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】

©2013-2023 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

API 文档
- API概览
- 开通服务
  - 绑定数据万象服务
  - 查询数据万象服务
  - 解绑数据万象服务
- 存储桶配置
  - 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
  - 云查毒接口
    - 提交病毒检测任务
    - 查询病毒检测任务结果
- 图片处理
  - 图片处理机制介绍
  - 图片样式
    - 增加样式
    - 查询样式
    - 删除样式
  - 基础图片处理
    - 缩放
    - 裁剪
    - 旋转
    - 格式转换
    - 质量变换
    - 渐进显示
    - 动图截帧
    - 高斯模糊
    - 亮度
    - 对比度
    - 锐化
    - 灰度图
    - 获取图片基本信息

版权所有：腾讯云计算（北京）有限责任公司

获取图片 EXIF
获取图片主色调
去除元信息
快速缩略模板
限制图片大小
管道操作符
支持字体列表
合成动图
获取原图
图片基础压缩（WebP）
图片高级压缩
HEIF 压缩
TPG 压缩
AVIF 压缩
SVG 压缩
图片Guetzli 压缩
开通 Guetzli 压缩
查询 Guetzli 状态
关闭 Guetzli 压缩
图片极智压缩
开通极智压缩
查询极智压缩状态
关闭极智压缩
通过API使用极智压缩
图片明水印
图片水印
文字水印
图片盲水印
异常图片检测
异常图片检测同步请求
异常图片检测异步任务
提交任务
任务回调
图片异步处理
提交任务
任务回调
创建模板
更新模板

版权所有：腾讯云计算（北京）有限责任公司

海报合成任务
海报合成模板接口
查询模板接口
查询海报模板
批量拉取海报模板
删除模板接口
上传文件解析模板接口
海报合成任务接口
查询海报合成任务
查询海报合成任务
批量拉取海报合成任务
取消海报合成任务
提交海报合成任务
异步服务和队列配置
开通图片处理（异步）服务
查询图片处理（异步）服务
关闭图片处理（异步）服务
查询图片处理（异步）队列
更新图片处理（异步）队列
媒体处理
媒体处理机制介绍
媒体处理接口列表
服务和队列配置
开通媒体处理服务
查询媒体处理服务
关闭媒体处理服务
搜索媒体处理队列
更新媒体处理队列
获取媒体信息
获取媒体信息同步请求
获取媒体信息异步任务
提交任务
回调信息
获取媒体截帧
读取私有 M3U8
获取私有 m3u8
更新 HLS 解密 Token
视频 DNA

版权所有：腾讯云计算（北京）有限责任公司

- 提交视频 DNA 任务
- 获取 DNA 库中文件列表
- 查询 DNA 库列表

音视频转码
- 提交任务
- 回调信息
- 创建模板
- 创建模板 转码pro（广电专业格式）
- 更新模板
- 更新模板 转码 pro

极速高清转码
- 提交任务
- 回调信息
- 创建模板
- 更新模板

闲时转码
- 提交任务

音视频转封装（音视频分段）
- 提交任务
- 回调信息

音视频流分离
- 提交任务
- 回调信息

音视频拼接
- 提交任务
- 回调信息
- 创建模板
- 更新模板

精彩集锦
- 提交任务
- 回调信息
- 创建模板
- 更新模板

视频截帧
- 提交任务
- 回调信息
- 创建模板
- 更新模板

版权所有：腾讯云计算（北京）有限责任公司

画质增强
提交任务
回调信息
创建模板
更新模板
视频明水印
提交任务
回调信息
创建模板
更新模板
视频标签
提交任务
回调信息
视频质量评分
提交任务
回调信息
视频转动图
提交任务
回调信息
创建模板
更新模板
数字水印
提交任务
回调信息
提取数字水印
提交任务
回调信息
视频拆条
提交任务
回调信息
智能封面
提交任务
回调信息
创建模板
更新模板
视频合成
提交任务
内容审核

版权所有：腾讯云计算（北京）有限责任公司

图片审核
图片单次审核
图片批量审核
查询图片审核任务结果
图片审核回调内容
图片审核结果反馈
视频审核
提交视频审核任务
查询视频审核任务结果
视频审核回调内容
音频审核
提交音频审核任务
查询音频审核任务结果
音频审核回调内容
文本审核
文本文件审核
文本内容审核
查询文本审核任务结果
文本审核回调内容
文本审核结果反馈
文档审核
提交文档审核任务
查询文档审核任务结果
文档审核回调内容
网页审核
提交网页审核任务
查询网页审核任务结果
网页审核回调内容
直播审核
提交直播审核任务
取消直播审核任务
查询直播审核任务结果
直播审核回调内容
AI 内容识别
异步服务和队列配置
开通AI 内容识别（异步）服务
查询AI 内容识别（异步）服务
关闭AI 内容识别（异步）服务

版权所有：腾讯云计算（北京）有限责任公司

查询AI 内容识别（异步）队列
更新AI 内容识别（异步）队列
图片上色
图像超分
图像增强
图像智能裁剪
实时文字翻译
图像修复
Logo 识别
图片二维码识别
图片二维码生成
图片标签
游戏场景识别
图片质量评分
人脸识别
人脸检测
人脸特效
人体识别
宠物识别
人脸核身
身份证识别
活体人脸核身
获取数字验证码
获取动作顺序
车辆车牌识别
图片文字识别(OCR)
商品抠图
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
内容翻译
提交任务
回调信息
词性分析
提交任务
回调信息

版权所有：腾讯云计算（北京）有限责任公司

视频目标检测
提交任务
回调信息
创建模板
更新模板
视频人像分割
提交任务
回调信息
文档预览
查询文档预览开通状态
文档转码
文档转码同步请求
异步处理任务接口
提交文档转码任务
查询指定的文档转码任务
拉取符合条件的文档转码任务
异步处理队列接口
查询文档转码队列
更新文档转码队列
支持的功能和效果
文档转html
文档转html同步请求
自定义配置概述
配置项说明
高级功能
支持的字体和动画
更新日志
文件处理
服务和队列配置
开通文件处理服务
查询文件处理服务
关闭文件处理服务
查询文件处理队列
更新文件处理队列
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

版权所有：腾讯云计算（北京）有限责任公司

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
压缩包预览
智能语音
服务和队列配置
开通智能语音服务
查询智能语音服务
关闭智能语音服务
查询智能语音队列
更新智能语音队列
人声分离
提交任务
任务回调
创建模板
更新模板
音频降噪
提交任务
任务回调
创建模板
更新模板
语音合成
提交任务
任务回调
创建模板
更新模板
语音识别
提交任务
任务回调
创建模板
更新模板
听歌识曲
提交任务
任务回调
音乐评分

版权所有：腾讯云计算（北京）有限责任公司

提交任务
任务回调
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
获取工作流实例详情
获取工作流实例详情列表
任务接口
取消任务
提交多任务处理
查询指定任务
获取符合条件的任务列表
模板接口
删除模板
查询模板
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
请求签名
附录
公共请求头部
公共响应头部
结构体

版权所有：腾讯云计算（北京）有限责任公司

API 文档
API概览

最近更新时间：2023-08-14 11:59:58

腾讯云数据万象服务（CI）API接口及说明如下：

开通服务

接口名称	说明
绑定数据万象服务	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。
查询数据万象服务	用于查询该 Bucket 是否已开通数据万象服务。
解绑数据万象服务	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））。

注意：
调用接口为存储桶开通数据万象，并未进行角色授权，可能会影响部分功能使用，如您首次开通数据万象，建议您通过控制台进行操作，并完成角色授权，点击登录 数据万象控制台 。

存储桶配置

接口名称	说明
查询防盗链	用于查询 Bucket 中防盗链详情（黑白名单）。
设置防盗链	用于对存储桶添加防盗链（设置黑白名单）。
开通原图保护	对 Bucket 开通原图保护功能。
查询原图保护状态	用于查询原图保护功能是否开启。
关闭原图保护	用于关闭原图保护功能。
提交病毒检测任务	云查毒功能为异步请求方式，您可以通过本接口对云上的文件进行文件病毒（例如木马病毒、蠕虫病毒等）检测。
查询病毒检测任务结果	用于查询一个病毒检测任务的状态或结果。

版权所有：腾讯云计算（北京）有限责任公司

图片处理

数据万象CI提供了一套便捷的 图片处理机制 ，通过这套机制可以对存储在对象存储 COS 中的图片进行一系列的图片处理操作。

功能分类	接口名称	说明
图片基础处理	图片样式	将基础图片处理操作设置为样式，用户可直接使用样式参数快速对图片进行处理。
	基础图片处理	对图片进行裁剪、缩放、水印、获取信息等基础处理。
	图片基础压缩（WebP）	将图片转换为 webp 压缩图片格式。
	合成动图	将多张 jpg 或 png 格式的图片合成为一个 GIF 动图。
	获取原图	获取某些场景下的原图。
	图片明水印	为图片添加水印，支持添加 图片水印 与 文字水印 两种方式。
图片高级压缩		将图片转换为 TPG 、AVIF、 HEIF 等高压缩比格式。
图片Guetzli压缩		对 JPG 图片进行 Guetzli 画质无损压缩。
图片盲水印		将水印以不可见的形式添加到图片中。
异常图片检测		检测图片中是否隐含其他类型的可疑文件。
海报合成任务		使用模板批量生成多类型成品海报。
图片异步处理		提交与查询图片异步处理任务，并支持创建与更新模板。
异步服务队列配置	查询队列	搜索图片处理队列。
	更新队列	更新图片处理队列。

注意：
异步服务均支持取消任务、删除模板、查询模板与批量任务；相关接口所有功能通用，具体请查看 任务与工作流 。

媒体处理

数据万象提供了一套完善便捷的 媒体处理机制，通过这套机制可以对存储在对象存储 COS 中的媒体对象进行各类型处理操作。

功能分类	接口名称	说明
服务配置	绑定媒体处理服务	授权媒体处理服务角色访问存储桶。
	查询媒体处理服务	查询已经开通媒体处理功能的存储桶。
	解绑媒体处理服务	关闭媒体处理服务并删除队列。
	搜索媒体处理队列	搜索媒体处理队列。
	更新媒体处理队列	更新媒体处理队列
同步服务	获取媒体信息	获取媒体文件的信息
	获取媒体截图	获取媒体文件某个时间的截图
	读取私有 M3U8	用于获取私有 M3U8 ts 资源的下载授权。
异步服务	获取媒体信息	提交一个获取媒体信息任务，查看回调信息。
	音视频转码	提交一个音视频转码任务，查看回调信息；并支持创建与更新模板。
	极速高清转码	提交一个极速高清转码任务，查看回调信息；并支持创建与更新模板。
	音视频转封装（音视频分段）	提交一个音视频转封装任务，此任务支持分段，查看回调信息。
	音视频流分离	提交一个音视频流分离任务，查看回调信息。
	音视频拼接	提交一个音视频拼接任务，查看回调信息；并支持创建与更新模板。
	精彩集锦	提交一个精彩集锦任务，查看回调信息；并支持创建与更新模板。
	视频截图	提交一个视频截图任务，查看回调信息；并支持创建与更新模板。
	画质增强	提交一个画质增强任务，查看回调信息；并支持创建与更新模板。
	视频明水印	提交一个视频明水印任务，查看回调信息；并支持创建与更

		新模板。
	视频标签	提交一个视频标签任务，查看回调信息。
	视频质量评分	提交一个视频质量评分任务，查看回调信息。
	视频转动图	提交一个视频转动图任务，查看回调信息；并支持创建与更新模板。
	数字水印	提交一个数字水印任务，查看回调信息。
	提取数字水印	提交一个提取数字水印任务，查看回调信息。
	视频拆条	提交一个视频拆条任务，查看回调信息。
	智能封面	提交一个智能封面任务，查看回调信息；并支持创建与更新模板。

注意：
异步服务均支持取消任务、删除模板、查询模板与批量任务；相关接口所有功能通用，具体请查看 任务与工作流 。

内容审核

功能分类	说明
图片审核	图片审核功能支持同步、异步请求方式。
视频审核	视频审核功能为异步任务方式。
音频审核	音频审核功能为异步任务方式。
文本审核	文本文件审核为异步请求方式，文本内容审核为同步请求。
文档审核	文档审核功能为异步任务方式。
网页审核	网页审核功能为异步任务方式。
直播审核	直播流审核功能为异步任务方式。

内容识别

版权所有：腾讯云计算（北京）有限责任公司

功能分类	接口名称	说明
服务配置	开通 AI 内容识别	开通 AI 内容识别服务并生成队列。
	查询 AI 内容识别	查询 AI 内容识别服务状态。
	查询队列	搜索 AI 内容识别队列。
	更新队列	更新 AI 内容识别队列。
同步服务	图片上色	对黑白图像进行上色。
	图像超分	对图像进行超分辨率处理，当前默认超分为宽高的2倍。
	图像增强	对图像进行增强处理。
	图像智能裁剪	对图像进行智能裁剪，支持持久化、云上处理及下载时处理。
	实时文字翻译	对文字块进行翻译。
	图像修复	检测并擦除图片中常见的标志,并对擦除部分进行智能修复。
	Logo识别	对图片内电商 Logo 的识别，返回图片中 Logo 的名称、坐标、置信度分值。返回图片中Logo的名称、坐标、置信度分值。
	二维码识别	识别图片中有效二维码的位置及内容，输出图像中二维码包含的文本信息，并可对识别出的二维码添加马赛克。
	二维码生成	根据用户指定的文本信息（URL 或文本），生成对应的二维码或条形码。
	图片标签	实现对 COS 存量数据的图片标签识别，返回图片中置信度较高的主题标签。
	游戏场景识别	实现游戏图片场景的识别，返回图片中置信度较高的游戏类别标签。
	图片质量评分	对图片文件进行检测，从多方面评估，最终给出综合可观的清晰度评分和主观的美观度评分。
	人脸检测	检测图片中的人脸位置。
	人脸特效	提供人脸美颜、人像变换、人像分割功能。
	人体识别	识别并输出画面中人体，输出其位置（矩形框）和置信度。
	身份证识别	支持中国大陆居民二代身份证正反面所有字段的识别。

版权所有：腾讯云计算（北京）有限责任公司

	活体人脸核身	对录制的自拍视频进行活体检测，从而确认当前用户为真人。检测为真人后，再判断该视频中的人与权威库的证件照是否属于同一个人，实现用户身份信息核实。
	车辆车牌识别	检测图片中的车辆，识别出车辆的品牌、颜色、位置、车牌位置等信息。
	通用文字识别	将图片上的文字内容，智能识别为可编辑的文本。
	商品抠图	检测图片中的商品信息，生成只包含商品信息的图片。
	以图搜图	需开通以图搜图服务，通过添加图片，对图片进行搜索。
异步服务	内容翻译	提交一个内容翻译任务，查看回调信息。
	词性分析	提交一个词性分析任务，查看回调信息。
	视频目标检测	提交一个视频目标检测任务，查看回调信息；并支持创建与更新模板。
	视频人像分割	提交一个视频人像分割任务，查看回调信息。

注意：

异步服务均支持取消任务、删除模板、查询模板与批量任务；相关接口所有功能通用，具体请查看 任务与工作流 。

文档处理

功能分类	接口名称	说明
服务配置	查询文档预览开通状态	用于查询已经开通文档预览功能的 Bucket。
	查询队列	用于查询文档转码队列。
	更新队列	用于更新文档转码队列
文档转码	文档转码同步请求	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。
	文档转码异步任务	提交、查询文档转码任务
文档转HTML	文档转html同步请求	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。

	文档转html高级用法	JS-SDK 提供了高级功能可直接对文档进行操作。

> **注意：**
> 异步服务均支持取消任务、删除模板、查询模板与批量任务；相关接口所有功能通用，具体请查看 任务与工作流 。

文件处理

功能分类	接口名称	说明
服务配置	开通文件处理服务	于开通文件处理服务。
	更新队列	用于更新文件处理的队列
	查询队列	用于搜索文件处理队列。
文件哈希值计算	哈希值计算同步请求	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。
	提交哈希值计算任务	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。
文件解压	提交文件解压任务	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。
压缩包预览	压缩包预览同步请求	不解压文件的情况下预览压缩包内的内容，包含文件数量、名称、文件时间等
多文件打包压缩	提交多文件打包压缩任务	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

> **注意：**
> 异步服务均支持取消任务、删除模板、查询模板与批量任务；相关接口所有功能通用，具体请查看 任务与工作流 。

智能语音

功能分类	接口名称	说明

服务配置	查询语音识别开通状态	用于查询存储桶是否已开通语音识别功能。
	查询队列	用于查询语音识别队列。
	更新队列	用于更新语音识别队列
音频降噪	提交任务	以提交任务的方式进行音频降噪处理。
	创建模板	创建音频降噪模板。
人声分离	提交任务	以提交任务的方式进行人声分类处理。
	创建模板	创建音频降噪模板。
语音合成	提交任务	以提交任务的方式进行语音合成处理。
	创建模板	创建音频降噪模板。
语音识别	提交任务	以提交任务的方式进行语音识别处理。
	创建模板	创建音频降噪模板。
听歌识曲	提交任务	以提交任务的方式进行听歌识曲处理。

注意：
异步服务均支持取消任务、删除模板、查询模板与批量任务；相关接口所有功能通用，具体请查看 任务与工作流 。

任务与工作流

工作流接口

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。

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。

功能分类	接口名称
工作流	创建工作流
	查询工作流

	删除工作流
	更新工作流
	测试工作流
	工作流回调
工作流实例	获取工作流实例详情
	获取工作流实例详情列表
任务设置	取消任务
	提交多任务处理
	查询指定任务
	获取符合条件的任务列表
模板设置	删除模板
	查询模板
批量拉取任务	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

说明：
创建任务、任务回调信息、创建模板、更新模板：各功能接口不同，请查看对应功能相关信息。

版权所有：腾讯云计算（北京）有限责任公司

开通服务
绑定数据万象服务

最近更新时间：2023-06-07 17:07:09

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。
开通后可使用数据万象基础图片处理接口、内容识别同步接口与内容审核接口，如使用其他能力，需进行单独绑定。
存储桶的其他功能如查询、修改等，请使用 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

该响应的响应体为空。

版权所有：腾讯云计算（北京）有限责任公司

查询数据万象服务

最近更新时间：2023-06-07 17:07:16

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

```
<?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：2023-06-07 17:07:21

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））。
当前接口仅支持解绑数据万象图片处理服务、内容识别同步服务与内容审核服务，其余能力解绑需单独处理。

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

该响应的响应体为空。

存储桶配置
查询防盗链

最近更新时间：2023-06-07 17:07:38

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

具体内容包括防盗链的种类并枚举所有地址 A。

```
<?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：2023-06-07 17:07:33

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）。
- 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
Ty	Hotli	防盗链类型，white 为白名单，black 为黑名单，off 为关	String	是

pe	nk	闭		

响应

响应头

此接口仅返回公共响应头部，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>
```

响应

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

开通原图保护

最近更新时间：2023-06-07 17:07:46

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

该响应的响应体为空。

版权所有：腾讯云计算（北京）有限责任公司

查询原图保护状态

最近更新时间：2023-06-07 17:07:52

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

```
<?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。

版权所有：腾讯云计算（北京）有限责任公司

关闭原图保护

最近更新时间：2023-06-07 17:07:58

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

该响应的响应体为空。

云查毒接口
提交病毒检测任务

最近更新时间：2023-06-20 15:21:02

功能描述

云查毒功能为异步请求方式，您可以通过本接口对云上的文件进行文件病毒（例如木马病毒、蠕虫病毒等）检测，然后通过查询病毒检测结果接口，查询病毒检测的结果。

费用说明

- 如果文件属于 COS 上的低频存储类型，调用检测会产生 COS 低频数据取回费用 。
- 不支持检测 COS 上的归档存储类型和深度归档存储类型的文件，此类型文件请先 恢复归档文件 。

限制说明

- 文件大小支持：**文件＜1GB**。
- 文件支持格式：EXE、DLL、SCR、SYS、MSI、SWF、JAR、APPLETJAR、JS、VBS、HTML、BAT、WSF、PS1、HTM、MHT、HTA、CHM、LNK、DOC、DOCX、DOCX、DOTX、DOCM、DOTM、XLS、XLSX、XLSM、XLSB、XLTM、 XLTX、XLAM、PPT、PPTX、POTX、PPSX、PPTM、POTM、PPSM、XML、RTF、PDF、ELF、 XML、MAC、VBE、3DMAX、ISO、CPIO、RPM、XZ、DEB。
- 支持检测压缩文件，例如：zip、7z、rar4、rar5、inno、cab、autoit、nsis、gz、tar、iso、mime、chm、jar、dmg、apk 等。

请求

请求示例

```
POST /virus/detect HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

> **说明**
> - Authorization: Auth String（详情请参见 请求签名 文档）。
> - 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>
  </Input>
  <Conf>
    <DetectType>Virus</DetectType>
    <Callback></Callback>
  </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	病毒检测的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要检测的文件信息。	Container	是
Conf	Request	检测规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选

Object	Request.Input	存储在 COS 存储桶中的病毒文件名称，例如在目录 test 中的文件 virus.doc，则文件名称为 test/virus.doc。 注意：Object 和 Url 只能选择其中一种。	String	是
Url	Request.Input	病毒文件的链接地址，例如：`http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/virus.doc`。 注意：Object 和 Url 只能选择其中一种。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
DetectType	Request.Conf	检测的病毒类型，当前固定为：Virus。	String	是
Callback	Request.Conf	检测结果回调通知到您设置的地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com`。	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <JobsDetail>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
  </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	病毒检测返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	病毒检测任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
JobId	Response.JobsDetail	本次病毒检测任务的 ID。	String
State	Response.JobsDetail	病毒检测任务的状态，值为 Submitted（已提交检测）、Success（检测成功）、Failed（检测失败）、Auditing（检测中）其中一个。	String
CreationTime	Response.JobsDetail	病毒检测任务的创建时间。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

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

<Request>
 <Input>
  <Object>a.doc</Object>
 </Input>
 <Conf>
  <DetectType>Virus</DetectType>
```

```
    <Callback>http://callback.com/</Callback>
  </Conf>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>ssb1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Submitted</State>
    <CreationTime>2021-07-07T12:12:12+0800</CreationTime>
  </JobsDetail>
</Response>
```

查询病毒检测任务结果

最近更新时间：2023-06-20 17:32:16

功能描述

本接口用于查询一个病毒检测任务的状态或结果。

请求

请求示例

```
GET /virus/detect/<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>
    <Message></Message>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <Suggestion></Suggestion>
    <DetectDetail>
      <Result>
        <FileName></FileName>
        <VirusName></VirusName>
      </Result>
      <Result>
        <FileName></FileName>
        <VirusName></VirusName>
      </Result>
    </DetectDetail>
  </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	病毒检测任务结果。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	病毒检测任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
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.J	本次病毒检测任务的 ID。	String

	obsDetail		
State	Response.JobsDetail	病毒检测任务的状态，值为 Submitted（已提交检测）、Success（检测成功）、Failed（检测失败）、Auditing（检测中）其中一个。	String
CreationTime	Response.JobsDetail	病毒检测任务的创建时间。	String
Object	Response.JobsDetail	检测的文件为存储在 COS 中的文件时，该字段表示本次检测的文件名称。	String
Url	Response.JobsDetail	检测的文件为一条文件链接时，该字段表示本次检测的文件链接。	String
Suggestion	Response.JobsDetail	该字段表示本次判定的检测结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的检测结果进行相应处理。有效值：**nomal**（检测正常），**block**（检测到文件含有病毒）。	String
DetectDetail	Response.JobsDetail	本次检测的详细信息。	Container Array

Container 节点 DetectDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Result	Response.JobsDetail.DetectDetail	单个文件的检测结果，可能有多个结果。	Array

Array 节点 Result 的内容：

节点名称（关键字）	父节点	描述	类型
FileName	Response.JobsDetail.DetectDetail.Result	检测的文件名称，如果是压缩文件，该字段表示解压后内部的具体文件名。	String
VirusName	Response.JobsDetail.DetectDetail.Result	检测到的病毒名称。	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 /virus/detect/ssb1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <Code>Success</Code>
    <Message>Success</Message>
    <JobId>ss364c726b370211ec8fb35254008618d9</JobId>
    <State>Success</State>
    <CreationTime>2021-07-07T12:12:12+0800</CreationTime>
    <Object>a.doc</Object>
    <Suggestion>block</Suggestion>
    <DetectDetail>
      <Result>
        <FileName>xxx</FileName>
        <VirusName>Bk.YDWebShell.Php.Small.11100752</VirusName>
      </Result>
      <Result>
      <FileName>xxx</FileName>
        <VirusName>Bk.YDWebShell.Php.Small.11100752</VirusName>
      </Result>
      </DetectDetail>
  </JobsDetail>
</Response>
```

图片处理
图片处理机制介绍

最近更新时间：2023-06-14 11:50:30

简介

数据万象提供了一套便捷的图片处理机制，通过这套机制可以对存储在对象存储 COS 中的图片进行一系列的图片处理操作。
图片处理机制分为以下三种方式：

方式	类别	说明
下载时处理	同步处理	在图片的 URL 后加上 图片处理参数
上传时处理	持久化处理	上传图片时，在请求包头部中加入 Pic-Operations 项并设置好相应的图片处理参数
云上数据处理	持久化处理	针对 COS 上已经存在的图片进行处理，处理方式与上传时处理类似，在请求包头部中加入 Pic-Operations 项并设置好相应的图片处理参数

下载时处理

您可以通过在图片的 URL 后加上 图片处理参数，实现访问图片的时候同步获取到处理后的图片。

请求语法

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

示例

请求

```
GET /filename.jpg?imageMogr2/format/png HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Wed, 28 Oct 2015 20:32:00 GMT
```

```
Authorization:XXXXXXXXXXX
```

返回

```
HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 565
Date: Fri, 10 Apr 2020 09:35:16 GMT
Last-Modified: Fri, 10 Apr 2020 09:35:05 GMT
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****
```

上传时处理

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 项并设置好相应的 图片处理参数，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以内的图片处理。

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）。
- 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。 当 `fileid` 与 `ObjectKey` 相同时，处理后的文件将覆盖原文件存储；不相同时将分别存储原文件和处理后文件。名称规则说明： 1. 名称需要经过 URL 安全的 Base64 编码 2. 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 3. 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 4. **注意：请不要以 / 结尾，否则会产生空文件名**
rule	String	是	图片处理参数，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	图片格式
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，格式为：0xRRGGBB（RR、GG、BB都是十六进制数，表示红、绿、蓝三种颜色）
Orientation	Int	图片旋转角度，取值为1-8的整数： 1：旋转0° 2：水平翻转 3：旋转180° 4：垂直翻转 5：顺时针旋转90°+水平翻转 6：顺时针旋转90° 7：顺时针旋转90°+垂直翻转 8：逆时针旋转90°
FrameCount	Int	图片的帧数。静态图为1，动图为对应的帧数.

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
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
FrameCount	Int	图片的帧数。静态图为1，动图为对应的帧数.

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:XXXXXXXXXXX
Pic-Operations: {"is_pic_info":1,"rules":[{"fileid":"test.png","rule":"imageMogr2/format/pr
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>
          <FrameCount>1</FrameCount>
        </ImageInfo>
      </OriginalInfo>
      <ProcessResults>
        <Object>
          <Key>test.png</Key>
          <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test.pı
          <Format>png</Format>
          <Width>640</Width>
          <Height>427</Height>
          <Size>463092</Size>
          <Quality>100</Quality>
          <ETag>&amp;quot;eaa4e3d8fd498bbc63be3b71c46b9c61f23e3f0c&amp;quot;</ETag>
          <FrameCount>1</FrameCount>
        </Object>
      </ProcessResults>
</UploadResult>
```

注意

上传时处理支持 COS V5 的分块上传功能，您在使用 COS V5 的 Complete Multipart Upload 接口时只需在请求包头部中加入 Pic-Operations 项，即可实现图片处理。

```
POST /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations: <PicOperations>
```

说明

COS 接口请参见 COS Complete Multipart Upload 文档。

云上数据处理

对象存储的图片处理 API 能够对已存储在 COS 的图片进行相应处理操作，并将结果存入到 COS。

请求语法

```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations: <PicOperations>
```

请求内容

Pic-Operations 为 json 格式的字符串，具体参数如下：

参数名称	类型	是否必选	描述
is_pic_info	Int	否	是否返回原图信息，0不返回原图信息，1返回原图信息，默认为0
rules	Array	否	处理规则，一条规则对应一个处理结果（目前支持五条规则），不填则不进行图片处理

rules（json 数组）中每一项具体参数如下：

参数名称	类型	是否必选	描述
bucket	String	否	存储结果的目标 bucket 名称，形如 BucketName-APPID，如果不指定的话默认保存到当前存储桶
fileid	String	是	处理后文件的保存路径及名称。 当 `fileid` 与 `ObjectKey` 相同时，处理后的文件将覆盖原文件存储；不相同时将分别存储原文件和处理后文件。名称规则说明： 1. 名称需要经过 URL 安全的 Base64 编码 2. 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 3. 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 4. **注意：请不要以 `/` 结尾，否则会产生空文件名**
rule	Str	是	处理参数，参见对象存储图片处理 API。若按指定样式处理，则以 `style/`

	ing		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，格式为：0xRRGGBB（RR、GG、BB都是十六进制数，表示红、绿、蓝三种颜色）

版权所有：腾讯云计算（北京）有限责任公司

Orientation	Int	图片旋转角度，取值为1-8的整数： 1：旋转0° 2：水平翻转 3：旋转180° 4：垂直翻转 5：顺时针旋转90°+水平翻转 6：顺时针旋转90° 7：顺时针旋转90°+垂直翻转 8：逆时针旋转90°
FrameCount	Int	图片的帧数。静态图为1，动图为对应的帧数.

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
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
FrameCount	Int	图片的帧数。静态图为1，动图为对应的帧数

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.png","rule":"imageMogr2/format/pr
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/filenar
          <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>
            <FrameCount>1</FrameCount>
          </ImageInfo>
        </OriginalInfo>
        <ProcessResults>
          <Object>
            <Key>test.png</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
            <FrameCount>1</FrameCount>
          </Object>
        </ProcessResults>
</UploadResult>
```

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：2023-06-19 16:47:01

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: xxxxxxxxxxxxxxxxx
```

查询样式

最近更新时间：2023-06-05 15:08:21

功能描述

用于查询该 Bucket 下已有的图片样式。若不使用请求体，则查询该 Bucket 下的全部图片样式，若使用请求体，则查询请求体中指定的图片样式。

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。

请求头

此接口仅使用公共请求头，详情请参见 公共请求头部 。

请求体

若不使用请求体，则查询该 Bucket 下的全部图片样式；若使用请求体，则查询请求体中指定的图片样式。

```
<GetStyle>
  <StyleName>string</StyleName>
  <StyleName>string</StyleName>
</GetStyle>
```

请求参数说明

节点名称	父节点	描述	类型	是否必选
GetStyle	无	查询图片样式名称的容器。	Container	否
StyleName	GetStyle	查询的图片样式名称。	String	否

版权所有：腾讯云计算（北京）有限责任公司

响应

响应头

此接口仅使用公共响应头，详情请参见 公共响应头部 。

响应体

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

响应参数说明

节点名称	父节点	描述	类型
StyleList	无	查询到的图片样式集合。	Container
StyleRule	StyleList	图片样式名称和参数。	Container
StyleName	StyleRule	图片样式名称。	String
StyleBody	StyleRule	图片样式参数。	String

删除样式

最近更新时间：2023-06-05 15:08:05

功能描述

用于删除指定的图片样式。

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。

请求头

此接口仅使用公共请求头，详情请参见 公共请求头部 。

请求体

```
<DeleteStyle>
  <StyleName>string</StyleName>
</DeleteStyle>
```

请求参数说明：

节点名称	父节点	描述	类型	是否必选
DeleteStyle	无	待删除样式名称容器	Container	是
StyleName	DeleteStyle	待删除样式名称	String	是

响应

响应头

此接口仅使用公共响应头，详情请参见 公共响应头部 。

响应体

该响应体返回为空。

基础图片处理
缩放

最近更新时间：2023-07-11 11:48:22

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供图片缩放功能。
该功能支持以下处理方式：

- 下载时处理
- 上传时处理
- 云上数据处理

限制说明

- 支持格式：默认支持处理 JPG/JPEG、BMP、GIF、PNG、WebP 格式，开通 图片高级压缩功能 后，可支持AVIF、HEIF、TPG 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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:
```

版权所有：腾讯云计算（北京）有限责任公司

```
{
 "is_pic_info": 1,
 "rules": [{
    "fileid": "exampleobject",
    "rule": "imageMogr2/thumbnail/<imageSizeAndOffsetGeometry>"
 }]
}
```

3. 云上数据处理

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

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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码

处理参数说明

操作名称：thumbnail。

参数	含义	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg。	是
/thumbnail/!	指定图片的宽高为原图的 Scale%	否

<Scale>p		
/thumbnail/!<Scale>px	指定图片的宽为原图的 Scale%，高度不变	否
/thumbnail/!x<Scale>p	指定图片的高为原图的 Scale%，宽度不变	否
/thumbnail/<Width>x	指定目标图片宽度为 Width，高度等比缩放	否
/thumbnail/x<Height>	指定目标图片高度为 Height，宽度等比缩放	否
/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	否
/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	否
/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	否
/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	否
/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	否
/thumbnail/<Area>@	等比缩放图片，缩放后的图像，总像素数量不超过 Area	否
/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	否
/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	否
/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	否

说明：

表格中的多个 thumbnail 参数，需至少选择一个，若选择了多个，仅会生效已选的第一个。
例如选择了两个参数：/thumbnail/!<Scale>p/!x<Scale>p
仅会生效!<Scale>p

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/t
```

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/t
```

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/t
```

最终效果如下：



案例四：Pad 模式缩放并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：

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

> **说明**
> <signature> 为签名部分，获取方式请参考 请求签名 。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名 。

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

裁剪

最近更新时间：2023-04-20 09:59:58

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。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过9999像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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>"
  }]
}
```

3. 云上数据处理

```
POST /<ObjectKey>?image_process HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
     "fileid": "exampleobject",
     "rule": "imageMogr2/cut/<width>x<height>x<dx>x<dy>
                  /crop/<imageSizeAndOffsetGeometry>
                  /iradius/<radius>
                  /rradius/<radius>
                  /scrop/<Width>x<Height>"
  }]
}
```

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后

图片。

- fileid 的值需要经过 URL 安全的 Base64 编码

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。 处理后图片如果是PNG、WebP或BMP等支持透明通道的图片，那么图片非圆形区域的部分将会以透明填充；如果是JPG图片，不支持透明通道，非圆形区域是以白色进行填充。

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。 处理后图片如果是PNG、WebP或BMP等支持透明通道的图片，那么图片非圆形区域的部分将会以透明填充；如果是JPG图片，不支持透明通道，非圆形区域是以白色进行填充。

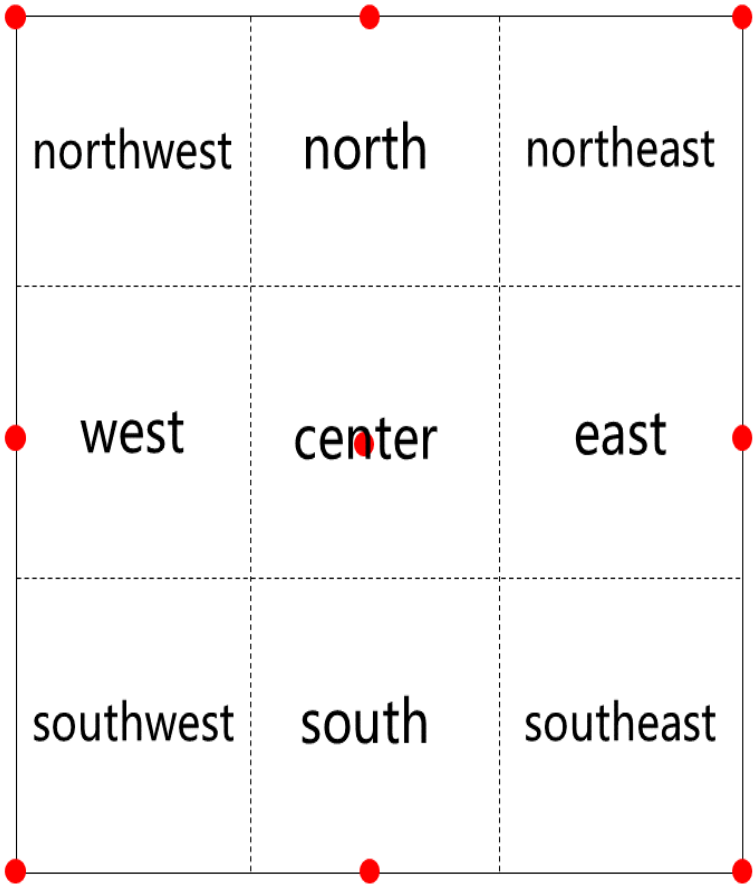
人脸智能裁剪参数说明

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/c
```

原图如下：



版权所有：腾讯云计算（北京）有限责任公司

最终效果如下：



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/c
```

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/i
```

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/r
```

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/s
```

> **说明**
> 如果图片中没有识别到人脸，则返回原图。

案例六：人脸智能裁剪并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：

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

> **说明**
> <signature> 为签名部分，获取方式请参考 请求签名 。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名 。

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

旋转

最近更新时间：2023-07-06 18:24:43

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：默认支持处理 JPG/JPEG、BMP、GIF、PNG、WebP 格式，开通 图片高级压缩功能 后，可支持AVIF、HEIF、TPG 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口形式

1. 下载时处理

```
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"
  }]
}
```

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。
- 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。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码。

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/r
```

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

版权所有：腾讯云计算（北京）有限责任公司

格式转换

最近更新时间：2023-06-05 15:08:49

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供格式转换、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、webp、avif、tpg、heif 等静态图片，支持互相转换。
- 转换为avif、tpg、heif需先开通图片高级压缩服务，详情请查看 图片高级压缩 。

接口形式

1. 下载时处理

```
GET /<ObjectKey>?imageMogr2/format/<Format> 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>"
         }]
}
```

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

- 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。
- Pic-Operations 为 json 格式的字符串，具体参数信息可参考 图片处理机制介绍 。
- 当 fileid 与 ObjectKey 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- fileid 的值需要经过 URL 安全的 Base64 编码

处理参数说明

参数	含义	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg。	是
/format/<Format>	有效值：jpg，bmp，gif，png，webp，heif，avif，tpg。 如需转换为avif、tpg、heif，请先开通图片高级压缩服务。	是
/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	否

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 图片处理机制介绍 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/f
```

案例二：将 jpeg 格式的原图片转换为 png 格式并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：

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

说明

`<signature>` 为签名部分，获取方式请参考 请求签名 。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名 。

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

质量变换

最近更新时间：2023-04-20 10:00:16

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>"
 }]
}
```

版权所有：腾讯云计算（北京）有限责任公司

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。
- 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 。
- 当 fileid 与 ObjectKey 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- fileid 的值需要经过 URL 安全的 Base64 编码

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/<q	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

uality>	• 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/r
```

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
```

渐进显示

最近更新时间：2023-02-15 14:40:30

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供jpg图片的渐进显示功能。

限制说明

- 支持格式：支持处理原图为 JPG、BMP、GIF、PNG、WebP、HEIF的格式，但渐进显示仅支持JPG图进行显示，其他格式图片需搭配格式转换 `format` 参数将原图转换为JPG格式。
- 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。

接口示例

```
GET /<ObjectKey>?imageMogr2/interlace/<Mode> 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

参数	含义
Object Key	对象文件名，例如 folder/sample.gif。
/interlace/	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。

版权所有：腾讯云计算（北京）有限责任公司

实际案例

将原图格式为PNG的图片保存为JPG格式，并设置成渐进显示：

```
http://examples-125xxxxxx.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/for
```

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
```

动图截帧

最近更新时间：2023-05-30 16:25:50

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供动图截帧功能。
该功能支持以下处理方式：

- 下载时处理
- 上传时处理
- 云上数据处理

限制说明

- 支持格式：支持处理 GIF、WebP 格式。
- 体积限制：GIF 原图宽 x 高 x 帧数不超过2.5亿像素；WebP原图宽高不超过16383像素。
- 动图帧数限制：帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?imageMogr2/cgif/<FrameNumber>
                    /egif/<FrameInterval> 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": [{
```

版权所有：腾讯云计算（北京）有限责任公司

```
    "fileid": "exampleobject",
    "rule": "imageMogr2/cgif/<FrameNumber>
                /egif/<FrameInterval>"
 }]
}
```

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/cgif/<FrameNumber>
                /egif/<FrameInterval>"
 }]
}
```

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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码

处理参数说明

参数	含义
ObjectKey	对象文件名，例如 folder/sample.gif。
/frame/	动图截单帧（仅支持下载时处理）：可指定截取动图的第N帧，N取值为正整数（1,2,3……），如果N大于动图的最大帧数，则取最后一帧。例如截取某动图的第1帧，则参数为 /frame/1。

/cgif/	动图格式优化（截取型）：针对原图为 gif、webp 格式，对动图进行的优化，截帧降颜色。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）。
/egif/	动图格式优化（抽帧型）：只针对原图为 gif、webp 格式，对动图进行的优化，进行降颜色处理，并每隔一定帧数（FrameInterval）抽取目标帧进行输出。 • FrameInterval 取值[1,10]。
/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。

注意

cgif与egif参数同时传递时，优先处理cgif。

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**。

对某GIF动图截取前10帧：

```
http://examples-125xxxxxx.cos.ap-shanghai.myqcloud.com/sample.gif?imageMogr2/cgif/
```

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名 。

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

高斯模糊

最近更新时间：2023-06-05 10:08:00

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。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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>"
  }]
}
```

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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。

- `fileid` 的值需要经过 URL 安全的 Base64 编码

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>	模糊半径，取值范围为1 - 50
<sigma>	正态分布的标准差，必须大于0
/ignore-	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

error/1	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/b
```

高斯模糊处理后效果如下：



案例二：模糊半径取8，sigma 值取5，进行高斯模糊处理，并携带私有文件签名

处理方式同上，仅增加签名部分，并与图片处理参数以“&”连接，示例如下：

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

> **说明**
> `<signature>` 为签名部分，获取方式请参考 请求签名 。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名 。

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

亮度

最近更新时间：2023-04-20 10:00:26

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。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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": [{
```

版权所有：腾讯云计算（北京）有限责任公司

```
    "fileid": "exampleobject",
    "rule": "imageMogr2/bright/<value>"
  }]
}
```

3. 云上数据处理

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

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码

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/b
```

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：2023-04-20 10:00:34

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。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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,
```

版权所有：腾讯云计算（北京）有限责任公司

```
  "rules": [{
     "fileid": "exampleobject",
     "rule": "imageMogr2/contrast/<value>"
  }]
}
```

3. 云上数据处理

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

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码

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/c
```

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：2023-04-20 10:00:38

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。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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": [{
```

版权所有：腾讯云计算（北京）有限责任公司

```
    "fileid": "exampleobject",
    "rule": "imageMogr2/sharpen/<value>"
  }]
}
```

3. 云上数据处理

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

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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。
- 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。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码。

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/s
```

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
```

版权所有：腾讯云计算（北京）有限责任公司

灰度图

最近更新时间：2023-04-20 10:00:42

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。

限制说明

- 支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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>"
  }]
}
```

版权所有：腾讯云计算（北京）有限责任公司

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>"
  }]
}
```

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 。
- 当 fileid 与 ObjectKey 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- fileid 的值需要经过 URL 安全的 Base64 编码

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/g
```

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
```

获取图片基本信息

最近更新时间：2023-02-15 14:42:48

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。

限制说明

- 支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

```
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）

版权所有：腾讯云计算（北京）有限责任公司

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
bit_depth	图片的位深
vertical_dpi	图片的垂直分辨率
horizontal_dpi	图片的水平分辨率

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": "77a16fa70e
```

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
```

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

获取图片 EXIF

最近更新时间：2023-02-15 14:42:52

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。

限制说明

- 支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

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

接口示例

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

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
```

版权所有：腾讯云计算（北京）有限责任公司

获取图片主色调

最近更新时间：2023-02-15 14:42:56

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。

限制说明

- 支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

```
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
```

版权所有：腾讯云计算（北京）有限责任公司

去除元信息

最近更新时间：2023-04-20 10:01:07

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。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?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": [{
```

版权所有：腾讯云计算（北京）有限责任公司

```
    "fileid": "exampleobject",
    "rule": "imageMogr2/strip"
  }]
}
```

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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码 。

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/s
```

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
```

版权所有：腾讯云计算（北京）有限责任公司

快速缩略模板

最近更新时间：2023-06-26 18:50:36

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。
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：默认支持处理 JPG/JPEG、BMP、GIF、PNG、WebP 格式，开通 图片高级压缩功能 后，可支持AVIF、HEIF、TPG 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?imageView2/<mode>/w/<Width>
                        /h/<Height>
                        /format/<Format>
                        /q/<Quality>
                        /rq/<Quality>
                        /lq/<Quality> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

> **注意**
> 质量变换参数仅针对 **jpg** 和 **webp** 格式图片。

版权所有：腾讯云计算（北京）有限责任公司

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>"
  }]
}
```

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。
- 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 。
- 当 fileid 与 ObjectKey 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- fileid 的值需要经过 URL 安全的 Base64 编码 。

处理参数说明

参数	含义	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg。	是
/0/w/<LongEdge>/h/<ShortEdge>	限定缩略图的长边最多为<LongEdge>，短边最多为<ShortEdge>，进行等比缩放，不裁剪。如果只指定w参数则表示限定长边（短边自适应），只指定h参数则表示限定短边（长边自适应）。 当图片长宽相同时，默认长边为图片的高。	否
/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	否
/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	否
/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。如果仅指定了一边的值，则另一边将默认设为相同的值。	否
/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 当图片长宽相同时，默认长边为图片的高。	否
/5/w/<LongEdge>/h/<S	限定缩略图的长边和短边的最大值分别为 LongEdge 和	否

hortEdge>	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。 当图片长宽相同时，默认长边为图片的高。	
/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。	否
/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。	否
/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%）。	否
/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。	否
/ignore-error/1	当处理参数中携带此参数时，针对文件过大、文件格式有误等外部因素导致的处理失败的场景，会直接返回原图而不报错。	否

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
```

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
```

版权所有：腾讯云计算（北京）有限责任公司

限制图片大小

最近更新时间：2023-03-13 14:12:18

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

```
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/st
```

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-algorith
```

> **说明**
> `<signature>` 为签名部分，获取方式请参考 请求签名 。

注意事项

为了避免未授权人员通过访问不携带处理参数的链接实现访问和下载原图的情况，您可同时将处理参数签入到请求签名中，处理参数整体是参数的 key，value 为空，如下是简单的示例（仅做样式参考，可能已经过期无法直接访问），详细计算方法请参见 请求签名 。

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

管道操作符

最近更新时间：2023-02-09 10:36:27

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；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：

```
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/t
```

图片显示如下：



使用管道操作符再添加文字水印的样式处理，则最终 URL 为：

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

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
```

支持字体列表

最近更新时间：2020-11-13 16:16:49

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

合成动图

最近更新时间：2023-06-27 10:41:41

功能概述

腾讯云数据万象通过 **imageMogr2** 接口提供合成功能，可将多张 jpg 或 png 格式的图片合成为一个 GIF 动图。该功能支持以下处理方式：

- 下载时处理
- 上传时处理
- 云上数据处理

限制说明

- 动图合成接口目前仅适用于存储在 COS 上的图片。
- 支持合成的原图格式：jpg/jpeg、png。
- 支持的最大帧数：20。
- 支持的最大图片尺寸：1080*1080。
- 原图限制：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。
- 合成后的动图限制：宽 x 高 x 帧数不超过2.5亿像素。

计费说明

该功能计入基础图片处理用量，其中处理用量按原图大小之和计算，例如使用了 10 张大小为 5MB 的 jpg 图片进行合成动图，则统计用量计为 50MB。有关基础图片处理的定价，请查看 图片处理费用 。

接口示例

1. 下载时处理

```
GET /<ObjectKey>?imageMogr2/animate/duration/<duration>/images/<imageurl1>/<im
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/animate/duration/<duration>/images/<imageurl1>/<imageurl2
  }]
}
```

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/animate/duration/<duration>/images/<imageurl1>/<imageurl2
  }]
}
```

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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码

处理参数说明

参数	参数说明	是否

		必填
ObjectKey	COS中的文件名称，例如 folder/sample.jpg，在这里将作为合成后的动图第一帧。	是
duration	GIF 动图的每帧间隔时间(单位: ms)，取值为大于 0 的整数。例如设置间隔为10ms，则参数为/duration/10。	是
images	待合成的图片原图链接，最多可添加19个，需要经过 URL安全Base64编码 。例如原图1的链接经过编码后的字符串为ahxxxxxxxxxmc，原图2的链接经过编码后的字符串为wywxxxxxxxxjih，则参数为/images/ahxxxxxxxxxmc/wywxxxxxxxxjih。	是

注意：
如果待合成的原图尺寸不相同，最终合成的动图将以第一帧的体积为标准，对所有图片进行等比缩放。

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**。

以存储桶examples-125xxxxxx中的image-1.jpg为动图首帧，和同个存储桶内的image-2.jpg合成为一个间隔50ms的动图。

```
http://examples-125xxxxxx.cos.ap-shanghai.myqcloud.com/image-1.jpg?imageMogr2/ani
```

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
```

获取原图

最近更新时间：2023-04-28 10:43:38

功能概述

如果您开启了图片原图功能，同时在某些场景下希望获取到原图，可通过本接口进行获取。

> **注意**
>
> 该请求操作不支持匿名请求，必须携带签名。

请求

请求示例

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

请求参数说明

参数	描述	类型	是否必选
ci-process	万象通用参数，需要获取原图的时，该参数固定为：originImage	string	是

请求头

该接口无特殊请求头。

请求体

该接口无请求体。

响应

响应头

该接口无特殊响应头。

响应体

该接口的响应体为对象（文件）内容。

实际案例

访问一张已开启了原图保护的存储桶中的图片，并携带获取原图参数：

```
http://example-1250000000.cos.ap-chengdu.myqcloud.com/?ci-process=originImage&q-s
```

> **说明**
> `<signature>` 为签名部分，获取方式请参考 请求签名，计算的签名中需将处理参数 `ci-process` 也一并签入。

图片基础压缩（WebP）

最近更新时间：2023-07-06 18:24:44

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

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

版权所有：腾讯云计算（北京）有限责任公司

```
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/fo
```

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%）

图片高级压缩
HEIF 压缩

最近更新时间：2023-06-05 15:07:44

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 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

```
GET /<ObjectKey>?imageMogr2/format/heif 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/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	描述	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg。	是
/format/<Format>	压缩格式，此处为 heif。	是

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/for
```

压缩率对比

格式	图片大小
png（原图）	1335.2KB
heif	52.87KB（压缩率96.0%）

版权所有：腾讯云计算（北京）有限责任公司

TPG 压缩

最近更新时间：2022-09-19 15:01:44

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 、Windows 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 。

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：

http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/for

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：2022-12-12 16:20:17

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>
```

版权所有：腾讯云计算（北京）有限责任公司

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：

```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageMogr2/for
```

压缩率对比

格式	图片大小

png（原图）	1335.2KB
avif	62.8KB（压缩率95.3%）

SVG 压缩

最近更新时间：2022-09-19 15:02:48

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
```

版权所有：腾讯云计算（北京）有限责任公司

```
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Guetzli 压缩
开通 Guetzli 压缩

最近更新时间：2023-03-07 09:51:51

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。

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：2023-03-07 09:53:33

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：2023-03-07 09:53:44

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。

图片极智压缩
开通极智压缩

最近更新时间：2023-07-14 14:47:01

功能描述

开通当前存储桶的图片极智压缩功能。

请求

请求示例

```
PUT /?image-slim 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。

请求头

公共头部

该请求操作无特殊的请求头部信息，使用的公共请求头，请参阅 公共请求头部 章节。

请求体

```
<ImageSlim>
    <SlimMode>Auto</SimMode>
    <Suffixs>
        <Suffix>jpg</Suffix>
        <Suffix>png</Suffix>
    </Suffixs>
</ImageSlim>
```

节点名	父节点	描述	类型	是否

称				必填
ImageSlim	无	极智压缩配置信息。	Container	是
SlimMode	ImageSlim	极智压缩的使用模式，包含两种： • API：开通极智压缩的 API 使用方式，开通后可在图片下载时通过极智压缩参数对图片进行压缩； • Auto：开通极智压缩的自动使用方式，开通后无需携带任何参数，存储桶内指定格式的图片将在访问时自动进行极智压缩。 注意：支持同时开通两种模式，多个值通过逗号分隔。	是	是
Suffixs	ImageSlim	当SlimMode的值包含Auto时生效，用于指定需要自动进行压缩的图片格式。	Container	否
Suffix	Suffixs	需要自动进行压缩的图片格式，可选值：jpg、png。	string	否，当SlimMode包含Auto时必填

响应

响应头

该响应无特殊的响应头，仅使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

响应体

该响应体返回为空。

实际案例

请求

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

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

```
<ImageSlim>
    <SlimMode>Auto,API</SimMode>
    <Suffixs>
        <Suffix>jpg</Suffix>
    </Suffixs>
</ImageSlim>
```

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
```

查询极智压缩状态

最近更新时间：2023-07-14 14:47:01

功能描述

该接口用于查询图片极智压缩功能的状态。

请求

请求示例

```
GET /?image-slim 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。

请求头

该请求操作仅使用了公共请求头，了解公共请求头详情，请参阅 公共请求头部 章节。

请求体

该请求的请求体为空。

响应

响应头

该响应仅使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

响应体

```
<?xml version="1.0" encoding="UTF-8" ?>
<ImageSlim>
  <Status>off</Status>
  <SlimMode>API</SlimMode>
  <Suffixs>
    <Suffix>jpg</Suffix>
```

```
    <Suffix>png</Suffix>
  </Suffixs>
</ImageSlim>
```

参数说明

节点名称（关键字）	父节点	描述
Status	ImageSlim	极智压缩的状态，包含值： on：极智压缩已开启； off：极智压缩已关闭。
SlimMode	ImageSlim	极智压缩的使用模式，包含两种： API：开通极智压缩的 API 使用方式，开通后可在图片下载时通过极智压缩参数对图片进行压缩； Auto：开通极智压缩的自动使用方式，开通后无需携带任何参数，存储桶内指定格式的图片将在访问时自动进行极智压缩。
Suffixs	ImageSlim	当SlimMode的值包含Auto时返回，表示需要自动进行压缩的图片格式。
Suffix	Suffixs	标识自动进行压缩的图片格式，包含值：jpg、png。

该接口支持json格式返回，当请求header指定 accept= application/json，响应体如下：

```
{
  "ImageSlim":{
    "Status":"on",
    "SlimMode":"Auto",
    "Suffixs":[
      {
        "Suffix":"jpg"
      },
      {
        "Suffix":"png"
      }
    ]
  }
}
```

关闭极智压缩

最近更新时间：2023-07-14 14:47:01

功能描述

该接口用于关闭图片极智压缩功能。

请求

请求语法

```
DELETE /?image-slim 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。

请求头

该请求操作仅使用公共请求头，了解公共请求头详情，请参阅 公共请求头部 章节。

请求体

该请求的请求体为空。

响应

响应头

该响应仅使用公共响应头，了解公共响应头详情，请参阅 公共响应头部 章节。

响应体

该响应体返回为空。

通过API使用极智压缩

最近更新时间：2023-07-10 09:47:51

功能描述

图片压缩指在图片质量保持不变的情况，尽可能的减小图片大小，以达到节省图片存储空间、减少图片访问流量、提升图片访问速度的效果。
极智压缩通过智能判断图片的主观质量进行自动调节，在不改变图片原格式的基础上，使图片体积相比原图有显著的降低，同时在视觉效果上可以最大程度贴近原图。

> **说明：**
> 该功能为付费服务，具体费用请参见 图片处理费用 。

限制说明

- 支持压缩的图片格式：PNG、JPG、JPEG
- 支持使用极智压缩的存储桶地域：北京、上海。

使用前提

通过控制台或接口开通极智压缩 API 使用功能。

使用方式

通过 imageSlim 参数在图片下载时处理。

接口示例

```
GET /<ObjectKey>?imageSlim 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

参数	含义

ObjectKey	对象文件名，例如 folder/sample.jpg。

实际案例

假设原图格式为 png，图片大小为1335.2KB，如下图所示：



将图片进行极智压缩处理，请求 URL 如下：

```
http://example-1258125638.cos.ap-shanghai.myqcloud.com/sample.png?imageSlim
```

效果如下：



压缩率对比

格式	图片大小
png（原图）	1335.2KB
png（极智压缩图）	549KB（压缩率58.88%）

图片明水印
图片水印

最近更新时间：2023-07-27 11:27:57

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。
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，开通 图片高级压缩功能 后，可支持AVIF、HEIF、TPG 格式的解码和处理操作。
- 静图体积限制：处理图片的原图及结果图的大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素。
- WebP图片限制：处理图片原图及结果图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：不超过300帧。

> **说明**
> - 图片水印为付费服务，费用同基础图片处理，具体费用请参见 图片处理费用 。
> - 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

```
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>"
  }]
}
```

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。
- Pic-Operations 为 json 格式的字符串，具体参数信息可参见 图片持久化处理 。
- 当 fileid 与 ObjectKey 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- fileid 的值需要经过 URL 安全的 Base64 编码 。

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	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg。	是
/image/	水印图片的链接，需要经过 URL 安全的 Base64 编码 。例如，水印图片为： http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/shuiyin_2.png ，则该处编码后的字符串为 aHR0cDovL2V4YW1wbGVidWNrZXQtMTI1MDAwMDAwMC5jb3MuYXAtc2hhbmdoYWkubXlxY2xvdWQuY29tL3NodWl5aW5fMi5wbmc	与/image_key/参数二选一
/image_key/	如果您在添加水印时不希望暴露水印所在的图片地址，可使用该参数。 该参数表示存储桶中水印图片所在的路径及名称，需要经过 URL 安全的 Base64 编码 。例如，水印图片在本存储桶中的路径及名称为： image/shuiyin_2.png 则该处编码后的字符串为 aW1hZ2Uvc2h1aXlpbl8yLnBuZw	与/image/参数二选一
/gravity/	图片水印位置，九宫格位置（ 参考九宫格方位图 ），默认值 SouthEast	否
/dx/	水平（横轴）边距，单位为像素，缺省值为0	否
/dy/	垂直（纵轴）边距，单位为像素，默认值为0	否
/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	否

	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	否
/spcent/	与 scatype 搭配使用： • 当 scatype 设置为1时，该有效值为[1,1000]，单位为千分比 • 当 scatype 设置为2时，该有效值为[1,1000]，单位为千分比 • 当 scatype 设置为3时，该有效值为[1,250]，单位为千分比。示例： `http://examples-1251000004.cos.ap-shanghai.myqcloud.com/sample.jpeg?watermark/1/image/xxxxxxx/scatype/3/spcent/250`	否
/dissolve/	图片水印的透明度，取值为1 - 100，默认90（90%不透明度）	否
/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	否
/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	否

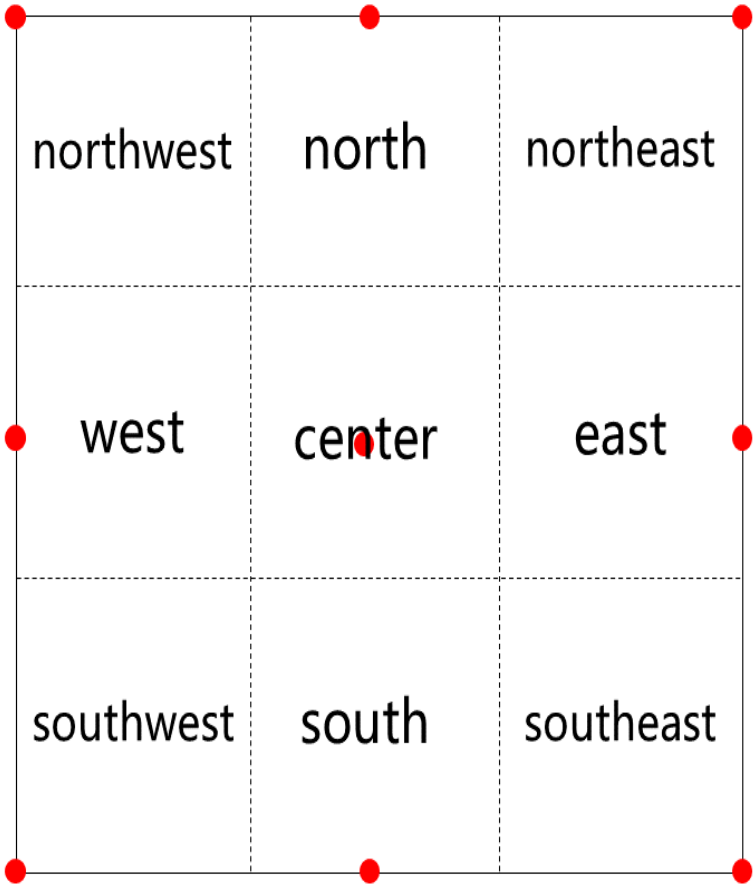
注意

指定的水印图片必须同时满足如下3个条件：

- **水印图片与原图片必须位于同一个存储桶下。**
- **水印图片链接需使用 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）。**
- **水印图片链接必须以 `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原点为参照）。



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/i
```

添加图片水印后效果如下：



案例二：添加图片水印并设置平铺、透明度

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

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
```

版权所有：腾讯云计算（北京）有限责任公司

文字水印

最近更新时间：2023-07-27 11:27:57

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。
该功能支持以下的处理方式：

- 下载时处理
- 上传时处理
- 云上数据处理

限制说明

- 支持格式：支持处理 JPG、BMP、GIF、PNG、WebP 格式，并且支持 HEIF 格式的解码和处理操作。
- 静图体积限制：处理图片原图大小不超过32MB、宽高不超过50000像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过50000像素。
- WebP图片限制：处理图片原图大小不超过32MB、宽高不超过16383像素且总像素不超过2.5亿像素，处理结果图宽高设置不超过16383像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

> **说明：**
> 文字水印为付费服务，费用同基础图片处理，具体费用请参见 图片处理费用 。

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
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>"
 }]
}
```

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。
- 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 。
- 当 `fileid` 与 `ObjectKey` 相同时，处理后的图片将覆盖原图存储；不相同时将分别存储原图和处理后图片。
- `fileid` 的值需要经过 URL 安全的 Base64 编码 。

处理参数说明

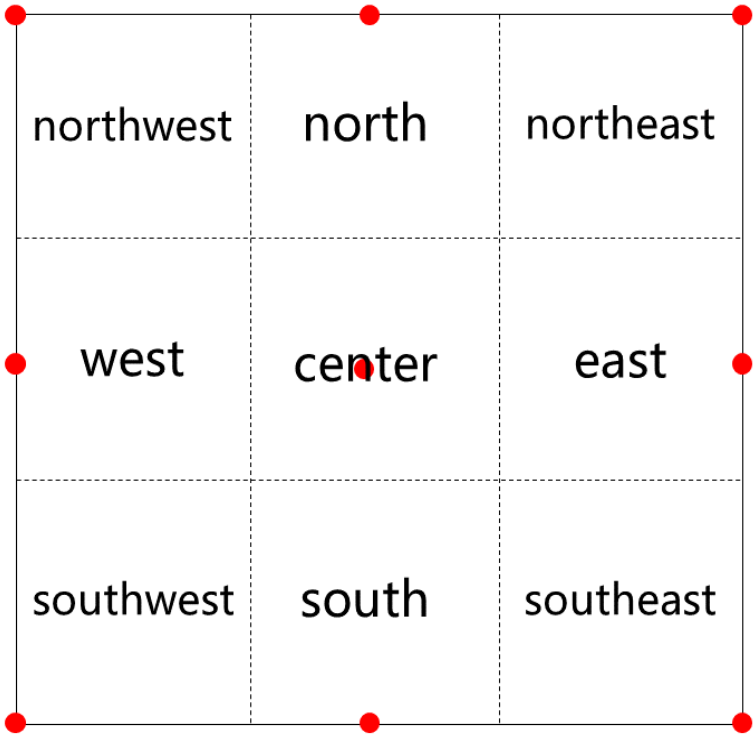
操作名称：watermark，数字2，表示水印类型为文字水印。

参数	含义	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg	是
/text/	水印内容，需要经过 URL 安全的 Base64 编码	是
/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	否
/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	否
/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	否

	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）	否
/gravity/	文字水印位置，九宫格位置（参见九宫格方位图），默认值 SouthEast	否
/dx/	水平（横轴）边距，单位为像素，缺省值为0	否
/dy/	垂直（纵轴）边距，单位为像素，默认值为0	否
/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	否
/spacing/	平铺模式下的水平、垂直间距相对文字水印贴图的宽高百分比，范围为[0,100]，默认10	否
/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	否
/shadow/	文字阴影效果，有效值为[0,100]，默认为0，表示无阴影	否
/scatype/	根据原图的大小，缩放调整文字水印的大小： • 当 scatype 设置为1时，按原图的宽缩放 • 当 scatype 设置为2时，按原图的高缩放 • 当 scatype 设置为3时，按原图的整体面积缩放	否
/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	否

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原点为参照）。



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：添加文字水印

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

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
```

图片盲水印

最近更新时间：2023-05-25 14:50:03

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，这样的水印不会影响图片的观感，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图片半盲水印、图片全盲水印和文字盲水印三种类型：

水印类型	特性	适用场景
图片半盲水印（type1）	抗攻击性强，但提取水印需原图	小图（640px x 640px以下）使用
图片全盲水印（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。数据万象在每个账户首次产生该服务用量后，将发放一个用量为6000次，有效期为2个月的免费额度资源包，超出用量或资源包到期后将正常计费。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图片半盲水印，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图片全盲水印提取，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）；若不存在图片全盲水印则添加图片全盲水印，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为图片半盲水印；2为图片全盲水印；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，	String	否

	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	只对图片全盲水印（type=2）有效。level 的取值范围为{1,2,3}，默认值为1，level 值越大则图片受影响程度越大、盲水印效果越好。	Int	否

请求参数

此接口无请求参数。

请求体

此接口请求的请求体为需要添加盲水印的原文件（图片）内容。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

响应体具体数据内容如下：

参数名称	描述	类型
UploadResult	原图信息	Container

UploadResult 节点内容：

参数名称	描述	类型
OriginalInfo	原图信息	Container
ProcessResults	图片处理结果	Container

OriginalInfo 节点内容：

节点名称	描述	类型
Key	原图文件名	String

Location	图片路径	String
ImageInfo	原图图片信息	Container

ImageInfo 节点内容：

节点名称	描述	类型
Format	格式	String
Width	图片宽度	Int
Height	图片高度	Int
Quality	图片质量	Int
Ave	图片主色调	String
Orientation	图片旋转角度	Int

ProcessResults 节点内容：

节点名称	描述	类型
Object	每一个图片处理结果	Container

Object 节点内容：

节点名称	描述	类型
Key	文件名	String
Location	图片路径	String
Format	图片格式	String
Width	图片宽度	Int
Height	图片高度	Int
Size	图片大小	Int
Quality	图片质量	Int

实际案例

请求

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

Pic-Operations:
{
   "is_pic_info": 1,
   "rules": [{
      "fileid": "exampleobject",
      "rule": "watermark/3/type/1/image/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQu
   }]
}
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/exar
              <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为图片半盲水印；2为图片全盲水印；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. 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-	String	否

	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	只对图片全盲水印（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?waterm
```

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	Arra	否

		y	

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为图片半盲水印；2为图片全盲水印；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	只对图片全盲水印（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:XXXXXXXXXXX
```

版权所有：腾讯云计算（北京）有限责任公司

```
Pic-Operations:
{
   "is_pic_info": 1,
   "rules": [{
      "fileid": "exampleobject",
      "rule": "watermark/3/type/1/image/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQu
   }]
}
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/exar
              <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为图片半盲水印；2为图片全盲水印；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： • 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	String	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，表示提取到图片全盲水印的可信度。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/aHR0cDovL2V4YW1wbGVzLTEyNTEwMDAwMDQu
   }]
}
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/exampleo
        <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/examp
            <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。

版权所有：腾讯云计算（北京）有限责任公司

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
异常图片检测同步请求

最近更新时间：2023-08-04 14:45:42

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。

> ⚠ **注意：**
> 异常图片检测功能为收费项，由数据万象收取，详细的计费说明请参见数据万象 图片处理费用 。

限制说明

- 支持格式：JPG/JPEG、PNG、BMP、GIF，以及纯TS视频流（更改为任意后缀）。
- 静态图体积限制：可检测的原图大小不超过32MB，宽高不超过50000像素且总像素不超过2.5亿像素。
- 动图体积限制：原图及处理图宽 x 高 x 帧数不超过2.5亿像素。
- 动图帧数限制：gif 帧数限300帧。

接口示例

```
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	类型	是否必选
ObjectKey	对象文件名，例如 folder/sample.jpg	String	是

响应参数

参数	含义	类型
picSize	检测的原图大小，单位为 Bytes	Strin

版权所有：腾讯云计算（北京）有限责任公司

		g
picType	检测的原图类型。例如 jpg、png 等	String
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	Boolean
suspiciousBeginByte	检测到的可疑文件的起始字节位置，单位为 Bytes	String
suspiciousEndByte	检测到的可疑文件的末尾字节位置，单位为 Bytes	String
suspiciousSize	检测到的可疑文件的大小	String
suspiciousType	检测到的可疑文件的类型，例如 MPEG-TS	String

实际案例

该接口需要携带签名进行使用，签名与检测参数以“&”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，需要将检测参数也签入签名中，获取方式请参考 请求签名 。

异常图片检测异步任务
提交任务

最近更新时间：2023-08-17 17:55:43

功能描述

提交一个异常图片检测任务。

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>ImageInspect</Tag>
  <Input>
    <Object>input/test.png</Object>
  </Input>
  <Operation>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
```

```
    </Operation>
    <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：ImageInspect	String	是
Input	Request	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	否
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
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
ImageInspect	Request.Operation	该任务的参数	Container	否

Container 类型 ImageInspect 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
AutoProcess	Request.Operation.ImageInspect	是否开启识别到图片异常后自动对图片进行如移动到其他目录、设置为私有权限、删除等动作。取值：true/false，默认为false	String	否
ProcessType	Request.Operation.ImageInspect	指定检测到异常图片后的处理动作 • BackupObject：将图片移动abnormal_images_backup下，该目录由后台自动创建 • SwitchObjectToPrivate：将图片权限设置为私有 • DeleteObject：删除图片 • 默认值：BackupObject • 当 AutoProcess 为 true 时，该参数生效	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	文件路径	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>cda071f3af9e011ed8f900133752da9af</JobId>
    <State>Submitted</State>
    <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
    <StartTime>-</StartTime>
    <EndTime>-</EndTime>
    <QueueId>p3a3b198088e74b76b31d553a77b656be</QueueId>
    <Tag>ImageInspect</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/test.png</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 数组

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：ImageInspect	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String
ImageInspect	Response.JobsDetail.Operation	同请求中的 Request.Operation.ImageInspect	Container
ImageInspectResult	Response.JobsDetail.Operation	异常图片检测结果，任务未完成时不返回	Container

Container 节点 ImageInspectResult 的内容：

节点名称（关键字）	父节点	描述	类型
State	Response.JobsDetail.Operation.ImageInspectResult	当前节点的任务状态，为 Success、Failed 其中一个	String
Code	Response.JobsDetail.Operation.ImageInspectResult	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail.Operation.ImageInspectResult	错误描述，只有 State 为 Failed 时有意义	String
InputObjectName	Response.JobsDetail.Operation.ImageInspectResult	当前节点的输入对象名	String
InputObjectUrl	Response.JobsDetail.Operation.ImageInspectResult	当前节点的输入对象Url	String
ProcessResult	Response.JobsDetail.Operation.ImageInspectResult	当前节点的处理结果详情	Container

Container 节点 ProcessResult 的内容：

节点名称（关	父节点	描述	类型

键字）			
PicSize	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测的原图大小，单位为 Bytes	Int
PicType	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测的原图类型。例如 jpg、png 等	String
Suspicious	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	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	String
SuspiciousBeginByte	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测到的可疑文件的起始字节位置，单位为 Bytes	Int
SuspiciousEndByte	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测到的可疑文件的末尾字节位置，单位为 Bytes	Int
SuspiciousSize	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测到的可疑文件的大小	Int
SuspiciousType	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测到的可疑文件的类型，例如 MPEG-TS	String
AutoProcessResult	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult	检测到为可疑文件后的图片自动处理结果	Container

Container 节点 AutoProcessResult 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult.AutoProcessResult	自动处理结果的错误码，为 Success、Failed 其中一个	String
Message	Response.JobsDetail.Operation.ImageInspectResult.ProcessResult.AutoProcessResult	自动处理结果的错误信息，当State为Failed时有效	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>ImageInspect</Tag>
    <Input>
        <Object>input/test.png</Object>
    </Input>
    <Operation>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>cda071f3af9e011ed8f900133752da9af</JobId>
        <State>Submitted</State>
        <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
```

```
        <StartTime>-</StartTime>
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>ImageInspect</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/test.png</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

任务回调

最近更新时间：2023-08-04 16:46:42

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，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2023-05-24T11:12:44+0800</CreationTime>
    <EndTime>2023-05-24T11:12:45+0800</EndTime>
    <Input>
        <BucketId>test-1253960000</BucketId>
        <Object>test.png</Object>
        <Region>ap-chongqing</Region>
    </Input>
    <JobId>cda071f3af9e011ed8f900133752da9af</JobId>
    <Message/>
    <Operation>
      <ImageInspectResult>
        <Code>Success</Code>
        <InputObjectName>test.png</InputObjectName>
        <InputObjectUrl>https://.cos.ap-chongqing.myqcloud.com/test.png</InputObje
        <Message/>
        <ProcessResult>
          <PicSize>304491</PicSize>
          <PicType>png</PicType>
          <Suspicious>true</Suspicious>
          <SuspiciousBeginByte>119</SuspiciousBeginByte>
          <SuspiciousEndByte>304490</SuspiciousEndByte>
          <SuspiciousSize>304372</SuspiciousSize>
          <SuspiciousType>MPEG-TS</SuspiciousType>
          <AutoProcessResult>
            <Code>Success</Code>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                <Message>Success</Message>
              </AutoProcessResult>
            </ProcessResult>
            <State>Success</State>
          </ImageInspectResult>
          <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>p3a3b198088e74b76b31d553a77b656be</QueueId>
        <StartTime>2023-05-24T11:12:45+0800</StartTime>
        <State>Success</State>
        <Tag>ImageInspect</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	同提交异常图片检测任务接口中的 Response.JobsDetail	Container 数组

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。

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

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>2023-05-24T11:12:44+0800</CreationTime>
    <EndTime>2023-05-24T11:12:45+0800</EndTime>
    <Input>
        <BucketId>test-1253960000</BucketId>
        <Object>test.png</Object>
        <Region>ap-chongqing</Region>
    </Input>
    <JobId>cda071f3af9e011ed8f900133752da9af</JobId>
    <Message/>
    <Operation>
      <ImageInspectResult>
        <Code>Success</Code>
        <InputObjectName>test.png</InputObjectName>
        <InputObjectUrl>https://test-1253960000.cos.ap-chongqing.myqcloud.com/tes
        <Message/>
        <ProcessResult>
          <PicSize>304491</PicSize>
          <PicType>png</PicType>
          <Suspicious>true</Suspicious>
          <SuspiciousBeginByte>119</SuspiciousBeginByte>
          <SuspiciousEndByte>304490</SuspiciousEndByte>
          <SuspiciousSize>304372</SuspiciousSize>
          <SuspiciousType>MPEG-TS</SuspiciousType>
          <AutoProcessResult>
            <Code>Success</Code>
```

```
            <Message>Success</Message>
          </AutoProcessResult>
        </ProcessResult>
        <State>Success</State>
      </ImageInspectResult>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p3a3b198088e74b76b31d553a77b656be</QueueId>
    <StartTime>2023-05-24T11:12:45+0800</StartTime>
    <State>Success</State>
    <Tag>ImageInspect</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": "2023-05-24T14:24:07+0800",
      "EndTime": "2023-05-24T14:24:08+0800",
      "Input": {
        "BucketId": "testpic-1253960000",
        "Object": "test.png",
        "Region": "ap-chongqing"
      },
      "JobId": "c966a67d0f9fb11eda05029775a03955f",
      "Message": "",
      "Operation": {
        "ImageInspectResult": [
          {
            "Code": "Success",
            "InputObjectName": "test.png",
            "InputObjectUrl": "https://testpic-1253960000.cos.ap-chongqing.myqcloud
            "Message": "",
            "ProcessResult": {
              "PicSize": 304491,
              "PicType": "png",
              "Suspicious": true,
              "SuspiciousBeginByte": 119,
              "SuspiciousEndByte": 304490,
```

```
                    "SuspiciousSize": 304372,
                    "SuspiciousType": "MPEG-TS"
                },
                "State": "Success"
            }
        ],
        "JobLevel": "0",
    },
    "QueueId": "p3a3b198088e74b76b31d553a77b656be",
    "StartTime": "2023-05-24T14:24:08+0800",
    "State": "Success",
    "Tag": "ImageInspect"
  }
 ]
}
```

图片异步处理
提交任务

最近更新时间：2023-06-20 17:32:21

功能描述

提交一个图片处理任务。

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>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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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	模板 ID	String	否
PicProcess	Request.Operation	该任务的参数，同创建图片处理模板接口中的 Request.PicProcess	Container	否
Output	Request.Operation	结果输出配置	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 PicProcess。

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

版权所有：腾讯云计算（北京）有限责任公司

JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
JobId	Response.JobsDetail	新创建任务的 ID	String
Tag	Response.JobsDetail	新创建任务的 Tag：PicProcess	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.JobsDetail	任务的开始时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
QueueId	Response.JobsDetail	任务所属的队列 ID	String
Input	Response.JobsDetail	该任务的输入资源地址	Container
Operation	Response.JobsDetail	该任务的规则	Container

Container 节点 Input 的内容：

节点名称（关键字）	父节点	描述	类型
Region	Response.JobsDetail.Input	存储桶的地域	String

Bucket	Response.JobsDetail.Input	存储结果的存储桶	String
Object	Response.JobsDetail.Input	输出结果的文件名	String

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.JobsDetail.Operation	任务的模板 ID	String
TemplateName	Response.JobsDetail.Operation	任务的模板名称，当 TemplateId 存在时返回	String
PicProcess	Response.JobsDetail.Operation	同请求中的 Request.Operation.PicProcess	Container
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
PicProcessResult	Response.JobsDetail.Operation	图片处理后输出图片的信息，没有时不返回	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String

Container 节点 PicProcessResult 的内容：
同 图片持久化处理 接口中的 UploadResult 节点。

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求1：使用图片处理模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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-1234567890</Bucket>
      <Object>output/out.jpg</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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-12345678900</BucketId>
      <Object>input/deer.jpg</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <Output>
        <Bucket>test-12345678900</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 /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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-1234567890</Bucket>
       <Object>output/out.jpg</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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>
```

任务回调

最近更新时间：2023-02-07 14:08:52

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            </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"
      }
    }]
}
```

创建模板

最近更新时间：2023-08-17 11:05:43

功能描述

创建图片处理模板。

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>PicProcess</Tag>
   <Name>TemplateName</Name>
   <PicProcess>
      <IsPicInfo>true</IsPicInfo>
      <ProcessRule>imageMogr2/rotate/90</ProcessRule>
   </PicProcess>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

具体数据描述如下：

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：PicProcess	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64	String	是
PicProcess	Request	图片处理参数	Container	是

Container 类型PicProcess 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
IsPicInfo	Request.PicProcess	是否返回原图信息，取值 true/false	String	false	否
ProcessRule	Request.PicProcess	图片处理规则 • 基础图片处理参见 基础图片处理 文档 • 图片压缩参见 图片压缩 文档 • 盲水印参见 盲水印 文档	String	无	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>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <Template>
```

```
      <Tag>PicProcess</Tag>
      <TemplateId>t1460606hgd52148c4ab182ffds5163ba7cd</TemplateId>
      <Name>TemplateName</Name>
      <BucketId>test-1234567890</BucketId>
      <Category>Custom</Category>
      <PicProcess>
        <IsPicInfo>true</IsPicInfo>
        <ProcessRule>imageMogr2/rotate/90</ProcessRule>
      </PicProcess>
      <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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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，PicProcess	String

版权所有：腾讯云计算（北京）有限责任公司

UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
PicProcess	Response.Template	同请求体中的 Request.PicProcess	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-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>PicProcess</Tag>
    <Name>TemplateName</Name>
    <PicProcess>
        <IsPicInfo>true</IsPicInfo>
        <ProcessRule>imageMogr2/rotate/90</ProcessRule>
    </PicProcess>
</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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <Template>
```

```
      <Tag>PicProcess</Tag>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <Name>TemplateName</Name>
      <BucketId>test-1234567890</BucketId>
      <Category>Custom</Category>
      <PicProcess>
        <IsPicInfo>true</IsPicInfo>
        <ProcessRule>imageMogr2/rotate/90</ProcessRule>
      </PicProcess>
      <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
      <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

更新模板

最近更新时间：2023-06-20 17:32:26

功能描述

更新图片处理模板。

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>PicProcess</Tag>
    <Name>TemplateName</Name>
    <PicProcess>
        <IsPicInfo>true</IsPicInfo>
        <ProcessRule>imageMogr2/rotate/90</ProcessRule>
    </PicProcess>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建图片处理模板接口的 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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <Template>
    <Tag>PicProcess</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <PicProcess>
      <IsPicInfo>true</IsPicInfo>
      <ProcessRule>imageMogr2/rotate/90</ProcessRule>
    </PicProcess>
    <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	同创建图片处理模板接口的 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-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>PicProcess</Tag>
    <Name>TemplateName</Name>
    <PicProcess>
        <IsPicInfo>true</IsPicInfo>
        <ProcessRule>imageMogr2/rotate/90</ProcessRule>
    </PicProcess>
</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>
    <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
    <Template>
        <Tag>PicProcess</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <BucketId>test-12345678900</BucketId>
        <Category>Custom</Category>
        <PicProcess>
            <IsPicInfo>true</IsPicInfo>
            <ProcessRule>imageMogr2/rotate/90</ProcessRule>
        </PicProcess>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

海报合成任务
海报合成模板接口
查询模板接口
查询海报模板

最近更新时间：2023-03-20 17:59:53

功能描述

查询海报合成模板接口。

请求

请求示例

```
GET /posterproduction/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

此接口仅使用公共请求头部，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>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
  <Template>
    <TemplateId>A</TemplateId>
    <Height></Height>
    <Width></Width>
    <Name></Name>
    <Thumb></Thumb>
  </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
RequestId	Response	请求的唯一 ID	String
Template	Response	模板具体可替换参数，该 Container 中的参数根据具体模板中可替换的参数返回	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

```
GET /posterproduction/template/62e27c0cd73bf6bb0e903721 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

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: 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>
    <Template>
        <TemplateId>62e27c0cd73bf6bb0e903721</TemplateId>
        <width>1280</width>,
        <height>720</height>
        <name>央视频下载观看1<name>
        <thumb>https://static.taishan.qq.com/xrdnest/psd/e9c62e867425f9188f86a319b1f
    </Template>
</Response>
```

批量拉取海报模板

最近更新时间：2022-12-22 10:07:37

功能描述

查询海报合成模板列表接口。

请求

请求示例

```
GET /posterproduction/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
pageNumber	无	第几页，默认值:1	Integer	否
pageSize	无	每页个数，默认值:10	Integer	否
categoryId	无		Stri	否

s		模板分类ID，支持传入多个，以 , 符号分割字符串	ng	
type	无	Official(系统预设模板)，Custom(自定义模板)，All(所有模板)，默认值: Custom	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>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TotalCount>1</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <TemplateList>
        <TemplateId>A</TemplateId>
        <Height></Height>
        <Width></Width>
        <Name></Name>
        <Thumb></Thumb>
    </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中参数为模板具体参数	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

```
GET /posterproduction/template HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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>62e27c0cd73bf6bb0e903721</TemplateId>
        <width>1280</width>,
        <height>720</height>
        <name>央视频下载观看1<name>
        <thumb>https://static.taishan.qq.com/xrdnest/psd/e9c62e867425f9188f86a319b1f
    </TemplateList>
```

```
    <TemplateList>
      <TemplateId>62e279bad73bf6bb0e903707</TemplateId>
      <width>1125</width>,
      <height>1125</height>
      <name>0点折上折<name>
      <thumb>https://static.taishan.qq.com/xrdnest/psd/e9c62e867425f9188f86a319b1f
    </TemplateList>
</Response>
```

删除模板接口

最近更新时间：2023-03-23 10:51:15

功能描述

删除模板。

请求

请求实例

```
DELETE /posterproduction/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。

版权所有：腾讯云计算（北京）有限责任公司

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 /posterproduction/template/t143ae6e040af6431aa772c9ec3f0a3f36 HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR98****-s
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: NjJjYmU0YmRfOTBmYTUwNjRfODhjXzE=

<Response>
    <RequestId>NjJjYmU0YmRfOTBmYTUwNjRfODhjXzE=</RequestId>
    <TemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</TemplateId>
<Response>
```

上传文件解析模板接口

最近更新时间：2023-03-23 10:51:19

功能描述

解析保存 PSD 文件模板。

请求

请求实例

```
POST /posterproduction/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>
  <Input>
    <Object></Object>
  </Input>
```

```
    <Name></Name>
</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
Input	Request	输入参数	Container	是
Name	Request	模板名称	String	是
CategoryIds	Request	模板分类 ID，支持传入多个，以 `,` 符号分割字符串	String	否

Container 类型 Input 的具体数据d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
Object	Request.Input	COS 桶中 PSD 文件，大小限制100M	String	是	无	无

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></RequestId>
  <Template>
    <Info>
      <Backgroud></Backgroud>
      <EXPIRY></EXPIRY>
      <MM></MM>
      <YY></YY>
      <base></base>
    </Info>
    <TemplateId></TemplateId>
  </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
RequestId	Response	请求 ID	Container
Template	Response	解析出的 PSD 文件的具体内容	Container

Container 节点 Template 的内容：

节点名称（关键字）	父节点	描述	类型
Info	Response	模板内容	Container
TemplateId	Response	模板 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

```
POST /posterproduction/template HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR98****-s
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Input>
        <Object>demo.psd</Object>
    </Input>
    <Name></Name>
    <CategoryIds>61849dcf18044201b378f699,61849dcf18044201b378f69b</CategoryI
</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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <RequestId>NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=</RequestId>
    <Template>
        <Info>
            <Background>https://background.png</Background>
            <EXPIRY>EXPIRY</EXPIRY>
            <MM>MM</MM>
            <YY>YY</YY>
            <base>https://example.png</base>
        </Info>
        <TemplateId>634f9a1a412c273727f440fd</TemplateId>
    </Template>
</Response>
```

海报合成任务接口
查询海报合成任务
查询海报合成任务

最近更新时间：2023-03-07 09:41:54

功能描述

查询指定的任务。

请求

请求示例

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

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <JobsDetail>
  </JobsDetail>
  <NonExistJobIds></NonExistJobIds>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

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，请参考 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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>c93984788066911ed89ed352d4d9d2084</JobId>
    <State>Submitted</State>
    <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
    <EndTime>-</EndTime>
    <StartTime>-</StartTime>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <Tag>PosterProduction</Tag>
    <Operation>
      <PosterProduction>
        <TemplateId>6324349b569067d4d11a2c2c</TemplateId>
        <Info>
          <Background>https://static.taishan.qq.com/xrdnest/pstemplate/assets/e96c
          <EXPIRY>EXPIRY</EXPIRY>
          <MM>MM</MM>
          <YY>YY</YY>
          <base>https://static.taishan.qq.com/xrdnest/pstemplate/assets/ba2064980
        </Info>
      </PosterProduction>
      <Output>
        <Bucket>test-1234567890</Bucket>
        <Object>output/out.png</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
      <TemplateName>test</TemplateName>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

批量拉取海报合成任务

最近更新时间：2023-03-07 09:41:09

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
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	Strin	否

			g	
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
图片处理	PosterProduction

响应

响应头

此接口仅返回公共响应头部，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
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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
```

版权所有：腾讯云计算（北京）有限责任公司

```
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>PosterProduction</Tag>
    <Operation>
      <PosterProduction>
        <TemplateId>6324349b569067d4d11a2c2c</TemplateId>
        <Info>
          <Background>https://static.taishan.qq.com/xrdnest/pstemplate/assets/e96c
          <EXPIRY>EXPIRY</EXPIRY>
          <MM>MM</MM>
          <YY>YY</YY>
          <base>https://static.taishan.qq.com/xrdnest/pstemplate/assets/ba2064980
        </Info>
      </PosterProduction>
      <Output>
        <Bucket>test-1234567890</Bucket>
        <Object>output/out.png</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>PosterProduction</Tag>
      <Operation>
        <PosterProduction>
          <TemplateId>6324349b569067d4d11a2c2c</TemplateId>
          <Info>
            <Background>https://static.taishan.qq.com/xrdnest/pstemplate/assets/e96c
            <EXPIRY>EXPIRY</EXPIRY>
            <MM>MM</MM>
            <YY>YY</YY>
            <base>https://static.taishan.qq.com/xrdnest/pstemplate/assets/ba2064980
          </Info>
        </PosterProduction>
        <Output>
          <Bucket>test-1234567890</Bucket>
          <Object>output/out.png</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
        <TemplateName>test</TemplateName>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

取消海报合成任务

最近更新时间：2023-06-19 16:47:02

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。

版权所有：腾讯云计算（北京）有限责任公司

实际案例

请求

```
PUT /jobs/c93984788066911ed89ed352d4d9d2084?cancel HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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=
```

版权所有：腾讯云计算（北京）有限责任公司

提交海报合成任务

最近更新时间：2023-06-21 16:50:53

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>PosterProduction</Tag>
  <Operation>
    <Output>
      <Region>ap-chongqing</Region>
      <Bucket>test-123456789</Bucket>
      <Object>output/out.jpg</Object>
    </Output>
    <PosterProduction>
      <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Info>
          <main>https://test-123456789.cos.ap-chongqing.myqcloud.com/demo.jpg</m
          <text_main>demo</text_main>
        </Info>
      </PosterProduction>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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	否
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
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
CallBackMq	Requ	任务回调TDMQ配置，当 CallBackType 为	Cont	否

Config	est	TDMQ 时必填。详情见 CallBackMqConfig	ainer	

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
PosterProduction	Request.Operation	指定该任务的参数	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

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

节点名称（关键字）	父节点	描述	类型	是否必选
TemplateId	Request.Operation.PosterProduction	海报合成任务模板 ID	String	是
Info	Request.Operation.PosterProduction	模板可替换信息，同 查询海报合成模板接口 中的 Response.Template ，Info 中的图片 URL 需要保证可访问，如有特殊字符需要 URLencode	Container	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.Out	存储桶的地域	String	是

	put			
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>PosterProduction</Tag>
    <Operation>
      <Output>
        <Bucket>test-1234567890</Bucket>
        <Object>output/out.jpg</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <PosterProduction>
        <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
      </PosterProduction>
      <TemplateName>test</TemplateName>
      <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

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	任务的详细信息	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时有意义	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时有意义	String
JobId	Response.JobsDetail	新创建任务的 ID	String
Tag	Response.JobsDetail	新创建任务的 Tag：PicProcess	String
State	Response.JobsDetail	任务的状态，为 Submitted、Running、Success、Failed、Pause、Cancel 其中一个	String
CreationTime	Response.JobsDetail	任务的创建时间	String
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

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：

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
PosterProduction	Response.JobsDetail.Operation	海报合成参数	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

Container 节点 PosterProdcution 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TemplateId	Request.Operation.PosterProduction	海报合成任务模板 ID	String	否

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>PosterProduction</Tag>
    <Operation>
        <PosterProduction>
            <TemplateId>6324349b569067d4d11a2c2c</TemplateId>
            <Info>
                <Background>https://static.taishan.qq.com/xrdnest/pstemplate/assets/e96c14
                <EXPIRY>EXPIRY</EXPIRY>
                <MM>MM</MM>
                <YY>YY</YY>
                <base>https://static.taishan.qq.com/xrdnest/pstemplate/assets/ba206498074
            </Info>
        </PosterProduction>
        <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>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8z****

<Response>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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>PosterProduction</Tag>
      <Operation>
        <PosterProduction>
          <TemplateId>6324349b569067d4d11a2c2c</TemplateId>
          <Info>
            <Background>https://static.taishan.qq.com/xrdnest/pstemplate/assets/e96c
            <EXPIRY>EXPIRY</EXPIRY>
            <MM>MM</MM>
            <YY>YY</YY>
            <base>https://static.taishan.qq.com/xrdnest/pstemplate/assets/ba2064980
          </Info>
        </PosterProduction>
        <Output>
          <Bucket>test-1234567890</Bucket>
          <Object>output/out.png</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t10461fe2bd5a649db9022452ec71e0381</TemplateId>
        <TemplateName>test</TemplateName>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

异步服务和队列配置
开通图片处理（异步）服务

最近更新时间：2023-08-01 16:46:27

功能描述

本接口用于开通图片处理（异步）服务并生成队列。

请求

请求示例

```
POST /picbucket 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>
    <PicBucket>
        <BucketId>test-1234567890</BucketId>
        <Name>test-1234567890</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
    </PicBucket>
</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
PicBucket	Response	存储桶信息	Container

Container 节点 PicBucket 的内容：

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

实际案例

请求

```
POST /picbucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

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: 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>
    <PicBucket>
        <BucketId>test-1234567890</BucketId>
        <Name>test-1234567890</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
    </PicBucket>
</Response>
```

