

# 语音识别 API 文档



腾讯云

#### 【版权声明】

©2013–2025 腾讯云版权所有

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

#### 【商标声明】



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

#### 【服务声明】

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

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

#### 【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

## 文档目录

### 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

语音流异步识别相关接口

语音流异步识别任务关闭

语音流异步识别任务列表

语音流异步识别任务创建

录音文件识别相关接口

录音文件识别结果查询

录音文件识别请求

录音文件识别回调说明

一句话识别相关接口

一句话识别

自学习相关接口

创建自学习模型

查询自学习模型列表

更新自学习模型

删除自学习模型

修改自学习模型状态

下载自学习模型语料

获取自学习模型信息

热词相关接口

创建热词表

删除热词表

下载热词表

获取热词表

列举热词表

设置热词表状态

更新热词表

说话人验证相关接口

说话人注册

说话人认证

说话人更新

说话人删除

说话人比对

说话人注册数量统计

说话人验证1:N

关键字检索相关接口

创建关键词表

删除关键词表

- 列举关键词表
  - 更新关键词表
- 业务用量相关接口
  - 通过日期查询用量
- 实时语音识别相关接口
  - 实时语音识别（WebSocket）
- 录音文件识别极速版相关接口
  - 录音文件识别极速版
- 虚拟号真人判定相关接口
  - 虚拟号真人判定（websocket）
- 数据结构
- 错误码



# API 文档

## 更新历史

最近更新时间：2025-05-13 01:09:56

### 第 43 次发布

发布时间：2025-05-13 01:09:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：ReplaceTextId

### 第 42 次发布

发布时间：2025-05-09 01:09:46

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetAsrVocabList](#)
  - 废弃入参：TagInfos

修改数据结构：

- [Model](#)
  - 废弃成员：TagInfos

### 第 41 次发布

发布时间：2025-03-19 01:10:31

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAsrKeyWordLib](#)
  - 修改入参：KeyWordFile
- [UpdateAsrKeyWordLib](#)
  - 修改入参：Name, KeyWordFile

### 第 40 次发布

发布时间：2024-09-23 01:07:54

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetUsageByDate](#)

新增数据结构：

- [UsageByDateInfo](#)
- [UsageByDateInfoData](#)

## 第 39 次发布

发布时间：2024-09-02 01:09:58

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAsrKeyWordLib](#)
- [DeleteAsrKeyWordLib](#)
- [GetAsrKeyWordLibList](#)
- [UpdateAsrKeyWordLib](#)

新增数据结构：

- [KeyWordLib](#)
- [KeyWordLibIdData](#)
- [KeyWordLibListData](#)

## 第 38 次发布

发布时间：2024-08-30 01:05:12

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：KeyWordLibIdList

新增数据结构：

- [KeyWordResult](#)

修改数据结构：

- [SentenceDetail](#)
  - 新增成员：KeyWordResults

## 第 37 次发布

发布时间：2024-06-27 01:05:14

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：HotwordList

## 第 36 次发布

发布时间：2024-06-25 01:07:52

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [VoicePrintGroupVerify](#)

修改接口：

- [VoicePrintCount](#)
  - 新增入参：GroupId, CountMod
- [VoicePrintDelete](#)
  - 新增入参：GroupId, DelMod
- [VoicePrintEnroll](#)
  - 新增入参：GroupId

新增数据结构：

- [VerifyTop](#)
- [VerifyTopResult](#)

修改数据结构：

- [VoicePrintCountData](#)
  - 新增成员：VoicePrintList

## 第 35 次发布

发布时间：2024-06-21 01:10:11

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [VoicePrintDelete](#)
  - 修改入参：VoicePrintId

## 第 34 次发布

发布时间：2024-04-11 01:07:46

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)

- **废弃入参**：ReinforceHotword
- [SentenceRecognition](#)
  - **废弃入参**：ReinforceHotword

## 第 33 次发布

发布时间：2024-02-23 01:07:51

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateCustomization](#)
  - **废弃入参**：TagInfos
- [GetCustomizationList](#)
  - **废弃入参**：TagInfos

## 第 32 次发布

发布时间：2024-02-02 01:05:31

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [VoicePrintCompare](#)

新增数据结构：

- [VoicePrintCompareData](#)

## 第 31 次发布

发布时间：2023-09-22 01:10:19

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SentenceDetail](#)
  - 新增成员：WrittenText
  - **修改成员**：EmotionType

## 第 30 次发布

发布时间：2023-08-08 01:06:35

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [VoicePrintCount](#)

新增数据结构：

- [VoicePrintCountData](#)

## 第 29 次发布

发布时间：2023-07-18 01:07:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SentenceRecognition](#)
  - 新增入参：InputSampleRate

## 第 28 次发布

发布时间：2023-07-05 01:07:50

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [VoicePrintDelete](#)
- [VoicePrintEnroll](#)
- [VoicePrintUpdate](#)
- [VoicePrintVerify](#)

新增数据结构：

- [VoicePrintBaseData](#)
- [VoicePrintVerifyData](#)

## 第 27 次发布

发布时间：2023-05-22 01:03:30

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SentenceRecognition](#)
  - 废弃入参：ProjectId, SubServiceType, UsrAudioKey

## 第 26 次发布

发布时间：2023-05-11 01:03:26

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SentenceRecognition](#)
  - 新增入参：HotwordList

## 第 25 次发布

发布时间：2023-05-08 01:07:44

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetModelInfo](#)

## 第 24 次发布

发布时间：2023-04-05 01:07:23

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SentenceRecognition](#)
  - 修改入参：ProjectId, SubServiceType, UsrAudioKey

## 第 23 次发布

发布时间：2023-02-09 01:04:17

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：EmotionRecognition

修改数据结构：

- [SentenceDetail](#)
  - 新增成员：EmotionType

## 第 22 次发布

发布时间：2022-12-28 01:04:09

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：SentenceMaxLength

## 第 21 次发布

发布时间：2022-11-28 06:10:45

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：EmotionalEnergy, ReinforceHotword
- [SentenceRecognition](#)
  - 新增入参：ReinforceHotword

修改数据结构：

- [SentenceDetail](#)
  - 新增成员：EmotionalEnergy, SilenceTime

## 第 20 次发布

发布时间：2022-09-02 06:09:54

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SentenceRecognition](#)
  - 新增入参：CustomizationId

## 第 19 次发布

发布时间：2022-04-12 06:06:07

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SentenceDetail](#)
  - 新增成员：SpeakerId

## 第 18 次发布

发布时间：2022-03-18 08:06:40

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：CustomizationId

## 第 17 次发布

发布时间：2021-12-22 08:06:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateAsyncRecognitionTask](#)

- 新增入参：AudioData

## 第 16 次发布

发布时间：2021-12-09 08:01:20

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [TaskStatus](#)
  - 新增成员：AudioDuration

## 第 15 次发布

发布时间：2021-01-18 08:00:30

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CloseAsyncRecognitionTask](#)

## 第 14 次发布

发布时间：2021-01-15 08:00:22

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAsyncRecognitionTask](#)
- [DescribeAsyncRecognitionTasks](#)

新增数据结构：

- [AsyncRecognitionTaskInfo](#)
- [AsyncRecognitionTasks](#)

## 第 13 次发布

发布时间：2020-09-24 08:00:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [SentenceDetail](#)
  - 新增成员：SpeechSpeed

## 第 12 次发布

发布时间：2020-09-23 08:00:03



本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateCustomization](#)
- [DeleteCustomization](#)
- [DownloadCustomization](#)
- [ModifyCustomization](#)
- [ModifyCustomizationState](#)

## 第 11 次发布

发布时间：2020-09-18 08:00:03

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetCustomizationList](#)

修改接口：

- [GetAsrVocabList](#)
  - 新增入参：TagInfos, Offset, Limit
  - 新增出参：TotalCount
- [SentenceRecognition](#)
  - 新增入参：WordInfo
  - 新增出参：WordSize, WordList

新增数据结构：

- [Model](#)
- [SentenceWord](#)

修改数据结构：

- [Vocab](#)
  - 新增成员：TagInfos

## 第 10 次发布

发布时间：2020-08-11 08:00:18

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：FilterPunc

## 第 9 次发布

发布时间：2020-07-27 08:00:08

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：Extra, SpeakerDiarization, SpeakerNumber
- [SentenceRecognition](#)
  - 新增出参：AudioDuration

