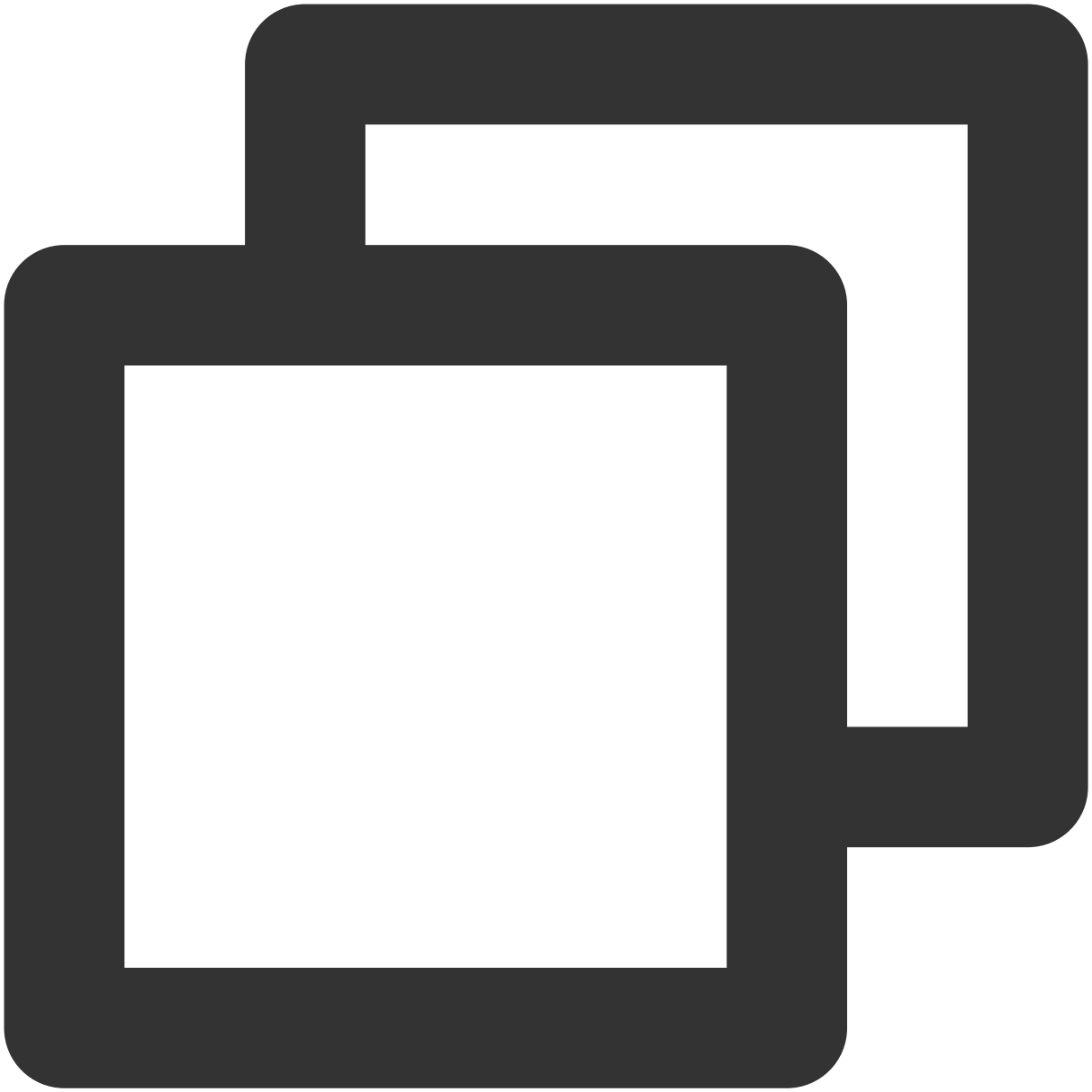
响应

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。

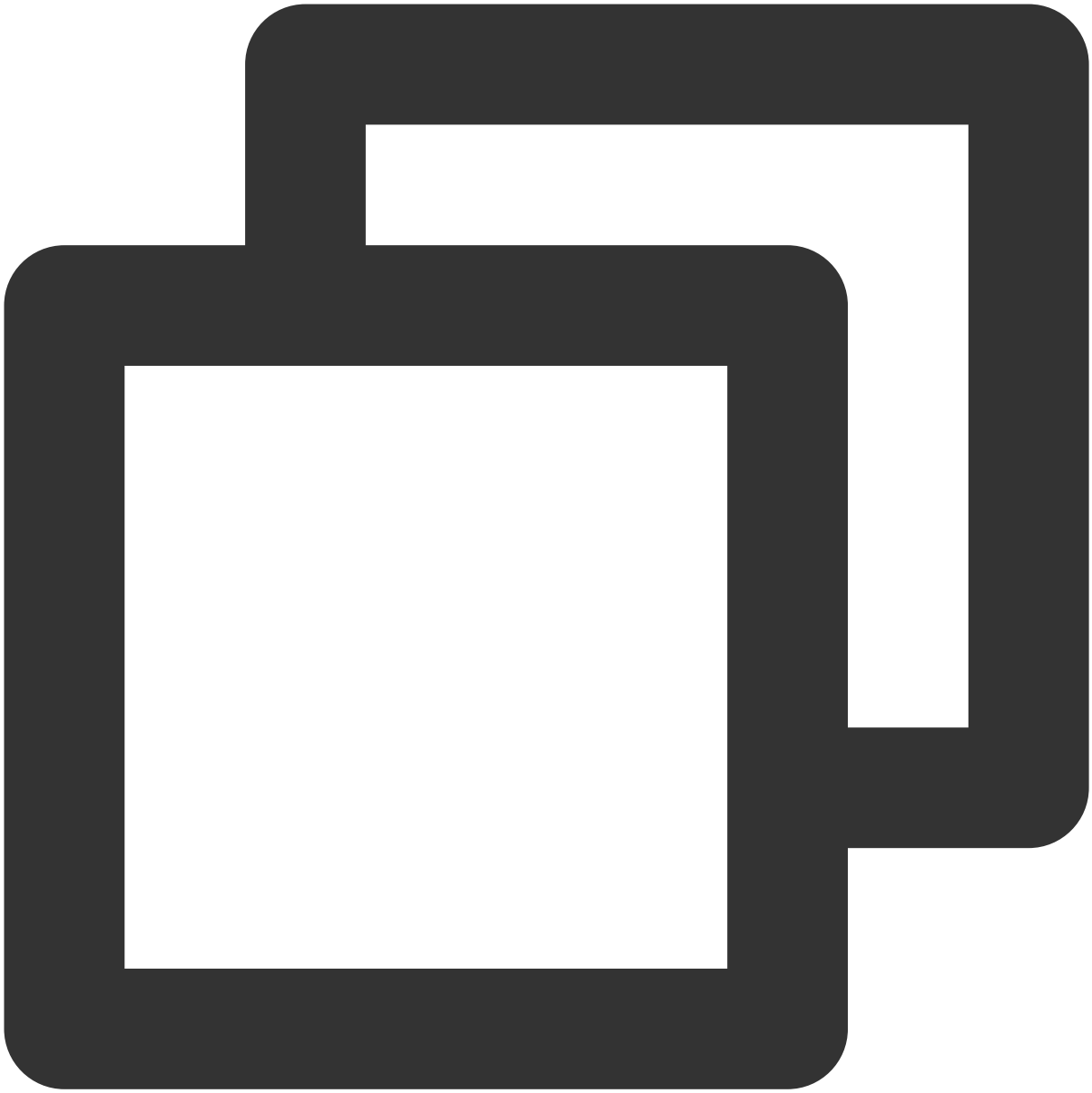
错误码

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

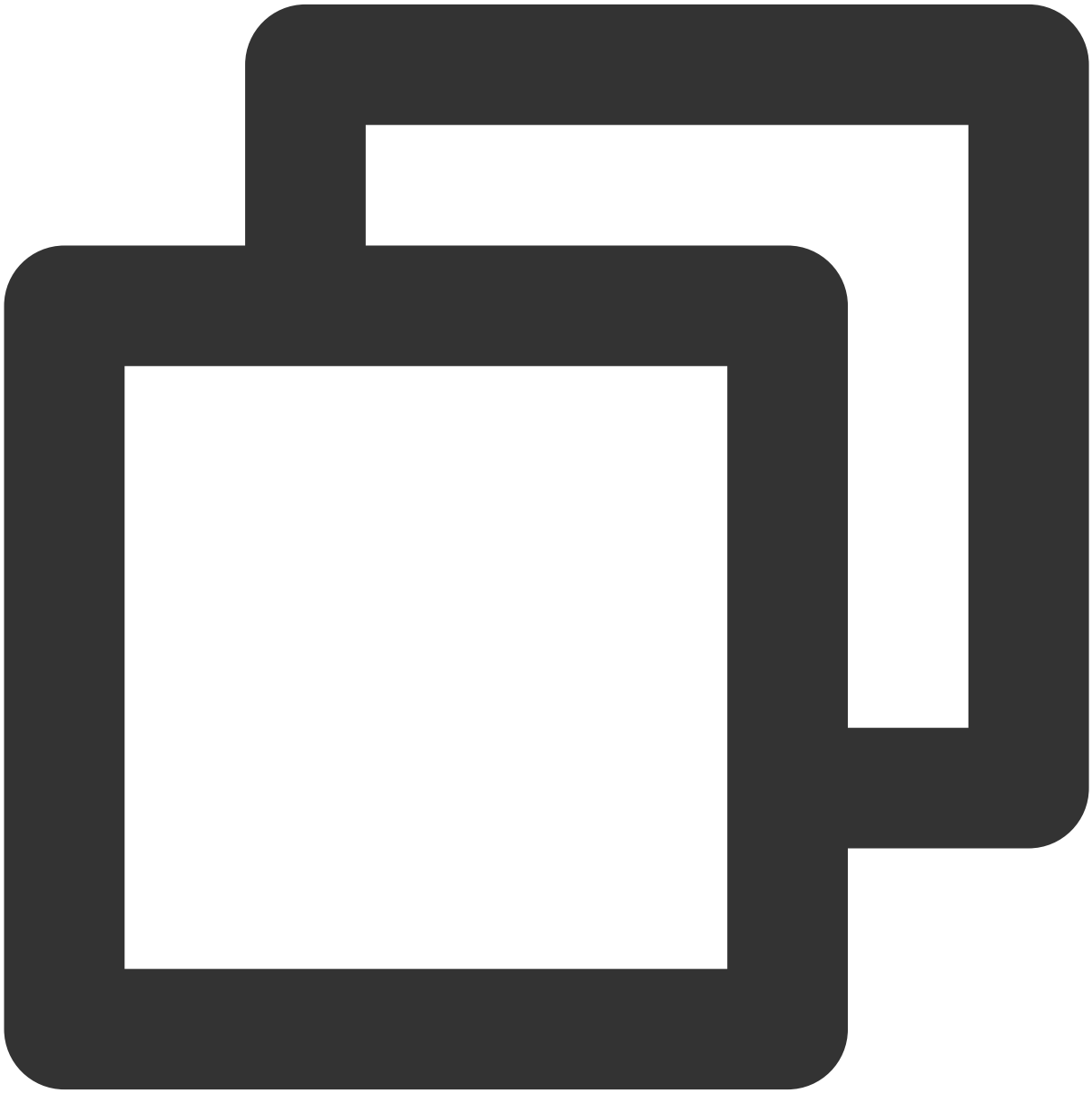
案例一：按照模板 ID 维度查询

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



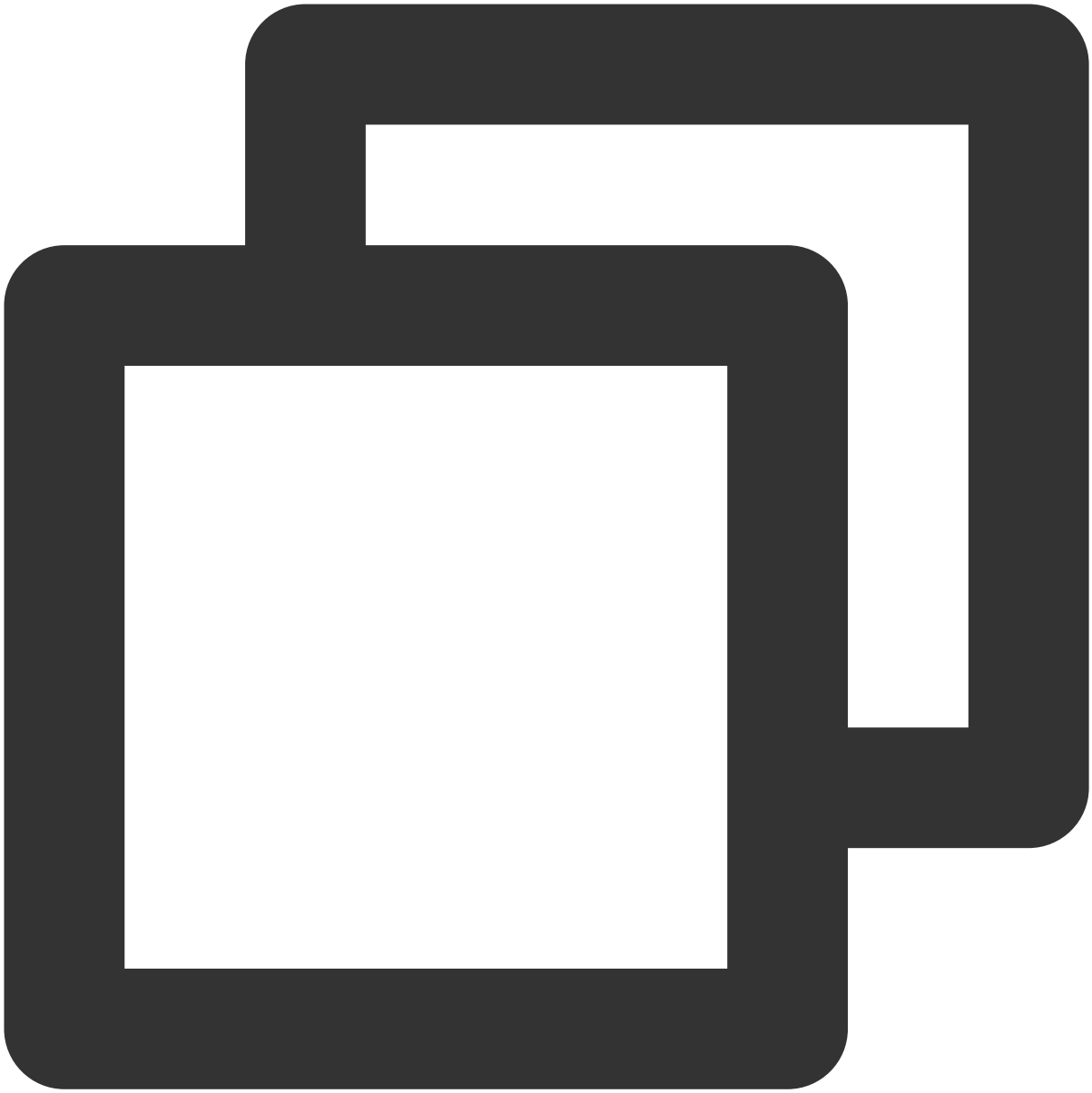
```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<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>
```

