

互动白板 服务端 API



腾讯云

【版权声明】

©2013–2025 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】



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或95716。

文档目录

服务端 API

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

参数类型

白板板书生成相关接口

创建白板板书生成任务

获取白板板书生成任务信息

白板推流相关接口

查询白板推流任务

查询白板推流回调地址

设置白板推流回调地址

设置白板推流回调密钥

结束白板推流

开始白板推流

录制视频生成（内测中）相关接口

创建录制视频生成任务

查询录制视频生成任务

查询录制视频生成回调地址

设置录制视频生成回调地址

设置录制视频生成回调密钥

文档转码相关接口

创建文档转码任务

查询文档转码任务

查询文档转码回调地址

设置文档转码回调地址

设置文档转码回调密钥

查询PPT检测任务状态

设置PPT检测任务回调地址

设置PPT检测任务回调密钥

查询PPT检测任务回调地址

创建PPT检测任务

通过文档URL查询转码任务状态

实时录制相关接口

查询实时录制任务

查询实时录制回调地址

暂停实时录制

恢复实时录制

设置实时录制回调地址

设置实时录制回调密钥

开始实时录制

停止实时录制

统计查询相关接口

 查询当前执行中的任务列表

告警（内测中）相关接口

 查询告警回调地址

 设置告警回调地址

数据结构

错误码

服务端 API

更新历史

最近更新时间：2025-03-14 18:17:33

第 39 次发布

发布时间：2025-03-14 18:17:27

本次发布包含了以下内容：

改善已有的文档。

删除接口：

- ApplyTiwTrial
- CreateApplication
- DescribeAPIService
- DescribeApplicationInfos
- DescribeApplicationUsage
- DescribeBoardSDKLog
- DescribeIMApplications
- DescribePostpaidUsage
- DescribeQualityMetrics
- DescribeRecordSearch
- DescribeRoomList
- DescribeTIWDailyUsage
- DescribeTIWRoomDailyUsage
- DescribeTranscodeSearch
- DescribeUsageSummary
- DescribeUserList
- DescribeUserResources
- DescribeUserStatus
- DescribeWhiteboardApplicationConfig
- DescribeWhiteboardBucketConfig
- DescribeWhiteboardPushSearch
- ModifyApplication
- ModifyAutoRenewFlag
- ModifyWhiteboardApplicationConfig
- ModifyWhiteboardBucketConfig

删除数据结构：

- ApplicationItem
- DataItem
- Detail
- RecordTaskResult
- RecordTaskSearchResult
- RoomListItem
- RoomUsageDataItem
- Tag
- TimeValue
- TranscodeTaskResult

- TranscodeTaskSearchResult
- UsageDataItem
- UserListItem
- WhiteboardApplicationConfig
- WhiteboardPushResult
- WhiteboardPushTaskSearchResult

第 38 次发布

发布时间：2023-12-29 01:26:38

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreatePPTCheckTask](#)
 - 新增入参：AutoHandleUnsupportedElementTypes
- [CreateTranscode](#)
 - 新增入参：AutoHandleUnsupportedElementTypes

第 37 次发布

发布时间：2023-12-01 01:26:06

本次发布包含了以下内容：

改善已有的文档。

删除接口：

- CreateOfflineRecord
- DescribeOfflineRecord
- DescribeOfflineRecordCallback
- SetOfflineRecordCallback

修改接口：

- [CreateTranscode](#)
 - 新增入参：ExcelParam

新增数据结构：

- [ExcelParam](#)

第 36 次发布

发布时间：2023-11-01 01:23:35

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribePPTCheck](#)
 - 新增出参：Errs

修改数据结构：

- [PTErr](#)
 - 修改成员：Name, Type, Detail

第 35 次发布

发布时间：2023-10-31 02:34:57

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [StartOnlineRecord](#)
 - 新增入参：TRTCRoomId, TRTCRoomIdStr
- [StartWhiteboardPush](#)
 - 新增入参：GroupId

第 34 次发布

发布时间：2023-09-01 01:24:40

本次发布包含了以下内容：

改善已有的文档。

预下线接口：

- CreateOfflineRecord
- DescribeOfflineRecord
- DescribeOfflineRecordCallback
- SetOfflineRecordCallback

第 33 次发布

发布时间：2023-07-24 01:52:09

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeTranscodeByUrl](#)

第 32 次发布

发布时间：2023-07-21 01:44:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeTranscode](#)
 - 新增出参：CreateTime, AssignTime, FinishedTime

第 31 次发布

发布时间：2023-04-11 01:42:49

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreatePPTCheckTask](#)
- [DescribePPTCheck](#)
- [DescribePPTCheckCallback](#)
- [DescribeRunningTasks](#)
- [DescribeWarningCallback](#)
- [SetPPTCheckCallback](#)
- [SetPPTCheckCallbackKey](#)
- [SetWarningCallback](#)

新增数据结构：

- [PPTErr](#)
- [PPTErrSlide](#)
- [RunningTaskItem](#)

第 30 次发布

发布时间：2023-03-29 04:22:17

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeRecordSearch](#)

新增数据结构：

- [RecordTaskResult](#)
- [RecordTaskSearchResult](#)

第 29 次发布

发布时间：2023-03-27 02:48:49

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ApplyTiwTrial](#)
- [CreateApplication](#)
- [CreateOfflineRecord](#)
- [DescribeAPIService](#)
- [DescribeApplicationInfos](#)
- [DescribeApplicationUsage](#)
- [DescribeBoardSDKLog](#)
- [DescribeIMApplications](#)
- [DescribeOfflineRecord](#)

- DescribeOfflineRecordCallback
- DescribePostpaidUsage
- DescribeRoomList
- DescribeTranscodeSearch
- DescribeUsageSummary
- DescribeUserList
- DescribeUserResources
- DescribeUserStatus
- DescribeWhiteboardApplicationConfig
- DescribeWhiteboardBucketConfig
- DescribeWhiteboardPushSearch
- ModifyApplication
- ModifyAutoRenewFlag
- ModifyWhiteboardApplicationConfig
- ModifyWhiteboardBucketConfig
- SetOfflineRecordCallback

新增数据结构：

- ApplicationItem
- DataItem
- Detail
- RoomListItem
- Tag
- TranscodeTaskResult
- TranscodeTaskSearchResult
- UserListItem
- WhiteboardApplicationConfig
- WhiteboardPushResult
- WhiteboardPushTaskSearchResult

第 28 次发布

发布时间：2023-01-16 02:13:33

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeTranscode](#)
 - 新增出参：ResourceListUrl, Ext
- [StartWhiteboardPush](#)
 - 新增入参：TRTCEnterRoomMode

第 27 次发布

发布时间：2022-12-12 01:52:52

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateTranscode](#)
 - 新增入参: AutoHandleUnsupportedElement

第 26 次发布

发布时间: 2022-12-07 07:09:41

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- DescribeTIWRoomDailyUsage

新增数据结构:

- RoomUsageDataItem

第 25 次发布

发布时间: 2022-05-17 06:17:19

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [StartWhiteboardPush](#)
 - 新增入参: IMAuthParam, TRTCAuthParam

新增数据结构:

- [AuthParam](#)

第 24 次发布

发布时间: 2021-12-27 08:11:32

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateSnapshotTask](#)
 - 新增入参: SnapshotMode

第 23 次发布

发布时间: 2021-11-19 08:15:40

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [StartWhiteboardPush](#)
 - 新增入参: TRTCRoomId, TRTCRoomIdStr

第 22 次发布

发布时间：2021-10-29 08:10:34

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeTIWDailyUsage](#)

新增数据结构：

- [UsageDataItem](#)

第 21 次发布

发布时间：2021-09-07 08:10:23

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateTranscode](#)
 - 新增入参：MinScaleResolution

第 20 次发布

发布时间：2021-07-21 08:10:28

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateSnapshotTask](#)
- [DescribeSnapshotTask](#)

新增数据结构：

- [SnapshotCOS](#)
- [SnapshotResult](#)
- [SnapshotWhiteboard](#)

第 19 次发布

发布时间：2021-07-12 08:10:18

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeOnlineRecord](#)
 - 新增出参：Interrupts

新增数据结构：

- [Interrupt](#)

第 18 次发布

发布时间：2021-06-29 08:16:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateTranscode](#)
 - 新增入参：Priority

第 17 次发布

发布时间：2021-04-23 08:09:10

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- DescribeQualityMetrics

新增数据结构：

- TimeValue

第 16 次发布

发布时间：2021-03-22 08:09:35

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [StartWhiteboardPush](#)
 - 新增入参：VideoFPS, VideoBitrate, UserDefinedRecordId, AutoPublish, UserDefinedStreamId

第 15 次发布

发布时间：2021-02-25 08:08:57

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [StartWhiteboardPush](#)
 - 新增入参：PrivateMapKey

第 14 次发布

发布时间：2021-01-29 08:09:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [StartOnlineRecord](#)
 - 新增入参：AutoStopTimeout

第 13 次发布

发布时间：2021-01-21 08:08:33

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeWhiteboardPush](#)
- [DescribeWhiteboardPushCallback](#)
- [SetWhiteboardPushCallback](#)
- [SetWhiteboardPushCallbackKey](#)

修改接口：

- [CreateTranscode](#)
 - 新增入参：ExtraData
- [CreateVideoGenerationTask](#)
 - 新增入参：ExtraData
- [StartOnlineRecord](#)
 - 新增入参：ExtraData

第 12 次发布

发布时间：2021-01-15 08:08:46

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [StartWhiteboardPush](#)
- [StopWhiteboardPush](#)

新增数据结构：

- [WhiteboardPushBackupParam](#)

第 11 次发布

发布时间：2020-11-04 08:07:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateTranscode](#)
 - 新增出参：TaskId

- [DescribeOnlineRecordCallback](#)
 - 新增出参：Callback, CallbackKey
- [DescribeTranscode](#)
 - 新增出参：Pages, Progress, Resolution, ResultUrl, Status, TaskId, Title, ThumbnailUrl, ThumbnailResolution, CompressFileUrl
- [DescribeTranscodeCallback](#)
 - 新增出参：Callback, CallbackKey

第 10 次发布

发布时间：2020-11-03 08:09:01

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateTranscode](#)
 - 删除出参：TaskId
- [DescribeOnlineRecordCallback](#)
 - 删除出参：Callback, CallbackKey
- [DescribeTranscode](#)
 - 删除出参：Pages, Progress, Resolution, ResultUrl, Status, TaskId, Title, ThumbnailUrl, ThumbnailResolution, CompressFileUrl
- [DescribeTranscodeCallback](#)
 - 删除出参：Callback, CallbackKey

第 9 次发布

发布时间：2020-09-25 08:08:01

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeVideoGenerationTask](#)
 - 新增出参：VideoInfoList

第 8 次发布

发布时间：2020-09-22 08:08:51

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateVideoGenerationTask](#)
- [DescribeVideoGenerationTask](#)
- [DescribeVideoGenerationTaskCallback](#)
- [SetVideoGenerationTaskCallback](#)
- [SetVideoGenerationTaskCallbackKey](#)

第 7 次发布

发布时间：2020-09-14 08:06:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeOnlineRecord](#)
 - 新增出参：ReplayUrl
- [StartOnlineRecord](#)
 - 新增入参：RecordMode, ChatGroupId

第 6 次发布

发布时间：2020-08-21 08:24:00

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [StartOnlineRecord](#)
 - 新增入参：RecordControl

新增数据结构：

- [RecordControl](#)
- [StreamControl](#)

修改数据结构：

- [StreamLayout](#)
 - 新增成员：FillMode

第 5 次发布

发布时间：2020-08-18 08:25:38

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [VideoInfo](#)
 - 新增成员：Width, Height

第 4 次发布

发布时间：2020-06-01 08:21:16

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [SetOnlineRecordCallbackKey](#)
- [SetTranscodeCallbackKey](#)

第 3 次发布

发布时间：2020-05-12 08:19:24

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeOnlineRecordCallback](#)
 - 新增出参：CallbackKey
- [DescribeTranscodeCallback](#)
 - 新增出参：CallbackKey

第 2 次发布

发布时间：2020-01-02 14:51:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeOnlineRecordCallback](#)
- [DescribeTranscodeCallback](#)

修改数据结构：

- [CustomLayout](#)
 - 修改成员：InputStreamList

第 1 次发布

发布时间：2019-12-05 16:48:15

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateTranscode](#)
- [DescribeOnlineRecord](#)
- [DescribeTranscode](#)
- [PauseOnlineRecord](#)
- [ResumeOnlineRecord](#)
- [SetOnlineRecordCallback](#)
- [SetTranscodeCallback](#)
- [StartOnlineRecord](#)
- [StopOnlineRecord](#)

新增数据结构：

- [Canvas](#)
- [Concat](#)
- [CustomLayout](#)
- [LayoutParams](#)
- [MixStream](#)

- [OmittedDuration](#)
- [StreamLayout](#)
- [VideoInfo](#)
- [Whiteboard](#)

简介

最近更新时间：2025-04-25 02:02:12

腾讯云互动白板（Tencent Interactive Whiteboard）提供整套完备的多人实时白板互动服务，打破线上教学中师生信息传递障碍，拥有比面授教学板书更丰富、直观和多样的功能。包含的功能有互动涂鸦、轨迹实时同步、文档共享、媒体共享、文档转码、白板与音视频实时同步录制、回放等功能。具备灵活易用、扩展性强的优点，帮助用户高度还原线下面授教学的互动体验，显著提升线上教学质量。

API 概览

最近更新时间：2025-03-14 18:22:27

白板板书生成相关接口

接口名称	接口功能	频率限制（次/秒）
CreateSnapshotTask	创建白板板书生成任务	20
DescribeSnapshotTask	获取白板板书生成任务信息	20

白板推流相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeWhiteboardPush	查询白板推流任务	20
DescribeWhiteboardPushCallback	查询白板推流回调地址	20
SetWhiteboardPushCallback	设置白板推流回调地址	20
SetWhiteboardPushCallbackKey	设置白板推流回调密钥	20
StopWhiteboardPush	结束白板推流	20
StartWhiteboardPush	开始白板推流	20

录制视频生成（内测中）相关接口

接口名称	接口功能	频率限制（次/秒）
CreateVideoGenerationTask	创建录制视频生成任务	20
DescribeVideoGenerationTask	查询录制视频生成任务	20
DescribeVideoGenerationTaskCallback	查询录制视频生成回调地址	20
SetVideoGenerationTaskCallback	设置录制视频生成回调地址	20
SetVideoGenerationTaskCallbackKey	设置录制视频生成回调密钥	20

文档转码相关接口

接口名称	接口功能	频率限制（次/秒）
CreateTranscode	创建文档转码任务	20
DescribeTranscode	查询文档转码任务	20
DescribeTranscodeCallback	查询文档转码回调地址	20
SetTranscodeCallback	设置文档转码回调地址	20
SetTranscodeCallbackKey	设置文档转码回调密钥	20
DescribePPTCheck	查询PPT检测任务状态	20

接口名称	接口功能	频率限制（次/秒）
SetPPTCheckCallback	设置PPT检测任务回调地址	20
SetPPTCheckCallbackKey	设置PPT检测任务回调密钥	20
DescribePPTCheckCallback	查询PPT检测任务回调地址	20
CreatePPTCheckTask	创建PPT检测任务	20
DescribeTranscodeByUrl	通过文档URL查询转码任务状态	20

实时录制相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeOnlineRecord	查询实时录制任务	20
DescribeOnlineRecordCallback	查询实时录制回调地址	20
PauseOnlineRecord	暂停实时录制	20
ResumeOnlineRecord	恢复实时录制	20
SetOnlineRecordCallback	设置实时录制回调地址	20
SetOnlineRecordCallbackKey	设置实时录制回调密钥	20
StartOnlineRecord	开始实时录制	20
StopOnlineRecord	停止实时录制	20

统计查询相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeRunningTasks	查询当前执行中的任务列表	20

告警（内测中）相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeWarningCallback	查询告警回调地址	20
SetWarningCallback	设置告警回调地址	20

⚠ 注意：

以上给出的接口频率限制维度为 API + 接入地域 + 子账号，有关限频更多说明参考：[API 频率限制说明](#)