## 第 8 次发布

发布时间：2020-04-30 08:03:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：ConvertNumMode
- [SentenceRecognition](#)
  - 新增入参：ConvertNumMode

## 第 7 次发布

发布时间：2020-03-19 08:03:01

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateRecTask](#)
  - 新增入参：FilterDirty, FilterModal
- [SentenceRecognition](#)
  - 新增入参：FilterDirty, FilterModal, FilterPunc

## 第 6 次发布

发布时间：2020-02-26 14:16:23

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [SentenceDetail](#)
- [SentenceWords](#)

修改数据结构：

- [TaskStatus](#)
  - 新增成员：ResultDetail

## 第 5 次发布

发布时间：2020-02-24 14:41:30

本次发布包含了以下内容：

改善已有的文档。

**删除数据结构：**

- SentenceDetail
- SentenceWords

**修改数据结构：**

- [TaskStatus](#)
  - 删除成员：ResultDetail

## 第 4 次发布

发布时间：2020-02-24 14:04:06

本次发布包含了以下内容：

改善已有的文档。

**新增接口：**

- [DownloadAsrVocab](#)
- [GetAsrVocabList](#)
- [SetVocabState](#)

**新增数据结构：**

- [SentenceDetail](#)
- [SentenceWords](#)
- [Vocab](#)

**修改数据结构：**

- [TaskStatus](#)
  - 新增成员：ResultDetail

## 第 3 次发布

发布时间：2020-01-15 19:33:53

本次发布包含了以下内容：

改善已有的文档。

**新增接口：**

- [CreateAsrVocab](#)
- [DeleteAsrVocab](#)
- [GetAsrVocab](#)
- [UpdateAsrVocab](#)

**修改接口：**

- [CreateRecTask](#)
  - 新增入参：HotwordId

- [SentenceRecognition](#)
  - 新增入参：HotwordId

新增数据结构：

- [HotWord](#)

## 第 2 次发布

发布时间：2019-09-06 08:30:06

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateRecTask](#)
- [DescribeTaskStatus](#)

新增数据结构：

- [Task](#)
- [TaskStatus](#)

## 第 1 次发布

发布时间：2019-06-21 19:59:11

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [SentenceRecognition](#)

# 简介

最近更新时间：2025-04-25 01:09:12

## 产品介绍

语音识别（Automatic Speech Recognition，ASR）为开发者提供语音转文字服务的最佳体验，开放实时语音识别、一句话识别和录音文件识别三种服务形式，满足不同类型开发者需求，具备识别准确率高、接入便捷、性能稳定等特点。

## 产品功能

### 实时语音识别

对实时音频流进行识别，达到“边说边出文字”的效果，可应用于语音输入、电话机器人等实时音频流场景

### 一句话识别

对60秒之内的短音频文件进行识别，达到快速准确识别较短语音的效果，可应用于语音消息转写等场景；

### 录音文件识别

对录音文件进行识别，达到识别较长的非实时语音的效果，可用于字幕生成、录音资料转写等场景

## 产品优势

### 海量数据积累

立足于腾讯庞大的社交数据平台，积累了数十万小时的语音标注数据，拥有丰富多样的语料库，为高识别率奠定数据基础。

### 算法业界领先

基于多种序列神经网络结构(LSTM、Attention Model、DeepCNN)，采用 Multitask 训练方法，结合T/S方式，在通用以及垂直领域有业内领先的识别精度。

### 支持多平台设备

提供 REST API 和 SDK，支持智能硬件、移动应用、网站、桌面客户端和物联网等多种设备终端。

### 支持语种丰富

现阶段支持中文普通话、英语、粤语和韩语语音识别，后续将陆续开放其他语种或方言的识别能力。

### 噪声环境识别佳

语音识别模型鲁棒性佳，识别精度高，抗噪声的干扰能力强，能够识别来自嘈杂环境的音频信息，不需要客户进行降噪处理。

### 海量内外部业务验证

经过微信、腾讯视频、王者荣耀等内部业务充分验证，也在大量互联网、金融、教育等领域的外部客户业务场景成功落地，日服务亿级用户，性能稳定。

## 应用场景

### 语音输入法

智能语音输入，由实时语音识别实现，为用户节省输入时间、提升输入体验。

### 语音消息转写

将用户的语音信息转成文字信息，由一句话识别服务实现，提升用户阅读效率。

### 字幕生成

将直播和录播视频中的语音转换为文字，由录音文件识别服务实现，轻松便捷地生成字幕文件。

### 会议纪要

将会议、庭审、采访等场景的音频信息转换为文字，由实时语音识别服务实现，降低人工记录成本、提升效率。

### 电话质检

将坐席通话转成文字，由实时语音识别服务或录音文件识别服务实现，全面覆盖质检内容、提升质检效率

# API 概览

最近更新时间：2024-09-23 01:08:08

## 语音流异步识别相关接口

| 接口名称                                          | 接口功能        | 频率限制（次/秒） |
|-----------------------------------------------|-------------|-----------|
| <a href="#">CloseAsyncRecognitionTask</a>     | 语音流异步识别任务关闭 | 20        |
| <a href="#">DescribeAsyncRecognitionTasks</a> | 语音流异步识别任务列表 | 20        |
| <a href="#">CreateAsyncRecognitionTask</a>    | 语音流异步识别任务创建 | 20        |

## 录音文件识别相关接口

| 接口名称                               | 接口功能       | 频率限制（次/秒） |
|------------------------------------|------------|-----------|
| <a href="#">DescribeTaskStatus</a> | 录音文件识别结果查询 | 50        |
| <a href="#">CreateRecTask</a>      | 录音文件识别请求   | 20        |

## 一句话识别相关接口

| 接口名称                                | 接口功能  | 频率限制（次/秒） |
|-------------------------------------|-------|-----------|
| <a href="#">SentenceRecognition</a> | 一句话识别 | -         |

## 自学习相关接口

| 接口名称                                     | 接口功能      | 频率限制（次/秒） |
|------------------------------------------|-----------|-----------|
| <a href="#">CreateCustomization</a>      | 创建自学习模型   | 20        |
| <a href="#">GetCustomizationList</a>     | 查询自学习模型列表 | 20        |
| <a href="#">ModifyCustomization</a>      | 更新自学习模型   | 20        |
| <a href="#">DeleteCustomization</a>      | 删除自学习模型   | 20        |
| <a href="#">ModifyCustomizationState</a> | 修改自学习模型状态 | 20        |
| <a href="#">DownloadCustomization</a>    | 下载自学习模型语料 | 20        |
| <a href="#">GetModelInfo</a>             | 获取自学习模型信息 | 20        |

## 热词相关接口

| 接口名称                             | 接口功能  | 频率限制（次/秒） |
|----------------------------------|-------|-----------|
| <a href="#">CreateAsrVocab</a>   | 创建热词表 | 20        |
| <a href="#">DeleteAsrVocab</a>   | 删除热词表 | 20        |
| <a href="#">DownloadAsrVocab</a> | 下载热词表 | 20        |

| 接口名称                            | 接口功能    | 频率限制（次/秒） |
|---------------------------------|---------|-----------|
| <a href="#">GetAsrVocab</a>     | 获取热词表   | 20        |
| <a href="#">GetAsrVocabList</a> | 列举热词表   | 20        |
| <a href="#">SetVocabState</a>   | 设置热词表状态 | 20        |
| <a href="#">UpdateAsrVocab</a>  | 更新热词表   | 20        |

## 说话人验证相关接口

| 接口名称                                  | 接口功能      | 频率限制（次/秒） |
|---------------------------------------|-----------|-----------|
| <a href="#">VoicePrintEnroll</a>      | 说话人注册     | 20        |
| <a href="#">VoicePrintVerify</a>      | 说话人认证     | 20        |
| <a href="#">VoicePrintUpdate</a>      | 说话人更新     | 20        |
| <a href="#">VoicePrintDelete</a>      | 说话人删除     | 20        |
| <a href="#">VoicePrintCompare</a>     | 说话人比对     | 20        |
| <a href="#">VoicePrintCount</a>       | 说话人注册数量统计 | 20        |
| <a href="#">VoicePrintGroupVerify</a> | 说话人验证1:N  | 20        |