查询图片处理（异步）服务

最近更新时间：2023-08-01 16:46:27

功能描述

本接口用于查询已经开通图片处理（异步）功能的存储桶。

请求

请求示例

```
GET /picbucket 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	默认值	是否必填
regions	地域信息，例如 ap-shanghai、ap-beijing，若查询多个地域以“,”分隔字符串，详情请参见 地域与域名	无	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	无	否
bucketNam	存储桶名称前缀，前缀搜索	无	否

e			
pageNumber	第几页	1	否
pageSize	每页个数，大于0且小于等于100 的整数	10	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>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <PicBucketList>
    <BucketId>examplebucket-1250000000</BucketId>
    <Name>examplebucket-1250000000</Name>
    <Region>ap-beijing</Region>
    <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
  </PicBucketList>
  <PicBucketList>
    <BucketId>examplebucket-1-1250000000</BucketId>
    <Name>examplebucket-1-1250000000</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
  </PicBucketList>
</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	开通图片处理（异步）服务的 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
PicBucketList	Response	异步图片处理 Bucket 详情	Container 数组

Container 节点 PicBucketList 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.PicBucketList	存储桶 ID	String
Name	Response.PicBucketList	存储桶名称，同 BucketId	String
Region	Response.PicBucketList	所在的地域	String
CreateTime	Response.PicBucketList	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 /picbucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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 2021 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>
    <PicBucketList>
        <BucketId>examplebucket-1250000000</BucketId>
        <Name>examplebucket-1250000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </PicBucketList>
    <PicBucketList>
        <BucketId>examplebucket-1-1250000000</BucketId>
        <Name>examplebucket-1-1250000000</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
    </PicBucketList>
</Response>
```

关闭图片处理（异步）服务

最近更新时间：2023-08-01 16:46:27

功能描述

本接口用于关闭图片处理（异步）服务并删除队列。

请求

请求示例

```
DELETE /picbucket 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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
  <BucketName>test-1234567890</BucketName>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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
BucketName	Response	存储桶名称	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 /picbucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
    <BucketName>test-1234567890</BucketName>
```

```
</Response>
```

查询图片处理（异步）队列

最近更新时间：2023-08-01 16:46:22

功能描述

本接口用于查询图片处理（异步）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	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
  <TotalCount>1</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <QueueList>
    <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</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>
    <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
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

版权所有：腾讯云计算（北京）有限责任公司

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	String
Type	Response.QueueList.NotifyConfig	回调类型	String
Event	Response.QueueList.NotifyConfig	回调事件	String
ResultFormat	Response.QueueList.NotifyConfig	回调类型	String
MqMode	Response.QueueList.NotifyConfig	TDMQ 使用模式	String
MqRegion	Response.QueueList.NotifyConfig	TDMQ 所属园区	String
MqName	Response.QueueList.NotifyConfig	TDMQ 主题名称	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 /picqueue?queueIds=p4a17eeea29334bf499b7e20e2fbfd99d,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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: NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85

<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</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>
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

更新图片处理（异步）队列

最近更新时间：2023-08-01 16:46:22

功能描述

本接口用于更新图片处理（异步）的队列。

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	默认值	是否必选
Name	Request	队列名称，仅支持中文、英文、数字、_、-和*，长度不超过 128	String	无	是
State	Request	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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	无	是
NotifyConfig	Request	回调配置	Container	无	是

Container 节点 NotifyConfig 的内容：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
State	Request.NotifyConfig	回调开关 • Off • On	String	Off	否
Event	Request.NotifyConfig	回调事件 • TaskFinish：任务完成 • WorkflowFinish：工作流完成	String	无	当 State=On 时，必选
ResultFormat	Request.NotifyConfig	回调格式 • XML • JSON	String	XML	否

Type	Request.NotifyConfig	回调类型 • Url • TDMQ	String	无	当 State=On 时，必选
Url	Request.NotifyConfig	回调地址，不能为内网地址。	String	无	当 State=On，且 Type=Url 时，必选
MqMode	Request.NotifyConfig	TDMQ 使用模式 • Topic：主题订阅 • Queue: 队列服务	String	无	当 State=On，且 Type=TDMQ 时，必选
MqRegion	Request.NotifyConfig	TDMQ 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	无	当 State=On，且 Type=TDMQ 时，必选
MqName	Request.NotifyConfig	TDMQ 主题名称	String	无	当 State=On，且 Type=TDMQ 时，必选

响应

响应头

此接口仅返回公共响应头部，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-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>
        <MqMode/>
        <MqName/>
        <MqRegion/>
      </NotifyConfig>
      <MaxSize>10000</MaxSize>
      <MaxConcurrent>10</MaxConcurrent>
      <Category>PicProcessing</Category>
      <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 /file_queue/p4a17eeea29334bf499b7e20e2fbfd99d HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>p4a17eeea29334bf499b7e20e2fbfd99d</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>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <Category>PicProcessing</Category>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

媒体处理
媒体处理机制介绍

最近更新时间：2023-06-21 16:50:53

简介

数据万象提供了一套完善便捷的媒体处理机制，通过这套机制可以对存储在对象存储 COS 中的媒体对象进行各类型处理操作。
媒体处理机制分为以下几种方式：

方式	类别	说明
通过裸参数提交任务	持久化处理	提交媒体处理任务，并在提交任务时设置好 媒体处理任务所需参数 。
通过模板设置提交任务	持久化处理	于提交媒体处理任务前设置好处理模板，并在提交任务 选择所需模板 。
工作流处理	持久化处理	可将不同的任务类型组合成为工作流，将工作流设置好并开启工作流后，上传 COS 后的文件会 自动触发工作流处理流程 。
批量处理	持久化处理	针对 COS 上已经存在的数据进行批量处理，处理方式与提交任务类似，确认需要处理的数据范围与处理类型后 提交批量任务 。
同步任务	同步处理	在媒体对象的 URL 后加上 媒体处理参数 。

通过裸参数提交任务

您可以在每次提交媒体处理任务时，灵活地配置 媒体处理任务所需参数 。

> **说明**
> 通过裸参数提交任务时不需要设置模板。

请求语法

```
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>
```

请求示例：通过裸参数提交音视频转码任务

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host: test-1234567890.ci.ap-chongqing.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>
      <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>
    </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>
    <RemoveWatermark>
      <Dx>150</Dx>
      <Dy>150</Dy>
```

```
        <Width>75</Width>
        <Height>75</Height>
      </RemoveWatermark>
      <Subtitles>
        <Subtitle>
          <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
        </Subtitle>
        <Subtitle>
          <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
        </Subtitle>
      </Subtitles>
      <DigitalWatermark>
        <Type>Text</Type>
        <Message>123456789ab</Message>
        <Version>V1</Version>
        <IgnoreError>false</IgnoreError>
      </DigitalWatermark>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-1234567890</Bucket>
        <Object>output/out.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

返回

```
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>
    <AudioMixArray>
      <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.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>
        </AudioMixArray>
        <AudioMixArray>
          <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.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>
        </AudioMixArray>
      </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>
      <Subtitles>
```

```
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
            </Subtitle>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
            </Subtitle>
        </Subtitles>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
            <State>Running</State>
        </DigitalWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

通过模板设置提交任务

通过设置模板 能帮助您简化操作，设置模板后，您后期在业务中可以多次复用，帮助您提升处理效率。

请求语法

```
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>
```

请求示例：设置模板后提交音视频转码任务

```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.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</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <RemoveWatermark>
            <Dx>150</Dx>
            <Dy>150</Dy>
            <Width>75</Width>
            <Height>75</Height>
        </RemoveWatermark>
        <Subtitles>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
            </Subtitle>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
            </Subtitle>
        </Subtitles>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </DigitalWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

返回内容

```
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</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <RemoveWatermark>
        <Dx>150</Dx>
        <Dy>150</Dy>
        <Width>75</Width>
        <Height>75</Height>
      </RemoveWatermark>
      <Subtitles>
        <Subtitle>
          <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
        </Subtitle>
        <Subtitle>
          <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
```

```
            </Subtitle>
        </Subtitles>
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
            <State>Running</State>
        </DigitalWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

工作流处理

工作流可将不同任务节点进行组合以适配您的业务，您将文件上传 COS 时可 自动触发工作流处理，并将结果存入到 COS。

请求语法

```
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>
```

请求示例：对于输入的视频进行截图和转码处理。转码后，同时转封装和生成智能封面，最后做图片处理。

```
POST /workflow HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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_15816(
        <Segment_15816659605667>End</Segment_15816659605667>
        <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
        <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-12345678900</Bucket>
              <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
```

```
            <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</SpriteObject>
        </Output>
    </Operation>
</Snapshot_15816659605536>
<Transcode_15816659605538>
    <Type>Transcode</Type>
    <Operation>
        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>bcd/${RunId}/trans.{Ext}</Object>
        </Output>
    </Operation>
</Transcode_15816659605538>
<Segment_15816659605667>
    <Type>Segment</Type>
    <Operation>
        <Segment>
            <Format>mkv</Format>
            <Duration>20</Duration>
        </Segment>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>test-trans${Number}.{Ext}</Object>
        </Output>
    </Operation>
</Segment_15816659605667>
<SmartCover_15816659605539>
    <Type>SmartCover</Type>
    <Operation>
        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
        </Output>
    </Operation>
</SmartCover_15816659605539>
<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>
```

返回内容

```
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_15816(
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
```

```
        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicProce
        <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_158166596053 6>
      <Type>Snapshot</Type>
      <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-12345678 90</Bucket>
          <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
          <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</SpriteObject>
        </Output>
      </Operation>
    </Snapshot_158166596053 6>
    <Transcode_158166596053 8>
      <Type>Transcode</Type>
      <Operation>
        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-12345678 90</Bucket>
          <Object>bcd/${RunId}/trans.{Ext}</Object>
        </Output>
      </Operation>
    </Transcode_158166596053 8>
    <Segment_15816659605667>
      <Type>Segment</Type>
      <Operation>
```

```
            <Segment>
              <Format>mkv</Format>
              <Duration>20</Duration>
            </Segment>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</Bucket>
              <Object>test-trans${Number}.{Ext}</Object>
            </Output>
          </Operation>
        </Segment_1581665960567>
        <SmartCover_1581665960539>
          <Type>SmartCover</Type>
          <Operation>
            <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</Bucket>
              <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
            </Output>
          </Operation>
        </SmartCover_1581665960539>
        <PicProcess_1581665960568>
          <Type>PicProcess</Type>
          <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</Bucket>
              <Object>bcd/${RunId}/pic.{Ext}</Object>
            </Output>
          </Operation>
        </PicProcess_1581665960568>
      </Nodes>
    </Topology>
  </MediaWorkflow>
</Response>
```

批量处理

批量任务处理可对已上传 COS 的历史文件进行单任务类型处理或按照工作流中配置的操作自动化处理，您可指定处理的数据范围后 提交批量任务 。

请求语法

```
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
```

请求示例：提交一个音视频转码批量任务

```
POST /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
        <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-12345678900</Bucket>
            <Object>output/${InventoryTriggerJobId}/out.mp4</Object>
        </Output>
        <JobLevel>0</JobLevel>
    </Operation>
</Request>
```

返回

```
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>
```

同步任务

您可以通过在媒体文件的 URL 后加上 媒体处理参数，实现访问视频文件的时候 同步获取视频截帧 、封面等功能。

请求示例1：通过 GET 请求方式

```
https://<BucketName-APPID>.cos.<Region>.myqcloud.com/for-test.mp4?ci-process=sna
```

请求示例2：通过 URL 地址添加参数方式

```
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>
```

媒体处理接口列表

最近更新时间：2023-07-12 16:55:02

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（分段）	提交转封装任务
	音视频拼接	提交视频拼接任务
	视频水印	视频明水印
	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 。

版权所有：腾讯云计算（北京）有限责任公司

服务和队列配置
开通媒体处理服务

最近更新时间：2023-08-01 15:45:52

功能描述

本接口用于开通媒体处理服务并生成队列。

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>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <MediaBucket>
        <BucketId>examplebucket-12500000000</BucketId>
        <Name>examplebucket-12500000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</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	Container

Container 节点 MediaBucket 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.MediaBucket	存储桶 ID	String
Name	Response.MediaBucket	存储桶名称，同 BucketId	String
Region	Response.MediaBucket	所在的地域	String
CreateTime	Response.MediaBucket	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 /mediabucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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, 09 Aug 2022 16:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <MediaBucket>
        <BucketId>examplebucket-1250000000</BucketId>
        <Name>examplebucket-1250000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </MediaBucket>
</Response>
```

查询媒体处理服务

最近更新时间：2023-08-01 16:46:22

功能描述

本接口用于查询已经开通媒体处理处理功能的存储桶。

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	默认值	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，详情请参见 地域与域名	无	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	无	否
bucket	存储桶名称前缀，前缀搜索	无	否

Name			
pageNumber	第几页	1	否
pageSize	每页个数，大于 0且小于等于 100 的整数	10	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>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <MediaBucketList>
    <BucketId>examplebucket-1250000000</BucketId>
    <Name>examplebucket-1250000000</Name>
    <Region>ap-beijing</Region>
    <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
  </MediaBucketList>
  <MediaBucketList>
    <BucketId>examplebucket-1-1250000000</BucketId>
    <Name>examplebucket-1-1250000000</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2020-11-16T14:13:15+0800</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	父节点	描述	类型

字）			
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 数组

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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 2021 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>
    <MediaBucketList>
        <BucketId>examplebucket-1250000000</BucketId>
        <Name>examplebucket-1250000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </MediaBucketList>
    <MediaBucketList>
        <BucketId>examplebucket-1-1250000000</BucketId>
        <Name>examplebucket-1-1250000000</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
    </MediaBucketList>
</Response>
```

关闭媒体处理服务

最近更新时间：2023-08-01 15:45:53

功能描述

本接口用于关闭媒体处理服务并删除队列。

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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
  <BucketName>test-123456789</BucketName>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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
BucketName	Response	存储桶名称	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 /mediabucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
    <BucketName>test-123456789</BucketName>
```

```
</Response>
```

搜索媒体处理队列

最近更新时间：2023-08-01 16:46:22

功能描述

本接口用于查询媒体处理队列。

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，支持搜索多个，以“,”符号分割字符串，最多填10个。	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>
      <MqMode/>
      <MqName/>
      <MqRegion/>
    </NotifyConfig>
    <MaxSize>10000</MaxSize>
    <MaxConcurrent>10</MaxConcurrent>
    <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
    <CreateTime>2022-08-09T16:23:11+0800</CreateTime>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <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>
        <MqMode/>
        <MqName/>
        <MqRegion/>
      </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	回调类型，Url	String

版权所有：腾讯云计算（北京）有限责任公司

	.NotifyConfig		
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
MqRegion	Response.QueueList .NotifyConfig	TDMQ 所属园区	String
MqMode	Response.QueueList .NotifyConfig	TDMQ 使用模式	String
MqName	Response.QueueList .NotifyConfig	TDMQ 主题名称	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1：搜索指定队列

```
GET /queue?queueIds=p4a17eeea29334bf499b7e20e2fbfd99d,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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>
         <MqMode/>
         <MqName/>
         <MqRegion/>
      </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-
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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </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>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </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：2023-08-01 16:46:22

功能描述

本接口用于更新媒体处理的队列。

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
Request	无	保存结果的容器	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	默认值
Name	Request	队列名称，仅支持中文、英文、数字、_、-和*，长度不超过 128	String	是	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
NotifyConfig	Request	回调配置	Container	是	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
State	Request.NotifyConfig	回调开关 • Off • On	String	否	Off
Event	Request.NotifyConfig	回调事件 • TaskFinish：任务完成 • WorkflowFinish：工作流完成	String	当 State=On 时，必选	无

ResultFormat	Request.NotifyConfig	回调格式 • XML • JSON	String	否	XML
Type	Request.NotifyConfig	回调类型 • Url • TDMQ	String	当 State=On 时，必选	无
Url	Request.NotifyConfig	回调地址，不能为内网地址。	String	当 State=On，且 Type=Url 时，必选	无
MqMode	Request.NotifyConfig	TDMQ 使用模式 • Topic：主题订阅 • Queue: 队列服务	String	当 State=On，且 Type=TDMQ 时，必选	无
MqRegion	Request.NotifyConfig	TDMQ 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	当 State=On，且 Type=TDMQ 时，必选	无
MqName	Request.NotifyConfig	TDMQ 主题名称	String	当 State=On，且 Type=TDMQ 时，必选	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-file</Name>
      <State>Active</State>
      <NotifyConfig>
        <Url>http://callback.demo.com</Url>
        <Event>TaskFinish,WorkflowFinish</Event>
        <Type>Url</Type>
        <State>On</State>
        <ResultFormat>JSON</ResultFormat>
        <MqMode/>
        <MqName/>
        <MqRegion/>
      </NotifyConfig>
      <MaxSize>10000</MaxSize>
      <MaxConcurrent>10</MaxConcurrent>
      <Category>Transcoding</Category>
      <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-
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>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <Category>Transcoding</Category>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

获取媒体信息
获取媒体信息同步请求

最近更新时间：2023-07-18 19:12:42

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。
- 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

版权所有：腾讯云计算（北京）有限责任公司

Format	Response.MediaInfo	格式信息	Container
Stream	Response.MediaInfo	流信息	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 数组
Audio	Response.MediaInfo.Stream	音频信息	Container 数组
Subtitle	Response.MediaInfo.Stream	字幕信息	Container 数组

Container 节点 Format 的内容（某字段未获取到有效值时不返回）：

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	String
NumProgram	Response.MediaInfo.Format	节目的数量	String
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	String
Duration	Response.MediaInfo.Format	时长，单位为秒	String
Bitrate	Response.MediaInfo.Format	比特率，单位为 kbps	String
Size	Response.MediaInfo.Format	大小，单位为 Byte	String

Container 节点 Video 的内容（某字段未获取到有效值时不返回）：

节点名称（关键字）	父节点	描述	类型
Index	Response.MediaInfo.Stream.Video	该流的编号	String
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	String
Width	Response.MediaInfo.Stream.Video	视频宽，单位 px	String
HasBFrame	Response.MediaInfo.Stream.Video	是否有B帧。1表示有，0表示无	String
RefFrames	Response.MediaInfo.Stream.Video	视频编码的参考帧个数	String
Sar	Response.MediaInfo.Stream.Video	采样宽高比	String

版权所有：腾讯云计算（北京）有限责任公司

Dar	Response.MediaInfo.Stream.Video	显示宽高比	String
PixFormat	Response.MediaInfo.Stream.Video	像素格式	String
FieldOrder	Response.MediaInfo.Stream.Video	场的顺序	String
Level	Response.MediaInfo.Stream.Video	视频编码等级	String
Fps	Response.MediaInfo.Stream.Video	视频帧率	String
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	String
Duration	Response.MediaInfo.Stream.Video	视频时长，单位为秒	String
Bitrate	Response.MediaInfo.Stream.Video	比特率，单位为 kbps	String
NumFrames	Response.MediaInfo.Stream.Video	总帧数	String
Language	Response.MediaInfo.Stream.Video	语言	String

Container 节点 Audio 的内容（某字段未获取到有效值时不返回）：

节点名称（关键字）	父节点	描述	类型
Index	Response.MediaInfo.Stream.Audio	该流的编号	String
CodecName	Response.MediaInfo.Stream.Audio	编解码格式名字	String
CodecLongNa	Response.MediaInfo.Strea	编解码格式的详细名字	String

版权所有：腾讯云计算（北京）有限责任公司

me	m.Audio		
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	String
Channel	Response.MediaInfo.Stream.Audio	通道数量	String
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	String
Duration	Response.MediaInfo.Stream.Audio	音频时长，单位秒	String
Bitrate	Response.MediaInfo.Stream.Audio	比特率，单位 kbps	String
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	String
Language	Response.MediaInfo.Stream.Subtitle	语言，und 表示无查询结果	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

```
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*
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)</CodecLongName>
        <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 part 10</CodecLongN
            <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>
```


版权所有：腾讯云计算（北京）有限责任公司

获取媒体信息异步任务
提交任务

最近更新时间：2023-08-01 16:46:23

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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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 数组

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 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，详见 MediaInfo，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

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.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>
    <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

回调信息

最近更新时间：2023-08-10 18:13:42

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，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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</CodecL
            <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>
      <JobLevel>0</JobLevel>
    </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	同提交获取媒体信息任务接口中的 Response.JobsDetail	Container 数组

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。

Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String

Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T18:45:14+0800</CreationTime>
    <EndTime>2022-06-30T18:45:15+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>jb976c9e0f86111ec9452c1c27f56a659</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format>
          <Bitrate>834.736000</Bitrate>
          <Duration>13.654000</Duration>
          <FormatLongName>QuickTime / MOV</FormatLongName>
          <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
          <NumProgram>0</NumProgram>
          <NumStream>2</NumStream>
          <Size>1424687</Size>
          <StartTime>0.000000</StartTime>
        </Format>
        <Stream>
          <Audio>
            <Bitrate>104.047000</Bitrate>
            <Channel>2</Channel>
            <ChannelLayout>stereo</ChannelLayout>
            <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
            <CodecName>aac</CodecName>
            <CodecTag>0x6134706d</CodecTag>
            <CodecTagString>mp4a</CodecTagString>
            <CodecTimeBase>1/44100</CodecTimeBase>
            <Duration>13.653311</Duration>
            <Index>1</Index>
            <Language>und</Language>
```

```
            <SampleFmt>fltp</SampleFmt>
            <SampleRate>44100</SampleRate>
            <StartTime>0.000000</StartTime>
            <Timebase>1/44100</Timebase>
          </Audio>
          <Subtitle/>
          <Video>
            <AvgFps>25.000000</AvgFps>
            <Bitrate>763.774000</Bitrate>
            <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</CodecL
            <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>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <StartTime>2022-06-30T18:45:15+0800</StartTime>
    <State>Success</State>
    <Tag>MediaInfo</Tag>
  </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调，格式为 JSON


```
{
  "EventName": "TaskFinish",
  "JobsDetail": [{
    "Code": "Success",
    "CreationTime": "2022-06-30T18:45:14+0800",
    "EndTime": "2022-06-30T18:45:15+0800",
    "Input": {
      "BucketId": "test-123456789",
      "Object": "input/demo.mp4",
      "Region": "ap-chongqing"
    },
    "JobId": "jb976c9e0f86111ec9452c1c27f56a659",
    "Operation": {
      "MediaInfo": {
        "Format": {
          "Bitrate": "834.736000",
          "Duration": "13.654000",
          "FormatLongName": "QuickTime / MOV",
          "FormatName": "mov,mp4,m4a,3gp,3g2,mj2",
          "NumProgram": "0",
          "NumStream": "2",
          "Size": "1424687",
          "StartTime": "0.000000"
        },
        "Stream": {
          "Audio": [{
            "Bitrate": "104.047000",
            "Channel": "2",
            "ChannelLayout": "stereo",
            "CodecLongName": "AAC (Advanced Audio Coding)",
            "CodecName": "aac",
            "CodecTag": "0x6134706d",
            "CodecTagString": "mp4a",
            "CodecTimeBase": "1/44100",
            "Duration": "13.653311",
            "Index": "1",
            "Language": "und",
            "SampleFmt": "fltp",
            "SampleRate": "44100",
            "StartTime": "0.000000",
            "Timebase": "1/44100"
          }],
          "Video": [{
```

```
                "AvgFps": "25.000000",
                "Bitrate": "763.774000",
                "CodecLongName": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10",
                "CodecName": "h264",
                "CodecTag": "0x31637661",
                "CodecTagString": "avc1",
                "CodecTimeBase": "1/12800",
                "ColorPrimaries": "bt470bg",
                "ColorRange": "tv",
                "ColorTransfer": "smpte170m",
                "Duration": "12.960000",
                "Fps": "25.000000",
                "HasBFrame": "0",
                "Height": "960",
                "Index": "0",
                "Language": "und",
                "Level": "10",
                "NumFrames": "324",
                "PixFormat": "yuv420p",
                "Profile": "High",
                "RefFrames": "1",
                "Rotation": "0.000000",
                "StartTime": "0.000000",
                "Timebase": "1/12800",
                "Width": "544"
            }]
        }
      },
      "UserData": "This is my MediaInfo job.",
      "JobLevel": 0,
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T18:45:15+0800",
    "State": "Success",
    "Tag": "MediaInfo"
  }]
}
```

获取媒体截帧

最近更新时间：2023-07-20 18:09:01

功能描述

GetSnapshot 接口用于获取媒体文件某个时间的截图。

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=sna
```

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。
- 此接口为同步模式，异步模式请参考 截图任务 。

请求头

此接口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	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

说明：
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。

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*
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>
```

读取私有 M3U8
获取私有 m3u8

最近更新时间：2023-07-20 18:09:02

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 m3u8 的内容，其中的 ts 为可下载的链接。

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*
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****

#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:11
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-KEY:METHOD=AES-128,URI="http://website.com/aes.key?Ciphertext=xxxxx&KMS
#EXTINF:11.333333,
test-00000.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:9.416667,
test-00001.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:9.375000,
```

```
test-00002.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:11.291667,
test-00003.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:3.500000,
test-00004.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXT-X-ENDLIST
```

版权所有：腾讯云计算（北京）有限责任公司

更新 HLS 解密 Token

最近更新时间：2023-07-20 18:09:02

功能描述

ModifyM3U8Token 接口用于在加密 M3U8 的 key uri 中增加 Token。

请求

请求示例

```
GET /modifym3u8token?object=&token= HTTP/1.1
Host: <BucketName-APPID>.cos.<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

参数的具体内容如下：

节点名称（关键字）	描述	类型	是否必选
object	M3U8 文件的对象名	String	是
token	需要嵌入 M3U8 key uri中的 token，长度小于2048个UTF-8字符。	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

该响应体为添加 token 的 m3u8 文件内容。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:11
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-KEY:METHOD=AES-128,URI="http://website.com/aes.key?Ciphertext=xxxxx&KMS
#EXTINF:11.333333,
test-00000.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:9.416667,
test-00001.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:9.375000,
test-00002.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:11.291667,
test-00003.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:3.500000,
test-00004.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXT-X-ENDLIST
```

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 /modifym3u8token?expires=3600&object=test.m3u8&token=abc HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-type: application/xml
```

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-ci-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZm****

#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:11
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-KEY:METHOD=AES-128,URI="http://website.com/aes.key?Ciphertext=xxxxx&KMS
#EXTINF:11.333333,
test-00000.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:9.416667,
test-00001.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:9.375000,
test-00002.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:11.291667,
test-00003.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXTINF:3.500000,
test-00004.ts?q-sign-algorithm=sha1&q-ak=&q-sign-time=&q-key-time=&q-header-list=
#EXT-X-ENDLIST
```

版权所有：腾讯云计算（北京）有限责任公司

视频 DNA
提交视频 DNA 任务

最近更新时间：2023-08-09 17:33:03

功能描述

提交一个视频 DNA 任务，视频 DNA 任务可实现视频入库、视频查重、视频出库操作。

> **注意：**
> - 视频 DNA 计费标准与视频标签一致，根据视频处理时长计费，例：入库时长为10分钟的视频，查重时长为5分钟的视频，费用为：10*0.05元/分钟+5*0.05元/分钟=0.75元，详情见 按量计费（后付费）。
> - 使用视频DNA需提前创建视频 DNA 库，点击 提交工单 创建 DNA 库。

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>DNA</Tag>
  <Input>
    <Object>input/demo.mp4</Object>
  </Input>
  <Operation>
    <DnaConfig>
      <RuleType>GetFingerPrint</RuleType>
      <DnaDbId>Db0fafe624fec211edbdc852540083a8ba</DnaDbId>
    </DnaConfig>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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：DNA	String	是
Input	Request	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否

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	输入文件路径，目前输入视频只支持 mp4 格式，视频长度小于1800s	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
DnaConfig	Request.Operation	视频 DNA 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	Request.Operation	透传用户信息	String	否

Container 类型 DnaConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
RuleType	Request.Operation.DnaConfig	DNA 任务类型 • GetFingerPrint：视频存入 DNA 库获取视频指纹，并进行查重 • FindSimilarVideo：视频查重，不会存入 DNA 库 • DeleteFingerPrint：从 DNA 库中删除视频	String	是
DnaDbId	Request.Operation.DnaConfig	DNA 库 ID	String	是
VideoId	Request.Operation.DnaConfig	视频唯一 ID，当 RuleType 为 DeleteVideoPrint 时生效且必选	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>SUCCESS</Message>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j8d121820f5e411ec926ef19d53ba9c6f</JobId>
    <State>Submitted</State>
    <CreationTime>2022-06-27T15:23:10+0800</CreationTime>
    <StartTime>-</StartTime>
    <EndTime>-</EndTime>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <Tag>DNA</Tag>
    <Operation>
      <DNAResult>
        <Detection>
          <Audio>
            <Similar>100</Similar>
          </Audio>
          <Duration>10</Duration>
          <MatchDetail>
            <MatchEndTime>9</MatchEndTime>
            <MatchStartTime>0</MatchStartTime>
            <SrcEndTime>9</SrcEndTime>
            <SrcStartTime>0</SrcStartTime>
          </MatchDetail>
          <Similar>100</Similar>
          <SimilarDuration>10</SimilarDuration>
          <VideoId>dv2b162b7d005011eeb67352540083a8ba</VideoId>
        </Detection>
        <Detection>
          <Audio>
```

```
            <Similar>100</Similar>
          </Audio>
          <Duration>10</Duration>
          <MatchDetail>
            <MatchEndTime>9</MatchEndTime>
            <MatchStartTime>0</MatchStartTime>
            <SrcEndTime>9</SrcEndTime>
            <SrcStartTime>0</SrcStartTime>
          </MatchDetail>
          <Similar>100</Similar>
          <SimilarDuration>10</SimilarDuration>
          <VideoId>dv9fb4b5d3004d11eeb67352540083a8ba</VideoId>
        </Detection>
        <Duration>10</Duration>
        <VideoId>dvef8cd08e006511eeb67352540083a8ba</VideoId>
      </DNAResult>
      <DnaConfig>
        <DnaDbId>Db0fafe624fec211edbdc852540083a8ba</DnaDbId>
        <RuleType>FindSimilarVideo</RuleType>
      </DnaConfig>
      <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 数组

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	创建任务的 ID	String
Tag	Response.JobsDetail	创建任务的 Tag：DNA	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
DnaConfig	Response.JobsDetail.Operation	同请求中的 Request.Operation.DnaConfig	Container
DnaResult	Response.JobsDetail.Operation	DNA 任务结果，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 DnaResult 的内容：

节点名称（关键字）	父节点	描述	类型
VideoId	Response.JobsDetail.Operation.DnaResult	输入视频的唯一标识 ID	String
Duration	Response.JobsDetail.Operation.DnaResult	输入视频的时长，单位秒	Integer
Detection	Response.JobsDetail.Operation.DnaResult	相似视频详情	Container 数组

Container 节点 Detection 的内容：

节点名称（关键字）	父节点	描述	类型
VideoId	Response.JobsDetail.Operation.DnaResult.Detection	匹配视频的 VideoId	String
Similar	Response.JobsDetail.Operation.DnaResult.Detection	相似度	Integer
SimilarDuration	Response.JobsDetail.Operation.DnaResult.Detection	相似片段时长	Integer

版权所有：腾讯云计算（北京）有限责任公司

Duration	Response.JobsDetail.Operation.DnaResult.Detection	视频时长	Integer
MatchDetail	Response.JobsDetail.Operation.DnaResult.Detection	匹配详情	Container 数组
Audio	Response.JobsDetail.Operation.DnaResult.Detection	音频相似详情	Container

Container 节点 MatchDetail 的内容：

节点名称（关键字）	父节点	描述	类型
MatchStartTime	Response.JobsDetail.Operation.DnaResult.Detection.MatchDetail	相似视频匹配的开始时间，单位秒	Integer
MatchEndTime	Response.JobsDetail.Operation.DnaResult.Detection.MatchDetail	相似视频匹配的结束时间，单位秒	Integer
SrcStartTime	Response.JobsDetail.Operation.DnaResult.Detection.MatchDetail	源视频匹配的开始时间，单位秒	Integer
SrcEndTime	Response.JobsDetail.Operation.DnaResult.Detection.MatchDetail	源视频匹配的结束时间，单位秒	Integer

Container 节点 Audio 的内容：

节点名称（关键字）	父节点	描述	类型
Similar	Response.JobsDetail.Operation.DnaResult.Detection.Audio	音频相似度	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

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

```
Content-Type: application/xml

<Request>
    <Tag>DNA</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <DnaConfig>
            <RuleType>GetFingerPrint</RuleType>
            <DnaDbId>Db0fafe624fec211edbdc852540083a8ba</DnaDbId>
        </DnaConfig>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>DNA</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
```

```
    </Input>
    <Operation>
      <DNAResult>
        <Detection>
          <Audio>
            <Similar>100</Similar>
          </Audio>
          <Duration>10</Duration>
          <MatchDetail>
            <MatchEndTime>9</MatchEndTime>
            <MatchStartTime>0</MatchStartTime>
            <SrcEndTime>9</SrcEndTime>
            <SrcStartTime>0</SrcStartTime>
          </MatchDetail>
          <Similar>100</Similar>
          <SimilarDuration>10</SimilarDuration>
          <VideoId>dv2b162b7d005011eeb67352540083a8ba</VideoId>
        </Detection>
        <Detection>
          <Audio>
            <Similar>100</Similar>
          </Audio>
          <Duration>10</Duration>
          <MatchDetail>
            <MatchEndTime>9</MatchEndTime>
            <MatchStartTime>0</MatchStartTime>
            <SrcEndTime>9</SrcEndTime>
            <SrcStartTime>0</SrcStartTime>
          </MatchDetail>
          <Similar>100</Similar>
          <SimilarDuration>10</SimilarDuration>
          <VideoId>dv9fb4b5d3004d11eeb67352540083a8ba</VideoId>
        </Detection>
        <Duration>10</Duration>
        <VideoId>dvef8cd08e006511eeb67352540083a8ba</VideoId>
      </DNAResult>
      <DnaConfig>
        <DnaDbId>Db0fafe624fec211edbdc852540083a8ba</DnaDbId>
        <RuleType>GetFingerPrint</RuleType>
      </DnaConfig>
      <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

获取 DNA 库中文件列表

最近更新时间：2023-08-09 17:33:04

功能描述

查询 DNA 库内文件列表。

请求

请求示例

```
GET /dnadb_files 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。

请求参数

参数的具体内容如下：

参数名称	描述	是否必选
object	文件名	否
dnaDbId	DNA 库 ID	否
pageNumber	第几页，默认值:1	否
pageSize	每页个数，默认值:10，最大值 100	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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TotalCount>2</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <DNADbFiles>
        <BucketId>test-1234567890</BucketId>
        <Region>ap-chongqing</Region>
        <DNADbId>xxx1</DNADbId>
        <VideoId>xxx</VideoId>
        <Object>input/test1.mp4</Object>
        <ETag>&amp;quot;3a64323ae8ed5633c02fc96def134b92-23&amp;quot;</ETag>
        <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
        <UpdateTime>2023-06-31T16:15:20+0800</UpdateTime>
    </DNADbFiles>
    <DNADbFiles>
        <BucketId>test-1234567890</BucketId>
        <Region>ap-chongqing</Region>
        <DNADbId>xxx1</DNADbId>
        <VideoId>xxx</VideoId>
        <Object>input/test2.mp4</Object>
        <ETag>&amp;quot;234f8b0b58874328039ac66352820cc8&amp;quot;</ETag>
        <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
        <UpdateTime>2023-06-31T16:15:20+0800</UpdateTime>
    </DNADbFiles>
</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	文件总数	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
DNADbFiles	Response	文件的详细信息	Container 数组

Container 节点 DNADbConfig 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.DNADbConfig	存储桶 ID	String
Region	Response.DNADbConfig	存储桶所在园区	String
DNADbId	Response.DNADbConfig	DNA 库 ID	String
VideoId	Response.DNADbConfig	文件唯一标识符	String
Object	Response.DNADbConfig	文件名	String
ETag	Response.DNADbConfig	文件 ETag	String
UpdateTime	Response.DNADbConfig	文件更新时间	String
CreateTime	Response.DNADbConfig	文件创建时间	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

```
GET /dnadb_filesHTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
    <TotalCount>2</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <DNADbFiles>
        <BucketId>test-1234567890</BucketId>
        <Region>ap-chongqing</Region>
        <DNADbId>xxx1</DNADbId>
        <VideoId>xxx</VideoId>
        <Object>input/test1.mp4</Object>
        <ETag>&quot;3a64323ae8ed5633c02fc96def134b92-23&quot;</ETag>
        <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
        <UpdateTime>2023-06-31T16:15:20+0800</UpdateTime>
    </DNADbFiles>
    <DNADbFiles>
        <BucketId>test-1234567890</BucketId>
        <Region>ap-chongqing</Region>
        <DNADbId>xxx1</DNADbId>
        <VideoId>xxx</VideoId>
        <Object>input/test2.mp4</Object>
        <ETag>&quot;234f8b0b58874328039ac66352820cc8&quot;</ETag>
        <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
        <UpdateTime>2023-06-31T16:15:20+0800</UpdateTime>
    </DNADbFiles>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

查询 DNA 库列表

最近更新时间：2023-08-09 17:33:04

功能描述

获取 DNA 库列表

请求

请求示例

```
GET /dnadb 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。

请求参数

参数的具体内容如下：

参数名称	描述	是否必选
ids	DNA库 ID，支持以 , 符号分割字符串，最多填 20 个	否
pageNumber	第几页，默认值:1	否
pageSize	每页个数，默认值:10, 最大值 100	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>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
  <TotalCount>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <DNADbConfig>
    <BucketId>test-1234567890</BucketId>
    <Region>ap-chongqing</Region>
    <DNADbId>xxx1</DNADbId>
    <DNADbName>测试1</DNADbName>
    <Capacity>10000</Capacity>
    <Description>这是测试库1</Description>
    <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
    <UpdateTime>2023-06-31T16:15:20+0800</UpdateTime>
  </DNADbConfig>
  <DNADbConfig>
    <BucketId>test-1234567890</BucketId>
    <Region>ap-chongqing</Region>
    <DNADbId>xxx2</DNADbId>
    <DNADbName>测试2</DNADbName>
    <Capacity>10000</Capacity>
    <Description>这是测试库2</Description>
    <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
    <UpdateTime>2023-06-25T17:21:20+0800</UpdateTime>
  </DNADbConfig>
  <NonExistIDs>A</NonExistIDs>
  <NonExistIDs>B</NonExistIDs>
  <NonExistIDs>C</NonExistIDs>
</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

版权所有：腾讯云计算（北京）有限责任公司

RequestId	Response	请求的唯一 ID	String
TotalCount	Response	库总数	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
DNADbConfig	Response	库的详细信息	Container 数组
NonExistIDs	Response	查询的 ID 中不存在的库，所有 ID 都存在时不返回	String 数组

Container 节点 DNADbConfig 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.DNADbConfig	存储桶 ID	String
Region	Response.DNADbConfig	存储桶所在园区	String
DNADbId	Response.DNADbConfig	DNA 库 ID	String
DNADbName	Response.DNADbConfig	DNA 库名称	String
Capacity	Response.DNADbConfig	DNA 库容量	Int
Description	Response.DNADbConfig	DNA 库描述	String
UpdateTime	Response.DNADbConfig	DNA 更新时间	String
CreateTime	Response.DNADbConfig	DNA 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

```
GET /dnadb?ids=xxx1,xxx2,A,B,C HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
```

版权所有：腾讯云计算（北京）有限责任公司

```
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>
    <RequestId>NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=</RequestId>
    <TotalCount>2</TotalCount>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <DNADbConfig>
        <BucketId>test-1234567890</BucketId>
        <Region>ap-chongqing</Region>
        <DNADbId>xxx1</DNADbId>
        <DNADbName>测试1</DNADbName>
        <Capacity>10000</Capacity>
        <Description>这是测试库1</Description>
        <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
        <UpdateTime>2023-06-31T16:15:20+0800</UpdateTime>
    </DNADbConfig>
    <DNADbConfig>
        <BucketId>test-1234567890</BucketId>
        <Region>ap-chongqing</Region>
        <DNADbId>xxx2</DNADbId>
        <DNADbName>测试2</DNADbName>
        <Capacity>10000</Capacity>
        <Description>这是测试库2</Description>
        <CreateTime>2023-06-05T11:35:24+0800</CreateTime>
        <UpdateTime>2023-06-25T17:21:20+0800</UpdateTime>
    </DNADbConfig>
    <NonExistIDs>A</NonExistIDs>
    <NonExistIDs>B</NonExistIDs>
    <NonExistIDs>C</NonExistIDs>
</Response>
```

音视频转码
提交任务

最近更新时间：2023-08-01 16:46:23

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</WatermarkTemp
```

```
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <RemoveWatermark>
            <Dx>150</Dx>
            <Dy>150</Dy>
            <Width>75</Width>
            <Height>75</Height>
        </RemoveWatermark>
        <Subtitles>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
            </Subtitle>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
            </Subtitle>
        </Subtitles>
        <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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
QueueType	Request	任务所在的队列类型，限制为 SpeedTranscoding, 表示为开启倍速转码	String	否
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
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	文件路径	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	转码模板 ID	String	否
Transcode	Request.Operation	转码模板参数	Container	否

WatermarkTemplateId	Request.Operation	水印模板 ID，可以传多个水印模板 ID，最多传3个	String数组	否
Watermark	Request.Operation	水印模板参数，同创建水印模板接口中的 Request.Watermark ，最多传3个	Container 数组	否
RemoveWatermark	Request.Operation	去除水印参数，H265、AV1编码暂不支持该参数	Container	否
Subtitles	Request.Operation	字幕参数，H265、AV1编码和非mkv封装暂不支持该参数	Container	否
DigitalWatermark	Request.Operation	数字水印参数，详情见 DigitalWatermark	Container	否
Output	Request.Operation	结果输出配置	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

- 转码参数必须设置，可以通过 TemplateId 或 Transcode 设置， TemplateId 优先级更高。
- 水印参数可选，可以通过 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	是否必选
TimeInterval	Request.Operation.Transcode	同创建转码模板接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板接口中的 Request.Video	Container	否
Audio	Request.Operation.Transcode	同创建转码模板接口中的 Request.Audio	Container	否

版权所有：腾讯云计算（北京）有限责任公司

TransConfig	Request.Operation.Transcode	同创建转码模板接口中的 Request.TransConfig	Container	否
AudioMix	Request.Operation.Transcode	混音参数，详情见 AudioMix	Container	否
AudioMixArray	Request.Operation.Transcode	混音参数数组，最多同时传2个	Container 数组	否

说明

- 不传 Video，相当于删除视频信息。不传 Audio，相当于删除音频信息。Video 和 Audio 不可以同时不传。
- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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	是否必选
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 Subtitles 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Subtitle	Request.Operation.Subtitles	字幕参数	Container数组	是

Container 类型 Subtitle 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Url	Request.Operation.Subtitles.Subtitle	同 bucket 的字幕地址，需要 encode	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

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <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</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
        <RemoveWatermark>
          <Dx>150</Dx>
          <Dy>150</Dy>
          <Width>75</Width>
          <Height>75</Height>
        </RemoveWatermark>
        <Subtitles>
          <Subtitle>
            <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
          </Subtitle>
          <Subtitle>
            <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
          </Subtitle>
        </Subtitles>
        <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.${ext}</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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	String
Progress	Response.JobsDetail	任务进度百分比，范围为[0, 100]	String
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.	任务的开始时间	Strin

	JobsDetail		g
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
Transcode	Response.JobsDetail.Operation	同请求中的 Request.Operation.Transcode	Container
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组
WatermarkTemplateId	Response.JobsDetail.Operation	同请求中的 Request.Operation.WatermarkTemplateId	String 数组
RemoveWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.RemoveWatermark	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，详见 MediaInfo，任务未完成时不返回	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，详见 MediaResult，任务未完成时不返回	Container
DigitalWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.DigitalWatermark	Container
UserData	Response.JobsDetail.Operation	同请求中的 Request.Operation.UserData	String
JobLevel	Response.JobsDetail.Operation	同请求中的 Request.Operation.JobLevel	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-
Host: test-1234567890.ci.ap-chongqing.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</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <RemoveWatermark>
            <Dx>150</Dx>
```

```
        <Dy>150</Dy>
        <Width>75</Width>
        <Height>75</Height>
      </RemoveWatermark>
      <Subtitles>
        <Subtitle>
          <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
        </Subtitle>
        <Subtitle>
          <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
        </Subtitle>
      </Subtitles>
      <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.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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</WatermarkTe
  <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
  <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
  <RemoveWatermark>
    <Dx>150</Dx>
    <Dy>150</Dy>
    <Width>75</Width>
    <Height>75</Height>
  </RemoveWatermark>
  <Subtitles>
    <Subtitle>
      <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
    </Subtitle>
    <Subtitle>
      <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
    </Subtitle>
  </Subtitles>
  <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.${ext}</Object>
  </Output>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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-s
Host: test-1234567890.ci.ap-chongqing.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>
        <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>
    </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>
    <RemoveWatermark>
      <Dx>150</Dx>
      <Dy>150</Dy>
      <Width>75</Width>
      <Height>75</Height>
    </RemoveWatermark>
    <Subtitles>
      <Subtitle>
        <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
      </Subtitle>
      <Subtitle>
        <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
      </Subtitle>
    </Subtitles>
    <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.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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>
      <AudioMixArray>
```

```
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.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>
         </AudioMixArray>
         <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.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>
         </AudioMixArray>
      </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>
        <Subtitles>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test1.srt</Url>
            </Subtitle>
            <Subtitle>
                <Url>https://test-xxx.cos.ap-chongqing.myqcloud.com/test2.srt</Url>
            </Subtitle>
        </Subtitles>
        <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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-11 18:10:32

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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)</CodecL
            <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.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
        <TemplateName>trans_993874</TemplateName>
        <UserData>This is my Transcode 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	同提交转码任务接口中的 Response.JobsDetail	Container 数组

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)</CodecL
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </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.${ext}</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
      <TemplateName>trans_993874</TemplateName>
      <UserData>This is my Transcode 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)</CodecL
          <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.${ext}</Object>
```

```
            <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
        <TemplateName>trans_993874</TemplateName>
        <UserData>This is my Transcode 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 Coding)",
                  "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.${ext}",
        "Region": "ap-chongqing"
    },
    "TemplateId": "t1995d523e42df4c5e858f244b4174360c",
    "TemplateName": "trans_993874",
    "UserData": "This is my Transcode job.",
    "JobLevel": 0
},
"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"
      }
    }]
}
```

创建模板

最近更新时间：2023-07-24 14:18:43

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</Audios
    <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</Audios
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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：Transcode	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64	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
Audio	Request	音频信息	Container	否
TransConfig	Request	转码配置	Container	否
AudioMix	Request	混音参数，详情见 AudioMix	Container	否
AudioMixArray	Req	混音参数，最多同时传2个	Contain	否

	uest		er 数组	

说明：

- 不传 Video，相当于删除视频信息。不传 Audio，相当于删除音频信息。Video 和 Audio 不可以同时不传。
- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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
ts/flv/avi/mov	aac、mp3	H.264
dash	aac	H.264
WebM	Vorbis、opus	VP8、VP9、AV1
aac	aac	不支持
mp3	mp3	不支持

flac	flac	不支持
amr	amr	不支持
adts	aac	不支持
m4a	aac	不支持
wav	pcm_s16le	不支持

Container 类型 Video 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
Codec	Request.Video	编解码格式 • H.264 • H.265 • VP8 • VP9 • AV1	String	• H.264 • 当 Format 为 WebM 时，为 VP8	否
Width	Request.Video	宽 • 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	String	视频原始宽度	否
Height	Request.Video	高 • 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	String	视频原始高度	否
Fps	Request.Video	帧率 • 值范围：(0, 60] • 单位：fps	String	无	否
Remove	Reques	是否删除视频流，取值 true	Str	false	否

版权所有：腾讯云计算（北京）有限责任公司

	t.Video	或 false。	ing		
Profile	Request.Video	编码级别 • 仅H.264支持此参数 • 支持 baseline、main、high、auto • 当 Pixfmt 为 auto 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	String	high	否
Bitrate	Request.Video	视频输出文件的码率 • 值范围：[10, 50000]，单位：Kbps • 设置为 auto 表示自适应码率	String	无	否
Crf	Request.Video	码率-质量控制因子 • 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	String	无	否
Gop	Request.Video	关键帧间最大帧数，值范围：[1, 100000]	String	无	否
Preset	Request.Video	视频算法器预置 • 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 universal、medium • H.265 和 VP9 不支持该参数	String	• 当 Codec 为 H.264 时，为 medium • 当 Codec 为 VP8 时，为 good	否
Bufsize	Request.Video	缓冲区大小 • 值范围：[1000, 128000]	String	无	否

版权所有：腾讯云计算（北京）有限责任公司

		• 单位：Kb • Codec 为 VP8/VP9时不支持此参数			
Maxrate	Request.Video	视频码率峰值 • 值范围：[10, 50000] • 单位：Kbps • Codec 为 VP8/VP9 时不支持此参数	String	无	否
Pixfmt	Request.Video	视频颜色格式 • 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	String	无	否
LongShortMode	Request.Video	长短边自适应，取值 true或 false。Codec 为 VP8/VP9/AV1 时不支持此参数。	String	false	否
Rotate	Request.Video	旋转角度 • 值范围：[0, 360) • 单位：度	String	无	否
Roi	Request.Video	Roi 强度 • 取值为 none、low、medium、high • Codec 为 VP8/VP9 时不支持此参数	String	none	否
Crop	Request.Video	自由裁剪 • 自定义裁切: width:height:left:top。示例:1280:800:0:140 • width和height的值需要大于0，left和top的值需要大于等于0	String	无	否

		• Codec 为 H.265/AV1 时不支持此参数。 • 开启自适应编码时，不支持此参数。 • 开启roi时，不支持此参数。			
Interlaced	Request.Video	开启隔行扫描 • false/true • Codec 为 H.265/AV1 时不支持此参数。 • 开启自适应码率时，不支持此参数。 • 开启roi时，不支持此参数。	String	false	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	默认值	是否必选
Start	Request.TimeInterval	开始时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	0	否
Duration	Request.TimeInterval	持续时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	视频原始时长	否

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	默认值	是否必选
Codec	Request.Audio	编解码格式，取值 aac、mp3、flac、amr、Vorbis、opus、pcm_s16le	String	• aac • 当 format 为 WebM	否

				时，为 Vorbis • 当 format 为 wav 时，为 pcm_s16le	
Samplerate	Request.Audio	采样率 • 单位：Hz • 可选 8000、11025、12000、16000、22050、24000、32000、44100、48000、88200、96000 • 不同的封装，mp3 支持不同的采样率，如下表所示 • 当 Codec 设置为 amr 时，只支持8000 • 当 Codec 设置为 opus 时，支持8000，16000，24000，48000	String	44100 • 当 Codec 为 opus 时，默认值为 48000	否
Bitrate	Request.Audio	原始音频码率 • 单位：Kbps • 值范围：[8，1000]	String	无	否
Channels	Request.Audio	声道数 • 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	String	无	否
Remove	Request.Audio	是否删除源音频流，取值 true、false	String	false	否

SampleFormat	Request.Audio	采样位宽 • 当 Codec 设置为 aac，支持 fltp • 当 Codec 设置为 mp3，支持 fltp、s16p、s32p • 当 Codec 设置为 flac，支持 s16、s32、s16p、s32p • 当 Codec 设置为 amr，支持 s16、s16p • 当 Codec 设置为 opus，支持 s16 • 当 Codec 设置为 pcm_s16le，支持s16 • 当 Codec 设置为 Vorbis，支持 fltp • 当 Video.Codec 为 H.265 时，此参数无效	String	无	否

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	

版权所有：腾讯云计算（北京）有限责任公司

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	48000	88200	96000
flv	N	N	N	N	Y	N	N	Y	N	N	N
mp4	N	N	N	Y	Y	Y	Y	Y	Y	N	N
hls/ts/mp3/mkv/avi/mov	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N

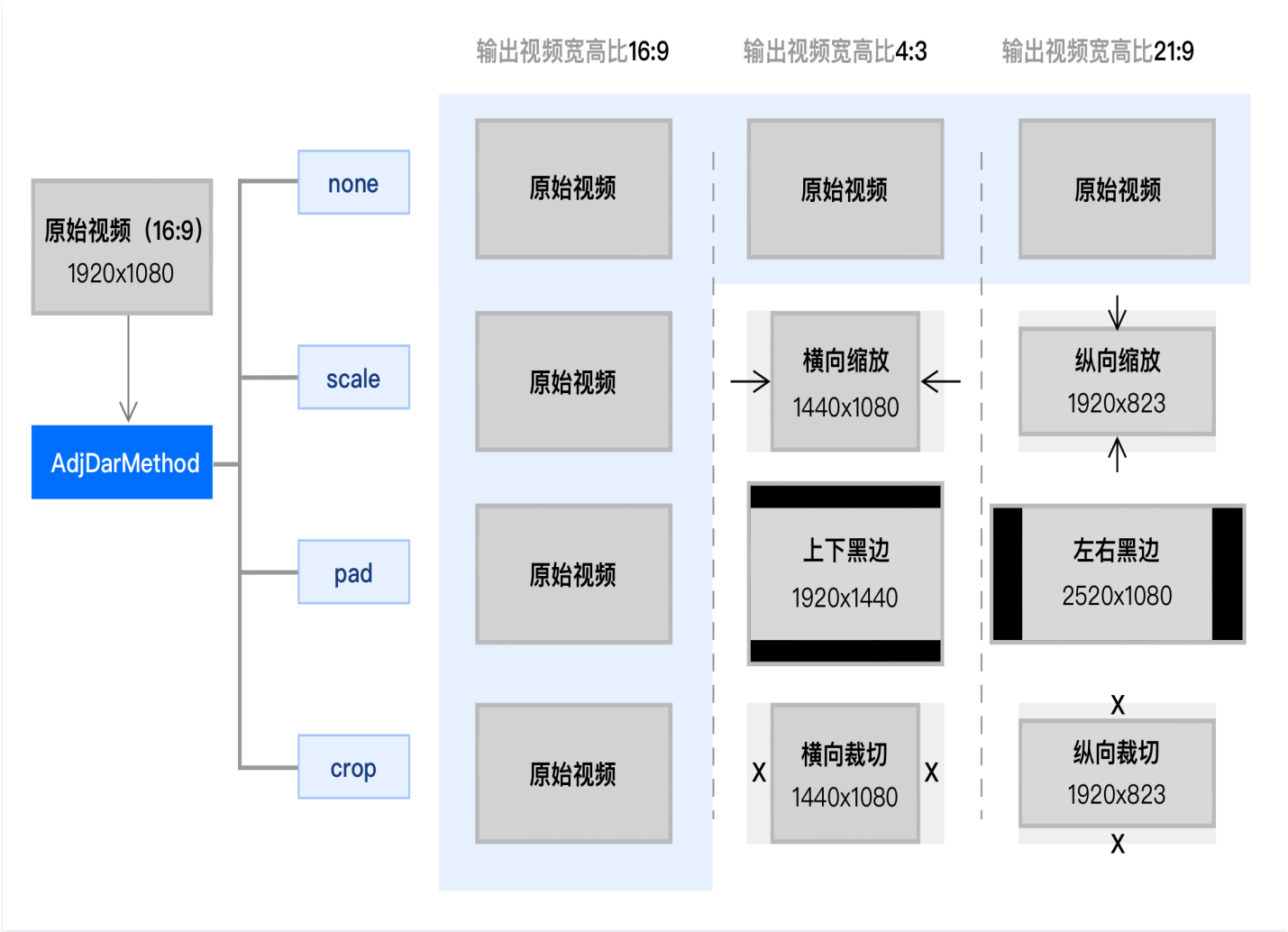
Container 类型 TransConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
AdjDarMethod	Request.TransConfig	分辨率调整方式 • 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	String	none	否
IsCheckReso	Request.TransConfig	是否检查分辨率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
ResoAdjMethod	Request.TransConfig	分辨率调整方式 • 当 IsCheckReso 为 true 时生效，取值0、1； • 0 表示使用原视频分辨率； • 1 表示返回转码失败	String	0	否
IsCheckVideoBitrate	Request.TransConfig	是否检查视频码率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
VideoBitrateAdjMethod	Request.TransConfig	视频码率调整方式 • IsCheckVideoBitrate 为 true	String	0	否

		时生效，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			
IsCheckAudioBitrate	Request.TransConfig	是否检查音频码率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
AudioBitrateAdjMethod	Request.TransConfig	音频码率调整方式 • IsCheckAudioBitrate 为 true 时生效，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	String	0	否
IsCheckVideoFps	Request.TransConfig	是否检查视频帧率，取值 true、false 当为 false 时，按照配置参数转码	String	false	否
VideoFpsAdjMethod	Request.TransConfig	视频帧率调整方式 • IsCheckVideoFps 为 true 时生效，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	String	0	否
DeleteMetadata	Request.TransConfig	是否删除文件中的 MetaData 信息，取值 true、false	String	false	否
IsHdr2Sdr	Request.TransConfig	是否开启 HDR 转 SDR，取值 true、false	String	false	否
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	无	否
HlsEncrypt	Request.TransConfig	hls 加密配置	Container	无	否

DashEncrypt	Request.TransConfig	dash 加密配置	Container	无	否

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	默认值	是否必选
IsHlsEncrypt	Request.TransConfig.HlsEncrypt	是否开启 HLS 加密，取值 true、false • 当 Container.Format 为 hls 时支持加密	String	false	否
UriKey	Request.TransCon	HLS 加密的 key	St	无	否

	fig.HlsEncrypt	• 当 IsHlsEncrypt 为 true 时，该参数才有意义	ri n g		

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	默认值	是否必选
IsEncrypt	Request.TransConfig.DashEncrypt	是否开启 DASH 加密，取值 true、false • 当 Container.Format 为 dash 时支持加密	String	false	否
UriKey	Request.TransConfig.DashEncrypt	DASH 加密的 key • 当 IsEncrypt 为 true 时，该参数才有意义	String	无	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>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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</Aud
        <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</Aud
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
        <EffectConfig>
            <EnableStartFadein>true</EnableStartFadein>
            <StartFadeinTime>3</StartFadeinTime>
            <EnableEndFadeout>false</EnableEndFadeout>
            <EndFadeoutTime>0</EndFadeoutTime>
            <EnableBgmFade>true</EnableBgmFade>
            <BgmFadeTime>1.7</BgmFadeTime>
        </EffectConfig>
    </AudioMixArray>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
```

```
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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.Templat	详细的模板参数	Containe

	e		r

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
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-
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>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Audios
        <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</Audios
        <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>
    <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
    <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</Aud
        <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</Aud
        <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>
```

创建模板 转码pro（广电专业格式）

最近更新时间：2023-08-01 16:46:23

功能描述

创建音视频转码 pro 模板。

> **说明：**
> 此功能为广电专业格式转码，可处理 Apple Prores、Sony XAVC 等格式。

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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
Tag	Request	模板类型：TranscodePro	String	是
Name	Request	模板名称 仅支持中文、英文、数字、_、-和*，长度不超过 64	String	是
Container	Request	容器格式	Container	是
Video	Request	视频信息	Container	是

TimeInterval	Request	时间区间	Container	否
Audio	Request	音频信息，不传 Audio，相当于删除音频信息	Container	否
TransConfig	Request	转码配置	Container	否

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：有效值为 mxf、mov、mkv	String	是

设定 container，音频视频支持的编解码格式如下表：

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
Codec	Request.Video	编解码格式，支持 xavc、apple_prores	String	是	无
Profile	Request.Video	视频算法器预置，取值范围见下表	String	是	无
Width	Request.Video	宽，取值范围见下表	String	是	无
Height	Request.Video	高，取值范围见下表	String	是	无
Interlaced	Request.Video	场模式，取值范围见下表	String	是	无
Fps	Request.Video	帧率，取值范围见下表	String	是	无

版权所有：腾讯云计算（北京）有限责任公司

Bitrate	Request.Video	视频输出文件的码率，取值范围见下表	String	否	无
Rotate	Request.Video	旋转角度 • 值范围：[0, 360) • 单位：度	String	否	无

注意：

1、'-' 表示留空。

2、码率可选值与帧率可选值序号对应。

当 Video.Codec 为 xavc 时：

Profile	Width	Height	Interlaced	Fps	Bitrate(单位:kbps)
XAVC-HD_intra_420_10bit_class50	1440	1080	true	1、25 2、30000/1001	-
XAVC-HD_intra_420_10bit_class50	1440	1080	false	1、25 2、24000/1001 3、30000/1001	-
XAVC-HD_intra_422_10bit_class100	1280	720	false	1、50 2、60000/1001	-
XAVC-HD_intra_422_10bit_class100	1920	1080	true	1、25 2、30000/1001	-
XAVC-HD_intra_422_10bit_class100	1920	1080	false	1、25 2、50 3、24000/1001	-

				4、30000/1001 5、60000/1001	
XAVC-HD_intra_422_10bit_class200	1920	1080	true	1、25 2、30000/1001	-
XAVC-HD_intra_422_10bit_class200	1920	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class100	2048	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class300	3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/10	-

				01	
XAVC-4K_intra_422_10bit_class300	4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class480	3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit_class480	4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	-
XAVC-4K_intra_422_10bit	2048	1080	false	1、25 2、50 3、24000/1001 4、30000/1001	1、93000 2、185000 3、89000 4、111000 5、222000

				5、60000/1001	
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50000 5、35000或50000

版权所有：腾讯云计算（北京）有限责任公司

				5、60000/1001	
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 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
1280	720	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
1280	1080	false	1、24000/1001 2、30000/1001	取值范围:[2000,3000000]
1280	1080	true	1、60000/1001	取值范围:[2000,3000000]
1440	1080	false	1、24000/1001 2、25 3、30000/1001	取值范围:[2000,3000000]
1440	1080	true	1、50 2、60000/1001	取值范围:[2000,3000000]
1920	1080	false	1、25 2、50 3、24000/1001	取值范围:[2000,3000000]

			4、30000/1001 5、60000/1001	
1920	1080	true	1、50 2、60000/1001	取值范围:[2000,3000000]
2048	1080	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
2048	1556	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
3840	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
4096	2160	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
5120	2700	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
6144	3240	false	1、25 2、50 3、24000/1001 4、30000/1001 5、60000/1001	取值范围:[2000,3000000]
8192	4320	false	1、25 2、50 3、24000/1001	取值范围:[2000,3000000]

			4、30000/1001 5、60000/1001	

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	默认值	是否必选
Start	Request.TimeInterval	开始时间 • [0， 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	0	否
Duration	Request.TimeInterval	持续时间 • [0， 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	视频原始时长	否

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	默认值	是否必选
Codec	Request.Audio	编解码格式，pcm_s24le、aac、mp3	String	无	是
Remove	Request.Audio	是否删除源音频流，取值 true、false	String	false	否

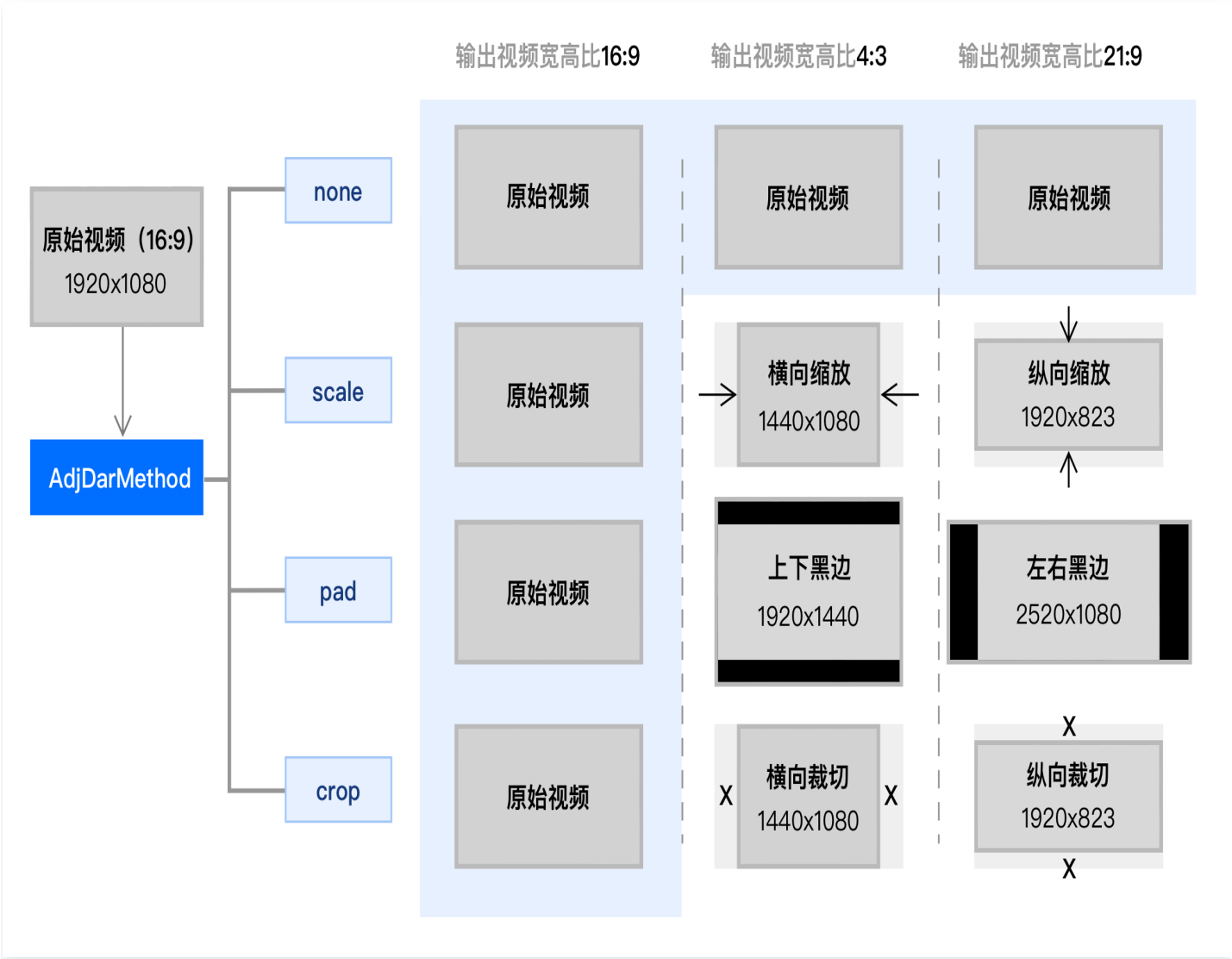
Container 类型 TransConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
AdjDarMethod	Request.TransConfig	分辨率调整方式 • 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	String	none	否

IsCheckReso	Request.TransConfig	是否检查分辨率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
ResoAdjMethod	Request.TransConfig	分辨率调整方式 • 当 IsCheckReso 为 true 时生效，取值0、1； • 0 表示使用原视频分辨率； • 1 表示返回转码失败	String	0	否
IsCheckVideoBitrate	Request.TransConfig	是否检查视频码率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
VideoBitrateAdjMethod	Request.TransConfig	视频码率调整方式 • IsCheckVideoBitrate 为 true 时生效，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	String	0	否
IsCheckAudioBitrate	Request.TransConfig	是否检查音频码率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
AudioBitrateAdjMethod	Request.TransConfig	音频码率调整方式 • IsCheckAudioBitrate 为 true 时生效，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	String	0	否
IsCheckVideoFps	Request.TransConfig	是否检查视频帧率，取值 true、false 当为 false 时，按照配置参数转码	String	false	否
VideoFpsAdjMethod	Request.TransConfig	视频帧率调整方式 • IsCheckVideoFps 为 true 时生效，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	String	0	否
DeleteMetadata	Request.TransConfig	是否删除文件中的 MetaData 信息，取值 true、false	String	fals	否

				e	
IsHdr2Sdr	Request.TransConfig	是否开启 HDR 转 SDR，取值 true、false	String	false	否

AdjDarMethod 参数图示：



响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <Template>
    <Tag>TranscodePro</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <TransProTpl>
      <Container>
        <Format>mxf</Format>
      </Container>
      <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
      </Video>
      <Audio>
        <Codec>pcm_s24le</Codec>
      </Audio>
      <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
      </TransConfig>
      <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
      </TimeInterval>
    </TransProTpl>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
    <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
  </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
TemplateId	Response.Template	模板 ID	String
Name	Response.Template	模板名称	String
BucketId	Response.Template	模板所属存储桶	String
Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，TranscodePro	String
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

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>TranscodePro</Tag>
    <Name>TemplateName</Name>
    <Container>
        <Format>mxf</Format>
    </Container>
    <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
    </Video>
    <Audio>
        <Codec>pcm_s24le</Codec>
    </Audio>
    <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
    </TransConfig>
    <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
    </TimeInterval>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <Template>
    <Tag>TranscodePro</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <TransProTpl>
      <Container>
        <Format>mxf</Format>
      </Container>
      <Video>
        <Codec>xavc</Codec>
        <Profile>XAVC-HD_422_10bit</Profile>
        <Width>1920</Width>
        <Height>1080</Height>
        <Interlaced>true</Interlaced>
        <Fps>30000/1001</Fps>
        <Bitrate>50000</Bitrate>
      </Video>
      <Audio>
        <Codec>pcm_s24le</Codec>
      </Audio>
      <TransConfig>
        <AdjDarMethod>scale</AdjDarMethod>
        <IsCheckReso>false</IsCheckReso>
        <ResoAdjMethod>1</ResoAdjMethod>
      </TransConfig>
      <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
      </TimeInterval>
    </TransProTpl>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
    <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
  </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

版权所有：腾讯云计算（北京）有限责任公司

更新模板

最近更新时间：2023-08-01 15:45:53

功能描述

更新音视频转码模板。

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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</AudioS
    <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</AudioS
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <TimeInterval>
        <Start>0</Start>
        <Duration>60</Duration>
      </TimeInterval>
      <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Aud
        <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</Aud
        <MixMode>Once</MixMode>
        <Replace>true</Replace>
        <EffectConfig>
          <EnableStartFadein>true</EnableStartFadein>
          <StartFadeinTime>3</StartFadeinTime>
          <EnableEndFadeout>false</EnableEndFadeout>
          <EndFadeoutTime>0</EndFadeoutTime>
          <EnableBgmFade>true</EnableBgmFade>
          <BgmFadeTime>1.7</BgmFadeTime>
        </EffectConfig>
      </AudioMixArray>
      <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
      <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	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-
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>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</AudioS
        <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</AudioS
      <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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <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>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Aud
        <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</Aud
        <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>
```

更新模板 转码 pro

最近更新时间：2023-08-01 16:46:23

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <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-
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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <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>
```

极速高清转码
提交任务

最近更新时间：2023-08-01 16:46:23

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</WatermarkTemp
```

```
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否

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	文件路径	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	模板 ID, 此处需要传入极速高清模板 ID	String	是
WatermarkTemplateId	Request.Operation	水印模板 ID，可以传多个水印模板 ID，最多传3个	String 数组	否
Watermark	Request.Operation	水印模板参数，同创建水印模板接口中的 Request.Watermark，最多传3个	Container 数组	否
DigitalWatermark	Request.Operation	数字水印参数，详情见 DigitalWatermark	Container	否
Output	Request.Operation	结果输出配置	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

水印参数可选，可以通过 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
        <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.${ext}</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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	String
Progress	Response.JobsDetail	任务进度百分比，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	String
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组
WatermarkTemplateId	Response.JobsDetail.Operation	同请求中的 Request.Operation.WatermarkTemplateId	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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	Container
DigitalWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.DigitalWatermark	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
```

```
Host: test-1234567890.ci.ap-chongqing.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</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <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.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-10 18:13:42

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，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)</CodecL
            <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.${ext}</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	同提交极速高清转码任务接口中的 Response.JobsDetail	Container 数组

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)</CodecL
            <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </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.${ext}</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)</CodecL
          <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.${ext}</Object>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <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 Coding)",
                    "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.${ext}",
        "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"
      }
    }]
}
```

创建模板

最近更新时间：2023-07-24 14:18:43

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Fps>30</Fps>
        <Bufsize>1000</Bufsize>
        <Maxrate>10</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
Tag	Request	模板类型: HighSpeedHd	String	是
Name	Request	模板名称。仅支持中文、英文、数字、_、-和 *，长度不超过 64	String	是
Container	Request	容器格式	Container	是
Video	Request	视频信息	Container	是
TimeInterval	Requ	时间区间	Conta	否

	est		iner	
Audio	Request	音频信息	Container	否
TransConfig	Request	转码配置	Container	否

说明：
不传 Audio，相当于删除音频信息。

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: 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	默认值	是否必选
Codec	Request.Video	编解码格式 • H.264	String	H.264	否

		• H.265			
Width	Request.Video	宽 • 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	String	视频原始宽度	否
Height	Request.Video	高 • 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	String	视频原始高度	否
Fps	Request.Video	帧率 • 值范围：(0，60] • 单位：fps	String	无	否
Profile	Request.Video	编码级别 • 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。	String	high	否
Bitrate	Request.Video	视频输出文件的码率 • 值范围：[10，50000],单位：Kbps • auto 表示自适应码率	String	无	否
Crf	Request.Video	码率-质量控制因子 • 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	String	无	否
Gop	Request.Video	关键帧间最大帧数，值范围：[0，100000]	String	无	否
Preset	Request.	视频算法器预置	Stri	medium	否

版权所有：腾讯云计算（北京）有限责任公司

	Video	• 仅H.264支持该参数 • 取值 veryfast、fast、medium、slow、slower	ng		
Bufsize	Request.Video	缓冲区大小 • 值范围：[1000，128000] • 单位：Kb	String	无	否
Maxrate	Request.Video	视频码率峰值 • 值范围：[10，50000] • 单位：Kbps	String	无	否
Pixfmt	Request.Video	视频颜色格式 • H.264支持：yuv420p、auto • H.265支持：yuv420p、yuv420p10le、auto	String	无	否
Rotate	Request.Video	旋转角度 • 值范围：[0, 360) • 单位：度	String	无	否
Roi	Request.Video	Roi 强度，取值 none、low、medium、high	String	none	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	默认值	是否必选
Start	Request.TimeInterval	开始时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	无	否
Duration	Request.TimeInterval	持续时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	无	否

Container 类型 Audio 的具体数据描述如下：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
Codec	Request.Audio	编解码格式，取值 aac、mp3	String	aac	否
Samplerate	Request.Audio	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	String	44100	否
Bitrate	Request.Audio	原始音频码率 • 单位：Kbps • 值范围：[8，1000]	String	无	否
Channels	Request.Audio	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	String	无	否
Remove	Request.Audio	是否删除源音频流，取值 true、false	String	false	否

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

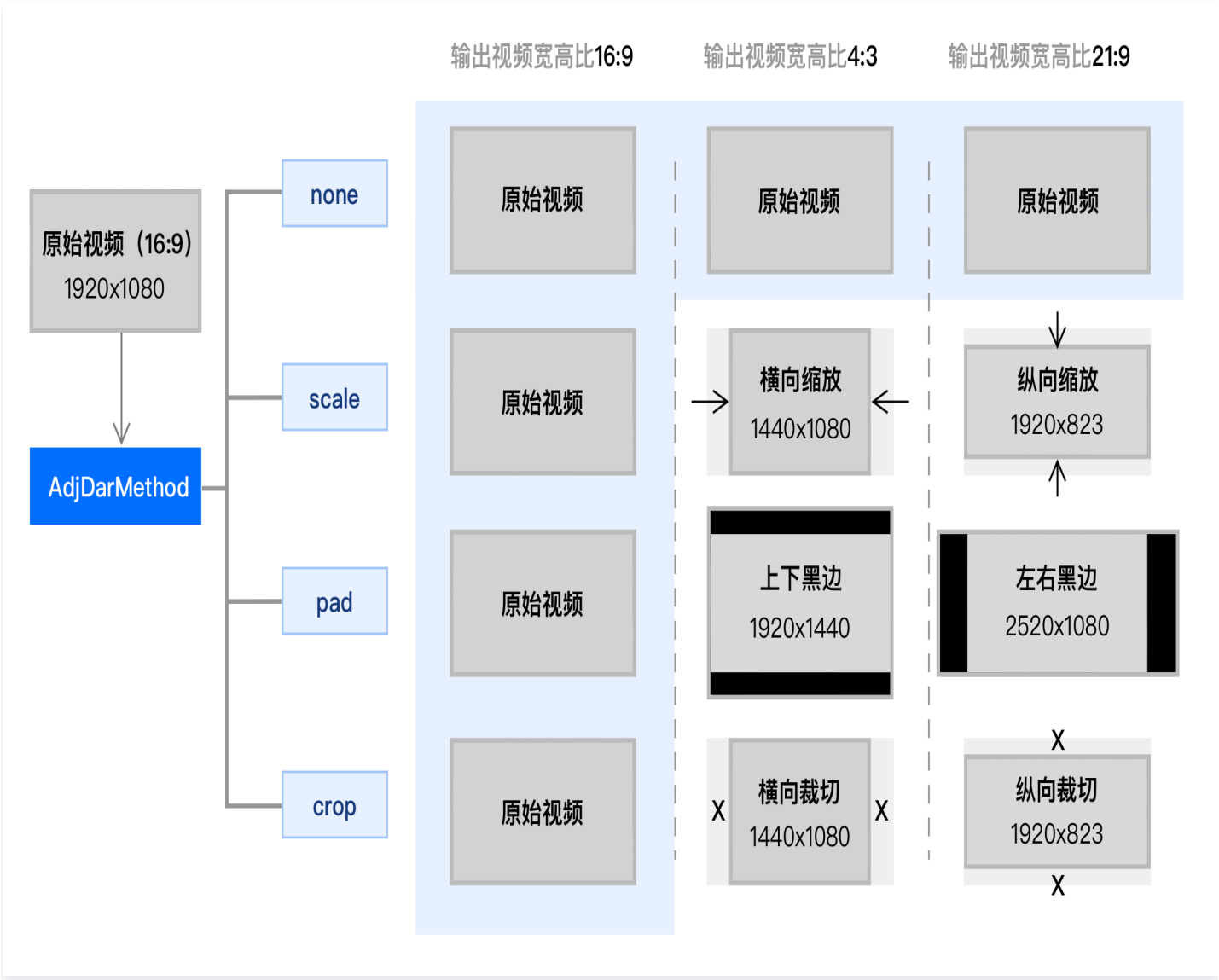
Container 类型 TransConfig 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认	是否必选

				值	
AdjDarMethod	Request.TransConfig	分辨率调整方式 • 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	String	none	否
IsCheckReso	Request.TransConfig	是否检查分辨率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
ResoAdjMethod	Request.TransConfig	分辨率调整方式 • 当 IsCheckReso 为 true 时生效，取值0、1； • 0 表示使用原视频分辨率； • 1 表示返回转码失败	String	0	否
IsCheckVideoBitrate	Request.TransConfig	是否检查视频码率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
VideoBitrateAdjMethod	Request.TransConfig	视频码率调整方式 • IsCheckVideoBitrate 为 true 时生效，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	String	0	否
IsCheckAudioBitrate	Request.TransConfig	是否检查音频码率，取值 true、false • 当为 false 时，按照配置参数转码	String	false	否
AudioBitrateAdjMethod	Request.TransConfig	音频码率调整方式 • IsCheckAudioBitrate 为 true 时生效，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	String	0	否
IsCheckVideoFps	Request.TransConfig	是否检查视频帧率，取值 true、false 当为 false 时，按照配置参数转码	String	false	否
VideoFpsAdj	Request.Tr	视频帧率调整方式	Stri	0	否

Method	ansConfig	• IsCheckVideoFps 为 true 时生效，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	ng		
DeleteMetadata	Request.TransConfig	是否删除文件中的 MetaData 信息，取值 true、false	String	false	否
IsHdr2Sdr	Request.TransConfig	是否开启 HDR 转 SDR，取值 true、false	String	false	否
HlsEncrypt	Request.TransConfig	hls 加密配置	Container	无	无

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
IsHlsEncrypt	Request.TransConfig.HlsEncrypt	是否开启HLS加密，取值true/false	String	否	false
UriKey	Request.TransConfig.HlsEncrypt	HLS加密的key，当IsHlsEncrypt为true时，该参数才有意义	String	否	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>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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>1000</Bufsize>
        <Maxrate>10</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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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

版权所有：腾讯云计算（北京）有限责任公司

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-s
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>1000</Bufsize>
```

```
      <Maxrate>10</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>
    <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
    <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>1000</Bufsize>
            <Maxrate>10</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>
```

更新模板

最近更新时间：2023-08-10 18:13:42

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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-s
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
```

版权所有：腾讯云计算（北京）有限责任公司

```
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

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

闲时转码
提交任务

最近更新时间：2023-06-20 17:32:17

功能描述

提交一个闲时转码任务。

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</WatermarkTemp
```

```
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <RemoveWatermark>
            <Dx>150</Dx>
            <Dy>150</Dy>
            <Width>75</Width>
            <Height>75</Height>
        </RemoveWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否

CallBack	Request	任务回调地址，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	文件路径	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	转码模板 ID	String	否
Transcode	Request.Operation	转码模板参数	Container	否
RemoveWatermark	Request.Operation	去除水印参数, H265、AV1编码暂不支持该参数	Container	否
Output	Request.Operation	结果输出配置	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
FreeTranscode	Request.Operation	闲时转码标记位, true/false	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	限制
Container	Request.Operation.Transcode	容器格式	Container	是	无
Video	Request.Operation.Transcode	视频信息	Container	否	不传 Video，相当于删除视频信息
TimeInterval	Request.Operation.Transcode	时间区间	Container	否	无
Audio	Request.Operation.Transcode	音频信息	Container	否	不传 Audio，相当于删除音频信息
TransConfig	Request.Operation.Transcode	转码配置	Container	否	无

说明

Video 和 Audio 不支持同时不传。

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.Operation.Transcode.Container	封装格式：取值见下表	String	是

设定 container，音频视频支持的格式如下表所示：

Container	Audio Codecs	Video Codecs
mp4	aac、mp3	H.264、H.265、AV1
hls	aac、mp3	H.264、H.265
flv	aac、mp3	H.264
mp3	mp3	不支持
flac	flac	不支持
m4a	aac	不支持

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.Operation.Transcode.Video	编解码格式	String	否	H.264	• H.264 • H.265 • AV1
Fps	Request.Operation.Transcode.Video	帧率	String	否	无	• 值范围：(0, 60] • 单位：fps
Bitrate	Request.Operation.Transcode.Video	视频输出文件的码率	String	否	无	• 值范围：[128, 35000] • 单位：Kbps
Width	Request.Operation.Transcode.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.Operation.Transcode.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
Gop	Request.Operation.Transcode.Video	关键帧间最大帧数	String	否	无	值范围：[1, 100000]
Crf	Request.Operation.Transcode.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

版权所有：腾讯云计算（北京）有限责任公司

Remove	Request.Operation.Transcode.Video	是否删除视频流	String	否	false	true、false, 不可以与 Audio.Remove 同时为true

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.Operation.Transcode.Audio	编解码格式	String	否	aac	aac、mp3、flac
Bitrate	Request.Operation.Transcode.Audio	原始音频码率	String	否	无	• 单位：Kbps • 值范围：[26，256]
Samplerate	Request.Operation.Transcode.Audio	采样率	String	否	44100	44100
Channels	Request.Operation.Transcode.Audio	声道数	String	否	2	1、2
Remove	Request.Operation.Transcode.Audio	是否删除源音频流	String	否	false	true、false，不可以与 Video.Remove 同时为true

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.Transcode.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.Operation.Transcode.TimeInterval	持续时间	String	否	视频原	• [0 视频时长] • 单位为秒 • 支持 float 格式，执行精

					始时长	度精确到毫秒

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.Operation.Transcode.TransConfig	分辨率调整方式	String	否	无	• scale(拉伸)、pad(黑边填充) • 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

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 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

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <RemoveWatermark>
        <Dx>150</Dx>
```

```
            <Dy>150</Dy>
            <Width>75</Width>
            <Height>75</Height>
        </RemoveWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/out.${ext}</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

版权所有：腾讯云计算（北京）有限责任公司

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
Transcode	Response.JobsDetail.Operation	同请求中的 Request.Operation.Transcode	Container
Watermark	Response.JobsDetai	同请求中的	Containe

	l.Operation	Request.Operation.Watermark	r 数组
RemoveWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.RemoveWatermark	Container
WatermarkTemplateId	Response.JobsDetail.Operation	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	同请求中的 Request.Operation.DigitalWatermark	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：
详情见 MediaResult 。

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-
Host: test-1234567890.ci.ap-chongqing.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</WatermarkTemp
    <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
    <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
    <RemoveWatermark>
      <Dx>150</Dx>
      <Dy>150</Dy>
      <Width>75</Width>
      <Height>75</Height>
    </RemoveWatermark>
    <Output>
      <Region>ap-chongqing</Region>
      <Bucket>test-123456789</Bucket>
      <Object>output/out.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
        <RemoveWatermark>
          <Dx>150</Dx>
          <Dy>150</Dy>
          <Width>75</Width>
          <Height>75</Height>
        </RemoveWatermark>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-123456789</Bucket>
          <Object>output/out.${ext}</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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host: test-1234567890.ci.ap-chongqing.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>
  </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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Transparency>30</Transparency>
        </Text>
    </Watermark>
    <RemoveWatermark>
        <Dx>150</Dx>
        <Dy>150</Dy>
        <Width>75</Width>
        <Height>75</Height>
    </RemoveWatermark>
    <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-123456789</Bucket>
        <Object>output/out.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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>
        </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-1234567890</Bucket>
          <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

音视频转封装（音视频分段）提交任务

最近更新时间：2023-08-14 17:01:43

功能描述

提交一个转封装（或音视频分段）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>mp4</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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否

CallBack	Request	任务回调地址，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	文件路径	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	转封装参数	Container	是
Output	Request.Operation	结果输出配置	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

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
Format	Request.Operation.Segment	封装格式，支持 aac、mp3、flac、mp4、ts、mkv、avi、hls、m3u8	String	是
Duration	Request.Operation.Segment	转封装时长 • 单位：秒 • 不小于5的整数 • 不设置 Duration 时，表示只转封装格式不分段	String	否
TranscodeIndex	Request.Operation.Segment	处理的流编号，对应媒体信息中的 Response.MediaInfo.Stream.Video.I	String	否

		ndex 和 Response.MediaInfo.Stream.Audio.Index，详见 获取媒体信息接口		
HlsEncrypt	Request.Operation.Segment	hls 加密配置,当封装格式为 hls 和 m3u8 时生效	Container	否
StartTime	Request.Operation.Segment	开始时间 • 取值范围：[0,视频时长]，默认值为0 • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	否
EndTime	Request.Operation.Segment	结束时间 • 取值范围：[0, 视频时长]，默认值为视频结束时间 • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	否

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
IsHlsEncrypt	Request.Operation.Segment.HlsEncrypt	是否开启 HLS 加密，取值 true/false，默认值 false	String	否
UriKey	Request.Operation.Segment.HlsEncrypt	HLS 加密的 key，当 IsHlsEncrypt 为 true 时，该参数才有意义	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

版权所有：腾讯云计算（北京）有限责任公司

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} 参数作为自定义转封装后每一小段音/视频流的输出序号	String	是

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id
${number}	产物索引，从0开始计数

响应

响应头

此接口仅返回公共响应头部，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>mp4</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>
      <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 数组

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：Segment	String

State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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.JobsDe	同请求中的 Request.Operation.Segment	Contai

版权所有：腾讯云计算（北京）有限责任公司

	tail.Operation		ner
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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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1：mp4转封装

```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.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-1234567890</Bucket>
            <Object>output/out-${number}</Object>
        </Output>
```

```
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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-1234567890</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>
      <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-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Segment</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <Segment>
            <Format>mp4</Format>
            <Duration>5</Duration>
            <HlsEncrypt>
                <IsHlsEncrypt>true</IsHlsEncrypt>
                <UriKey>test-key</UriKey>
            </HlsEncrypt>
        </Segment>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/out-${number}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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-1234567890</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <Segment>
        <Format>mp4</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>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
  </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

回调信息

最近更新时间：2023-08-10 18:13:42

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，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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</CodecL
            <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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>
        <JobLevel>0</JobLevel>
      </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	同提交转封装任务接口中的 Response.JobsDetail	Container 数组

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T19:14:31+0800</CreationTime>
    <EndTime>2022-06-30T19:14:36+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>jd0c0da74f86511ec8a5a87e016101404</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format>
          <Bitrate>2629.749000</Bitrate>
          <Duration>215.680000</Duration>
          <FormatLongName>QuickTime / MOV</FormatLongName>
          <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
```

```
        <NumProgram>0</NumProgram>
        <NumStream>2</NumStream>
        <Size>70898041</Size>
        <StartTime>0.000000</StartTime>
    </Format>
    <Stream>
        <Audio>
            <Bitrate>256.000000</Bitrate>
            <Channel>2</Channel>
            <ChannelLayout>stereo</ChannelLayout>
            <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
            <CodecName>aac</CodecName>
            <CodecTag>0x6134706d</CodecTag>
            <CodecTagString>mp4a</CodecTagString>
            <CodecTimeBase>1/48000</CodecTimeBase>
            <Duration>215.637000</Duration>
            <Index>1</Index>
            <Language>eng</Language>
            <SampleFmt>fltp</SampleFmt>
            <SampleRate>48000</SampleRate>
            <StartTime>0.000000</StartTime>
            <Timebase>1/48000</Timebase>
        </Audio>
        <Subtitle/>
        <Video>
            <AvgFps>24.000000</AvgFps>
            <Bitrate>2369.296000</Bitrate>
            <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</CodecL
            <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>
      <JobLevel>0</JobLevel>
    </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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <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</CodecL
            <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>
        <JobLevel>0</JobLevel>
    </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.",
        "JobLevel": 0
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T19:14:32+0800",
    "State": "Success",
    "Tag": "Segment",
    "Workflow": {
        "Name": "Segment_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
    }
  }]
}
```

音视频流分离
提交任务

最近更新时间：2023-08-01 16:46:23

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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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

CallBack MqConfig	Req uest	任务回调TDMQ配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig	Cont ainer	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	文件路径	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	结果输出配置	Con tain er	是
UserData	Request. Operation	透传用户信息，可打印的 ASCII 码，长度不超过1024	Stri ng	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	Stri ng	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.Operatio n.Output	存储桶的地域	String	是
Bucket	Request.Operatio n.Output	存储结果的存储桶	String	是
StreamExtrac t	Request.Operatio n.Output	结果文件配置信息，最多支持10个流配置	Contain er 数组	是

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

版权所有：腾讯云计算（北京）有限责任公司

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	是
Object	Request.Operation.Output.StreamExtract	结果输出文件名	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>2022-06-27T15:23:11+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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任	String

版权所有：腾讯云计算（北京）有限责任公司

		务会变为暂停状态 • Cancel：任务被取消执行	
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，详情见 MediaResult	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

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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>
    <CallBack>http://callback.demo.com</CallBack>
    <CallBackFormat>JSON<CallBackFormat>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
```

版权所有：腾讯云计算（北京）有限责任公司

```
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>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>
```

回调信息

最近更新时间：2023-08-10 18:13:42

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，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>
        <JobLevel>0</JobLevel>
      </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	同提交流分离任务接口中的 Response.JobsDetail	Container数组

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>
        <JobLevel>0</JobLevel>
      </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.",
      "JobLevel": 0
```

```
        },
        "QueueId": "p2242ab62c7c94486915508540933a2c6",
        "StartTime": "2022-06-30T19:27:40+0800",
        "State": "Success",
        "Tag": "StreamExtract"
    }]
}
```

音视频拼接
提交任务

最近更新时间：2023-08-01 16:46:23

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-1234567890</Bucket>
        <Object>output/out.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	否
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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	拼接模板 ID	String	否
ConcatTemplate	Request.Operation	拼接参数	Container	否
Output	Request.Operation	结果输出配置	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

注意

拼接参数必须设置，可以通过 TemplateId 或 ConcatTemplate 设置， TemplateId 优先级更高。

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	默认值	是否必选
ConcatFragment	Request.Operation.ConcatTemplate	拼接节点，支持多个文件，按照文件顺序拼接	Container 数组	无	否
Audio	Request.Operation.C	音频参数，同创建拼接模板接口中的 Request.ConcatTemplate.Audio	Container	无	否

	oncatTemplate				
Video	Request.Operation.ConcatTemplate	视频参数，同创建拼接模板接口中的 Request.ConcatTemplate.Video	Container	无	否
Container	Request.Operation.ConcatTemplate	封装格式，同创建拼接模板接口中的 Request.ConcatTemplate.Container	Container	无	是
AudioMix	Request.Operation.ConcatTemplate	混音参数，详情见 AudioMix	Container	无	否
AudioMixArray	Request.Operation.Transcode	混音参数数组，最多同时传2个	Container 数组	无	否
Index	Request.Operation.ConcatTemplate	Input 节点位于 ConcatFragment 序列索引，不能大于 ConcatFragment 长度	String	0	否
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，取值 true/false	String	false	否

说明

- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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	默认值	是否必选
Url	Request.Operation.	同 bucket 对象地址，需要 urlEncode	Stri	无	是

	ConcatTemplate.ConcatFragment		ng		
FragmentIndex	Request.Operation.ConcatTemplate.ConcatFragment	拼接对象的索引位置，大于等于0的整数	String	0	否
StartTime	Request.Operation.ConcatTemplate.ConcatFragment	开始时间 • [0, 视频时长] • 单位为秒 • 当 Request.Operation.ConcatTemplate.DirectConcat 为 true 时不生效	String	视频开始	否
EndTime	Request.Operation.ConcatTemplate.ConcatFragment	结束时间 • [0, 视频时长] • 单位为秒 • 当 Request.Operation.ConcatTemplate.DirectConcat 为 true 时不生效	String	视频结束	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

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义

${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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.${ext}</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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDet	任务优先级	Strin

	ail.Operation		g

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****-s
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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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.${ext}</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****-s
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</Url>
      </ConcatFragment>
      <ConcatFragment>
        <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp4</Url>
      </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>
      <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>
      </ConcatTemplate>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-123456789</Bucket>
        <Object>output/out.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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.mp4</Ur
  </ConcatFragment>
  <ConcatFragment>
    <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp4</Url
  </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>
  <AudioMixArray>
    <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.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>
  </AudioMixArray>
  <AudioMixArray>
    <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            </AudioMixArray>
            <Index>1</Index>
        </ConcatTemplate>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.${ext}</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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR98****-s
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</Url>
                <FragmentIndex>0</FragmentIndex>
                <StartTime>0</StartTime>
                <EndTime>6</EndTime>
            </ConcatFragment>
            <ConcatFragment>
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/middle.mp4</Ur
                <FragmentIndex>1</FragmentIndex>
            </ConcatFragment>
            <ConcatFragment>
                <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp4</Url>
                <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.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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.mp4</Ur
            <StartTime>0</StartTime>
            <EndTime>6</EndTime>
          </ConcatFragment>
          <ConcatFragment>
            <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/middle.mp4</
          </ConcatFragment>
          <ConcatFragment>
            <Url>http://test-123456789.cos.ap-chongqing.myqcloud.com/end.mp4</Url
            <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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-10 18:13:42

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，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</CodecL
            <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.${ext}</Object>
            <Region>ap-chongqing</Region>
         </Output>
         <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
         <TemplateName>template_name3544697</TemplateName>
         <UserData>This is my Concat job.</UserData>
         <JobLevel>0</JobLevel>
      </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	同提交拼接任务接口中的 Response.JobsDetail	Container 数组