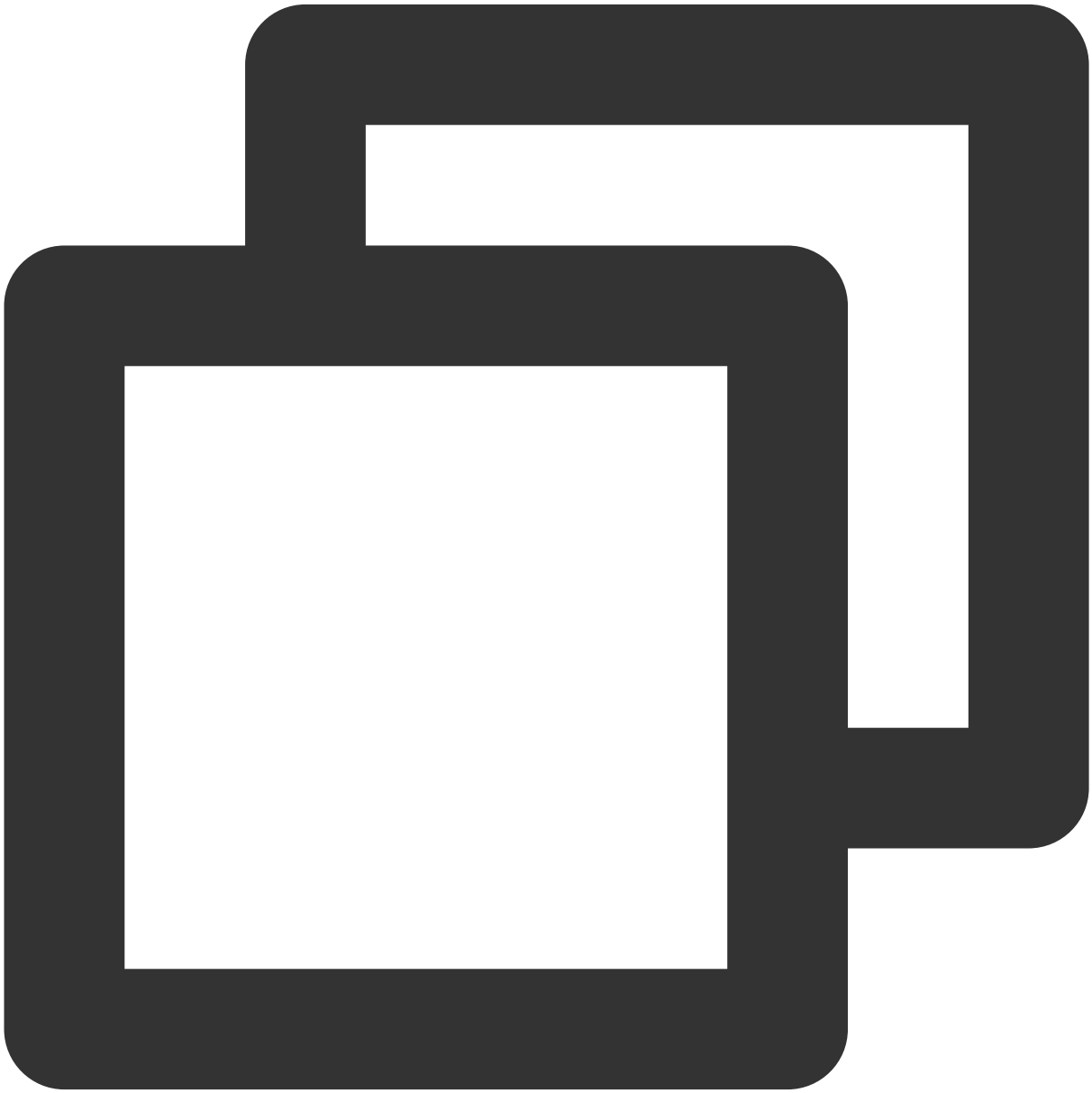
案例二：按照分页列表维度查询

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>
```

更新 AI 内容识别模板
更新语音识别模板

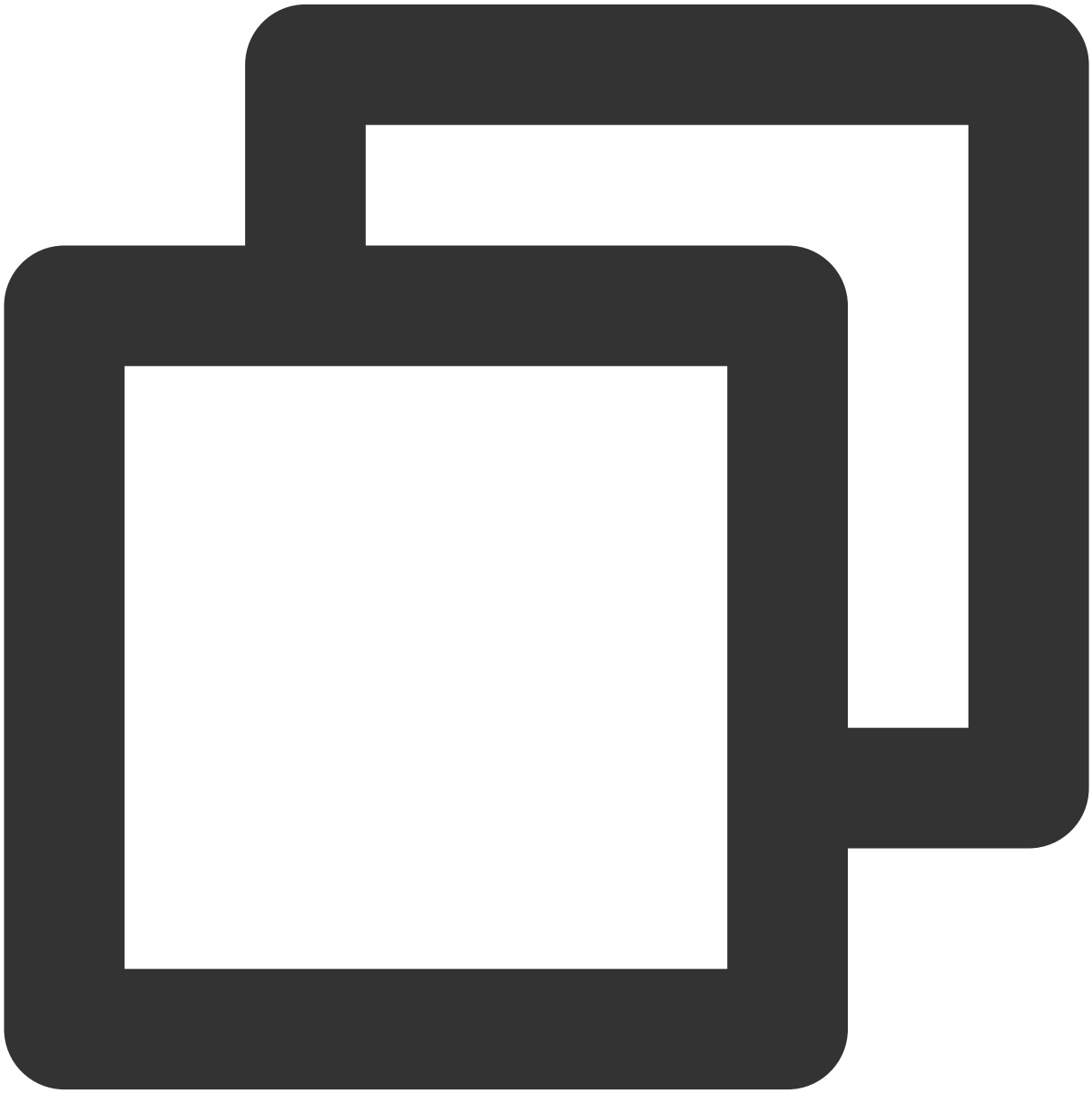
最近更新时间：2024-01-23 10:00:02

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：

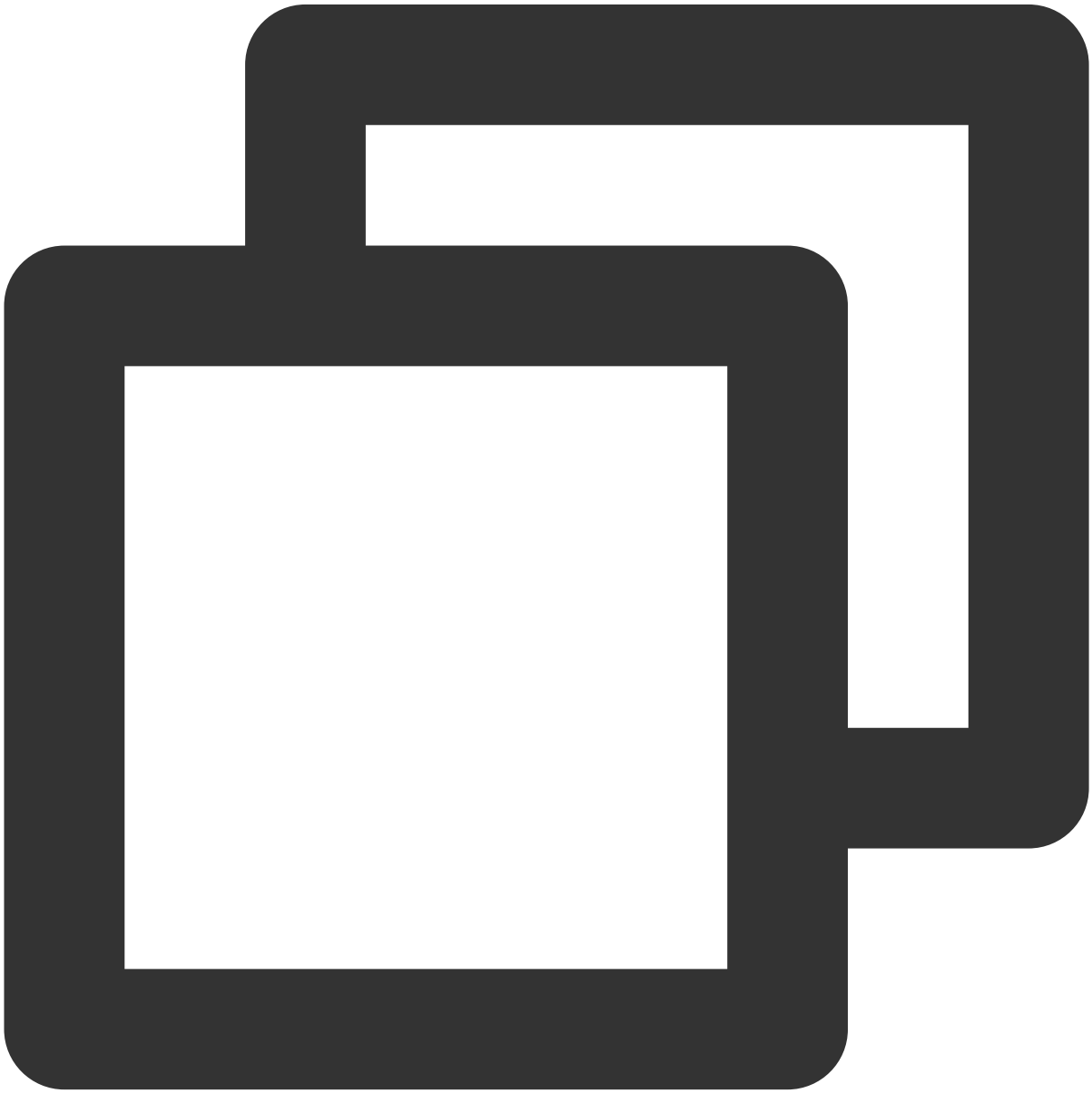
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

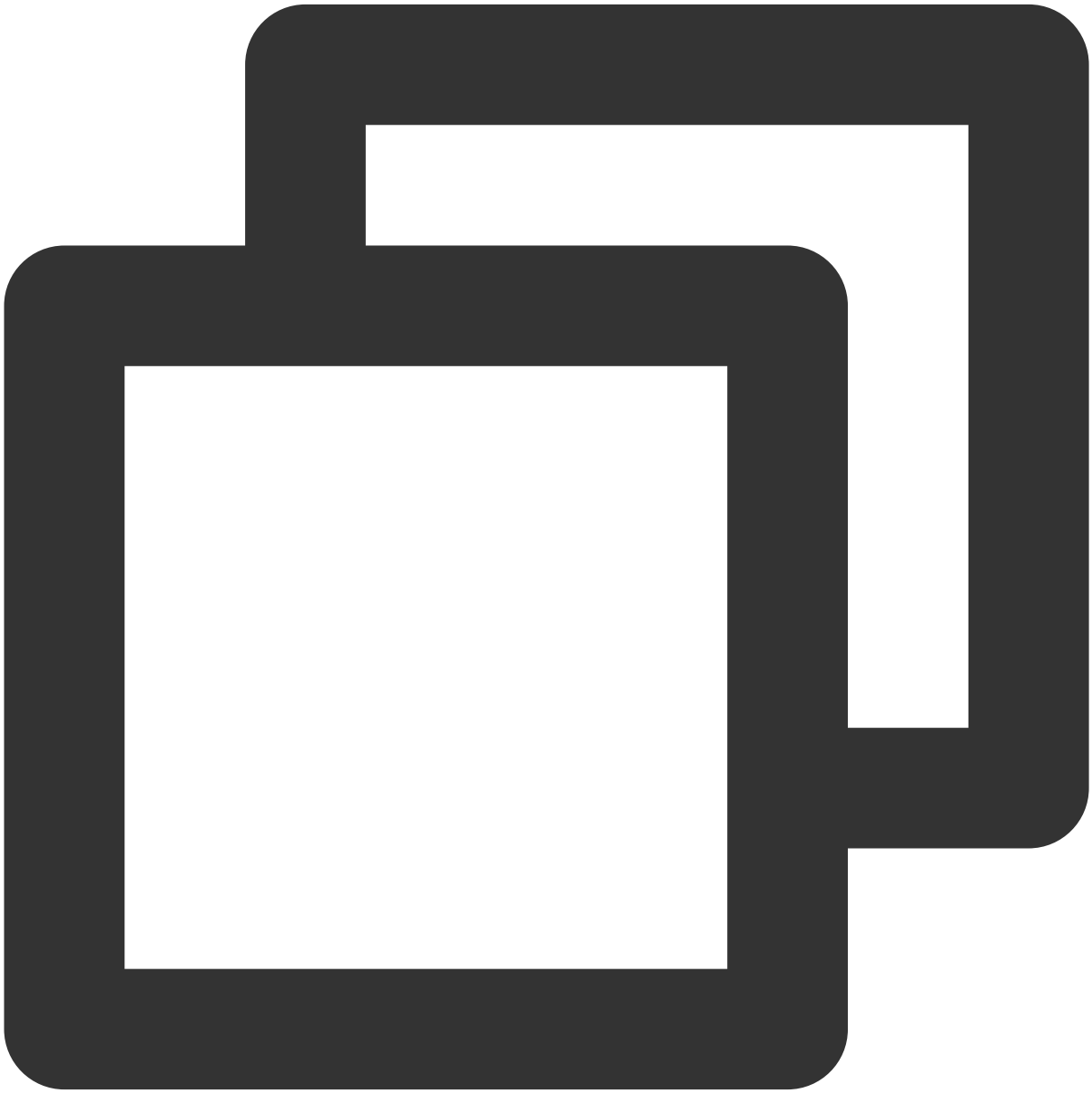
响应

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：

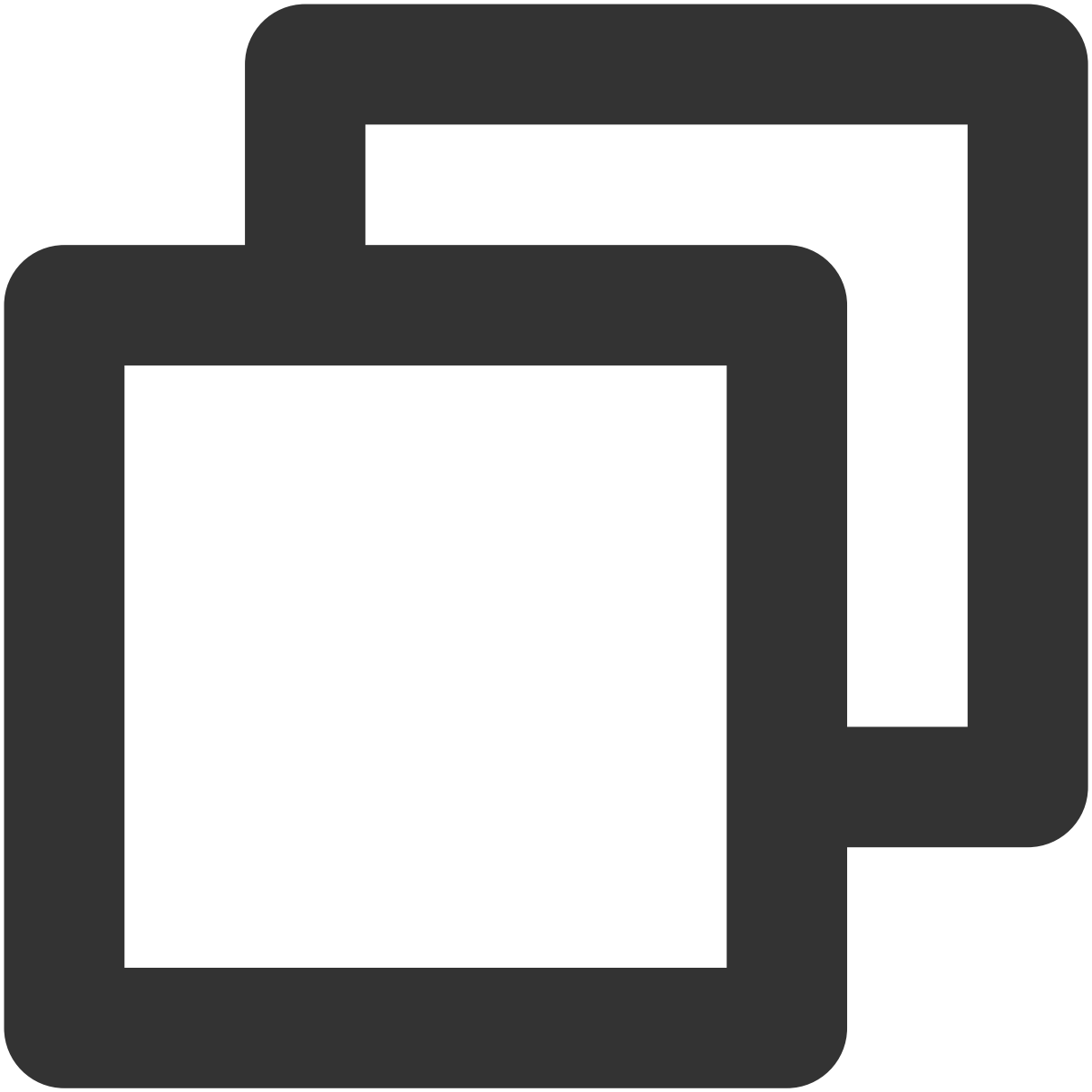
节点名称（关键字）	父节点	描述	类型
Response	无	同创建语音合成模板接口的 Response	Container

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

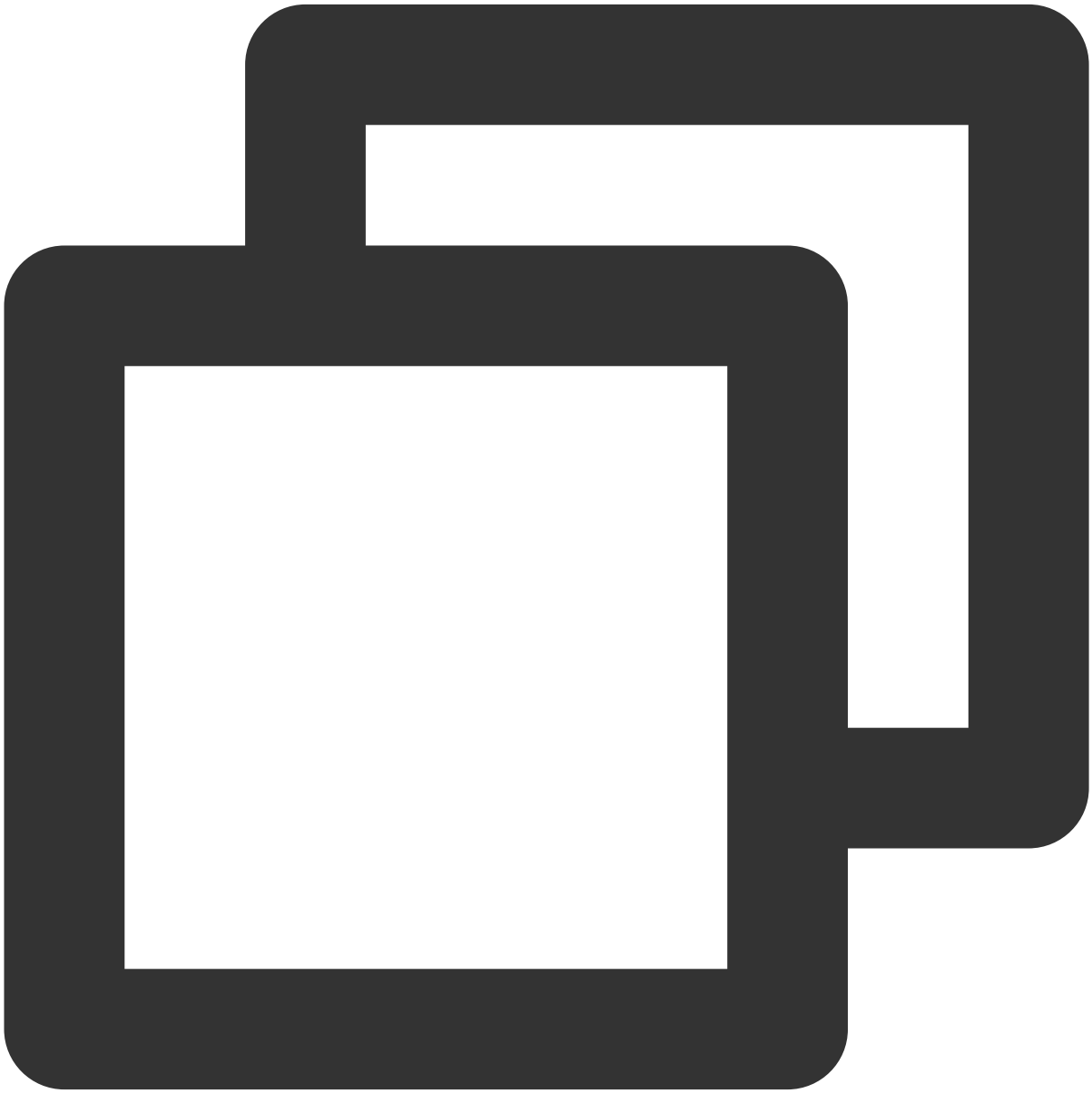


