

智能创作 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

平台管理相关接口

获取平台列表

获取账号信息

项目管理相关接口

修改项目

获取项目列表

删除项目

创建项目

在项目中导入媒体

导出视频编辑项目

操作云转推项目 (废弃)

复制项目

操作媒体转推项目

Headless媒体合成相关接口

使用视频合成协议导出视频

使用视频剪辑模板导出视频

任务管理相关接口

获取任务列表

获取任务详情

视频编辑项目相关接口

视频智能拆条

使用视频智能拆条数据导出视频

媒体资源管理相关接口

平铺浏览媒体

搜索媒体

修改媒体信息

浏览媒体

导入媒体

获取媒体详情

删除媒体

创建链接

移动资源

媒资授权相关接口

发起媒资授权

获取资源授权列表

获取共享空间

撤销资源授权

团队管理相关接口

创建团队

添加团队成员

获取团队信息

获取团队成员信息

获取成员加入的团队列表

修改团队成员

修改团队信息

删除团队成员

删除团队

分类管理相关接口

获取分类列表

创建分类

移动分类

删除分类

登录态管理相关接口

查询登录态

删除登录态

参数模板相关接口

创建视频编码配置

删除视频编码配置

修改视频编码配置

查询视频编码配置

其他接口

回调事件内容解析

数据结构

错误码

服务端 API

更新历史

最近更新时间：2025-06-04 01:21:17

第 61 次发布

发布时间：2025-06-04 01:21:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 废弃入参：AspectRatio

修改数据结构：

- [ExternalMediaInfo](#)
 - 废弃成员：Definition
- [MaterialBasicInfo](#)
 - 废弃成员：TagInfoSet
- [StorageNewFileCreatedEvent](#)
 - 废弃成员：Operator

第 60 次发布

发布时间：2025-03-27 20:17:54

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MediaCastDestinationStatus](#)
 - 修改成员：Id
- [MediaCastPlayInfo](#)
 - 修改成员：CurrentSourceId
- [TeamInfo](#)
 - 修改成员：MemberCount, CreateTime

第 59 次发布

发布时间：2024-09-25 01:13:30

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SlotInfo](#)
 - 新增成员：StartTime

第 58 次发布

发布时间：2024-07-29 01:23:21

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [VideoEditProjectOutput](#)
 - 修改成员：CoverURL

第 57 次发布

发布时间：2023-10-19 16:00:29

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ExportVideoByEditorTrackData](#)
 - 新增入参：ExportExtensionArgs
- [ExportVideoByTemplate](#)
 - 新增入参：ExportExtensionArgs

第 56 次发布

发布时间：2023-08-09 01:16:08

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MediaCastOutputMediaSetting](#)
 - 新增成员：FollowSourceInfo

第 55 次发布

发布时间：2023-07-21 01:14:51

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MediaCastProjectInfo](#)
 - 新增成员：Duration

第 54 次发布

发布时间：2023-06-29 01:11:59

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MediaCastPlaySetting](#)
 - 新增成员：AutoStartTime
- [MediaCastSourceInfo](#)
 - 新增成员：Offset, Duration, Url

第 53 次发布

发布时间：2023-03-24 01:25:23

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [HandleMediaCastProject](#)

修改接口：

- [CreateProject](#)
 - 新增入参：MediaCastProjectInput

新增数据结构：

- [MediaCastDestinationStatus](#)
- [MediaCastPlayInfo](#)
- [MediaCastProjectInput](#)

修改数据结构：

- [MediaCastDestinationInfo](#)
 - 新增成员：Id
 - 删除成员：Index
- [MediaCastSourceInfo](#)
 - 新增成员：Id
 - 删除成员：Index
 - 修改成员：Type
- [VideoEditProjectOutput](#)
 - 新增成员：CoverURL

第 52 次发布

发布时间：2022-12-05 06:32:09

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MaterialInfo](#)
 - 修改成员：BasicInfo, VideoMaterial, AudioMaterial, ImageMaterial, LinkMaterial, VideoEditTemplateMaterial, OtherMaterial

第 51 次发布

发布时间：2022-08-24 06:19:05

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [MediaCastDestinationInfo](#)
- [MediaCastDestinationInterruptInfo](#)
- [MediaCastOutputMediaSetting](#)
- [MediaCastPlaySetting](#)
- [MediaCastProjectInfo](#)
- [MediaCastSourceInfo](#)
- [MediaCastSourceInterruptInfo](#)
- [MediaCastVideoSetting](#)
- [ProjectMediaCastStatusChangedEvent](#)

修改数据结构：

- [EventContent](#)
 - 新增成员：ProjectMediaCastStatusChangedEvent
- [ProjectInfo](#)
 - 新增成员：MediaCastProjectInfo

第 50 次发布

发布时间：2022-06-24 06:09:00

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [HandleStreamConnectProject](#)
 - 新增出参：VodPullInputPlayInfo

新增数据结构：

- [VodPullInputPlayInfo](#)

第 49 次发布

发布时间：2022-06-10 06:04:57

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [StreamConnectInputInterruptInfo](#)
- [StreamConnectOutputInterruptInfo](#)

修改数据结构：

- [ProjectStreamConnectStatusChangedEvent](#)
 - 新增成员：InputInterruptInfo, OutputInterruptInfo

第 48 次发布

发布时间：2022-04-22 06:07:06

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [VideoExportCompletedEvent](#)

修改数据结构：

- [EventContent](#)
 - 新增成员：VideoExportCompletedEvent

第 47 次发布

发布时间：2022-01-27 08:22:16

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [ClassCreatedEvent](#)
- [ClassDeletedEvent](#)
- [ClassMovedEvent](#)
- [ImportMediaInfo](#)
- [MaterialAddedEvent](#)
- [MaterialDeletedEvent](#)
- [MaterialImportedEvent](#)
- [MaterialModifiedEvent](#)
- [MaterialMovedEvent](#)

- [ProjectSwitcherStatusChangedEvent](#)

修改数据结构：

- [EventContent](#)
 - 新增成员：Operator, ProjectSwitcherStatusChangedEvent, MaterialImportedEvent, MaterialAddedEvent, MaterialMovedEvent, MaterialModifiedEvent, MaterialDeletedEvent, ClassCreatedEvent, ClassMovedEvent, ClassDeletedEvent
- [StorageNewFileCreatedEvent](#)
 - 新增成员：TaskId, SourceContext

第 46 次发布

发布时间：2021-12-24 08:07:24

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [TextSlotInfo](#)

修改数据结构：

- [SlotInfo](#)
 - 新增成员：DefaultTextSlotInfo

第 45 次发布

发布时间：2021-12-14 08:07:36

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ExternalMediaInfo](#)
 - 新增成员：StorageId
 - 修改成员：Definition

第 44 次发布

发布时间：2021-12-02 08:04:53

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [VideoExportExtensionArgs](#)
 - 新增成员：FrameRate

第 43 次发布

发布时间：2021-11-26 08:06:20

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [HandleStreamConnectProject](#)
 - 新增入参：Operator

第 42 次发布

发布时间：2021-11-17 08:08:41

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateVideoEncodingPreset](#)
- [DeleteVideoEncodingPreset](#)
- [DescribeVideoEncodingPresets](#)
- [ModifyVideoEncodingPreset](#)

修改接口：

- [ExportVideoEditProject](#)
 - 新增入参：ExportExtensionArgs

新增数据结构：

- [VideoEncodingPreset](#)
- [VideoEncodingPresetAudioSetting](#)
- [VideoEncodingPresetAudioSettingForUpdate](#)
- [VideoEncodingPresetVideoSetting](#)
- [VideoEncodingPresetVideoSettingForUpdate](#)
- [VideoExportExtensionArgs](#)

第 41 次发布

发布时间：2021-11-16 08:08:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeResourceAuthorization](#)
 - 修改入参：Owner, Resource

第 40 次发布

发布时间：2021-11-01 08:07:26

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ExportVideoByEditorTrackData](#)
 - 新增入参：AspectRatio

第 39 次发布

发布时间：2021-09-15 08:03:30

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [PlatformInfo](#)
 - 新增成员：Status

第 38 次发布

发布时间：2021-08-12 08:03:28

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [MediaPreprocessOperation](#)

修改数据结构：

- [MediaReplacementInfo](#)
 - 新增成员：MediaType, MediaUrl, PreprocessOperation
 - 修改成员：MaterialId

第 37 次发布

发布时间：2021-07-16 08:03:02

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ExportVideoByEditorTrackData](#)
 - 新增入参：CoverData

第 36 次发布

发布时间：2021-07-08 08:02:59

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CopyProject](#)

第 35 次发布

发布时间：2021-06-22 08:03:11

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [TextReplacementInfo](#)

修改数据结构：

- [SlotReplacementInfo](#)
 - 新增成员：TextReplacementInfo

第 34 次发布

发布时间：2021-06-17 08:02:44

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [VideoEditTemplateMaterial](#)
 - 新增成员：PreviewVideoUrl

第 33 次发布

发布时间：2021-06-16 08:02:54

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [ProjectStreamConnectStatusChangedEvent](#)

修改数据结构：

- [EventContent](#)
 - 新增成员：ProjectStreamConnectStatusChangedEvent

第 32 次发布

发布时间：2021-05-26 08:02:29

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [CosPublishInputInfo](#)

修改数据结构：

- [ThirdPartyPublishInfo](#)
 - 新增成员：CosPublishInfo

第 31 次发布

发布时间：2021-05-24 08:02:19

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ParseEvent](#)

新增数据结构：

- [EventContent](#)
- [StorageNewFileCreatedEvent](#)

第 30 次发布

发布时间：2021-04-30 08:02:08

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 新增出参：RtmpPushInputInfoSet
- [HandleStreamConnectProject](#)
 - 新增出参：StreamInputRtmpPushUrl

第 29 次发布

发布时间：2021-04-28 08:03:12

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 新增入参：Mode
- [DescribeProjects](#)
 - 新增入参：Modes
- [ModifyProject](#)
 - 新增入参：Mode

第 28 次发布

发布时间：2021-04-27 08:02:11

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeAccounts](#)

新增数据结构：

- [AccountInfo](#)

第 27 次发布

发布时间：2021-04-23 08:01:59

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [HandleStreamConnectProject](#)

新增数据结构：

- [StreamConnectOutputInfo](#)
- [StreamConnectProjectInfo](#)

修改数据结构：

- [ProjectInfo](#)
 - 新增成员：StreamConnectProjectInfo

第 26 次发布

发布时间：2021-04-15 08:02:12

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 新增入参：RecordReplayProjectInput
- [ExportVideoEditProject](#)
 - 新增入参：CoverData

新增数据结构：

- [RecordReplayProjectInput](#)

第 25 次发布

发布时间：2021-04-07 08:02:01

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [SlotInfo](#)
- [VideoEditTemplateMaterial](#)

修改数据结构：

- [MaterialInfo](#)
 - 新增成员：VideoEditTemplateMaterial

第 24 次发布

发布时间：2021-04-05 08:02:04

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 新增入参：StreamConnectProjectInput

新增数据结构：

- [LivePullInputInfo](#)
- [MediaTransitionItem](#)
- [RtmpPushInputInfo](#)
- [StreamConnectOutput](#)
- [StreamConnectProjectInput](#)
- [StreamInputInfo](#)
- [VodPullInputInfo](#)

修改数据结构：

- [MediaTrackItem](#)
 - 新增成员：TransitionItem

第 23 次发布

发布时间：2021-03-22 08:02:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ExportVideoByTemplate](#)

新增数据结构：

- [MediaReplacementInfo](#)
- [SlotReplacementInfo](#)

第 22 次发布

发布时间：2021-03-17 08:02:14

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [MoveResource](#)

修改接口：

- [DeleteProject](#)
 - 新增入参：Operator
- [DescribeProjects](#)
 - 新增入参：Operator
- [DescribeTaskDetail](#)
 - 新增入参：Operator
- [DescribeTasks](#)
 - 新增入参：Operator
- [ModifyMaterial](#)
 - 修改入参：Owner

新增数据结构：

- [OtherMaterial](#)
- [PresetTagInfo](#)
- [ResourceInfo](#)

修改数据结构：

- [MaterialBasicInfo](#)
 - 新增成员：PresetTagSet, TagSet
- [MaterialInfo](#)
 - 新增成员：OtherMaterial

第 21 次发布

发布时间：2021-03-01 08:01:55

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [VideoEditProjectInput](#)
 - 新增成员：AspectRatio
- [VideoSegmentationProjectInput](#)
 - 新增成员：AspectRatio

第 20 次发布

发布时间：2021-01-19 08:01:58

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 修改入参：AspectRatio
- [ExportVideoEditProject](#)
 - 新增入参：Operator
- [ImportMediaToProject](#)
 - 新增入参：Operator

第 19 次发布

发布时间：2020-12-29 08:01:48

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeTeams](#)
 - 新增入参：Offset, Limit
 - 修改入参：TeamIds
 - 新增出参：TotalCount
- [ImportMaterial](#)
 - 新增入参：SourceType, ExternalMediaInfo
 - 修改入参：VodFileId
- [ImportMediaToProject](#)
 - 新增入参：SourceType, ExternalMediaInfo
 - 修改入参：VodFileId

新增数据结构：

- [ExternalMediaInfo](#)

修改数据结构：

- [AddMemberInfo](#)
 - 新增成员：Role

第 18 次发布

发布时间：2020-12-17 08:01:45

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribePlatforms](#)

新增数据结构：

- [PlatformInfo](#)

第 17 次发布

发布时间：2020-12-15 08:03:01

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ExportVideoByVideoSegmentationData](#)
- [GenerateVideoSegmentationSchemeByAi](#)

修改接口：

- [CreateProject](#)
 - 新增入参：VideoSegmentationProjectInput

新增数据结构：

- [VideoSegmentationProjectInput](#)

第 16 次发布

发布时间：2020-12-04 08:01:32

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [KuaishouPublishInfo](#)
- [PenguinMediaPlatformPublishInfo](#)
- [ThirdPartyPublishInfo](#)
- [WeiboPublishInfo](#)

修改数据结构：

- [CMEExportInfo](#)
 - 新增成员：ThirdPartyPublishInfos
- [VODExportInfo](#)
 - 新增成员：ThirdPartyPublishInfos
- [VideoEditProjectInput](#)
 - 新增成员：VideoEditTemplateId

第 15 次发布

发布时间：2020-10-30 08:01:35

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 新增入参：LiveStreamClipProjectInput, VideoEditProjectInput

新增数据结构：

- [AudioTrackItem](#)
- [EmptyTrackItem](#)
- [LiveStreamClipProjectInput](#)
- [MediaTrack](#)
- [MediaTrackItem](#)
- [VideoEditProjectInput](#)
- [VideoTrackItem](#)

第 14 次发布

发布时间：2020-08-24 08:01:09

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateProject](#)
 - 新增入参：Description, SwitcherProjectInput

新增数据结构：

- [SwitcherPgmOutputConfig](#)
- [SwitcherProjectInput](#)

第 13 次发布

发布时间：2020-08-20 08:04:39

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [VideoEditProjectOutput](#)
 - 新增成员：MaterialId

第 12 次发布

发布时间：2020-07-10 08:03:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [AudioMaterial](#)
 - 新增成员：OriginalUrl, VodFileId
- [ImageMaterial](#)
 - 新增成员：OriginalUrl, VodFileId
- [VideoMaterial](#)
 - 新增成员：OriginalUrl, VodFileId

第 11 次发布

发布时间：2020-05-26 08:03:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SearchMaterial](#)
 - 新增入参：Tags

新增数据结构：

- [MaterialTagInfo](#)

修改数据结构：

- [MaterialBasicInfo](#)
 - 新增成员：TagInfoSet

第 10 次发布

发布时间：2020-05-15 08:03:10

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLink](#)
 - 删除入参：Tags
- [SearchMaterial](#)
 - 删除入参：Tags

第 9 次发布

发布时间：2020-05-14 08:04:13

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ImportMaterial](#)
 - 删除入参：Tags
- [ModifyMaterial](#)
 - 删除入参：Tags

修改数据结构：

- [MaterialBasicInfo](#)
 - 删除成员：TagSet

第 8 次发布

发布时间：2020-04-30 08:03:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ExportVideoByEditorTrackData](#)

第 7 次发布

发布时间：2020-04-29 08:03:24

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [MaterialStatus](#)

删除数据结构：

- MaterialBaseInfo

修改数据结构：

- [AudioMaterial](#)
 - 新增成员：MaterialStatus
- [VideoEditProjectOutput](#)
 - 删除成员：MaterialBaseInfo
- [VideoMaterial](#)
 - 新增成员：MaterialStatus

第 6 次发布

发布时间：2020-04-13 08:02:23

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateLink](#)

第 5 次发布

发布时间：2020-04-10 08:02:25

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [MoveClass](#)

第 4 次发布

发布时间：2020-04-09 08:02:18

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AddTeamMember](#)
- [CreateClass](#)
- [CreateTeam](#)
- [DeleteClass](#)
- [DeleteMaterial](#)
- [DeleteTeam](#)
- [DeleteTeamMembers](#)
- [DescribeClass](#)
- [DescribeJoinTeams](#)
- [DescribeMaterials](#)
- [DescribeResourceAuthorization](#)
- [DescribeSharedSpace](#)
- [DescribeTeamMembers](#)
- [DescribeTeams](#)
- [FlattenListMedia](#)
- [GrantResourceAuthorization](#)
- [ImportMaterial](#)
- [ListMedia](#)
- [ModifyMaterial](#)
- [ModifyTeam](#)
- [ModifyTeamMember](#)
- [RevokeResourceAuthorization](#)
- [SearchMaterial](#)

修改接口：

- [DescribeProjects](#)
 - 新增入参：Sort
- [ModifyProject](#)
 - 新增入参：AspectRatio

新增数据结构：

- [AddMemberInfo](#)
- [AudioMaterial](#)
- [AuthorizationInfo](#)
- [Authorizer](#)
- [ClassInfo](#)
- [ImageMaterial](#)
- [IntegerRange](#)
- [JoinTeamInfo](#)
- [LinkMaterial](#)
- [LinkMaterialInfo](#)
- [MaterialBasicInfo](#)
- [MaterialInfo](#)
- [MediaImageSpriteInfo](#)

- [Resource](#)
- [SearchScope](#)
- [SortBy](#)
- [TeamInfo](#)
- [TeamMemberInfo](#)
- [TimeRange](#)
- [VideoMaterial](#)

第 3 次发布

发布时间：2020-03-27 08:01:55

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MediaMetaData](#)
 - 新增成员：Rotate

第 2 次发布

发布时间：2020-03-16 08:01:06

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ProjectInfo](#)
 - 新增成员：CoverUrl, CreateTime, UpdateTime

第 1 次发布

发布时间：2019-12-18 16:28:24

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateProject](#)
- [DeleteLoginStatus](#)
- [DeleteProject](#)
- [DescribeLoginStatus](#)
- [DescribeProjects](#)
- [DescribeTaskDetail](#)
- [DescribeTasks](#)
- [ExportVideoEditProject](#)
- [ImportMediaToProject](#)
- [ModifyProject](#)

新增数据结构：

- [AudioStreamInfo](#)
- [CMEEExportInfo](#)
- [Entity](#)
- [LoginStatusInfo](#)
- [MaterialBaseInfo](#)
- [MediaMetaData](#)
- [ProjectInfo](#)
- [TaskBaseInfo](#)

- [VODExportInfo](#)
- [VideoEditProjectOutput](#)
- [VideoStreamInfo](#)

简介

最近更新时间：2025-04-25 01:19:30

多媒体创作引擎（Cloud Media Engine，CME）是一站式的长短视频内容创作和协作平台，提供内容创作协作、专业内容交付、高清视频处理、媒资存储和管理、云端剪辑、多平台同步直播等能力

协作审片

高效精确的协作审片工具，可通过画笔、箭头、文字等形式逐帧表达视频制作建议，支持 PC、小程序等多端分享协同，极大提升视频审核及多端协同制作效率。

视频剪辑

便捷高效的云端视频剪辑工具，支持逐帧剪辑、画中画、帧定格等专业的剪辑能力，同时提供模板剪辑和小程序剪辑多种剪辑方式，以及海量在线版权素材。

云创网盘

智能易用的云端网盘，自动识别标签、人脸、语音等信息，提供团队协作存储空间，支持人员管理、直播录制、活动投稿等功能，极大提升文件管理和使用效率。

企业直播

支持多种类型输入源的切换和混流、支持实时编辑水印字幕、点播转直播、轮播列表、定时开播，多平台同步直播等多种专业级直播导播能力。

请您在使用多媒体创作引擎 API 前，确保已充分理解多媒体创作引擎的功能，如有问题，欢迎咨询。

API 概览

最近更新時間：2024-08-02 01:19:34

平台管理相關接口

接口名称	接口功能	频率限制（次/秒）
DescribeAccounts	获取账号信息	20
DescribePlatforms	获取平台列表	100

项目管理相關接口

接口名称	接口功能	频率限制（次/秒）
CopyProject	复制项目	20
CreateProject	创建项目	20
DeleteProject	删除项目	20
DescribeProjects	获取项目列表	20
ExportVideoEditProject	导出视频编辑项目	20
HandleMediaCastProject	操作媒体转推项目	20
HandleStreamConnectProject	操作云转推项目（废弃）	20
ImportMediaToProject	在项目中导入媒体	20
ModifyProject	修改项目	20

Headless媒体合成相關接口

接口名称	接口功能	频率限制（次/秒）
ExportVideoByEditorTrackData	使用视频合成协议导出视频	20
ExportVideoByTemplate	使用视频剪辑模板导出视频	20

任务管理相關接口

接口名称	接口功能	频率限制（次/秒）
DescribeTaskDetail	获取任务详情	20
DescribeTasks	获取任务列表	20

视频编辑项目相關接口

接口名称	接口功能	频率限制（次/秒）
ExportVideoByVideoSegmentationData	使用视频智能拆条数据导出视频	20
GenerateVideoSegmentationSchemeByAi	视频智能拆条	20

媒体资源管理相關接口

接口名称	接口功能	频率限制（次/秒）
FlattenListMedia	平铺浏览媒体	20

接口名称	接口功能	频率限制（次/秒）
CreateLink	创建链接	20
DeleteMaterial	删除媒体	20
DescribeMaterials	获取媒体详情	20
ImportMaterial	导入媒体	20
ListMedia	浏览媒体	20
ModifyMaterial	修改媒体信息	20
MoveResource	移动资源	20
SearchMaterial	搜索媒体	20

媒资授权相关接口

接口名称	接口功能	频率限制（次/秒）
GrantResourceAuthorization	发起媒资授权	20
DescribeResourceAuthorization	获取资源授权列表	20
DescribeSharedSpace	获取共享空间	20
RevokeResourceAuthorization	撤销资源授权	20

团队管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateTeam	创建团队	20
AddTeamMember	添加团队成员	20
DescribeJoinTeams	获取成员加入的团队列表	20
DescribeTeamMembers	获取团队成员信息	20
DescribeTeams	获取团队信息	20
ModifyTeam	修改团队信息	20
ModifyTeamMember	修改团队成员	20
DeleteTeam	删除团队	20
DeleteTeamMembers	删除团队成员	20

分类管理相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeClass	获取分类列表	20
CreateClass	创建分类	20
MoveClass	移动分类	20
DeleteClass	删除分类	20

登录态管理相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteLoginStatus	删除登录态	20
DescribeLoginStatus	查询登录态	20

参数模板相关接口

接口名称	接口功能	频率限制（次/秒）
CreateVideoEncodingPreset	创建视频编码配置	20
DeleteVideoEncodingPreset	删除视频编码配置	20
ModifyVideoEncodingPreset	修改视频编码配置	20
DescribeVideoEncodingPresets	查询视频编码配置	20

其他接口

接口名称	接口功能	频率限制（次/秒）
ParseEvent	回调事件内容解析	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 01:13:22

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 `cme.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `cme.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 `cme.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）	<code>cme.tencentcloudapi.com</code>
华南地区（广州）	<code>cme.ap-guangzhou.tencentcloudapi.com</code>
华东地区（上海）	<code>cme.ap-shanghai.tencentcloudapi.com</code>
华东地区（南京）	<code>cme.ap-nanjing.tencentcloudapi.com</code>
华北地区（北京）	<code>cme.ap-beijing.tencentcloudapi.com</code>
西南地区（成都）	<code>cme.ap-chengdu.tencentcloudapi.com</code>
西南地区（重庆）	<code>cme.ap-chongqing.tencentcloudapi.com</code>
港澳台地区（中国香港）	<code>cme.ap-hongkong.tencentcloudapi.com</code>
亚太东南（新加坡）	<code>cme.ap-singapore.tencentcloudapi.com</code>
亚太东南（雅加达）	<code>cme.ap-jakarta.tencentcloudapi.com</code>
亚太东南（曼谷）	<code>cme.ap-bangkok.tencentcloudapi.com</code>
亚太东北（首尔）	<code>cme.ap-seoul.tencentcloudapi.com</code>
亚太东北（东京）	<code>cme.ap-tokyo.tencentcloudapi.com</code>
美国东部（弗吉尼亚）	<code>cme.na-ashburn.tencentcloudapi.com</code>
美国西部（硅谷）	<code>cme.na-siliconvalley.tencentcloudapi.com</code>
南美地区（圣保罗）	<code>cme.sa-saopaulo.tencentcloudapi.com</code>
欧洲地区（法兰克福）	<code>cme.eu-frankfurt.tencentcloudapi.com</code>

注意：由于金融区和非金融区是隔离不通的，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）	<code>cme.ap-shanghai-fsi.tencentcloudapi.com</code>
华南地区（深圳金融）	<code>cme.ap-shenzhen-fsi.tencentcloudapi.com</code>

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 01:21:50

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 cme；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 。

参数名称	类型	必选	描述
Version	String	是	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。
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*****
***
```

签名方法 v3

最近更新时间：2024-12-25 01:22:06

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：

- 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。
- 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：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云API密钥](#) 的控制台页面。
- 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：

- 云服务器默认已开通，该接口很常用；
- 该接口是只读的，不会改变现有资源的状态；
- 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 *****。用户想查看广州区云服务器名为“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 中间号 (?) 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； 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 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n。 注意：content-type 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 charset 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。content-type 和 host 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 content-type;host;x-tc-action
HashedRequestPayload	请求正文（payload，即 body，此示例为 {"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编

字段名称	解释
	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 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

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 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 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=content-type;hos  
t;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=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
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.utcfromtimestamp(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"
.$credentialScope."\n"
.$hashedCanonicalRequest;
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}\nxc-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, 则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"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);

    #if 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++)
    {
        snprintf(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 OPENSSSL_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 OPENSSSL_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* 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, "\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\": \"instance-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相符合，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 01:22:06

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
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-gu
angzhou&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&R
egion=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 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的网络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
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 sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
    string sigOutParam = Sign(SECRET_KEY, sigInParam);
    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：2024-03-15 16:02:03

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 06:23:12

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。

平台管理相关接口

获取平台列表

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

- 支持获取所创建的所有平台列表信息；
- 支持获取指定的平台列表信息。

关于平台概念，请参见文档 [平台](#)。

默认接口请求频率限制：100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：DescribePlatforms。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platforms.N	否	Array of String	平台 Id 列表。如果不填，则不按平台 Id 进行过滤。 示例值：["1000000009"]
LicenseIds.N	否	Array of String	平台绑定的 License Id 列表。如果不填，则不按平台绑定的 License Id 进行过滤。 示例值：cmelid_8621ba0483c211ea90dc6c92bf621f6e
Offset	否	Integer	分页返回的起始偏移量，默认值：0。 示例值：0
Limit	否	Integer	分页返回的记录条数，默认值：10，最大值：20。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合查询条件的记录总数。 示例值：10
PlatformInfoSet	Array of PlatformInfo	平台信息列表。
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 获取所有创建的平台列表

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "PlatformInfoSet": [
      {
        "Platform": "1000000009",
        "Status": "Valid",
        "Description": "my_platform",
        "VodSubAppId": 140000303,
        "LicenseId": "cmelid_8621ba0483c211ea90dc6c92bf621f6e",
        "CreateTime": "2018-12-01T13:00:00Z",
        "UpdateTime": "2018-12-01T13:00:00Z"
      }
    ],
    "RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
  }
}
```

示例2 指定绑定的 License Id 获取平台列表

输入示例

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

{
  "LicenseIds": [
    "cmelid_8621ba0483c211ea90dc6c92bf621f6e"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "PlatformInfoSet": [
      {
        "Platform": "1000000009",
        "Status": "Valid",
        "Description": "my_platform",
        "VodSubAppId": 140000303,
```

```
"LicenseId": "cmelid_8621ba0483c211ea90dc6c92bf621f6e",
"CreateTime": "2018-12-01T13:00:00Z",
"UpdateTime": "2018-12-01T13:00:00Z"
}
],
"RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
}
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Limit	返回记录条数不合法。
UnauthorizedOperation	未授权操作。

获取账号信息

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取平台中所有的已注册账号。

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：DescribeAccounts。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Phone	否	String	手机号码。指定手机号获取账号信息，目前仅支持国内手机号，且号码不加地区码 +86 等。 示例值：15515555055
Offset	否	Integer	分页返回的起始偏移量，默认值：0。 示例值：0
Limit	否	Integer	分页返回的记录条数，默认值：10，最大值：20。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合搜索条件的记录总数。 示例值：10
AccountInfoSet	Array of AccountInfo	账号信息列表。
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，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 指定手机号获取用户信息

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeAccounts
&Platform=1000000009
&Phone=15517177077
&Offset=0
```

```
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "AccountInfoSet": [
      {
        "UserId": "72392ca4-2c79-44f3-ad5a-63cbfa72367c",
        "Phone": "15515555055",
        "Nick": "Nick",
        "Status": "Normal"
      }
    ],
    "RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
  }
}
```

示例2 分布获取平台所有账号信息

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeAccounts
&Platform=1000000009
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "AccountInfoSet": [
      {
        "UserId": "32392ca4-2c79-44f3-ad5a-63cbfa72367c",
        "Phone": "15515555056",
        "Nick": "小李",
        "Status": "Normal"
      },
      {
        "UserId": "62392ca4-2c79-44f3-ad5a-63cbfa72365d",
        "Phone": "13415555056",
        "Nick": "小张",
        "Status": "Normal"
      }
    ],
    "RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21ad"
  }
}
```

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。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Offset	分页偏移不合法。
InvalidParameterValue.Platform	平台参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
ResourceNotFound.PlatformNotFound	平台不存在。
UnauthorizedOperation	未授权操作。

项目管理相关接口

修改项目

最近更新时间：2025-04-25 01:19:23

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

修改项目信息。

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：ModifyProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectId	是	String	项目 Id。 示例值：67e105de4fb8950001052fff
Name	否	String	项目名称，不可超过30个字符。 示例值：first_project
AspectRatio	否	String	画布宽高比，值为视频编辑项目画布宽与高的像素值的比值，如 16:9、9:16 等。 示例值：16:9
Owner	否	Entity	项目所有者。目前仅支持个人项目，不支持团队项目。
Mode	否	String	项目模式，一个项目可以有多种模式并相互切换。 当 Category 为 VIDEO_EDIT 时，可选模式有： - Default：默认模式，即普通视频编辑项目。 - VideoEditTemplate：剪辑模板制作模式，用于制作剪辑模板。 示例值：Default

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 修改项目名称

输入示例

```
https://cme.tencentcloudapi.com/?Action=ModifyProject
&Platform=1000000009
&ProjectId=3f1699f3f97b9f0001920f29
&Name=项目名称
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5d"
  }
}
```

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。
InvalidParameterValue.AspectRatio	画布宽高比不合法。
InvalidParameterValue.Name	名称字段编辑错误，需使用 UTF8 编码。
InvalidParameterValue.NameLenLimt	名称字段长度超长，限制 30 个中文字符。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。