## 关键字检索相关接口

| 接口名称                                 | 接口功能   | 频率限制（次/秒） |
|--------------------------------------|--------|-----------|
| <a href="#">CreateAsrKeyWordLib</a>  | 创建关键词表 | 20        |
| <a href="#">DeleteAsrKeyWordLib</a>  | 删除关键词表 | 20        |
| <a href="#">GetAsrKeyWordLibList</a> | 列举关键词表 | 20        |
| <a href="#">UpdateAsrKeyWordLib</a>  | 更新关键词表 | 20        |

## 业务用量相关接口

| 接口名称                           | 接口功能     | 频率限制（次/秒） |
|--------------------------------|----------|-----------|
| <a href="#">GetUsageByDate</a> | 通过日期查询用量 | 20        |

### ⚠ 注意：

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

# 调用方式

## 请求结构

最近更新时间：2025-05-21 01:08:34

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

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



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

## 2. 通信协议

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

## 3. 请求方法

支持的 HTTP 请求方法：

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

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

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

## 4. 字符编码

均使用 UTF-8 编码。

# 公共参数

最近更新时间：2024-11-15 01:09:56

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。 <b>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。</b>                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 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。 <b>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。</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 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：<br>TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request,<br>SignedHeaders=content-type;host,<br>Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024<br>其中，<br>- TC3-HMAC-SHA256：签名方法，目前固定取该值；<br>- Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 asr；tc3_request 为固定字符串；<br>- SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部；<br>- Signature：签名摘要，计算过程详见 <a href="#">文档</a> 。 |
| Token         | String  | 否  | HTTP 请求头：X-TC-Token。即 <a href="#">安全凭证服务</a> 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。                                                                                                                                                                                                                                                                                                                                                                                                    |
| Language      | String  | 否  | HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

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

HTTP GET 请求结构示例：

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

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

#### HTTP POST (application/json) 请求结构示例:

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

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

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

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

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

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

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

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

10
--58731222010402--
```

## 使用签名方法 v1 的公共参数

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

| 参数名称            | 类型      | 必选 | 描述                                                                                                                                      |
|-----------------|---------|----|-----------------------------------------------------------------------------------------------------------------------------------------|
| Action          | String  | 是  | 操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。                                                         |
| Region          | String  | -  | 地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 <b>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。</b>                           |
| 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 <a href="#">云API密钥</a> 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 <a href="#">文档</a> 。                                                                       |
| 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 <a href="#">安全凭证服务</a> 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。 |
| Language        | String  | 否  | 指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。                                                                              |

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

HTTP GET 请求结构示例：

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

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

HTTP POST 请求结构示例：

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

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

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

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

### 地域列表

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

| 地域       | 取值           |
|----------|--------------|
| 华南地区（广州） | ap-guangzhou |

# 签名方法 v3

最近更新时间：2024-12-25 01:10:56

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

推荐使用 API Explorer

</> 点击调试

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

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

## 为什么要进行签名

签名通过以下方式帮助保护请求：

### 1. 验证请求者的身份

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

### 2. 保护传输中的数据

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

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

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

## 申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

## 签名版本 v3 签名过程

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

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

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

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

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

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

下面详细解释签名计算过程。

## 1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

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

| 字段名称                 | 解释                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | <p>2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。</p> <p>此示例计算结果是 <code>content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n</code>。</p> <p>注意：<code>content-type</code> 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 <code>charset</code> 值，如果签名时和发送时不一致，服务器会返回签名校验失败。</p>                                                                             |
| SignedHeaders        | <p>参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。<code>content-type</code> 和 <code>host</code> 为必选头部。</p> <p>拼接规则：</p> <ol style="list-style-type: none"><li>1. 头部 key 统一转成小写；</li><li>2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号（;）分隔。</li></ol> <p>此示例为 <code>content-type;host;x-tc-action</code></p>                                                                        |
| HashedRequestPayload | <p>请求正文（payload，即 body，此示例为 <code>{"Limit": 1, "Filters": [{"Values": [{"\u672a\u547d\u540d"}], "Name": "instance-name"}]}</code>）的哈希值，计算伪代码为 <code>Lowercase(HexEncode(Hash.SHA256(RequestPayload)))</code>，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 <code>35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064</code>。</p> |

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

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

## 2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

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



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

注意：

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

根据以上规则，示例中得到的待签名字符串如下：

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

### 3. 计算签名

1) 计算派生签名密钥，伪代码如下：

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

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

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

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

2) 计算签名，伪代码如下：

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

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

## 4. 拼接 Authorization

按如下格式拼接 Authorization:

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

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

根据以上规则，示例中得到的值为：

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

最终完整的调用信息如下：

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

### ⚠ 注意：

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

打印实际请求内容，网络抓包等方式对比排查。

## 签名演示

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

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

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

- [Signature Demo](#)

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

### Java

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

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

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

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

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

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

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

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

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

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

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

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

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

## Python

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

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

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

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

# ***** 步骤 1: 拼接规范请求串 *****

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

# ***** 步骤 2: 拼接待签名字符串 *****

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

# ***** 步骤 3: 计算签名 *****

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

# ***** 步骤 4: 拼接 Authorization *****

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

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

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

## Golang

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

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

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

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

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

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



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

## PHP

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

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

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

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

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

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

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

## Ruby

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

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

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

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

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

puts canonical_request

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

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

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

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

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

## DotNet

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## NodeJS

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

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

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

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

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

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

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

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

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

```
console.log(canonicalRequest)

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

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

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

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

## C++

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



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

using namespace std;

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

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

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

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

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

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

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

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

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

return (ss.str());
}

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

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

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

string service = "cvm";
```

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

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

// ***** 步骤 2: 拼接待签名字符串 *****
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;

// ***** 步骤 3: 计算签名 *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;

// ***** 步骤 4: 拼接 Authorization *****
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;
```

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

## C

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

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

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

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

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

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

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

}

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

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

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

int main()
```

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

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

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

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

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

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

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

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

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

## 其他语言

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

## 签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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



# 签名方法

最近更新时间：2024-12-25 01:10:57

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

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

推荐使用 API Explorer

<> 点击调试

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

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

## 1. 申请安全凭证

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

安全凭证包括 SecretId 和 SecretKey：

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

## 2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

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

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

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

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

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

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

## 2.1. 对参数排序

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

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

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

## 2.2. 拼接请求字符串

此步骤生成请求字符串。

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

**注意：**“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

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

## 2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

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

4. 请求字符串: 即上一步生成的请求字符串。

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

示例的拼接结果为:

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

## 2.4. 生成签名串

此步骤生成签名串。

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

具体代码如下, 以 PHP 语言为例:

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

最终得到的签名串为:

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

## 3. 签名串编码

生成的签名串并不能直接作为请求参数, 需要对其进行 URL 编码。

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

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

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

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

## 4. 签名失败

根据实际情况, 存在以下签名失败的错误码, 请根据实际情况处理。

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

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

## 5. 签名演示

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

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

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

- [Signature Demo](#)

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Signature=7RAM2xfNM09EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12`。

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

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

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

### Java

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

public class TencentCloudAPIDemo {
```

```
private final static String CHARSET = "UTF-8";

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

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

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

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

## Python

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

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

import requests

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

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

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

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

## Golang

```
package main

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

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

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

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

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

```
}

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

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

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

## PHP

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

ksort($params);

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

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

## Ruby



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

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

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

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

## DotNet

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

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

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

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

        string endpoint = "cvm.tencentcloudapi.com";
        string region = "ap-guangzhou";
        string action = "DescribeInstances";
        string version = "2017-03-12";
        double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25, 此参数作为示例, 以实际为准
        // long timestamp = ToTimestamp() / 1000;
        // string requestTimestamp = timestamp.ToString();
        Dictionary<string, string> param = new Dictionary<string, string>();
        param.Add("Limit", "20");
        param.Add("Offset", "0");
        param.Add("InstanceIds.0", "ins-09dx96dg");
        param.Add("Action", action);
        param.Add("Nonce", "11886");
        // param.Add("Nonce", Math.Abs(new Random().Next()).ToString());
    }
}
```