```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-si
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<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）

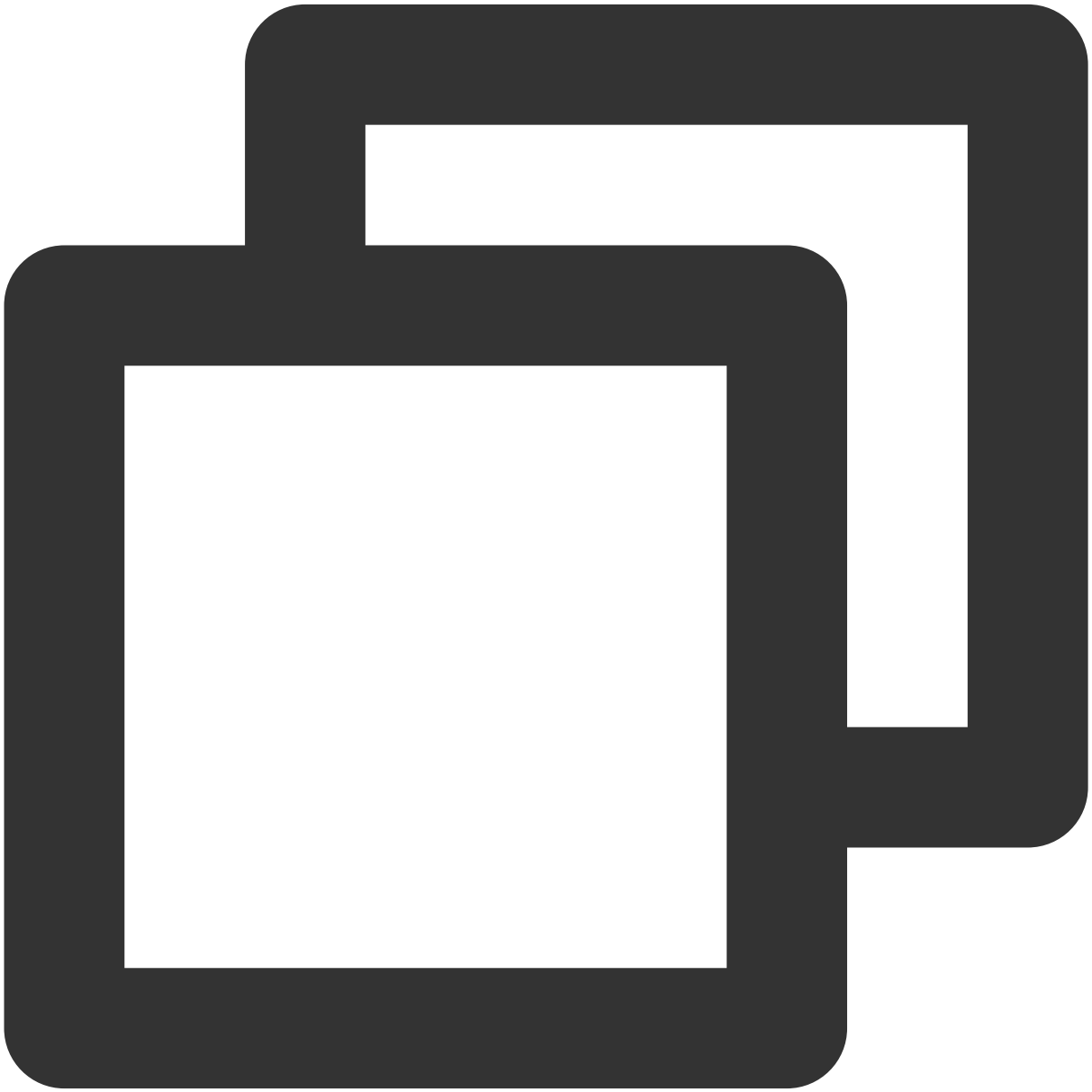
最近更新时间：2024-01-23 10:00:02

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）。

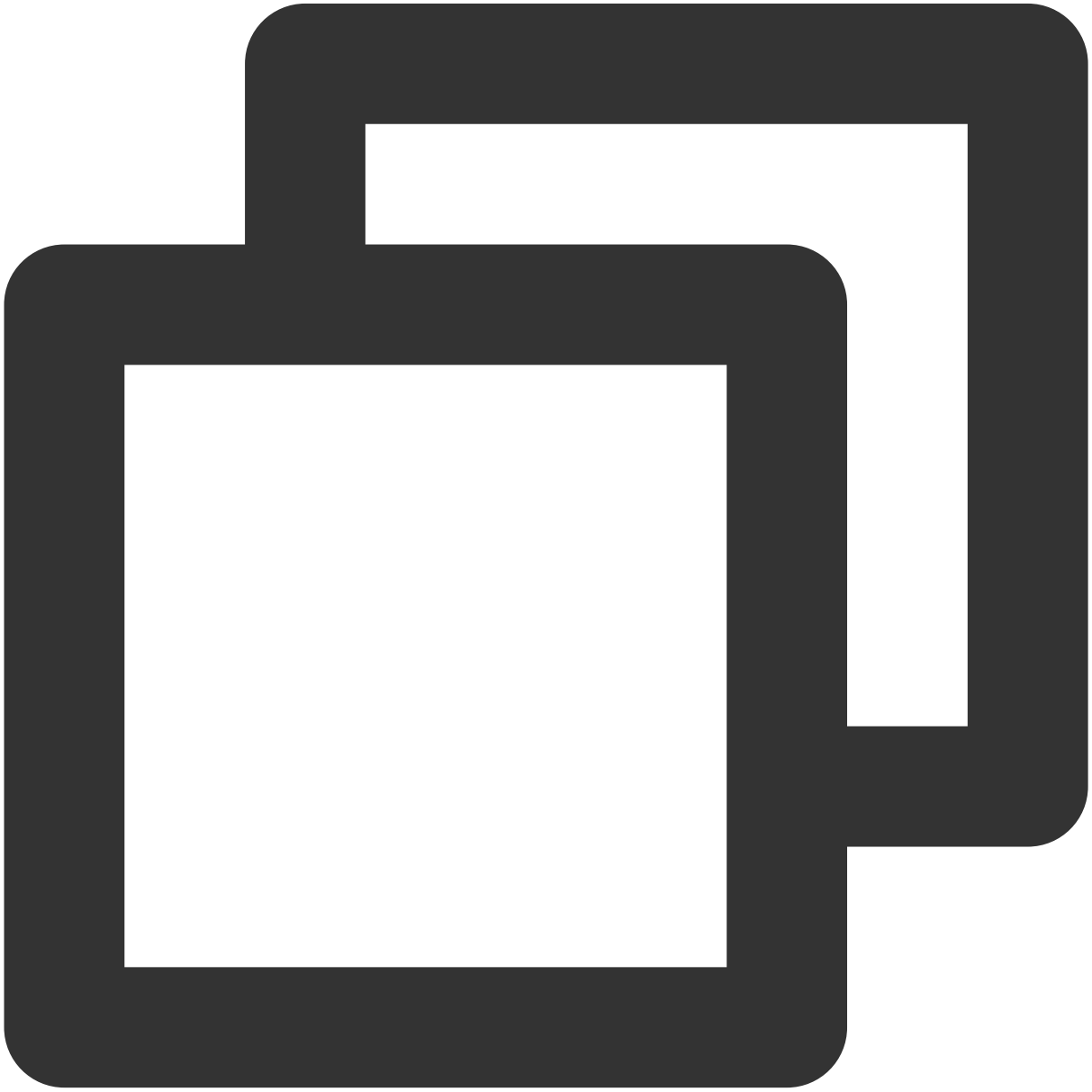
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

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
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

版权所有：腾讯云计算（北京）有限责任公司

CallBackMqConfig	Request.Operation	任务回调 TDMQ 配置，当 CallBackType 为 TDMQ 时必填	Container	否

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
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