错误码	描述
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取项目列表

最近更新时间：2025-04-25 01:19:24

-

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

支持根据多种条件过滤出项目列表。

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：DescribeProjects。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectIds.N	否	Array of String	项目 Id 过滤参数列表，最大支持20个项目 Id 过滤。如果不填不需要项目 Id 进行过滤。 示例值：["a82b830e765aa979"]
AspectRatioSet.N	否	Array of String	画布宽高比过滤参数列表。如果不填则不用画布宽高比进行过滤。 示例值：["16:9"]
CategorySet.N	否	Array of String	项目类型过滤参数列表，取值有： - VIDEO_EDIT：视频编辑。 - SWITCHER：导播台。 - VIDEO_SEGMENTATION：视频拆条。 - STREAM_CONNECT：云转推。 - RECORD_REPLAY：录制回放。 - MEDIA_CAST：点播转直播。 注：如果不填则不使用项目类型进行过滤。 示例值：["VIDEO_EDIT"]
Modes.N	否	Array of String	项目模式过滤参数列表，一个项目可以有多种模式并相互切换。 当 Category 为 VIDEO_EDIT 时，可选模式有： - Default：默认模式。 - VideoEditTemplate：视频编辑模板制作模式。 注：不填不使用项目模式进行过滤。 示例值：["Default"]
Sort	否	SortBy	结果排序方式，支持下列排序字段： - CreateTime：创建时间； - UpdateTime：更新时间。 注：如不填，则使用项目创建时间倒序排列。
Owner	否	Entity	项目所有者，目前仅支持个人项目过滤。
Offset	否	Integer	分页返回的起始偏移量，默认值：0。 示例值：0
Limit	否	Integer	分页返回的记录条数，默认值：10。 示例值：10

参数名称	必选	类型	描述
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以查询一切用户项目信息。如果指定操作者，则操作者必须为项目所有者。 示例值：378b753f-f91f-4c42-8fa3-34a2a3bf3c56

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的记录总数。 示例值：1
ProjectInfoSet	Array of ProjectInfo	项目信息列表。
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 获取项目列表

输入示例

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

{
  "Platform": "1000000009",
  "ProjectIds": [
    "cmepid_60599df66a6b440001518159"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "ProjectInfoSet": [
      {
        "ProjectId": "cmepid_60599df66a6b440001518159",
        "Name": "my_project",
        "AspectRatio": "16:9",
        "Category": "VIDEO_EDIT",
        "Owner": {
          "Type": "PERSON",
          "Id": "user_31345df76a6b44104571916a"
        },
        "StreamConnectProjectInfo": null,
        "MediaCastProjectInfo": null,
        "UpdateTime": "2020-11-13T06:41:34.808Z",
        "CreateTime": "2020-11-13T06:41:34.808Z",
        "CoverUrl": ""
      }
    ]
  }
}
```

```
}
],
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5c"
}
}
```

示例2 获取云转推项目信息

输入示例

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

{
  "Platform": "1000000009",
  "CategorySet": [
    "STREAM_CONNECT"
  ],
  "ProjectIds": [
    "cmepid_60599df66a6b440001518169"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "ProjectInfoSet": [
      {
        "ProjectId": "cmepid_60599df66a6b440001518169",
        "Name": "my_project",
        "AspectRatio": "",
        "Category": "STREAM_CONNECT",
        "Owner": {
          "Type": "PERSON",
          "Id": "user_31345df76a6b44104571916a"
        },
        "UpdateTime": "2020-11-13T06:41:34.808Z",
        "CreateTime": "2020-11-13T06:41:34.808Z",
        "CoverUrl": "",
        "MediaCastProjectInfo": null,
        "StreamConnectProjectInfo": {
          "Status": "Working",
          "CurrentInputEndpoint": "Main",
          "CurrentStartTime": "2020-11-13T06:41:34.808Z",
          "CurrentStopTime": "2020-11-13T08:41:34.808Z",
          "LastStopTime": "2020-11-12T07:41:34.808Z",
          "MainInput": {
            "InputType": "LivePull",
            "LivePullInputInfo": {
              "InputUrl": "rtmp://liveplay.video-studio.myqcloud.com/output/1250000001-600e8e7fb1cc1c0001293759?txSecret=d5f0d5e9eadda723d7fbf6822313b1c0&txTime=6017C8FF"
            }
          }
        }
      }
    ]
  }
}
```

```

    }
  },
  "OutputSet": [
    {
      "PushSwitch": "On",
      "StreamConnectOutput": {
        "Id": "124522",
        "Type": "URL",
        "PushUrl": "rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600e8e66194ef500012d9b08",
        "Name": "my_output"
      }
    }
  ]
},
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5c"
}
}

```

示例3 获取点播转直播项目信息

输入示例

```

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

{
  "Platform": "1000000009",
  "CategorySet": [
    "MEDIA_CAST"
  ],
  "ProjectIds": [
    "cmepid_60599df66a6b440001518169"
  ]
}

```

输出示例

```

{
  "Response": {
    "TotalCount": 1,
    "ProjectInfoSet": [
      {
        "ProjectId": "cmepid_60599df66a6b440001518169",
        "Name": "my_project",
        "AspectRatio": "",
        "Category": "MEDIA_CAST",
        "Owner": {
          "Type": "PERSON",
          "Id": "user_31345df76a6b44104571916a"
        },
        "UpdateTime": "2020-11-13T06:41:34.808Z",

```

```
"CreateTime": "2020-11-13T06:41:34.808Z",
"CoverUrl": "",
"StreamConnectProjectInfo": null,
"MediaCastProjectInfo": {
  "Status": "Working",
  "StartTime": "2020-11-13T06:41:34.808Z",
  "StopTime": "2020-11-14T06:41:34.808Z",
  "Duration": 60,
  "SourceInfos": [
    {
      "Id": "fsdf",
      "Type": "CME",
      "MaterialId": "a123",
      "Offset": 10,
      "Duration": 60,
      "FileId": "bee50aae5845",
      "Url": "http://1000000009.vod2.myqcloud.com/6cac69e3vodcq123/c53b8681557667123/123.mp4"
    },
    {
      "Id": "sdfs",
      "Type": "VOD",
      "MaterialId": "a345",
      "Offset": 10,
      "Duration": 60,
      "FileId": "bee50aae1234",
      "Url": "http://1000000009.vod2.myqcloud.com/6cac69e3vodcq123/c53b8681557667123/345.mp4"
    }
  ],
  "DestinationInfos": [
    {
      "Id": "2342342",
      "Name": "my_output",
      "PushUrl": "rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600e8e66194ef500012d9b08"
    }
  ],
  "PlaySetting": {
    "LoopCount": 2,
    "EndTime": "2023-09-13T02:19:06Z",
    "AutoStartTime": "0001-01-01T00:00:00Z"
  },
  "OutputMediaSetting": {
    "VideoSetting": {
      "Width": 1920,
      "Height": 1080,
      "Bitrate": 2500,
      "FrameRate": 25
    }
  }
},
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5c"
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AspectRatioSet	画布宽高比不合法。
InvalidParameterValue.CategorySet	项目类别不合法。
InvalidParameterValue.Limit	返回记录条数不合法。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.SortOrder	排序方式无效。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

删除项目

最近更新时间：2025-04-25 01:19:24

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

删除项目。处于推流状态的云转推和点播转直播项目不允许删除，若强行调用删除项目接口会返回失败。

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：DeleteProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectId	是	String	要删除的项目 Id。 示例值：1112
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以删除一切项目。如果指定操作者，则操作者必须为项目所有者。 示例值：user_id_100001

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 删除项目

删除项目

输入示例

```
https://cme.tencentcloudapi.com/?Action=DeleteProject
&Platform=1000000009
&ProjectId=cmeqid_5f16967b64436100015fb026
&<公共请求参数>
```

输出示例

```
{
  "Response": {
```

```
"RequestId": "a64cbb5b-b809-4061-8c36-7369b98e8e6x"
}
```

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
FailedOperation	操作失败。
FailedOperation.SwitcherOnWorking	非法操作，导播中正在工作中。
InternalError	内部错误。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

创建项目

最近更新时间：2025-06-04 01:21:14

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

创建多媒体创作引擎项目，目前支持的项目类型有：

- 视频剪辑项目：用于普通视频剪辑；
- 直播剪辑项目：用于直播流剪辑；
- 导播台项目：用于云导播台；
- 视频拆条：用于视频拆条；
- 录制回放项目：用于直播录制回放；
- 媒体转推项目：用于媒体文件转直播输出。

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：CreateProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。平台概念，请参见文档 平台 。 示例值：1000000009
Name	是	String	项目名称，不可超过30个字符。 示例值：新建项目
Owner	是	Entity	项目归属者，即项目的所有者，后续操作只有该所有者有权限操作。 注：目前所有项目只能设置归属个人，暂不支持团队项目。
Category	是	String	项目类别，取值有： • VIDEO_EDIT：视频编辑。 • SWITCHER：导播台。 • VIDEO_SEGMENTATION：视频拆条。 • STREAM_CONNECT：云转推。 • RECORD_REPLAY：录制回放。 • MEDIA_CAST：媒体转推。 示例值：VIDEO_EDIT
Mode	否	String	项目模式，一个项目可以有多种模式并相互切换。 当 Category 为 VIDEO_EDIT 时，可选模式有： • Default：默认模式，即普通视频编辑项目。 • VideoEditTemplate：剪辑模板制作模式，用于制作剪辑模板。 注：不填则为默认模式。 示例值：Default
Description	否	String	项目描述信息。 示例值：新建

参数名称	必选	类型	描述
SwitcherProjectInput	否	SwitcherProjectInput	导播台项目输入信息，仅当项目类型为 SWITCHER 时必填。
LiveStreamClipProjectInput	否	LiveStreamClipProjectInput	直播剪辑项目输入信息，暂未开放，请勿使用。
VideoEditProjectInput	否	VideoEditProjectInput	视频编辑项目输入信息，仅当项目类型为 VIDEO_EDIT 时必填。
VideoSegmentationProjectInput	否	VideoSegmentationProjectInput	视频拆条项目输入信息，仅当项目类型为 VIDEO_SEGMENTATION 时必填。
StreamConnectProjectInput	否	StreamConnectProjectInput	云转推项目输入信息，仅当项目类型为 STREAM_CONNECT 时必填。
RecordReplayProjectInput	否	RecordReplayProjectInput	录制回放项目输入信息，仅当项目类型为 RECORD_REPLAY 时必填。
MediaCastProjectInput	否	MediaCastProjectInput	媒体转推项目输入信息，仅当项目类型为 MEDIA_CAST 时必填。

3. 输出参数

参数名称	类型	描述
ProjectId	String	项目 Id。 示例值：60336ab29997a500019xxx
RtmpPushInputInfoSet	Array of RtmpPushInputInfo	• 当 Catagory 为 STREAM_CONNECT 时，数组返回长度为2，第0个代表主输入源推流信息，第1个代表备输入源推流信息。只有当各自输入源类型为推流时才有有效内容。
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 创建普通剪辑项目

输入示例

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

{
  "Category": "VIDEO_EDIT",
  "Platform": "1000000009",
  "Owner": {
    "Type": "PERSON",
    "Id": "user_id_61978823e6a253000100fb0f"
  },
  "Name": "first_project",
  "VideoEditProjectInput": {
    "AspectRatio": "16:9"
  }
}
```

输出示例

```
{
  "Response": {
```

```
"ProjectId": "cmepid_5f16967b64436100015fb025",
"RtmpPushInputInfoSet": [],
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
}
}
```

示例2 创建一个导播台项目

输入示例

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

{
  "Category": "SWITCHER",
  "Platform": "1000000009",
  "Owner": {
    "Type": "PERSON",
    "Id": "user_id_61978823e6a253000100fb0f"
  },
  "Name": "导播台",
  "SwitcherProjectInput": {
    "PgmOutputConfig": {
      "TemplateId": "10001"
    }
  }
}
```

输出示例

```
{
  "Response": {
    "ProjectId": "3f1699f3f97b9f0001920f29",
    "RtmpPushInputInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5c"
  }
}
```

示例3 使用初始轨道创建项目

输入示例

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

{
  "Category": "VIDEO_EDIT",
  "Platform": "1000000009",
  "Owner": {
```

```
"Type": "PERSON",
"Id": "user_id_61978823e6a253000100fb0f"
},
"Name": "视频剪辑项目并初始化轨道",
"VideoEditProjectInput": {
  "InitTracks": [
    {
      "TrackItems": [
        {
          "Type": "Video",
          "VideoItem": {
            "Duration": "10",
            "SourceMedia": "52858908113623182311",
            "SourceMediaStartTime": "0",
            "SourceType": "VOD"
          }
        }
      ]
    },
    {
      "Type": "Video"
    }
  ],
  "AspectRatio": "16:9"
}
}
```

输出示例

```
{
  "Response": {
    "ProjectId": "3f1699f3f97b9f0001920f31",
    "RtmpPushInputInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e6a"
  }
}
```

示例4 使用剪辑模板创建项目

输入示例

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

{
  "Category": "VIDEO_EDIT",
  "Platform": "1000000009",
  "Owner": {
    "Type": "PERSON",
    "Id": "user_id_61978823e6a253000100fb0f"
  },
  "Name": "剪辑模板项目",
  "VideoEditProjectInput": {
    "AspectRatio": "16:9",
    "VideoEditTemplateId": "61385efc24827f05859d3765@Public@CME"
```

```
}  
}
```

输出示例

```
{  
  "Response": {  
    "ProjectId": "3f1699f3f97b9f0001920f30",  
    "RtmpPushInputInfoSet": [],  
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5a"  
  }  
}
```

示例5 创建一个云转推项目，初始化输入源为直播拉流

云转推项目初始化输入输出源

输入示例

```
POST / HTTP/1.1  
Host: cme.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: CreateProject  
<公共请求参数>  
  
{  
  "Category": "STREAM_CONNECT",  
  "Platform": "1000000009",  
  "Owner": {  
    "Type": "PERSON",  
    "Id": "user_id_61978823e6a253000100fb0f"  
  },  
  "Name": "stream_connect",  
  "StreamConnectProjectInput": {  
    "MainInput": {  
      "LivePullInputInfo": {  
        "InputUrl": "rtmp://liveplay.video-studio.myqcloud.com/output/1250000001-600e8e7fb1cc1c0001293759"  
      },  
      "InputType": "LivePull"  
    },  
    "Outputs": [  
      {  
        "PushUrl": "rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600e8e66194ef500012d9b08"  
      }  
    ]  
  }  
}
```

输出示例

```
{  
  "Response": {  
    "ProjectId": "3f1699f3f97b9f0001920f29",  
    "RtmpPushInputInfoSet": [],  
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"  
  }  
}
```

```
}  
}
```

示例6 创建一个云转推项目，初始化输入源为直播推流

云转推项目初始化输入输出源

输入示例

```
POST / HTTP/1.1  
Host: cme.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: CreateProject  
<公共请求参数>  
  
{  
  "Category": "STREAM_CONNECT",  
  "Platform": "1000000009",  
  "Owner": {  
    "Type": "PERSON",  
    "Id": "user_id_61978823e6a253000100fb0f"  
  },  
  "Name": "云转推",  
  "StreamConnectProjectInput": {  
    "MainInput": {  
      "InputType": "RtmpPush",  
      "RtmpPushInputInfo": {  
        "ExpiredSecond": "3600"  
      }  
    },  
    "Outputs": [  
      {  
        "PushUrl": "rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600e8e66194ef500012d9b08"  
      }  
    ]  
  }  
}
```

输出示例

```
{  
  "Response": {  
    "ProjectId": "3f1699f3f97b9f0001920f29",  
    "RtmpPushInputInfoSet": [  
      {  
        "ExpiredSecond": 3600,  
        "PushUrl": "rtmp://livepush-xx.video-studio.myqcloud.com/output/1250000001-6086674e265b450001837f8e?txSecret=4478cfdfe0fd0eb3820705aebaa328xx&txTime=608FA1CE"  
      },  
      {  
        "ExpiredSecond": 0,  
        "PushUrl": ""  
      }  
    ],  
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"  
  }  
}
```

```
}  
}
```

示例7 创建一个点播转直播项目，播放一次，输出视频的帧率为25帧，码率为2500K，视频大小为1920*1080。

输入示例

```
POST / HTTP/1.1  
Host: cme.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: CreateProject  
<公共请求参数>  
  
{  
  "Category": "MEDIA_CAST",  
  "Platform": "1000000009",  
  "Owner": {  
    "Type": "PERSON",  
    "Id": "user_id_61978823e6a253000100fb0f"  
  },  
  "Name": "点播转直播项目",  
  "MediaCastProjectInput": {  
    "PlaySetting": {  
      "LoopCount": "1"  
    },  
    "OutputMediaSetting": {  
      "VideoSetting": {  
        "Width": "1920",  
        "FrameRate": "25",  
        "Bitrate": "2500",  
        "Height": "1080"  
      }  
    },  
    "SourceInfos": [  
      {  
        "Type": "VOD",  
        "FileId": "122"  
      },  
      {  
        "Type": "CME",  
        "MaterialId": "aaa12323"  
      }  
    ],  
    "DestinationInfos": [  
      {  
        "PushUrl": "rtmp://livetest.com/live/a?t=11",  
        "Name": "media_name"  
      }  
    ]  
  }  
}
```

输出示例

```
{  
  "Response": {
```

```
"ProjectId": "3f1699f3f97b9f0001920f31",
"RtmpPushInputInfoSet": [],
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e6a"
}
}
```

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
FailedOperation.CreateRecordTask	创建录制任务失败。
FailedOperation.RecordNotSupport	直播录制功能暂未对使用腾讯云点播存储的平台开放。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AspectRatio	画布宽高比不合法。
InvalidParameterValue.Category	项目类别不合法。
InvalidParameterValue.ExternalMediaInfoNotExist	原始媒资信息不存在。
InvalidParameterValue.MaterialId	媒体 ID 无效。
InvalidParameterValue.Name	名称字段编辑错误，需使用 UTF8 编码。
InvalidParameterValue.NameLenLimit	名称字段长度超长，限制 30 个中文字符。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.SwitcherProjectInput	导播台项目输入信息无效。
InvalidParameterValue.TrackItem	轨道元素无效。

错误码	描述
InvalidParameterValue.VodSubAppid	绑定的点播子应用无效。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

在项目中导入媒体

最近更新时间：2025-04-25 01:19:23

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

将云点播中的媒资或者用户自有媒资文件添加到项目中与项目关联，供后续视频编辑使用。目前仅视频编辑项目和智能视频拆条项目有效。

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：ImportMediaToProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectId	是	String	项目 Id。 示例值：600aae0bb1cc1c000129214b
SourceType	否	String	导入媒资类型，取值： - VOD：云点播文件； - EXTERNAL：媒资绑定。 注意：如果不填默认为云点播文件，如果媒体存储在非腾讯云点播中，都需要使用媒资绑定。 示例值：VOD
VodFileId	否	String	云点播媒资文件 Id，当 SourceType 取值 VOD 或者缺省的时候必填。 示例值：5285890813075978720
ExternalMediaInfo	否	ExternalMediaInfo	原始媒资文件信息，当 SourceType 取值 EXTERNAL 的时候必填。
Name	否	String	媒体名称，不能超过30个字符。如果不填，则媒体名称为点播媒资文件名称。 示例值：素材一
PreProcessDefinition	否	Integer	媒体预处理配置 ID，取值： 10：进行视频编辑预处理。 注意：如果填0或者不填则不进行处理，如果原始视频不可在浏览器直接播放将无法在编辑页面编辑。 示例值：10
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以向所有视频编辑项目导入媒体；如果指定操作者，则操作者必须为项目所有者。 示例值：abf00b8b-9482-4c34-b72c-bea5da0a27ce

3. 输出参数

参数名称	类型	描述
MaterialId	String	媒体 Id。

参数名称	类型	描述
		示例值：600a2c94e5c6aa0001b9ef78
TaskId	String	媒体预处理任务 ID，如果未指定发起预处理任务则为空。 示例值：1900001191-tfusion-5e299c12c63f6af4c95f34159f19137345t0
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 在项目中导入外部媒体

在项目中导入一个 COS 桶中媒体，并发起预处理。

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMediaToProject
&Platform=1000000009
&ProjectId=cmepid_6f16967b64436100015fb046
&SourceType=EXTERNAL
&Name=视频2
&ExternalMediaInfo.MediaKey=example-folder/example-object.mp4
&PreProcessDefinition=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "248208104522663991",
    "TaskId": "125xxxxx65-procedurev2-3ab512f07530b88bc1aabb01fac345bcd",
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e738"
  }
}
```

示例2 在项目中导入云点播文件

在项目中导入云点播文件，并发起预处理。

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMediaToProject
&Platform=1000000009
&ProjectId=cmepid_6f16967b64436100015fb046
&SourceType=VOD
&VodFileId=52858908114334690679
&Name=视频1
&PreProcessDefinition=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "248208104522663991",
    "TaskId": "125xxx65-procedurev2-bfffb15f07530b57bc1aabb01fac74bca",
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e56"
  }
}
```

```
}  
}
```

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
FailedOperation	操作失败。
FailedOperation.InvalidVodUser	云点播用户无效。
InternalServerError.CreateTask	发起预处理任务失败。
InvalidParameter	参数错误。
InvalidParameterValue.ExternalMediaInfoNotExist	原始媒资信息不存在。
InvalidParameterValue.MaterialId	媒体 ID 无效。
InvalidParameterValue.Name	名称字段编辑错误，需使用 UTF8 编码。
InvalidParameterValue.NameLenLimit	名称字段长度超长，限制 30 个中文字符。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.PreProcessDefinition	预处理模板 ID 无效。
InvalidParameterValue.ProjectId	项目不存在。
InvalidParameterValue.VodFileId	参数缺少 VodFile。
InvalidParameterValue.VodFileNotExist	点播文件不存在。
InvalidParameterValue.VodSubAppid	绑定的点播子应用无效。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

导出视频编辑项目

最近更新时间：2025-04-25 01:19:24

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

导出视频编辑项目，支持指定输出的模板。

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：ExportVideoEditProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
ProjectId	是	String	项目 Id。 示例值：cmepid_503f1c85000152217601dfa6
Definition	是	Integer	视频编码配置 ID，支持自定义创建，推荐优先使用系统预置的导出配置。 10：分辨率为 480P，输出视频格式为 MP4； 11：分辨率为 720P，输出视频格式为 MP4； 12：分辨率为 1080P，输出视频格式为 MP4。 示例值：10
ExportDestination	是	String	导出目标，指定导出视频的目标媒资库，可取值有： - CME：多媒体创作引擎，即导出为多媒体创作引擎媒资库，此导出目标在云点播媒资库依然可见； - VOD：云点播，即导出为云点播媒资库，此导出目标在多媒体创作引擎媒资库将不可见。 示例值：CME
CoverData	否	String	视频封面图片文件（如 jpeg、png 等）进行 Base64 编码后的字符串，仅支持 gif、jpeg、png 三种图片格式，原图片文件不能超过 2 M 大小。 示例值：Y292ZXJJbWFnZQ==
CMEExportInfo	否	CMEExportInfo	导出的多媒体创作引擎媒体信息。当导出目标为 CME 时必填。
VODExportInfo	否	VODExportInfo	导出的云点播媒资信息。当导出目标为 VOD 时必填。
ExportExtensionArgs	否	VideoExportExtensionArgs	视频导出扩展参数。可以覆盖导出模板中的参数，灵活的指定导出规格及参数。
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验项目导出权限。 示例值：9b92a8e7-a5ef-4e28-d72e-4eff571fabf0

3. 输出参数

参数名称	类型	描述
TaskId	String	任务 Id。 示例值：24820810452266399
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 导出视频到云点播

输入示例

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

{
  "Platform": "1000000009",
  "ProjectId": "cmepid_eb94392053f840539229ed9",
  "Definition": 10,
  "ExportDestination": "VOD",
  "VODExportInfo": {
    "Name": "在线编辑视频",
    "ClassId": 10
  },
  "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125xxxxxx-tfusion-19f876370688d7d08eb10d6e4ecd1cd9",
    "RequestId": "725d0791-97d7-4e4a-8740-e2128b98ddf0"
  }
}
```

示例2 导出视频到云点播并发布到企鹅号

输入示例

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

{
  "Platform": "1000000009",
  "Definition": 10,
  "ProjectId": "cmepid_eb94392053f840539229ed9",
  "ExportDestination": "VOD",
  "VODExportInfo": {
    "Name": "在线编辑视频",
    "ClassId": 10,
    "ThirdPartyPublishInfos": [
      {
        "ChannelMaterialId": "5fd8ad3d628dc30001bd0895",
        "PenguinMediaPlatformPublishInfo": {
          "Title": "企鹅视频",
          "Description": "新闻",
          "Tags": [
            "娱乐"
          ],
          "Category": 100
        }
      }
    ]
  }
}
```

```
}
}
],
},
"Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125xxxxxxx-tfusion-00f08kd8e6370688d7d08eb10d6e4ec9kdu73",
    "RequestId": "789d5d0791-97d7-4e4a-8740-e2128b98ddf0"
  }
}
```

示例3 导出视频到多媒体创作引擎媒资库

输入示例

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

{
  "Platform": "1000000009",
  "ProjectId": "cmepid_eb94392053f840539229ed9",
  "Definition": 10,
  "ExportDestination": "CME",
  "CMEEExportInfo": {
    "Owner": {
      "Type": "PERSON",
      "Id": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
    }
  },
  "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125xxxxxxx-tfusion-62f876370688d7d08eb10d6e4ecd1cd9",
    "RequestId": "233d0791-97d7-4e4a-8740-e2128b032df8"
  }
}
```

示例4 导出视频到多媒体创作引擎媒资库并发布到企鹅号

输入示例

```
POST / HTTP/1.1
Host: cme.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: ExportVideoEditProject
<公共请求参数>

{
  "Platform": "1000000009",
  "Definition": 10,
  "ExportDestination": "CME",
  "ProjectId": "cmepid_eb94392053f840539229ed9",
  "CMEExportInfo": {
    "Owner": {
      "Type": "PERSON",
      "Id": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
    },
    "Name": "新建项目",
    "ClassPath": "/成片",
    "ThirdPartyPublishInfos": [
      {
        "ChannelMaterialId": "5fd8ad3d628dc30001bd0895",
        "PenguinMediaPlatformPublishInfo": {
          "Title": "企鹅视频",
          "Description": "新闻",
          "Tags": [
            "娱乐"
          ],
          "Category": 100
        }
      }
    ],
    "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
  }
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125xxxxxxx-tfusion-82f876370688d7d08eb10d6e4ecd1cd9",
    "RequestId": "733d0791-97d7-4e4a-8740-e2128b032df2"
  }
}
```

示例5 导出视频多媒体创作引擎媒资库并指定导出视频的码率

导出视频多媒体创作引擎媒资库并指定导出视频的码率为 3M bps

输入示例

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

{
  "Platform": "1000000009",
  "ProjectId": "cmepid_eb94392053f840539229ed9",
```

```
"Definition": 10,
"ExportDestination": "CME",
"CMEExportInfo": {
  "Owner": {
    "Type": "PERSON",
    "Id": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
  }
},
"ExportExtensionArgs": {
  "VideoBitrate": 3145728
},
"Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125xxxxxxx-tfusion-72f876370688d7d08eb10d6e4ecd1cd9",
    "RequestId": "533d0791-97d7-4e4a-8740-e2128b032df9"
  }
}
```

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
FailedOperation	操作失败。
InternalError	内部错误。
InternalError.CreateTask	发起预处理任务失败。

错误码	描述
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.Definition	模板 ID 无效。
InvalidParameterValue.ExportDestination	导出目标不合法。
InvalidParameterValue.MaterialId	媒体 ID 无效。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
InvalidParameterValue.ThirdyPartyPublishChannelId	发布通道 ID 无效。
InvalidParameterValue.TrackData	轨道数据无效。
InvalidParameterValue.VodSubAppid	绑定的点播子应用无效。
LimitExceeded.BillItemStorage	存储空间不足，请查看会员权益，购买资源包或会员版本后继续使用。
LimitExceeded.BillItemVideoEditExportDuration	您已达到导出视频时长上限，若还需使用，请升级会员版本或购买资源包。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

操作云转推项目（废弃）

最近更新时间：2025-05-27 01:20:42

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

本接口废弃，可创建媒体转推项目替代，操作媒体转推项目可使用 [HandleMediaCastProject](#) 接口 实现。

操作类型

- `AddInput`（添加输入源），包括：
 - 添加直播拉流输入源，参见 [示例1](#)；
 - 添加直播推流输入源，参见 [示例2](#)；
 - 添加点播拉流输入源，参见 [示例3](#)、[示例4](#)；
- `DeleteInput`（删除输入源），参见 [示例5](#)；
- `ModifyInput`（修改输入源），参见 [示例6](#)；
- `AddOutput`（添加输出源），参见 [示例7](#)；
- `DeleteOutput`（删除输出源），参见 [示例8](#)；
- `ModifyOutput`（修改输出源），参见 [示例9](#)；
- `Start`（开启转推），参见 [示例10](#)；
- `Stop`（停止转推），参见 [示例11](#)；
- `SwitchInput`（切换输入源），参见 [示例12](#)；
- `ModifyCurrentStopTime`（修改当前计划结束时间），参见 [示例13](#)；
- `DescribeInputPlayInfo`（查询播放进度），参见 [示例14](#)。

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： <code>HandleStreamConnectProject</code> 。
Version	是	String	公共参数 ，本接口取值： <code>2019-10-29</code> 。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值： <code>1000000009</code>
ProjectId	是	String	云转推项目 Id。 示例值： <code>67e105de4fb8950001052fff</code>
Operation	是	String	请参考 操作类型 示例值： <code>AddInput</code>
InputInfo	否	StreamInputInfo	转推输入源操作参数。具体操作方式详见 操作类型 及下文示例。
InputEndpoint	否	String	主备输入源标识，取值有： • Main：主源； • Backup：备源。 示例值： <code>Main</code>
OutputInfo	否	StreamConnectOutput	转推输出源操作参数。具体操作方式详见 操作类型 及下文示例。

参数名称	必选	类型	描述
CurrentStopTime	否	String	云转推当前预计结束时间，采用 ISO 日期格式 。具体操作方式详见 操作类型 及下文示例。 示例值：2021-03-23T07:51:18.029Z
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以操作所有云转推项目。如果指定操作者，则操作者必须为项目所有者。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
StreamInputRtmpPushUrl	String	输入源推流地址，当 Operation 取值 AddInput 且 InputType 为 RtmpPush 类型时有效。 示例值：rtmp://livepush.video-studio.myqcloud.com/output/1250000001-6086
VodPullInputPlayInfo	VodPullInputPlayInfo	点播输入源播放进度信息，当 Operation 取值 DescribeInputPlayInfo 且 InputType 为 VodPull 类型时有效。 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 修改输入源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyInput
InputEndpoint	是	String	Main 或 Backup
InputInfo	是	StreamInputInfo	输入源

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyInput",
  "InputEndpoint": "Main",
  "InputInfo": {
    "VodPullInputInfo": {
      "LoopTimes": "-1",
      "InputUrls": [
        "https://1810000001.vod2.myqcloud.com/b64e98advodcq1810000001/85f47b535285890787805464143/test.mp4",
        "https://1810000001.vod2.myqcloud.com/b64e98advodcq1810000001/85f47b535285890787805464143/AfJuJAzie5QA.mp4"
      ]
    },
    "InputType": "VodPull"
  },
}
```

```
"ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "983cbb5b-b809-4061-8c45-7469b64e8e41"
  }
}
```

示例2 修改输出源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyOutput
OutputInfo	是	StreamConnectOutput	输出源，Id 字段必填，从获取项目列表中拿到对应输出源的 Id

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyOutput",
  "OutputInfo": {
    "PushUrl": "rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600e8e66194ef500012d9b08xxxx",
    "Id": "12357877"
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "832dbb5b-b809-4061-8c45-7469b648329d1"
  }
}
```

示例3 停止云转推

停止云转推包括停止输入源、输出源转推。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 Stop

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "Stop",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "7819bb5b-b809-4061-8c45-7469b64e8370"
  }
}
```

示例4 切换输入源

云转推主备输入源切换，当前转推输入源为Main，切换到Backup输入源。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 SwitchInput
InputEndPoint	是	String	Main 或 Backup

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "SwitchInput",
  "InputEndpoint": "Backup",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "9303bb5b-b809-4061-8c45-7469b64e7630"
  }
}
```

示例5 删除输入源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DeleteInput
InputEndpoint	是	String	Main 或 Backup

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "DeleteInput",
  "InputEndpoint": "Main",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "a48cbb3b-b901-4052-8c48-9869b64e8e69"
  }
}
```

示例6 删除输出源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DeleteOutput
OutputInfo	是	StreamConnectOutput	输出源，Id 字段必填，从获取项目列表接口中拿到对应输出源的 Id