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

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

## NodeJS

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

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

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

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

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

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

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

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

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

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

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

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

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

## 返回结果

最近更新时间：2024-03-12 01:09:40

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

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

### 正确返回结果

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

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

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

### 错误返回结果

若调用失败，其返回值示例如下为：

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

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

### 公共错误码

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

## 参数类型

最近更新时间：2022-08-10 06:14:26

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。

# 语音流异步识别相关接口

## 语音流异步识别任务关闭

最近更新时间：2025-04-25 01:09:09

### 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

本接口用于关闭语音流异步识别任务。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

### 2. 输入参数

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

| 参数名称    | 必选 | 类型      | 描述                                                     |
|---------|----|---------|--------------------------------------------------------|
| Action  | 是  | String  | <a href="#">公共参数</a> ，本接口取值：CloseAsyncRecognitionTask。 |
| Version | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                |
| Region  | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                        |
| TaskId  | 是  | Integer | 语音流异步识别任务的唯一标识，在创建任务时会返回<br>示例值：9000254964             |

### 3. 输出参数

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

### 4. 示例

#### 示例1 语音流异步识别任务关闭

##### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: CloseAsyncRecognitionTask
X-TC-Timestamp: 1599140162
```



```
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=0615f73a69c6b054affd69e4b1cbb68fbe84ae8024a9347df4aa5054933adac8
```

<公共请求参数>

```
{
  "TaskId": 1013279283
}
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "fabcd63-a1b7-40a0-b4c3-640f78974919"
  }
}
```

## 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.NoSuchTask | 错误的TaskId。 |

| 错误码                                    | 描述       |
|----------------------------------------|----------|
| InternalServerError.FailAccessDatabase | 访问数据库失败。 |
| InvalidParameterValue                  | 参数取值错误。  |
| MissingParameter                       | 缺少参数错误。  |

# 语音流异步识别任务列表

最近更新时间：2025-04-25 01:09:08

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

本接口用于查询当前在运行的语音流异步识别任务列表。

• 签名方法参考 [公共参数](#) 中签名方法v3。

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 | <a href="#">公共参数</a> ，本接口取值：DescribeAsyncRecognitionTasks。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                    |
| Region  | 否  | String | <a href="#">公共参数</a> ，此参数为可选参数。                            |

## 3. 输出参数

| 参数名称      | 类型                                    | 描述                                                                                     |
|-----------|---------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">AsyncRecognitionTasks</a> | 任务列表<br>注意：此字段可能返回 null，表示取不到有效值。                                                      |
| RequestId | String                                | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 查询运行中的任务列表

#### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: DescribeAsyncRecognitionTasks
X-TC-Timestamp: 1599140162
```

```
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=0615f73a69c6b054affd69e4b1cbb68fbe84ae8024a9347df4aa5054933adac8
```

<公共请求参数>

```
{}
```

## 输出示例

```
{
  "Response": {
    "RequestId": "8824366f-0e8f-4bd4-8924-af5e84127caa",
    "Data": {
      "Tasks": [
        {
          "TaskId": 9266418,
          "Url": ""
        }
      ]
    }
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

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

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                                    | 描述       |
|----------------------------------------|----------|
| InternalServerError.FailAccessDatabase | 访问数据库失败。 |

# 语音流异步识别任务创建

最近更新时间：2025-05-09 01:09:53

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

本接口用于对语音流进行准实时识别，通过异步回调来返回识别结果。适用于直播审核等场景。

- 支持rtmp、rtsp等流媒体协议，以及各类基于http协议的直播流(不支持hls)
- 音频流时长无限制，服务会自动拉取音频流数据，若连续10分钟拉不到流数据时，服务会终止识别任务
- 服务通过回调的方式来提供识别结果，用户需要提供CallbackUrl。回调时机为一小段话(最长15秒)回调一次。
- 签名方法参考 [公共参数](#) 中签名方法v3。
- 默认单账号限制并发数为20路，如您有提高并发限制的需求，请提[工单](#)进行咨询。

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 | <a href="#">公共参数</a> ，本接口取值：CreateAsyncRecognitionTask。                                                                                                                                                                                                     |
| Version    | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                                                                                                                                                     |
| Region     | 否  | String | <a href="#">公共参数</a> ，此参数为可选参数。                                                                                                                                                                                                                             |
| EngineType | 是  | String | 引擎模型类型。<br>• 16k_zh：中文普通话通用；<br>• 16k_en：英语；<br>• 16k_yue：粤语；<br>• 16k_id：印度尼西亚语；<br>• 16k_fil：菲律宾语；<br>• 16k_th：泰语；<br>• 16k_pt：葡萄牙语；<br>• 16k_tr：土耳其语；<br>• 16k_ar：阿拉伯语；<br>• 16k_es：西班牙语；<br>• 16k_hi：印地语；<br>• 16k_fr：法语；<br>• 16k_de：德语；<br>示例值：16k_zh |

| 参数名称           | 必选 | 类型      | 描述                                                                                                                   |
|----------------|----|---------|----------------------------------------------------------------------------------------------------------------------|
| Url            | 是  | String  | 语音流地址，支持rtmp、rtsp等流媒体协议，以及各类基于http协议的直播流(不支持hls, m3u8)<br>示例值：https://www.audio.com/audio.wav                        |
| CallbackUrl    | 是  | String  | 支持HTTP和HTTPS协议，用于接收识别结果，您需要自行搭建公网可调用的服务。回调格式&内容详见： <a href="#">语音流异步识别回调说明</a><br>示例值：https://www.audio.com/callback |
| SignToken      | 否  | String  | 用于生成回调通知中的签名<br>示例值：xadjjfshjask**lksadlakd                                                                          |
| FilterDirty    | 否  | Integer | 是否过滤脏词（目前支持中文普通话引擎）。0：不过滤脏词；1：过滤脏词；2：将脏词替换为 *。<br>默认值为 0<br>示例值：0                                                    |
| FilterModal    | 否  | Integer | 是否过滤语气词（目前支持中文普通话引擎）。0：不过滤语气词；1：部分过滤；2：严格过滤。<br>默认值为 0<br>示例值：0                                                      |
| FilterPunc     | 否  | Integer | 是否过滤标点符号（目前支持中文普通话引擎）。0：不过滤，1：过滤句末标点，2：过滤所有标点。<br>默认0<br>示例值：0                                                       |
| ConvertNumMode | 否  | Integer | 是否进行阿拉伯数字智能转换。0：不转换，直接输出中文数字，1：根据场景智能转换为阿拉伯数字。<br>默认值为1<br>示例值：1                                                     |
| WordInfo       | 否  | Integer | 是否显示词级别时间戳。0：不显示；1：显示，不包含标点时间戳，2：显示，包含标点时间戳。<br>默认为0<br>示例值：0                                                        |
| HotwordId      | 否  | String  | 热词id。用于调用对应的热词表，如果在调用语音识别服务时，不进行单独的热词id设置，自动生效默认热词；如果进行了单独的热词id设置，那么将生效单独设置的热词id。<br>示例值：sifhsadhuasjh*djsahdsg      |
| AudioData      | 否  | Boolean | 回调数据中，是否需要对应音频数据。<br>示例值：false                                                                                       |

### 3. 输出参数

| 参数名称      | 类型                   | 描述                                                                                    |
|-----------|----------------------|---------------------------------------------------------------------------------------|
| Data      | <a href="#">Task</a> | 请求返回结果，包含本次的任务ID(TaskId)                                                              |
| RequestId | String               | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。 |

### 4. 示例

#### 示例1 语音流异步识别任务创建

创建一个异步识别任务，通过接口返回拿到任务ID

##### 输入示例

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

```
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: CreateAsyncRecognitionTask
X-TC-Timestamp: 1599140162
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=0615f73a69c6b054affd69e4b1cbb68fbe84ae8024a9347df4aa5054933adac8
```

<公共请求参数>

