

正版曲库直通车

API 文档



腾讯云

【 版权声明 】

© 2013–2023 腾讯云版权所有

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

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

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

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

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

文档目录

API 文档

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

云音乐相关接口

获取授权项目列表

获取授权项目已购云音乐列表

获取云音乐播放信息

曲库包相关接口

获取已购曲库包列表

获取曲库包已核销歌曲列表

获取曲库包歌曲播放信息

曲库包获取已核销歌曲回退数据

获取歌词信息

根据歌曲ID查询歌曲信息

数据上报相关接口

上报歌曲播放数据

直播互动曲库相关接口

获取直播互动曲库歌曲排行榜

获取直播互动曲库标签信息

搜索直播互动曲库歌曲

获取直播互动曲库联想词

获取直播互动曲目详情

批量获取直播互动曲目详情

直播互动曲库歌单相关接口

获取直播互动曲库推荐歌单列表

获取直播互动曲库推荐歌单详情

直播互动歌手相关接口

获取直播互动曲库歌手分类信息

获取直播互动曲库歌手

获取歌手下歌曲列表

直播互动机器人相关接口

创建直播互动机器人

获取直播互动机器人信息

销毁直播互动机器人

同步直播互动机器人指令

其他接口

获取分类内容（Station）列表

获取分类内容下歌曲列表

查询歌曲在售状态

数据结构

错误码

API 文档

更新历史

最近更新时间：2023-05-26 01:01:58

第 36 次发布

发布时间：2023-05-26 01:01:42

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [TRTCJoinRoomInput](#)
 - 新增成员：RoomIdType

第 35 次发布

发布时间：2023-02-23 01:01:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SyncKTVRobotCommand](#)
 - 新增入参：SetRealVolumeCommandInput

新增数据结构：

- [SetRealVolumeCommandInput](#)

修改数据结构：

- [KTVRobotInfo](#)
 - 新增成员：SetRealVolumeInput
- [SyncRobotCommand](#)
 - 新增成员：SetRealVolumeCommandInput

第 34 次发布

发布时间：2023-02-20 01:06:05

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SetPlaylistCommandInput](#)
 - 新增成员：MusicURLs
- [TRTCJoinRoomInput](#)
 - 新增成员：PrivateMapKey, Role

第 33 次发布

发布时间：2022-04-15 06:01:25

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateKTVRobot](#)
 - 新增入参: SyncRobotCommands

新增数据结构:

- [SyncRobotCommand](#)

第 32 次发布

发布时间: 2022-04-04 06:03:13

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [DescribeKTVPlaylists](#)
 - 新增入参: Type

第 31 次发布

发布时间: 2022-03-21 08:06:03

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [SyncKTVRobotCommand](#)
 - 新增入参: SetVolumeCommandInput

新增数据结构:

- [SetVolumeCommandInput](#)

修改数据结构:

- [KTVRobotInfo](#)
 - 新增成员: SetVolumeInput

第 30 次发布

发布时间: 2022-03-10 08:03:53

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [DescribeKTVMusicTags](#)
- [DescribeKTVSuggestions](#)

修改接口:

- [CreateKTVRobot](#)
 - 新增入参: ApplicationLicenseInput
- [SearchKTVMusics](#)
 - 新增入参: TagIds

新增数据结构:

- [ApplicationLicenseInput](#)
- [KTVMusicTagGroup](#)
- [KTVMusicTagInfo](#)
- [KTVSuggestionInfo](#)

第 29 次发布

发布时间：2022-02-28 08:05:12

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SetPlaylistCommandInput](#)
 - 新增成员：ChangedIndex
 - 修改成员：Index

第 28 次发布

发布时间：2022-02-25 08:03:09

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeKTVMusicDetail](#)
 - 新增出参：PreludeInterval

修改数据结构：

- [KTVMusicDetailInfo](#)
 - 新增成员：PreludeInterval

第 27 次发布

发布时间：2022-02-23 08:01:52

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [BatchDescribeKTVMusicDetails](#)

新增数据结构：

- [KTVMusicDetailInfo](#)

第 26 次发布

发布时间：2022-02-15 08:02:33

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SyncKTVRobotCommand](#)
 - 新增入参：SetDestroyModeCommandInput

新增数据结构：

- [SetDestroyModeCommandInput](#)

第 25 次发布

发布时间：2022-02-14 08:04:52

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SearchKTMusics](#)
 - 新增入参：Sort

第 24 次发布

发布时间：2022-01-28 08:00:20

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeKTVTopList](#)

新增数据结构：

- [KTMusicTopInfo](#)

第 23 次发布

发布时间：2022-01-27 08:04:32

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeKTVSingerCategories](#)
- [DescribeKTVSingerMusics](#)
- [DescribeKTVSingers](#)

修改接口：

- [DescribeKTMusicDetail](#)
 - 新增出参：MidiJsonUrl, ChorusClipSet

新增数据结构：

- [ChorusClip](#)
- [KTVSingerBaseInfo](#)
- [KTVSingerCategoryInfo](#)
- [KTVSingerInfo](#)
- [SortBy](#)

修改数据结构：

- [KTMusicBaseInfo](#)
 - 新增成员：SingerInfoSet

第 22 次发布

发布时间：2022-01-26 08:02:19

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeKTVRobots](#)
 - 新增入参：CreateTime

新增数据结构：

- [TimeRange](#)

第 21 次发布

发布时间：2022-01-24 08:04:15

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SyncKTVRobotCommand](#)
 - 新增入参：SetPlayModeCommandInput

新增数据结构：

- [SetPlayModeCommandInput](#)

修改数据结构：

- [KTVRobotInfo](#)
 - 新增成员：SetPlayModelInput

第 20 次发布

发布时间：2022-01-18 08:03:18

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateKTVRobot](#)
- [DescribeKTVRobots](#)
- [DestroyKTVRobot](#)
- [SyncKTVRobotCommand](#)

新增数据结构：

- [JoinRoomInput](#)
- [KTVRobotInfo](#)
- [PlayCommandInput](#)
- [SeekCommandInput](#)
- [SendMessageCommandInput](#)
- [SetAudioParamCommandInput](#)
- [SetPlaylistCommandInput](#)
- [TRTCJoinRoomInput](#)

第 19 次发布

发布时间：2021-12-03 08:01:26

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [PackageItem](#)
 - 新增成员：Tags

第 18 次发布

发布时间：2021-10-13 08:02:42

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeKTVMusicDetail](#)
 - 新增出参：DefinitionInfoSet

新增数据结构：

- [KTVMusicDefinitionInfo](#)

第 17 次发布

发布时间：2021-09-06 08:10:10

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribePkgOfflineMusic](#)

新增数据结构：

- [OfflineMusicDetail](#)

第 16 次发布

发布时间：2021-08-25 08:10:23

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeMusicSaleStatus](#)

新增数据结构：

- [MusicStatus](#)

第 15 次发布

发布时间：2021-08-24 08:11:45

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataInfo](#)
 - 新增成员：TagNames

第 14 次发布

发布时间：2021-08-17 08:10:54

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MusicOpenDetail](#)

- 新增成员：WaveformUrl

第 13 次发布

发布时间：2021-08-16 08:10:16

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeKTVPlaylistDetail](#)
- [DescribeKTVPlaylists](#)

新增数据结构：

- [KTVPlaylistBaseInfo](#)

第 12 次发布

发布时间：2021-07-14 08:10:35

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

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

第 11 次发布

发布时间：2021-06-22 08:11:20

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeKTVMusicDetail](#)
 - 新增出参：LyricsUrl

第 10 次发布

发布时间：2021-05-31 08:10:00

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeKTVMusicDetail](#)
- [SearchKTVMusics](#)

新增数据结构：

- [KTVMusicBaseInfo](#)

第 9 次发布

发布时间：2021-05-28 08:09:40

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [MusicDetailInfo](#)
 - 新增成员：Scene, Region, AuthPeriod, Commercialization, Platform, Channel

第 8 次发布

发布时间：2021-04-27 08:10:58

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyMusicOnShelves](#)
 - 新增入参：AmeKey

第 7 次发布

发布时间：2021-04-14 08:09:38

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ModifyMusicOnShelves](#)
- [PutMusicOnTheShelves](#)
- [TakeMusicOffShelves](#)

新增数据结构：

- [MusicDetailInfo](#)
- [TakeMusicOffShelves](#)

第 6 次发布

发布时间：2021-01-07 08:10:08

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeAuthInfo](#)
- [DescribeCloudMusic](#)
- [DescribeCloudMusicPurchased](#)

新增数据结构：

- [AuthInfo](#)
- [MusicOpenDetail](#)

第 5 次发布

发布时间：2020-08-13 08:22:27

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribePackageItems](#)
- [DescribePackages](#)

新增数据结构：

- [Package](#)
- [PackageItem](#)
- [UseRange](#)

修改数据结构：

- [Item](#)
 - 新增成员：Status

第 4 次发布

发布时间：2020-06-02 08:21:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeItemById](#)

第 3 次发布

发布时间：2020-02-18 14:36:11

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Music](#)
 - 新增成员：FullUrl

第 2 次发布

发布时间：2020-01-15 21:47:50

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ReportData](#)

第 1 次发布

发布时间：2019-12-05 20:17:39

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeItems](#)
- [DescribeLyric](#)
- [DescribeMusic](#)
- [DescribeStations](#)

新增数据结构：

- [Album](#)
- [Artist](#)
- [DataInfo](#)
- [ImagePath](#)

- [Item](#)
- [Lyric](#)
- [Music](#)
- [Station](#)

简介

最近更新时间：2021-06-23 11:38:37

欢迎使用 正版曲库直通车 API 3.0 版本。全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验。支持全地域就近接入让您更快连接腾讯云产品。更多腾讯云 API 3.0 使用介绍请查看：[快速入门](#)

正版曲库直通车（版权音乐助手，Authorized Music Express，AME）是基于腾讯音乐娱乐集团（TME）海量线上背景音乐专用曲库资源，结合腾讯云存储、内容加速分发等基础能力，为解决内容创作过程中的正版背景音乐素材应用问题设计的 PaaS 产品。开发者在控制台一键域名接入，三步完成 API 调用，即可便捷地实现海量正版背景音乐素材在多端的顺畅播放与应用。

云音乐单曲订购接口调用步骤

- 第一步：调用[获取授权项目列表](#)接口，获取授权项目列表。
- 第二步：调用[获取授权项目已购云音乐列表](#)接口，获取授权项目下已购云音乐列表。
- 第三步：调用[获取云音乐播放信息](#)接口，获取指定歌曲的基础信息及播放链接。

高级/商用版接口调用步骤

- 第一步：调用[查询已购曲库包列表](#)接口，获取已购曲库包信息。
- 第二步：调用[查询曲库包已核验歌曲列表](#)接口，获取曲库包中已选歌曲列表信息。
- 第三步：调用[获取歌曲播放信息](#)接口或[获取歌词信息](#)接口，获取指定歌曲的播放链接或歌词信息。