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "DeleteOutput",
  "OutputInfo": {
    "Id": "12357877"
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "kd83bb5b-b809-4061-8c45-7469b64e9183"
  }
}
```

示例7 开启云转推

开启云转推包括开启输入源、输出源转推。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 Start

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "Start",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
```

```
"RequestId": "190dbb5b-b809-4061-8c45-7469b64e928d"
}
```

示例8 添加输出源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddOutput
OutputInfo	是	StreamConnectOutput	输出源，Id 字段若不填则由后端生成

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "AddOutput",
  "OutputInfo": {
    "PushUrl": "rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600e8e66194ef500012d9b08xx",
    "Name": "媒体"
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "1938bb5b-b809-4061-8c45-7469b64e8e893"
  }
}
```

示例9 修改转推结束时间

修改当前正在转推项目的转推预计结束时间。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyCurrentStopTime
CurrentStopTime	是	String	当前转推预计结束时间，ISO 日期格式，不得超过当前时间30天

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyCurrentStopTime",
  "CurrentStopTime": "2021-03-23T07:51:18.029Z",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "kkd89b5b-b809-4061-8c45-7469b64e9263"
  }
}
```

示例10 添加直播拉流输入源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddInput
InputEndpoint	是	String	Main 或 Backup
InputInfo	是	StreamInputInfo	输入源，InputType 取值 LivePull

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "AddInput",
  "InputEndpoint": "Main",
  "InputInfo": {
    "LivePullInputInfo": {
      "InputUrl": "rtmp://liveplay.video-studio.myqcloud.com/output/1250000001-600e8e7fb1cc1c0001293759"
    },
    "InputType": "LivePull"
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例11 添加直播推流输入源

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddInput
InputEndpoint	是	String	Main 或 Backup
InputInfo	是	StreamInputInfo	输入源，InputType 取值 RtmpPush ， 从获取项目列表接口可拿到对应推流输入源的 PushUrl

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "AddInput",
  "InputEndpoint": "Main",
  "InputInfo": {
    "InputType": "RtmpPush",
    "RtmpPushInputInfo": {
      "ExpiredSecond": "3600"
    }
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "rtmp://liveplay-xx.video-studio.myqcloud.com/output/1250000001-6086674e265b4500018xxx?txSecret=*****&txTime=608FA1CE",
    "VodPullInputPlayInfo": null,
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例12 查询点播输入源播放进度

查询当前正在转推项目的点播输入源播放进度。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DescribeInputPlayInfo
InputEndpoint	是	String	Main 或 Backup

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "DescribeInputPlayInfo",
  "InputEndpoint": "Main ",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": {
      "Url": "https://1810000001.vod2.myqcloud.com/b64e98advodcq1810000001/85f47b5352858907805464143/AfJuJAzie5QA.mp4",
      "TimeOffset": 15
    },
    "RequestId": "kkd89b5b-b809-4061-8c45-7469b64e9263"
  }
}
```

示例13 添加点播拉流输入源且单次播放

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddInput
InputEndpoint	是	String	Main 或 Backup
InputInfo	是	StreamInputInfo	输入源，InputType 取值 VodPull ， VodPullInputInfo.LoopTimes 取值 0

输入示例

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

{
```

```
"Platform": "1000000009",
"Operation": "AddInput",
"InputEndpoint": "Main",
"InputInfo": {
  "VodPullInputInfo": {
    "LoopTimes": "0",
    "InputUrls": [
      "https://1810000001.vod2.myqcloud.com/b64e98advodcq1810000001/85f47b535285890787805464143/AfJuJAzie5QA.mp4"
    ]
  },
  "InputType": "VodPull"
},
"ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "c93cbb5b-b809-4061-8c45-7469b64e8e69"
  }
}
```

示例14 添加点播拉流输入源且循环播放

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddInput
InputEndpoint	是	String	Main 或 Backup
InputInfo	是	StreamInputInfo	输入源，InputType 取值 VodPull ， VodPullInputInfo.LoopTimes 取值-1

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "AddInput",
  "InputEndpoint": "Main",
  "InputInfo": {
    "VodPullInputInfo": {
      "LoopTimes": "-1",
      "InputUrls": [
        "https://1810000001.vod2.myqcloud.com/b64e98advodcq1810000001/85f47b535285890787805464143/AfJuJAzie5QA.mp4"
      ]
    },
    "InputType": "VodPull"
  }
}
```

```
},
"ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "StreamInputRtmpPushUrl": "",
    "VodPullInputPlayInfo": null,
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e85a"
  }
}
```

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
FailedOperation	操作失败。
FailedOperation.StartStreamConnectPartlyErr	转推部分失败。
FailedOperation.StreamConnect	转推操作失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Category	项目类别不合法。
InvalidParameterValue.ExtInfoInvalid	拓展信息不合法。

错误码	描述
InvalidParameterValue.Input	转推输入流参数非法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
InvalidParameterValue.StreamConnect	云转推信息非法。
InvalidParameterValue.StreamConnectInputInvalid	转推输入源信息非法。
InvalidParameterValue.StreamConnectOutputInvalid	转推输出源信息非法。
InvalidParameterValue.StreamInput	输入流参数错误。
LimitExceeded.BillItemLiveDispatchDuration	您已超过推流时长限制，若还需使用，请升级会员版本或购买资源包。
LimitExceeded.BillItemLiveDispatchMaxCount	您已达到转推并发数上限，若还需使用，请升级会员版本或购买资源包。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。

复制项目

最近更新时间：2025-04-25 01:19:24

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

复制一个项目，包括项目素材及轨道数据。目前仅普通剪辑及模板制作项目可复制，其它类型的项目不支持复制。

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：CopyProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
ProjectId	是	String	被复制的项目 ID。 示例值：cmepid_88kkdkkkkd009d9d9d9d9
Name	否	String	复制后的项目名称，不填为原项目名称+"(副本)"。 示例值：复制项目
Owner	否	Entity	复制后的项目归属者，不填为原项目归属者。
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验操作权限。 示例值：user_id_1000001

3. 输出参数

参数名称	类型	描述
ProjectId	String	复制后的项目 ID。 示例值：mepid_88kkdkkkkd009d9duujde77
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 复制一个项目

复制一个项目

输入示例

```
https://cme.tencentcloudapi.com/?Action=CopyProject
&Platform=1000000009
&ProjectId=cmepid_88kkdkkkkd009d9d9d9d9
```

&Name=复制项目

&<公共请求参数>

输出示例

```
{
  "Response": {
    "ProjectId": "mepid_88kkdkkkkd009d9duujde77",
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

操作媒体转推项目

最近更新时间：2025-05-27 01:20:42

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

对媒体转推项目进行操作。

操作类型

- `AddSource`（添加输入源），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)；
- `DeleteSource`（删除输入源），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)；
- `SwitchSource`（切换当前播放的输入源），项目状态为 `Working` 时可以操作。参见 [示例](#)；
- `AddDestination`（添加输出源），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)；
- `DeleteDestination`（删除输出源），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)；
- `EnableDestination`（启动输出源），项目状态为 `Working` 时可以操作。参见 [示例](#)；
- `DisableDestination`（停止输出源），项目状态为 `Working` 时可以操作。参见 [示例](#)；
- `ModifyDestination`（修改输出源），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)；
- `Start`（启动媒体转推），项目状态为 `Idle` 时可以操作。参见 [示例](#)。发起 `Start` 请求成功后，媒体转推项目开始启动，30秒内还需要再进行一次 `Confirm`操作进行确认；
- `Confirm`（确认媒体转推项目启动），项目状态为 `Working` 时可以操作。参见 [示例](#)；
- `Stop`（停止媒体转推），项目状态为 `Working` 时可以操作。参见 [示例](#)；
- `ModifyOutputMediaSetting`（修改媒体输出配置），项目状态为 `Idle` 时可以操作。参见 [示例](#)；
- `ModifyPlaySetting`（修改播放结束时间），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)；
- `DescribePlayInfo`（查询播放信息），项目状态为 `Idle`、`Working` 时均可以操作。参见 [示例](#)。

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：HandleMediaCastProject。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectId	是	String	媒体转推项目 Id。 示例值：67e105de4fb8950001052fff
Operation	是	String	请参考 操作类型 。 示例值：AddSource
SourceInfos.N	否	Array of MediaCastSourceInfo	输入源信息。具体操作方式详见 操作类型 及下文示例。 当 Operation 为 AddSource、DeleteSource、SwitchSource 时必填。
DestinationInfos.N	否	Array of MediaCastDestinationInfo	输出源信息。具体操作方式详见 操作类型 及下文示例。 当 Operation 为 AddDestination、DeleteDestination、EnableDestination、DisableDestination、ModifyDestination 时必填。
OutputMediaSetting	否	MediaCastOutputMediaSetting	输出媒体配置。具体操作方式详见 操作类型 及下文示例。 当 Operation 为 ModifyOutputSetting 时必填。

参数名称	必选	类型	描述
PlaySetting	否	MediaCastPlaySetting	播放控制参数。具体操作方式详见 操作类型 及下文示例。 当 Operation 为 ModifyPlaySetting 时必填。
Position	否	Integer	新添加的输入源位于输入源列表的位置，从0开始。默认加在输入源列表的后面。具体操作方式详见 操作类型 及下文示例。 当 Operation 为 AddSource 时必填。 示例值：1
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以操作所有媒体转推项目。如果指定操作者，则操作者必须为项目所有者。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
PlayInfo	MediaCastPlayInfo	播放信息，Operation 为 DescribePlayInfo 时返回。 注意：此字段可能返回 null，表示取不到有效值。
SourceInfoSet	Array of MediaCastSourceInfo	输入源信息，Operation 为 AddSource 时返回添加成功的输入源信息。 注意：此字段可能返回 null，表示取不到有效值。
DestinationInfoSet	Array of MediaCastDestinationInfo	输出源信息，Operation 为 AddDestination 时返回添加成功的输出源信息。 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 修改输出源

修改输出源的名称或者输出地址，如果项目状态是 Working 状态并且修改了输出地址，将断开原来输出源地址的直播流，向新的输出源地址输出直播流。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyDestination。
DestinationInfos	是	Array of MediaCastDestinationInfo	输出源信息。一次修改一个输出源。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyDestination",
  "DestinationInfos": [
    {
      "Id": "dt_01",
      "Name": "new name",
      "PushUrl": "rtmp://test.com/new?t=abc"
    }
  ],
}
```

```
"ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例2 停止输出源

项目状态是 Working 状态时，停止输出源将断开该输出源的直播流。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DisableDestination。
DestinationInfos	是	Array of MediaCastDestinationInfo	输出源信息。只需要带上输出源的 Id 参数。一次只能停止一个输出源。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "DisableDestination",
  "DestinationInfos": [
    {
      "Id": "dt_01"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例3 删除输入源

如果项目状态是 Working 状态，不能删除正在播放的输入源。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DeleteSource。
SourceInfos	是	Array of MediaCastSourceInfo	输入源信息，只需要带上输入源的 Id 。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "DeleteSource",
  "SourceInfos": [
    {
      "Id": "st_02"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例4 删除输出源

项目状态是 Working 状态时，删除输出源将停止该输出源的直播输出。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DeleteDestination。
DestinationInfos	是	Array of MediaCastDestinationInfo	输出源信息，只需要带上输出源的 Id 参数。一次只能删除一个输出源。

输入示例

```
POST / HTTP/1.1
Host: cme.tencentcloudapi.com
Content-Type: application/json
```

X-TC-Action: HandleMediaCastProject

<公共请求参数>

```
{
  "Platform": "1000000009",
  "Operation": "DeleteDestination",
  "DestinationInfos": [
    {
      "Id": "dt_02"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例5 启动输出源

重新启动已经停止的输出源，项目状态是 Working 状态时进行操作。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 EnableDestination。
DestinationInfos	是	Array of MediaCastDestinationInfo	输出源信息。只需要带上输出源的 Id 参数。一次只能启动一个输出源。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "EnableDestination",
  "DestinationInfos": [
    {
      "Id": "dt_02"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例6 添加输入源

新添加的输入源添加到输入源列表的后面。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddSource。
Position	否	Integer	输入源列表中的位置，0表示第一个位置。默认添加到输入源列表的后面。
SourceInfos	是	Array of MediaCastSourceInfo	输入源信息。输入源个数最大为100个。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "AddSource",
  "Position": 0,
  "SourceInfos": [
    {
      "Type": "CME",
      "MaterialId": "5fd8ad3d628dc30001bd0895"
    },
    {
      "Type": "VOD",
      "FileId": "5285485487985271487"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [
      {
        "Id": "st_01",
        "Type": "CME",
```

```
"MaterialId": "7d7ac177781a48c14198",
"FileId": "a82b830e765aa979",
"Offset": 0,
"Duration": 23,
"Url": "rtmp://push.example.com/cme/live1"
},
{
  "Id": "st_02",
  "Type": "VOD",
  "MaterialId": "7d7ac177781a48c12223",
  "FileId": "5285485487985271487",
  "Offset": 0,
  "Duration": 23,
  "Url": "rtmp://push.example.com/cme/live1"
}
],
"DestinationInfoSet": [],
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
}
}
```

示例7 添加输出源

项目状态为 Working 状态时，添加输出源后，将向新的输出源推送直播流。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 AddDestination。
DestinationInfos	是	Array of MediaCastDestinationInfo	输出源信息。一次添加一个输出源。输出源个数最大为10。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "AddDestination",
  "DestinationInfos": [
    {
      "Name": "demo",
      "PushUrl": "rtmp://test.com/live/abc?t=xxx"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
```

```
"PlayInfo": null,
"SourceInfoSet": [],
"DestinationInfoSet": [
{
  "Id": "dt_02",
  "PushUrl": "rtmp://test.com/live/aa?t=xx",
  "Name": "demo"
}
],
"RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
}
```

示例8 修改结束时间

如果项目是 Working 状态，将按照修改后的结束时间结束项目。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyPlaySetting。
PlaySetting	是	MediaCastPlaySetting	播放参数。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyPlaySetting",
  "PlaySetting": {
    "EndTime": "2022-12-20T20:00:00Z"
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例9 停止媒体转推

停止媒体转推。停止操作将停止输入源的播放并断开直播流，项目状态变为 Idle 状态。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 Stop。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "Stop",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例10 启动媒体转推

启动媒体转推。启动成功后项目状态变为 Working 状态，从输入源列表的第一个输入源开始播放，向输出源列表推送直播流。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 Start。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "Start",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例11 设置定时转推

项目需要是 Idle 状态，设置自动转推时间后在指定时间内启动项目进行转推。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyPlaySetting。
PlaySetting	是	MediaCastPlaySetting	播放参数。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyPlaySetting",
  "PlaySetting": {
    "AutoStartTime": "2022-12-20T20:00:00Z"
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例12 修改输出的媒体配置

修改输出的媒体配置，项目状态是 Working 状态时不能修改输出的媒体配置。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 ModifyOutputMediaSetting。
OutputMediaSetting	是	MediaCastOutputMediaSetting	输出的媒体配置。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "ModifyOutputMediaSetting",
  "OutputMediaSetting": {
    "VideoSetting": {
      "Width": 1920,
      "Height": 1080,
      "Bitrate": 2500,
      "FrameRate": 30
    }
  },
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例13 切换当前播放的输入源

项目状态为 Working 状态时切换生效，切换后从该输入源开始播放。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 SwitchSource。
SourceInfos	是	Array of MediaCastSourceInfo	输入源信息，只需要带上输入源的 Id 。

输入示例

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

```
{
  "Platform": "1000000009",
  "Operation": "SwitchSource",
  "SourceInfos": [
    {
      "Id": "st_02"
    }
  ],
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

示例14 确认媒体转推项目启动

发起 Start 操作成功后，媒体转推项目开始启动，在30秒内还需要发起 Confirm 操作进行确认。如果超时没有确认，该项目会被自动停止掉。

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 Confirm。

输入示例

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

{
  "Platform": "1000000009",
  "Operation": "Confirm",
  "ProjectId": "12522d74de35ff"
}
```

输出示例

```
{
  "Response": {
    "PlayInfo": null,
    "SourceInfoSet": [],
    "DestinationInfoSet": [],
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

```
}  
}
```

示例15 查询媒体转推项目的播放信息

输入参数：

除公共参数及 Platform、ProjectId 之外，其余参数填写规范如下：

参数名称	必选	类型	描述
Operation	是	String	请填写 DescribePlayInfo。

输入示例

```
POST / HTTP/1.1  
Host: cme.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: HandleMediaCastProject  
<公共请求参数>  
  
{  
  "Platform": "1000000009",  
  "Operation": "DescribePlayInfo",  
  "ProjectId": "12522d74de35ff"  
}
```

输出示例

```
{  
  "Response": {  
    "PlayInfo": {  
      "Status": "Working",  
      "CurrentSourceId": "st_0123",  
      "CurrentSourcePosition": 100,  
      "CurrentSourceDuration": 3490,  
      "LoopCount": 1,  
      "DestinationStatusSet": [  
        {  
          "Id": "dt_123",  
          "PushUrl": "rtmp://test.com/live/aa?t=xx",  
          "Status": "Working"  
        }  
      ],  
      "SourceInfoSet": [],  
      "DestinationInfoSet": [],  
      "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"  
    }  
  }  
}
```

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
FailedOperation	操作失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.EndTime	结束时间参数不合法。
InvalidParameterValue.MediaCastSourceInfosInvalid	输入源信息非法。
InvalidParameterValue.OperationInvalid	转推操作非法。
InvalidParameterValue.Operator	无效的操作者。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。

Headless媒体合成相关接口

使用视频合成协议导出视频

最近更新时间：2025-04-25 01:19:30

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

使用 [视频合成协议](#) 合成视频，支持导出视频到 CME 云媒资或者云点播媒资。

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：ExportVideoByEditorTrackData。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Definition	是	Integer	导出视频预设配置 Id，推荐优先使用下面的默认预设配置 Id，有其他需求可通过接口定制预设配置。 10：分辨率为 480P，输出视频格式为 MP4； 11：分辨率为 720P，输出视频格式为 MP4； 12：分辨率为 1080P，输出视频格式为 MP4。 示例值：10
ExportDestination	是	String	导出目标，指定导出视频的目标媒资库，可取值有： - CME：多媒体创建引擎，即导出到多媒体创作引擎媒资库，此导出目标在云点播媒资库依然可见； - VOD：云点播，即导出为云点播媒资库，此导出目标在多媒体创作引擎媒资库将不可见。 示例值：VOD
TrackData	是	String	轨道数据，用于描述待导出视频的内容。关于轨道数据的格式请查看 视频合成协议 。文档中也描述了如何在页面上查看一个剪辑项目的轨道数据，该能力可以帮助开发者更方便地构造自己的轨道数据。 示例值：[{"id":"469f6d6e","type":"video","items":[]}]
AspectRatio	否	String	轨道数据对应的画布宽高比，配合预设配置中的视频短边尺寸，可决定导出画面的尺寸。 例： - 如果 AspectRatio 取值 16:9，预设配置选为12（短边1080），则导出尺寸为 1920 * 1080； - 如果 AspectRatio 取值 9:16，预设配置选为11（短边720），则导出尺寸为 720 * 1280。 示例值：16:9
CoverData	否	String	视频封面图片文件（如 jpeg、png 等）进行 Base64 编码后的字符串，仅支持 gif、jpeg、png 三种图片格式，原图片文件不能超过2 M大小。 示例值：SGVsbG8sIFdvcmxklQ==

参数名称	必选	类型	描述
CMEExportInfo	否	CMEExportInfo	导出的多媒体创作引擎媒体信息。当导出目标为 CME 时必填。
VODExportInfo	否	VODExportInfo	导出的云点播媒资信息。当导出目标为 VOD 时必填。
ExportExtensionArgs	否	VideoExportExtensionArgs	视频导出扩展参数。可以覆盖导出模板中的参数，灵活的指定导出规格及参数。
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，无权限限制。如果指定操作者，轨道数据中使用的媒资该操作者需要拥有使用权限。 示例值：9b92a8e7-a5ef-4e28-d72e-4eff571fabf0

3. 输出参数

参数名称	类型	描述
TaskId	String	任务 Id。 示例值：125009388839-tfusion-0a54392053f84053942f9308c49404d7
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 使用视频编辑轨道数据导出视频到云点播

使用视频编辑轨道数据导出视频到云点播，并指定操作者。

输入示例

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

{
  "Platform": "1000000009",
  "Definition": 10,
  "ExportDestination": "VOD",
  "TrackData": "{\n\"id\": \"baa03b1c-6773-4459-a7b3-7f1e8b9e5b56\", \"type\": \"video\", \"items\": [{\n\"id\": \"f244311c-ca04-447a-b49a-1226356d46ba\", \"start_time\": 0, \"duration\": 3000, \"type\": \"image\", \"asset_id\": \"6503fd9d17509729a6100011\", \"filter_asset_id\": \"\", \"operations\": [{\n\"type\": \"image_rotate\", \"params\": {\n\"angle\": 0\n}], \"width\": 540, \"height\": 540, \"position\": {\n\"x\": 480, \"y\": 300\n}, \"section\": {\n\"from\": 0, \"to\": 3000\n}}]\",
  \"VODExportInfo\": {
    \"Name\": \"在线编辑视频\",
    \"ClassId\": 10
  },
  \"ExportExtensionArgs\": {
    \"FrameRate\": 30
  },
  \"Operator\": \"9b92a8e7-a5ef-4e28-d72e-4eff571fabf0\"
}
```

输出示例

```
{
  \"Response\": {
    \"TaskId\": \"1250093889-tfusion-0a54392053f84053942f9308c49404d7\",
    \"RequestId\": \"0a543920-53f8-4053-942f-9308c49404d7\"
  }
```

```
}  
}
```

示例2 使用视频编辑轨道数据导出视频到云点播并发布到企鹅号

使用视频编辑轨道数据导出视频到云点播，并将导出的视频发布到企鹅号。

输入示例

```
POST / HTTP/1.1  
Host: cme.tencentcloudapi.com  
Content-Type: application/json  
X-TC-Action: ExportVideoByEditorTrackData  
<公共请求参数>  
  
{  
  "Platform": "1000000009",  
  "Definition": 10,  
  "ExportDestination": "VOD",  
  "TrackData": "{\n\"id\": \"baa03b1c-6773-4459-a7b3-7f1e8b9e5b56\", \"type\": \"video\", \"items\": [{\n\"id\": \"f244311c-ca04-447a-b49a-1226356d46ba\", \"start_time\": 0, \"duration\": 3000, \"type\": \"image\", \"asset_id\": \"6503fd9d17509729a6100011\", \"filter_asset_id\": \"\", \"operations\": [{\n\"type\": \"image_rotate\", \"params\": {\n\"angle\": 0}], \"width\": 540, \"height\": 540, \"position\": {\n\"x\": 480, \"y\": 300}, \"section\": {\n\"from\": 0, \"to\": 3000}}]}",  
  "AspectRatio": "16:9",  
  "VODExportInfo": {  
    "Name": "在线编辑视频",  
    "ClassId": 10,  
    "ThirdPartyPublishInfos": [  
      {  
        "ChannelMaterialId": "5fd8ad3d628dc30001bd0895",  
        "PenguinMediaPlatformPublishInfo": {  
          "Title": "企鹅视频",  
          "Description": "新闻",  
          "Tags": [  
            "娱乐"  
          ],  
          "Category": 100  
        }  
      }  
    ],  
    "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"  
  }  
}
```

输出示例

```
{  
  "Response": {  
    "TaskId": "e86c730b-97c8-4e7d-b4a1-f5b28291e2b3",  
    "RequestId": "181xxxxxx01-FusionCME-c85bf7f3b9b4c594fbd558800a6fecb7t"  
  }  
}
```

示例3 使用视频编辑轨道数据导出视频到多媒体创作引擎媒资库

使用视频编辑轨道数据导出视频到多媒体创作引擎媒资库。

输入示例

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

{
  "Platform": "1000000009",
  "Definition": 1,
  "ExportDestination": "CME",
  "AspectRatio": "16:9",
  "TrackData": "{\n\"id\": \"baa03b1c-6773-4459-a7b3-7f1e8b9e5b56\", \"type\": \"video\", \"items\": [{\n\"id\": \"f244311c-ca04-447a-b49a-1226356d46ba\", \"start_time\": 0, \"duration\": 3000, \"type\": \"image\", \"asset_id\": \"6503fd9d17509729a6100011\", \"filter_asset_id\": \"\", \"operations\": [{\n\"type\": \"image_rotate\", \"params\": {\n\"angle\": 0}], \"width\": 540, \"height\": 540, \"position\": {\n\"x\": 480, \"y\": 300}, \"section\": {\n\"from\": 0, \"to\": 3000}}]}\n",
  "CMEExportInfo": {
    "Owner": {
      "Type": "PERSON",
      "Id": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
    },
    "Name": "新建项目",
    "ClassPath": "/成片"
  },
  "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e86c730b-97c8-4e7d-b4a1-f5b28291e2b3",
    "TaskId": "181xxxxxx01-FusionCME-c85bf7f3b9b4c594fbd558800a6fecb5t"
  }
}
```

示例4 使用视频编辑轨道数据导出视频到云点播并指定导出视频宽高比

使用视频编辑轨道数据导出视频到云点播，同时指定导出视频宽高比且指定操作者。

输入示例

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

{
  "Platform": "1000000009",
  "Definition": 10,
  "ExportDestination": "VOD",
  "AspectRatio": "16:9",
  "TrackData": "{\n\"id\": \"baa03b1c-6773-4459-a7b3-7f1e8b9e5b56\", \"type\": \"video\", \"items\": [{\n\"id\": \"f244311c-ca04-447a-b49a-1226356d46ba\", \"start_time\": 0, \"duration\": 3000, \"type\": \"image\", \"asset_id\": \"6503fd9d17509729a6100011\", \"filter_asset_id\": \"\", \"operations\": [{\n\"type\": \"image_rotate\", \"params\": {\n\"angle\": 0}], \"width\": 540, \"height\": 540, \"po
```

```
sition\":{"x\:480,"y\:300},"section\":{"from\:0,"to\:3000}}}],
"VOExportInfo": {
  "Name": "在线编辑视频",
  "ClassId": 10
},
"Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "181xxxxxx01-FusionCME-c85bf7f3b9b4c594fbd558800a6fecb5t",
    "RequestId": "e86c730b-97c8-4e7d-b4a1-f5b28291e2b3"
  }
}
```

示例5 使用视频编辑轨道数据导出视频到多媒体创作引擎媒资库并发布到企鹅号

使用视频编辑轨道数据导出视频到多媒体创作引擎媒资库并发布到企鹅号。

输入示例

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

{
  "Platform": "1000000009",
  "Definition": 1,
  "ExportDestination": "CME",
  "AspectRatio": "16:9",
  "TrackData": [{"id\:\"baa03b1c-6773-4459-a7b3-7f1e8b9e5b56\", \"type\":"video\", \"items\":[{ \"id\":"f244311c-ca04-447a-b49a-1226356d46ba\", \"start_time\:0, \"duration\:3000, \"type\":\"image\", \"asset_id\":"6503fd9d17509729a6100011\", \"filter_asset_id\":"\", \"operations\":[{ \"type\":\"image_rotate\", \"params\":{"angle\:0}}], \"width\:540, \"height\:540, \"position\":{"x\:480,\"y\:300},"section\":{"from\:0,\"to\:3000}}}],
  "CMEExportInfo": {
    "Owner": {
      "Type": "PERSON",
      "Id": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
    },
    "Name": "新建项目",
    "ClassPath": "/成片",
    "ThirdPartyPublishInfos": [
      {
        "ChannelMaterialId": "5fd8ad3d628dc30001bd0895",
        "PenguinMediaPlatformPublishInfo": {
          "Title": "企鹅视频",
          "Description": "新闻",
          "Tags": [
            "娱乐"
          ],
          "Category": 100
        }
      }
    ]
  }
}
```

```

    ],
    },
    "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
  }
}

```

输出示例

```

{
  "Response": {
    "RequestId": "e86c730b-97c8-4e7d-b4a1-f5b28291e2b3",
    "TaskId": "181xxxxxx01-FusionCME-c82bf7f3b9b4c594fbd558800a6fe7c5t"
  }
}

```

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 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
FailedOperation	操作失败。
InternalServerError	内部错误。
InternalServerError.CreateTask	发起预处理任务失败。
InvalidParameter	参数错误。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Definition	模板 ID 无效。
InvalidParameterValue.ExportDestination	导出目标不合法。
InvalidParameterValue.MaterialId	媒体 ID 无效。

错误码	描述
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ThirdyPartyPublishChannelId	发布通道 ID 无效。
InvalidParameterValue.TrackData	轨道数据无效。
InvalidParameterValue.VodSubAppid	绑定的点播子应用无效。
LimitExceeded.BillItemStorage	存储空间不足，请查看会员权益，购买资源包或会员版本后继续使用。
LimitExceeded.BillItemVideoEditExportDuration	您已达到导出视频时长上限，若还需使用，请升级会员版本或购买资源包。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

使用视频剪辑模板导出视频

最近更新时间：2025-04-25 01:19:29

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

使用视频剪辑模板直接导出视频。

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：ExportVideoByTemplate。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TemplateId	是	String	视频编辑模板 Id。 示例值：60097a8a8ea83d00017b484b
Definition	是	Integer	导出视频预设配置 Id，推荐优先使用下面的默认预设配置 Id，有其他需求可通过接口定制预设配置。 10：分辨率为 480P，输出视频格式为 MP4； 11：分辨率为 720P，输出视频格式为 MP4； 12：分辨率为 1080P，输出视频格式为 MP4。 示例值：10
ExportDestination	是	String	导出目标，指定导出视频的目标媒资库，可取值有： - CME：多媒体创作引擎，即导出为多媒体创作引擎媒资库，此导出目标在云点播媒资库依然可见； - VOD：云点播，即导出为云点播媒资库，此导出目标在多媒体创作引擎媒资库将不可见。 示例值：VOD
SlotReplacements.N	否	Array of SlotReplacementInfo	需要替换的素材信息。
CMEExportInfo	否	CMEExportInfo	导出的多媒体创作引擎媒资信息。当导出目标为 CME 时必填。
VODExportInfo	否	VODExportInfo	导出的云点播媒资信息。当导出目标为 VOD 时必填。
ExportExtensionArgs	否	VideoExportExtensionArgs	视频导出扩展参数。可以覆盖导出模板中的参数，灵活的指定导出规格及参数。
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，无权限限制。如果指定操作者，则操作者需要有替换媒体及剪辑模板的权限。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
TaskId	String	导出任务 Id。 示例值：1250000000-tfusion-32f87637061242508eb10d6e4ecd0cd6
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 使用剪辑模板导出视频到云点播

输入示例

```
https://cme.tencentcloudapi.com/?Action=ExportVideoByTemplate
&Platform=1000000009
&TemplateId=60498a50adaf8f000164a07b@prod@125000000
&SlotReplacements.0.Id=1
&SlotReplacements.0.ReplacementType=VIDEO
&SlotReplacements.0.MediaReplacementInfo.MediaType=CMEMaterialId
&SlotReplacements.0.MediaReplacementInfo.MaterialId=5fc4ad7f69ec74000179072345
&SlotReplacements.0.MediaReplacementInfo.StartTimeOffset=0
&Definition=10
&ExportDestination=VOD
&VODExportInfo.Name=测试视频
&VODExportInfo.ClassId=10000
&Operator=user_id_9394392053f8405394299d8
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TaskId": "1250000000-tfusion-32f87637061242508eb10d6e4ecd0cd6",
    "RequestId": "0b480791-97d7-4e4a-8740-e2128b871df6"
  }
}
```

示例2 使用剪辑模板导出视频到多媒体创作引擎媒资

输入示例

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

{
  "Platform": "1000000009",
  "TemplateId": "60498a50adaf8f000164a07b@prod@125000000",
  "SlotReplacements": [
    {
      "Id": 1,
      "ReplacementType": "IMAGE",
      "MediaReplacementInfo": {
        "MediaType": "CMEMaterialId",
```

```
"MaterialId": "becff2c5e7d77a9e0353"
}
}
],
"Definition": 10,
"ExportDestination": "CME",
"CMEExportInfo": {
  "Owner": {
    "Type": "PERSON",
    "Id": "1a69f09d-1a28-4437-a71a"
  },
  "Name": "模板测试1009",
  "ClassPath": "/媒资"
},
"ExportExtensionArgs": {
  "VideoBitrate": 102400,
  "FrameRate": 30
},
"Operator": "user_id_9394392053f8405394299d8"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "1250000000-tfusion-32f87637061242508eb10d6e4ecd0cd7",
    "RequestId": "323d0791-97d7-4e4a-8740-e2128b031df7"
  }
}
```

示例3 使用剪辑模板导出视频到云点播并发布到企鹅号

输入示例