版权所有：腾讯云计算（北京）有限责任公司

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 Sna 、StreamExtract 时，该 当媒体任务类型为SmartC Request.Operation.JobPa 大于1时，必须包含 ${Nu 当媒体任务类型为 Snaps Snapshot.Count 大于1时， 参数
AuObject	Request.Operation.Output	人声结果文件名	String	否	当媒体任务类型为 VideoS 效，且不能和 Object 同时
SpriteObject	Request.Operation.Output	雪碧图名	String	否	当媒体任务类型为 Snaps 且不能和 Object 同时为空 当 Snapshot.Count 大于1 ${Number} 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	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、	String	否

版权所有：腾讯云计算（北京）有限责任公司

		VideoProcess、SuperResolution、SDR2HDR 时该参数有效		
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、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
Snapshot	Request.Operation.JobParam	截图任务参数，当媒体任务类型为 Snapshot 时有效	Container	否

版权所有：腾讯云计算（北京）有限责任公司

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
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。

版权所有：腾讯云计算（北京）有限责任公司

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

Container 类型 SpeechRecognition 的内容：

同 提交语音识别任务 的 Request.Operation.SpeechRecognition

版权所有：腾讯云计算（北京）有限责任公司

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

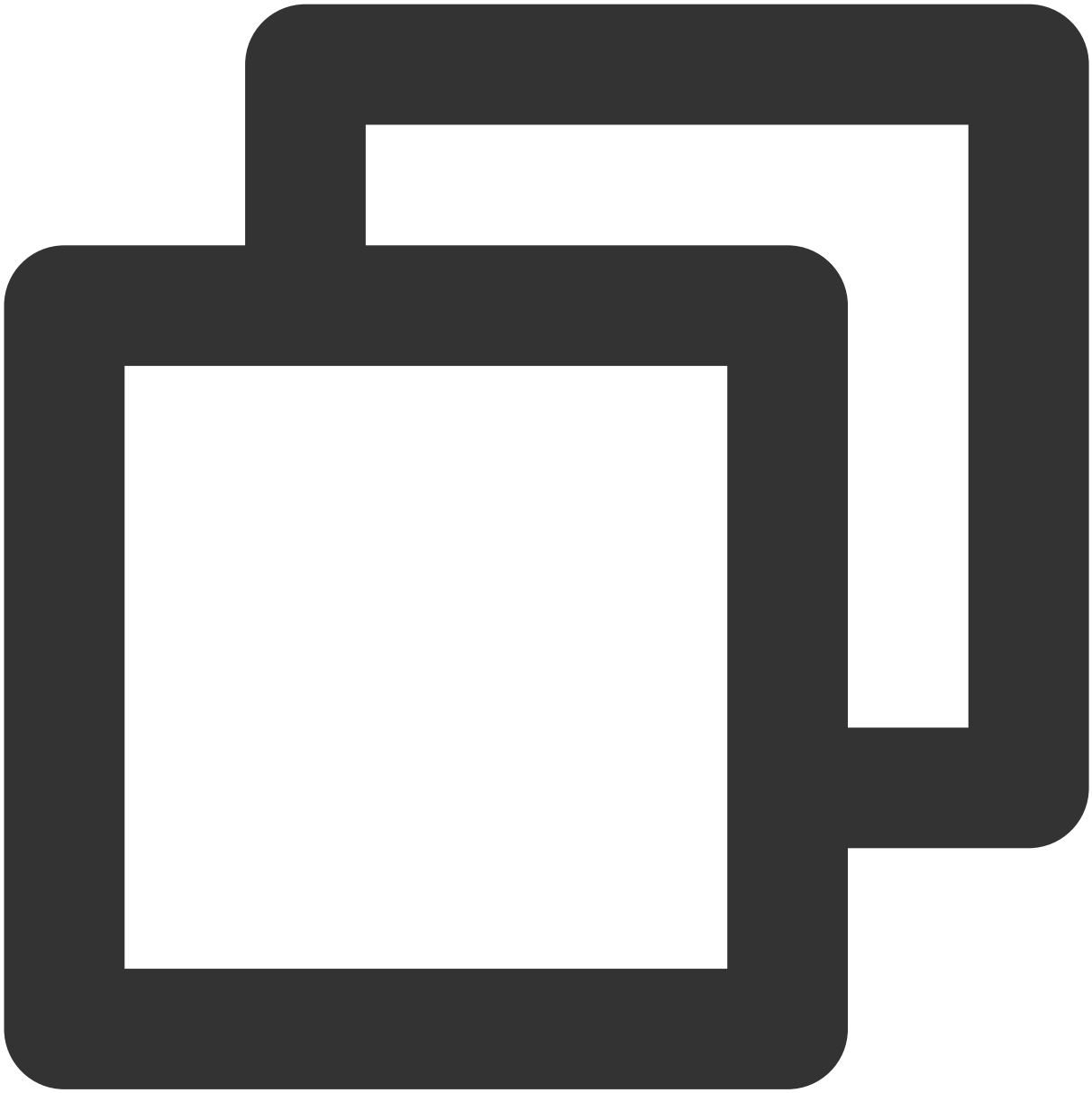
响应

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

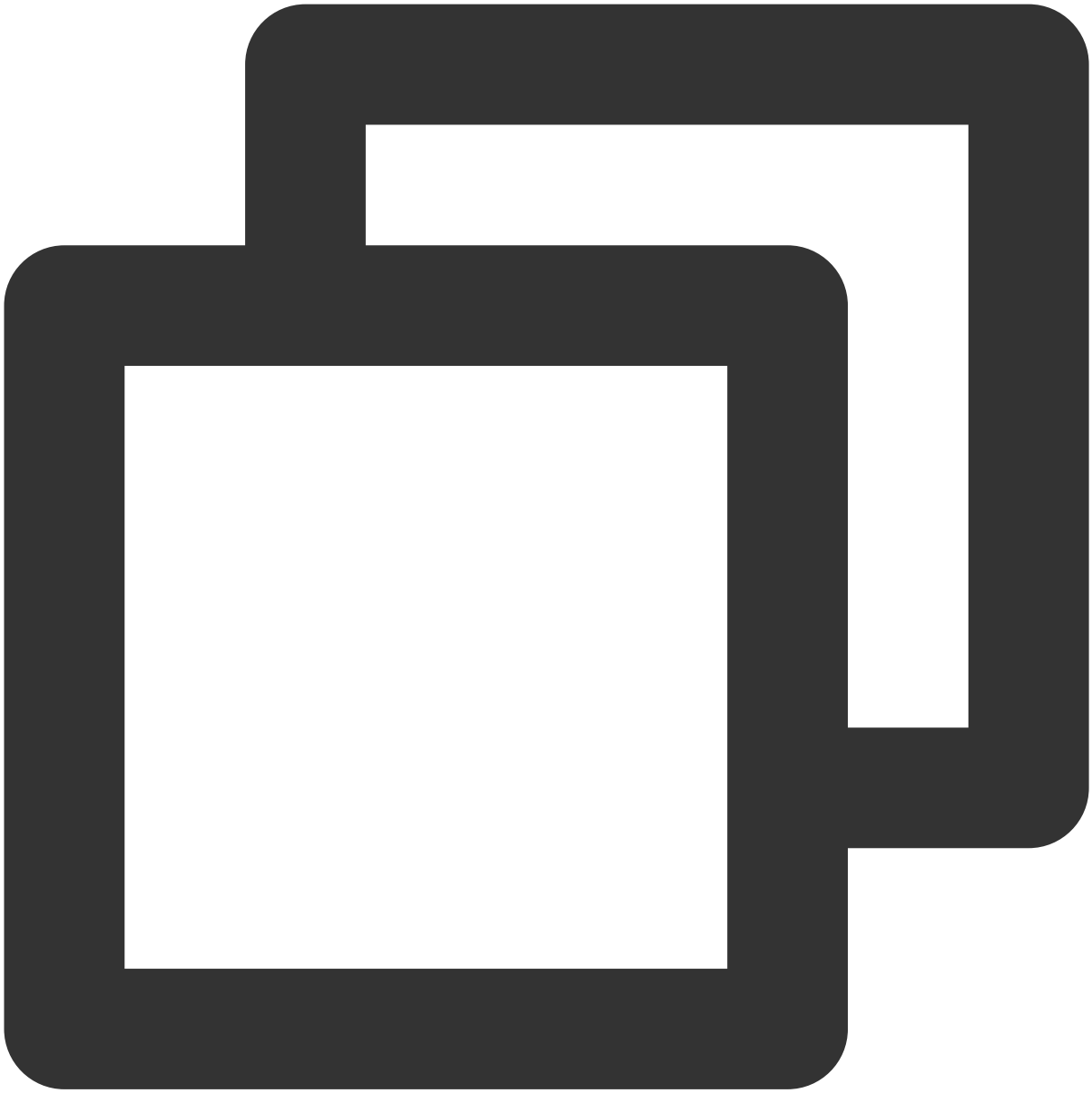
Message	Response.JobsDetail	错误描述，只有 State为 Failed 时有意义	String
Type	Response.JobsDetail	批量处理任务类型，Job	String
Name	Response.JobsDetail	新创建任务的名称	String
JobId	Response.JobsDetail	新创建任务的 ID	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	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

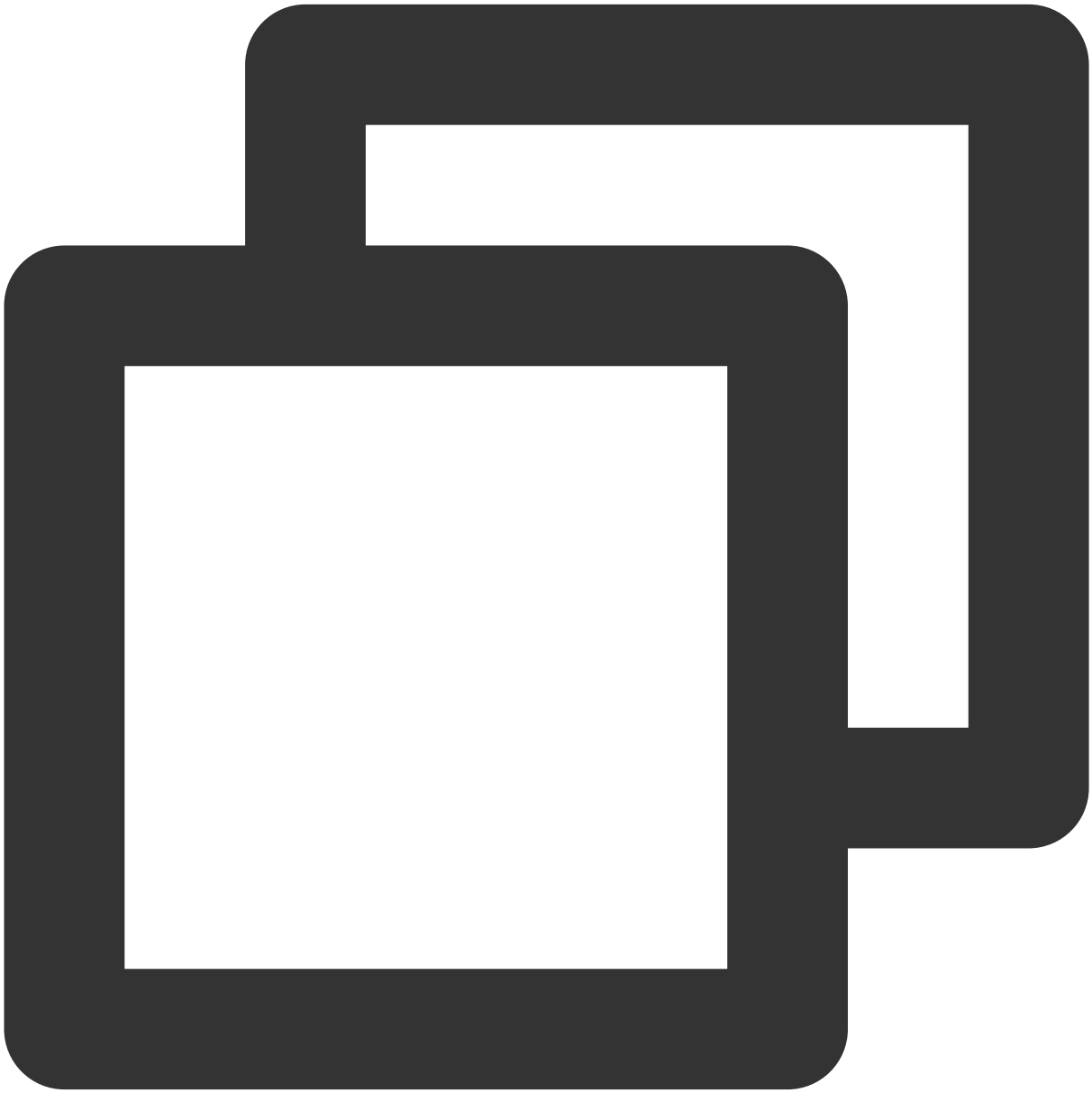


```
POST /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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>
```