```
{
  "EngineType": "16k_zh",
  "Url": "rtmp://test.com/test_stream",
  "CallbackUrl": "http://test.com/callback",
  "SignToken": ""
}
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "fabcd63-a1b7-40a0-b4c3-640f78974919",
    "Data": {
      "TaskId": 1000000007
    }
  }
}
```

## 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.ServiceIsolate      | 账号因为欠费停止服务，请在腾讯云账户充值。     |
| FailedOperation.UserHasNoAmount     | 资源包耗尽，请购买资源包或开通后付费        |
| FailedOperation.UserHasNoFreeAmount | 资源包耗尽，请开通后付费或者购买资源包       |
| FailedOperation.UserNotRegistered   | 服务未开通，请在腾讯云官网语音识别控制台开通服务。 |
| InternalError.FailAccessDatabase    | 访问数据库失败。                  |
| InvalidParameter                    | 参数错误。                     |
| InvalidParameterValue               | 参数取值错误。                   |
| MissingParameter                    | 缺少参数错误。                   |
| RequestLimitExceeded                | 请求的次数超过了频率限制。             |
| UnknownParameter                    | 未知参数错误。                   |

# 录音文件识别相关接口

## 录音文件识别结果查询

最近更新时间：2025-04-25 01:09:11

### 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

调用录音文件识别请求接口后，有回调和轮询两种方式获取识别结果。

- 注意任务有效期为24小时，超过24小时的任务请不要再查询，且不要依赖TaskId作为业务唯一ID，不同日期可能出现重复TaskId。
- 当采用回调方式时，识别完成后会将结果通过 POST 请求的形式通知到用户在请求时填写的回调 URL，具体请参见 [录音识别结果回调](#)。
- 当采用轮询方式时，需要主动提交任务ID来轮询识别结果，共有任务成功、等待、执行中和失败四种结果，具体信息请参见下文说明。
- 请求方法为 HTTP POST，Content-Type为"application/json; charset=utf-8"
- 签名方法参考 [公共参数](#) 中签名方法v3。
- 默认接口请求频率限制：50次/秒，如您有提高请求频率限制的需求，请提[工单](#)进行咨询。

默认接口请求频率限制：50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  | <a href="#">公共参数</a> ，本接口取值：DescribeTaskStatus。                                                    |
| Version | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                            |
| Region  | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                                                                    |
| TaskId  | 是  | Integer | 从CreateRecTask接口获取的TaskId，用于获取任务状态与结果。<br>注意：TaskId有效期为24小时，超过24小时的TaskId请不要再查询。<br>示例值：4500055927 |

### 3. 输出参数

| 参数名称      | 类型                         | 描述                                                                                    |
|-----------|----------------------------|---------------------------------------------------------------------------------------|
| Data      | <a href="#">TaskStatus</a> | 录音文件识别的请求返回结果。                                                                        |
| RequestId | String                     | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。 |

### 4. 示例

#### 示例1 轮询结果——任务失败

用户通过轮询方式获取识别结果，任务失败，具体原因请用户查看返回中ErrorMsg提示。

## 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: DescribeTaskStatus
X-TC-Timestamp: 1599142563
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=6647fa852c9329ac5aa355c68c838358139c2d941f4ed175dd5858978f5d79c8
Content-Length: 21
<公共请求参数>

{
  "TaskId": 522931820
}
```

## 输出示例

```
{
  "Response": {
    "RequestId": "8824366f-0e8f-4bd4-8924-af5e84127caa",
    "Data": {
      "TaskId": 522931820,
      "Status": 3,
      "StatusStr": "failed",
      "AudioDuration": 0,
      "Result": "",
      "ErrorMsg": "Failed to download audio file!",
      "ResultDetail": []
    }
  }
}
```

## 示例2 轮询结果——任务成功

用户通过轮询方式获取识别结果，任务成功，并返回识别结果。

## 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: DescribeTaskStatus
X-TC-Timestamp: 1599142563
Authorization: TC3-HMAC-SHA256 Credential=*****/2020
```

```
-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=6647fa852c9329ac5aa355c68c838358139c2d94
1f4ed175dd5858978f5d79c8
Content-Length: 21
<公共请求参数>

{
  "TaskId": 522931820
}
```

## 输出示例

```
{
  "Response": {
    "RequestId": "a73b14a6-5044-41cb-bf32-e735d5bd69de",
    "Data": {
      "TaskId": 9266418,
      "Status": 2,
      "StatusStr": "success",
      "AudioDuration": 2.38,
      "Result": "[0:0.020,0:2.380] 腾讯云语音识别欢迎您。\\n",
      "ResultDetail": [
        {
          "FinalSentence": "腾讯云语音识别欢迎您。",
          "SliceSentence": "腾讯云 语音识别 欢迎 您",
          "StartMs": 20,
          "EndMs": 2380,
          "SpeechSpeed": 5.9,
          "WordsNum": 4,
          "EmotionalEnergy": 0,
          "SpeakerId": 0,
          "SilenceTime": 0,
          "EmotionType": [
            "happy"
          ],
          "Words": [
            {
              "OffsetStartMs": 120,
              "OffsetEndMs": 780,
              "Word": "腾讯云"
            },
            {
              "OffsetStartMs": 780,
              "OffsetEndMs": 1530,
              "Word": "语音识别"
            },
            {
              "OffsetStartMs": 1530,
              "OffsetEndMs": 1860,
              "Word": "欢迎"
            }
          ]
        }
      ]
    }
  }
}
```

```
},
{
  "OffsetStartMs": 1860,
  "OffsetEndMs": 2250,
  "Word": "您"
}
]
}
],
"ErrorMsg": ""
}
}
}
```

### 示例3 轮询结果——任务等待

用户通过轮询方式获取识别结果，任务等待，说明任务还在识别队列中，请用户耐心等待。

#### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: DescribeTaskStatus
X-TC-Timestamp: 1599142563
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=6647fa852c9329ac5aa355c68c838358139c2d941f4ed175dd5858978f5d79c8
Content-Length: 21
<公共请求参数>

{
  "TaskId": 522931820
}
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "8824366f-0e8f-4bd4-8924-af5e84127caa",
    "Data": {
      "TaskId": 522931820,
      "Status": 0,
      "StatusStr": "waiting",
      "AudioDuration": 0,
      "Result": "",
      "ErrorMsg": "",
      "ResultDetail": []
    }
  }
}
```

```
}  
  
}  
  
}
```

#### 示例4 轮询结果——任务执行中

用户通过轮询方式获取识别结果，任务执行中，说明任务已经开始识别，请用户耐心等待。

##### 输入示例

```
POST / HTTP/1.1  
Host: asr.tencentcloudapi.com  
Content-Type: application/json; charset=utf-8  
X-TC-Version: 2019-06-14  
X-TC-Region: ap-shanghai  
X-TC-Action: DescribeTaskStatus  
X-TC-Timestamp: 1599142563  
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=6647fa852c9329ac5aa355c68c838358139c2d941f4ed175dd5858978f5d79c8  
Content-Length: 21  
<公共请求参数>  
  
{  
  "TaskId": 522931820  
}
```

##### 输出示例

```
{  
  "Response": {  
    "RequestId": "8824366f-0e8f-4bd4-8924-af5e84127caa",  
    "Data": {  
      "TaskId": 522931820,  
      "Status": 1,  
      "StatusStr": "doing",  
      "AudioDuration": 0,  
      "Result": "",  
      "ErrorMsg": "",  
      "ResultDetail": []  
    }  
  }  
}
```

## 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.ErrorDownFile          | 下载音频文件失败。             |
| FailedOperation.ErrorRecognize         | 识别失败。                 |
| FailedOperation.NoSuchTask             | 错误的TaskId。            |
| FailedOperation.ServiceIsolate         | 账号因为欠费停止服务，请在腾讯云账户充值。 |
| FailedOperation.UserHasNoFreeAmount    | 资源包耗尽，请开通后付费或者购买资源包   |
| InternalServerError.FailAccessDatabase | 访问数据库失败。              |
| InternalServerError.FailAccessRedis    | 访问Redis失败。            |
| InvalidParameter                       | 参数错误。                 |
| MissingParameter                       | 缺少参数错误。               |
| UnknownParameter                       | 未知参数错误。               |