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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</CodecL
            <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <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.${ext}</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
      <TemplateName>template_name3544697</TemplateName>
      <UserData>This is my Concat job.</UserData>
      <JobLevel>0</JobLevel>
    </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</CodecL
            <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
        </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.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
        <TemplateName>template_name3544697</TemplateName>
        <UserData>This is my Concat job.</UserData>
        <JobLevel>0</JobLevel>
      </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.${ext}",
        "Region": "ap-chongqing"
      },
      "TemplateId": "t140325e8918ac423da53b7d78dbbab564",
      "TemplateName": "template_name3544697",
      "UserData": "This is my Concat job.",
      "JobLevel": 0
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T18:33:08+0800",
    "State": "Success",
    "Tag": "Concat",
    "Workflow": {
      "Name": "Concat_1581665960537",
      "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
      "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
      "WorkflowName": "workflow-test"
    }
  }]
}
```


版权所有：腾讯云计算（北京）有限责任公司

创建模板

最近更新时间：2023-08-01 16:46:24

功能描述

创建音视频拼接模板。

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</Url>
</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>
<AudioMixArray>
  <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Aud
  <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</Aud
  <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>
</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*，长度不超过 64	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	默认值	是否必选
ConcatFragment	Request.ConcatTemplate	拼接节点	Container 数组	无	是
Audio	Request.ConcatTemplate	音频参数	Container	媒体原始值	否
Video	Request.ConcatTemplate	视频参数	Container	媒体原始值	否
Container	Request.ConcatTemplate	封装格式	Container	无	是
AudioMix	Request.Conc	混音参数，详情见	Contain	无	否

版权所有：腾讯云计算（北京）有限责任公司

	atTemplate	AudioMix	er		
AudioMixArray	Request.ConcatTemplate	混音参数，最多同时传2个	Container 数组	无	否
DirectConcat	Request.ConcatTemplate	只拼接不转码，取值 true/false	String	false	否
SceneChangeInfo	Request.ConcatTemplate	转场参数	Container	无	否

说明

- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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	默认值	是否必选
Url	Request.ConcatTemplate.ConcatFragment	拼接对象地址	String	无	是
Mode	Request.ConcatTemplate.ConcatFragment	节点类型 • Start：开头 • End：结尾	String	无	是
StartTime	Request.ConcatTemplate.ConcatFragment	开始时间，单位为秒，支持float格式，执行精度精确到毫秒，当 Request.ConcatTemplate.DirectConcat 为 true 时不生效	String	无	否
EndTime	Request.ConcatTemplate.ConcatFragment	结束时间，单位为秒，支持float格式，执行精度精确到毫秒，当 Request.ConcatTemplate.DirectConcat 为 true 时不生效	String	无	否

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	默认值	是否必选
Codec	Request.ConcatTemplate.Audio	编解码格式，取值 aac、mp3	String	无	是
Samplerate	Request.ConcatTemplate.Audio	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	String	文件原采样率	否
Bitrate	Request.ConcatTemplate.Audio	音频码率 • 单位：Kbps • 值范围：[8，1000]	String	文件原音频码率	否
Channels	Request.ConcatTemplate.Audio	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	String	文件原声道数	否

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	是否必选
Format	Request.ConcatTemplate.Container	封装格式：mp4，flv，hls，ts, mp3, aac	String	是

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	默认值	是否必选
Codec	Request.ConcatTemplate.Video	编解码格式，取值H.264	String	H.264	是
Width	Request.ConcatTemplate.Video	宽 • 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	String	视频原始宽度	否
Height	Request.ConcatTemplate.Video	高 • 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	String	视频原始高度	否
Fps	Request.ConcatTemplate.Video	帧率 • 值范围：(0，60] • 单位：fps	String	视频原始帧率	否
Bitrate	Request.ConcatTemplate.Video	视频输出文件的码率 • 值范围：[10，50000] • 单位：Kbps	String	视频原始码率	否
Crf	Request.ConcatTemplate.Video	码率-质量控制因子 • 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	String	视频原始码率	否
Remove	Request.ConcatTemplate.Video	是否删除视频流，取值 true、false	String	false	否

Rotate	Request.ConcatTemplate.Video	旋转角度 • 值范围：[0, 360) • 单位：度	String	无	否

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

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
Mode	Request.ConcatTemplate.SceneChangeInfo	转场模式 • Default：不添加转场特效 • FADE：淡入淡出 • GRADIENT：渐变	String	无	是
Time	Request.ConcatTemplate.SceneChangeInfo	转场时长 单位：秒(s) 取值范围：(0, 5]，支持小数	String	3	• 单位：秒(s) • 取值范围：(0, 5]，支持小数

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>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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</Url>
  </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>
  <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>
      </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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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.Templ	更新时间	String

版权所有：腾讯云计算（北京）有限责任公司

	ate		
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****-s
Host:test-1234567890.ci.ap-chongqing.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</Url>
    </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>
      <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Aud
        <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</Aud
        <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>
    </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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
```

版权所有：腾讯云计算（北京）有限责任公司

```
<RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
<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</Url>
    </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>
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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>
      </ConcatTemplate>
      <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
      <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

更新模板

最近更新时间：2023-08-18 15:28:02

功能描述

更新音视频拼接模板。

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
<ConcatFragment>
  <Mode>Start</Mode>
  <Url>http://bucket-1250000000.cos.ap-beijing.myqcloud.com/start.mp4</Url>
</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>
<AudioMixArray>
  <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Aud
  <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</Aud
  <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>
  </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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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</Url>
      </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>
        <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>
    </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****-s
Host:test-1234567890.ci.ap-chongqing.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</Url>
        </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>
      <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</Aud
        <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</Aud
        <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>
    </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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=
```

版权所有：腾讯云计算（北京）有限责任公司

```
<Response>
  <RequestId>NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=</RequestId>
  <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</Url>
      </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>
      <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>
      </ConcatTemplate>
      <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
      <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
   </Template>
</Response>
```

精彩集锦
提交任务

最近更新时间：2023-08-01 16:46:24

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.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	父节点	描述	类型	是否必选

键字）				
Object	Request.Input	同存储桶文件路径	String	否
Url	Request.Input	支持公网下载的Url	String	否

说明：
Url 与 Object 不可同时为空，且当两者都传入时，优先使用Object

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	模板 ID	String	否
VideoMontage	Request.Operation	精彩集锦模板参数	Container	否
Output	Request.Operation	结果输出配置	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
精彩集锦参数必须设置，可以通过 TemplateId 或 VideoMontage 设置， TemplateId 优先级更高。

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
TimeInterval	Request.Operation.Transcode	同创建精彩集锦模板接口中的 Request.TimeInterval	Container	否

Container	Request.Operation.Transcode	同创建精彩集锦模板接口中的 Request.Container	Container	是
Video	Request.Operation.Transcode	同创建精彩集锦模板接口中的 Request.Video	Container	是
Audio	Request.Operation.Transcode	同创建精彩集锦模板接口中的 Request.Audio	Container	否
Scene	Request.Operation.Transcode	同创建精彩集锦模板接口中的 Request.Scene	String	否
AudioMix	Request.Operation.Transcode	混音参数，详情见 AudioMix	Container 数组	否
AudioMixArray	Request.Operation.Transcode	混音参数数组，最多同时传2个	Container 数组	否

说明

- 不传 Video，相当于删除视频信息。不传 Audio，相当于删除音频信息。Video 和 Audio 不可以同时不传。
- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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	输出结果的文件名	String	是

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式

${jobid}	任务 Id

响应

响应头

此接口仅返回公共响应头部，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.${ext}</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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
VideoMontage	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoMontage	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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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

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-
Host: test-1234567890.ci.ap-chongqing.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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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.${ext}</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-
Host: test-1234567890.ci.ap-chongqing.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>
    <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>
    <Duration>15</Duration>
  </VideoMontage>
```

```
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>
        <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.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>
        </AudioMixArray>
        <AudioMixArray>
            <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix2.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>
        </AudioMixArray>
        <Duration>15</Duration>
    </VideoMontage>
    <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-123456789</Bucket>
        <Object>output/out.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
```

```
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-09 17:33:02

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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</CodecL
            <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.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t164ac18da8bdc403fa27ac4b077a962a1</TemplateId>
        <TemplateName>videoMontage</TemplateName>
        <UserData>This is my VideoMontage job.</UserData>
        <JobLevel>0</JobLevel>
      </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	同提交精彩集锦任务接口中的 Response.JobsDetail	Container 数组

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</CodecL
            <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <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.${ext}</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t164ac18da8bdc403fa27ac4b077a962a1</TemplateId>
      <TemplateName>videoMontage</TemplateName>
      <UserData>This is my VideoMontage job.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
    <StartTime>2022-06-30T19:55:05+0800</StartTime>
    <State>Success</State>
    <Tag>VideoMontage</Tag>
  </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T19:55:05+0800</CreationTime>
```

```
<EndTime>2022-06-30T19:55:19+0800</EndTime>
<Input>
  <BucketId>test-123456789</BucketId>
  <Object>input/demo.mp4</Object>
  <Region>ap-chongqing</Region>
  <CosHeaders>
    <Key>Content-Type</Key>
    <Value>video/mp4</Value>
  </CosHeaders>
  <CosHeaders>
    <Key>x-cos-request-id</Key>
    <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
  </CosHeaders>
  <CosHeaders>
    <Key>EventName</Key>
    <Value>cos:ObjectCreated:Put</Value>
  </CosHeaders>
  <CosHeaders>
    <Key>Size</Key>
    <Value>1424687</Value>
  </CosHeaders>
</Input>
<JobId>j7b0dfbe2f86b11ec995acf5cffaf6664</JobId>
<Message/>
<Operation>
  <MediaInfo>
    <Format>
      <Bitrate>14044.868000</Bitrate>
      <Duration>2.067000</Duration>
      <FormatLongName>QuickTime / MOV</FormatLongName>
      <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
      <NumProgram>0</NumProgram>
      <NumStream>2</NumStream>
      <Size>3628843</Size>
      <StartTime>0.000000</StartTime>
    </Format>
    <Stream>
      <Audio>
        <Bitrate>126.400000</Bitrate>
        <Channel>2</Channel>
        <ChannelLayout>stereo</ChannelLayout>
        <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
        <CodecName>aac</CodecName>
        <CodecTag>0x6134706d</CodecTag>
        <CodecTagString>mp4a</CodecTagString>
```

```
            <CodecTimeBase>1/44100</CodecTimeBase>
            <Duration>2.066576</Duration>
            <Index>1</Index>
            <Language>und</Language>
            <SampleFmt>fltp</SampleFmt>
            <SampleRate>44100</SampleRate>
            <StartTime>0.000000</StartTime>
            <Timebase>1/44100</Timebase>
         </Audio>
         <Subtitle/>
         <Video>
            <AvgFps>25.000000</AvgFps>
            <Bitrate>14664.836000</Bitrate>
            <CodecLongName>H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10</CodecL
            <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.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t164ac18da8bdc403fa27ac4b077a962a1</TemplateId>
        <TemplateName>videoMontage</TemplateName>
        <UserData>This is my VideoMontage job.</UserData>
        <JobLevel>0</JobLevel>
      </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.${ext}",
          "Region": "ap-chongqing"
```

```
        },
        "TemplateId": "t164ac18da8bdc403fa27ac4b077a962a1",
        "TemplateName": "videoMontage",
        "UserData": "This is my VideoMontage job.",
        "JobLevel": 0
      },
      "QueueId": "p09d709939fef48a0a5c247ef39d90cec",
      "StartTime": "2022-06-30T19:55:05+0800",
      "State": "Success",
      "Tag": "VideoMontage",
      "Workflow": {
        "Name": "VideoMontage_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
      }
    }]
}
```

创建模板

最近更新时间：2023-07-24 14:18:44

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>
    <AudioMixArray>
      <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</AudioS
      <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</AudioS
      <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	默认值
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*，长度不超过 64	String	是	无
Duration	Request	集锦时长 • 默认自动分析时长 • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	否	无
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
Scene	Request	精彩集锦场景 • Video：普通视频 • Soccer： 足球	String	否	Video
AudioMix	Request	混音参数，详情见 AudioMix	Container	否	无
AudioMixArray	Request	混音参数数组，最多同时传2个	Container 数组	否	无

说明

- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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: 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	默认值	是否必选
Codec	Request.Video	编解码格式，取值H.264、H.265	String	H.264	否
Width	Request.Video	宽 • 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	String	视频原始宽度	否
Height	Request.Video	高 • 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	String	视频原始高度	否
Fps	Request.Video	帧率 • 值范围：(0，60] • 单位：fps	String	无	否
Bitrate	Request.Video	视频输出文件的码率 • 值范围：[10，50000]	Stri	无	否

版权所有：腾讯云计算（北京）有限责任公司

		• 单位：Kbps	ng		
Crf	Request.Video	码率-质量控制因子 • 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	String	无	否
Rotate	Request.Video	旋转角度 • 值范围：[0, 360) • 单位：度	String	无	否

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	默认值	是否必选
Codec	Request.Audio	编解码格式，取值 aac、mp3	String	aac	否
Samplerate	Request.Audio	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	String	44100	否
Bitrate	Request.Audio	原始音频码率 • 单位：Kbps • 值范围：[8，1000]	String	无	否
Channels	Request.Audio	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	String	无	否
Remove	Request.Audio	是否删除音频流，取值 true、false	String	false	否

> **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>
        <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>
      </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.Template.VideoMontage	同请求体中的 Request.Duration
TimeInterval	Response.Template.VideoMontage	同请求体中的 Request.TimeInterval
Container	Response.Template.VideoMontage	同请求体中的 Request.Container
Video	Response.Template.VideoMontage	同请求体中的 Request.Video
Audio	Response.Template.VideoMontage	同请求体中的 Request.Audio
Scene	Response.Template.VideoMontage	同请求体中的 Request.Scene
AudioMix	Response.Template.VideoMontage	同请求体中的 Request.AudioMix
AudioMixArray	Response.Template.VideoMon	同请求体中的

版权所有：腾讯云计算（北京）有限责任公司

	tage	Request.AudioMixArray

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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>
    <AudioMixArray>
        <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</AudioS
        <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</AudioS
      <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>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>
          <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>
        </VideoMontage>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
      </Template>
    </Response>
```

更新模板

最近更新时间：2023-08-01 16:46:24

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <Width>1280</Width>
      <Fps>30</Fps>
    </Video>
    <Audio>
      <Codec>aac</Codec>
      <Samplerate>44100</Samplerate>
      <Bitrate>128</Bitrate>
      <Channels>4</Channels>
      <Remove>false</Remove>
    </Audio>
    <AudioMixArray>
      <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</AudioS
      <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</AudioS
      <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	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>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <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>
      <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>
      </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-
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>
    <AudioMixArray>
      <AudioSource>https://test-xxx.cos.ap-chongqing.myqcloud.com/mix1.mp3</AudioS
      <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</AudioS
      <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>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <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>
      <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>
      </VideoMontage>
      <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
      <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    </Template>
</Response>
```

视频截帧
提交任务

最近更新时间：2023-08-01 16:46:24

功能描述

提交一个截帧任务。

> **说明：**
> - 支持的视频编码格式：H264、H265、VP8、VP9、AV1等。
> - 支持的视频封装格式：avi、mp4、mkv、mov、webm、flv、hls、ts、dash、3GP等。
> - 视频截帧任务并发任务默认限制为10QPS。

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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Requ	任务回调地址，优先级高于队列的回调地址。设置为	String	否

	est	no 时，表示队列的回调地址不产生回调		
CallBack MqConfig	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	文件路径	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	截帧模板 ID	String	否
Snapshot	Request.Operation	该任务的参数，同创建截帧模板接口中的 Request.Snapshot	Container	否
Output	Request.Operation	结果输出配置	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

截帧参数必须设置，可以通过 TemplateId 或 Snapshot 设置， TemplateId 优先级更高。

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。当有多个输出文件时必须包含 ${number} 通配符	String	否
SpriteObject	Request.Operation.Output	雪碧图的名字。当有多个输出文件时必须包含 ${number} 通配符。仅支持 jpg 格式	String	否

Request.Operation.Output.Object 和 Request.Operation.Output.SpriteObject 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id
${number}	产物索引，从0开始计数
${time}	截帧时间点（单位：毫秒）

响应

响应头

此接口仅返回公共响应头部，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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功	String

		• Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	
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.O	同请求中的	Contai

	peration	Request.Operation.Snapshot	ner
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，详见 MediaResult	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1：使用截帧模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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>
  <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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：使用截帧处理参数

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
```

回调信息

最近更新时间：2023-08-09 17:33:02

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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>
        <JobLevel>0</JobLevel>
      </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	同提交截图任务接口中的 Response.JobsDetail	Container 数组

版权所有：腾讯云计算（北京）有限责任公司

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>
          <JobLevel>0</JobLevel>
        </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>
        <JobLevel>0</JobLevel>
      </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.",
        "JobLevel": 0
      },
      "QueueId": "p2242ab62c7c94486915508540933a2c6",
      "StartTime": "2022-06-30T19:25:12+0800",
      "State": "Success",
      "Tag": "Snapshot",
      "Workflow": {
        "Name": "Snapshot_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
      }
    }]
}
```


版权所有：腾讯云计算（北京）有限责任公司

创建模板

最近更新时间：2023-08-08 14:20:52

功能描述

创建视频截图模板。

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*，长度不超过 64	String	是
Snapshot	Request	截图参数	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	默认值	是否必选
Mode	Request.Snapshot	截图模式 • 值范围：{Interval, Average, KeyFrame} • Interval 表示间隔模式，Start，TimeInterval，Count参数生效。当设置Count，未设置TimeInterval 时，表示截取所有帧，共Count张图片 • Average 表示平均模式，Start，Count参数生效。表示	String	Interval	否

		从Start开始到视频结束，按平均间隔截取共Count张图片 • KeyFrame 表示关键帧模式			
Start	Request.Snapshot	开始时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	0	否
TimeInterval	Request.Snapshot	截图时间间隔 • (0 3600] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	无	否
Count	Request.Snapshot	截图数量，取值范围 (0，10000]	String	无	是
Width	Request.Snapshot	宽 • 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	String	视频原始宽度	否
Height	Request.Snapshot	高 • 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	String	视频原始高度	否
CIParam	Request.Snapshot	截图图片处理参数，参考 图片处理	String	无	否
IsCheckCount	Request.Snapshot	是否强制检查截图个数。使用自定义间隔模式截图时，视频时长不够截取 Count 个截图，可以转为平均截图模式截取 Count 个截图	String	false	否
IsCheckBlack	Request.Snapshot	是否开启黑屏检测，取值 true、false	String	false	否
BlackLevel	Request.Snapshot	截图黑屏检测参数 • 当 IsCheckBlack=true 时有	String	空	否

版权所有：腾讯云计算（北京）有限责任公司

		效 • 值参考范围[30，100]，表示黑色像素的占比值，值越小，黑色占比越小 • Start>0，参数设置无效，不做过滤黑屏 • Start =0 参数有效，截帧的开始时间为第一帧非黑屏开始			
PixelBlackThreshold	Request.Snapshot	截图黑屏检测参数 • 当 IsCheckBlack=true 时有效 • 判断像素点是否为黑色点的阈值，取值范围：[0，255]	String	空	否
SnapshotOutMode	Request.Snapshot	截图输出模式参数 • OnlySnapshot：仅输出截图模式 • OnlySprite：仅输出雪碧图模式 • SnapshotAndSprite：输出截图与雪碧图模式	String	OnlySnapshot	否
SpriteSnapshotConfig	Request.Snapshot	雪碧图输出配置	Container	无	否

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	默认值	是否必选
CellWidth	Request.Snapshot.SpriteSnapshotConfig	单图宽度 • 值范围：[8，4096] • 单位：px	String	截图宽度	否
CellHeight	Request.Snapshot.SpriteSnapshotConfig	单图高度 • 值范围：[8，4096] • 单位：px	String	截图高度	否
Padding	Request.Snapshot.SpriteSnapshotConfig	雪碧图内边距大小 • 值范围：[0，1024] • 单位：px	String	0	否

版权所有：腾讯云计算（北京）有限责任公司

Margin	Request.Snapshot.SpriteSnapshotConfig	雪碧图外边距大小 • 值范围：[0，1024] • 单位：px	String	0	否
Color	Request.Snapshot.SpriteSnapshotConfig	背景颜色，支持颜色详见 FFmpeg	String	无	是
Columns	Request.Snapshot.SpriteSnapshotConfig	雪碧图列数，值范围：[1，10000]	String	无	是
Lines	Request.Snapshot.SpriteSnapshotConfig	雪碧图行数，值范围：[1，10000]	String	无	是
ScaleMethod	Request.Snapshot.SpriteSnapshotConfig	雪碧图缩放模式 • DirectScale: 指定宽高缩放 • MaxWHScaleAndPad: 指定最大宽高缩放填充 • MaxWHScale: 指定最大宽高缩放 • 主动设置 CellWidth 和 CellHeight 时生效	String	DirectScale	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>
  <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
Template	Response	保存结果的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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 Request.Snapshot	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-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>Snapshot</Tag>
    <Name>TemplateName</Name>
    <Snapshot>
        <Mode>Interval</Mode>
        <Width>1280</Width>
        <Height>960</Height>
        <Start>0</Start>
        <TimeInterval>0.5</TimeInterval>
        <Count>10</Count>
        <SnapshotOutMode>OnlySprite</SnapshotOutMode>
        <SpriteSnapshotConfig>
            <Color>Aquamarine</Color>
            <Columns>3</Columns>
            <Lines>3</Lines>
        </SpriteSnapshotConfig>
    </Snapshot>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
```

```
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <RequestId>NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=</RequestId>
    <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>
```

更新模板

最近更新时间：2023-08-14 18:07:21

功能描述

更新视频截图模板。

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <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>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <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-s
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
  <Tag>Snapshot</Tag>
  <Name>TemplateName</Name>
  <Snapshot>
    <Mode>Interval</Mode>
    <Width>1280</Width>
    <Height>960</Height>
    <Start>0</Start>
    <TimeInterval>0.5</TimeInterval>
    <Count>10</Count>
    <SnapshotOutMode>OnlySprite</SnapshotOutMode>
    <SpriteSnapshotConfig>
      <Color>Aquamarine</Color>
      <Columns>3</Columns>
      <Lines>3</Lines>
    </SpriteSnapshotConfig>
  </Snapshot>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

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

版权所有：腾讯云计算（北京）有限责任公司

画质增强
提交任务

最近更新时间：2023-08-01 16:46:24

功能描述

提交一个画质增强任务。

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>VideoEnhance</Tag>
  <Input>
    <Object>input/demo.mp4</Object>
  </Input>
  <Operation>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemp
```

```
    <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTem
    <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.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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：VideoEnhance	String	是
Input	Request	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否

CallBack	Request	任务回调地址，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	文件路径	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	画质增强模板 ID	String	否
VideoEnhance	Request.Operation	画质增强模板参数，同创建画质增强模板接口中的 Request.VideoEnhance	Container	否
WatermarkTemplateId	Request.Operation	水印模板 ID，可以传多个水印模板 ID，最多传3个	String 数组	否
Watermark	Request.Operation	水印模板参数，同创建水印模板接口中的 Request.Watermark，最多传3个	Container 数组	否
DigitalWatermark	Request.Operation	数字水印参数，详情见 DigitalWatermark	Container	否
Output	Request.Operation	结果输出配置	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

- 画质增强参数必须设置，可以通过 TemplateId 或 VideoEnhance 设置， TemplateId 优先级更

高。

- 水印参数可选，可以通过 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务 ID

响应

响应头

此接口仅返回公共响应头部，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>2022-06-27T14:44:10+0800</CreationTime>
    <StartTime>-</StartTime>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <EndTime>-</EndTime>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <Tag>VideoEnhance</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/demo.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>VideoEnhance_demo</TemplateName>
            <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTe
            <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</Watermark
            <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.${ext}</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 数组

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：VideoEnhance	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	String
Progress	Response.JobsDetail	任务进度百分比，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
VideoEnhance	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoEnhance	Container
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组
WatermarkTemplateId	Response.JobsDetail.Operation	同请求中的 Request.Operation.WatermarkTemplateId	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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	Container
DigitalWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.DigitalWatermark	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail	任务优先级	String

	.Operation		

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoEnhance</Tag>
    <Input>
        <Object>test.${ext}</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplat
        <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemp
        <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkTem
        <DigitalWatermark>
            <Type>Text</Type>
            <Message>123456789ab</Message>
            <Version>V1</Version>
            <IgnoreError>false</IgnoreError>
        </DigitalWatermark>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>
    <Progress>0</Progress>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <StartTime>-</StartTime>
    <EndTime>-</EndTime>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <Tag>VideoEnhance</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <TemplateName>VideoEnhance_demo</TemplateName>
      <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTer
      <WatermarkTemplateId>t12a74d11687d444deba8a6cc52051ac27</WatermarkT
      <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/out.${ext}</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-s
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoEnhance</Tag>
    <Input>
        <Object>test.${ext}</Object>
    </Input>
    <Operation>
        <VideoEnhance>
            <Transcode>
                <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>
                </Audio>
            </Transcode>
            <SuperResolution>
                <Resolution>sdtohd</Resolution>
                <EnableScaleUp>true</EnableScaleUp>
                <Version>Enhance</Version>
            </SuperResolution>
            <ColorEnhance>
                <Contrast>50</Contrast>
                <Correction>100</Correction>
                <Saturation>100</Saturation>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </ColorEnhance>
        <MsSharpen>
          <SharpenLevel>5</SharpenLevel>
        </MsSharpen>
        <SDRtoHDR>
          <HdrMode>HDR10</HdrMode>
        </SDRtoHDR>
      </VideoEnhance>
      <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.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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>
    <Progress>0</Progress>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <StartTime>-</StartTime>
    <EndTime>-</EndTime>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <Tag>VideoEnhance</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <VideoEnhance>
        <Transcode>
          <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>
          </Audio>
        </Transcode>
        <SuperResolution>
```

```
            <Resolution>sdtohd</Resolution>
            <EnableScaleUp>true</EnableScaleUp>
            <Version>Enhance</Version>
        </SuperResolution>
        <ColorEnhance>
            <Contrast>50</Contrast>
            <Correction>100</Correction>
            <Saturation>100</Saturation>
        </ColorEnhance>
        <MsSharpen>
            <SharpenLevel>5</SharpenLevel>
        </MsSharpen>
        <SDRtoHDR>
            <HdrMode>HDR10</HdrMode>
        </SDRtoHDR>
    </VideoEnhance>
    <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/out.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
```

```
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-10 18:13:43

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，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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)</CodecL
            <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.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1f4ac18da8bdc403fa27ac4bfn77a62a1</TemplateId>
        <TemplateName>VideoEnhance_12654</TemplateName>
        <UserData>This is my VideoEnhance job.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
      <StartTime>2022-06-30T20:02:18+0800</StartTime>
      <State>Success</State>
      <Progress>100</Progress>
      <Tag>VideoEnhance</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	同提交画质增强任务接口中的 Response.JobsDetail	Container 数组

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)</CodecL
                <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.${ext}</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t1f4ac18da8bdc403fa27ac4bfn77a62a1</TemplateId>
      <TemplateName>VideoEnhance_12654</TemplateName>
      <UserData>This is my VideoEnhance job.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <StartTime>2022-06-30T20:02:18+0800</StartTime>
    <State>Success</State>
    <Progress>100</Progress>
    <Tag>VideoEnhance</Tag>
  </JobsDetail>
</Response>
```

案例 2：通过工作流触发的任务回调

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T20:02:17+0800</CreationTime>
    <EndTime>2022-06-30T20:02:34+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
      <CosHeaders>
        <Key>Content-Type</Key>
        <Value>video/mp4</Value>
      </CosHeaders>
      <CosHeaders>
        <Key>x-cos-request-id</Key>
        <Value>NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5</Value>
      </CosHeaders>
      <CosHeaders>
        <Key>EventName</Key>
        <Value>cos:ObjectCreated:Put</Value>
      </CosHeaders>
      <CosHeaders>
        <Key>Size</Key>
        <Value>1424687</Value>
      </CosHeaders>
    </Input>
    <JobId>j7d0eafdaf86c11ecbdf419856486c0ab</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format>
          <Bitrate>10269.090000</Bitrate>
          <Duration>13.654000</Duration>
          <FormatLongName>QuickTime / MOV</FormatLongName>
          <FormatName>mov,mp4,m4a,3gp,3g2,mj2</FormatName>
          <NumProgram>0</NumProgram>
          <NumStream>2</NumStream>
          <Size>17526771</Size>
          <StartTime>0.000000</StartTime>
        </Format>
        <Stream>
          <Audio>
            <Bitrate>128.726000</Bitrate>
            <Channel>2</Channel>
```

```
            <ChannelLayout>stereo</ChannelLayout>
            <CodecLongName>AAC (Advanced Audio Coding)</CodecLongName>
            <CodecName>aac</CodecName>
            <CodecTag>0x6134706d</CodecTag>
            <CodecTagString>mp4a</CodecTagString>
            <CodecTimeBase>1/44100</CodecTimeBase>
            <Duration>13.652993</Duration>
            <Index>1</Index>
            <Language>und</Language>
            <SampleFmt>fltp</SampleFmt>
            <SampleRate>44100</SampleRate>
            <StartTime>0.000000</StartTime>
            <Timebase>1/44100</Timebase>
         </Audio>
         <Subtitle/>
         <Video>
            <AvgFps>24.922840</AvgFps>
            <Bitrate>10674.365000</Bitrate>
            <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)</CodecL
            <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.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <TemplateId>t1f4ac18da8bdc403fa27ac4bfn77a62a1</TemplateId>
        <TemplateName>VideoEnhance_12654</TemplateName>
        <UserData>This is my VideoEnhance job.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
      <StartTime>2022-06-30T20:02:18+0800</StartTime>
      <State>Success</State>
      <Progress>100</Progress>
      <Tag>VideoEnhance</Tag>
      <Workflow>
        <Name>VideoEnhance_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 Coding)",
            "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.${ext}",
            "Region": "ap-chongqing"
        },
        "TemplateId": "t1f4ac18da8bdc403fa27ac4bfn77a62a1",
        "TemplateName": "VideoEnhance_12654",
        "UserData": "This is my VideoEnhance job.",
        "JobLevel": 0
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T20:02:18+0800",
    "State": "Success",
    "Progress": "100",
    "Tag": "VideoEnhance",
    "Workflow": {
        "Name": "VideoEnhance_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
    }
  }]
}
```


版权所有：腾讯云计算（北京）有限责任公司

创建模板

最近更新时间：2023-08-18 15:28:02

功能描述

创建画质增强模板。

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>VideoEnhance</Tag>
  <Name>TemplateName</Name>
  <VideoEnhance>
    <Transcode>
      <Container>
        <Format>mp4</Format>
      </Container>
      <Video>
        <Codec>H.264</Codec>
```

```
            <Width>1280</Width>
            <Fps>30</Fps>
        </Video>
        <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
        </Audio>
    </Transcode>
    <SuperResolution>
        <Resolution>sdtohd</Resolution>
        <EnableScaleUp>true</EnableScaleUp>
        <Version>Enhance</Version>
    </SuperResolution>
    <ColorEnhance>
        <Contrast>50</Contrast>
        <Correction>100</Correction>
        <Saturation>100</Saturation>
    </ColorEnhance>
    <MsSharpen>
        <SharpenLevel>5</SharpenLevel>
    </MsSharpen>
    <SDRtoHDR>
        <HdrMode>HDR10</HdrMode>
    </SDRtoHDR>
    <FrameEnhance>
        <FrameDoubling>true</FrameDoubling>
    </FrameEnhance>
  </VideoEnhance>
</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	默认值	是否必选

Tag	Request	模板类型：VideoEnhance	String	无	是
Name	Request	模板名称：仅支持中文、英文、数字、_、-和*，长度不超过 64	String	无	是
VideoEnhance	Request	画质增强参数	Container	无	是

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	默认值	是否必选
Transcode	Request.VideoEnhance	转码参数	Container	无	是
SuperResolution	Request.VideoEnhance	超分辨率参数	Container	无	否
SDRtoHDR	Request.VideoEnhance	SDR to HDR 参数	Container	无	否
ColorEnhance	Request.VideoEnhance	色彩增强参数	Container	无	否
MsSharpen	Request.VideoEnhance	细节增强参数	Container	无	否
FrameEnhance	Request.FrameEnhance	帧率增强参数	Container	无	否

说明

SuperResolution、SDRtoHDR、ColorEnhance、MsSharpen、FrameEnhance 不能同时为空。

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	默认值	是否必选
Container	Request.VideoEnha	容器格式	Cont	无	是

版权所有：腾讯云计算（北京）有限责任公司

	nce.Transcode		ainer		
Video	Request.VideoEnhance.Transcode	视频信息	Container	无	是
Audio	Request.VideoEnhance.Transcode	音频信息，不传 Audio，相当于删除音频信息	Container	无	否

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：mp4/webm	String	是

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	默认值	是否必选
Codec	Request.VideoEnhance.Transcode.Video	编解码格式 • H.264 • H.265 • VP8 • VP9	String	H.264，当 Format 为 WebM 时，为 VP8	否
Width	Request.VideoEnhance.Transcode.Video	宽 • 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	String	当 Request.VideoEnhance.SuperResolution 参数为空时，默认值为视频原始宽度 当 Request.VideoEnhance.SuperResolution 参数不为空时，默认值为视频原始宽度两倍	否
Heigh	Request.VideoEnhance.	高	St	• 当	否

t	Transcode.Video	• 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	ri n g	Request.VideoEnhance.SuperResolution 参数为空时，默认值为视频原始高度 • 当 Request.VideoEnhance.SuperResolution 参数不为空时，默认值为视频原始高度两倍	
Fps	Request.VideoEnhance.Transcode.Video	帧率 • 值范围：(0, 60] • 单位：fps	St ri n g	无	否

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	默认值	是否必选
Codec	Request.VideoEnhance.Transcode.Audio	编解码格式，取值 aac、mp3、Vorbis、opus	St ri n g	aac，当 format 为 WebM 时，为 Vorbis	否
Samplerate	Request.VideoEnhance.Transcode.Audio	采样率 • 单位：Hz • 可选 8000、11025、12000、16000、22050、24000、32000、	St ri n g	44100，当 Codec 为 opus 时，默认值为 48000	否

		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，仅支持8000，16000，24000，48000			
Bitrate	Request.VideoEnhance.Transcode.Audio	原始音频码率 • 单位：Kbps • 值范围：[8，1000]	String	无	否
Channels	Request.VideoEnhance.Transcode.Audio	声道数	String	无	否

设定 container，音频视频支持的格式如下表所示：

Container	Audio Codecs	Video Codecs
mp4	aac、mp3	H.264、H.265
WebM	Vorbis、opus	VP8、VP9

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	默认值	是否必选
Contrast	Request.VideoEnhance.ColorEnhance	对比度 • 值范围：[0, 100] • auto（表示自动分析）	String	无	是

Correction	Request.VideoEnhance.ColorEnhance	色彩校正 • 值范围：[0, 1000] • auto（表示自动分析）	String	无	是
Saturation	Request.VideoEnhance.ColorEnhance	饱和度 • 值范围：[0, 300] • auto（表示自动分析）	String	无	是

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	默认值	是否必选
SharpenLevel	Request.VideoEnhance.MsSharpen	增强等级 • 值范围：[0, 10] • auto（表示自动分析）	String	无	是

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	默认值	是否必选
Resolution	Request.VideoEnhance.SuperResolution	分辨率选项 • sdtohd：标清到超清 • hdto4k：高清到4K	String	无	是
EnableScaleUp	Request.VideoEnhance.SuperResolution	自动缩放开关，取值 true、false	String	false	否
Version	Request.VideoEnhance.SuperResolution	版本 • Base：基础版 • Enhance：增强版	String	Base	否

版权所有：腾讯云计算（北京）有限责任公司

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	默认值	是否必选
HdrMode	Request.VideoEnhance.SDRtoHDR	HDR 标准 • HLG • HDR10	String	无	• HLG • HDR10

Container 类型 FrameEnhance 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
FrameDoubling	Request.VideoEnhance.FrameEnhance	是否开启插帧，取值 true、false	String	false	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>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <Template>
    <Tag>VideoEnhance</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
```

```
      <VideoEnhance>
        <transcode>
          <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>
          </Audio>
        </Transcode>
        <SuperResolution>
          <Resolution>sdtohd</Resolution>
          <EnableScaleUp>true</EnableScaleUp>
          <Version>Enhance</Version>
        </SuperResolution>
        <ColorEnhance>
          <Contrast>50</Contrast>
          <Correction>100</Correction>
          <Saturation>100</Saturation>
        </ColorEnhance>
        <MsSharpen>
          <SharpenLevel>5</SharpenLevel>
        </MsSharpen>
        <SDRtoHDR>
          <HdrMode>HDR10</HdrMode>
        </SDRtoHDR>
        <FrameEnhance>
          <FrameDoubling>true</FrameDoubling>
        </FrameEnhance>
      </VideoEnhance>
      <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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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
VideoEnhance	Response.Template	详细的模板参数	Container

Container 节点 VideoEnhance 的内容：

节点名称（关键字）	父节点	描述	类型

Transcode	Response.Template.VideoEnhance	同请求体中的 Request.Transcode	Container
SuperResolution	Response.Template.VideoEnhance	同请求体中的 Request.SuperResolution	Container
SDRtoHDR	Response.Template.VideoEnhance	同请求体中的 Request.SDRtoHDR	Container
ColorEnhance	Response.Template.VideoEnhance	同请求体中的 Request.ColorEnhance	Container
MsSharpen	Response.Template.VideoEnhance	同请求体中的 Request.MsSharpen	Container
FrameEnhance	Response.Template.VideoEnhance	同请求体中的 Request.FrameEnhance	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
   <Tag>VideoEnhance</Tag>
   <Name>TemplateName</Name>
   <VideoEnhance>
      <Transcode>
         <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>
        </Audio>
      </Transcode>
      <SuperResolution>
        <Resolution>sdtohd</Resolution>
        <EnableScaleUp>true</EnableScaleUp>
        <Version>Enhance</Version>
      </SuperResolution>
      <ColorEnhance>
        <Contrast>50</Contrast>
        <Correction>100</Correction>
        <Saturation>100</Saturation>
      </ColorEnhance>
      <MsSharpen>
        <SharpenLevel>5</SharpenLevel>
      </MsSharpen>
      <SDRtoHDR>
        <HdrMode>HDR10</HdrMode>
      </SDRtoHDR>
      <FrameEnhance>
        <FrameDoubling>true</FrameDoubling>
      </FrameEnhance>
    </VideoEnhance>
</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: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x

<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <Template>
        <Tag>VideoEnhance</Tag>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
```

```
    <Name>TemplateName</Name>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <VideoEnhance>
      <transcode>
        <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>
        </Audio>
      </Transcode>
      <SuperResolution>
        <Resolution>sdtohd</Resolution>
        <EnableScaleUp>true</EnableScaleUp>
        <Version>Enhance</Version>
      </SuperResolution>
      <ColorEnhance>
        <Contrast>50</Contrast>
        <Correction>100</Correction>
        <Saturation>100</Saturation>
      </ColorEnhance>
      <MsSharpen>
        <SharpenLevel>5</SharpenLevel>
      </MsSharpen>
      <SDRtoHDR>
        <HdrMode>HDR10</HdrMode>
      </SDRtoHDR>
      <FrameEnhance>
        <FrameDoubling>true</FrameDoubling>
      </FrameEnhance>
    </VideoEnhance>
    <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

更新模板

最近更新时间：2023-08-08 17:34:10

功能描述

更新画质增强模板。

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>VideoEnhance</Tag>
  <Name>TemplateName</Name>
  <VideoEnhance>
    <Transcode>
      <Container>
        <Format>mp4</Format>
      </Container>
      <Video>
        <Codec>H.264</Codec>
```

```
            <Width>1280</Width>
            <Fps>30</Fps>
        </Video>
        <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
        </Audio>
    </Transcode>
    <SuperResolution>
        <Resolution>sdtohd</Resolution>
        <EnableScaleUp>true</EnableScaleUp>
        <Version>Enhance</Version>
    </SuperResolution>
    <ColorEnhance>
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
    </ColorEnhance>
    <MsSharpen>
        <SharpenLevel></SharpenLevel>
    </MsSharpen>
    <SDRtoHDR>
        <HdrMode>HDR10</HdrMode>
    </SDRtoHDR>
  </VideoEnhance>
</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	同创建画质增强模板接口的 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>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <Template>
    <Tag>VideoEnhance</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <VideoEnhance>
      <transcode>
        <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>
        </Audio>
      </Transcode>
      <SuperResolution>
        <Resolution>sdtohd</Resolution>
        <EnableScaleUp>true</EnableScaleUp>
        <Version>Enhance</Version>
      </SuperResolution>
      <ColorEnhance>
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
      </ColorEnhance>
      <MsSharpen>
        <SharpenLevel></SharpenLevel>
      </MsSharpen>
      <SDRtoHDR>
        <HdrMode>HDR10</HdrMode>
      </SDRtoHDR>
    </VideoEnhance>
    <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	同创建画质增强模板接口的 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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoEnhance</Tag>
    <Name>TemplateName</Name>
    <VideoEnhance>
        <Transcode>
            <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>
            </Audio>
        </Transcode>
        <SuperResolution>
```

```
        <Resolution>sdtohd</Resolution>
        <EnableScaleUp>true</EnableScaleUp>
        <Version>Enhance</Version>
      </SuperResolution>
      <ColorEnhance>
        <Contrast></Contrast>
        <Correction></Correction>
        <Saturation></Saturation>
      </ColorEnhance>
      <MsSharpen>
        <SharpenLevel></SharpenLevel>
      </MsSharpen>
      <SDRtoHDR>
        <HdrMode>HDR10</HdrMode>
      </SDRtoHDR>
    </VideoEnhance>
</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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <Template>
      <Tag>VideoEnhance</Tag>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <Name>TemplateName</Name>
      <BucketId>test-1234567890</BucketId>
      <Category>Custom</Category>
      <VideoEnhance>
        <transcode>
          <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>
          </Audio>
        </Transcode>
        <SuperResolution>
          <Resolution>sdtohd</Resolution>
          <EnableScaleUp>true</EnableScaleUp>
          <Version>Enhance</Version>
        </SuperResolution>
        <ColorEnhance>
          <Contrast></Contrast>
          <Correction></Correction>
          <Saturation></Saturation>
        </ColorEnhance>
        <MsSharpen>
          <SharpenLevel></SharpenLevel>
        </MsSharpen>
        <SDRtoHDR>
          <HdrMode>HDR10</HdrMode>
        </SDRtoHDR>
      </VideoEnhance>
      <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

视频明水印
提交任务

最近更新时间：2023-08-11 18:10:32

功能描述

提交一个视频明水印任务。

> **说明：**
> 视频明水印为付费服务，费用同音视频转码，具体费用请参见 媒体处理费用 。

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>Watermark</Tag>
  <Input>
```

```
    <Object>input/demo.mp4</Object>
  </Input>
  <Operation>
    <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemp
    <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
    <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
    <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>
  <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：Watermark	String	是
Input	Request	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	待处理的文件路径	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
WatermarkTemplateId	Request.Operation	水印模板 ID，可以传多个水印模板 ID ，最多传3个	String 数组	否
Watermark	Request.Operation	水印模板参数，同创建水印模板接口中的 Request.Watermark ，最多传3个	Container 数组	否
Output	Request.Operation	结果输出配置	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

水印参数必须设置，可以通过 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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

版权所有：腾讯云计算（北京）有限责任公司

Object	Request.Operation.Output	输出结果的文件名	String	是

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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>Watermark</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <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 数组

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：Watermark	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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	源文件路径	String

Container 节点 Operation 的内容：

节点名称（关键字）	父节点	描述	类型
WatermarkTemplateId	Response.JobsDetail.Operation	同请求中的 Request.Operation.WatermarkTemplateId	String 数组
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output	Container
MediaInfo	Response.JobsDe	输出文件的媒体信息，任务未完成时不返回，	Contain

	tail.Operation	详见 MediaInfo	er
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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1：使用水印模板id

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

<Request>
    <Tag>Watermark</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTemp
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTemp
        <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>
    <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>2022-06-27T15:23:11+0800</CreationTime>
    <StartTime>-</StartTime>
    <EndTime>-</EndTime>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <Tag>Watermark</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <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：使用水印参数

```
POST /jobs HTTP/1.1
```

```
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>Watermark</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <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>
        <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-chongqing.myqcloud.com/shuiyin_2.png</U
                <Mode>Proportion</Mode>
                <Width>10</Width>
                <Height></Height>
                <Transparency>100</Transparency>
            </Image>
        </Watermark>
        <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>
    <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>Watermark</Tag>
      <Input>
        <BucketId>test-123456789</BucketId>
        <Object>input/demo.mp4</Object>
        <Region>ap-chongqing</Region>
      </Input>
      <Operation>
        <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>
         <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-chongqing.myqcloud.com/shuiyin_2.png</Url>
               <Mode>Proportion</Mode>
               <Width>10</Width>
               <Height></Height>
               <Transparency>100</Transparency>
            </Image>
         </Watermark>
         <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>
```

回调信息

最近更新时间：2023-08-10 18:13:43

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，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)</CodecL
            <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/Watermark.mp4</ObjectName>
         </Md5Info>
```

```
            <ObjectName>output/Watermark.mp4</ObjectName>
            <ObjectPrefix/>
            <Region>ap-chongqing</Region>
          </OutputFile>
        </MediaResult>
        <Output>
          <Bucket>test-123456789</Bucket>
          <Object>output/Watermark.mp4</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
        <UserData>This is my Watermark job.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
      <StartTime>2022-06-30T15:29:42+0800</StartTime>
      <State>Success</State>
      <Tag>Watermark</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	同提交视频明水印接口中的 Response.JobsDetail	Container 数组

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)</CodecL
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </Stream>
      </MediaInfo>
      <MediaResult>
        <OutputFile>
          <Bucket>test-123456789</Bucket>
          <Md5Info>
            <Md5>2db78d168edf2f883adce89a1b4ca033</Md5>
            <ObjectName>output/Watermark.mp4</ObjectName>
          </Md5Info>
          <ObjectName>output/Watermark.mp4</ObjectName>
          <ObjectPrefix/>
          <Region>ap-chongqing</Region>
        </OutputFile>
      </MediaResult>
      <Output>
        <Bucket>test-123456789</Bucket>
        <Object>output/Watermark.mp4</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <UserData>This is my Watermark job.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <StartTime>2022-06-30T15:29:42+0800</StartTime>
    <State>Success</State>
    <Tag>Watermark</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)</CodecL
          <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/Watermark.mp4</ObjectName>
        </Md5Info>
        <ObjectName>output/Watermark.mp4</ObjectName>
        <ObjectPrefix/>
        <Region>ap-chongqing</Region>
      </OutputFile>
    </MediaResult>
    <Output>
      <Bucket>test-123456789</Bucket>
      <Object>output/Watermark.mp4</Object>
```

```
        <Region>ap-chongqing</Region>
      </Output>
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <UserData>This is my Watermark job.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <StartTime>2022-06-30T15:29:42+0800</StartTime>
    <State>Success</State>
    <Tag>Watermark</Tag>
    <Workflow>
      <Name>Watermark_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 Coding)",
                    "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/Watermark.mp4"
            }],
            "ObjectName": ["output/Watermark.mp4"],
            "Region": "ap-chongqing"
        }
    },
    "Output": {
        "Bucket": "test-123456789",
        "Object": "output/Watermark.mp4",
        "Region": "ap-chongqing"
    },
    "WatermarkTemplateId": [
        "t1318c5f428d474afba1797f84091cbe22",
        "t1318c5f428d474afba1797f84091cbe23",
        "t1318c5f428d474afba1797f84091cbe24"
    ],
    "UserData": "This is my Watermark job.",
    "JobLevel": 0
},
"QueueId": "p2242ab62c7c94486915508540933a2c6",
"StartTime": "2022-06-30T15:29:42+0800",
```

```
      "State": "Success",
      "Tag": "Watermark",
      "Workflow": {
        "Name": "Watermark_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
      }
    }]
}
```

创建模板

最近更新时间：2023-07-24 14:18:45

功能描述

创建明水印模板。

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>simfang.ttf</FontType>
        <FontColor>0x000000</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-chongqing.myqcloud.com/shuiyin_2.png</Url:
        <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*，长度不超过 64	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	默认值	是否必选
Type	Request.Watermark	水印类型 • Text：文字水印 • Image：图片水印	String	无	是
Pos	Request.Watermark	基准位置 • TopRight • TopLeft • BottomRight • BottomLeft • Left • Right • Top • Bottom • Center	String	无	是
LocMode	Request.Watermark	偏移方式 • Relativity：按比例 • Absolute：固定位置	String	无	是
Dx	Request.Watermark	水平偏移 • 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]。	String	无	是

		• 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 • 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。	String	无	是
StartTime	Request.Watermark	水印开始时间 • [0，视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	0	否
EndTime	Request.Watermark	水印结束时间 • [0，视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	视频结束时间	否
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	无	否

版权所有：腾讯云计算（北京）有限责任公司

Image	Request.Watermark	图片水印节点	Container	无	否
Text	Request.Watermark	文本水印节点	Container	无	否

Container 类型 SlideConfig 的具体数据描述如下：

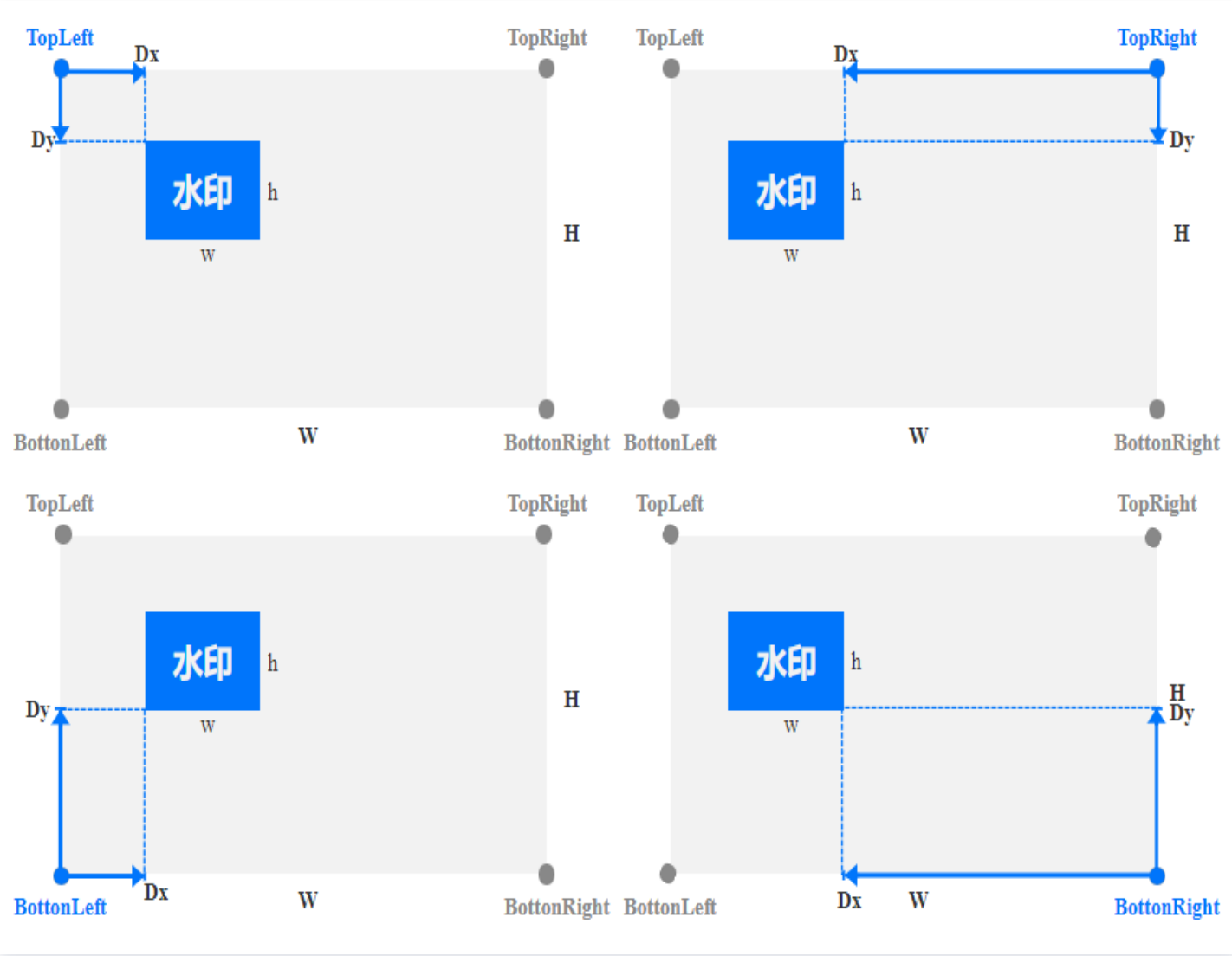
节点名称（关键字）	父节点	描述	类型	默认值	是否必选
SlideMode	Request.Watermark.SlideConfig	滑动模式 • Default: 默认不开启 • ScrollFromLeft: 从左到右滚动 • 若设置了ScrollFromLeft模式，则Watermark.Pos参数不生效	String	无	是
XSlideSpeed	Request.Watermark.SlideConfig	横向滑动速度，取值范围：[0,10]内的整数	String	0	否
YSlideSpeed	Request.Watermark.SlideConfig	纵向滑动速度，取值范围：[0,10]内的整数	String	0	否

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	默认值	是否必选
Url	Request.Watermark.Image	水印图地址(需要 Urlencode 后传入)	String	无	是
Mode	Request.Watermark.Image	尺寸模式 • Original：原有尺寸 • Proportion：按比例 • Fixed：固定大小	String	无	是
Width	Request.Watermark.	宽 • 当 Mode 为 Original 时，不支持	String	无	否

	Image	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 • 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	String	无	否
Transparency	Request.Watermark.Image	透明度，值范围：[1 100]，单位%	String	无	是
Background	Request.Watermark.Image	是否背景图，取值 true、false	String	false	否

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	默认值	是否必选
FontSize	Request.Watermark.Text	字体大小，值范围：[5 100]，单位 px	String	无	是
FontType	Request.Watermark.Text	字体类型，参考下表	String	无	是

FontColor	Request.Watermark.Text	字体颜色，格式：0xRRGGBB	String	无	是
Transparency	Request.Watermark.Text	透明度，值范围：[1 100]，单位%	String	无	是
Text	Request.Watermark.Text	水印内容，长度不超过64个字符，仅支持中文、英文、数字、_、-和*	String	无	是

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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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>simfang.ttf</FontType>
        <FontColor>0x000000</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
Template	Response	保存模板详情的容器	Container

RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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=AKIDZfbOAo7cIlgPvF9cXFrJD0****&q-sign-tir
Host: test-1234567890.ci.ap-chongqing.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>simfang.ttf</FontType>
      <FontColor>0x000000</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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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>simfang.ttf</FontType>
          <FontColor>0x000000</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-tir
Host: test-1234567890.ci.ap-chongqing.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-chongqing.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>
    <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
    <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-chongqing.myqcloud.com/shuiyin_2.png</U
                <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

更新模板

最近更新时间：2023-08-08 17:34:10

功能描述

更新明水印模板。

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-
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>
```

视频标签
提交任务

最近更新时间：2023-08-01 16:46:24

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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	Reques	执行视频标签任务的文件路径，目前支持 mp4、avi、	St	是

版权所有：腾讯云计算（北京）有限责任公司

	t.Input	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	ri n g

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	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	Request.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
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。当前版本只支持 Stream 场景	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>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

版权所有：腾讯云计算（北京）有限责任公司

State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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

版权所有：腾讯云计算（北京）有限责任公司

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 场景视频标签任务的结果列表	Container
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

版权所有：腾讯云计算（北京）有限责任公司

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	标签模型预测分数，取值范围[0,1]	Float

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	String
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	标签模型预测分数	Float
Count	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	任务出现频次	String
DetailPerSecond	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags	具体识别到人物出现的位置和时间信息	Container 数组

Container 节点 DetailPerSecond 的内容：

节点名称（关键字）	父节点	描述	类型
TimeStamp	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond	出现时间，单位为秒	String
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond	标签模型预测分数	Float
BBox	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond	左上角为原点，物体的相对坐标	Container 数组

Container 节点 BBox 的内容：

节点名称（关键字）	父节点	描述	类型
X1	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox	坐标X1的相对位置	String
Y1	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox	坐标Y1的相对位置	String
X2	Response.JobsDetail.Operation.VideoTagRes	坐标X2的相对位	Stri

	ult.StreamData.Data.PersonTags.DetailPerSecond.BBox	置	ng
Y2	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PersonTags.DetailPerSecond.BBox	坐标Y2的相对位置	String

Container 节点 PlaceTags 的内容：

节点名称（关键字）	父节点	描述	类型
Tags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	视频场景标签信息，可能不返回，内容同 Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	Container 数组
StartTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段起始时间，单位为秒	String
EndTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段结束时间，单位为秒	String
StartIndex	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段起始帧数	String
EndIndex	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	片段结束帧数	String
ClipFrameResult	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.PlaceTags	单帧识别结果top1	String

Container 节点 ActionTags 的内容：

节点名称（关键字）	父节点	描述	类型
Tags	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ActionTags	视频动作标签信息，可能不返回，内容同 Response.JobsDetail.Operat	Container 数组

版权所有：腾讯云计算（北京）有限责任公司

		ion.VideoTagResult.Stream Data.Data.Tags	
StartTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ActionTags	片段起始时间，单位为秒	String
EndTime	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ActionTags	片段结束时间，单位为秒	String

Container 节点 ObjectTags 的内容：

节点名称（关键字）	父节点	描述	类型
Objects	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags	视频物体标签信息，可能不返回，内容同 Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.Tags	Container 数组
TimeStamp	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags	识别物体时间戳，单位为秒	String

Container 节点 Object 的内容：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags.Objects	物体名称	string
Confidence	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags.Objects	标签模型预测分数，取值范围[0,1]	Float
BBox	Response.JobsDetail.Operation.VideoTagResult.StreamData.Data.ObjectTags.Objects	左上角为原点，物体的相对坐标，内容同 Response.JobsDetail.Operation.VideoTagResult.Stream	Container 数组

		Data.Data.PersonTags.DetailPerSecond.BBox	

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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>
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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>
```

回调信息

最近更新时间：2023-08-10 18:13:43

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，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>
      <JobLevel>0</JobLevel>
      <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>
  <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
        </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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <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	同提交视频标签接口中的 Response.JobsDetail	Container数组

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>
      <JobLevel>0</JobLevel>
      <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>
  <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
        </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.",
      "JobLevel": 0,
      "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",
```


版权所有：腾讯云计算（北京）有限责任公司

```
                    "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"
}]
}
```

视频质量评分
提交任务

最近更新时间：2023-08-01 16:46:24

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>
    <QualityEstimateConfig>
      <Rotate>90</Rotate>
```

```
        <Mode>vqaPlus</Mode>
      </QualityEstimateConfig>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	String	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
QualityEstimateConfig	Request.Operation	质量评分配置	Container	否
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 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
Rotate	Request.Operation.QualityEstimateConfig	旋转角度，只支持 0，90，180，270	String	0	否
Mode	Request.Operation.QualityEstimateConfig	分析模式 • general: 普通分析, 只返回视频质量分数 • vqaPlus: 返回更详细的分析结果 • 设置为vqaPlus时，Rotate 参数无效	String	general	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>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>
      <QualityEstimateConfig>
        <Rotate>90</Rotate>
        <Mode>vqaPlus</Mode>
      </QualityEstimateConfig>
      <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 数组

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	任任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
MediaInfo	Response.JobsDetail.Operation	输出视频的信息，任务未完成时不返回，详见 MediaInfo	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
VqaPlusResult	Response.JobsDetail.Operation.QualityEstimate	模式为 vqaPlus 时的分析结果	Container

Container 节点 QualityEstimate 的内容：

节点名称（关键字）	父节点	描述	类型
NoAudio	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult	无音频	bool
NoVideo	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult	无视频	bool
DetailedResult	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult	详细的分析结果，	Container 数组

Container 节点 DetailedResult 的内容：

节点名称（关键字）	父节点	描述	类型
Type	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult.DetailedResult	异常项，取值范围： • Jitter: 抖动 • Blur: 模糊 • LowLighting: 低光照 • HighLighting: 过曝 • CrashScreen: 花屏 • BlackWhiteEdge: 黑白边 • SolidColorScreen: 纯色屏 • Noise: 噪点 • Mosaic: 马赛克 • QRCode: 二维码 • AppletCode: 小程序码 • BarCode: 条形码 • LowVoice: 低音 • HighVoice: 爆音 • NoVoice: 静音 • LowEvaluation: 无参考打分低于阈值	String
Items	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult.DetailedResult	异常项信息。	Container数组

Container 节点 Items 的内容：

节点名称（关键字）	父节点	描述	类型
Confidence	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult.DetailedResult.Items	置信度，取值范围是 0 到 100。某些场景下该字段可能不返回。	Integer
StartTimeOffset	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult.DetailedResult.Items	出现的起始时间，单位：秒	Float
EndTimeOffset	Response.JobsDetail.Operation.QualityEstimate.VqaPlu	出现的结束时间，单位：秒	Float

版权所有：腾讯云计算（北京）有限责任公司

	sResult.DetailedResult.Items		
AreaCoordSet	Response.JobsDetail.Operation.QualityEstimate.VqaPlusResult.DetailedResult.Items	区域坐标(px)，即左上角坐标、右下角坐标	Integer 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 /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>QualityEstimate</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <QualityEstimateConfig>
            <Rotate>90</Rotate>
            <Mode>vqaPlus</Mode>
        </QualityEstimateConfig>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>
            <QualityEstimateConfig>
                <Rotate>90</Rotate>
                <Mode>vqaPlus</Mode>
            </QualityEstimateConfig>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
    </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-10 18:13:43

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，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T19:06:27+0800</CreationTime>
    <EndTime>2022-06-30T19:06:34+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>jb01fe2f2f86411ec9ca1073b78d316d3</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format>
          <Bitrate>834736</Bitrate>
          <Duration>13.654</Duration>
          <NumStream>1</NumStream>
          <Size>1424687</Size>
        </Format>
        <Stream>
          <Audio>
            <Bitrate>104047</Bitrate>
            <Channel>2</Channel>
            <CodecName>aac</CodecName>
            <Duration>13.653</Duration>
            <Index>1</Index>
            <SampleFmt>fltp</SampleFmt>
            <SampleRate>44100</SampleRate>
          </Audio>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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>
        <JobLevel>0</JobLevel>
      </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	同提交视频质量分析任务接口中的 Response.JobsDetail	Container 数组

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>2022-06-30T19:06:27+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>
          <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
        <StartTime>2022-06-30T19:06:28+0800</StartTime>
        <State>Success</State>
        <Tag>QualityEstimate</Tag>
    </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调，格式为 JSON

```
{
    "EventName": "TaskFinish",
    "JobsDetail": [{
        "Code": "Success",
        "CreationTime": "2022-06-30T19:06:27+0800",
        "EndTime": "2022-06-30T19:06:34+0800",
        "Input": {
            "BucketId": "test-123456789",
            "Object": "input/demo.mp4",
            "Region": "ap-chongqing"
        },
        "JobId": "jb01fe2f2f86411ec9ca1073b78d316d3",
        "Operation": {
            "MediaInfo": {
                "Format": {
                    "Bitrate": "834736",
                    "Duration": "13.654",
                    "NumStream": "1",
                    "Size": "1424687"
                },
                "Stream": {
                    "Audio": [{
```

```
                "Bitrate": "104047",
                "Channel": "2",
                "CodecName": "aac",
                "Duration": "13.653",
                "Index": "1",
                "SampleFmt": "fltp",
                "SampleRate": "44100"
            }],
            "Video": [{
                "AvgFps": "25.000",
                "Bitrate": "763774",
                "CodecName": "h264",
                "ColorPrimaries": "unknown",
                "ColorRange": "tv",
                "ColorTransfer": "smpte170m",
                "Duration": "12.960",
                "Fps": "25.000",
                "Height": "960",
                "Index": "0",
                "Level": "1.000",
                "NumFrames": "324",
                "PixFormat": "yuv420p",
                "Profile": "High",
                "Rotation": "0",
                "StartTime": "0.000",
                "Timebase": "1/12800",
                "Width": "544"
            }]
        }
    },
    "MediaResult": {

    },
    "QualityEstimate": {
        "Score": "6.460358"
    },
    "UserData": "This is my QualityEstimate job.",
    "JobLevel": 0
  },
  "QueueId": "p2242ab62c7c94486915508540933a2c6",
  "StartTime": "2022-06-30T19:06:28+0800",
  "State": "Success",
  "Tag": "QualityEstimate"
 }]
}
```

版权所有：腾讯云计算（北京）有限责任公司

视频转动图
提交任务

最近更新时间：2023-08-01 16:46:25

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.${ext}</Object>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	Contain	否

			er	

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	文件路径	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	视频转动图模板 ID	String	否
Animation	Request.Operation	视频转动图参数	Container	否
Output	Request.Operation	结果输出配置	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

视频转动图参数必须设置，可以通过 TemplateId 或 Animation 设置， TemplateId 优先级更高。

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	同创建动图模板接口中的 Request.Container	Container	否
Video	Request.Operati	同创建动图模板接口中的	Con	否

	on.Animation	Request.Video	tainer	
TimeInterval	Request.Operation.Animation	同创建动图模板接口中的 Request.TimeInterval	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
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

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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.${ext}</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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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

使用动图模板 ID

请求

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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.${ext}</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-s
Host:test-1234567890.ci.ap-chongqing.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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-11 18:10:32

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
    <EndTime>2022-06-30T15:29:48+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format>
          <Bitrate>97624.424000</Bitrate>
          <Duration>0.870000</Duration>
          <FormatLongName>CompuServe Graphics Interchange Format (GIF)</Form
          <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)</Coc
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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.${ext}</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
      <TemplateName>template_name1512456</TemplateName>
      <UserData>This is my Animation job.</UserData>
      <JobLevel>0</JobLevel>
    </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	同提交动图任务接口中的 Response.JobsDetail	Container 数组

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

版权所有：腾讯云计算（北京）有限责任公司

Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
    <EndTime>2022-06-30T15:29:48+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j682b9662f84611ecb8546d80f2baf56f</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format>
          <Bitrate>97624.424000</Bitrate>
          <Duration>0.870000</Duration>
          <FormatLongName>CompuServe Graphics Interchange Format (GIF)</Form
          <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)</Coc
            <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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.${ext}</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
      <TemplateName>template_name1512456</TemplateName>
      <UserData>This is my Animation job.</UserData>
      <JobLevel>0</JobLevel>
    </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)</Form
          <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)</Coc
            <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.${ext}</Object>
    <Region>ap-chongqing</Region>
</Output>
<TemplateId>t1f16e1dfbdc994105b31292d45710642a</TemplateId>
<TemplateName>template_name1512456</TemplateName>
<UserData>This is my Animation job.</UserData>
```

```
        <JobLevel>0</JobLevel>
      </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 Format)",
            "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": [{
```


版权所有：腾讯云计算（北京）有限责任公司

```
                "Md5": "3df1f845d2ffd20a525a93ec40014d90",
                "ObjectName": "output/animation.gif"
            }],
            "ObjectName": ["output/animation.gif"],
            "Region": "ap-chongqing"
        }
      },
      "Output": {
        "Bucket": "test-123456789",
        "Object": "output/animation.${ext}",
        "Region": "ap-chongqing"
      },
      "TemplateId": "t1f16e1dfbdc994105b31292d45710642a",
      "TemplateName": "template_name1512456",
      "UserData": "This is my Animation job.",
      "JobLevel": 0
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T15:29:43+0800",
    "State": "Success",
    "Tag": "Animation",
    "Workflow": {
      "Name": "Animation_1581665960537",
      "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
      "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
      "WorkflowName": "workflow-test"
    }
  }]
}
```

创建模板

最近更新时间：2023-07-25 09:06:34

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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*，长度不超过 64	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	默认值	是否必选
Format	Request.Container	封装格式：gif，hgif，webp	String	无	是

说明：
hgif 为高质量 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	默认值	是否必选
Codec	Request.Video	编解码格式，取值 gif,、webp	String	无	是
Width	Request.Video	宽 • 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	String	视频原始宽度	否
Height	Request.Video	高 • 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	String	视频原始高度	否
Fps	Request.Video	帧率 • 值范围：(0，60] • 单位：fps • 如果不设置，那么播放速度按照原来的时间戳。这里设置 fps 为动图的播放帧率	String	视频原始帧率	否
AnimateOnlyKeepKeyFrame	Request.Video	动图只保留关键帧 • true：AnimateTimeIntervalOfFrame 和 AnimateFramesPerSecond 无效 • false：AnimateTimeIntervalOfFrame 和 AnimateFramesPerSecond 必填	String	false	否
AnimateTimeInt	Reques	动图抽帧间隔时间	Str	无	否

ervalOfFrame	t.Video	• 取值范围：（0，视频时长] • 若设置 TimeInterval.Duration，则小于该值	ing		
AnimateFramesPerSecond	Request.Video	每秒抽帧帧数，取值范围：（0，视频帧率）	String	无	否
Quality	Request.Video	相对质量 • 取值范围：[1, 100) • webp 图像质量设定生效，gif 没有质量参数	String	无	否

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	默认值	是否必选
Start	Request.TimeInterval	开始时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	0	否
Duration	Request.TimeInterval	持续时间 • [0 视频时长] • 单位为秒 • 支持 float 格式，执行精度精确到毫秒	String	视频时长	否

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

Container 节点 Response 的内容：

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

版权所有：腾讯云计算（北京）有限责任公司

CreateTime	Response	创建时间	String
TransTpl	Response	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
Container	Request	同请求体中的 Request.Container	Container
Video	Request	同请求体中的 Request.Video	Container
TimeInterval	Request	同请求体中的 Request.TimeInterval	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=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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>
```

更新模板

最近更新时间：2023-08-08 17:34:10

功能描述

更新视频转动图模板。

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>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <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-s
Host:test-1234567890.ci.ap-chongqing.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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

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

版权所有：腾讯云计算（北京）有限责任公司

数字水印
提交任务

最近更新时间：2023-08-01 16:46:25

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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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	结果输出配置	Container	是
DigitalWatermark	Request.Operation	数字水印配置，详情见 DigitalWatermark	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

响应头

此接口仅返回公共响应头部，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

版权所有：腾讯云计算（北京）有限责任公司

JobsDetail	Response	任务的详细信息	Container 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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>
  <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
```

回调信息

最近更新时间：2023-08-11 18:10:32

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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</CodecL
            <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>
      <JobLevel>0</JobLevel>
    </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	同提交数字水印任务接口中的 Response.JobsDetail	Container 数组

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</CodecL
            <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>
      <JobLevel>0</JobLevel>
    </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.",
        "JobLevel": 0
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T18:37:38+0800",
    "State": "Success",
    "SubTag": "DigitalWatermark",
    "Tag": "DigitalWatermark"
  }]
}
```

提取数字水印
提交任务

最近更新时间：2023-08-01 16:46:25

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>
  <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	Reques	任务回调TDMQ配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig	Container	否

	t			

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	文件路径	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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型
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-s
Host:test-1234567890.ci.ap-chongqing.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>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
```

回调信息

最近更新时间：2023-08-11 18:10:32

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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>
      <JobLevel>0</JobLevel>
    </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：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型
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	同提交提取数字水印任务接口中的 Response.JobsDetail	Container 数组

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>
        <JobLevel>0</JobLevel>
      </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.",
        "JobLevel": 0
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T18:42:21+0800",
    "State": "Success",
    "Tag": "ExtractDigitalWatermark"
  }]
}
```

视频拆条
提交任务

最近更新时间：2023-08-01 15:45:53

功能描述

提交一个视频拆条任务。

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>SplitVideoParts</Tag>
  <Input>
    <Object>input/test.mp4</Object>
  </Input>
  <Operation>
    <SplitVideoParts>
      <Mode>SHOTDETECT</Mode>
```

```
        </SplitVideoParts>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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：SplitVideoParts	String	是
Input	Request	待操作的对象信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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
SplitVideoParts	Request.Operation	视频拆条配置	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 SplitVideoParts 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Mode	Request.Operation.SplitVideoParts	拆条模式，当前只支持 SHOTDETECT (分镜拆分)	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>SplitVideoParts</Tag>
        <Input>
          <BucketId>test-123456789</BucketId>
          <Object>input/test.mp4</Object>
          <Region>ap-chongqing</Region>
        </Input>
        <Operation>
          <SplitVideoParts>
            <Mode>SHOTDETECT</Mode>
          </SplitVideoParts>
          <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 数组

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.J	新创建任务的 ID	Stri

	obsDetail		ng
Tag	Response.JobsDetail	新创建任务的 Tag：SegmentVideoBody	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
SplitVideoParts	Response.JobsDetail.Operation	同请求中的 Request.Operation.SplitVideoParts	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String

JobLevel	Response.JobsDetail.Operation	任务优先级	String
SplitVideoInfoResult	Response.JobsDetail.Operation	输出拆条后的信息，任务未完成时不返回	Container

Container 节点 SplitVideoInfoResult 的内容：

节点名称（关键字）	父节点	描述	类型
TimeInfo	Response.JobsDetail.Operation.SplitVideoInfoResult	各片段详情	Container 数组

Container 节点 TimeInfo 的内容：

节点名称（关键字）	父节点	描述	类型
Index	Response.JobsDetail.Operation.SplitVideoInfoResult.TimeInfo	片段索引	String
PartBegin	Response.JobsDetail.Operation.SplitVideoInfoResult.TimeInfo	片段起始时间	String
PartEnd	Response.JobsDetail.Operation.SplitVideoInfoResult.TimeInfo	片段结束时间	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 /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SplitVideoParts</Tag>
    <Input>
        <Object>input/test.mp4</Object>
```

```
    </Input>
    <Operation>
      <SplitVideoParts>
        <Mode>SHOTDETECT</Mode>
      </SplitVideoParts>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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>SplitVideoParts</Tag>
      <Input>
        <BucketId>test-123456789</BucketId>
        <Object>input/test.mp4</Object>
        <Region>ap-chongqing</Region>
      </Input>
      <Operation>
        <SplitVideoParts>
          <Mode>SHOTDETECT</Mode>
        </SplitVideoParts>
        <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

回调信息

最近更新时间：2023-08-11 18:10:32

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2023-05-06T16:04:13+0800</CreationTime>
    <EndTime>2023-05-06T16:04:41+0800</EndTime>
    <Input>
      <BucketId>testpic-1253960454</BucketId>
      <Object>test.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j96ca39faebe411eda9b5*********</JobId>
    <Message/>
    <Operation>
      <JobLevel>0</JobLevel>
      <SplitVideoInfoResult>
        <TimeInfo>
          <Index>0</Index>
          <PartBegin>0.000000</PartBegin>
          <PartEnd>5.320000</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>1</Index>
          <PartBegin>5.320000</PartBegin>
          <PartEnd>13.280000</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>2</Index>
          <PartBegin>13.280000</PartBegin>
          <PartEnd>16.799999</PartEnd>
        </TimeInfo>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <TimeInfo>
                <Index>3</Index>
                <PartBegin>16.799999</PartBegin>
                <PartEnd>20.040001</PartEnd>
            </TimeInfo>
            <TimeInfo>
                <Index>4</Index>
                <PartBegin>20.040001</PartBegin>
                <PartEnd>24.440001</PartEnd>
            </TimeInfo>
            <TimeInfo>
                <Index>5</Index>
                <PartBegin>24.440001</PartBegin>
                <PartEnd>104.400002</PartEnd>
            </TimeInfo>
            <TimeInfo>
                <Index>6</Index>
                <PartBegin>104.400002</PartBegin>
                <PartEnd>105.480003</PartEnd>
            </TimeInfo>
            <TimeInfo>
                <Index>7</Index>
                <PartBegin>105.480003</PartBegin>
                <PartEnd>123.120003</PartEnd>
            </TimeInfo>
            <TimeInfo>
                <Index>8</Index>
                <PartBegin>123.120003</PartBegin>
                <PartEnd>134.600006</PartEnd>
            </TimeInfo>
            <TimeInfo>
                <Index>9</Index>
                <PartBegin>134.600006</PartBegin>
                <PartEnd>137.479996</PartEnd>
            </TimeInfo>
        </SplitVideoInfoResult>
        <SplitVideoParts>
            <Mode>SHOTDETECT</Mode>
        </SplitVideoParts>
    </Operation>
    <QueueId>p5f30ea2dad4443a08d53c6e2351ca31e</QueueId>
    <StartTime>2023-05-06T16:04:14+0800</StartTime>
    <State>Success</State>
    <Tag>SplitVideoParts</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	同提交视频拆条接口中的 Response.JobsDetail	Container 数组

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>
    <Code>Success</Code>
    <CreationTime>2023-05-06T16:04:13+0800</CreationTime>
    <EndTime>2023-05-06T16:04:41+0800</EndTime>
    <Input>
      <BucketId>testpic-1253960454</BucketId>
      <Object>test.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j96ca39faebe411eda9b5*********</JobId>
    <Message/>
    <Operation>
      <JobLevel>0</JobLevel>
      <SplitVideoInfoResult>
        <TimeInfo>
          <Index>0</Index>
          <PartBegin>0.000000</PartBegin>
          <PartEnd>5.320000</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>1</Index>
          <PartBegin>5.320000</PartBegin>
          <PartEnd>13.280000</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>2</Index>
          <PartBegin>13.280000</PartBegin>
          <PartEnd>16.799999</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>3</Index>
          <PartBegin>16.799999</PartBegin>
          <PartEnd>20.040001</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>4</Index>
          <PartBegin>20.040001</PartBegin>
          <PartEnd>24.440001</PartEnd>
        </TimeInfo>
        <TimeInfo>
          <Index>5</Index>
```

```
            <PartBegin>24.440001</PartBegin>
            <PartEnd>104.400002</PartEnd>
          </TimeInfo>
          <TimeInfo>
            <Index>6</Index>
            <PartBegin>104.400002</PartBegin>
            <PartEnd>105.480003</PartEnd>
          </TimeInfo>
          <TimeInfo>
            <Index>7</Index>
            <PartBegin>105.480003</PartBegin>
            <PartEnd>123.120003</PartEnd>
          </TimeInfo>
          <TimeInfo>
            <Index>8</Index>
            <PartBegin>123.120003</PartBegin>
            <PartEnd>134.600006</PartEnd>
          </TimeInfo>
          <TimeInfo>
            <Index>9</Index>
            <PartBegin>134.600006</PartBegin>
            <PartEnd>137.479996</PartEnd>
          </TimeInfo>
        </SplitVideoInfoResult>
        <SplitVideoParts>
            <Mode>SHOTDETECT</Mode>
        </SplitVideoParts>
      </Operation>
      <QueueId>p5f30ea2dad4443a08d53c6e2351ca31e</QueueId>
      <StartTime>2023-05-06T16:04:14+0800</StartTime>
      <State>Success</State>
      <Tag>SplitVideoParts</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":"2023-05-08T14:53:35+0800",
```

```
        "EndTime":"2023-05-08T14:54:01+0800",
        "Input":{
            "BucketId":"testpic-1253960454",
            "Object":"test.mp4",
            "Region":"ap-chongqing"
        },
        "JobId":"j0d7759e6ed6d11edb28**********",
        "Message":"",
        "Operation":{
            "JobLevel":"0",
            "SplitVideoInfoResult":{
                "TimeInfo":[
                    {
                        "Index":"0",
                        "PartBegin":"0.000000",
                        "PartEnd":"5.320000"
                    },
                    {
                        "Index":"1",
                        "PartBegin":"5.320000",
                        "PartEnd":"13.280000"
                    },
                    {
                        "Index":"2",
                        "PartBegin":"13.280000",
                        "PartEnd":"16.799999"
                    },
                    {
                        "Index":"3",
                        "PartBegin":"16.799999",
                        "PartEnd":"20.040001"
                    },
                    {
                        "Index":"4",
                        "PartBegin":"20.040001",
                        "PartEnd":"24.440001"
                    },
                    {
                        "Index":"5",
                        "PartBegin":"24.440001",
                        "PartEnd":"104.400002"
                    },
                    {
                        "Index":"6",
                        "PartBegin":"104.400002",
```

```
                    "PartEnd":"105.480003"
                },
                {
                    "Index":"7",
                    "PartBegin":"105.480003",
                    "PartEnd":"123.120003"
                },
                {
                    "Index":"8",
                    "PartBegin":"123.120003",
                    "PartEnd":"134.600006"
                },
                {
                    "Index":"9",
                    "PartBegin":"134.600006",
                    "PartEnd":"137.479996"
                },
            ]
        },
        "SplitVideoParts":{
            "Mode":"SHOTDETECT"
        }
    },
    "QueueId":"p5f30ea2dad4443a08d5********",
    "StartTime":"2023-05-08T14:53:36+0800",
    "State":"Success",
    "Tag":"SplitVideoParts"
  }
 ]
}
```

智能封面
提交任务

最近更新时间：2023-08-11 18:10:33

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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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	智能封面参数, 同创建智能封面模板接口中的 Request.SmartCover	Container	否
Output	Request.Operation	结果输出配置	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

智能封面参数必须设置，可以通过 TemplateId 或 SmartCover 设置， TemplateId 优先级更高。

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，当有多个输出文件时必须包含 ${number} 通配符	Stri	是

			n g	

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id
${number}	产物索引，从0开始计数

响应

响应头

此接口仅返回公共响应头部，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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	String

CreationTime	Response.JobsDetail	任务的创建时间	String
EndTime	Response.JobsDetail	任务的结束时间	String
QueueId	Response.JobsDetail	任务所属的 队列 ID	String
Input	Response.JobsDetail	该任务的输入资源地址	Container
Operation	Response.JobsDetail	该任务的规则	Container

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，详见 MediaResult	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1：使用智能封面模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:test-1234567890.ci.ap-chongqing.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>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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-s
Host:test-1234567890.ci.ap-chongqing.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>
    <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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
```

回调信息

最近更新时间：2023-08-17 17:55:43

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，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
    <EndTime>2022-06-30T15:29:48+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format/>
        <Stream>
          <Audio/>
          <Subtitle/>
          <Video/>
        </Stream>
      </MediaInfo>
      <MediaResult>
        <OutputFile>
          <Bucket>test-123456789</Bucket>
          <Md5Info/>
          <ObjectName>output/smartcover-0.jpg</ObjectName>
          <ObjectName>output/smartcover-1.jpg</ObjectName>
          <ObjectName>output/smartcover-2.jpg</ObjectName>
          <ObjectName>output/smartcover-3.jpg</ObjectName>
          <ObjectName>output/smartcover-4.jpg</ObjectName>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <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>
        <JobLevel>0</JobLevel>
      </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	同提交智能封面任务接口中的 Response.JobsDetail	Container 数组

如果任务是通过工作流触发的，Response.JobsDetail.Input 还会包含 CosHeaders 节点，类型为 Container 数组。

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.JobsDetail.Input.CosHeaders	自定义 Header 的名称	String
Value	Response.JobsDetail.Input.CosHeaders	自定义 Header 的值	String

如果任务是通过工作流触发的，Response.JobsDetail 还会包含 Workflow 节点，类型为 Container。
Container 节点 Workflow 的内容：

节点名称（关键字）	父节点	描述	类型
RunId	Response.Workflow	工作流实例 ID	String
WorkflowId	Response.Workflow	工作流 ID	String
WorkflowName	Response.Workflow	工作流名称	String
Name	Response.Workflow	工作流节点名称	String

实际案例

案例 1：通过任务接口触发的任务回调

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T15:29:42+0800</CreationTime>
    <EndTime>2022-06-30T15:29:48+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j6842765cf84611ecb8546d80f2baf56f</JobId>
    <Message/>
    <Operation>
      <MediaInfo>
        <Format/>
        <Stream>
          <Audio/>
          <Subtitle/>
          <Video/>
        </Stream>
```

```
        </MediaInfo>
        <MediaResult>
          <OutputFile>
            <Bucket>test-123456789</Bucket>
            <Md5Info/>
            <ObjectName>output/smartcover-0.jpg</ObjectName>
            <ObjectName>output/smartcover-1.jpg</ObjectName>
            <ObjectName>output/smartcover-2.jpg</ObjectName>
            <ObjectName>output/smartcover-3.jpg</ObjectName>
            <ObjectName>output/smartcover-4.jpg</ObjectName>
            <ObjectPrefix/>
            <Region>ap-chongqing</Region>
          </OutputFile>
        </MediaResult>
        <Output>
          <Bucket>test-123456789</Bucket>
          <Object>output/smartcover-${number}.jpg</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <SmartCover>
          <Count>5</Count>
          <DeleteDuplicates>false</DeleteDuplicates>
          <Format>jpg</Format>
          <Height>960</Height>
          <Width>1280</Width>
        </SmartCover>
        <UserData>This is my SmartCover job.</UserData>
        <JobLevel>0</JobLevel>
      </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>
        <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
      <StartTime>2022-06-30T15:29:43+0800</StartTime>
      <State>Success</State>
      <Tag>SmartCover</Tag>
      <Workflow>
        <Name>SmartCover_1581665960537</Name>
        <RunId>ic90edd59f84f11ec9d4f525400a3c59f</RunId>
        <WorkflowId>web6ac56c1ef54dbfa44d7f4103203be9</WorkflowId>
        <WorkflowName>workflow-test</WorkflowName>
      </Workflow>
    </JobsDetail>
</Response>
```

案例 3：通过工作流触发的任务回调，格式为 JSON

```
{
  "EventName": "TaskFinish",
  "JobsDetail": [{
    "Code": "Success",
    "CreationTime": "2022-06-30T15:29:42+0800",
    "EndTime": "2022-06-30T15:29:48+0800",
    "Input": {
      "BucketId": "test-123456789",
      "Object": "input/demo.mp4",
      "Region": "ap-chongqing",
      "CosHeaders": [{
            "Key": "Content-Type",
            "Value": "video/mp4"
        },
```

```
            {
                "Key": "x-cos-request-id",
                "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
            },
            {
                "Key": "EventName",
                "Value": "cos:ObjectCreated:Put"
            },
            {
                "Key": "Size",
                "Value": "1424687"
            }
        ]
    },
    "JobId": "j6842765cf84611ecb8546d80f2baf56f",
    "Operation": {
        "MediaInfo": {
            "Stream": {

            }
        },
        "MediaResult": {
            "OutputFile": {
                "Bucket": "test-123456789",
                "ObjectName": [
                    "output/smartcover-0.jpg",
                    "output/smartcover-1.jpg",
                    "output/smartcover-2.jpg",
                    "output/smartcover-3.jpg",
                    "output/smartcover-4.jpg"
                ],
                "Region": "ap-chongqing"
            }
        },
        "Output": {
            "Bucket": "test-123456789",
            "Object": "output/smartcover-${number}.jpg",
            "Region": "ap-chongqing"
        },
        "SmartCover": {
            "Count": "5",
            "DeleteDuplicates": "false",
            "Format": "jpg",
            "Height": "960",
            "Width": "1280"
```

```
        },
        "UserData": "This is my SmartCover job.",
        "JobLevel": 0
      },
      "QueueId": "p2242ab62c7c94486915508540933a2c6",
      "StartTime": "2022-06-30T15:29:43+0800",
      "State": "Success",
      "Tag": "SmartCover",
      "Workflow": {
        "Name": "SmartCover_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
      }
    }]
}
```

创建模板

最近更新时间：2023-08-01 15:45:53

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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 *，长度不超过 64	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	默认值	是否必选
Format	Request.SmartCover	图片格式，支持 jpg、png 、webp	String	jpg	否
Width	Request.SmartCover	宽 • 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	String	视频原始宽度	否
Height	Request.SmartCover	高 • 值范围：[128，4096] • 单位：px • 若只设置 Height 时，按照视频原始比	String	视频原始	否

		例计算 Width		高度	
Count	Request.SmartCover	截图数量，取值范围 [1,10]	String	3	否
DeleteDuplicates	Request.SmartCover	封面去重，取值 true/false	String	false	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>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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
RequestId	无	请求的唯一 ID	Container

Container 节点 Template 的内容如下

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，同请求体 Request.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-
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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
    <RequestId>NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=</RequestId>
    <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>
```

更新模板

最近更新时间：2023-08-11 17:56:15

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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-
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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <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>
```

视频合成
提交任务

最近更新时间：2023-08-18 15:28:01

功能描述

提交一个视频合成任务。

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>VideoSynthesis</Tag>
  <Operation>
    <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplat
    <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemp
    <VideoSynthesis>
      <KeepAudioTrack>false</KeepAudioTrack>
      <SpliceInfo>
```

```
            <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/car.mp4<
            <Width>640</Width>
        </SpliceInfo>
        <SpliceInfo>
            <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/game.mp
            <X>700</X>
            <Width>640</Width>
            <Height>960</Height>
        </SpliceInfo>
    </VideoSynthesis>
    <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-123456789</Bucket>
        <Object>output/out.${ext}</Object>
    </Output>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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：VideoSynthesis	String	是
Input	Request	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackT	Req	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于	Strin	否

ype	uest	队列的回调类型	g	
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	文件路径	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
VideoSynthesis	Request.Operation	视频合成模板参数	Container	是
TranscodeTemplateId	Request.Operation	转码模板 ID	String	否
Transcode	Request.Operation	转码模板参数	Container	否
WatermarkTemplateId	Request.Operation	水印模板 ID，可以传多个水印模板 ID，最多传3个	String 数组	否
Watermark	Request.Operation	水印模板参数，同创建水印模板接口中的 Request.Watermark，最多传3个	Container 数组	否
Output	Request.Operation	结果输出配置	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：

- 转码参数必须设置，可以通过 TranscodeTemplateId 或 Transcode 设置，TranscodeTemplateId 优先级更高。

- 水印参数可选，可以通过 WatermarkTemplateId 或 Watermark 设置，WatermarkTemplateId 优先级更高。

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

节点名称（关键字）	父节点	描述	类型	是否必选
KeepAudioTrack	Request.Operation.VideoSynthesis	是否保留源视频音轨，取值 true/false，默认值 false	String	否
SpliceInfo	Request.Operation.VideoSynthesis	待合成的视频	Container 数组	是

Container 类型 SpliceInfo 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Url	Request.Operation.VideoSynthesis.SpliceInfo	同 bucket 的视频文件地址	String	是
X	Request.Operation.VideoSynthesis.SpliceInfo	该视频在画面中的横坐标，默认值 0	String	否
Y	Request.Operation.VideoSynthesis.SpliceInfo	该视频在画面中的纵坐标，默认值 0	String	否
Width	Request.Operation.VideoSynthesis.SpliceInfo	该视频在画面中的宽，默认原视频宽	String	否
Height	Request.Operation.VideoSynthesis.SpliceInfo	该视频在画面中的高,默认原视频高	String	否
Name	Request.Operation.VideoSynthesis.SpliceI	该视频流的自定义名称, 长度不超过64个字符，仅支持中文、英文、数字、	Stri	否

	nfo	_、-和*	n g	

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
TimeInterval	Request.Operation.Transcode	同创建转码模板接口中的 Request.TimeInterval	Container	否
Container	Request.Operation.Transcode	同创建转码模板接口中的 Request.Container	Container	否
Video	Request.Operation.Transcode	同创建转码模板接口中的 Request.Video	Container	否
Audio	Request.Operation.Transcode	同创建转码模板接口中的 Request.Audio	Container	否
TransConfig	Request.Operation.Transcode	同创建转码模板接口中的 Request.TransConfig	Container	否
AudioMix	Request.Operation.Transcode	混音参数，详情见 AudioMix	Container	否
AudioMixArray	Request.Operation.Transcode	混音参数数组，最多同时传2个	Container 数组	否

说明：

- 不传 Video，相当于删除视频信息。不传 Audio，相当于删除音频信息。Video 和 Audio 不可以同时不传。
- AudioMix 和 AudioMixArray 在 Audio.Remove 为 false 时生效。
- AudioMixArray 是一个数组，每一个子项内容同 AudioMix 。AudioMix 优先级更高，设置 AudioMix 时，AudioMixArray 无效。

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.Out	存储结果的存储桶	Strin	是

	put		g	
Object	Request.Operation.Output	输出结果的文件名	String	是

Request.Operation.Output.Object 支持的通配符如下：

通配符	含义
${ext}	封装格式
${jobid}	任务Id

响应

响应头

此接口仅返回公共响应头部，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>VideoSynthesis</Tag>
    <Operation>
      <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemp
      <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTer
      <VideoSynthesis>
        <KeepAudioTrack>false</KeepAudioTrack>
        <SpliceInfo>
          <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/car.mp
          <Width>640</Width>
        </SpliceInfo>
        <SpliceInfo>
          <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/game.m
          <X>700</X>
```

```
            <Width>640</Width>
            <Height>960</Height>
          </SpliceInfo>
        </VideoSynthesis>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-123456789</Bucket>
          <Object>output/out.${ext}</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 数组

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：VideoSynthesis	String
State	Response.J	任务状态	Stri

	obsDetail	• Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	ng
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

版权所有：腾讯云计算（北京）有限责任公司

VideoSynthesis	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoSynthesis	Container
TranscodeTemplateId	Response.JobsDetail.Operation	同请求中的 Request.Operation.TranscodeTemplateId	String
Transcode	Response.JobsDetail.Operation	同请求中的 Request.Operation.Transcode	Container
Watermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.Watermark	Container 数组
WatermarkTemplateId	Response.JobsDetail.Operation	同请求中的 Request.Operation.WatermarkTemplateId	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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	Container
DigitalWatermark	Response.JobsDetail.Operation	同请求中的 Request.Operation.DigitalWatermark	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=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
```

```
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoSynthesis</Tag>
    <Operation>
        <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemplat
        <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTemp
        <VideoSynthesis>
            <KeepAudioTrack>false</KeepAudioTrack>
            <SpliceInfo>
                <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/car.mp4<
                <Width>640</Width>
            </SpliceInfo>
            <SpliceInfo>
                <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/game.mp
                <X>700</X>
                <Width>640</Width>
                <Height>960</Height>
            </SpliceInfo>
        </VideoSynthesis>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.${ext}</Object>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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****
```