详细信息，您可以[点此查看](#)。另外您也可以：

- 查看[示例demo源码](#)（go语言）
- 通过[API 3.0 Explorer](#)进行在线调用、签名验证、SDK 代码生成和快速检索接口等能力。
- 通过[腾讯云命令行工具](#)来调用腾讯云 API，管理您的腾讯云资源，基于腾讯云 CLI 来做自动化和脚本处理，从而简化部分复杂 API 的服务。

API 概览

最近更新时间：2023-05-17 01:02:00

云音乐相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeAuthInfo	获取授权项目列表	20
DescribeCloudMusicPurchased	获取授权项目已购云音乐列表	20
DescribeCloudMusic	获取云音乐播放信息	20

曲库包相关接口

接口名称	接口功能	频率限制（次/秒）
DescribePackages	获取已购曲库包列表	20
DescribePackageItems	获取曲库包已核销歌曲列表	20
DescribeMusic	获取曲库包歌曲播放信息	500
DescribePkgOfflineMusic	曲库包获取已核销歌曲回退数据	20
DescribeLyric	获取歌词信息	500
DescribeItemById	根据歌曲ID查询歌曲信息	20

数据上报相关接口

接口名称	接口功能	频率限制（次/秒）
ReportData	上报歌曲播放数据	500

直播互动曲库相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeKTVTopList	获取直播互动曲库歌曲排行榜	20
DescribeKTVMusicTags	获取直播互动曲库标签信息	20
SearchKTVMusics	搜索直播互动曲库歌曲	20
DescribeKTVSuggestions	获取直播互动曲库联想词	20
DescribeKTVMusicDetail	获取直播互动曲目详情	20
BatchDescribeKTVMusicDetails	批量获取直播互动曲目详情	20

直播互动曲库歌单相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeKTVPlaylists	获取直播互动曲库推荐歌单列表	20
DescribeKTVPlaylistDetail	获取直播互动曲库推荐歌单详情	20

直播互动歌手相关接口

接口名称	接口功能	频率限制（次/秒）
------	------	-----------

接口名称	接口功能	频率限制（次/秒）
DescribeKTVSingerCategories	获取直播互动曲库歌手分类信息	20
DescribeKTVSingers	获取直播互动曲库歌手	20
DescribeKTVSingerMusics	获取歌手下歌曲列表	20

直播互动机器人相关接口

接口名称	接口功能	频率限制（次/秒）
CreateKTVRobot	创建直播互动机器人	20
DescribeKTVRobots	获取直播互动机器人信息	20
DestroyKTVRobot	销毁直播互动机器人	20
SyncKTVRobotCommand	同步直播互动机器人指令	20

其他接口

接口名称	接口功能	频率限制（次/秒）
DescribeStations	获取分类内容（Station）列表	500
DescribeItems	获取分类内容下歌曲列表	500
DescribeMusicSaleStatus	查询歌曲在售状态	20
ModifyMusicOnShelves	变更歌曲参数	20
PutMusicOnTheShelves	上架歌曲	20
TakeMusicOffShelves	下架歌曲	20

调用方式

请求结构

最近更新时间：2023-02-22 01:02:25

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 `ame.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `ame.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 `ame.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>ame.tencentcloudapi.com</code>
华南地区(广州)	<code>ame.ap-guangzhou.tencentcloudapi.com</code>
华东地区(上海)	<code>ame.ap-shanghai.tencentcloudapi.com</code>
华北地区(北京)	<code>ame.ap-beijing.tencentcloudapi.com</code>
西南地区(成都)	<code>ame.ap-chengdu.tencentcloudapi.com</code>
西南地区(重庆)	<code>ame.ap-chongqing.tencentcloudapi.com</code>
港澳台地区(中国香港)	<code>ame.ap-hongkong.tencentcloudapi.com</code>
亚太东南(新加坡)	<code>ame.ap-singapore.tencentcloudapi.com</code>
亚太东南(曼谷)	<code>ame.ap-bangkok.tencentcloudapi.com</code>
亚太南部(孟买)	<code>ame.ap-mumbai.tencentcloudapi.com</code>
亚太东北(首尔)	<code>ame.ap-seoul.tencentcloudapi.com</code>
亚太东北(东京)	<code>ame.ap-tokyo.tencentcloudapi.com</code>
美国东部(弗吉尼亚)	<code>ame.na-ashburn.tencentcloudapi.com</code>
美国西部(硅谷)	<code>ame.na-siliconvalley.tencentcloudapi.com</code>
北美地区(多伦多)	<code>ame.na-toronto.tencentcloudapi.com</code>
欧洲地区(法兰克福)	<code>ame.eu-frankfurt.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：2023-07-28 01:06:12

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=AKIDEXAMPLE/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，AKIDEXAMPLE 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 ame；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*****EXAMPLE/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*****EXAMPLE/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*****EXAMPLE/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*****EXAMPLE

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*****EXAMPLE
```

地域列表

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

地域	取值
华北地区（北京）	ap-beijing
华南地区（广州）	ap-guangzhou
华东地区（上海）	ap-shanghai
欧洲地区（法兰克福）	eu-frankfurt

签名方法 v3

最近更新时间：2023-09-05 01:07:43

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

推荐使用 API Explorer

</> 点击调试

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

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

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份
签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 [云 API 密钥](#) 页面获取密钥相关信息。
2. 保护传输中的数据
为了防止请求在传输过程中被篡改，腾讯云 API 会使用请求参数来计算请求的哈希值，并将生成的哈希值加密后作为请求的一部分，发送到腾讯云 API 服务器。服务器会使用收到的请求参数以同样的过程计算哈希值，并验证请求中的哈希值。如果请求被篡改，将导致哈希值不一致，腾讯云 API 将拒绝本次请求。

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

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以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 7 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++](#)。

申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

签名版本 v3 签名过程

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

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

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

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

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3***** 和 Gu5t9xGARNpq86cd98joQYCN3*****。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=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3" \
-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，字符集 UTF-8，推荐使用编程语言标准库，所有特殊字符均需编码，大写形式。
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，请求必定失败，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 = "Gu5t9xGARNpq86cd98joQYCN3*****"
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：
f1cb4d518a0eda9d5cbbfdb7850983f1e603eeae484edea76e4dd8d8deb5556e，
e7c609ce81bea53546bed2cc904778bef9ca14082e48e67883443ed64e227cd7，
8aa8ab5755582f576e94bcfe383b8e29325b0ca90c3590d569221c6a63a091ed。

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 Gu5t9xGARNpq86cd98joQYCN3*****。
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 be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3。

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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****。
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 be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-t
```

```
ype;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3
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"}]}
```

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++](#)

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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    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 Gu5t9xGARNpq86cd98joQYCN3*****
    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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
$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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98JoQYCN3*****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]
```



```
service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

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

puts canonical_request

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

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

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

puts 'curl -X POST ' + endpoint \
```

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

DotNet

```
using System;\nusing System.Collections.Generic;\nusing System.Security.Cryptography;\nusing System.Text;\n\npublic class Application\n{\n    public static string SHA256Hex(string s)\n    {\n        using (SHA256 algo = SHA256.Create())\n        {\n            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));\n            StringBuilder builder = new StringBuilder();\n            for (int i = 0; i < hashbytes.Length; ++i)\n            {\n                builder.Append(hashbytes[i].ToString("x2"));\n            }\n            return builder.ToString();\n        }\n    }\n\n    public static byte[] HmacSHA256(byte[] key, byte[] msg)\n    {\n        using (HMACSHA256 mac = new HMACSHA256(key))\n        {\n            return mac.ComputeHash(msg);\n        }\n    }\n\n    public static Dictionary<String, String> BuildHeaders(string secretid,\n        string secretkey, string service, string endpoint, string region,\n        string action, string version, DateTime date, string requestPayload)\n    {\n        string datestr = date.ToString("yyyy-MM-dd");\n        DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);\n        long requestTimestamp = (long)Math.Round(((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000);\n        // ***** 步骤 1: 拼接规范请求串 *****\n        string algorithm = "TC3-HMAC-SHA256";\n        string httpRequestMethod = "POST";\n        string canonicalUri = "/";\n        string canonicalQueryString = "";\n        string contentType = "application/json";\n        string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\\n"\n            + "host:" + endpoint + "\\n"\n            + "x-tc-action:" + action.ToLower() + "\\n";\n    }\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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

string service = "cvm";
string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
```

```
string version = "2017-03-12";

// 此处由于示例规范的原因，采用时间戳2019-02-26 00:44:25，此参数作为示例，如果在项目中，您应当使用：
// DateTime date = DateTime.UtcNow;
// 注意时区，建议此时间统一采用UTC时间戳，否则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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 buff[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++)  
    {  
        sprintf(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;
#ifdef OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

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

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

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

return (ss.str());
}

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

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

int main()
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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(char* key, const char* input, char* result)
{
    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, strlen(key), EVP_sha256(), NULL);
    HMAC_Update(h, ( unsigned char* )input, strlen(input));
    unsigned int len = 32;
    HMAC_Final(h, hash, &len);
}
```

```
#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif
strncpy(result, (const char*)hash, len);
}

void hex_encode(const char* input, char* output)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = strlen(input);
    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < 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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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, "\nxc-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%s", "TC3", SECRET_KEY);
char k_date[64] = {0};
hmac_sha256(k_key, date, k_date);
char k_service[64] = {0};
hmac_sha256(k_date, service, k_service);
char k_signing[64] = {0};
hmac_sha256(k_service, "tc3_request", k_signing);
char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, string_to_sign, k_hmac_sha_sign);

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, 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;
}
```

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：2023-01-11 02:03:48

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 7 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++](#)。

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: AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3*****

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	AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
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': 'AKIDz8krbsj5yKBZQpn74WFkmLPx3*****',
  '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=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&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-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&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 = 'Gu5t9xGARNpq86cd98joYCN3*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&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：

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

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 zmmjn35mikh6pM3V7sUEuX4wyYM=，最终得到的签名串请求参数（Signature）为：
zmmjn35mikh6pM3V7sUEuX4wyYM%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++](#)

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&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap->

guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
        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 Gu5t9xGARNpq86cd98joQYCN3*****
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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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",
        "InstanceId.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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
$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["InstanceId.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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
```

```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceId.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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    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("InstanceId.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'] = escape(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 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
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['InstanceId.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：2022-07-21 06:01:18

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，可以联系我们，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，可以联系我们，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 06:12:50

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。

云音乐相关接口

获取授权项目列表

最近更新时间：2023-09-05 01:07:42

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取授权项目信息列表

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：DescribeAuthInfo。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
Offset	否	Integer	偏移量：Offset=Offset+Limit
Limit	否	Integer	数据条数
Key	否	String	搜索关键字

3. 输出参数

参数名称	类型	描述
AuthInfo	Array of AuthInfo	授权项目列表
TotalCount	Integer	总数
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 DescribeAuthInfo

获取授权项目列表

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeAuthInfo
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 0,
```

```
"AuthInfo": [
  {
    "ProjectName": "xx",
    "Commercialization": 0,
    "Platform": 0,
    "SubjectName": "xx",
    "AuthPeriod": 0,
    "AppRegion": 0,
    "AppScene": 0,
    "Id": "xx"
  }
],
"RequestId": "xx"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	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。
ResourceInUse	资源被占用。

错误码	描述
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

获取授权项目已购云音乐列表

最近更新时间：2023-09-05 01:07:41

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取授权项目下已购云音乐列表

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：DescribeCloudMusicPurchased。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
AuthInfold	是	String	授权项目Id

3. 输出参数

参数名称	类型	描述
MusicOpenDetail	Array of MusicOpenDetail	云音乐列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 DescribeCloudMusicPurchased

获取授权项目已购云音乐列表

输入示例

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

{
  "AuthInfold": "xx"
}
```