# 录音文件识别请求

最近更新时间：2025-06-23 01:09:50

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

- 本接口可对较长的录音文件进行识别。如希望直接使用带界面的语音识别产品，请访问[产品体验中心](#)。产品计费标准请查阅 [计费概述（在线版）](#)
- 接口默认限频：20次/秒。此处仅限制任务提交频次，与识别结果返回时效无关
  - 返回时效：异步回调，非实时返回。最长3小时返回识别结果，**大多数情况下，1小时的音频1-3分钟即可完成识别**。请注意：上述返回时长不含音频下载时延，且30分钟内发送超过1000小时录音或2万条任务的情况除外
  - 音频格式：wav、mp3、m4a、flv、mp4、wma、3gp、amr、aac、ogg-opus、flac
  - 支持语言：在本页面上搜索 `EngineModelType`，或前往 [产品功能](#) 查看
  - 音频提交方式：本接口支持[音频 URL](#)、[本地音频文件](#)两种请求方式。推荐使用 [腾讯云COS](#) 来存储、生成URL并提交任务，此种方式将不产生外网和流量下行费用，可节约成本、提升任务速度（可参考COS预签名指南：[使用预签名 URL 访问 COS](#)，获取COS预签名url）
  - 音频限制：音频 URL 时长不能大于5小时，文件大小不超过1GB；本地音频文件不能大于5MB
  - 如何获取识别结果：支持[回调](#)或[轮询](#)的方式获取结果，具体请参考 [录音文件识别结果查询](#)
  - 识别结果有效时间：识别结果在服务端保存24小时
  - 签名方法参考 [公共参数](#) 中签名方法 v3

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 | <a href="#">公共参数</a> ，本接口取值：CreateRecTask。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Version         | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Region          | 否  | String | <a href="#">公共参数</a> ，此参数为可选参数。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| EngineModelType | 是  | String | <div>引擎模型类型</div> <div>识别引擎采用分级计费方案，标记为“大模型版”的引擎适用大模型计费方案，<a href="#">点击这里</a> 查看产品计费说明</div> <div>电话通讯场景引擎：</div> <div><b>注意：电话通讯场景，请务必使用以下8k引擎</b></div> <div><ul style="list-style-type: none"><li>• 8k_zh：中文电话通讯；</li><li>• 8k_en：英文电话通讯；</li><li>• 8k_zh_large：中文电话场景专用大模型引擎。通过显著提升模型参数规模与语言建模能力，实现对电话音频中复杂场景（如口音干扰、背景噪声）的高精度识别，识别准确率较常规版本大幅提升。<a href="#">点击这里</a> 对比常规版本与普方大模型效果；</li></ul></div> <div>注意：如您有电话通讯场景识别需求，但发现需求语种仅支持16k，可将8k音频传入下方16k引擎，亦能获得识别结果。但16k引擎并非基于电话通讯数据训练，无法承诺此种调用方式的识别效果，需由您自行验证识别结果是否可用</div> |



| 参数名称          | 必选 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|----|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |    |         | <p>通用场景引擎：</p> <p><b>注意：除电话通讯场景以外的其它识别场景，请务必使用以下16k引擎</b></p> <ul style="list-style-type: none"> <li>• <b>16k_zh</b>：中文普通话通用引擎，支持中文普通话和少量英语，使用丰富的中文普通话语料训练，覆盖场景广泛，适用于除电话通讯外的所有中文普通话识别场景；</li> <li>• <b>16k_zh_large</b>：普方英大模型引擎【大模型版】。当前模型同时支持中文、英文、<b>多种中文方言</b>等语言的识别，模型参数量极大，语言模型性能增强，针对噪声大、回音大、人声小、人声远等低质量音频的识别准确率极大提升，<a href="#">点击这里</a> 对比中文普通话常规版本与普方英大模型版本的识别效果；</li> <li>• <b>16k_multi_lang</b>：多语种大模型引擎【大模型版】。当前模型同时支持英语、日语、韩语、阿拉伯语、菲律宾语、法语、印地语、印尼语、马来语、葡萄牙语、西班牙语、泰语、土耳其语、越南语、德语的识别，可实现15个语种的自动识别(句子/段落级别)；</li> <li>• <b>16k_zh_dialect</b>：中文普通话+多方言混合引擎，除普通话外支持23种方言（上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话）；</li> <li>• <b>16k_zh_en</b>：中英大模型引擎【大模型版】。当前模型同时支持中文、英语识别，模型参数量极大，语言模型性能增强，针对噪声大、回音大、人声小、人声远等低质量音频的识别准确率极大提升；</li> <li>• <b>16k_en</b>：英语；</li> <li>• <b>16k_yue</b>：粤语；</li> <li>• <b>16k_zh-PY</b>：中英粤混合引擎，使用一个引擎同时识别中文普通话、英语、粤语三个语言；</li> <li>• <b>16k_zh-TW</b>：中文繁体；</li> <li>• <b>16k_ja</b>：日语；</li> <li>• <b>16k_ko</b>：韩语；</li> <li>• <b>16k_vi</b>：越南语；</li> <li>• <b>16k_ms</b>：马来语；</li> <li>• <b>16k_id</b>：印度尼西亚语；</li> <li>• <b>16k_fil</b>：菲律宾语；</li> <li>• <b>16k_th</b>：泰语；</li> <li>• <b>16k_pt</b>：葡萄牙语；</li> <li>• <b>16k_tr</b>：土耳其语；</li> <li>• <b>16k_ar</b>：阿拉伯语；</li> <li>• <b>16k_es</b>：西班牙语；</li> <li>• <b>16k_hi</b>：印地语；</li> <li>• <b>16k_fr</b>：法语；</li> <li>• <b>16k_zh_medical</b>：中文医疗引擎；</li> <li>• <b>16k_de</b>：德语；</li> </ul> <p>示例值：16k_zh</p> |
| ChannelNum    | 是  | Integer | <p>识别声道数</p> <p>1：单声道（16k音频仅支持单声道，<b>请勿</b>设置为双声道）；</p> <p>2：双声道（仅支持8k电话音频，且双声道应分别为通话双方）</p> <p><b>注意：</b></p> <ul style="list-style-type: none"> <li>• 16k音频：仅支持单声道识别，<b>需设置ChannelNum=1</b>；</li> <li>• 8k电话音频：支持单声道、双声道识别，<b>建议设置ChannelNum=2，即双声道</b>。双声道能够物理区分说话人、避免说话双方重叠产生的识别错误，能达到最好的说话人分离效果和识别效果。设置双声道后，将自动区分说话人，因此<b>无需再开启说话人分离功能</b>，相关参数（<b>SpeakerDiarization</b>、<b>SpeakerNumber</b>）使用默认值即可，返回的ResultDetail中的speakerId的值为0代表左声道，值为1代表右声道。</li> </ul> <p>示例值：1</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| ResTextFormat | 是  | Integer | <p>识别结果返回样式</p> <p>0：基础识别结果（仅包含有效人声时间戳，无词粒度的<a href="#">详细识别结果</a>）；</p> <p>1：基础识别结果之上，增加词粒度的<a href="#">详细识别结果</a>（包含词级别时间戳、语速值，<b>不含标点</b>）；</p> <p>2：基础识别结果之上，增加词粒度的<a href="#">详细识别结果</a>（包含词级别时间戳、语速值和标点）；</p> <p>3：基础识别结果之上，增加词粒度的<a href="#">详细识别结果</a>（包含词级别时间戳、语速值和标点），且识别结果按标点符号分段，<b>适用字幕场景</b>；</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| 参数名称               | 必选 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|----|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |    |         | <p>4：【<b>增值付费功能</b>】基础识别结果之上，增加词粒度的<b>详细识别结果</b>（包含词级别时间戳、语速值和标点），且识别结果按nlp语义分段，<b>适用会议、庭审记录转写等场景</b>，仅支持8k_zh/16k_zh引擎</p> <p>5：【<b>增值付费功能</b>】基础识别结果之上，增加词粒度的<b>详细识别结果</b>（包含词级别时间戳、语速值和标点），并输出口语转书面语转写结果，该结果去除语气词、重复词、精简冗余表达，并修正发言人口误，实现口语转书面语的效果，<b>适用于线上、线下会议直接总结为书面会议纪要的场景</b>，仅支持8k_zh/16k_zh引擎</p> <p>注意：<br/>如果传入参数值4，需确保账号已购买<b>语义分段资源包</b>，或账号开启后付费；若当前账号已开启<b>后付费功能</b>，并传入参数值4，将<b>自动计费</b><br/>如果传入参数值5，需确保账号已购买<b>口语转书面语资源包</b>，或账号开启后付费；若当前账号已开启<b>后付费功能</b>，并传入参数值5，将自动计费<b>自动计费</b><br/>示例值：0</p> |
| SourceType         | 是  | Integer | <p>音频数据来源<br/>0：音频URL；<br/>1：音频数据（post body）<br/>示例值：0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Data               | 否  | String  | <p>音频数据base64编码<br/><b>当 SourceType 值为 1 时须填写该字段，为 0 时不需要填写</b></p> <p>注意：音频数据要小于5MB（含）<br/>示例值：UklGRka1CQBxQVZFZm10zOjQzAABJU0ZUDgAAAE</p>                                                                                                                                                                                                                                                                                                                                                                         |
| DataLen            | 否  | Integer | <p>数据长度（此数据长度为数据未进行base64编码时的长度）<br/>示例值：50</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Url                | 否  | String  | <p>音频URL的地址（需要公网环境浏览器可下载）<br/><b>当 SourceType 值为 0 时须填写该字段，为 1 时不需要填写</b></p> <p>注意：<br/>1. 请确保录音文件时长在5个小时（含）之内，否则可能识别失败；<br/>2. 请保证文件的下载速度，否则可能下载失败<br/>示例值：https://audio.cos.ap-guangzhou.myqcloud.com/example.wav</p>                                                                                                                                                                                                                                                                                            |
| CallbackUrl        | 否  | String  | <p>回调 URL<br/>用户自行搭建的用于接收识别结果的服务URL<br/>回调格式和内容详见：<a href="#">录音识别回调说明</a></p> <p>注意：<br/>- 如果用户使用轮询方式获取识别结果，则无需提交该参数<br/>- 建议在回调URL中带上您的业务ID等信息，以便处理业务逻辑<br/>示例值：https://xxxx.xxx.xxx/callback</p>                                                                                                                                                                                                                                                                                                                 |
| SpeakerDiarization | 否  | Integer | <p>是否开启说话人分离<br/>0：不开启；<br/>1：开启（仅支持以下引擎：<br/>8k_zh/16k_zh/16k_ms/16k_en/16k_id/16k_zh_large/16k_zh_dialect，且 ChannelNum=1时可用）；<br/>默认值为 0</p> <p>注意：<br/>8k双声道电话音频请按 <b>ChannelNum 识别声道数</b> 的参数描述使用默认值<br/>示例值：0</p>                                                                                                                                                                                                                                                                                              |
| SpeakerNumber      | 否  | Integer | <p>说话人分离人数<br/><b>需配合开启说话人分离使用，不开启无效</b>，取值范围：0-10</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| 参数名称               | 必选 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|----|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |    |         | 0：自动分离（最多分离出20个人）；<br>1-10：指定人数分离；<br>默认值为 0<br>示例值：0                                                                                                                                                                                                                                                                                                                           |
| HotwordId          | 否  | String  | 热词表id<br>如不设置该参数，将自动生效默认热词表；<br>如设置该参数，将生效对应id的热词表；<br>点击 <a href="#">这里查看热词表配置方法</a><br>示例值：769762aexxxxxxxxxxec3cda3142e                                                                                                                                                                                                                                                      |
| CustomizationId    | 否  | String  | 自学习定制模型 id<br>如设置了该参数，将生效对应id的自学习定制模型；<br>点击 <a href="#">这里查看自学习定制模型配置方法</a><br>示例值：769762aexxxxxxxxxxec3cda3142e                                                                                                                                                                                                                                                               |
| EmotionRecognition | 否  | Integer | <b>【增值付费功能】</b> 情绪识别能力（目前仅支持16k_zh,8k_zh）<br>0：不开启；<br>1：开启情绪识别，但不在文本展示情绪标签；<br>2：开启情绪识别，并且在文本展示情绪标签（ <b>该功能需要设置ResTextFormat 大于0</b> ）<br>默认值为0<br>支持的情绪分类为：高兴、伤心、愤怒<br><br>注意：<br>1. <b>本功能为增值服务</b> ，需将参数设置为1或2时方可按对应方式生效；<br>2. 如果传入参数值1或2，需确保账号已购买 <a href="#">情绪识别资源包</a> ，或账号开启后付费； <b>若当前账号已开启后付费功能，并传入参数值1或2，将自动计费</b> ）；<br>3. 参数设置为0时，无需购买资源包，也不会消耗情绪识别对应资源<br>示例值：0 |
| EmotionalEnergy    | 否  | Integer | 情绪能量值<br>取值为音量分贝值/10，取值范围：[1,10]，值越高情绪越强烈<br>0：不开启；<br>1：开启；<br>默认值为0<br>示例值：0                                                                                                                                                                                                                                                                                                  |
| ConvertNumMode     | 否  | Integer | 阿拉伯数字智能转换（目前支持中文普通话引擎）<br>0：不转换，直接输出中文数字；<br>1：根据场景智能转换为阿拉伯数字；<br>3：打开数学相关数字转换（如：阿尔法转写为 $\alpha$ ）；<br>默认值为 1<br>示例值：0                                                                                                                                                                                                                                                          |
| FilterDirty        | 否  | Integer | 脏词过滤（目前支持中文普通话引擎）<br>0：不过滤脏词；<br>1：过滤脏词；<br>2：将脏词替换为 * ；<br>默认值为 0<br>示例值：0                                                                                                                                                                                                                                                                                                     |
| FilterPunc         | 否  | Integer | 标点符号过滤（目前支持中文普通话引擎）<br>0：不过滤标点；<br>1：过滤句末标点；<br>2：过滤所有标点；<br>默认值为 0<br>示例值：0                                                                                                                                                                                                                                                                                                    |
| FilterModal        | 否  | Integer | 语气词过滤（目前支持中文普通话引擎）<br>0：不过滤语气词；                                                                                                                                                                                                                                                                                                                                                 |