版权所有：腾讯云计算（北京）有限责任公司

```
<Response>
  <JobsDetail>
    <Code>Success</Code>
    <Message/>
    <JobId>je8f65004eb8511eaaed4f377124a303c</JobId>
    <State>Submitted</State>
    <Progress>0</Progress>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <StartTime>-</StartTime>
    <EndTime>-</EndTime>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <Tag>VideoSynthesis</Tag>
    <Operation>
      <TranscodeTemplateId>t160606b9752148c4absdfaf2f55163b1f</TranscodeTemp
      <WatermarkTemplateId>t146d70eb241c44c63b6efc1cc93ccfc5d</WatermarkTer
      <VideoSynthesis>
        <KeepAudioTrack>false</KeepAudioTrack>
        <SpliceInfo>
          <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/car.mp
          <Width>640</Width>
        </SpliceInfo>
        <SpliceInfo>
          <Url>https://test-123456789.cos.ap-chongqing.myqcloud.com/input/game.m
          <X>700</X>
          <Width>640</Width>
          <Height>960</Height>
        </SpliceInfo>
      </VideoSynthesis>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-123456789</Bucket>
        <Object>output/out.${ext}</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

内容审核
图片审核
图片单次审核

最近更新时间：2023-06-27 10:41:42

功能描述

图片审核功能支持同步、异步请求方式，您可以通过本接口对图片文件进行内容审核。该接口属于 GET 请求。
该接口支持情况如下：

> **注意**
> - 支持审核腾讯云 COS 上的图片文件。
> - 支持审核第三方云存储厂商的图片 URL。
> - 支持审核本地图片文件，需要先进行 base64编码。审核本地图片文件请调用 图片批量审核接口 。

- 支持对 COS 上的图片文件或链接进行检测，通过深度学习技术，识别可能令人反感、不安全或不适宜的违规图片内容。
- 支持对 GIF 图进行截帧审核。
- 支持识别多种违规场景，包括：低俗、违法违规、色情、广告等场景。
- 支持多种物体检测（实体、广告台标、二维码等）及图片中文本的 OCR 文本识别。
- 支持根据不同的业务场景 配置自定义的审核策略 。
- 支持用户通过配置自定义风险库，打击自定义的违规内容。

费用说明

- 每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1张**图片，**计2次**审核费用。
- 调用接口会产生 图片审核费用 和 COS 读请求费用 。
- 如果图片为 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用 。
- 不支持审核 COS 上的归档存储类型和深度归档存储类型的文件，如果需要审核此类型文件，请先 恢复归档文件 。
- 审核第三方云存储厂商的图片 URL 时，将产生所在云厂商的下行流量费用。

限制说明

- 图片文件大小支持：支持审核32MB以下的图片。对于大小超过5MB的图片，您需要在调用请求时，使用 large-image-detect 参数。

版权所有：腾讯云计算（北京）有限责任公司

- 图片文件分辨率支持：建议**分辨率大于256x256**，否则可能会影响识别效果。
- 图片文件支持格式：PNG、JPG/JPEG、BMP、GIF、WEBP 、HEIF/HEIC格式。
- 接口默认并发数量：100。
- 图片文件链接支持的传输协议：HTTP、HTTPS。
- 调用接口需携带签名，具体规则请参见 请求签名 文档。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档 。

请求

请求示例

```
GET /<ObjectKey>?ci-process=sensitive-content-recognition&detect-url=<detect-url>&ii
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求参数

具体参数如下：

参数名称	描述	类型	是否必选
ObjectKey	COS 存储桶中的图片文件名称，COS 存储桶由 Host 指定，例如在北京的 examplebucket-1250000000存储桶中的目录 test 下的文件 img.jpg，则 Host 填写 examplebucket-1250000000.cos.ap-beijing.myqcloud.com，ObjectKey 填写 test/img.jpg。	String	否

ci-process	标识数据处理功能的字段，内容审核的值为：sensitive-content-recognition。	String	是
biz-type	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如涉黄、广告、违法违规等，配置指引： 设置公共审核策略 。您可以在控制台上获取到 biz-type。biz-type 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。biz-type 不填写时，将自动使用默认的审核策略。	String	否
detect-url	您可以通过填写 `detect-url` 审核任意公网可访问的图片链接不填写 `detect-url` 时，后台会默认审核 `ObjectKey` 填写了 `detect-url` 时，后台会审核 `detect-url` 链接，无需再填写 `ObjectKey` ，`detect-url` 示例：http://www.example.com/abc.jpg。	String	否
interval	审核 GIF 动图时，可使用该参数进行截帧配置，代表截帧的间隔。例如值设为5，则表示从第1帧开始截取，每隔5帧截取一帧，默认值5。	Int	否
max-frames	针对 GIF 动图审核的最大截帧数量，需大于0。例如值设为5，则表示最大截取5帧，默认值为5。	Int	否
large-image-detect	对于超过大小限制的图片是否进行压缩后再审核，取值为： 0（不压缩），1（压缩）。默认为0。注：压缩最大支持32MB的图片，且会收取图片压缩费用。对于 GIF 等动态图过大时，压缩时间较长，可能会导致审核超时失败。	Int	否
dataid	图片标识，该字段在结果中返回原始内容，长度限制为512字节。	String	否
async	是否异步进行审核，取值 0：同步返回结果，1：异步进行审核，默认为0。	Int	否
callback	审核结果（Detail版本）以回调形式发送至您的回调地址，异步审核时生效，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。	String	否

注意

- 通过 ObjectKey 进行审核为内网操作，不会产生额外的外网流量。
- 通过 detect-url 进行审核，会产生图片所在源站对应的外网流量。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

```
<RecognitionResult>
    <JobId>xxxxxxxxxxxxxxx</JobId>
    <Result>1</Result>
    <Label>Porn</Label>
    <SubLabel>SexBehavior</SubLabel>
    <Score>90</Score>
    <PornInfo>
      <Code>0</Code>
      <Msg>OK</Msg>
      <HitFlag>1</HitFlag>
      <Label>xxx</Label>
      <SubLabel>SexBehavior</SubLabel>
      <Score>100</Score>
      <OcrResults>
        <Text></Text>
        <Keywords></Keywords>
        <Location>
          <X></X>
          <Y></Y>
          <Width></Width>
          <Height></Height>
          <Rotate></Rotate>
        </Location>
      </OcrResults>
    </PornInfo>
</RecognitionResult>
```

响应包体具体数据内容如下：

参数名称	类型	描述
RecognitionResult	Container	图片审核结果。

异步审核返回的RecognitionResult 节点内容：

参数名称	类型	描述
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。

JobId	String	图片审核任务的 ID。
State	String	审核任务的状态，值为Submitted（已提交审核）。
Object	String	存储在 COS 桶中的图片名称，创建任务使用 ObjectKey 时返回。
Url	String	图片文件的链接地址，创建任务使用 detect-url 时返回。

同步审核返回的RecognitionResult 节点内容：

参数名称	类型	描述
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	String	图片审核任务的 ID。
State	String	审核任务的状态，值为 Success（审核成功）。
Object	String	存储在 COS 桶中的图片名称，创建任务使用 ObjectKey 时返回。
Url	String	图片文件的链接地址，创建任务使用 detect-url 时返回。
CompressionResult	Int	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。
Result	Int	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。
Label	String	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal** 表示正常，**Porn** 表示色情，**Ads** 表示广告，以及其他不安全或不适宜的类型。
Category	String	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。
SubLabel	String	该图命中的二级标签结果。
Score	Int	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。
Text	String	该图里的文字内容（OCR），当审核策略开启文本内容检测时返回。
PornInfo	Container	审核场景为**涉黄**的审核结果信息。

AdsInfo	Container	审核场景为**广告引导**的审核结果信息。

审核信息（PornInfo、AdsInfo）中包含如下内容：

参数名称	类型	描述
Code	Int	错误码，0为正确，其他数字对应相应错误。详情请参见 错误码 。
Msg	String	具体错误信息，如正常则为 OK。
HitFlag	Int	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。
Score	Int	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。其中0 - 60分表示图片正常，61 - 90分表示图片疑似敏感，91 - 100分表示图片确定敏感。例如：色情 99，表明该内容非常有可能属于色情内容。
Label	String	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。
Category	String	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。
SubLabel	String	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。
OcrResults	Container Array	该字段表示 OCR 文本识别的详细检测结果，包括文本识别结果、命中的关键词等信息，有相关违规内容时返回。
LibResults	Container Array	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。

Container 节点 LibResults 的内容：

参数名称	类型	描述
Imageld	String	该字段表示命中的风险库中的图片样本 ID。
Score	Integer	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有

		可能命中库中的色情样本。

Container 节点 OcrResults 的内容：

参数名称	类型	描述
Text	String	图片 OCR 文本识别出的具体文本内容。
Keywords	String Array	在当前审核场景下命中的关键词。
Location	Container	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。

Container 节点 Location 的内容：

名称	类型	描述
X	Float	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Y	Float	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Width	Float	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Height	Float	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Rotate	Float	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。

Container 节点 ObjectResults 的内容：

名称	类型	描述
Name	String	该标签用于返回所识别出的实体名称，例如人名。
Location	Container	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。

版权所有：腾讯云计算（北京）有限责任公司

实际案例

请求1：同步图片审核

```
GET /picture.jpg?ci-process=sensitive-content-recognition&interval=0&max-frames=1&b
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX
```

响应1

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

<RecognitionResult>
    <JobId>xxxxxxxxxxxxxxx</JobId>
    <CompressionResult>0</CompressionResult>
    <Result>1</Result>
    <Label>Porn</Label>
    <SubLabel>SexBehavior</SubLabel>
    <Score>95</Score>
    <PornInfo>
      <Code>0</Code>
      <Msg>OK</Msg>
      <HitFlag>1</HitFlag>
      <Label>xxx</Label>
      <SubLabel>SexBehavior</SubLabel>
      <Score>95</Score>
    </PornInfo>
</RecognitionResult>
```

请求2：异步图片审核

```
GET /picture.jpg?ci-process=sensitive-content-recognition&biz-type=***&async=1&callba
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX
```

响应2

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

<RecognitionResult>
    <JobId>xxxxxxxxxxxxxxx</JobId>
</RecognitionResult>
```

图片批量审核

最近更新时间：2023-07-10 15:44:31

功能描述

图片批量审核接口支持同步、异步请求方式，您可以通过本接口对多个图片文件进行内容审核。该接口属于 POST 请求。

> **说明**
> - 支持审核腾讯云 COS 上的图片文件。
> - 支持审核第三方云存储厂商的图片 URL。
> - 支持审核加密数据，如有需要，请 联系服务人员 咨询。
> - 支持根据不同的业务场景 配置自定义的审核策略 。

限制说明

- 图片文件大小支持：支持审核32MB以下的图片。对于大小超过5MB的图片，您需要在调用请求时，使用 large-image-detect 参数。
- 单次请求图片数量不可超过**100**张。
- 图片文件分辨率支持：建议**分辨率大于256x256**，否则可能会影响识别效果。
- 图片文件支持格式：PNG、JPG/JPEG、BMP、GIF、WEBP、HEIF/HEIC 格式。
- 接口默认并发数量：100。
- 图片文件链接支持的传输协议：HTTP、HTTPS。
- 调用接口需携带签名，具体规则请参见 请求签名 文档。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档 。

请求

请求示例

```
POST /image/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
```

```
Content-Type: application/xml

<body>
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。

```
<Request>
  <Input>
   <DataId></DataId>
   <Url></Url>
   <Object></Object>
   <Content></Content>
   <Interval></Interval>
   <MaxFrames></MaxFrames>
   <LargeImageDetect></LargeImageDetect>
   <UserInfo>
    <TokenId></TokenId>
   </UserInfo>
  </Input>
  <Input>
   <DataId></DataId>
   <Url></Url>
   <Object></Object>
   <Content></Content>
   <Interval></Interval>
   <MaxFrames></MaxFrames>
  </Input>
  <Conf>
   <BizType>b81d45f94b91a683255e9a9506f45a11</BizType>
   <Async></Async>
   <Callback></Callback>
   <Freeze>
    <PornScore></PornScore>
   </Freeze>
```

```
    </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	图片批量审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容，如有多个图片，请传入多个 Input 结构。	Container Array	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下，使用其中一种：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的图片文件名称，例如在目录 test 中的文件 image.jpg，则文件名称为 test/image.jpg。Content，Object 和 Url 只能选择其中一种，传入多个时仅一个生效，按 Content，Object， Url 顺序。	String	否
Url	Request.Input	图片文件的链接地址，例如 http://a-1250000.cos.ap-shanghai.myqcloud.com/image.jpg。Content，Object 和 Url 只能选择其中一种，传入多个时仅一个生效，按 Content，Object， Url 顺序。	String	否
Content	Request.Input	图片文件的内容，需要先经过 base64 编码。Content，Object 和 Url 只能选择其中一种，传入多个时仅一个生效，按 Content，Object， Url 顺序。	String	否
Interva	Request.I	截帧频率，GIF 图检测专用，默认值为5，表示从第一	Int	否

I	nput	帧（包含）开始每隔5帧截取一帧。		
MaxFrames	Request.Input	最大截帧数量，GIF 图检测专用，默认值为5，表示只截取 GIF 的5帧图片进行审核，必须大于0。	Int	否
DataId	Request.Input	图片标识，该字段在结果中返回原始内容，长度限制为512字节。	String	否
LargeImageDetect	Request.Input	对于超过大小限制的图片是否进行压缩后再审核，取值为：0（不压缩），1（压缩）。默认为0。注：压缩最大支持32M的图片，且会收取压缩费用。	Int	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称	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

（关键字）				
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如涉黄、广告、违法违规等，配置指引： 设置公共审核策略 。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Async	Request.Conf	是否异步进行审核，0：同步返回结果，1：异步进行审核。默认值为 0。	Integer	否
Callback	Request.Conf	审核结果（Detail版本）以回调形式发送至您的回调地址，异步审核时生效，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。	String	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对图片进行自动冻结，仅当 `input` 中审核的图片为 `object` 时有效。	Container	否

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请联系服务人员 咨询。

注意

- 通过 Object 进行审核为内网操作，不会产生额外的外网流量。
- 通过 Url 进行审核，会产生图片所在源站对应的外网流量。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <JobsDetail>
    <DataId></DataId>
    <Url></Url>
    <PornInfo>
      <HitFlag>1</HitFlag>
      <Score>100</Score>
      <Category>xxx</Category>
      <SubLabel>xxx</SubLabel>
    </PornInfo>
  </JobsDetail>
  <JobsDetail>
    <Code>InvalidArgument</Code>
    <Message>Param MaxFrames is illegal</Message>
    <DataId></DataId>
  </JobsDetail>
</Response>
```

响应包体具体数据内容如下：

节点名称（关键字）	父节点	描述
Response	Container	图片批量审核返回的具体响应内容。

Container 节点 Response 的内容：

参数名称	类型	描述
JobsDetail	Container Array	图片审核的结果
RequestId	String	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。

异步审核结果的 JobsDetail 节点内容如下：

节点名称（关键字）	类型	描述
Code	String	错误码，只有失败时返回。详情请查看 错误码列表 。
Message	String	错误描述，只有失败时返回。
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	String	本次审核任务的 ID。
State	String	审核任务的状态，值为 Submitted（已提交审核）、Failed（审核失败）其中一个。
Object	String	存储在 COS 存储桶中的图片文件名称，创建任务使用 Object 时返回。
Url	String	图片文件的链接地址，创建任务使用Url时返回。

同步审核结果的 JobsDetail 节点内容如下：

参数名称	类型	描述
Code	String	错误码，只有失败时返回。详情请查看 错误码列表 。
Message	String	错误描述，只有失败时返回。
DataId	String	图片标识，审核结果会返回原始内容，长度限制为512字节。
JobId	Str	本次审核任务的 ID。

	ing	
State	String	审核任务的状态，值为 Success（审核成功）、Failed（审核失败）其中一个。
Object	String	存储在 COS 存储桶中的图片文件名称，创建任务使用Object时返回。
Url	String	图片文件的链接地址，创建任务使用Url时返回。
CompressionResult	Integer	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。
Label	String	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告等。
Result	Integer	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。
Score	Integer	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。

Category	String	该图命中的审核类别结果。
SubLabel	String	该图命中的二级标签结果。
Text	String	该图里的文字内容（OCR），当审核策略开启文本内容检测时返回。
PornInfo	Container	审核场景为**涉黄**的审核结果信息。
AdsInfo	Container	审核场景为**广告引导**的审核结果信息。
UserInfo	Container	请求中设置的 UserInfo 原样返回。
ListInfo	Contai	账号黑白名单结果。

	ner	
ForbidState	Integer	若您在请求时设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结。

审核信息（PornInfo、AdsInfo等）中的内容如下：

参数名称	类型	描述
Code	Integer	单个审核场景的错误码，0为成功，其他为失败。详情请参见 错误码 。
Msg	String	具体错误信息，如正常则为 OK。
HitFlag	Integer	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。
Score	Integer	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。
Label	String	该图的结果标签（为综合标签，可能为 SubLabel，可能为人物名字等）。
Category	String	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。
SubL	String	该图的二级标签结果。

abel		
OcrResults	ContainerArray	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。
LibResults	ContainerArray	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。

Container 节点 LibResults 的内容：

节点名称（关键字）	类型	描述
ImageId	String	该字段表示命中的风险库中的图片样本 ID。
Score	Integer	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，则表明该数据非常有可能命中库中的色情样本。

OcrResults 的内容如下：

节点名称（关键字）	类型	描述
Text	String	有敏感信息的 Ocr 文本内容。
Keywords	String Array	该段内容中的敏感文字。
Location	Container	该段文字在图片中的位置信息。

版权所有：腾讯云计算（北京）有限责任公司

ObjectResults 的内容如下：

节点名称（关键字）	类型	描述
Name	String	敏感内容的字称。
Location	Container	该内容在图片中的位置信息。

Location 的内容如下：

节点名称（关键字）	类型	描述
X	Float	相对于原点（图片左上角）的 X 坐标。
Y	Float	相对于原点（图片左上角）的 Y 坐标。
Height	Float	敏感区域的高度。
Width	Float	敏感区域的宽度。
Rotate	Float	敏感区域的旋转角度。

UserInfo 的内容（与请求中的UserInfo一致）：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	类型	描述
ListResults	Container Array	命中的所有名单结果。

Container 节点 ListResults 的内容：

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

实际案例

请求

```
POST /image/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-s
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Input>
    <Object>a.jpg</Object>
  </Input>
  <Input>
    <Url>http://exsample.com/b.jpg</Url>
  </Input>
```

```
  <Conf>
    <BizType>b81d45f94xxxxxxxxa9506f45a11</BizType>
  </Conf>
</Request>
```

响应

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

<Response>
    <JobsDetail>
        <JobId>xxxxxxxxxx</JobId>
        <CompressionResult>0</CompressionResult>
        <Object>a.jpg</Object>
        <Label>Porn</Label>
        <Result>1</Result>
        <Score>100</Score>
        <SubLabel>xxx</SubLabel>
        <PornInfo>
            <Code>0</Code>
            <Msg>OK</Msg>
            <HitFlag>1</HitFlag>
            <Score>100</Score>
            <Label>xxx</Label>
            <SubLabel>xxx</SubLabel>
        </PornInfo>
    </JobsDetail>
    <JobsDetail>
        <JobId>yyyyyyyyyy</JobId>
        <CompressionResult>0</CompressionResult>
        <Url>http://exsample.com/b.jpg</Url>
        <Label>Porn</Label>
        <Result>1</Result>
        <Score>100</Score>
        <SubLabel>xxx</SubLabel>
        <PornInfo>
            <Code>0</Code>
            <Msg>OK</Msg>
            <HitFlag>1</HitFlag>
```

```
        <Score>100</Score>
        <Label>xxx</Label>
        <SubLabel>xxx</SubLabel>
      </PornInfo>
    </JobsDetail>
    <RequestId>xxxxxxxxxxx</RequestId>
</Response>
```

查询图片审核任务结果

最近更新时间：2023-05-06 18:06:29

功能描述

本接口用于主动查询指定的图片审核任务结果。

> **说明**
> 图片审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

数据万象 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

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

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数名称	描述	类型	是否必选
jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <Label></Label>
    <Result>1</Result>
    <PornInfo>
      <HitFlag>1</HitFlag>
      <Label>xxx</Label>
      <SubLabel>SexBehavior</SubLabel>
      <Score>100</Score>
      <OcrResults>
        <Text></Text>
        <Keywords></Keywords>
        <Location>
          <X></X>
          <Y></Y>
          <Width></Width>
          <Height></Height>
          <Rotate></Rotate>
        </Location>
      </OcrResults>
    </PornInfo>
    <AdsInfo>
      <HitFlag>1</HitFlag>
      <Label>xxx</Label>
      <SubLabel>SexBehavior</SubLabel>
      <Score>100</Score>
      <OcrResults>
        <Text></Text>
        <Keywords></Keywords>
        <Location>
          <X></X>
          <Y></Y>
```

```
            <Width></Width>
            <Height></Height>
            <Rotate></Rotate>
          </Location>
        </OcrResults>
      </AdsInfo>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

参数名称	类型	描述
Response	Container	审核任务结果。

Container 节点 Response 的内容：

参数名称	类型	描述
JobsDetail	Container	任务的详细信息。
RequestId	String	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。

Container 节点 JobsDetail 的内容：

参数名称	类型	描述
Code	String	错误码，只有 State 为 Failed 时返回。详情请参见 错误码列表 。
Message	String	错误描述，只有 State 为 Failed 时返回。
DataId	String	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。

JobId	String	本次审核任务的 ID
State	String	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。
CreationTime	String	审核任务的创建时间。
Object	String	本次审核的文件名称，创建任务使用 Object 时返回。
Url	String	本次审核的文件链接，创建任务使用 Url 时返回。
CompressionResult	Integeger	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。
Text	String	该图里的文字内容（OCR），当审核策略开启文本内容检测时返回。
Label	String	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告等。
Result	Inte	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）。

	ger	
Category	String	该图命中的审核类别结果。
SubLabel	String	该图命中的二级标签结果。
PornInfo	Container	审核场景为**涉黄**的审核结果信息。
AdsInfo	Container	审核场景为**广告引导**的审核结果信息。
UserInfo	Container	用户业务字段。创建任务时未设置 UserInfo 则无此字段。
ListInfo	Contain	账号黑白名单结果。

	er	
ForbidState	Integer	若您在请求时设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。

JobsDetail中Container节点PornInfo，AdsInfo的内容：

参数名称	类型	描述
HitFlag	Integer	用于返回该审核场景的审核结果，返回值：0：正常1：确认为当前场景的违规内容2：疑似为当前场景的违规内容。
Score	Integer	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。其中0 - 60分表示图片正常，61 - 90分表示图片疑似敏感，91 - 100分表示图片确定敏感。例如：色情 99，表明该内容非常有可能属于色情内容。
Label	String	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。
Category	String	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。
Su	Stri	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。

bLabel	ng	
OcrResults	ContainerArray	该字段表示 OCR 文本识别的详细检测结果，包括文本识别结果、命中的关键词等信息，有相关违规内容时返回。
LibResults	ContainerArray	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。

Container 节点 LibResults 的内容：

参数名称	类型	描述
Imageld	String	该字段表示命中的风险库中的图片样本 ID。
Sco	Inte	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。

re	ger	

Container 节点 OcrResults 的内容：

参数名称	类型	描述
Text	String	图片 OCR 文本识别出的具体文本内容。
Keywords	String Array	在当前审核场景下命中的关键词。
Location	Container	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。

Container 节点 Location 的内容：

名称	类型	描述
X	Float	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Y	Float	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。
Width	Float	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Height	Float	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。
Rotate	Float	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。

Container 节点 ObjectResults 的内容：

名称	类型	描述
Name	String	该标签用于返回所识别出的实体名称，例如人名。
Location	Container	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 ListInfo 的内容：

节点名称（关键字）	类型	描述
ListResults	Container Array	命中的所有名单结果。

Container 节点 ListResults 的内容：

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

实际案例

请求

```
GET /image/auditing/iab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<Response>
    <JobsDetail>
        <JobId>iab1ca9fc8a3ed11ea834c5254008639004</JobId>
        <State>Success</State>
        <CreationTime>2021-05-21T10:44:32+08:00</CreationTime>
        <Object>example.jpg</Object>
        <Label>Ads</Label>
        <Result>1</Result>
        <AdsInfo>
```

```
        <HitFlag>1</HitFlag>
        <Label>xxx</Label>
        <SubLabel>SexBehavior</SubLabel>
        <Score>100</Score>
        <OcrResults>
          <Text></Text>
          <Keywords></Keywords>
          <Location>
            <X></X>
            <Y></Y>
            <Width></Width>
            <Height></Height>
            <Rotate></Rotate>
          </Location>
        </OcrResults>
      </AdsInfo>
    </JobsDetail>
    <RequestId></RequestId>
</Response>
```

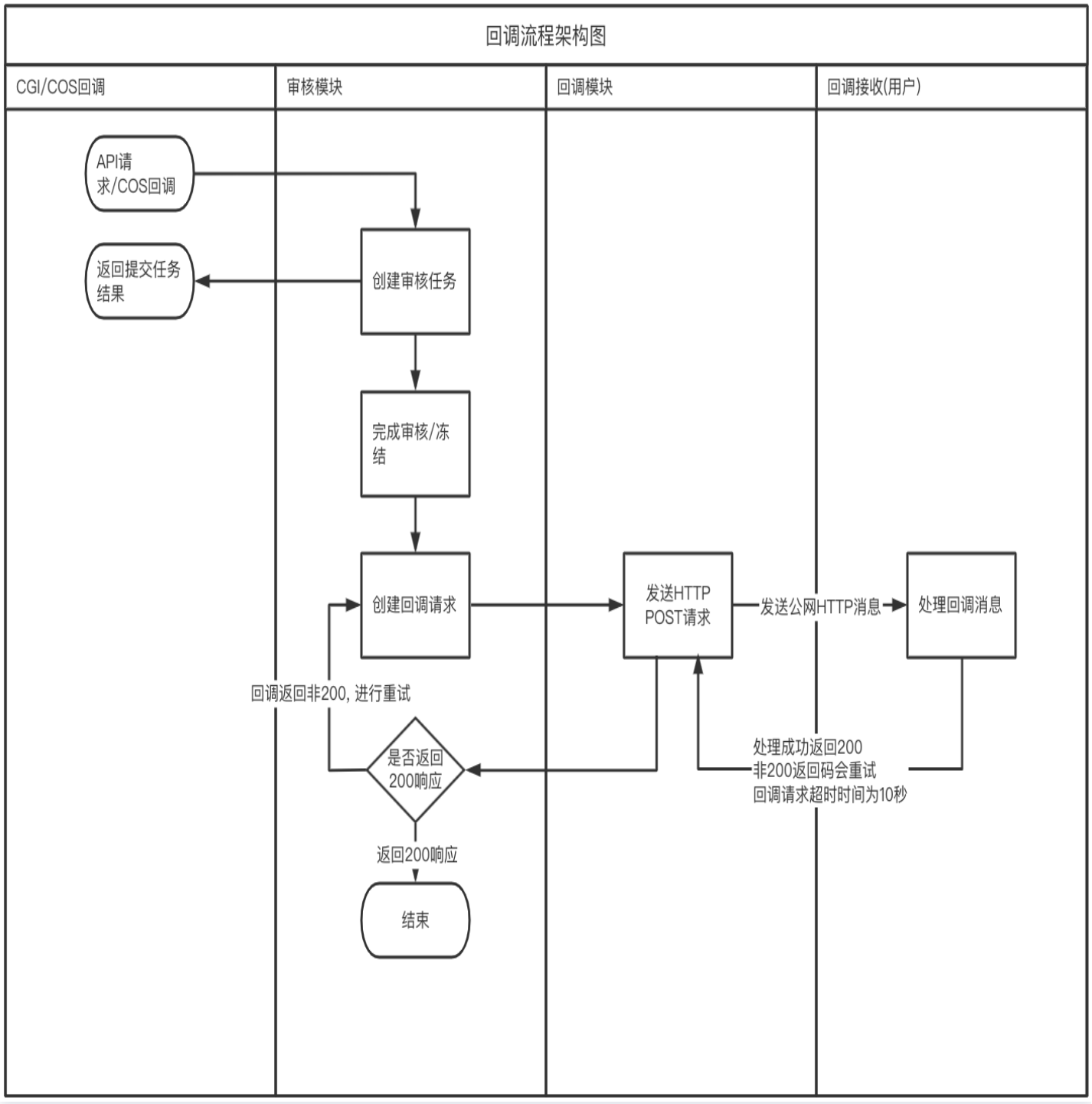
图片审核回调内容

最近更新时间：2022-08-03 14:55:19

功能描述

若您配置了审核回调地址，在图片审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。

版权所有：腾讯云计算（北京）有限责任公司



回调内容说明

简洁回调（Simple）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Simple` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：

```
{
  "code": 0,
  "data": {
    "forbidden_status": 0,
    "event": "ReviewImage",
    "porn_info": {
      "hit_flag": 0,
      "label": "",
      "score": 9
    },
    "result": 0,
    "trace_id": "test_trace_id",
    "url": "https://examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test.jpg",
    "cos_headers": {
      "x-cos-meta-xx": "xx"
    }
  },
  "message": "Test request when setting callback url"
}
```


具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码 。	Integer	是
message	错误描述。	String	是
data	图片审核结果的详细信息。	Object	是

data 的具体内容如下：

参数名称	描述	类型	是否必选
trace_id	唯一 ID 标识。	String	是

版权所有：腾讯云计算（北京）有限责任公司

url	被审核图片的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewImage。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integeger	是
forbidden_status	若您设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integeger	是
cos_headers	资源上传时设置的自定义 header 内容，若未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否

xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer	是
score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer	是
label	命中的标签名称。	String	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：

```
{
 "EventName": "ReviewImage",
 "JobsDetail": {
  "JobId": "xxxx",
  "State": "Success",
  "CreationTime": "",
  "Object": "",
  "Url": "https://examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test.jpg",
  "Label": "Normal",
  "Result": 0,
  "Score": 0,
  "Category": "",
  "SubLabel": "",
  "Text": "",
```

```
    "PornInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": "",
      "Category": "",
      "SubLabel": ""
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": "",
      "Category": "",
      "SubLabel": ""
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
        "x-cos-meta-id": "xxxx"
    }
  }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息	Object
EventName	任务的类型，为 ReviewImage	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	Str

		ing
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
State	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	审核任务的创建时间。	String
CompressionResult	图片是否被压缩处理，值为 0（未压缩），1（正常压缩）。	Integer
Object	被审核的图片文件名称。创建任务使用 Object 时返回。	String
Url	被审核的图片文件链接。创建任务使用 Url 时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String

Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Category	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签	String
Text	图片 OCR 文本识别出的具体文本内容，当审核策略开启 OCR 内容检测时返回。	String
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
BucketId	审核操作所属的存储桶名称。	String

Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer
Label	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String

Category	该字段为 Label 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本识别结果、命中的关键词等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	ContainerArray

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String

Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，例如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯	Fl

版权所有：腾讯云计算（北京）有限责任公司

	一确定检测框的大小和位置。	oat
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0 – 360（角度制）**，方向为**逆时针旋转**。	Float

实际案例

案例一：简洁回调（Simple）

```
{
  "code": 0,
  "data": {
     "event":"ReviewImage",
     "forbidden_status": 0,
     "porn_info": {
        "hit_flag": 0,
        "label": "",
        "score": 9
     },
     "result": 0,
     "trace_id": "ixzt90jl2dfscxxxxxxxxxxxxxxxxx",
     "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/test.jpg",
     "cos_headers": {
        "x-cos-meta-id": "666666"
     }
  },
  "message": "success"
}
```

案例二：详细回调（Detail）

```
{
  "EventName": "ReviewImage",
  "JobsDetail": {
    "JobId": "xxxx",
    "State": "Success",
    "CreationTime": "2021-08-10T21:01:10+08:00",
    "CompressionResult": 0,
    "Object": "1.jpg",
    "Label": "Normal",
    "Result": 0,
    "Score": 0,
    "Category": "",
    "SubLabel": "",
    "Text": "",
    "PornInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": "",
      "Category": "",
      "SubLabel": ""
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": "",
      "Category": "",
      "SubLabel": ""
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
        "x-cos-meta-id": "xxxx"
    }
  }
}
```


版权所有：腾讯云计算（北京）有限责任公司

图片审核结果反馈

最近更新时间：2022-07-06 17:56:18

功能描述

您可通过本接口反馈与预期不符的审核结果，例如色情图片被审核判定为正常或正常图片被判定为色情时可通过该接口直接反馈。
本接口不会直接修改审核结果，您反馈的错误审核结果将在后台进行确认，并在后续的审核任务中生效。

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 /report/badcase HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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>
  <ContentType></ContentType>
  <JobId></JobId>
```

```
    <ModerationTime></ModerationTime>
    <Url></Url>
    <SuggestedLabel></SuggestedLabel>
    <Label></Label>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	反馈审核错误的具体参数	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
ContentType	Request	需要反馈的数据类型，反馈图片错误样本，取值为2。	Integer	是
Url	Request	图片类型的样本，需要填写图片的 url 链接，ContentType 为2时必填。	String	否
Label	Request	数据万象审核判定的审核结果标签，例如 Porn。	String	是
SuggestedLabel	Request	您自己期望的正确审核结果的标签，例如期望是正常，则填 Normal。	String	是
JobId	Request	该数据样本对应的审核任务 ID，有助于定位审核记录。	String	否
ModerationTime	Request	该数据样本之前审核的时间，有助于定位审核记录。格式为 2021-08-07T12:12:12+08:00	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
```

```
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	返回的具体响应内容	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
POST /report/badcase HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1*******&q-s
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 66
Content-Type: application/xml
<Request>
    <ContentType>1</ContentType>
    <Url>http://www.example.com/abc.jpg</Url>
    <Label>Porn</Label>
    <SuggestedLabel>Normal</SuggestedLabel>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 May 2022 12:37:29 GMT
```

```
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

视频审核
提交视频审核任务

最近更新时间：2023-08-14 17:01:44

功能描述

本接口用于提交一个视频审核任务。视频审核功能为异步任务方式，您可以通过提交视频审核任务审核您的视频文件，然后通过 查询视频审核任务接口 或 视频审核回调 查询审核结果。
该接口支持情况如下：

> **说明**
> - 支持审核腾讯云 COS 上的视频文件。
> - 支持审核第三方云存储厂商的视频 URL。
> - 支持审核加密数据，如有需要，请 联系服务人员 咨询。

- 支持对视频文件进行自动检测，从 OCR 文本识别、物体检测（实体、广告台标、二维码等）、图像识别及音频审核四个维度，通过深度学习技术，识别视频中的违规内容。
- 支持设置回调地址 Callback 获取检测结果，或通过 查询视频审核任务结果接口 主动轮询获取审核结果详情。
- 支持识别多种违规场景，包括：色情、违法、广告等场景。
- 支持根据不同的业务场景 配置自定义的审核策略 。
- 支持用户 自定义配置黑白图片库 ，打击自定义违规内容。

费用说明

视频的审核分为**视频画面审核**、**视频截帧**、**视频声音审核**，其中：
- 视频画面审核：基于视频截帧能力，通过视频截帧将视频截取为多张图片进行审核，审核费用与图片审核一致。
- 视频截帧：会产生相应的 视频截帧费用 。
- 视频声音审核：将视频声音分离出来进行音频审核，审核费用与音频审核一致。
- 每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1个视频**，**计2次**审核费用。
- 调用接口会产生 图片审核费用 、音频审核费用 和 COS 读请求费用 。
- 如果音频文件属于 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用 。
- 不支持审核 COS 上的归档存储类型和深度归档存储类型的音频，如果需要审核此类型音频，请先 恢复归档文件 。

限制说明

- 视频文件大小支持：**文件 < 5GB**。

版权所有：腾讯云计算（北京）有限责任公司

- 视频文件支持格式：flv、mkv、mp4 、rmvb、avi、wmv、3gp、mov、m3u8、m4v 等。
- 接口默认并发数量：10。
- 支持用户配置审核视频画面或视频声音。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

```
POST /video/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。

```
<Request>
 <Input>
  <Object></Object>
  <DataId></DataId>
 </Input>
 <Conf>
  <Snapshot>
    <Mode>Interval</Mode>
    <TimeInterval></TimeInterval>
```

```
      <Count></Count>
    </Snapshot>
    <Callback></Callback>
    <BizType></BizType>
    <DetectContent></DetectContent>
  </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	视频审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的视频。	Container	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	当前 COS 存储桶中的视频文件名称，例如在目录 test 中的文件 video.mp4，则文件名称为 test/video.mp4。	String	否
Url	Request.Input	视频文件的链接地址，例如 http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/test.mp4。Object 和 Url 只能选择其中一种。	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	用户业务 TokenId，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	用户业务 Nickname，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	用户业务 DeviceId，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	用户业务 AppId，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	用户业务 Room，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	用户业务 IP，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	用户业务 Type，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	用户业务 ReceiveTokenId，长度不超过128字节。	String	否
Gender	Request.Input.UserInfo	用户业务 Gender，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	用户业务 Level，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	用户业务 Role，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如色情、广告、违法违规等，配置指引： 设置公共审核策略 。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策	String	否

		略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。		
Snapshot	Request.Conf	视频画面的审核通过视频截帧能力截取出一定量的截图，通过对截图逐一审核而实现的，该参数用于指定视频截帧的配置。	Container	是
Callback	Request.Conf	回调地址，以 http:// 或者 https:// 开头的地址。	String	否
Callback Version	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	String	否
DetectContent	Request.Conf	用于指定是否审核视频声音，当值为0时：表示只审核视频画面截图；值为1时：表示同时审核视频画面截图和视频声音。默认值为0。	Integer	否
Callback Type	Request.Conf	回调片段类型，有效值：1（回调全部截帧和音频片段）、2（回调违规截帧和音频片段）。默认为 1。	Integer	否
Freeze	Request.Conf	自动冻结配置项，可配置指定审核分数的结果进行自动冻结。	Container	否

Container 类型 Snapshot 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Mode	Request.Conf.Snapshot	表示视频的截帧模式，有效值：Interval、Average、Fps。 • Interval 表示间隔模式：每间隔 TimeInterval 秒截取一帧；当设置 Count，未设置 TimeInterval 时，表示截取所有帧，共 Count 帧。 • Average 表示平均模式：选择该模式时 TimeInterval 参数不生效，需要结合 Count 参数使用，表示整个视频，按平均间隔截取共 Count 张图片。 • Fps 表示固定帧率模式：选择该模式时，TimeInterval 参数表示每秒截取多少帧，未设置 TimeInterval 时，表示截取所有帧，Count 表示共截取多少帧。	String	否
Count	Request.Conf	视频截帧数量，范围为(0, 10000]。	Int	是

版权所有：腾讯云计算（北京）有限责任公司

	.Snapshot		eger	
TimeInterval	Request.Conf.Snapshot	视频截帧频率，范围为(0, 60]，单位为秒，支持 float 格式，执行精度精确到毫秒。	Float	否

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请 联系服务人员 咨询。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <JobsDetail>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	视频审核返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	视频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

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

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

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

```
<Request>
  <Input>
    <Object>a.mp4</Object>
    <DataId>123-fdrsg-123</DataID>
  </Input>
  <Conf>
    <Snapshot>
       <Mode>Interval</Mode>
       <TimeInterval>50</TimeInterval>
       <Count>100</Count>
    </Snapshot>
    <Callback>http://callback.com/</Callback>
    <BizType>b81d45f94b91a683255e9a9506f45a11</BizType>
    <DetectContent>1</DetectContent>
  </Conf>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <DataId>123-fdrsg-123</DataID>
    <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Submitted</State>
    <CreationTime>2021-08-07T12:12:12+0800</CreationTime>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

查询视频审核任务结果

最近更新时间：2023-06-21 10:22:01

功能描述

本接口用于主动查询指定的视频审核任务结果。视频审核功能为异步任务方式，您可以通过提交视频审核任务审核您的视频文件，然后通过查询视频审核任务接口查询审核结果。

> **说明**
> 视频审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

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

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数名称	描述	类型	是否必选

jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>Success</Message>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <SnapshotCount></SnapshotCount>
    <Result>1</Result>
    <PornInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </PornInfo>
    <AdsInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </AdsInfo>
    <Snapshot>
      <Url></Url>
      <Text></Text>
      <SnapshotTime></SnapshotTime>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Label></Label>
        <SubLabel></SubLabel>
      </PornInfo>
      <AdsInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Label></Label>
        <SubLabel></SubLabel>
```

```
      </AdsInfo>
    </Snapshot>
    <AudioSection>
      <Url></Url>
       <OffsetTime>0</OffsetTime>
       <Duration>3000</Duration>
      <Text>哈哈哈</Text>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </PornInfo>
      <AdsInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </AdsInfo>
    </AudioSection>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	视频审核的全部结果信息。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	视频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关	父节点	描述	类型

键字）			
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	视频审核任务的 ID。	String
State	Response.JobsDetail	视频审核任务的状态，值为 Submitted（已提交审核）、Snapshoting（视频截帧中）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	视频审核任务的创建时间。	String
Object	Response.JobsDetail	本次审核的文件名称，创建任务使用 Object 时返回。	String
Url	Response.JobsDetail	本次审核的文件链接，创建任务使用 Url 时返回。	String
SnapshotCount	Response.JobsDetail	视频截图的总数量。	String
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，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
PornInfo	Response.JobsDetail	审核场景为**色情**的审核结果信息。	Container

AdsInfo	Response.JobsDetail	审核场景为**广告**的审核结果信息。	Container
Snapshot	Response.JobsDetail	该字段用于返回视频中视频画面截图审核的结果。	Container Array
AudioSection	Response.JobsDetail	该字段用于返回视频中视频声音审核的结果。如果未检测音频，则不返回。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。创建任务未设置 UserInfo 时无此字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container
ForbidState	Response.JobsDetail	若您设置了自动冻结，该字段表示图片的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Count	Response.JobsDetail.*Info	命中该审核分类的截图张数。	Integer

Container 节点 Snapshot 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.Snapshot	视频截图的访问地址，您可以通过该地址查看该截图内容，地址格式为标准 URL 格式。注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Sna	Response.JobsD	该字段用于返回当前截图位于视频中的时间，单位为毫秒。	Integ

pshotTime	etail.Snapshot	例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前截图的时间戳，单位为毫秒。例如1649387157000。	er
Text	Response.JobsDetail.Snapshot	该字段用于返回当前截图的图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	Response.JobsDetail.AudioSection	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
SubLabel	Response.JobsDetail.AudioSection	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Result	Response.JobsDetail.AudioSection	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.Snapshot	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.Snapshot	审核场景为**广告**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Snapshot.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.Snapshot.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回的审核信息。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Label	Response.JobsDet	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String

	ail.Snapshot.*Info		
Category	Response.JobsDetail.Snapshot.*Info	该字段为 Label 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	Response.JobsDetail.Snapshot.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	Response.JobsDetail.Snapshot.*Info	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Container Array
LibResults	Response.JobsDetail.Snapshot.*Info	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
ImageId	Response.JobsDetail.Snapshot.*Info.LibResults	该字段表示命中的风险库中的图片样本 ID。	String
Score	Response.JobsDetail.Snapshot.*Info.LibResults	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）–100（置信度最高），越高代表当前的图片越有可能命中库中的样本；如：色情 99，则表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关	父节点	描述	类型

版权所有：腾讯云计算（北京）有限责任公司

键字）			
Text	Response.JobsDetail.Snapshot.*Info.OcrResults	图片 OCR 文本识别出的具体文本内容。	String
Keywords	Response.JobsDetail.Snapshot.*Info.OcrResults	在当前审核场景下命中的关键词。	String Array
Location	Response.JobsDetail.Snapshot.*Info.OcrResults	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Container

ObjectResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Snapshot.*Info.ObjectResults	该标签用于返回所识别出的实体名称，例如人名。	String
Location	Response.JobsDetail.Snapshot.*Info.ObjectResults	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Container

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 AudioSection 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.AudioSection	视频声音片段的访问地址，您可以通过该地址获取该声音片段的内容，地址格式为标准 URL 格式。注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	Response.JobsDetail.AudioSection	该字段用于返回当前视频声音的 ASR 文本识别的检测结果。	String
OffsetTime	Response.JobsDetail.AudioSection	该字段用于返回当前声音片段位于视频中的时间，单位为毫秒，例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前声音片段的时间戳，单位为毫秒，例如1649387157000。	Integer
Duration	Response.JobsDetail.AudioSection	当前视频声音片段的时长，单位毫秒。	Integer
Label	Response.JobsDetail.Snapshot	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail.Snapshot	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.AudioSection	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.AudioSection	审核场景为**广告**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关	父节点	描述	类型

键字）			
HitFlag	Response.JobsDetail.AudioSection.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	String
Score	Response.JobsDetail.AudioSection.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。	String
Category	Response.JobsDetail.AudioSection.*Info	该字段为 Label 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
SubLabel	Response.JobsDetail.AudioSection.*Info	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Keywords	Response.JobsDetail.AudioSection.*Info	在当前审核场景下命中的关键词，没有时不返回	String Array
LibResults	Response.JobsDetail.AudioSection.*Info	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
LibType	Response.JobsDetail.AudioSection.*Info.LibResults	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	Response.JobsDetail.AudioSection.*Info.LibResults	命中的风险库名称。	String
Keywords	Response.JobsDetail.AudioSection.*Info.LibResults	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

Container 节点 UserInfo 的内容：

节点名称（关键	描述	类型	是否必

字）			选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ListResults	Response.JobsDetail.ListInfo	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

节点名称（关键字）	父节点	描述	类型

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

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /video/auditing/vab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>vab1ca9fc8a3ed11ea834c5254008639 04</JobId>
    <State>Success</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Object>a.mp4</Object>
    <SnapshotCount>1</SnapshotCount>
    <Label>Normal</Label>
    <Result>0</Result>
    <PornInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
```

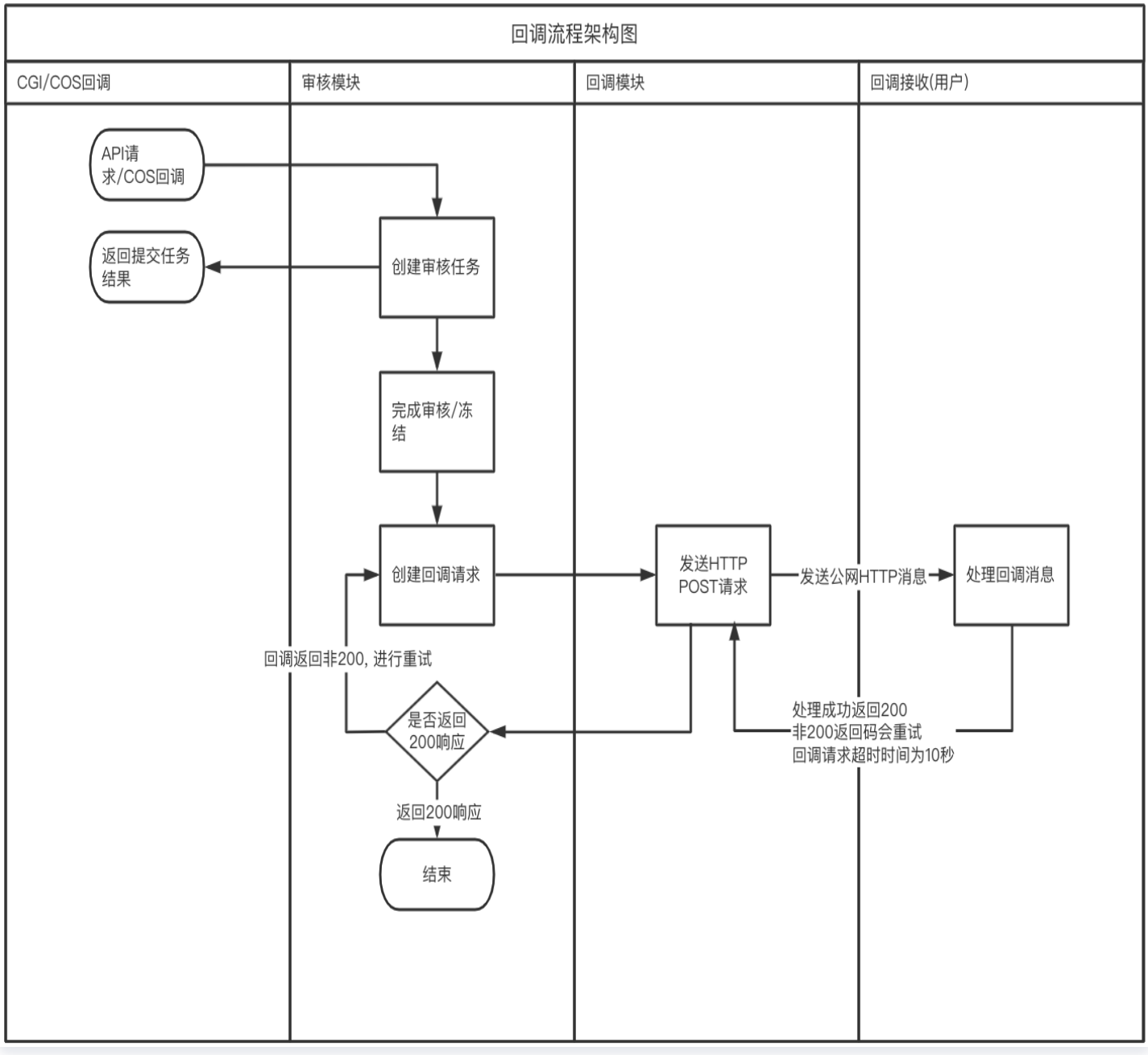
```
    </PornInfo>
    <AdsInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </AdsInfo>
    <Snapshot>
      <Url>http://examplebucket-1250000000.cos.ap-chongqing.myqcloud.com/0.jpg</U
      <SnapshotTime>0</SnapshotTime>
      <Text>哈哈</Text>
      <Label>Normal</Label>
      <Result>0</Result>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Label></Label>
        <SubLabel></SubLabel>
      </PornInfo>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Label></Label>
        <SubLabel></SubLabel>
      </AdsInfo>
    </Snapshot>
    <AudioSection>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
      </AdsInfo>
      <Duration>30000</Duration>
      <OffsetTime>0</OffsetTime>
      <Label>Normal</Label>
      <Result>0</Result>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
      </PornInfo>
      <Text>wc xx</Text>
      <Url>https://auditing-1250000000.cos.ap-guangzhou.myqcloud.com/q9MhdztF/0.m
    </AudioSection>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

视频审核回调内容

最近更新时间：2022-11-15 14:37:00

功能描述

若您配置了审核回调地址，在视频审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。



回调内容说明

简洁回调（Simple）

回调通知为 HTTP 的 POST 方法，含有 X-Ci-Content-Version: Simple 的头部。
该响应体返回为 application/json 数据，包含完整节点数据的内容展示如下：

```
{
  "code": 0,
  "data": {
     "event":"ReviewVideo",
     "forbidden_status": 0,
     "porn_info": {
        "hit_flag": 0,
        "label": "",
        "count": 0
     },
     "result": 0,
     "trace_id": "test_trace_id",
     "url": "test_url",
     "cos_headers": {
        "x-cos-meta-xx": "xx"
     }
  },
  "message": "Test request when setting callback url"
}
```

具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码 。	Integer	是
message	错误描述。	String	是
data	视频审核结果的详细信息。	Object	是

data 的具体内容如下：

参数名称	描述	类型	是否必选

trace_id	请求的 ID，如果是接口创建的异步任务，则为接口返回的 JobId。	String	是
url	被审核视频的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewVideo。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示视频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer	是
cos_headers	资源上传时设置的自定义 header 内容，若未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否
data_id	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String	否

xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer	是
label	命中的标签名称，目前为空。	String	是
count	命中该审核分类的敏感截图张数。	Integer	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：

```
{
 "EventName": "ReviewVideo",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "",
  "Object": "",
  "Label": "Normal",
  "Result": 0,
  "SnapshotCount": 1,
  "PornInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "AdsInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "Snapshot": [
   {
    "Url": "",
    "Text": "",
    "SnapshotTime": 0,
    "Label": "Normal",
    "Result": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Label": "",
     "SubLabel": ""
    },
    "AdsInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Label": "",
     "SubLabel": ""
    }
   }
  ],
  "AudioSection": [
   {
    "Url": "",
    "Text": "",
    "OffsetTime": 0,
    "Duration": 30000,
```

```
      "Label": "Normal",
      "Result": 0,
      "PornInfo": {
        "HitFlag": 0,
        "Score": 0
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0
      }
    }
  ],
  "BucketId": "examplebucket-1250000000",
  "Region": "ap-chongqing",
  "ForbidState": 0,
  "CosHeaders": {
    "x-cos-meta-id": "xxx"
  }
 }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object
EventName	任务的类型，为 ReviewVideo 。	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
DataId	提交任务时设置了DataId参数时返回，返回原始内容，长度限制为512字节。	String

State	视频审核任务的状态，值为 Submitted（已提交审核）、Snapshoting（截图中）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	视频审核任务的创建时间。	String
Object	被审核的视频文件的名称，创建任务使用 Object 时返回。	String
Url	本次审核的文件链接，创建任务使用 Url 时返回。	String
SnapshotCount	视频截图的总数量。直播流状态为 Auditing 的回调中该字段为本次回调包含的截图数量。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
Snapshot	该字段用于返回视频中视频画面截图审核的结果。	Array
AudioSection	该字段用于返回视频中视频声音审核的结果。如果未检测音频，则不返回。	Array
BucketId	创建的审核任务所属的存储桶名称。	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示视频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object

版权所有：腾讯云计算（北京）有限责任公司

UserInfo	用户业务字段。创建任务时未设置 UserInfo 则无此字段。	Object
Type	审核类型，直播流审核返回 live_video。	String
ListInfo	账号黑白名单结果。	Object

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Count	命中该审核分类的截图张数。	Integer

Snapshot 的具体内容如下：

节点名称（关键字）	描述	类型
Url	视频截图的访问地址，您可以通过该地址查看该截图内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
Snapshot Time	该字段用于返回当前截图位于视频中的时间，单位为毫秒。例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前截图的时间戳，单位为毫秒。例如1649387157000。	Integer
Text	该字段用于返回当前截图的图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object

AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Snapshot 中 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高** ），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Label	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Object Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高 ），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称	描述	类型

（关键字）		
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

AudioSection 的具体内容如下：

节点名称（关	描述	类型

键字）		
Url	视频声音片段的访问地址，您可以通过该地址获取该声音片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
Text	该字段用于返回当前视频声音的 ASR 文本识别的检测结果。	String
OffsetTime	该字段用于返回当前声音片段位于视频中的时间，单位为毫秒，例如5000（视频开始后5000毫秒）。审核直播流时，该字段返回当前声音片段的时间戳，单位为毫秒，例如1649387157000。	Integer
Duration	当前视频声音片段的时长，单位毫秒。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	供参考的识别结果，0（正常），1（敏感），2（疑似敏感）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

AudioSection 中 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情99，表明该内容非常有可能属于色情内容。	Integer
Category	该字段表示命中的审核类别。 注意：该字段可能返回空。	String
Keywords	在当前审核场景下命中的关键词，没有时不返回。	Array

LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
LibType	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	命中的风险库名称。	String
Keywords	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	Strin	否

版权所有：腾讯云计算（北京）有限责任公司

		g	
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

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

实际案例

案例一：简洁回调（Simple）

```
{
  "code": 0,
  "data": {
    "event":"ReviewVideo",
    "forbidden_status": 0,
    "porn_info": {
      "hit_flag": 0,
      "label": "",
      "count": 0
    },
    "result": 0,
    "trace_id": "vxzt90jl2dfscxxxxxxxxxxxxxxxxx",
    "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/video.mp
    "cos_headers": {
      "x-cos-meta-id": "666666"
    }
  },
  "message": "success"
}
```

案例二：详细回调（Detail）


```
{
 "EventName": "ReviewVideo",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "2021-08-10T21:01:10+08:00",
  "Object": "1.mp4",
  "Label": "Normal",
  "Result": 0,
  "SnapshotCount": 1,
  "PornInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "AdsInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "Snapshot": [
   {
    "Url": "https://video-1250000000.cos.ap-chongqing.myqcloud.com/test/0.jpg",
    "Text": "",
    "SnapshotTime": 41,
    "Label": "Normal",
    "Result": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Label": "",
     "SubLabel": ""
    },
    "AdsInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Label": "",
     "SubLabel": ""
    }
   }
  ],
  "AudioSection": [
   {
    "Url": "https://audio-1250000000.cos.ap-guangzhou.myqcloud.com/0.mp3",
    "Text": "",
```

```
        "OffsetTime": 0,
        "Duration": 30000,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      }
    ],
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "xxxx"
    }
  }
}
```

音频审核
提交音频审核任务

最近更新时间：2023-06-27 10:41:43

功能描述

本接口用于提交一个音频审核任务。音频审核功能为异步任务方式，您可以通过提交音频审核任务审核您的音频文件，然后通过 查询音频审核任务接口 或 音频审核回调 查询审核结果。
该接口支持情况如下：

> **说明**
> - 支持审核腾讯云 COS 上的音频文件。
> - 支持审核第三方云存储厂商的音频 URL。

- 支持对音频文件进行自动检测，从 ASR 语音文本识别、色情呻吟两个维度，通过深度学习技术识别音频中的违规内容。
- 支持设置回调地址 Callback 获取检测结果，或通过 查询音频审核任务结果接口 主动轮询获取审核结果详情。
- 支持识别多种违规场景，包括：色情、广告等场景。
- 支持根据不同的业务场景 配置自定义的审核策略 。

费用说明

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

限制说明

- 音频文件大小支持：**文件 < 600MB**。
- 音频文件时长支持：**< 3小时**。
- 音频码率类型支持：128Kbps - 256Kbps。
- 音频文件支持格式：mp3、wav、aac、flac、amr、3gp、m4a、wma、ogg、ape。
- 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。

版权所有：腾讯云计算（北京）有限责任公司

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

```
POST /audio/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
<body>
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体：

```
<Request>
  <Input>
    <Object></Object>
    <Url></Url>
    <DataId></DataId>
  </Input>
  <Conf>
    <Callback></Callback>
    <BizType></BizType>
  </Conf>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	音频审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容。	Container	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的音频文件名称，例如在目录 test 中的文件 audio.mp3，则文件名称为 test/audio.mp3。Object 和 Url 只能选择其中一种。	String	否
Url	Request.Input	音频文件的链接地址，例如 http://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/audio.mp3。Object 和 Url 只能选择其中一种。	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	用户业务 TokenId，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	用户业务 Nickname，长度不超过128字节。	String	否

DeviceId	Request.Input.UserInfo	用户业务 DeviceId，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	用户业务 AppId，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	用户业务 Room，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	用户业务 IP，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	用户业务 Type，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	用户业务 ReceiveTokenId，长度不超过128字节。	String	否
Gender	Request.Input.UserInfo	用户业务 Gender，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	用户业务 Level，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	用户业务 Role，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如色情、广告等，配置指引： 设置公共审核策略 。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Callback	Request.Conf	审核结果以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。	String	否
CallbackVersion	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	string	否

CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部音频片段）、2（回调违规音频片段）。默认为 1。	Integer	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对音频文件进行自动冻结，仅当 input 中审核的音频为 object 时有效。	Container	否

Container 类型 Freeze 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
PornScore	Request.Conf.Freeze	取值为[0,100]，表示当色情审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请 联系服务人员 咨询。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <JobsDetail>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	音频审核返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	音频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

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

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
POST /audio/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>
```

查询音频审核任务结果

最近更新时间：2023-06-20 15:10:36

功能描述

本接口用于查询一个音频审核任务的状态或结果。

> **说明**
> 音频审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

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

> **说明**
> - Authorization: Auth String（详情请参见 请求签名 文档）。
> - 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

参数名称	描述	类型	是否必选

jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>Success</Message>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <Result>1</Result>
    <AudioText></AudioText>
    <PornInfo>
      <HitFlag></HitFlag>
      <Score></Score>
      <Label></Label>
    </PornInfo>
    <Section>
      <Url>https://cos.ap-guangzhou.myqcloud.com/0.mp3</Url>
        <OffsetTime>0</OffsetTime>
        <Duration>3000</Duration>
      <Text>哈哈哈</Text>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </PornInfo>
    </Section>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	音频审核任务结果。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	音频审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	本次音频审核任务的 ID。	String
State	Response.JobsDetail	音频审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	音频审核任务的创建时间。	String
Object	Response.JobsDetail	审核的音频文件为存储在 COS 中的文件时，该字段表示本次审核的音频文件名称。	String
Url	Response.JobsDetail	审核的音频文件为一条文件链接时，该字段表示本次审核的音频文件链接。	String

Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
AudioText	Response.JobsDetail	该字段用于返回音频文件中已识别的对应文本内容。	String
PornInfo	Response.JobsDetail	审核场景为**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
Section	Response.JobsDetail	当音频过长时，会对音频进行分段，该字段用于返回音频片段的审核结果，主要包括开始时间和音频审核的相应结果。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。创建任务未设置 UserInfo 时无此字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container
ForbidState	Response.JobsDetail	若您设置了自动冻结，该字段表示音频文件的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情99，表明该内容非常有可能属于色情内容。	Integer

Label	Response.JobsDetail.*Info	本次审核的综合结果标签，如果命中了敏感的关键词，该字段返回对应的关键词。	String
Category	Response.JobsDetail.*Info	该字段表示审核命中的具体审核类别。注意：该字段可能返回空。	String
SubLabel	Response.JobsDetail.*Info	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String

Container 节点 Section 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.Section	当前音频片段的访问地址，您可以通过该地址获取该音频片段的内容，地址格式为标准 URL 格式。注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	Response.JobsDetail.Section	该字段用于返回当前音频片段的 ASR 文本识别的检测结果。	String
OffsetTime	Response.JobsDetail.Section	该字段用于返回当前音频片段位于总音频中的时间，单位为毫秒，例如5000（音频开始后5000毫秒）。	Integer
Duration	Response.JobsDetail.Section	当前音频片段的时长，单位毫秒。	Integer
Label	Response.JobsDetail.Section	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
SubLabel	Response.JobsDetail	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Result	Response.JobsDetail.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

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Section.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.Section.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer
Category	Response.JobsDetail.Section.*Info	该字段为 `Label` 的子集，表示审核命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。	String
Keywords	Response.JobsDetail.Section.*Info	本次审核命中的敏感关键词，没有时不返回。	String Array
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

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（预设风险库）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

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ListResults	Response.JobsDetail.ListInfo	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

版权所有：腾讯云计算（北京）有限责任公司

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

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /audio/auditing/vab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>aab1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Success</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Object>a.mp3</Object>
    <Result>1</Result>
    <AudioText>wc xx</AudioText>
    <PornInfo>
      <HitFlag>1</HitFlag>
```

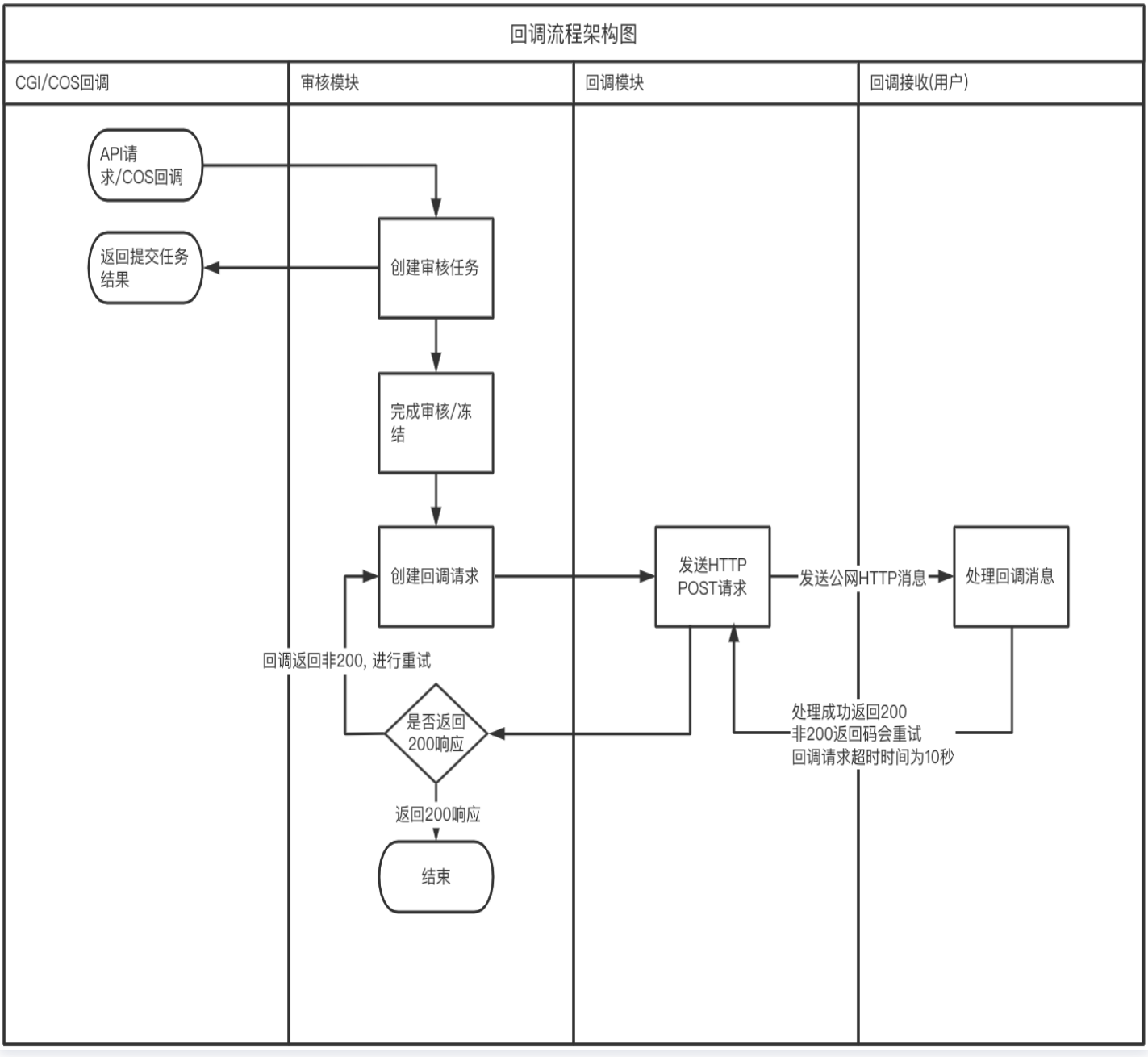
```
      <Score>91</Score>
      <Label>xx</Label>
    </PornInfo>
    <AdsInfo>
      <HitFlag>0</HitFlag>
      <Score>0</Score>
      <Label></Label>
    </AdsInfo>
    <Section>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
      </AdsInfo>
      <Duration>30000</Duration>
      <OffsetTime>0</OffsetTime>
      <PornInfo>
        <HitFlag>1</HitFlag>
        <Score>91</Score>
        <Keywords>xx</Keywords>
        <Keywords>wc</Keywords>
      </PornInfo>
      <Text>wc xx</Text>
      <Url>https://auditing-1250000000.cos.ap-guangzhou.myqcloud.com/q9MhdztF/0.m
    </Section>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

音频审核回调内容

最近更新时间：2022-11-15 14:39:09

功能描述

若您配置了审核回调地址，在音频审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。



回调内容说明

简洁回调（Simple）


```
{
  "code": 0,
  "data": {
    "forbidden_status": 0,
    "porn_info": {
      "hit_flag": 0,
      "label": "",
      "score": 9
    },
    "result": 0,
    "trace_id": "test_trace_id",
    "url": "test_audio",
    "cos_headers": {
      "x-cos-meta-xx": "xx"
    }
  },
  "message": "Test request when setting callback url"
}
```


具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码 。	Integer	是
message	错误描述。	String	是
data	音频审核结果的详细信息。	Object	是

data的具体内容如下：

参数名称	描述	类型	是否必选
trace_id	请求的 ID，如果是接口创建的异步任务，则为接口返回的 JobId。	String	是

url	被审核音频的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewAudio。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示音频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer	是
cos_headers	资源上传时设置的自定义 header 内容，如果未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否
data_id	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String	否

xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer	是
score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer	是
label	本次审核的结果标签，如果命中了敏感的关键词，该字段返回对应的关键词。	String	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：

```
{
 "EventName": "ReviewAudio",
 "JobsDetail": {
```

```
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "",
    "Object": "",
    "Label": "Normal",
    "Result": 0,
    "AudioText": "       ",
    "PornInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": ""
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": ""
    },
    "Section": [
      {
        "Url": "",
        "Text": "",
        "OffsetTime": 0,
        "Duration": 30000,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
        }
      }
    ],
    "BucketId": "",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "xxxx"
    }
  }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息	Object
EventName	任务的类型，为 ReviewAudio	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State为 Failed 时返回。详情请查看 错误码列表 。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	本次音频审核任务的 ID。	String
DataId	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String
State	音频审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	音频审核任务的创建时间。	String
Object	本次审核的音频文件名称，创建任务使用 Object 时返回。	String
Url	本次审核的音频文件链接，创建任务使用 Url 时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
SubLabel	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
AudioText	该字段用于返回音频文件识别出的对应文本内容。	Stri

		ng
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
Section	当音频过长时，会对音频进行分段，该字段用于返回音频片段的审核结果，主要包括开始时间和音频审核的相应结果。	Array
BucketId	创建的音频审核任务所属的存储桶名称。	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示音频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段，创建任务未设置 UserInfo 时无此字段。	Object
ListInfo	账号黑白名单结果。	Object

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	本次审核的结果分数，分数越高表示越敏感。	Integer
Category	该字段表示审核命中的具体审核类别。注意：该字段可能返回空。	String
SubLabel	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Label	本次审核的综合结果标签，如果命中了敏感关键词，该字段返回对应的关键词。	String

Section 的具体内容如下：

节点名称（关键字）	描述	类型
Url	当前音频片段的访问地址，您可以通过该地址获取该音频片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
Text	该字段用于返回当前音频片段的 ASR 文本识别的检测结果。	String
OffsetTime	该字段用于返回当前音频片段位于总音频中的时间，单位为毫秒，例如5000（音频开始后5000毫秒）。	Integer
Duration	当前音频片段的时长，单位毫秒。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
SubLabel	该字段表示审核命中的具体子标签。注意：该字段可能返回空。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Section 中的 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情	Integer

	99，表明该内容非常有可能属于色情内容。	
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
Keywords	本次审核命中的敏感关键词，没有时不返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
LibType	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	命中的风险库名称。	String
Keywords	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否

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）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

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

实际案例

案例一：简洁回调（Simple）

```
{
  "code": 0,
  "data": {
    "event":"ReviewAudio",
    "forbidden_status": 0,
    "porn_info": {
      "hit_flag": 0,
      "label": "",
      "score": 9
    },
```

```
      "result": 0,
      "trace_id": "ixzt90jl2dfscxxxxxxxxxxxxxxxx",
      "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/music.mp
      "cos_headers": {
          "x-cos-meta-id": "666666"
      }
   },
   "message": "success"
}
```


案例二：详细回调（Detail）


```
{
  "EventName": "ReviewAudio",
  "JobsDetail": {
    "JobId": "xxxxxx",
    "State": "Success",
    "CreationTime": "2021-08-10T21:01:08+08:00",
    "Object": "1.mp3",
    "Result": 0,
    "AudioText": "       ",
    "PornInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": ""
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Score": 0,
      "Label": ""
    },
    "Section": [
      {
        "Url": "https://audio-1250000000.cos.ap-guangzhou.myqcloud.com/0.mp3",
        "Text": "",
        "OffsetTime": 0,
        "Duration": 30000,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0
```

```
      }
     }
    ],
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "xxxx"
    }
  }
}
```

文本审核
文本文件审核

最近更新时间：2023-06-27 15:00:22

功能描述

本接口用于进行文本文件审核，为异步请求方式，您可以通过本接口提交审核任务，然后通过查询文本审核任务接口查询审核结果。
该接口支持情况如下：

> **说明**
> - 支持审核腾讯云 COS 上的文本文件（仅支持 UTF8 编码和 GBK 编码的内容）。
> - 支持审核第三方云存储厂商的文本文件 URL（仅支持 UTF8 编码和 GBK 编码的内容）。
> - 支持设置回调地址 Callback 获取检测结果，或通过 查询文本审核任务结果接口 主动轮询获取审核结果详情。
> - 支持识别多种违规场景，包括：色情、违法、广告等场景。
> - 支持根据不同的业务场景 配置自定义的审核策略 。

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个文本**，**计2次**审核费用。
- 调用接口会产生 文本审核费用 和 COS 读请求费用 。
- 如果文本属于 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文本，如果需要审核此类型文本，请先 恢复归档文件 。

限制说明

- 大小限制：文件不能超过1MB。
- 文件格式限制：无后缀限制，文件的内容需要是 UTF8 编码和 GBK 编码的。
- 接口默认并发数量：10。
- 字数限制：每 10000 个字符算一次，超出部分按多条计费。
- 语言支持：目前支持中文、中英混合、阿拉伯数字的检测。

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 。

版权所有：腾讯云计算（北京）有限责任公司

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>
  <Object></Object>
  <Url></Url>
  <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
Object	Request.Input	当前 COS 存储桶中的文本文件名称，例如在目录 test 中的文件 test.txt，则文件名称为 test/test.txt，文本文件仅支持 UTF8 编码和 GBK 编码的内容，且文件大小不得超过1MB。	String	否
Url	Request.Input	文本文件的完整链接，例如：`https://www.test.com/test.txt` 。	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否

AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如色情、广告、违法等，配置指引： 设置公共审核策略 。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Callback	Request.Conf	审核结果可以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如： `http://www.callback.com` 。当 Input 使用 Content 时，该参数不生效，结果会直接返回。	String	否
CallbackVersion	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	String	否

CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部文本片段）、2（回调违规文本片段）。默认为 1。	Integer	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对文本文件进行自动冻结，仅当 `input` 中审核的文本为 `object` 时有效。	Container	否

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>
  </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	文本审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
DataId	Response.JobsDetail	请求中添加的唯一业务标识。	String
JobId	Response.JobsDetail	本次文本审核任务的 ID。	String
State	Response.JobsDetail	文本审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	文本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 /text/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Input>
    <Object>a.txt</Object>
    <DataId>123-fdrsg-123</DataID>
  </Input>
  <Conf>
    <Callback>http://callback.com/</Callback>
    <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>
    <DataId>123-fdrsg-123</DataID>
    <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

文本内容审核

最近更新时间：2023-06-27 15:00:22

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 。

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。

限制说明

- 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。

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
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

版权所有：腾讯云计算（北京）有限责任公司

Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如色情、广告、违法等，配置指引： 设置公共审核策略 。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
Callback	Request.Conf	审核结果可以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。当 Input 使用 Content 时，该参数不生效，结果会直接返回。	String	否
CallbackVersion	Request.Conf	回调内容的结构，有效值：Simple（回调内容包含基本信息）、Detail（回调内容包含详细信息）。默认为 Simple。	String	否
CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部文本片段）、2（回调违规文本片段）。默认为 1。	Integer	否
Freeze	Request.Conf	可通过该字段，设置根据审核结果给出的不同分值，对文本文件进行自动冻结，仅当 `input` 中审核的文本为 `object` 时有效。	Container	否

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
PornSc	Request.Conf	取值为[0,100]，表示当色情审核结果大于或等于	Integ	否

ore	.Freeze	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。	er	
AdsScore	Request.Conf.Freeze	取值为[0,100]，表示当广告审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
IllegalScore	Request.Conf.Freeze	取值为[0,100]，表示当违法审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否
AbuseScore	Request.Conf.Freeze	取值为[0,100]，表示当谩骂审核结果大于或等于该分数时，自动进行冻结操作。不填写则表示不自动冻结，默认值为空。	Integer	否

其他审核场景的冻结参数，请联系服务人员咨询。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
 <JobsDetail>
  <DataId></DataId>
  <JobId></JobId>
  <State></State>
  <CreationTime></CreationTime>
  <Code>Success</Code>
  <Message>Success</Message>
  <SectionCount></SectionCount>
  <Result>1</Result>
  <PornInfo>
   <HitFlag></HitFlag>
   <Count></Count>
  </PornInfo>
  <Section>
   <StartByte></StartByte>
   <PornInfo>
    <HitFlag></HitFlag>
    <Score></Score>
```

```
      <Keywords></Keywords>
    </PornInfo>
  </Section>
 </JobsDetail>
 <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	文本审核返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	文本审核的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回，详情请查看 错误码列表 。	String
DataId	Response.JobsDetail	请求中添加的唯一业务标识。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
JobId	Response.JobsDetail	本次文本审核的任务ID。	String
CreationTime	Response.JobsDetail	文本审核的创建时间。	String

State	Response.JobsDetail	文本审核的状态，值为 Success（审核成功）或者 Failed（审核失败）。	String
Content	Response.JobsDetail	文本内容的 Base64 编码。	String
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
SectionCount	Response.JobsDetail	审核的文本内容分片数，固定为1。	Integer
PornInfo	Response.JobsDetail	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail	审核场景为**广告**的审核结果信息。	Container
IllegalInfo	Response.JobsDetail	审核场景为**违法**的审核结果信息。	Container
AbuseInfo	Response.JobsDetail	审核场景为**谩骂**的审核结果信息。	Container
Section	Response.JobsDetail	文本审核的具体结果信息。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container

Container 节点 PornInfo，AdsInfo，IllegalInfo，AbuseInfo 的内容：

节点名称（关键	父节点	描述	类型

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	审核场景为**广告**的审核结果信息。	Container

	.Section		
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	父节点	描述	类型

（关键字）			
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
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

版权所有：腾讯云计算（北京）有限责任公司

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-
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>
```

查询文本审核任务结果

最近更新时间：2023-06-21 10:22:02

功能描述

DescribeAuditingTextJob 接口用于查询指定的文本审核任务。

> **说明**
> 文本审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

数据万象 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口，点此查看 SDK 文档。

请求

请求示例

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

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数名称	描述	类型	是否必选
jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>
    <Message>Success</Message>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Object></Object>
    <SectionCount></SectionCount>
    <Result>1</Result>
    <PornInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </PornInfo>
    <AdsInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </AdsInfo>
    <IllegalInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </IllegalInfo>
    <AbuseInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </AbuseInfo>
    </Section>
      <StartByte></StartByte>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </PornInfo>
      <AdsInfo>
        <HitFlag></HitFlag>
```

```
        <Score></Score>
        <Keywords></Keywords>
      </AdsInfo>
      <IllegalInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </IllegalInfo>
      <AbuseInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </AbuseInfo>
    </Section>
    </Section>
      <StartByte></StartByte>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </PornInfo>
      <AdsInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </AdsInfo>
      <IllegalInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </IllegalInfo>
      <AbuseInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </AbuseInfo>
    </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	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
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回，详情请查看 错误码列表 。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	本次审核任务的 ID。	String
State	Response.JobsDetail	任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	Response.JobsDetail	任务的创建时间。	String
Object	Response.JobsDetail	审核的对象名称，当创建任务使用 Object 时返回。	String
Content	Response.JobsDetail	文本内容的 Base64 编码。当创建任务使用 Content 时返回。	String
SectionCount	Response.JobsDetail	文本审核会将文本进行分段送审，每10000个 utf8 编码字符分一段，该参数表示分段的数量。	Integer

版权所有：腾讯云计算（北京）有限责任公司

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
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	具体文本分片的审核结果信息。	Container Array
UserInfo	Response.JobsDetail	用户业务字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container
ForbidState	Response.JobsDetail	若您设置了自动冻结，该字段表示文本文件的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	String
Count	Response.Jobs	命中该审核分类的分片数。	Integ

	Detail.*Info		er

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.*Info	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	String
Score	Response.JobsDetail.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（预设风险库）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

版权所有：腾讯云计算（北京）有限责任公司

Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ListResults	Response.JobsDetail.ListInfo	命中的所有名单结果。	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

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 /text/auditing/stb1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>

    <JobId>st78925397b9de11eba532525400946f76</JobId>
    <State>Success</State>
    <CreationTime>2021-05-21T10:44:32+08:00</CreationTime>
    <Object>test.txt</Object>
    <SectionCount>1</SectionCount>
    <Label>Illegal</Label>
    <Result>1</Result>
    <PornInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </PornInfo>
    <AdsInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </AdsInfo>
    <IllegalInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </IllegalInfo>
    <AbuseInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </AbuseInfo>
    <Section>
```

版权所有：腾讯云计算（北京）有限责任公司

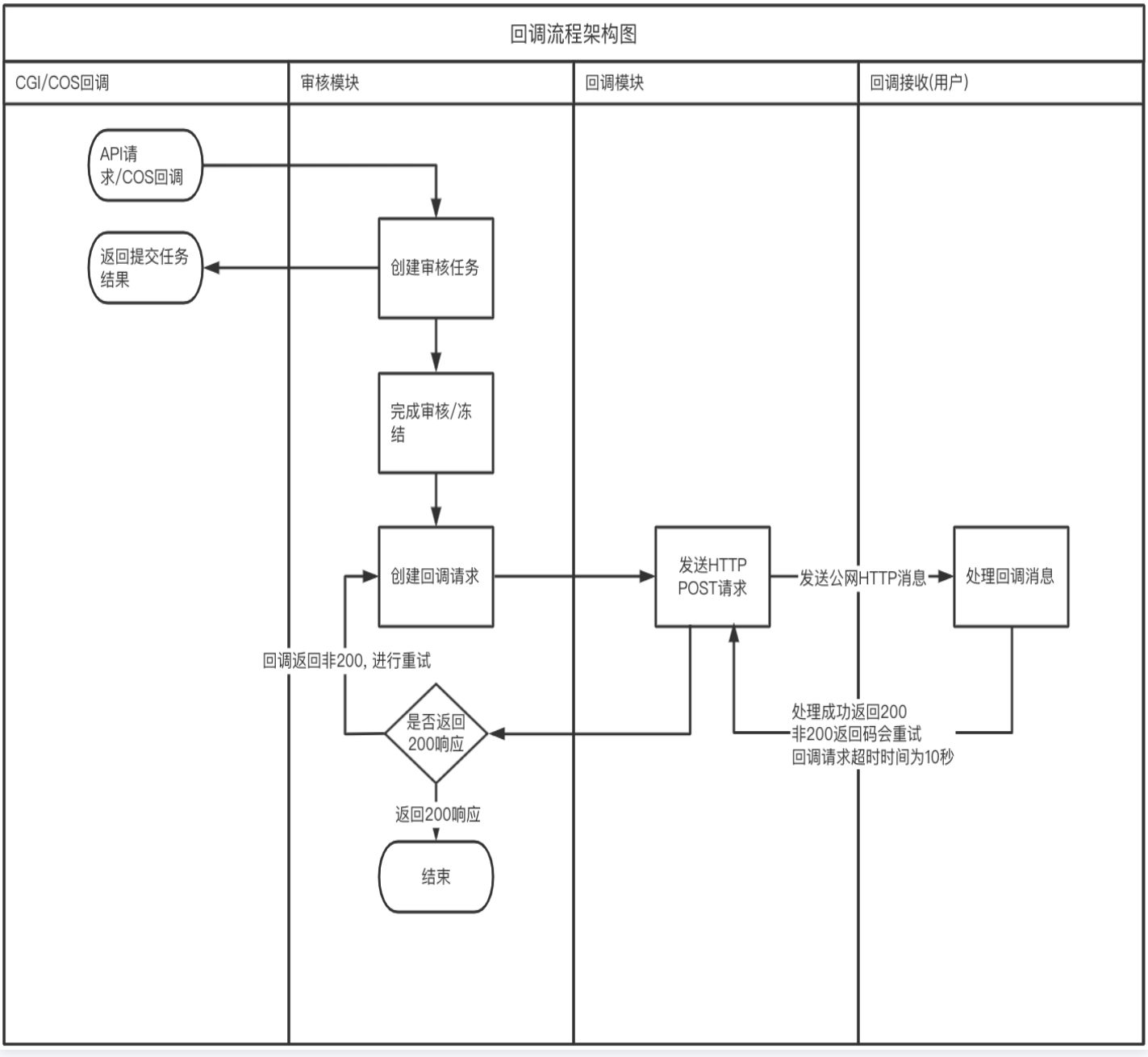
```
      <StartByte>0</StartByte>
      <Label>Illegal</Label>
      <Result>1</Result>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Keywords></Keywords>
      </PornInfo>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Keywords></Keywords>
      </AdsInfo>
      <IllegalInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Keywords></Keywords>
      </IllegalInfo>
      <AbuseInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Keywords></Keywords>
      </AbuseInfo>
    </Section>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

文本审核回调内容

最近更新时间：2022-08-03 15:01:34

功能描述

若您配置了审核回调地址，在文本审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。



回调内容说明

简洁回调（Simple）

回调通知为 HTTP 的 POST 方法，含有 X-Ci-Content-Version: Simple 的头部。
该响应体返回为 application/json 数据，包含完整节点数据的内容展示如下：

```
{
  "code": 0,
  "data": {
    "forbidden_status": 0,
    "event": "ReviewText",
    "porn_info": {
      "hit_flag": 0,
      "label": "",
      "count": 0
    },
    "result": 0,
    "trace_id": "test_trace_id",
    "url": "test_url",
    "cos_headers": {
      "x-cos-meta-xx": "xx"
    }
  },
  "message": "Test request when setting callback url"
}
```

具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码 。	Integer	是
message	错误描述。	String	是
data	文本审核结果的详细信息。	Object	是

data的具体内容如下：

参数名称	描述	类型	是否必选

trace_id	唯一 ID 标识。如果是通过接口创建的异步任务，则为接口返回的 JobId。	String	是
url	被审核文本的完整 URL 链接。	String	是
event	触发的事件，此处固定为 ReviewText。	String	是
result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示文本的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer	是
cos_headers	资源上传时设置的自定义 header 内容，若未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否
abuse_info	审核场景为**谩骂**的审核结果信息。	Object	否
illegal_info	审核场景为**违法**的审核结果信息。	Object	否
data_id	提交任务时设置了DataId参数时返回，返回原始内容，长度限制为512字节。	String	否

xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer	是
label	命中的关键字。	String	是
count	命中该审核分类的敏感文本段数。	Integer	是

版权所有：腾讯云计算（北京）有限责任公司

详细回调（Detail）

回调通知为 HTTP 的 POST 方法，含有 X-Ci-Content-Version: Detail 的头部。
该响应体返回为 application/json 数据，包含完整节点数据的内容展示如下：


```
{
 "EventName": "ReviewText",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "",
  "Object": "",
  "Label": "Normal",
  "Result": 0,
  "SectionCount": 1,
  "PornInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "AdsInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "IllegalInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "AbuseInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "Section": [
   {
    "StartByte": 0,
    "Label": "Normal",
    "Result": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Keywords": ""
    },
    "IllegalInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Keywords": ""
```

```
      },
      "AbuseInfo": {
        "HitFlag": 0,
        "Score": 0,
        "Keywords": ""
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0,
        "Keywords": ""
      }
    }
  ],
  "BucketId": "",
  "Region": "ap-chongqing",
  "ForbidState": 0,
  "CosHeaders": {
    "x-cos-meta-id": "xxxxx"
  }
 }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object
EventName	任务的类型，为 ReviewText 。	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	本次文本审核任务的 ID。	String
DataId	提交任务时设置了DataId参数时返回，返回原始内容，长度限制为512字节。	String
State	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成	String

	功）、Failed（审核失败）、Auditing（审核中）其中一个。	
CreationTime	审核任务的创建时间。	String
Object	被审核的文本文件的名称，创建任务使用 Object 时返回。	String
Content	本次审核的文本内容的 Base64编码，当创建任务使用 Content 时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，**Illegal**：违法，**Abuse**：谩骂。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
IllegalInfo	审核场景为**违法**的审核结果信息。	Object
AbuseInfo	审核场景为**谩骂**的审核结果信息。	Object
SectionCount	文本审核会将文本进行分段送审，每10000个 utf8 编码字符分一段，该参数表示分段的数量。	Integer
Section	文本内容每一段的审核结果。	Array
BucketId	创建的审核任务所属的存储桶名称。	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示文本的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段。	Container
ListInfo	账号黑白名单结果。	Container

PornInfo，AdsInfo，IllegalInfo，AbuseInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Count	命中该审核分类的文本段数。	Integer

Section 的具体内容如下：

节点名称（关键字）	描述	类型
StartByte	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	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，**Illegal**：违法，**Abuse**：谩骂。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
IllegalInfo	审核场景为**违法**的审核结果信息。	Object
AbuseInfo	审核场景为**谩骂**的审核结果信息。	Object

Section 中的 PornInfo，AdsInfo，IllegalInfo，AbuseInfo 的具体内容如下：

节点名称（关键字）	描述	类型

HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Keywords	在当前审核场景下命中的关键词，多关键词以 , 分割。	String
LibResults	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。	String

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
LibType	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	命中的风险库名称。	String
Keywords	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

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

实际案例

案例一：简洁回调（Simple）

```
{
  "code": 0,
  "data": {
     "event":"ReviewText",
     "forbidden_status": 0,
```

```
    "porn_info": {
      "hit_flag": 0,
      "label": "",
      "count": 0
    },
    "result": 0,
    "trace_id": "ixzt90jl2dfscxxxxxxxxxxxxxxxxx",
    "url": "https://examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/test.txt",
    "cos_headers": {
      "x-cos-meta-id": "xxxxxx"
    }
  },
  "message": "success"
}
```

案例二：详细回调（Detail）

```
{
 "EventName": "ReviewText",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "2021-08-10T21:01:08+08:00",
  "Object": "1.txt",
  "Label": "Normal",
  "Result": 0,
  "SectionCount": 1,
  "PornInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "AdsInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "Section": [
   {
    "StartByte": 0,
    "Label": "Normal",
    "Result": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Keywords": ""
```

```
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0,
        "Keywords": ""
      }
    }
  ],
  "BucketId": "examplebucket-1250000000",
  "Region": "ap-chongqing",
  "ForbidState": 0,
  "CosHeaders": {
    "x-cos-meta-id": "xxx"
  }
 }
}
```

文本审核结果反馈

最近更新时间：2022-07-06 17:57:15

功能描述

您可通过本接口反馈与预期不符的审核结果，例如色情文本被审核判定为正常或正常文本被判定为色情时可通过该接口直接反馈。
本接口不会直接修改审核结果，您反馈的错误审核结果将在后台进行确认，并在后续的审核任务中生效。

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 /report/badcase HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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>
  <ContentType></ContentType>
  <JobId></JobId>
```

```
    <ModerationTime></ModerationTime>
    <Text></Text>
    <SuggestedLabel></SuggestedLabel>
    <Label></Label>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	反馈审核错误的具体参数	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
ContentType	Request	需要反馈的数据类型，错误文本样本取值为1。	Integer	是
Text	Request	文本类型的样本，需要填写 base64 的文本内容，ContentType 为1时必填。	String	否
Label	Request	数据万象审核判定的审核结果标签，例如 Porn。	String	是
SuggestedLabel	Request	您自己期望的正确审核结果的标签，例如期望是正常，则填 Normal。	String	是
JobId	Request	该数据样本对应的审核任务 ID，有助于定位审核记录。	String	否
ModerationTime	Request	该数据样本之前审核的时间，有助于定位审核记录。格式为 2021-08-07T12:12:12+08:00	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
```

```
    <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	返回的具体响应内容	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
POST /report/badcase HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1********&q-s
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 66
Content-Type: application/xml
<Request>
    <ContentType>1</ContentType>
    <Text>6L+Z5piv5Li65LuA5LmI</Text>
    <Label>Porn</Label>
    <SuggestedLabel>Normal</SuggestedLabel>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 May 2022 12:37:29 GMT
```

版权所有：腾讯云计算（北京）有限责任公司

```
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

文档审核
提交文档审核任务

最近更新时间：2023-06-27 15:00:23

功能描述

本接口用于提交一个文档审核任务。文档审核功能为异步任务方式，您可以通过提交文档审核任务审核您的文档文件，然后通过 查询文档审核任务接口 或 文档审核回调 查询审核结果。

> **说明**
> - 支持审核腾讯云 COS 上的文档文件。
> - 支持审核第三方云存储厂商的文档 URL。

- 支持对文档文件进行自动检测，通过将文档页面转换截图，从 OCR 文本识别、图像识别维度，利用深度学习技术，识别文档中的违规内容。
- 支持设置回调地址 Callback 获取检测结果，或通过 查询文档审核任务结果接口 主动轮询获取审核结果详情。
- 支持识别多种违规场景，包括：色情、违法、广告等场景。
- 支持根据不同的业务场景 配置自定义的审核策略 。

费用说明

文档审核费用包含：

- 文档页面审核：基于文档转图片能力，将文档按页面转换为多张图片进行审核，再通过图片 OCR 技术识别其中的文字内容，审核费用与图片审核一致。
- 文档转图片：会产生相应的 文档转图片费用 。
- 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篇文档**，**计2次**审核费用。
- 调用接口会产生 图片审核费用 和 COS 读请求费用 。
- 如果文档文件属于 COS 上的低频存储类型，调用审核会产生 COS 低频数据取回费用 。
- 不支持审核 COS 上的归档存储类型和深度归档存储类型的文档，如果需要审核此类型文档，请先 恢复归档文件 。

限制说明

- 接口默认并发数量：10。
- 大小页数限制：文档大小限制在200MB之内，页数限制在5000页之内。
- 目前支持审核的文档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。

版权所有：腾讯云计算（北京）有限责任公司

- 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）
- PDF 文件。
- 其他格式文件：txt、log、htm、html、lrc、c、cpp、h、asm、s、java、asp、bat、bas、prg、cmd、rtf、xml。

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 /document/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 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>
      <Url></Url>
      <Type></Type>
      <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	文档审核的具体配置项。	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Input	Request	需要审核的内容。	Container	是
Conf	Request	审核规则配置。	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Object	Request.Input	存储在 COS 存储桶中的文件名称，例如在目录 test 中的文件 test.doc，则文件名称为 test/test.doc。Object 和 Url 只能选择其中一种。	String	否
Url	Request.Input	文档文件的链接地址，例如 http://www.example.com/doctest.doc，Object 和 Url 只能选择其中一种。	String	否
Type	Request.Input	指定文档文件的类型，如未指定则默认以文件的后缀为类型。 如果文件没有后缀，该字段必须指定，否则会审核失败。例如：doc、docx、ppt、pptx 等。	String	否
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为 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
UserInf	Request.I	用户业务字段。	Conta	否

o	nput		iner	

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	String	否

		告、违法违规等，配置指引： 设置公共审核策略 。 您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。 BizType 不填写时，将自动使用默认的审核策略。		
Callback	Request.Conf	审核结果可以回调形式发送至您的回调地址，支持以 `http://` 或者 `https://` 开头的地址，例如：`http://www.callback.com` 。	String	否
CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部分页）、2（回调违规分页）。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，对文档进行自动冻结。仅当 `input` 中审核的文档为 `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：