输出示例

```
{
  "Response": {
    "MusicOpenDetail": [
      {
```

```
"MusicName": "xx",
"Tags": [
  "xx"
],
"MusicId": "xx",
"MusicImageUrl": "xx",
"AlbumName": "xx",
"Duration": 1,
"AlbumImageUrl": "xx",
"Singers": [
  "xx"
],
"LyricUrl": "xx"
}
},
"RequestId": "xx"
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的 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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。

错误码	描述
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResourcesSoldOut	资源售罄。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

获取云音乐播放信息

最近更新时间：2023-09-05 01:07:41

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取云音乐播放信息接口

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：DescribeCloudMusic。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
MusicId	是	String	歌曲Id 示例值：xxx
MusicType	否	String	歌曲类型，可选值有： - MP3-128K-FTW：含有水印的试听资源； - MP3-320K-FTD-P：320kbps歌曲热门片段； - MP3-320K-FTD：320kbps已核验歌曲完整资源。 默认为：MP3-128K-FTW 示例值：MP3-320K-FTD-P

3. 输出参数

参数名称	类型	描述
MusicId	String	歌曲Id 示例值：xxx
MusicName	String	歌曲名称 示例值：低温xx
Duration	Integer	歌曲时长 注意：此字段可能返回 null，表示取不到有效值。 示例值：288
MusicUrl	String	歌曲链接 示例值：https://xxx.com/wm/10/384/10384721-MP3-128K-FTW-P.mp3?sign=4d7428e523c0d419e6c05db5912c8c67&t=5fcf112a&transDeliveryCode=CA@0@1607400466@S@c56ea
MusicImageUrl	String	歌曲图片 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://xxx.com/album/097/976740-JPG-240X240-ALBUM.jpg
Singers	Array of String	歌手列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：["\u9648xx"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取云音乐播放信息

获取云音乐播放信息

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeCloudMusic
&MusicId=xxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Duration": 288,
    "MusicId": "xxx",
    "MusicImageUrl": "http://xxx.com/album/097/976740-JPG-240X240-ALBUM.jpg",
    "MusicName": "低温xx",
    "MusicUrl": "https://xxx.com/wm/10/384/10384721-MP3-128K-FTW-P.mp3?sign=4d7428e523c0d419e6c05db5912c8c67&t=5fcf112a&transDeliveryCode=CA@0@1607400466@S@c56ea136420f5044",
    "RequestId": "s1607400465431774000",
    "Singers": [
      "陈xx"
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
-----	----

错误码	描述
FailedOperation	操作失败。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。

曲库包相关接口

获取已购曲库包列表

最近更新时间：2023-09-05 01:07:32

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取已购曲库包列表接口

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：DescribePackages。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
Offset	否	Integer	默认0，Offset=Offset+Length
Length	否	Integer	默认20

3. 输出参数

参数名称	类型	描述
Packages	Array of Package	已购曲库包列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取已购曲库包列表

获取已购曲库包列表

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribePackages
&Offset=0
&Length=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Packages": [
      {
```

```
"AuthorizedArea": "CN",
"AuthorizedLimit": 2000,
"Commercial": 0,
"EffectTime": "2020-07-03 16:05:01",
"ExpireTime": "2021-07-03 16:05:01",
"Name": "商用通用-在线",
"OrderId": "10070316050031100",
"PackagePrice": 200000,
"TermOfValidity": 365,
"UseRanges": [
{
"Name": "API音乐使用场景/短视频背景音（免费）",
"UseRangeId": 32
},
{
"Name": "API音乐使用场景/图文背景音（免费）",
"UseRangeId": 33
},
{
"Name": "API音乐使用场景/应用程序背景音（免费）",
"UseRangeId": 34
},
{
"Name": "API音乐使用场景/网页/个人空间背景音乐定制（付费）",
"UseRangeId": 1005
},
{
"Name": "API音乐使用场景/客户端背景音乐内置（付费）",
"UseRangeId": 1008
},
{
"Name": "API音乐使用场景/视频背景音乐（付费）",
"UseRangeId": 1013
}
],
"UsedCount": 10
},
{
"RequestId": "s1596075862723742000"
}
}
```

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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

获取曲库包已核销歌曲列表

最近更新时间：2023-09-05 01:07:33

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取曲库包下已核销歌曲列表接口

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：DescribePackageItems。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
OrderId	是	String	订单id，从获取已购曲库包列表中获得
Offset	否	Integer	默认0，Offset=Offset+Length
Length	否	Integer	默认20

3. 输出参数

参数名称	类型	描述
PackageItems	Array of PackageItem	已核销歌曲信息列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取曲库包已核验歌曲列表

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribePackageItems
&OrderId=xxx
&Offset=0
&Length=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "PackageItems": [
      {
```

```
"ArtistName": "格子兮,willen",
"AuthorizedArea": "CN",
"Duration": "00:03:13",
"Img": "http://csimage.tingmall.com/album/095/954833-JPG-600X600-ALBUM.jpg",
"ItemId": "EB9DF7B88A3CF5BF5569BB40AF7AEF08",
"OrderId": "00070710331231100",
"TrackName": "十年寒冬"
}
],
"RequestId": "s1596078603574525000"
}
}
```

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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

获取曲库包歌曲播放信息

最近更新时间：2023-09-05 01:07:34

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取曲库包歌曲播放信息接口

默认接口请求频率限制：500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：DescribeMusic。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
ItemId	是	String	歌曲ID 示例值：xxxx
IdentityId	是	String	在应用前端播放音乐C端用户的唯一标识。无需是账户信息，用户唯一标识即可。
SubItemType	否	String	MP3-320K-FTD-P 为获取320kbps歌曲热门片段。 MP3-320K-FTD 为获取320kbps已核验歌曲完整资源。 示例值：MP3-64K-FTD-P
Ssl	否	String	CDN URL Protocol:HTTP or HTTPS/SSL Values:Y , N(default)

3. 输出参数

参数名称	类型	描述
Music	Music	音乐相关信息 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，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取曲库包歌曲播放信息-正常示例

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeMusic
&ItemId=xxxx
&IdentityId=xx
&SubItemType=MP3-64K-FTD-P
&<公共请求参数>
```

输出示例


```
{
  "Response": {
    "Music": {
      "AuditionBegin": 0,
      "AuditionEnd": 0,
      "FileExtension": "mp3",
      "FileSize": 3475953,
      "Url": "/content/09/800/9860965-MP3-64K-FTD.mp3?sign=71a19ae9f06313071f5876fbc4fc5c30&t=5d81f154&transDeliveryCode=YQUFAR@0@1568791612@S@3e760189cf194262",
      "FullUrl": "http://xxx.com/content/09/800/9860965-MP3-64K-FTD.mp3?sign=71a19ae9f06313071f5876fbc4fc5c30&t=5d81f154&transDeliveryCode=YQUFAR@0@1568791612@S@3e760189cf194262"
    },
    "RequestId": "s1568791637852491000"
  }
}
```

示例2 获取曲库包歌曲播放信息-版权方错误示例

在Error为ResourceNotFound时，Message中包含Code信息为版权方的错误码，具体含义如下：

101 无结果
102 内容无权限获取
103 内容因版权原因下架
104 内容仅支持下载，不可在线播放
105 内容需要付费获取
106 部分资源内容状态不同
1001 输入参数错误
1004 第三方API错误
1005 支付错误
1006 非支持支付模式
1007 系统错误
1008 其他系统错误
1009 Token过期
1205 用户账户不存在
5023 超过当日下载最大量
5024 超过当月下载最大量
5000-5999 其他功能错误定义

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeMusic
&ItemId=xxxx
&IdentityId=xx
&SubItemType=MP3-64K-FTD-P
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Error": {
      "Code": "ResourceNotFound",
      "Message": "Code:1001, Msg:Invalid parameter (itemid)"
    },
    "RequestId": "s1575025791424544000"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。

曲库包获取已核销歌曲回退数据

最近更新时间：2023-09-05 01:07:32

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据购买曲库包用户可查询已回退的歌曲信息

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：DescribePkgOfflineMusic。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
PackageOrderId	是	String	订单id
Limit	否	Integer	分页返回的起始偏移量，默认值：0。将返回第 Offset 到第 Offset+Limit-1 条(注：单次上限为100)。
Offset	否	Integer	分页返回的记录条数，默认值：50。将返回第 Offset 到第 Offset+Limit-1 条。

3. 输出参数

参数名称	类型	描述
OfflineMusicSet	Array of OfflineMusicDetail	曲库包中不可用歌曲信息
TotalCount	Integer	返回总量
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 曲库包获取已核销歌曲回退数据

根据购买曲库包用户可查询已回退的歌曲信息

输入示例

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

{
  "PackageOrderId": "xx",
  "Limit": 0,
  "Offset": 0
}
```

输出示例

```
{
  "Response": {
    "OfflineMusicSet": [
      {
        "ItemId": "ABCD",
        "MusicName": "A",
        "OffRemark": "授权到期",
        "OffTime": "2006-01-01 00:00:00"
      }
    ],
    "RequestId": "xxx",
    "TotalCount": 1
  }
}
```

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。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResourcesSoldOut	资源售罄。

错误码	描述
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

获取歌词信息

最近更新时间：2023-09-05 01:07:34

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据接口的模式及歌曲ID来取得歌词信息或者波形图信息。

默认接口请求频率限制：500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：DescribeLyric。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
ItemId	是	String	歌曲ID 示例值：xxxxxx
SubItemType	否	String	格式，可选项，可不填写，默认值为：LRC-LRC。 - LRC-LRC：歌词； - JSON-ST：波形图。 示例值：LRC-LRC

3. 输出参数

参数名称	类型	描述
Lyric	Lyric	歌词或者波形图详情 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，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取歌词信息

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeLyric
&ItemId=xxxxxx
&SubItemType=LRC-LRC
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Lyric": {
      "FileNameExt": "lrc",
      "SubItemType": "LRC-LRC",
```

```
"Url": "http://xxx.com/lyric/09/860/9860965-LRC-LRC.lrc"
},
"RequestId": "s1568793079544475000"
}
}
```

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。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

根据歌曲ID查询歌曲信息

最近更新时间：2023-09-05 01:07:35

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据歌曲ID查询歌曲信息

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：DescribeItemById。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
ItemIDs	是	String	歌曲ID，目前暂不支持批量查询

3. 输出参数

参数名称	类型	描述
Items	Array of Item	歌曲信息 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，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 根据歌曲ID查询歌曲信息

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeItemById
&ItemIDs=xxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Items": [
      {
        "Album": {
          "AlbumName": "你像她",
          "ImagePathMap": [
            {
              "Key": "JPG-240X240-ALBUM",
```