| 参数名称               | 必选 | 类型              | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|----|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |    |                 | 1: 过滤部分语气词;<br>2: 严格过滤语气词;<br>默认值为 0<br>示例值: 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| SentenceMaxLength  | 否  | Integer         | 单标点最多字数（目前支持中文普通话引擎）<br>可控制单行字幕最大字数，适用于字幕生成场景，取值范围：[6, 40]<br>0: 不开启该功能；<br>默认值为0<br><br>注意：需设置ResTextFormat为3，解析返回的ResultDetail列表，通过结构中FinalSentence获取单个标点断句结果<br>示例值：0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Extra              | 否  | String          | 附加参数（该参数无意义，忽略即可）<br>示例值：无意义参数                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| HotwordList        | 否  | String          | 临时热词表：该参数用于提升识别准确率。<br><br>- 单个热词限制："热词 权重"，单个热词不超过30个字符（最多10个汉字），权重[1-11]或者100，如：“腾讯云 5”或“ASR 11”；<br><br>- 临时热词表限制：多个热词用英文逗号分割，最多支持128个热词，如：“腾讯云 10,语音识别 5,ASR 11”；<br><br>- 参数 hotword_id（热词表）与 hotword_list（临时热词表）区别：<br><br>- hotword_id：热词表。需要先在控制台或接口创建热词表，获得对应hotword_id传入参数来使用热词功能；<br><br>- hotword_list：临时热词表。每次请求时直接传入临时热词表来使用热词功能，云端不保留临时热词表。适用于有极大量热词需求的用户；<br><br>注意：<br><br>- 如果同时传入了 hotword_id 和 hotword_list，只有hotword_list 生效；<br><br>- 热词权重设置为11时，当前热词将升级为超级热词，建议仅将重要且必须生效的热词设置到11，设置过多权重为11的热词将影响整体字准确率。<br><br>- 热词权重设置为100时，当前热词开启热词增强同音替换功能（仅支持8k_zh,16k_zh），举例：热词配置“蜜制 100”时，与“蜜制”同拼音（mizhi）的“秘制”的识别结果会被强制替换成“蜜制”。因此建议客户根据自己的实际情况开启该功能。建议仅将重要且必须生效的热词设置到100，设置过多权重为100的热词将影响整体字准确率。<br>示例值：腾讯云 10,语音识别 5,ASR 11 |
| KeyWordLibIdList.N | 否  | Array of String | 关键词识别ID列表，默认空为不进行识别，最多10个<br><br>示例值：["sdsdasdasf","asdfasdasd"]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ReplaceTextId      | 否  | String          | 替换词汇总表id，适用于热词和自学习场景也无法解决的极端case词组，会对识别结果强制替换。具体可参考 <a href="#">配置控制台</a> ；强制替换功能可能会影响正常识别结果，请谨慎使用<br><br>注意：<br>1. 本功能配置完成后，预计在10分钟后生效<br><br>示例值：26910027**d9fa530f9f39dcd35d8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