```
<Request>
  <JobsDetail>
    <DataId></DataId>
```

```
      <JobId></JobId>
      <State></State>
      <CreationTime></CreationTime>
    </JobsDetail>
    <RequestId></RequestId>
</Request>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	提交文档审核任务返回的具体响应内容。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	文档审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

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

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
POST /document/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml
<Request>
    <Input>
      <Url>http://www.example.com/doctest.doc</Url>
      <DataId>123-fdrsg-123</DataID>
    </Input>
    <Conf>
      <DetectType>Porn,Ads</DetectType>
      <Callback></Callback>
      <BizType>b81d45f94b91a683255e9a9506f4****</BizType>
    </Conf>
</Request>
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <DataId>123-fdrsg-123</DataID>
    <JobId>vab1ca9fc8a3ed11ea834c52540086****</JobId>
    <State>Submitted</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxx</RequestId>
</Response>
```

查询文档审核任务结果

最近更新时间：2023-06-20 17:32:13

功能描述

本接口用于查询指定的文档审核任务结果。

> **说明**
> 文档审核任务结果保存时效为1个月，您可通过本接口查询近1个月的审核结果。

推荐使用 SDK

数据万象 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

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

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数名称	描述	类型	是否必选
jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>
    <JobId>aab1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Success</State>
    <Code>Success</Code>
    <Message>Success</Message>
    <DataId></DataId>
    <Suggestion>1</Suggestion>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Url>http://www.test.com/123</Url>
    <PageCount>3</PageCount>
    <Labels>
      <PornInfo>
        <HitFlag>1</HitFlag>
        <Score>96</Score>
      </PornInfo>
    </Labels>
    <PageSegment>
      <Results>
        <Url></Url>
        <Text></Text>
        <PageNumber>1</PageNumber>
        <SheetNumber>1</SheetNumber>
        <PornInfo>
          <HitFlag></HitFlag>
          <SubLabel></SubLabel>
          <Score></Score>
          <OcrResults>
            <Text></Text>
            <Keywords></Keywords>
          </OcrResults>
        </PornInfo>
      </Results>
    </PageSegment>
  </JobsDetail>
  <RequestId></RequestId>
```

```
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	文档审核的全部信息。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	文档审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回，详情请查看 错误码列表 。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	文档审核任务的 ID。	String
State	Response.JobsDetail	任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
Suggestion	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer
Label	Response.J	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**	String

	obsDetail	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。	
CreationTime	Response.JobsDetail	任务的创建时间。	String
Object	Response.JobsDetail	审核的文件为存储在 COS 中的文件时，该字段表示本次审核的文件名称。	String
Url	Response.JobsDetail	审核的文件链接，创建任务使用 Url 时返回。	String
PageCount	Response.JobsDetail	文档审核会将文档转换为图片进行审核，该字段表示转换的图片总数量。	Integer
Labels	Response.JobsDetail	该字段用于返回命中的审核场景及对应的结果。	Container
PageSegment	Response.JobsDetail	文档转换为图片后，具体每张图片的审核结果信息。	Container
UserInfo	Response.JobsDetail	用户业务字段。创建任务时未设置 UserInfo 则无此字段。	Container
ListInfo	Response.JobsDetail	账号黑白名单结果。	Container
ForbidState	Response.JobsDetail	若您设置了自动冻结，该字段表示文档的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer

Container 节点 Labels 的内容：

节点名称（关键字）	父节点	描述	类型
PornInfo	Response.JobsDetail.Labels	审核场景为色情的审核结果信息。	Container
AdsInfo	Response.JobsDetail.Labels	审核场景为广告的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型

HitFlag	Response.JobsDetail.Labels.*Info	用于返回该审核场景的审核结果，返回值：0：正常 1：确认为当前场景的违规内容2：疑似为当前场景的违规内容。	Integer
Score	Response.JobsDetail.Labels.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer

Container 节点 PageSegment 的内容：

节点名称（关键字）	父节点	描述	类型
Results	Response.JobsDetail.PageSegment	文档转换为图片后，每张图片的详细审核结果信息。	Container Array

Container 节点 Results 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.PageSegment.Results	文档转换成图片后，您可以通过该地址查看该图片内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	Response.JobsDetail.PageSegment.Results	该字段用于返回当前图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
PageNumber	Response.JobsDetail.PageSegment.Results	该图片的页码，通常是文档的页码。	Integer
SheetNumber	Response.JobsDetail.PageSegment.Results	如果审核的为表格文件，该字段表示表格内的 Sheet 页码。	Integer
Label	Response.JobsDetail.PageSegment.Results	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：	String

版权所有：腾讯云计算（北京）有限责任公司

		Normal：正常，Porn：色情，Ads：广告，以及其他不安全或不适宜的类型。	
Suggestion	Response.JobsDetail.PageSegment.Results	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.PageSegment.Results	审核场景为**色情**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.PageSegment.Results	审核场景为**广告**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.PageSegment.Results.*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
Category	Response.JobsDetail.PageSegment.Results.*Info	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。 注意：该字段可能返回空。	String
SubLabel	Response.JobsDetail.PageSegment.Results.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
Score	Response.JobsDetail.PageSegment.Results.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer
OcrResults	Response.JobsDetail.Pag	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返	Container Array

	eSegment.Results.*Info	回。	
LibResults	Response.JobsDetail.PageSegment.Results.*Info	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
ImageId	Response.JobsDetail.PageSegment.Results.*Info.LibResults	该字段表示命中的风险库中的图片样本 ID。	String
Score	Response.JobsDetail.PageSegment.Results.*Info.LibResults	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本；如：色情 99，则表明该数据非常有可能命中库中的色情样本。	Integer

Container 节点 OcrResults 的内容：

节点名称（关键字）	父节点	描述	类型
Text	Response.JobsDetail.PageSegment.Results.*Info.OcrResults	图片 OCR 文本识别出的具体文本内容。	String
Keywords	Response.JobsDetail.PageSegment.Results.*Info.OcrResults	在当前审核场景下命中的关键词。	String Array
Location	Response.JobsDetail.PageSegment.Results.*Info.OcrResults	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Container

ObjectResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.PageSegment.Results.*Info.ObjectResults	该标签用于返回所识别出的实体名称，例如人名。	String
Location	Response.JobsDetail.PageSegment.Results.*Info.ObjectResults	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Container

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

Container 节点 UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否

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 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

版权所有：腾讯云计算（北京）有限责任公司

错误码

此接口遵循统一的错误响应和错误码，详情请参见 错误码 文档。

实际案例

请求

```
GET /document/auditing/stb1ca9fc8a34c525400863904**** HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****
<Response>
  <JobsDetail>
    <JobId>stb1ca9fc8a34c525400863904****</JobId>
    <State>Success</State>
    <Suggestion>1</Suggestion>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Url>http://www.test.com/123.doc</Url>
    <PageCount>3</PageCount>
    <Labels>
      <PornInfo>
        <HitFlag>1</HitFlag>
        <Score>96</Score>
      </PornInfo>
    </Labels>
    <PageSegment>
      <Results>
        <Url>https://examplebucket-1250000000.cos.ap-nanjing.myqcloud.com/audit-1.jp
        <Text>哈哈哈，您好</Text>
        <PageNumber>1</PageNumber>
        <PornInfo>
          <HitFlag>0</HitFlag>
          <SubLabel>sex</SubLabel>
```

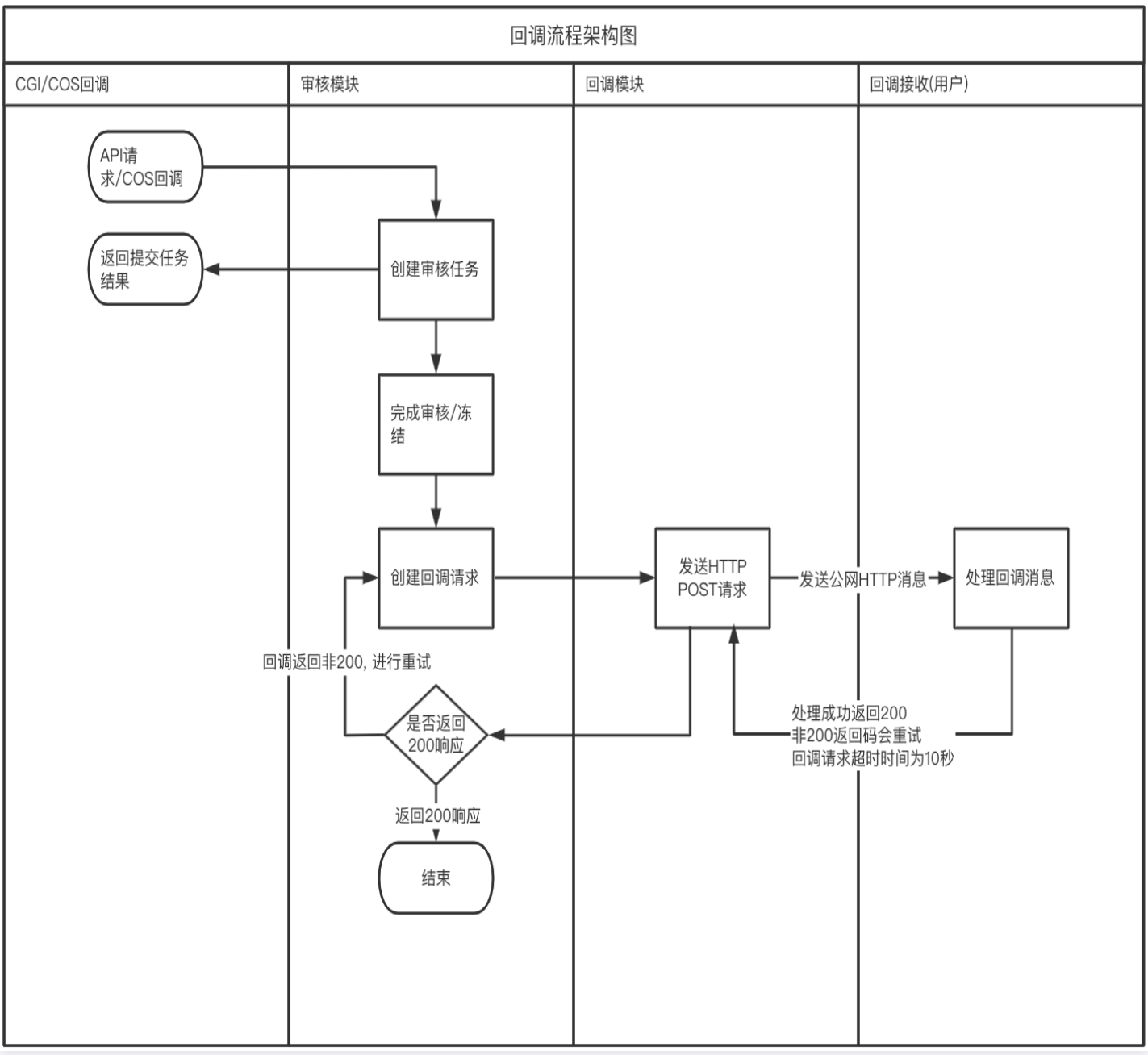
```
        <Score>0</Score>
        <OcrResults>
          <Text>哈哈哈，您好</Text>
          <Keywords>您好</Keywords>
        </OcrResults>
      </PornInfo>
    </Results>
    <Results>
      <Url>https://examplebucket-1250000000.cos.ap-nanjing.myqcloud.com/audit-2.jp
      <Text>哈哈哈，您好</Text>
      <PageNumber>1</PageNumber>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <SubLabel>sex</SubLabel>
        <Score>0</Score>
        <OcrResults>
          <Text>哈哈哈，您好</Text>
          <Keywords>您好</Keywords>
        </OcrResults>
      </PornInfo>
    </Results>
  </PageSegment>
 </JobsDetail>
 <RequestId>xxxxxxxxxxxxx</RequestId>
</Response>
```

文档审核回调内容

最近更新时间：2022-10-17 16:14:41

功能描述

若您配置了审核回调地址，在文档审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。



回调内容说明

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


```
{
 "EventName": "ReviewDocument",
 "JobsDetail": {
  "JobId": "6666666666666666666666666666666666",
  "State": "Success",
  "CreationTime": "",
  "Url": "",
  "Label": "Normal",
  "Suggestion": 0,
  "PageCount": 1,
  "Labels": {
   "PornInfo": {
    "HitFlag": 0,
    "Score": 0
   },
   "AdsInfo": {
    "HitFlag": 0,
    "Score": 0
   }
  },
  "PageSegment": {
   "Results": [
   {
    "Url": "",
    "Text": "",
    "PageNumber": 0,
    "SheetNumber": 0,
    "Label": "Normal",
    "Suggestion": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "SubLabel": "",
     "OcrResults": [{
      "Text": "",
      "Keywords":["",""],
      "Location": {
       "X": 0,
       "Y": 0,
       "Width": 0,
       "Height": 0,
       "Rotate": 0
      }
```

```
        }]
       },
       "AdsInfo": {
        "HitFlag": 0,
        "Score": 0,
        "SubLabel": ""
       }
      }
     ]
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "666666"
    }
  }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object
EventName	任务的类型，为 ReviewDocument 。	String

JobsDetail 的内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
DataId	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。	String
State	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String

Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：0（审核正常），1（判定为违规敏感文件），2（疑似敏感，建议人工复核）。	Integer
CreationTime	审核任务的创建时间。	String
Object	审核的文件为存储在 COS 中的文件时，该字段表示本次审核的文件名称。	String
Url	审核的文件链接，创建任务使用Url时返回。	String
PageCount	文档审核会将文档转换为图片进行审核，该字段表示转换的图片总数量。	Integer
Labels	该字段用于返回命中的审核场景及对应的结果。	Object
PageSegment	文档转换为图片后，具体每张图片的审核结果信息。	Object
ForbidState	若您设置了自动冻结，该字段表示文档的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段。创建任务时未设置UserInfo则无此字段。	Object
ListInfo	账号黑白名单结果。	Container

Labels 的内容如下：

节点名称（关键字）	描述	类型
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Labels 中 PornInfo，AdsInfo 的内容如下：

节点名称（关键字）	描述	类型
HitFla	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	Integ

版权所有：腾讯云计算（北京）有限责任公司

g	容。2：疑似为当前场景的违规内容。	er
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情99，表明该内容非常有可能属于色情内容。	Integer

PageSegment 的内容如下：

节点名称（关键字）	描述	类型
Results	文档转换为图片后，每张图片的详细审核结果信息。	Array

Results 的内容如下：

节点名称（关键字）	描述	类型
Url	文档转换成图片后，您可以通过该地址查看该图片内容，地址格式为标准 URL 格式。注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	该字段用于返回当前图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
PageNumber	该图片的页码，通常是文档的页码。	Integer
SheetNumber	如果审核的为表格文件，该字段表示表格内的 Sheet 页码。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Results 中 PornInfo，AdsInfo 的内容如下：

节点名称（关键字）	描述	类型

HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

版权所有：腾讯云计算（北京）有限责任公司

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，例如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

Room	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

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

实际案例

```
{
 "EventName": "ReviewDocument",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "2021-08-10T21:01:10+08:00",
  "Url": "http://test.com/test.doc",
```

```
    "Label": "Normal",
    "Suggestion": 0,
    "PageCount": 1,
    "Labels": {
      "PornInfo": {
        "HitFlag": 0,
        "Score": 0
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0
      }
    },
    "PageSegment": {
      "Results": [
      {
        "Url": "http://audit-125000000.cos.ap-chongqing.myqcloud.com/1.jpg",
        "Text": "",
        "PageNumber": 1,
        "SheetNumber": 0,
        "Label": "Normal",
        "Suggestion": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0,
          "SubLabel": ""
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0,
          "SubLabel": ""
        }
      }
      ]
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "xxxx"
    }
  }
}
```

网页审核
提交网页审核任务

最近更新时间：2023-06-27 15:00:23

功能描述

本接口用于提交一个网页审核任务。网页审核功能为异步任务方式，您可以通过提交网页审核任务审核您的网页，然后通过 查询网页审核任务接口 或 网页审核回调 查询审核结果。
该接口支持情况如下：

- 支持对网页文件进行自动检测，从 OCR 文本识别、物体检测（实体、广告台标、二维码等）、图像识别几个维度，通过深度学习技术，识别网页中的违规内容。
- 支持识别多种违规场景，包括：色情、违法、广告等场景。

费用说明

网页的审核分为**网页画面审核**、**网页文字审核**，其中：

- 网页画面审核：通过爬取网页内图片的方式，将爬取的多张图片进行审核，审核费用与图片审核一致。
- 网页文本审核：将网页内的文本分离出来进行文本审核，审核费用与文本审核一致。
- 每个审核场景单独计费，例如您选择审核涉黄、广告两种场景，则审核**1个网页**，**计2次**审核费用。
- 调用接口会产生 图片审核费用 、文本审核费用 和 COS 请求费用 。

限制说明

- 接口默认并发数量：10。

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 /webpage/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

版权所有：腾讯云计算（北京）有限责任公司

```
<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>
  <Url></Url>
    <DataId></DataId>
    <UserInfo></UserInfo>
 </Input>
 <Conf>
  <Biztype></Biztype>
  <Callback></Callback>
  <ReturnHighlightHtml>true</ReturnHighlightHtml>
 </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	网页审核的具体配置项。	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	需要审核的网页信息。	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
Url	Request.Input	网页文件的链接地址，例如 `http://www.test.com` 。	String	是
DataId	Request.Input	该字段在审核结果中会返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String	否
UserInfo	Request.Input	用户业务字段。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	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	Request.Input.UserInfo	一般用于表示性别信息，长度不超过128字节。	String	否
Level	Request.Input.	一般用于表示等级信息，长度不超过128	Strin	否

版权所有：腾讯云计算（北京）有限责任公司

	UserInfo	字节。	g	
Role	Request.Input.UserInfo	一般用于表示角色信息，长度不超过128字节。	String	否

Container 类型 Conf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
BizType	Request.Conf	表示审核策略的唯一标识，您可以通过控制台上的审核策略页面，配置您希望审核的场景，如涉黄、广告、违法违规等。您可以在控制台上获取到 BizType。BizType 填写时，此条审核请求将按照该审核策略中配置的场景进行审核。BizType 不填写时，将自动使用默认的审核策略。	String	否
DetectType	Request.Conf	（该参数后续不再维护，请使用BizType参数）审核的场景类型，有效值：Porn（涉黄）、Ads（广告）等，可以传入多种类型，不同类型以逗号分隔，例如：Porn,Ads。如您有更多场景的审核需要，请使用 BizType 参数。	String	否
Callback	Request.Conf	回调地址，以 `http://` 或者 `https://` 开头的地址。	String	否
ReturnHighlightHtml	Request.Conf	指定是否需要高亮展示网页内的违规文本，查询及回调结果时会根据此参数决定是否返回高亮展示的 html 内容。取值为 true 或者 false，默认为 false。	Boolean	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>
    <JobId></JobId>
```

```
    <State></State>
    <CreationTime></CreationTime>
  </JobsDetail>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	网页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	网页审核任务的详细信息。	Container

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
DataId	Response.JobsDetail	请求中添加的 DataId 字段。	String
JobId	Response.JobsDetail	本次网页审核任务的 ID。	String
State	Response.JobsDetail	网页审核任务的状态，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	网页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 /webpage/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
```

```
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Input>
    <Url>http://test.com</Url>
  </Input>
  <Conf>
    <ReturnHighlightHtml>true</ReturnHighlightHtml>
    <DetectType>Porn,Ads</DetectType>
  </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>
        <CreationTime>2021-11-09T09:55:53+08:00</CreationTime>
        <JobId>sh2c1260a4410011eca1f1525400276c76</JobId>
        <State>Submitted</State>
        <Url>http://test.com</Url>
    </JobsDetail>
    <RequestId>NjE4OWQ1Mjlf*****MzQ0OF85</RequestId>
</Response>
```

查询网页审核任务结果

最近更新时间：2023-06-20 17:32:13

功能描述

本接口用于主动查询指定的网页审核任务结果。网页审核功能为异步任务方式，您可以通过提交网页审核任务审核您的网页文件，然后通过查询网页审核任务接口查询审核结果。

> **说明**
> 网页审核任务结果保存时效为3个月，您可通过本接口查询近3个月的审核结果。

推荐使用 SDK

数据万象 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

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

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数名称	描述	类型	是否必选

jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>
    <JobId></JobId>
    <State>Success</State>
    <Code>Success</Code>
    <Message>Success</Message>
    <Suggestion>1</Suggestion>
    <CreationTime></CreationTime>
    <Url></Url>
    <PageCount>3</PageCount>
    <Labels>
      <PornInfo>
        <HitFlag>1</HitFlag>
        <Score>96</Score>
      </PornInfo>
    </Labels>
    <ImageResults>
      <Results>
        <Url></Url>
        <Text></Text>
        <PornInfo>
          <HitFlag></HitFlag>
          <SubLabel></SubLabel>
          <Score></Score>
          <OcrResults>
            <Text></Text>
            <Keywords></Keywords>
            <Location>
              <X></X>
              <Y></Y>
              <Width></Width>
              <Height></Height>
              <Rotate></Rotate>
            </Location>
```

```
        </OcrResults>
      </PornInfo>
    </Results>
  </ImageResults>
  <TextResults>
    <Results>
      <Text></Text>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Keywords></Keywords>
      </PornInfo>
    </Results>
  </TextResults>
 </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	网页审核的全部结果信息。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	网页审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型
Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回，详情请查看 错误码列表 。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String

JobId	Response.JobsDetail	网页审核任务的 ID。	String
DataId	Response.JobsDetail	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
State	Response.JobsDetail	网页审核任务的状态，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	网页审核任务的创建时间。	String
Url	Response.JobsDetail	本次审核的文件链接，创建任务使用 Url 时返回。	String
Labels	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。	Container
Suggestion	Response.JobsDetail	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告等。	String
PageCount	Response.JobsDetail	网页审核会将网页中的图片链接和文本分开送审，该字段表示送审的链接和文本总数量。	Integer
ImageResults	Response.JobsDetail	网页内图片的审核结果。	Container
TextResults	Response.JobsDetail	网页内文字的审核结果。	Container
HighlightHtml	Response.JobsDetail	对违规关键字高亮处理的Html网页内容，请求内容指定 ReturnHighlightHtml 时返回。	String
UserInfo	Response.JobsDetail	用户业务字段。创建任务未设置 UserInfo 时无此字段。	Container

ListInfo	Response.JobsDetail	账号黑白名单结果。	Container

Container 节点 Labels 的内容：

节点名称（关键字）	父节点	描述	类型
PornInfo	Response.JobsDetail.Labels	审核场景为**涉黄**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.Labels	审核场景为**广告引导**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Labels.*Info	Response.JobsDetail.Labels.*Info	用于返回该审核场景的审核结果，返回值：0：正常1：确认为当前场景的违规内容2：疑似为当前场景的违规内容。
Score	Response.JobsDetail.Labels.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer

Container 节点 ImageResults 的内容：

节点名称（关键字）	父节点	描述	类型
Results	Response.JobsDetail.ImageResults	网页内图片审核的结果，审核多张图片将返回多个 `Results` 。	Container Array

Container 节点 Results 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.ImageResults.Results	网页中图片审核，被审核图片的访问地址，您可以通过该地址查看图片，地址格式为标准 URL 格式。	String
Text	Response.JobsDetail.ImageResults.Results	该字段用于返回当前图片的 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	Response.JobsDetail.ImageResults.Results	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	Response.JobsDetail.ImageResults.Results	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.ImageResults.Results	审核场景为**涉黄**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.ImageResults.Results	审核场景为**广告引导**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.ImageResults.Results.*Info	Response.JobsDetail.Labels.*Info	用于返回该审核场景的审核结果，返回值：0：正常1：确认为当前场景的

			违规内容2：疑似为当前场景的违规内容。
Score	Response.JobsDetail.ImageResults.Results.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Category	Response.JobsDetail.ImageResults.Results.*Info	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	Response.JobsDetail.ImageResults.Results.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	Response.JobsDetail.ImageResults.Results.*Info	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Container Array
LibResults	Response.JobsDetail.ImageResults.Results.*Info	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
ImageId	Response.JobsDetail.ImageResults.Results.*Info.LibResults	该字段表示命中的风险库中的图片样本 ID。	String
Score	Response.JobsDetail.ImageResults.Results.*Info.LibResults	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

Container 节点 OcrResults 的内容：

节点名称（关键字）	父节点	描述	类型
Text	Response.JobsDetail.ImageResults.Results.*Info.OcrResults	图片 OCR 文本识别出的具体文本内容。	String
Keywords	Response.JobsDetail.ImageResults.Results.*Info.OcrResults	在当前审核场景下命中的关键词。	String Array
Location	Response.JobsDetail.ImageResults.Results.*Info.OcrResults	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Container

ObjectResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.ImageResults.Results.*Info.ObjectResults	该标签用于返回所识别出的实体名称，例如人名。	String
Location	Response.JobsDetail.ImageResults.Results.*Info.ObjectResults	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Container

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float

Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0 – 360（角度制）**，方向为**逆时针旋转**。	Float

Container 节点 TextResults 的内容：

节点名称（关键字）	父节点	描述	类型
Results	Response.JobsDetail.TextResults	网页内文本审核的结果，审核多条文本将返回多个 `Results` 。	Container Array

Container 节点 Results 的内容：

节点名称（关键字）	父节点	描述	类型
Text	Response.JobsDetail.TextResults.Results	网页文本将进行分段送审，每10000个 utf8 编码字符分一段，该参数返回本段文本内容。	String
Label	Response.JobsDetail.TextResults.Results	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	Response.JobsDetail.TextResults.Results	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.TextResults.Results	审核场景为**涉黄**的审核结果信息。	Container

AdsInfo	Response.JobsDetail.TextResults.Results	审核场景为**广告引导**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.TextResults.Results.*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.TextResults.Results.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。	Integer
Keywords	Response.JobsDetail.TextResults.Results.*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.TextResults.Results.*Info	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
LibType	Response.JobsDetail.TextResults.Results.*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.TextResults.Results.*Info.LibResults	命中的风险库名称。	String
Keywords	Response.JobsDetail.TextResults.Results.*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）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	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	Request.Input.UserInfo	一般用于表示性别信息，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	一般用于表示等级信息，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	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

```
GET /webpage/auditing/vab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Tue, 09 Nov 2021 01:56:09 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <CreationTime>2021-11-09T10:03:32+08:00</CreationTime>
    <HighlightHtml>xxxxx</HighlightHtml>
    <ImageResults>
      <Results>
        <AdsInfo>
          <HitFlag>0</HitFlag>
          <Score>0</Score>
          <SubLabel/>
        </AdsInfo>
        <Label>Porn</Label>
        <PornInfo>
```

```
        <HitFlag>1</HitFlag>
        <Score>99</Score>
        <SubLabel>ACGPorn</SubLabel>
      </PornInfo>
      <Suggestion>1</Suggestion>
      <Text>xxxxxxx</Text>
      <Url>xxxxxx</Url>
    </Results>
  </ImageResults>
  <JobId>sh3dba0967410111eca1f1525400276c76</JobId>
  <Label>Porn</Label>
  <Labels>
    <AdsInfo>
      <HitFlag>0</HitFlag>
      <Score>0</Score>
    </AdsInfo>
    <PornInfo>
      <HitFlag>1</HitFlag>
      <Score>99</Score>
    </PornInfo>
  </Labels>
  <PageCount>2</PageCount>
  <State>Success</State>
  <Suggestion>1</Suggestion>
  <TextResults>
    <Results>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Keywords/>
        <Score>0</Score>
      </AdsInfo>
      <Label>Porn</Label>
      <PornInfo>
        <HitFlag>1</HitFlag>
        <Keywords>xxxx</Keywords>
        <Score>98</Score>
      </PornInfo>
      <Suggestion>1</Suggestion>
      <Text>xxxx</Text>
    </Results>
  </TextResults>
  <Url>https://xxxx</Url>
 </JobsDetail>
 <RequestId>NjE4OWQ3MDNfY****RfMzQ0OF9h</RequestId>
</Response>
```

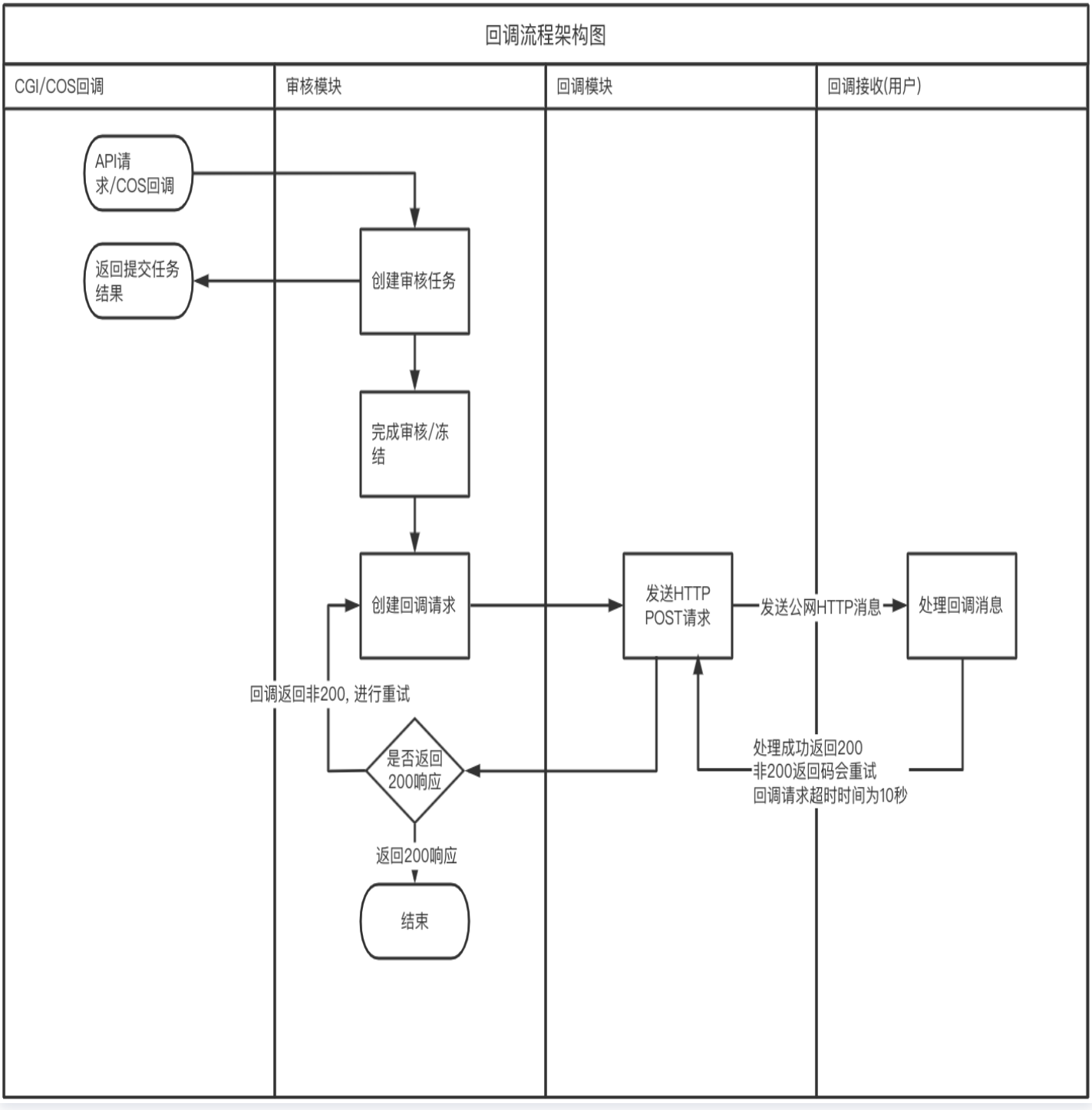
版权所有：腾讯云计算（北京）有限责任公司

网页审核回调内容

最近更新时间：2022-10-17 16:19:45

功能描述

若您配置了审核回调地址，在网页审核结束后，后台会将审核结果以 json 形式回调至您的回调地址。您可以通过回调内容，进行后续的文件处理操作。



回调内容说明

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


```
{
 "EventName": "ReviewHtml",
 "JobsDetail": {
  "JobId": "6666666666666666666666666666666666",
  "State": "Success",
  "CreationTime": "",
  "Url": "",
  "Label": "Normal",
  "Suggestion": 0,
  "PageCount": 1,
  "Labels": {
   "PornInfo": {
    "HitFlag": 0,
    "Score": 0
   },
   "AdsInfo": {
    "HitFlag": 0,
    "Score": 0
   }
  },
  "ImageResults": {
   "Results": [
   {
    "Url": "",
    "Text": "",
    "Label": "Normal",
    "Suggestion": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "SubLabel": "",
     "OcrResults": [{
      "Text": "",
      "Keywords":["",""],
      "Location": {
       "X": 0,
       "Y": 0,
       "Width": 0,
       "Height": 0,
```

```
          "Rotate": 0
        }
       }]
      },
      "AdsInfo": {
       "HitFlag": 0,
       "Score": 0,
       "SubLabel": ""
      }
     }
     ]
    },
    "TextResults": {
     "Results": [
     {
      "Text": "",
      "Label": "Normal",
      "Suggestion": 0,
      "PornInfo": {
       "HitFlag": 0,
       "Score": 0,
       "Keywords": ""
      },
      "AdsInfo": {
       "HitFlag": 0,
       "Score": 0,
       "Keywords": ""
      }
     }
     ]
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "666666"
    }
   }
}
```


具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object

EventName	任务的类型，为 ReviewHtml 。	String

JobsDetail 的内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
DataId	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
State	审核任务的状态，值为 Submitted（已提交审核）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告等。	String
Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
CreationTime	审核任务的创建时间。	String
Url	审核的网页链接，创建任务使用 Url 时返回。	String
PageCount	网页审核会将网页中的图片链接和文本分开送审，该字段表示送审的链接和文本总数量。	Integer
Labels	该字段用于返回命中的审核场景及对应的结果。	Object
ImageResults	网页内图片的审核结果。	Object
TextResults	网页内文字的审核结果。	Object

HighlightHtml	对违规关键字高亮处理的 HTML 网页内容，请求内容指定 ReturnHighlightHtml 时返回。	String
UserInfo	用户业务字段。当创建任务时未设置 UserInfo 则无此字段。	Object
ListInfo	账号黑白名单结果。	Container

Labels 的内容如下：

节点名称（关键字）	描述	类型
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Labels 中 PornInfo，AdsInfo 的内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer

ImageResults 的内容如下：

节点名称（关键字）	描述	类型
Results	网页内图片审核的结果。	Array

Results 的内容如下：

节点名称（关键字）	描述	类型
Url	网页中图片审核，被审核图片的访问地址，您可以通过该地址查看图片，地址格式为标准 URL 格式。	String
Text	该字段用于返回当前图片的 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String

版权所有：腾讯云计算（北京）有限责任公司

Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Results 中 PornInfo，AdsInfo 的内容如下：

节点名称（关键字）	描述	类型
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。通常[0,60]分可以认为图片属于正常，[61,90]分表示图片疑似敏感，[91,100]分表示图片为敏感的。例如：色情 99，表明该内容非常有可能属于色情内容。	Integer
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。	String
SubLabel	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本。例如：色情 99，表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，例如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一	Flo

	确定检测框的大小和位置。	at
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0-360（角度制）**，方向为**逆时针旋转**。	Float

TextResults 的具体内容如下：

节点名称（关键字）	描述	类型
Results	网页内文本审核的结果。	Array

Results 的具体内容如下：

节点名称（关键字）	描述	类型
Text	网页文本将进行分段送审，每10000个 utf8 编码字符分一段，该参数返回本段文本内容。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Suggestion	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键	描述	类型

字）		
HitFlag	用于返回该审核场景的审核结果，返回值：0：正常。1：确认为当前场景的违规内容。2：疑似为当前场景的违规内容。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。	Integer
Keywords	在当前审核场景下命中的关键词，多关键词以 , 分割。	String
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
LibType	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	命中的风险库名称。	String
Keywords	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

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	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	一般用于表示设备信息，长度不超过128字节。	String	否

AppId	Request.Input.UserInfo	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	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	Request.Input.UserInfo	一般用于表示性别信息，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	一般用于表示等级信息，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

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

实际案例


```
{
 "EventName": "ReviewHtml",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "2021-08-10T21:01:10+08:00",
  "Url": "http://test.com/test.html",
  "Label": "Normal",
  "Suggestion": 0,
  "PageCount": 2,
  "Labels": {
   "PornInfo": {
    "HitFlag": 0,
    "Score": 0
   },
   "AdsInfo": {
    "HitFlag": 0,
    "Score": 0
   }
  },
  "ImageResults": {
   "Results": [
   {
    "Url": "http://xxx.xxx.com/a.jpg",
    "Text": "",
    "Label": "Normal",
    "Suggestion": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "SubLabel": "",
    },
    "AdsInfo": {
     "HitFlag": 0,
```

```
        "Score": 0,
        "SubLabel": ""
       }
      }
      ]
    },
    "TextResults": {
     "Results": [
     {
      "Text": "xxxxxxx",
      "Label": "Normal",
      "Suggestion": 0,
      "PornInfo": {
       "HitFlag": 0,
       "Score": 0,
       "Keywords": ""
      },
      "AdsInfo": {
       "HitFlag": 0,
       "Score": 0,
       "Keywords": ""
      }
     }
     ]
    },
    "BucketId": "examplebucket-1250000000",
    "Region": "ap-chongqing",
    "ForbidState": 0,
    "CosHeaders": {
       "x-cos-meta-id": "xxxx"
    }
  }
}
```

直播审核
提交直播审核任务

最近更新时间：2023-06-26 18:50:37

功能描述

本接口用于提交一个直播流审核任务。直播流审核功能为异步任务方式，您可以通过提交直播流审核任务审核您的直播流，然后通过查询直播流审核任务接口查询审核结果。
该接口支持情况如下：

> **说明**
> - 支持审核视频直播，例如娱乐直播、线上授课等场景。
> - 支持审核语音直播，例如语音聊天房、在线会议等场景。

- 支持对直播流进行自动检测，从 OCR 文本识别、物体检测（实体、广告台标、二维码等）、图像识别及音频审核四个维度，通过深度学习技术，识别直播流中的违规内容。
- 支持设置回调地址 Callback 获取检测结果，或通过 查询直播审核任务结果 主动轮询获取审核结果详情。
- 支持识别多种违规场景，包括：色情、违法、广告等场景。
- 支持根据不同的业务场景 配置自定义的审核策略 。
- 支持用户 自定义配置风险库 ，打击自定义违规内容。

费用说明

直播流的审核分为**直播画面审核**、**直播声音审核**，其中：
- 直播画面审核：通过直播流截帧将直播流截取为多张图片进行审核。
- 直播声音审核：将直播流声音分离出来进行音频审核。
- 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个直播流**，**计2次**审核费用。
- 调用接口会产生 直播审核费用 。

> **说明**
> 直播审核不会产生截帧费用，已合并至审核费用中。

限制说明

- 直播流时长支持：**5小时以内**。
- 直播流分辨率支持：最大支持1920x1080 (1080p)。
- 接口默认并发数量：10。

版权所有：腾讯云计算（北京）有限责任公司

- 支持的直播流媒体协议：rmtp、hls、http、https 等主流协议。
- 默认并发审核路数限制：10路，超过并发审核路数，接口会返回错误。

推荐使用 SDK

对象存储 SDK 提供了完整的 Demo、自动集成、计算签名等能力。您可通过 SDK 方便快捷地调用接口。点此查看 SDK 文档。

请求

请求示例

```
POST /video/auditing HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 通过子账号使用时，需要授予相关的权限，详情请参见 数据万象授权粒度详情 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>
  <Type>live_video</Type>
  <Input>
    <Url></Url>
    <DataId></DataId>
  </Input>
  <Conf>
    <BizType></BizType>
    <Callback></Callback>
    <CallbackType></CallbackType>
```

```
  </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	直播流审核的具体配置项。	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
Type	Request	审核的任务类型，直播流审核固定为 live_video。	String	是
Input	Request	包含需要审核的直播流信息。	Container	是
Conf	Request	包含审核规则的配置信息。	Container	是
StorageConf	Request	包含直播流转存的配置信息。	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
Url	Request.Input	需要审核的直播流播放地址，例如 rtmp://example.com/live/123。	String	是
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	自定义字段，可用于辅助行为数据分析。	Container	否

Container 节点 UserInfo 的内容：

节点名称（关键字）	父节点	描述	类型	是否必选
TokenId	Request.Input.UserInfo	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	Request.Input.UserInfo	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	Request.Input.UserInfo	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	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	Request.Input.UserInfo	一般用于表示性别信息，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	一般用于表示等级信息，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	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，例如涉黄、广告、违法违规等，配置指引： 设置审核策略 。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	String	是

		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	回调地址，以 `http://` 或者 `https://` 开头的地址。	String	否
CallbackType	Request.Conf	回调片段类型，有效值：1（回调全部截帧和音频片段）、2（仅回调违规截帧和音频片段）。默认为 1。	Integer	否

Container 类型 StorageConf 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Path	Request.StorageConf	表示直播流所要转存的路径，直播流的 ts 文件和 m3u8 文件将保存在本桶该目录下。m3u8 文件保存文件名为 Path/{$JobId}.m3u8，ts 文件的保存文件名为 Path/{$JobId}-{$Realtime}.ts，其中 Realtime 为17位 `年月日时分秒毫秒` 时间。	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>
    <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	直播流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	直播流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	本次直播流审核任务的 ID。	String
State	Response.JobsDetail	直播流审核任务的状态，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	直播流审核任务的创建时间。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
POST /video/auditing HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
```

```
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Type>live_video</Type>
  <Input>
    <Url>rtmp://example.com/live/123</Url>
    <DataId>123-fdrsg-123</DataID>
  </Input>
  <Conf>
    <Callback>http://callback.com/</Callback>
    <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>
    <DataId>123-fdrsg-123</DataID>
    <JobId>vab1ca9fc8a3ed11ea834c525400863904</JobId>
    <State>Submitted</State>
    <CreationTime>2021-08-07T12:12:12+0800</CreationTime>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

取消直播审核任务

最近更新时间：2022-07-04 11:09:57

功能描述

本接口用于取消一个在进行中的直播审核任务，成功取消后将返回已终止任务的 JobID。

限制说明

默认接口请求频率限制：20次/秒。

请求

请求示例

```
POST /video/cancel_auditing/<jobId> HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>

<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 数据万象授权粒度详情 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>
    <DataId></DataId>
    <JobId></JobId>
    <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	直播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	直播流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	本次直播审核任务的 ID。	String
CreationTime	Response.JobsDetail	直播审核任务的创建时间。	String

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
POST /video/cancel_auditing/<jobId> HTTP/1.1
```

```
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

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>
    <CreationTime>2021-08-07T12:12:12+0800</CreationTime>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

查询直播审核任务结果

最近更新时间：2023-05-06 18:10:13

功能描述

本接口用于主动查询指定的直播审核任务结果。直播审核功能为异步任务方式，您可以通过提交直播审核任务审核您的直播流，然后通过查询直播审核任务接口轮询获取审核结果。

请求

请求示例

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

说明

- Authorization: Auth String（详情请参见 请求签名 文档）。
- 通过子账号使用时，需要授予相关的权限，详情请参见 数据万象授权粒度详情 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

参数名称	描述	类型	是否必选
jobId	要查询的任务ID，可通过提交审核任务的响应中获取。	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>Success</Message>
    <DataId></DataId>
    <JobId></JobId>
    <State></State>
    <CreationTime></CreationTime>
    <Type>live_video</Tpye>
    <SnapshotCount></SnapshotCount>
    <Result>1</Result>
    <Label></Label>
    <PornInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </PornInfo>
    <AdsInfo>
      <HitFlag></HitFlag>
      <Count></Count>
    </AdsInfo>
    <Snapshot>
      <Url></Url>
      <Text></Text>
      <Label></Label>
      <Result>1</Result>
      <SnapshotTime></SnapshotTime>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Label></Label>
        <Category></Category>
        <SubLabel></SubLabel>
      </PornInfo>
      <AdsInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Label></Label>
        <Category></Category>
        <SubLabel></SubLabel>
      </AdsInfo>
    </Snapshot>
    <AudioSection>
      <Url></Url>
```

```
        <OffsetTime>0</OffsetTime>
        <Duration>3000</Duration>
      <Text>哈哈哈</Text>
      <Label></Label>
      <Result>1</Result>
      <PornInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Category></Category>
        <Keywords></Keywords>
      </PornInfo>
      <AdsInfo>
        <HitFlag></HitFlag>
        <Score></Score>
        <Category></Category>
        <Keywords></Keywords>
      </AdsInfo>
    </AudioSection>
  </JobsDetail>
  <RequestId></RequestId>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	直播审核的全部结果信息。	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
JobsDetail	Response	直播审核任务的详细信息。	Container
RequestId	Response	每次请求发送时，服务端将会自动为请求生成一个 ID，遇到问题时，该 ID 能更快地协助定位问题。	String

Container 节点 JobsDetail 的内容：

节点名称（关键字）	父节点	描述	类型

Code	Response.JobsDetail	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	Response.JobsDetail	错误描述，只有 State 为 Failed 时返回。	String
DataId	Response.JobsDetail	提交任务时设置了 DataId 参数时返回，返回原始内容，长度限制为512字节。您可以使用该字段对待审核的数据进行唯一业务标识。	String
JobId	Response.JobsDetail	直播审核任务的 ID。	String
State	Response.JobsDetail	直播审核任务的状态，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	直播审核任务的创建时间。	String
SnapshotCount	Response.JobsDetail	当审核的直播包含视频流时，该字段表示直播截图的总数量。	String
Label	Response.JobsDetail	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，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**（判定为违规敏感内容），**2**（疑似敏感内容，建议人工复核）。	Integer
PornInfo	Response.JobsDetail	审核场景为**涉黄**的审核结果信息。	Container
AdsInfo	Response.JobsDetail	审核场景为**广告引导**的审核结果信息。	Container
Snapshot	Response.JobsDetail	该字段用于返回直播中画面截图审核的结果。	Container Array
AudioSection	Response.JobsDetail	当审核的直播包含音频流时，该字段用于返回直播中声音审核的结果。如果未检测音频，则不返回。	Container Array
UserInf	Response.J	用户业务字段。创建任务未设置 UserInfo 时无此字	Containe

o	obsDetail	段。	r
Type	Response.JobsDetail	审核类型，直播审核固定为 live_video。	String

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.*Info	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Count	Response.JobsDetail.*Info	命中该审核分类的截图张数。	Integer

Container 节点 Snapshot 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.Snapshot	直播截图的访问地址，您可以通过该地址查看该截图内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
SnapshotTime	Response.JobsDetail.Snapshot	该字段返回当前截图的时间戳，单位为毫秒。例如1649387157000。	Integer
Text	Response.JobsDetail.Snapshot	该字段用于返回当前截图的图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	Response.JobsDetail.Snapshot	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail.Snapshot	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感内容），**2**（疑似敏感内容，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.Snapshot	审核场景为**涉黄**的审核结果信息。	Container

AdsInfo	Response.JobsDetail.Snapshot	审核场景为**广告引导**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.Snapshot.*Info	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	Response.JobsDetail.Snapshot.*Info	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回的审核信息。 例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Label	Response.JobsDetail.Snapshot.*Info	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String
Category	Response.JobsDetail.Snapshot.*Info	该字段表示命中的审核类别。 注意：该字段可能返回空。	String
SubLabel	Response.JobsDetail.Snapshot.*Info	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标签。 注意：该字段可能返回空，表示未命中具体的子标签。	String
OcrResults	Response.JobsDetail.Snapshot.*Info	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Container Array
LibResults	Response.JobsDetail.Snapshot.*Info	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
ImageId	Response.JobsDetail.Snapshot.*Info.LibResults	该字段表示命中的风险库中的图片样本 ID。	String
Score	Response.JobsDetail.Snapshot.*Info.LibResults	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高），越高代表当前的图片越有可能命中库中的样本；如：色情 99，则表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Text	Response.JobsDetail.Snapshot.*Info.OcrResults	图片 OCR 文本识别出的具体文本内容。	String
Keywords	Response.JobsDetail.Snapshot.*Info.OcrResults	在当前审核场景下命中的关键词。	String Array
Location	Response.JobsDetail.Snapshot.*Info.OcrResults	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Container

ObjectResults 的具体内容如下：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Snapshot.*Info.ObjectResults	该标签用于返回所识别出的实体名称，例如人名。	String
Location	Response.JobsDetail.Snapshot.*Info.Obje	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关	Container

	ctResults	信息。	

Location 的具体内容如下：

节点名称（关键字）	描述	类型
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0 – 360（角度制）**，方向为**逆时针旋转**。	Float

Container 节点 AudioSection 的内容：

节点名称（关键字）	父节点	描述	类型
Url	Response.JobsDetail.AudioSection	直播声音片段的访问地址，您可以通过该地址获取该声音片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请重新发起查询请求。	String
Text	Response.JobsDetail.AudioSection	该字段用于返回当前直播声音片段的 ASR 文本识别的检测结果。	String
OffsetTime	Response.JobsDetail.Aud	该字段返回当前声音片段的时间戳，单位为毫秒，例如1649387157000。	Integer

	ioSection		
Duration	Response.JobsDetail.AudioSection	当前声音片段的长度，单位毫秒。	Integer
Label	Response.JobsDetail.Snapshot	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告，以及其他不安全或不适宜的类型。	String
Result	Response.JobsDetail.Snapshot	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感内容），**2**（疑似敏感内容，建议人工复核）。	Integer
PornInfo	Response.JobsDetail.AudioSection	审核场景为**涉黄**的审核结果信息。	Container
AdsInfo	Response.JobsDetail.AudioSection	审核场景为**广告引导**的审核结果信息。	Container

Container 节点 PornInfo，AdsInfo 的内容：

节点名称（关键字）	父节点	描述	类型
HitFlag	Response.JobsDetail.AudioSection.*Info	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	String
Score	Response.JobsDetail.Au	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息	String

	dioSection.*Info	例如：色情 99，则表明该内容非常有可能属于色情内容。	
Category	Response.JobsDetail.AudioSection.*Info	该字段表示命中的审核类别。 注意：该字段可能返回空。	String
Keywords	Response.JobsDetail.AudioSection.*Info	在当前审核场景下命中的关键词，没有时不返回。	String Array
LibResults	Response.JobsDetail.AudioSection.*Info	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	父节点	描述	类型
LibType	Response.JobsDetail.AudioSection.*Info.LibResults	命中的风险库类型，取值为1（预设风险库）和2（自定义风险库）。	Integer
LibName	Response.JobsDetail.AudioSection.*Info.LibResults	命中的风险库名称。	String
Keywords	Response.JobsDetail.AudioSection.*Info.LibResults	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

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	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	Request.Input.	一般用于表示昵称信息，长度不超过128字	St	否

	UserInfo	节。	ring	
DeviceId	Request.Input.UserInfo	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	Request.Input.UserInfo	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	Request.Input.UserInfo	一般用于表示房间号信息，长度不超过128字节。	String	否
IP	Request.Input.UserInfo	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	Request.Input.UserInfo	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	Request.Input.UserInfo	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	Request.Input.UserInfo	一般用于表示性别信息，长度不超过128字节。	String	否
Level	Request.Input.UserInfo	一般用于表示等级信息，长度不超过128字节。	String	否
Role	Request.Input.UserInfo	一般用于表示角色信息，长度不超过128字节。	Stri	否

版权所有：腾讯云计算（北京）有限责任公司

			n g	

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /video/auditing/vab1ca9fc8a3ed11ea834c525400863904 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 666
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <JobsDetail>
    <JobId>vab1ca9fc8a3ed11ea834c5254008639 04</JobId>
    <State>Success</State>
    <CreationTime>2019-07-07T12:12:12+0800</CreationTime>
    <Type>live_video</Tpye>
    <SnapshotCount>1</SnapshotCount>
    <Label>Normal</Label>
    <Result>0</Result>
    <PornInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </PornInfo>
    <AdsInfo>
      <HitFlag>0</HitFlag>
      <Count>0</Count>
    </AdsInfo>
    <Snapshot>
```

```
      <Url>http://examplebucket-1250000000.cos.ap-chongqing.myqcloud.com/0.jpg</U
      <SnapshotTime>1649387157000</SnapshotTime>
      <Text>哈哈</Text>
      <Label>Normal</Label>
      <Result>0</Result>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Label></Label>
        <SubLabel></SubLabel>
      </PornInfo>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
        <Label></Label>
        <SubLabel></SubLabel>
      </AdsInfo>
    </Snapshot>
    <AudioSection>
      <AdsInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
      </AdsInfo>
      <Duration>30000</Duration>
      <OffsetTime>1649387157000</OffsetTime>
      <Label>Normal</Label>
      <Result>0</Result>
      <PornInfo>
        <HitFlag>0</HitFlag>
        <Score>0</Score>
      </PornInfo>
      <Text>wc xx</Text>
      <Url>https://auditing-1250000000.cos.ap-guangzhou.myqcloud.com/q9MhdztF/0.m
    </AudioSection>
  </JobsDetail>
  <RequestId>xxxxxxxxxxxxxx</RequestId>
</Response>
```

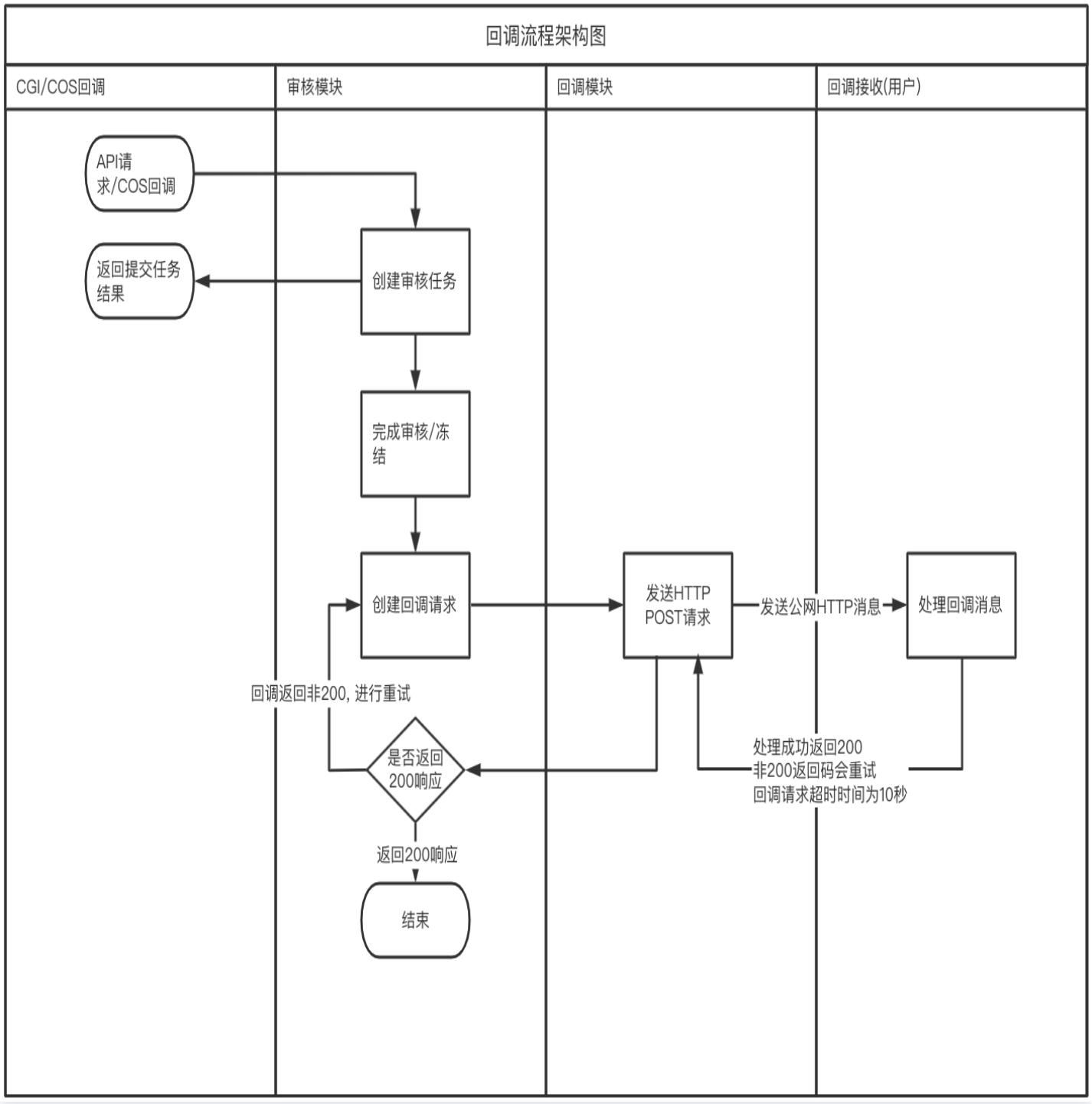
直播审核回调内容

最近更新时间：2022-10-18 14:19:20

功能描述

若您配置了审核回调地址，在直播审核结束后，后台会将审核结果以 JSON 形式回调至您的回调地址。您可以通过回调内容进行后续的处理操作。
回调内容分为简洁回调（Simple）和详细回调（Detail）。

版权所有：腾讯云计算（北京）有限责任公司



回调内容说明

简洁回调（Simple）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Simple` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：

```
{
  "code": 0,
  "data": {
     "event":"ReviewVideo",
     "forbidden_status": 0,
     "porn_info": {
        "hit_flag": 0,
        "label": "",
        "count": 0
     },
     "result": 0,
     "trace_id": "test_trace_id",
     "url": "test_url",
     "cos_headers": {
        "x-cos-meta-xx": "xx"
     }
  },
  "message": "Test request when setting callback url"
}
```


具体的数据内容如下：

参数名称	描述	类型	是否必选
code	错误码，值为0时表示审核成功，非0表示审核失败。详情请查看 内容审核错误码 。	Integer	是
message	错误描述。	String	是
data	直播审核结果的详细信息。	Object	是

data 的具体内容如下：

参数名称	描述	类型	是否必选
trace_id	请求的 ID，如果是接口创建的异步任务，则为接口返回的 JobId。	String	是
url	被审核直播的直播地址。	String	是
event	触发的事件，此处固定为 ReviewVideo。	String	是

result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：0（审核正常），1（判定为违规敏感的内容），2（疑似敏感，建议人工复核）。	Integer	是
forbidden_status	若您设置了自动冻结，该字段表示冻结状态。直播审核不涉及该字段，默认为0：未冻结。	Integer	是
cos_headers	资源上传时设置的自定义 header 内容，若未设置则不返回。	Object	否
porn_info	审核场景为**涉黄**的审核结果信息。	Object	否
ads_info	审核场景为**广告引导**的审核结果信息。	Object	否
data_id	提交任务时设置了DataId参数时返回，返回原始内容，长度限制为512字节。	String	否

xx_info 的具体内容如下：

参数名称	描述	类型	是否必选
hit_flag	用于返回该审核场景的审核结果，返回值： • 0：正常。 • 1：确认为当前场景的违规内容。 • 2：疑似为当前场景的违规内容。	Integer	是
label	命中的标签名称，目前为空。	String	是
count	命中该审核分类的敏感截图张数。	Integer	是

详细回调（Detail）

回调通知为 `HTTP` 的 `POST` 方法，含有 `X-Ci-Content-Version: Detail` 的头部。
该响应体返回为 **application/json** 数据，包含完整节点数据的内容展示如下：

```
{
 "EventName": "ReviewVideo",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "",
```

```
    "Object": "",
    "Label": "Normal",
    "Result": 0,
    "SnapshotCount": 1,
    "PornInfo": {
      "HitFlag": 0,
      "Count": 0
    },
    "AdsInfo": {
      "HitFlag": 0,
      "Count": 0
    },
    "Snapshot": [
      {
        "Url": "",
        "Text": "",
        "SnapshotTime": 0,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0,
          "Label": "",
          "SubLabel": ""
        },
        "AdsInfo": {
          "HitFlag": 0,
          "Score": 0,
          "Label": "",
          "SubLabel": ""
        }
      }
    ],
    "AudioSection": [
      {
        "Url": "",
        "Text": "",
        "OffsetTime": 0,
        "Duration": 30000,
        "Label": "Normal",
        "Result": 0,
        "PornInfo": {
          "HitFlag": 0,
          "Score": 0
        },
```

```
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0
      }
    }
  ],
  "BucketId": "examplebucket-1250000000",
  "Region": "ap-chongqing",
  "ForbidState": 0,
  "CosHeaders": {
    "x-cos-meta-id": "xxx"
  }
 }
}
```

具体的数据内容如下：

节点名称（关键字）	描述	类型
JobsDetail	任务的详细结果信息。	Object
EventName	任务的类型，为 ReviewVideo 。	String

JobsDetail 的具体内容如下：

节点名称（关键字）	描述	类型
Code	错误码，只有 State 为 Failed 时返回。详情请查看 错误码列表 。	String
Message	错误描述，只有 State 为 Failed 时返回。	String
JobId	审核任务的 ID。	String
DataId	提交任务时设置了DataId参数时返回，返回原始内容，长度限制为512字节。	String
State	直播审核任务的状态，值为 Submitted（已提交审核）、Snapshoting（截图中）、Success（审核成功）、Failed（审核失败）、Auditing（审核中）其中一个。	String
CreationTime	直播审核任务的创建时间。	String
Url	本次审核的直播链接。	String
SnapshotCount	截图的总数量。直播流状态为 Auditing 的回调中该字段为本次回调包含的截图数量。	Integer

Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object
Snapshot	该字段用于返回直播中画面截图审核的结果。	Array
AudioSection	该字段用于返回直播中声音审核的结果。如果未检测到声音，则不返回。	Array
BucketId	创建的审核任务所属的存储桶名称。	String
Region	存储桶所在的地域。	String
ForbidState	若您设置了自动冻结，该字段表示视频的冻结状态。0：未冻结，1：已被冻结，2：已转移文件。	Integer
CosHeaders	资源上传到 COS 时设置的自定义 Header 内容，如果未设置则不返回。map 结构，key 为自定义 Header 的名称，value 为内容。	Object
UserInfo	用户业务字段。创建任务时未设置 UserInfo 则无此字段。	Object
Type	审核类型，直播流审核返回 live_video。	String
ListInfo	账号黑白名单结果。	Container

PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Count	命中该审核分类的截图张数。	Integer

Snapshot 的具体内容如下：

节点名称（关键	描述	类型

字）		
Url	直播截图的访问地址，您可以通过该地址查看该截图内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
SnapshotTime	该字段返回当前截图的时间戳，单位为毫秒。例如1649387157000。	Integer
Text	该字段用于返回当前截图的图片 OCR 文本识别的检测结果，仅在审核策略开启文本内容检测时返回。	String
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	该字段表示本次判定的审核结果，您可以根据该结果，进行后续的操作；建议您按照业务所需，对不同的审核结果进行相应处理。 有效值：**0**（审核正常），**1**（判定为违规敏感文件），**2**（疑似敏感，建议人工复核）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

Snapshot 中 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息。 例如：色情 99，则表明该内容非常有可能属于色情内容。	Integer
Label	该字段表示该截图的综合结果标签（可能为 SubLabel，可能为人物名字等）。	String
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。 注意：该字段可能返回空。	String
SubLa	该字段表示审核命中的具体子标签，例如：Porn 下的 SexBehavior 子标	String

bel	签。 注意：该字段可能返回空，表示未命中具体的子标签。	
OcrResults	该字段表示 OCR 文本识别的详细检测结果，包括文本坐标信息、文本识别结果等信息，有相关违规内容时返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。 注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
ImageId	该字段表示命中的风险库中的图片样本 ID。	String
Score	该字段用于返回当前标签下的置信度，取值范围：0（置信度最低）-100（置信度最高 ），越高代表当前的图片越有可能命中库中的样本；如：色情 99，则表明该数据非常有可能命中库中的色情样本。	Integer

OcrResults 的具体内容如下：

节点名称（关键字）	描述	类型
Text	图片 OCR 文本识别出的具体文本内容。	String
Keywords	在当前审核场景下命中的关键词。	Array
Location	该参数用于返回 OCR 检测框在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位识别文字的相关信息。	Object

ObjectResults 的具体内容如下：

节点名称（关键字）	描述	类型
Name	该标签用于返回所识别出的实体名称，如人名。	String
Location	该参数用于返回检测结果在图片中的位置（左上角 xy 坐标、长宽、旋转角度），以方便快速定位相关信息。	Object

Location 的具体内容如下：

节点名称	描述	类型

（关键字）		
X	该参数用于返回检测框**左上角位置的横坐标**（x）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Y	该参数用于返回检测框**左上角位置的纵坐标**（y）所在的像素位置，结合剩余参数可唯一确定检测框的大小和位置。	Float
Height	该参数用于返回**检测框的高度**（由左上角出发在 y 轴向下延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Width	该参数用于返回**检测框的宽度**（由左上角出发在 x 轴向右延伸的长度），结合剩余参数可唯一确定检测框的大小和位置。	Float
Rotate	该参数用于返回**检测框的旋转角度**，该参数结合 X 和 Y 两个坐标参数可唯一确定检测框的具体位置；取值：**0 - 360（角度制）**，方向为**逆时针旋转**。	Float

AudioSection 的具体内容如下：

节点名称（关键字）	描述	类型
Url	直播声音片段的访问地址，您可以通过该地址获取该声音片段的内容，地址格式为标准 URL 格式。 注意：每次查看数据的有效期为2小时，2小时后如还需查看，请发起查询请求。	String
Text	该字段用于返回当前直播声音片段的 ASR 文本识别的检测结果。	String
OffsetTime	该字段返回当前声音片段的时间戳，单位为毫秒，例如1649387157000。	Integer
Duration	当前声音片段的时长，单位毫秒。	Integer
Label	该字段用于返回检测结果中所对应的**优先级最高的恶意标签**，表示模型推荐的审核结果，建议您按照业务所需，对不同违规类型与建议值进行处理。 返回值：**Normal**：正常，**Porn**：色情，**Ads**：广告。	String
Result	供参考的识别结果，0（正常），1（敏感），2（疑似敏感）。	Integer
PornInfo	审核场景为**涉黄**的审核结果信息。	Object
AdsInfo	审核场景为**广告引导**的审核结果信息。	Object

AudioSection 中 PornInfo，AdsInfo 的具体内容如下：

节点名称（关键字）	描述	类型
HitFlag	是否命中该审核分类，0表示未命中，1表示命中，2表示疑似。	Integer
Score	该字段表示审核结果命中审核信息的置信度，取值范围：0（**置信度最低**）-100（**置信度最高**），越高代表该内容越有可能属于当前返回审核信息例如：色情99，则表明该内容非常有可能属于色情内容。	Integer
Category	表示命中的具体审核类别。例如 Sexy，表示色情标签中的性感类别。该字段可能为空，表示未命中或暂无相关的类别。 注意：该字段可能返回空。	String
Keywords	在当前审核场景下命中的关键词，没有时不返回。	Array
LibResults	该字段用于返回基于风险库识别的结果。注意：未命中风险库中样本时，此字段不返回。	Container Array

Container 节点 LibResults 的内容：

节点名称（关键字）	描述	类型
LibType	命中的风险库类型，取值为1（预设黑白库）和2（自定义风险库）。	Integer
LibName	命中的风险库名称。	String
Keywords	命中的库中关键词。该参数可能会有多个返回值，代表命中的多个关键词。	String Array

UserInfo 的内容：

节点名称（关键字）	描述	类型	是否必选
TokenId	一般用于表示账号信息，长度不超过128字节。	String	否
Nickname	一般用于表示昵称信息，长度不超过128字节。	String	否
DeviceId	一般用于表示设备信息，长度不超过128字节。	String	否
AppId	一般用于表示 App 的唯一标识，长度不超过128字节。	String	否
Room	一般用于表示房间号信息，长度不超过128字节。	String	否

版权所有：腾讯云计算（北京）有限责任公司

IP	一般用于表示 IP 地址信息，长度不超过128字节。	String	否
Type	一般用于表示业务类型，长度不超过128字节。	String	否
ReceiveTokenId	一般用于表示接收消息的用户账号，长度不超过128字节。	String	否
Gender	一般用于表示性别信息，长度不超过128字节。	String	否
Level	一般用于表示等级信息，长度不超过128字节。	String	否
Role	一般用于表示角色信息，长度不超过128字节。	String	否

Container 节点 ListInfo 的内容：

节点名称（关键字）	描述	类型
ListResults	命中的所有名单结果。	Container Array

Container 节点 ListResults 的内容：

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

实际案例

案例一：简洁回调（Simple）


```
{
  "code": 0,
  "data": {
     "event":"ReviewVideo",
     "forbidden_status": 0,
     "porn_info": {
        "hit_flag": 0,
        "label": "",
        "count": 0
     },
     "result": 0,
     "trace_id": "vxzt90jl2dfscxxxxxxxxxxxxxxxx",
     "url": "https://66665.livepush.myqcloud.com/video.flv",
```

```
    "cos_headers": {
      "x-cos-meta-id": "666666"
    }
  },
  "message": "success"
}
```

案例二：详细回调（Detail）

```
{
 "EventName": "ReviewVideo",
 "JobsDetail": {
  "JobId": "xxxxxx",
  "State": "Success",
  "CreationTime": "2021-08-10T21:01:10+08:00",
  "Url": "https://66665.livepush.myqcloud.com/video.flv",
  "Label": "Normal",
  "Result": 0,
  "SnapshotCount": 1,
  "PornInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "AdsInfo": {
   "HitFlag": 0,
   "Count": 0
  },
  "Snapshot": [
   {
    "Url": "https://video-1250000000.cos.ap-chongqing.myqcloud.com/test/0.jpg",
    "Text": "",
    "SnapshotTime": 41,
    "Label": "Normal",
    "Result": 0,
    "PornInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Label": "",
     "SubLabel": ""
    },
    "AdsInfo": {
     "HitFlag": 0,
     "Score": 0,
     "Label": "",
```

```
        "SubLabel": ""
      }
    }
  ],
  "AudioSection": [
    {
      "Url": "https://audio-1250000000.cos.ap-guangzhou.myqcloud.com/0.mp3",
      "Text": "",
      "OffsetTime": 0,
      "Duration": 30000,
      "Label": "Normal",
      "Result": 0,
      "PornInfo": {
        "HitFlag": 0,
        "Score": 0
      },
      "AdsInfo": {
        "HitFlag": 0,
        "Score": 0
      }
    }
  ],
  "BucketId": "examplebucket-1250000000",
  "Region": "ap-chongqing",
  "ForbidState": 0,
  "CosHeaders": {
     "x-cos-meta-id": "xxxx"
  }
 }
}
```

AI 内容识别
异步服务和队列配置
开通AI 内容识别（异步）服务

最近更新时间：2023-08-01 16:46:25

功能描述

本接口用于开通AI 内容识别（异步）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>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <AiBucket>
    <BucketId>test-1234567890</BucketId>
    <Name>test-1234567890</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2022-08-09T16:23:11+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

实际案例

请求

```
POST /ai_bucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
        <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
    </AiBucket>
</Response>
```

查询AI 内容识别（异步）服务

最近更新时间：2023-08-01 16:46:25

功能描述

本接口用于查询已经开通AI 内容识别（异步）服务的存储桶。

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	默认值	是否必填
regions	地域信息，例如 ap-shanghai、ap-beijing，若查询多个地域以“,”分隔字符串，详情请参见 地域与域名	无	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	无	否
bucketNam	存储桶名称前缀，前缀搜索	无	否

e			
pageNumber	第几页	1	否
pageSize	每页个数，大于0且小于等于100的整数	10	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>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <AiBucketList>
    <BucketId>examplebucket-1250000000</BucketId>
    <Name>examplebucket-1250000000</Name>
    <Region>ap-beijing</Region>
    <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
  </AiBucketList>
  <AiBucketList>
    <BucketId>examplebucket-1-1250000000</BucketId>
    <Name>examplebucket-1-1250000000</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2020-11-16T14:13:15+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	开通AI 内容识别（异步）的 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	AI 内容识别（异步） Bucket 详情	Container 数组

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 HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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 2021 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>
    <AiBucketList>
        <BucketId>examplebucket-1250000000</BucketId>
        <Name>examplebucket-1250000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </FileBucketList>
    <FileBucketList>
        <BucketId>examplebucket-1-1250000000</BucketId>
        <Name>examplebucket-1-1250000000</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
    </AiBucketList>
</Response>
```

关闭AI 内容识别（异步）服务

最近更新时间：2023-08-01 16:46:27

功能描述

本接口用于关闭AI 内容识别（异步）服务并删除队列。

请求

请求示例

```
DELETE /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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
  <BucketName>test-1234567890</BucketName>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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
BucketName	Response	存储桶名称	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 /ai_bucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
    <BucketName>test-1234567890</BucketName>
```

```
</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（异步）队列

最近更新时间：2023-08-01 16:46:25

功能描述

本接口用于查询AI 内容识别（异步）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	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
  <TotalCount>1</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <QueueList>
    <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</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>
    <MaxSize>10000</MaxSize>
    <MaxConcurrent>10</MaxConcurrent>
    <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
    <CreateTime>2022-08-09T16:23:11+0800</CreateTime>
    <Category>AIProcess</Category>
  </QueueList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	队列总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
QueueList	Response	队列数组	Container 数组
NonExistPIDs	Response	不存在的队列 ID 列表	String 数组

Container 节点 QueueList 的内容：

节点名称（关键字）	父节点	描述	类型
QueueId	Response.QueueList	队列 ID	String
Name	Response.QueueList	队列名字	String
State	Response.QueueList	当前状态，Active 或者 Paused	String
NotifyConfig	Response.QueueList	回调配置	Container
MaxSize	Response.QueueList	队列最大长度	Int
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

版权所有：腾讯云计算（北京）有限责任公司

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	String
Type	Response.QueueList.NotifyConfig	回调类型	String
Event	Response.QueueList.NotifyConfig	回调事件	String
ResultFormat	Response.QueueList.NotifyConfig	回调类型	String
MqMode	Response.QueueList.NotifyConfig	TDMQ 使用模式	String
MqRegion	Response.QueueList.NotifyConfig	TDMQ 所属园区	String
MqName	Response.QueueList.NotifyConfig	TDMQ 主题名称	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 /ai_queue?queueIds=p4a17eeea29334bf499b7e20e2fbfd99d,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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: NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85

<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</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>
        <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>
```

更新AI 内容识别（异步）队列

最近更新时间：2023-08-01 16:46:25

功能描述

本接口用于更新AI 内容识别（异步）的队列。

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	默认值	是否必选
Name	Request	队列名称，仅支持中文、英文、数字、_、-和*，长度不超过 128	String	无	是
State	Request	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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	无	是
NotifyConfig	Request	回调配置	Container	无	是

Container 节点 NotifyConfig 的内容：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
State	Request.NotifyConfig	回调开关 • Off • On	String	Off	否
Event	Request.NotifyConfig	回调事件 • TaskFinish：任务完成 • WorkflowFinish：工作流完成	String	无	当 State=On 时，必选
ResultFormat	Request.NotifyConfig	回调格式 • XML • JSON	String	XML	否

Type	Request.NotifyConfig	回调类型 • Url • TDMQ	String	无	当 State=On 时，必选
Url	Request.NotifyConfig	回调地址，不能为内网地址。	String	无	当 State=On，且 Type=Url 时，必选
MqMode	Request.NotifyConfig	TDMQ 使用模式 • Topic：主题订阅 • Queue: 队列服务	String	无	当 State=On，且 Type=TDMQ 时，必选
MqRegion	Request.NotifyConfig	TDMQ 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	无	当 State=On，且 Type=TDMQ 时，必选
MqName	Request.NotifyConfig	TDMQ 主题名称	String	无	当 State=On，且 Type=TDMQ 时，必选

响应

响应头

此接口仅返回公共响应头部，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-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>
        <MqMode/>
        <MqName/>
        <MqRegion/>
      </NotifyConfig>
      <MaxSize>10000</MaxSize>
      <MaxConcurrent>10</MaxConcurrent>
      <Category>AIProcess</Category>
      <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/p4a17eeea29334bf499b7e20e2fbfd99d HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>p4a17eeea29334bf499b7e20e2fbfd99d</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>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <Category>AIProcess</Category>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

图片上色

最近更新时间：2023-06-19 16:47:02

功能描述

腾讯云数据万象通过 **AIImageColoring** 接口对黑白图像进行上色。

> **使用限制：**
> - 待处理的图片，原图大小不超过5MB，分辨率大于32 x 32像素。
> - 接口携带签名，下载时处理，请求参数需要记入签名内

1. 下载时处理

请求

- 原图存储在COS：

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

- 原图来自其他链接：

```
GET /?ci-process=AIImageColoring&detect-url=<detect-url> 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：AIImageCrop。

参数	含义	类型	是否必选
ObjectKey	对象文件名，例如：folder/document.jpg。	String	否
ci-proce	数据万象处理能力，图片上色参固定为AIImageColoring。	String	是

ss			
detect-url	待上色图片url，需要进行urlencode，与ObjectKey二选其一，如果同时存在，则默认以ObjectKey为准	String	否

响应

```
HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 98645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

[pic-content]
```

处理后的图片直接返回至响应体。

2.上传时处理

请求语法

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

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	处理参数，固定为：ci-process=AIImageColoring

请求体

该请求body内容为待上色图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	图片路径	String
ETag	OriginalInfo	原图的Etag值	

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	上色后图片文件名	String
Location	Object	上色后的图片链接	String
ETag	Object	上色后图片的ETag值	String
Size	Object	图片大小	Int

示例

请求：

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

版权所有：腾讯云计算（北京）有限责任公司

```
Date: GMT Date
Authorization: Auth String
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
     "fileid": "exampleobject",
     "rule": "ci-process=AIImageColoring"
  }]
}
```

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

3.云上数据处理

请求语法

```
POST  /<ObjectKey> HTTP/1.1
```

```
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>
```

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	处理参数，固定为：ci-process=AIImageColoring

请求体

空

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	图片路径	String
ETag	OriginalInfo	原图ETag	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型

Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	上色后图片文件名	String
Location	Object	上色后的图片链接	String
ETag	Object	上色后图片的ETag值	String
Size	Object	上色后图片大小	Int

示例

请求：

```
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:
{
 "rules": [{
    "fileid": "exampleobject",
    "rule": "ci-process=AIImageColoring"
 }]
}
```

处理后的图片存至存储桶，并以fileid字段的值作为对象名称。

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
   <OriginalInfo>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

图像超分

最近更新时间：2023-06-19 16:47:02

功能描述

腾讯云数据万象通过 AISuperResolution 接口对图像进行超分辨率处理，当前默认超分为宽高的2倍。

使用限制：

- 待处理的图片，原图大小不超过5MB，分辨率不得超过 1920px * 1080px。
- 支持处理JPG PNG JPEG格式图片。

1. 下载时处理

请求

- 原图存储在COS：

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

- 原图来自其他链接：

```
GET /?ci-process=AISuperResolution&detect-url=<detect-url> 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：AIImageCrop。

参数	含义	类型	是否必选
Object Key	对象文件名，例如：folder/document.jpg。	String	否
ci-proces	数据万象处理能力，只能裁剪参固定为AISuperResolution。	String	是

s			
detect-url	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 detect-url 时，后台会默认处理 ObjectKey ，填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey，detect-url 示例：http://www.example.com/abc.jpg ，需要进行 UrlEncode，处理后为 http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg。	String	否

响应

```
HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 98645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

[pic-content]
```

处理后的图片直接返回至响应体。

2.上传时处理

请求语法

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

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为 <ObjectKey>
rule	String	是	处理参数，固定为：ci-process=AISuperResolution

请求体

该请求body内容为待上色图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	原图路径	String
ETag	OriginalInfo	原图ETag值	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	超分后图片文件名	String
Location	Object	超分后的图片链接	String
ETag	Object	超分后图片的ETag值	String

版权所有：腾讯云计算（北京）有限责任公司

Size	Object	超分后图片大小	Int

示例

请求

```
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": "ci-process=AISuperResolution"
  }]
}
```

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
```

```
</UploadResult>
```

3.云上数据处理

请求语法

```
POST /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>
```

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	处理参数，固定为：ci-process=AISuperResolution

请求体

该请求body内容为待上色图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String

Location	OriginalInfo	原图路径	String
ETag	OriginalInfo	原图ETag值	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	超分后图片文件名	String
Location	Object	超分后的图片链接	String
ETag	Object	超分后图片的ETag值	String
Size	Object	图片大小	Int

示例

请求：

```
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:
{
 "rules": [{
    "fileid": "exampleobject",
    "rule": "ci-process=AISuperResolution"
 }]
}
```

处理后的图片存至存储桶，并以fileid字段的值作为对象名称。

响应：

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

版权所有：腾讯云计算（北京）有限责任公司

```
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

图像增强

最近更新时间：2023-06-19 16:47:02

功能描述

腾讯云数据万象通过 AIEnhanceImage 接口对图像进行增强处理。

> **注意**
> 使用限制：待处理的图片，原图大小不超过5MB。

1. 下载时处理

请求

- 原图存储在COS：

```
GET /<ObjectKey>?ci-process=AIEnhanceImage&denoise=<denoise>&sharpen=<sha
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

- 原图来自其他链接：

```
GET /<ObjectKey>?ci-process=AIEnhanceImage&denoise=<denoise>&sharpen=<sha
Host: <BucketName-APPID>.pic.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

处理参数说明

操作名称：AIImageCrop。

参数	含义	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，只能裁剪参固定为 AIEnhanceImage。	String	是

denoise	去噪强度值，取值范围为 0 – 5 之间的整数，值为 0 时不进行去噪操作，默认值为3。	Integer	否
sharpen	锐化强度值，取值范围为 0 – 5 之间的整数，值为 0 时不进行锐化操作，默认值为3。	Integer	否
detect-url	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 `detect-url` 时，后台会默认处理 `ObjectKey` ，填写了 `detect-url` 时，后台会处理 `detect-url` 链接，无需再填写 `ObjectKey` ，`detect-url` 示例：`http://www.example.com/abc.jpg` ，需要进行 UrlEncode，处理后为 `http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg`	String	否
ignore-error	当此参数为1时，针对文件过大等导致处理失败的场景，会直接返回原图而不报错。	Integer	-

响应

```
HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 98645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

[pic-content]
```

处理后的图片直接返回至响应体。

2.上传时处理

请求

```
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": "ci-process=AIEnhanceImage"
  }]
}
```

处理后的图片存至存储桶，并以 fileid 字段的值作为对象名称。

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

3.云上数据处理

请求

```
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:
```

```
{
 "rules": [{
    "fileid": "exampleobject",
    "rule": "ci-process=AIEnhanceImage"
 }]
}
```

处理后的图片存至存储桶，并以 fileid 字段的值作为对象名称。

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

版权所有：腾讯云计算（北京）有限责任公司

图像智能裁剪

最近更新时间：2023-06-19 16:47:02

功能描述

腾讯云数据万象通过 **AIImageCrop** 接口对图像进行智能裁剪，支持持久化、云上处理及下载时处理。

使用限制：

- 待处理的图片，原图大小不超过5MB。
- 支持处理PNG JPG JPEG格式图片。
- 输入图片分辨率需大于32 x 32，此功能需携带签名。

1. 下载时处理

请求

- 原图存储在COS：

```
GET /<ObjectKey>?ci-process=AIImageCrop&width=<width>&height=<height>&fixe
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

- 原图来自其他链接：

```
GET /<ObjectKey>?ci-process=AIImageCrop&width=<width>&height=<height>&fixe
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

处理参数说明

操作名称：AIImageCrop。

参数	含义	类型	是否必选
ObjectKey	对象文件名，例如：folder/document.jpg	String	是
ci-	数据万象处理能力，智能裁剪固定为AIImageCrop	Stri	是

process		ng	
detect-url	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 detect-url 时，后台会默认处理 ObjectKey ，填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey detect-url 示例： `http://www.example.com/abc.jpg` ，需要进行 UrlEncode，处理后为 `http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg`	String	否
width	需要裁剪区域的宽度，与height共同组成所需裁剪的图片宽高比例；输入数字请大于0、小于图片宽度的像素值	Integer	是
height	需要裁剪区域的高度，与width共同组成所需裁剪的图片宽高比例；输入数字请大于0、小于图片高度的像素值；width : height建议取值在[1, 2.5]之间，超过这个范围可能会影响效果	Integer	是
fixed	是否严格按照 width 和 height 的值进行输出。取值为0时，宽高比例（width : height）会简化为最简分数，即如果width输入10、height输入20，会简化为1：2；取值为1时，输出图片的宽度等于width，高度等于height；默认值为0	Integer	否
ignore-error	当此参数为1时，针对文件过大等导致处理失败的场景，会直接返回原图而不报错	Integer	否

响应

```
HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 98645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

[pic-content]
```

处理后的图片直接返回至响应体。

2.上传时处理

请求语法

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

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	处理参数

rule包含的处理参数为：

参数名称	类型	是否必选	描述
ci-process	String	是	数据万象处理能力，智能裁剪参固定为 AIImageCrop
width	Integer	是	需要裁剪区域的宽度，与height共同组成所需裁剪的图片宽高比例；输入数字请大于0、小于图片宽度的像素值
height	Integer	是	需要裁剪区域的高度，与 width 共同组成所需裁剪的图片宽高比例；输入数字请大于0、小于图片高度的像素值；width : height建议取值在[1, 2.5]之间，超过这个范围可能会影响效果；
fixed	Integer	否	是否严格按照 width 和 height 的值进行输出。取值为0时，宽高比例（width : height）会简化为最简分数，即如果 width 输入10、height 输入20，会简化为1：2；取值为1时，输出图片的宽度等于 width，高度等于 height；默认值为0

请求体

该请求body内容为待裁剪图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	图片路径	String
Etag	OriginalInfo	原图ETag值	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	裁剪后图片文件名	String
Location	Object	裁剪后的图片链接	String
ETag	Object	裁剪后图片的ETag值	String
Size	Object	裁剪后图片大小	Int

示例

请求：

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

版权所有：腾讯云计算（北京）有限责任公司

```
Pic-Operations:
{
  "is_pic_info": 1,
  "rules": [{
     "fileid": "exampleobject",
     "rule": "ci-process=AIImageCrop&width=<width>&height=<height>&fixed=<fixed>
  }]
}
```

处理后的图片存至存储桶，并以fileid字段的值作为对象名称。

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

3.云上数据处理

请求语法

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

```
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>
```

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>

rule	String	是	处理参数

rule包含的处理参数为：

参数名称	类型	是否必选	描述
ci-process	String	是	数据万象处理能力，智能裁剪参固定为AIImageCrop。
width	Integer	是	需要裁剪区域的宽度，与height共同组成所需裁剪的图片宽高比例；输入数字请大于0、小于图片宽度的像素值
height	Integer	是	需要裁剪区域的高度，与width共同组成所需裁剪的图片宽高比例；输入数字请大于0、小于图片高度的像素值；width : height建议取值在[1, 2.5]之间，超过这个范围可能会影响效果；
fixed	Integer	否	是否严格按照 width 和 height 的值进行输出。取值为0时，宽高比例（width : height）会简化为最简分数，即如果 width 输入10、height 输入20，会简化为1：2；取值为1时，输出图片的宽度等于 width，高度等于 height；默认值为0

请求体

空

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container

版权所有：腾讯云计算（北京）有限责任公司

ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	图片路径	String
Etag	OriginalInfo	原图ETag值	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	裁剪后图片文件名	String
Location	Object	裁剪后的图片链接	String
ETag	Object	裁剪后图片的ETag值	String
Size	Object	裁剪后图片大小	Int

示例

请求：

```
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:
{
 "rules": [{
    "fileid": "exampleobject",
    "rule": "ci-process=AIImageCrop&width=<width>&height=<height>&fixed=<fixed>
 }]
```

版权所有：腾讯云计算（北京）有限责任公司

```
}
```

处理后的图片存至存储桶，并以fileid字段的值作为对象名称。

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>ObjectKey</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>exampleobject</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

实时文字翻译

最近更新时间：2023-06-19 16:47:03

文本翻译

腾讯云数据万象通过 **AutoTranslationBlock** 接口对文字块进行翻译，请求时需要携带签名。

请求

请求语法

```
GET /?ci-process=AutoTranslationBlock&InputText=<InputText>&SourceLang=<SourceLa
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）。

请求内容

参数名称	描述	类型	是否必选
ci-process	数据万象处理能力，文本块翻译固定为 AutoTranslationBlock。	String	是
InputText	待翻译的文本	String	是
SourceLang	输入语言，如 "zh"	String	是
TargetLang	输出语言，如 "en"	String	是
TextDomain	文本所属业务领域，如: "ecommerce", //缺省值为 general	String	否
TextStyle	文本类型，如: "title", //缺省值为 sentence	String	否

SourceLang、TargetLang 可填参数参考这个：

中文（代码zh）与下列语言之间的双向互译：

英语 en 阿拉伯语 ar 德语 de 西班牙语 es 法语 fr 印尼语 id 意大利语 it / 日语 ja葡萄牙语 pt 俄语 ru 韩语 ko 高棉语 km 老挝语 lo

英文（代码en）与下列语言之间的双向互译：

中文 zh 阿拉伯语 ar 德语 de 西班牙语 es 法语 fr 印尼语 id 意大利语 it / 日语 ja葡萄牙语 pt 俄语 ru 韩语 ko 高棉语 km 老挝语 lo其中，中文除了常规的简体中文（zh）外，还有繁体中文（zh-tr）、中国香港繁体中文（zh-hk）、中国台湾繁体中文（zh-tw）。

我们已开放了数个专业场景，通过text_domain和text_style参数获得相应能力：

(1) 电商：text_domain=ecommerce
(2) 电商标题: text_domain=ecommerce；text_style=title
(3) 网络文学：text_domain=net-literature

请求头

此接口仅使用公共请求头部，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，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
TranslationResult	无	翻译结果	StringResponse 的内容

示例

请求

```
GET /?ci-process=AutoTranslationBlock&InputText=%E6%88%91%E6%98%AF%E6%B5%
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: xxxxx
Authorization:XXXXXXXXXXXX
```

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: 414641
Date: xxxxxx
Server: tencent-ci
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<TranslationResult>I&amp;apos;m testing.</TranslationResult>
```

图像修复

最近更新时间：2023-07-06 18:24:46

功能概述

腾讯云数据万象通过 ImageRepair 接　能检测并擦除图片中常见的标志,并对擦除部分进行智能修复，此功能需携带签名。

> **说明：**
> 当前图像修复功能的图片限制为：图片格式支持 PNG、JPEG、JPG，图片大小不超过5MB，图片宽高大于32像素 x 32像素。

1.下载时处理

请求

- 原图存储在COS：

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

- 原图来自其他链接：

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

2.上传时处理

请求语法

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

版权所有：腾讯云计算（北京）有限责任公司

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	处理参数，ci-process=ImageRepair&MaskPic=<MaskPic>&MaskPoly=<MaskPoly>，具体参数描述见下表

图像修复相关参数如下：

参数名称	描述	类型	是否必选
ci-process	数据万象处理能　，图像标志擦除固定为 ImageRepair	String	是
MaskPic	遮罩（白色区域为需要去除的水印位置）图片地址，私有图片需携带签名，需要经过 URL 安全的 Base64 编码 。例如，遮罩图片为 `http://examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/shuiyin_2.png` ，则该处编码后的字符串为 `aHR0cDovL2V4YW1wbGVidWNrZXQtMTI1MDAwMDAwMC5jb3MuYXAtZ3Vhbmd6aG91Lm15cWNsb3VkLmNvbS9zaHVpeWluXzIucG5n`	String	否
MaskPoly	例如：[[[608, 794], [1024, 794], [1024, 842], [608, 842]], [[1295, 62], [1295, 30], [1597, 32],[1597,64]]]，顺时针输　多边形的每个点的坐标,每个多边形: [[x1, y1], [x2, y2]...] , 形式为三维矩阵（多个多边形：[多边形1,多边形2]）或　维矩阵（单个多边形），且需要经过 URL 安全的 Base64 编码。MaskPoly同时与MaskPic填写时，优先采　MaskPic的值	String	否

请求体

该请求body内容为待修复图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	图片路径	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	修复后图片文件名	String
Location	Object	修复后的图片链接	String
ETag	Object	修复后图片的ETag值	String
Size	Object	图片大小	Int

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: {"rules":[{"fileid":"test.jpg","rule":"ci-process=ImageRepair&MaskPic=aH
Content-Length: 64

[Image 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>456.jpeg</Key>
                <Location>examplebucket-1250000000.cos.ap-chongqing.myqcloud.com
            </OriginalInfo>
        <ProcessResults>
            <Object>
                <Key>test.jpg</Key>
                <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/test.jp
                <ETag>"eaa4e3d8fd498bbc63be3b71c46b9c61f23e3f0c"</ETag>
            </Object>
        </ProcessResults>
</UploadResult>
```

3.云上数据处理

请求语法

```
POST  /<ObjectKey>& HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Pic-Operations: <PicOperations>
```

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

请求

参数名称	描述	类型
ObjectKey	待修复图片名称，需为存储在COS上的文件名，例如 folder/sample.jpg	String

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	处理参数，ci-process=ImageRepair&MaskPic=<MaskPic>&MaskPoly=<MaskPoly>，具体参数描述见下表

参数名称	描述	类型	是否必选
ci-process	数据万象处理能 ，图像标志擦除固定为 ImageRepair	String	是

MaskPic	遮罩（白色区域为需要去除的水印位置）图片地址，私有图片需携带签名，需要经过 URL 安全的 Base64 编码 。例如，遮罩图片为 `http://examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/shuiyin_2.png` ，则该处编码后的字符串为 `aHR0cDovL2V4YW1wbGVidWNrZXQtMTI1MDAwMDAwMC5jb3MuYXAtZ3Vhbmd6aG91Lm15cWNsb3VkLmNvbS9zaHVpeWluXzIucG5n`	String	否
MaskPoly	例如： [[[608, 794], [1024, 794], [1024, 842], [608, 842]],[[1295, 62], [1295, 30], [1597, 32],[1597,64]]] ，顺时针输　多边形的每个点的坐标,每个多边形: [[x1, y1], [x2, y2]...] , 形式为三维矩阵（多个多边形： [多边形1,多边形2] ）或　维矩阵（单个多边形），且需要经过 URL 安全的 Base64 编码。MaskPoly同时与MaskPic填写时，优先采MaskPic的值	String	否

请求体

该请求没有请求体

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

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

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	修复后图片文件名	String
Location	Object	修复后的图片链接	String
ETag	Object	修复后图片的ETag值	String
Size	Object	图片大小	Int

示例

请求