触发任务（工作流）

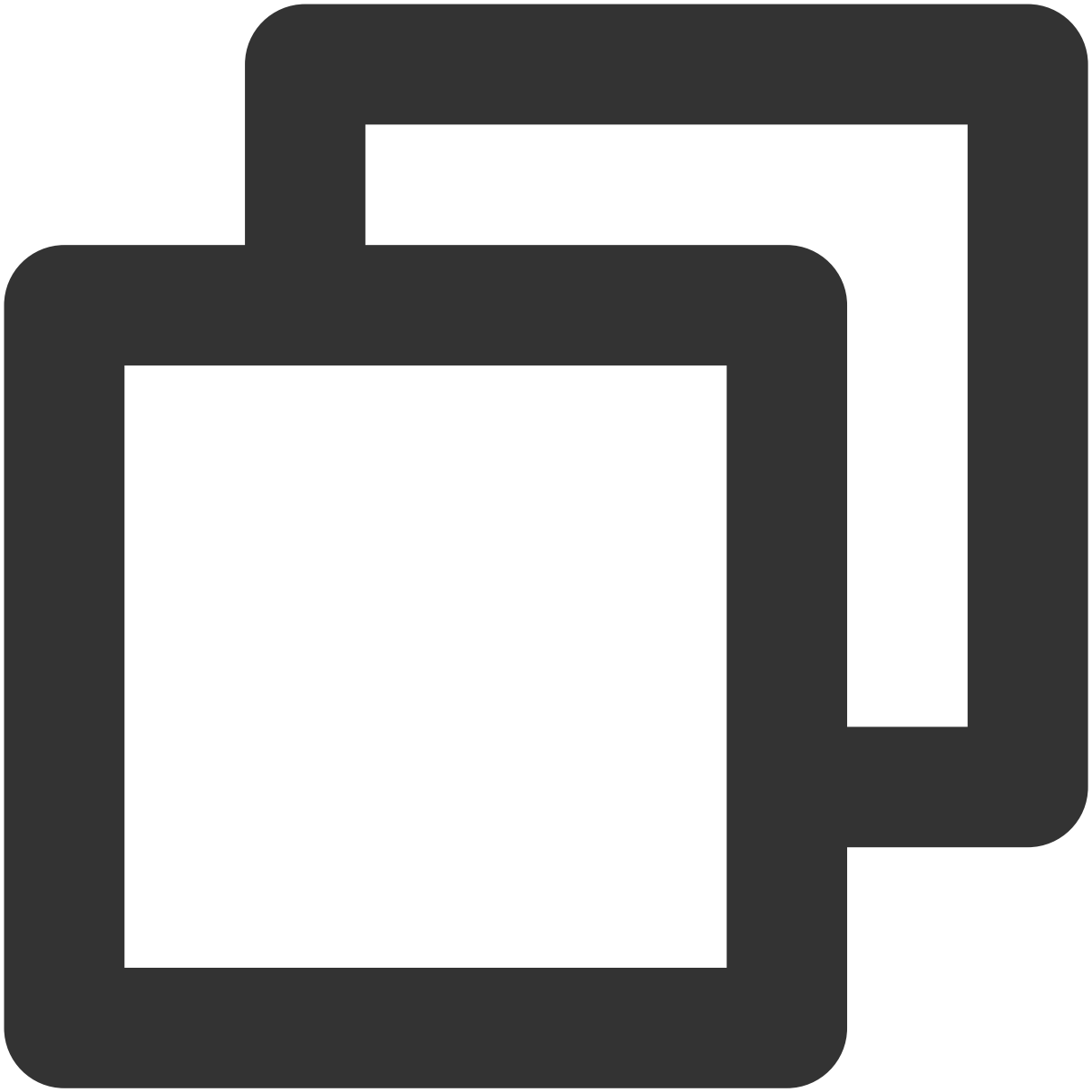
最近更新时间：2024-01-23 10:00:02

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：

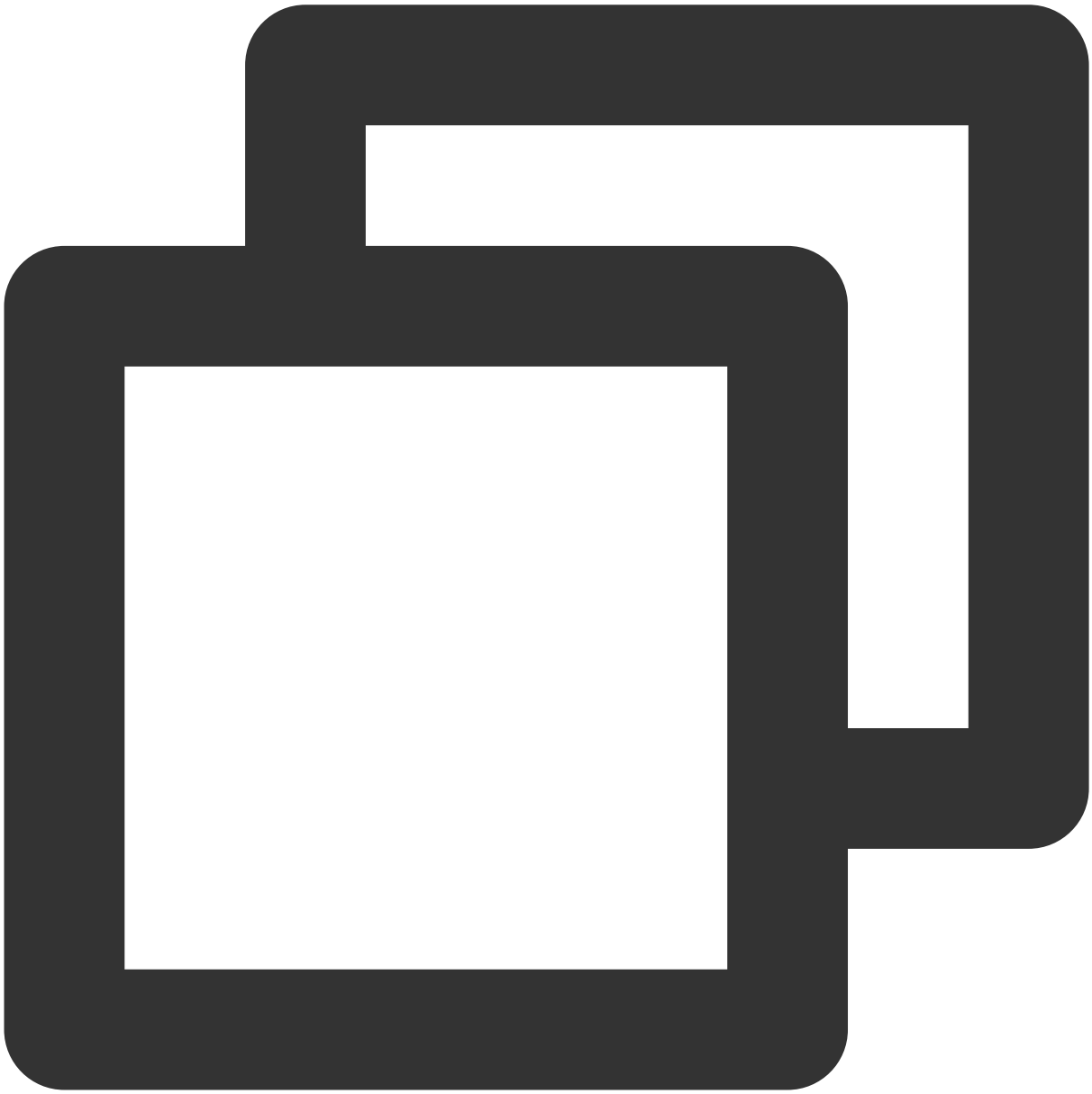
Authorization: Auth String（详情请参见 请求签名 文档）。
通过子账号使用时，需要授予相关的权限，详情请参见 授权粒度详情 文档。

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

版权所有：腾讯云计算（北京）有限责任公司

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
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

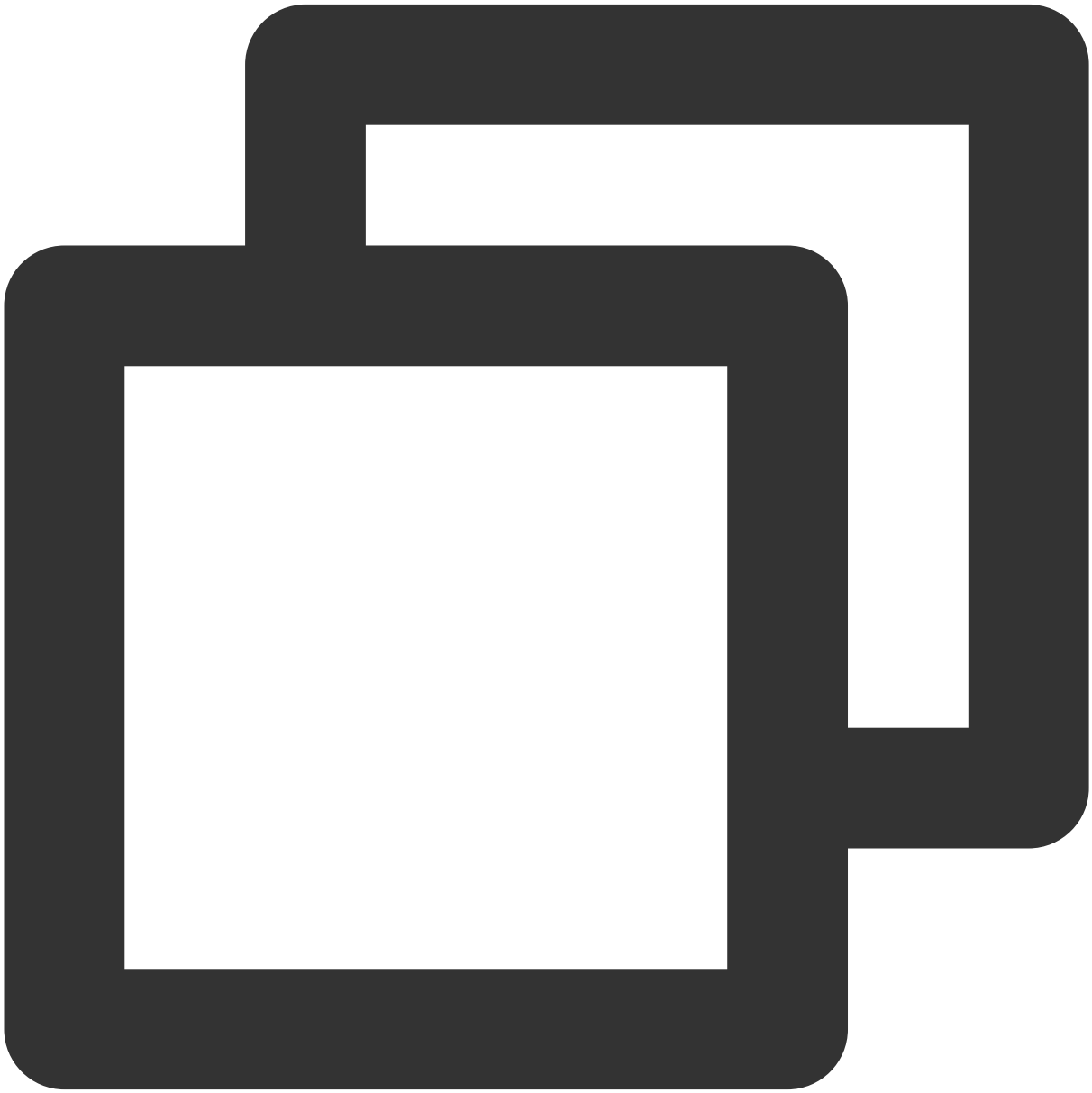
响应

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>
```

```
        <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

版权所有：腾讯云计算（北京）有限责任公司

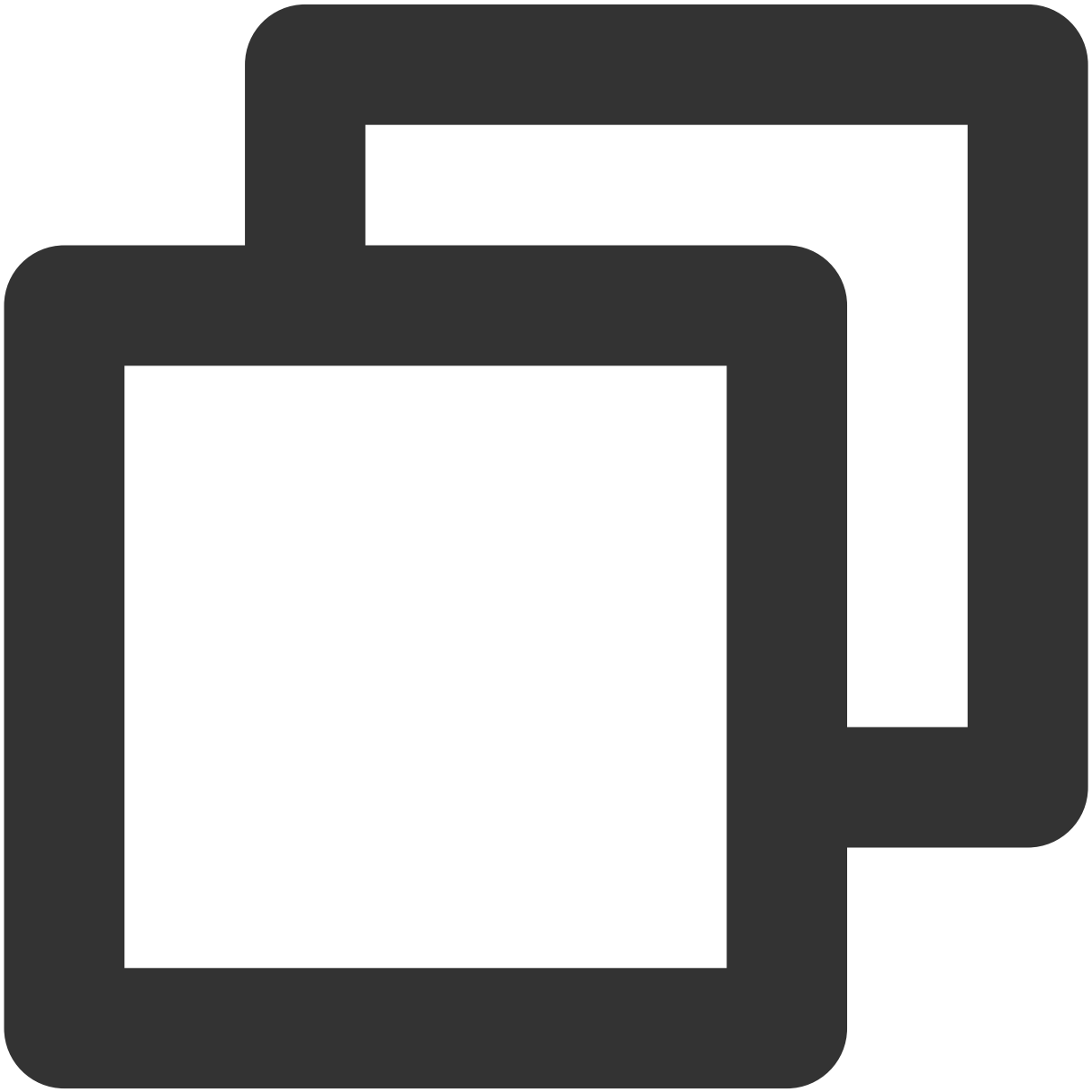
StartTime	Response.JobsDetail	任务的开始时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
Input	Response.JobsDetail	同请求中的 Request.Input	Container
Operation	Response.JobsDetail	同请求中的 Request.Operation	Container