```
"Value": "http://image.tingmall.com/album/172/1727096-JPG-240X240-ALBUM.jpg"
},
{
  "Key": "JPG-320X320-ALBUM",
  "Value": "http://image.tingmall.com/album/172/1727096-JPG-320X320-ALBUM.jpg"
},
{
  "Key": "JPG-600X600-ALBUM",
  "Value": "http://image.tingmall.com/album/172/1727096-JPG-600X600-ALBUM.jpg"
},
{
  "Key": "JPG-1000X1000-ALBUM",
  "Value": "http://image.tingmall.com/album/172/1727096-JPG-1000X1000-ALBUM.jpg"
}
],
},
"Artists": [
  {
    "ArtistName": "炫青"
  },
],
"DataInfo": {
  "AuditionBegin": 31000,
  "AuditionEnd": 58000,
  "Duration": "00:04:25",
  "Name": "你像她",
  "Version": "伴奏"
},
"ItemID": "58E772961AA8DD0D5569BB40AF7AEF08"
},
"RequestId": "s1590115749573388000"
}
```

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。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

数据上报相关接口

上报歌曲播放数据

最近更新时间：2023-09-05 01:07:36

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

客户上报用户数据功能，为了更好地为用户提供优质服务

默认接口请求频率限制：500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：ReportData。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
ReportData	是	String	上报数据 注:reportData为客户端压缩后的上报数据进行16进制转换的字符串数据 压缩说明： a) 上报的json格式字符串通过流的转换（ByteArrayInputStream, java.util.zip.GZIPOutputStream），获取到压缩后的字节数组。 b) 将压缩后的字节数组转成16进制字符串。 reportData由两部分数据组成： 1) report_type（上报类型） 2) data（歌曲上报数据） 不同的report_type对应的data数据结构不一样。 详细说明请参考文档reportdata.docx： https://github.com/tencentyun/ame-documents 示例值：1f8b08000000000000000007xxxxxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 上报歌曲播放数据

输入示例

```
https://ame.tencentcloudapi.com/?Action=ReportData
&ReportData=1f8b08000000000000000007xxxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "s1578300559628971903"
  }
}
```

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。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

直播互动曲库相关接口

获取直播互动曲库歌曲排行榜

最近更新时间：2023-09-05 01:07:27

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取直播互动曲库歌曲的周榜和月榜

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：DescribeKTVTopList。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Type	否	String	榜单类型。默认Hot Hot, 热歌榜。 示例值：Hot
Period	否	String	榜单周期 默认为Week - Week, 周榜。 - Month, 月榜。 示例值：Week

3. 输出参数

参数名称	类型	描述
KTVMusicTopInfoSet	Array of KTVMusicTopInfo	歌曲基础信息列表
TotalCount	Integer	返回总条数 示例值：30
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取直播互动曲库歌曲的周榜和月榜

获取直播互动曲库歌曲的周榜和月榜

输入示例

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

<公共请求参数>

```
{
  "Type": "Hot",
  "Period": "Week"
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "RequestId": "1999xff-11122",
    "KTVMusicTopInfoSet": [
      {
        "ComposerSet": [
          "方文山"
        ],
        "SingerInfoSet": [
          {
            "SingerId": "amesid-98dssf",
            "Name": "周杰伦"
          }
        ],
        "MusicId": "ameid-34dd8d9s",
        "TagSet": [
          "情歌"
        ],
        "Duration": 240,
        "LyricistSet": [
          "方文山"
        ],
        "Name": "七里香"
      }
    ]
  }
}
```

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。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnknownParameter	未知参数错误。

获取直播互动曲库标签信息

最近更新时间：2023-09-05 01:07:28

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取直播互动曲库标签分组信息和标签信息

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：DescribeKTVMusicTags。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
TagGroupSet	Array of KTVMusicTagGroup	标签分组列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取直播互动曲库标签信息

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "TagGroupSet": [
      {
        "EnglishGroupName": "language",
        "ChineseGroupName": "语种",
        "TagSet": [
```



```
{
  "TagId": "12",
  "TagName": "国语"
}
],
"RequestId": "xx"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

搜索直播互动曲库歌曲

最近更新时间：2023-09-05 01:07:26

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据搜索条件，返回匹配的歌曲列表。

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：SearchKTVMusics。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
KeyWord	是	String	搜索关键词 示例值：""
Offset	是	Integer	分页返回的起始偏移量，默认值：0。将返回第 Offset 到第 Offset+Limit-1 条。 取值范围：Offset + Limit 不超过5000。 示例值：0
Limit	是	Integer	分页返回的起始偏移量，默认值：50。将返回第 Offset 到第 Offset+Limit-1 条。 示例值：10
Sort	否	SortBy	排序方式。默认按照匹配度排序 • Sort.Field 可选 CreateTime • Sort.Order 可选 Desc • 当 KeyWord 不为空时，Sort.Field 字段无效， 搜索结果将以匹配度排序。 示例值：""
TagIds.N	否	Array of String	标签 ID 集合，匹配集合指定所有 ID 。 • 数组长度限制：10。 示例值：""

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总记录数 示例值：1
KTVMusicInfoSet	Array of KTVMusicBaseInfo	KTV 曲目列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 搜索 KTV 歌曲

输入示例

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

```
{
  "Limit": 10,
  "Keyword": "周杰伦",
  "Sort": {
    "Field": "xx",
    "Order": "xx"
  },
  "Offset": 0,
  "TagIds": [
    "xx"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "KTMusicInfoSet": [
      {
        "ComposerSet": [
          "方文山"
        ],
        "SingerInfoSet": [
          {
            "SingerId": "xx",
            "Name": "xx"
          }
        ],
        "MusicId": "xx",
        "SingerSet": [
          "周杰伦"
        ],
        "TagSet": [
          "华语",
          "流行"
        ],
        "Duration": 1,
        "LyricistSet": [
          "周杰伦"
        ],
        "Name": "xx"
      }
    ],
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
InvalidParameter	参数错误。
ResourceUnavailable	资源不可用。
UnsupportedOperation	操作不支持。

获取直播互动曲库联想词

最近更新时间：2023-09-05 01:07:28

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取直播互动曲库联想词

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：DescribeKTVSuggestions。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
KeyWord	是	String	联想关键词 示例值：""

3. 输出参数

参数名称	类型	描述
KTVSuggestionInfoSet	Array of KTVSuggestionInfo	联想词信息列表。返回总数最大为10。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取直播互动曲库联想词

输入示例

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

{
  "KeyWord": "zhou"
}
```

输出示例

```
{
  "Response": {
    "KTVSuggestionInfoSet": [
```

```
{
  "Suggestion": "周杰伦"
},
{
  "RequestId": "xx"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
ResourceUnavailable	资源不可用。

获取直播互动曲目详情

最近更新时间：2023-09-05 01:07:29

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据 Id 查询歌曲的详细信息，包含基础信息及播放信息。

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：DescribeKTVMusicDetail。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MusicId	是	String	曲目 Id 示例值：""

3. 输出参数

参数名称	类型	描述
KTVMusicBaseInfo	KTVMusicBaseInfo	歌曲基础信息
PlayToken	String	播放凭证 示例值：""
LyricsUrl	String	歌词下载地址 示例值：""
DefinitionInfoSet	Array of KTVMusicDefinitionInfo	歌曲规格信息列表 示例值：""
MidiJsonUrl	String	音高数据文件下载地址 示例值：""
ChorusClipSet	Array of ChorusClip	副歌片段数据列表 示例值：""
PreludeInterval	Integer	前奏间隔，单位：毫秒；注：若参数返回为0则无人声部分
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询 KTV 曲目详情

输入示例

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

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

```
{
  "MusicId": "ame-78d2xxx"
}
```

输出示例

```
{
  "Response": {
    "LyricsUrl": "http://125xxx.myqcloud.com/ly.vtt",
    "PreludeInterval": 24000,
    "DefinitionInfoSet": [
      {
        "Definition": "audio/lo",
        "Bitrate": 64000,
        "Size": 3298220
      }
    ],
    "KTVMusicBaseInfo": {
      "ComposerSet": [
        "方文山"
      ],
      "SingerInfoSet": [
        {
          "SingerId": "sid-88djidd",
          "Name": "周杰伦"
        }
      ],
      "MusicId": "mid-1287djxxjx",
      "SingerSet": [
        "周杰伦"
      ],
      "TagSet": [
        "华语",
        "流行"
      ],
      "Duration": 241,
      "LyricistSet": [
        "方文山"
      ],
      "Name": "七里香"
    },
    "ChorusClipSet": [
      {
        "EndTime": 48000,
        "StartTime": 24000
      }
    ],
    "MidijsonUrl": "http://125xxx.myqcloud.com/midi.json",
    "PlayToken": "e1173xx993311ddx3h3gg3d9g4j",
    "RequestId": "1901xx-233xxx-221-xx"
  }
}
```


5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnsupportedOperation	操作不支持。

批量获取直播互动曲目详情

最近更新时间：2023-09-05 01:07:30

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据 Id 列表查询歌曲的详细信息，包含基础信息及播放信息。

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：BatchDescribeKTVMusicDetails。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MusicIds.N	是	Array of String	歌曲Id列表，注：列表最大长度为50

3. 输出参数

参数名称	类型	描述
KTVMusicDetailInfoSet	Array of KTVMusicDetailInfo	歌曲详情列表信息
NotExistMusicIdSet	Array of String	不存在的歌曲 ID 列表。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询直播互动曲目详情列表

查询直播互动曲目详情列表

输入示例

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

{
  "MusicIds": [
    "0000e1f4-502f-4fbc-857b-6dbad90f9802",
    "a"
  ]
}
```

输出示例