```
POST /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: {"rules":[{"fileid":"bigpic_process.jpg","rule":"ci-process=ImageRepair&M
Content-Length: 0
```

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>456.jpeg</Key>
```

```
    <Location>examplebucket-1250000000.cos.ap-chongqing.myqcloud.com/456.jpeg
  </OriginalInfo>
  <ProcessResults>
    <Object>
      <Key>bigpic_process.jpeg</Key>
      <Location>examplebucket-1250000000.cos.ap-chongqing.myqcloud.com/bigpic_
      <Format/>
      <Size>288132</Size>
      <ETag>"fa2a3d6081c1037f154401a64eec4b19"</ETag>
    </Object>
  </ProcessResults>
</UploadResult>
```

Logo 识别

最近更新时间：2023-06-19 16:47:03

功能概述

腾讯云数据万象通过 **RecognizeLogo** 接口实现对图片内电商 Logo 的识别，返回图片中 Logo 的名称、坐标、置信度分值。，返回图片中Logo的名称、坐标、置信度分值。图片Logo识别请求包属于 GET 请求，请求时需要携带签名。

> **说明：**
> 当前Logo识别的图片限制为：图片格式支持 PNG、JPEG、JPG，图片大小不超过5MB，图片宽高大于32像素 x 32像素，小于7680像素 x 7680像素。

请求语法

```
GET /?ci-process=RecognizeLogo&detect-url=<detect-url> 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）。

请求内容

参数名称	描述	类型	是否必选
ci-process	数据万象处理能力，Logo识别固定为RecognizeLogo	String	是
detect-url	待检查图片url，需要进行urlencode	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

该响应体返回为 application/xml 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
RecognitionResult	无	保存Logo识别结果的容器	Container

Response 的内容：

节点名称（关键字）	父节点	描述	类型
Status	RecognitionResult	Logo识别结果。0表示未识别到，1表示识别到	Int
LogoInfo	RecognitionResult	Logo识别结果，可能有多个	Container

LogoInfo 节点内容：

节点名称（关键字）	父节点	描述	类型
Name	LogoInfo	Logo的名称	String
Sorce	LogoInfo	Logo的置信度，取值范围为[0-100]。值越高概率越大。	Int
Location	LogoInfo	图中识别到Logo的坐标	Container

Location 节点内容：

节点名称（关键字）	父节点	描述	类型
Point	Location	Logo坐标点（X坐标,Y坐标）	String

示例

请求

```
GET /?ci-process=RecognizeLogo&detect-url=xxx HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: xxxxx
Authorization:XXXXXXXXXXXX
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 414641
Date: xxxxxx
Server: tencent-ci
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<RecognitionResult>
  <Status>1</Status>
  <LogoInfo>
    <Name>xxxxxxxx</Name>
    <Sorce>99</Sorce>
    <Location>
      <Point>100,100</Point>
      <Point>100,200</Point>
      <Point>200,200</Point>
      <Point>200,100</Point>
    </Location>
  </LogoInfo>
  <LogoInfo>
    <Name>xxxxxxxx</Name>
    <Sorce>95</Sorce>
    <Location>
      <Point>1000,1000</Point>
      <Point>1000,2000</Point>
      <Point>2000,2000</Point>
      <Point>2000,1000</Point>
    </Location>
  </LogoInfo>
</RecognitionResult>
```

版权所有：腾讯云计算（北京）有限责任公司

图片二维码识别

最近更新时间：2023-08-17 17:55:43

功能描述

数据万象二维码识别功能可识别图片中有效二维码的位置及内容，输出图像中二维码包含的文本信息（每个二维码对应的 URL 或文本），并可对识别出的二维码添加马赛克。

请求一：上传时识别

图片上传时识别二维码的请求包与 COS 简单上传文件接口一致，只需在请求包头部增加图片处理参数 Pic-Operations，并将请求 Host 更改为对象存储域名即可。

请求

```
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，详细请参见 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。

请求头

此接口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

此接口无请求参数。

请求体

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	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	In	否

_info		t	
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，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	处理结果的文件路径名称，例如以 `/` 开头，则存入指定文件夹中，否则存入与原图文件相同的目录位置	String	否
rule	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， 例如样式名为 `test` ，则 rule 字段为 `style/test`	String	是

使用二维码识别功能需在 rule 中添加二维码识别参数（QRcode），相关内容如下：

```
QRcode/cover/<mode>
```

示例参数说明如下：

参数	描述	类型	是否必选
cover	二维码覆盖功能。可为0或1： • 0表示不开启二维码覆盖 • 1表示开启二维码覆盖 功能开启后，将对识别出的二维码覆盖上马赛克，默认值0	Int	否
bar-type	二维码/条形码识别功能，将对识别出的二维码/条形码 覆盖马赛克。取值为0，1，2： • 0表示都识别， • 1表示识别二维码 • 2表示识别条形码	Int	否

	默认值0		
segment	通用的切片开关参数，指定是否需要切片，有效值： • 0（表示不切片） • 1（表示切片） 默认值0，需要切片时，后台会根据图片尺寸进行切片识别	Int	否
size	当segment取值为1时生效，默认1000像素，取值范围为大于等于500的整数。当size指定的数值大于图片像素时，则不进行切片，直接识别	Int	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

响应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

ImageInfo 节点内容：

节点名称	类型	描述
Format	String	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
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
Width	Int	图片宽度
Height	Int	图片高度
Size	Int	图片大小
Quality	Int	图片质量
codeStatus	Int	二维码识别结果。0表示未识别到二维码，1表示识别到二维码
QRcodeInfo	Container	二维码识别结果，可能有多个

QRcodeInfo 节点内容：

节点名称	类型	描述
codeUrl	String	二维码的内容。可能识别不出
codelocation	Container	图中识别到的二维码位置坐标

版权所有：腾讯云计算（北京）有限责任公司

codelocation 节点内容：

节点名称	类型	描述
point	String	二维码坐标点

实际案例

请求

```
PUT /picture.jpg 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":"test.jpg","rule":" QRcode/cover/1"}]}
Content-Length: 64


[Object]
```

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>picture.jpg</Key>
     <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/picture.jpg
     <ImageInfo>
          <Format>JPEG</Format>
          <Width>640</Width>
          <Height>427</Height>
          <Quality>100</Quality>
```

```
            <Ave>0xa18454</Ave>
            <Orientation>1</Orientation>
        </ImageInfo>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>picture-2.jpg</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/picture-2.
            <Format>png</Format>
            <Width>640</Width>
            <Height>427</Height>
            <Size>463092</Size>
            <Quality>100</Quality>
            <codeStatus>1</codeStatus>
            <QRcodeInfo>
                <codeUrl>xxxxxxxxxxxx</codeUrl>
                <codelocation>
                    <point>100,100</point>
                    <point>100,200</point>
                    <point>200,200</point>
                    <point>200,100</point>
                </codelocation>
            </QRcodeInfo>
                <codeUrl>xxxxxxxxxxxx</codeUrl>
                <codelocation>
                    <point>1000,1000</point>
                    <point>1000,2000</point>
                    <point>2000,2000</point>
                    <point>2000,1000</point>
                </codelocation>
        </Object>
    </ProcessResults>
</UploadResult>
```

请求二：下载时识别

请求

```
GET /<ObjectKey>?ci-process=QRcode&cover=<cover> 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。

请求头

此接口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

参数名称	描述	是否必选	类型
ObjectKey	对象文件名，例如 folder/sample.jpg	是	String
ci-process	万象处理能力，二维码识别固定为 QRcode	是	String
cover	二维码覆盖功能，将对识别出的二维码覆盖上马赛克。取值为0或1。0表示不开启二维码覆盖，1表示开启二维码覆盖，默认值0	否	Int

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>
    <CodeStatus>1</CodeStatus>
    <QRcodeInfo>
        <CodeUrl>xxxx</CodeUrl>
        <CodeLocation>
            <Point>xxx,xxx</Point>
            <Point>xxx,xxx</Point>
            <Point>xxx,xxx</Point>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Point>xxx,xxx</Point>
        </CodeLocation>
    </QRcodeInfo>
    <ResultImage>
        base64编码的图片
    </ResultImage>
</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
CodeStatus	Response	二维码识别结果。0表示未识别到二维码，1表示识别到二维码	Int
QRcodeInfo	Response	二维码识别结果，可能有多个	Container
ResultImage	Response	处理后的图片 base64数据，请求参数 cover 为1时返回	String

QRcodeInfo 节点内容：

节点名称（关键字）	父节点	描述	类型
CodeUrl	QRcodeInfo	二维码的内容。可能识别不到内容	String
CodeLocation	QRcodeInfo	图中识别到的二维码位置坐标	Container

CodeLocation 节点内容：

节点名称（关键字）	父节点	描述	类型
Point	CodeLocation	二维码坐标点（X坐标,Y坐标）	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 /<ObjectKey>?ci-process=QRcode&cover=1 HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-s
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-ci
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****



<Response>
    <CodeStatus>1</CodeStatus>
      <QRcodeInfo>
        <CodeUrl>xxxxxxxxxxxx</CodeUrl>
        <CodeLocation>
            <Point>100,100</Point>
         <Point>100,200</Point>
         <Point>200,200</Point>
         <Point>200,100</Point>
        </CodeLocation>
    </QRcodeInfo>
      <QRcodeInfo>
        <CodeUrl>xxxxxxxxxxxx</CodeUrl>
        <CodeLocation>
        <Point>1000,1000</Point>
        <Point>1000,2000</Point>
        <Point>2000,2000</Point>
        <Point>2000,1000</Point>
        </CodeLocation>
    </QRcodeInfo>
    <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

图片二维码生成

最近更新时间：2023-06-19 16:47:03

功能描述

数据万象二维码生成功能可根据用户指定的文本信息（URL 或文本），生成对应的二维码或条形码。

请求

请求示例

```
GET /?ci-process=qrcode-generate&qrcode-content=<qrcode-content>&mode=<mode>
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
ci-process	数据万象处理能力，二维码生成参数为 qrcode-generate	String	是
qrcode-content	可识别的二维码文本信息	String	是
mode	生成的二维码类型，可选值：0或1。0为二维码，1为条形码，默认值为0	Int	否
width	指定生成的二维码或条形码的宽度，高度会进行等比压缩	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>
</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
ResultImage	Response	二维码图片 base64 数据	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 /?ci-process=qrcode-generate&qrcode-content=<https://www.example.com>&mode
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Apr 2020 09:35:16 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0****&q-sign-tir
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 414641
Date: Fri, 10 Apr 2020 09:35:16 GMT
Server: tencent-ci
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****



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

图片标签

最近更新时间：2023-08-09 17:33:03

功能概述

图片标签功能通过借助数据万象的持久化处理接口，实现对 COS 存量数据的图片标签识别，返回图片中置信度较高的主题标签。图片标签识别请求包属于 GET 请求，请求时需要携带签名。

> **注意**
> 当前内容识别的图片限制为：图片格式支持 PNG、JPEG、JPG，图片大小不超过3MB，图片宽高大于32 x 32像素，小于7680 x 4320像素。

请求语法

- 原图存储在COS：

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

- 原图来自其他链接：

```
GET /?ci-process=detect-label&scenes=<scenes>&detect-url=<detect-url> 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）。

请求内容

参数名称	类型	是否必选	描述
ObjectKey	string	是	对象文件名，例如 picture.jpg
ci-	string	是	固定值detect-label

process			
scenes	String	否	本次调用支持的识别场景，可选值如下： • web，针对网络图片优化； • camera，针对手机摄像头拍摄图片优化； • album，针对手机相册、网盘产品优化； • news，针对新闻、资讯、广电等行业优化； • 如果不传此参数，则默认为camera。 支持多场景（scenes）一起检测，以 `,` 分隔。例如，使用 scenes=web，camera 即对一张图片使用两个模型同时检测，输出两套识别结果。
detect-url	String	否	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 detect-url 时，后台会默认处理 ObjectKey，填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey detect-url 示例：`http://www.example.com/abc.jpg`，需要进行 UrlEncode，处理后为 `http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg`

返回内容

响应包体具体数据内容如下：

参数名称	类型	描述
RecognitionResult	Container	图片标签识别结果

RecognitionResult 节点内容：

参数名称	类型	描述
WebLabels	Container	Web网络版标签结果数组。如未选择web场景，则不存在该项。此字段可能为空，表示取不到有效值
CameraLabels	Container	Camera摄像头版标签结果数组。如未选择camera场景，则不存在该项。此字段可能为空，表示取不到有效值
AlbumLabels	Container	Album相册版标签结果数组。如未选择album场景，则不存在该项。此字段可能为空，表示取不到有效值
NewsLabels	Container	News新闻版标签结果数组。如未选择news场景，则不存在该项。此字段可能为空，表示取不到有效值

版权所有：腾讯云计算（北京）有限责任公司

WebLabels，CameraLabels，AlbumLabels，NewsLabels 节点内容：

参数名称	类型	描述
Labels	Container	标签信息

Labels节点内容：

参数名称	类型	描述
Confidence	Int	该标签的置信度分数，分数越高则该标签准确度越高
Name	String	识别出的图片标签
FirstCategory	String	标签的一级分类
SecondCategory	String	标签的二级分类

示例

请求

```
GET /picture.jpg?ci-process=detect-label&scenes=web,camera,album,news HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX
```

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****

<RecognitionResult>
  <AlbumLabels>
    <Labels>
      <Confidence>72</Confidence>
      <FirstCategory>动植物</FirstCategory>
      <Name>鹿</Name>
      <SecondCategory>哺乳动物</SecondCategory>
    </Labels>
```

```
    </AlbumLabels>
    <CameraLabels>
      <Labels>
        <Confidence>26</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>羚羊</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>21</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>兔子</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>18</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>鹿</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>10</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>狮子</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>5</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>豹</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
    </CameraLabels>
    <NewsLabels>
      <Labels>
        <Confidence>60</Confidence>
        <FirstCategory>植物及其他生物</FirstCategory>
        <Name>植物</Name>
        <SecondCategory>植物类别</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>38</Confidence>
        <FirstCategory>植物</FirstCategory>
        <Name>树</Name>
        <SecondCategory>树</SecondCategory>
```

```
      </Labels>
    </NewsLabels>
    <WebLabels>
      <Labels>
        <Confidence>97</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>野生动物</Name>
        <SecondCategory>其他</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>82</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>哺乳动物</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>78</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>动物</Name>
        <SecondCategory>其他</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>71</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>鹿</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
      <Labels>
        <Confidence>41</Confidence>
        <FirstCategory>动植物</FirstCategory>
        <Name>白尾鹿</Name>
        <SecondCategory>哺乳动物</SecondCategory>
      </Labels>
    </WebLabels>
</RecognitionResult>
```

版权所有：腾讯云计算（北京）有限责任公司

游戏场景识别

最近更新时间：2023-08-14 17:01:45

功能概述

游戏标签功能实现游戏图片场景的识别，返回图片中置信度较高的游戏类别标签。游戏标签识别请求包属于 GET 请求，请求时需要携带签名。

> **注意**
> 当前内容识别的图片限制为：图片格式支持 PNG、JPEG、JPG，图片大小不超过5MB，图片宽高大于32 x 32像素，小于7680 x 4320像素。
> 此功能计费使用图像标签计费项，比例为1:2，即执行1次游戏标签将产生2次图片标签用量，计费详情见 AI 智能计费 。

请求语法

请求：

- 原图存储在COS：

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

- 原图来自其他链接：

```
GET /?ci-process=AIGameRec&detect-url=<detect-url> 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）。

请求内容

参数名称	类型	是否必选	描述
-			

ObjectKey	图片地址	是（与 detect-url 二选一）	String
detect-url	您可以通过填写 detect-url 对任意公网可访问的图片进行游戏场景识别。不填写 detect-url 时，后台会默认处理 ObjectKey ；填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey ，detect-url 示例：http://www.example.com/abc.jpg。	是	String

返回内容

响应包体具体数据内容如下：

参数名称	类型	描述
RecognitionResult	Container	图片游戏标签识别结果

RecognitionResult 节点内容：

参数名称	类型	描述
GameLabels	Container	游戏标签信息

标签信息中包含如下内容：

参数名称	类型	描述
Confidence	Int	该标签的置信度分数，分数越高则该标签准确度越高
FirstCategory	String	一级标签
SecondCategory	String	二级标签
GameName	String	识别出的游戏标签名称

示例

请求

```
GET /ObjectKey?ci-process=AIGameRec HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization:XXXXXXXXXXXX
```

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: 645
Date: Tue, 03 Apr 2018 09:06:16 GMT
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<RecognitionResult>
    <GameLabels>
        <Confidence>100</Confidence>
        <FirstCategory>游戏</FirstCategory>
        <SecondCategory>游戏</SecondCategory>
        <GameName>王者荣耀</GameName>
    </GameLabels>
</RecognitionResult>
```

图片质量评分

最近更新时间：2023-06-19 16:47:03

功能描述

图片质量评分功能为同步请求方式，您可以通过本接口对图片文件进行检测，从多方面评估，最终给出综合可观的清晰度评分和主观的美观度评分。该接口属于 GET 请求。

费用说明

- 成功调用接口会产生 图片质量评估费用 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。
- 图片大小：所下载图片经 Base64 编码后**不超过4MB**。
- 图片像素：建议大于**50*50像素**，否则影响识别效果。
- 长宽比：建议**长边与短边的比例小于5:1**。
- 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=AssessQuality 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，图像质量检测固定为 AssessQuality。	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <LongImage>FALSE</LongImage>
  <BlackAndWhite>FALSE</BlackAndWhite>
  <SmallImage>FALSE</SmallImage>
  <BigImage>TRUE</BigImage>
  <PureImage>FALSE</PureImage>
  <ClarityScore>89</ClarityScore>
  <AestheticScore>70</AestheticScore>
  <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：

参数名称	描述	类型
LongImag	取值为 TRUE 或 FALSE，TRUE 为长图，FALSE 为正常图，长图定义为	Boolea

e	长宽比大于等于3或小于等于1/3的图片。	n
BlackAndWhite	取值为 TRUE 或 FALSE，TRUE 为黑白图，FALSE 为否。黑白图即灰度图，指红绿蓝三个通道都是以灰度色阶显示的图片，并非视觉上的“黑白图片”。	Boolean
SmallImage	取值为 TRUE 或 FALSE，TRUE 为小图，FALSE 为否, 小图定义为最长边小于179像素的图片。当一张图片被判断为小图时，不建议做推荐和展示，其他字段统一输出为0或 FALSE。	Boolean
BigImage	取值为 TRUE 或 FALSE，TRUE 为大图，FALSE 为否，定义为最短边大于1000像素的图片	Boolean
PureImage	取值为 TRUE 或 FALSE，TRUE 为纯色图或纯文字图，即没有内容或只有简单内容的图片，FALSE 为正常图片。	Boolean
ClarityScore	综合评分。图像清晰度的得分，对图片的噪声、曝光、模糊、压缩等因素进行综合评估，取值为[0, 100]，值越大，越清晰。一般大于50为较清晰图片，标准可以自行把握。	Integer
AestheticScore	综合评分。图像美观度得分，从构图、色彩等多个艺术性维度评价图片，取值为[0, 100]，值越大，越美观。一般大于50为较美观图片，标准可以自行把握。	Integer
RequestId	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

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=AssessQuality HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 03 Apr 2019 09:06:15 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<Response>
  <LongImage>FALSE</LongImage>
  <BlackAndWhite>FALSE</BlackAndWhite>
  <SmallImage>FALSE</SmallImage>
  <BigImage>TRUE</BigImage>
  <PureImage>FALSE</PureImage>
  <ClarityScore>89</ClarityScore>
  <AestheticScore>70</AestheticScore>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
</Response>
```

人脸识别
人脸检测

最近更新时间：2023-06-19 16:47:04

功能描述

人脸检测功能为同步请求方式，您可以通过本接口检测图片中的人脸位置。该接口属于 GET 请求。

费用说明

- 成功调用接口会产生 人脸检测费用 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。
- 图片大小：所下载图片经 Base64 编码后**不超过5MB**。
- 图片像素：JPG 格式长边像素**不可超过4000**，其他格式图片长边像素**不可超过2000**。
- 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=DetectFace&max-face-num=<FaceNum> 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 DetectFace。	String	是
max-face-num	最多处理的人脸数目。默认值为1（仅检测图片中面积最大的那张人脸），最大值为120。此参数用于控制处理待检测图片中的人脸个数，值越小，处理速度越快。	Integer	否

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

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>
  <ImageWidth>616</ImageWidth>
  <ImageHeight>442</ImageHeight>
  <FaceModelVersion>3.0</FaceModelVersion>
  <RequestId>NjEzZjBlZGRfOTBmYTUwNjRfZWUx****</RequestId>
  <FaceInfos>
    <X>312</X>
    <Y>-5</Y>
    <Width>117</Width>
    <Height>173</Height>
  </FaceInfos>
</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
ImageWidth	Response	请求的图片宽度。	Integer
ImageHeight	Response	请求的图片高度。	Integer
FaceModelVersion	Response	人脸识别所用的算法模型版本。	String
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
FaceInfos	Response	人脸信息列表。包含人脸坐标信息。	Container

FaceInfos 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	FaceInfos	人脸框左上角横坐标。人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。	Integer
Y	FaceInfos	人脸框左上角纵坐标。人脸框包含人脸五官位置并在此基础上进行一定的扩展，若人脸框超出图片范围，会导致坐标负值。	Integer
Width	FaceInfos	人脸框宽度。	Integer
Height	FaceInfos	人脸框高度。	Integer

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.ConflictOperation	操作冲突，请勿同时操作相同的 Person。
FailedOperation.GroupInDeletedState	当前组正处于删除状态，请等待。

版权所有：腾讯云计算（北京）有限责任公司

FailedOperation.GroupPersonMapExist	组中已包含对应的人员 ID。
FailedOperation.GroupPersonMapNotExist	组中不包含对应的人员 ID。
FailedOperation.ImageDecodeFailed	图片解码失败。
FailedOperation.ImageDownloadError	图片下载错误。
FailedOperation.ImageFacedetectFailed	人脸检测失败。
FailedOperation.ImageResolutionExceed	图片分辨率过大。
FailedOperation.ImageResolutionTooSmall	图片短边分辨率小于64。
FailedOperation.ImageSizeExceed	base64编码后的图片数据大小不超过5MB。
FailedOperation.RequestLimitExceeded	请求频率超过限制。
FailedOperation.RequestTimeout	后端服务超时。
FailedOperation.ServerError	算法服务异常，请重试。
InternalError	内部错误。
InvalidParameter.InvalidParameter	参数不合法。
InvalidParameterValue.AccountFaceNumExceed	账号脸数量超出限制。
InvalidParameterValue.FaceModelVersionIllegal	算法模型版本不合法。
InvalidParameterValue.ImageEmpty	图片为空。
InvalidParameterValue.LimitExceed	返回数量超出限制。
InvalidParameterValue.NoFaceInPhoto	图片中没有人脸。
InvalidParameterValue.UrlIllegal	URL 格式不合法。
LimitExceeded.ErrorFaceNumExceed	人脸个数超过限制。
MissingParameter.ErrorParameterEmpty	必选参数为空。
ResourceUnavailable.Freeze	帐号已被冻结。
ResourceUnavailable.InArrears	帐号已欠费。
ResourceUnavailable.LowBalance	余额不足。

版权所有：腾讯云计算（北京）有限责任公司

ResourceUnavailable.StopUsing	帐号已停服。
ResourceUnavailable.UnknownStatus	计费状态未知。
ResourcesSoldOut.ChargeStatusException	计费状态异常。
UnsupportedOperation.UnknowMethod	未知方法名。

实际案例

请求

```
GET /test.jpg?max-face-num=1&ci-process=detect-face HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <ImageWidth>616</ImageWidth>
    <ImageHeight>442</ImageHeight>
    <FaceModelVersion>3.0</FaceModelVersion>
    <RequestId>NjEzZjBlZGRfOTBmYTUwNjRfZWUxXzE=</RequestId>
    <FaceInfos>
        <X>312</X>
        <Y>-5</Y>
        <Width>117</Width>
        <Height>173</Height>
    </FaceInfos>
    <FaceInfos>
        <X>600</X>
        <Y>-5</Y>
        <Width>117</Width>
        <Height>173</Height>
    </FaceInfos>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

人脸特效

最近更新时间：2023-06-21 09:18:42

功能描述

人脸特效，提供人脸美颜、人像变换、人像分割功能。

使用限制

图片 base64 编码后大小不可超过5M。支持PNG、JPG、JPEG、BMP，不支持 GIF 图片。
当type为face-segmentation时，图片分辨率须小于2000*2000。

请求

请求示例

- 原图存储在COS：

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

- 原图来自其他链接：

```
GET /?ci-process=face-effect&type=<type>&detect-url=<detect-url> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

> Authorization: Auth String（详情请查阅 请求签名 文档）。

请求参数

参数名称	是否必选	类型	描述
ObjectKey	是	String	对象文件名，例如 folder/document.pdf
ci-process	是	String	万象处理能力，人脸特效固定为face-effect

detect-url	否	String	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 detect-url 时，后台会默认处理 ObjectKey ，填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey detect-url 示例：`http://www.example.com/abc.jpg` ，需要进行 UrlEncode，处理后为 `http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg` 。
type	是	String	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
whitening	否	Integer	type为face-beautify时生效，美白程度，取值范围[0,100]。0不美白，100代表最高程度。默认值30
smoothing	否	Integer	type为face-beautify时生效，磨皮程度，取值范围[0,100]。0不磨皮，100代表最高程度。默认值10
faceLifting	否	Integer	type为face-beautify时生效，瘦脸程度，取值范围[0,100]。0不瘦脸，100代表最高程度。默认值70
eyeEnlarging	否	Integer	type为face-beautify时生效，大眼程度，取值范围[0,100]。0不大眼，100代表最高程度。默认值70
gender	否	Integer	type为face-gender-transformation时生效，选择转换方向，0：男变女，1：女变男。无默认值，为必选项。限制：仅对图片中面积最大的人脸进行转换。
age	否	Integer	type为face-age-transformation时生效，变化到的人脸年龄，[10,80]。无默认值，为必选项。限制：仅对图片中面积最大的人脸进行转换。

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 文档。

非公共头部

该请求操作无特殊的请求头部信息。

版权所有：腾讯云计算（北京）有限责任公司

请求体

该请求无请求体。

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阅 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>
  <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	String

		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）	

错误码

常见的错误信息请参阅 错误码 文档。

实际案例

使用模版ID

请求

```
GET /test.jpg?ci-process=face-effect&type=face-beautify&whitening=70&smoothing=80
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
Host:bucket-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: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

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

人体识别

最近更新时间：2023-06-19 16:47:04

功能概述

腾讯云数据万象通过 **AIBodyRecognition** 接口识别并输出画面中人体，输出其位置（矩形框）和置信度。图片人体识别请求包属于 GET 请求，请求时需要携带签名。

> **说明：**
> 当前人体识别的图片限制为：图片格式支持 PNG、JPEG、JPG，图片大小不超过10MB，图片宽高大于50像素 x 50像素。

请求语法

```
GET /?ci-process=AIBodyRecognition 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）。

请求内容

参数名称	描述	类型	是否必选
ci-process	数据万象处理能力，人体识别固定为AIBodyRecognition	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

该响应体返回为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
RecognitionResult	无	保存Logo识别结果的容器	Container

Response 的内容：

节点名称（关键字）	父节点	描述	类型
Status	RecognitionResult	人体识别结果。0表示未识别到，1表示识别到	Int
PedestrianInfo	RecognitionResult	人体识别结果，可能有多个	Container

LogoInfo 节点内容：

节点名称（关键字）	父节点	描述	类型
Name	PedestrianInfo	识别类型，人体识别默认：person	String
Sorce	PedestrianInfo	人体的置信度，取值范围为[0-100]。值越高概率越大。	String
Location	PedestrianInfo	图中识别到人体的坐标	Container

Location 节点内容：

节点名称（关键字）	父节点	描述	类型
Point	Location	人体坐标点（X坐标,Y坐标）	String

示例

请求

```
GET /?ci-process=RecognizeLogo&detect-url=xxx HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
```

```
Date: xxxxx
Authorization:XXXXXXXXXXXX
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 414641
Date: xxxxxx
Server: tencent-ci
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<RecognitionResult>
  <Status>1</Status>
  <PedestrianInfo>
    <Name>person</Name>
    <Score>90</Score>
    <Location>
      <Point>162,266</Point>
      <Point>162,716</Point>
      <Point>392,716</Point>
      <Point>392,266</Point>
    </Location>
    <Name>person</Name>
    <Score>89</Score>
    <Location>
      <Point>682,162</Point>
      <Point>682,479</Point>
      <Point>783,479</Point>
      <Point>783,162</Point>
    </Location>
  </PedestrianInfo>
</RecognitionResult>
```

宠物识别

最近更新时间：2023-08-01 16:46:27

功能概述

腾讯云数据万象通过 **detect-pet** 接口识别并输出画面中宠物，输出其位置（矩形框）和置信度。图片宠物识别请求包属于 GET 请求，请求时需要携带签名。

> **说明：**
> 当前宠物识别的图片限制为：图片格式支持 PNG、JPEG、JPG，图片大小不超过10MB，图片宽高大于50 x 50像素。

请求

请求示例

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

请求参数

参数名称	描述	类型	是否必选
ci-process	数据万象处理能力，宠物识别固定为 detect-pet	String	是

请求头
此接口仅使用公共请求头部。
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。
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>
    <ResultInfo>
```

```
            <Score>64</Score>
            <Name>dog</Name>
            <Location>
                <X>59</X>
                <Y>51</Y>
                <Height>225</Height>
                <Width>175</Width>
            </Location>
        </ResultInfo>
        <ResultInfo>
            ...
        </ResultInfo>
</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
ResultInfo	Response	保存结果的容器	Container
Score	ResultInfo	分数，范围在0-100之间，值越高，表示目标为相应结果的可能性越高	int
Name	ResultInfo	宠物名称	string
Location	ResultInfo	保存宠物位置的容器	Container
X	Location	x轴坐标	int
Y	Location	y轴坐标	int
Height	Location	(x,y)坐标距离高度	int

Width	Location	(x,y)坐标距离长度	int

错误码

错误码	描述	HTTP状态码
FailedOperation.NoObjectDetected	未检测到目标	400 Bad Request
InternalError	文件下载失败/其他未知错误	400 Bad Request

实际案例

请求

```
GET /test.jpg?ci-process=detect-pet HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 359
Date: Mon, 26 Apr 2021 02:19:16 GMT
Server: tencent-image
x-ci-request-id: NjA4NjIzMjNfOTBmYTUwNjRfNjAx****

<Response>
    <ResultInfo>
        <Score>64</Score>
        <Name>dog</Name>
        <Location>
            <X>59</X>
            <Y>51</Y>
            <Height>225</Height>
            <Width>175</Width>
        </Location>
    </ResultInfo>
    <ResultInfo>
        <Score>62</Score>
        <Name>dog</Name>
        <Location>
            <X>237</X>
```

```
                <Y>60</Y>
                <Height>208</Height>
                <Width>177</Width>
            </Location>
        </ResultInfo>
</Response>
```

人脸核身
身份证识别

最近更新时间：2023-07-12 20:11:01

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。

```
PUT /<ObjectKey>?ci-process=IDCardOCR&CardSide=<CardSide>&Config=<Config> HT
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> HT
```

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

- 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
Object Key	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
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, "C
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
  <IdInfo>
    <Name>李明</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

版权所有：腾讯云计算（北京）有限责任公司

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

活体人脸核身

最近更新时间：2023-06-19 16:47:04

功能描述

集成了活体检测和跟权威库进行比对的能力，传入一段视频和姓名、身份证号信息即可进行验证。对录制的自拍视频进行活体检测，从而确认当前用户为真人，可防止照片、视频、静态3D建模等各种不同类型的攻击。检测为真人后，再判断该视频中的人与权威库的证件照是否属于同一个人，实现用户身份信息核实。

说明

- 当前活体人脸核身的视频片限制为：用于活体检测的视频 BASE64 编码后的大小不超过8M，支持 mp4、avi、flv 格式。
- 活体人脸核身为付费服务，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

视频上传时检测的请求包与 COS 简单上传文件接口一致，只需增加相关处理参数，即可在视频上传时返回检测结果，并可将原视频存入到 COS。

```
PUT /<ObjectKey>?ci-process=LivenessRecognition&IdCard=<IdCard>&Name=<Name>
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 的人脸视频进行相应识别操作，返回检测结果。

```
GET /<ObjectKey>?ci-process=LivenessRecognition&IdCard=<IdCard>&Name=<Name>
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
```

版权所有：腾讯云计算（北京）有限责任公司

```
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.mp4，BASE64 编码后的大小不超过8M，支持 mp4、avi、flv 格式	String	是
ci-process	数据万象处理能力，人脸核身固定为 LivenessRecognition	String	是
IdCard	身份证号	String	是
Name	姓名。中文请使用 UTF-8编码	String	是
LivenessType	• 活体检测类型，取值：LIP/ACTION/SILENT • LIP 为数字模式，ACTION 为动作模式，SILENT 为静默模式，三种模式选择一种传入	String	是
ValidateData	• 数字模式传参：数字验证码（1234），需先调用接口获取数字验证码 • 动作模式传参：传动作顺序（2，1 or 1，2），需先调用接口获取动作顺序 • 静默模式传参：空	String	是。当 LivenessType 填的是 SILENT 时，该参数为可选项
BestFrameNum	需要返回多张最佳截图，取值范围1 - 10，不设置默认返回一张最佳截图	Int	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。

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

具体的返回内容如下：

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
BestFrameBase64	Response	验证通过后的视频最佳截图照片，照片为 BASE64 编码后的值，JPG 格式 注意：此字段可能返回 null，表示取不到有效值	String
Sim	Response	• 相似度，取值范围 [0.00, 100.00]。推荐相似度大于等于70时可判断为同一人，可根据具体场景自行调整阈值（阈值70的误通过率为千分之一，阈值80的误通过率是万分之一） • 检测成功返回	Float
BestFrameList	Response	最佳截图列表，仅在配置了返回多张最佳截图时返回 注意：此字段可能返回 null，表示取不到有效值	String

实际案例

请求

```
GET /test.mp4?ci-process=LivenessRecognition&IdCard=112044165412202 43X&Name=
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-cos-request-id: NWY2OWFlYjdfNDQ0NjYzNjRfMTBm****



<Response>
    <BestFrameBase64>/9j/4AAQSkZJRgABAQAAAQA</BestFrameBase64>
    <Sim>100</Sim>
</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.ActionCloseEye	未检测到闭眼动作
FailedOperation.ActionFaceClose	脸离屏幕太近
FailedOperation.ActionFaceFar	脸离屏幕太远
FailedOperation.ActionFaceLeft	脸离屏幕太左
FailedOperation.ActionFaceRight	脸离屏幕太右
FailedOperation.ActionFirstAction	未检测到第一个动作
FailedOperation.ActionLightDark	光线太暗
FailedOperation.ActionLightStrong	光线太强
FailedOperation.ActionNodetectFace	未能检测到完整人脸
FailedOperation.ActionOpenMouth	未检测到张嘴动作
FailedOperation.CompareFail	比对失败
FailedOperation.CompareLowSimilarity	调用比对引擎接口出错
FailedOperation.FileSaveError	文件存储失败，请稍后重试
FailedOperation.IdFormatError	输入的身份证号有误
FailedOperation.IdNameMisMatch	姓名和身份证号不一致，请核实后重试

版权所有：腾讯云计算（北京）有限责任公司

FailedOperation.IdNoExistSystem	库中无此号，请到户籍所在地进行核实
FailedOperation.IdPhotoNoExist	库中无此号照片，请到户籍所在地进行核实
FailedOperation.IdPhotoPoorQuality	证件图片分辨率太低，请重新上传
FailedOperation.IdPhotoSystemNoanswer	客户库自建库或认证中心返照失败，请稍后再试
FailedOperation.LifePhotoDetectFaces	检测到多张人脸
FailedOperation.LifePhotoDetectFake	实人比对没通过
FailedOperation.LifePhotoDetectNoFaces	未能检测到完整人脸
FailedOperation.LifePhotoPoorQuality	传入图片分辨率太低，请重新上传
FailedOperation.LifePhotoSizeError	传入图片过大或过小
FailedOperation.LipFaceIncomplete	脸部未完整露出
FailedOperation.LipMoveSmall	嘴唇动作幅度过小
FailedOperation.LipNetFailed	视频拉取失败，请重试
FailedOperation.LipSizeError	视频为空，或大小不合适，请控制录制时长在6s左右
FailedOperation.LipVideoInvalid	视频格式有误
FailedOperation.LipVideoQuaility	视频像素太低
FailedOperation.LipVoiceDetect	未检测到声音
FailedOperation.LipVoiceLow	视频声音太小
FailedOperation.LipVoiceRecognize	声音识别失败
FailedOperation.LivessBestFrameError	人脸检测失败，无法提取比对照
FailedOperation.LivessDetectFail	活体检测没通过
FailedOperation.LivessDetectFake	疑似非真人录制
FailedOperation.LivessSystemError	调用活体引擎接口出错
FailedOperation.LivessUnknownError	视频实人检测没通过

版权所有：腾讯云计算（北京）有限责任公司

FailedOperation.NameFormatError	输入的姓名有误
FailedOperation.RequestLimitExceeded	调用次数超出限制
FailedOperation.SilentDetectFail	实人检测失败
FailedOperation.SilentThreshold	实人检测未达到通过标准
FailedOperation.SilentTooShort	视频录制时间过短，请录制2s以上的视频。
FailedOperation.UnKnown	内部未知错误
InvalidParameter	参数错误
InvalidParameter.RuleId	RuleId 不存在
InvalidParameterValue	参数取值错误
InvalidParameterValue.BizTokenExpired	BizToken 过期
InvalidParameterValue.RuleIdNotExist	RuleId 不存在，请到人脸核身控制台申请
UnauthorizedOperation	未授权操作
UnauthorizedOperation.Arrears	帐号已欠费
UnauthorizedOperation.NonAuthorize	账号未实名
UnauthorizedOperation.Nonactivated	未开通服务
UnsupportedOperation	操作不支持

版权所有：腾讯云计算（北京）有限责任公司

获取数字验证码

最近更新时间：2023-06-19 16:47:04

接口描述

使用数字活体检测模式前，需调用本接口获取数字验证码。

请求

请求示例

```
GET /?ci-process=GetLiveCode 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
ci-process	数据万象处理能力，获取数字验证码固定为 GetLiveCode	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

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

```
<Response>
  <LiveCode>数字验证码</LiveCode>
</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
LiveCode	Response	数字验证码，如：1234	String

实际案例

请求

```
GET /?ci-process=GetLiveCode HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-cos-request-id: NWY2OWFlZjVfOTA0MDYzNjRfNjY5Zl8zNQ==

<Response>
  <LiveCode>0521</LiveCode>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

获取动作顺序

最近更新时间：2023-06-20 15:10:27

接口描述

使用动作活体检测模式前，需调用本接口获取动作顺序。

请求

请求示例

```
GET /?ci-process=GetActionSequence 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
ci-process	数据万象处理能力，获取动作顺序固定为 GetActionSequence	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

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

```
<Response>
  <ActionSequence>动作顺序</ActionSequence>
</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
ActionSequence	Response	动作顺序（2，1 or 1，2），1代表张嘴，2代表闭眼	String

实际案例

请求

```
GET /?ci-process=GetActionSequence HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-cos-request-id: NWY2OWFlZjVfOTA0MDYzNjRfNjY5Zl8zNQ==

<Response>
  <ActionSequence>2,1</ActionSequence>
</Response>
```

车辆车牌识别

最近更新时间：2023-06-20 17:32:12

功能描述

车辆车牌检测功能为同步请求方式，您可以通过本接口检测图片中的车辆，识别出车辆的品牌、颜色、位置、车牌位置等信息。该接口属于 GET 请求。

费用说明

- 成功调用接口会产生 车辆车牌检测费用 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。
- 图片大小：所下载图片经 Base64 编码后**不超过4MB**。
- 图片像素：建议大于**50*50像素**，否则影响识别效果。
- 长宽比：建议**长边与短边的比例小于5：1**。
- 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=DetectCar 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，车辆识别固定为 DetectCar	String	是

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <RequestId></RequestId>
  <CarTags>
    <Serial></Serial>
    <Brand></Brand>
    <Type></Type>
    <Color><Color>
    <Confidence></Confidence>
    <Year></Year>
    <CarLocation>
      <X></X>
      <Y></Y>
    </CarLocation>
    <PlateContent>
      <Plate></Plate>
      <Color>Color>
      <Type></Type>
      <PlateLocation>
        <X></X>
        <Y></Y>
      </PlateLocation>
    </PlateContent>
  </CarTags>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	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
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
CarTags	Response	车辆属性识别的结果数组，支持返回多个车辆信息	Container

CarTags 节点的内容：

节点名称（关键字）	父节点	描述	类型
Serial	CarTags	车系	String
Brand	CarTags	车辆品牌	String
Type	CarTags	车辆类型	String
Color	CarTags	车辆颜色	String
Confidence	CarTags	置信度，0 – 100	Int
Year	CarTags	年份，识别不出年份时返回0	Int
CarLocation	CarTags	车辆在图片中的坐标信息，可能返回多个坐标点的值	Container
PlateContent	CarTags	车牌信息，包含车牌号、车牌颜色、车牌位置。支持返回多个车牌	Container

版权所有：腾讯云计算（北京）有限责任公司

CarLocation 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	CarLocation	横坐标 x	Int
Y	CarLocation	纵坐标 y	Int

PlateContent 节点的内容：

节点名称（关键字）	父节点	描述	类型
Plate	PlateContent	车牌号信息	String
Color	PlateContent	车牌的颜色	String
Type	PlateContent	车牌的种类，例如普通蓝牌	String
PlateLocation	PlateContent	车牌的位置	Container

PlateLocation 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	PlateLocation	定位出的车牌左上角、右上角、左下角、右下角的 X 坐标	Int
Y	PlateLocation	定位出的车牌左上角、右上角、左下角、右下角的 Y 坐标	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 test.jpg?ci-process=DetectCar HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
    <RequestId>NjE1MzEyMDlfOTBmYTUwNjRfNzUz****</RequestId>
    <CarTags>
        <Serial>奔驰S级</Serial>
        <Brand>奔驰</Brand>
        <Type>轿车</Type>
        <Color>黑</Color>
        <Confidence>98</Confidence>
        <Year>0</Year>
        <CarLocation>
            <X>8</X>
            <Y>364</Y>
        </CarLocation>
        <CarLocation>
            <X>8</X>
            <Y>5</Y>
        </CarLocation>
        <CarLocation>
            <X>637</X>
            <Y>5</Y>
        </CarLocation>
        <CarLocation>
            <X>637</X>
            <Y>364</Y>
        </CarLocation>
        <PlateContent>
            <Plate>陕ASxxxL</Plate>
            <Color>蓝色</Color>
            <Type>普通蓝牌</Type>
            <PlateLocation>
                <X>263</X>
                <Y>309</Y>
            </PlateLocation>
            <PlateLocation>
                <X>263</X>
                <Y>239</Y>
            </PlateLocation>
            <PlateLocation>
```

```
            <X>459</X>
            <Y>239</Y>
          </PlateLocation>
          <PlateLocation>
            <X>459</X>
            <Y>309</Y>
          </PlateLocation>
        </PlateContent>
      </CarTags>
      <CarTags>
        <Serial>五菱宏光</Serial>
        <Brand>五菱</Brand>
        <Type>面包车</Type>
        <Color>白</Color>
        <Confidence>0</Confidence>
        <Year>0</Year>
        <CarLocation>
          <X>0</X>
          <Y>228</Y>
        </CarLocation>
        <CarLocation>
          <X>0</X>
          <Y>81</Y>
        </CarLocation>
        <CarLocation>
          <X>73</X>
          <Y>81</Y>
        </CarLocation>
        <CarLocation>
          <X>73</X>
          <Y>228</Y>
        </CarLocation>
      </CarTags>
    </Response>
```

图片文字识别(OCR)

最近更新时间：2023-06-20 15:10:28

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，需携带签名。

> **说明：**
> - 图片格式：PNG、JPG、JPEG、BMP，不支持 GIF 图片。
> - 支持PDF
> - 图片大小：所下载图片经Base64编码后不超过7M。
> - 图片像素：建议分辨率在600 x 800以上，限制输入图片最大分辨率为8K（7680 x 4320像素），最小分辨率为32 x 32像素；

请求

请求示例

- 原图存储在COS：

```
GET /<ObjectKey>?ci-process=OCR&type=general&language-type=zh&ispdf=ture&p
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

- 原图来自其他链接：

```
GET /?ci-process=OCR&detect-url=<detect-url>&type=general&language-type=zh&i
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

Authorization: Auth String（详情请查阅 请求签名 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
detect-url	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 detect-url 时，后台会默认处理 ObjectKey ，填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey detect-url 示例：http://www.example.com/abc.jpg ，需要进行 UrlEncode，处理后为 http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg	String	否
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： zh：中英混合 zh_rare：支持英文、数字、中文生僻字、繁体字，特殊符号等 auto：自动 mix：混合语种 jap：日语 kor：韩语 spa：西班牙语 fre：法语 ger：德语 por：葡萄牙语 vie：越语 may：马来语 rus：俄语 ita：意大利语 hol：荷兰语 swe：瑞典语 fin：芬兰语	String	否

	dan：丹麦语 nor：挪威语 hun：匈牙利语 tha：泰语 hi：印地语 ara：阿拉伯语		
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

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 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

该请求无请求体。

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阅 公共响应头部 文档。

特有响应头

该响应无特殊的响应头。

版权所有：腾讯云计算（北京）有限责任公司

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。	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
Character	Words	候选字Character	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

版权所有：腾讯云计算（北京）有限责任公司

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	左上顶点坐标	Container
RightBottom	WordPolygon	左上顶点坐标	Container
LeftBottom	WordPolygon	左上顶点坐标	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

常见的错误信息请参阅 错误码 文档。

实际案例

使用模板ID

请求

```
GET /<ObjectKey>?ci-process=OCR&type=general&language-type=zh&ispdf=true&iswo
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
Host:bucket-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-ci
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<Response>
    <Angel>359.99</Angel>
```

版权所有：腾讯云计算（北京）有限责任公司

```
<Language>mix</Language>
<PdfPageSize>0</PdfPageSize>
<RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</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>
```

商品抠图

最近更新时间：2023-06-20 15:21:01

功能描述

腾讯云数据万象通过 **GoodsMatting** 接口检测图片中的商品信息，生成只包含商品信息的图片，支持持久化、云上处理及下载时处理。

使用限制：

- 待处理的图片，原图大小不超过5MB。
- 宽高不超过7680 x 7680像素。
- 支持处理JPG JPEG PNG。

1. 下载时处理

请求

- 原图存储在COS：

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

- 原图来自其他链接：

```
GET /<ObjectKey>?ci-process=GoodsMatting&detect-url=<detect-url> 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：GoodsMatting。

参数	含义	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，商品抠图固定为GoodsMatting。	String	是
detect-url	您可以通过填写 detect-url 处理任意公网可访问的图片链接。不填写 detect-url 时，后台会默认处理 ObjectKey ，填写了 detect-url 时，后台会处理 detect-url 链接，无需再填写 ObjectKey detect-url 示例： `http://www.example.com/abc.jpg` ，需要进行 UrlEncode，处理后为 `http%25253A%25252F%25252Fwww.example.com%25252Fabc.jpg` 。	String	否

响应

```
HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 98645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Server: tencent-ci
Status: 200 OK
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

[pic-content]
```

处理后的图片直接返回至响应体。

2.上传时处理

请求语法

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

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部。

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	固定为：ci-process=GoodsMatting

请求体

该请求body内容为待裁剪图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型
Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	图片路径	String
Etag	OriginalInfo	原图ETag值	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	抠图后图片文件名	String
Location	Object	抠图后的图片链接	String
ETag	Object	抠图后图片的ETag值	String

版权所有：腾讯云计算（北京）有限责任公司

Size	Object	抠图后图片大小	Int

示例

请求：

```
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": "ci-process=GoodsMatting"
  }]
}
```

处理后的图片存至存储桶，并以fileid字段的值作为对象名称。

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>goods_matting.jpg</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>goods_matting_process.jpeg</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </Object>
  </ProcessResults>
</UploadResult>
```

3.云上数据处理

请求语法

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

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

请求

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。
除公共头部外，还需要传入如下头部。

头部名称	描述	类型
Pic-Operations	对图片进行操作的参数	String

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
bucke	Str	否	存储结果的目标存储桶名称，格式为 BucketName-AppId，如果不指定

t	ing		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 4. 如希望修复后的图片覆盖原图，请将fileid字段设置为<ObjectKey>
rule	String	是	固定为：ci-process=GoodsMatting

请求体

该请求body内容为待裁剪图片原图
[Image Content]

响应

响应头

此接口返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

该响应体返回content-type为 **application/xml** 数据，具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
UploadResult	无	请求结果	Container

UploadResult 的内容：

节点名称（关键字）	父节点	描述	类型
OriginalInfo	UploadResult	原图信息	Container
ProcessResults	UploadResult	请求处理结果	Container

OriginalInfo 节点内容：

节点名称	父节点	描述	类型

Key	OriginalInfo	原图文件名	String
Location	OriginalInfo	原图路径	String
Etag	OriginalInfo	原图ETag值	String

ProcessResults 节点内容：

节点名称（关键字）	父节点	描述	类型
Object	ProcessResults	每个图片的处理结果	Container

Object 节点内容：

节点名称（关键字）	父节点	描述	类型
Key	Object	抠图后图片文件名	String
Location	Object	抠图后的图片链接	String
ETag	Object	抠图后图片的ETag值	String
Size	Object	抠图后图片大小	Int

示例

请求：

```
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:
{
 "rules": [{
    "fileid": "exampleobject",
    "rule": "ci-process=GoodsMatting"
 }]
}
```

处理后的图片存至存储桶，并以fileid字段的值作为对象名称。

响应：

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 645
Date: Tue, 23 Apr 2022 09:06:16 GMT
Status: 200 OK
Server: tencent-ci
x-cos-request-id: NWFjMzQ0MDZfOTBmYTUwXzZkZV8z****

<UploadResult>
    <OriginalInfo>
        <Key>goods_matting.jpg</Key>
        <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/filename.jp
        <ETag>"ed60e3598e6862d8c8443a06acfb2153"</ETag>
    </OriginalInfo>
    <ProcessResults>
        <Object>
            <Key>goods_matting_process.jpeg</Key>
            <Location>examplebucket-1250000000.cos.ap-chengdu.myqcloud.com/example
            <Format/>
            <Size>354256</Size>
            <ETag>"521451f63c331becf533e466732c1091"</ETag>
        </Object>
    </ProcessResults>
</UploadResult>
```

版权所有：腾讯云计算（北京）有限责任公司

以图搜图
开通以图搜图

最近更新时间：2023-06-20 15:10:28

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****&
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：2023-07-14 18:05:21

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）。
- 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>
   <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

具体数据描述如下：

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
ObjectKey	Request	对象文件名，例如：folder/document.jpg。	String	是
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（例 key1>=1 key1>='aa'）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****&
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>
```

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****
```

图片搜索接口

最近更新时间：2023-07-14 18:05:21

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=&O
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）。
- 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）	描述	类型	是否必选
ObjectKey	对象文件名，例如：folder/document.jpg。	String	是
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

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

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，最多不超过10个，json 字符串，格式为 key:value（例 key1>=1 key1>='aa'）对	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
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&
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：2023-06-20 15:10:28

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）。
- 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>
   <EntityId>apple</EntityId>
</Request>
```

具体数据描述如下：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型	是否必选
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-s
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****
```

内容翻译
提交任务

最近更新时间：2023-08-01 16:46:25

功能描述

提交一个翻译任务。

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>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>
    <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
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	Req	任务回调地址，优先级高于队列的回调地址。设置为 no	Stri	否

	uest	时，表示队列的回调地址不产生回调	ng	
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	默认值	是否必选
Object	Request.Input	源文档文件名 • 单文件（docx/xlsx/html/markdown/txt）：不超过800万字符 • 有页数的（pdf/pptx）：不超过300页 • 文本文件（txt）：不超过10MB • 二进制文件（pdf/docx/pptx/xlsx）：不超过60MB • 图片文件（jpg/jpeg/png/webp）：不超过10MB	String	无	是
Lang	Request.Input	文档语言类型 • 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	String	无	是

Type	Request.Input	文档类型 • pdf • docx • pptx • xlsx • 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 • webp	String	无	是
BasicType	Request.Input	原始文档类型 • 仅在 Type=pdf/jpg/jpeg/png/webp 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	String	无	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	默认值	是否必选
Translation	Request.Operation	翻译参数	Container	无	是
Output	Request.Operation	结果输出地址，当NoNeedOutput为true时非必选	Container	无	否
UserData	Request.Operation	透传用户信息，可打印的 ASCII 码，长度不超过1024	String	无	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	无	否
NoNeedOutput	Request.Operation	仅输出结果，不生成结果文件。取值：true/false。该参数原文档类型为图片时有效。默认值 false	String	无	否

版权所有：腾讯云计算（北京）有限责任公司

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	默认值	是否必选
Lang	Request.Operation.Translation	目标语言类型 • 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、km、lo、pt • 源语言类型为 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	String	无	是
Type	Request.Operation.Translation	文档类型，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 • webp：txt	String	无	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.Outp	存储结果的存储桶	Stri	是

版权所有：腾讯云计算（北京）有限责任公司

	ut		ng	
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 数组

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	创建任务的 ID	String
Tag	Response.JobsDetail	创建任务的 Tag：Translation	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	String
CreationTime	Response.JobsDetail	任务的创建时间	String
StartTime	Response.J	任务的开始时间	Stri

	obsDetail		ng
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 /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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

回调信息

最近更新时间：2023-08-11 17:56:15

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，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </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	同提交翻译任务接口中的 Response.JobsDetail	Container 数组

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>
    </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"
    }]
}
```

词性分析
提交任务

最近更新时间：2023-08-01 16:46:25

功能描述

提交一个分词任务。

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>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>
    <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
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	文件路径	String	是

版权所有：腾讯云计算（北京）有限责任公司

Container 类型 Operation 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
WordsGeneralize	Request.Operation	指定分词参数	Container	是
UserData	Request.Operation	透传用户信息，可打印的 ASCII 码，长度不超过1024	String	否
JobLevel	Request.Operation	任务优先级，级别限制：0 、1 、2。级别越大任务优先级越高，默认为0	String	否

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

节点名称（关键字）	父节点	描述	类型	是否必选
NerMethod	Request.Operation.WordsGeneralize	ner 方式，支持 NerBasic 和 DL，默认值 DL	String	否
SegMethod	Request.Operation.WordsGeneralize	分词粒度，支持 SegBasic 和 MIX，默认值 MIX	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>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 数组

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	创建任务的 ID	String

Tag	Response.JobsDetail	创建任务的 Tag：WordsGeneralize	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
WordsGeneralize	Response.JobsDetail.Operation	同请求中的 Request.Operation.WordsGeneralize	Container
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	Container

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：

节点名称（关键字）	父节点	描述	类型
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
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

版权所有：腾讯云计算（北京）有限责任公司

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
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 /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>
    <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>
```

回调信息

最近更新时间：2023-08-10 18:13:43

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，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            </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	同提交分词任务接口中的 Response.JobsDetail	Container 数组

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>
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

案例 2：通过任务触发的任务回调，格式为 JSON


```
{
  "EventName": "TaskFinish",
  "JobsDetail": [{
    "Tag": "WordsGeneralize",
    "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": {
      "UserData": "This is my WordsGeneralize job.",
      "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"
            }
         ]
      }
   }
}]
}
```

视频目标检测
提交任务

最近更新时间：2023-08-11 17:56:16

功能描述

提交一个视频目标检测任务。

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>VideoTargetRec</Tag>
  <Input>
    <Object>input/demo.mp4</Object>
  </Input>
  <Operation>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <UserData>This is my data.</UserData>
```

```
        <JobLevel>0</JobLevel>
    </Operation>
    <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：VideoTargetRec	String	是
Operation	Request	操作规则	Container	是
Input	Request	待操作的媒体信息	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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。详情请参见 CallBackMqConfig	Container	否

Container 类型 Input 的具体数据描述如下：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型	是否必选
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
TemplateId	Request.Operation	视频目标检测模板 ID	String	否
VideoTargetRec	Request.Operation	视频目标检测参数，同创建视频目标检测模板接口中的 Request.VideoTargetRec	Container	否
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**
> 视频目标检测参数必须设置，可以通过 TemplateId 或 VideoTargetRec 设置， TemplateId 优先级更高。

响应

响应头

此接口仅返回公共响应头部，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>VideoTargetRec</Tag>
      <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <TemplateName>video_target_rec_demo</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 数组

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：VideoTargetRec	String
State	Response.J	任务状态	Stri

	obsDetail	• Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	ng
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
VideoTargetRec	Response.JobsDetail.Operation	同请求中的 Request.Operation.VideoTargetRec	Container
VideoTargetRecResult	Response.JobsDetail.Operation	视频目标检测结果	Container
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.	任务优先级	String

	Operation		

Container 节点 VideoTargetRecResult 的内容：

节点名称（关键字）	父节点	描述	类型
BodyRecognition	Response.JobsDetail.Operation.VideoTargetRecResult	人体识别结果	Container 数组
PetRecognition	Response.JobsDetail.Operation.VideoTargetRecResult	宠物识别结果	Container 数组
CarRecognition	Response.JobsDetail.Operation.VideoTargetRecResult	车辆识别结果	Container 数组

Container 节点 BodyRecognition 的内容：

节点名称（关键字）	父节点	描述	类型
Time	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition	截图的时间点，单位为秒	String
Url	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition	截图 URL	String
BodyInfo	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition	人体识别结果，可能有多个	Container 数组

Container 节点 PetRecognition 的内容：

节点名称（关键字）	父节点	描述	类型
Time	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition	截图的时间点，单位为秒	String
Url	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition	截图 URL	String
PetInfo	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition	宠物识别结果结果，可能有多个	Container 数组

Container 节点 CarRecognition 的内容：

节点名称（关	父节点	描述	类型

键字）			
Time	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition	截图的时间点，单位为秒	String
Url	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition	截图 URL	String
CarInfo	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition	车辆识别结果，可能有多个	Container 数组

Container 节点 BodyInfo 节点的内容：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo	识别类型	String
Score	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo	识别的置信度，取值范围为[0-100]。值越高概率越大。	Int
Location	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo	图中识别到人体的坐标	Container

Container 节点 Location 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo.Location	X坐标	String
Y	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo.Location	Y坐标	String
Height	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo.Location	(X,Y)坐标距离高度	String

版权所有：腾讯云计算（北京）有限责任公司

Width	Response.JobsDetail.Operation.VideoTargetRecResult.BodyRecognition.BodyInfo.Location	(X,Y)坐标距离长度	String

Container 节点 PetInfo 节点的内容：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo	识别类型	String
Score	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo	识别的置信度，取值范围为[0-100]。值越高概率越大。	Int
Location	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo	图中识别到宠物的坐标	Container

Container 节点 Location 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo.Location	X坐标	Int
Y	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo.Location	Y坐标	Int
Height	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo.Location	(X,Y)坐标距离高度	Int
Width	Response.JobsDetail.Operation.VideoTargetRecResult.PetRecognition.PetInfo.Location	(X,Y)坐标距离长度	Int

Container 节点 CarInfo 节点的内容：

节点名称（关键字）	父节点	描述	类型
Name	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo	识别类型	String
Score	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo	识别的置信度，取值范围为[0-100]。值越高概率越大。	Int
Location	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo	图中识别到车辆的坐标	Container

Container 节点 Location 节点的内容：

节点名称（关键字）	父节点	描述	类型
X	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo.Location	X坐标	Int
Y	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo.Location	Y坐标	Int
Height	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo.Location	(X,Y)坐标距离高度	Int
Width	Response.JobsDetail.Operation.VideoTargetRecResult.CarRecognition.CarInfo.Location	(X,Y)坐标距离长度	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1：使用视频目标检测模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
```

```
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>VideoTargetRec</Tag>
    <Input>
        <Object>input/demo.mp4</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>VideoTargetRec</Tag>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <TemplateName>video_target_rec_demo</TemplateName>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
```

```
    </Operation>
  </JobsDetail>
</Response>
```

请求2：使用视频目标检测处理参数

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
  <Tag>VideoTargetRec</Tag>
  <Input>
    <Object>input/demo.mp4</Object>
  </Input>
  <Operation>
    <VideoTargetRec>
      <Body>true</Body>
      <Car>true</Car>
      <Pet>true</Pet>
    </VideoTargetRec>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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>VideoTargetRec</Tag>
      <Operation>
        <VideoTargetRec>
          <Body>true</Body>
          <Car>true</Car>
          <Pet>true</Pet>
        </VideoTargetRec>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
    </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-18 15:28:02

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-11-21T20:32:44+0800</CreationTime>
    <EndTime>2022-11-21T20:32:54+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>input/car.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>a993e661c699811ed9588038f7fb96507</JobId>
    <Message>TaskFinish</Message>
    <Operation>
      <JobLevel>0</JobLevel>
      <VideoTargetRec>
        <Body>true</Body>
        <Car>true</Car>
        <Pet>true</Pet>
      </VideoTargetRec>
      <VideoTargetRecResult>
        <BodyRecognition>
          <BodyInfo>
            <Location>
              <Height>382</Height>
              <Width>193</Width>
              <X>469</X>
              <Y>67</Y>
            </Location>
            <Name>person</Name>
            <Score>86</Score>
```

```
    </BodyInfo>
    <Time>1.206000</Time>
    <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
  </BodyRecognition>
  <BodyRecognition>
    <BodyInfo>
      <Location>
        <Height>349</Height>
        <Width>198</Width>
        <X>315</X>
        <Y>177</Y>
      </Location>
      <Name>person</Name>
      <Score>83</Score>
    </BodyInfo>
    <Time>2.413000</Time>
    <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
  </BodyRecognition>
  <BodyRecognition>
    <BodyInfo>
      <Location>
        <Height>366</Height>
        <Width>190</Width>
        <X>322</X>
        <Y>126</Y>
      </Location>
      <Name>person</Name>
      <Score>85</Score>
    </BodyInfo>
    <Time>3.620000</Time>
    <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
  </BodyRecognition>
  <BodyRecognition>
    <BodyInfo>
      <Location>
        <Height>29</Height>
        <Width>21</Width>
        <X>807</X>
        <Y>195</Y>
      </Location>
      <Name>person</Name>
      <Score>74</Score>
    </BodyInfo>
    <Time>9.655000</Time>
    <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
```

```
        </BodyRecognition>
        <CarRecognition>
          <CarInfo>
            <Location>
              <Height>526</Height>
              <Width>911</Width>
              <X>48</X>
              <Y>0</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>72</Score>
          </CarInfo>
          <Time>0.000000</Time>
          <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
        </CarRecognition>
        <CarRecognition>
          <CarInfo>
            <Location>
              <Height>459</Height>
              <Width>744</Width>
              <X>9</X>
              <Y>48</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>87</Score>
          </CarInfo>
          <Time>6.034000</Time>
          <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
        </CarRecognition>
        <CarRecognition>
          <CarInfo>
            <Location>
              <Height>284</Height>
              <Width>176</Width>
              <X>781</X>
              <Y>235</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>92</Score>
          </CarInfo>
          <CarInfo>
            <Location>
              <Height>231</Height>
              <Width>290</Width>
              <X>0</X>
```

```
            <Y>231</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>91</Score>
        </CarInfo>
        <CarInfo>
          <Location>
            <Height>132</Height>
            <Width>165</Width>
            <X>0</X>
            <Y>216</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>85</Score>
        </CarInfo>
        <CarInfo>
          <Location>
            <Height>84</Height>
            <Width>107</Width>
            <X>733</X>
            <Y>219</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>81</Score>
        </CarInfo>
        <CarInfo>
          <Location>
            <Height>457</Height>
            <Width>525</Width>
            <X>237</X>
            <Y>40</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>71</Score>
        </CarInfo>
        <Time>8.448000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-123456
      </CarRecognition>
      <CarRecognition>
        <CarInfo>
          <Location>
            <Height>233</Height>
            <Width>270</Width>
            <X>0</X>
            <Y>232</Y>
```

```
            </Location>
            <Name>vehicle</Name>
            <Score>89</Score>
         </CarInfo>
         <CarInfo>
            <Location>
               <Height>272</Height>
               <Width>199</Width>
               <X>760</X>
               <Y>234</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>78</Score>
         </CarInfo>
         <CarInfo>
            <Location>
               <Height>140</Height>
               <Width>153</Width>
               <X>0</X>
               <Y>214</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>70</Score>
         </CarInfo>
         <Time>9.655000</Time>
         <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-123456
      </CarRecognition>
      <CarRecognition>
         <CarInfo>
            <Location>
               <Height>125</Height>
               <Width>150</Width>
               <X>189</X>
               <Y>282</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>90</Score>
         </CarInfo>
         <CarInfo>
            <Location>
               <Height>207</Height>
               <Width>191</Width>
               <X>0</X>
               <Y>330</Y>
            </Location>
```

```
            <Name>vehicle</Name>
            <Score>86</Score>
        </CarInfo>
        <CarInfo>
            <Location>
                <Height>62</Height>
                <Width>96</Width>
                <X>99</X>
                <Y>288</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>81</Score>
        </CarInfo>
        <CarInfo>
            <Location>
                <Height>58</Height>
                <Width>43</Width>
                <X>378</X>
                <Y>282</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>74</Score>
        </CarInfo>
        <CarInfo>
            <Location>
                <Height>22</Height>
                <Width>27</Width>
                <X>342</X>
                <Y>271</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>73</Score>
        </CarInfo>
        <CarInfo>
            <Location>
                <Height>85</Height>
                <Width>107</Width>
                <X>208</X>
                <Y>229</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>70</Score>
        </CarInfo>
        <CarInfo>
            <Location>
```

```
            <Height>43</Height>
            <Width>31</Width>
            <X>313</X>
            <Y>271</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>70</Score>
        </CarInfo>
        <Time>10.862000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345&
      </CarRecognition>
      <CarRecognition>
        <CarInfo>
          <Location>
            <Height>338</Height>
            <Width>817</Width>
            <X>61</X>
            <Y>144</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>88</Score>
        </CarInfo>
        <Time>19.309999</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345&
      </CarRecognition>
      <CarRecognition>
        <CarInfo>
          <Location>
            <Height>465</Height>
            <Width>854</Width>
            <X>48</X>
            <Y>15</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>80</Score>
        </CarInfo>
        <Time>27.757999</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345&
      </CarRecognition>
      <Duration>33171</Duration>
      <PetRecognition>
        <PetInfo>
          <Location>
            <Height>92</Height>
            <Width>210</Width>
```

```
                <X>622</X>
                <Y>148</Y>
              </Location>
              <Name>dog</Name>
              <Score>81</Score>
            </PetInfo>
            <Time>19.309999</Time>
            <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
          </PetRecognition>
        </VideoTargetRecResult>
      </Operation>
      <QueueId>p56347f79bc2d4551a4651e01a54a6fce</QueueId>
      <StartTime>2022-11-21T20:32:45+0800</StartTime>
      <State>Success</State>
      <Tag>VideoTargetRec</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	同提交视频目标检测接口中的 Response.JobsDetail	Container 数组

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.CosH	自定义 Header 的值	Strin

	eaders		g

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，格式为 XML

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-11-21T20:32:44+0800</CreationTime>
    <EndTime>2022-11-21T20:32:54+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>input/car.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>a993e661c699811ed9588038f7fb96507</JobId>
    <Message>TaskFinish</Message>
    <Operation>
      <JobLevel>0</JobLevel>
      <VideoTargetRec>
        <Body>true</Body>
        <Car>true</Car>
        <Pet>true</Pet>
      </VideoTargetRec>
      <VideoTargetRecResult>
        <BodyRecognition>
          <BodyInfo>
            <Location>
              <Height>382</Height>
              <Width>193</Width>
```

```
            <X>469</X>
            <Y>67</Y>
          </Location>
          <Name>person</Name>
          <Score>86</Score>
        </BodyInfo>
        <Time>1.206000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
      </BodyRecognition>
      <BodyRecognition>
        <BodyInfo>
          <Location>
            <Height>349</Height>
            <Width>198</Width>
            <X>315</X>
            <Y>177</Y>
          </Location>
          <Name>person</Name>
          <Score>83</Score>
        </BodyInfo>
        <Time>2.413000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
      </BodyRecognition>
      <BodyRecognition>
        <BodyInfo>
          <Location>
            <Height>366</Height>
            <Width>190</Width>
            <X>322</X>
            <Y>126</Y>
          </Location>
          <Name>person</Name>
          <Score>85</Score>
        </BodyInfo>
        <Time>3.620000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
      </BodyRecognition>
      <BodyRecognition>
        <BodyInfo>
          <Location>
            <Height>29</Height>
            <Width>21</Width>
            <X>807</X>
            <Y>195</Y>
          </Location>
```

```
            <Name>person</Name>
            <Score>74</Score>
        </BodyInfo>
        <Time>9.655000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
    </BodyRecognition>
    <CarRecognition>
        <CarInfo>
            <Location>
                <Height>526</Height>
                <Width>911</Width>
                <X>48</X>
                <Y>0</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>72</Score>
        </CarInfo>
        <Time>0.000000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
    </CarRecognition>
    <CarRecognition>
        <CarInfo>
            <Location>
                <Height>459</Height>
                <Width>744</Width>
                <X>9</X>
                <Y>48</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>87</Score>
        </CarInfo>
        <Time>6.034000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
    </CarRecognition>
    <CarRecognition>
        <CarInfo>
            <Location>
                <Height>284</Height>
                <Width>176</Width>
                <X>781</X>
                <Y>235</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>92</Score>
        </CarInfo>
```

```
        <CarInfo>
          <Location>
            <Height>231</Height>
            <Width>290</Width>
            <X>0</X>
            <Y>231</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>91</Score>
        </CarInfo>
        <CarInfo>
          <Location>
            <Height>132</Height>
            <Width>165</Width>
            <X>0</X>
            <Y>216</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>85</Score>
        </CarInfo>
        <CarInfo>
          <Location>
            <Height>84</Height>
            <Width>107</Width>
            <X>733</X>
            <Y>219</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>81</Score>
        </CarInfo>
        <CarInfo>
          <Location>
            <Height>457</Height>
            <Width>525</Width>
            <X>237</X>
            <Y>40</Y>
          </Location>
          <Name>vehicle</Name>
          <Score>71</Score>
        </CarInfo>
        <Time>8.448000</Time>
        <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-123456
      </CarRecognition>
      <CarRecognition>
        <CarInfo>
```

```
            <Location>
              <Height>233</Height>
              <Width>270</Width>
              <X>0</X>
              <Y>232</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>89</Score>
          </CarInfo>
          <CarInfo>
            <Location>
              <Height>272</Height>
              <Width>199</Width>
              <X>760</X>
              <Y>234</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>78</Score>
          </CarInfo>
          <CarInfo>
            <Location>
              <Height>140</Height>
              <Width>153</Width>
              <X>0</X>
              <Y>214</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>70</Score>
          </CarInfo>
          <Time>9.655000</Time>
          <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-123456
        </CarRecognition>
        <CarRecognition>
          <CarInfo>
            <Location>
              <Height>125</Height>
              <Width>150</Width>
              <X>189</X>
              <Y>282</Y>
            </Location>
            <Name>vehicle</Name>
            <Score>90</Score>
          </CarInfo>
          <CarInfo>
            <Location>
```

```
            <Height>207</Height>
            <Width>191</Width>
            <X>0</X>
            <Y>330</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>86</Score>
    </CarInfo>
    <CarInfo>
        <Location>
            <Height>62</Height>
            <Width>96</Width>
            <X>99</X>
            <Y>288</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>81</Score>
    </CarInfo>
    <CarInfo>
        <Location>
            <Height>58</Height>
            <Width>43</Width>
            <X>378</X>
            <Y>282</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>74</Score>
    </CarInfo>
    <CarInfo>
        <Location>
            <Height>22</Height>
            <Width>27</Width>
            <X>342</X>
            <Y>271</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>73</Score>
    </CarInfo>
    <CarInfo>
        <Location>
            <Height>85</Height>
            <Width>107</Width>
            <X>208</X>
            <Y>229</Y>
        </Location>
```

```
        <Name>vehicle</Name>
        <Score>70</Score>
      </CarInfo>
      <CarInfo>
        <Location>
          <Height>43</Height>
          <Width>31</Width>
          <X>313</X>
          <Y>271</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>70</Score>
      </CarInfo>
      <Time>10.862000</Time>
      <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
    </CarRecognition>
    <CarRecognition>
      <CarInfo>
        <Location>
          <Height>338</Height>
          <Width>817</Width>
          <X>61</X>
          <Y>144</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>88</Score>
      </CarInfo>
      <Time>19.309999</Time>
      <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
    </CarRecognition>
    <CarRecognition>
      <CarInfo>
        <Location>
          <Height>465</Height>
          <Width>854</Width>
          <X>48</X>
          <Y>15</Y>
        </Location>
        <Name>vehicle</Name>
        <Score>80</Score>
      </CarInfo>
      <Time>27.757999</Time>
      <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345
    </CarRecognition>
    <Duration>33171</Duration>
```

```
            <PetRecognition>
               <PetInfo>
                  <Location>
                     <Height>92</Height>
                     <Width>210</Width>
                     <X>622</X>
                     <Y>148</Y>
                  </Location>
                  <Name>dog</Name>
                  <Score>81</Score>
               </PetInfo>
               <Time>19.309999</Time>
               <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345(
            </PetRecognition>
         </VideoTargetRecResult>
      </Operation>
      <QueueId>p56347f79bc2d4551a4651e01a54a6fce</QueueId>
      <StartTime>2022-11-21T20:32:45+0800</StartTime>
      <State>Success</State>
      <Tag>VideoTargetRec</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": "2022-11-21T20:32:44+0800",
   "EndTime": "2022-11-21T20:32:54+0800",
   "Input": {
    "BucketId": "test-1234567890",
    "Object": "input/car.mp4",
    "Region": "ap-chongqing"
   },
   "JobId": "a993e661c699811ed9588038f7fb96507",
   "Message": "TaskFinish",
   "Operation": {
    "JobLevel": "0",
    "VideoTargetRec": {
     "Body": "true",
     "Car": "true",
```

```
      "Pet": "true"
    },
    "VideoTargetRecResult": {
      "BodyRecognition": [
        {
          "BodyInfo": {
            "Location": {
              "Height": "382",
              "Width": "193",
              "X": "469",
              "Y": "67"
            },
            "Name": "person",
            "Score": "86"
          },
          "Time": "1.206000",
          "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
        },
        {
          "BodyInfo": {
            "Location": {
              "Height": "349",
              "Width": "198",
              "X": "315",
              "Y": "177"
            },
            "Name": "person",
            "Score": "83"
          },
          "Time": "2.413000",
          "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
        },
        {
          "BodyInfo": {
            "Location": {
              "Height": "366",
              "Width": "190",
              "X": "322",
              "Y": "126"
            },
            "Name": "person",
            "Score": "85"
          },
          "Time": "3.620000",
          "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
```


版权所有：腾讯云计算（北京）有限责任公司

```
    },
    {
      "BodyInfo": {
        "Location": {
          "Height": "29",
          "Width": "21",
          "X": "807",
          "Y": "195"
        },
        "Name": "person",
        "Score": "74"
      },
      "Time": "9.655000",
      "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
    }
  ],
  "CarRecognition": [
    {
      "CarInfo": {
        "Location": {
          "Height": "526",
          "Width": "911",
          "X": "48",
          "Y": "0"
        },
        "Name": "vehicle",
        "Score": "72"
      },
      "Time": "0.000000",
      "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
    },
    {
      "CarInfo": {
        "Location": {
          "Height": "459",
          "Width": "744",
          "X": "9",
          "Y": "48"
        },
        "Name": "vehicle",
        "Score": "87"
      },
      "Time": "6.034000",
      "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
    },
```

```
{
 "CarInfo": [
  {
   "Location": {
    "Height": "284",
    "Width": "176",
    "X": "781",
    "Y": "235"
   },
   "Name": "vehicle",
   "Score": "92"
  },
  {
   "Location": {
    "Height": "231",
    "Width": "290",
    "X": "0",
    "Y": "231"
   },
   "Name": "vehicle",
   "Score": "91"
  },
  {
   "Location": {
    "Height": "132",
    "Width": "165",
    "X": "0",
    "Y": "216"
   },
   "Name": "vehicle",
   "Score": "85"
  },
  {
   "Location": {
    "Height": "84",
    "Width": "107",
    "X": "733",
    "Y": "219"
   },
   "Name": "vehicle",
   "Score": "81"
  },
  {
   "Location": {
    "Height": "457",
```


版权所有：腾讯云计算（北京）有限责任公司

```
            "Width": "525",
            "X": "237",
            "Y": "40"
          },
          "Name": "vehicle",
          "Score": "71"
        }
      ],
      "Time": "8.448000",
      "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
    },
    {
      "CarInfo": [
        {
          "Location": {
            "Height": "233",
            "Width": "270",
            "X": "0",
            "Y": "232"
          },
          "Name": "vehicle",
          "Score": "89"
        },
        {
          "Location": {
            "Height": "272",
            "Width": "199",
            "X": "760",
            "Y": "234"
          },
          "Name": "vehicle",
          "Score": "78"
        },
        {
          "Location": {
            "Height": "140",
            "Width": "153",
            "X": "0",
            "Y": "214"
          },
          "Name": "vehicle",
          "Score": "70"
        }
      ],
      "Time": "9.655000",
```

```
      "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
    },
    {
      "CarInfo": [
        {
          "Location": {
            "Height": "125",
            "Width": "150",
            "X": "189",
            "Y": "282"
          },
          "Name": "vehicle",
          "Score": "90"
        },
        {
          "Location": {
            "Height": "207",
            "Width": "191",
            "X": "0",
            "Y": "330"
          },
          "Name": "vehicle",
          "Score": "86"
        },
        {
          "Location": {
            "Height": "62",
            "Width": "96",
            "X": "99",
            "Y": "288"
          },
          "Name": "vehicle",
          "Score": "81"
        },
        {
          "Location": {
            "Height": "58",
            "Width": "43",
            "X": "378",
            "Y": "282"
          },
          "Name": "vehicle",
          "Score": "74"
        },
        {
```

```
        "Location": {
          "Height": "22",
          "Width": "27",
          "X": "342",
          "Y": "271"
        },
        "Name": "vehicle",
        "Score": "73"
      },
      {
        "Location": {
          "Height": "85",
          "Width": "107",
          "X": "208",
          "Y": "229"
        },
        "Name": "vehicle",
        "Score": "70"
      },
      {
        "Location": {
          "Height": "43",
          "Width": "31",
          "X": "313",
          "Y": "271"
        },
        "Name": "vehicle",
        "Score": "70"
      }
    ],
    "Time": "10.862000",
    "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
  },
  {
    "CarInfo": {
      "Location": {
        "Height": "338",
        "Width": "817",
        "X": "61",
        "Y": "144"
      },
      "Name": "vehicle",
      "Score": "88"
    },
    "Time": "19.309999",
```

```
          "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
        },
        {
          "CarInfo": {
            "Location": {
              "Height": "465",
              "Width": "854",
              "X": "48",
              "Y": "15"
            },
            "Name": "vehicle",
            "Score": "80"
          },
          "Time": "27.757999",
          "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
        }
      ],
      "Duration": "33171",
      "PetRecognition": [
        {
          "PetInfo": {
            "Location": {
              "Height": "92",
              "Width": "210",
              "X": "622",
              "Y": "148"
            },
            "Name": "dog",
            "Score": "81"
          },
          "Time": "19.309999",
          "Url": "https://test-1234567890.cos.ap-chongqing.myqcloud.com/test-12345678
        }]
    }
  },
  "QueueId": "p56347f79bc2d4551a4651e01a54a6fce",
  "StartTime": "2022-11-21T20:32:45+0800",
  "State": "Success",
  "Tag": "VideoTargetRec"
 }]
}
```


版权所有：腾讯云计算（北京）有限责任公司

创建模板

最近更新时间：2023-08-01 15:45:53

功能描述

创建视频目标检测模板。

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>VideoTargetRec</Tag>
  <Name>TemplateName</Name>
  <VideoTargetRec>
    <Body>true</Body>
    <Pet>false</Pet>
    <Car>true</Car>
  </VideoTargetRec>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

具体数据描述如下：

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：VideoTargetRec	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64	String	是
VideoTargetRec	Request	视频目标检测 参数	Container	是

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

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
Body	Request.VideoTargetRec	是否开启人体检测，取值 true/false	string	false	否
Pet	Request.VideoTargetRec	是否开启宠物检测，取值 true/false	string	false	否
Car	Request.VideoTargetRec	是否开启车辆检测，取值 true/false	string	false	否

> **说明**
> Body、Pet、Car 不能同时为 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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <Template>
    <Tag>VideoTargetRec</Tag>
    <Name>TemplateName</Name>
    <State>Normal</State>
    <Tag>VideoTargetRec</Tag>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
    <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <VideoTargetRec>
      <Body>true</Body>
      <Pet>false</Pet>
      <Car>true</Car>
    </VideoTargetRec>
  </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
RequestId	Response	请求的唯一 ID	String
Template	Response	保存模板详情的容器	Container

Container 节点 Template 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

版权所有：腾讯云计算（北京）有限责任公司

Category	Response.Template	模板属性，Custom 或者 Official	String
Tag	Response.Template	模板类型，VideoTargetRec	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
VideoTargetRec	Response.Template	同请求体中的 Request.VideoTargetRec	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-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoTargetRec</Tag>
    <Name>TemplateName</Name>
    <VideoTargetRec>
        <Body>true</Body>
        <Pet>false</Pet>
        <Car>true</Car>
    </VideoTargetRec>
</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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>VideoTargetRec</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VideoTargetRec>
            <Body>true</Body>
            <Pet>false</Pet>
            <Car>true</Car>
        </VideoTargetRec>
    </Template>
</Response>
```

更新模板

最近更新时间：2023-08-11 17:56:16

功能描述

更新视频目标检测模板。

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>VideoTargetRec</Tag>
  <Name>TemplateName</Name>
  <VideoTargetRec>
    <Body>true</Body>
    <Pet>false</Pet>
    <Car>true</Car>
  </VideoTargetRec>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

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	同创建视频目标检测模板接口的 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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <Template>
    <Tag>VideoTargetRec</Tag>
    <Name>TemplateName</Name>
    <State>Normal</State>
    <Tag>VideoTargetRec</Tag>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
    <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <VideoTargetRec>
      <Body>true</Body>
      <Pet>false</Pet>
      <Car>true</Car>
    </VideoTargetRec>
  </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	同创建视频目标检测模板接口的 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-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>VideoTargetRec</Tag>
    <Name>TemplateName</Name>
    <VideoTargetRec>
        <Body>true</Body>
        <Pet>false</Pet>
        <Car>true</Car>
    </VideoTargetRec>
</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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>VideoTargetRec</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <VideoTargetRec>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Body>true</Body>
        <Pet>false</Pet>
        <Car>true</Car>
      </VideoTargetRec>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

视频人像分割
提交任务

最近更新时间：2023-08-11 18:10:33

功能描述

提交一个视频人像抠图任务。

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>SegmentVideoBody</Tag>
  <Input>
    <Object>input/test.mp4</Object>
  </Input>
  <Operation>
    <SegmentVideoBody>
      <Mode>Mask</Mode>
```

```
    </SegmentVideoBody>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
    <Output>
      <Region>ap-chongqing</Region>
      <Bucket>test-123456789</Bucket>
      <Object>output/out.mp4</Object>
    </Output>
  </Operation>
  <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：SegmentVideoBody	String	是
Input	Request	待操作的对象信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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
SegmentVideoBody	Request.Operation	视频人像抠图配置	Container	否
Output	Request.Operation	结果输出配置	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 SegmentVideoBody 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Mode	Request.Operation.SegmentVideoBody	抠图模式 • Mask：输出alpha通道结果 • Foreground：输出前景视频 • Combination：输出抠图后的前景与自定义背景合成后的视频 • 默认值：Mask	String	否
SegmentType	Request.Operation.SegmentVideoBody	抠图类型 • HumanSeg：人像抠图 • GreenScreenSeg：绿幕抠图 • SolidColorSeg：纯色背景抠图 • 默认值：HumanSeg	String	否
BackgroundRed	Request.Operation.SegmentVideoBody	mode为 Foreground 时参数生效，背景颜色为红色，取值范围 [0, 255]， 默认值为 0	String	否
BackgroundGreen	Request.Operation.SegmentVideoBody	mode为 Foreground 时参数生效，背景颜色为绿色，取值范围 [0,	String	否

		255]，默认值为 0		
BackgroundBlue	Request.Operation.SegmentVideoBody	mode为 Foreground 时参数生效，背景颜色为蓝色，取值范围 [0, 255]，默认值为 0	String	否
BackgroundLogoUrl	Request.Operation.SegmentVideoBody	传入背景文件。mode为 Combination 时，此参数必填，背景文件需与源文件在同存储桶下	String	否
BinaryThreshold	Request.Operation.SegmentVideoBody	调整抠图的边缘位置，取值范围为[0, 255]，默认值为 0	String	否
RemoveRed	Request.Operation.SegmentVideoBody	纯色背景抠图的背景色（红），当 SegmentType 为 SolidColorSeg 生效，取值范围为 [0, 255]，默认值为 0	String	否
RemoveGreen	Request.Operation.SegmentVideoBody	纯色背景抠图的背景色（绿），当 SegmentType 为 SolidColorSeg 生效，取值范围为 [0, 255]，默认值为 0	String	否
RemoveBlue	Request.Operation.SegmentVideoBody	纯色背景抠图的背景色（蓝），当 SegmentType 为 SolidColorSeg 生效，取值范围为 [0, 255]，默认值为 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
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>SegmentVideoBody</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/test.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <SegmentVideoBody>
        <Mode>Mask</Mode>
      </SegmentVideoBody>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-123456789</Bucket>
        <Object>output/out.mp4</Object>
      </Output>
    </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

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	创建任务的 ID	String
Tag	Response.JobsDetail	创建任务的 Tag：SegmentVideoBody	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
SegmentVideoBody	Response.JobsDetail.Operation	同请求中的 Request.Operation.SegmentVideoBody	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
MediaInfo	Response.JobsDetail.Operation	输出文件的媒体信息，任务未完成时不返回，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SegmentVideoBody</Tag>
    <Input>
        <Object>input/test.mp4</Object>
    </Input>
    <Operation>
        <SegmentVideoBody>
            <Mode>Mask</Mode>
        </SegmentVideoBody>
```

```
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.mp4</Object>
        </Output>
    </Operation>
    <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>SegmentVideoBody</Tag>
        <Input>
            <BucketId>test-123456789</BucketId>
            <Object>input/test.mp4</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <SegmentVideoBody>
                <Mode>Mask</Mode>
            </SegmentVideoBody>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
```

```
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-123456789</Bucket>
            <Object>output/out.mp4</Object>
        </Output>
    </Operation>
  </JobsDetail>
</Response>
```

回调信息

最近更新时间：2023-08-14 17:01:45

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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>
      <BucketId>test-123456789</BucketId>
      <Object>input/test.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <Index>1</Index>
            <Language>und</Language>
            <SampleFmt>fltp</SampleFmt>
            <SampleRate>44100</SampleRate>
            <StartTime>0.000000</StartTime>
            <Timebase>1/44100</Timebase>
          </Audio>
          <Video>
            <AvgFps>50.000</AvgFps>
            <Bitrate>1031968</Bitrate>
            <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)</CodecL
            <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/out.mp4</ObjectName>
          </Md5Info>
          <ObjectName>output/out.mp4</ObjectName>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <ObjectPrefix/>
            <Region>ap-chongqing</Region>
          </OutputFile>
        </MediaResult>
        <SegmentVideoBody>
          <Mode>Mask</Mode>
        </SegmentVideoBody>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-123456789</Bucket>
          <Object>output/out.mp4</Object>
        </Output>
        <UserData>This is my SegmentVideoBody job.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>pbde29b842b1b4e2ea4f0b20d264fcec2</QueueId>
      <StartTime>2022-07-25T16:35:39+0800</StartTime>
      <State>Success</State>
      <Tag>SegmentVideoBody</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	同提交视频人像抠图接口中的 Response.JobsDetail	Container 数组

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>
      <BucketId>test-123456789</BucketId>
      <Object>input/test.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>ac34be3aa0bf411ed9ce39d7cc972e427</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>
        <Video>
            <AvgFps>50.000</AvgFps>
            <Bitrate>1031968</Bitrate>
            <CodecLongName>H.265 / HEVC (High Efficiency Video Coding)</CodecL
            <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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
        </MediaInfo>
        <MediaResult>
          <OutputFile>
            <Bucket>test-123456789</Bucket>
            <Md5Info>
              <Md5>2db78d168edf2f883adce89a1b4ca033</Md5>
              <ObjectName>output/out.mp4</ObjectName>
            </Md5Info>
            <ObjectName>output/out.mp4</ObjectName>
            <ObjectPrefix/>
            <Region>ap-chongqing</Region>
          </OutputFile>
        </MediaResult>
        <SegmentVideoBody>
          <Mode>Mask</Mode>
        </SegmentVideoBody>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-123456789</Bucket>
          <Object>output/out.mp4</Object>
        </Output>
        <UserData>This is my SegmentVideoBody job.</UserData>
        <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>pbde29b842b1b4e2ea4f0b20d264fcec2</QueueId>
      <StartTime>2022-07-25T16:35:39+0800</StartTime>
      <State>Success</State>
      <Tag>SegmentVideoBody</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": {
      "BucketId": "test-123456789",
      "Object": "input/test.mp4",
      "Region": "ap-chongqing"
    },
```

```
    "JobId": "ac34be3aa0bf411ed9ce39d7cc972e427",
    "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 Coding)",
            "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",
```


版权所有：腾讯云计算（北京）有限责任公司

```
                "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/out.mp4"
            }],
            "ObjectName": ["output/out.mp4"],
            "Region": "ap-chongqing"
        }
    },
    "SegmentVideoBody": {
        "Mode": "Mask"
    },
    "Output": {
        "Region": "ap-chongqing",
        "Bucket": "test-123456789",
        "Object": "output/out.mp4"
    },
    "UserData": "This is my SegmentVideoBody job.",
    "JobLevel": "0"
  },
  "QueueId": "pbde29b842b1b4e2ea4f0b20d264fcec2",
  "StartTime": "2022-07-25T16:35:39+0800",
  "State": "Success",
  "Tag": "SegmentVideoBody"
}]
}
```

文档预览
查询文档预览开通状态

最近更新时间：2023-06-20 17:32:13

功能描述

DescribeDocProcessBuckets 接口用于查询已经开通文档预览功能的 Bucket。

请求

请求示例

```
GET /docbucket?pageNumber=1&pageSize=2 HTTP/1.1
Host: ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

请求参数

名称	描述	类型	是否必选
regions	地域信息，以“,”分隔字符串，支持 All、ap-shanghai、ap-beijing	string	否
bucketNam	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	stri	否

es		ng	
bucketName	存储桶名称前缀，前缀搜索	string	否
pageNumber	第几页	string	否
pageSize	每页个数	string	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
    <TotalCount>1</TotalCount>
    <RequestId>RequestId</RequestId>
    <PageNumber>1</PageNumber>
    <PageSize>2</PageSize>
    <DocBucketList>
        <Name>BucketName-APPID</Name>
        <CreateTime>Time</CreateTime>
        <Region>Region</Region>
        <AliasBucketId/>
        <BucketId>BucketName-APPID</BucketId>
    </DocBucketList>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

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

TotalCount	Response	文档预览 Bucket 总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
DocBucketList	Response	文档预览 Bucket 列表	Container

Container 节点 DocBucketList 的内容：

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

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /docbucket?regions=ap-chongqing HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: cos-python-sdk-v5.3.2
Host: ci.ap-chongqing.myqcloud.com
Content-Type: application/xml
Authorization: Authorization
```

响应

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 07:00:43 GMT
Content-Type: application/xml
Content-Length: 843
```

```
Connection: keep-alive
Server: tencent-ci
x-ci-request-id: NWYxZTdiOWJfYzc2OTQzNjRfM2Qz****

<?xml version="1.0" encoding="utf-8"?>
<Response>
    <TotalCount>3</TotalCount>
    <RequestId>NWYxZTdiOWJfYzc2OTQzNjRfM2Qz****</RequestId>
    <PageNumber>1</PageNumber>
    <PageSize>10</PageSize>
    <DocBucketList>
        <Name>examplebucket-1250000000</Name>
        <CreateTime>2020-07-27T10:54:42+0800</CreateTime>
        <Region>ap-chongqing</Region>
        <AliasBucketId/>
        <BucketId>examplebucket-1250000000</BucketId>
    </DocBucketList>
    <DocBucketList>
        <Name>examplebucket-1250000000</Name>
        <CreateTime>2020-07-24T22:42:26+0800</CreateTime>
        <Region>ap-chongqing</Region>
        <AliasBucketId/>
        <BucketId>examplebucket-1250000000</BucketId>
    </DocBucketList>
</Response>
```

文档转码
文档转码同步请求

最近更新时间：2023-07-12 11:42:54

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

```
GET /<ObjectKey>?ci-process=doc-preview&page=<page>&srcType=<srcType>&Image
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。

版权所有：腾讯云计算（北京）有限责任公司

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
```

版权所有：腾讯云计算（北京）有限责任公司

```
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/thumbna
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: -3015
```

版权所有：腾讯云计算（北京）有限责任公司

异步处理任务接口
提交文档转码任务

最近更新时间：2023-07-12 16:55:02

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>
</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
Operatio	Reques	操作规则	Containe	是

n	t		r	
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	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	否
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	String	否

		• pdf，转成 pdf 格式文件（暂不支持指定页数）		
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

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` ，按页导出图片，`TgtType="png"/"jpg"` 时生效	Int	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` ，	String	是

		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		

响应

响应头

此接口仅返回公共响应头部，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/aHR0cDovL3Rlc3QwMDUtMTI1MTcwNDcwOC5
    </DocProcess>
  </Operation>
  <Tag>DocProcess</Tag>
</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/aHR0cDovL3Rlc3QwMDUtMTI1M
                <SrcType/>
                <StartPage>1</StartPage>
                <TgtType>png</TgtType>
            </DocProcess>
            <Output>
                <Bucket>examplebucket-1250000000</Bucket>
                <Object>big/test-${Number}</Object>
                <Region>ap-chongqing</Region>
            </Output>
```