```
https://cme.tencentcloudapi.com/?Action=ExportVideoByTemplate
&Platform=1000000009
&TemplateId=60498a50adaf8f000164a07b@prod@12500000
&SlotReplacements.0.Id=1
&SlotReplacements.0.ReplacementType=VIDEO
&SlotReplacements.0.MediaReplacementInfo.MediaType=CMEMaterialId
&SlotReplacements.0.MediaReplacementInfo.MaterialId=5fc4ad7f69ec74000179072345
&SlotReplacements.0.MediaReplacementInfo.StartTimeOffset=0
&Definition=10
&ExportDestination=VOD
&VODExportInfo.Name=测试视频
&VODExportInfo.ClassId=10000
&VODExportInfo.ThirdPartyPublishInfos.0.ChannelMaterialId=5fd8ad3d628dc30001bd0895
&VODExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Title=企鹅视频
&VODExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Description=新闻
&VODExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Tags.0=娱乐
&VODExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Category=100
&Operator=user_id_9394392053f8405394299d8
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TaskId": "1250000000-tfusion-32f87637061242508eb10d6e4ecd0cd8",
    "RequestId": "323d0791-97d7-4e4a-8740-e2128b031df8"
  }
}
```

示例4 使用剪辑模板导出视频到多媒体创作引擎媒资并发布到企鹅号

输入示例

```
https://cme.tencentcloudapi.com/?Action=ExportVideoByTemplate
&Platform=1000000009
&TemplateId=60498a50adaf8f000164a07b@prod@12500000
&SlotReplacements.0.Id=1
&SlotReplacements.0.ReplacementType=VIDEO
&SlotReplacements.0.MediaReplacementInfo.MediaType=CMEMaterialId
&SlotReplacements.0.MediaReplacementInfo.MaterialId=5fc4ad7f69ec74000179072345
&SlotReplacements.0.MediaReplacementInfo.StartTimeOffset=0
&Definition=10
&ExportDestination=CME
&CMEExportInfo.Owner.Type=PERSON
&CMEExportInfo.Owner.Id=119993993939390
&CMEExportInfo.Name=测试视频
&CMEExportInfo.ClassPath=/成片
&CMEExportInfo.ThirdPartyPublishInfos.0.ChannelMaterialId=5fd8ad3d628dc30001bd0895
&CMEExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Title=企鹅视频
&CMEExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Description=新闻
&CMEExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Tags.0=娱乐
&CMEExportInfo.ThirdPartyPublishInfos.0.PenguinMediaPlatformPublishInfo.Category=100
&Operator=user_id_9394392053f8405394299d8
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TaskId": "1250000000-tfusion-32f87637061242508eb10d6e4ecd0cd8",
    "RequestId": "89d1e791-97d7-4e4a-8740-e2128b038ad9"
  }
}
```

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。
InternalError	内部错误。
InternalError.CreateTask	发起预处理任务失败。
InvalidParameter	参数错误。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.Definition	模板 ID 无效。
InvalidParameterValue.ExportDestination	导出目标不合法。
InvalidParameterValue.MaterialId	媒体 ID 无效。
InvalidParameterValue.MediaReplacementInfo	媒体替换信息无效。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ReplacementType	轨道素材替换类型无效或不匹配。
InvalidParameterValue.TrackData	轨道数据无效。
InvalidParameterValue.VideoEditTemplateIdNotExist	视频编辑模板不存在。
LimitExceeded	超过配额限制。
LimitExceeded.BillItemStorage	存储空间不足，请查看会员权益，购买资源包或会员版本后继续使用。
LimitExceeded.BillItemVideoEditExportDuration	您已达到导出视频时长上限，若还需使用，请升级会员版本或购买资源包。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。

任务管理相关接口

获取任务列表

最近更新时间：2025-04-25 01:19:29

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取任务列表，支持条件筛选，返回对应的任务基础信息列表。

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：DescribeTasks。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectId	否	String	项目 Id，使用项目 Id 进行过滤。 示例值：a82b830e765aa979
TaskTypeSet.N	否	Array of String	任务类型集合，取值有： VIDEO_EDIT_PROJECT_EXPORT：视频编辑项目导出。 注：不填不使用任务类型进行过滤。 示例值：["VIDEO_EDIT_PROJECT_EXPORT"]
StatusSet.N	否	Array of String	任务状态集合，取值有： - PROCESSING：处理中； - SUCCESS：成功； - FAIL：失败。 注：不填则不使用任务状态进行过滤。 示例值：["PROCESSING"]
Offset	否	Integer	分页返回的起始偏移量，默认值：0。 示例值：0
Limit	否	Integer	分页返回的记录条数，默认值：10。最大值：20。 示例值：10
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以获取所有任务信息。如果指定操作者，则操作者需要是任务发起者。 示例值：378b753f-f91f-4c42-8fa3-34a2a3bf3c56

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合搜索条件的记录总数。 示例值：1
TaskBaseInfoSet	Array of TaskBaseInfo	任务基础信息列表。
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，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 分页获取视频导出任务列表

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeTasks
&Platform=1000000009
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "TaskBaseInfoSet": [
      {
        "CreateTime": "2020-04-30T03:26:37Z",
        "ErrCode": 0,
        "ErrMsg": "SUCCESS",
        "Progress": 100,
        "Status": "SUCCESS",
        "TaskId": "181xxxxxxxx9-procedurev2-b718fcee09a72470bf665cf55351d810t0",
        "TaskType": "VIDEO_EDIT_PROJECT_EXPORT"
      }
    ],
    "RequestId": "efde1a47-f1c9-4c1b-911c-9865a6a4d1ac"
  }
}
```

示例2 指定项目 Id 获取视频导出任务列表

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeTasks
&Platform=1000000009
&Offset=0
&ProjectId=cmePid_5fd8ad3d628dc30001bd0895
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "TaskBaseInfoSet": [
      {
        "CreateTime": "2020-04-30T03:26:37Z",
        "ErrCode": 0,
        "ErrMsg": "SUCCESS",
        "Progress": 100,
        "Status": "SUCCESS",
        "TaskId": "181xxxxxxxx9-procedurev2-b718fcee09a72470bf665cf55351d810t0",
        "TaskType": "VIDEO_EDIT_PROJECT_EXPORT"
      }
    ],
    "RequestId": "efde1a47-f1c9-4c1b-911c-9865a6a4d1ad"
  }
}
```

示例3 指定任务状态获取视频导出任务列表

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeTasks
&Platform=1000000009
&StatusSet.0=SUCCESS
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "TaskBaseInfoSet": [
      {
        "CreateTime": "2020-04-30T03:26:37Z",
        "ErrCode": 0,
        "ErrMsg": "SUCCESS",
        "Progress": 100,
        "Status": "SUCCESS",
        "TaskId": "181xxxxxxxx9-procedurev2-b718fcee09a72470bf665cf55351d810t0",
        "TaskType": "VIDEO_EDIT_PROJECT_EXPORT"
      }
    ],
    "RequestId": "efde1a47-f1c9-4c1b-911c-9865a6a4d1ae"
  }
}
```

示例4 指定任务类型获取视频导出任务列表

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeTasks
&Platform=1000000009
```

```
&TaskTypeSet.0=VIDEO_EDIT_PROJECT_EXPORT
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "TaskBaseInfoSet": [
      {
        "CreateTime": "2020-04-30T03:26:37Z",
        "ErrCode": 0,
        "ErrMsg": "SUCCESS",
        "Progress": 100,
        "Status": "SUCCESS",
        "TaskId": "181xxxxxxxx9-procedurev2-b718fcee09a72470bf665cf55351d810t0",
        "TaskType": "VIDEO_EDIT_PROJECT_EXPORT"
      }
    ],
    "RequestId": "efde1a47-f1c9-4c1b-911c-9865a6a4d133"
  }
}
```

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。
InvalidParameterValue.Platform	平台参数取值错误。

错误码	描述
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取任务详情

最近更新时间：2025-04-25 01:19:29

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

获取任务详情信息，包含下面几个部分：

- 任务基础信息：包括任务状态、错误信息、创建时间等；
- 导出项目输出信息：包括输出的素材 Id 等。

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：DescribeTaskDetail。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TaskId	是	String	任务 Id。 示例值：1400-tfusion-ct0
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以获取任意任务信息。如果指定操作者，则操作者需要是任务发起者。 示例值：85b8cdac-8e18-430a-aeae-08076ac26a11

3. 输出参数

参数名称	类型	描述
Status	String	任务状态，取值有： - PROCESSING：处理中； - SUCCESS：成功； - FAIL：失败。 示例值：SUCCESS
Progress	Integer	任务进度，取值为：0~100。 示例值：100
ErrCode	Integer	错误码。 0：成功； - 其他值：失败。 示例值：0
ErrMsg	String	错误信息。 示例值：Success.
TaskType	String	任务类型，取值有： VIDEO_EDIT_PROJECT_EXPORT：视频编辑项目导出。 示例值：VIDEO_EDIT_PROJECT_EXPORT

参数名称	类型	描述
VideoEditProjectOutput	VideoEditProjectOutput	导出项目输出信息。仅当 TaskType 为 VIDEO_EDIT_PROJECT_EXPORT 时有效。 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	创建时间，格式按照 ISO 8601 标准表示。 示例值：2020-04-28T18:16:58Z
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 获取任务详情

获取任务详情。

输入示例

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

{
  "Platform": "1000000009",
  "TaskId": "181xxxxxxxx9-procedurev2-b718fcee09a72470bf665cf55351d810t0"
}
```

输出示例

```
{
  "Response": {
    "CreateTime": "2020-04-28T18:16:58Z",
    "ErrCode": 0,
    "ErrMsg": "Success.",
    "Progress": 100,
    "RequestId": "85b8cdac-8e18-430a-aeae-08076ac26a11",
    "Status": "SUCCESS",
    "TaskType": "VIDEO_EDIT_PROJECT_EXPORT",
    "VideoEditProjectOutput": {
      "MaterialId": "6866c9a7-a2ca-4d",
      "MetaData": {
        "AudioStreamInfoSet": [
          {
            "Bitrate": 46552,
            "Codec": "aac",
            "SamplingRate": 16000
          }
        ],
        "Bitrate": 6917072,
        "Container": "mov,mp4,m4a,3gp,3g2,mj2",
        "Duration": 346.55999755859375,
        "Height": 1080,
        "Rotate": 0,
        "Size": 299688882,
        "VideoStreamInfoSet": [
          {
```

```
"Bitrate": 6870520,
"Codec": "h264",
"Fps": 25,
"Height": 1080,
"Width": 1920
},
"Width": 1920
},
"URL": "http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.mp4",
"CoverURL": "http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.jpg",
"VodFileId": "5285890802029089"
}
}
}
```

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。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TaskId	任务 Id 无效。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

视频编辑项目相关接口

视频智能拆条

最近更新时间：2025-04-25 01:19:24

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

- 发起视频智能拆条任务，支持智能生成和平精英集锦、王者荣耀集锦、足球集锦、篮球集锦、人物集锦、新闻拆条等任务。
 - 和平精英集锦和王者荣耀集锦根据击杀场景进行拆条，足球集锦和篮球集锦根据进球场景进行拆条，人物集锦根据人物人脸特征进行拆条，新闻拆条根据导播进行拆条。
 - 【本接口内测中，暂不建议使用】
- 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：GenerateVideoSegmentationSchemeByAi。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ProjectId	是	String	视频拆条项目 Id。 示例值：60336ab29997a50001942xxx
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以对任务视频拆条项目发起拆条任务。如果指定操作者，则操作者必须为项目所有者。 示例值：cmeid_system

3. 输出参数

参数名称	类型	描述
TaskId	String	视频智能拆条任务 Id。 示例值：1400xxx8-procxxx
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 智能拆条

输入示例

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

```
{
  "Platform": "1000000009",
  "ProjectId": "67e105de4fb8950001052fff"
}
```

输出示例

```
{
  "Response": {
    "TaskId": "5f16967b64436100015fb025",
    "RequestId": "c44cbb5b-b809-4061-8c45-7469b64e8e5x"
  }
}
```

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。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.ProjectId	项目不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。

使用视频智能拆条数据导出视频

最近更新时间：2025-04-25 01:19:24

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

使用视频智能拆条数据导出视频，将指定的视频拆条片段导出为一个视频(内测中，请勿使用)。

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：ExportVideoByVideoSegmentationData。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：default_platform
ProjectId	是	String	视频拆条项目 Id。 示例值：cmepid_5fd1dfaf0e1b530001f5d9b0
SegmentGroupId	是	String	指定需要导出的智能拆条片段的组 Id。 示例值：5fd6e20b0e1b530001f5daed
SegmentIds.N	是	Array of String	指定需要导出的智能拆条片段 Id 集合。 示例值：["seg_5fd6e20b0e1b530001f5daee\n"]
Definition	是	Integer	导出模板 Id，目前不支持自定义创建，只支持下面的预置模板 Id。 10：分辨率为 480P，输出视频格式为 MP4； 11：分辨率为 720P，输出视频格式为 MP4； 12：分辨率为 1080P，输出视频格式为 MP4。 示例值：10
ExportDestination	是	String	导出目标，指定导出视频的目标媒资库，可取值有： - CME：多媒体创作引擎，即导出为多媒体创作引擎媒资库，此导出目标在云点播媒资库依然可见； - VOD：云点播，即导出为云点播媒资库，此导出目标在多媒体创作引擎媒资库将不可见。 示例值：VOD
CMEExportInfo	否	CMEExportInfo	导出的多媒体创作引擎媒体信息。当导出目标为 CME 时必填。
VODExportInfo	否	VODExportInfo	导出的云点播媒资信息。当导出目标为 VOD 时必填。
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以操作任意智能拆条项目。如果指定操作者，则操作者必须为项目所有。 示例值：16556266637489

3. 输出参数

参数名称	类型	描述
TaskId	String	任务 Id。

参数名称	类型	描述
		示例值：125009388839-tfusion-0a54392053f84053942f9308c49404d7
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 使用视频智能拆条数据导出视频到CME

输入示例

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

{
  "Definition": "10",
  "ProjectId": "cmepid_5fd1dfaf0e1b530001f5d9b0",
  "CMEExportInfo": {
    "Owner": {
      "Type": "PERSON",
      "Id": "08f06c7c1fdddbea083f899b932c4987"
    },
    "ClassPath": "/媒资/视频",
    "Name": "在线编辑视频"
  },
  "ExportDestination": "CME",
  "Platform": "default_platform",
  "Operator": "16556266637489",
  "SegmentGroupId": "5fd6e20b0e1b530001f5daed",
  "SegmentIds": [
    "seg_5fd6e2100e1b530001f5daef",
    "seg_5fd6e20b0e1b530001f5daee"
  ]
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125009388839-tfusion-0a54392053f84053942f9308c49404d7",
    "RequestId": "0a543920-53f8-4053-942f-9308c49404d7"
  }
}
```

示例2 使用视频智能拆条数据导出视频到VOD

输入示例

```
POST / HTTP/1.1
Host: cme.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ExportVideoByVideoSegmentationData
```

<公共请求参数>

```
{
  "Definition": "10",
  "ProjectId": "cmepid_5fd1dfaf0e1b530001f5d9b0",
  "ExportDestination": "VOD",
  "Platform": "default_platform",
  "VODExportInfo": {
    "Name": "在线编辑视频"
  },
  "Operator": "16556266637489",
  "SegmentGroupId": "5fd6e20b0e1b530001f5daed",
  "SegmentIds": [
    "seg_5fd6e20b0e1b530001f5daee"
  ]
}
```

输出示例

```
{
  "Response": {
    "TaskId": "125009388839-tfusion-0a54392053f84053942f9308c49404d7",
    "RequestId": "0a543920-53f8-4053-942f-9308c49404d7"
  }
}
```

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
InvalidParameter	参数错误。

错误码	描述
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Definition	模板 ID 无效。
InvalidParameterValue.ExportDestination	导出目标不合法。
InvalidParameterValue.ProjectId	项目不存在。
LimitExceeded.BillItemStorage	存储空间不足，请查看会员权益，购买资源包或会员版本后继续使用。
LimitExceeded.BillItemVideoEditExportDuration	您已达到导出视频时长上限，若还需使用，请升级会员版本或购买资源包。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。

媒体资源管理相关接口

平铺浏览媒体

最近更新时间：2025-06-04 01:21:15

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

平铺分类路径下及其子分类下的所有媒体基础信息，返回当前分类及子分类中的所有媒体的基础信息。

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：FlattenListMedia。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ClassPath	是	String	媒体分类路径，例如填写"/a/b"，则代表平铺该分类路径下及其子分类路径下的媒体信息。 示例值：/a/b
Owner	是	Entity	媒体分类的归属者。
Offset	否	Integer	分页偏移量，默认值：0。 示例值：0
Limit	否	Integer	返回记录条数，默认值：10，最大值：50。 示例值：10
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以平铺查询任意分类下的媒体信息。如果指定操作者，则操作者必须对当前分类有读权限。 示例值：cmeid_system

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的记录总数。 示例值：1
MaterialInfoSet	Array of MaterialInfo	该分类路径下及其子分类下的所有媒体基础信息列表。
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，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 平铺浏览媒体

输入示例

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

{
  "Owner": {
    "Type": "PERSON",
    "Id": "user_id_8ad3d628dc30001bd0343"
  },
  "Platform": "10000000009",
  "ClassPath": "/a/b"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "MaterialInfoSet": [
      {
        "BasicInfo": {
          "MaterialId": "281921553743710210",
          "MaterialType": "VIDEO",
          "Name": "9Qvacao2r7EA",
          "CreateTime": "2019-12-31T03:59:14Z",
          "UpdateTime": "2019-12-31T03:59:14Z",
          "ClassPath": "/a/b",
          "TagSet": [],
          "PresetTagSet": [],
          "PreviewUrl": "http://1810xxxxxxx2.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/coverBySnaps
hot/1577764755_4069421039.100_0.jpg",
          "Owner": {
            "Type": "PERSON",
            "Id": "user_id_8ad3d628dc30001bd0343"
          }
        },
        "ImageMaterial": null,
        "LinkMaterial": null,
        "AudioMaterial": null,
        "VideoMaterial": null,
        "VideoEditTemplateMaterial": null,
        "OtherMaterial": null
      }
    ],
    "RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a05"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

搜索媒体

最近更新时间：2025-04-25 01:19:26

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

根据检索条件搜索媒体，返回媒体的基本信息。

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：SearchMaterial。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
SearchScopes.N	是	Array of SearchScope	指定搜索空间，数组长度不得超过5。
MaterialTypes.N	否	Array of String	媒体类型，可取值有： - AUDIO：音频； - VIDEO：视频； - IMAGE：图片； - VIDEO_EDIT_TEMPLATE：剪辑模板。 示例值：["VIDEO"]
Text	否	String	搜索文本，模糊匹配媒体名称或描述信息，匹配项越多，匹配度越高，排序越优先。长度限制：15个字符。 示例值：综艺
Resolution	否	String	按画质检索，取值为：LD/SD/HD/FHD/2K/4K。 示例值：HD
DurationRange	否	IntegerRange	按媒体时长检索，单位s。
CreateTimeRange	否	TimeRange	按照媒体创建时间检索。
Tags.N	否	Array of String	按标签检索，填入检索的标签名。 示例值：["zoo"]
Sort	否	SortBy	排序方式。Sort.Field 可选值：CreateTime。指定 Text 搜索时，将根据匹配度排序，该字段无效。
Offset	否	Integer	偏移量。默认值：0。 示例值：0
Limit	否	Integer	返回记录条数，默认值：50。 示例值：10
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以搜索任意媒体的信息。如果指定操作者，则操作者必须对媒体有读权限。 示例值：user_id

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合记录总条数。 示例值：1
MaterialInfoSet	Array of MaterialInfo	媒体信息，仅返回基础信息。
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 根据关键字检索媒体

检索某搜索空间下包含关键字的媒体。

输入示例

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

{
  "Platform": "1000000009",
  "Text": "滨海",
  "SearchScopes": [
    {
      "Owner": {
        "Type": "PERSON",
        "Id": "3bfd339c-4e0d-4742-933e-eb71abde0965"
      },
      "ClassPath": "/a/b"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "MaterialInfoSet": [
      {
        "BasicInfo": {
          "MaterialId": "5feb08c4b4e0960001ed96a5",
          "MaterialType": "VIDEO",
          "Name": "腾讯滨海大厦",
          "CreateTime": "2019-12-31T03:59:14Z",
          "UpdateTime": "2019-12-31T03:59:14Z",
          "ClassPath": "/a/b",
          "TagSet": [],
          "TagInfoSet": [],
          "PresetTagSet": [],
          "PreviewUrl": "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/coverBySnapsho
```

```
t/1577764755_4069421039.100_0.jpg",
"Owner": {
  "Type": "PERSON",
  "Id": "3bfd339c-4e0d-4742-933e-eb71abde0965"
},
"ImageMaterial": null,
"LinkMaterial": null,
"AudioMaterial": null,
"VideoMaterial": null,
"VideoEditTemplateMaterial": null,
"OtherMaterial": null
},
"RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a05"
}
```

示例2 根据媒体类型检索媒体

检索某搜索空间下媒体类型为VIDEO的媒体。

输入示例

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

{
  "Platform": "1000000009",
  "MaterialTypes": [
    "VIDEO"
  ],
  "SearchScopes": [
    {
      "Owner": {
        "Type": "PERSON",
        "Id": "3bfd339c-4e0d-4742-933e-eb71abde0965"
      },
      "ClassPath": "/a/b"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "MaterialInfoSet": [
      {
        "BasicInfo": {
          "MaterialId": "5feb08c4b4e0960001ed96a4",
          "MaterialType": "VIDEO",
          "Name": "9Qvacao2r7EA",
```

```
"CreateTime": "2019-12-31T03:59:14Z",
"UpdateTime": "2019-12-31T03:59:14Z",
"ClassPath": "/a/b",
"TagSet": [],
"TagInfoSet": [],
"PresetTagSet": [],
"PreviewUrl": "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/coverBySnapshot/1577764755_4069421039.100_0.jpg",
"Owner": {
  "Type": "PERSON",
  "Id": "3bfd339c-4e0d-4742-933e-eb71abde0965"
},
"ImageMaterial": null,
"LinkMaterial": null,
"AudioMaterial": null,
"VideoMaterial": null,
"VideoEditTemplateMaterial": null,
"OtherMaterial": null
}
],
"RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a05"
}
}
```

示例3 根据搜索空间检索媒体

检索某搜索空间下的媒体。

输入示例

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

{
  "Platform": "1000000009",
  "MaterialTypes": [
    "VIDEO"
  ],
  "SearchScopes": [
    {
      "Owner": {
        "Type": "PERSON",
        "Id": "3bfd339c-4e0d-4742-933e-eb71abde0965"
      },
      "ClassPath": "/a/b"
    }
  ]
}
```

输出示例

```
{
  "Response": {
```

```
"TotalCount": 1,
"MaterialInfoSet": [
{
  "BasicInfo": {
    "MaterialId": "5feb08c4b4e0960001ed96a4",
    "MaterialType": "VIDEO",
    "Name": "9Qvacao2r7EA",
    "CreateTime": "2019-12-31T03:59:14Z",
    "UpdateTime": "2019-12-31T03:59:14Z",
    "ClassPath": "/a/b",
    "TagSet": [],
    "TagInfoSet": [],
    "PresetTagSet": [],
    "PreviewUrl": "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/coverBySnapshot/1577764755_4069421039.100_0.jpg",
    "Owner": {
      "Type": "PERSON",
      "Id": "3bfd339c-4e0d-4742-933e-eb71abde0965"
    }
  },
  "ImageMaterial": null,
  "LinkMaterial": null,
  "AudioMaterial": null,
  "VideoMaterial": null,
  "VideoEditTemplateMaterial": null,
  "OtherMaterial": null
}
],
"RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a05"
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.DataNotFoundInDB	查询不到数据。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

修改媒体信息

最近更新时间：2025-04-25 01:19:26

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

修改媒体信息，支持修改媒体名称、分类路径、标签等信息。

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：ModifyMaterial。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
MaterialId	是	String	要修改的媒体 Id。 示例值：67e0e02c1ac4270001501fff
Owner	否	Entity	媒体归属者。
Name	否	String	媒体名称，不能超过30个字符，不填则不修改。 示例值：name_modify
ClassPath	否	String	媒体分类路径，例如填写"/a/b"，则代表该媒体存储的路径为"/a/b"。若修改分类路径，则 Owner 字段必填。 示例值：/a/b
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以修改任意媒体的信息。如果指定操作者，则操作者必须对媒体有写权限。 示例值：user_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 修改媒体名称

输入示例

```
https://cme.tencentcloudapi.com/?Action=ModifyMaterial
&Platform=1000000009
&MaterialId=5fd8ad3d628dc30001bd0895
```

&Name=修改媒体名称
&<公共请求参数>

输出示例

```
{
  "Response": {
    "RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a05"
  }
}
```

示例2 修改媒体的分类

输入示例

```
https://cme.tencentcloudapi.com/?Action=ModifyMaterial
&Platform=1000000009
&MaterialId=5fd8ad3d628dc30001bd0895
&ClassPath=/媒体/视频
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a12"
  }
}
```

示例3 修改媒体归属为团队

输入示例

```
https://cme.tencentcloudapi.com/?Action=ModifyMaterial
&Platform=1000000009
&MaterialId=5fd8ad3d628dc30001bd0895
&Owner.Id=cmetid_acbde8ad3do0303dc1b33
&Owner.Type=TEAM
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a09"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

浏览媒体

最近更新时间：2025-06-04 01:21:14

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

浏览当前分类路径下的资源，包括媒体文件和子分类，返回媒资基础信息和分类信息。

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：ListMedia。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
ClassPath	是	String	媒体分类路径，例如填写"/a/b"，则代表浏览该分类路径下的媒体和子分类信息。 示例值：/a/b
Owner	是	Entity	媒体和分类的归属者。
Offset	否	Integer	分页偏移量，默认值：0。 示例值：0
Limit	否	Integer	返回记录条数，默认值：10，最大值：50。 示例值：10
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以浏览任意分类的信息。如果指定操作者，则操作者必须对分类有读权限。 示例值：user_id

3. 输出参数

参数名称	类型	描述
MaterialTotalCount	Integer	符合条件的媒体记录总数。 示例值：1
MaterialInfoSet	Array of MaterialInfo	浏览分类路径下的媒体列表信息。
ClassInfoSet	Array of ClassInfo	浏览分类路径下的一级子类。
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 浏览媒体

浏览媒体

输入示例

```
https://cme.tencentcloudapi.com/?Action=ListMedia
&Platform=1000000009
&ClassPath=/素材/视频
&Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
&Owner.Type=PERSON
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialInfoSet": [
      {
        "AudioMaterial": null,
        "BasicInfo": {
          "ClassPath": "/素材/视频",
          "CreateTime": "2021-11-24T16:08:07Z",
          "MaterialId": "619e6367b2e891000160aaac0",
          "MaterialType": "VIDEO",
          "Name": "a.mp4",
          "Owner": {
            "Id": "6b6ef043-85f3-4614-b735-768bb466ae5b",
            "Type": "PERSON"
          },
          "PresetTagSet": [],
          "PreviewUrl": "http://1xxxxx98.vod2.myqcloud.com/334751dbvodtranssh1400293698/e37cf1313877022919226xxxx/coverBySnapshot/coverBySnapshot_10_0.jpg",
          "TagSet": [],
          "UpdateTime": "2021-11-24T16:08:07Z"
        },
        "ImageMaterial": null,
        "LinkMaterial": null,
        "OtherMaterial": null,
        "VideoEditTemplateMaterial": null,
        "VideoMaterial": null
      },
      {
        "AudioMaterial": null,
        "BasicInfo": {
          "ClassPath": "/素材/视频",
          "CreateTime": "2021-11-14T15:42:09Z",
          "MaterialId": "61912e513a2f7c0001827ddda",
          "MaterialType": "VIDEO",
          "Name": "片段1.mp4",
          "Owner": {
            "Id": "6b6ef043-85f3-4614-b735-768bb466ae5b",
            "Type": "PERSON"
          },
          "PresetTagSet": [],
          "PreviewUrl": "http://14xxxxxx98.vod2.myqcloud.com/334751dbvodtranssh1400293698/a958de3738770229144662908d/coverBySnapshot/coverBySnapshot_10_0.jpg",
          "TagSet": [],
          "UpdateTime": "2021-11-14T15:53:17Z"
        }
      }
    ]
  }
}
```

```

    },
    "ImageMaterial": null,
    "LinkMaterial": null,
    "OtherMaterial": null,
    "VideoEditTemplateMaterial": null,
    "VideoMaterial": null
  }
],
"ClassInfoSet": [
  {
    "Owner": {
      "Id": "6b6ef043-85f3-4614-b735-768bb466ae5b",
      "Type": "PERSON"
    },
    "ClassPath": "/素材/视频/体育"
  }
],
"RequestId": "60d1b6f5-f3f3-440f-943d-a6c2d56341ec",
"MaterialTotalCount": 2
}
}

```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClassNotExist	分类不存在。
InvalidParameterValue.ClassPath	分类路径不合法。

错误码	描述
InvalidParameterValue.Limit	返回记录条数不合法。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

导入媒体

最近更新时间：2025-04-25 01:19:26

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

将云点播媒资文件导入到多媒体创作引擎媒体资源库。支持导入媒体归属团队或者个人。

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：ImportMaterial。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Owner	是	Entity	媒体归属者，可支持归属团队或个人。
Name	是	String	媒体名称，不能超过30个字符。 示例值：material_name
SourceType	否	String	导入媒资类型，取值： - VOD：云点播文件； - EXTERNAL：媒资绑定。 注意：如果不填默认为云点播文件，如果媒体存储在非腾讯云点播中，都需要使用媒资绑定。另外，导入云点播的文件，使用云点播的子应用 Id 必须与创建多媒体创作引擎平台时使用的云点播子应用一致。 示例值：VOD
VodFileId	否	String	云点播媒资 FileId，仅当 SourceType 为 VOD 时有效。 示例值：5285890796182734267
ExternalMedialInfo	否	ExternalMedialInfo	原始媒资文件信息，当 SourceType 取值 EXTERNAL 的时候必填。
ClassPath	否	String	媒体分类路径，形如："/a/b/"，层级数不能超过10，每个层级长度不能超过15字符。若不填则默认为根路径。 示例值：/media
PreProcessDefinition	否	Integer	媒体预处理任务参数 ID。可取值有： 10：进行编辑预处理。 示例值：10
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以向任意团队或者个人导入媒体。如果指定操作者，如果媒体归属为个人，则操作者必须与归属者一致；如果媒体归属为团队，则必须为团队可导入媒体的团队成员(如果没有特殊设置，所有团队成员可导入媒体)。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
MaterialId	String	媒体 Id。 示例值：materialId
PreProcessTaskId	String	媒体文件预处理任务 ID，如果未指定发起预处理任务则为空。 示例值：taskId
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 导入云点播媒体到团队

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMaterial
&Platform=1000000009
&SourceType=VOD
&VodFileId=5285890796182734267
&Owner.Id=cmetid_37b1bb3bf8fb2eb3ba00e41dcca0ce
&Owner.Type=TEAM
&Name=媒体名称
&ClassPath=/媒体
&Operator=user_id_0a54392053f84053942f930
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "3dFd8ad3d628dc30001bd09dd6",
    "PreProcessTaskId": "",
    "RequestId": "0a543920-53f8-4053-942f-9308c49404d30"
  }
}
```

示例2 导入云点播文件到个人

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMaterial
&Platform=1000000009
&SourceType=VOD
&VodFileId=528589079618273426711
&Owner.Id=user_id_0a54392053f84053942f930
&Owner.Type=PERSON
&Name=媒体名称
&ClassPath=/媒体
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "3dFd8ad3d628dc30001bd09dd5",
```

```
"PreProcessTaskId": "",
"RequestId": "0a543920-53f8-4053-942f-9308c49404d7"
}
}
```