```
{
  "Response": {
    "KTVMusicDetailInfoSet": [
      {
        "ChorusClipSet": [
          {
            "EndTime": 109250,
            "StartTime": 64387
          }
        ],
        "DefinitionInfoSet": [
          {
            "Bitrate": 64000,
            "Definition": "audio/mi",
            "Size": 1992000
          },
          {
            "Bitrate": 128000,
            "Definition": "audio/lo",
            "Size": 3984000
          }
        ],
        "KTVMusicBaseInfo": {
          "ComposerSet": [
            ""
          ],
          "Duration": 249,
          "LyricistSet": [
            ""
          ],
          "MusicId": "0000e1f4-502f-4fbc-857b-6dbad90f9802",
          "Name": "我的故事我的歌",
          "SingerInfoSet": [
            {
              "Name": "后来者",
              "SingerId": "05ce39873cc7476ca987d0c621b18ee6"
            }
          ],
          "SingerSet": [
            "后来者"
          ],
          "TagSet": [
            "故事情节",
            "情歌"
          ],
          "LyricsUrl": "https://1500005072.vod2.myqcloud.com/6c9a2f96vodcq1500005072/799b51c33701925920835211299/subtitles/ame-subtitle.vtt?sign=4378ec47c23996d25fe07dc823dff306&t=620dcad9&us=1300054767",
          "PreludeInterval": 0,
          "MidijsonUrl": "https://smart-media-prod-1258642396.file.myqcloud.com/midijson/0000e1f4-502f-4fbc-857b-6dbad90f9802/b2f9a8d1b8f34e86baece70246a8a55b_1640942219276.json?sign=e471c221033bdca7692909a339ac79fe&t=1645067465",
          "PlayToken": "eyJNdXNpY0lkIjoimDAwMGUxZjQ0NTAyZi00ZmjlLTg1N2ltNmRiYWQ5MGY5ODAyliwiRmlsZUIkljoiMzcwMTkyNTkyMDgzNTIxMTI5OSIsIlBsYXItaWduljoiZXIKaGJHY2lPaUpjVXpjMU5pSXNjbll1Y0NjNkIrcFhWQ0o5LmV5SmhjSEJlKwNjNk1UVXdnNREF3TIRBM01pd2lZM1Z5Y21WdWRGUnBiV1ZUZEdGdGNDSTNVFkwTIRBMk56UTJOU3dpWlhod2FYSmxWR2x0WIZOMFhXMXdJam94TmPRMU1EY3hNRkxTENKbWFXeGxTV1FpT2lJek56QXhPVEkxT1Rjd09ETTFNakV4TVprNUlpd2ljR05tWnllNkltRnRaVjY5IRd01ETWIMQ0oxY214QlkyTmxjM05KYm1adklqcDdjblFpT2lJMK1qQmtZMKZrT1NKOWZ
```

```
RLkJKVmg4U0RJShhGX3REWkIDZjZ4NXRtQzZYNmNEZDFLa0s0WDJGUld1S2ciLCJSZXBvcnRLZXkiOijleUpoYkdjaU9pSkIveIV4TWIjc0luUjVjQ0k2SWtw
WFZDSjkuZXIKQmNIQkpaQ0k2TVRNd01EQTFORGMtYnI3aVRYVnphV05KWkNjNklqQXdNREJsTVdZMExUVXdnbVl0TkdaaVI5MDROVGRpTFRaa1ltRm
tPVEJtT1Rnd01pSXNja2xRSWpvaU1URXVNVVE0xTGpVdU1UY3IjaXdpVW1Wd2IzSjBWR2x0WINjNklqSXdnNkl0TURJdE1UY2dNVEU2TVRFNk1EVWIMQ0
psZUhBaU9qRTJORFV3TnpRMk5qVXVOak0zTnpRNUxDSnBjM01pT2IKVFpYSjJhV05sSW4wLmVYQ2RiQzRFXzVtTHItNIBZZGMyVmIcSFE0VXd1akUxU
DZYNFo3T3E0VXc0enZfMXJtcVRcChBfQng3YWFnaVV3al9iRF85cG01eVhEbC16bkNnX3Z3liwiVm9kQXBwSWQiOjE1MDAwMDUwNzJ9"
}
],
"NotExistMusicIdSet": [
  "a"
],
"RequestId": "c163a5b7-b4c9-4b71-9202-81532d9c69dc"
}
}
```

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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。

直播互动曲库歌单相关接口

获取直播互动曲库推荐歌单列表

最近更新时间：2023-09-05 01:07:30

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取直播互动曲库推荐歌单列表。

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：DescribeKTVPlaylists。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Type	否	String	歌单类型，取值有： · OfficialRec：官方推荐 · Normal：自定义 当该字段未填时，默认为取OfficialRec 示例值：1
Offset	否	Integer	分页返回的起始偏移量，默认值：0。将返回第 Offset 到第 Offset+Limit-1 条。 取值范围：Offset + Limit 不超过5000 示例值：10
Limit	否	Integer	分页返回的记录条数，默认值：50。将返回第 Offset 到第 Offset+Limit-1 条。 取值范围：Offset + Limit 不超过5000 示例值：100

3. 输出参数

参数名称	类型	描述
PlaylistBaseInfoSet	Array of KTVPlaylistBaseInfo	推荐歌单列表
TotalCount	Integer	推荐歌单列表总数 示例值：30
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取即时广播曲库推荐歌单

获取即时广播曲库推荐歌单

输入示例

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

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

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "PlaylistBaseInfoSet": [
      {
        "Description": "xx",
        "MusicNum": 0,
        "PlaylistId": "xx",
        "Title": "xx"
      }
    ],
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

获取直播互动曲库推荐歌单详情

最近更新时间：2023-09-05 01:07:31

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据歌单 Id 获取歌单详情，包括歌单的基础信息以及歌曲列表。

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：DescribeKTVPlaylistDetail。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
PlaylistId	是	String	歌单Id
Offset	否	Integer	分页返回的起始偏移量，默认值：0。将返回第 Offset 到第 Offset+Limit-1 条。 示例值：10
Limit	否	Integer	分页返回的记录条数，默认值：50。将返回第 Offset 到第 Offset+Limit-1 条。 示例值：10

3. 输出参数

参数名称	类型	描述
KTVMusicInfoSet	Array of KTVMusicBaseInfo	歌曲基础信息列表
PlaylistBaseInfo	KTVPlaylistBaseInfo	歌单基础信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取即时广播歌单接口

获取即时广播歌单接口

输入示例

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

{
  "PlaylistId": "AME-201803241",
  "Limit": 0,
```



```
"Offset": 10
}
```

输出示例

```
{
  "Response": {
    "PlaylistBaseInfo": {
      "Description": "国内热门",
      "MusicNum": 1,
      "PlaylistId": "pid-xxfdds1",
      "Title": "国内热门"
    },
    "RequestId": "998d8d-11223-xxxd1-192j",
    "KTVMusicInfoSet": [
      {
        "ComposerSet": [
          "方文山"
        ],
        "MusicId": "mid-xd998aa",
        "SingerSet": [
          "周杰伦"
        ],
        "SingerInfoSet": [
          {
            "SingerId": "sid-19xfj8d",
            "Name": "周杰伦"
          }
        ],
        "TagSet": [
          "情歌"
        ],
        "Duration": 241,
        "LyricistSet": [
          "方文山"
        ],
        "Name": "七里香"
      }
    ]
  }
}
```

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。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

直播互动歌手相关接口

获取直播互动曲库歌手分类信息

最近更新时间：2023-09-05 01:07:23

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取直播互动曲库歌手分类信息

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：DescribeKTVSingerCategories。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
GenderSet	Array of KTVSingerCategoryInfo	歌手性别分类列表
AreaSet	Array of KTVSingerCategoryInfo	歌手区域分类列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取直播互动曲库歌手分类信息

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "GenderSet": [
      {
```

```
"EnglishName": "MALE",
"ChineseName": "男"
},
{
"EnglishName": "CH",
"ChineseName": "中国大陆"
},
"RequestId": "xx"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
FailedOperation	操作失败。
InvalidParameterValue	参数取值错误。
UnsupportedOperation	操作不支持。

获取直播互动曲库歌手

最近更新时间：2023-09-05 01:07:21

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据过滤条件，返回匹配的歌手列表。

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：DescribeKTVSingers。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
SingerIds.N	否	Array of String	歌手id集合，精确匹配歌手id 数组长度限制10
Genders.N	否	Array of String	歌手性别集合，不传为全部，精确匹配歌手性别类型， - 数组长度限制1 - 取值范围：直播互动曲库歌手分类信息接口，返回性别分类信息列表中，分类英文名
Areas.N	否	Array of String	歌手区域集合，不传为全部，精确匹配歌手区域 - 数组长度限制10 - 取值范围：直播互动曲库歌手分类信息接口，返回的区域分类信息列表中，分类英文名
Sort	否	SortBy	排序方式。默认按照播放数倒序 Sort.Field 可选 PlayCount。
Offset	否	Integer	分页偏移量，默认值：0。
Limit	否	Integer	分页返回的记录条数，默认值：50。将返回第 Offset 到第 Offset+Limit-1 条。

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总歌手数 示例值：100
KTVSingerInfoSet	Array of KTVSingerInfo	KTV歌手列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取直播互动曲库歌手

输入示例

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

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

```
{
  "Sort": {
    "Field": "PlayCount",
    "Order": "Desc"
  },
  "Limit": 1,
  "Offset": 0,
  "Genders": [
    "MALE"
  ],
  "Areas": [
    "CH"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "KTVSingerInfoSet": [
      {
        "SingerId": "sid-8h3h3gs",
        "Name": "周杰伦",
        "Area": "港台",
        "Gender": "男",
        "PlayCount": 10,
        "MusicCount": 10
      }
    ],
    "RequestId": "2hs73j3-2fur9j4-3jke44-hdhh2"
  }
}
```

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。
InvalidParameter	参数错误。
UnsupportedOperation	操作不支持。

获取歌手下歌曲列表

最近更新時間：2023-09-05 01:07:22

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据歌手id，返回该歌手下歌曲列表。

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：DescribeKTVSingerMusics。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
SingerId	是	String	歌手id
Offset	否	Integer	分页偏移量，默认值：0。 示例值：0
Limit	否	Integer	分页返回的记录条数，默认值：50。将返回第 Offset 到第 Offset+Limit-1 条。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总曲目数 示例值：10
KTVMusicInfoSet	Array of KTVMusicBaseInfo	KTV 曲目列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取歌手下歌曲列表

获取歌手下歌曲列表

输入示例

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

{
  "SingerId": "sid-38xhdd0",
```



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

输出示例

```
{  
  "Response": {  
    "TotalCount": 1,  
    "KTVMusicInfoSet": [  
      {  
        "ComposerSet": [  
          "方文山"  
        ],  
        "MusicId": "mid-399x9e91",  
        "SingerSet": [  
          "周杰伦"  
        ],  
        "SingerInfoSet": [  
          {  
            "SingerId": "sid-38xhhdd0",  
            "Name": "周杰伦"  
          }  
        ],  
        "TagSet": [  
          "伤感"  
        ],  
        "Duration": 241,  
        "LyricistSet": [  
          "方文山"  
        ],  
        "Name": "七里香"  
      }  
    ],  
    "RequestId": "c163a5b7-b4c9-4b71-9202-81532d9c69dc"  
  }  
}
```

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。
InvalidParameter	参数错误。
UnsupportedOperation	操作不支持。

直播互动机器人相关接口

创建直播互动机器人

最近更新时间：2023-09-05 01:07:26

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

创建机器人，支持进入 RTC 房间，播放直播互动曲库歌曲。

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：CreateKTVRobot。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
RTCSystem	是	String	RTC厂商类型，取值有： • TRTC 示例值：TRTC
JoinRoomInput	是	JoinRoomInput	进房参数。
ApplicationLicenseInput	否	ApplicationLicenseInput	license基础信息
SyncRobotCommands.N	否	Array of SyncRobotCommand	创建机器人时初始化参数。

3. 输出参数

参数名称	类型	描述
RobotId	String	机器人Id。 示例值：ame-xxxxxx
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建直播互动机器人

输入示例

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

{
```

```
"RTCSystem": "TRTC",
"JoinRoomInput": {
  "TRTCJoinRoomInput": {
    "Sign": "xxxxxxxxxxxx",
    "RoomId": "12345",
    "SdkAppId": "140000001",
    "UserId": "tom"
  }
}
```