调用方式

请求结构

最近更新时间：2025-05-21 01:33:02

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 tiw.tencentcloudapi.com，也支持指定地域域名访问，例如广州地域的域名为 tiw.ap-guangzhou.tencentcloudapi.com。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 tiw.ap-guangzhou.tencentcloudapi.com 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的地域。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	tiw.tencentcloudapi.com
华南地区（广州）	tiw.ap-guangzhou.tencentcloudapi.com
华东地区（上海）	tiw.ap-shanghai.tencentcloudapi.com
华东地区（南京）	tiw.ap-nanjing.tencentcloudapi.com
华北地区（北京）	tiw.ap-beijing.tencentcloudapi.com
西南地区（成都）	tiw.ap-chengdu.tencentcloudapi.com
西南地区（重庆）	tiw.ap-chongqing.tencentcloudapi.com
港澳台地区（中国香港）	tiw.ap-hongkong.tencentcloudapi.com
亚太东南（新加坡）	tiw.ap-singapore.tencentcloudapi.com
亚太东南（雅加达）	tiw.ap-jakarta.tencentcloudapi.com
亚太东南（曼谷）	tiw.ap-bangkok.tencentcloudapi.com
亚太东北（首尔）	tiw.ap-seoul.tencentcloudapi.com
亚太东北（东京）	tiw.ap-tokyo.tencentcloudapi.com
美国东部（弗吉尼亚）	tiw.na-ashburn.tencentcloudapi.com
美国西部（硅谷）	tiw.na-siliconvalley.tencentcloudapi.com
南美地区（圣保罗）	tiw.sa-saopaulo.tencentcloudapi.com
欧洲地区（法兰克福）	tiw.eu-frankfurt.tencentcloudapi.com

注意：由于金融区和非金融区是隔离不互通的，因此当访问金融区服务时（公共参数 Region 为金融区地域），需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致。

金融区接入地域	金融区域名
华东地区（上海金融）	tiw.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	tiw.ap-shenzhen-fsi.tencentcloudapi.com

2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法：

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

- application/json（推荐），必须使用签名方法 v3（TC3-HMAC-SHA256）。
- application/x-www-form-urlencoded，必须使用签名方法 v1（HmacSHA1 或 HmacSHA256）。
- multipart/form-data（仅部分接口支持），必须使用签名方法 v3（TC3-HMAC-SHA256）。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1（HmacSHA1、HmacSHA256）时不得超过1MB。POST 请求使用签名方法 v3（TC3-HMAC-SHA256）时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2024-11-15 02:08:06

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见 [签名方法 v3](#)。

注意：出于简化的目的，部分接口文档中的示例使用的是签名方法 v1 GET 请求，而不是更安全的签名方法 v3。

使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下表所示：

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 tiw； tc3_request 为固定字符串； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-10-09/cvm/tc3_request,
SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST (application/json) 请求结构示例:

```
https://cvm.tencentcloudapi.com/
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-05-30/cvm/tc3_request,
SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
{"Offset":0,"Limit":10}
```

HTTP POST (multipart/form-data) 请求结构示例 (仅特定的接口支持):

```
https://cvm.tencentcloudapi.com/
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-05-30/cvm/tc3_request,
SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```


使用签名方法 v1 的公共参数

使用签名方法 v1（有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见[文档](#)

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	—	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。
Version	String	是	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****
```

```
Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/
```

```
Host: cvm.tencentcloudapi.com
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****
```

地域列表

本产品所有接口 Region 字段的可选值如下表所示。如果接口不支持该表中的所有地域，则会在接口文档中单独说明。

地域	取值
华南地区（广州）	ap-guangzhou

签名方法 v3

最近更新时间：2024-12-25 02:04:47

以下文档说明了签名方法 v3 的签名过程，但仅在您编写自己的代码来调用腾讯云 API 时才有用。我们推荐您使用 [腾讯云 API Explorer](#)，[腾讯云 SDK](#) 和 [腾讯云命令行工具（TCCLI）](#) 等开发者工具，从而无需学习如何对 API 请求进行签名。

推荐使用 API Explorer

</> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共参数中指定该签名结果并以指定的方式和格式发送请求。

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 [云 API 密钥](#) 页面获取密钥相关信息。

2. 保护传输中的数据

为了防止请求在传输过程中被篡改，腾讯云 API 会使用请求参数来计算请求的哈希值，并将生成的哈希值加密后作为请求的一部分，发送到腾讯云 API 服务器。服务器会使用收到的请求参数以同样的过程计算哈希值，并验证请求中的哈希值。如果请求被篡改，将导致哈希值不一致，腾讯云 API 将拒绝本次请求。

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 JSON 格式，POST 请求支持传空数组和空字符串，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以对生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- 用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄露，请立刻禁用该安全凭证。

申请安全凭证的具体步骤如下：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云API密钥](#) 的控制台页面。
3. 在 [云API密钥](#) 页面，单击【新建密钥】创建一对密钥。

签名版本 v3 签名过程

云 API 支持 GET 和 POST 请求。对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于 POST 方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式绝大多数接口均支持，multipart 格式只有特定接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

1. 云服务器默认已开通，该接口很常用；
2. 该接口是只读的，不会改变现有资源的状态；
3. 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC- 前缀）。

假设用户的 SecretId 和 SecretKey 分别是：AKID***** 和 *****。用户想查看广州云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠 (/)。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；

字段名称	解释
	2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 <code>content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n</code>。 注意：<code>content-type</code> 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 <code>charset</code> 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。<code>content-type</code> 和 <code>host</code> 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号（;）分隔。 此示例为 <code>content-type;host;x-tc-action</code>
HashedRequestPayload	请求正文（payload，即 body，此示例为 <code>{"Limit": 1, "Filters": [{"Values": [{"u672a\u547d\u540d}], "Name": "instance-name"}]}</code>）的哈希值，计算伪代码为 <code>Lowercase(HexEncode(Hash.SHA256(RequestPayload)))</code>，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 <code>35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064</code>。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com
x-tc-action:describeinstances

content-type;host;x-tc-action
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + "\n" +
RequestTimestamp + "\n" +
CredentialScope + "\n" +
HashedCanonicalRequest
```

字段名称	解释
Algorithm	签名算法，目前固定为 <code>TC3-HMAC-SHA256</code> 。
RequestTimestamp	请求时间戳，即请求头部的公共参数 <code>X-TC-Timestamp</code> 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 <code>1551113065</code> 。
CredentialScope	凭证范围，格式为 <code>Date/service/tc3_request</code> ，包含日期、所请求的服务和终止字符串（ <code>tc3_request</code> ）。 Date 为 UTC 标准时间的日期，取值需要和公共参数 <code>X-TC-Timestamp</code> 换算的

字段名称	解释
	UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能运行一段时间后，请求失败，返回签名过期错误。

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "*****"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：da98fb70dcf6b112dc21038d1eeeb3a95c74b4dcb12c1131f864f6066bd02be0，8d70cbefb03939f929db64d32dc2ba89b1095620119fe3e050e2b18c5bd2752f，b596b923aad85185e2d1f6659d2a062e0a86731226e021e61bfe06f7ed05f5af。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，请以实际情况为准。此处的伪代码密钥参数 key 在前，消息参数 data 在后。通常标准库函数会提供二进制格式的返回值，也可能会提供打印友好的十六进制格式的返回值，此处使用的是二进制格式。

字段名称	解释
SecretKey	原始的 SecretKey，即 *****。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名，伪代码如下：

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

4. 拼接 Authorization

按如下格式拼接 Authorization:

```
Authorization =  
Algorithm + ' ' +  
'Credential=' + SecretId + '/' + CredentialScope + ', ' +  
'SignedHeaders=' + SignedHeaders + ', ' +  
'Signature=' + Signature
```

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKID*****。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host;x-tc-action。
Signature	签名值。此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=c  
ontent-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/  
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request,  
SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b4  
6093f6ab6c4f  
Content-Type: application/json; charset=utf-8  
Host: cvm.tencentcloudapi.com  
X-TC-Action: DescribeInstances  
X-TC-Version: 2017-03-12  
X-TC-Timestamp: 1551113065  
X-TC-Region: ap-guangzhou  
  
{ "Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}
```

⚠ 注意：

请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过

打印实际请求内容，网络抓包等方式对比排查。

签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚地解释签名过程，下面以实际编程语言为例，将上述的签名过程完整实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
    private final static String CT_JSON = "application/json; charset=utf-8";

    public static byte[] hmac256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(UTF8));
    }
}
```



```
public static String sha256Hex(String s) throws Exception {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] d = md.digest(s.getBytes(UTF8));
    return DatatypeConverter.printHexBinary(d).toLowerCase();
}

public static void main(String[] args) throws Exception {
    String service = "cvm";
    String host = "cvm.tencentcloudapi.com";
    String region = "ap-guangzhou";
    String action = "DescribeInstances";
    String version = "2017-03-12";
    String algorithm = "TC3-HMAC-SHA256";
    String timestamp = "1551113065";
    //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    // 注意时区, 否则容易出错
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
    String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

    // ***** 步骤 1: 拼接规范请求串 *****
    String httpRequestMethod = "POST";
    String canonicalUri = "/";
    String canonicalQueryString = "";
    String canonicalHeaders = "content-type:application/json; charset=utf-8\n"
    + "host:" + host + "\n" + "x-tc-action:" + action.toLowerCase() + "\n";
    String signedHeaders = "content-type;host;x-tc-action";

    String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
    String hashedRequestPayload = sha256Hex(payload);
    String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
    + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
    System.out.println(canonicalRequest);

    // ***** 步骤 2: 拼接待签名字符串 *****
    String credentialScope = date + "/" + service + "/" + "tc3_request";
    String hashedCanonicalRequest = sha256Hex(canonicalRequest);
    String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
    System.out.println(stringToSign);

    // ***** 步骤 3: 计算签名 *****
    byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
    byte[] secretService = hmac256(secretDate, service);
    byte[] secretSigning = hmac256(secretService, "tc3_request");
    String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
    System.out.println(signature);

    // ***** 步骤 4: 拼接 Authorization *****
```

```
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: \").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\")")
.append(" -H \"Host: \").append(host).append("\")")
.append(" -H \"X-TC-Action: \").append(action).append("\")")
.append(" -H \"X-TC-Timestamp: \").append(timestamp).append("\")")
.append(" -H \"X-TC-Version: \").append(version).append("\")")
.append(" -H \"X-TC-Region: \").append(region).append("\")")
.append(" -d '").append(payload).append("'");
System.out.println(sb.toString());
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcnow().timestamp(timestamp).strftime("%Y-%m-%d")
```

```
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}
```

```
# ***** 步骤 1: 拼接规范请求串 *****
```

```
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)
```

```
# ***** 步骤 2: 拼接待签名字符串 *****
```

```
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)
```

```
# ***** 步骤 3: 计算签名 *****
```

```
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)
```

```
# ***** 步骤 4: 拼接 Authorization *****
```

```
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)
```

```
print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"
+ ' -H "Host: ' + host + '"')
```

```
+ ' -H "X-TC-Action: ' + action + '"'  
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"'  
+ ' -H "X-TC-Version: ' + version + '"'  
+ ' -H "X-TC-Region: ' + region + '"'  
+ " -d '" + payload + '"')
```

Golang

```
package main  
  
import (  
    "crypto/hmac"  
    "crypto/sha256"  
    "encoding/hex"  
    "fmt"  
    "os"  
    "strings"  
    "time"  
)  
  
func sha256hex(s string) string {  
    b := sha256.Sum256([]byte(s))  
    return hex.EncodeToString(b[:])  
}  
  
func hmacsha256(s, key string) string {  
    hashed := hmac.New(sha256.New, []byte(key))  
    hashed.Write([]byte(s))  
    return string(hashed.Sum(nil))  
}  
  
func main() {  
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****  
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")  
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****  
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")  
    host := "cvm.tencentcloudapi.com"  
    algorithm := "TC3-HMAC-SHA256"  
    service := "cvm"  
    version := "2017-03-12"  
    action := "DescribeInstances"  
    region := "ap-guangzhou"  
    //var timestamp int64 = time.Now().Unix()  
    var timestamp int64 = 1551113065  
  
    // step 1: build canonical request string  
    httpRequestMethod := "POST"  
    canonicalURI := "/"  
    canonicalQueryString := ""
```

```
canonicalHeaders := fmt.Sprintf("content-type:%s\nhost:%s\nx-tc-action:%s\n",
    "application/json; charset=utf-8", host, strings.ToLower(action))
signedHeaders := "content-type;host;x-tc-action"
payload := `{ "Limit": 1, "Filters": [{ "Values": [ "\u672a\u547d\u540d" ], "Name": "instance-name" } ] }`
hashedRequestPayload := sha256hex(payload)
canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
    httpRequestMethod,
    canonicalURI,
    canonicalQueryString,
    canonicalHeaders,
    signedHeaders,
    hashedRequestPayload)
fmt.Println(canonicalRequest)

// step 2: build string to sign
date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
hashedCanonicalRequest := sha256hex(canonicalRequest)
string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
    algorithm,
    timestamp,
    credentialScope,
    hashedCanonicalRequest)
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
    algorithm,
    secretId,
    credentialScope,
    signedHeaders,
    signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
```

```
fmt.Println(curl)
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = implode("\n", [
    "content-type:application/json; charset=utf-8",
    "host:".$host,
    "x-tc-action:".strtolower($action),
    ""
]);
$signedHeaders = implode(";", [
    "content-type",
    "host",
    "x-tc-action",
]);
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/".$service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
```

```
.$timestamp."\n"
.$credentialScope."\n"
.$shashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=".$secretId."/". $credentialScope
.", SignedHeaders=".$signedHeaders.", Signature=".$signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"
.' -H "Content-Type: application/json; charset=utf-8"
.' -H "Host: '.$host.'"
.' -H "X-TC-Action: '.$action.'"
.' -H "X-TC-Timestamp: '.$timestamp.'"
.' -H "X-TC-Version: '.$version.'"
.' -H "X-TC-Region: '.$region.'"
." -d '". $payload.'"";
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
require 'json'
require 'time'
require 'openssl'

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
```

```
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\nx-tc-action:#{action.downcase}\n"
signed_headers = 'content-type;host;x-tc-action'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['未命名'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
  http_request_method,
  canonical_uri,
  canonical_querystring,
  canonical_headers,
  signed_headers,
  hashed_request_payload,
].join("\n")

puts canonical_request

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
  algorithm,
  timestamp.to_s,
  credential_scope,
  hashed_request_payload,
].join("\n")
puts string_to_sign

# ***** 步骤 3: 计算签名 *****
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ***** 步骤 4: 拼接 Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers},
```



```
Signature=#{signature}"}
puts authorization

puts 'curl -X POST ' + endpoint \
+ ' -H "Authorization: ' + authorization + '"' \
+ ' -H "Content-Type: application/json; charset=utf-8"' \
+ ' -H "Host: ' + host + '"' \
+ ' -H "X-TC-Action: ' + action + '"' \
+ ' -H "X-TC-Timestamp: ' + timestamp.to_s + '"' \
+ ' -H "X-TC-Version: ' + version + '"' \
+ ' -H "X-TC-Region: ' + region + '"' \
+ " -d '" + payload + '"'
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

public class Application
{
    public static string SHA256Hex(string s)
    {
        using (SHA256 algo = SHA256.Create())
        {
            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
            StringBuilder builder = new StringBuilder();
            for (int i = 0; i < hashbytes.Length; ++i)
            {
                builder.Append(hashbytes[i].ToString("x2"));
            }
            return builder.ToString();
        }
    }

    public static byte[] HmacSHA256(byte[] key, byte[] msg)
    {
        using (HMACSHA256 mac = new HMACSHA256(key))
        {
            return mac.ComputeHash(msg);
        }
    }

    public static Dictionary<String, String> BuildHeaders(string secretid,
        string secretkey, string service, string endpoint, string region,
        string action, string version, DateTime date, string requestPayload)
    {
        string datestr = date.ToString("yyyy-MM-dd");
```

```
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;

// ***** 步骤 1: 拼接规范请求串 *****
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

// ***** 步骤 2: 拼接待签名字符串 *****
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n"
+ requestTimestamp.ToString() + "\n"
+ credentialScope + "\n"
+ hashedCanonicalRequest;
Console.WriteLine(stringToSign);

// ***** 步骤 3: 计算签名 *****
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);

// ***** 步骤 4: 拼接 Authorization *****
string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
```

```
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}

public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

    string service = "cvm";
    string endpoint = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";

    // 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
    // DateTime date = DateTime.UtcNow;
    // 注意时区, 建议此时间统一采用UTC时间戳, 否则容易出错
    DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
    string requestPayload = "{\n\"Limit\": 1, \"Filters\": [{\n\"Values\": [\n\"\\u672a\\u547d\\u540d\"], \"Name\":\n\"instance-name\"}]}";

    Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service
, endpoint, region, action, version, date, requestPayload);

    Console.WriteLine("POST https://cvm.tencentcloudapi.com");
    foreach (KeyValuePair<string, string> kv in headers)
    {
        Console.WriteLine(kv.Key + ": " + kv.Value);
    }
    Console.WriteLine();
    Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
    const hmac = crypto.createHmac('sha256', secret)
    return hmac.update(message).digest(encoding)
}
```

```
function getHash(message, encoding = 'hex') {
  const hash = crypto.createHash('sha256')
  return hash.update(message).digest(encoding)
}

function getDate(timestamp) {
  const date = new Date(timestamp * 1000)
  const year = date.getUTCFullYear()
  const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
  const day = ('0' + date.getUTCDate()).slice(-2)
  return `${year}-${month}-${day}`
}

function main(){
  // 密钥参数
  // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
  const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
  // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
  const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

  const endpoint = "cvm.tencentcloudapi.com"
  const service = "cvm"
  const region = "ap-guangzhou"
  const action = "DescribeInstances"
  const version = "2017-03-12"
  //const timestamp = getTime()
  const timestamp = 1551113065
  //时间处理, 获取世界时间日期
  const date = getDate(timestamp)