示例3 导入外部媒体到个人并发起预处理

导入一个 COS 桶中媒体到个人，且发起预处理。

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMaterial
&Platform=1000000009
&SourceType=EXTERNAL
&Owner.Id=user_id_0a54392053f84053942f930
&Owner.Type=PERSON
&Name=媒体名称
&ClassPath=/媒体
&ExternalMediaInfo.MediaKey=example-folder/example-object.mp4
&PreProcessDefinition=10
&Operator=user_id_0a54392053f84053942f930
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "3dfd8ad3d628dc30001bd09dd9",
    "PreProcessTaskId": "125xxxxxxx9-tfusion-0a54392053f84053942f93089kji3d3",
    "RequestId": "0a543920-53f8-4053-942f-9308c49408kd8"
  }
}
```

示例4 导入外部媒体到团队并发起预处理

导入一个 COS 桶中媒体到团队，且发起预处理。

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMaterial
&Platform=1000000009
&SourceType=EXTERNAL
&Owner.Id=cmetid_37b1bb3bf8fb2eb3ba00e41dcca0ce
&Owner.Type=TEAM
&Name=媒体名称
&ClassPath=/媒体
&ExternalMediaInfo.MediaKey=example-folder/example-object.mp4
&PreProcessDefinition=10
&Operator=user_id_0a54392053f84053942f930
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "3dfd8ad3d628dc30001bd09da0",
```

```
"PreProcessTaskId": "125xxxxxxx9-tfusion-0a54392053f84053942f93089kji3d4",
"RequestId": "0a543920-53f8-4053-942f-9308c49409634"
}
}
```

示例5 导入云点播媒体到团队并发起预处理

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMaterial
&Platform=1000000009
&SourceType=VOD
&VodFileId=5285890796182734267
&Owner.Id=cmetid_37b1bb3bf8fb2eb3ba00e41dcca0ce
&Owner.Type=TEAM
&Name=媒体名称
&ClassPath=/媒体
&PreProcessDefinition=10
&Operator=user_id_0a54392053f84053942f930
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "3dfd8ad3d628dc30001bd09dd6",
    "PreProcessTaskId": "125xxxxxxx9-tfusion-0a54392053f84053942f9308978d3",
    "RequestId": "0a543920-53f8-4053-942f-9308c49404d30"
  }
}
```

示例6 导入云点播文件到个人并发起预处理

输入示例

```
https://cme.tencentcloudapi.com/?Action=ImportMaterial
&Platform=1000000009
&SourceType=VOD
&VodFileId=528589079618273426711
&Owner.Id=user_id_0a54392053f84053942f930
&Owner.Type=PERSON
&Name=媒体名称
&ClassPath=/媒体
&PreProcessDefinition=10
&Operator=user_id_0a54392053f84053942f930
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "3dfd8ad3d628dc30001bd09dd6",
    "PreProcessTaskId": "125xxxxxxx9-tfusion-0a54392053f84053942f9308c49404d8",
    "RequestId": "0a738920-53f8-4053-942f-9308c494099d1d"
  }
}
```

```
}  
  
}
```

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
FailedOperation.InvalidVodUser	云点播用户无效。
InternalServerError	内部错误。
InternalServerError.CreateTask	发起预处理任务失败。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.ExternalMediaInfoNotExist	原始媒资信息不存在。
InvalidParameterValue.Name	名称字段编辑错误，需使用 UTF8 编码。
InvalidParameterValue.NameLenLimt	名称字段长度超长，限制 30 个中文字符。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.PreProcessDefinition	预处理模板 ID 无效。
InvalidParameterValue.VodFileId	参数缺少 VodFile。
InvalidParameterValue.VodFileNotExist	点播文件不存在。
InvalidParameterValue.VodSubAppid	绑定的点播子应用无效。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。

错误码	描述
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取媒体详情

最近更新时间：2025-06-04 01:21:15

- - - - -

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

根据媒体 Id 批量获取媒体详情。

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：DescribeMaterials。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
MaterialIds.N	是	Array of String	媒体 ID 列表，一次最多可拉取20个媒体的信息。 示例值：["5fd8ad3d628dc30001bd0895"]
Sort	否	SortBy	列表排序，支持下列排序字段： - CreateTime：创建时间； - UpdateTime：更新时间。
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以获取任意媒体的信息。如果指定操作者，则操作者必须对媒体有读权限。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
MaterialInfoSet	Array of MaterialInfo	媒体列表信息。
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 获取图片媒体详情

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeMaterials
&Platform=1000000009
&MaterialIds.0=5fd8ad3d628dc30001bd08894
&Operator=user_id4114428941bd0420
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialInfoSet": [
      {
        "AudioMaterial": null,
        "BasicInfo": {
          "ClassPath": "",
          "CreateTime": "2019-10-28T12:10:25Z",
          "MaterialId": "5fd8ad3d628dc30001bd08894",
          "MaterialType": "IMAGE",
          "Name": "dzq1.png",
          "Owner": {
            "Id": "03ce2cd6-a889-4e4d-95a7-876e44f21831",
            "Type": "PERSON"
          },
          "PreviewUrl": "https://1810000002.vod2.myqcloud.com/b64e98aevodcq1810000002/0d014fa85285890795320102944/j5dCbqBV72QA.png",
          "TagSet": [],
          "PresetTagSet": [],
          "UpdateTime": "2019-10-28T12:10:25Z"
        },
        "ImageMaterial": {
          "Height": 153,
          "MaterialUrl": "https://1810000002.vod2.myqcloud.com/b64e98aevodcq1810000002/0d014fa85285890795320102944/j5dCbqBV72QA.png",
          "Size": 52421,
          "Width": 213,
          "VodFileId": "15e83cf8b48b147",
          "OriginalUrl": "http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.png"
        },
        "LinkMaterial": null,
        "VideoMaterial": null,
        "VideoEditTemplateMaterial": null,
        "OtherMaterial": null
      }
    ],
    "RequestId": "964aae95-7344-4200-8962-92a44ff7a2aa"
  }
}
```

示例2 获取视频媒体详情

视频媒体

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeMaterials
&Platform=1000000009
&MaterialIds.0=5fd8ad3d628dc30001bd0895
&Operator=user_id4114428941bd0420
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialInfoSet": [
```

```
{
  "BasicInfo": {
    "MaterialId": "5fd8ad3d628dc30001bd0895",
    "MaterialType": "VIDEO",
    "Name": "9Qvacao2r7EA",
    "CreateTime": "2019-12-31T03:59:14Z",
    "UpdateTime": "2019-12-31T03:59:14Z",
    "ClassPath": "/media",
    "TagSet": [],
    "PresetTagSet": [],
    "PreviewUrl": "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/coverBySnapshot/1577764755_4069421039.100_0.jpg",
    "Owner": {
      "Type": "PERSON",
      "Id": "6b6ef043-85f3-4614-b735-768bb466ae5b"
    }
  },
  "ImageMaterial": null,
  "LinkMaterial": null,
  "AudioMaterial": null,
  "VideoEditTemplateMaterial": null,
  "OtherMaterial": null,
  "VideoMaterial": {
    "VodFileId": "15e83cf8b48b1472",
    "OriginalUrl": "http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.mp4",
    "MaterialStatus": {
      "EditorUsableStatus": "NORMAL"
    },
    "MetaData": {
      "Size": 2148950,
      "Container": "mov,mp4,m4a,3gp,3g2,mj2",
      "Bitrate": 242237,
      "Height": 270,
      "Width": 480,
      "Duration": 69.4729995727539,
      "Rotate": 0,
      "VideoStreamInfoSet": [
        {
          "Bitrate": 209241,
          "Height": 270,
          "Width": 480,
          "Codec": "h264",
          "Fps": 24
        }
      ],
      "AudioStreamInfoSet": [
        {
          "Bitrate": 32996,
          "SamplingRate": 48000,
          "Codec": "aac"
        }
      ]
    },
    "ImageSpriteInfo": {
      "Height": 80,
      "Width": 142,
      "TotalCount": 70,
```

```
"ImageUrlSet": [
  "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/imageSprite/1577764755_3800596325.100_0.jpg"
],
"WebVttUrl": "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/imageSprite/1577764755.vtt"
},
"MaterialUrl": "https://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/v.f50000.m3u8?t=5e572824&sign=f3a588b4c37d047bd2df6b02ecf11ed8",
"CoverUrl": "http://1810000002.vod2.myqcloud.com/8d388656vodtranscq1810000002/accfa04c5285890797261730431/coverBySnapshot/1577764755_4069421039.100_0.jpg",
"Resolution": "SD"
}
}
],
"RequestId": "85b8cdac-8e18-430a-aeae-08076ac26a11"
}
}
```

示例3 获取音频媒体详情

音频媒体

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeMaterials
&Platform=1000000009
&MaterialIds.0=5fd8ad3d628dc30001bd08893
&Operator=user_id4114428941bd0420
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialInfoSet": [
      {
        "AudioMaterial": {
          "VodFileId": "15e83cf8b48b1473",
          "OriginalUrl": "http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.mp4",
          "CoverUrl": "",
          "MaterialStatus": {
            "EditorUsableStatus": "NORMAL"
          },
          "MaterialUrl": "https://1810000002.vod2.myqcloud.com/b64e98aevodcq1810000002/6df9fb785285890795345485561/FOMbrvesytgA.mp3?t=5eaa3d36&sign=72576faefb2863396f2775bd45259a44",
          "MetaData": {
            "AudioStreamInfoSet": [
              {
                "Bitrate": 320000,
                "Codec": "mp3",
                "SamplingRate": 44100
              }
            ],
            "Bitrate": 320047,
            "Container": "mp3",
            "Duration": 221.04815673828,
```

```
"Height": 0,
"Rotate": 0,
"Size": 8843241,
"VideoStreamInfoSet": [],
"Width": 0
},
},
"BasicInfo": {
"ClassPath": "/media",
"CreateTime": "2019-10-29T09:36:33Z",
"MaterialId": "5fd8ad3d628dc30001bd08893",
"MaterialType": "AUDIO",
"Name": "Fgi.mp3",
"Owner": {
"Id": "03ce2cd6-a889-4e4d-95a7-876e44f21831",
"Type": "PERSON"
},
"PreviewUrl": "",
"TagSet": [],
"PresetTagSet": [],
"UpdateTime": "2019-10-29T09:37:29Z"
},
"ImageMaterial": null,
"LinkMaterial": null,
"VideoMaterial": null,
"VideoEditTemplateMaterial": null,
"OtherMaterial": null
}
],
"RequestId": "964aae95-7344-4200-8962-92a44ff7a2ac"
}
}
```

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
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

删除媒体

最近更新时间：2025-04-25 01:19:26

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

根据媒体 Id 删除媒体。

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：DeleteMaterial。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
MaterialId	是	String	媒体 Id。 示例值：397ff3d4730c2e09ba24298f742add6346107826
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验媒体删除权限。 示例值：user_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 删除媒体

删除媒体

输入示例

```
https://cme.tencentcloudapi.com/?Action=DeleteMaterial
&Platform=1000000009
&MaterialId=397ff3d4730c2e09ba24298f742add6346107826
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "7bf08361-4455-4cbd-afd6-423b62c54a05"
```

```
}  
  
}
```

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。
InvalidParameterValue	参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

创建链接

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

创建媒体链接或分类路径链接，将资源信息链接到目标。

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：CreateLink。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Type	是	String	链接类型，可取值有： - CLASS: 分类链接； - MATERIAL: 媒体文件链接。 示例值：MATERIAL
Name	是	String	链接名称，不能超过30个字符。 示例值：link
Owner	是	Entity	链接归属者。
DestinationId	是	String	目标资源Id。可取值有： - 当 Type 为 MATERIAL 时填媒体 ID； - 当 Type 为 CLASS 时填写分类路径。 示例值：5fec3806b4e0960001ed9ce2
DestinationOwner	是	Entity	目标资源归属者。
ClassPath	否	String	链接的分类路径，如填"/a/b"则代表链接属于该分类路径，不填则默认为根路径。 示例值：/a/b
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以创建任意源及目标资源的链接。如果指定操作者，则操作者必须对源资源有读权限，对目标媒体有写权限。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
MaterialId	String	新建链接的媒体 Id。 示例值：5fec3806b4e0960001ed9ce5
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 创建指向分类的链接

创建指向分类路径为 /a/c 的链接。

输入示例

```
https://cme.tencentcloudapi.com/?Action=CreateLink
&Platform=1000000009
&Type=CLASS
&Name=link
&ClassPath=/a/b
&Owner.Id=38f5d76a1c4fbad03b26d258253fe468
&Owner.Type=PERSON
&DestinationId=/a/c
&DestinationOwner.Id=38f5d76a1c4fbad03b26d258253fe468
&DestinationOwner.Type=PERSON
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "5fec3806b4e0960001ed9ce61",
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb656"
  }
}
```

示例2 创建指向别人媒体的链接

为媒体 ID 为 5fec38060544f4e4260001ed9ce5 且属于用户 ID 为 38f5d76a1c4fbad03b26d258253fe468 的媒体创建链接，其分类路径为 /a/b 。

输入示例

```
https://cme.tencentcloudapi.com/?Action=CreateLink
&Platform=1000000009
&Type=MATERIAL
&Name=link
&ClassPath=/a/b
&Owner.Id=b4f5d76a1c4fbad03b26d258847c6217
&Owner.Type=PERSON
&DestinationId=5fec38060544f4e4260001ed9ce5
&DestinationOwner.Id=38f5d76a1c4fbad03b26d258253fe468
&DestinationOwner.Type=PERSON
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "5fec3806b4e0960001ed9ce6",
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb657"
  }
}
```

示例3 创建指向自己媒体的链接

为媒体 ID 为 5fec3806b4e0960001ed9ce2 且属于自己的媒体创建链接，其分类路径为 /a/b 。

输入示例

```
https://cme.tencentcloudapi.com/?Action=CreateLink
&Platform=1000000009
&Type=MATERIAL
&Name=link
&ClassPath=/a/b
&Owner.Id=b4f5d76a1c4fbad03b26d258847c6217
&Owner.Type=PERSON
&DestinationId=5fec3806b4e0960001ed9ce2
&DestinationOwner.Id=b4f5d76a1c4fbad03b26d258847c6217
&DestinationOwner.Type=PERSON
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "5fec3806b4e0960001ed9ce5",
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb65b"
  }
}
```

示例4 创建指向别人团队媒体的链接

为媒体 ID 为 5feb077183bac900010975bf 且属于团队 ID 为 cmetid_1eaea85c92e98f87ea8984441ae40280 的媒体创建链接，其分类路径为 /a/b 。

输入示例

```
https://cme.tencentcloudapi.com/?Action=CreateLink
&Platform=1000000009
&Type=MATERIAL
&Name=link
&ClassPath=/a/b
&Owner.Id=b4f5d76a1c4fbad03b26d258847c6217
&Owner.Type=PERSON
&DestinationId=5feb077183bac900010975bf
&DestinationOwner.Id=cmetid_1eaea85c92e98f87ea8984441ae40280
&DestinationOwner.Type=TEAM
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "MaterialId": "5fec3806b4e0960001ed9ce6",
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb657"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClassNotExist	分类不存在。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.MaterialId	媒体 ID 无效。
InvalidParameterValue.NameLenLimit	名称字段长度超长，限制 30 个中文字符。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.VodSubAppId	绑定的点播子应用无效。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

移动资源

最近更新时间：2025-04-25 01:19:26

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

移动资源，支持跨个人或团队移动媒体以及分类。如果填写了Operator，则需要校验用户对媒体和分类资源的访问以及写权限。

- 当原始资源为媒体时，该接口效果为将该媒体移动到目标分类下面；
- 当原始资源为分类时，该接口效果为将原始分类移动到目标分类或者是重命名。

如果 SourceResource.Resource.Id = /素材/视频/NBA，DestinationResource.Resource.Id= /素材/视频/篮球

- 当 DestinationResource.Resource.Id 不存在时候且原始资源与目标资源归属相同，操作结果为重命名原始分类；
- 当 DestinationResource.Resource.Id 存在时候，操作结果为产生新目录 /素材/视频/篮球/NBA

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：MoveResource。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
SourceResource	是	ResourceInfo	待移动的原始资源信息，包含原始媒体或分类资源，以及资源归属。
DestinationResource	是	ResourceInfo	目标信息，包含分类及归属，仅支持移动资源到分类。
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以移动任务资源。如果指定操作者，则操作者必须对源及目标资源有写权限。 示例值：c9bf9e57-1685-4c89-bafb-ff5af830be8a

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 跨归属移动媒体

将归属于个人的媒体文件移动到团队

输入示例

```
https://cme.tencentcloudapi.com/?Action=MoveResource
&Platform=1000000009
&SourceResource.Resource.Id=123245678
&SourceResource.Resource.Type=MATERIAL
```

```
&SourceResource.Owner.Id=6b6ef043-85f3-4614-b735-xxx
&SourceResource.Owner.Type=PERSON
&DestinationResource.Resource.Id=/个人/视频
&DestinationResource.Resource.Type=CLASS
&DestinationResource.Owner.Id=cmetid_5fa4f63306e69dxx
&DestinationResource.Owner.Type=TEAM
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb65b"
  }
}
```

示例2 移动媒体到某分类下面

输入示例

```
https://cme.tencentcloudapi.com/?Action=MoveResource
&Platform=1000000009
&SourceResource.Resource.Id=123245678
&SourceResource.Resource.Type=MATERIAL
&SourceResource.Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
&SourceResource.Owner.Type=PERSON
&DestinationResource.Resource.Id=/个人/视频
&DestinationResource.Resource.Type=CLASS
&DestinationResource.Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
&DestinationResource.Owner.Type=PERSON
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "d1c5eb0e-e499-4419-b465-60c0973b3d2"
  }
}
```

示例3 移动分类到已经存在的分类下

目标分类已经存在，操作结果为将原始分类的资源移到新分类 /个人/娱乐/综艺

输入示例

```
https://cme.tencentcloudapi.com/?Action=MoveResource
&Platform=1000000009
&SourceResource.Resource.Id=/个人/综艺
&SourceResource.Resource.Type=CLASS
&SourceResource.Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
&SourceResource.Owner.Type=PERSON
&DestinationResource.Resource.Id=/个人/娱乐
&DestinationResource.Resource.Type=CLASS
&DestinationResource.Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
```

```
&DestinationResource.Owner.Type=PERSON
```

```
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb65b"
  }
}
```

示例4 移动分类到不存在的分类下，资源归属相同

目标分类不存在，操作结果为将原始分类重命名为 /个人/娱乐

输入示例

```
https://cme.tencentcloudapi.com/?Action=MoveResource
&Platform=1000000009
&SourceResource.Resource.Id=/个人/综艺
&SourceResource.Resource.Type=CLASS
&SourceResource.Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
&SourceResource.Owner.Type=PERSON
&DestinationResource.Resource.Id=/个人/娱乐
&DestinationResource.Resource.Type=CLASS
&DestinationResource.Owner.Id=6b6ef043-85f3-4614-b735-768bb466ae5b
&DestinationResource.Owner.Type=PERSON
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb65b"
  }
}
```

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
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.MaterialId	媒体 ID 无效。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。

媒资授权相关接口

发起媒资授权

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

资源归属者对个人或团队授予目标资源的相应权限。

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：GrantResourceAuthorization。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Owner	是	Entity	资源归属者，个人或者团队。
Resources.N	是	Array of Resource	被授权资源。
Authorizees.N	是	Array of Entity	被授权目标，个人或者团队。
Permissions.N	是	Array of String	详细授权值。取值有： - R：可读，可以浏览媒体，但不能使用该媒体文件（将其添加到 Project），或复制到自己的媒资库中 - X：可用，可以使用该素材（将其添加到 Project），但不能将其复制到自己的媒资库中，意味着被授权者无法将该资源进一步扩散给其他个人或团队。 - C：可复制，既可以使用该素材（将其添加到 Project），也可以将其复制到自己的媒资库中。 - W：可修改、删除媒资。 示例值：["R\n"]
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以授权任意归属者的资源。如果指定操作者，则操作者必须对资源拥有写权限。 示例值： <code>cmeid_system</code>

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 发起媒资授权

输入示例

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

{
  "Authorizes": [
    {
      "Type": "TEAM",
      "Id": "teamA"
    }
  ],
  "Platform": "1000000009",
  "Operator": "cmeid_system",
  "Owner": {
    "Type": "PERSON",
    "Id": "aa2b2421-425f-435d-b580-d417bdd73dd8"
  },
  "Resources": [
    {
      "Type": "MATERIAL",
      "Id": "mid_abc"
    }
  ],
  "Permissions": [
    "R"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d1c5eb0e-e499-4419-b465-60c0973b3d2"
  }
}
```

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。
InvalidParameterValue.ClassNotExist	分类不存在。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取资源授权列表

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

查询资源被授权的情况。

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：DescribeResourceAuthorization。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000001
Owner	否	Entity	归属者。 示例值：{"Type":"PERSON","Id":"e7abc68e-6908-0000-2222-3221728e1111"}
Resource	否	Resource	资源。 示例值：{"Type":"CLASS","Id":"1"}
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以查询任意资源的被授权情况。如果指定操作者，则操作者必须对被授权资源有读权限。 示例值：1234c68e-1111-2222-3333-3221728e4444

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的资源授权记录总数。 示例值：1
AuthorizationInfoSet	Array of AuthorizationInfo	授权信息列表。 示例值：[{"Authorizee":{"Type":"PERSON","Id":"234"},"PermissionSet":["R"
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 获取媒资授权信息

输入示例

```
POST / HTTP/1.1
Host: cme.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeResourceAuthorization
```

<公共请求参数>

```
{
  "Operator": "cmeid_system",
  "Platform": "1000000001",
  "Resource": {
    "Type": "MATERIAL",
    "Id": "67e105de4fb8950001052fff"
  },
  "Owner": {
    "Type": "TEAM",
    "Id": "b718fcee09a72470bf665cffff"
  }
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "AuthorizationInfoSet": [
      {
        "Authorizee": {
          "Type": "PERSON",
          "Id": "d1c5eb0ee4994419b46560"
        },
        "PermissionSet": [
          "R",
          "W"
        ]
      }
    ],
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb65b"
  }
}
```

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
InternalError	内部错误。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取共享空间

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取共享空间。当个人或团队A对个人或团队B授权某资源以后，个人或团队B的共享空间就会增加个人或团队A。

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：DescribeSharedSpace。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Authorizee	是	Entity	被授权目标，个人或团队。
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以查询任意个人或者团队的共享空间。如果指定操作者，则操作者必须本人或者团队成员。 示例值：teamAmemberXXX

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	查询到的共享空间总数。 示例值：1
AuthorizerSet	Array of Authorizer	各个共享空间对应的授权者信息。
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 获取共享空间

输入示例

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

{
```

```
"Operator": "user_id_d1c5eb0ee4994419b465",
"Platform": "1000000009",
"Authorizee": {
  "Type": "TEAM",
  "Id": "cmetid_acc5dde49933d2b4338ud8d9"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "AuthorizerSet": [
      {
        "Type": "PERSON",
        "Id": "user_id_d1c5eb0ee4994419b465"
      }
    ],
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb6c9"
  }
}
```

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。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。

错误码	描述
UnauthorizedOperation	未授权操作。

撤销资源授权

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

资源所属实体对目标实体撤销目标资源的相应权限，若原本没有相应权限则不产生变更。

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：RevokeResourceAuthorization。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Owner	是	Entity	资源所属实体。
Resources.N	是	Array of Resource	被授权资源。
Authorizees.N	是	Array of Entity	被授权目标实体。
Permissions.N	是	Array of String	详细授权值。取值有： - R：可读，可以浏览素材，但不能使用该素材（将其添加到 Project），或复制到自己的媒资库中 - X：可用，可以使用该素材（将其添加到 Project），但不能将其复制到自己的媒资库中，意味着被授权者无法将该资源进一步扩散给其他个人或团队。 - C：可复制，既可以使用该素材（将其添加到 Project），也可以将其复制到自己的媒资库中。 - W：可修改、删除媒资。 示例值：["R\r\n"]
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，撤销任意资源的授权权限。如果指定操作者，则操作者必须对被授权资源有写权限。 示例值： <code>cmeid_system</code>

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 撤销个人资源权限

撤销素材 Id `5fd8ad3d628dc30001bd0895` 对团队 `cmetid_afd3ad3d6289k390d9dk09ad0` 授予的读权限。

输入示例

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

```
{
  "Authorizes": [
    {
      "Type": "TEAM",
      "Id": "cmetid_afd3ad3d6289k390d9dk09ad0"
    }
  ],
  "Operator": "user_id_1c5ebabd4392343b6489560",
  "Platform": "1000000009",
  "Owner": {
    "Type": "PERSON",
    "Id": "user_id_1c5ebabd4392343b6489560"
  },
  "Resources": [
    {
      "Type": "MATERIAL",
      "Id": "5fd8ad3d628dc30001bd0895"
    }
  ],
  "Permissions": [
    "R"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d1c5eb0e-e499-4419-b465-60c097fbb65b"
  }
}
```

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
InternalError	内部错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound	资源不存在。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

团队管理相关接口

创建团队

最近更新时间：2025-04-25 01:19:28

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

创建一个团队。

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：CreateTeam。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
Name	是	String	团队名称，限30个字符。 示例值：测试团队名称
OwnerId	是	String	团队所有者，指定用户 ID。 示例值：997988688
OwnerRemark	否	String	团队所有者的备注，限30个字符。 示例值：管理员李四
TeamId	否	String	自定义团队 ID。创建后不可修改，限20个英文字符及“-”。同时不能以 cmetid_ 开头。不填会生成默认团队 ID。 示例值：fa3a12938b007acff7adb59a9e4ae7b9

3. 输出参数

参数名称	类型	描述
TeamId	String	创建的团队 ID。 示例值：cmetid_19882998399iiei87neu78
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 创建一个团队

输入示例

```
POST / HTTP/1.1
Host: cme.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: CreateTeam
<公共请求参数>

{
  "Platform": "1000000009",
  "Name": "测试团队名称",
  "OwnerRemark": "管理员李四",
  "OwnerId": "997988688"
}
```

输出示例

```
{
  "Response": {
    "TeamId": "cmetid_19882998399iiei87neu78",
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.OwnerRemark	团队 Owner 的备注不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TeamId	团队 ID 已经存在，不能重复创建。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。

错误码	描述
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

添加团队成员

最近更新时间：2025-04-25 01:19:28

-

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

向一个团队中添加团队成员，并且指定成员的角色。

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：AddTeamMember。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamId	是	String	团队 ID。 示例值：1kdk882ddd88338ddd3k3
TeamMembers.N	是	Array of AddMemberInfo	要添加的成员列表，一次最多添加30个成员。
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以向任意团队中添加成员。如果指定操作者，则操作者必须为管理员或者团队所有者。 示例值：60097a8a8ea83d00017b484b

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 将一个成员添加到团队

输入示例

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

{
  "Platform": "1000000009",
  "TeamId": "1kdk882ddd88338ddd3k3",
  "TeamMembers": [
    {
      "Remark": "编辑张三",
      "MemberId": "1kd3ej3okjdije3lejjdk99"
```

```
}  
]  
}
```

输出示例

```
{  
  "Response": {  
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"  
  }  
}
```

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。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TeamId	团队 ID 已经存在，不能重复创建。
InvalidParameterValue.TeamNotExist	团队不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取团队信息

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取指定团队的信息，拉取团队信息列表。

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：DescribeTeams。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamIds.N	否	Array of String	团队 ID 列表，限30个。若不填，则默认获取平台下所有团队。 示例值：["1kdk882ddd88338ddd3k3\n","1kdk882ddd88338ddd3k4\n"]
Offset	否	Integer	分页偏移量，默认值：0。 示例值：0
Limit	否	Integer	返回记录条数，默认值：20，最大值：30。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的记录总数。 示例值：2
TeamSet	Array of TeamInfo	团队列表。
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，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 分布获取团队信息

输入示例

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

```
{
  "Platform": "1000000009",
  "Limit": "10",
  "Offset": "0"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "TeamSet": [
      {
        "TeamId": "1kdk882ddd88338ddd3k3",
        "Name": "工作室1",
        "MemberCount": 30,
        "CreateTime": "2020-03-09T02:35:21Z",
        "UpdateTime": "2020-07-06T04:55:21Z"
      }
    ],
    "RequestId": "98534d9f-aeec-4817-bbe2-d3499f95a2af"
  }
}
```

示例2 获取指定团队信息

输入示例

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

{
  "TeamIds": [
    "1kdk882ddd88338ddd3k4",
    "1kdk882ddd88338ddd3k3"
  ],
  "Platform": "1000000009"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "TeamSet": [
      {
        "TeamId": "1kdk882ddd88338ddd3k3",
        "Name": "工作室1",
        "MemberCount": 30,
        "CreateTime": "2020-03-09T02:35:21Z",
        "UpdateTime": "2020-07-06T04:55:21Z"
      }
    ]
  }
}
```

```
},
{
  "TeamId": "1kdk882ddd88338ddd3k4",
  "Name": "工作室2",
  "MemberCount": 10,
  "CreateTime": "2020-04-03T09:25:21Z",
  "UpdateTime": "2020-06-06T03:16:21Z"
}
],
"RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
}
```

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.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TeamNotExist	团队不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取团队成员信息

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取指定团队的成员信息。支持获取指定成员的信息，同时也支持分页拉取指定团队的所有成员信息。

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：DescribeTeamMembers。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamId	是	String	团队 ID。 示例值：1kdk882ddd88338ddd3k3
MemberIds.N	否	Array of String	成员 ID 列表，限指定30个指定成员。如不填，则返回指定团队下的所有成员。 示例值：["378b753f-f91f-4c42-8fa3-34a2a3bf3c56"]
Offset	否	Integer	分页偏移量，默认值：0 示例值：0
Limit	否	Integer	返回记录条数，默认值：30，最大值：30。 示例值：10
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以拉取任意团队成员的信息。如果指定操作者，则操作者必须为团队成员。 示例值：60097a8a8ea83d00017b484b

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的记录总数。 示例值：30
MemberSet	Array of TeamMemberInfo	团队成员列表。
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，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 拉取指定成员的信息

输入示例

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

```
{
  "Platform": "1000000009",
  "TeamId": "1kdk882ddd88338ddd3k3",
  "MemberIds": [
    "1993939392iidkei8ei"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "MemberSet": [
      {
        "MemberId": "1993939392iidkei8ei",
        "Remark": "主编辑李四",
        "Role": "Members"
      }
    ],
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
  }
}
```

示例2 分页拉取团队成员信息

输入示例

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

```
{
  "Platform": "1000000009",
  "TeamId": "1kdk882ddd88338ddd3k3",
  "Limit": "10",
  "Offset": "0"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 3,
    "MemberSet": [
      {
        "MemberId": "299addf39392iidkei38",
```

```
"Remark": "主编辑李四",
"Role": "Owner"
},
{
  "MemberId": "39a39a9b92iidkei9ab",
  "Remark": "主编辑李四",
  "Role": "Admin"
},
{
  "MemberId": "i9d1ld9393d2iidkei89i0",
  "Remark": "主编辑李四",
  "Role": "Members"
}
],
"RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
}
```

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.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Limit	返回记录条数不合法。
InvalidParameterValue.Offset	分页偏移不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TeamNotExist	团队不存在。

错误码	描述
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

获取成员加入的团队列表

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取用户所加入的团队列表

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：DescribeJoinTeams。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
MemberId	是	String	团队成员 ID。 示例值：1993939392iidkei8ei
Offset	否	Integer	分页偏移量，默认值：0。 示例值：0
Limit	否	Integer	返回记录条数，默认值：30，最大值：30。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的记录总数。 示例值：2
TeamSet	Array of JoinTeamInfo	团队列表。
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 获取用户所加入的团队列表