```
            </Operation>
            <QueueId>p532fdead78444e649e1a4467c1cd19d3</QueueId>
            <State>Submitted</State>
            <Tag>DocProcess</Tag>
        </JobsDetail>
</Response>
```

查询指定的文档转码任务

最近更新时间：2023-07-12 00:33:51

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。

请求参数

参数名称	描述	类型	是否必选
jobId	要查询的任务ID，可通过提交文档预览任务的响应中获取。	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> </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>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	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

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

版权所有：腾讯云计算（北京）有限责任公司

SuccPage Count	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
TotalSheet Count	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
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-s
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>
```

版权所有：腾讯云计算（北京）有限责任公司

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/>
```

版权所有：腾讯云计算（北京）有限责任公司

```
                <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：2023-07-13 14:45:31

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
tag	无	任务的 Tag：DocProcess	String	是
queueId	无	拉取该队列 ID 下的任务，可在任务响应内容或控制台中	String	否

		获取		
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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
```

异步处理队列接口
查询文档转码队列

最近更新时间：2023-06-20 17:32:14

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </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

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：2023-06-20 17:32:14

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>NTk0MjdmODIfMjQ4OGY3XzYzYzhf****</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	Resp	请求的唯一 ID	Strin

	onse		g
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>
```

版权所有：腾讯云计算（北京）有限责任公司

支持的功能和效果

最近更新时间：2021-07-09 10:45:42

以下是文档转图片功能支持的功能和效果：

类型	文件名后缀	支持的功能和效果
演示	ppt、pptx、pptm、ppsx、ppsm、pps、potx、potm、dpt、dps	幻灯片背景、对象边框/填充、对象阴影/倒影、文本（横/竖向文本框、占位符、形状）、图片、链接、艺术字、形状、表格、图片、图表、公式、符号、链接、脑图、流程图
文字	doc、dot、wps、wpt、docx、dotx、docm、dotm	字属性、段属性、图片、文字表、链接、形状、代码块、日期、水印、评论、页眉页脚
表格	xls、xlt、et、xlsx、xltx、xlsm、xltm	字属性、段属性、自选图形、艺术字、公式、图片、流程图、脑图
PDF	pdf	背景、文字（字体、字号、字色、加粗、斜体）、画笔、高亮、下划线、删除线

版权所有：腾讯云计算（北京）有限责任公司

文档转html
文档转html同步请求

最近更新时间：2023-08-04 14:34:52

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。
- 您可在 文档预览接入体验工具 中自行调整参数，体验接入文档预览后的预览效果，同时该工具可以根据相应配置项生成代码，助力您快速接入文档预览。
- 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, language: "zh" }}，支持的配置参考 自定义配置项说明 。 htmlParams支持的特殊配置：语言切换，通过 commonOptions 的 language 参数指定预览语言，支持"zh"、"en “，默认为"zh"。	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、

版权所有：腾讯云计算（北京）有限责任公司

rtf、txt、log、xml、htm、html。
- 输入文件大小限制在200MB之内。
- 输入文件页数限制在5000页之内。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

此接口请求的响应体为html预览内容，展示效果可参见下方实际案例。

错误码

该请求无特有错误信息，常见的错误信息请参见 错误码 文档。

实际案例

案例一：简单预览

将存储桶 examples-1258125638 中的 ObjectKey 为 test.pptx 的 PPT 文件进行在线预览，最终 URL 如下：

```
https://examples-1258125638.cos.ap-guangzhou.myqcloud.com/test.pptx?ci-process=do
```

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=do
```

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-sca
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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>
```

案例三：切换预览语言为英文

将存将存储桶 examples-1258125638 中的 ObjectKey 为 test.pptx 的 PPT 文件进行在线预览。

设置 htmlParams 参数（htmlParams为 {"commonOptions":{"language": "en"}}，通过 URL 安全的 Base64编码得到 htmlParams=eyJjb21tb25PcHRpb25zIjp7Imxhbmd1YWdlIjoiZW4ifX0 ）

最终 URL 如下：

```
https://examples-1258125638.cos.ap-guangzhou.myqcloud.com/test.pptx?ci-process=do
```

直接在浏览器中打开链接，生成的预览页面如下图所示：


1
数据万象
文档预览服务介绍
2
3
4
5
6
7
8
9
腾讯云
数据万象
文档预览服务介绍
No notes
Full Screen
PowerPoint 1/11
70%

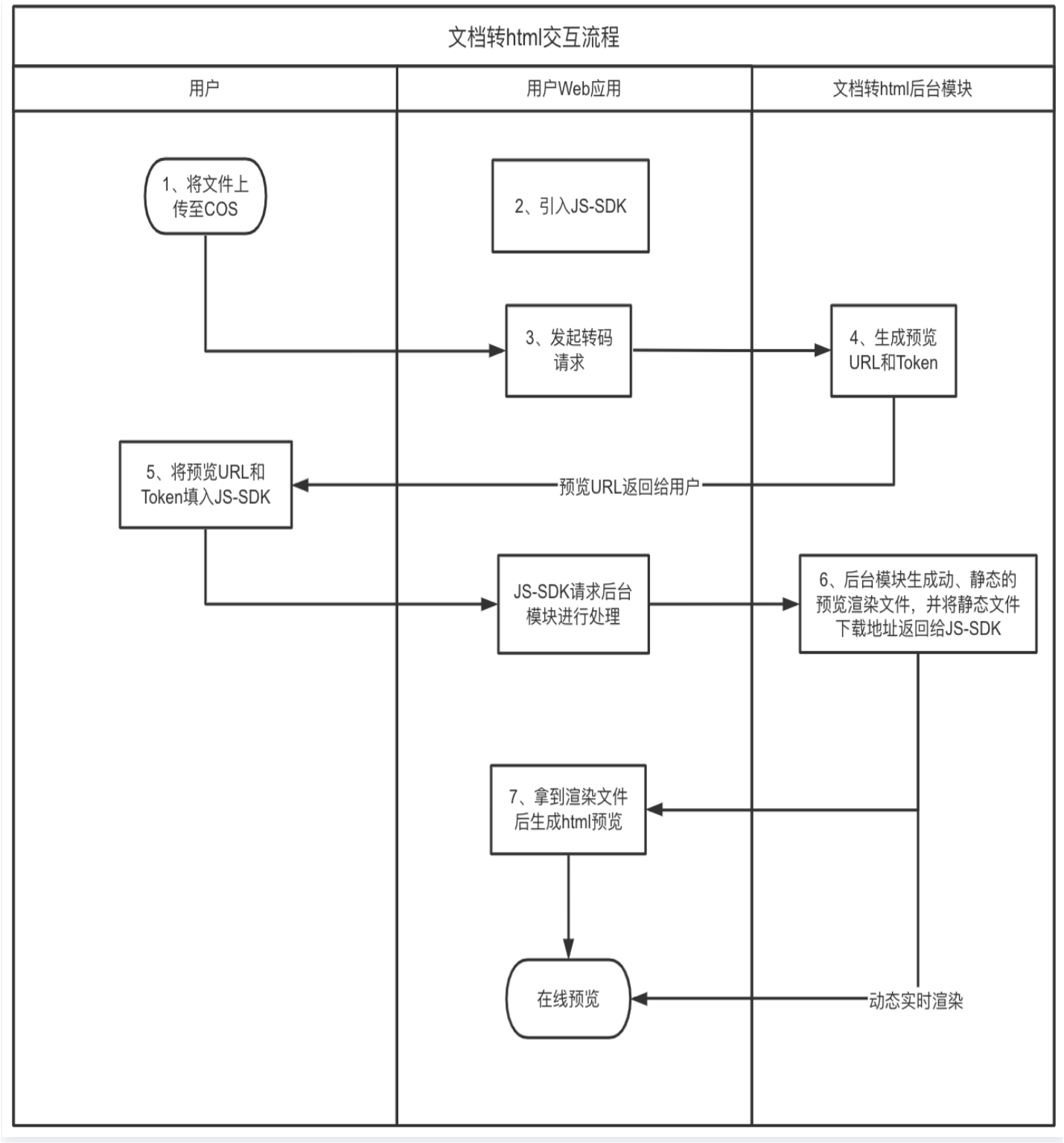
自定义配置概述

最近更新时间：2023-06-02 11:24:09

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（仅支持 webview 方

版权所有：腾讯云计算（北京）有限责任公司

	式接入）
Android	QQ 内置浏览器、QQ 小程序、微信内置浏览器、微信小程序（仅支持 webview 方式接入）
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/test.p
    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=Auth
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

```
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>
```

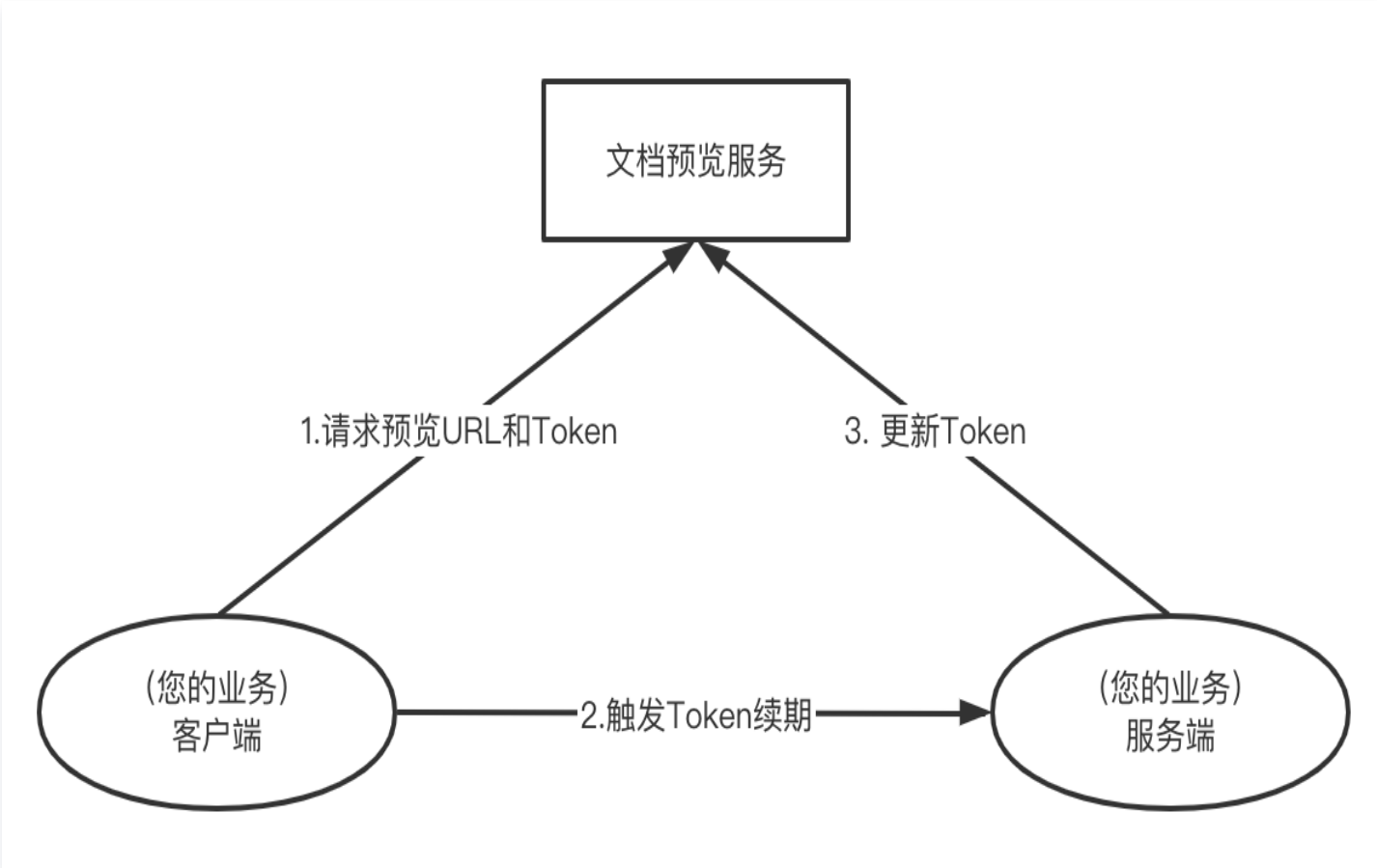
```
var demo = COSDocPreviewSDK.config({
    url: '在线文档预览地址',    // 步骤3中获取到的在线文档预览地址
    mount: document.querySelector('.custom-mount')    // 挂载节点，默认会挂载到 body 下，
})
```

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&tokenui
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****
```

版权所有：腾讯云计算（北京）有限责任公司

```
{
 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
});
```

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 。

版权所有：腾讯云计算（北京）有限责任公司

配置项说明

最近更新时间：2022-06-24 17:53:30

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

- 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

版权所有：腾讯云计算（北京）有限责任公司

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：2022-06-24 17:53:34

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
```

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

展开全部

- 获取总页数
- 获取当前页
- 跳转到指定页
- 设置评论显示

版权所有：腾讯云计算（北京）有限责任公司

设置目录显示

视图缩放

表格文件

展开全部

获取所有sheet名称

获取当前sheet名称

切换到指定sheet

视图缩放

演示文件

展开全部

获取总页数

获取当前页

跳转到指定页

播放幻灯片

退出播放

幻灯片下一步动画

幻灯片上一动画

获取播放状态

视图播放

版权所有：腾讯云计算（北京）有限责任公司

PDF 文件

展开全部

- 获取总页数
- 获取当前页
- 跳转到指定页
- 设置页面渲染模式
- 设置目录显示
- 视图缩放
- 事件监听

版权所有：腾讯云计算（北京）有限责任公司

支持的字体和动画

最近更新时间：2021-05-17 09:44:59

简介

文档转 HTML 功能支持预览多种字体，并可预览演示文件中的动画效果，具体请参见下方的字体支持表和动画支持表。

> **说明**
> 使用文档转 HTML 功能，实际的预览效果可能因文档格式的不同而有所差异。

支持的字体

文档预览支持如下所示的字体，部分字体**仅支持 Word 文档**。不支持的字体会在展示时匹配相近字体，可能会出现字体效果与本地不符的情况。

中文字体（简/繁）

仅支持 Word 文档

字体名称	字体名称	字体名称
幼圆	华文行楷	方正姚体
新宋体	华文仿宋	方正舒体
微软雅黑	华文彩云	等线
微软雅黑 Light	黑体	等线 Light
腾讯体 W3	汉仪智楷繁	PMingLiU-ExtB （新　明　-ExtB）
腾讯体 W7	汉仪智草繁	MingLiU-ExtB （　明　-ExtB）
宋体	汉仪赵楷繁	MingLiU_HKSCS-ExtB （　明　_HKSCS-ExtB）
隶书	汉仪王行繁	Microsoft YaHei UI
楷体	汉仪唐隶繁	Microsoft YaHei UI Light
华文中宋	汉仪欧行繁	Microsoft JhengHei UI
华文新魏	汉仪陆行繁	Microsoft JhengHei UI Light
华文细黑	汉仪柳楷繁	Microsoft JhengHei

版权所有：腾讯云计算（北京）有限责任公司

华文宋体	汉仪衡方碑繁	Microsoft JhengHei Light
华文隶书	汉仪褚楷繁	Adobe 宋体 Std
华文楷体	汉仪曹隶繁	–
华文琥珀	仿宋	–

支持全部文档类型

字体名称	字体名称
汉仪中黑 KW	汉仪楷体 KW
汉仪中等线 KW	汉仪君黑 KW 55J
汉仪书宋二 KW	汉仪君黑 KW 85J
汉仪旗黑 KW 85S	汉仪仿宋 KW
汉仪旗黑 KW 55S	–

其他语言字体

仅支持 Word 文档

字体名称	字体名称	字体名称
Agency FB	Gadugi	Palatino Linotype
Algerian	Garamond	Papyrus
Arial Black	Gigi	Parchment
Arial Narrow	Gill Sans MT	Perpetua
Arial Rounded MT Bold	Gill Sans MT Condensed	Perpetua Titling MT
Bahnschrift	Gill Sans MT Ext Condensed Bold	Playbill
Baskerville Old Face	Gill Sans Ultra Bold	Poor Richard
Bauhaus 93	Gill Sans Ultra Bold Condensed	Pristina
Bell MT	Gloucester MT Extra	Rage Italic

版权所有：腾讯云计算（北京）有限责任公司

	Condensed	
Berlin Sans FB	Goudy Old Style	Raleway
Berlin Sans FB Demi	Goudy Stout	Raleway Black
Bernard MT Condensed	Haettenschweiler	Raleway ExtraBold
Blackadder ITC	Harlow Solid Italic	Raleway ExtraLight
Bodoni MT	Harrington	Raleway Light
Bodoni MT Black	High Tower Text	Raleway Medium
Bodoni MT Condensed	HoloLens MDL2 Assets	Raleway SemiBold
Bodoni MT Poster Compressed	Impact	Raleway Thin
Book Antiqua	Imprint MT Shadow	Ravie
Bookman Old Style	Informal Roman	Roboto
Bookshelf Symbol 7	Ink Free	Roboto Black
Bradley Hand ITC	Javanese Text	Roboto Light
Britannic Bold	Jokerman	Roboto Medium
Broadway	Juice ITC	Roboto Thin
Brush Script MT	Kristen ITC	Roboto Slab
C059	Kunstler Script	Rockwell
Calibri	Leelawadee	Rockwell Condensed
Calibri Light	Leelawadee UI	Rockwell Extra Bold
Californian FB	Leelawadee UI Semilight	Script MT Bold
Calisto MT	Lucida Bright	Segoe MDL2 Assets
Cambria	Lucida Calligraphy	Segoe Print
Cambria Math	Lucida Console	Segoe Script
Candara	Lucida Fax	Segoe UI

版权所有：腾讯云计算（北京）有限责任公司

Candara Light	Lucida Handwriting	Segoe UI Black
Castellar	Lucida Sans	Segoe UI Light
Centaur	Lucida Sans Typewriter	Segoe UI Semibold
Century	Lucida Sans Unicode	Segoe UI Semilight
Century Gothic	Magneto	Segoe UI Emoji
Century Schoolbook	Maiandra GD	Segoe UI Historic
Chiller	Malgun Gothic ( )	Segoe UI Symbol
Colonna MT	Malgun Gothic Semilight (Semilight)	Showcard Gothic
Consolas	Matura MT Script Capitals	SimSun-ExtB
Constantia	Merriweather	Sitka Banner
Cooper Black	Merriweather Black	Sitka Display
Copperplate Gothic Bold	Merriweather Light	Sitka Heading
Copperplate Gothic Light	Microsoft Himalaya	Sitka Small
Corbel	Microsoft New Tai Lue	Sitka Subheading
Corbel Light	Microsoft PhagsPa	Sitka Text
Courier	Microsoft Sans Serif	Small Fonts
Curlz MT	Microsoft Tai Le	Snap ITC
D050000L	Microsoft Uighur	Standard Symbols PS
DejaVu Sans	Microsoft Yi Baiti	Stencil
DejaVu Sans Condensed	Mistral	Sylfaen
DejaVu Sans Light	Modern No. 20	Symbol
Dubai	Mongolian Baiti	Symbol Tiger
Dubai Light	Monotype Corsiva	Symbol Tiger Expert

版权所有：腾讯云计算（北京）有限责任公司

Dubai Medium	MS Gothic (MS ゴシック)	System
Ebrima	MS Outlook	Tahoma
Edwardian Script ITC	MS Pgothic (MS Pゴシック)	Tempus Sans ITC
Elephant	MS Reference Sans Serif	TencentSans Korean W3 (W3)
Engravers MT	MS Reference Specialty	TencentSans Korean W7 (W7)
Eras Bold ITC	MS Sans Serif	Terminal
Eras Demi ITC	MS Serif	Tiger
Eras Light ITC	MS UI Gothic	Tiger Expert
Eras Medium ITC	MT Extra	Trebuchet MS
Euclid	MT Extra Tiger	Tw Cen MT
Euclid Extra	MV Boli	Tw Cen MT Condensed
Euclid Fraktur	Myanmar Text	Tw Cen MT Condensed Extra Bold
Euclid Math One	Niagara Engraved	URW Bookman
Euclid Math Two	Niagara Solid	URW Gothic
Euclid Symbol	Nimbus Mono PS	Viner Hand ITC
Felix Titling	Nimbus Roman	Vivaldi
Fences	Nimbus Sans	Vladimir Script
Fixedsys	Nimbus Sans Narrow	Wide Latin
Footlight MT Light	Nirmala UI	Wingdings
Forte	Nirmala UI Semilight	Wingdings 2
Franklin Gothic Book	OCR A Extended	Wingdings 3
Franklin Gothic Demi	Old English Text MT	Yu Gothic (游ゴシック)
Franklin Gothic Demi Cond	Onyx	Yu Gothic Light (游ゴシック Light)

版权所有：腾讯云计算（北京）有限责任公司

Franklin Gothic Heavy	Open Sans	Yu Gothic Medium（游ゴシック Medium）
Franklin Gothic Medium	Open Sans Extrabold	Yu Gothic UI
Franklin Gothic Medium Cond	Open Sans Light	Yu Gothic UI Light
Freestyle Script	Open Sans Semibold	Yu Gothic UI Semibold
French Script MT	P052	Yu Gothic UI Semilight
Gabriola	Palace Script MT	Z003

支持全部文档类型

字体名称	字体名称	字体名称
Arial	Noto Sans Gujarati UI	Noto Sans Shavian
Comic Sans MS	Noto Sans Gurmukhi	Noto Sans Sinhala
Courier New	Noto Sans Gurmukhi UI	Noto Sans Sinhala UI
Georgia	Noto Sans Hanunoo	Noto Sans Sundanese
Kingsoft Confetti	Noto Sans Hebrew	Noto Sans Syloti Nagri
Kingsoft Extra	Noto Sans Imperial Aramaic (Noto Sans ImpAramaic)	Noto Sans Symbols
Kingsoft Mark	Noto Sans Inscriptional Pahlavi (Noto Sans InsPahlavi)	Noto Sans Symbols2
Kingsoft Math	Noto Sans Inscriptional Parthian (Noto Sans InsParthi)	Noto Sans Syriac Eastern
Kingsoft Sign	Noto Sans Javanese	Noto Sans Syriac Estrangela
Kingsoft Stress	Noto Sans Kaithi	Noto Sans Syriac

版权所有：腾讯云计算（北京）有限责任公司

		Western
Noto Emoji	Noto Sans Kannada	Noto Sans Tagalog
Noto Kufi Arabic	Noto Sans Kannada UI	Noto Sans Tagbanwa
Noto Mono	Noto Sans Kayah Li	Noto Sans Tai Le
Noto Naskh Arabic	Noto Sans Kharoshthi	Noto Sans Tai Tham
Noto Naskh Arabic UI	Noto Sans Khmer	Noto Sans Tai Viet
Noto Nastaliq Urdu	Noto Sans Khmer UI	Noto Sans Tamil
Noto Sans	Noto Sans Lao	Noto Sans Tamil UI
Noto Sans Adlam	Noto Sans Lao UI	Noto Sans Telugu
Noto Sans Adlam Unjoined	Noto Sans Lepcha	Noto Sans Telugu UI
Noto Sans Anatolian Hieroglyphs (Noto Sans AnatoHiero)	Noto Sans Limbu	Noto Sans Thaana
Noto Sans Arabic	Noto Sans Linear B	Noto Sans Thai
Noto Sans Arabic UI	Noto Sans Lisu	Noto Sans Thai UI
Noto Sans Armenian	Noto Sans Lycian	Noto Sans Tibetan
Noto Sans Avestan	Noto Sans Lydian	Noto Sans Tifinagh
Noto Sans Balinese	Noto Sans Malayalam	Noto Sans Ugaritic
Noto Sans Bamum	Noto Sans Malayalam UI	Noto Sans Vai
Noto Sans Batak	Noto Sans Mandaic	Noto Sans Yi
Noto Sans Bengali	Noto Sans Meetei Mayek	Noto Serif
Noto Sans Bengali UI	Noto Sans Mongolian	Noto Serif Armenian

版权所有：腾讯云计算（北京）有限责任公司

Noto Sans Brahmi	Noto Sans Mono	Noto Serif Bengali
Noto Sans Buginese	Noto Sans Myanmar	Noto Serif CJK SC
Noto Sans Buhid	Noto Sans Myanmar UI	Noto Serif CJK TC
Noto Sans Canadian Aboriginal	Noto Sans New Tai Lue	Noto Serif Devanagari
Noto Sans Carian	Noto Sans Nko (Noto Sans N'Ko)	Noto Serif Display (Noto Serif Disp)
Noto Sans Chakma	Noto Sans Ogham	Noto Serif Ethiopic
Noto Sans Cham	Noto Sans Ol Chiki	Noto Serif Georgian
Noto Sans Cherokee	Noto Sans Old Italic	Noto Serif Gujarati
Noto Sans Coptic	Noto Sans Old Persian	Noto Serif Hebrew
Noto Sans Cuneiform	Noto Sans Old South Arabian (Noto Sans OldSouArab)	Noto Serif Kannada
Noto Sans Cypriot	Noto Sans Old Turkic	Noto Serif Khmer
Noto Sans Deseret	Noto Sans Oriya	Noto Serif Lao
Noto Sans Devanagari	Noto Sans Oriya UI	Noto Serif Malayalam
Noto Sans Devanagari UI	Noto Sans Osage	Noto Serif Myanmar
Noto Sans Display (Noto Sans Disp)	Noto Sans Osmanya	Noto Serif Sinhala
Noto Sans Egyptian Hieroglyphs (Noto Sans EgyptHiero)	Noto Sans Phags Pa	Noto Serif Tamil
Noto Sans Ethiopic	Noto Sans Phoenician	Noto Serif Telugu
Noto Sans Georgian	Noto Sans Rejang	Noto Serif Thai
Noto Sans Glagolitic	Noto Sans Runic	Times New Roman
Noto Sans Gothic	Noto Sans Samaritan	Verdana

版权所有：腾讯云计算（北京）有限责任公司

Noto Sans Gujarati	Noto Sans Saurashtra	Webdings

支持的演示文件动画

演示文件包含的格式有：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。以下表中的动画，除**随机效果**外，均支持**播放时机**、**速度**、**重复**和**延迟**的调节。

进入和退出动画

动画名称	属性	动画名称	属性	动画名称	属性
百叶窗	支持方向	圆形扩展	支持方向	放大	不支持
擦除	支持方向	渐变	不支持	飞旋	不支持
出现	不支持	渐变式回旋	不支持	浮动	不支持
飞入、飞出	支持方向	渐变式缩放	不支持	光速	不支持
盒状	支持方向	展开-细微型	不支持	滑翔	不支持
缓慢进入、缓慢退出	支持方向	翻转式由远及近	不支持	挥鞭式	不支持
阶梯状	支持方向	回旋	不支持	挥舞	不支持
菱形	支持方向	渐入,渐出	不支持	空翻	不支持
轮子	支持辐射状	上升	不支持	螺旋飞入、螺旋飞出	不支持
劈裂	支持方向	伸展	支持方向	曲线向上	不支持
棋盘	支持方向	升起	不支持	投掷	不支持
切入	支持方向	缩放	支持缩放	玩具风车	不支持
闪烁一次	不支持	下降	不支持	线形	不支持
扇形展开	不支持	压缩	不支持	旋转	支持方向
十字形扩展	支持方向	颜色打字机	不支持	折叠	不支持
随机线条	支持方向	展开-温和型	不支持	字幕式	不支持

版权所有：腾讯云计算（北京）有限责任公司

随机效果	不支持	中心旋转	不支持	–	–
向内溶解	不支持	弹跳	不支持	–	–

强调动画

动画名称	属性	动画名称	属性
放大/缩小	尺寸	忽明忽暗	不支持
更改填充颜色	充填色	混色	颜色
更改线条颜色	线条色	加粗闪烁	不支持
更改字号	字号	加深	不支持
更改字体	字体	添加下划线	不支持
更改字体颜色	字体颜色	着色	颜色
更改字形	字形	温和型	不支持
透明	数量	彩色延伸	颜色
陀螺旋	数量	跷跷板	不支持
细微型	不支持	闪动	颜色
变淡	不支持	闪现	不支持
补色	不支持	华丽型	不支持
补色2	不支持	爆炸	颜色
不饱和	不支持	波浪型	不支持
彩色波纹	颜色	加粗展示	不支持
垂直突出显示	颜色	闪烁	不支持
对比色	不支持	样式强调	颜色

版权所有：腾讯云计算（北京）有限责任公司

更新日志

最近更新时间：2022-06-24 17:53:38

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
服务和队列配置
开通文件处理服务

最近更新时间：2023-08-01 15:45:53

功能描述

本接口用于开通文件处理服务并生成队列。

请求

请求示例

```
POST /file_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>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
  <FileBucket>
    <BucketId>test-1234567890</BucketId>
    <Name>test-1234567890</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
  </FileBucket>
</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
FileBucket	Response	存储桶信息	Container

Container 节点 FileBucket 的内容：

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

实际案例

请求

```
POST /file_bucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
    <FileBucket>
        <BucketId>test-1234567890</BucketId>
        <Name>test-1234567890</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
    </FileBucket>
</Response>
```

查询文件处理服务

最近更新时间：2023-08-01 15:45:52

功能描述

本接口用于查询已经开通文件处理功能的存储桶。

请求

请求示例

```
GET /file_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	默认值	是否必填
regions	地域信息，例如 ap-shanghai、ap-beijing，若查询多个地域以“,”分隔字符串，详情请参见 地域与域名	无	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	无	否
bucketN	存储桶名称前缀，前缀搜索	无	否

ame			
pageNumber	第几页	1	否
pageSize	每页个数，大于0且小于等于100的整数	10	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>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <FileBucketList>
    <BucketId>examplebucket-1250000000</BucketId>
    <Name>examplebucket-1250000000</Name>
    <Region>ap-beijing</Region>
    <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
  </FileBucketList>
  <FileBucketList>
    <BucketId>examplebucket-1-1250000000</BucketId>
    <Name>examplebucket-1-1250000000</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
  </FileBucketList>
</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	开通文件处理的 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
FileBucketList	Response	文件处理 Bucket 详情	Container 数组

Container 节点 FileBucketList 的内容：

节点名称（关键字）	父节点	描述	类型
BucketId	Response.FileBucketList	存储桶 ID	String
Name	Response.FileBucketList	存储桶名称，同 BucketId	String
Region	Response.FileBucketList	所在的地域	String
CreateTime	Response.FileBucketList	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 /file_bucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-s
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 2021 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>
    <FileBucketList>
        <BucketId>examplebucket-1250000000</BucketId>
        <Name>examplebucket-1250000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </FileBucketList>
    <FileBucketList>
        <BucketId>examplebucket-1-1250000000</BucketId>
        <Name>examplebucket-1-1250000000</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
    </FileBucketList>
</Response>
```

关闭文件处理服务

最近更新时间：2023-08-01 15:45:52

功能描述

本接口用于关闭文件处理服务并删除队列。

请求

请求示例

```
DELETE /file_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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
  <BucketName>test-1234567890</BucketName>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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
BucketName	Response	存储桶名称	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 /file_bucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
    <BucketName>test-1234567890</BucketName>
```

```
</Response>
```

查询文件处理队列

最近更新时间：2023-08-01 15:45:54

功能描述

本接口用于查询文件处理队列。

请求

请求示例

```
GET /file_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	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
  <TotalCount>1</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <QueueList>
    <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
    <Name>My-Queue-file</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>FileProcess</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

版权所有：腾讯云计算（北京）有限责任公司

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

版权所有：腾讯云计算（北京）有限责任公司

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	String
Type	Response.QueueList.NotifyConfig	回调类型	String
Event	Response.QueueList.NotifyConfig	回调事件	String
ResultFormat	Response.QueueList.NotifyConfig	回调类型	String
MqMode	Response.QueueList.NotifyConfig	TDMQ 使用模式	String
MqRegion	Response.QueueList.NotifyConfig	TDMQ 所属园区	String
MqName	Response.QueueList.NotifyConfig	TDMQ 主题名称	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 /file_queue?queueIds=p4a17eeea29334bf499b7e20e2fbfd99d,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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: NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85

<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>My-Queue-file</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>FileProcess</Category>
    </QueueList>
    <NonExistPIDs>A</NonExistPIDs>
    <NonExistPIDs>B</NonExistPIDs>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新文件处理队列

最近更新时间：2023-08-01 16:46:26

功能描述

本接口用于更新文件处理的队列。

请求

请求示例

```
PUT /file_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-file</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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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	默认值	是否必选
Name	Request	队列名称，仅支持中文、英文、数字、_、-和*，长度不超过 128	String	无	是
State	Request	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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	无	是
NotifyConfig	Request	回调配置	Container	无	是

Container 节点 NotifyConfig 的内容：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
State	Request.NotifyConfig	回调开关 • Off • On	String	Off	否
Event	Request.NotifyConfig	回调事件 • TaskFinish：任务完成 • WorkflowFinish：工作流完成	String	无	当 State=On 时，必选
ResultFormat	Request.NotifyConfig	回调格式 • XML • JSON	String	XML	否

Type	Request.NotifyConfig	回调类型 • Url • TDMQ	String	无	当 State=On 时，必选
Url	Request.NotifyConfig	回调地址，不能为内网地址。	String	无	当 State=On，且 Type=Url 时，必选
MqMode	Request.NotifyConfig	TDMQ 使用模式 • Topic：主题订阅 • Queue: 队列服务	String	无	当 State=On，且 Type=TDMQ 时，必选
MqRegion	Request.NotifyConfig	TDMQ 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	无	当 State=On，且 Type=TDMQ 时，必选
MqName	Request.NotifyConfig	TDMQ 主题名称	String	无	当 State=On，且 Type=TDMQ 时，必选

响应

响应头

此接口仅返回公共响应头部，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-file</Name>
    <State>Active</State>
    <NotifyConfig>
      <Url>http://callback.demo.com</Url>
```

```
        <Event>TaskFinish,WorkflowFinish</Event>
        <Type>Url</Type>
        <State>On</State>
        <ResultFormat>JSON</ResultFormat>
        <MqMode/>
        <MqName/>
        <MqRegion/>
      </NotifyConfig>
      <MaxSize>10000</MaxSize>
      <MaxConcurrent>10</MaxConcurrent>
      <Category>FileProcess</Category>
      <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 /file_queue/p4a17eeea29334bf499b7e20e2fbfd99d HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 162
Content-Type: application/xml

<Request>
    <Name>My-Queue-file</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-file</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
            <ResultFormat>JSON</ResultFormat>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <Category>FileProcess</Category>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

文件哈希值计算
哈希值计算同步请求

最近更新时间：2023-07-14 14:47:02

功能描述

以同步请求的方式进行文件哈希值计算，实时返回计算得到的哈希值，该接口属于 GET 请求。

费用说明

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

限制说明

- 文件大小：支持计算小于 128 MB 大小的文件，更大文件请使用 哈希值计算异步接口 。
- 请求超时时间：10秒。
- 调用接口需携带签名，具体规则请参见 请求签名 文档。
- 本接口当前支持以下地域：北京、上海、广州、成都、南京、重庆、中国香港、新加坡、硅谷。

请求

请求示例

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

版权所有：腾讯云计算（北京）有限责任公司

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
LastModified	Response.FileHashCodeResult	对象的最近一次修改的时间	String
Etag	Response.FileHashCodeResult	ETag 全称为 Entity Tag，是对象被创建时标识对象内容的信息标签，可用于检查对象的内容是否发生变化	String

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

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

```
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*
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：2023-08-01 16:46:26

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 /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。

版权所有：腾讯云计算（北京）有限责任公司

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>
  <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
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

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，支持：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>
    <Progress>0</Progress>
    <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 数组

版权所有：腾讯云计算（北京）有限责任公司

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	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	任务状态 • Submitted：已提交，待执行。 • Running：执行中。 • Success：执行成功。 • Failed：执行失败。 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态。 • Cancel：任务被取消执行。	String
Progress	Response.JobsDetail	任务进度百分比，范围为[0, 100]。	Int
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。	Container
FileHashCodeResult	Response.JobsDetail.Operation	计算得到的文件 hash 值信息，任务未完成时不返回。	Container

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.FileHa	文件大小。	Int

版权所有：腾讯云计算（北京）有限责任公司

	shCodeResult		
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 /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-s
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>
    <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>
    <Progress>0</Progress>
    <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
    <EndTime>-</EndTime>
    <StartTime>-</StartTime>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <Tag>FileHashCode</Tag>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>input/deer.jpg</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <FileHashCodeConfig>
        <Type>MD5</Type>
        <AddToHeader>true</AddToHeader>
      </FileHashCodeConfig>
      <UserData>This is my data.</UserData>
    </Operation>
  </JobsDetail>
</Response>
```

查询哈希值计算结果

最近更新时间：2023-08-11 17:56:17

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/>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
      <State>Submitted</State>
      <Progress>0</Progress>
      <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	同提交哈希值计算接口中的 Response.JobsDetail	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 数组

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

```
GET /file_jobs/f93984788066911ed89ed352d4d9d2084 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-s
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>
    <Progress>0</Progress>
    <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
    <EndTime>-</EndTime>
    <StartTime>-</StartTime>
    <QueueId>p2911917386e148639319e13c285cc774</QueueId>
    <Tag>FileHashCode</Tag>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>input/test.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <FileHashCodeConfig>
        <Type>MD5</Type>
```

```
            <AddToHeader>true</AddToHeader>
          </FileHashCodeConfig>
          <FileHashCodeResult>
            <MD5>xxxxxxxx</MD5>
            <FileSize>123</FileSize>
            <LastModified>2022-06-27T15:23:11+0800</LastModified>
            <Etag>xxxxxx</Etag>
          </FileHashCodeResult>
          <UserData>This is my data.</UserData>
        </Operation>
    </JobsDetail>
</Response>
```

文件解压
提交文件解压任务

最近更新时间：2023-08-11 17:56:17

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5TB 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、南京、重庆、中国香港、新加坡、硅谷。

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>FileUncompress</Tag>
    <Input>
        <Object>input/test.zip</Object>
    </Input>
    <Operation>
        <FileUncompressConfig>
            <Prefix>output/</Prefix>
            <PrefixReplaced>1</PrefixReplaced>
            <UnCompressKey>MTIzNDU2Nzg5MA==</UnCompressKey>
        </FileUncompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <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

	t			
Operation	Request	包含文件解压的处理规则。	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式。	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型。	String	否
CallBack	Request	任务回调的地址，优先级高于队列的回调地址。	String	否
CallBackMqConfig	Request	任务回调 TDMQ 配置，当 CallBackType 为 TDMQ 时必填。详情见 CallBackMqConfig 。	Container	否

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	父节点	描述	类	是否

键字）			型	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
UnCompressKey	Request.Operation.FileUnCompressConfig	解压密钥，传入时需先经过 base64 编码。	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 • 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。	String	否

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.O	保存解压后文件的存储	Stri	是

	utput	桶。	ng	

响应

响应头

此接口仅返回公共响应头部，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>
    <Progress>0</Progress>
    <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>
        <UnCompressKey>MTIzNDU2Nzg5MA==</UnCompressKey>
      </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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	String
Progress	Response.JobsDetail	任务进度百分比，范围为[0, 100]。	Int
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	类型
UserData	Response.JobsDetail.Operation	透传用户信息。	String
Output	Response.JobsDetail.Operation	同请求中的 Request.Operation.Output。	Container
FileUncompressConfig	Response.JobsDetail.Operation	同请求中的 Request.Operation.FileUncompressConfig。	Container
FileUncompressResult	Response.JobsDetail.Operation	文件解压的结果，任务未完成时不返回。	Container

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 /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-s
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>
            <UnCompressKey>MTIzNDU2Nzg5MA==</UnCompressKey>
        </FileUncompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <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>
    <Progress>0</Progress>
    <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>
        <UnCompressKey>MTIzNDU2Nzg5MA==</UnCompressKey>
      </FileUncompressConfig>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-1234567890</Bucket>
      </Output>
      <UserData>This is my data.</UserData>
    </Operation>
  </JobsDetail>
</Response>
```

查询文件解压结果

最近更新时间：2023-08-18 15:28:02

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/>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
        <State>Submitted</State>
        <Progress>0</Progress>
        <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	同提交文件解压任务接口中的 Response.JobsDetail	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 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

```
GET /file_jobs/f93984788066911ed89ed352d4d9d2084 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
        <Progress>0</Progress>
        <CreationTime>2022-07-18T15:16:43+0800</CreationTime>
        <EndTime>-</EndTime>
        <StartTime>-</StartTime>
        <QueueId>p2911917386e148639319e13c285cc774</QueueId>
        <Tag>FileUncompress</Tag>
        <Input>
            <BucketId>test-1234567890</BucketId>
            <Object>input/test.zip</Object>
            <Region>ap-chongqing</Region>
        </Input>
        <Operation>
            <FileUncompressConfig>
                <Prefix>output/</Prefix>
                <PrefixReplaced>1</PrefixReplaced>
            </FileUncompressConfig>
```

```
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-1234567890</Bucket>
        </Output>
        <UserData>This is my data.</UserData>
      </Operation>
    </JobsDetail>
</Response>
```

多文件打包压缩
提交多文件打包压缩任务

最近更新时间：2023-08-01 16:46:26

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、南京、重庆、中国香港、新加坡、硅谷。

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。

版权所有：腾讯云计算（北京）有限责任公司

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求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>
      <Type>better</Type>
      <CompressKey>MTIzNDU2Nzg5MA==</CompressKey>
    </FileCompressConfig>
    <Output>
      <Region>ap-chongqing</Region>
      <Bucket>test-1234567890</Bucket>
      <Object>output/test.zip</Object>
    </Output>
    <UserData>This is my data.</UserData>
  </Operation>
  <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

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

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.Fi	打包压缩的类型，有效值：zip、tar、tar.gz。	String	是

	leCompressConfig			
Type	Request.Operation.FileCompressConfig	压缩类型，仅在Format为tar.gz或zip时有效。 • faster：压缩速度较快 • better：压缩质量较高，体积较小 • default：适中的压缩方式 • 默认值为default	String	否
CompressKey	Request.Operation.FileCompressConfig	压缩包密钥，传入时需先经过 base64 编码，编码后长度不能超过128。当 Format 为 zip 时生效。	String	否
UrlList	Request.Operation.FileCompressConfig	• 支持将需要打包的文件整理成索引文件，后台将根据索引文件内提供的文件 url，打包为一个压缩包文件。 • 索引文件需要保存在当前存储桶中，本字段需要提供索引文件的对象地址，例如：/test/index.csv。 • 索引文件格式：仅支持 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。	String	否
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>
    <Progress>0</Progress>
    <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>
        <Type>better</Type>
        <CompressKey>MTIzNDU2Nzg5MA==</CompressKey>
      </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 数组

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	任务状态 • Submitted：已提交，待执行。 • Running：执行中。 • Success：执行成功。 • Failed：执行失败。 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态。 • Cancel：任务被取消执行。	String
Progress	Response.J	任务进度百分比，范围为[0, 100]。	Int

	obsDetail		
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。	Container
FileCompressConfig	Response.JobsDetail.Operation	同请求中的 Request.Operation.FileCompressConfig。	Container
FileCompressResult	Response.JobsDetail.Operation	多文件打包压缩的结果，任务未完成时不返回。	Container

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.Fi	打包压缩后文件的名称。	Str

版权所有：腾讯云计算（北京）有限责任公司

	leCompressResult		ing

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
            <Type>better</Type>
            <CompressKey>MTIzNDU2Nzg5MA==</CompressKey>
        </FileCompressConfig>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-1234567890</Bucket>
            <Object>output/test.zip</Object>
        </Output>
        <UserData>This is my data.</UserData>
    </Operation>
    <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
```

版权所有：腾讯云计算（北京）有限责任公司

```
Server: tencent-ci
x-ci-request-id: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<Response>
  <JobsDetail>
    <Code>Success</Code>
    <Message/>
    <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
    <State>Submitted</State>
    <Progress>0</Progress>
    <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>
        <Type>better</Type>
        <CompressKey>MTIzNDU2Nzg5MA==</CompressKey>
      </FileCompressConfig>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-1234567890</Bucket>
        <Object>output/test.zip</Object>
      </Output>
      <UserData>This is my data.</UserData>
    </Operation>
  </JobsDetail>
</Response>
```

查询多文件打包压缩结果

最近更新时间：2023-08-11 17:56:17

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/>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <JobId>f93984788066911ed89ed352d4d9d2084</JobId>
      <State>Submitted</State>
      <Progress>0</Progress>
      <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	同提交多文件打包任务接口中的 Response.JobsDetail	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 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

```
GET /file_jobs/f93984788066911ed89ed352d4d9d2084 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
        <Progress>0</Progress>
        <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>
```

版权所有：腾讯云计算（北京）有限责任公司

压缩包预览

最近更新时间：2023-08-01 16:46:26

功能描述

该接口可以在不解压文件的情况下预览压缩包内的内容，包含文件数量、名称、文件时间等，接口为同步请求方式。

费用说明

- 调用接口会产生 文件解压费用 。
- 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

- 支持预览的压缩包格式：zip、tar、gz、7zip、rar。
- 预览压缩包大小限制：zip/7zip无大小限制，tar/gz/rar需小于128MB。
- 预览压缩包内的文件数限制：仅支持预览内部文件数量小于1000的压缩包。
- 请求超时时间：10秒。
- 调用接口需携带签名，具体规则请参见 请求签名 文档。
- 本接口当前支持以下地域：北京、上海、广州、成都、南京、重庆、中国香港、新加坡、硅谷。

请求

请求示例

```
GET /for-test.zip?ci-process=zippreview 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。
- 该请求为 cos 域名。

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。

请求参数

参数说明如下：

节点名称（关键字）	描述	是否必选
ci-process	操作类型，压缩包预览计算固定为：zippreview	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>
  <FileNumber>2</FileNumber>
  <Contents>
    <Key>test1.jpg</Key>
    <LastModified>2021-11-21T12:30:42+0000</LastModified>
    <UncompressedSize>46743</UncompressedSize>
  </Contents>
  <Contents>
    <Key>test2.mp3</Key>
    <LastModified>2021-11-22T12:11:26+0000</LastModified>
    <UncompressedSize>733318</UncompressedSize>
  </Contents>
</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
FileNumber	Response	压缩包中的文件数量	Int
Contents	Response	保存文件信息的容器	Container 数组

Container 节点 Contents 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.Contents	文件名称	String
LastModified	Response.Contents	文件的最近一次修改的时间	String
UncompressedSize	Response.Contents	文件压缩前的大小	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 /for-test.zip?ci-process=zippreview HTTP/1.1
Host: test-1234567890.cos.ap-beijing.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfG*
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>
    <FileNumber>2</FileNumber>
    <Contents>
        <Key>test1.jpg</Key>
        <LastModified>2021-11-21T12:30:42+0000</LastModified>
        <UncompressedSize>46743</UncompressedSize>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </Contents>
    <Contents>
        <Key>test2.mp3</Key>
        <LastModified>2021-11-22T12:11:26+0000</LastModified>
        <UncompressedSize>733318</UncompressedSize>
    </Contents>
</Response>
```

智能语音
服务和队列配置
开通智能语音服务

最近更新时间：2023-08-01 16:46:28

功能描述

本接口用于开通智能语音服务并生成队列。

请求

请求示例

```
POST /asrbucket 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>
  <AsrBucket>
    <BucketId>test-1234567890</BucketId>
    <Name>test-1234567890</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
  </AsrBucket>
</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
AsrBucket	Response	存储桶信息	Container

Container 节点 AsrBucket 的内容：

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

实际案例

请求

```
POST /asrbucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
    <AsrBucket>
        <BucketId>test-1234567890</BucketId>
        <Name>test-1234567890</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2022-08-09T16:23:11+0800></CreateTime>
    </AsrBucket>
</Response>
```

查询智能语音服务

最近更新时间：2023-08-01 16:46:26

功能描述

本接口用于查询已经开通智能语音功能的存储桶。

请求

请求示例

```
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	描述	默认值	是否必填
regions	地域信息，例如 ap-shanghai、ap-beijing，若查询多个地域以“,”分隔字符串，详情请参见 地域与域名	无	否
bucketNames	存储桶名称，以“,”分隔，支持多个存储桶，精确搜索	无	否
bucketNam	存储桶名称前缀，前缀搜索	无	否

e			
pageNumber	第几页	1	否
pageSize	每页个数，大于0且小于等于100的整数	10	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>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <AsrBucketList>
    <BucketId>examplebucket-1250000000</BucketId>
    <Name>examplebucket-1250000000</Name>
    <Region>ap-beijing</Region>
    <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
  </AsrBucketList>
  <AsrBucketList>
    <BucketId>examplebucket-1-1250000000</BucketId>
    <Name>examplebucket-1-1250000000</Name>
    <Region>ap-chongqing</Region>
    <CreateTime>2020-11-16T14:13:15+0800</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	开通智能语音的 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	智能语音 Bucket 详情	Container 数组

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 /asrbucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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 2021 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>
    <AsrBucketList>
        <BucketId>examplebucket-1250000000</BucketId>
        <Name>examplebucket-1250000000</Name>
        <Region>ap-beijing</Region>
        <CreateTime>2019-11-15T12:12:12+0800</CreateTime>
    </AsrBucketList>
    <AsrBucketList>
        <BucketId>examplebucket-1-1250000000</BucketId>
        <Name>examplebucket-1-1250000000</Name>
        <Region>ap-chongqing</Region>
        <CreateTime>2020-11-16T14:13:15+0800</CreateTime>
    </AsrBucketList>
</Response>
```

关闭智能语音服务

最近更新时间：2023-08-01 16:46:28

功能描述

本接口用于关闭智能语音服务并删除队列。

请求

请求示例

```
DELETE /asrbucket 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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
  <BucketName>test-1234567890</BucketName>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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
BucketName	Response	存储桶名称	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 /asrbucket HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>NjJmMzYxMDlfOTBmYTUwNjRfNGE2YV83</RequestId>
    <BucketName>test-1234567890</BucketName>
```

```
</Response>
```

查询智能语音队列

最近更新时间：2023-08-08 14:20:52

功能描述

本接口用于查询智能语音队列。

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

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	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
  <TotalCount>1</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <QueueList>
    <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
    <Name>My-Queue-asr</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>Speeching</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

版权所有：腾讯云计算（北京）有限责任公司

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	String
Type	Response.QueueList.NotifyConfig	回调类型	String
Event	Response.QueueList.NotifyConfig	回调事件	String
ResultFormat	Response.QueueList.NotifyConfig	回调类型	String
MqMode	Response.QueueList.NotifyConfig	TDMQ 使用模式	String
MqRegion	Response.QueueList.NotifyConfig	TDMQ 所属园区	String
MqName	Response.QueueList.NotifyConfig	TDMQ 主题名称	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 /asrqueue?queueIds=p4a17eeea29334bf499b7e20e2fbfd99d,A,B HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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: NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85

<Response>
    <RequestId>NjJmNDczY2RfOTBmYTUwNjRfNTA4ZV85</RequestId>
    <QueueList>
        <QueueId>p4a17eeea29334bf499b7e20e2fbfd99d</QueueId>
        <Name>My-Queue-asr</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>Speeching</Category>
    </QueueList>
    <NonExistPIDs>A</NonExistPIDs>
    <NonExistPIDs>B</NonExistPIDs>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新智能语音队列

最近更新时间：2023-08-01 16:46:26

功能描述

本接口用于更新智能语音的队列。

请求

请求示例

```
PUT /asrqueue/<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-asr</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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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	默认值	是否必选
Name	Request	队列名称，仅支持中文、英文、数字、_、-和*，长度不超过128	String	无	是
State	Request	• Active 表示队列内的作业会被调度执行 • Paused 表示队列暂停，作业不再会被调度执行，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	无	是
NotifyConfig	Request	回调配置	Container	无	是

Container 节点 NotifyConfig 的内容：

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
State	Request.NotifyConfig	回调开关 • Off • On	String	Off	否
Event	Request.NotifyConfig	回调事件 • TaskFinish：任务完成 • WorkflowFinish：工作流完成	String	无	当 State=On 时，必选
ResultFormat	Request.NotifyConfig	回调格式 • XML • JSON	String	XML	否

Type	Request.NotifyConfig	回调类型 • Url • TDMQ	String	无	当 State=On 时，必选
Url	Request.NotifyConfig	回调地址，不能为内网地址。	String	无	当 State=On，且 Type=Url 时，必选
MqMode	Request.NotifyConfig	TDMQ 使用模式 • Topic：主题订阅 • Queue: 队列服务	String	无	当 State=On，且 Type=TDMQ 时，必选
MqRegion	Request.NotifyConfig	TDMQ 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	无	当 State=On，且 Type=TDMQ 时，必选
MqName	Request.NotifyConfig	TDMQ 主题名称	String	无	当 State=On，且 Type=TDMQ 时，必选

响应

响应头

此接口仅返回公共响应头部，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-asr</Name>
    <State>Active</State>
    <NotifyConfig>
      <Url>http://callback.demo.com</Url>
```

```
        <Event>TaskFinish,WorkflowFinish</Event>
        <Type>Url</Type>
        <State>On</State>
        <ResultFormat>JSON</ResultFormat>
        <MqMode/>
        <MqName/>
        <MqRegion/>
      </NotifyConfig>
      <MaxSize>10000</MaxSize>
      <MaxConcurrent>10</MaxConcurrent>
      <Category>Speeching</Category>
      <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 /asrqueue/p4a17eeea29334bf499b7e20e2fbfd99d HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 162
Content-Type: application/xml

<Request>
    <Name>My-Queue-asr</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-asr</Name>
        <State>Active</State>
        <NotifyConfig>
            <Url>http://callback.demo.com</Url>
            <Event>TaskFinish,WorkflowFinish</Event>
            <Type>Url</Type>
            <State>On</State>
            <ResultFormat>JSON</ResultFormat>
            <MqMode/>
            <MqName/>
            <MqRegion/>
        </NotifyConfig>
        <MaxSize>10000</MaxSize>
        <MaxConcurrent>10</MaxConcurrent>
        <Category>Speeching</Category>
        <UpdateTime>2022-08-09T16:23:11+0800</UpdateTime>
        <CreateTime>2022-08-09T16:13:05+0800</CreateTime>
    </Queue>
</Response>
```

人声分离
提交任务

最近更新时间：2023-08-08 17:34:11

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.${ext}</Object>
        <AuObject>output/audio.${ext}</AuObject>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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	人声分离模板 ID	String	否
VoiceSeparate	Request.Operation	人声分离模板参数	Container	否
Output	Request.Operation	结果输出配置	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

⚠ 注意

人声分离参数必须设置，可以通过 TemplateId 或 VoiceSeparate 设置， TemplateId 优先级更高。

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接口中的 Request.AudioMode	String	是
AudioConfig	Request.Operation.VoiceSeparate	同创建人声分离模板接口中的 Request.AudioConfig	Container	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.Operatio	存储结果的存储桶	Str	是

	n.Output		ing	
Object	Request.Operation.Output	背景音结果文件名	String	否
AuObject	Request.Operation.Output	人声结果文件名	String	否
BassObject	Request.Operation.Output	Bass声结果文件名	String	否
DrumObject	Request.Operation.Output	鼓声结果文件名	String	否

> **注意：**
> Object、AuObject、BassObject、DrumObject 不可以同时为空

Request.Operation.Output.Object、Request.Operation.Output.AuObject、Request.Operation.Output.AuObject、Request.Operation.Output.AuObject 支持的通配符如下：

通配符	含义
${ext}	封装格式

响应

响应头

此接口仅返回公共响应头部，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.${ext}</Object>
        <AuObject>output/audio.${ext}</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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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1：使用人声分离模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
Host:test-1234567890.ci.ap-chongqing.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.${ext}</Object>
        <AuObject>output/audio.${ext}</AuObject>
      </Output>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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.${ext}</Object>
```

```
            <AuObject>output/audio.${ext}</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-s
Host: test-1234567890.ci.ap-chongqing.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.${ext}</Object>
            <AuObject>output/audio.${ext}</AuObject>
        </Output>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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.${ext}</Object>
        <AuObject>output/audio.${ext}</AuObject>
      </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

任务回调

最近更新时间：2023-08-18 15:28:02

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 application/xml 数据，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-07-01T10:12:26+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>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <ObjectPrefix/>
            <Region>ap-chongqing</Region>
          </OutputFile>
        </MediaResult>
        <Output>
          <AuObject>output/audio.${ext}</AuObject>
          <Bucket>test-123456789</Bucket>
          <Object>output/backgroud.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <UserData>This is my VoiceSeparate job.</UserData>
        <JobLevel>0</JobLevel>
        <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	同提交人声分离任务接口中的 Response.JobsDetail	Container 数组

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.CosH	自定义 Header 的名	Strin

	eaders	称	g
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.${ext}</AuObject>
          <Bucket>test-123456789</Bucket>
          <Object>output/backgroud.${ext}</Object>
          <Region>ap-chongqing</Region>
        </Output>
        <UserData>This is my VoiceSeparate job.</UserData>
        <JobLevel>0</JobLevel>
        <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.${ext}</AuObject>
      <Bucket>test-123456789</Bucket>
      <Object>output/backgroud.${ext}</Object>
      <Region>ap-chongqing</Region>
    </Output>
    <UserData>This is my VoiceSeparate job.</UserData>
    <JobLevel>0</JobLevel>
```

```
        <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.${ext}",
    "Bucket": "test-123456789",
    "Object": "output/backgroud.${ext}",
    "Region": "ap-chongqing"
},
"UserData": "This is my VoiceSeparate job.",
"JobLevel": 0,
"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"
```

```
      }
    }]
  }
```

创建模板

最近更新时间：2023-08-08 17:34:11

功能描述

创建人声分离模板。

请求

请求示例

```
POST /template HTTP/1.1
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
Content-Length: <length>
Content-Type: application/xml

<body>
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求操作的实现需要有如下请求体。

```
<Request>
  <Tag>VoiceSeparate</Tag>
  <Name>TemplateName</Name>
  <AudioMode>IsAudio</AudioMode>
  <AudioConfig>
    <Codec>aac</Codec>
    <Samplerate>44100</Samplerate>
    <Bitrate>128</Bitrate>
    <Channels>4</Channels>
  </AudioConfig>
```

```
</Request>
```

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Tag	Request	模板类型: VoiceSeparate	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64	String	是
AudioMode	Request	输出音频 • IsAudio：输出人声 • IsBackground：输出背景声 • AudioAndBackground：输出人声和背景声 • MusicMode：输出人声、背景声、Bass声、鼓声	String	是
AudioConfig	Request	音频配置	Container	是

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	默认值	是否必选
Codec	Request.AudioConfig	编解码格式，取值 aac、mp3、flac、amr。当 Request.AudioMode 为 MusicMode 时，仅支持 mp3、wav、acc	String	aac	否
Samplerate	Request.AudioConfig	采样率 • 单位：Hz	String	44100	否

		• 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 • 当 Request.AudioMode 为 MusicMode 时，该参数无效			
Bitrate	Request.AudioConfig	音频码率 • 单位：Kbps • 值范围：[8，1000] • 当 Request.AudioMode 为 MusicMode 时，该参数无效	String	原始音频码率	否
Channels	Request.AudioConfig	声道数 • 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 • 当 Request.AudioMode 为 MusicMode 时，该参数无效	String	原始音频声道数	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>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <Template>
    <Tag>VoiceSeparate</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
```

版权所有：腾讯云计算（北京）有限责任公司

```
      <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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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

版权所有：腾讯云计算（北京）有限责任公司

Tag	Response.Template	模板类型，VoiceSeparate	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
VoiceSeparate	Response.Template	详细的模板参数	Container

Container 节点 VoiceSeparate 的内容：

节点名称（关键字）	父节点	描述	类型
AudioMode	Response.Template.VoiceSeparate	同请求体中的 Request.AudioMode	String
AudioConfig	Response.Template.VoiceSeparate	同请求体中的 Request.AudioConfig	Container

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-
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: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x

<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <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>
```

更新模板

最近更新时间：2023-08-08 17:34:11

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
</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>
  <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <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-
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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <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>
```

音频降噪
提交任务

最近更新时间：2023-08-11 17:56:17

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>
    <NoiseReduction>
      <Format>wav</Format>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <SampleRate>16000</SampleRate>
      </NoiseReduction>
      <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>
    <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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	执行音频降噪任务的文件路径 • 目前只支持文件大小在10M之内的音频 • 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 • 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
TemplateId	Request.Operation	降噪模板ID	String	否
NoiseReduction	Request.Operation	降噪任务参数，同创建降噪模板接口中的 Request.NoiseReduction	Container	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
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	结果输出配置	Container	是

说明

降噪参数必须设置，可以通过 TemplateId 或 NoiseReduction 设置， TemplateId 优先级更高。

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.Out	存储结果的存储桶	Strin	是

	put		g	
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>NoiseReduction</Tag>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp4</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <Operation>
      <NoiseReduction>
        <Format>wav</Format>
        <SampleRate>16000</SampleRate>
      </NoiseReduction>
      <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>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	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

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：NoiseReduction	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
TemplateId	Response.JobsDetail.Operation	降噪模板ID	String
TemplateName	Response.JobsDetail.Operation	任务的模板名称，当 TemplateId 存在时返回	String
NoiseReuction	Response.JobsDetail.Operation	同请求中的 Request.Operation.NoiseReduction	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

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，使用降噪模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>NoiseReduction</Tag>
    <Input>
        <Object>input/demo.mp3</Object>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd<TemplateId>
        <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>
    <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: 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>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd<TemplateId>
        <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>
```

请求2，使用降噪参数

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0**********&q-si
Host:test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>NoiseReduction</Tag>
    <Input>
        <Object>input/demo.mp3</Object>
    </Input>
    <Operation>
        <NoiseReduction>
            <Format>wav</Format>
            <SampleRate>16000</SampleRate>
        </NoiseReduction>
        <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>
    <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: 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>
        <NoiseReduction>
          <Format>wav</Format>
          <SampleRate>16000</SampleRate>
        </NoiseReduction>
        <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>
```

任务回调

最近更新时间：2023-08-14 17:01:45

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-06-30T18:47:18+0800</CreationTime>
    <EndTime>2022-06-30T18:47:30+0800</EndTime>
    <Input>
      <BucketId>test-123456789</BucketId>
      <Object>input/demo.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>j035aca66f86211ec8a5a87e016101404</JobId>
    <Message>SUCCESS</Message>
    <Operation>
      <Output>
        <Bucket>test-123456789</Bucket>
        <Object>output/nr.wav</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <UserData>This is my NoiseReduction job.</UserData>
      <JobLevel>0</JobLevel>
    </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	同提交音频降噪任务接口中的 Response.JobsDetail	Container 数组

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>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <StartTime>2022-06-30T18:47:20+0800</StartTime>
    <State>Success</State>
    <Tag>NoiseReduction</Tag>
  </JobsDetail>
</Response>
```

案例 2：通过任务接口触发的任务回调，格式为 JSON

```
{
  "EventName": "TaskFinish",
  "JobsDetail": [{
    "Code": "Success",
    "CreationTime": "2022-06-30T18:47:18+0800",
    "EndTime": "2022-06-30T18:47:30+0800",
    "Input": {
      "BucketId": "test-123456789",
      "Object": "input/demo.mp3",
      "Region": "ap-chongqing"
    },
    "JobId": "j035aca66f86211ec8a5a87e016101404",
    "Message": "SUCCESS",
    "Operation": {
      "Output": {
```

```
            "Bucket": "test-123456789",
            "Object": "output/nr.wav",
            "Region": "ap-chongqing"
        },
        "UserData": "This is my NoiseReduction job.",
        "JobLevel": "0"
    },
    "QueueId": "p2242ab62c7c94486915508540933a2c6",
    "StartTime": "2022-06-30T18:47:20+0800",
    "State": "Success",
    "Tag": "NoiseReduction"
  }]
}
```

创建模板

最近更新时间：2023-08-01 16:46:27

功能描述

创建音频降噪模板。

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>NoiseReduction</Tag>
   <Name>TemplateName</Name>
   <NoiseReduction>
      <Format>wav</Format>
      <Samplerate>16000</Samplerate>
   </NoiseReduction>
</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	NoiseReduction	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64。	String	是
NoiseReduction	Request	降噪参数	Container	是

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

节点名称（关键字）	父节点	描述	类型	默认值	是否必选
Format	Request.NoiseReduction	封装格式，支持 mp3、m4a、wav	String	wav	否
Samplerate	Request.NoiseReduction	采样率 • 单位：Hz • 可选 8000、12000、16000、24000、32000、44100、48000	String	16000	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>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <Template>
```

```
        <Tag>NoiseReduction</Tag>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>NoiseReduction</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <NoiseReduction>
          <Format>wav</Format>
          <Samplerate>16000</Samplerate>
        </NoiseReduction>
    </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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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

版权所有：腾讯云计算（北京）有限责任公司

Tag	Response.Template	模板类型，NoiseReduction	String
UpdateTime	Response.Template	更新时间	String
CreateTime	Response.Template	创建时间	String
NoiseReduction	Response.Template	同请求体中的 Request.NoiseReduction	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
Host: test-1234567890.ci.ap-chongqing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>NoiseReduction</Tag>
    <Name>TemplateName</Name>
    <NoiseReduction>
        <SampleRate>16000</SampleRate>
        <Format>wav</Format>
    </NoiseReduction>
</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****
```

版权所有：腾讯云计算（北京）有限责任公司

```
<Response>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>NoiseReduction</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <NoiseReduction>
            <SampleRate>16000</SampleRate>
            <Format>wav</Format>
        </NoiseReduction>
    </Template>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

更新模板

最近更新时间：2023-08-14 17:01:45

功能描述

更新音频降噪模板。

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>NoiseReduction</Tag>
   <Name>TemplateName</Name>
   <NoiseReduction>
      <Format>wav</Format>
      <Samplerate>16000</Samplerate>
   </NoiseReduction>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	同创建音频降噪模板接口的 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>
  <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <Template>
    <Tag>NoiseReduction</Tag>
    <Name>TemplateName</Name>
    <State>Normal</State>
    <Tag>NoiseReduction</Tag>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
    <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
    <BucketId>test-1234567890</BucketId>
    <Category>Custom</Category>
    <NoiseReduction>
      <Format>wav</Format>
      <Samplerate>16000</Samplerate>
    </NoiseReduction>
  </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	同创建音频降噪模板接口的 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-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 1666
Content-Type: application/xml

<Request>
    <Tag>NoiseReduction</Tag>
    <Name>TemplateName</Name>
    <NoiseReduction>
        <Format>wav</Format>
        <Samplerate>16000</Samplerate>
    </NoiseReduction>
</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>
    <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
    <Template>
        <Tag>NoiseReduction</Tag>
        <Name>TemplateName</Name>
        <State>Normal</State>
        <Tag>NoiseReduction</Tag>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
        <BucketId>test-1234567890</BucketId>
        <Category>Custom</Category>
        <NoiseReduction>
            <Format>wav</Format>
            <Samplerate>16000</Samplerate>
        </NoiseReduction>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </Template>
</Response>
```

语音合成
提交任务

最近更新时间：2023-08-11 18:10:33

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>
  <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
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

TemplateId	Request.Operation	语音合成模板 ID	String	否
TtsTpl	Request.Operation	语音合成参数	Container	否
TtsConfig	Request.Operation	语音合成任务参数	Container	是
Output	Request.Operation	结果输出配置	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

语音合成参数必须设置，可以通过 TemplateId 或 TtsTpl设置， TemplateId 优先级更高。

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	同创建语音合成模板接口中的 Request.Mode	String	否
Codec	Request.Operation.TtsTpl	同创建语音合成模板接口中的 Request.Codec	String	否
VoiceType	Request.Operation.TtsTpl	同创建语音合成模板接口中的 Request.VoiceType	String	否
Volume	Request.Operation.TtsTpl	同创建语音合成模板接口中的 Request.Volume	String	否

Speed	Request.Operation.TtsTpl	同创建语音合成模板接口中的 Request.Speed	String	否
Emotion	Request.Operation.TtsTpl	同创建语音合成模板接口中的 Request.Emotion	String	否

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	• 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；如果合成方式为异步处理，则文件内容不超过 10000 个 utf-8 字符。 • 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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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	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，详见 MediaInfo	Container
MediaResult	Response.JobsDetail.Operation	输出文件的基本信息，任务未完成时不返回，详见 MediaResult	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1：使用语音合成模板 ID

```
POST /jobs HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&q-si
Host:test-1234567890.ci.ap-chongqing.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>
  <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-s
Host: test-1234567890.ci.ap-chongqing.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>
    <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>
```

版权所有：腾讯云计算（北京）有限责任公司

任务回调

最近更新时间：2023-08-14 17:01:45

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </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>
        <UserData>This is my Tts job.</UserData>
        <JobLevel>0</JobLevel>
      </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	同提交语音合成任务接口中的 Response.JobsDetail	Container 数组

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.CosH	自定义 Header 的名	Strin

	eaders	称	g
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>
        <UserData>This is my tts job.</UserData>
        <JobLevel>0</JobLevel>
      </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/text.txt
```

```
            <InputType>Url</InputType>
        </TtsConfig>
        <TemplateId>t1f1ae1dfsdc9ds41dsb31632d457ds6s2a</TemplateId>
        <TemplateName>template_tts</TemplateName>
        <UserData>This is my Tts job.</UserData>
        <JobLevel>0</JobLevel>
      </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/input/text.tx
        "InputType": "Url"
    },
    "TemplateId": "t1f1ae1dfsdc9ds41dsb31632d457ds6s2a",
    "TemplateName": "template_tts",
    "UserData": "This is my Tts job.",
    "JobLevel": 0
},
"QueueId": "p2242ab62c7c94486915508540933a2c6",
"StartTime": "2022-06-30T19:33:19+0800",
"State": "Success",
"Tag": "Tts",
"Workflow": {
    "Name": "Tts_1581665960537",
    "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
```


版权所有：腾讯云计算（北京）有限责任公司

```
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
      }
    }]
}
```

创建模板

最近更新时间：2023-08-01 16:46:27

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>aixiaoxing</VoiceType>
   <Volume>2</Volume>
   <Speed>200</Speed>
   <Emotion>arousal</Emotion>
</Request>
```

版权所有：腾讯云计算（北京）有限责任公司

具体数据描述如下：

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	默认值	是否必选
Tag	Request	模板类型：Tts	String	无	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64	String	无	是
Mode	Request	处理模式 • Asyc（异步合成） • Sync（同步合成）	String	Asyc	否
Codec	Request	音频格式，支持 wav、mp3、pcm	String	• 当 Mode=Sync 时，默认值为 wav • 当 Mode=Asyc 时，默认值为 pcm	否
VoiceType	Request	音色，取值和限制介绍请见下表	String	ruxue	否
Volume	Request	音量，取值范围 [-10,10]	String	0	否
Speed	Request	语速，取值范围 [50,200]	String	100	否
Emotion	Request	情绪，不同音色支持的情绪不同，详见下表	String	无	否

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
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

版权所有：腾讯云计算（北京）有限责任公司

音色支持情绪说明

名称	Voice 参数值	情绪分类
爱小星	aixiaoxing	neutral(中性)、arousal(兴奋)
爱小诚	aixiaocheng	neutral(中性)、newscast(广播)
爱小雪	aixiaoxue	neutral(中性)、newscast(广播)、customer-service(客服)
爱小璐	aixiaolu	neutral(中性)、story(故事)、customer-service(客服)
爱小栋	aixiaodong	neutral(中性)、newscast(广播)
爱小阳	aixiaoyang	neutral(中性)、newscast(广播)

响应

响应头

此接口仅返回公共响应头部，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>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
  <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>aixiaoxing</VoiceType>
      <Volume>2</Volume>
      <Speed>200</Speed>
      <Emotion>arousal</Emotion>
    </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

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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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

版权所有：腾讯云计算（北京）有限责任公司

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
Emotion	Response.Template.TtsTpl	同请求体中的 Request.Emotion

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-
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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <RequestId>NjQ0NGNlNDZfMThhYTk0MGFfY*********</RequestId>
    <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>aixiaoxing</VoiceType>
        <Volume>2</Volume>
        <Speed>200</Speed>
        <Emotion>arousal</Emotion>
      </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

更新模板

最近更新时间：2023-08-14 17:01:45

功能描述

更新语音合成模板。

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>
```

版权所有：腾讯云计算（北京）有限责任公司

具体数据描述如下：

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>
  <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <Template>
    <Tag>Tts</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <TtsTpl>
      <Mode>Sync</Mode>
      <Codec>pcm</Codec>
      <VoiceType>ruxue</VoiceType>
      <Volume>2</Volume>
      <Speed>200</Speed>
    </TtsTpl>
    <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
    <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
  </Template>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	同创建语音合成模板接口的 Response	Container

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
PUT /template/t1460606b9752148c4ab182f55163ba7cd HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x

<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <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>
```

语音识别
提交任务

最近更新时间：2023-08-14 17:01:45

功能描述

提交一个语音识别任务。

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。
- 极速ASR仅支持100 MB以内且时长不超过2小时的音频文件
- 普通ASR的音频时长不能长于5小时，文件大小不超过512MB

请求头

此接口仅使用公共请求头部，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>
  </Input>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <Operation>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <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>
    <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
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	文件路径	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
TemplateId	Request.Operation	语音识别模板 ID	String	否
SpeechRecognition	Request.Operation	语音识别参数，同创建语音识别模板接口中的 Request.SpeechRecognition	Container	否
Output	Request.Operation	结果输出配置	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

语音识别参数必须设置，可以通过 TemplateId 或 SpeechRecognition 设置， TemplateId 优先级更高。

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.O	存储结果的存储桶	Stri	是

版权所有：腾讯云计算（北京）有限责任公司

	utput		ng	
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>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>
      <Output>
        <Bucket>test-1234567890</Bucket>
        <Object>output/asr.txt</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
      <TemplateName>speech_demo</TemplateName>
      <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 数组

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	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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.J	任务的结束时间	Strin

	obsDetail		g
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
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
SpeechRecognition	Response.JobsDetail.Operation	同请求中的 Request.Operation.SpeechRecognition	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
SpeechRecognitionResult	Response.JobsDetail.Operation	语音识别任务结果，没有时不返回	Container

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	Int
text	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult	声道音频完整识别结果	String
sentence_list	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult	句子/段落级别的识别结果列表	Container 数组

Container 节点 sentence_list 的内容：

节点名称（关键字）	父节点	描述	类型
text	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	句子/段落级别文本	String
start_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	开始时间	Int
end_time	Response.JobsDetail.Operation.S	结束时间	Int

版权所有：腾讯云计算（北京）有限责任公司

	peechRecognitionResult.FlashResult.sentence_list		
speaker_id	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	说话人 Id（请求中如果设置了 speaker_diarization，可以按照 speaker_id 来区分说话人）	Int
word_list	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list	词级别的识别结果列表	Container 数组

Container 节点 word_list 的内容：

节点名称（关键字）	父节点	描述	类型
word	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list.word_list	词级别文本	String
start_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list.word_list	开始时间	Int
end_time	Response.JobsDetail.Operation.SpeechRecognitionResult.FlashResult.sentence_list.word_list	结束时间	Int

Container 节点 ResultDetail 的内容：

节点名称（关键字）	父节点	描述	类型
FinalSentence	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句最终识别结果	String
SliceSentence	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句中间识别结果，使用空格拆分为多个词	String
StartMs	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句开始时间（毫秒）	String

版权所有：腾讯云计算（北京）有限责任公司

EndMs	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句结束时间（毫秒）	String
WordsNum	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句中词个数	String
SpeechSpeed	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句语速，单位：字数/秒	String
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）	String
Words	Response.JobsDetail.Operation.SpeechRecognitionResult.ResultDetail	单句中词详情	Container 数组

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：使用语音识别模板 ID

```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <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>
    <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>
         <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
         <TemplateName>speech_demo</TemplateName>
         <UserData>This is my data.</UserData>
         <JobLevel>0</JobLevel>
      </Operation>
      <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
      <StartTime>-</StartTime>
      <State>Submitted</State>
      <Tag>SpeechRecognition</Tag>
   </JobsDetail>
</Response>
```

请求2：使用语音识别参数

```
POST /jobs HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>
    <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>
        <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>
```

任务回调

最近更新时间：2023-01-31 14:09:55

功能说明

数据万象支持自定义设置回调 URL，在任务完成后，系统向该 URL 发送 HTTP POST 请求，请求体中包含通知内容。您可通过配置的回调地址及时了解任务处理的进展和状态，以便进行其他业务操作。

回调内容

任务完成后，系统会向您设置的回调地址发送回调内容，该响应体返回为 application/xml 数据，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <State>Success</State>
    <Tag>SpeechRecognition</Tag>
    <Code>Success</Code>
    <CreationTime>2022-09-02T10:25:38+0800</CreationTime>
    <StartTime>2022-09-02T10:25:39+0800</StartTime>
    <EndTime>2022-09-02T10:25:46+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>minnong.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>s886c0a7c2a6611ed847a618901112dcf</JobId>
    <QueueId>pe91d0af11fc14337987ff0c34f8b0886</QueueId>
    <Message/>
    <Operation>
      <UserData>This is my SpeechRecognition job.</UserData>
      <JobLevel>0</JobLevel>
      <Output>
        <Bucket>test-1234567890</Bucket>
        <Object>/example.txt</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <SpeechRecognitionResult>
        <AudioTime>19.7124</AudioTime>
        <DetailObjectName>/example.txt.detail</DetailObjectName>
        <ObjectName>/example.txt</ObjectName>
        <Result>[0:3.320,0:19.712] 悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒
```

```
        <ResultDetail>
          <EndMs>19712</EndMs>
          <FinalSentence>悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒粒皆辛苦。
          <SliceSentence>悯 农 ， 李 绅 ， 锄禾 日 当午 ， 汗滴 禾 下 土 。 谁知 盘中餐 ，
          <SpeakerId>0</SpeakerId>
          <SpeechSpeed>2.4</SpeechSpeed>
          <StartMs>3320</StartMs>
          <Words>
            <OffsetEndMs>2190</OffsetEndMs>
            <OffsetStartMs>0</OffsetStartMs>
            <VoiceType>0</VoiceType>
            <Word>悯</Word>
          </Words>
          <Words>
            <OffsetEndMs>2625</OffsetEndMs>
            <OffsetStartMs>2190</OffsetStartMs>
            <VoiceType>0</VoiceType>
            <Word>农</Word>
          </Words>
          <Words>
            <OffsetEndMs>2625</OffsetEndMs>
            <OffsetStartMs>2190</OffsetStartMs>
            <VoiceType>1</VoiceType>
            <Word>，</Word>
          </Words>
          <Words>
            <OffsetEndMs>3060</OffsetEndMs>
            <OffsetStartMs>2625</OffsetStartMs>
            <VoiceType>0</VoiceType>
            <Word>李</Word>
          </Words>
          <Words>
            <OffsetEndMs>3825</OffsetEndMs>
            <OffsetStartMs>3060</OffsetStartMs>
            <VoiceType>0</VoiceType>
            <Word>绅</Word>
          </Words>
          <Words>
            <OffsetEndMs>3825</OffsetEndMs>
            <OffsetStartMs>3060</OffsetStartMs>
            <VoiceType>1</VoiceType>
            <Word>，</Word>
          </Words>
          <Words>
            <OffsetEndMs>5040</OffsetEndMs>
```

```
        <OffsetStartMs>3825</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>锄禾</Word>
      </Words>
      <Words>
        <OffsetEndMs>5475</OffsetEndMs>
        <OffsetStartMs>5040</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>日</Word>
      </Words>
      <Words>
        <OffsetEndMs>6390</OffsetEndMs>
        <OffsetStartMs>5475</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>当午</Word>
      </Words>
      <Words>
        <OffsetEndMs>6390</OffsetEndMs>
        <OffsetStartMs>5475</OffsetStartMs>
        <VoiceType>1</VoiceType>
        <Word>，</Word>
      </Words>
      <Words>
        <OffsetEndMs>7365</OffsetEndMs>
        <OffsetStartMs>6390</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>汗滴</Word>
      </Words>
      <Words>
        <OffsetEndMs>7755</OffsetEndMs>
        <OffsetStartMs>7365</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>禾</Word>
      </Words>
      <Words>
        <OffsetEndMs>8130</OffsetEndMs>
        <OffsetStartMs>7755</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>下</Word>
      </Words>
      <Words>
        <OffsetEndMs>8850</OffsetEndMs>
        <OffsetStartMs>8130</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>土</Word>
```

```
        </Words>
        <Words>
          <OffsetEndMs>8850</OffsetEndMs>
          <OffsetStartMs>8130</OffsetStartMs>
          <VoiceType>1</VoiceType>
          <Word>。</Word>
        </Words>
        <Words>
          <OffsetEndMs>9915</OffsetEndMs>
          <OffsetStartMs>8850</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>谁知</Word>
        </Words>
        <Words>
          <OffsetEndMs>11130</OffsetEndMs>
          <OffsetStartMs>9915</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>盘中餐</Word>
        </Words>
        <Words>
          <OffsetEndMs>11130</OffsetEndMs>
          <OffsetStartMs>9915</OffsetStartMs>
          <VoiceType>1</VoiceType>
          <Word>，</Word>
        </Words>
        <Words>
          <OffsetEndMs>12045</OffsetEndMs>
          <OffsetStartMs>11130</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>粒粒</Word>
        </Words>
        <Words>
          <OffsetEndMs>12540</OffsetEndMs>
          <OffsetStartMs>12045</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>皆</Word>
        </Words>
        <Words>
          <OffsetEndMs>14745</OffsetEndMs>
          <OffsetStartMs>12540</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>辛苦</Word>
        </Words>
        <Words>
          <OffsetEndMs>14745</OffsetEndMs>
```

```
                <OffsetStartMs>12540</OffsetStartMs>
                <VoiceType>1</VoiceType>
                <Word>。</Word>
            </Words>
            <WordsNum>22</WordsNum>
        </ResultDetail>
    </SpeechRecognitionResult>
    <TemplateId>t14aea553bb55c468b963fafa472afc538</TemplateId>
    <TemplateName>test</TemplateName>
  </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数组

Container 节点 JobsDetail 的内容：
同提交语音识别任务接口中的 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>
    <State>Success</State>
    <Tag>SpeechRecognition</Tag>
    <Code>Success</Code>
    <CreationTime>2022-09-02T10:25:38+0800</CreationTime>
    <StartTime>2022-09-02T10:25:39+0800</StartTime>
    <EndTime>2022-09-02T10:25:46+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>minnong.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>s886c0a7c2a6611ed847a618901112dcf</JobId>
    <QueueId>pe91d0af11fc14337987ff0c34f8b0886</QueueId>
    <Message/>
    <Operation>
      <UserData>This is my SpeechRecognition job.</UserData>
      <JobLevel>0</JobLevel>
      <Output>
        <Bucket>test-1234567890</Bucket>
        <Object>/example.txt</Object>
        <Region>ap-chongqing</Region>
      </Output>
      <SpeechRecognitionResult>
        <AudioTime>19.7124</AudioTime>
        <DetailObjectName>/example.txt.detail</DetailObjectName>
        <ObjectName>/example.txt</ObjectName>
        <Result>[0:3.320,0:19.712] 悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒
        <ResultDetail>
```

```
<EndMs>19712</EndMs>
<FinalSentence>悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒粒皆辛苦。
<SliceSentence>悯 农 ， 李 绅 ， 锄禾 日 当午 ， 汗滴 禾 下 土 。 谁知 盘中餐 ，
<SpeakerId>0</SpeakerId>
<SpeechSpeed>2.4</SpeechSpeed>
<StartMs>3320</StartMs>
<Words>
  <OffsetEndMs>2190</OffsetEndMs>
  <OffsetStartMs>0</OffsetStartMs>
  <VoiceType>0</VoiceType>
  <Word>悯</Word>
</Words>
<Words>
  <OffsetEndMs>2625</OffsetEndMs>
  <OffsetStartMs>2190</OffsetStartMs>
  <VoiceType>0</VoiceType>
  <Word>农</Word>
</Words>
<Words>
  <OffsetEndMs>2625</OffsetEndMs>
  <OffsetStartMs>2190</OffsetStartMs>
  <VoiceType>1</VoiceType>
  <Word>，</Word>
</Words>
<Words>
  <OffsetEndMs>3060</OffsetEndMs>
  <OffsetStartMs>2625</OffsetStartMs>
  <VoiceType>0</VoiceType>
  <Word>李</Word>
</Words>
<Words>
  <OffsetEndMs>3825</OffsetEndMs>
  <OffsetStartMs>3060</OffsetStartMs>
  <VoiceType>0</VoiceType>
  <Word>绅</Word>
</Words>
<Words>
  <OffsetEndMs>3825</OffsetEndMs>
  <OffsetStartMs>3060</OffsetStartMs>
  <VoiceType>1</VoiceType>
  <Word>，</Word>
</Words>
<Words>
  <OffsetEndMs>5040</OffsetEndMs>
  <OffsetStartMs>3825</OffsetStartMs>
```

```
        <VoiceType>0</VoiceType>
        <Word>锄禾</Word>
      </Words>
      <Words>
        <OffsetEndMs>5475</OffsetEndMs>
        <OffsetStartMs>5040</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>日</Word>
      </Words>
      <Words>
        <OffsetEndMs>6390</OffsetEndMs>
        <OffsetStartMs>5475</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>当午</Word>
      </Words>
      <Words>
        <OffsetEndMs>6390</OffsetEndMs>
        <OffsetStartMs>5475</OffsetStartMs>
        <VoiceType>1</VoiceType>
        <Word>，</Word>
      </Words>
      <Words>
        <OffsetEndMs>7365</OffsetEndMs>
        <OffsetStartMs>6390</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>汗滴</Word>
      </Words>
      <Words>
        <OffsetEndMs>7755</OffsetEndMs>
        <OffsetStartMs>7365</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>禾</Word>
      </Words>
      <Words>
        <OffsetEndMs>8130</OffsetEndMs>
        <OffsetStartMs>7755</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>下</Word>
      </Words>
      <Words>
        <OffsetEndMs>8850</OffsetEndMs>
        <OffsetStartMs>8130</OffsetStartMs>
        <VoiceType>0</VoiceType>
        <Word>土</Word>
      </Words>
```

```
        <Words>
          <OffsetEndMs>8850</OffsetEndMs>
          <OffsetStartMs>8130</OffsetStartMs>
          <VoiceType>1</VoiceType>
          <Word>。</Word>
        </Words>
        <Words>
          <OffsetEndMs>9915</OffsetEndMs>
          <OffsetStartMs>8850</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>谁知</Word>
        </Words>
        <Words>
          <OffsetEndMs>11130</OffsetEndMs>
          <OffsetStartMs>9915</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>盘中餐</Word>
        </Words>
        <Words>
          <OffsetEndMs>11130</OffsetEndMs>
          <OffsetStartMs>9915</OffsetStartMs>
          <VoiceType>1</VoiceType>
          <Word>，</Word>
        </Words>
        <Words>
          <OffsetEndMs>12045</OffsetEndMs>
          <OffsetStartMs>11130</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>粒粒</Word>
        </Words>
        <Words>
          <OffsetEndMs>12540</OffsetEndMs>
          <OffsetStartMs>12045</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>皆</Word>
        </Words>
        <Words>
          <OffsetEndMs>14745</OffsetEndMs>
          <OffsetStartMs>12540</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>辛苦</Word>
        </Words>
        <Words>
          <OffsetEndMs>14745</OffsetEndMs>
          <OffsetStartMs>12540</OffsetStartMs>
```

```
            <VoiceType>1</VoiceType>
            <Word>。</Word>
          </Words>
          <WordsNum>22</WordsNum>
        </ResultDetail>
      </SpeechRecognitionResult>
      <TemplateId>t14aea553bb55c468b963fafa472afc538</TemplateId>
      <TemplateName>test</TemplateName>
    </Operation>
  </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>2022-09-02T10:25:38+0800</CreationTime>
    <EndTime>2022-09-02T10:25:46+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>minnong.mp3</Object>
      <Region>ap-chongqing</Region>
      <CosHeaders>
        <Key>Content-Type</Key>
        <Value>Audio/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>1424687</Value>
      </CosHeaders>
    </Input>
    <JobId>s886c0a7c2a6611ed847a618901112dcf</JobId>
    <Message/>
    <Operation>
```

```
<UserData>This is my SpeechRecognition job.</UserData>
<JobLevel>0</JobLevel>
<Output>
  <Bucket>test-1234567890</Bucket>
  <Object>/example.txt</Object>
  <Region>ap-chongqing</Region>
</Output>
<SpeechRecognitionResult>
  <AudioTime>19.7124</AudioTime>
  <DetailObjectName>/example.txt.detail</DetailObjectName>
  <ObjectName>/example.txt</ObjectName>
  <Result>[0:3.320,0:19.712] 悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒
  <ResultDetail>
    <EndMs>19712</EndMs>
    <FinalSentence>悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒粒皆辛苦。
    <SliceSentence>悯 农 ， 李 绅 ， 锄禾 日 当午 ， 汗滴 禾 下 土 。 谁知 盘中餐 ，
    <SpeakerId>0</SpeakerId>
    <SpeechSpeed>2.4</SpeechSpeed>
    <StartMs>3320</StartMs>
    <Words>
      <OffsetEndMs>2190</OffsetEndMs>
      <OffsetStartMs>0</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>悯</Word>
    </Words>
    <Words>
      <OffsetEndMs>2625</OffsetEndMs>
      <OffsetStartMs>2190</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>农</Word>
    </Words>
    <Words>
      <OffsetEndMs>2625</OffsetEndMs>
      <OffsetStartMs>2190</OffsetStartMs>
      <VoiceType>1</VoiceType>
      <Word>，</Word>
    </Words>
    <Words>
      <OffsetEndMs>3060</OffsetEndMs>
      <OffsetStartMs>2625</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>李</Word>
    </Words>
    <Words>
      <OffsetEndMs>3825</OffsetEndMs>
```

```
      <OffsetStartMs>3060</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>绅</Word>
    </Words>
    <Words>
      <OffsetEndMs>3825</OffsetEndMs>
      <OffsetStartMs>3060</OffsetStartMs>
      <VoiceType>1</VoiceType>
      <Word>，</Word>
    </Words>
    <Words>
      <OffsetEndMs>5040</OffsetEndMs>
      <OffsetStartMs>3825</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>锄禾</Word>
    </Words>
    <Words>
      <OffsetEndMs>5475</OffsetEndMs>
      <OffsetStartMs>5040</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>日</Word>
    </Words>
    <Words>
      <OffsetEndMs>6390</OffsetEndMs>
      <OffsetStartMs>5475</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>当午</Word>
    </Words>
    <Words>
      <OffsetEndMs>6390</OffsetEndMs>
      <OffsetStartMs>5475</OffsetStartMs>
      <VoiceType>1</VoiceType>
      <Word>，</Word>
    </Words>
    <Words>
      <OffsetEndMs>7365</OffsetEndMs>
      <OffsetStartMs>6390</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>汗滴</Word>
    </Words>
    <Words>
      <OffsetEndMs>7755</OffsetEndMs>
      <OffsetStartMs>7365</OffsetStartMs>
      <VoiceType>0</VoiceType>
      <Word>禾</Word>
```

```
        </Words>
        <Words>
          <OffsetEndMs>8130</OffsetEndMs>
          <OffsetStartMs>7755</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>下</Word>
        </Words>
        <Words>
          <OffsetEndMs>8850</OffsetEndMs>
          <OffsetStartMs>8130</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>土</Word>
        </Words>
        <Words>
          <OffsetEndMs>8850</OffsetEndMs>
          <OffsetStartMs>8130</OffsetStartMs>
          <VoiceType>1</VoiceType>
          <Word>。</Word>
        </Words>
        <Words>
          <OffsetEndMs>9915</OffsetEndMs>
          <OffsetStartMs>8850</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>谁知</Word>
        </Words>
        <Words>
          <OffsetEndMs>11130</OffsetEndMs>
          <OffsetStartMs>9915</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>盘中餐</Word>
        </Words>
        <Words>
          <OffsetEndMs>11130</OffsetEndMs>
          <OffsetStartMs>9915</OffsetStartMs>
          <VoiceType>1</VoiceType>
          <Word>，</Word>
        </Words>
        <Words>
          <OffsetEndMs>12045</OffsetEndMs>
          <OffsetStartMs>11130</OffsetStartMs>
          <VoiceType>0</VoiceType>
          <Word>粒粒</Word>
        </Words>
        <Words>
          <OffsetEndMs>12540</OffsetEndMs>
```

```
            <OffsetStartMs>12045</OffsetStartMs>
            <VoiceType>0</VoiceType>
            <Word>皆</Word>
          </Words>
          <Words>
            <OffsetEndMs>14745</OffsetEndMs>
            <OffsetStartMs>12540</OffsetStartMs>
            <VoiceType>0</VoiceType>
            <Word>辛苦</Word>
          </Words>
          <Words>
            <OffsetEndMs>14745</OffsetEndMs>
            <OffsetStartMs>12540</OffsetStartMs>
            <VoiceType>1</VoiceType>
            <Word>。</Word>
          </Words>
          <WordsNum>22</WordsNum>
        </ResultDetail>
      </SpeechRecognitionResult>
      <TemplateId>t14aea553bb55c468b963fafa472afc538</TemplateId>
      <TemplateName>test</TemplateName>
    </Operation>
    <QueueId>p2242ab62c7c94486915508540933a2c6</QueueId>
    <StartTime>2022-09-02T10:25:39+0800</StartTime>
    <State>Success</State>
    <Tag>SpeechRecognition</Tag>
    <Workflow>
      <Name>SpeechRecognition_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": "2022-09-02T10:25:38+0800",
    "EndTime": "2022-09-02T10:25:46+0800",
    "Input": {
```

```
        "BucketId": "test-1234567890",
        "Object": "minnong.mp3",
        "Region": "ap-chongqing",
        "CosHeaders": [{
              "Key": "Content-Type",
              "Value": "Audio/mp3"
            },
            {
              "Key": "x-cos-request-id",
              "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
            },
            {
              "Key": "EventName",
              "Value": "cos:ObjectCreated:Put"
            },
            {
              "Key": "Size",
              "Value": "1424687"
            }
        ]
    },
    "JobId": "s886c0a7c2a6611ed847a618901112dcf",
    "Operation": {
        "UserData": "This is my SpeechRecognition job.",
        "JobLevel": "0",
        "Output": {
            "Bucket": "test-1234567890",
            "Object": "/example.txt",
            "Region": "ap-chongqing"
        },
        "SpeechRecognitionResult": {
            "AudioTime": "19.7124",
            "DetailObjectName": "/example.txt.detail",
            "ObjectName": "/example.txt",
            "Result": "[0:3.320,0:19.712]  悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒
            "ResultDetail": {
                "EndMs": "19712",
                "FinalSentence": "悯农，李绅，锄禾日当午，汗滴禾下土。谁知盘中餐，粒粒皆辛苦。
                "SliceSentence": "悯 农 ， 李 绅 ， 锄禾 日 当午 ， 汗滴 禾 下 土 。 谁知 盘中餐 ，
                "SpeakerId": "0",
                "SpeechSpeed": "2.4",
                "StartMs": "3320",
                "Words": [{
                        "OffsetEndMs": "2190",
                        "OffsetStartMs": "0",
```