  // ***** 步骤 1: 拼接规范请求串 *****
  const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"

  const hashedRequestPayload = getHash(payload);
  const httpRequestMethod = "POST"
  const canonicalUri = "/"
  const canonicalQueryString = ""
  const canonicalHeaders = "content-type:application/json; charset=utf-8\\n"
  + "host:" + endpoint + "\\n"
  + "x-tc-action:" + action.toLowerCase() + "\\n"
  const signedHeaders = "content-type;host;x-tc-action"

  const canonicalRequest = httpRequestMethod + "\\n"
  + canonicalUri + "\\n"
  + canonicalQueryString + "\\n"
  + canonicalHeaders + "\\n"
  + signedHeaders + "\\n"
  + hashedRequestPayload
```

```
console.log(canonicalRequest)

// ***** 步骤 2: 拼接待签名字符串 *****
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\n" +
timestamp + "\n" +
credentialScope + "\n" +
hashedCanonicalRequest
console.log(stringToSign)

// ***** 步骤 3: 计算签名 *****
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)

// ***** 步骤 4: 拼接 Authorization *****
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)

const curlcmd = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + '"'
console.log(curlcmd)
}
main()
```

C++

```
#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
```

```
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
    utcDate = string(buff);
    return utcDate;
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str.c_str(), str.size());
    SHA256_Final(hash, &sha256);
    std::string NewString = "";
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        NewString = NewString + buf;
    }
    return NewString;
}

string HmacSha256(const string &key, const string &input)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
```

```
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )&input[0], input.length());
unsigned int len = 32;
HMAC_Final(h, hash, &len);

#ifdef OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif

std::stringstream ss;
ss << std::setfill('0');
for (int i = 0; i < len; i++)
{
ss << hash[i];
}

return (ss.str());
}

string HexEncode(const string &input)
{
static const char* const lut = "0123456789abcdef";
size_t len = input.length();

string output;
output.reserve(2 * len);
for (size_t i = 0; i < len; ++i)
{
const unsigned char c = input[i];
output.push_back(lut[c >> 4]);
output.push_back(lut[c & 15]);
}
return output;
}

int main()
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

string service = "cvm";
```

```
string host = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
int64_t timestamp = 1551113065;
string date = get_data(timestamp);

// ***** 步骤 1: 拼接规范请求串 *****

string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string lower = action;
std::transform(action.begin(), action.end(), lower.begin(), ::tolower);
string canonicalHeaders = string("content-type:application/json; charset=utf-8\n")
+ "host:" + host + "\n"
+ "x-tc-action:" + lower + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
cout << canonicalRequest << endl;

// ***** 步骤 2: 拼接待签名字符串 *****

string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;

// ***** 步骤 3: 计算签名 *****

string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;

// ***** 步骤 4: 拼接 Authorization *****

string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;
```



```
string curlcmd = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: \" + authorization + "\"\n"
+ " -H \"Content-Type: application/json; charset=utf-8\" + "\n"
+ " -H \"Host: \" + host + "\"\n"
+ " -H \"X-TC-Action: \" + action + "\"\n"
+ " -H \"X-TC-Timestamp: \" + RequestTimestamp + "\"\n"
+ " -H \"X-TC-Version: \" + version + "\"\n"
+ " -H \"X-TC-Region: \" + region + "\"\n"
+ " -d '" + payload + "'";
cout << curlcmd << endl;
return 0;
};
```

C

```
#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdint.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

void get_utc_date(int64_t timestamp, char* utc, int len)
{
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(utc, len, "%Y-%m-%d", &sttime);
}

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        sprintf(buf, sizeof(buf), "%02x", hash[i]);
        strcat(result, buf);
    }
}

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
```

```
unsigned char* output, unsigned int* output_len)
{
    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
    #else
    h = HMAC_CTX_new();
    #endif

    HMAC_Init_ex(h, key, key_len, EVP_sha256(), NULL);
    HMAC_Update(h, ( unsigned char* )input, input_len);
    HMAC_Final(h, output, output_len);

    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
    #else
    HMAC_CTX_free(h);
    #endif

}

void hex_encode(const char* input, int input_len, char* output)
{
    static const char* const lut = "0123456789abcdef";

    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < input_len; ++i)
    {
        const unsigned char c = input[i];
        temp[0] = lut[c >> 4];
        strcat(add_out, temp);
        temp[0] = lut[c & 15];
        strcat(add_out, temp);
    }
    strncpy(output, add_out, 128);
}

void lowercase(const char * src, char * dst)
{
    for (int i = 0; src[i]; i++)
    {
        dst[i] = tolower(src[i]);
    }
}

int main()
```

```
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
const char* service = "cvm";
const char* host = "cvm.tencentcloudapi.com";
const char* region = "ap-guangzhou";
const char* action = "DescribeInstances";
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

// ***** 步骤 1: 拼接规范请求串 *****
const char* http_request_method = "POST";
const char* canonical_uri = "/";
const char* canonical_query_string = "";
char canonical_headers[100] = {"content-type:application/json; charset=utf-8\nhost:"};
strcat(canonical_headers, host);
strcat(canonical_headers, "\n\nx-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"i\n\nstance-name\"}] }";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

char canonical_request[256] = {0};
sprintf(canonical_request, "%s\n%s\n%s\n%s\n%s\n%s", http_request_method,
canonical_uri, canonical_query_string, canonical_headers,
signed_headers, hashed_request_payload);
printf("%s\n", canonical_request);

// ***** 步骤 2: 拼接待签名字符串 *****
const char* algorithm = "TC3-HMAC-SHA256";
char request_timestamp[16] = {0};
sprintf(request_timestamp, "%d", timestamp);
char credential_scope[64] = {0};
strcat(credential_scope, date);
sprintf(credential_scope, "%s/%s/tc3_request", date, service);
char hashed_canonical_request[100] = {0};
sha256_hex(canonical_request, hashed_canonical_request);
char string_to_sign[256] = {0};
sprintf(string_to_sign, "%s\n%s\n%s\n%s", algorithm, request_timestamp,
```

```
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s", "TC3", SECRET_KEY);
unsigned char k_date[64] = {0};
unsigned int output_len = 0;
hmac_sha256(k_key, strlen(k_key), date, strlen(date), k_date, &output_len);
unsigned char k_service[64] = {0};
hmac_sha256(k_date, output_len, service, strlen(service), k_service, &output_len);
unsigned char k_signing[64] = {0};
hmac_sha256(k_service, output_len, "tc3_request", strlen("tc3_request"), k_signing, &output_len);
unsigned char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, output_len, string_to_sign, strlen(string_to_sign), k_hmac_sha_sign, &output_len);

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

// ***** 步骤 4: 拼接 Authorization *****
char authorization[512] = {0};
sprintf(authorization, "%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm, SECRET_ID, credential_scope, signed_headers, signature);
printf("%s\n", authorization);

char curlcmd[10240] = {0};
sprintf(curlcmd, "curl -X POST https://%s\n \
-H \"Authorization: %s\"\n \
-H \"Content-Type: application/json; charset=utf-8\"\n \
-H \"Host: %s\"\n \
-H \"X-TC-Action: %s\"\n \
-H \"X-TC-Timestamp: %s\"\n \
-H \"X-TC-Version: %s\"\n \
-H \"X-TC-Region: %s\"\n \
-d '%s'",
host, authorization, host, action, request_timestamp, version, region, payload);
printf("%s\n", curlcmd);
return 0;
}
```

其他语言

- Lua: [GitHub](#)
- Swift: [GitHub](#)
- Dart: [GitHub](#)
- Shell(Bash): [GitHub](#)

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2024-12-25 02:04:47

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

推荐使用 API Explorer

<> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。

1. 申请安全凭证

在第一次使用云 API 之前，请前往 [云 API 密钥页面](#) 申请安全凭证。

安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云 API 密钥](#) 的控制台页面
3. 在 [云 API 密钥](#) 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。以下是使用签名方法 v1 生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKID*****
- SecretKey: *****

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表（DescribeInstances）请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKID*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886

参数名称	中文	参数值
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数。

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action' : 'DescribeInstances',
  'InstanceIds.0' : 'ins-09dx96dg',
  'Limit' : 20,
  'Nonce' : 11886,
  'Offset' : 0,
  'Region' : 'ap-guangzhou',
  'SecretId' : 'AKID*****',
  'Timestamp' : 1465185768,
  'Version' : '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 "DescribeInstances"，因此格式化后即为 Action=DescribeInstances。

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为: cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。

4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 + 请求路径 + ? + 请求字符串。

示例的拼接结果为:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。

首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签名, 然后将生成的签名串使用 Base64 进行编码, 即可获得最终的签名串。

具体代码如下, 以 PHP 语言为例:

```
$secretKey = '*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

使用其它程序设计语言开发时, 可用上面示例中的原文进行签名验证, 得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数, 需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ=, 最终得到的签名串请求参数 (Signature) 为: 7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D, 它将用于生成最终的请求 URL。

注意: 如果用户的请求方法是 GET, 或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded, 则发送请求时所有请求参数的值均需要做 URL 编码, 参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先用 UTF-8 进行编码。

注意: 有些编程语言的库会自动为所有参数进行 urlencode, 在这种情况下, 就不需要对签名串进行 URL 编码了, 否则两次 URL 编码会导致签名失败。

注意: 其他参数值也需要进行编码, 编码采用 [RFC 3986](#)。使用 %XY 对特殊字符例如汉字进行百分比编码, 其中 “X” 和 “Y” 为十六进制字符 (0-9 和大写字母 A-F), 使用小写将引发错误。

4. 签名失败

根据实际情况, 存在以下签名失败的错误码, 请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期

错误代码	错误描述
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12。`

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";
```

```
public static String sign(String s, String key, String method) throws Exception {
    Mac mac = Mac.getInstance(method);
    SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
    mac.init(secretKeySpec);
    byte[] hash = mac.doFinal(s.getBytes(CHARSET));
    return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
    StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
    // 签名时要求对参数进行字典排序, 此处用TreeMap保证顺序
    for (String k : params.keySet()) {
        s2s.append(k).append("=").append(params.get(k).toString()).append("&");
    }
    return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode, 由于key都是英文字母, 故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数, 例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间, 例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA1")); // 公共参数
    System.out.println(getUrl(params));
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： `pip install requests` 。

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import os
import time

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

Golang

```
package main

import (
    "bytes"
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "fmt"
    "os"
    "sort"
    "strconv"
)

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": strconv.Itoa(1465185768),
        "Region": "ap-guangzhou",
        "SecretId": secretId,
        "Version": "2017-03-12",
        "Action": "DescribeInstances",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": strconv.Itoa(20),
        "Offset": strconv.Itoa(0),
    }

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("cvm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

    // sort keys by ascii asc order
    keys := make([]string, 0, len(params))
    for k, _ := range params {
        keys = append(keys, k)
    }
    sort.Strings(keys)

    for i := range keys {
        k := keys[i]
        buf.WriteString(k)
        buf.WriteString("=")
        buf.WriteString(params[k])
        buf.WriteString("&")
    }
}
```

```
}

buf.Truncate(buf.Len() - 1)

hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $params as $key => $value ) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
  'SecretId' => secret_id,
  'Timestamp' => 1465185768, # Time.now.to_i
  'Version' => '2017-03-12',
}
sign = method + endpoint + '/'
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;
```

```
public class Application {
    public static string Sign(string signKey, string secret)
    {
        string signRet = string.Empty;
        using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
        {
            byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
            signRet = Convert.ToBase64String(hash);
        }
        return signRet;
    }

    public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
    {
        string retStr = "";
        retStr += requestMethod;
        retStr += requestHost;
        retStr += requestPath;
        retStr += "?";
        string v = "";
        foreach (string key in requestParams.Keys)
        {
            v += string.Format("{0}={1}&", key, requestParams[key]);
        }
        retStr += v.TrimEnd('&');
        return retStr;
    }

    public static void Main(string[] args)
    {
        // 密钥参数
        // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
        string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
        // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
        string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

        string endpoint = "cvm.tencentcloudapi.com";
        string region = "ap-guangzhou";
        string action = "DescribeInstances";
        string version = "2017-03-12";

        double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25, 此参数作为示例, 以实际为准
        // long timestamp = ToTimestamp() / 1000;
        // string requestTimestamp = timestamp.ToString();
        Dictionary<string, string> param = new Dictionary<string, string>();
        param.Add("Limit", "20");
        param.Add("Offset", "0");
        param.Add("InstanceIds.0", "ins-09dx96dg");
        param.Add("Action", action);
        param.Add("Nonce", "11886");
        // param.Add("Nonce", Math.Abs(new Random().Next()).ToString());
    }
}
```

```
param.Add("Timestamp", RequestTimestamp.ToString());
param.Add("Version", version);

param.Add("SecretId", SECRET_ID);
param.Add("Region", region);
SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
string sigInParameter = MakeSignPlainText(headers, "GET", endpoint, "/");
string sigOutParam = Sign(SECRET_KEY, sigInParameter);
Console.WriteLine(sigOutParam);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
    params['Signature'] = encodeURIComponent(params['Signature']);
    const url_strParam = sort_params(params)
    return "https://" + endpoint + "/" + url_strParam.slice(1);
}

function formatSignString(reqMethod, endpoint, path, strParam){
    let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
    return strSign;
}

function sha1(secretKey, strsign){
    let signMethodMap = {'HmacSHA1': 'sha1'};
    let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
    return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
    let strParam = "";
    let keys = Object.keys(params);
    keys.sort();
    for (let k in keys) {
        //k = k.replace(/_/g, '.');
        strParam += ("&" + keys[k] + "=" + params[keys[k]]);
    }
    return strParam
}

function main(){
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
```



```
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768 // 时间戳 2016-06-06 12:02:48, 此参数作为示例, 以实际为准
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886 // 随机正整数
//const nonce = Math.round(Math.random() * 65535)

let params = {};
params['Action'] = Action;
params['InstanceIds.0'] = 'ins-09dx96dg';
params['Limit'] = 20;
params['Offset'] = 0;
params['Nonce'] = Nonce;
params['Region'] = Region;
params['SecretId'] = SECRET_ID;
params['Timestamp'] = Timestamp;
params['Version'] = Version;

// 1. 对参数排序, 并拼接请求字符串
strParam = sort_params(params)

// 2. 拼接签名原字符串
const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)

// 3. 生成签名串
params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])

// 4. 进行url编码并拼接请求url
// const req_url = get_req_url(params, endpoint)
// console.log(params['Signature'])
// console.log(req_url)
}
main()
```

返回结果

最近更新时间：2024-03-12 01:56:30

云 API 3.0 接口默认返回 JSON 数据，返回非 JSON 格式的接口会在文档中做出说明。返回 JSON 数据时最大限制为 50 MB，如果返回的数据超过最大限制，请求会失败并返回内部错误。请根据接口文档中给出的过滤功能（例如时间范围）或者分页功能，控制返回数据不要过大。

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为 200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是 200，而不是 401。

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系 [腾讯云客服](#) 或 [提交工单](#)，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系 [腾讯云客服](#) 或 [提交工单](#)，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码。完整的错误码列表请参考本产品“API 文档”目录下的“错误码”页面。

参数类型

最近更新时间：2022-08-10 06:54:08

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

- String: 字符串。
- Integer: 整型，上限为无符号64位整数。SDK 3.0 不同编程语言支持的类型有所差异，建议以所使用编程语言的最大整型定义，例如 Golang 的 `uint64`。
- Boolean: 布尔型。
- Float: 浮点型。
- Double: 双精度浮点型。
- Date: 字符串，日期格式。例如：2022-01-01。
- Timestamp: 字符串，时间格式。例如：2022-01-01 00:00:00。
- Timestamp ISO8601: ISO 8601 是由国际标准化组织（International Organization for Standardization，ISO）发布的关于日期和时间格式的国际标准，对应国标《GB/T 7408-2005数据元和交换格式信息交换日期和时间表示法》。建议以所使用编程语言的标准库进行格式解析。例如：2022-01-01T00:00:00+08:00。
- Binary: 二进制内容，需要以特定协议请求和解析。

白板板书生成相关接口

创建白板板书生成任务

最近更新时间：2025-05-27 02:06:06

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

创建白板板书生成任务, 在任务结束后, 如果提供了回调地址, 将通过回调地址通知板书生成结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateSnapshotTask。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Whiteboard	是	SnapshotWhiteboard	白板相关参数
SdkAppId	是	Integer	白板房间 SdkAppId 示例值：1400124720
RoomId	是	Integer	白板房间号 示例值：1234
CallbackURL	否	String	白板板书生成结果通知回调地址 示例值：http://xxx
COS	否	SnapshotCOS	白板板书文件 COS 存储参数，不填默认存储在公共存储桶，公共存储桶的数据仅保存 3天
SnapshotMode	否	String	白板板书生成模式，默认为 AllMarks。取值说明如下： AllMarks - 全量模式，即对于客户端每一次调用 addSnapshotMark 接口打上的白板板书生成标志全部都会生成对应的白板板书图片。 LatestMarksOnly - 单页去重模式，即对于客户端在同一页白板上多次调用 addSnapshotMark 打上的白板板书生成标志仅保留最新一次标志来生成对应白板页的白板板书图片。 (注意：LatestMarksOnly 模式只有客户端使用v2.6.8及以上版本的白板SDK调用 addSnapshotMark 时才生效，否则即使在调用本API是指定了