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

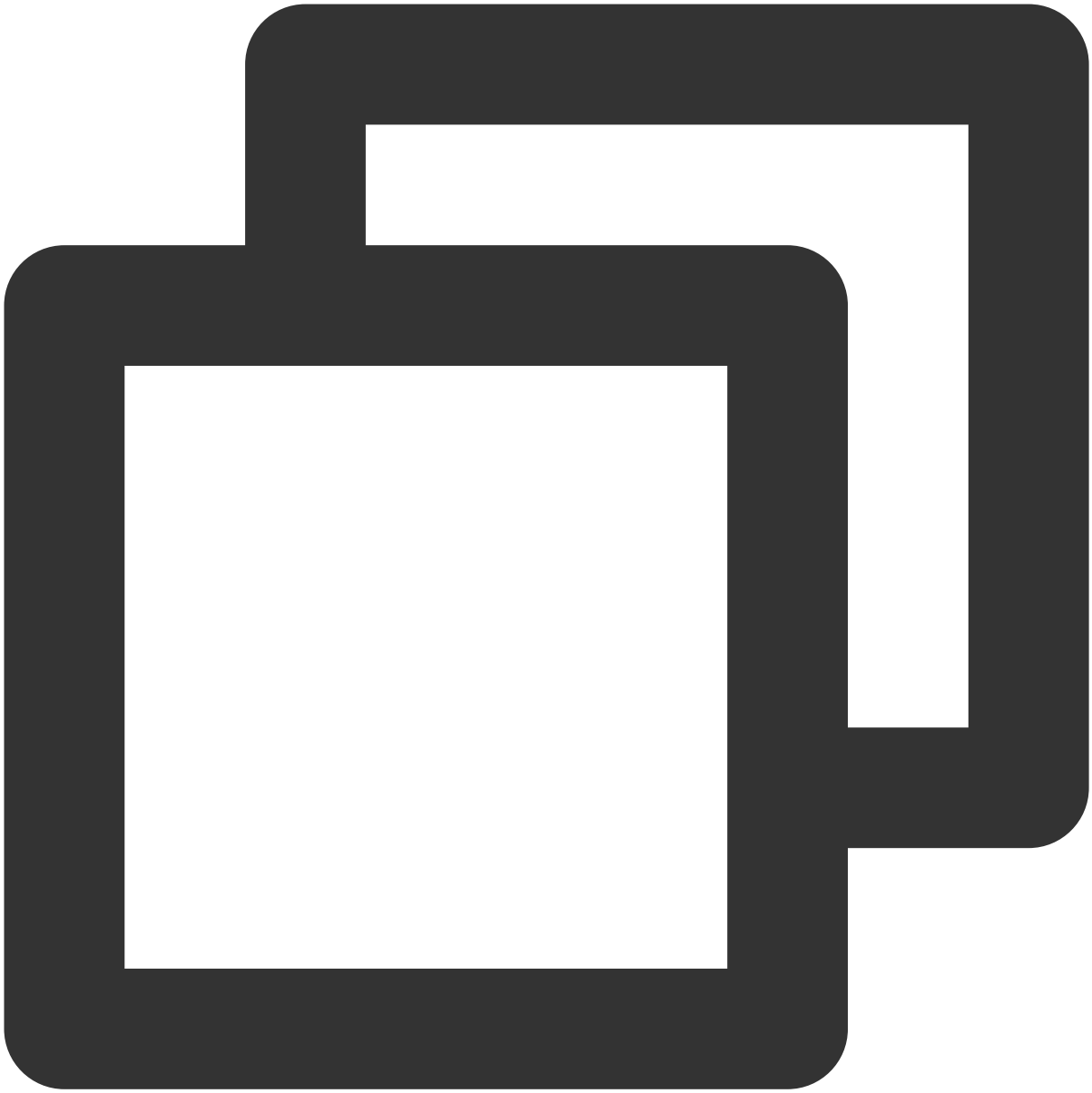


```
POST /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-sig
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<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>
```

```
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

取消任务

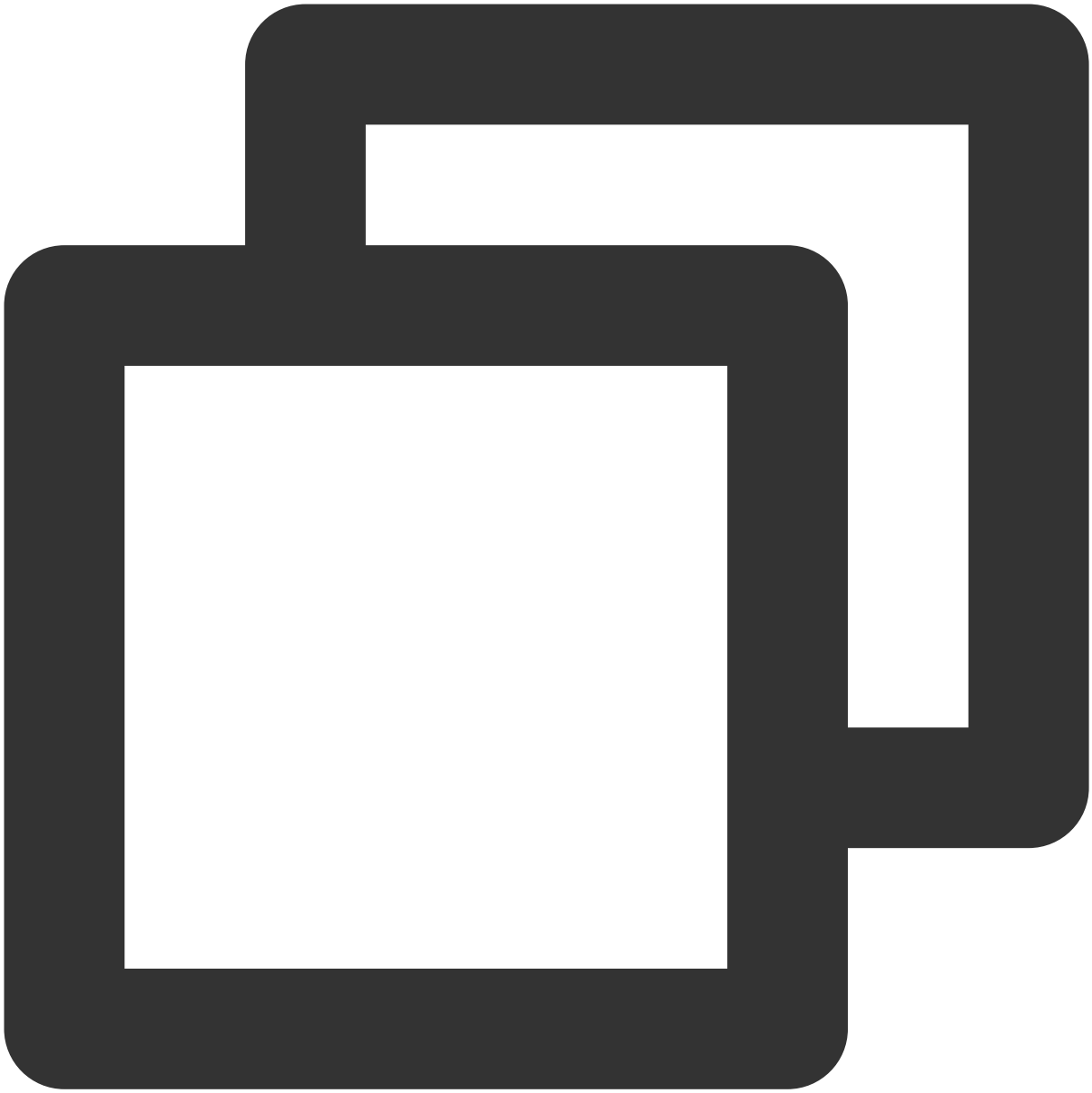
最近更新时间：2024-01-23 10:00:02

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

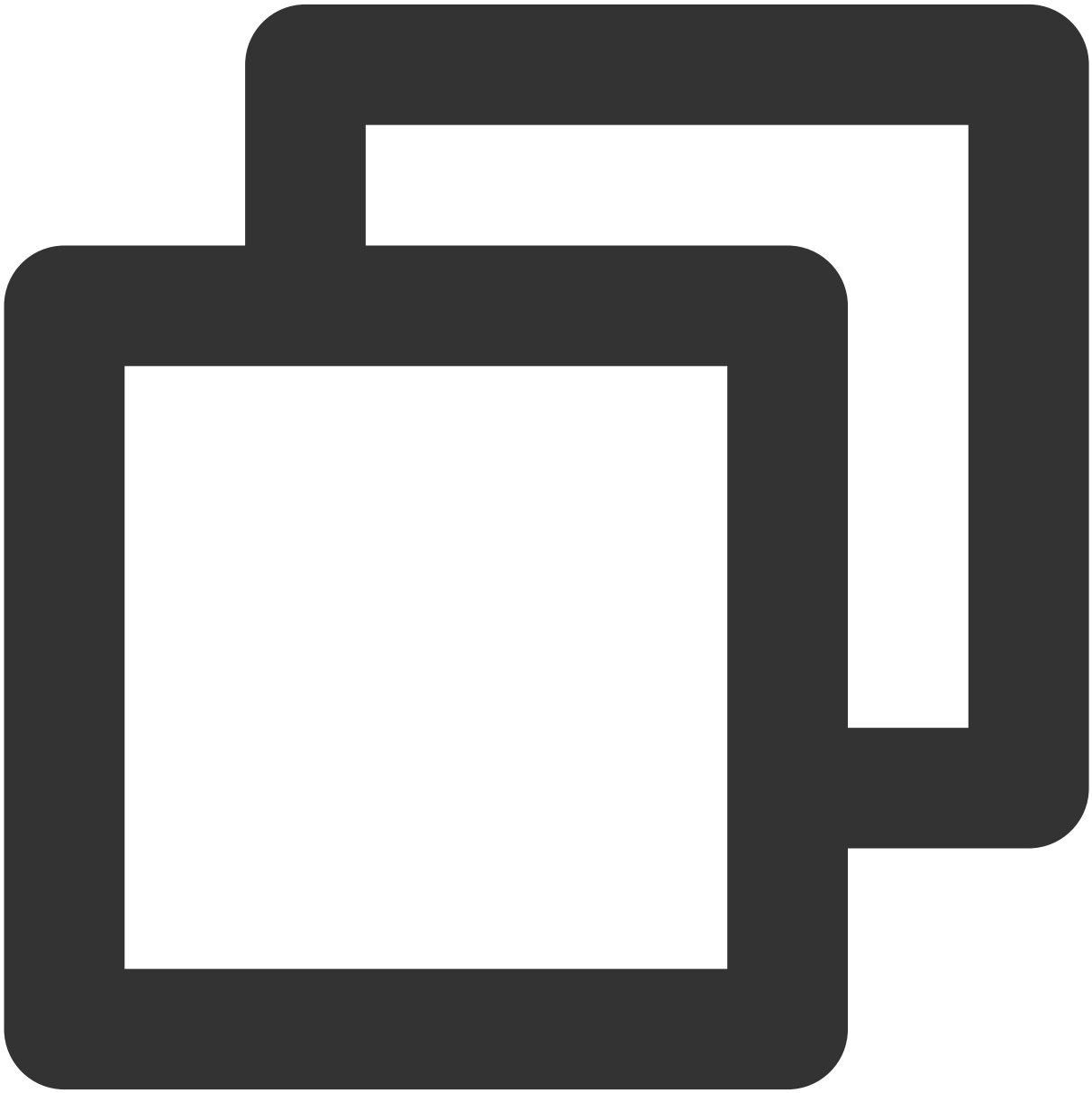
该请求无响应体。

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。

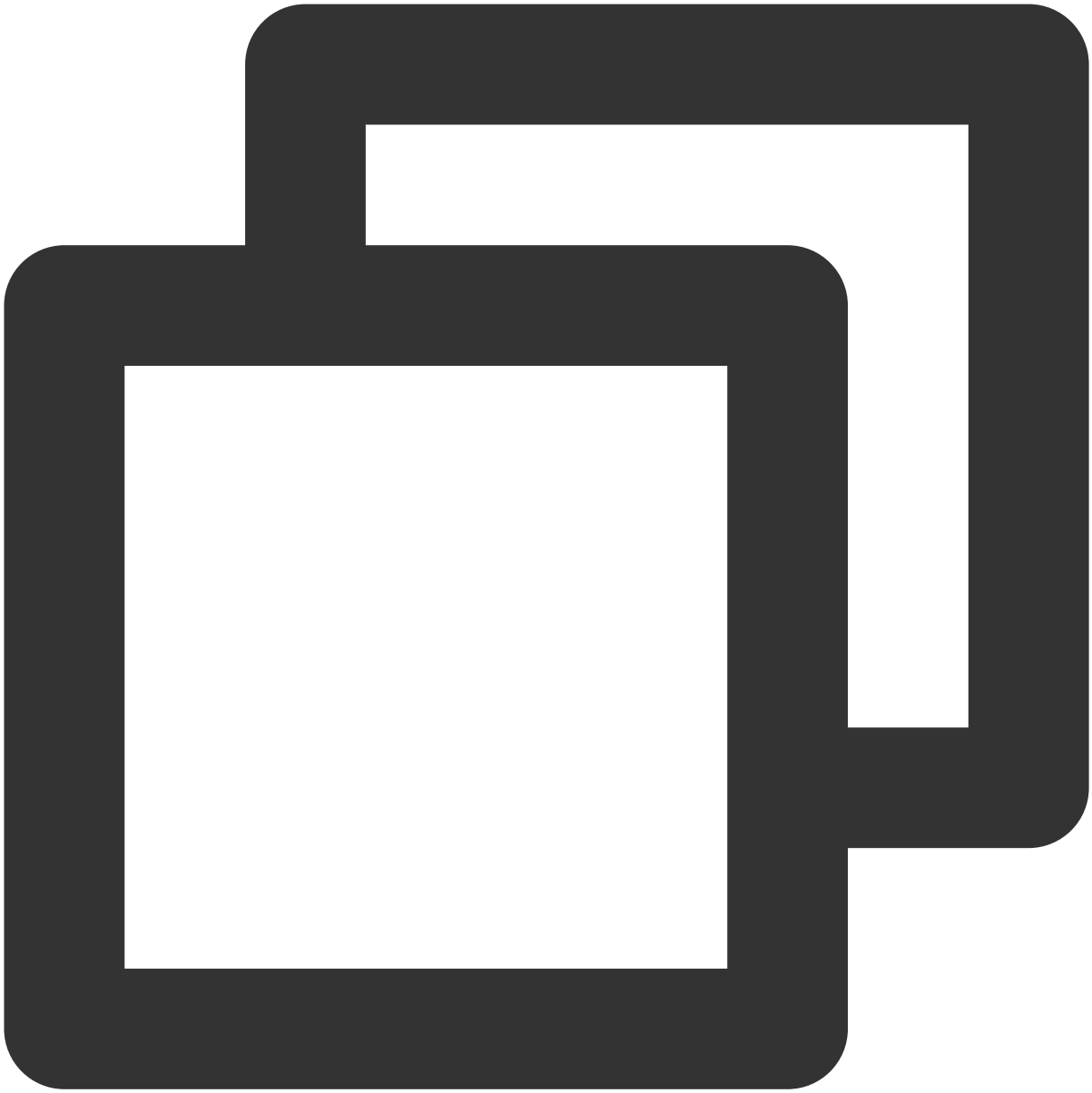
实际案例

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

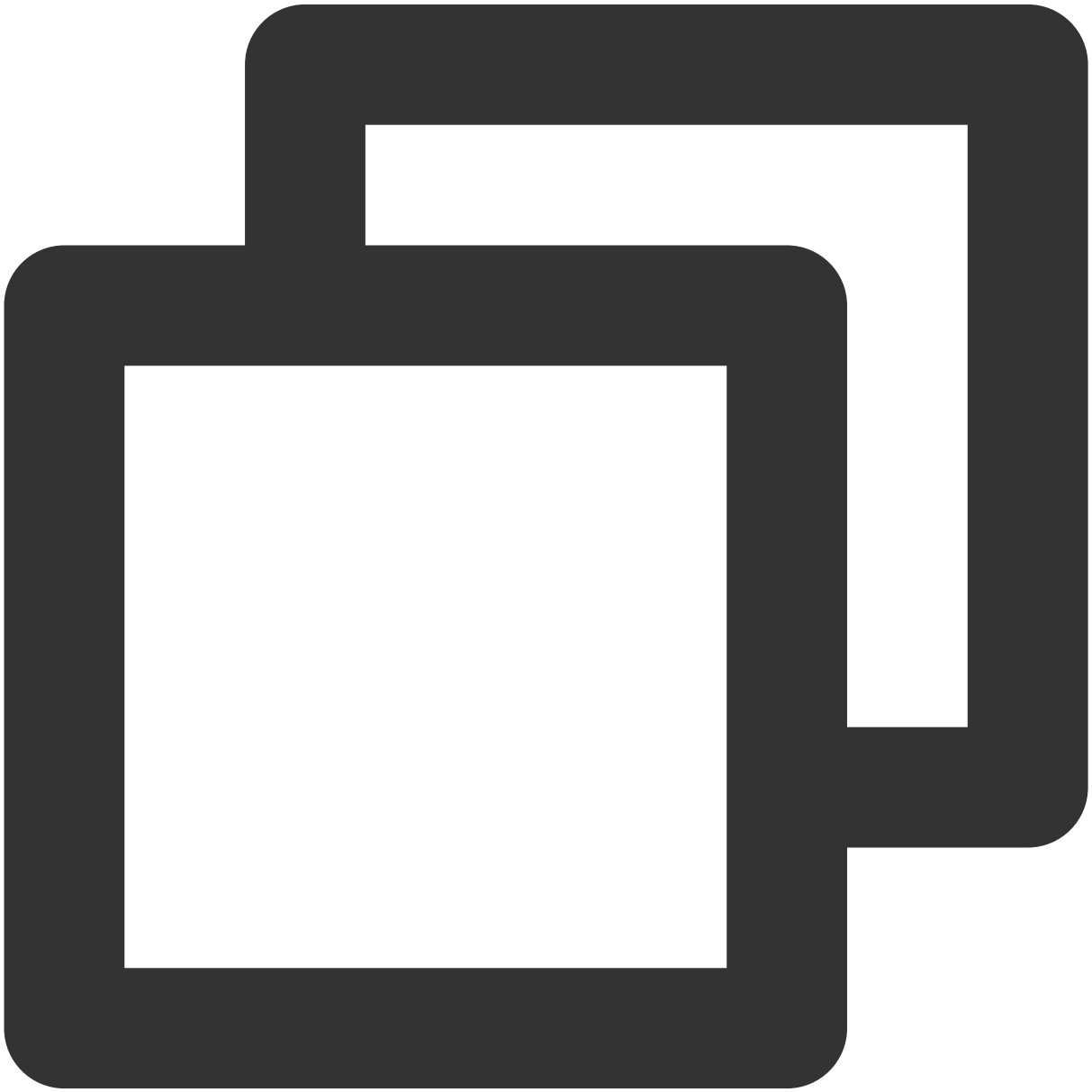
最近更新时间：2024-03-01 18:06:24

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。

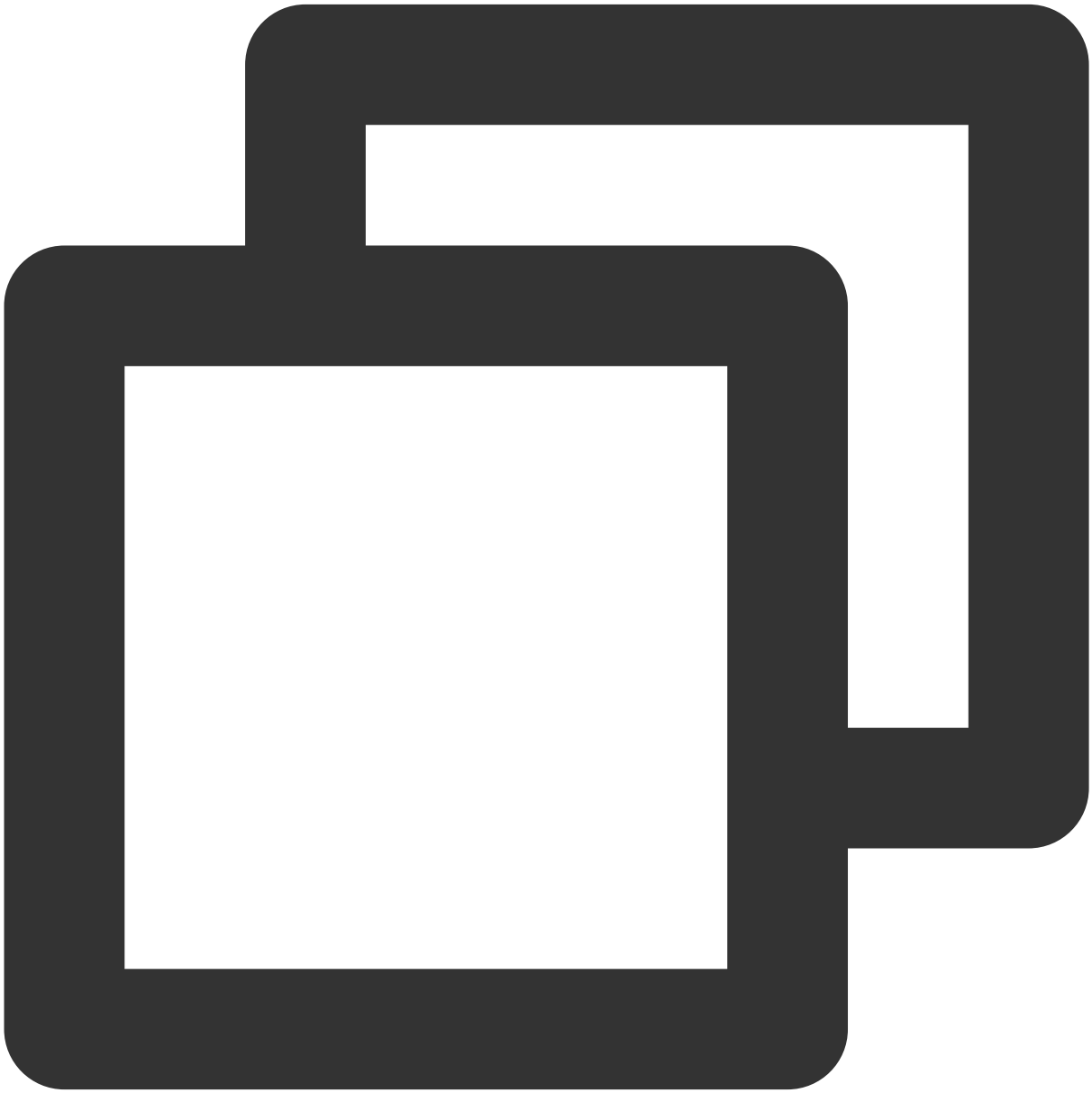
响应