```
            "VoiceType": "0",
            "Word": "悯"
        },
        {
            "OffsetEndMs": "2625",
            "OffsetStartMs": "2190",
            "VoiceType": "0",
            "Word": "农"
        },
        {
            "OffsetEndMs": "2625",
            "OffsetStartMs": "2190",
            "VoiceType": "1",
            "Word": "，"
        },
        {
            "OffsetEndMs": "3060",
            "OffsetStartMs": "2625",
            "VoiceType": "0",
            "Word": "李"
        },
        {
            "OffsetEndMs": "3825",
            "OffsetStartMs": "3060",
            "VoiceType": "0",
            "Word": "绅"
        },
        {
            "OffsetEndMs": "3825",
            "OffsetStartMs": "3060",
            "VoiceType": "1",
            "Word": "，"
        },
        {
            "OffsetEndMs": "5040",
            "OffsetStartMs": "3825",
            "VoiceType": "0",
            "Word": "锄禾"
        },
        {
            "OffsetEndMs": "5475",
            "OffsetStartMs": "5040",
            "VoiceType": "0",
            "Word": "日"
        },
```


版权所有：腾讯云计算（北京）有限责任公司

```
            {
                "OffsetEndMs": "6390",
                "OffsetStartMs": "5475",
                "VoiceType": "0",
                "Word": "当午"
            },
            {
                "OffsetEndMs": "6390",
                "OffsetStartMs": "5475",
                "VoiceType": "1",
                "Word": "，"
            },
            {
                "OffsetEndMs": "7365",
                "OffsetStartMs": "6390",
                "VoiceType": "0",
                "Word": "汗滴"
            },
            {
                "OffsetEndMs": "7755",
                "OffsetStartMs": "7365",
                "VoiceType": "0",
                "Word": "禾"
            },
            {
                "OffsetEndMs": "8130",
                "OffsetStartMs": "7755",
                "VoiceType": "0",
                "Word": "下"
            },
            {
                "OffsetEndMs": "8850",
                "OffsetStartMs": "8130",
                "VoiceType": "0",
                "Word": "土"
            },
            {
                "OffsetEndMs": "8850",
                "OffsetStartMs": "8130",
                "VoiceType": "1",
                "Word": "。"
            },
            {
                "OffsetEndMs": "9915",
                "OffsetStartMs": "8850",
```


版权所有：腾讯云计算（北京）有限责任公司

```
                "VoiceType": "0",
                "Word": "谁知"
            },
            {
                "OffsetEndMs": "11130",
                "OffsetStartMs": "9915",
                "VoiceType": "0",
                "Word": "盘中餐"
            },
            {
                "OffsetEndMs": "11130",
                "OffsetStartMs": "9915",
                "VoiceType": "1",
                "Word": "，"
            },
            {
                "OffsetEndMs": "12045",
                "OffsetStartMs": "11130",
                "VoiceType": "0",
                "Word": "粒粒"
            },
            {
                "OffsetEndMs": "12540",
                "OffsetStartMs": "12045",
                "VoiceType": "0",
                "Word": "皆"
            },
            {
                "OffsetEndMs": "14745",
                "OffsetStartMs": "12540",
                "VoiceType": "0",
                "Word": "辛苦"
            },
            {
                "OffsetEndMs": "14745",
                "OffsetStartMs": "12540",
                "VoiceType": "1",
                "Word": "。"
            }
        ],
        "WordsNum": "22"
    }
},
"TemplateId": "t14aea553bb55c468b963fafa472afc538",
"TemplateName": "test"
```

```
      },
      "QueueId": "p2242ab62c7c94486915508540933a2c6",
      "StartTime": "2022-09-02T10:25:39+0800",
      "State": "Success",
      "Tag": "SpeechRecognition",
      "Workflow": {
        "Name": "SpeechRecognition_1581665960537",
        "RunId": "ic90edd59f84f11ec9d4f525400a3c59f",
        "WorkflowId": "web6ac56c1ef54dbfa44d7f4103203be9",
        "WorkflowName": "workflow-test"
      }
    }]
}
```

创建模板

最近更新时间：2023-08-01 16:46:27

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>
    <ChannelNum>2</ChannelNum>
    <ResTextFormat>1</ResTextFormat>
    <FilterDirty>0</FilterDirty>
    <FilterModal>1</FilterModal>
    <ConvertNumMode>0</ConvertNumMode>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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
Tag	Request	模板类型：SpeechRecognition	String	是
Name	Request	模板名称，仅支持中文、英文、数字、_、-和*，长度不超过 64	String	是
SpeechRecognition	Request	语音识别参数	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	默认值	是否必选
FlashAsr	Request.SpeechRecognition	开启极速ASR，取值 true/false	String	false	否
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（仅适用于单声道音频）	String	无	是

		• 8k_en：电话 8k 英语 非电话场景： • 6k_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 支持8k_zh、16k_zh、16k_en、16k_zh_video、16k_zh_dialect、16k_ms（马来语）、16k_zh-PY（中英粤）			
ChannelNum	Request.SpeechRecognition	语音声道数： • 1 表示单声道。EngineModelType为非电话场景仅支持单声道 • 2 表示双声道（仅支持 8k_zh 引擎模型 双声道应分别对应通话双方） • 仅支持非极速ASR，为非极速ASR时，该参数必填	String	无	否
ResTextFormat	Request.SpeechRecognition	识别结果返回形式： • 0：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） • 3：标点符号分段，包含每段时间戳，特别适用于字幕场景（包含词级时间、标点、语速值） • 仅支持非极速ASR	String	无	否
FilterDirty	Request.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 *	String	0	否

FilterModal	Request.SpeechRecognition	是否过滤语气词（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	String	0	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：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 • 3：打开数学相关数字转换 • 仅支持非极速ASR	String	0	否
SpeakerDiarization	Request.SpeechRecognition	是否开启说话人分离 • 0：不开启 • 1：开启(仅支持8k_zh，16k_zh，16k_zh_video，单声道音频) • 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。	String	0	否
SpeakerNumber	Request.SpeechRecognition	• 说话人分离人数（需配合开启说话人分离使用），取值范围：[0, 10] • 0 代表自动分离（目前仅支持≤6个人） • 1-10代表指定说话人数分离 • 仅支持非极速ASR	String	0	否
FilterPunc	Request.SpeechRecognition	是否过滤标点符号（目前支持中文普通话引擎） • 0：不过滤。 • 1：过滤句末标点 • 2：过滤所有标点	String	0	否
OutputFileType	Request.SpeechRecognition	输出文件类型，可选txt、srt • 极速ASR仅支持txt • 非极速Asr且ResTextFormat为3时仅支持txt	String	txt	否
Format	Request.SpeechRecognition	极速ASR音频格式，支持 wav、pcm、ogg-opus、speex、silk、mp3、m4a、aac • 极速ASR时，该参数必填	String	无	否

版权所有：腾讯云计算（北京）有限责任公司

FirstChannelOnly	Request.SpeechRecognition	是否识别首个声道 • 0：识别所有声道 • 1：识别首个声道 • 仅支持极速ASR	String	1	否
WordInfo	Request.SpeechRecognition	是否显示词级别时间戳 • 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 • 仅支持极速ASR	String	0	否
SentenceMaxLength	Request.SpeechRecognition	单标点最多字数，取值范围：[6，40] • 默认值为 0 表示不开启该功能 • 该参数可用于字幕生成场景，控制单行字幕最大字数 • 当FlashAsr为false时，仅ResTextFormat为3时参数有效	String	0	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>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <Template>
    <Tag>SpeechRecognition</Tag>
    <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
    <Name>TemplateName</Name>
    <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>2</ChannelNum>
      <ResTextFormat>1</ResTextFormat>
      <FilterDirty>0</FilterDirty>
```

```
        <FilterModal>1</FilterModal>
        <ConvertNumMode>0</ConvertNumMode>
        <SpeakerDiarization>1</SpeakerDiarization>
        <SpeakerNumber>0</SpeakerNumber>
        <FilterPunc>0</FilterPunc>
        <OutputFileType>txt</OutputFileType>
        <SentenceMaxLength>0</SentenceMaxLength>
      </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
Template	Response	保存模板详情的容器	Container
RequestId	Response	请求的唯一 ID	String

Container 节点 Template 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-
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>
      <ChannelNum>1</ChannelNum>
      <ResTextFormat>1</ResTextFormat>
      <FilterDirty>0</FilterDirty>
      <FilterModal>1</FilterModal>
      <ConvertNumMode>0</ConvertNumMode>
      <SpeakerDiarization>1</SpeakerDiarization>
      <SpeakerNumber>0</SpeakerNumber>
      <FilterPunc>0</FilterPunc>
      <OutputFileType>txt</OutputFileType>
      <SentenceMaxLength>0</SentenceMaxLength>
   </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
```

版权所有：腾讯云计算（北京）有限责任公司

```
Server: tencent-ci
x-ci-request-id: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x

<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <Template>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Name>TemplateName</Name>
        <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>
            <SentenceMaxLength>0</SentenceMaxLength>
        </SpeechRecognition>
    </Template>
</Response>
```

更新模板

最近更新时间：2023-08-14 17:01:46

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>
    <ChannelNum>1</ChannelNum>
    <ResTextFormat>1</ResTextFormat>
    <FilterDirty>0</FilterDirty>
    <FilterModal>1</FilterModal>
    <ConvertNumMode>0</ConvertNumMode>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <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>
  <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
  <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-
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>
    <ChannelNum>1</ChannelNum>
    <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: NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x

<Response>
    <RequestId>NjJmMWQxYjNfOTBmYTUwNjRfNWYyY18x</RequestId>
    <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

听歌识曲
提交任务

最近更新时间：2023-08-17 11:05:46

功能描述

提交一个听歌识曲任务。

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>SoundHound</Tag>
  <Input>
    <Object>input/test.mp3</Object>
  </Input>
  <Operation>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
```

```
    </Operation>
    <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：SoundHound	String	是
Input	Request	待操作的对象信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	文件路径	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
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

响应

响应头

此接口仅返回公共响应头部，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>s03f2ff944ece11edb780e50b111e7626</JobId>
    <Message/>
    <Operation>
      <UserData>This is my data.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <StartTime>-</StartTime>
        <State>Submitted</State>
        <Tag>SoundHound</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 数组

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：SoundHound	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String
SoundHoundResult	Response.JobsDetail.Operation	听歌识曲任务结果，没有时不返回	Container

Container 节点 SoundHoundResult 的内容：

节点名称（关键字）	父节点	描述	类型
SongList	Response.JobsDetail.Operation.SoundHoundResult	识别结果详情	Container 数组

Container 节点 SongList 的内容：

节点名称（关键字）	父节点	描述	类型
Inlier	Response.JobsDetail.Operation.SoundHoundResult.SongList	识别置信度	Int

版权所有：腾讯云计算（北京）有限责任公司

SingerName	Response.JobsDetail.Operation.SoundHoundResult.SongList	歌手姓名	String
SongName	Response.JobsDetail.Operation.SoundHoundResult.SongList	歌曲名	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-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 166
Content-Type: application/xml

<Request>
    <Tag>SoundHound</Tag>
    <Input>
        <Object>input/test.mp3</Object>
    </Input>
    <Operation>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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>s03f2ff944ece11edb780e50b111e7626</JobId>
        <Message/>
        <Operation>
            <UserData>This is my data.</UserData>
            <JobLevel>0</JobLevel>
        </Operation>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <StartTime>-</StartTime>
        <State>Submitted</State>
        <Tag>SoundHound</Tag>
    </JobsDetail>
</Response>
```

任务回调

最近更新时间：2023-08-17 11:05:46

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，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2022-10-18T18:22:05+0800</CreationTime>
    <EndTime>2022-10-18T18:22:10+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>input/test.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>sb6b64de84ece11eda7ff3f95cf53fd82</JobId>
    <Message/>
    <Operation>
      <JobLevel>1</JobLevel>
      <SoundHoundResult>
        <SongList>
          <Inlier>0.98</Inlier>
          <SingerName>周杰伦</SingerName>
          <SongName>青花瓷</SongName>
        </SongList>
      </SoundHoundResult>
      <UserData>This is my SoundHound job.</UserData>
    </Operation>
    <QueueId>p4986e06c00fe4ead808bd7e4942089e1</QueueId>
    <StartTime>2022-10-18T18:22:06+0800</StartTime>
    <State>Success</State>
    <Tag>SoundHound</Tag>
  </JobsDetail>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	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	同提交听歌识曲任务接口中的 Response.JobsDetail	Container 数组

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>2022-10-18T18:22:05+0800</CreationTime>
    <EndTime>2022-10-18T18:22:10+0800</EndTime>
    <Input>
      <BucketId>test-1234567890</BucketId>
      <Object>input/test.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>sb6b64de84ece11eda7ff3f95cf53fd82</JobId>
    <Message/>
    <Operation>
      <JobLevel>1</JobLevel>
      <SoundHoundResult>
        <SongList>
          <Inlier>0.98</Inlier>
          <SingerName>周杰伦</SingerName>
          <SongName>青花瓷</SongName>
        </SongList>
      </SoundHoundResult>
      <UserData>This is my SoundHound job.</UserData>
    </Operation>
    <QueueId>p4986e06c00fe4ead808bd7e4942089e1</QueueId>
    <StartTime>2022-10-18T18:22:06+0800</StartTime>
    <State>Success</State>
    <Tag>SoundHound</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": "2022-10-18T18:22:05+0800",
    "EndTime": "2022-10-18T18:22:10+0800",
    "Input": {
      "BucketId": "test-1234567890",
      "Object": "input/test.mp3",
      "Region": "ap-chongqing"
    },
```

```
      "JobId": "sb6b64de84ece11eda7ff3f95cf53fd82",
      "Operation": {
        "JobLevel": "1",
        "SoundHoundResult": {
          "SongList": {
            "Inlier": "98",
            "SingerName": "周杰伦",
            "SongName": "青花瓷",
          }
        },
        "UserData": "This is my SoundHound job."
      },
      "QueueId": "p4986e06c00fe4ead808bd7e4942089e1",
      "StartTime": "2022-10-18T18:22:06+0800",
      "State": "Success",
      "Tag": "SoundHound"
    }]
}
```

音乐评分
提交任务

最近更新时间：2023-08-08 17:34:01

功能描述

提交一个音乐评分任务。

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>VocalScore</Tag>
  <Input>
    <Object>input/test.mp3</Object>
  </Input>
  <Operation>
    <VocalScore>
      <StandardObject>input/base.mp3</StandardObject>
```

```
        </VocalScore>
        <UserData>This is my data.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
    <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：VocalScore	String	是
Input	Request	待操作的对象信息	Container	是
Operation	Request	操作规则	Container	是
CallBackFormat	Request	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request	任务回调地址，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	父节点	描述	类	是否

键字）			型	必选
Object	Request.Input	文件路径	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
VocalScore	Request.Operation	音乐评分参数配置	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 VocalScore 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
StandardObject	Request.Operation	比对基准文件路径	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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>2000-07-18T10:54:49+0800</CreationTime>
    <EndTime>-</EndTime>
    <Input>
```

```
        <BucketId>test-1234567890</BucketId>
        <Object>test.mp3</Object>
        <Region>ap-chongqing</Region>
      </Input>
      <JobId>s76520462251611eeb22df1c8c02341ac</JobId>
      <Message/>
      <Operation>
        <JobLevel>0</JobLevel>
        <VocalScore>
          <StandardObject>test.mp3</StandardObject>
        </VocalScore>
      </Operation>
      <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
      <StartTime>-</StartTime>
      <State>Submitted</State>
      <Tag>VocalScore</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 数组

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：VocalScore	String
State	Response.JobsDetail	任务状态 • Submitted：已提交，待执行 • Running：执行中 • Success：执行成功 • Failed：执行失败 • Pause：任务暂停，当暂停队列时，待执行的任务会变为暂停状态 • Cancel：任务被取消执行	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
UserData	Response.JobsDetail.Operation	透传用户信息	String
JobLevel	Response.JobsDetail.Operation	任务优先级	String
VocalScore	Response.JobsDetail.Operation	同请求中的 Request.Operation.VocalScore	Container

版权所有：腾讯云计算（北京）有限责任公司

VocalScoreResult	Response.JobsDetail.Operation	音乐评分任务结果，没有时不返回	Container

Container 节点 VocalScoreResult 的内容：

节点名称（关键字）	父节点	描述	类型
PitchScore	Response.JobsDetail.Operation.VocalScoreResult	音调评分结果详情	Container
RhythemScore	Response.JobsDetail.Operation.VocalScoreResult	节奏评分结果详情	Container

Container 节点 PitchScore 的内容：

节点名称（关键字）	父节点	描述	类型
SentenceScores	Response.JobsDetail.Operation.VocalScoreResult.PitchScore	句子评分详情	Container数组
TotalScore	Response.JobsDetail.Operation.VocalScoreResult.PitchScore	音调总分	Int

Container 节点 SentenceScores 的内容：

节点名称（关键字）	父节点	描述	类型
StartTime	Response.JobsDetail.Operation.VocalScoreResult.PitchScore.SentenceScores	句子开始时间	Float
EndTime	Response.JobsDetail.Operation.VocalScoreResult.PitchScore.SentenceScores	句子结束时间	Float
Score	Response.JobsDetail.Operation.VocalScoreResult.PitchScore.SentenceScores	句子评分	Int

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 RhythemScore 的内容：

节点名称（关键字）	父节点	描述	类型
SentenceScores	Response.JobsDetail.Operation.VocalScoreResult.RhythemScore	句子评分详情	Container 数组
TotalScore	Response.JobsDetail.Operation.VocalScoreResult.RhythemScore	音调总分	Int

Container 节点 SentenceScores 的内容：

节点名称（关键字）	父节点	描述	类型
StartTime	Response.JobsDetail.Operation.VocalScoreResult.RhythemScore.SentenceScores	句子开始时间	Float
EndTime	Response.JobsDetail.Operation.VocalScoreResult.RhythemScore.SentenceScores	句子结束时间	Float
Score	Response.JobsDetail.Operation.VocalScoreResult.RhythemScore.SentenceScores	句子评分	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

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

```
<Request>
  <Tag>VocalScore</Tag>
  <Input>
    <Object>input/test.mp3</Object>
  </Input>
  <Operation>
    <VocalScore>
      <StandardObject>base.mp3</StandardObject>
    </VocalScore>
    <UserData>This is my data.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <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>2023-07-18T16:14:36+0800</CreationTime>
    <EndTime>-</EndTime>
    <Input>
      <BucketId>testpic-1253960454</BucketId>
      <Object>input/test.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>s229aa658254311eeaf90856735d7e87d</JobId>
    <Message/>
    <Operation>
      <JobLevel>0</JobLevel>
      <VocalScore>
        <StandardObject>test.mp3</StandardObject>
      </VocalScore>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </Operation>
        <QueueId>pcd463e1467964d39ad2d3f66aacd8199</QueueId>
        <StartTime>-</StartTime>
        <State>Submitted</State>
        <Tag>VocalScore</Tag>
    </JobsDetail>
</Response>
```

任务回调

最近更新时间：2023-08-18 15:28:01

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，包含完整节点数据的内容展示如下：

```
<Response>
  <EventName>TaskFinish</EventName>
  <JobsDetail>
    <Code>Success</Code>
    <CreationTime>2023-07-18T16:14:36+0800</CreationTime>
    <EndTime>2023-07-18T16:15:54+0800</EndTime>
    <Input>
      <BucketId>testpic-1253960454</BucketId>
      <Object>test.mp3</Object>
      <Region>ap-chongqing</Region>
    </Input>
    <JobId>s229aa658254311eeaf90856735d7e87d</JobId>
    <Message>Success</Message>
    <Operation>
      <JobLevel>0</JobLevel>
      <VocalScore>
          <StandardObject>test.mp3</StandardObject>
          <StandardUrl/>
      </VocalScore>
      <VocalScoreResult>
        <PitchScore>
          <SentenceScores>
              <EndTime>200.811</EndTime>
              <Score>100</Score>
              <StartTime>0.572727</StartTime>
          </SentenceScores>
          <SentenceScores>
              <EndTime>210.4</EndTime>
              <Score>100</Score>
              <StartTime>200.811</StartTime>
          </SentenceScores>
```

版权所有：腾讯云计算（北京）有限责任公司

```
            <TotalScore>100</TotalScore>
          </PitchScore>
          <RhythemScore>
            <SentenceScores>
              <EndTime>200.811</EndTime>
              <Score>100</Score>
              <StartTime>0.572727</StartTime>
            </SentenceScores>
            <SentenceScores>
              <EndTime>210.4</EndTime>
              <Score>100</Score>
              <StartTime>200.811</StartTime>
            </SentenceScores>
            <TotalScore>100</TotalScore>
          </RhythemScore>
        </VocalScoreResult>
      </Operation>
      <QueueId>p6617fd4048fa41faxxxxxxxxxxxx</QueueId>
      <QueueType>Speeching</QueueType>
      <StartTime>2023-07-18T16:14:37+0800</StartTime>
      <State>Success</State>
      <Tag>VocalScore</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	同提交音乐评分任务接口中的 Response.JobsDetail	Container 数组

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>2023-07-18T16:14:36+0800</CreationTime>
  <EndTime>2023-07-18T16:15:54+0800</EndTime>
  <Input>
    <BucketId>testpic-1253960454</BucketId>
    <Object>test.mp3</Object>
    <Region>ap-chongqing</Region>
  </Input>
  <JobId>s229aa658254311eeaf90856735d7e87d</JobId>
  <Message>Success</Message>
  <Operation>
    <JobLevel>0</JobLevel>
    <VocalScore>
        <StandardObject>test.mp3</StandardObject>
        <StandardUrl/>
    </VocalScore>
    <VocalScoreResult>
      <PitchScore>
        <SentenceScores>
            <EndTime>200.811</EndTime>
            <Score>100</Score>
            <StartTime>0.572727</StartTime>
        </SentenceScores>
        <SentenceScores>
            <EndTime>210.4</EndTime>
            <Score>100</Score>
            <StartTime>200.811</StartTime>
        </SentenceScores>
        <TotalScore>100</TotalScore>
      </PitchScore>
      <RhythemScore>
        <SentenceScores>
          <EndTime>200.811</EndTime>
          <Score>100</Score>
          <StartTime>0.572727</StartTime>
        </SentenceScores>
        <SentenceScores>
          <EndTime>210.4</EndTime>
          <Score>100</Score>
          <StartTime>200.811</StartTime>
        </SentenceScores>
        <TotalScore>100</TotalScore>
      </RhythemScore>
```

```
        </VocalScoreResult>
      </Operation>
      <QueueId>p6617fd4048fa41faxxxxxxxxxxxx</QueueId>
      <QueueType>Speeching</QueueType>
      <StartTime>2023-07-18T16:14:37+0800</StartTime>
      <State>Success</State>
      <Tag>VocalScore</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":"2023-07-18T16:14:36+0800",
      "EndTime":"2023-07-18T16:15:54+0800",
      "Input":{
         "BucketId":"testpic-1253960454",
         "Object":"test.mp3",
         "Region":"ap-chongqing"
      },
      "JobId":"s229aa658254311eeaf90856735d7e87d",
      "Message":"Success",
      "Operation":{
         "JobLevel":"0",
         "VocalScore":{
            "StandardObject":"test.mp3",
            "StandardUrl":""
         },
         "VocalScoreResult":{
            "PitchScore":{
               "SentenceScores":[
                  {
                     "EndTime":"200.811",
                     "Score":"100",
                     "StartTime":"0.572727"
                  },
                  {
                     "EndTime":"210.4",
                     "Score":"100",
                     "StartTime":"200.811"
                  }
```

```
            ],
            "TotalScore":"100"
         },
         "RhythemScore":{
            "SentenceScores":[
               {
                  "EndTime":"200.811",
                  "Score":"100",
                  "StartTime":"0.572727"
               },
               {
                  "EndTime":"210.4",
                  "Score":"100",
                  "StartTime":"200.811"
               }
            ],
            "TotalScore":"100"
         }
      }
   },
   "QueueId":"p6617fd4048fa41faxxxxxxxxxxxx",
   "QueueType":"Speeching",
   "StartTime":"2023-07-18T16:14:37+0800",
   "State":"Success",
   "Tag":"VocalScore"
}]
}
```

任务与工作流
工作流接口
创建工作流

最近更新时间：2023-08-17 17:55:43

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_15
        <Segment_15816659605667>End</Segment_15816659605667>
        <SmartCover_1581665960539>PicProcess_15816659605668</SmartCov
        <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}</SpriteOb
            </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>
```

具体数据描述如下：

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

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.MediaWor	节点依赖关	Contai	是	无

版权所有：腾讯云计算（北京）有限责任公司

	kflow.Topology	系	ner		
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	限制
Start	Request.MediaWorkflow.Topology.Nodes	开始节点	Container	是	只有唯一一个开始节点
Animation_***	Request.MediaWorkflow.Topology.Nodes	动图类型节点	Container	否	节点名称以 Animation 为前缀，可能有多个动图节点
Snapshot_***	Request.MediaWorkflow.Topology.Nodes	截图类型节点	Container	否	节点名称以 Snapshot 为前缀，可能有多个截图节点
SmartCover_***	Request.MediaWorkflow.Topology.Nodes	智能封面节点	Container	否	节点名称以 SmartCover 为前缀，可能有多个智能封面节点
Transcode_***	Request.MediaWorkflow.Topology.Nodes	转码节点	Container	否	节点名称以 Transcode 为前缀，可能有多个转码节点
Concat_***	Request.MediaWorkflow.Topology.Nodes	音视频拼接节点	Container	否	节点名称以 Concat 为前缀，可能有多个音视频拼接节点
VoiceSeparate_***	Request.MediaWorkflow.Topology.Nodes	人声节点	Container	否	节点名称以 VoiceSeparate 为前缀，可能有多个人声分离节点

版权所有：腾讯云计算（北京）有限责任公司

VideoMontage_***	Request.MediaWorkflow.Topology.Nodes	精彩集锦节点	Container	否	节点名称以 VideoMontage 为前缀，可能有多个精彩集锦节点
StreamPackConfig_***	Request.MediaWorkflow.Topology.Nodes	自适应码流节点	Container	否	节点名称以 StreamPackConfig 为前缀，只能有一个 自适应码流节点。只能在 start 节点之后，后面只能是视频子流节点，可以有多个视频子流节点
VideoStream_***	Request.MediaWorkflow.Topology.Nodes	视频子流节点	Container	否	节点名称以 VideoStream 为前缀，可能有多个视频子流节点 ，只能在 StreamPackConfig 节点之后，后面只能是 StreamPack 节点
StreamPack_***	Request.MediaWorkflow.Topology.Nodes	自适应码流打包节点	Container	否	节点名称以 StreamPack 为前缀，只能有一个自适应码流打包节点 ，只能在视频子流节点之后，后面只能是 End 节点
SDRtoHDR_***	Request.MediaWorkflow.Topology.Nodes	SDRtoHDR 节点	Container	否	节点名称以 SDRtoHDR 为前缀，可能有多个 SDRtoHDR 节点
VideoProcess_***	Request.MediaWorkflow.Topology.Nodes	视频处理节点	Container	否	节点名称以 VideoProcess 为前缀，可能有多个视频处理节点
SCF_***	Request.MediaWorkflow.Topology.Nodes	SCF 函数节点	Container	否	节点名称以 SCF 为前缀，可能有多个 SCF 函数节点
SuperResolution_***	Request.MediaWorkflow.Topology.Nodes	超分辨率节点	Container	否	节点名称以 SuperResolution 为前缀，可能有多个超分辨率节点
Segment_***	Request.Medi	音视频转封	Con	否	节点名称以 Segment 为

	aWorkflow.Topology.Nodes	装节点	tainer		前缀，可能有多个音视频转封装节点
PicProcess_***	Request.MediaWorkflow.Topology.Nodes	图片处理节点	Container	否	节点名称以 PicProcess 为前缀，可能有多个图片处理节点
Tts_***	Request.MediaWorkflow.Topology.Nodes	语音合成节点	Container	否	节点名称以 Tts 为前缀，可能有多个语音合成节点
SpeechRecognition_***	Request.MediaWorkflow.Topology.Nodes	语音识别节点	Container	否	节点名称以 SpeechRecognition 为前缀，可能有多个语音识别节点
VideoEnhance_***	Request.MediaWorkflow.Topology.Nodes	画质增强节点	Container	否	节点名称以 VideoEnhance 为前缀，可能有多个语音识别节点
Condition_***	Request.MediaWorkflow.Topology.Nodes	条件判断节点	Container	否	节点名称以 Condition 为前缀，可能有多个条件判断节点
MediaInfo_***	Request.MediaWorkflow.Topology.Nodes	媒体信息获取节点	Container	否	节点名称以 MediaInfo 为前缀，可能有多个媒体信息获取节点
AIRecognition_***	Request.MediaWorkflow.Topology.Nodes	AI识别节点	Container	否	节点名称以 AIRecognition 为前缀
SegmentVideoBody_***	Request.MediaWorkflow.Topology.Nodes	视频人像抠图节点	Container	否	节点名称以 SegmentVideoBody 为前缀
NoiseReduction_***	Request.MediaWorkflow.	音频降噪节点	Container	否	节点名称以 NoiseReduction 为前缀

版权所有：腾讯云计算（北京）有限责任公司

	Topology.Nodes				
FileInfo_***	Request.MediaWorkflow.Topology.Nodes	文件信息节点	Container	否	节点名称以 FileInfo 为前缀
ImageInspect_***	Request.MediaWorkflow.Topology.Nodes	异常图片检测节点	Container	否	节点名称以 ImageInspect 为前缀

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	Animation
Operation	Request.MediaWorkflow.Topology.Nodes.Animation_***	操作规则，详情见 Animation	Container	是	无

Container 类型 Snapshot_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflo	节点类型	String	是	Snapsh

	w. Topology.Nodes.Snapshot_***				ot
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	限制
Type	Request.MediaWorkflow. Topology.Nodes.SmartCover_***	节点类型	String	是	SmartCover
Operation	Request.MediaWorkflow. Topology.Nodes.SmartCover_***	操作规则，详情见 SmartCover	Container	是	无

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	限制
Type	Request.MediaWorkflow. Topology.Nodes.Transcode_***	节点类型	String	是	Transcode
Operation	Request.MediaWorkflow. Topology.Nodes.Transcode_***	操作规则，详情见 Transcode	Container	是	无
Input	Request.MediaWorkflow. Topology.Nodes.Transcode_***	输入配置	Container	否	无

版权所有：腾讯云计算（北京）有限责任公司

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

Container 类型 Concat_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.Concat_***	节点类型	String	是	Concat
Operation	Request.MediaWorkflow.Topology.Nodes.Concat_***	操作规则，详情见 Concat	Container	是	无

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	限制
Type	Request.MediaWorkflow.Topology.Nodes.VoiceSeparate_***	节点类型	String	是	VoiceSeparate
Operation	Request.MediaWorkflow.Topology.Nodes.VoiceSeparate_***	操作规则，详情见 VoiceSeparate	Container	是	无

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	限制
Type	Request.MediaWorkflow.	节点类型	String	是	VideoMontage

	Topology.Nodes.VideoMontage_***				
Operation	Request.MediaWorkflow.Topology.Nodes.VideoMontage_***	操作规则，详情见 VideoMontage	Container	是	无

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	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.StreamPackConfig_***	节点类型	String	是	StreamPackConfig
Operation	Request.MediaWorkflow.Topology.Nodes.StreamPackConfig_***	操作规则，详情见 StreamPackConfig	Container	是	无

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	限制
Type	Request.MediaWorkflow.Topology.Nodes.VideoStream_***	节点类型	String	是	VideoStream
Operation	Request.MediaWorkflow.Topology.Nodes.VideoStream_***	操作规则，详情见 VideoStream	Container	是	无

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	限制

Type	Request.MediaWorkflow.Topology.Nodes.StreamPack_***	节点类型	String	是	StreamPack
Operation	Request.MediaWorkflow.Topology.Nodes.StreamPack_***	操作规则，详情见 StreamPack	Container	是	无

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
Type	Request.MediaWorkflow.Topology.Nodes.SDRtoHDR_***	节点类型	String	是	SDRtoHDR
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	限制
Type	Request.MediaWorkflow.Topology.Nodes.VideoProcess_***	节点类型	String	是	VideoProcess
Operation	Request.MediaWorkflow.Topology.Nodes.VideoProcess_***	操作规则，详情见 VideoProcess	Container	是	无

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	限制

				必选	
Type	Request.MediaWorkflow.Topology.Nodes.SCF_***	节点类型	String	是	SCF
Operation	Request.MediaWorkflow.Topology.Nodes.SCF_***	操作规则，详情见 SCF	Container	是	无

Container 类型 SuperResolution_*** 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.SuperResolution_***	节点类型	String	是	SuperResolution
Operation	Request.MediaWorkflow.Topology.Nodes.SuperResolution_***	操作规则，详情见 SuperResolution	Container	是	无

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	Segment
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	限制

版权所有：腾讯云计算（北京）有限责任公司

（关键字）				否必选	
Type	Request.MediaWorkflow.Topology.Nodes.PicProcess_***	节点类型	String	是	PicProcess
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	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.SpeechRecognition_***	节点类型	String	是	SpeechRecognition
Operation	Request.MediaWorkflow.Topology.Nodes.SpeechRecognition_***	操作规则，详情见 SpeechRecognition	Container	是	无

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	限制
Type	Request.MediaWork	节点类型	Stri	是	VideoE

	flow.Topology.Nodes.VideoEnhance_***		ng		nhance
Operation	Request.MediaWorkflow.Topology.Nodes.VideoEnhance_***	操作规则，详情见 VideoEnhance	Container	是	无

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	Condition
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 FileInfo_*** 的具体数据描述如下：

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.FileInfo_***	节点类型	String	是	FileInfo
Operation	Request.MediaWorkflow.	操作规则，详情见 FileInfo	Container	是	无

版权所有：腾讯云计算（北京）有限责任公司

	Topology.Nodes.FileInfo_***				

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	是否必选	限制
Type	Request.MediaWorkflow.Topology.Nodes.SegmentVideoBody_***	节点类型	String	是	SegmentVideoBody
Operation	Request.MediaWorkflow.Topology.Nodes.SegmentVideoBody_***	操作规则，详情见 SegmentVideoBody	Container	是	无

Container 类型 NoiseReduction_*** 的具体数据描述如下：

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.NoiseReduction_***	节点类型	String	是	NoiseReduction
Operation	Request.MediaWorkflow.Topology.Nodes.NoiseReduction_***	操作规则，详情见 NoiseReduction	Container	是	无

Container 类型 ImageInspect_*** 的具体数据描述如下：

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.ImageInspect_***	节点类型	String	是	ImageInspect
Operation	Request.MediaWorkflow.Topology.Nodes.ImageInspect_***	操作规则，详情见 ImageInspect	Container	是	无

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,StreamPackConfig,SDRtoHDR,VideoProcess,SCF,SuperResolution,Segment,PicProcess,Tts,SpeechRecognition,VideoEnhance,MediaInfo,AIRecognition,VideoEnhance,SegmentVideoBody,NoiseReduction,FileInfo
Snapshot	PicProcess,SCF,FileInfo,End
Animation	PicProcess,SCF,FileInfo,End
SmartCover	PicProcess,SCF,FileInfo,End
Transcode	Snapshot,Animation,SmartCover,Concat,VideoMontage,SDRtoHDR,VideoProcess,SCF,Segment,SpeechRecognition,VideoEnhance,MediaInfo,SegmentVideoBody,NoiseReduction,FileInfo,End
Concat	Transcode,Snapshot,Animation,SmartCover,SDRtoHDR,VideoProcess,SCF,Segment,VideoEnhance,MediaInfo,SegmentVideoBody,NoiseReduction,FileInfo,End
VoiceSeparate	SCF,SpeechRecognition,End
VideoMontage	Snapshot,Animation,SDRtoHDR,VideoProcess,SCF,SuperResolution,VideoEnhance,Segment,MediaInfo,SegmentVideoBody,NoiseReduction,FileInfo,End
StreamPackConfig	VideoStream
VideoStream	StreamPack

StreamPack	End
SDRtoHDR	Transcode,Concat,VideoMontage,VoiceSeparate,SmartCover,Animation,VideoProcess,SCF,SuperResolution,SegmentVideoBody,End
VideoProcess	Transcode,Concat,VideoMontage,VoiceSeparate,SmartCover,Animation,SDRtoHDR,SCF,Segment,SuperResolution,MediaInfo,SegmentVideoBody,End
SCF	SDRtoHDR,Snapshot,Transcode,Concat,VideoMontage,VoiceSeparate,SmartCover,Animation,VideoProcess,End
SuperResolution	VideoMontage,Transcode,SmartCover,Animation,Snapshot,SCF,Segment,SDRtoHDR,VideoProcess,MediaInfo,SegmentVideoBody,End
Segment	VideoEnhance,SegmentVideoBody,End
PicProcess	SCF,PicProcess,FileInfo,End
Tts	SCF,Transcode,FileInfo,End
SpeechRecognition	SCF,Tts,End
VideoEnhance	Transcode,VideoMontage,SmartCover,Animation,Snapshot,SCF,Segment,SpeechRecognition,VoiceSeparate,MediaInfo,FileInfo,End
MediaInfo	Condition,End
FileInfo	Condition,End
Condition	Transcode,SuperResolution,VideoProcess,Animation,Snapshot,VideoMontage,SmartCover,Concat,VideoEnhance,SpeechRecognition,VoiceSeparate,Segment,SDRtoHDR,SCF,End
NoiseReduction	Transcode,Segment,MediaInfo,Concat,SpeechRecognition,FileInfo,SCF,End
AIRecognition	End
SegmentVideoBody	FileInfo,End
ImageInspect	End

说明

- 当 Condition 类型节点前为 FileInfo 类型节点时，后续能连接的节点根据 FileInfo 前置的节点能连

> 接的节点来判断（FileInfo 、MediInfo 类型除外，Condition 类型节点后续不能连接 FileInfo 、MediaInfo 节点）。

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：
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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR***
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_15
      <Segment_15816659605667>End</Segment_15816659605667>
      <SmartCover_1581665960539>PicProcess_15816659605668</SmartCov
      <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}</SpriteOb
          </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>
```

版权所有：腾讯云计算（北京）有限责任公司

查询工作流

最近更新时间：2023-06-20 17:32:16

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-12345678900</BucketId>
    <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
    <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
    <Topology>
      <Dependencies>
        <Start>Snapshot_1581665960536,Transcode_15816659605380</Start>
        <Snapshot_1581665960536>End</Snapshot_1581665960536>
        <Transcode_15816659605380>Segment_15816659605667,SmartCover_15816(
        <Segment_15816659605667>End</Segment_15816659605667>
        <SmartCover_15816659605390>PicProcess_15816659605668</SmartCover_15
        <PicProcess_15816659605668>End</PicProcess_15816659605668>
      </Dependencies>
      <Nodes>
        <Start>
          <Type>Start</Type>
          <Input>
            <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
            <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicProce
            <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-12345678900</Bucket>
            <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
            <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</SpriteObject>
        </Output>
    </Operation>
</Snapshot_1581665960536>
<Transcode_1581665960538>
    <Type>Transcode</Type>
    <Operation>
        <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>bcd/${RunId}/trans.{Ext}</Object>
        </Output>
    </Operation>
</Transcode_1581665960538>
<Segment_15816659605667>
    <Type>Segment</Type>
    <Operation>
        <Segment>
            <Format>mkv</Format>
            <Duration>20</Duration>
        </Segment>
        <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>test-trans${Number}.{Ext}</Object>
        </Output>
    </Operation>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        </Segment_158166596056 67>
        <SmartCover_1581665960539>
          <Type>SmartCover</Type>
          <Operation>
            <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</Bucket>
              <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
            </Output>
          </Operation>
        </SmartCover_1581665960539>
        <PicProcess_158166596056 68>
          <Type>PicProcess</Type>
          <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</Bucket>
              <Object>bcd/${RunId}/pic.{Ext}</Object>
            </Output>
          </Operation>
        </PicProcess_158166596056 68>
      </Nodes>
    </Topology>
  </MediaWorkflowList>
  <MediaWorkflowList>
    <Name>workflow-2</Name>
    <State>Active</State>
    <WorkflowId>w93aa43ba105347169fa093ed857b2a900</WorkflowId>
    <BucketId>test-12345678900</BucketId>
    <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
    <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
    <Topology>
      <Dependencies>
        <Start>StreamPackConfig_15816659605320</Start>
        <StreamPackConfig_15816659605320>VideoStream_1581665960536,VideoStre
        <VideoStream_15816659605360>StreamPack_15816659605380</VideoStream_
        <VideoStream_15816659605370>StreamPack_15816659605380</VideoStream_
        <StreamPack_15816659605380>End</StreamPack_15816659605380>
      </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_15816659605320>
      <Type>StreamPackConfig</Type>
      <Operation>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-12345678900</Bucket>
          <Object>${InputPath}/${InputName}._${RunId}.${ext}</Object>
        </Output>
        <StreamPackConfig>
          <PackType>HLS</PackType>
          <IgnoreFailedStream>true</IgnoreFailedStream>
        </StreamPackConfig>
      </Operation>
    </StreamPackConfig_15816659605320>
    <VideoStream_15816659605360>
      <Type>VideoStream</Type>
      <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <Output>
          <Region>ap-chongqing</Region>
          <Bucket>test-12345678900</Bucket>
          <Object>${RunId}_Substream_1/video.m3u8</Object>
        </Output>
      </Operation>
    </VideoStream_15816659605360>
    <VideoStream_15816659605370>
```

```
            <Type>VideoStream</Type>
            <Operation>
              <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId>
              <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-12345678900</Bucket>
                <Object>${RunId}_Substream_2/video.m3u8</Object>
              </Output>
            </Operation>
          </VideoStream_15816659605377>
          <StreamPack_15816659605388>
            <Type>StreamPack</Type>
            <Operation>
              <StreamPackInfo>
                <VideoStreamConfig>
                  <VideoStreamName>VideoStream_15816659605366</VideoStreamN
                  <BandWidth>2000000000</BandWidth>
                </VideoStreamConfig>
                <VideoStreamConfig>
                  <VideoStreamName>VideoStream_15816659605377</VideoStreamN
                  <BandWidth>2000000000</BandWidth>
                </VideoStreamConfig>
              </StreamPackInfo>
            </Operation>
          </StreamPack_15816659605388>
        </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

版权所有：腾讯云计算（北京）有限责任公司

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-
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-12345678900</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_15816(
                <Segment_15816659605667>End</Segment_15816659605667>
                <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
                <PicProcess_15816659605668>End</PicProcess_15816659605668>
            </Dependencies>
            <Nodes>
                <Start>
                    <Type>Start</Type>
                    <Input>
                        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
                        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicProce
```

```
            <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_15816659605536>
        <Type>Snapshot</Type>
        <Operation>
            <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-12345678900</Bucket>
                <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
                <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</SpriteObject>
            </Output>
        </Operation>
    </Snapshot_15816659605536>
    <Transcode_15816659605538>
        <Type>Transcode</Type>
        <Operation>
            <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
            <Output>
                <Region>ap-chongqing</Region>
                <Bucket>test-12345678900</Bucket>
                <Object>bcd/${RunId}/trans.{Ext}</Object>
            </Output>
        </Operation>
    </Transcode_15816659605538>
    <Segment_15816659605667>
        <Type>Segment</Type>
        <Operation>
            <Segment>
```

```
            <Format>mkv</Format>
            <Duration>20</Duration>
          </Segment>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>test-trans${Number}.{Ext}</Object>
          </Output>
        </Operation>
      </Segment_158166596056667>
      <SmartCover_15816659605399>
        <Type>SmartCover</Type>
        <Operation>
          <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
          </Output>
        </Operation>
      </SmartCover_15816659605399>
      <PicProcess_158166596056668>
        <Type>PicProcess</Type>
        <Operation>
          <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>bcd/${RunId}/pic.{Ext}</Object>
          </Output>
        </Operation>
      </PicProcess_158166596056668>
    </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=AKIDZfbOAo7cIlgPvF9cXFrJD0**********&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, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=

<Response>
  <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=</RequestId>
  <TotalCount>2</TotalCount>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <MediaWorkflowList>
    <Name>workflow-1</Name>
    <State>Active</State>
    <WorkflowId>wc666d0b9f9dd47ae9137a096252d49f7</WorkflowId>
    <BucketId>test-12345678900</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_158166
        <Segment_15816659605667>End</Segment_15816659605667>
        <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
        <PicProcess_15816659605668>End</PicProcess_15816659605668>
      </Dependencies>
      <Nodes>
        <Start>
          <Type>Start</Type>
          <Input>
            <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
            <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicProce
            <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_15816659605366>
        <Type>Snapshot</Type>
        <Operation>
          <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>abc/${RunId}/snapshot-${number}.${Ext}</Object>
            <SpriteObject>abc/${RunId}/sprite-${number}.${Ext}</SpriteObject>
          </Output>
        </Operation>
      </Snapshot_15816659605366>
      <Transcode_15816659605388>
        <Type>Transcode</Type>
        <Operation>
          <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>bcd/${RunId}/trans.{Ext}</Object>
          </Output>
        </Operation>
      </Transcode_15816659605388>
      <Segment_15816659605667>
        <Type>Segment</Type>
        <Operation>
          <Segment>
            <Format>mkv</Format>
            <Duration>20</Duration>
          </Segment>
          <Output>
```

```
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>test-trans${Number}.{Ext}</Object>
          </Output>
        </Operation>
      </Segment_1581665960566 7>
      <SmartCover_1581665960539>
        <Type>SmartCover</Type>
        <Operation>
          <TemplateId>t16e81a29fe48c4e23acefc247a7792b63</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>abc/${RunId}/cover-${Number}.{Ext}</Object>
          </Output>
        </Operation>
      </SmartCover_1581665960539>
      <PicProcess_1581665960566 8>
        <Type>PicProcess</Type>
        <Operation>
          <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>bcd/${RunId}/pic.{Ext}</Object>
          </Output>
        </Operation>
      </PicProcess_1581665960566 8>
    </Nodes>
  </Topology>
</MediaWorkflowList>
<MediaWorkflowList>
  <Name>workflow-2</Name>
  <State>Active</State>
  <WorkflowId>w93aa43ba105347169fa093ed857b2a90</WorkflowId>
  <BucketId>test-12345678900</BucketId>
  <CreateTime>2022-06-29T14:37:44+0800</CreateTime>
  <UpdateTime>2022-06-29T14:37:44+0800</UpdateTime>
  <Topology>
    <Dependencies>
      <Start>StreamPackConfig_15816659605 32</Start>
      <StreamPackConfig_15816659605 32>VideoStream_1581665960536,VideoStre
      <VideoStream_15816659605 36>StreamPack_15816659605 38</VideoStream_
      <VideoStream_15816659605 37>StreamPack_15816659605 38</VideoStream_
      <StreamPack_15816659605 38>End</StreamPack_15816659605 38>
```

```
    </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_15816659605320>
        <Type>StreamPackConfig</Type>
        <Operation>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>${InputPath}/${InputName}._${RunId}.${ext}</Object>
          </Output>
          <StreamPackConfig>
            <PackType>HLS</PackType>
            <IgnoreFailedStream>true</IgnoreFailedStream>
          </StreamPackConfig>
        </Operation>
      </StreamPackConfig_15816659605320>
      <VideoStream_15816659605360>
        <Type>VideoStream</Type>
        <Operation>
          <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
```

```
            <Object>${RunId}_Substream_1/video.m3u8</Object>
          </Output>
        </Operation>
      </VideoStream_1581665960536>
      <VideoStream_1581665960537>
        <Type>VideoStream</Type>
        <Operation>
          <TemplateId>t1460606bgfdg2148c4ab182f55163ba7bj</TemplateId>
          <Output>
            <Region>ap-chongqing</Region>
            <Bucket>test-12345678900</Bucket>
            <Object>${RunId}_Substream_2/video.m3u8</Object>
          </Output>
        </Operation>
      </VideoStream_1581665960537>
      <StreamPack_1581665960538>
        <Type>StreamPack</Type>
        <Operation>
          <StreamPackInfo>
            <VideoStreamConfig>
              <VideoStreamName>VideoStream_1581665960536</VideoStreamN
              <BandWidth>2000000000</BandWidth>
            </VideoStreamConfig>
            <VideoStreamConfig>
              <VideoStreamName>VideoStream_1581665960537</VideoStreamN
              <BandWidth>2000000000</BandWidth>
            </VideoStreamConfig>
          </StreamPackInfo>
        </Operation>
      </StreamPack_1581665960538>
    </Nodes>
  </Topology>
 </MediaWorkflowList>
</Response>
```

版权所有：腾讯云计算（北京）有限责任公司

删除工作流

最近更新时间：2023-06-20 17:32:16

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
<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-
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>
```

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

更新工作流

最近更新时间：2023-06-20 17:32:17

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    <Snapshot_1581665960536>End</Snapshot_1581665960536>
    <Transcode_1581665960538>Segment_15816659605667,SmartCover_15816(
    <Segment_15816659605667>End</Segment_15816659605667>
    <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
    <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}</SpriteObject>
        </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_158166
        <Segment_15816659605667>End</Segment_15816659605667>
        <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
        <PicProcess_15816659605668>End</PicProcess_15816659605668>
      </Dependencies>
      <Nodes>
        <Start>
          <Type>Start</Type>
          <Input>
            <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
            <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicProce
```

```
        <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}</SpriteObject>
        </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
Respon	无	同创建工作流接口的 Response	Conta	是

se			iner	

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-
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_158166
            <Segment_15816659605667>End</Segment_15816659605667>
            <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
            <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}</SpriteObject>
        </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_158166
    <Segment_15816659605667>End</Segment_15816659605667>
    <SmartCover_1581665960539>PicProcess_15816659605668</SmartCover_15
    <PicProcess_15816659605668>End</PicProcess_15816659605668>
  </Dependencies>
  <Nodes>
    <Start>
      <Type>Start</Type>
      <Input>
        <QueueId>p09d709939fef48a0a5c247ef39d90cec</QueueId>
        <PicProcessQueueId>p2911917386e148639319e13c285cc774</PicProce
        <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}</SpriteObject>
        </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-
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-
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
```

版权所有：腾讯云计算（北京）有限责任公司

测试工作流

最近更新时间：2023-06-21 10:22:02

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
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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>
```

版权所有：腾讯云计算（北京）有限责任公司

工作流回调

最近更新时间：2023-05-08 19:40:20

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <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.Workflo	被处理文件所属的存储桶	String

	wExecution		
Object	Response.WorkflowExecution	该实例处理的文件名	String
CreateTime	Response.WorkflowExecution	实例的创建时间	String
CosHeaders	Response.WorkflowExecution	该 Object 上传时设定的自定义 Header 列表，没有设置时无此内容	Container 数组
Tasks	Response.WorkflowExecution	任务信息列表	Container 数组

Container 节点 CosHeaders 的内容：

节点名称（关键字）	父节点	描述	类型
Key	Response.WorkflowExecution.CosHeaders	自定义 Header 的名称	String
Value	Response.WorkflowExecution.CosHeaders	自定义 Header 的值	String

Container 节点 Tasks 的内容：

节点名称（关键字）	父节点	描述	类型
Type	Response.WorkflowExecution.Tasks	任务类型	String
Createtime	Response.WorkflowExecution.Tasks	任务的创建时间	String
StartTime	Response.WorkflowExecution.Tasks	任务开始时间	String
EndTime	Response.WorkflowExecution.Tasks	任务的完成时间	String
State	Response.WorkflowExecution.Tasks	任务的状态，值为 Success、Failed 等	String
JobId	Response.WorkflowExecution.Tasks	任务的 ID	String
Name	Response.WorkflowExecut	任务在工作流中的节点名称	String

版权所有：腾讯云计算（北京）有限责任公司

	ion.Tasks		
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
ResultInfo	Response.WorkflowExecution.Tasks	任务结果详情	Container

Container 节点 ResultInfo 的内容：

节点名称（关键字）	父节点	描述	类型
SpriteObjectInfo	Response.WorkflowExecution.Tasks.ResultInfo	雪碧图对象详情	Container 数组
ObjectInfo	Response.WorkflowExecution.Tasks.ResultInfo	输出对象详情	Container 数组

Container 节点 SpriteObjectInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.WorkflowExecution.Tasks.ResultInfo.SpriteObjectInfo	对象名	String
ObjectUrl	Response.WorkflowExecution.Tasks.ResultInfo.SpriteObjectInfo	对象 Url	String

Container 节点 ObjectInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	对象名	String
ObjectUrl	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	对象Url	String
InputObjectName	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	输入对象名	String
Code	Response.WorkflowExecution.	InputObjectName对应错误码，	Str

版权所有：腾讯云计算（北京）有限责任公司

	Tasks.ResultInfo.ObjectInfo	0为处理成功，非0处理失败	ing
Message	Response.WorkflowExecution. Tasks.ResultInfo.ObjectInfo	InputObjectName对应错误信息，处理失败时有效	Str ing

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": [{
                "Key": "Content-Type",
                "Value": "video/mp4"
            },
            {
                "Key": "x-cos-request-id",
                "Value": "NjJiZDYwYTFfNjUzYTYyNjRfZjEwZl8xMmZhYzY5"
            }
        ],
        "Tasks": [{
                "Type": "Snapshot",
                "CreateTime": "2022-08-09 14:54:40+0800",
                "EndTime": "2022-08-09 14:54:42+0800",
                "State": "Success",
                "JobId": "j23c11e1e17b011edaab4ab15ec33d076",
                "Name": "Snapshot_1581665960536",
                "TemplateId": "t07740e32081b44ad7a0aea03adcffd54a",
                "TemplateName": "snapshot_1280*720_3"
            },
            {
                "Type": "Transcode",
                "CreateTime": "2022-08-09 14:54:18+0800",
                "EndTime": "2022-08-09 14:54:39+0800",
                "State": "Success",
                "JobId": "j168668b217b011ed8efb27bb229e2d11",
                "Name": "Transcode_1581665960537",
                "TemplateId": "t01e57db1c2d154d2fb57aa5de9313a897",
```

```
            "TemplateName": "MP4-265-SD"
        }
    ]
  }
}
```

工作流实例
获取工作流实例详情

最近更新时间：2023-07-25 09:06:34

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。
- 为工作流实例id

请求头

此接口仅使用公共请求头部，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_158166
      </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</PicProce
          </Input>
        </Start>
        <Snapshot_1581665960536>
          <Type>Snapshot</Type>
          <Operation>
            <TemplateId>t07740e32081b44ad7a0aea03adcffd54a</TemplateId>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</Bucket>
              <Object>snapshot-${number}.jpg</Object>
            </Output>
          </Operation>
        </Snapshot_1581665960536>
```


版权所有：腾讯云计算（北京）有限责任公司

```
            <Transcode_1581665960537>
              <Type>Transcode</Type>
              <Operation>
                <TemplateId>t01e57db1c2d154d2fb57aa5de9313a897</TemplateId>
                <Output>
                  <Region>ap-chongqing</Region>
                  <Bucket>test-12345678900</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
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，当该节点为判断节点时，PartialSuccess 表示部分输入文件通过判断	String

版权所有：腾讯云计算（北京）有限责任公司

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name	String
ResultInfo	Response.WorkflowExecution.Tasks	任务结果详情	Container
JudgementInfo	Response.WorkflowExecution.Tasks	判断节点结果详情，当节点为判断节点时返回	Container
FileInfo	Response.WorkflowExecution.Tasks	判断节点结果详情，当节点为判断节点时返回	Container 数组

Container 节点 ResultInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectCount	Response.WorkflowExecution.Tasks.ResultInfo	任务生成对象文件个数	String
SpriteObjectCount	Response.WorkflowExecution.Tasks.ResultInfo	仅在截图生成雪碧图有效，表示生成雪碧图个数	String
ObjectInfo	Response.WorkflowExecution.Tasks.ResultInfo	任务生成对象详情	Container 数组
SpriteObjectInfo	Response.WorkflowExecution.Tasks.ResultInfo	任务生成雪碧图对象详情	Container 数组

Container 节点 SpriteObjectInfo 的内容：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.WorkflowExecution.Tasks.ResultInfo.SpriteObjectInfo	对象名	String
ObjectUrl	Response.WorkflowExecution.Tasks.ResultInfo.SpriteObjectInfo	对象Url	String

Container 节点 ObjectInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectName	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	对象名	String
ObjectUrl	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	对象Url	String
InputObjectName	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	输入对象名	String
Code	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	InputObjectName对应错误码，0为处理成功，非0处理失败	String
Message	Response.WorkflowExecution.Tasks.ResultInfo.ObjectInfo	InputObjectName对应错误信息，处理失败时有效	String

Container 节点 JudgementInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectCount	Response.WorkflowExecution.Tasks.JudgementInfo	判断节点输入对象文件个数	Int
JudgementResult	Response.WorkflowExecution.Tasks.JudgementInfo	判断节点输入对象具体判断结果	Container数组

Container 节点 JudgementResult 的内容：

节点名称（关键字）	父节点	描述	类型
ObjectNa	Response.WorkflowExecution.	对象名	String

版权所有：腾讯云计算（北京）有限责任公司

me	Tasks.JudgementInfo.JudgementResult		
ObjectUrl	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult	对象链接	String
State	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult	判断结果状态，Success 表示通过条件判断，Failed 表示未通过条件判断	String
InputObjectInfo	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult	输入文件的判断参数	Container

Container 节点 InputObjectInfo 的内容：

节点名称（关键字）	父节点	描述	类型
Width	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	视频宽	String
Height	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	视频高	String
Dar	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	视频宽高比	String
Duration	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	音视频时长	String
Size	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	文件大小	String
ImageWidth	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	图片宽	String
ImageHeight	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	图片高	String
ImageDar	Response.WorkflowExecution.Tasks.JudgementInfo.JudgementResult.InputObjectInfo	图片宽高比	String

版权所有：腾讯云计算（北京）有限责任公司

Container 节点 FileInfo 的内容：

节点名称（关键字）	父节点	描述	类型
BasicInfo	Response.WorkflowExecution.Tasks.FileInfo	文件基础信息	Container
MediaInfo	Response.WorkflowExecution.Tasks.FileInfo	音视频信息	Container
ImageInfo	Response.WorkflowExecution.Tasks.FileInfo	图片信息	Container

Container 节点 BasicInfo 的内容：

节点名称（关键字）	父节点	描述	类型
ContentType	Response.WorkflowExecution.Tasks.FileInfo.BasicInfo	文件类型	String
Size	Response.WorkflowExecution.Tasks.FileInfo.BasicInfo	文件大小	String
ETag	Response.WorkflowExecution.Tasks.FileInfo.BasicInfo	文件Etag	String
LastModified	Response.WorkflowExecution.Tasks.FileInfo.BasicInfo	文件上修改时间	String
Object	Response.WorkflowExecution.Tasks.FileInfo.BasicInfo	文件名	String

Container 节点 MediaInfo 的内容：

节点名称（关键字）	父节点	描述	类型
Video	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo	视频信息	Container
Audio	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo	音频信息	Container
Format	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo	格式信息	Container

Container 节点 Video 的内容：

版权所有：腾讯云计算（北京）有限责任公司

节点名称（关键字）	父节点	描述	类型
Height	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo.Video	视频高	String
Width	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo.Video	视频宽	String
Dar	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo.Video	视频宽高比	String
Duration	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo.Video	视频时长，单位秒	String

Container 节点 Audio 的内容：

节点名称（关键字）	父节点	描述	类型
Duration	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo.Audio	音频时长，单位秒	String

Container 节点 Formt 的内容：

节点名称（关键字）	父节点	描述	类型
Duration	Response.WorkflowExecution.Tasks.FileInfo.MediaInfo.Format	时长，单位秒	String

Container 节点 ImageInfo 的内容：

节点名称（关键字）	父节点	描述	类型
Format	Response.WorkflowExecution.Tasks.FileInfo.ImageInfo	图片高	String
Height	Response.WorkflowExecution.Tasks.FileInfo.ImageInfo	图片宽	String
Width	Response.WorkflowExecution.Tasks.FileInfo.ImageInfo	图片宽	String
Md5	Response.WorkflowExecution.Tasks.FileInfo.ImageInfo	图片Md5	String

版权所有：腾讯云计算（北京）有限责任公司

FrameCount	Response.WorkflowExecution.Tasks.FileInfo.ImageInfo	图片帧数，静态图为1，动图为对应的帧数	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

```
GET /workflowexecution/i166ee19017b011eda8a5525400c540df HTTP/1.1
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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_158166
      </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</PicProce
          </Input>
        </Start>
        <Snapshot_1581665960536>
          <Type>Snapshot</Type>
          <Operation>
            <TemplateId>t07740e32081b44ad7a0aea03adcffd54a</TemplateId>
            <Output>
              <Region>ap-chongqing</Region>
              <Bucket>test-12345678900</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-12345678900</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_15816659605 37</Name>
        <State>Success</State>
        <StartTime>2022-08-09T14:54:18+08:00</StartTime>
        <EndTime>2022-08-09T14:54:39+08:00</EndTime>
        <Code>Success</Code>
        <Message>success</Message>
    </Tasks>
</WorkflowExecution>
```

获取工作流实例详情列表

最近更新时间：2023-06-20 17:32:17

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	否
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>NjJmMjEyMmFfZWM0YTYyNjRfN2U2NV8yNml0</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-
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
取消任务

最近更新时间：2023-06-20 17:32:23

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-
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=
```

版权所有：腾讯云计算（北京）有限责任公司

提交多任务处理

最近更新时间：2023-06-21 10:35:31

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
        <Object>output/animation.gif</Object>
      </Output>
      <UserData>This is my Animation job.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <Operation>
      <Tag>Transcode</Tag>
      <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
      <Output>
        <Region>ap-chongqing</Region>
        <Bucket>test-1234567890</Bucket>
        <Object>output/transcode.mp4</Object>
      </Output>
      <UserData>This is my Trancode job.</UserData>
      <JobLevel>0</JobLevel>
    </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-1234567890</Bucket>
        <Object>output/smartcover-${Number}.jpg</Object>
      </Output>
      <UserData>This is my SmartCover job.</UserData>
      <JobLevel>0</JobLevel>
    </Operation>
    <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
Input	Request	待操作的文件信息	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	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	文件路径	String	是

对于不同的任务类型，Operation 的内容不同，请参照以下链接：

媒体处理：

- 超分辨率
- 画质增强
- 获取媒体信息
- 极速高清转码
- 精彩集锦
- 视频标签
- 视频截图
- 视频明水印
- 视频增强
- 视频质量评分
- 视频转动图
- 数字水印
- 提取数字水印
- 音视频流分离
- 音视频拼接
- 音视频转封装
- 音视频转码
- 智能封面
- SDR to HDR

图片处理：

- 图片处理

智能语音：

- 人声分离

- 听歌识曲
- 音频降噪
- 语音合成
- 语音识别

AI 内容识别

- 翻译
- 分词
- 视频目标检测
- 视频人像抠图
- 视频拆条

响应

响应头

此接口仅返回公共响应头部，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>
      <JobLevel>0</JobLevel>
    </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>
      <JobLevel>0</JobLevel>
    </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-1234567890</Bucket>
          <Object>output/smartcover-${Number}.jpg</Object>
        </Output>
        <UserData>This is my SmartCover job.</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 数组

对于不同的任务类型，JobsDetail 的内容不同，请参照以下链接：

媒体处理：

- 超分辨率
- 画质增强
- 获取媒体信息
- 极速高清转码
- 精彩集锦
- 视频标签
- 视频截图

版权所有：腾讯云计算（北京）有限责任公司

- 视频明水印
- 视频增强
- 视频质量评分
- 视频转动图
- 数字水印
- 提取数字水印
- 音视频流分离
- 音视频拼接
- 音视频转封装
- 音视频转码
- 智能封面
- SDR to HDR

图片处理：

- 图片处理

智能语音：

- 人声分离
- 听歌识曲
- 音频降噪
- 语音合成
- 语音识别

AI 内容识别

- 翻译
- 分词
- 视频目标检测
- 视频人像抠图
- 视频拆条

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-s
Host: test-1234567890.ci.ap-chongqing.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-1234567890</Bucket>
      <Object>output/animation.gif</Object>
    </Output>
    <UserData>This is my Animation job.</UserData>
    <JobLevel>0</JobLevel>
  </Operation>
  <Operation>
    <Tag>Transcode</Tag>
    <TemplateId>t1995d523e42df4c5e858f244b4174360c</TemplateId>
    <Output>
      <Region>ap-chongqing</Region>
      <Bucket>test-1234567890</Bucket>
      <Object>output/transcode.mp4</Object>
    </Output>
    <UserData>This is my Trancode job.</UserData>
    <JobLevel>0</JobLevel>
  </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-1234567890</Bucket>
      <Object>output/smartcover-${Number}.jpg</Object>
    </Output>
```

```
    <UserData>This is my SmartCover job.</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: NjMxMDJhYTNfMThhYTk0MGFfYmU1OV8zZjc=

<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>
      <JobLevel>0</JobLevel>
```

```
    </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>
      <JobLevel>0</JobLevel>
    </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-1234567899</Bucket>
            <Object>output/smartcover-${Number}.jpg</Object>
        </Output>
        <UserData>This is my SmartCover job.</UserData>
        <JobLevel>0</JobLevel>
    </Operation>
  </JobsDetail>
</Response>
```

查询指定任务

最近更新时间：2023-06-20 17:32:24

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。
- jobId 为提交任务时返回的任务Id，支持传多个，以逗号分隔，最多传10个。

请求头

此接口仅使用公共请求头部，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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </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：

媒体处理：

- 超分辨率
- 画质增强
- 获取媒体信息
- 极速高清转码
- 精彩集锦
- 视频标签
- 视频截图
- 视频明水印
- 视频增强
- 视频质量评分
- 视频转动图
- 数字水印
- 提取数字水印
- 音视频流分离
- 音视频拼接
- 音视频转封装
- 音视频转码

- 智能封面
- SDR to HDR

图片处理：

- 图片处理

智能语音：

- 人声分离
- 听歌识曲
- 音频降噪
- 语音合成
- 语音识别

AI 内容识别

- 翻译
- 分词
- 视频目标检测
- 视频人像抠图
- 视频拆条

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：查询一个音视频转码任务，该任务正在执行中

```
GET /jobs/j8d121820f5e411ec926ef19d53ba9c6f HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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=
```

版权所有：腾讯云计算（北京）有限责任公司

```
<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</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
      <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
      <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

请求2：查询一个音视频转码任务，该任务执行成功

```
GET /jobs/j9c0a4726f6ac11ec96aaa9b64ab18d00 HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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</WatermarkTe
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
            <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</CodecL
              <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>
```

获取符合条件的任务列表

最近更新时间：2023-06-20 17:32:24

功能描述

获取符合条件的任务列表。

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	否
queueType	无	拉取队列类型下的任务，和 queueId 不同时生效，同时存在时 queueId 优先	String	否
tag	无	任务的 Tag	String	是
workflowId	无	触发该任务的工作流ID	String	否
inventoryTrigg	无	触发该任务的存量触发任务ID	Strin	否

erJobId			g	
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

queueType 支持以下几种类型：

队列类型	
媒体处理	Transcoding
智能语音	Speeching
图片处理	PicProcess
AI 内容识别	AIProcess
倍速转码	SpeedTranscoding

tag 支持以下几种类型：

任务类型		tag
媒体处理	音视频转码	Transcode

版权所有：腾讯云计算（北京）有限责任公司

	视频转动图	Animation
	视频截图	Snapshot
	智能封面	SmartCover
	音视频拼接	Concat
	精彩集锦	VideoMontage
	SDR to HDR	SDRtoHDR
	视频增强	VideoProcess
	超分辨率	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
	视频明水印	Watermark
图片处理	图片处理	PicProcess
智能语音	语音识别	SpeechRecognition
	音频降噪	NoiseReduction
	人声分离	VoiceSeparate
	语音合成	Tts
	听歌识曲	SoundHound
AI 内容识别	翻译	Translation
	分词	WordsGeneralize
	视频目标检测	VideoTargetRec

版权所有：腾讯云计算（北京）有限责任公司

	视频人像抠图	SegmentVideoBody
	视频拆条	SplitVideoParts

请求头

此接口仅使用公共请求头部，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：
媒体处理：

- 超分辨率
- 画质增强
- 获取媒体信息
- 极速高清转码
- 精彩集锦
- 视频标签
- 视频截图
- 视频明水印
- 视频增强
- 视频质量评分
- 视频转动图
- 数字水印
- 提取数字水印
- 音视频流分离
- 音视频拼接
- 音视频转封装
- 音视频转码
- 智能封面
- SDR to HDR

图片处理

- 图片处理

智能语音：

- 人声分离
- 听歌识曲
- 音频降噪
- 语音合成
- 语音识别

AI 内容识别

- 翻译
- 分词
- 视频目标检测
- 视频人像抠图
- 视频拆条

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-s
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</WatermarkTe
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
            <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
            <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-1234567890</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-1234567890</BucketId>
        <Object>input/demo.mp4</Object>
        <Region>ap-chongqing</Region>
    </Input>
    <Operation>
        <TemplateId>t1460606b9752148c4ab182f55163ba7cd</TemplateId>
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe22</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe23</WatermarkTe
        <WatermarkTemplateId>t1318c5f428d474afba1797f84091cbe24</WatermarkTe
        <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-1234567890</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</CodecL
               <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-1234567890</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>
```

模板接口
删除模板

最近更新时间：2023-06-20 17:32:27

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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR98****-s
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: NjJjYmU0YmRfOTBmYTUwNjRfODhjXzE=

<Response>
    <RequestId>NjJjYmU0YmRfOTBmYTUwNjRfODhjXzE=</RequestId>
    <TemplateId>t143ae6e040af6431aa772c9ec3f0a3f36</TemplateId>
<Response>
```

查询模板

最近更新时间：2023-06-20 15:23:41

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

版权所有：腾讯云计算（北京）有限责任公司

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
智能封面	SmartCover
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
语音合成	Tts
音视频转码Pro	TranscodePro
极速高清	HighSpeedHd
图片处理	PicProcess
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>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>
```

版权所有：腾讯云计算（北京）有限责任公司

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
TotalCount	Response	模板总数	Int
PageNumber	Response	当前页数，同请求中的 pageNumber	Int
PageSize	Response	每页个数，同请求中的 pageSize	Int
TemplateList	Response	模板数组	Container 数组

对于不同的模板类型，TemplateList 内容同创建模板接口的 Response，请参照以下链接：

- 音视频转码
- 视频转动图
- 视频截图
- 智能封面
- 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,t1a30a323f55434a29b25bf
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-
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-12345678900</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-12345678900</BucketId>
      <Category>Custom</Category>
      <ConcatTemplate>
        <ConcatFragment>
          <Mode>Start</Mode>
          <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/input/ad.mp4<
        </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-
Host: test-1234567890.ci.ap-beijing.myqcloud.com
Content-Length: 0
Content-Type: application/xml
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 100
Connection: keep-alive
Date: Thu, 14 Jul 2022 12:37:29 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhf****

<Response>
   <RequestId>NTk0MjdmODlfMjQ4OGY3XzYzYzhf****</RequestId>
   <TotalCount>2</TotalCount>
   <PageNumber>1</PageNumber>
   <PageSize>10</PageSize>
   <TemplateList>
      <TemplateId>t1847cd4ca57f543e89f551dbe68169eb9</TemplateId>
      <Name>TemplateName</Name>
      <Tag>Transcode</Tag>
      <TransTpl>
         <Container>
            <Format>mp4</Format>
         </Container>
         <Video>
            <Codec>H.264</Codec>
            <Profile>high</Profile>
            <Bitrate>1000</Bitrate>
```

```
            <Width>1280</Width>
            <Fps>30</Fps>
            <Preset>medium</Preset>
          </Video>
          <Audio>
            <Codec>aac</Codec>
            <Samplerate>44100</Samplerate>
            <Bitrate>128</Bitrate>
            <Channels>4</Channels>
          </Audio>
          <TransConfig>
            <AdjDarMethod>scale</AdjDarMethod>
            <IsCheckReso>false</IsCheckReso>
            <ResoAdjMethod>1</ResoAdjMethod>
          </TransConfig>
          <TimeInterval>
            <Start>0</Start>
            <Duration>60</Duration>
          </TimeInterval>
        </TransTpl>
        <CreateTime>2020-08-05T11:35:24+0800</CreateTime>
        <UpdateTime>2020-08-31T16:15:20+0800</UpdateTime>
      </TemplateList>
      <TemplateList>
        <TemplateId>t140325e8918ac423da53b7d78dbbab564</TemplateId>
        <Name>template_name3544697</Name>
        <Tag>Concat</Tag>
        <BucketId>test-12345678900</BucketId>
        <Category>Custom</Category>
        <ConcatTemplate>
          <ConcatFragment>
            <Mode>Start</Mode>
            <Url>https://test-1234567890.cos.ap-chongqing.myqcloud.com/input/ad.mp4<
          </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>
```

批量任务接口
触发任务（独立节点）

最近更新时间：2023-08-11 18:10:33

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

<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>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>
      <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-12345678900</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	待操作的文件信息	Container	是
Operation	Request	操作规则	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Manifest	Request.Input	COS 清单列表文件名，COS 清单生成的 manifest.json 文件	String	否
UrlFile	Request.Input	URL 文件名，文件中每行的 URL 为一个 COS 中对象的访问地址	String	否
Prefix	Request.Input	Object 前缀	String	否
Object	Request.Input	文件路径	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	否
QueueType	Request.Operation	队列类型，限制为 SpeedTranscoding ，当 Tag 为 Transcode 且 QueueId 为空时生效，表示为开启倍速转码	String	否
UserData	Request.Operation	用户信息	String	否

版权所有：腾讯云计算（北京）有限责任公司

JobLevel	Request.Operation	任务优先级，级别限制：0 、1 、2。级别越大任务优先级越高，默认为0	String	否
CallBackFormat	Request.Operation	任务回调格式，JSON 或 XML，默认 XML，优先级高于队列的回调格式	String	否
CallBackType	Request.Operation	任务回调类型，Url 或 TDMQ，默认 Url，优先级高于队列的回调类型	String	否
CallBack	Request.Operation	任务回调地址，优先级高于队列的回调地址。设置为 no 时，表示队列的回调地址不产生回调	String	否
CallBackMqConfig	Request.Operation	任务回调 TDMQ 配置，当 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
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
异常图片检测	ImageInspect

Container 类型 TimeInterval 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	默认值	限制
Start	Request.Operation.TimeInterval	开始时间	String	否	0	• 扫描对象的上传时间 • %Y-%m-%dT%H:%m:%S%z

End	Request.Operation.TimeInterval	结束时间	String	否	当前时间	• 扫描对象的上传时间 • %Y-%m-%dT%H:%m:%S%z

Container 类型 Output 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选	限制
Region	Request.Operation.Output	存储桶的地域	String	是	无
Bucket	Request.Operation.Output	存储桶的名称	String	是	无
Object	Request.Operation.Output	结果文件名称	String	否	• 当媒体任务类型不为 Snapshot 、VoiceSeparate 、StreamExtract 时，该参数必选 • 当媒体任务类型为 SmartCover，且 Request.Operation.JobParam.SmartCover.Count 大于1时，必须包含 ${Number} 参数 • 当媒体任务类型为 Snapshot，且 Snapshot.Count 大于1时，必须包含 ${Number} 参数
AuObject	Request.Operation.Output	人声结果文件名	String	否	当媒体任务类型为 VideoSeParete 时，该参数有效，且不能和 Object 同时为空
SpriteObject	Request.Operation.Output	雪碧图名	String	否	• 当媒体任务类型为 Snapshot 时，该参数有效，且不能和 Object 同时为空

					• 当 Snapshot.Count 大于1时，必须包含 ${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
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	是
Object	Request.Operation.Output.StreamExtract	结果输出文件名	String	是

Container 类型 JobParam 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
TemplateId	Request.Operation.JobParam	模板 ID	String	否
TranscodeTemplateId	Request.Operation.JobParam	转码模板 ID，当媒体任务类型为 VideoProcess 、SuperResolution	Container	否

		、SDR2HDR 时有效		
Animation	Request.Operation.JobParam	动图任务参数，当媒体任务类型为 Animation 时有效	Container	否
Transcode	Request.Operation.JobParam	转码任务参数，当媒体任务类型为 Transcode 、VideoProcess 、SuperResolution 、SDR2HDR 时该参数有效	String	否
SmartCover	Request.Operation.JobParam	智能封面任务参数，当媒体任务类型为 SmartCover 时有效	Container	否
DigitalWatermark	Request.Operation.JobParam	数字水印参数，当媒体任务类型为 Transcode 、VideoProcess 、SuperResolution 、DigitalWatermark 时该参数有效	Container	否
Watermark	Request.Operation.JobParam	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
WatermarkTemplateId	Request.Operation.JobParam	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	去除水印参数，当媒体任务类型为 Transcode 时有效	Container	否
Subtitles	Request.Operation.JobParam	字幕参数，当媒体任务类型为 Transcode 时有效	Container	否
Snapshot	Request.Operation.JobParam	截图任务参数，当媒体任务类型为 Snapshot 时有效	Container	否
SpeechRecognition	Request.Operation.JobParam	语音识别任务参数，当媒体任务类型为 SpeechRecognition 时有效	Container	否
ConcatTemplate	Request.Operation.JobParam	拼接任务参数，当媒体任务类型为 Concat 时有效	Container	否
VoiceSeparat	Request.Operatio	人声分离任务参数，当媒体任务类型为	Cont	否

e	n.JobParam	VoiceSeparate 时有效	ainer	
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
NoiseReduction	Request.Operation.JobParam	音频降噪模板参数，当媒体任务类型为 NoiseReduction 时有效	Container	否

注意

优先使用 TemplateId，无 TemplateId 时使用对应任务类型的参数。

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

节点名称（关键字）	父节点	描述	类型	是否必选
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
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 提交转封装任务 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

版权所有：腾讯云计算（北京）有限责任公司

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
Container 类型 NoiseReduction 的内容：
同 提交音频降噪任务 的 Request.Operation.NoiseReduction

响应

响应头

此接口仅返回公共响应头部，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
CreationTi	Response.	任务的创建时间	Stri

me	JobsDetail		ng
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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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>
        <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-12345678900</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: NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18x****

<Response>
    <RequestId>NjJiZDU1ZmZfOTBmYTUwNjRfNzdjY18x****<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：2023-06-20 17:32:27

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>
```

版权所有：腾讯云计算（北京）有限责任公司

```
    </TimeInterval>
    <WorkflowIds>w7476ff3564ee45b7b490d64bccaba6cc</WorkflowIds>
  </Operation>
</Request>
```

具体的数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Request	无	保存请求的容器	Container	是

Container 类型 Request 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Name	Request	批量触发任务名称，支持中文、英文、数字、—和_，长度限制128字符	String	是
Type	Request	批量处理任务类型: Workflow	String	否
Input	Request	待操作的媒体信息	Container	是
Operation	Request	操作规则	Container	是

Container 类型 Input 的具体数据描述如下：

节点名称（关键字）	父节点	描述	类型	是否必选
Manifest	Request.Input	COS 清单列表文件名，COS 清单生成的 manifest.json 文件	String	否
UrlFile	Request.Input	URL 文件名，文件中每行的 URL 为一个 COS 中对象的访问地址	String	否
Prefix	Request.Input	Object 前缀	String	否
Object	Request.Input	媒体文件名	String	否

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.J	任务的状态，为 Submitted、Running、Success、	Strin

	obsDetail	Failed、Pause、Cancel 其中一个	g
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	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=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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：2023-06-20 17:32:28

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。

版权所有：腾讯云计算（北京）有限责任公司

实际案例

请求

```
PUT /inventorytriggerjob/b-xxx-xxx-xxx-xxxx?cancel HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
Host:bucket-1250000000.ci.ap-beijing.myqcloud.com
```

响应

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Mon, 28 Jun 2022 15:23:12 GMT
Server: tencent-ci
x-ci-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzh****=
```

查询任务

最近更新时间：2023-06-20 17:32:28

功能描述

查询批量数据处理任务。

请求

请求示例

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

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求的请求体为空。

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
  <RequestId><RequestId>
  <JobsDetail>
  ...
```

```
    </JobsDetail>
    <NonExistJobIds></NonExistJobIds>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型
Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
JobsDetail	Response	任务的详细信息	Container
NonExistJobIds	Response	查询的 ID 中不存在的任务，所有任务都存在时不返回	String

对于不同的触发方式，JobsDetail 的内容不同，请参照以下链接：

- 工作流
- 独立任务

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /inventorytriggerjob/b3deffea2f84911ec9cb15254008618d9 HTTP/1.1
Accept: */*
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR****&q-s
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>
```

版权所有：腾讯云计算（北京）有限责任公司

批量拉取任务

最近更新时间：2023-06-20 17:32:28

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
Host: <BucketName-APPID>.ci.<Region>.myqcloud.com
Date: <GMT Date>
Authorization: <Auth String>
```

说明

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

请求头

此接口仅使用公共请求头部，详情请参见 公共请求头部 文档。

请求体

该请求无请求体。

请求参数

参数的具体内容如下：

节点名称（关键字）	父节点	描述	类型	必选
nextToken	无	请求的上下文，用于翻页。上次返回的值	String	否
size	无	拉取的最大任务数。默认为10。最大为100	Integer	否

type	无	拉取批量任务类型，工作流类型 Workflow 、任务类型 Job	String	否
orderByTime	无	Desc 或者 Asc。默认为 Desc	String	否
states	无	拉取该状态的任务，以,分割支持多状态 All，Submitted，Running，Success，Failed，Pause，Cancel。默认为 All	String	否
startCreationTime	无	拉取创建时间大于等于该时间的任务。格式为：%Y-%m-%dT%H:%m:%S%z	String	否
endCreationTime	无	拉取创建时间小于等于该时间的任务。格式为：%Y-%m-%dT%H:%m:%S%z	String	否
workflowId	无	工作流 ID	String	否
jobId	无	批量触发任务 ID	String	否
name	无	批量触发任务名称	String	否

响应

响应头

此接口仅返回公共响应头部，详情请参见 公共响应头部 文档。

响应体

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

```
<Response>
    <RequestId></RequestId>
    <JobsDetail>
    ...
    </JobsDetail>
    <NextToken></NextToken>
</Response>
```

具体的数据内容如下：

节点名称（关键字）	父节点	描述	类型

Response	无	保存结果的容器	Container

Container 节点 Response 的内容：

节点名称（关键字）	父节点	描述	类型
RequestId	Response	请求的唯一 ID	String
JobsDetail	Response	任务的详细信息	Container 数组
NextToken	Response	翻页的上下文 Token	String

对于不同的触发方式，JobsDetail 的内容不同，请参照以下链接：

- 工作流
- 独立任务

错误码

该请求操作无特殊错误信息，常见的错误信息请参见 错误码 文档。

实际案例

请求

```
GET /inventorytriggerjob HTTP/1.1
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cIlgPvF9cXFrJD0a1ICvR****&q-s
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：2023-07-14 14:47:02

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 断点续传偏移量可能设置错误。	请检查断点续传偏移量的相关配置。

版权所有：腾讯云计算（北京）有限责任公司

-154	原图保护机制禁止该 URL 下载。	请使用样式访问。
-156	强制执行302流程。	-
-162	业务配置为欠费，禁止下载图片。	请及时充值。
-163	业务配置不存在。	请检查相关配置项。
-164	热图下载，限制。	请降低请求频率，或 提交工单 联系我们处理。
-165	攻击性下载请求，限制。	-
-166	消息格式错误。	-
-167	文件太大，无法支持图片下载。	请降低图片大小。
-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。
-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。
-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。

版权所有：腾讯云计算（北京）有限责任公司

-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	错误码	错误信息	说明
-65001	InternalError	internal error, please try again.	内部错误
-65002	InvalidArgument	invalid params	参数/消息体解析失败。
-65003	InvalidArgument	invalid params	参数不合法。
		missing Object	
		invalid DetectType	
		Param CallbackVersion is	

		illegal	
		No such bucket	
		The image is empty file	
		Param is illegal	
-65004	AuditingJobNotFound	JobId not found	任务不存在。
-65006	SlowDown	RequestLimitExceeded	请求频率过高。
-65007	ResourceNotSupported	Decode file failed	文件解码失败，请检查文件是否损坏。
-65010	InvalidArgument	Download file failed	下载文件失败，请检查文件是否损坏。
		url unreachable	
-65011	InvalidArgument	File is too large	文件过大，不支持审核。
		File is too large or empty	
-65012	InvalidArgument	Resolution is too small	图片分辨率过低，不支持审核。
-65014	CIRoleNotExist	No permission for CI_QCSRole	审核失败，请先开通数据万象角色。(万象角色不存在/未授权)
-65015	InvalidArgument	There is not any label opened	未使用审核策略，请添加审核策略后重试。
-65016	InvalidArgument	The bizType provided does not exist	提供的BizType不存在，请检查后重试。
-65019	AccessDenied	No permission to do this	鉴权失败, 没有passrole权限。
-65020	ResourceNotFound	The file does not exist	文件不存在。
		No such key	
		No such object	
-65023	InvalidArgument	File size is too small	文件大小太小。

-65024	InvalidArgument	Image aspect ratio is too large	图片宽高比过大。

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
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

版权所有：腾讯云计算（北京）有限责任公司

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

请求签名

最近更新时间：2023-07-12 16:17:52

腾讯云数据万象服务通过签名来验证请求的合法性。开发者通过将签名授权给客户端，使其具备上传下载及管理指定资源的能力。
使用数据万象服务时，可通过 RESTful API 发起 HTTP 匿名请求或 HTTP 签名请求，对于签名请求，服务器端将会对请求发起者进行身份验证。

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

数据万象基于 对象存储（Cloud Object Storage，COS），当前签名算法与 COS 保持一致，基于密钥 HMAC（Hash Message Authentication Code）的自定义方案进行身份验证。

签名算法

数据万象已实现和 COS 域名的合并，使用 COS 域名即可实现图片处理操作。对于数据万象域名和 COS 域名，在使用上有所区别，需要分别使用不同的签名版本：

- **数据万象域名**：形式为 `<BucketName-APPID>.<picture region>.myqcloud.com/<picture name>`，例如 `examplebucket-1250000000.picsh.myqcloud.com/picture.jpeg`，操作需使用 **JSON** 版本签名，可参阅 JSON 版本 请求签名 。
- **COS 域名**：形式为 `<BucketName-APPID>.cos.<Region>.myqcloud.com`，例如 `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg`，操作需使用 **XML** 版本签名，可参见 XML 版本 请求签名 。

签名适用场景

场景		适用签名
下载时数据处理	无防盗链	不验证签名
	开启防盗链	验证签名
上传时数据处理	持久化处理	XML 版本签名
Bucket 接口操作	查询、开通、删除等	XML 版本签名
内容识别	鉴黄、违法违规	XML 版本签名

签名工具

生成签名所需信息包括 APPID（例如1250000000），Bucket 名称（例如 examplebucket-ci），项目的 SecretID 和 SecretKey。

版权所有：腾讯云计算（北京）有限责任公司

获取这些信息的方法如下：

1. 登录 数据万象控制台 ，单击左侧菜单栏**存储桶管理**。
2. 单击需要管理的存储桶，进入存储桶管理页面。
3. 单击【存储桶配置】，即可查看存储桶名称和 Bucket ID 。若当前项目无存储桶，您可参考 创建存储桶 文档进行创建。
4. 进入 访问管理 API 密钥管理页面，获取 SecretID 和 SecretKey。

数据万象签名计算过程与 COS 一致，您可以根据**签名适用场景**，使用 COS 提供的 在线签名工具 生成所需版本签名。

签名使用

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

1. 通过标准的 HTTP Authorization 头，如

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

2. 作为 HTTP 请求参数，请注意 UrlEncode，如

```
/exampleobject?q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753%3B1557996953&...&q-signature=...
```

> **注意**
> 上述示例中使用 `...` 省略了部分具体签名内容。

附录
公共请求头部

最近更新时间：2022-12-12 16:24:21

描述

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

请求头部列表

Header 名称	描述	类型	是否必选
Authorization	携带鉴权信息，用以验证请求合法性的签名信息。针对公有读的对象可不携带此头部，如通过请求参数传递鉴权信息也无需携带此头部，详情请参见 请求签名 文档。	string	是。
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，数据万象可复用对象存储 临时安全凭证方法，详情请参见 临时安全凭证 相关说明。	string	否。当使用临时密钥并通过 Authorization

			携带鉴权信息时，此头部为必选项。

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：2020-03-24 16:48:00

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

结构体

最近更新时间：2023-08-10 18:13:43

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
NotifyConfig	回调信息，如果不设置，则使用队列的回调信息	Container	否	无
ExtFilter	文件后缀过滤器	Container	否	无

Container 类型 NotifyConfig 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	默认值	限制
Url	回调地址	String	是	无	不能为内网地址
Type	回调类型	String	是	无	• Url：url 回调 • TDMQ：tdmq 消息回调
Event	回调信息	String	是	无	• TaskFinish：任务完成 • WorkflowFinish：工作流完成 • 支持多种事件，以逗号分隔
ResultFormat	回调格式	String	否	XML	• XML：xml 格式 • JSON：json 格式
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

版权所有：腾讯云计算（北京）有限责任公司

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	Stri	是	• 当工作流节点类型为 Snapshot 或

	称	ng		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

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
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

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
TranscodeTe	音视频转码	String	是	无

版权所有：腾讯云计算（北京）有限责任公司

mplateId	模板 ID			
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_VideoEnhance_***_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
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_FileInfo_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
FileInfo	文件信息节点参数	Container	是	无

Container 类型 FileInfo 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
AdvancedInformation	启用高级信息	String	否	false/true，默认为false

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

版权所有：腾讯云计算（北京）有限责任公司

Output	输出地址	Container，详情请参见 Output	是	无

Container 类型 Segment 的具体数据描述如下：

节点名称（关键字）	描述	类型	是否必选	限制
Format	封装格式	String	是	aac、mp3、flac、mp4、ts、mkv、avi、hls、m3u8
Duration	转封装时长,单位:秒	String	否	不小于5的整数
HlsEncrypt	hls 加密配置	Container	否	只有当封装格式为 hls 时生效，详情请参见 HlsEncrypt

MediaWorkflow_Topology_Nodes_PicProcess_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	图片处理模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_Tts_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	语音合成模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_SpeechRecognition_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	语音识别模板 ID	String	是	无
Output	输出地址	Container，详情请参见 Output	是	无

版权所有：腾讯云计算（北京）有限责任公司

MediaWorkflow_Topology_Nodes_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，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_StreamPack_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
StreamPackInfo	打包规则	Container	否	无

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

节点名称（关键字）	描述	类型	是否必选	限制

版权所有：腾讯云计算（北京）有限责任公司

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

MediaWorkflow_Topology_Nodes_Condition_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
Condition	条件配置,详情请参见 Condition	Container	是	无

MediaWorkflow_Topology_Nodes_SegmentVideoBody_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
SegmentVideoBody	视频人像抠图配置	Container	否	无
Output	输出地址	Container，详情请参见 Output	是	无

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

节点名称（关键字）	描述	类型	是否必选
Mode	抠图模式，当前只支持 Mask (输出alpha通道结果)	String	否

MediaWorkflow_Topology_Nodes_NoiseReduction_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
TemplateId	音频降噪模板 ID	String	否	无

版权所有：腾讯云计算（北京）有限责任公司

Output	输出地址	Container，详情请参见 Output	是	无

MediaWorkflow_Topology_Nodes_ImageInspect_***_Operation

节点名称（关键字）	描述	类型	是否必选	限制
ImageInspect	异常图片检测处理规则	Container	否	无

Container 类型 ImageInspect 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
AutoProcess	是否开启识别到图片异常后自动对图片进行如移动到其他目录、设置为私有权限、删除等动作。取值：true/false，默认为false	String	否	无
ProcessType	指定检测到异常图片后的处理动作 • BackupObject：将图片移动abnormal_images_backup下，该目录由后台自动创建 • SwitchObjectToPrivate：将图片权限设置为私有 • DeleteObject：删除图片 • 默认值：BackupObject • 当 AutoProcess 为 true 时，该参数生效	String	否	无

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，需要做 URLEncode	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: 混音循环 • Once: 混音一次播放
Replace	是否用混音音轨媒体替换Input媒体文件的原音频	String	否	false	true/false

EffectConfig	混音淡入淡出配置	Container	否	false	无

Container 类型 EffectConfig 的具体数据描述如下：

节点名称（关键字）	描述	类型	必选	默认值	限制
EnableStartFadein	开启淡入	String	否	false	true/false
StartFadeinTime	淡入时长	String	否	无	大于0, 支持浮点数
EnableEndFadeout	开启淡出	String	否	false	true/false
EndFadeoutTime	淡出时长	String	否	无	大于0, 支持浮点数
EnableBgmFade	开启 bgm 转换淡入	String	否	false	true/false
BgmFadeTime	bgm 转换淡入时长	String	否	无	大于0, 支持浮点数

AudioMixArray

AudioMixArray是数组，每一个子项内容同 AudioMix

> **说明**
> 仅第一个 Replace 生效

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	Stri	是

		ng	

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	限制

				值	
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

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

版权所有：腾讯云计算（北京）有限责任公司

					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

Condition

节点名称（关键字）	描述	类型	是否必选	限制
Express	表达式详情	String	是	详情请参见 jsonLogic

Express

Express 表达式中的参数内容：

参数	描述	是否必选	限制
Duration	音视频时长	否	无
Height	音视频高	否	无
Width	音视频宽	否	无
Dar	音视频宽高比	否	无
ImageHeight	图片高	否	无
ImageWidth	图片音视频宽	否	无
ImageDar	图片宽高比	否	无
Size	文件大小	否	无

版权所有：腾讯云计算（北京）有限责任公司

MediaResult

Container 节点 MediaResult 的内容：

节点名称（关键字）	描述	类型
OutputFile	输出文件的基本信息	Container

Container 节点 OutputFile 的内容：

节点名称（关键字）	描述	类型
Bucket	输出文件所在的存储桶	String
Region	输出文件所在的存储桶所在的园区	String
ObjectName	输出文件名，可能有多个	String 数组
Md5Info	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）	描述	类型
ObjectName	输出文件名	String
Md5	输出文件的 MD5 值	String

版权所有：腾讯云计算（北京）有限责任公司