参数名称	必选	类型	描述
			LatestMarksOnly 模式，服务后台会使用默认的 AllMarks 模式生成白板板书) 示例值：AllMarks

3. 输出参数

参数名称	类型	描述
TaskID	String	白板板书生成任务ID，只有任务创建成功的时候才会返回此字段 示例值：d1806f20-25b8-4c30-8176-c0832bf84e02
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建白板板书生成任务

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateSnapshotTask
<公共请求参数>

{
  "COS": {
    "TargetDir": "snapshot/",
    "Region": "ap-guangzhou",
    "Bucket": "test-123070000",
    "Domain": "xxx.file.com",
    "Uin": 130001358
  },
  "Whiteboard": {
    "Width": 1280,
    "InitParams": "{\"ratio\":\"16:9\"}",
    "Height": 720
  },
  "RoomId": 12345,
  "SdkAppId": 1400123480,
  "CallbackURL": "http://xx/callback"
}
```

输出示例

```
{
  "Response": {
    "TaskID": "d1806f20-25b8-4c30-8176-c0832bf84e02",
    "RequestId": "fe36df20-25b8-4c30-8176-cff28afacb25"
  }
}
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceUnavailable	资源不可用。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

获取白板板书生成任务信息

最近更新时间：2025-05-27 02:06:05

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

获取指定白板板书生成任务信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSnapshotTask。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TaskID	是	String	查询任务ID 示例值：f2e1728c-6e87-4481-abe1-cde8542a5a10
SdkAppId	是	Integer	任务SdkAppId 示例值：1400124720

3. 输出参数

参数名称	类型	描述
TaskID	String	任务ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：f2e1728c-6e87-4481-abe1-cde8542a5a10
Status	String	任务状态 Running – 任务执行中 Finished – 任务已结束 注意：此字段可能返回 null，表示取不到有效值。 示例值：Finished
CreateTime	Integer	任务创建时间，单位s 注意：此字段可能返回 null，表示取不到有效值。 示例值：1620982173
FinishTime	Integer	任务完成时间，单位s 注意：此字段可能返回 null，表示取不到有效值。 示例值：1621082195

参数名称	类型	描述
Result	SnapshotResult	任务结果信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取任务信息

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeSnapshotTask
<公共请求参数>

{
  "SdkAppId": 1400127230,
  "TaskID": "f2e1728c-6e87-4481-abe1-cde8542a5a10"
}
```

输出示例

```
{
  "Response": {
    "CreateTime": 1626249427,
    "FinishTime": 1626250455,
    "RequestId": "38d75752-85e0-4931-9837-d41be5409a0a",
    "Result": {
      "ErrorCode": "OK",
      "ErrorMessage": "",
      "Snapshots": [
        "http://tiw-snapshot-1259648581.cos.ap-guangzhou.myqcloud.com/880520/f2e1728c-6e87-4481-abe1-cde8542a5a10/53255368873480.png",
        "http://tiw-snapshot-1259648581.cos.ap-guangzhou.myqcloud.com/880520/f2e1728c-6e87-4481-abe1-cde8542a5a10/53255373174978.png",
        "http://tiw-snapshot-1259648581.cos.ap-guangzhou.myqcloud.com/880520/f2e1728c-6e87-4481-abe1-cde8542a5a10/53255375472670.png"
      ],
      "Total": 3
    },
    "Status": "Finished",
    "TaskID": "f2e1728c-6e87-4481-abe1-cde8542a5a10"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceUnavailable	资源不可用。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

白板推流相关接口

查询白板推流任务

最近更新时间：2025-05-27 02:06:07

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询推流任务状态与结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeWhiteboardPush。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
TaskId	是	String	白板推流任务Id 示例值：ghucnligtggtvk2624mb

3. 输出参数

参数名称	类型	描述
FinishReason	String	推流结束原因， – AUTO: 房间内长时间没有音视频上行及白板操作导致自动停止推流 – USER_CALL: 主动调用了停止推流接口 – EXCEPTION: 推流异常结束 示例值：USER_CALL
TaskId	String	需要查询结果的白板推流任务Id 示例值：ghucnligtggtvk2624mb
Status	String	推流任务状态 – PREPARED: 表示推流正在准备中（进房/启动推流服务等操作） – PUSHING: 表示推流已开始 – STOPPED: 表示推流已停止 示例值：STOPPED
RoomId	Integer	房间号

参数名称	类型	描述
		示例值：12345
GroupId	String	白板的群组 Id 示例值：12345
PushUserId	String	推流用户 Id 示例值：tic_push_user_12345_abcde
PushStartTime	Integer	实际开始推流时间，Unix 时间戳，单位秒 示例值：1611128339
PushStopTime	Integer	实际停止推流时间，Unix 时间戳，单位秒 示例值：1611128339
ExceptionCnt	Integer	推流过程中出现异常的次数 示例值：0
IMSyncTime	Integer	白板推流首帧对应的IM时间戳，可用于录制回放时IM聊天消息与白板推流视频进行同步对时。 示例值：0
Backup	String	备份推流任务结果信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：{}
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询白板推流任务

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeWhiteboardPush
&SdkAppId=1400000001
&TaskId=ghucnliggtgtvk2624mb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "ExceptionCnt": 0,
    "FinishReason": "USER_CALL",
    "GroupId": "880528",
    "IMSyncTime": 0,
    "PushStartTime": 1610591726,
    "PushStopTime": 1610591759,
    "PushUserId": "tic_push_user_880528_test1",
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "RoomId": 880528,
    "Status": "STOPPED",
    "Backup": "",
```

```
"TaskId": "g42lfmu5uc3lpoqrqvbb"
}
```

示例2 查询推流失败的白板推流任务

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeWhiteboardPush
&SdkAppId=1400000001
&TaskId=ghucnligtgtvkv2624mb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Error": {
      "Code": "FailedOperation.WhiteboardPush",
      "Message": "code: 40006, msg: 2021-01-13 22:34:42, TRTC onError, code: -70003, msg: The UserSig in use is illegal. Please regenerate UserSig through official API. (FYI: https://cloud.tencent.com/document/product/269/32688 )\n"
    },
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.WhiteboardPush	白板推流失败，具体请参考错误描述。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询白板推流回调地址

最近更新时间：2025-05-27 02:06:07

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询白板推流回调地址

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeWhiteboardPushCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId 示例值：1400000001

3. 输出参数

参数名称	类型	描述
Callback	String	白板推流事件回调地址，如果未设置回调地址，该字段为空字符串 示例值：https://example.com/online/callback
CallbackKey	String	白板推流回调鉴权密钥 示例值：6vg9G7Fd
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询白板推流回调地址

查询白板推流回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeWhiteboardPushCallback
&SdkAppId=1400000001
```

&<公共请求参数>

输出示例

```
{
  "Response": {
    "Callback": "https://example.com/online/callback",
    "CallbackKey": "6vg9G7Fd",
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。

错误码	描述
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置白板推流回调地址

最近更新时间：2025-05-27 02:06:06

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置白板推流回调地址，回调数据格式请参考文档：<https://cloud.tencent.com/document/product/1137/40257>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetWhiteboardPushCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Callback	是	String	白板推流任务结果回调地址，如果传空字符串会删除原来的回调地址配置，回调地址仅支持 http或https协议，即回调地址以http://或https://开头。回调数据格式请参考文档： https://cloud.tencent.com/document/product/1137/40257 示例值： https://example.com/online/callback

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置白板推流回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetWhiteboardPushCallback
&SdkAppId=1400000001
&Callback=https://example.com/online/callback
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置白板推流回调密钥

最近更新时间：2025-05-27 02:06:06

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置白板推流回调鉴权密钥，回调鉴权方式请参考文档：<https://cloud.tencent.com/document/product/1137/40257>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetWhiteboardPushCallbackKey。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId
CallbackKey	是	String	设置白板推流回调鉴权密钥，最长64字符，如果传入空字符串，那么删除现有的鉴权回调密钥。回调鉴权方式请参考文档： https://cloud.tencent.com/document/product/1137/40257

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置白板推流回调鉴权密钥

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetWhiteboardPushCallbackKey
&SdkAppId=1400000001
&CallbackKey=6vg9G7Fd
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppld	SdkAppld不存在或者SdkAppld与当前腾讯云账号不对应。

结束白板推流

最近更新时间：2025-05-27 02:06:06

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

停止白板推流任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StopWhiteboardPush。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId
TaskId	是	String	需要停止的白板推流任务 Id

3. 输出参数

参数名称	类型	描述
Backup	String	备份任务相关参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 停止白板推流

输入示例

```
https://tiw.tencentcloudapi.com/?Action=StopWhiteboardPush
&SdkAppId=1400000001
&TaskId=07b4c70bpafm1b4majmb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Backup": "{ \"RequestId\": \"87c334b8-dbf6-4013-8d65-9a9ea60eff27\" }",
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.SdkAppIdNotFound	SdkAppId不存在或格式错误。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

错误码	描述
UnsupportedOperation.TaskHasAlreadyStopped	任务结束，不能执行指定操作。

开始白板推流

最近更新时间：2025-05-27 02:06:06

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

发起一个白板推流任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：StartWhiteboardPush。
Version	是	String	公共参数，本接口取值：2019-09-19。
Region	是	String	公共参数，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
RoomId	是	Integer	需要推流的白板房间号，取值范围: (1, 4294967295)。 1. 在没有指定GroupId的情况下，白板推流默认以RoomId的字符串表达形式作为IM群组ID（比如RoomId为1234，则IM群组ID为"1234"），并加群进行信令同步，请在开始推流前确保相应IM群组已创建完成，否则会导致推流失败。 2. 在没有指定TRTCRoomId和TRTCRoomIdStr的情况下，默认会以RoomId作为白板流进行推流的TRTC房间号。 示例值：1268
PushUserId	是	String	用于白板推流服务进入白板房间的用户ID。在没有额外指定IMAuthParam和TRTCAuthParam的情况下，这个用户ID同时会用于IM登录、IM加群、TRTC进房推流等操作。 用户ID最大长度不能大于60个字节，该用户ID必须是一个单独的未同时在其他地方使用的用户ID，白板推流服务使用这个用户ID进入房间进行白板音视频推流，若该用户ID和其他地方同时在使用用户ID重复，会导致白板推流服务与其他使用场景账号互踢，影响正常推流。 示例值：tic_push_1268_xxx
PushUserSig	是	String	与PushUserId对应的IM签名(usersig)。 示例值：eJwtjb0OgjAAhN*IK4a0tcVC4iDGkIg-g7K4NEiLViLUUsFgfHcRmS733eXuDY6b
Whiteboard	否	Whiteboard	白板参数，例如白板宽高、背景颜色等

参数名称	必选	类型	描述
AutoStopTimeout	否	Integer	自动停止推流超时时间，单位秒，取值范围[300, 259200]，默认为1800秒。 当白板超过设定时间没有操作的时候，白板推流服务会自动停止白板推流。 示例值：300
AutoManageBackup	否	Boolean	对主白板推流任务进行操作时，是否同时同步操作备份任务 示例值：false
Backup	否	WhiteboardPushBackupParam	备份白板推流相关参数。 指定了备份参数的情况下，白板推流服务会在房间内新增一路白板画面视频流，即同一个房间内会有两路白板画面推流。
PrivateMapKey	否	String	TRTC高级权限控制参数，如果在实时音视频开启了高级权限控制功能，必须提供PrivateMapKey才能保证正常推流。
VideoFPS	否	Integer	白板推流视频帧率，取值范围[0, 30]，默认20fps 示例值：15
VideoBitrate	否	Integer	白板推流码率，取值范围[0, 2000]，默认1200kbps。 这里的码率设置是一个参考值，实际推流的时候使用的是动态码率，所以真实码率不会固定为指定值，会在指定值附近波动。 示例值：1200
AutoRecord	否	Boolean	在实时音视频云端录制模式选择为 指定用户录制 模式的时候是否自动录制白板推流。 默认在实时音视频的云端录制模式选择为 指定用户录制 模式的情况下，不会自动进行白板推流录制，如果希望进行白板推流录制，请将此参数设置为true。 如果实时音视频的云端录制模式选择为 全局自动录制 模式，可忽略此参数。 示例值：false
UserDefinedRecordId	否	String	指定白板推流这路流在音视频云端录制中的RecordID，指定的RecordID会用于填充实时音视频云端录制完成后的回调消息中的 "userdefinerecordid" 字段内容，便于您更方便的识别录制回调，以及在点播媒体资源管理中查找相应的录制视频文件。 限制长度为64字节，只允许包含大小写英文字母（a-zA-Z）、数字（0-9）及下划线和连词符。 此字段设置后，不管AutoRecord字段取值如何，都将自动进行白板推流录制。 默认RecordId生成规则如下： urlencode(SdkAppID_RoomID_PushUserID) 例如： SdkAppID = 12345678，RoomID = 12345， PushUserID = push_user_1 那么：RecordId = 12345678_12345_push_user_1 示例值：12345678_12345_push_user_1

参数名称	必选	类型	描述
AutoPublish	否	Boolean	在实时音视频旁路推流模式选择为指定用户旁路模式的时候，是否自动旁路白板推流。 默认在实时音视频的旁路推流模式选择为 指定用户旁路 模式的情况下，不会自动旁路白板推流，如果希望旁路白板推流，请将此参数设置为true。 如果实时音视频的旁路推流模式选择为 全局自动旁路 模式，可忽略此参数。 示例值：false
UserDefinedStreamId	否	String	指定实时音视频在旁路白板推流这路流时的StreamID，设置之后，您就可以在腾讯云直播 CDN 上通过标准直播方案（FLV或HLS）播放该用户的音视频流。 限制长度为64字节，只允许包含大小写英文字母（a-zA-Z）、数字（0-9）及下划线和连词符。 此字段设置后，不管AutoPublish字段取值如何，都将自动旁路白板推流。 默认StreamID生成规则如下： urlencode(SdkAppID_RoomID_PushUserID_main) 例如： SdkAppID = 12345678, RoomID = 12345, PushUserID = push_user_1 那么：StreamID = 12345678_12345_push_user_1_main 示例值：12345678_12345_push_user_1_main
ExtraData	否	String	内部参数，不需要关注此参数
TRTCRoomId	否	Integer	TRTC数字类型房间号，取值范围: (1, 4294967295)。 在同时指定了RoomId与TRTCRoomId的情况下，优先使用TRTCRoomId作为白板流进行推流的TRTC房间号。 当指定了TRTCRoomIdStr的情况下，此字段将被忽略。 示例值：1368
TRTCRoomIdStr	否	String	TRTC字符串类型房间号。 在指定了TRTCRoomIdStr的情况下，会优先使用TRTCRoomIdStr作为白板流进行推流的TRTC房间号。 示例值：wbpush-12345
IMAuthParam	否	AuthParam	IM鉴权信息参数，用于IM鉴权。 当白板信令所使用的IM应用与白板应用的SdkAppId不一致时，可以通过此参数提供对应IM应用鉴权信息。 如果提供了此参数，白板推流服务会优先使用此参数指定的SdkAppId作为白板信令的传输通道，否则使用公共参数中的SdkAppId作为白板信令的传输通道。
TRTCAuthParam	否	AuthParam	TRTC鉴权信息参数，用于TRTC进房推流鉴权。 当需要推流到的TRTC房间所对应的TRTC应用与白板应用的SdkAppId不一致时，可以通过此参数提供对应的TRTC应用鉴

参数名称	必选	类型	描述
			权信息。 如果提供了此参数，白板推流服务会优先使用此参数指定的 SdkAppId 作为白板推流的目标 TRTC 应用，否则使用公共参数中的 SdkAppId 作为白板推流的目标 TRTC 应用。
TRTCEnterRoomMode	否	String	指定白板推流时推流用户进 TRTC 房间的进房模式。默认为 TRTCApSceneVideoCall TRTCApSceneVideoCall – 视频通话场景，即绝大多数时间都是两人或两人以上视频通话的场景，内部编码器和网络协议优化侧重流畅性，降低通话延迟和卡顿率。 TRTCApSceneLIVE – 直播场景，即绝大多数时间都是一人直播，偶尔有多人视频互动的场景，内部编码器和网络协议优化侧重性能和兼容性，性能和清晰度表现更佳。 示例值：TRTCApSceneVideoCall
GroupId	否	String	白板进行信令同步的 IM 群组 ID。 在没有指定 GroupId 的情况下，白板推流服务将使用 RoomId 的字符串形式作为同步白板信令的 IM 群组 ID。 在指定了 GroupId 的情况下，白板推流将优先 GroupId 作为同步白板信令的群组 ID。请在开始推流前确保指定的 IM 群组已创建完成，否则会导致推流失败。 示例值：1368