输入示例

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

{
```

```
"Platform": "1000000009",
"MemberId": "1993939392iidkei8ei"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "TeamSet": [
      {
        "TeamId": "1kdk882ddd88338ddd3k3",
        "Name": "工作室1",
        "MemberCount": 30,
        "Role": "Admin"
      },
      {
        "TeamId": "1kdk882ddd88338ddd3k4",
        "Name": "工作室2",
        "MemberCount": 10,
        "Role": "Member"
      }
    ],
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
  }
}
```

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。

错误码	描述
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

修改团队成员

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

修改团队成员信息，包括成员备注、角色等。

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：ModifyTeamMember。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamId	是	String	团队 ID。 示例值：1kdk882ddd88338ddd3k3
MemberId	是	String	团队成员 ID。 示例值：1kd3ej3okjdije3lejddjk99
Remark	否	String	成员备注，长度不能超过15个字符。 示例值：主编辑李四
Role	否	String	成员角色，可取值有： - Admin：团队管理员； - Member：普通成员。 示例值：Admin
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以修改任意团队成员的信息。如果指定操作者，则操作者必须为团队的管理员或者所有者。 示例值：60097a8a8ea83d00017b484b

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 修改一个团队成员的备注

输入示例

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

{
  "Platform": "1000000009",
  "TeamId": "1kdk882ddd88338ddd3k3",
  "Remark": "主编辑李四",
  "MemberId": "1kd3ej3okjdije3lejddjk99"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a2bf"
  }
}
```

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。
InvalidParameterValue.MemberNotExist	团队成员不存在。
InvalidParameterValue.NotTeamMemberError	非团队成员。
InvalidParameterValue.Operator	无效的操作者。

错误码	描述
InvalidParameterValue.Role	角色不合法。
InvalidParameterValue.TeamNotExist	团队不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

修改团队信息

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

修改团队信息，目前支持修改的操作有：

- 修改团队名称。

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：ModifyTeam。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamId	是	String	团队 ID。 示例值：997988688
Name	否	String	团队名称。团队名称不能置空，并且不能超过30个字符。 示例值：测试团队名称
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以修改所有团队的信息。如果指定操作者，则操作者必须为团队管理员或者所有者。 示例值：60097a8a8ea83d00017b484b

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 修改团队名称

输入示例

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

{
```

```
"Platform": "1000000009",
"TeamId": "997988688",
"Name": "测试团队名称"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "99004d9f-aeec-4817-bbe2-d3499f95a3ba"
  }
}
```

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。
InvalidParameterValue.Name	名称字段编辑错误，需使用 UTF8 编码。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TeamNotExist	团队不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

删除团队成员

最近更新时间：2025-04-25 01:19:27

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

将团队成员从团队中删除。

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：DeleteTeamMembers。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamId	是	String	团队 ID。 示例值：1kdk882ddd88338ddd3k3
MemberIds.N	是	Array of String	要删除的成员列表。 示例值：["999kd3ej3okjdije3lejjdk99\r\n","999kd3ej3okjdije3lejjdk98\r\n"]
Operator	否	String	操作者。如不填，默认为 <code>cmeid_system</code> ，表示平台管理员操作，可以删除所有团队的成员。如果指定操作者，则操作者必须为团队管理员或者所有者。 示例值：60097a8a8ea83d00017b484b

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 删除一个团队成员

输入示例

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

{
  "Platform": "1000000009",
```

```
"TeamId": "1kdk882ddd88338ddd3k3",
"MemberIds": [
  "999kd3ej3okjdije3lejddjk98",
  "999kd3ej3okjdije3lejddjk99"
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "89d14d9f-aeec-4817-bbe2-d3499f95ae8a"
  }
}
```

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。
InvalidParameterValue.MemberIds	无效的成员列表。
InvalidParameterValue.TeamNotExist	团队不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。

删除团队

最近更新时间：2025-04-25 01:19:28

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

删除一个团队。要删除团队，必须满足以下条件：

- 要删除的团队必须没有归属的素材；
- 要删除的团队必须没有归属的分类。

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：DeleteTeam。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
TeamId	是	String	要删除的团队 ID。 示例值：9aoodiiiekj899dkk99d7d6d5dff77f
Operator	否	String	操作者。如不填，默认为 cmeid_system，表示平台管理员操作，可以删除所有团队。如果指定操作者，则操作者必须为团队所有者。 示例值：60097a8a8ea83d00017b484b

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 删除团队

输入示例

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

{
  "Platform": "1000000009",
```

```
"TeamId": "9aoodiiekj899dkk99d7d6d5dff77f"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "9109d6d9f-aeec-4817-bbe2-d3499f95a3ac"
  }
}
```

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。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.TeamNotExist	团队不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

分类管理相关接口

获取分类列表

最近更新时间：2025-04-25 01:19:29

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

获取指定归属者下所有的分类信息。

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：DescribeClass。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
Owner	是	Entity	归属者。
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验操作权限。 示例值：c9bf9e57-1685-4c89-bafb-ff5af830be8a

3. 输出参数

参数名称	类型	描述
ClassInfoSet	Array of ClassInfo	分类信息列表。
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 获取分类列表

输入示例

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

{
  "Owner": {
```

```
"Type": "PERSON",
  "Id": "c9bf9e57-1685-4c89-bafb-ff5af830be8a"
},
  "Platform": "1000000009"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "ClassInfoSet": [
      {
        "Owner": {
          "Id": "c9bf9e57-1685-4c89-bafb-ff5af830be8a",
          "Type": "PERSON"
        },
        "ClassPath": "/a/b"
      }
    ]
  }
}
```

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。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。

错误码	描述
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

最近更新时间：2025-04-25 01:19:29

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

新增分类，用于管理素材。分类层数不能超过20。

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：CreateClass。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
Owner	是	Entity	归属者。
ClassPath	是	String	分类路径。 示例值：/a/b
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验分类创建权限。 示例值：aaa1e081-***-9162840cadb7

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 创建分类

输入示例

```
https://cme.tencentcloudapi.com/?Action=CreateClass
&Platform=1000000009
&Owner.Id=a-4d6c-8ee5-9a3c
&Owner.Type=PERSON
&ClassPath=/a/b
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "f47ac10b-58cc-4372-a567-0e02b2c3d479"
```

```
}  
  
}
```

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.Platform	平台参数非法。
InvalidParameterValue.ClassExist	分类已存在。
InvalidParameterValue.ClassNotExist	分类不存在。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.Platform	平台参数取值错误。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

移动分类

最近更新时间：2025-04-25 01:19:28

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

移动某一个分类到另外一个分类下，也可用于分类重命名。

如果 `SourceClassPath` = /素材/视频/NBA，`DestinationClassPath` = /素材/视频/篮球

- 当 `DestinationClassPath` 不存在时候，操作结果为重命名 `ClassPath`；
- 当 `DestinationClassPath` 存在时候，操作结果为产生新目录 /素材/视频/篮球/NBA

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：MoveClass。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Owner	是	Entity	归属者。
SourceClassPath	是	String	源分类路径。 示例值：/媒资
DestinationClassPath	是	String	目标分类路径。 示例值：/媒资/综艺
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验操作权限。 示例值：9b92a8e7-a5ef-4e28-d72e-4eff571fabf0

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 移动分类

移动分类

输入示例

```
POST / HTTP/1.1
Host: cme.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: MoveClass
```

<公共请求参数>

```
{
  "Platform": "1000000009",
  "Owner": {
    "Type": "PERSON",
    "Id": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
  },
  "SourceClassPath": "/媒资",
  "DestinationClassPath": "/媒资/综艺",
  "Operator": "9b92a8e7-a5ef-4e28-d72e-4eff571fabf0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "requestId"
  }
}
```

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。
InvalidParameterValue.ClassExist	分类已存在。
InvalidParameterValue.ClassNotExist	分类不存在。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.DstClassPathNotExist	目标分类不存在。

错误码	描述
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

删除分类

最近更新时间：2025-04-25 01:19:29

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

删除分类信息，删除时检验下述限制：

- 分类路径必须存在；
- 分类下没有绑定素材。

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：DeleteClass。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
Owner	是	Entity	归属者。
ClassPath	是	String	分类路径。 示例值：/a/b
Operator	否	String	操作者。填写用户的 Id，用于标识调用者及校验操作权限。 示例值：c9bf9e57-1685-4c89-bafb-ff5af830be8a

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 删除分类

输入示例

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

{
  "Owner": {
    "Type": "PERSON",
    "Id": "c9bf9e57-1685-4c89-bafb-ff5af830be8a"
```

```
},
"Platform": "1000000009",
"ClassPath": "/a/b"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "requestId"
  }
}
```

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。
InvalidParameterValue.ClassExist	分类已存在。
InvalidParameterValue.ClassNotEmpty	分类不为空。
InvalidParameterValue.ClassNotExist	分类不存在。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

登录态管理相关接口

查询登录态

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

查询指定用户的登录态。

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：DescribeLoginStatus。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台 Id，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
UserIds.N	是	Array of String	用户 Id 列表，N 从0开始取值，最大19。 示例值：["c9bf9e57-1685-4c89-bafb-ff5af830be8a"]

3. 输出参数

参数名称	类型	描述
LoginStatusInfoSet	Array of LoginStatusInfo	用户登录状态列表。
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 查询用户登录态

输入示例

```
https://cme.tencentcloudapi.com/?Action=DescribeLoginStatus
&Platform=1000000009
&UserIds.0=user_id_5285890811433511063
&UserIds.1=user_id_5285890811433511064
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "LoginStatusInfoSet": [
      {
        "UserId": "user_id_5285890811433511063",
        "Status": "Online"
      },
      {
        "UserId": "user_id_5285890811433511064",
        "Status": "Offline"
      }
    ],
    "RequestId": "requestId"
  }
}
```

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。
InvalidParameterValue.Platform	平台参数取值错误。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

删除登录态

最近更新时间：2025-04-25 01:19:25

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

删除用户登录态，使用户登出多媒体创作引擎平台。

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：DeleteLoginStatus。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
UserIds.N	是	Array of String	用户 Id 列表，N 从 0 开始取值，最大 19。 示例值：["111\r\n"]

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 删除用户登录态

输入示例

```
https://cme.tencentcloudapi.com/?Action=DeleteLoginStatus
&Platform=1000000009
&UserIds.0=866c9a7-a2ca-4
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "requestId"
  }
}
```

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。
ResourceNotFound.Platform	平台不存在。
UnauthorizedOperation	未授权操作。

参数模板相关接口

创建视频编码配置

最近更新时间：2025-04-25 01:19:28

1. 接口描述

接口请求域名：cme.tencentcloudapi.com 。

指定导出的参数，创建一个视频编码配置

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：CreateVideoEncodingPreset。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
Name	是	String	配置名，可用来简单描述该配置的作用。 示例值：测试模板
Container	否	String	封装格式，可选值： - mp4； - mov。 默认值：mp4。 示例值：mp4
RemoveVideo	否	Integer	是否去除视频数据，可选值： 0：保留； 1：去除。 默认值：0。 示例值：0
RemoveAudio	否	Integer	是否去除音频数据，可选值： 0：保留； 1：去除。 默认值：0。 示例值：1
VideoSetting	否	VideoEncodingPresetVideoSetting	编码配置的视频设置。默认值参考VideoEncodingPresetVideoSetting 定义。
AudioSetting	否	VideoEncodingPresetAudioSetting	编码配置的音频设置。默认值参考VideoEncodingPresetAudioSetting 定义。

3. 输出参数

参数名称	类型	描述
Id	Integer	模板 ID。 示例值：10
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360p的视频编码配置

创建360p的视频编码配置

详细功能：去除音频流，设置码率为500k bps

输入示例

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

{
  "Platform": "1000000009",
  "Name": "自定义360p编码配置",
  "RemoveAudio": "1",
  "VideoSetting": {
    "ShortEdge": "360",
    "Bitrate": "512000"
  }
}
```

输出示例

```
{
  "Response": {
    "Id": 100050,
    "RequestId": "e33fffc2-02c9-4074-9355-88443869e88b"
  }
}
```

示例2 创建单声道音频编码配置

输入示例

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

{
  "Platform": "1000000009",
  "Name": "自定义单声道音频编码配置",
  "AudioSetting": {
    "Channels": "1"
  }
}
```

```
}  
}
```

输出示例

```
{  
  "Response": {  
    "Id": 100049,  
    "RequestId": "22392574-2d1d-4418-94bb-a71ee67fbc56"  
  }  
}
```

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。
ResourceNotFound.Platform	平台不存在。

删除视频编码配置

最近更新时间：2025-04-25 01:19:28

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

删除指定 ID 的视频编码配置

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：DeleteVideoEncodingPreset。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。 示例值：1000000009
Id	是	Integer	要删除的视频编码配置 ID。 示例值：100050

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 删除视频编码配置

输入示例

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

{
  "Platform": "1000000009",
  "Id": 100050
}
```

输出示例

```
{
  "Response": {
    "RequestId": "117ff5e6-df9e-4a56-b48a-ecc1ddae02f0"
  }
}
```

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。
ResourceNotFound.Platform	平台不存在。

修改视频编码配置

最近更新时间：2025-04-25 01:19:28

1. 接口描述

接口请求域名：cme.tencentcloudapi.com。

修改视频编码配置信息。

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：ModifyVideoEncodingPreset。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Id	是	Integer	配置 ID。 示例值：100048
Name	否	String	更改后的视频编码配置名，不填则不修改。 示例值：demo
RemoveVideo	否	Integer	是否去除视频数据，可选值： 0：保留； 1：去除。 默认值：0。 示例值：0
RemoveAudio	否	Integer	是否去除音频数据，可选值： 0：保留； 1：去除。 默认值：0。 示例值：0
VideoSetting	否	VideoEncodingPresetVideoSettingForUpdate	更改后的编码配置的视频设置。
AudioSetting	否	VideoEncodingPresetAudioSettingForUpdate	更改后的编码配置的音频设置。

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 修改编码配置名称

输入示例

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

{
  "Platform": "1000000009",
  "Name": "demo",
  "Id": 100048
}
```

输出示例

```
{
  "Response": {
    "RequestId": "4e6ab84c-47c5-45af-8f47-ec39eaf0606d"
  }
}
```

示例2 修改编码配置中的视频尺寸

输入示例

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

{
  "Platform": "1000000009",
  "Id": 100048,
  "VideoSetting": {
    "ShortEdge": "1024"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7164f965-2923-4640-9010-05c51b282433"
  }
}
```

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
ResourceNotFound.Platform	平台不存在。

查询视频编码配置

最近更新时间：2025-05-27 01:20:43

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

查询视频编码配置信息。

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：DescribeVideoEncodingPresets。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
Ids.N	否	Array of Integer	要查询的配置 ID 列表。填写该参数则按照配置 ID 进行查询。 示例值：[100047]
Limit	否	Integer	分页大小，默认20。最大值50。 示例值：20
Offset	否	Integer	分页起始，默认0。 示例值：0

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	符合条件的编码配置总个数。 示例值：32
VideoEncodingPresetSet	Array of VideoEncodingPreset	视频编码配置信息。
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 指定 ID 查询配置

输入示例

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

```
{
  "Platform": "1000000009",
  "Ids": [
    100047
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "46ada659-495d-43b3-88c8-64dad423007d",
    "TotalCount": 1,
    "VideoEncodingPresetSet": [
      {
        "AudioSetting": {
          "Bitrate": 64000,
          "Channels": 2,
          "Codec": "AAC",
          "SampleRate": 16000
        },
        "Container": "mp4",
        "Id": 100047,
        "Name": "",
        "RemoveAudio": 0,
        "RemoveVideo": 0,
        "VideoSetting": {
          "Bitrate": 0,
          "Codec": "H264",
          "ShortEdge": 1080
        }
      }
    ]
  }
}
```

示例2 按照分页查询配置

输入示例

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

{
  "Platform": "1000000009",
  "Offset": 0,
  "Limit": 2
}
```

输出示例

```
{
  "Response": {
    "RequestId": "f16e0eea-20d0-4716-8195-599df164d8b4",
    "TotalCount": 32,
    "VideoEncodingPresetSet": [
      {
        "AudioSetting": {
          "Bitrate": 64000,
          "Channels": 2,
          "Codec": "AAC",
          "SampleRate": 16000
        },
        "Container": "mp4",
        "Id": 100047,
        "Name": "",
        "RemoveAudio": 0,
        "RemoveVideo": 0,
        "VideoSetting": {
          "Bitrate": 0,
          "Codec": "H264",
          "ShortEdge": 1080
        }
      },
      {
        "AudioSetting": {
          "Bitrate": 64000,
          "Channels": 1,
          "Codec": "AAC",
          "SampleRate": 32000
        },
        "Container": "mp4",
        "Id": 100048,
        "Name": "test1",
        "RemoveAudio": 0,
        "RemoveVideo": 0,
        "VideoSetting": {
          "Bitrate": 131072,
          "Codec": "H264",
          "ShortEdge": 1024
        }
      }
    ]
  }
}
```

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。
InvalidParameter.Id	Id 无效。
InvalidParameterValue.Platform	平台参数取值错误。
ResourceNotFound.Platform	平台不存在。

其他接口

回调事件内容解析

最近更新时间：2025-06-04 01:21:16

1. 接口描述

接口请求域名：`cme.tencentcloudapi.com`。

该接口接受多媒体创作引擎回调给业务的事件内容，将其转化为对应的 `EventContent` 结构。请不要实际调用该接口，只需要将接收到的事件内容直接使用 `JSON` 解析到 `EventContent` 结构即可使用。

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：ParseEvent。
Version	是	String	公共参数 ，本接口取值：2019-10-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Platform	是	String	平台名称，指定访问的平台。关于平台概念，请参见文档 平台 。 示例值：1000000009
EventContent	是	String	回调事件内容。 示例值：{"EventType":"Class.Created","Operator":"3fa6b09e-05c6-433e-bc1e-6457d290d9d9","MaterialImportedEvent":{"Owner":{"Id":"user_9949009d9d9","Type":"PERSON"},"ClassPath":"/媒体"}}

3. 输出参数

参数名称	类型	描述
EventContent	EventContent	事件内容。
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 分析分类创建事件内容

输入示例

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

{
  "Platform": "1000000009",
```

```
"EventContent": "{\n  \"EventType\": \"Class.Created\",\n  \"Operator\": \"3fa6b09e-05c6-433e-bc1e-6457d290d9d9\",\n  \"MaterialImportedEvent\": {\n    \"Owner\": {\n      \"Id\": \"user_9949009d9d9\",\n      \"Type\": \"PERSON\",\n      \"ClassPath\": \"媒体\"\n    }\n  }\n}"
```

输出示例

```
{\n  \"Response\": {\n    \"EventContent\": {\n      \"EventType\": \"Class.Created\",\n      \"Operator\": \"3fa6b09e-05c6-433e-bc1e-6457d290d9d9\",\n      \"ClassCreatedEvent\": {\n        \"Owner\": {\n          \"Id\": \"user_9949009d9d9\",\n          \"Type\": \"PERSON\"\n        },\n        \"ClassPath\": \"媒体\"\n      },\n      \"ProjectStreamConnectStatusChangedEvent\": {\n        \"ProjectId\": \"\",\n        \"Status\": \"\",\n        \"InputInterruptInfo\": null,\n        \"OutputInterruptInfo\": null\n      },\n      \"StorageNewFileCreatedEvent\": {\n        \"FileId\": \"\",\n        \"MaterialId\": \"\",\n        \"OperationType\": \"\",\n        \"Owner\": {\n          \"Type\": \"\",\n          \"Id\": \"\"\n        },\n        \"ClassPath\": \"\",\n        \"TaskId\": \"\",\n        \"SourceContext\": \"\"\n      },\n      \"ProjectSwitcherStatusChangedEvent\": null,\n      \"MaterialImportedEvent\": null,\n      \"MaterialAddedEvent\": null,\n      \"MaterialMovedEvent\": null,\n      \"MaterialModifiedEvent\": null,\n      \"MaterialDeletedEvent\": null,\n      \"ClassMovedEvent\": null,\n      \"ClassDeletedEvent\": null,\n      \"VideoExportCompletedEvent\": null,\n      \"ProjectMediaCastStatusChangedEvent\": null\n    },\n    \"RequestId\": \"946c2480-a14f-4d18-8a09-31a45cbd21af\"\n  }\n}
```

示例2 分析媒体导入事件内容

输入示例

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

{
  "Platform": "1000000009",
  "EventContent": "{\n\"EventType\": \"Material.Imported\", \"Operator\": \"3fa6b09e-05c6-433e-bc1e-6457d290d9d9\", \"MaterialImportedEvent\": {\n\"MediaInfoSet\": [\n\"9dd28590d08987ujd795d\", \"rdd28590d08187ujd7095d\"], \"Owner\": {\n\"Id\": \"user_9949009d9d9\", \"Type\": \"PERSON\"}, \"ClassPath\": \" /媒体 \"}}}"
}
```

输出示例

```
{
  "Response": {
    "EventContent": {
      "EventType": "Material.Imported",
      "Operator": "3fa6b09e-05c6-433e-bc1e-6457d290d9d9",
      "MaterialImportedEvent": {
        "MediaInfoSet": [
          {
            "MaterialId": "9dd28590d08987ujd795d"
          },
          {
            "MaterialId": "rdd28590d08187ujd7095d"
          }
        ],
        "Owner": {
          "Id": "user_9949009d9d9",
          "Type": "PERSON"
        },
        "ClassPath": " /媒体 "
      },
      "ProjectStreamConnectStatusChangedEvent": {
        "ProjectId": "",
        "Status": "",
        "InputInterruptInfo": null,
        "OutputInterruptInfo": null
      },
      "StorageNewFileCreatedEvent": {
        "FileId": "",
        "MaterialId": "",
        "OperationType": "",
        "Owner": {
          "Type": "",
          "Id": ""
        },
        "ClassPath": "",
        "TaskId": "",
        "SourceContext": ""
      },
      "ProjectSwitcherStatusChangedEvent": null,
      "MaterialAddedEvent": null,
      "MaterialMovedEvent": null,
    }
  }
}
```

```
"MaterialModifiedEvent": null,
"MaterialDeletedEvent": null,
"ClassCreatedEvent": null,
"ClassMovedEvent": null,
"ClassDeletedEvent": null,
"VideoExportCompletedEvent": null,
"ProjectMediaCastStatusChangedEvent": null
},
"RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
}
}
```

示例3 分析新文件产生事件内容

输入示例

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

{
  "Platform": "1000000009",
  "EventContent": "{\n\"EventType\": \"Storage.NewFileCreated\", \"Operator\": \"3fa6b09e-05c6-433e-bc1e-6457d290d9d9\", \"Storage\nNewFileCreatedEvent\": {\n\"FileId\": \"5285890818667693851\", \"MaterialId\": \"60a75c1c30fcbe0001cc09d5\", \"OperationType\n\": \"Upload\", \"Owner\": {\n\"Type\": \"PERSON\", \"Id\": \"3fa6b09e-05c6-433e-bc1e-6457d264f1dd\"}, \"ClassPath\": \"媒体\"}}}"
}
```

输出示例

```
{
  "Response": {
    "EventContent": {
      "EventType": "Storage.NewFileCreated",
      "Operator": "3fa6b09e-05c6-433e-bc1e-6457d290d9d9",
      "StorageNewFileCreatedEvent": {
        "FileId": "85890818667698972737",
        "MaterialId": "60a75c1c30fcbe00093ikd93k",
        "OperationType": "Upload",
        "Owner": {
          "Type": "PERSON",
          "Id": "3fa6b09e-05c6-433e-bc1e-6457d290d9d9"
        },
        "ClassPath": "媒体",
        "TaskId": "",
        "SourceContext": ""
      },
      "ProjectStreamConnectStatusChangedEvent": {
        "ProjectId": "",
        "Status": "",
        "InputInterruptInfo": null,
        "OutputInterruptInfo": null
      },
      "ProjectSwitcherStatusChangedEvent": null,
      "MaterialImportedEvent": null,
    }
  }
}
```

```
"MaterialAddedEvent": null,
"MaterialMovedEvent": null,
"MaterialModifiedEvent": null,
"MaterialDeletedEvent": null,
"ClassCreatedEvent": null,
"ClassMovedEvent": null,
"ClassDeletedEvent": null,
"VideoExportCompletedEvent": null,
"ProjectMediaCastStatusChangedEvent": null
},
"RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
}
}
```

示例4 分析云转推项目状态变更事件内容

输入示例

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

{
  "Platform": "10000000009",
  "EventContent": "{\n\"EventType\":\n\"Project.StreamConnect.StatusChanged\",\n\"Operator\":\n\"3fa6b09e-05c6-433e-bc1e-6457d290d9d9\",\n\"ProjectStreamConnectStatusChangedEvent\":{\n\"ProjectId\":\n\"5285890818667693851\",\n\"Status\":\n\"Working\"}\n}"
}
```

输出示例

```
{
  "Response": {
    "EventContent": {
      "EventType": "Project.StreamConnect.StatusChanged",
      "Operator": "3fa6b09e-05c6-433e-bc1e-6457d290d9d9",
      "ProjectStreamConnectStatusChangedEvent": {
        "ProjectId": "5285890818667693851",
        "Status": "Working",
        "InputInterruptInfo": null,
        "OutputInterruptInfo": null
      },
      "StorageNewFileCreatedEvent": {
        "FileId": "",
        "MaterialId": "",
        "OperationType": "",
        "Owner": {
          "Type": "",
          "Id": ""
        },
        "ClassPath": "",
        "TaskId": "",
        "SourceContext": ""
      },
      "ProjectSwitcherStatusChangedEvent": null,

```

```
"MaterialImportedEvent": null,
"MaterialAddedEvent": null,
"MaterialMovedEvent": null,
"MaterialModifiedEvent": null,
"MaterialDeletedEvent": null,
"ClassCreatedEvent": null,
"ClassMovedEvent": null,
"ClassDeletedEvent": null,
"VideoExportCompletedEvent": null,
"ProjectMediaCastStatusChangedEvent": null
},
"RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
}
}
```

示例5 分析导播台项目状态变更事件内容

输入示例

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

{
  "Platform": "1000000009",
  "EventContent": "{\"EventType\":\"Project.Switcher.StatusChanged\",\"Operator\":\"3fa6b09e-05c6-433e-bc1e-6457d290d9d9\", \"ProjectSwitcherStatusChangedEvent\":{\"ProjectId\":\"add28590d08187ujd7695d\",\"Status\":\"Started\"}}"
```

输出示例

```
{
  "Response": {
    "EventContent": {
      "EventType": "Project.Switcher.StatusChanged",
      "Operator": "3fa6b09e-05c6-433e-bc1e-6457d290d9d9",
      "ProjectSwitcherStatusChangedEvent": {
        "ProjectId": "add28590d08187ujd7695d",
        "Status": "Started"
      },
      "ProjectStreamConnectStatusChangedEvent": {
        "ProjectId": "",
        "Status": "",
        "InputInterruptInfo": null,
        "OutputInterruptInfo": null
      },
      "StorageNewFileCreatedEvent": {
        "FileId": "",
        "MaterialId": "",
        "OperationType": "",
        "Owner": {
          "Type": "",
          "Id": ""
        }
      },
    },
  },
}
```

```
"ClassPath": "",
"TaskId": "",
"SourceContext": ""
},
"MaterialImportedEvent": null,
"MaterialAddedEvent": null,
"MaterialMovedEvent": null,
"MaterialModifiedEvent": null,
"MaterialDeletedEvent": null,
"ClassCreatedEvent": null,
"ClassMovedEvent": null,
"ClassDeletedEvent": null,
"VideoExportCompletedEvent": null,
"ProjectMediaCastStatusChangedEvent": null
},
"RequestId": "946c2480-a14f-4d18-8a09-31a45cbd21af"
}
}
```

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。
InvalidParameter.Platform	平台参数非法。
ResourceNotFound.Platform	平台不存在。

数据结构

最近更新时间：2025-06-04 01:21:16

AccountInfo

制作云用户账号信息。

被如下接口引用：DescribeAccounts。

名称	类型	描述
UserId	String	用户 Id。 示例值：1000000001
Phone	String	用户手机号码。 示例值：15515555055
Nick	String	用户昵称。 示例值：Nick
Status	String	账号状态，取值： - Normal：有效； - Stopped：无效。 示例值：Normal

AddMemberInfo

添加的团队成员信息

被如下接口引用：AddTeamMember。

名称	类型	必选	描述
MemberId	String	是	团队成员 ID。 示例值：c9bf9e57-1685-4c89-bafb-ff5af830be8a
Remark	String	否	团队成员备注。 示例值：new_admin
Role	String	否	团队成员角色，不填则默认添加普通成员。可选值： - Admin：团队管理员； - Member：普通成员。 示例值：Member

AudioMaterial

音频素材信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
MetaData	MediaMetaData	素材元信息。
MaterialUrl	String	素材媒体文件的播放 URL 地址。 示例值：http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.mp3
CoverUrl	String	素材媒体文件的封面图片地址。 示例值：http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.jpg
MaterialStatus	MaterialStatus	素材状态。 注意：此字段可能返回 null，表示取不到有效值。
OriginalUrl	String	素材媒体文件的原始 URL 地址。 示例值：http://cmedemo.vod2.myqcloud.com/original/xxxxx/cemdemo.mp3

名称	类型	描述
VodFileId	String	云点播媒资 FileId。 示例值：15e83cf8b48b147223

AudioStreamInfo

音频流信息。

被如下接口引用：DescribeTaskDetail。

名称	类型	描述
Bitrate	Integer	码率，单位：bps。 示例值：1411000
SamplingRate	Integer	采样率，单位：hz。 示例值：44100
Codec	String	编码格式。 示例值：aac

AudioTrackItem

音频轨道上的音频片段信息。

被如下接口引用：CreateProject。

名称	类型	必选	描述
SourceType	String	是	音频媒体来源类型，取值有： • VOD：素材来源于云点播文件； • CME：视频来源于制作云媒体文件； • EXTERNAL：视频来源于媒资绑定，如果媒体不是存储在腾讯云点播中或者云创中，都需要使用媒资绑定。 示例值：EXTERNAL
SourceMedia	String	是	音频媒体，可取值为： • 当 SourceType 为 VOD 时，参数填云点播 FileId； • 当 SourceType 为 CME 时，参数填多媒体创作引擎媒体 Id； • 当 SourceType 为 EXTERNAL 时，目前仅支持外部媒体 URL(如 https://www.example.com/a.mp3)，参数填写规则请参见注意事项。 注意： • 当 SourceType 为 EXTERNAL 并且媒体 URL Scheme 为 https 时(如：https://www.example.com/a.mp3)，参数为：1000000:www.example.com/a.mp3。 • 当 SourceType 为 EXTERNAL 并且媒体 URL Scheme 为 http 时(如：http://www.example.com/b.mp3)，参数为：1000001:www.example.com/b.mp3。 示例值：1000000:www.qq.com/a.mp4
SourceMediaStartTime	Float	否	音频片段取自媒体文件的起始时间，单位为秒。0 表示从媒体开始位置截取。默认为0。 示例值：0
Duration	Float	否	音频片段的时长，单位为秒。默认和媒体本身长度一致，表示截取全部媒体。 示例值：20

AuthorizationInfo

资源权限信息

被如下接口引用：DescribeResourceAuthorization。

名称	类型	描述
Authorizee	Entity	被授权者实体。
PermissionSet	Array of String	详细授权值。取值有： • R：可读，可以浏览素材，但不能使用该素材（将其添加到 Project），或复制到自己的媒资库中。 • X：可用，可以使用该素材（将其添加到 Project），但不能将其复制到自己的媒资库中，意味着被授权者无法将该资源进一步扩散给其他个人或团队。 • C：可复制，既可以使用该素材（将其添加到 Project），也可以将其复制到自己的媒资库中。 • W：可修改、删除媒资。 示例值：["R"]

Authorizer

授权者

被如下接口引用：DescribeSharedSpace。

名称	类型	必选	描述
Type	String	是	授权者类型，取值有： • PERSON：个人。 • TEAM：团队。 示例值：PERSON
Id	String	是	Id，当 Type=PERSON，取值为用户 Id。当Type=TEAM，取值为团队 ID。 示例值：c9bf9e57-1685-psid-demo-ff5af830be8a

CMEExportInfo

多媒体创作引擎导出信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
Owner	Entity	是	导出媒体归属，个人或团队。
Name	String	否	导出的媒体名称，不得超过30个字符。 示例值：素材
Description	String	否	导出的媒体信息，不得超过50个字符。 示例值：素材
ClassPath	String	否	导出的媒体分类路径，长度不能超过15字符。不存在默认创建。 示例值：/素材
TagSet	Array of String	否	导出的媒体标签，单个标签不得超过10个字符。 示例值：["素材"]
ThirdPartyPublishInfos	Array of ThirdPartyPublishInfo	否	第三方平台发布信息列表。暂未正式对外，请勿使用。

ClassCreatedEvent

分类创建事件。

被如下接口引用：ParseEvent。

名称	类型	描述
Owner	Entity	分类归属。

名称	类型	描述
ClassPath	String	分类路径。 示例值：/媒体

ClassDeletedEvent

分类删除事件。

被如下接口引用：ParseEvent。

名称	类型	描述
Owner	Entity	删除的分类归属。
ClassPathSet	Array of String	删除的分类路径列表。 示例值：["/path"]