输出示例

```
{
  "Response": {
    "RobotId": "ame-xxxxx",
    "RequestId": "xx"
  }
}
```

示例2 创建直播互动机器人并初始化

输入示例

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

{
  "SyncRobotCommands": [
    {
      "Command": "SetPlaylist",
      "SetPlaylistCommandInput": {
        "Type": "Add",
        "MusicIds": [
          "xxxxxxx"
        ],
        "Index": -1
      }
    },
    {
      "Command": "SetVolume",
      "SetVolumeCommandInput": {
        "Volume": 50
      }
    },
    {
      "Command": "Play",
      "PlayCommandInput": {
        "Index": 0
      }
    }
  ],
  "RTCSystem": "TRTC",
```

```
"JoinRoomInput": {
  "TRTCJoinRoomInput": {
    "Sign": "xxxxxxxxxxx",
    "RoomId": "12345",
    "SdkAppId": "140000001",
    "UserId": "tom"
  }
}
```

输出示例

```
{
  "Response": {
    "RobotId": "ame-xxxxx",
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。

错误码	描述
ResourceUnavailable	资源不可用。

获取直播互动机器人信息

最近更新时间：2023-09-05 01:07:25

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

获取机器人列表，支持 Id、状态等过滤条件。

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：DescribeKTVRobots。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
RobotIds.N	否	Array of String	机器人Id列表。 示例值：ame-xxxxx
Statuses.N	否	Array of String	机器人状态，取值有： - Play：播放 - Pause：暂停 - Destroy：销毁 示例值：Play
CreateTime	否	TimeRange	匹配创建时间在此时间段内的机器人。 包含所指定的头尾时间点。
Offset	否	Integer	分页返回的起始偏移量，默认值：0。将返回第 Offset 到第 Offset+Limit-1 条。 示例值：0
Limit	否	Integer	分页返回的起始偏移量，默认值：10。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	机器人总数。 示例值：100
KTVRobotInfoSet	Array of KTVRobotInfo	机器人信息集合。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询机器人信息

输入示例

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

```
{
  "RobotIds": [
    "ame-xxxxx1"
  ],
  "Statuses": [
    "Play"
  ],
  "CreateTime": {
    "After": "2022-01-10T07:25:52Z",
    "Before": "2022-01-11T07:25:52Z"
  }
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "KTVRobotInfoSet": [
      {
        "RobotId": "ame-xxxxx1",
        "Status": "Play",
        "Playlists": [
          "xxx1",
          "xxx2",
          "xxx3",
          "xxx4"
        ],
        "CurIndex": 2,
        "Position": 120,
        "RTCSystem": "TRTC",
        "JoinRoomInput": {
          "TRTCJoinRoomInput": {
            "Sign": "xxxxxxxxxxx",
            "RoomId": "12345",
            "SdkAppld": "140000001",
            "UserId": "tom"
          }
        },
        "SetAudioParamInput": {
          "Definition": "audio/mi",
          "Type": "Original"
        },
        "SetPlayModelInput": {
          "PlayMode": "Order"
        }
      }
    ],
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。

销毁直播互动机器人

最近更新时间：2023-09-05 01:07:24

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

销毁机器人，机器人退出 RTC 房间。

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：DestroyKTVRobot。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
RobotId	是	String	机器人Id。 示例值：ame-xxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 销毁机器人

输入示例

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

{
  "RobotId": "ame-xxxxx1"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。

同步直播互动机器人指令

最近更新时间：2023-09-05 01:07:24

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

下发操作机器人指令，支持播放、暂停、恢复、歌单设置等操作指令，实现对机器人行为的控制。

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：SyncKTVRobotCommand。
Version	是	String	公共参数，本接口取值：2019-09-16。
Region	否	String	公共参数，此参数为可选参数。
RobotId	是	String	机器人Id。 示例值：ame-xxxxx
Command	是	String	指令，取值有： - Play：播放 - Pause：暂停 - SwitchPrevious：上一首 - SwitchNext：下一首 - SetPlayMode：设置播放模式 - Seek：调整播放进度 - SetPlaylist：歌单变更 - SetAudioParam：音频参数变更 - SendMessage：发送自定义消息 - SetDestroyMode：设置销毁模式 - SetVolume：设置音量（已废弃，请采用 SetRealVolume） - SetRealVolume：设置真实音量 示例值：Play
PlayCommandInput	否	PlayCommandInput	播放参数。
SetPlaylistCommandInput	否	SetPlaylistCommandInput	播放列表变更信息，当Command取SetPlaylist时，必填。
SeekCommandInput	否	SeekCommandInput	播放进度，当Command取Seek时，必填。
SetAudioParamCommandInput	否	SetAudioParamCommandInput	音频参数，当Command取SetAudioParam时，必填。
SendMessageCommandInput	否	SendMessageCommandInput	自定义消息，当Command取SendMessage时，必填。
SetPlayModeCommandInput	否	SetPlayModeCommandInput	播放模式，当Command取SetPlayMode时，必填。
SetDestroyModeCommandInput	否	SetDestroyModeCommandInput	销毁模式，当Command取SetDestroyMode时，必填。
SetVolumeCommandInput	否	SetVolumeCommandInput	音量，当Command取SetVolume时，必填。（已废弃，请采用 SetRealVolumeCommandInput）
SetRealVolumeCommandInput	否	SetRealVolumeCommandInput	真实音量，当Command取SetRealVolume时，必填。

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 播放列表添加歌曲

播放列表添加歌曲

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SetPlaylist",
  "SetPlaylistCommandInput": {
    "Type": "Add",
    "MusicIds": [
      "xxxxxxx"
    ],
    "Index": -1
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例2 播放指定音乐

播放第2首歌曲

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "Play",
  "PlayCommandInput": {
    "Index": 1
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例3 变更歌曲音频参数

变更歌曲音频参数

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SetAudioParam",
  "SetAudioParamCommandInput": {
    "Definition": "audio/hi",
    "Type": "Accompaniment"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例4 切换下一首歌曲

切换下一首歌曲

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SwitchNext"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例5 发送自定义消息

发送自定义消息

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SendMessage",
  "SendMessageCommandInput": {
    "Message": "{\\Field1\\":\\\"Value1\\\"}",
    "Repeat": 2
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例6 设置播放模式

设置播放模式为随机播放。

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SetPlayMode",
  "SetPlayModeCommandInput": {
    "PlayMode": "Shuffle"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例7 设置销毁模式

设置销毁模式为：房间没人后，2分钟内无人进入自动销毁。

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SetDestroyMode",
  "SetDestroyModeCommandInput": {
    "DestroyMode": "Expire",
    "DestroyExpireTime": 120
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

示例8 设置真实音量

设置真实音量

输入示例

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

{
  "RobotId": "xxxxx",
  "Command": "SetRealVolume",
  "SetRealVolumeCommandInput": {
    "RealVolume": 50
  }
}
```

输出示例


```
{
  "Response": {
    "RequestId": "xx"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。

其他接口

获取分类内容（Station）列表

最近更新时间：2023-09-05 01:07:39

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

该服务后续会停用，不再建议使用

默认接口请求频率限制：500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：DescribeStations。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
Limit	是	Integer	条数，必须大于0
Offset	是	Integer	offset (Default = 0)，Offset=Offset+Limit

3. 输出参数

参数名称	类型	描述
Total	Integer	总数量
Offset	Integer	分页偏移量
Size	Integer	当前页station数量
HaveMore	Integer	剩余数量（total-offset-size），通过这个值判断是否还有下一页
Stations	Array of Station	Stations 素材库列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取分类内容（Station）列表

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeStations
&Limit=1
&Offset=0
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "HaveMore": 0,
    "Offset": 0,
    "RequestId": "s1568780320026098000",
    "Size": 3,
    "Stations": [
      {
        "CategoryCode": "",
        "CategoryID": "DB5C207AE09E058C",
        "ImagePathMap": [
          {
            "Key": "JPG-320X320-STATION",
            "Value": ""
          },
          {
            "Key": "JPG-600X600-STATION",
            "Value": ""
          },
          {
            "Key": "JPG-NXN-STATION",
            "Value": ""
          }
        ],
        "Name": "素材-国风",
        "Rank": 1
      },
      {
        "Total": 3
      }
    ]
  }
}
```

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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。

获取分类内容下歌曲列表

最近更新时间：2023-09-05 01:07:40

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

该服务后续会停用，不再建议使用

默认接口请求频率限制：500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：DescribeItems。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
Offset	是	Integer	offset (Default = 0)，(当前页-1) * Limit
Limit	是	Integer	条数，必须大于0，最大值为30
CategoryId	否	String	(电台/歌单) ID，CategoryId和CategoryCode两个必传1个，可以从 获取分类内容（Station）列表接口 中获取。
CategoryCode	否	String	(电台/歌单) ID，CategoryId和CategoryCode两个必传1个，可以从 获取分类内容（Station）列表接口 中获取。

3. 输出参数

参数名称	类型	描述
Offset	Integer	分页偏移量
Size	Integer	当前页歌曲数量
Total	Integer	总数据条数
HaveMore	Integer	剩余数量（total-offset-size），通过这个值判断是否还有下一页
Items	Array of Item	Items 歌曲列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取分类内容下歌曲列表

输入示例

```
https://ame.tencentcloudapi.com/?Action=DescribeItems
&Limit=1
&Offset=0
```

&CategoryId=xxxxx

&<公共请求参数>

输出示例

```
{
  "Response": {
    "RequestId": "xx",
    "Items": [
      {
        "ItemID": "xx",
        "Album": {
          "ImagePathMap": [
            {
              "Value": "xx",
              "Key": "xx"
            },
            {
              "Key": "xx",
              "Value": "xx"
            },
            {
              "Key": "xx",
              "Value": "xx"
            },
            {
              "Key": "xx",
              "Value": "xx"
            }
          ],
          "AlbumName": "xx"
        },
        "Status": 0,
        "DataInfo": {
          "Duration": "xx",
          "AuditionBegin": 1,
          "Version": "xx",
          "Name": "xx",
          "AuditionEnd": 1
        },
        "Artists": [
          {
            "ArtistName": "xx"
          }
        ],
        "HaveMore": 1,
        "Offset": 1,
        "Total": 1,
        "Size": 1
      }
    ]
  }
}
```

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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。

查询歌曲在售状态

最近更新时间：2023-09-05 01:07:39

1. 接口描述

接口请求域名：ame.tencentcloudapi.com。

根据音乐信息查询音乐是否在售

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：DescribeMusicSaleStatus。
Version	是	String	公共参数 ，本接口取值：2019-09-16。
Region	否	String	公共参数 ，此参数为可选参数。
MusicIds.N	是	Array of String	歌曲Id集合，可传单个，也可传多个，上线查询单次50个
PurchaseType	是	Integer	查询哪个渠道的数据，1为曲库包，2为单曲 示例值：1

3. 输出参数

参数名称	类型	描述
MusicStatusSet	Array of MusicStatus	musicId对应歌曲状态
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询歌曲在售状态