3. 输出参数

参数名称	类型	描述
TaskId	String	推流任务Id 示例值：0c69baivvc3lp8t4f80b
Backup	String	备份任务结果参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 开始白板推流

创建一个白板推流任务

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StartWhiteboardPush
<公共请求参数>

{
  "PushUserId": "tic_push_user_1203_141551",
  "PushUserSig": "usersig_of_<tic_push_user_1203_141551>",
```

```
"RoomId": 1203,
"SdkAppId": 1400000001
}
```

输出示例

```
{
  "Response": {
    "Backup": "{\"RequestId\":\"8e9b2bc4-ec7f-46cd-823b-66676e2375f8\",\"TaskId\":\"0521fmu5uc3lp0rrqvbb\"}\",
    \"RequestId\": \"eac6b301-a322-493a-8e36-83b295459397\",
    \"TaskId\": \"bj0mt2l23osdj300h130\"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.InvalidExtra	额外指定的特殊功能不存在。

错误码	描述
InvalidParameter.RecordParameter	实时录制参数格式不正确。
InvalidParameter.SdkAppIdNotFound	SdkAppId不存在或格式错误。
LimitExceeded.TaskConcurrency	转码或录制任务并发数量超过限制，请参考错误描述或稍后重试。
ResourceInUse.RecordUserId	实时录制任务录制用户已被其他录制任务使用。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

录制视频生成（内测中）相关接口

创建录制视频生成任务

最近更新时间：2025-06-30 02:33:04

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

创建视频生成任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateVideoGenerationTask。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
OnlineRecordTaskId	是	String	实时录制任务的TaskId 示例值：bj0mt2l23osdj300hl30
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Whiteboard	否	Whiteboard	视频生成的白板参数，例如白板宽高等。 此参数与开始录制接口提供的Whiteboard参数互斥，在本接口与开始录制接口都提供了Whiteboard参数时，优先使用本接口指定的Whiteboard参数进行视频生成，否则使用开始录制接口提供的Whiteboard参数进行视频生成。 示例值：1280*960
Concat	否	Concat	视频拼接参数 此参数与开始录制接口提供的Concat参数互斥，在本接口与开始录制接口都提供了Concat参数时，优先使用本接口指定的Concat参数进行视频拼接，否则使用开始录制接口提供的Concat参数进行视频拼接。 示例值：否
MixStream	否	MixStream	视频生成混流参数 此参数与开始录制接口提供的MixStream参数互斥，在本接口与开始录制接口都提供了MixStream参数时，优先使用本接口指定的MixStream参数进行视频混流，否则使用开始录制接口提供的MixStream参数进行视频拼接混流。 示例值：否

参数名称	必选	类型	描述
RecordControl	否	RecordControl	视频生成控制参数，用于更精细地指定需要生成哪些流，某一路流是否禁用音频，是否只录制小画面等 此参数与开始录制接口提供的RecordControl参数互斥，在本接口与开始录制接口都提供了RecordControl参数时，优先使用本接口指定的RecordControl参数进行视频生成控制，否则使用开始录制接口提供的RecordControl参数进行视频拼生成控制。 示例值：否
ExtraData	否	String	内部参数 示例值：""

3. 输出参数

参数名称	类型	描述
TaskId	String	视频生成的任务Id 示例值：bj0mt2l23osdj300hl30
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 开始录制视频生成

输入示例

```
https://tiw.tencentcloudapi.com/?Action=CreateVideoGenerationTask
&SdkAppId=1400000001
&OnlineRecordTaskId=bj0mt2l23osdj300hl30
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "TaskId": "g7khk7f809vterp942mb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.InvalidExtra	额外指定的特殊功能不存在。
InvalidParameter.RecordParameter	实时录制参数格式不正确。
InvalidParameter.SdkAppIdNotFound	SdkAppId不存在或格式错误。
LimitExceeded.TaskConcurrency	转码或录制任务并发数量超过限制，请参考错误描述或稍后重试。
ResourceInUse.RecordUserId	实时录制任务录制用户已被其他录制任务使用。
ResourceUnavailable.InterfaceUnauthorized	资源不可用，接口未授权。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询录制视频生成任务

最近更新时间：2025-05-27 02:06:09

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询录制视频生成任务状态与结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeVideoGenerationTask。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId
TaskId	是	String	录制视频生成的任务Id

3. 输出参数

参数名称	类型	描述
GroupId	String	任务对应的群组Id
RoomId	Integer	任务对应的房间号
TaskId	String	任务的Id
Progress	Integer	已废弃
Status	String	录制视频生成任务状态 - QUEUED: 正在排队 - PROCESSING: 正在生成视频 - FINISHED: 生成视频结束（成功完成或失败结束，可以通过错误码和错误信息进一步判断）
TotalTime	Integer	回放视频总时长,单位：毫秒
VideoInfos	VideoInfo	已废弃，请使用VideoInfoList参数
VideoInfoList	Array of VideoInfo	录制视频生成视频列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获

参数名称	类型	描述
		得 RequestId)。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询录制视频生成

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeVideoGenerationTask
&SdkAppId=1400000001
&TaskId=ghucnliggtgtvk2624mb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "GroupId": "900822",
    "Progress": 100,
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "RoomId": 900822,
    "Status": "FINISHED",
    "TaskId": "00e1jv8ve0jcqk7da91b",
    "TotalTime": 156750,
    "VideoInfoList": [
      {
        "UserId": "",
        "VideoDuration": 156750,
        "VideoFormat": "mp4",
        "VideoId": "5285890792776509100",
        "VideoPlayTime": 0,
        "VideoSize": 318384,
        "VideoType": 2,
        "Width": 1024,
        "Height": 768,
        "VideoUrl": "http://1257240443.vod2.myqcloud.com/cc35b442vodcq1257240443/video.mp4"
      }
    ],
    "VideoInfos": {
      "VideoType": 0,
      "VideoUrl": "已废弃, 请使用VideoInfoList",
      "UserId": "xx",
      "VideoId": "xx",
      "VideoSize": 0,
      "VideoDuration": 0,
      "Width": 0,
      "Height": 0,
      "VideoPlayTime": 0,
    }
  }
}
```

```
"VideoFormat": "xx"
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.Record	录制失败，具体请参考错误描述。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询录制视频生成回调地址

最近更新时间：2025-05-27 02:06:09

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询录制视频生成回调地址

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeVideoGenerationTaskCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId

3. 输出参数

参数名称	类型	描述
Callback	String	录制视频生成回调地址
CallbackKey	String	录制视频生成回调鉴权密钥
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询录制视频生成回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeVideoGenerationTaskCallback
&SdkAppId=1400000001
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Callback": "https://example.com/offline/callback",
    "CallbackKey": "6vg9G7Fd",
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置录制视频生成回调地址

最近更新时间：2025-05-27 02:06:09

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置录制视频生成回调地址

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetVideoGenerationTaskCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId
Callback	是	String	课后录制任务结果回调地址，如果传空字符串会删除原来的回调地址配置，回调地址仅支持 http或https协议，即回调地址以http://或https://开头

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置录制视频生成回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetVideoGenerationTaskCallback
&SdkAppId=1400000001
&Callback=https://example.com/videoGeneration/callback
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppld	SdkAppld不存在或者SdkAppld与当前腾讯云账号不对应。

设置录制视频生成回调密钥

最近更新时间：2025-05-27 02:06:09

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置视频生成回调鉴权密钥

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetVideoGenerationTaskCallbackKey。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId
CallbackKey	是	String	设置视频生成回调鉴权密钥，最长64字符，如果传入空字符串，那么删除现有的鉴权回调密钥

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置视频生成回调鉴权密钥

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetVideoGenerationTaskCallbackKey
&SdkAppId=1400000001
&CallbackKey=6vg9G7Fd
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppld	SdkAppld不存在或者SdkAppld与当前腾讯云账号不对应。

文档转码相关接口

创建文档转码任务

最近更新时间：2025-05-27 02:06:08

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

创建一个文档转码任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateTranscode。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Url	是	String	经过URL编码后的转码文件地址。URL 编码会将字符转换为可通过因特网传输的格式，比如文档地址为 http://example.com/测试.pdf，经过URL编码之后为 http://example.com/%E6%B5%8B%E8%AF%95.pdf。为了提高URL解析的成功率，请对URL进行编码。 示例值：https://path/to/document.ppt
IsStaticPPT	否	Boolean	是否为静态PPT，默认为False； 如果IsStaticPPT为False，后缀名为.ppt或.pptx的文档会动态转码成HTML5页面，其他格式的文档会静态转码成图片；如果IsStaticPPT为True，所有格式的文档会静态转码成图片； 示例值：False
MinResolution	否	String	注意: 该参数已废弃, 请使用最新的 云API SDK ，使用 MinScaleResolution字段传递分辨率 转码后文档的最小分辨率，不传、传空字符串或分辨率格式错误则使用文档原分辨率 示例：1280x720，注意分辨率宽高中间为英文字母"xyz"的"x" 示例值：960x540
ThumbnailResolution	否	String	动态PPT转码可以为文件生成该分辨率的缩略图，不传、传空字符串或分辨率格式错误则不生成缩略图，分辨率格式同

参数名称	必选	类型	描述
			MinResolution 示例值：960x540
CompressFileType	否	String	转码文件压缩格式，不传、传空字符串或不是指定的格式则不生成压缩文件，目前支持如下压缩格式： zip：生成.zip压缩包 tar.gz：生成.tar.gz压缩包 示例值：zip
ExtraData	否	String	内部参数 示例值：空
Priority	否	String	文档转码优先级，只有对于PPT动态转码生效，支持填入以下值： - low: 低优先级转码，对于动态转码，能支持500MB（下载超时时间10分钟）以及2000页文档，但资源有限可能会有比较长时间的排队，请酌情使用该功能。 - 不填表示正常优先级转码，支持200MB文件（下载超时时间2分钟），500页以内的文档进行转码 注意：对于PDF等静态文件转码，无论是正常优先级或者低优先级，最大只能支持200MB 示例值：空
MinScaleResolution	否	String	转码后文档的最小分辨率，不传、传空字符串或分辨率格式错误则使用文档原分辨率。 分辨率越高，效果越清晰，转出来的图片资源体积会越大，课件加载耗时会变长，请根据实际使用场景配置此参数。 示例：1280x720，注意分辨率宽高中间为英文字母"xyz"的"x" 示例值：960x540
AutoHandleUnsupportedElement	否	Boolean	此参数仅对动态转码生效。 是否对不支持元素开启自动处理的功能，默认不开启。 true -- 开启 false -- 不开启 当设置为true时，可配合 AutoHandleUnsupportedElementTypes参数使用，具体有哪些不兼容元素类型，可参考 AutoHandleUnsupportedElementTypes参数的说明。 示例值：true
AutoHandleUnsupportedElementTypes.N	否	Array of Integer	此参数仅在AutoHandleUnsupportedElement参数为true的情况下有效。 指定需要自动处理的不兼容元素类型，默认对所有不兼容的元素进行自动处理。 目前支持检测的不兼容元素类型及对应的自动处理方式如下： 0: 不支持的墨迹类型 -- 自动处理方式：移除墨迹 1: 自动翻页

参数名称	必选	类型	描述
			-- 自动处理方式：移除自动翻页设置，并修改为单击切换 2: 已损坏音视频 -- 自动处理方式：移除对损坏音视频的引用 3: 不可访问资源 -- 自动处理方式：移除对不可访问的资源的引用 4: 只读文件 -- 自动处理方式：移除只读设置 5: 不支持的元素编辑锁定状态 -- 自动处理方式：移除锁定状态 6: 可能有兼容问题的字体 -- 自动处理方式：不支持处理 7: 设置了柔化边缘的GIF图片 -- 自动处理方式：移除柔化边缘设置 8: 存在不兼容的空格下划线 -- 自动处理方式：通过调整空格下划线前后文本的字体语言体系，保证空格下划线表现正常 9: 存在设置了分段动画的数学公式和文本混合内容 -- 自动处理方式：不支持处理 10: 存在设置了分段动画的渐变色文本 -- 自动处理方式：不支持处理 11: 存在不兼容的分散对齐方式 -- 自动处理方式：不支持处理 12: 存在不兼容的多倍行距设置 -- 自动处理方式：不支持处理 13: 存在带有特殊符号内容的datetime类型的a:fld标签元素 -- 自动处理方式：a:fld标签替换为普通文本 示例值：[0, 1, 2]
ExcelParam	否	ExcelParam	Excel表格转码参数，可设置转码时表格纸张大小及纸张方向等参数（仅对转码文件为Excel表格文件的静态转码任务生效）

3. 输出参数

参数名称	类型	描述
TaskId	String	文档转码任务的唯一标识Id，用于查询该任务的进度以及转码结果 示例值：g6ls63ps49vteb8bk1mb
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 创建PPT动态转码任务

创建PPT动态转码任务，并生成分辨率为960x540的缩略图

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateTranscode
<公共请求参数>

{
  "Url": "https://path/to/document.ppt",
  "ThumbnailResolution": "960x540",
  "SdkAppId": 1400000001
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "TaskId": "g61s63ps49vteb8bk1mb"
  }
}
```

示例2 创建PPT静态转码任务

创建PPT静态转码任务

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateTranscode
<公共请求参数>

{
  "Url": "https://path/to/document.ppt",
  "IsStaticPPT": "True",
  "SdkAppId": 1400000001
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "TaskId": "g61s63ps49vteb8bk1mb"
  }
}
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.Transcode	转码失败，具体请参考错误描述或联系客服人员。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.FileFormatUnsupported	文档后缀名对应的格式不支持。
InvalidParameter.PreprocessParameter	转码预处理参数格式不正确。
InvalidParameter.SdkAppIdNotFound	SdkAppId不存在或格式错误。
InvalidParameter.TranscodeParameter	文档转码参数格式不正确。
LimitExceeded.TaskConcurrency	转码或录制任务并发数量超过限制，请参考错误描述或稍后重试。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询文档转码任务

最近更新时间：2025-05-27 02:06:08

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询文档转码任务的执行进度与转码结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTranscode。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
TaskId	是	String	文档转码任务的唯一标识Id 示例值：g0jb42ps49vtebjshilb

3. 输出参数

参数名称	类型	描述
Pages	Integer	文档的总页数 示例值：1
Progress	Integer	转码的当前进度,取值范围为0~100 示例值：100
Resolution	String	文档的分辨率 示例值：793x1122
ResultUrl	String	转码完成后结果的URL 动态转码：PPT转动态H5的链接 静态转码：文档每一页的图片URL前缀，比如，该URL前缀为 http://example.com/g0jb42ps49vtebjshilb/，那么文档第1页的图片URL为 http://example.com/g0jb42ps49vtebjshilb/1.jpg，其它页以此类推 示例值：https://transcode-result/0agdnligqgtgvtkm65emb/index.html
Status	String	任务的当前状态 - QUEUED: 正在排队等待转换 - PROCESSING: 转换中 - FINISHED: 转换完成 示例值：FINISHED

参数名称	类型	描述
TaskId	String	转码任务的唯一标识Id 示例值：0agdnligqtgtvkm65emb
Title	String	文档的文件名 示例值：59378.docx
ThumbnailUrl	String	缩略图URL前缀，比如，该URL前缀为http://example.com/g0jb42ps49vtebjshilb/，那么动态PPT第1页的缩略图URL为 http://example.com/g0jb42ps49vtebjshilb/1.jpg，其它页以此类推 如果发起文档转码请求参数中带了ThumbnailResolution参数，并且转码类型为动态转码，该参数不为空，其余情况该参数为空字符串 示例值：https://transcode-thumbnail/0agdnligqtgtvkm65emb/
ThumbnailResolution	String	动态转码缩略图生成分辨率 示例值：793x1122
CompressFileUrl	String	转码压缩文件下载的URL，如果发起文档转码请求参数中CompressFileType为空或者不是支持的压缩格式，该参数为空字符串
ResourceListUrl	String	资源清单文件下载URL(内测体验) 注意：此字段可能返回 null，表示取不到有效值。
Ext	String	文档制作方式(内测体验) 注意：此字段可能返回 null，表示取不到有效值。 示例值：office
CreateTime	Integer	文档转码任务创建时间，单位s 注意：此字段可能返回 null，表示取不到有效值。 示例值：1688974545
AssignTime	Integer	文档转码任务分配时间，单位s 注意：此字段可能返回 null，表示取不到有效值。 示例值：1688974545
FinishedTime	Integer	文档转码任务完成时间，单位s 注意：此字段可能返回 null，表示取不到有效值。 示例值：1688974545
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询文档动态转码任务