ClassInfo

分类信息

被如下接口引用：DescribeClass, DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
Owner	Entity	归属者。
ClassPath	String	分类路径。 示例值：/my/media/class

ClassMovedEvent

分类移动事件。

被如下接口引用：ParseEvent。

名称	类型	描述
SourceOwner	Entity	源分类归属。
SourceClassPathSet	Array of String	源分类路径列表。 示例值：/素材
DestinationOwner	Entity	目标分类归属。
DestinationClassPath	String	目标分类归属。 示例值：/媒体

CosPublishInputInfo

COS 发布信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
Bucket	String	是	发布生成的对象存储文件所在的 COS Bucket 名，如 TopRankVideo-125xxx88。 示例值：Video-125xxx88
Region	String	是	发布生成的对象存储文件所在的 COS Bucket 所属园区，如 ap-chongqing。 示例值：ap-chongqing
VideoKey	String	是	发布生成的视频在 COS 存储的对象键。对象键（ObjectKey）是对象（Object）在存储桶（Bucket）中的唯一标识。 示例值：fadd70539811.mp4
CoverKey	String	是	发布生成的封面在 COS 存储的对象键。

名称	类型	必选	描述
			示例值：fadd70539812.png

EmptyTrackItem

空的轨道片段，用来进行时间轴的占位。如需要两个音频片段之间有一段时间的静音，可以用 EmptyTrackItem 来进行占位。

被如下接口引用：CreateProject。

名称	类型	必选	描述
Duration	Float	是	持续时间，单位为秒。 示例值：30.0

Entity

用于描述资源的归属，归属者为个人或者团队。

被如下接口引用：CopyProject, CreateClass, CreateLink, CreateProject, DeleteClass, DescribeClass, DescribeMaterials, DescribeProjects, DescribeResourceAuthorization, DescribeSharedSpace, ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject, FlattenListMedia, GrantResourceAuthorization, ImportMaterial, ListMedia, ModifyMaterial, ModifyProject, MoveClass, MoveResource, ParseEvent, RevokeResourceAuthorization, SearchMaterial。

名称	类型	必选	描述
Type	String	是	类型，取值有： - PERSON：个人。 - TEAM：团队。 示例值：PERSON
Id	String	是	Id，当 Type=PERSON，取值为用户 Id，当 Type=TEAM，取值为团队 Id。 示例值：c9bf9e57-1685-4c89-bafb-ff5af830be8a

EventContent

回调事件内容。

被如下接口引用：ParseEvent。

名称	类型	描述
EventType	String	事件类型，可取值有： - Storage.NewFileCreated：新文件产生事件； - Project.StreamConnect.StatusChanged：云转推项目状态变更事件； - Project.Switcher.StatusChanged：导播台项目状态变更事件； - Material.Imported：媒体导入事件； - Material.Added：媒体添加事件； - Material.Moved：媒体移动事件； - Material.Modified：媒体变更事件； - Material.Deleted：媒体删除事件； - Class.Created：分类新增事件； - Class.Moved：分类移动事件； - Class.Deleted：分类删除事件； - Task.VideoExportCompleted：视频导出完成事件； - Project.MediaCast.StatusChanged：点播转直播项目状态变更事件。 示例值：Storage.NewFileCreated

名称	类型	描述
Operator	String	操作者，表示触发事件的操作者。如果是 cmeid_system 表示平台管理员操作。 示例值：378b753f-f91f-4c42-8fa3-34a2a3bf3c56
StorageNewFileCreatedEvent	StorageNewFileCreatedEvent	新文件产生事件。仅当 EventType 为 Storage.NewFileCreated 时有效。
ProjectStreamConnectStatusChangedEvent	ProjectStreamConnectStatusChangedEvent	云转推项目状态变更事件。仅当 EventType 为 Project.StreamConnect.StatusChanged 时有效。
ProjectSwitcherStatusChangedEvent	ProjectSwitcherStatusChangedEvent	导播台项目状态变更事件。仅当 EventType 为 Project.Switcher.StatusChanged 时有效。 注意：此字段可能返回 null，表示取不到有效值。
MaterialImportedEvent	MaterialImportedEvent	媒体导入事件。仅当 EventType 为 Material.Imported 时有效。 注意：此字段可能返回 null，表示取不到有效值。
MaterialAddedEvent	MaterialAddedEvent	媒体添加事件。仅当 EventType 为 Material.Added 时有效。 注意：此字段可能返回 null，表示取不到有效值。
MaterialMovedEvent	MaterialMovedEvent	媒体移动事件。仅当 EventType 为 Material.Moved 时有效。 注意：此字段可能返回 null，表示取不到有效值。
MaterialModifiedEvent	MaterialModifiedEvent	媒体更新事件。仅当 EventType 为 Material.Modified 时有效。 注意：此字段可能返回 null，表示取不到有效值。
MaterialDeletedEvent	MaterialDeletedEvent	媒体删除事件。仅当 EventType 为 Material.Deleted 时有效。 注意：此字段可能返回 null，表示取不到有效值。
ClassCreatedEvent	ClassCreatedEvent	分类创建事件。仅当 EventType 为 Class.Created 时有效。 注意：此字段可能返回 null，表示取不到有效值。
ClassMovedEvent	ClassMovedEvent	分类移动事件。仅当 EventType 为 Class.Moved 时有效。 注意：此字段可能返回 null，表示取不到有效值。
ClassDeletedEvent	ClassDeletedEvent	分类删除事件。仅当 EventType 为 Class.Deleted 时有效。 注意：此字段可能返回 null，表示取不到有效值。
VideoExportCompletedEvent	VideoExportCompletedEvent	视频导出完成事件。仅当 EventType 为 Task.VideoExportCompleted 时有效。 注意：此字段可能返回 null，表示取不到有效值。
ProjectMediaCastStatusChangedEvent	ProjectMediaCastStatusChangedEvent	点播转直播项目状态变更事件。仅当 EventType 为 Project.MediaCast.StatusChanged 时有效。 注意：此字段可能返回 null，表示取不到有效值。

ExternalMediaInfo

媒资绑定资源信息，包含媒资绑定模板 ID 和文件信息。

被如下接口引用：ImportMaterial, ImportMediaToProject。

名称	类型	必选	描述
MediaKey	String	是	目前仅支持绑定 COS 桶的媒体，请填写存储对象 Key 值，例如：example-folder/example.mp4。 示例值：www.qq.com/a.mp4

名称	类型	必选	描述
StorageId	String	否	媒资挂载的存储 Id。 示例值：60097a8a8ea83d00017b484b

ImageMaterial

图片素材信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
Height	Integer	图片高度，单位：px。 示例值：1024
Width	Integer	图片宽度，单位：px。 示例值：1024
MaterialUrl	String	素材媒体文件的展示 URL 地址。 示例值：http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.png
Size	Integer	图片大小，单位：字节。 示例值：596106
OriginalUrl	String	素材媒体文件的原始 URL 地址。 示例值：http://cmedemo.vod2.myqcloud.com/original/xxxxx/cemdemo.png
VodFileId	String	云点播媒资 FileId。 示例值：15e83cf8b48b14712

ImportMediaInfo

导入媒资信息

被如下接口引用：ParseEvent。

名称	类型	描述
FileId	String	云点播文件 FileId。 示例值：362565566555118451552
MaterialId	String	媒体 Id。 示例值：60097a8a8ea83d00017b484b

IntegerRange

整型范围

被如下接口引用：SearchMaterial。

名称	类型	必选	描述
LowerBound	Integer	是	按整形代表值的下限检索。 示例值：0
UpperBound	Integer	是	按整形代表值的上限检索。 示例值：100

JoinTeamInfo

加入的团队信息

被如下接口引用：DescribeJoinTeams。

名称	类型	描述
TeamId	String	团队 ID。 示例值：teamid
Name	String	团队名称。 示例值：团队一
MemberCount	Integer	团队成员个数。 示例值：10
Role	String	成员在团队中的角色，取值有： - Owner：团队所有者，添加团队成员及修改团队成员解决时不能填此角色； - Admin：团队管理员； - Member：普通成员。 示例值：Member

KuaishouPublishInfo

快手视频发布信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
Title	String	否	视频发布标题，限30个字符。 示例值：my_news

LinkMaterial

链接类型的素材信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
LinkType	String	链接类型取值： - CLASS：分类链接； - MATERIAL：素材链接。 示例值：CLASS
LinkStatus	String	链接状态取值： - Normal：正常； - NotFound：链接目标不存在； - Forbidden：无权限。 示例值：Normal
LinkMaterialInfo	LinkMaterialInfo	素材链接详细信息，当LinkType="MATERIAL"时有值。 注意：此字段可能返回 null，表示取不到有效值。
LinkClassInfo	ClassInfo	分类链接目标信息，当LinkType=“CLASS”时有值。 注意：此字段可能返回 null，表示取不到有效值。

LinkMaterialInfo

链接素材信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
BasicInfo	MaterialBasicInfo	素材基本信息。
VideoMaterial	VideoMaterial	视频素材信息。 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
AudioMaterial	AudioMaterial	音频素材信息。 注意：此字段可能返回 null，表示取不到有效值。
ImageMaterial	ImageMaterial	图片素材信息。 注意：此字段可能返回 null，表示取不到有效值。

LivePullInputInfo

直播拉流信息

被如下接口引用：CreateProject, HandleStreamConnectProject。

名称	类型	必选	描述
InputUrl	String	是	直播拉流地址。 示例值：rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600c

LiveStreamClipProjectInput

直播剪辑项目输入参数。

被如下接口引用：CreateProject。

名称	类型	必选	描述
Url	String	是	直播流播放地址，目前仅支持 HLS 和 FLV 格式。 示例值：https://liveplay.video-studio.myqcloud.com/live.flv
StreamRecordDuration	Integer	是	直播流录制时长，单位为秒，最大值为 7200。 示例值：60

LoginStatusInfo

登录态信息

被如下接口引用：DescribeLoginStatus。

名称	类型	描述
UserId	String	用户 Id。 示例值：c9bf9e57-1685-4c89-bafb-ff5af830be8a
Status	String	用户登录状态。 - Online：在线； - Offline：离线。 示例值：Online

MaterialAddedEvent

媒体添加事件。

被如下接口引用：ParseEvent。

名称	类型	描述
MaterialIdSet	Array of String	添加的媒体 Id 列表。 示例值：["445ecdf88f7071998672"]
Owner	Entity	添加的媒体归属。
ClassPath	String	添加的媒体分类路径。 示例值：/媒体

MaterialBasicInfo

媒体基本信息。

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
MaterialId	String	媒体 Id。 示例值：445ecdf88f7071998672
MaterialType	String	媒体类型，取值为： - AUDIO :音频; - VIDEO :视频; - IMAGE :图片; - LINK :链接。 - OTHER : 其他。 示例值：VIDEO
Owner	Entity	媒体归属实体。
Name	String	媒体名称。 示例值：素材
CreateTime	String	媒体文件的创建时间，使用 ISO 日期格式。 示例值：2018-10-01T18:00:00+08:00
UpdateTime	String	媒体文件的最近更新时间（如修改视频属性、发起视频处理等会触发更新媒体文件信息的操作），使用 ISO 日期格式。 示例值：2018-10-01T18:00:00+08:00
ClassPath	String	媒体的分类路径。 示例值：/媒资
PresetTagSet	Array of PresetTagInfo	预置标签列表。
TagSet	Array of String	人工标签列表。 示例值：["tag"]
PreviewUrl	String	媒体文件的预览图。 示例值：http://cme.vod-qcloud.com/123/f0.jpg

MaterialDeletedEvent

媒体删除事件。

被如下接口引用：ParseEvent。

名称	类型	描述
MaterialIdSet	Array of String	删除的媒体 Id 列表。 示例值：["445ecdf88f7071998672"]

MaterialImportedEvent

媒体导入事件

被如下接口引用：ParseEvent。

名称	类型	描述
MediaInfoSet	Array of ImportMediaInfo	导入的媒体信息列表。
Owner	Entity	媒体归属。
ClassPath	String	媒体分类路径。 示例值：/媒体

MaterialInfo

媒体详情信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
BasicInfo	MaterialBasicInfo	媒体基本信息。
VideoMaterial	VideoMaterial	视频媒体信息。 注意：此字段可能返回 null，表示取不到有效值。
AudioMaterial	AudioMaterial	音频媒体信息。 注意：此字段可能返回 null，表示取不到有效值。
ImageMaterial	ImageMaterial	图片媒体信息。 注意：此字段可能返回 null，表示取不到有效值。
LinkMaterial	LinkMaterial	链接媒体信息。 注意：此字段可能返回 null，表示取不到有效值。
VideoEditTemplateMaterial	VideoEditTemplateMaterial	模板媒体信息。 注意：此字段可能返回 null，表示取不到有效值。
OtherMaterial	OtherMaterial	其他类型媒体信息。 注意：此字段可能返回 null，表示取不到有效值。

MaterialModifiedEvent

媒体更新事件。

被如下接口引用：ParseEvent。

名称	类型	描述
MaterialId	String	媒体 Id。 示例值：60097a8a8ea83d00017b484b
Name	String	更新后的媒体名称。如未更新则为空。 示例值：素材
PresetTagIdSet	Array of String	更新后的媒体预置标签列表。如未更新媒体预置标签，则该字段为空数组。 示例值：["tag"]
TagSet	Array of String	更新后的媒体自定义标签列表。如未更新媒体自定义标签，则该字段为空数组。 示例值：["tag"]

MaterialMovedEvent

媒体移动事件

被如下接口引用：ParseEvent。

名称	类型	描述
MaterialIdSet	Array of String	要移动的媒体 Id 列表。 示例值：["445ecdf88f7071998672"]
SourceOwner	Entity	源媒体归属。
SourceClassPath	String	源媒体分类路径。 示例值：/媒体
DestinationOwner	Entity	目标媒体分类归属。 示例值：0
DestinationClassPath	String	目标媒体分类路径。 示例值：/媒体/视频

MaterialStatus

素材的状态，目前仅包含素材编辑可用状态。

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
EditorUsableStatus	String	素材编辑可用状态，取值有： - NORMAL：正常，可直接用于编辑； - ABNORMAL：异常，不可用于编辑； - PROCESSING：处理中，暂不可用于编辑。 示例值：NORMAL

MediaCastDestinationInfo

点播转直播输出信息。

被如下接口引用：CreateProject, DescribeProjects, HandleMediaCastProject, ParseEvent。

名称	类型	必选	描述
Id	String	否	输出源 Id。由系统进行分配。 示例值：fa3a12938b007acff7adb59a9e4ae7b9
PushUrl	String	否	输出直播流地址。支持的直播流类型为 RTMP 和 SRT。 示例值：rtmp://push.example.com/cme/live1
Name	String	否	输出源的名称。 示例值：输出

MediaCastDestinationInterruptInfo

点播转直播输出断流信息。

被如下接口引用：ParseEvent。

名称	类型	描述
DestinationInfo	MediaCastDestinationInfo	发生断流的输出源信息。
Reason	String	输出源断流原因，取值有： - SystemError：系统错误； - Unknown：未知错误。 示例值：SystemError

MediaCastDestinationStatus

点播转直播输出源状态信息。

被如下接口引用：HandleMediaCastProject。

名称	类型	描述
Id	String	输出源 Id，由系统分配。 示例值：209dcbca1d7ba343ba51
PushUrl	String	输出源直播地址。 示例值：rtmp://push.example.com/cme/live1
Status	String	输出源的状态。取值有： - Working：运行中； - Stopped：停止输出； - Failed：输出失败。 示例值：Working

MediaCastOutputMediaSetting

点播转直播输出媒体配置。

被如下接口引用：CreateProject, DescribeProjects, HandleMediaCastProject。

名称	类型	必选	描述
VideoSetting	MediaCastVideoSetting	是	视频配置。
FollowSourceInfo	Boolean	否	视频配置是否和第一个输入源的视频配置相同，默认值：false。如果 FollowSourceInfo 的值为 true，忽略 VideoSetting 参数。 示例值：false

MediaCastPlayInfo

点播转直播播放信息。

被如下接口引用：HandleMediaCastProject。

名称	类型	描述
Status	String	点播转直播项目运行状态，取值有： - Working：运行中； - Idle：空闲状态。 示例值：Working
CurrentSourceId	String	当前播放的输入源 Id。 示例值：209dcbca1d7ba343ba51
CurrentSourcePosition	Float	当前播放的输入源的播放位置，单位：秒。 示例值：100
CurrentSourceDuration	Float	当前播放的输入源时长，单位：秒。 示例值：100
DestinationStatusSet	Array of MediaCastDestinationStatus	输出源状态信息。
LoopCount	Integer	已经循环播放的次数。 示例值：1

MediaCastPlaySetting

播放控制参数。

被如下接口引用：CreateProject, DescribeProjects, HandleMediaCastProject。

名称	类型	必选	描述
LoopCount	Integer	否	循环播放次数。LoopCount 和 EndTime 同时只能有一个生效。默认循环播放次数为一次。如果同时设置了 LoopCount 和 EndTime 参数，优先使用 LoopCount 参数。 示例值：1
EndTime	String	否	结束时间，采用 ISO 日期格式 。 示例值：2022-12-10T15:00:00Z
AutoStartTime	String	否	自动启动时间，采用 ISO 日期格式 。 示例值：2022-12-10T15:00:00Z

MediaCastProjectInfo

点播转直播项目信息。

被如下接口引用：DescribeProjects。

名称	类型	描述
Status	String	点播转直播项目状态，取值有：

名称	类型	描述
		- Working：运行中； - Idle：空闲。 示例值：Working
SourceInfos	Array of MediaCastSourceInfo	输入源列表。
DestinationInfos	Array of MediaCastDestinationInfo	输出源列表。
OutputMediaSetting	MediaCastOutputMediaSetting	输出媒体配置。
PlaySetting	MediaCastPlaySetting	播放参数。
StartTime	String	项目启动时间。采用 ISO 日期格式 。 示例值：2021-03-23T07:51:18.029Z
StopTime	String	项目结束时间。采用 ISO 日期格式 。如果项目还在运行中，该字段为空。 示例值：2021-03-23T07:51:18.029Z
Duration	Float	推流时长，单位：秒。项目结束后，返回上次项目运行时的推流时长。如果项目是 Working 状态，返回的时长是0。 示例值：23

MediaCastProjectInput

点播转直播项目输入信息。

被如下接口引用：CreateProject。

名称	类型	必选	描述
SourceInfos	Array of MediaCastSourceInfo	否	输入源列表。输入源列表最大个数为100。
DestinationInfos	Array of MediaCastDestinationInfo	否	输出源列表。输出源列表最大个数为10。
OutputMediaSetting	MediaCastOutputMediaSetting	否	输出媒体配置。
PlaySetting	MediaCastPlaySetting	否	播放控制参数。

MediaCastSourceInfo

点播转直播输入源信息。

被如下接口引用：CreateProject, DescribeProjects, HandleMediaCastProject, ParseEvent。

名称	类型	必选	描述
Id	String	否	输入源 Id，由系统分配。 示例值：fa98bf72ad3d334bb3a8
Type	String	否	输入源的媒体类型，取值有： - CME：多媒体创作引擎的媒体文件； - VOD：云点播的媒资文件。 - EXTERNAL：非多媒体创建引擎或者云点播的媒资文件。 示例值：CME
FileId	String	否	云点播媒体文件 ID。当 Type = VOD 时必填。 示例值：a82b830e765aa979
MaterialId	String	否	多媒体创作引擎的媒体 ID。当 Type = CME 时必填。 示例值：7d7ac177781a48c14198
Offset	Float	否	文件播放的起始位置，单位：秒。默认为0，从文件头开始播放。当 Type = CME 或者 VOD 时有效。 示例值：0

名称	类型	必选	描述
Duration	Float	否	播放时长，单位：秒。默认播放整个文件。当 Type = CME 或者 VOD 时有效。 示例值：23
Url	String	否	外部文件的 Url，Type=EXTERNAL 时必填，可以是点播文件或者直播文件，支持的 Scheme 包括HTTP、HTTPS、RTMP。 示例值：rtmp://push.example.com/cme/live1

MediaCastSourceInterruptInfo

点播转直播输入断流信息。

被如下接口引用：ParseEvent。

名称	类型	描述
SourceInfo	MediaCastSourceInfo	发生断流的输入源信息。
Reason	String	输入源断开原因。取值有： - SystemError：系统错误； - Unknown：未知错误。 示例值：SystemError

MediaCastVideoSetting

点播转直播视频配置

被如下接口引用：CreateProject, HandleMediaCastProject。

名称	类型	必选	描述
Width	Integer	是	视频宽度，单位：px，默认值为1280。 示例值：1280
Height	Integer	是	视频高度，单位：px，默认值为720。支持的视频分辨率最大为1920*1080。 示例值：720
Bitrate	Integer	否	视频码率，单位：kbps，默认值为2500。最大值为10000 kbps。 示例值：2000
FrameRate	Float	否	视频帧率，单位：Hz，默认值为25。最大值为60。 示例值：25

MediaImageSpriteInfo

雪碧图

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
Height	Integer	雪碧图小图的高度。 示例值：100
Width	Integer	雪碧图小图的宽度。 示例值：100
TotalCount	Integer	雪碧图小图的总数量。 示例值：200
ImageUrlSet	Array of String	截取雪碧图输出的地址。 示例值：["http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.jpg"]
WebVttUrl	String	雪碧图子图位置与时间关系的 WebVtt 文件地址。WebVtt 文件表明了各个雪碧图小图对应的时间点，以及在雪碧大图里的坐标位置，一般被播放器用于实现预览。 示例值：https://xxx.vod2.myqcloud.com/xxx/xxx.vtt

MediaMetaData

文件元信息。

被如下接口引用：DescribeMaterials, DescribeTaskDetail, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
Size	Integer	大小。 示例值：97561
Container	String	容器类型。 示例值：mp4
Bitrate	Integer	视频流码率平均值与音频流码率平均值之和，单位：bps。 示例值：64000
Height	Integer	视频流高度的最大值，单位：px。 示例值：540
Width	Integer	视频流宽度的最大值，单位：px。 示例值：960
Duration	Float	时长，单位：秒。 示例值：20
Rotate	Integer	视频拍摄时的选择角度，单位：度 示例值：0
VideoStreamInfoSet	Array of VideoStreamInfo	视频流信息。
AudioStreamInfoSet	Array of AudioStreamInfo	音频流信息。

MediaPreprocessOperation

媒体处理视频合成任务的预处理操作。

被如下接口引用：ExportVideoByTemplate。

名称	类型	必选	描述
Type	String	是	预处理操作的类型，取值范围： ImageTextMask：图片文字遮罩。 示例值：ImageTextMask
Args	Array of String	否	预处理操作参数。 当 Type 取值 ImageTextMask 时，参数为要保留的文字。 示例值：["文字"]

MediaReplacementInfo

媒体替换信息。

被如下接口引用：ExportVideoByTemplate。

名称	类型	必选	描述
MediaType	String	否	替换的媒体类型，取值有： - CMEMaterialId：替换的媒体类型为媒体 ID； - ImageUrl：替换的媒体类型为图片 URL； 注：默认为 CMEMaterialId。 示例值：ImageUrl
MaterialId	String	否	媒体 ID。 当媒体类型取值为 CMEMaterialId 时有效。 示例值：60097a8a8ea83d00017b484b

名称	类型	必选	描述
MediaUrl	String	否	媒体 URL。 当媒体类型取值为 ImageUrl 时有效， 图片仅支持 jpg、png 格式，且大小不超过 2M 。 示例值：http://cme.vod2.myqcloud.com/123/f0.jpg
StartTimeOffset	Float	否	替换媒体选取的开始时间，单位为秒，默认为 0。 示例值：0
PreprocessOperation	MediaPreprocessOperation	否	预处理操作。 注：目前该功能暂不支持，请勿使用。

MediaTrack

轨道信息

被如下接口引用：CreateProject。

名称	类型	必选	描述
Type	String	是	轨道类型，取值有： • Video：视频轨道。视频轨道由以下 Item 组成： - VideoTrackItem - EmptyTrackItem - MediaTransitionItem • Audio：音频轨道。音频轨道由以下 Item 组成： - AudioTrackItem - EmptyTrackItem 示例值：Video
TrackItems	Array of MediaTrackItem	是	轨道上的媒体片段列表。

MediaTrackItem

媒体轨道的片段信息

被如下接口引用：CreateProject。

名称	类型	必选	描述
Type	String	是	片段类型。取值有： - Video：视频片段； - Audio：音频片段； - Empty：空白片段； - Transition：转场。 示例值：Video
VideoItem	VideoTrackItem	否	视频片段，当 Type = Video 时有效。
AudioItem	AudioTrackItem	否	音频片段，当 Type = Audio 时有效。
EmptyItem	EmptyTrackItem	否	空白片段，当 Type = Empty 时有效。空片段用于时间轴的占位。 - 如需要两个音频片段之间有一段时间的静音，可以用 EmptyTrackItem 来进行占位。 - 使用 EmptyTrackItem 进行占位，来定位某个Item。
TransitionItem	MediaTransitionItem	否	转场，当 Type = Transition 时有效。 示例值：b64e98advodcq198uje890

MediaTransitionItem

转场信息

被如下接口引用：CreateProject。

名称	类型	必选	描述
TransitionId	String	是	转场 Id 。暂只支持一个转场。 示例值：60097a8a8ea83d00017b484b
Duration	Float	否	转场持续时间，单位为秒，默认为2秒。进行转场处理的两个媒体片段，第二个片段在轨道上的起始时间会自动进行调整，设置为前面一个片段的结束时间减去转场的持续时间。 示例值：2.0

OtherMaterial

其他类型素材

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
MaterialUrl	String	素材媒体文件的播放 URL 地址。 示例值：http://cme.vod2.myqcloud.com/123/f0.jpg
VodFileId	String	云点播媒资 FileId。 示例值：a754bf0c6d0e727c

PenguinMediaPlatformPublishInfo

企鹅号发布信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
Title	String	否	视频发布标题。 示例值：视频能力
Description	String	否	视频发布描述信息。 示例值：视频能力描述
Tags	Array of String	否	视频标签。 示例值：视频
Category	Integer	否	视频分类，详见 企鹅号官网 视频分类。 示例值：64

PlatformInfo

平台信息。

被如下接口引用：DescribePlatforms。

名称	类型	描述
Platform	String	平台标识。 示例值：1000000001
Description	String	平台描述。 示例值：description
VodSubAppld	Integer	云点播子应用 Id。 示例值：1402233125
LicenseId	String	平台绑定的 license Id。 示例值：60097a8a8ea83d00017b484b
Status	String	平台状态，可取值为： - Normal：正常，可使用。； - Stopped：已停用，暂无法使用； - Expired：已过期，需要重新购买会员包。

名称	类型	描述
		示例值：Normal
CreateTime	String	创建时间，格式按照 ISO 8601 标准表示。 示例值：2021-09-02T03:13:08Z
UpdateTime	String	更新时间，格式按照 ISO 8601 标准表示。 示例值：2021-09-02T03:13:08Z

PresetTagInfo

预置标签信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	必选	描述
Id	String	是	标签 Id。 示例值：ca24d5c6cb360a98a9dc6ae1245a0f90
Name	String	是	标签名称。 示例值：视频
ParentTagId	String	否	父级预设 Id。 示例值：755d4edf7550762526088161f5cf9ae0

ProjectInfo

项目信息。

被如下接口引用：DescribeProjects。

名称	类型	描述
ProjectId	String	项目 Id。 示例值：445ecdf88f7071998672
Name	String	项目名称。 示例值：新建
AspectRatio	String	画布宽高比。 示例值：16:9
Category	String	项目类别，取值有： - VIDEO_EDIT：视频编辑。 - SWITCHER：导播台。 - VIDEO_SEGMENTATION：视频拆条。 - STREAM_CONNECT：云转推。 - RECORD_REPLAY：录制回放。 示例值：STREAM_CONNECT
Owner	Entity	归属者。
CoverUrl	String	项目封面图片地址。 示例值：http://cme.vod2.myqcloud.com/123/f0.png
StreamConnectProjectInfo	StreamConnectProjectInfo	云转推项目信息，仅当项目类别取值 STREAM_CONNECT 时有效。 注意：此字段可能返回 null，表示取不到有效值。
MediaCastProjectInfo	MediaCastProjectInfo	点播转直播项目信息，仅当项目类别取值为 MEDIA_CAST 时有效。 注意：此字段可能返回 null，表示取不到有效值。
UpdateTime	String	项目更新时间，格式按照 ISO 8601 标准表示。 示例值：2021-03-23T07:51:18.029Z
CreateTime	String	项目创建时间，格式按照 ISO 8601 标准表示。 示例值：2021-03-23T07:51:18.029Z

ProjectMediaCastStatusChangedEvent

点播转直播项目状态变更事件。

被如下接口引用：ParseEvent。

名称	类型	描述
ProjectId	String	项目 Id。 示例值：60097a8a8ea83d00017b484b
Status	String	项目状态，取值有： - Started：点播转直播开始； - Stopped：点播转直播结束； - SourceInterrupted：点播转直播输入断流； - DestinationInterrupted：点播转直播输出断流。 示例值：Started
SourceInterruptInfo	MediaCastSourceInterruptInfo	点播转直播输入断流信息，仅当 Status 取值 SourceInterrupted 时有效。
DestinationInterruptInfo	MediaCastDestinationInterruptInfo	点播转直播输出断流信息，仅当 Status 取值 DestinationInterrupted 时有效。

ProjectStreamConnectStatusChangedEvent

云转推项目状态变更事件。

被如下接口引用：ParseEvent。

名称	类型	描述
ProjectId	String	项目 Id。 示例值：60097a8a8ea83d00017b484b
Status	String	项目状态，取值有： - Working：云转推推流开始； - Stopped：云转推推流结束； - InputInterrupted：云转推输入断流； - OutputInterrupted：云转推输出断流。 示例值：PushStreamFinish
InputInterruptInfo	StreamConnectInputInterruptInfo	云转推输入断流信息，仅当 Status 取值 InputInterrupted 时有效。 注意：此字段可能返回 null，表示取不到有效值。
OutputInterruptInfo	StreamConnectOutputInterruptInfo	云转推输出断流信息，仅当 Status 取值 OutputInterrupted 时有效。 注意：此字段可能返回 null，表示取不到有效值。

ProjectSwitcherStatusChangedEvent

导播台项目状态变更事件

被如下接口引用：ParseEvent。

名称	类型	描述
ProjectId	String	导播台项目 Id。 示例值：60097a8a8ea83d00017b484b
Status	String	导播台项目状态，可取值有： - Started：导播台启动； - Stopped：导播台停止； - PvwStarted：导播台 PVW 开启； - PgmStarted：导播台 PGM 开启，输出推流开始； - PvwStopped：导播台 PVW 停止； - PgmStopped：导播台 PGM 停止，输出推流结束。 示例值：Started

RecordReplayProjectInput

录制回放项目输入信息。

被如下接口引用：CreateProject。

名称	类型	必选	描述
PullStreamUrl	String	是	录制拉流地址。 示例值：https://liveplay.video-studio.myqcloud.com/live.flv
MaterialOwner	Entity	是	录制文件归属者。
MaterialClassPath	String	是	录制文件存储分类路径。 示例值：/录制
PushStreamUrl	String	是	回放推流地址。 示例值：rtmp://push.example.com/cme/live1

Resource

用于描述资源

被如下接口引用：DescribeResourceAuthorization, GrantResourceAuthorization, MoveResource, RevokeResourceAuthorization。

名称	类型	必选	描述
Type	String	是	类型，取值有： - MATERIAL：素材。 - CLASS：分类。 示例值：MATERIAL
Id	String	是	资源 Id，当 Type 为 MATERIAL 时，取值为素材 Id；当 Type 为 CLASS 时，取值为分类路径 ClassPath。 示例值：67e105de4fb8950001052fff

ResourceInfo

资源信息，包含资源以及归属信息

被如下接口引用：MoveResource。

名称	类型	必选	描述
Resource	Resource	是	媒资和分类资源。
Owner	Entity	是	资源归属，个人或团队。

RtmpPushInputInfo

直播推流信息，包括推流地址有效时长，多媒体创作引擎后端生成直播推流地址。

被如下接口引用：CreateProject, HandleStreamConnectProject。

名称	类型	必选	描述
ExpiredSecond	Integer	是	直播推流地址有效期，单位：秒。 示例值：60
PushUrl	String	否	直播推流地址，入参不填默认由多媒体创作引擎生成。 示例值：rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600c