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

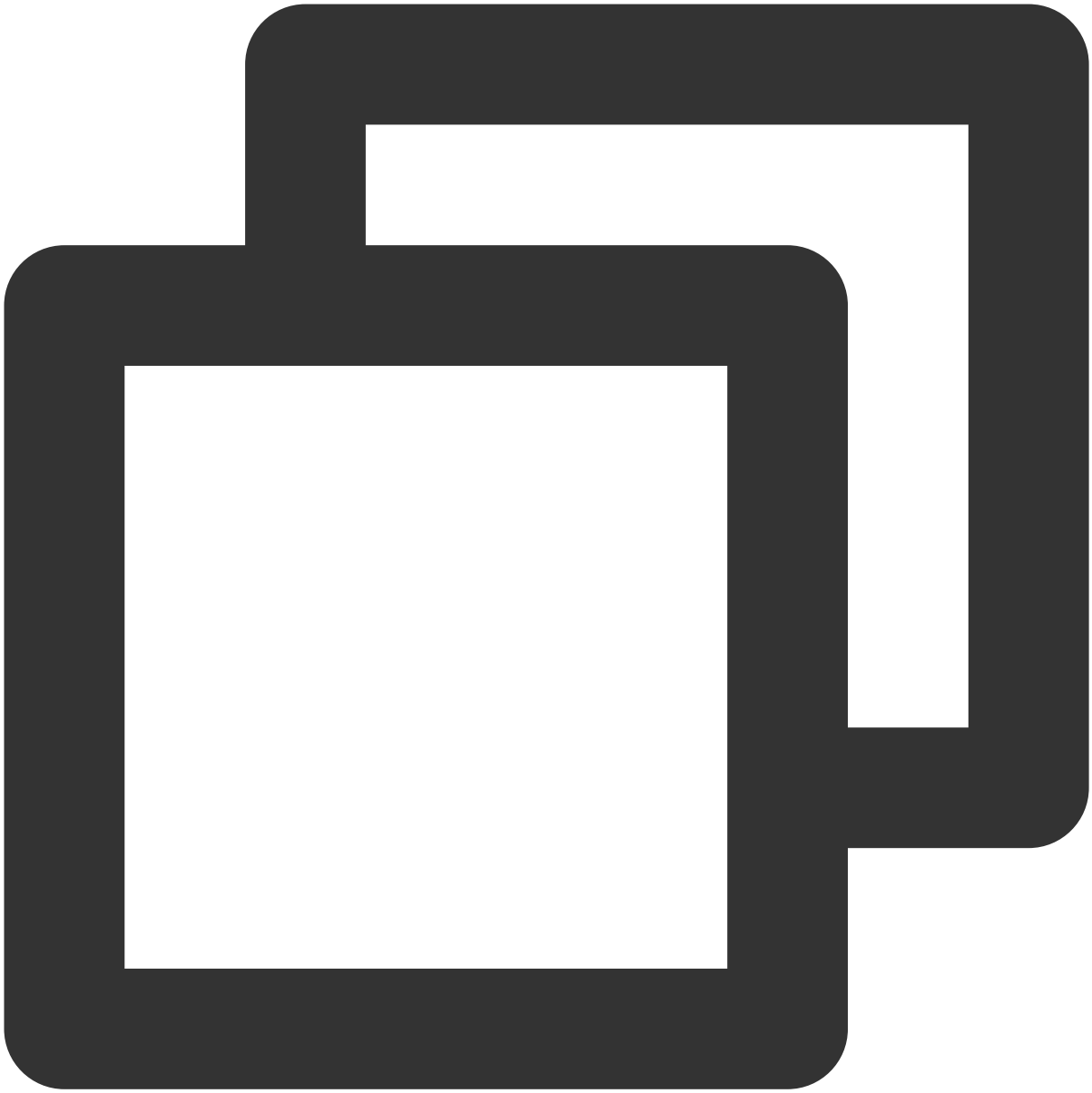
对于不同的触发方式，JobsDetail 的内容不同，请参照以下链接：
工作流

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。

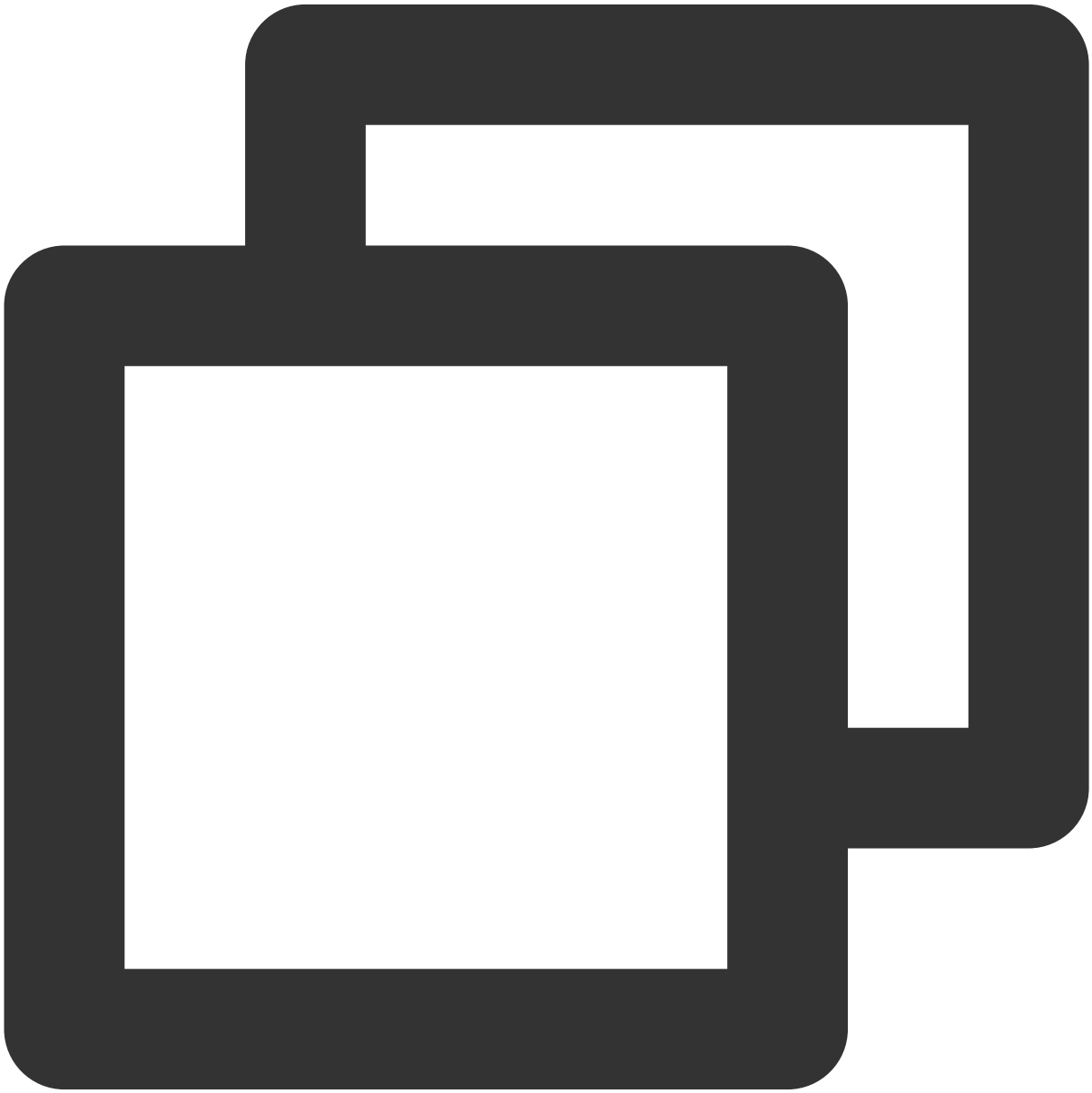
实际案例

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

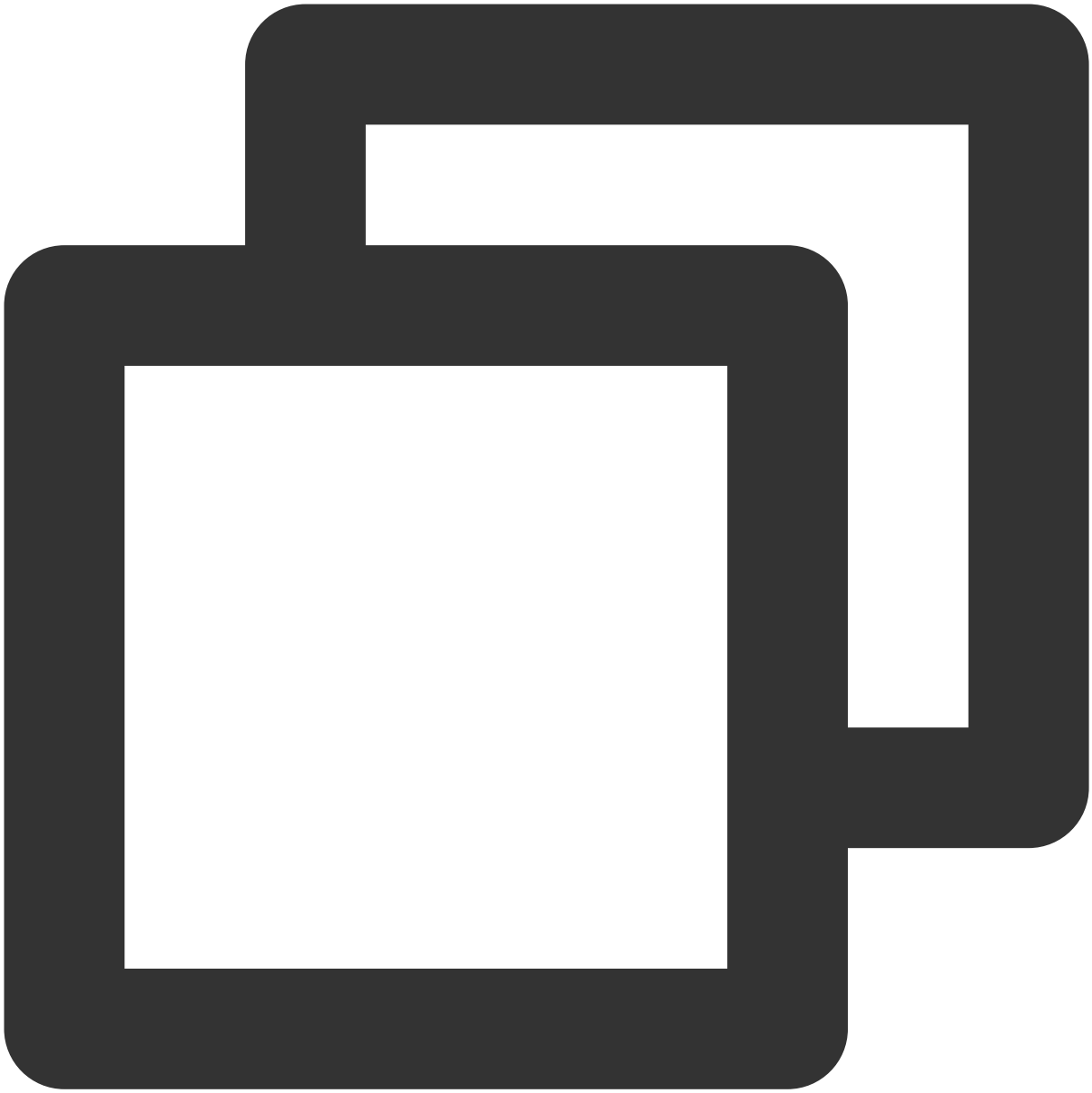
最近更新时间：2024-01-23 10:00:02

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

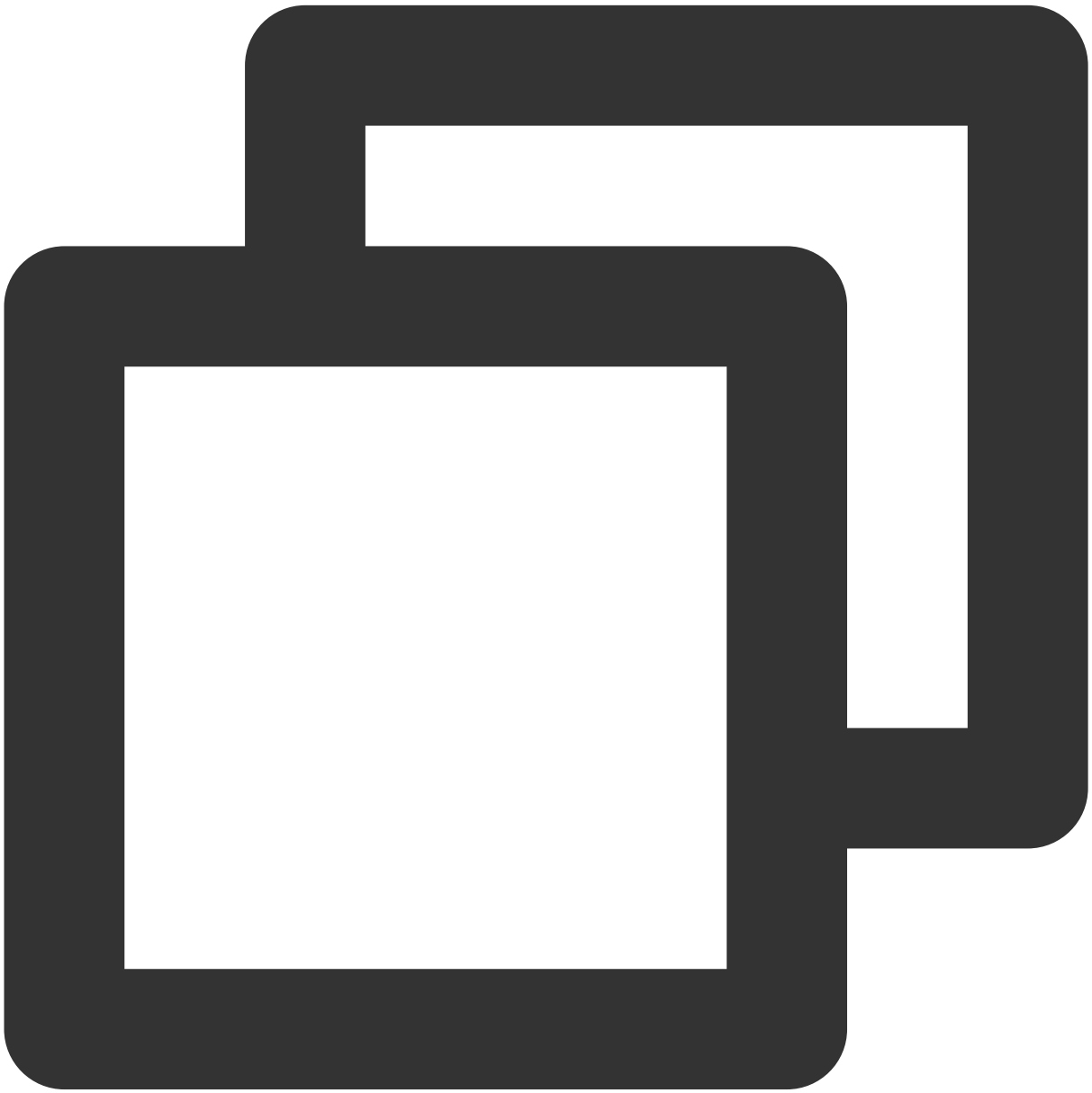
响应头

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

版权所有：腾讯云计算（北京）有限责任公司

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

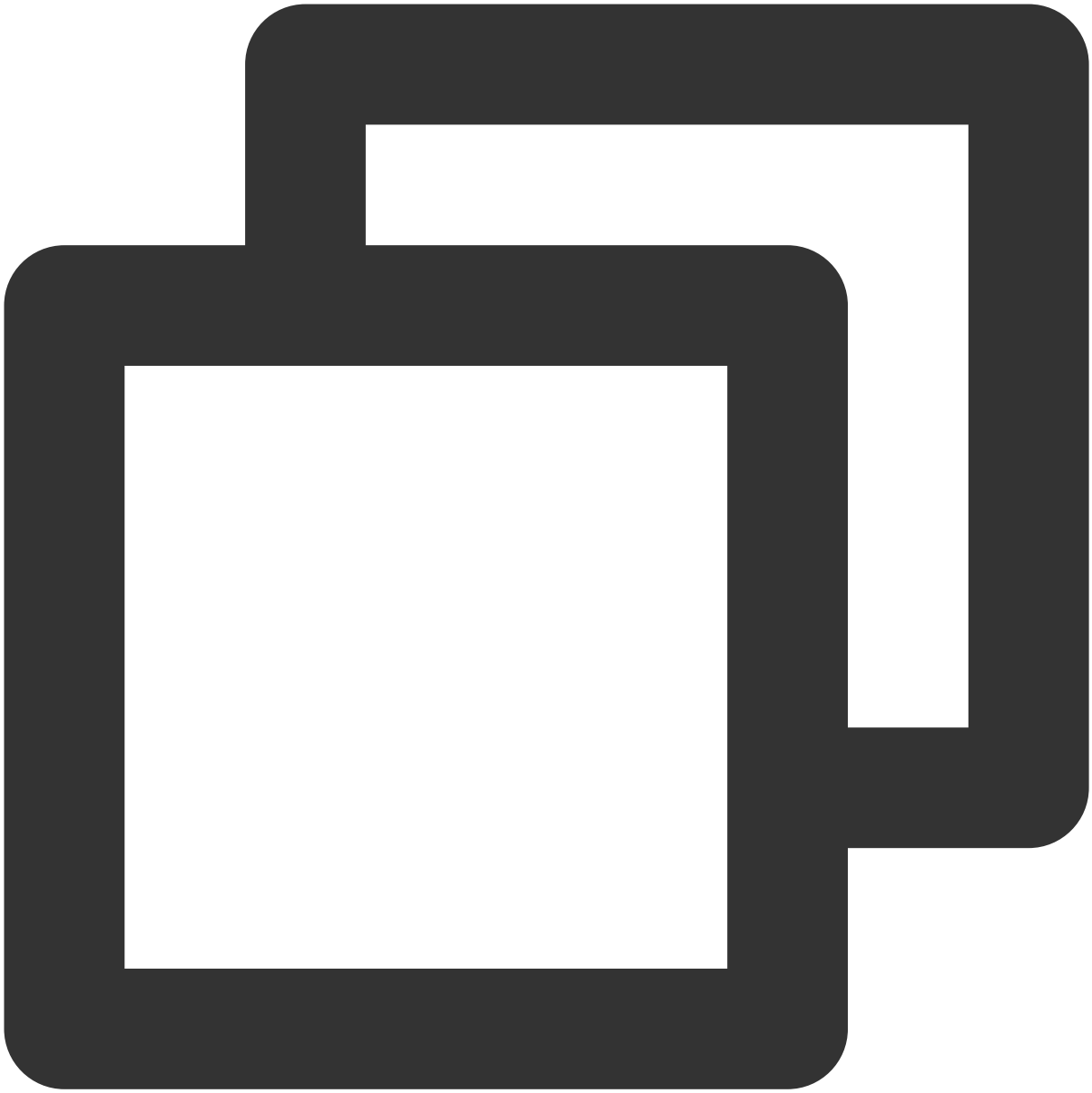
对于不同的触发方式，JobsDetail 的内容不同，请参照以下链接：
独立任务
工作流

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。

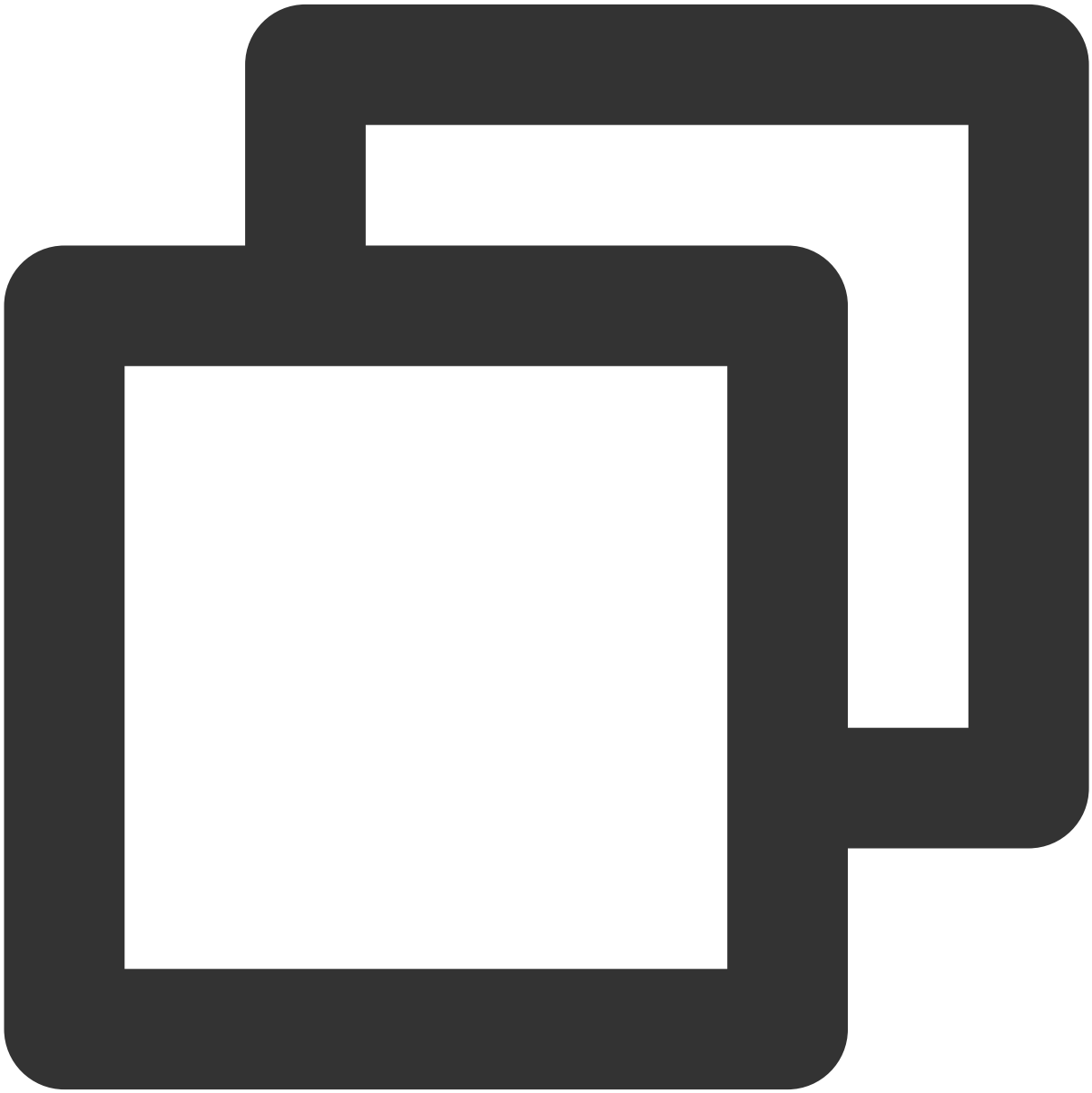
实际案例

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>
```