查询文档动态转码任务结果

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeTranscode
<公共请求参数>

{
```

```
"TaskId": "g0jb42ps49vtebjshilb",
"SdkAppId": "1400000001"
}
```

输出示例

```
{
  "Response": {
    "Pages": 1,
    "Progress": 100,
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "Resolution": "793x1122",
    "ResultUrl": "https://transcode-result/0agdnligqgtgvkm65emb/index.html",
    "Status": "FINISHED",
    "TaskId": "0agdnligqgtgvkm65emb",
    "Title": "59378.docx",
    "ThumbnailUrl": "https://transcode-thumbnail/0agdnligqgtgvkm65emb/",
    "ThumbnailResolution": "793x1122",
    "CompressFileUrl": "",
    "ResourceListUrl": "https://transcode-result/0agdnligqgtgvkm65emb/resources.txt",
    "Ext": "Office"
  }
}
```

示例2 查询文档静态转码任务

查询文档静态转码任务结果

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeTranscode
<公共请求参数>

{
  "TaskId": "g0jb42ps49vtebjshilb",
  "SdkAppId": "1400000001"
}
```

输出示例

```
{
  "Response": {
    "Pages": 1,
    "Progress": 100,
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "Resolution": "793x1122",
```

```
"ResultUrl": "https://transcode-result/0agdnligqgtvkm65emb/",
"Status": "FINISHED",
"TaskId": "0agdnligqgtvkm65emb",
"Title": "59378.docx",
"ThumbnailUrl": "",
"ThumbnailResolution": "",
"CompressFileUrl": "",
"ResourceListUrl": "https://transcode-result/0agdnligqgtvkm65emb/resources.txt",
"Ext": "Office"
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.FileDownloadFail	文档下载失败，请检查请求参数中URL是否正确，或者如果您使用其他的文件存储服务，请检查文件存储服务的上传带宽，文档转码服务仅提供1分钟的下载时间，如果下载不成功本次的转码请求将以失败终止。
FailedOperation.FileFormatError	文档格式错误，不支持转换只读文档或者已加密的文档。
FailedOperation.FileOpenFail	文档打开失败，请检查提交转码的文档是否加密或有其他格式问题。

错误码	描述
FailedOperation.FileUploadFail	转码后上传结果失败，请稍后重试。
FailedOperation.Transcode	转码失败，具体请参考错误描述或联系客服人员。
FailedOperation.TranscodeServerError	转码服务出现内部错误，请稍后重试或联系客服人员。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.FileFormatUnsupported	文档后缀名对应的格式不支持。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
InvalidParameter.TranscodeParameter	文档转码参数格式不正确。
InvalidParameter.UrlFormatError	文档下载Url格式错误，请检查请求参数里的Url。
LimitExceeded.TranscodePagesLimitation	超过文档最大页数限制，目前不支持超过500页的文件转码，如有特殊需求，请联系客服人员。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询文档转码回调地址

最近更新时间：2025-05-27 02:06:07

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询文档转码回调地址

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTranscodeCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId 示例值：1400000001

3. 输出参数

参数名称	类型	描述
Callback	String	文档转码回调地址 示例值：https://example.com/transcode/callback
CallbackKey	String	文档转码回调鉴权密钥 示例值：6vg9G7Fd
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询文档转码回调地址

查询文档转码回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeTranscodeCallback
&SdkAppId=1400000001
```

&<公共请求参数>

输出示例

```
{
  "Response": {
    "Callback": "https://example.com/transcode/callback",
    "CallbackKey": "6vg9G7Fd",
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。

错误码	描述
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置文档转码回调地址

最近更新时间：2025-05-27 02:06:07

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置文档转码回调地址，回调数据格式请参考文档：<https://cloud.tencent.com/document/product/1137/40260>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetTranscodeCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Callback	是	String	文档转码进度回调地址，如果传空字符串会删除原来的回调地址配置，回调地址仅支持http或https协议，即回调地址以http://或https://开头。 回调数据格式请参考文档： https://cloud.tencent.com/document/product/1137/40260 示例值： https://example.com/transcode/callback

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 设置文档转码回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetTranscodeCallback
&SdkAppId=1400000001
&Callback=https://example.com/transcode/callback
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置文档转码回调密钥

最近更新时间：2025-05-27 02:06:07

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置文档转码回调鉴权密钥，回调鉴权方式请参考文档：<https://cloud.tencent.com/document/product/1137/40257>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetTranscodeCallbackKey。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId
CallbackKey	是	String	设置文档转码回调鉴权密钥，最长64字符，如果传入空字符串，那么删除现有的鉴权回调密钥，回调鉴权方式请参考文档： https://cloud.tencent.com/document/product/1137/40257

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置文档转码回调鉴权密钥

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetTranscodeCallbackKey
&SdkAppId=1400000001
&CallbackKey=6vg9G7Fd
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询PPT检测任务状态

最近更新时间：2025-04-25 02:02:10

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询PPT检测任务的执行进度或结果，支持查询最近半年内的任务结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribePPTCheck。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
TaskId	是	String	任务的唯一标识Id 示例值：g0jb42ps49vtebjshilb

3. 输出参数

参数名称	类型	描述
TaskId	String	任务的唯一标识Id 示例值：0agdnligqgtvkm65emb
IsOK	Boolean	PPT文件是否正常 示例值：false
ResultUrl	String	修复后的PPT URL，只有创建任务时参数AutoHandleUnsupportedElement=true，才返回此参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://xxx/xxx/测试_fixed.ppt
Slides	Array of PPTErrSlide	错误PPT页面列表 注意：此字段可能返回 null，表示取不到有效值。
Status	String	任务的当前状态 - QUEUED: 正在排队等待 - PROCESSING: 执行中 - FINISHED: 执行完成 示例值：FINISHED
Progress	Integer	当前进度,取值范围为0~100 示例值：100

参数名称	类型	描述
Errs	Array of PPTErr	错误列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询PPT检测任务异常

查询PPT检测任务，存在异常

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribePPTCheck
<公共请求参数>

{
  "TaskId": "g0jb42ps49vtebjshilb",
  "SdkAppId": "1400000001"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "bj0mt2l23osdj300h130",
    "IsOK": false,
    "Status": "FINISHED",
    "Progress": 100,
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "ResultUrl": "https://xxx/xxx/测试_fixed.ppt",
    "Slides": [
      {
        "Page": "幻灯片5",
        "Errs": [
          {
            "Name": "Ink 1",
            "Type": 0,
            "Detail": "wps墨迹"
          }
        ]
      }
    ],
    "Errs": [
      {

```

```
"Type": 0,
"Detail": "wps墨迹"
}
]
}
}
```

示例2 查询PPT检测任务无异常

查询PPT检测任务，PPT无异常

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribePPTCheck
<公共请求参数>

{
  "TaskId": "g0j42ps49vtebjshilb",
  "SdkAppId": "1400000001"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "bj0mt2l23osdj300hl30",
    "IsOK": true,
    "Status": "FINISHED",
    "Progress": 100,
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "ResultUrl": null,
    "Slides": null,
    "Errs": null
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.FileDownloadFail	文档下载失败，请检查请求参数中URL是否正确，或者如果您使用其他的文件存储服务，请检查文件存储服务的上传带宽，文档转码服务仅提供1分钟的下载时间，如果下载不成功本次的转码请求将以失败终止。
FailedOperation.FileFormatError	文档格式错误，不支持转换只读文档或者已加密的文档。
FailedOperation.FileOpenFail	文档打开失败，请检查提交转码的文档是否加密或有其他格式问题。
FailedOperation.FileUploadFail	转码后上传结果失败，请稍后重试。
FailedOperation.Preprocess	转码预处理失败，具体请参考错误描述或联系客服人员。
FailedOperation.PreprocessServerError	预处理服务出现内部错误，请稍后重试或联系客户人员。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.FileFormatUnsupported	文档后缀名对应的格式不支持。
InvalidParameter.PreprocessParameter	转码预处理参数格式不正确。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
InvalidParameter.UrlFormatError	文档下载Url格式错误，请检查请求参数里的Url。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppld	SdkAppld不存在或者SdkAppld与当前腾讯云账号不对应。

设置PPT检测任务回调地址

最近更新时间：2025-04-25 02:02:09

1. 接口描述

接口请求域名： tiw.tencentcloudapi.com 。

设置PPT检测任务回调地址，回调数据格式请参考文档：<https://cloud.tencent.com/document/product/1137/40260#c9cbe05f-fe1a-4410-b4dc-40cc301c7b81>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetPPTCheckCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Callback	是	String	进度回调地址，如果传空字符串会删除原来的回调地址配置，回调地址仅支持http或https协议，即回调地址以http://或https://开头。回调数据格式请参考文档： https://cloud.tencent.com/document/product/1137/40260#c9cbe05f-fe1a-4410-b4dc-40cc301c7b81 示例值： https://example.com/transcode/callback

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置回调地址

设置回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetPPTCheckCallback
&SdkAppId=1400000001
```

```
&Callback=https://example.com/transcode/callback
```

```
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。

错误码	描述
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置PPT检测任务回调密钥

最近更新时间：2025-04-25 02:02:09

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置PPT检测任务回调密钥，回调鉴权方式请参考文档：<https://cloud.tencent.com/document/product/1137/40257>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetPPTCheckCallbackKey。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId 示例值：1400000001
CallbackKey	是	String	设置回调鉴权密钥，最长64字符，如果传入空字符串，那么删除现有的鉴权回调密钥，回调鉴权方式请参考文档： https://cloud.tencent.com/document/product/1137/40257

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置回调鉴权密钥

设置回调鉴权密钥

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetPPTCheckCallbackKey
&SdkAppId=1400000001
&CallbackKey=6vg9G7Fd
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询PPT检测任务回调地址

最近更新时间：2025-04-25 02:02:10

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询PPT检测任务回调地址

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribePPTCheckCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId 示例值：1400000001

3. 输出参数

参数名称	类型	描述
Callback	String	回调地址 示例值：https://example.com/transcode/callback
CallbackKey	String	回调鉴权密钥 示例值：6vg9G7Fd
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询回调地址

查询回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribePPTCheckCallback
&SdkAppId=1400000001
```

&<公共请求参数>

输出示例

```
{
  "Response": {
    "Callback": "https://example.com/transcode/callback",
    "CallbackKey": "6vg9G7Fd",
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。

错误码	描述
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

创建PPT检测任务

最近更新时间：2025-05-27 02:06:09

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

检测PPT文件，识别PPT中包含的动态转码任务（Transcode）不支持的元素

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：CreatePPTCheckTask。
Version	是	String	公共参数，本接口取值：2019-09-19。
Region	是	String	公共参数，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Url	是	String	经过URL编码后的PPT文件地址。URL 编码会将字符转换为可通过因特网传输的格式，例如文档地址为http://example.com/测试.pptx，经过URL编码之后为http://example.com/%E6%B5%8B%E8%AF%95.pptx。为了提高URL解析的成功率，请对URL进行编码。 示例值： http://example.com/%E6%B5%8B%E8%AF%95.pptx
AutoHandleUnsupportedElement	否	Boolean	是否对不支持元素开启自动处理的功能，默认不开启。 true -- 开启 false -- 不开启 当设置为true时，可配合AutoHandleUnsupportedElementTypes参数使用，具体有哪些不兼容元素类型，可参考AutoHandleUnsupportedElementTypes参数的说明。 示例值：true
AutoHandleUnsupportedElementTypes.N	否	Array of Integer	此参数仅在AutoHandleUnsupportedElement参数为true的情况下有效。 指定需要自动处理的不兼容元素类型，默认对所有不兼容的元素进行自动处理。 目前支持检测的不兼容元素类型及对应的自动处理方式如下： 0: 不支持的墨迹类型

参数名称	必选	类型	描述
			-- 自动处理方式：移除墨迹 1: 自动翻页 -- 自动处理方式：移除自动翻页设置，并修改为单击切换 2: 已损坏音视频 -- 自动处理方式：移除对损坏音视频的引用 3: 不可访问资源 -- 自动处理方式：移除对不可访问的资源的引用 4: 只读文件 -- 自动处理方式：移除只读设置 5: 不支持的元素编辑锁定状态 -- 自动处理方式：移除锁定状态 6: 可能有兼容问题的字体 -- 自动处理方式：不支持处理 7: 设置了柔化边缘的GIF图片 -- 自动处理方式：移除柔化边缘设置 8: 存在不兼容的空格下划线 -- 自动处理方式：通过调整空格下划线前后文本的字体语言体系，保证空格下划线表现正常 9: 存在设置了分段动画的数学公式和文本混合内容 -- 自动处理方式：不支持处理 10: 存在设置了分段动画的渐变色文本 -- 自动处理方式：不支持处理 11: 存在不兼容的分散对齐方式 -- 自动处理方式：不支持处理 12: 存在不兼容的多倍行距设置 -- 自动处理方式：不支持处理 13: 存在带有特殊符号内容的datetime类型的a:fld标签元素 -- 自动处理方式：a:fld标签替换为普通文本 示例值：[0, 1, 2]

3. 输出参数

参数名称	类型	描述
TaskId	String	检测任务的唯一标识Id，用于查询该任务的进度以及检测结果 示例值：g6ls63ps49vteb8bk1mb
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 创建PPT检测任务

创建PPT检测任务，识别动态转码不支持的元素列表

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreatePPTCheckTask
<公共请求参数>

{
  "Url": "http://example.com/%E6%B5%8B%E8%AF%95.pptx",
  "SdkAppId": 1400000001
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "TaskId": "g61s63ps49vteb8bk1mb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.Preprocess	转码预处理失败，具体请参考错误描述或联系客服人员。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.FileFormatUnsupported	文档后缀名对应的格式不支持。
InvalidParameter.PreprocessParameter	转码预处理参数格式不正确。
InvalidParameter.SdkAppIdNotFound	SdkAppId不存在或格式错误。
LimitExceeded.TaskConcurrency	转码或录制任务并发数量超过限制，请参考错误描述或稍后重试。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

通过文档URL查询转码任务状态

最近更新时间：2025-05-27 02:06:08

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

通过文档URL查询转码任务，返回最近一天内最新的转码任务状态

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTranscodeByUrl。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Url	是	String	经过URL编码后的转码文件地址。URL 编码会将字符转换为可通过因特网传输的格式，比如文档地址为 http://example.com/测试.pdf，经过URL编码之后为 http://example.com/%E6%B5%8B%E8%AF%95.pdf。为了提高URL解析的成功率，请对URL进行编码。 示例值：https://path/to/document.ppt

3. 输出参数

参数名称	类型	描述
Progress	Integer	转码的当前进度,取值范围为0~100 示例值：100
Status	String	任务的当前状态 – QUEUED: 正在排队等待转换 – PROCESSING: 转换中 – FINISHED: 转换完成 – EXCEPTION: 转换异常 示例值：FINISHED
TaskId	String	转码任务的唯一标识Id 示例值：0agdnligqtgtvkm65emb
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 通过url查询文档动态转码任务状态

通过url查询文档动态转码任务状态

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeTranscodeByUrl
<公共请求参数>

{
  "Url": "https://path/to/document.ppt",
  "SdkAppId": 1400000001
}
```

输出示例

```
{
  "Response": {
    "Progress": 100,
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "Status": "FINISHED",
    "TaskId": "0agdnligqtgtvkm65emb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.TranscodeServerError	转码服务出现内部错误，请稍后重试或联系客户人员。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

实时录制相关接口

查询实时录制任务

最近更新时间：2025-05-27 02:06:11

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询录制任务状态与结果

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeOnlineRecord。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
TaskId	是	String	实时录制任务Id 示例值：ghucnligtgtvk2624mb

3. 输出参数

参数名称	类型	描述
FinishReason	String	录制结束原因， - AUTO: 房间内长时间没有音视频上行及白板操作导致自动停止录制 - USER_CALL: 主动调用了停止录制接口 - EXCEPTION: 录制异常结束 - FORCE_STOP: 强制停止录制，一般是因为暂停超过90分钟或者录制总时长超过24小时。 示例值：USER_CALL
TaskId	String	需要查询结果的录制任务Id 示例值：ghucnligtgtvk2624mb
Status	String	录制任务状态 - PREPARED: 表示录制正在准备中（进房/启动录制服务等操作） - RECORDING: 表示录制已开始 - PAUSED: 表示录制已暂停 - STOPPED: 表示录制已停止，正在处理并上传视频

参数名称	类型	描述
		– FINISHED: 表示视频处理并上传完成，成功生成录制结果 示例值：FINISHED
RoomId	Integer	房间号 示例值：880528
GroupId	String	白板的群组 Id 示例值：880528
RecordUserId	String	录制用户 Id 示例值：tic_record_user_880528_test-01
RecordStartTime	Integer	实际开始录制时间，Unix 时间戳，单位秒 示例值：1568949369
RecordStopTime	Integer	实际停止录制时间，Unix 时间戳，单位秒 示例值：1568949392
TotalTime	Integer	回放视频总时长（单位：毫秒） 示例值：18317
ExceptionCnt	Integer	录制过程中出现异常的次数 示例值：0
OmittedDurations	Array of OmittedDuration	拼接视频中被忽略的时间段，只有开启视频拼接功能的时候，这个参数才是有效的
VideoInfos	Array of VideoInfo	录制视频列表
ReplayUrl	String	回放URL，需配合信令播放器使用。此字段仅适用于视频生成模式 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://replayurl
Interrupts	Array of Interrupt	视频流在录制过程中断流次数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询实时录制任务