根据音乐信息查询音乐是否在售

输入示例

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

{
  "PurchaseType": 1,
  "MusicIds": [
    "xx"
  ]
}
```


输出示例

```
{
  "Response": {
    "RequestId": "xx",
    "MusicStatusSet": [
      {
        "MusicId": "xx",
        "SaleStatus": 0
      }
    ]
  }
}
```

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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。

数据结构

最近更新时间：2023-08-17 01:06:48

Album

Album

被如下接口引用：DescribeItemById, DescribeItems。

名称	类型	描述
AlbumName	String	专辑名
ImagePathMap	Array of ImagePath	专辑图片大小及类别 注意：此字段可能返回 null，表示取不到有效值。

ApplicationLicenseInput

用户license基础信息

被如下接口引用：CreateKTVRobot。

名称	类型	必选	描述
AppName	String	是	应用名称，注：后面三个字段AndroidPackageName、IOSBundleId、PcIdentifier，三者选填一个
AndroidPackageName	String	否	app的安卓包名
IOSBundleId	String	否	app的IOS的BundleId名
PcIdentifier	String	否	PC标识名

Artist

Artist

被如下接口引用：DescribeItemById, DescribeItems。

名称	类型	描述
ArtistName	String	歌手名

AuthInfo

AuthInfo集合

被如下接口引用：DescribeAuthInfo。

名称	类型	描述
SubjectName	String	主体名称 注意：此字段可能返回 null，表示取不到有效值。
ProjectName	String	项目名称 注意：此字段可能返回 null，表示取不到有效值。
AppScene	Integer	应用场景
AppRegion	Integer	应用地域
AuthPeriod	Integer	授权时间
Commercialization	Integer	是否可商业化
Platform	Integer	是否可跨平台
Id	String	加密后Id

ChorusClip

副歌片段信息

被如下接口引用：BatchDescribeKTVMusicDetails, DescribeKTVMusicDetail。

名称	类型	描述
StartTime	Integer	副歌时间，单位：毫秒
EndTime	Integer	副歌结束时间，单位：毫秒

DataInfo

数据信息

被如下接口引用：DescribeItemById, DescribeItems。

名称	类型	描述
Name	String	Song Name
Version	String	歌曲版本
Duration	String	歌曲总时长（非试听时长）
AuditionBegin	Integer	试听开始时间
AuditionEnd	Integer	试听结束时间
TagNames	Array of String	标签名称

ImagePath

图片路径

被如下接口引用：DescribeItemById, DescribeItems, DescribeStations。

名称	类型	描述
Key	String	station图片大小及类别 注意：此字段可能返回 null，表示取不到有效值。
Value	String	station图片地址 注意：此字段可能返回 null，表示取不到有效值。

Item

歌曲信息

被如下接口引用：DescribeItemById, DescribeItems。

名称	类型	描述
ItemId	String	Song ID
DataInfo	DataInfo	Song info 注意：此字段可能返回 null，表示取不到有效值。
Album	Album	专辑信息 注意：此字段可能返回 null，表示取不到有效值。
Artists	Array of Artist	多个歌手集合 注意：此字段可能返回 null，表示取不到有效值。
Status	Integer	歌曲状态，1:添加进购物车；2:核销进曲库包 注意：此字段可能返回 null，表示取不到有效值。

JoinRoomInput

直播进房输入参数

被如下接口引用：CreateKTVRobot, DescribeKTVRobots。

名称	类型	必选	描述
TRTCJoinRoomInput	TRTCJoinRoomInput	否	TRTC进房参数

KTVMusicBaseInfo

KTV 曲目基础信息

被如下接口引用：BatchDescribeKTVMusicDetails, DescribeKTVMusicDetail, DescribeKTVPlaylistDetail, DescribeKTVSingerMusics, SearchKTVMusics。

名称	类型	描述
MusicId	String	歌曲 Id 示例值：""
Name	String	歌曲名称 示例值：""
SingerInfoSet	Array of KTVSingerBaseInfo	演唱者基础信息列表 示例值：""
SingerSet	Array of String	已弃用，请使用SingerInfoSet 示例值：""
LyricistSet	Array of String	作词者列表 示例值：""
ComposerSet	Array of String	作曲者列表 示例值：""
TagSet	Array of String	标签列表 示例值：""
Duration	Integer	歌曲时长 示例值：0

KTVMusicDefinitionInfo

直播互动歌曲规格信息。

被如下接口引用：BatchDescribeKTVMusicDetails, DescribeKTVMusicDetail。

名称	类型	描述
Definition	String	规格，取值有： - audio/mi：低规格； - audio/lo：中规格； - audio/hi：高规格。 示例值：""
Bitrate	Integer	码率，单位为 bps。 示例值：0
Size	Integer	文件大小，单位为字节。 示例值：0

KTVMusicDetailInfo

即使广播曲库歌曲信息详情列表

被如下接口引用：BatchDescribeKTVMusicDetails。

名称	类型	描述
KTVMusicBaseInfo	KTVMusicBaseInfo	即使广播曲库歌曲基础信息
PlayToken	String	播放凭证
LyricsUrl	String	歌词下载地址
DefinitionInfoSet	Array of KTVMusicDefinitionInfo	歌曲规格信息列表
MidiJsonUrl	String	音高数据文件下载地址
ChorusClipSet	Array of ChorusClip	副歌片段数据列表
PreludeInterval	Integer	前奏间隔，单位：毫秒；注：若参数返回为0则无人声部分

KTVMusicTagGroup

即使广播曲库歌曲标签分组信息

被如下接口引用：DescribeKTVMusicTags。

名称	类型	描述
EnglishGroupName	String	标签分组英文名 示例值：""
ChineseGroupName	String	标签分组中文名 示例值：""
TagSet	Array of KTVMusicTagInfo	标签分类下标签列表 示例值：""

KTVMusicTagInfo

即使广播曲库歌曲标签信息

被如下接口引用：DescribeKTVMusicTags。

名称	类型	描述
TagId	String	标签Id 示例值：""
TagName	String	标签 示例值：""

KTVMusicTopInfo

排行榜结构

被如下接口引用：DescribeKTVTopList。

名称	类型	描述
MusicId	String	歌曲Id
Name	String	歌曲名称
SingerInfoSet	Array of KTVSingerBaseInfo	歌手名称列表
LyricistSet	Array of String	歌词名称列表
ComposerSet	Array of String	作曲列表
TagSet	Array of String	标签列表
Duration	Integer	播放时长 示例值：10

KTVPlaylistBaseInfo

推荐歌单基础信息

被如下接口引用：DescribeKTVPlaylistDetail, DescribeKTVPlaylists。

名称	类型	描述
PlaylistId	String	歌单Id
Title	String	歌单标题
Description	String	歌单介绍
MusicNum	Integer	歌曲数量 示例值：1

KTVRobotInfo

机器人信息

被如下接口引用：DescribeKTVRobots。

名称	类型	描述
RobotId	String	机器人Id。 示例值：ame-xxxxx
Status	String	状态，取值有： - Play：播放 - Pause：暂停 - Destroy：销毁 示例值：Play
Playlists	Array of String	播放列表。 示例值：xxxx
CurIndex	Integer	当前歌单索引位置。 示例值：1
Position	Integer	播放进度，单位：毫秒。 示例值：120
SetAudioParamInput	SetAudioParamCommandInput	音频参数。
JoinRoomInput	JoinRoomInput	进房信息。
RTCSys tem	String	RTC厂商类型，取值有： TRTC 示例值：TRTC
SetPlayModeInput	SetPlayModeCommandInput	播放模式，PlayMode取值有： - RepeatPlaylist：列表循环 - Order：顺序播放 - RepeatSingle：单曲循环 - Shuffle：随机播放
SetVolumeInput	SetVolumeCommandInput	音量，范围 0~100，默认为 50。（已废弃，请采用 SetRealVolumeInput）
SetRealVolumeInput	SetRealVolumeCommandInput	真实音量，范围 0~100，默认为 50。

KTVSingerBaseInfo

KTV 歌手基础信息

被如下接口引用：DescribeKTVMusicDetail, DescribeKTVPlaylistDetail, DescribeKTVSingerMusics, DescribeKTVTopList, SearchKVMusics。

名称	类型	描述
----	----	----

名称	类型	描述
SingerId	String	歌手id 示例值: ""
Name	String	歌手名 示例值: ""

KTVSingerCategoryInfo

KTV歌手分类信息

被如下接口引用：DescribeKTVSingerCategories。

名称	类型	描述
ChineseName	String	分类中文名 示例值: 大陆
EnglishName	String	分类英文名 示例值: Japan

KTVSingerInfo

直播互动歌曲的歌手信息。

被如下接口引用：DescribeKTVSingers。

名称	类型	描述
SingerId	String	歌手id
Name	String	歌手名 示例值: 张学友
Gender	String	歌手性别: 男, 女, 组合 示例值: 男
Area	String	地区: 大陆, 港台, 欧美, 日本 示例值: 大陆
MusicCount	Integer	歌曲数 示例值: 201
PlayCount	Integer	歌曲总播放次数 示例值: 2000

KTVSuggestionInfo

即使广播曲库联想词信息

被如下接口引用：DescribeKTVSuggestions。

名称	类型	描述
Suggestion	String	联想词 示例值: ""

Lyric

歌词信息

被如下接口引用：DescribeLyric。

名称	类型	描述
Url	String	歌词cdn地址

名称	类型	描述
FileNameExt	String	歌词后缀名
SubItem Type	String	歌词类型

Music

音乐详情

被如下接口引用：DescribeMusic。

名称	类型	描述
Url	String	音乐播放链接相对路径，必须通过在正版曲库直通车控制台上登记的域名进行拼接。
FileSize	Integer	音频文件大小
FileExtension	String	音频文件类型
AuditionBegin	Integer	Song fragment start.试听片段开始时间，试听时长为auditionEnd-auditionBegin Unit :ms
AuditionEnd	Integer	Song fragment end.试听片段结束时间，试听时长为auditionEnd-auditionBegin Unit :ms
FullUrl	String	音乐播放链接全路径，前提是在正版曲库直通车控制台添加过域名，否则返回空字符。 如果添加过多个域名只返回第一个添加域名的播放全路径。

MusicDetailInfo

歌曲变更细节

被如下接口引用：ModifyMusicOnShelves。

名称	类型	必选	描述
MusicId	String	是	资源方音乐Id
AmelId	String	是	资源方识别信息 示例值：TME
Tags	Array of String	否	分类标签 示例值：["伴奏"]
HitWords	Array of String	否	关键词 示例值：[""]
Bpm	Integer	否	节奏信息 示例值：0
Score	Float	否	商业化权益 示例值：5.67
Scene	Array of String	否	应用歌曲信息,1.图文/短视频,2.网络直播,3.网络电台FM,4.免费游戏,5.商业游戏,6.网店网站设计,7.广告营销,8.网络长视频 示例值：["1"]
Region	Array of String	否	应用地域,1. 中国大陆,2. 中国含港澳台,3. 全球 示例值：["1"]
AuthPeriod	String	否	授权时间,1. 1年, 5. 随片永久 示例值："1"
Commercialization	String	否	商业化授权, 1. 支持商业化 ,2. 不支持商业化 示例值："1"
Platform	String	否	跨平台传播, 1. 支持跨平台传播 ,2. 不支持跨平台传播 示例值："1"

名称	类型	必选	描述