SearchScope

搜索空间

被如下接口引用：SearchMaterial。

名称	类型	必选	描述
Owner	Entity	是	分类路径归属。
ClassPath	String	否	按分类路径检索。不填则默认按根分类路径检索。 示例值：/media

SlotInfo

卡槽信息。

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
Id	Integer	卡槽 Id。 示例值：0
Type	String	卡槽类型，可取值有： - AUDIO：音频卡槽，可替换素材类型为 AUDIO 的音频素材； - VIDEO：视频卡槽，可替换素材类型为 VIDEO 的视频素材； - IMAGE：图片卡槽，可替换素材类型为 IMAGE 的图片素材； - TEXT：文本卡槽，可替换文本内容。 示例值：VIDEO
DefaultMaterialId	String	默认素材ID。当卡槽类型为 AUDIO，VIDEO，或 IMAGE 中的一种时有效。 示例值：60097a8a8ea83d00017b484b
DefaultTextSlotInfo	TextSlotInfo	默认文本卡槽信息。当卡槽类型为 TEXT 时有效。 注意：此字段可能返回 null，表示取不到有效值。
Duration	Float	素材时长，单位秒。 示例值：100
StartTime	Float	卡槽起始时间，单位秒。 示例值：0

SlotReplacementInfo

卡槽替换信息。

被如下接口引用：ExportVideoByTemplate。

名称	类型	必选	描述
Id	Integer	是	卡槽 Id。 示例值：0
ReplacementType	String	是	替换类型，可取值有： - AUDIO：音频； - VIDEO：视频； - IMAGE：图片； - TEXT：文本。 注意：这里必须保证替换的素材类型与模板轨道数据的素材类型一致。如果替换的类型为 Text，则必须保证模板轨道数据中相应卡槽的位置标记的是文本。 示例值：VIDEO
MediaReplacementInfo	MediaReplacementInfo	否	媒体替换信息，仅当要替换的媒体类型为音频、视频、图片时有效。 示例值：无
TextReplacementInfo	TextReplacementInfo	否	文本替换信息，仅当要替换的卡槽类型为文本时有效。

SortBy

排序

被如下接口引用：DescribeMaterials, DescribeProjects, SearchMaterial。

名称	类型	必选	描述
Field	String	是	排序字段。 示例值：CreateTime
Order	String	否	排序方式，可选值：Asc（升序）、Desc（降序），默认降序。 示例值：Asc

StorageNewFileCreatedEvent

新文件生成事件

被如下接口引用：ParseEvent。

名称	类型	描述
FileId	String	云点播文件 Id。 示例值：5285890818667693889292992
MaterialId	String	媒体 Id。 示例值：60a75c1c30fcbe0001cc8990d0
OperationType	String	操作类型，可取值有： - Upload：本地上传； - PullUpload：拉取上传； - VideoEdit：视频剪辑； - LiveStreamClip：直播流剪辑； - LiveStreamRecord：直播流录制。 示例值：Upload
Owner	Entity	媒体归属。
ClassPath	String	媒体分类路径。 示例值：/媒体
TaskId	String	生成文件的任务 Id。当生成新文件是拉取上传、视频剪辑、直播流剪辑时为任务 Id。 示例值：125000800-PullUpload-6fdaf790821a36dbe95ec8f557366c71t
SourceContext	String	来源上下文信息。视频剪辑生成新文件时此字段为项目 Id；直播流剪辑或者直播流录制生成新文件则为原始流地址。 示例值：60097a8a8ea83d00017b484b

StreamConnectInputInterruptInfo

云转推输入断流信息。

被如下接口引用：ParseEvent。

名称	类型	描述
EndPoint	String	云转推输入源标识，取值有： - Main：主源； - Backup：备源。 示例值：Main

StreamConnectOutput

云转推输出源。

被如下接口引用：CreateProject, DescribeProjects, HandleStreamConnectProject。

名称	类型	必选	描述
Id	String	否	云转推输出源标识，转推项目级别唯一。若不填则由后端生成。 示例值：135ed54d55d

名称	类型	必选	描述
Name	String	否	云转推输出源名称。 示例值：MyStreamName
Type	String	否	云转推输出源类型，取值： • URL：URL类型 不填默认为URL类型。 示例值：URL
PushUrl	String	否	云转推推流地址。 示例值：rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600c

StreamConnectOutputInfo

云转推输出源信息，包含输出源和输出源转推状态。

被如下接口引用：DescribeProjects。

名称	类型	描述
StreamConnectOutput	StreamConnectOutput	输出源。 注意：此字段可能返回 null，表示取不到有效值。
PushSwitch	String	输出流状态： • On：开； • Off：关。 示例值：On

StreamConnectOutputInterruptInfo

云转推输出断流信息

被如下接口引用：ParseEvent。

名称	类型	描述
Id	String	云转推输出标识。 示例值：outputId
Name	String	云转推输出名称。 示例值：MyStreamName
Url	String	云转推输出地址。 示例值：rtmp://livepush.video-studio.myqcloud.com/output/1250000001-600c

StreamConnectProjectInfo

云转推项目信息，包含输入源、输出源、当前转推开始时间等信息。

被如下接口引用：DescribeProjects。

名称	类型	描述
Status	String	转推项目状态，取值有： • Working：转推中； • Idle：空闲中。 示例值：Working
CurrentInputEndpoint	String	当前转推输入源，取值有： • Main：主输入源； • Backup：备输入源。 示例值：Main

名称	类型	描述
CurrentStartTime	String	当前转推开始时间，采用 ISO 日期格式。仅 Status 取值 Working 时有效。 示例值：2021-03-23T07:51:18.029Z
CurrentStopTime	String	当前转推计划结束时间，采用 ISO 日期格式。仅 Status 取值 Working 时有效。 示例值：2021-03-23T08:51:18.029Z
LastStopTime	String	上一次转推结束时间，采用 ISO 日期格式。仅 Status 取值 Idle 时有效。 示例值：2021-03-22T08:51:18.029Z
MainInput	StreamInputInfo	云转推主输入源。 注意：此字段可能返回 null，表示取不到有效值。
BackupInput	StreamInputInfo	云转推备输入源。 注意：此字段可能返回 null，表示取不到有效值。
OutputSet	Array of StreamConnectOutputInfo	云转推输出源。

StreamConnectProjectInput

云转推项目输入信息。

被如下接口引用：CreateProject。

名称	类型	必选	描述
MainInput	StreamInputInfo	否	云转推主输入源信息。
BackupInput	StreamInputInfo	否	云转推备输入源信息。
Outputs	Array of StreamConnectOutput	否	云转推输出源信息。

StreamInputInfo

输入流信息。

被如下接口引用：CreateProject, DescribeProjects, HandleStreamConnectProject。

名称	类型	必选	描述
InputType	String	是	流输入类型，取值： - VodPull：点播拉流； - LivePull：直播拉流； - RtmpPush：直播推流。 示例值：LivePull
VodPullInputInfo	VodPullInputInfo	否	点播拉流信息，当 InputType = VodPull 时必填。 注意：此字段可能返回 null，表示取不到有效值。
LivePullInputInfo	LivePullInputInfo	否	直播拉流信息，当 InputType = LivePull 时必填。 注意：此字段可能返回 null，表示取不到有效值。
RtmpPushInputInfo	RtmpPushInputInfo	否	直播推流信息，当 InputType = RtmpPush 时必填。 注意：此字段可能返回 null，表示取不到有效值。

SwitcherPgmOutputConfig

导播台主监输出配置信息

被如下接口引用：CreateProject。

名称	类型	必选	描述
TemplateId	Integer	否	导播台输出模板 ID，可取值： 10001：分辨率为1080 P； 10002：分辨率为720 P； 10003：分辨率为480 P。

名称	类型	必选	描述
			示例值：10001
Width	Integer	否	导播台输出宽，单位：像素。 示例值：960
Height	Integer	否	导播台输出高，单位：像素。 示例值：540
Fps	Integer	否	导播台输出帧率，单位：帧/秒 示例值：25
BitRate	Integer	否	导播台输出码率，单位：bit/s。 示例值：64000

SwitcherProjectInput

导播台项目输入信息

被如下接口引用：CreateProject。

名称	类型	必选	描述
StopTime	String	否	导播台停止时间，格式按照 ISO 8601 标准表示。若不填，该值默认为当前时间加七天。 示例值：2024-01-01T12:34:56+00:00
PgmOutputConfig	SwitcherPgmOutputConfig	否	导播台主监输出配置信息。若不填，默认输出 720P。

TaskBaseInfo

任务基础信息。

被如下接口引用：DescribeTasks。

名称	类型	描述
TaskId	String	任务 Id。 示例值：1000000009-procedurev2-b718fcee09a72470bf665cffff
TaskType	String	任务类型，取值有： • VIDEO_EDIT_PROJECT_EXPORT：项目导出。 示例值：VIDEO_EDIT_PROJECT_EXPORT
Status	String	任务状态，取值有： • PROCESSING：处理中； • SUCCESS：成功； • FAIL：失败。 示例值：SUCCESS
Progress	Integer	任务进度，取值为：0~100。 示例值：100
ErrCode	Integer	错误码。 • 0：成功； • 其他值：失败。 示例值：0
ErrMsg	String	错误信息。 示例值：SUCCESS
CreateTime	String	创建时间，格式按照 ISO 8601 标准表示。 示例值：2024-04-30T03:26:37Z

TeamInfo

团队信息

被如下接口引用：DescribeTeams。

名称	类型	描述
TeamId	String	团队 ID。 示例值：aeebb1372a0472e775047027e1f35e3f
Name	String	团队名称。 示例值：分队
MemberCount	Integer	团队成员个数 示例值：5
CreateTime	String	团队创建时间，格式按照 ISO 8601 标准表示。 示例值：2024-01-01T12:34:56+00:00
UpdateTime	String	团队最后更新时间，格式按照 ISO 8601 标准表示。 示例值：2024-02-01T12:34:56+00:00

TeamMemberInfo

团队成员信息

被如下接口引用：DescribeTeamMembers。

名称	类型	必选	描述
MemberId	String	是	团队成员 ID。 示例值：59198a95d991f120005
Remark	String	否	团队成员备注。 示例值：成员
Role	String	否	团队成员角色，取值： - Owner：团队所有者，添加团队成员及修改团队成员解决时不能填此角色； - Admin：团队管理员； - Member：普通成员。 示例值：Member

TextReplacementInfo

模板插槽文本替换信息。

被如下接口引用：ExportVideoByTemplate。

名称	类型	必选	描述
Text	String	是	替换的文本信息。 示例值：文字

TextSlotInfo

文本类型卡槽信息。

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
Text	String	文本内容。 示例值：智能创作

ThirdPartyPublishInfo

第三方平台视频发布信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
ChannelMaterialId	String	是	发布通道 ID。 示例值：9d64a35169404c4a250d
PenguinMediaPlatformPublishInfo	PenguinMediaPlatformPublishInfo	否	企鹅号发布信息，如果使用的发布通道为企鹅号时必填。
WeiboPublishInfo	WeiboPublishInfo	否	新浪微博发布信息，如果使用的发布通道为新浪微博时必填。
KuaishouPublishInfo	KuaishouPublishInfo	否	快手发布信息，如果使用的发布通道为快手时必填。
CosPublishInfo	CosPublishInputInfo	否	腾讯云对象存储发布信息，如果使用的发布通道为腾讯云对象存储时必填。

TimeRange

时间范围

被如下接口引用：SearchMaterial。

名称	类型	必选	描述
StartTime	String	是	开始时间，使用 ISO 日期格式 。 示例值：2018-10-01T18:00:00+08:00
EndTime	String	是	结束时间，使用 ISO 日期格式 。 示例值：2018-11-01T18:00:00+08:00

VODExportInfo

云点播导出信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
Name	String	否	导出的媒资名称。 示例值：MaterialName
ClassId	Integer	否	导出的媒资分类 Id。 示例值：9
ThirdPartyPublishInfos	Array of ThirdPartyPublishInfo	否	第三方平台发布信息列表。暂未正式对外，请勿使用。

VideoEditProjectInput

视频编辑项目输入参数

被如下接口引用：CreateProject。

名称	类型	必选	描述
AspectRatio	String	否	画布宽高比，取值有： 16:9； 9:16； 2:1。 默认值 16:9 。 示例值：16:9
VideoEditTemplateId	String	否	视频编辑模板媒体 ID，通过模板媒体导入项目轨道数据时填写。 示例值：36f19218d1c35ae5
InitTracks	Array of MediaTrack	否	输入的媒体轨道列表，包括视频、音频，等媒体组成的多个轨道信息。其中： - 输入的多个轨道在时间轴上和输出媒体文件的时间轴对齐； - 时间轴上相同时间点的各个轨道的素材进行重叠，视频或者图片按轨道顺序进行图像的叠加，轨道顺序高的素材叠加在上面，音频素材进行混音；

名称	类型	必选	描述
			视频、音频，每一种类型的轨道最多支持10个。 注：当从模板导入项目时（即 VideoEditTemplateId 不为空时），该参数无效。

VideoEditProjectOutput

项目导出信息。

被如下接口引用：DescribeTaskDetail, ParseEvent。

名称	类型	描述
MaterialId	String	导出的多媒体创作引擎媒体 Id，仅当导出目标为多媒体创作引擎媒体时有效。 示例值：5f73f8ad54953bce323f
VodFileId	String	云点播媒资 FileId。 示例值：2c65f425b86ad532
URL	String	导出的媒资 URL。 示例值：http://cme.vod2.myqcloud.com/123/f0.mp4
MetaData	MediaMetaData	元信息。 注意：此字段可能返回 null，表示取不到有效值。
CoverURL	String	导出视频的封面图片 URL。 示例值：http://cme.vod2.myqcloud.com/123/f0.jpg

VideoEditTemplateMaterial

视频编辑模板素材信息。

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
AspectRatio	String	视频编辑模板宽高比。 示例值：16:9
SlotSet	Array of SlotInfo	卡槽信息。
PreviewVideoUrl	String	模板预览视频 URL 地址。 示例值：http://cme.vod2.myqcloud.com/123/f0.mp4

VideoEncodingPreset

视频编码配置

被如下接口引用：DescribeVideoEncodingPresets。

名称	类型	描述
Id	Integer	配置 ID。 示例值：10
Name	String	配置名。 示例值：默认编码
Container	String	封装格式，可选值： - mp4； - mov。 示例值：mp4
RemoveVideo	Integer	是否去除视频数据，可选值： 0：保留； 1：去除。

名称	类型	描述
		默认值：0。 示例值：0
RemoveAudio	Integer	是否去除音频数据，可选值： 0：保留； 1：去除。 默认值：0。 示例值：0
VideoSetting	VideoEncodingPresetVideoSetting	视频编码配置中的视频设置。
AudioSetting	VideoEncodingPresetAudioSetting	视频编码配置中的音频设置。

VideoEncodingPresetAudioSetting

视频编码配置中的音频设置

被如下接口引用：CreateVideoEncodingPreset, DescribeVideoEncodingPresets。

名称	类型	必选	描述
Codec	String	否	音频流的编码格式，可选值： AAC：AAC 编码。 默认值：AAC。 示例值：AAC
Bitrate	Integer	否	音频码率，单位：bps。 默认值：64K。 示例值：0
Channels	Integer	否	音频声道数，可选值： 1：单声道； 2：双声道。 默认值：2。 示例值：2
SampleRate	Integer	否	音频流的采样率，仅支持 16000；32000；44100；48000。单位：Hz。 默认值：16000。 示例值：16000

VideoEncodingPresetAudioSettingForUpdate

视频编码配置中的音频设置更新信息

被如下接口引用：ModifyVideoEncodingPreset。

名称	类型	必选	描述
Bitrate	String	否	音频码率，单位：bps。 不填则不修改。 示例值：0
Channels	Integer	否	音频声道数，可选值： 1：单声道； 2：双声道。 不填则不修改。 示例值：2
SampleRate	Integer	否	音频流的采样率，目前仅支持：16000；32000；44100；48000。单位：Hz。 不填则不修改。 示例值：16000

VideoEncodingPresetVideoSetting

视频编码配置中的视频设置信息

被如下接口引用：CreateVideoEncodingPreset, DescribeVideoEncodingPresets。

名称	类型	必选	描述
Codec	String	否	视频流的编码格式，可选值： H264：H.264 编码。 示例值：H264
ShortEdge	Integer	否	视频短边尺寸，取值范围：[128, 4096]，单位：px。 视频最后的分辨率，根据短边尺寸和宽高比进行计算。 例：如果项目的宽高比是 16：9： - 短边尺寸为 1080，则导出视频的分辨率为 1920 * 1080。 - 短边尺寸为 720，则导出视频的分辨率为 1280 * 720。 如果项目的宽高比是 9：16： - 短边尺寸为 1080，则导出视频的分辨率为 1080 * 1920。 - 短边尺寸为 720，则导出视频的分辨率为 720 * 1280。 默认值：1080。 示例值：1080
Bitrate	Integer	否	指定码率，单位 bps。当该参数为'0'时则不强制限定码率。 默认值：0。 示例值：0

VideoEncodingPresetVideoSettingForUpdate

视频编码配置的视频设置更新信息

被如下接口引用：ModifyVideoEncodingPreset。

名称	类型	必选	描述
ShortEdge	Integer	否	视频短边尺寸，取值范围：[128, 4096]，单位：px。 视频最后的分辨率，根据短边尺寸和宽高比进行计算。 例：如果项目的宽高比是 16：9： - 短边尺寸为 1080，则导出视频的分辨率为 1920 * 1080。 - 短边尺寸为 720，则导出视频的分辨率为 1280 * 720。 如果项目的宽高比是 9：16： - 短边尺寸为 1080，则导出视频的分辨率为 1080 * 1920。 - 短边尺寸为 720，则导出视频的分辨率为 720 * 1280。 不填则不修改。 示例值：1080
Bitrate	Integer	否	指定码率，单位 bps。当该参数为'0' 时则不强制限定码率。 不填则不修改。 示例值：0
FrameRate	Float	否	指定帧率。单位 Hz。 不填则不修改。 示例值：25

VideoExportCompletedEvent

视频导出完成事件。

被如下接口引用：ParseEvent。

名称	类型	描述
TaskId	String	任务 Id。 示例值：123-FusionCME-391d2d669d70952a99e3be240e0acf11t

名称	类型	描述
Status	String	任务状态，取值有： - SUCCESS：成功； - FAIL：失败。 示例值：SUCCESS
ErrCode	Integer	错误码，取值有： 0：成功； - 其他值：失败。 示例值：0
ErrMsg	String	错误信息。 示例值：fail
Output	VideoEditProjectOutput	任务输出。 注意：此字段可能返回 null，表示取不到有效值。

VideoExportExtensionArgs

视频导出扩展参数

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoEditProject。

名称	类型	必选	描述
Container	String	否	封装格式，可选值： - mp4 - mov 不填则使用视频导出编码配置。 示例值：mp4
ShortEdge	Integer	否	视频短边尺寸，取值范围：[128, 4096]，单位：px。 视频最后的分辨率，根据短边尺寸和宽高比进行计算。 例如：项目的宽高比是 16：9： - 短边尺寸为 1080，则导出视频的分辨率为 1920 * 1080。 - 短边尺寸为 720，则导出视频的分辨率为 1280 * 720 不填则使用视频导出编码配置。 示例值：1080
VideoBitrate	Integer	否	指定码率，单位 bps。当该参数为 0 时则不强制限定码率。 不填则使用视频导出编码配置。 示例值：0
FrameRate	Float	否	帧率。取值范围：[15, 60]，不填默认值为 25。 示例值：0
RemoveVideo	Integer	否	是否去除视频数据，可选值： 0：保留； 1：去除。 不填则使用视频导出编码配置。 示例值：0
RemoveAudio	Integer	否	是否去除音频数据，可选值： 0：保留； 1：去除。 不填则使用视频导出编码配置。 示例值：0
StartTime	Integer	否	片段起始时间，单位：毫秒。 示例值：0
EndTime	Integer	否	片段结束时间，单位：毫秒。

名称	类型	必选	描述
			示例值：0

VideoMaterial

视频素材信息

被如下接口引用：DescribeMaterials, FlattenListMedia, ListMedia, SearchMaterial。

名称	类型	描述
MetaData	MediaMetaData	素材元信息。
ImageSpriteInfo	MediaImageSpriteInfo	雪碧图信息。
MaterialUrl	String	素材媒体文件的播放 URL 地址。 示例值：http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.mp4
CoverUrl	String	素材媒体文件的封面图片地址。 示例值：http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.jpg
Resolution	String	媒体文件分辨率。取值为：LD/SD/HD/FHD/2K/4K。 示例值：HD
MaterialStatus	MaterialStatus	素材状态。 注意：此字段可能返回 null，表示取不到有效值。
OriginalUrl	String	素材媒体文件的原始 URL 地址。 示例值：http://cmedemo.vod2.myqcloud.com/xxxxx/cemdemo.mp4
VodFileId	String	云点播媒资 FileId。 示例值：15e83cf8b48b147

VideoSegmentationProjectInput

视频拆条项目的输入信息。

被如下接口引用：CreateProject。

名称	类型	必选	描述
AspectRatio	String	否	画布宽高比，取值有： 16:9； 9:16； 2:1。 默认值 16:9 。 示例值：16:9
ProcessModel	String	否	视频拆条处理模型，不填则默认为手工分割视频。取值： - AI.GameHighlights.PUBG：和平精英集锦； - AI.GameHighlights.Honor OfKings：王者荣耀集锦； - AI.SportHighlights.Football：足球集锦 - AI.SportHighlights.Basketball：篮球集锦； - AI.PersonSegmentation：人物集锦； - AI.NewsSegmentation：新闻拆条。 示例值：AI.NewsSegmentation

VideoStreamInfo

视频流信息。

被如下接口引用：DescribeTaskDetail。

名称	类型	描述
Bitrate	Integer	码率，单位：bps。 示例值：64000
Height	Integer	高度，单位：px。 示例值：540
Width	Integer	宽度，单位：px。 示例值：960
Codec	String	编码格式。 示例值：h264
Fps	Integer	帧率，单位：hz。 示例值：24

VideoTrackItem

视频轨的视频片段信息。

被如下接口引用：CreateProject。

名称	类型	必选	描述
SourceType	String	是	视频媒体来源类型，取值有： - VOD：媒体来源于云点播文件。 - CME：视频来源制作云媒体文件。 - EXTERNAL：视频来源于媒资绑定，如果媒体不是存储在腾讯云点播中或者云创中，都需要使用媒资绑定。 示例值：VOD
SourceMedia	String	是	视频媒体，可取值为： - 当 SourceType 为 VOD 时，参数填云点播 FileId； - 当 SourceType 为 CME 时，参数填多媒体创作引擎媒体 Id； - 当 SourceType 为 EXTERNAL 时，目前仅支持外部媒体 URL(如 https://www.example.com/a.mp4)，参数填写规则请参见注意事项。 注意： - 当 SourceType 为 EXTERNAL 并且媒体 URL Scheme 为 https 时(如：https://www.example.com/a.mp4)，参数为：1000000:www.example.com/a.mp4。 - 当 SourceType 为 EXTERNAL 并且媒体 URL Scheme 为 http 时(如：http://www.example.com/b.mp4)，参数为：1000001:www.example.com/b.mp4。 示例值：60097a8a8ea83d00017b484b
SourceMediaStartTime	Float	否	视频片段取自媒体文件的起始时间，单位为秒。默认为0。 示例值：0
Duration	Float	否	视频片段时长，单位为秒。默认取视频媒体文件本身长度，表示截取全部媒体文件。如果源文件是图片，Duration 需要大于0。 示例值：20
XPos	String	否	视频片段原点距离画布原点的水平位置。支持 %、px 两种格式： - 当字符串以 % 结尾，表示视频片段 XPos 为画布宽度指定百分比的位置，如 10% 表示 XPos 为画布口宽度的 10%。 - 当字符串以 px 结尾，表示视频片段 XPos 单位为像素，如 100px 表示 XPos 为100像素。 默认值：0px。 示例值：10px

名称	类型	必选	描述
YPos	String	否	视频片段原点距离画布原点的垂直位置。支持 %、px 两种格式： - 当字符串以 % 结尾，表示视频片段 YPos 为画布高度指定百分比的位置，如 10% 表示 YPos 为画布高度的 10%。 - 当字符串以 px 结尾，表示视频片段 YPos 单位为像素，如 100px 表示 YPos 为100像素。 默认值：0px。 示例值：20px
CoordinateOrigin	String	否	视频原点位置，取值有： Center：坐标原点为中心位置，如画布中心。 默认值：Center。 示例值：Center
Height	String	否	视频片段的高度。支持 %、px 两种格式： - 当字符串以 % 结尾，表示视频片段 Height 为画布高度的百分比大小，如 10% 表示 Height 为画布高度的 10%； - 当字符串以 px 结尾，表示视频片段 Height 单位为像素，如 100px 表示 Height 为100像素； - 当 Width、Height 均为空，则 Width 和 Height 取视频媒体文件本身的 Width、Height； - 当 Width 为空，Height 非空，则 Width 按比例缩放； - 当 Width 非空，Height 为空，则 Height 按比例缩放。 示例值：1080px
Width	String	否	视频片段的宽度。支持 %、px 两种格式： - 当字符串以 % 结尾，表示视频片段 Width 为画布宽度的百分比大小，如 10% 表示 Width 为画布宽度的 10%； - 当字符串以 px 结尾，表示视频片段 Width 单位为像素，如 100px 表示 Width 为100像素； - 当 Width、Height 均为空，则 Width 和 Height 取视频媒体文件本身的 Width、Height； - 当 Width 为空，Height 非空，则 Width 按比例缩放； - 当 Width 非空，Height 为空，则 Height 按比例缩放。 示例值：1920px

VodPullInputInfo

点播拉流信息，包括输入拉流地址和播放次数。

被如下接口引用：CreateProject, HandleStreamConnectProject。

名称	类型	必选	描述
InputUrls	Array of String	是	点播输入拉流 URL 。 示例值：["http://example.com/file1.mp4"]
LoopTimes	Integer	否	播放次数，取值有： -1：循环播放，直到转推结束； 0：不循环； - 大于0：具体循环次数，次数和时间以先结束的为准。 默认不循环。 示例值：-1

VodPullInputPlayInfo

点播文件播放信息，包含当前在播地址和该地址已播时长。

被如下接口引用：HandleStreamConnectProject。

名称	类型	描述
Url	String	当前正在播放文件 Url 。 示例值：http://playdomain/patch/file.m3u8
TimeOffset	Float	点播文件已播放时长，单位：秒。 示例值：10

WeiboPublishInfo

微博发布信息。

被如下接口引用：ExportVideoByEditorTrackData, ExportVideoByTemplate, ExportVideoByVideoSegmentationData, ExportVideoEditProject。

名称	类型	必选	描述
Title	String	否	视频发布标题。 示例值：视频能力
Description	String	否	视频发布描述信息。 示例值：视频能力描述
Visible	String	否	微博可见性，可取值为： - Public：公开，所有人可见； - Private：私有，仅自己可见。 默认为 Public，所有人可见。 示例值：Public

错误码

最近更新时间：2025-05-27 01:20:44

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。
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。

错误码	说明
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.CreateRecordTask	创建录制任务失败。
FailedOperation.InvalidVodUser	云点播用户无效。
FailedOperation.RecordNotSupport	直播录制功能暂未对使用腾讯云点播存储的平台开放。
FailedOperation.StartStreamConnectPartlyErr	转推部分失败。
FailedOperation.StreamConnect	转推操作失败。
FailedOperation.SwitcherOnWorking	非法操作，导播中正在工作中。
InternalError.CreateTask	发起预处理任务失败。
InvalidParameter.Id	Id 无效。
InvalidParameter.Platform	平台参数非法。
InvalidParameterValue.AspectRatio	画布宽高比不合法。
InvalidParameterValue.AspectRatioSet	画布宽高比不合法。
InvalidParameterValue.Category	项目类别不合法。
InvalidParameterValue.CategorySet	项目类别不合法。
InvalidParameterValue.ClassExist	分类已存在。
InvalidParameterValue.ClassNotEmpty	分类不为空。

错误码	说明
InvalidParameterValue.ClassNotExist	分类不存在。
InvalidParameterValue.ClassPath	分类路径不合法。
InvalidParameterValue.DataNotFoundInDB	查询不到数据。
InvalidParameterValue.Definition	模板 ID 无效。
InvalidParameterValue.DstClassPathNotExist	目标分类不存在。
InvalidParameterValue.EndTime	结束时间参数不合法。
InvalidParameterValue.ExportDestination	导出目标不合法。
InvalidParameterValue.ExtInfoInvalid	拓展信息不合法。
InvalidParameterValue.ExternalMediaInfoNotExist	原始媒资信息不存在。
InvalidParameterValue.Input	转推输入流参数非法。
InvalidParameterValue.Limit	返回记录条数不合法。
InvalidParameterValue.MaterialId	媒体 ID 无效。
InvalidParameterValue.MediaCastSourceInfosInvalid	输入源信息非法。
InvalidParameterValue.MediaReplacementInfo	媒体替换信息无效。
InvalidParameterValue.MemberIds	无效的成员列表。
InvalidParameterValue.MemberNotExist	团队成员不存在。
InvalidParameterValue.Name	名称字段编辑错误，需使用 UTF8 编码。
InvalidParameterValue.NameLenLimit	名称字段长度超长，限制 30 个中文字符。
InvalidParameterValue.NotTeamMemberError	非团队成员。
InvalidParameterValue.Offset	分页偏移不合法。
InvalidParameterValue.OperationInvalid	转推操作非法。
InvalidParameterValue.Operator	无效的操作者。
InvalidParameterValue.OwnerId	归属者 ID 不合法。
InvalidParameterValue.OwnerRemark	团队 Owner 的备注不合法。
InvalidParameterValue.OwnerType	归属者类型不合法。
InvalidParameterValue.Platform	平台参数取值错误。
InvalidParameterValue.PreProcessDefinition	预处理模板 ID 无效。
InvalidParameterValue.ProjectId	项目不存在。
InvalidParameterValue.ReplacementType	轨道素材替换类型无效或不匹配。
InvalidParameterValue.Role	角色不合法。
InvalidParameterValue.SortOrder	排序方式无效。
InvalidParameterValue.StreamConnect	云转推信息非法。
InvalidParameterValue.StreamConnectInputInvalid	转推输入源信息非法。
InvalidParameterValue.StreamConnectOutputInvalid	转推输出源信息非法。
InvalidParameterValue.StreamInput	输入流参数错误。
InvalidParameterValue.SwitcherProjectInput	导播台项目输入信息无效。
InvalidParameterValue.TaskId	任务 Id 无效。

错误码	说明
InvalidParameterValue.TeamId	团队 ID 已经存在，不能重复创建。
InvalidParameterValue.TeamNotExist	团队不存在。
InvalidParameterValue.ThirdyPartyPublishChannelId	发布通道 ID 无效。
InvalidParameterValue.TrackData	轨道数据无效。
InvalidParameterValue.TrackItem	轨道元素无效。
InvalidParameterValue.VideoEditTemplateIdNotExist	视频编辑模板不存在。
InvalidParameterValue.VodFileId	参数缺少 VodFile。
InvalidParameterValue.VodFileNotExist	点播文件不存在。
InvalidParameterValue.VodSubAppid	绑定的点播子应用无效。
LimitExceeded.BillItemLiveDispatchDuration	您已超过推流时长限制，若还需使用，请升级会员版本或购买资源包。
LimitExceeded.BillItemLiveDispatchMaxCount	您已达到转推并发数上限，若还需使用，请升级会员版本或购买资源包。
LimitExceeded.BillItemStorage	存储空间不足，请查看会员权益，购买资源包或会员版本后继续使用。
LimitExceeded.BillItemVideoEditExportDuration	您已达到导出视频时长上限，若还需使用，请升级会员版本或购买资源包。
OperationDenied	操作被拒绝。
OperationDenied.PermissionDeny	权限不足，请检查对应操作者的权限。
ResourceNotFound.Platform	平台不存在。
ResourceNotFound.PlatformNotFound	平台不存在。