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeOnlineRecord
&SdkAppId=1400000001
&TaskId=ghucnliggtgtvk2624mb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "ExceptionCnt": 0,
    "FinishReason": "USER_CALL",
```

```
"GroupId": "880528",
"OmittedDurations": [],
"RecordStartTime": 1568949369,
"RecordStopTime": 1568949392,
"RecordUserId": "tic_record_user_880528_test-01",
"RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851",
"RoomId": 880528,
"Status": "FINISHED",
"TaskId": "ghucnligqgtgtvk2624mb",
"TotalTime": 18317,
"ReplayUrl": "http://replayurl",
"Interrupts": [],
"VideoInfos": [
{
  "UserId": "Mac_trtc_04",
  "VideoDuration": 17969,
  "VideoFormat": "mp4",
  "VideoId": "dace3518e865e76a9e36712c629822ba",
  "VideoPlayTime": 0,
  "VideoSize": 593418,
  "VideoType": 0,
  "Width": 640,
  "Height": 360,
  "VideoUrl": "http://online-recording-1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/d124f518e865e76a9e36712c629822ba.mp4"
},
{
  "UserId": "tic_mixstream_880528_101",
  "VideoDuration": 18205,
  "VideoFormat": "mp4",
  "VideoId": "763d1f6b8679c3f17fb118bd37d05c85",
  "VideoPlayTime": 3,
  "VideoSize": 765545,
  "VideoType": 3,
  "Width": 640,
  "Height": 360,
  "VideoUrl": "http://online-recording-1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/763d1f6b86724f51fb118bd37d05c85.mp4"
},
{
  "UserId": "tic_mixstream_880528_3",
  "VideoDuration": 18222,
  "VideoFormat": "mp4",
  "VideoId": "1b9623df0516dc7318df89f6e7fffc1e",
  "VideoPlayTime": 95,
  "VideoSize": 402038,
  "VideoType": 3,
  "Width": 640,
  "Height": 360,
  "VideoUrl": "http://online-recording-1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/1b9623df05124f51318d"
```

```
f89f6e7fffc1e.mp4"
},
{
  "UserId": "",
  "VideoDuration": 17605,
  "VideoFormat": "mp4",
  "VideoId": "a8152f8faa2cfe621dc965a066a5813c",
  "VideoPlayTime": 623,
  "VideoSize": 226337,
  "VideoType": 2,
  "Width": 640,
  "Height": 360,
  "VideoUrl": "http://online-recording-1259648581.file.myqcloud.com/00sp43mantgtv4r842mb/a815224f512cfe621dc965a066a5813c.mp4"
}
]
}
}
```

示例2 查询录制失败的实时录制任务

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeOnlineRecord
&SdkAppId=1400000001
&TaskId=ghucnligtgtvk2624mb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Error": {
      "Code": "FailedOperation.Record",
      "Message": "code: 40004, msg: 2019-09-23 10:31:47, im login failed with code 6206, desc serSig expired\n",
    },
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.Record	录制失败，具体请参考错误描述。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

查询实时录制回调地址

最近更新时间：2025-05-27 02:06:11

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询实时录制回调地址

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeOnlineRecordCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId 示例值：1400000001

3. 输出参数

参数名称	类型	描述
Callback	String	实时录制事件回调地址，如果未设置回调地址，该字段为空字符串 示例值：https://example.com/online/callback
CallbackKey	String	实时录制回调鉴权密钥 示例值：6vg9G7Fd
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询实时录制回调地址

查询实时录制回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=DescribeOnlineRecordCallback
&SdkAppId=1400000001
```

&<公共请求参数>

输出示例

```
{
  "Response": {
    "Callback": "https://example.com/online/callback",
    "CallbackKey": "6vg9G7Fd",
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。

错误码	描述
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

暂停实时录制

最近更新时间：2025-05-27 02:06:11

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

暂停实时录制

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：PauseOnlineRecord。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId
TaskId	是	String	实时录制任务 Id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 暂停实时录制

输入示例

```
https://tiw.tencentcloudapi.com/?Action=PauseOnlineRecord
&SdkAppId=1400000001
&TaskId=bj0mt2l23osdj300h130
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。
UnsupportedOperation.InvalidTaskStatus	当前未完成任务不能在此状态下执行指定操作，例如对正在录制的任务执行恢复录制等。
UnsupportedOperation.TaskHasAlreadyStopped	任务结束，不能执行指定操作。

恢复实时录制

最近更新时间：2025-05-27 02:06:11

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

恢复实时录制

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ResumeOnlineRecord。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId
TaskId	是	String	恢复录制的实时录制任务 Id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 恢复实时录制任务

输入示例

```
https://tiw.tencentcloudapi.com/?Action=ResumeOnlineRecord
&SdkAppId=1400000001
&TaskId=07b4c70bpafm1b4majmb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。
UnsupportedOperation.InvalidTaskStatus	当前未完成任务不能在此状态下执行指定操作，例如对正在录制的任务执行恢复录制等。
UnsupportedOperation.TaskHasAlreadyStopped	任务结束，不能执行指定操作。

设置实时录制回调地址

最近更新时间：2025-05-27 02:06:10

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置实时录制回调地址，回调数据格式请参考文档：<https://cloud.tencent.com/document/product/1137/40258>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetOnlineRecordCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Callback	是	String	实时录制任务结果回调地址，如果传空字符串会删除原来的回调地址配置，回调地址仅支持 http或https协议，即回调地址以http://或https://开头。回调数据格式请参考文档： https://cloud.tencent.com/document/product/1137/40258 示例值： https://example.com/online/callback

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置实时录制回调地址

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetOnlineRecordCallback
&SdkAppId=1400000001
&Callback=https://example.com/online/callback
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置实时录制回调密钥

最近更新时间：2025-05-27 02:06:10

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置实时录制回调鉴权密钥，回调鉴权方式请参考文档：<https://cloud.tencent.com/document/product/1137/40257>

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetOnlineRecordCallbackKey。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId
CallbackKey	是	String	设置实时录制回调鉴权密钥，最长64字符，如果传入空字符串，那么删除现有的鉴权回调密钥。回调鉴权方式请参考文档： https://cloud.tencent.com/document/product/1137/40257

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置实时录制回调鉴权密钥

输入示例

```
https://tiw.tencentcloudapi.com/?Action=SetOnlineRecordCallbackKey
&SdkAppId=1400000001
&CallbackKey=6vg9G7Fd
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppld	SdkAppld不存在或者SdkAppld与当前腾讯云账号不对应。

开始实时录制

最近更新时间：2025-05-27 02:06:10

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

发起一个实时录制任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StartOnlineRecord。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
RoomId	是	Integer	需要录制的白板房间号，取值范围：(1, 4294967295)。 1. 在没有指定GroupId的情况下，实时录制默认以RoomId的字符串表达形式作为同步白板信令的IM群组ID（比如RoomId为12358，则IM群组ID为"12358"），并加群进行信令同步，请在开始录制前确保相应IM群组已创建完成，否则会导致录制失败。 2. 在没有指定TRTCRoomId和TRTCRoomIdStr的情况下，默认会以RoomId作为TRTC房间号进房拉流进行录制。 示例值：12358
RecordUserId	是	String	用于录制服务进房的用户ID，最大长度不能大于60个字节，格式为tic_record_user_\${RoomId}_\${Random}，其中 \${RoomId}与录制房间号对应，\${Random}为一个随机字符串。 该ID必须是一个单独的未在SDK中使用的ID，录制服务使用这个用户ID进入房间进行音视频与白板录制，若该ID和SDK中使用的ID重复，会导致SDK和录制服务互踢，影响正常录制。 示例值：tic_record_user_12358_1258
RecordUserSig	是	String	与RecordUserId对应的IM签名 示例值：eJwtjb0OgjAAhN*K4a0tcVC4iDGklg-g7K4NEiLViLUUsFgfHcRmS733eXuDY6b
GroupId	否	String	白板进行信令同步的 IM 群组 ID。 在没有指定GroupId的情况下，实时录制服务将使用 RoomId 的字符串形式作为同步白板信令的IM群组ID。 在指定了GroupId的情况下，实时录制将优先使用GroupId作为同步白板信令的群组ID。

参数名称	必选	类型	描述
			请在开始录制前确保相应的IM群组已创建完成，否则会导致录制失败。 示例值：12358
Concat	否	Concat	录制视频拼接参数
Whiteboard	否	Whiteboard	录制白板参数，例如白板宽高等
MixStream	否	MixStream	录制混流参数 特别说明： 1. 混流功能需要根据额外开通，请联系腾讯云互动白板客服人员 2. 使用混流功能，必须提供 Extras 参数，且 Extras 参数中必须包含 "MIX_STREAM"
Extras.N	否	Array of String	使用到的高级功能列表 可以选值列表： MIX_STREAM – 混流功能 示例值：["MixStream"]
AudioFileNeeded	否	Boolean	是否需要在结果回调中返回各路流的纯音频录制文件，文件格式为mp3 示例值：false
RecordControl	否	RecordControl	录制控制参数，用于更精细地指定需要录制哪些流，某一路流是否禁用音频，是否只录制小画面等
RecordMode	否	String	录制模式 REALTIME_MODE – 实时录制模式（默认） VIDEO_GENERATION_MODE – 视频生成模式（内测中，需邮件申请开通） 示例值：REALTIME_MODE
ChatGroupId	否	String	聊天群组ID，此字段仅适用于视频生成模式 在视频生成模式下，默认会记录白板群组内的非白板指令消息，如果指定了 ChatGroupId，则会记录指定群ID的聊天消息。 示例值：12358
AutoStopTimeout	否	Integer	自动停止录制超时时间，单位秒，取值范围[300, 86400]，默认值为300秒。 当超过设定时间房间内没有音视频上行且没有白板操作的时候，录制服务会自动停止当前录制任务。 示例值：300
ExtraData	否	String	内部参数，可忽略
TRTCRoomId	否	Integer	TRTC数字类型房间号，取值范围: (1, 4294967295)。 在同时指定了RoomId与TRTCRoomId的情况下，优先使用TRTCRoomId作为实时录制拉TRTC流的TRTC房间号。 当指定了TRTCRoomIdStr的情况下，此字段将被忽略。 示例值：12358
TRTCRoomIdStr	否	String	TRTC字符串类型房间号。 在指定了TRTCRoomIdStr的情况下，会优先使用TRTCRoomIdStr作为实时录制拉TRTC流的TRTC房间号。 示例值：record-room

3. 输出参数

参数名称	类型	描述
TaskId	String	录制任务Id 示例值：0c69baivvc3lp8t4f80b
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 开始实时录制

输入示例

```
https://tiw.tencentcloudapi.com/?Action=StartOnlineRecord
&SdkAppId=1400000001
&RoomId=1203
&RecordUserId=tic_record_user_1203_141551
&RecordUserSig=usersig_of_<tic_record_user_1203_141551>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "TaskId": "bj0mt2l23osdj300h130"
  }
}
```

示例2 开始带混流的实时录制

使用内置混流布局进行混流录制

输入示例

```
https://tiw.tencentcloudapi.com/?Action=StartOnlineRecord
&SdkAppId=1400000001
&RoomId=1203
&RecordUserId=tic_record_user_1203_141551
&RecordUserSig=usersig_of_<tic_record_user_1203_141551>
&Whiteboard.Width=1280
&Whiteboard.Height=960
&Whiteboard.InitParam={"ratio":"16:9"}
&Concat.Enabled=true
&Concat.Image=http://background/image.jpg
&MixStream.Enabled=true
&MixStream.DisableAudio=false
&MixStream.ModelId=2
&MixStream.TeacherId=teacher
```

```
&Extras.0=MIX_STREAM
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "TaskId": "bj0mt2l23osdj300h130"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.InvalidExtra	额外指定的特殊功能不存在。
InvalidParameter.RecordParameter	实时录制参数格式不正确。
InvalidParameter.SdkAppldNotFound	SdkAppld不存在或格式错误。

错误码	描述
LimitExceeded.TaskConcurrency	转码或录制任务并发数量超过限制，请参考错误描述或稍后重试。
ResourceInUse.RecordUserId	实时录制任务录制用户已被其他录制任务使用。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

停止实时录制

最近更新时间：2025-05-27 02:06:10

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

停止实时录制

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：StopOnlineRecord。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId
TaskId	是	String	需要停止录制的任务 Id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 停止在线录制

停止在线录制

输入示例

```
https://tiw.tencentcloudapi.com/?Action=StopOnlineRecord
&SdkAppId=1400000001
&TaskId=07b4c70bpafm1b4majmb
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppld	SdkAppld不存在或者SdkAppld与当前腾讯云账号不对应。
UnsupportedOperation.TaskHasAlreadyStopped	任务结束，不能执行指定操作。

统计查询相关接口

查询当前执行中的任务列表

最近更新时间：2025-05-27 02:06:05

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

根据指定的任务类型，获取当前正在执行中的任务列表。只能查询最近3天内创建的任务。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeRunningTasks。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppID	是	Integer	应用的SdkAppID 示例值：1400000001
TaskType	是	String	指定需要获取的任务类型。 有效取值如下： - TranscodeH5: 动态转码任务，文档转HTML5页面 - TranscodeJPG: 静态转码任务，文档转图片 - WhiteboardPush: 白板推流任务 - OnlineRecord: 实时录制任务 示例值：WhiteboardPush
Offset	否	Integer	分页获取时的任务偏移量，默认为0。 示例值：0
Limit	否	Integer	每次获取任务列表时最大获取任务数，默认值为100。 有效取值范围：[1, 500] 示例值：100

3. 输出参数

参数名称	类型	描述
Total	Integer	当前正在执行中的任务总数 示例值：1
Tasks	Array of	任务信息列表

参数名称	类型	描述
	RunningTaskItem	
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取当前正在执行的白板推流任务列表

通过接口获取当前应用白板推流任务类型的正在执行中的任务列表

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeRunningTasks
<公共请求参数>

{
  "SdkAppID": 1400000001,
  "TaskType": "WhiteboardPush"
}
```

输出示例

```
{
  "Response": {
    "Total": 1,
    "Tasks": [
      {
        "TaskID": "bj0mt2l23osdj300hl31",
        "TaskType": "WhiteboardPush",
        "CreateTime": "2023-03-14 15:00:00",
        "CancelTime": "",
        "Status": "PROCESSING",
        "Progress": 11,
        "RoomID": 12345,
        "FileURL": "",
        "SdkAppID": 1400000001
      }
    ],
    "RequestId": "d290f1ee-6c54-4b01-90e6-d701748f0851"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation.FileDownloadFail	文档下载失败，请检查请求参数中URL是否正确，或者如果您使用其他的文件存储服务，请检查文件存储服务的上传带宽，文档转码服务仅提供1分钟的下载时间，如果下载不成功本次的转码请求将以失败终止。
FailedOperation.FileFormatError	文档格式错误，不支持转换只读文档或者已加密的文档。
FailedOperation.FileOpenFail	文档打开失败，请检查提交转码的文档是否加密或有其他格式问题。
FailedOperation.FileUploadFail	转码后上传结果失败，请稍后重试。
FailedOperation.Transcode	转码失败，具体请参考错误描述或联系客服人员。
FailedOperation.TranscodeServerError	转码服务出现内部错误，请稍后重试或联系客服人员。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.FileFormatUnsupported	文档后缀名对应的格式不支持。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
InvalidParameter.TranscodeParameter	文档转码参数格式不正确。
InvalidParameter.UrlFormatError	文档下载Url格式错误，请检查请求参数里的Url。
LimitExceeded.TranscodePagesLimitation	超过文档最大页数限制，目前不支持超过500页的文件转码，如有特殊需求，请联系客服人员。
ResourceUnavailable.NotRegistered	未开通互动白板。

错误码	描述
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

告警（内测中）相关接口

查询告警回调地址

最近更新时间：2025-05-27 02:06:12

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

查询告警回调地址。此功能需要申请白名单使用。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeWarningCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	应用的SdkAppId 示例值：1400000001

3. 输出参数

参数名称	类型	描述
Callback	String	告警事件回调地址，如果未设置回调地址，该字段为空字符串 示例值：https://example.com/online/callback
CallbackKey	String	告警回调鉴权密钥 示例值：6vg9G7Fd
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取告警回调地址信息

获取告警回调地址信息

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeWarningCallback
<公共请求参数>

{
  "SdkAppId": 123456
}
```

输出示例

```
{
  "Response": {
    "Callback": "http://callback",
    "CallbackKey": "12345",
    "RequestId": "786eafde-17a8-47a9-9e1c-05b2b282dccc"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter	参数错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