Channel	String	否	传播渠道 示例值: "1"

MusicOpenDetail

对外开放信息

被如下接口引用: DescribeCloudMusicPurchased。

名称	类型	描述
MusicId	String	音乐Id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""
AlbumName	String	专辑名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""
AlbumImageUrl	String	专辑图片路径 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""
MusicName	String	音乐名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""
MusicImageUrl	String	音乐图片路径 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""
Singers	Array of String	歌手 注意: 此字段可能返回 null, 表示取不到有效值。
Duration	Integer	播放时长 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
Tags	Array of String	标签 注意: 此字段可能返回 null, 表示取不到有效值。
LyricUrl	String	歌词url 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""
WaveformUrl	String	波形图url 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ""

MusicStatus

返回单曲页面歌曲是否在售状态

被如下接口引用: DescribeMusicSaleStatus。

名称	类型	描述
MusicId	String	歌曲Id
SaleStatus	Integer	在售状态,0为在售, 1为临时下架, 2为永久下架

OfflineMusicDetail

曲库包已下架歌曲详细信息

被如下接口引用: DescribePkgOfflineMusic。

名称	类型	描述
ItemId	String	歌曲Id
MusicName	String	歌曲名称
OffRemark	String	不可用原因
OffTime	String	不可用时间

Package

曲库包信息

被如下接口引用：DescribePackages。

名称	类型	描述
OrderId	String	订单id
Name	String	曲库包名称
AuthorizedArea	String	授权地区-global: 全球 CN: 中国
AuthorizedLimit	Integer	授权次数
TermOfValidity	Integer	套餐有效期，单位:天
Commercial	Integer	0:不可商业化；1:可商业化
PackagePrice	Float	套餐价格，单位：元
EffectTime	String	生效开始时间,格式yyyy-MM-dd HH:mm:ss
ExpireTime	String	生效结束时间,格式yyyy-MM-dd HH:mm:ss
UsedCount	Integer	剩余授权次数
UseRanges	Array of UseRange	曲库包用途信息

Packageltem

曲库包歌曲信息

被如下接口引用：DescribePackageltems。

名称	类型	描述
OrderId	String	订单id 示例值：空字符串
TrackName	String	歌曲名 示例值：空字符串
ItemID	String	歌曲ID 示例值：空字符串
Img	String	专辑图片 示例值：空字符串
ArtistName	String	歌手名 示例值：空字符串
Duration	String	歌曲时长 示例值：空字符串
AuthorizedArea	String	授权区域，global: 全球 CN: 中国 示例值：空字符串

名称	类型	描述
Tags	Array of String	标签数组 示例值：空数组

PlayCommandInput

播放指令输入参数

被如下接口引用：CreateKTVRobot, SyncKTVRobotCommand。

名称	类型	必选	描述
Index	Integer	是	歌曲位置索引。 示例值：2

SeekCommandInput

调整播放进度指令参数

被如下接口引用：CreateKTVRobot, SyncKTVRobotCommand。

名称	类型	必选	描述
Position	Integer	是	播放位置，单位：毫秒。 示例值：120

SendMessageCommandInput

发送自定义信息指令参数

被如下接口引用：CreateKTVRobot, SyncKTVRobotCommand。

名称	类型	必选	描述
Message	String	是	自定义消息，json格式字符串。 示例值：{"field":"value"}
Repeat	Integer	否	消息重复次数，默认为 1。 示例值：2

SetAudioParamCommandInput

音频参数信息

被如下接口引用：CreateKTVRobot, DescribeKTVRobots, SyncKTVRobotCommand。

名称	类型	必选	描述
Definition	String	否	规格，取值有： - audio/mi：低规格 - audio/lo：中规格 - audio/hi：高规格 示例值：audio/hi
Type	String	否	音频类型，取值有： - Original：原唱 - Accompaniment：伴奏 示例值：Original

SetDestroyModeCommandInput

设置销毁模式

被如下接口引用：CreateKTVRobot, SyncKTVRobotCommand。

名称	类型	必选	描述
DestroyMode	String	是	销毁模式，取值有： - Auto：房间没人时自动销毁 - Expire：房间没人时过期自动销毁 - Never：不自动销毁，需手动销毁 默认为：Auto。 示例值：Auto
DestroyExpireTime	Integer	否	过期销毁时间，单位：秒，当DestroyMode取Expire时必须填。 示例值：120

SetPlayModeCommandInput

设置播放模式

被如下接口引用：CreateKTVRobot, DescribeKTVRobots, SyncKTVRobotCommand。

名称	类型	必选	描述
PlayMode	String	是	播放模式，取值有： - RepeatPlaylist：列表循环 - Order：顺序播放 - RepeatSingle：单曲循环 - Shuffle：随机播放 示例值：RepeatPlaylist

SetPlaylistCommandInput

设置播放列表指令参数

被如下接口引用：CreateKTVRobot, SyncKTVRobotCommand。

名称	类型	必选	描述
Type	String	是	变更类型，取值有： - Add：添加 - Delete：删除 - ClearList：清空歌曲列表 - Move：移动歌曲 示例值：Add
Index	Integer	否	歌单索引位置， 当 Type 取 Add 时，-1表示添加在列表最后位置，大于-1表示要添加的位置； 当 Type 取 Delete 时，表示待删除歌曲的位置； 当 Type 取 Move 时，表示待调整歌曲的位置。 示例值：1
ChangedIndex	Integer	否	当 Type 取 Move 时，必填，表示移动歌曲的目标位置。 示例值：2
MusicIds	Array of String	否	歌曲 ID 列表，当 Type 取 Add 时，与MusicURLs必填其中一项。 示例值：xxxxxx
MusicURLs	Array of String	否	歌曲 URL 列表，当 Type 取 Add 时，与MusicIds必填其中一项。 注：URL必须以.mp3结尾且必须是mp3编码文件。 示例值：http://demo.mp3

SetRealVolumeCommandInput

设置真实音量。

被如下接口引用：CreateKTVRobot, DescribeKTVRobots, SyncKTVRobotCommand。

名称	类型	必选	描述
RealVolume	Integer	是	真实音量大小，取值范围为 0~100，默认值为 50。 示例值：50

SetVolumeCommandInput

设置音量。

被如下接口引用：CreateKTVRobot, DescribeKTVRobots, SyncKTVRobotCommand。

名称	类型	必选	描述
Volume	Integer	是	音量大小，取值范围为 0~100，默认值为 50。 示例值：50

SortBy

排序依据

被如下接口引用：DescribeKTVSingers, SearchKTVMusics。

名称	类型	必选	描述
Field	String	是	排序字段
Order	String	是	排序方式，可选值：Asc（升序）、Desc（降序） 示例值：Asc

Station

分类内容

被如下接口引用：DescribeStations。

名称	类型	描述
CategoryID	String	StationID
CategoryCode	String	Station MCCode 注意：此字段可能返回 null，表示取不到有效值。
Name	String	Category Name 注意：此字段可能返回 null，表示取不到有效值。
Rank	Integer	Station的排序值，供参考（返回结果已按其升序） 注意：此字段可能返回 null，表示取不到有效值。
ImagePathMap	Array of ImagePath	station图片集合 注意：此字段可能返回 null，表示取不到有效值。

SyncRobotCommand

KTV 机器人初始化参数，在创建后自动完成相关初始化工作。

被如下接口引用：CreateKTVRobot。

名称	类型	必选	描述
----	----	----	----

名称	类型	必选	描述
Command	String	是	可同时传入多个指令，顺序执行。取值有： - Play：播放 - Pause：暂停 - SwitchPrevious：上一首 - SwitchNext：下一首 - SetPlayMode：设置播放模式 - Seek：调整播放进度 - SetPlaylist：歌单变更 - SetAudioParam：音频参数变更 - SendMessage：发送自定义消息 - SetDestroyMode：设置销毁模式 SetVolume：设置音量（已废弃，请采用 SetRealVolume） - SetRealVolume：设置真实音量
PlayCommandInput	PlayCommandInput	否	播放参数。
SetPlaylistCommandInput	SetPlaylistCommandInput	否	播放列表变更信息，当Command取SetPlaylist时，必填。
SeekCommandInput	SeekCommandInput	否	播放进度，当Command取Seek时，必填。
SetAudioParamCommandInput	SetAudioParamCommandInput	否	音频参数，当Command取SetAudioParam时，必填。
SendMessageCommandInput	SendMessageCommandInput	否	自定义消息，当Command取SendMessage时，必填。
SetPlayModeCommandInput	SetPlayModeCommandInput	否	播放模式，当Command取SetPlayMode时，必填。
SetDestroyModeCommandInput	SetDestroyModeCommandInput	否	销毁模式，当Command取SetDestroyMode时，必填。
SetVolumeCommandInput	SetVolumeCommandInput	否	音量，当Command取SetVolume时，必填。 （已废弃，请采用 SetRealVolumeCommandInput）
SetRealVolumeCommandInput	SetRealVolumeCommandInput	否	真实音量，当Command取SetRealVolume时，必填。

TRTCJoinRoomInput

TRTC推流进房信息

被如下接口引用：CreateKTVRobot。

名称	类型	必选	描述
Sign	String	是	签名。 示例值：xxxxxxx
RoomId	String	是	房间号。 示例值：12345
SdkAppId	String	是	推流应用ID。 示例值：140000001
UserId	String	是	用户唯一标识。 示例值：tom
PrivateMapKey	String	否	进房钥匙，若需要权限控制请携带该参数。 privateMapKey 权限设置 示例值：xxxxxxx
Role	String	否	用户角色，目前支持两种角色： - anchor：主播 - audience：观众 示例值：anchor

名称	类型	必选	描述
RoomIdType	String	否	TRTC房间号的类型： - Integer：数字类型 - String：字符串类型 默认为：Integer。 示例值：Integer

TakeMusicOffShelves

下架歌曲复合结构

被如下接口引用：TakeMusicOffShelves。

名称	类型	必选	描述
MusicIds	String	是	资源方对应音乐Id
SaleStatus	String	是	当曲目临时下架时：已订购客户无影响，无需消息通知。当曲目封杀下架后，推送消息至已订购老客户，枚举值，判断是否上/下架 在售状态，0在售，1临时下架，2永久下架 示例值：1

TimeRange

时间范围

被如下接口引用：DescribeKTVRobots。

名称	类型	必选	描述
Before	String	否	- 大于等于此时间（起始时间）。 - 格式按照 ISO 8601标准表示，详见 ISO 日期格式说明。 示例值：2020-12-10T07:25:52Z
After	String	否	- 小于此时间（结束时间）。 - 格式按照 ISO 8601标准表示，详见 ISO 日期格式说明。 示例值：2020-12-10T07:25:52Z

UseRange

曲库包用途信息

被如下接口引用：DescribePackages。

名称	类型	描述
UseRangeId	Integer	用途id
Name	String	用途范围名称

错误码

最近更新时间：2022-02-28 08:05:18

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。
OperationDenied	操作被拒绝。
ResourcesSoldOut	资源售罄。