错误码

最近更新时间：2024-01-23 10:00:02

功能说明

图片下载失败时，返回包 Headers 中的 X-ErrNo 为错误码字段，请参考下表进行处理。

说明：

其他透传自 COS 下载侧错误码详见 COS 错误码。

错误码列表

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	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

	量可能设置错误。	
-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。
-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。

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。

版权所有：腾讯云计算（北京）有限责任公司

-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。
-3015	请求的页数超过了文档总页数	请检查请求页数是否正确。

内容审核错误码

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。

媒体处理错误码

错误码	错误信息	说明
1007	InternalError	内部错误
1010	InvalidUrl	无效的 url
1011	InvalidContent	无效的请求体
1012	MissingAuthorization	缺失签名
1013	InvalidAuthorization	无效的签名
1014	SignatureExpired	签名已过期
1015	InvalidArgument	无效的参数
1016	MissingContent	缺失请求体
1017	InvalidHost	无效的 host
1018	AccessDenied	拒绝访问
1019	SecretidNotExist	密钥不存在

版权所有：腾讯云计算（北京）有限责任公司

1020	SlowDown	频率控制
1021	InvalidContentType	请求体类型无效
1100	BucketAlreadyBinded	存储桶已绑定
1103	ResourceNotFound	资源未找到
1106	QueueNotFound	队列未找到
1107	TemplateNotFound	模板未找到
1109	BindedBucketNumLimit	绑定的存储桶数量超过上限
1110	CreatedTemplateNumLimit	创建的模板数量超过上限
1111	MediaBucketUnBinded	存储桶未绑定
1112	TemplateNameDuplicate	模板命名重复
1114	ActiveWorkflowForbiddenUpdate	开启的工作流禁止更新
1120	WorkFlowNumLimit	创建的工作流数量超过上限
1124	WriteTargetBucketDenied	禁止写入存储桶
1145	QueueNumLimit	创建的队列数量超过上限

版权所有：腾讯云计算（北京）有限责任公司

附录
结构体

最近更新时间：2024-03-05 14:13:34

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

版权所有：腾讯云计算（北京）有限责任公司

					JSON：json 格式
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
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

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

版权所有：腾讯云计算（北京）有限责任公司

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

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

版权所有：腾讯云计算（北京）有限责任公司

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

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

版权所有：腾讯云计算（北京）有限责任公司

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	是否必选	限制
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

版权所有：腾讯云计算（北京）有限责任公司

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

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	是	无

		参见 Output		

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	String	否	false	true/false

	原音频				
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，该参数才有意义

版权所有：腾讯云计算（北京）有限责任公司

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