设置告警回调地址

最近更新时间：2025-05-27 02:06:11

1. 接口描述

接口请求域名：tiw.tencentcloudapi.com。

设置告警回调地址。此功能需要申请白名单使用。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetWarningCallback。
Version	是	String	公共参数 ，本接口取值：2019-09-19。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SdkAppId	是	Integer	客户的SdkAppId 示例值：1400000001
Callback	是	String	告警回调地址，如果传空字符串会删除原来的回调地址配置，回调地址仅支持http或https协议，即回调地址以http://或https://开头。 回调数据格式请参考文档： https://cloud.tencent.com/document/product/1137/90112 示例值： https://example.com/transcode/callback
CallbackKey	是	String	设置告警回调鉴权密钥，最长64字符，如果传入空字符串，那么删除现有的鉴权回调密钥，回调鉴权方式请参考文档： https://cloud.tencent.com/document/product/1137/40257 示例值：123456

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 设置告警回调地址

设置告警回调地址

输入示例

```
POST / HTTP/1.1
Host: tiw.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: SetWarningCallback
<公共请求参数>

{
  "SdkAppId": 123456,
  "Callback": "http://callback",
  "CallbackKey": "123456"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "786eafde-17a8-47a9-9e1c-05b2b282dccc"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。

数据结构

最近更新时间：2025-03-14 18:17:33

AuthParam

鉴权参数

被如下接口引用：StartWhiteboardPush。

名称	类型	必选	描述
SdkAppId	Integer	是	应用SdkAppId 示例值：12345678
UserId	String	是	用户ID 示例值：userid
UserSig	String	是	用户ID对应的签名 示例值：usersig

Canvas

混流画布参数

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
LayoutParams	LayoutParams	是	混流画布宽高配置
BackgroundColor	String	否	背景颜色，默认为黑色，格式为RGB格式，如红色为"#FF0000"

Concat

实时录制视频拼接参数

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
Enabled	Boolean	是	是否开启拼接功能 在开启了视频拼接功能的情况下，实时录制服务会把同一个用户因为暂停导致的多段视频拼接成一个视频
Image	String	否	视频拼接时使用的垫片图片下载地址，不填默认用全黑的图片进行视频垫片

CustomLayout

自定义混流布局参数

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
Canvas	Canvas	是	混流画布参数
InputStreamList	Array of StreamLayout	是	流布局参数，每路流的布局不能超出画布区域

ExcelParam

Excel转码相关参数

被如下接口引用：CreateTranscode。

名称	类型	必选	描述
PaperSize	Integer	否	表格转码纸张（画布）大小，默认为0。 0 -- A4 1 -- A2 2 -- A0 注：当设置的值超出合法取值范围时，将采用默认值。 示例值：0
PaperDirection	Integer	否	表格文件转换纸张方向，默认为0。 0 -- 代表垂直方向 非0 -- 代表水平方向 示例值：0

Interrupt

实时录制中出现的用户视频流断流次数统计

被如下接口引用：DescribeOnlineRecord。

名称	类型	描述
UserId	String	用户ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：tiw_user_634
Count	Integer	视频流断流次数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2

LayoutParams

自定义混流配置布局参数

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
Width	Integer	是	流画面宽，取值范围[2,3000]
Height	Integer	是	流画面高，取值范围[2,3000]
X	Integer	否	当前画面左上角顶点相对于Canvas左上角顶点的x轴偏移量，默认为0，取值范围[0,3000]
Y	Integer	否	当前画面左上角顶点相对于Canvas左上角顶点的y轴偏移量，默认为0，取值范围[0,3000]
ZOrder	Integer	否	画面z轴位置，默认为0 z轴确定了重叠画面的遮盖顺序，z轴值大的画面处于顶层

MixStream

混流配置

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
Enabled	Boolean	是	是否开启混流
DisableAudio	Boolean	否	是否禁用音频混流
ModelId	Integer	否	内置混流布局模板ID, 取值 [1, 2], 区别见内置混流布局模板样式示例说明 在没有填Custom字段时候，ModelId是必填的
TeacherId	String	否	老师用户ID 此字段只有在ModelId填了的情况下生效 填写TeacherId的效果是把指定为TeacherId的用户视频流显示在内置模板的第一个小画面中
Custom	CustomLayout	否	自定义混流布局参数 当此字段存在时，ModelId 及 TeacherId 字段将被忽略

OmittedDuration

拼接视频中被忽略的时间段

被如下接口引用：DescribeOnlineRecord。

名称	类型	描述
VideoTime	Integer	录制暂停时间戳对应的视频播放时间(单位: 毫秒)
PauseTime	Integer	录制暂停时间戳(单位: 毫秒)
ResumeTime	Integer	录制恢复时间戳(单位: 毫秒)

PPTErr

PPT错误元素

被如下接口引用：DescribePPTCheck。

名称	类型	描述
Name	String	元素名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：Ink 1
Type	Integer	0: 不支持的墨迹类型，1: 不支持自动翻页，2: 存在已损坏音视频，3: 存在不可访问资源，4: 只读文件 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Detail	String	错误详情 注意：此字段可能返回 null，表示取不到有效值。 示例值：wps墨迹

PPTErrSlide

PPT错误页面列表

被如下接口引用：DescribePPTCheck。

名称	类型	描述
Page	String	异常元素存在的页面，由页面类型+页码组成，页码类型包括：幻灯片、幻灯片母版、幻灯片布局等 注意：此字段可能返回 null，表示取不到有效值。 示例值：幻灯片1
Errs	Array of PPTErr	错误元素列表 注意：此字段可能返回 null，表示取不到有效值。

RecordControl

录制控制参数，用于指定全局录制控制及具体流录制控制参数，比如设置需要对哪些流进行录制，是否只录制小画面等

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
Enabled	Boolean	是	设置是否开启录制控制参数，只有设置为true的时候，录制控制参数才生效。
DisableRecord	Boolean	否	设置是否禁用录制的全局控制参数。一般与StreamControls参数配合使用。 true – 所有流都不录制。 false – 所有流都录制。默认为false。 这里的设置对所有流都生效，如果同时在 StreamControls 列表中针对指定流设置了控制参数，则优先采用StreamControls中设置的控制参数。
DisableAudio	Boolean	否	设置是否禁用所有流的音频录制的全局控制参数。一般与StreamControls参数配合使用。 true – 所有流的录制都不对音频进行录制。 false – 所有流的录制都需要对音频进行录制。默认为false。 这里的设置对所有流都生效，如果同时在 StreamControls 列表中针对指定流设置了控制参数，则优先采用StreamControls中设置的控制参数。
PullSmallVideo	Boolean	否	设置是否所有流都只录制小画面的全局控制参数。一般与StreamControls参数配合使用。 true – 所有流都只录制小画面。设置为true时，请确保上行端在推流的时候同时上行小画面，否则录制视频可能是黑屏。 false – 所有流都录制大画面，默认为false。 这里的设置对所有流都生效，如果同时在 StreamControls 列表中针对指定流设置了控制参数，则优先采用StreamControls中设置的控制参数。
StreamControls	Array of StreamControl	否	针对具体流指定控制参数，如果列表为空，则所有流采用全局配置的控制参数进行录制。列表不为空，则列表中指定的流将优先按此列表指定的控制参数进行录制。

RunningTaskItem

正在运行的任务列表项

被如下接口引用：DescribeRunningTasks。

名称	类型	描述
SdkAppID	Integer	应用SdkAppID

名称	类型	描述
		示例值：1400000001
TaskID	String	任务ID 示例值：0agdnligqgtgvm65emb
TaskType	String	任务类型 – TranscodeH5: 动态转码任务，文档转HTML5页面 – TranscodeJPG: 静态转码任务，文档转图片 – WhiteboardPush: 白板推流任务 – OnlineRecord: 实时录制任务 示例值：WhiteboardPush
CreateTime	String	任务创建时间 示例值：2023-03-14 15:00:00
CancelTime	String	任务取消时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2023-03-14 20:00:00
Status	String	任务状态 – QUEUED: 任务正在排队等待执行中 – PROCESSING: 任务正在执行中 – FINISHED: 任务已完成 示例值：QUEUED
Progress	Integer	任务当前进度 示例值：50
FileURL	String	转码任务中转码文件的原始URL 此参数只有任务类型为TranscodeH5、TranscodeJPG类型时才会有有效值。其他任务类型为空字符串。 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://xxx.pptx
RoomID	Integer	房间号 当任务类型为TranscodeH5、TranscodeJPG时，房间号为0。 注意：此字段可能返回 null，表示取不到有效值。 示例值：12345

SnapshotCOS

板书文件存储cos参数

被如下接口引用：CreateSnapshotTask。

名称	类型	必选	描述
Uin	Integer	是	cos所在腾讯云账号uin 示例值：100006912345
Region	String	是	cos所在地区 示例值：ap-guangzhou
Bucket	String	是	cos存储桶名称 示例值：examplebucket-1250000000
TargetDir	String	否	板书文件存储根目录 示例值：snapshot

名称	类型	必选	描述
Domain	String	否	CDN加速域名 示例值：xxxx.file.qcloud.com

SnapshotResult

白板板书结果

被如下接口引用：DescribeSnapshotTask。

名称	类型	描述
ErrorCode	String	任务执行错误码 注意：此字段可能返回 null，表示取不到有效值。 示例值：InternalErr
ErrorMessage	String	任务执行错误信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：redis connect failed
Total	Integer	快照生成图片总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：12
Snapshots	Array of String	快照图片链接列表 注意：此字段可能返回 null，表示取不到有效值。

SnapshotWhiteboard

生成白板板书时的白板参数，例如白板宽高等

被如下接口引用：CreateSnapshotTask。

名称	类型	必选	描述
Width	Integer	否	白板宽度大小，默认为1280，有效取值范围[0，2560] 示例值：1280
Height	Integer	否	白板高度大小，默认为720，有效取值范围[0，2560] 示例值：960
InitParams	String	否	白板初始化参数的JSON转义字符串，透传到白板 SDK 示例值：{"ratio":"4:3"}

StreamControl

指定流录制的控制参数，比如是否禁用音频、视频是录制大画面还是录制小画面等

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
StreamId	String	是	视频流ID 视频流ID的取值含义如下： 1. tic_record_user – 表示白板视频流 2. tic_substream – 表示辅路视频流 3. 特定用户ID – 表示指定用户的视频流

名称	类型	必选	描述
			在实际录制过程中，视频流ID的匹配规则为前缀匹配，只要真实流ID的前缀与指定的流ID一致就认为匹配成功。
DisableRecord	Boolean	否	设置是否对此路流开启录制。 true – 表示不对这路流进行录制，录制结果将不包含这路流的视频。 false – 表示需要对这路流进行录制，录制结果会包含这路流的视频。 默认为 false。
DisableAudio	Boolean	否	设置是否禁用这路流的音频录制。 true – 表示不对这路流的音频进行录制，录制结果里这路流的视频将会没有声音。 false – 录制视频会保留音频，如果设置为true，则录制视频会丢弃这路流的音频。 默认为 false。
PullSmallVideo	Boolean	否	设置当前流录制视频是否只录制小画面。 true – 录制小画面。设置为true时，请确保上行端同时上行小画面，否则录制视频可能是黑屏。 false – 录制大画面。 默认为 false。

StreamLayout

流布局参数

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord。

名称	类型	必选	描述
LayoutParams	LayoutParams	是	流布局配置参数
InputStreamId	String	否	视频流ID 流ID的取值含义如下： - tic_record_user – 表示当前画面用于显示白板视频流 - tic_substream – 表示当前画面用于显示辅路视频流 - 特定用户ID – 表示当前画面用于显示指定用户的视频流 - 不填 – 表示当前画面用于备选，当有新的视频流加入时，会从这些备选的空中选择一个没有被占用的位置来显示新的视频流画面
BackgroundColor	String	否	背景颜色，默认为黑色，格式为RGB格式，如红色为"#FF0000"
FillMode	Integer	否	视频画面填充模式。 0 – 自适应模式，对视频画面进行等比例缩放，在指定区域内显示完整的画面。此模式可能存在黑边。 1 – 全屏模式，对视频画面进行等比例缩放，让画面填充满整个指定区域。此模式不会存在黑边，但会将超出区域的那一部分画面裁剪掉。

VideoInfo

视频信息

被如下接口引用：DescribeOnlineRecord, DescribeVideoGenerationTask。

名称	类型	描述
VideoPlayTime	Integer	视频开始播放的时间（单位：毫秒）
VideoSize	Integer	视频大小（字节）
VideoFormat	String	视频格式
VideoDuration	Integer	视频播放时长（单位：毫秒）
VideoUrl	String	视频文件URL
VideoId	String	视频文件Id
VideoType	Integer	视频流类型 – 0：摄像头视频 – 1：屏幕分享视频 – 2：白板视频 – 3：混流视频 – 4：纯音频（mp3）
UserId	String	摄像头/屏幕分享视频所属用户的 Id（白板视频为空、混流视频tic_mixstream_房间号_混流布局类型、辅路视频tic_substream_UserId）
Width	Integer	视频分辨率的宽
Height	Integer	视频分辨率的高

Whiteboard

实时录制白板参数，例如白板宽高等

被如下接口引用：CreateVideoGenerationTask, StartOnlineRecord, StartWhiteboardPush。

名称	类型	必选	描述
Width	Integer	否	实时录制结果里白板视频宽，取值必须大于等于2，默认为1280 示例值：1280
Height	Integer	否	实时录制结果里白板视频高，取值必须大于等于2，默认为960 示例值：960
InitParam	String	否	白板初始化参数，透传到白板 SDK

WhiteboardPushBackupParam

白板推流备份相关请求参数

被如下接口引用：StartWhiteboardPush。

名称	类型	必选	描述
PushUserId	String	是	用于白板推流服务进房的用户ID， 该ID必须是一个单独的未在SDK中使用的ID，白板推流服务将使用这个用户ID进入房间进行白板推流，若该ID和SDK中使用的ID重复，会导致SDK和录制服务互踢，影响正常推流。
PushUserSig	String	是	与PushUserId对应的签名

错误码

最近更新时间：2025-06-30 02:33:06

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
ActionOffline	接口已下线。
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在 控制台 检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。

错误码	说明
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP 地址在黑名单中。
IpNotInWhitelist	IP 地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP 限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
AuthFailure	CAM签名/鉴权错误。
FailedOperation.FileDownloadFail	文档下载失败，请检查请求参数中URL是否正确，或者如果您使用其他的文件存储服务，请检查文件存储服务的上传带宽，文档转码服务仅提供1分钟的下载时间，如果下载不成功本次的转码请求将以失败终止。
FailedOperation.FileFormatError	文档格式错误，不支持转换只读文档或者已加密的文档。
FailedOperation.FileOpenFail	文档打开失败，请检查提交转码的文档是否加密或有其他格式问题。
FailedOperation.FileUploadFail	转码后上传结果失败，请稍后重试。

错误码	说明
FailedOperation.Preprocess	转码预处理失败，具体请参考错误描述或联系客服人员。
FailedOperation.PreprocessServerError	预处理服务出现内部错误，请稍后重试或联系客户人员。
FailedOperation.Record	录制失败，具体请参考错误描述。
FailedOperation.Transcode	转码失败，具体请参考错误描述或联系客服人员。
FailedOperation.TranscodeServerError	转码服务出现内部错误，请稍后重试或联系客户人员。
FailedOperation.WhiteboardPush	白板推流失败，具体请参考错误描述。
InvalidParameter.BodyParameterTypeUnmatched	参数类型不匹配。
InvalidParameter.CallbackAddressFormatError	回调地址格式错误。
InvalidParameter.FileFormatUnsupported	文档后缀名对应的格式不支持。
InvalidParameter.InvalidExtra	额外指定的特殊功能不存在。
InvalidParameter.PreprocessParameter	转码预处理参数格式不正确。
InvalidParameter.RecordParameter	实时录制参数格式不正确。
InvalidParameter.SdkAppIdNotFound	SdkAppId不存在或格式错误。
InvalidParameter.TaskNotFound	需要查询的任务不存在。
InvalidParameter.TranscodeParameter	文档转码参数格式不正确。
InvalidParameter.UrlFormatError	文档下载Url格式错误，请检查请求参数里的Url。
LimitExceeded.TaskConcurrency	转码或录制任务并发数量超过限制，请参考错误描述或稍后重试。
LimitExceeded.TranscodePagesLimitation	超过文档最大页数限制，目前不支持超过500页的文件转码，如有特殊需求，请联系客服人员。
ResourceInUse.RecordUserId	实时录制任务录制用户已被其他录制任务使用。
ResourceUnavailable.InterfaceUnauthorized	资源不可用，接口未授权。
ResourceUnavailable.NotRegistered	未开通互动白板。
ResourceUnavailable.ServiceExpired	账户欠费或者互动白板服务已过期。
UnauthorizedOperation.SdkAppId	SdkAppId不存在或者SdkAppId与当前腾讯云账号不对应。
UnsupportedOperation.InvalidTaskStatus	当前未完成的任务不能在此状态下执行指定操作，例如对正在录制的任务执行恢复录制等。
UnsupportedOperation.TaskHasAlreadyStopped	任务结束，不能执行指定操作。