### 3. 输出参数

| 参数名称      | 类型     | 描述                                                                                                    |
|-----------|--------|-------------------------------------------------------------------------------------------------------|
| Data      | Task   | 录音文件识别的请求返回结果，包含结果查询需要的TaskId。<br><b>注意：TaskId有效期为24小时，不同日期可能出现重复TaskId，请不要依赖TaskId作为您业务系统里的唯一ID。</b> |
| RequestId | String | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。                 |

### 4. 示例

#### 示例1 通过音频Url来调用接口

用户通过音频Url的方式（SourceType为0）请求录音识别服务，请求模型为16k中文（EngineModelType = 16k\_zh），音频格式为wav（采样率为16k，单声道）

##### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: CreateRecTask
X-TC-Timestamp: 1599142560
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=524ed61a4a71de417f4fa41249dcb428a0c51013890f24f3492068ca7cd16953
<公共请求参数>
```

```
{
  "Url": "http://test.cos.ap-guangzhou.myqcloud.com/test.wav",
  "ChannelNum": 1,
  "EngineModelType": "16k_zh",
  "ResTextFormat": 0,
  "SourceType": 0
}
```

##### 输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03",
    "Data": {
      "TaskId": 1393265
    }
  }
}
```

#### 示例2 通过音频数据来调用接口

用户通过上传音频数据（Data）的方式（SourceType为1）请求录音识别服务，请求模型为16k中文（EngineModelType = 16k\_zh），音频格式为wav（采样率为16k，单声道）

#### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: CreateRecTask
X-TC-Timestamp: 1599142560
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=524ed61a4a71de417f4fa41249dcb428a0c51013890f24f3492068ca7cd16953
```

<公共请求参数>

```
{
  "ChannelNum": 1,
  "EngineModelType": "16k_zh",
  "ResTextFormat": 0,
  "Data": "eGNmYXNkZmFzZmFzZGZhc2RmCg==",
  "SourceType": 1
}
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03",
    "Data": {
      "TaskId": 1396665
    }
  }
}
```

## 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.InvalidAuthorization       | 鉴权错误。                     |
| FailedOperation.CheckAuthInfoFailed    | 鉴权错误。                     |
| FailedOperation.ErrorDownFile          | 下载音频文件失败。                 |
| FailedOperation.ErrorRecognize         | 识别失败。                     |
| FailedOperation.ServiceIsolate         | 账号因为欠费停止服务，请在腾讯云账户充值。     |
| FailedOperation.UserHasNoAmount        | 资源包耗尽，请购买资源包或开通后付费        |
| FailedOperation.UserHasNoFreeAmount    | 资源包耗尽，请开通后付费或者购买资源包       |
| FailedOperation.UserNotRegistered      | 服务未开通，请在腾讯云官网语音识别控制台开通服务。 |
| InternalServerError.ErrorDownFile      | 下载音频文件失败。                 |
| InternalServerError.FailAccessDatabase | 访问数据库失败。                  |
| InternalServerError.FailAccessRedis    | 访问Redis失败。                |
| InvalidParameter                       | 参数错误。                     |
| InvalidParameterValue                  | 参数取值错误。                   |
| MissingParameter                       | 缺少参数错误。                   |
| RequestLimitExceeded.UinLimitExceeded  | 超出请求频率。                   |
| UnknownParameter                       | 未知参数错误。                   |

# 录音文件识别回调说明

最近更新时间：2025-03-31 11:13:30

## 1. 回调说明

录音识别请求中，如果用户设置了 CallbackUrl 参数，则通过回调的方式来返回识别结果，用户需要自行搭建可公网访问的 HTTP 或者 HTTPS 服务，并在创建录音识别任务时，将回调 Url 填写到 CallbackUrl 中。回调时，会使用 HTTP 的 POST 方法，将所有内容会放入 Body 中，Content-Type 为 application/x-www-form-urlencoded 。

## 2. 回调 Body 说明

### 2.1 回调 Body 示例

```
code=0&requestId=4000048858&appid=12516****&projectid=0&text=%5B0%3A1.640%2C0%3A4.600%5D++%E6%88%91%E5%9C%A8%E9%A9%AC%E4%B8%8A%E4%B8%8A%E5%A4%9C%E7%8F%AD%E3%80%82%0A%5B0%3A5.420%2C0%3A7.820%5D++%E6%98%8E%E5%A4%A9%E6%97%A9%E4%B8%8A%E8%AF%B4%E5%93%88%E3%80%82%0A&audioTime=8.420000&message=&resultDetail=
```

### 2.2 回调参数说明

| 字段           | 类型     | 描述                                                         |
|--------------|--------|------------------------------------------------------------|
| code         | int64  | 任务状态码，0为成功，其他：失败；详见 <a href="#">状态码说明</a>                  |
| message      | string | 失败原因文字描述，成功时此值为空                                           |
| requestId    | uint64 | 任务唯一标识，与录音识别请求中返回的 TaskId 一致。数据格式必须设置为 Uint64              |
| appid        | uint64 | 腾讯云应用 ID                                                   |
| projectid    | int64  | 腾讯云项目 ID                                                   |
| audioUrl     | string | 语音 url，如创建任务时为上传数据的方式，则不包含该字段                              |
| text         | string | 识别出的结果文本                                                   |
| resultDetail | string | 包含 <a href="#">详细识别结果</a> ，如创建任务时 ResTextFormat 为0，则不包含该字段 |
| audioTime    | double | 语音总时长                                                      |

## 3. 回调 Response 说明

用户侧收到回调请求后，需要以 JSON 格式返回响应。

### 3.1 回调 Response 示例

```
{ "code" : 0, "message" : "success" }
```

### 3.2 回调 Response 参数说明

| 参数名称    | 类型     | 描述               |
|---------|--------|------------------|
| code    | int64  | 错误码，0为成功，其他值代表失败 |
| message | string | 失败原因说明           |

## 4. 回调状态码



| 数值    | 说明                 |
|-------|--------------------|
| 10000 | 转码失败，请确认音频格式是否符合标准 |
| 10001 | 识别失败               |
| 10002 | 语音时长太短             |
| 10003 | 语音时长太长             |
| 10004 | 无效的语音文件            |
| 10005 | 其他失败               |
| 10006 | 音轨个数不匹配            |
| 10007 | 音频下载失败             |

## 5. 常见问题

### 5.1 回调最多尝试几次？

最多2次。

### 5.2 过了很久都没有收到回调怎么办？

- 请确认回调服务正常，且回调地址是否在公网下可访问。
- 如果回调地址是 https 的，请确认证书经过合法 CA 机构认证。

# 一句话识别相关接口

## 一句话识别

最近更新时间：2025-05-09 01:09:55

### 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

本接口用于对60秒之内的短音频文件进行识别。

- 支持中文普通话、英语、粤语、日语、越南语、马来语、印度尼西亚语、菲律宾语、泰语、葡萄牙语、土耳其语、阿拉伯语、印地语、法语、德语、上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话。
- 支持本地语音文件上传和语音URL上传两种请求方式，音频时长不能超过60s，音频文件大小不能超过3MB。推荐使用 [腾讯云COS](#) 来存储音频、生成URL并提交请求，此种方式会走内网下载音频，极大降低整体请求时延；并且不会产生外网和流量下行费用，可节约成本（可参考COS预签名指南：[使用预签名 URL 访问 COS](#)，获取COS预签名url）
- 音频格式支持wav、pcm、ogg-opus、speex、silk、mp3、m4a、aac、amr。
- 请求方法为 HTTP POST，Content-Type为"application/json; charset=utf-8"
- 签名方法参考 [公共参数](#) 中签名方法v3。
- 默认接口请求频率限制：30次/秒，如您有提高请求频率限制的需求，请[前往购买](#)。

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 | <a href="#">公共参数</a> ，本接口取值：SentenceRecognition。                                                                                                                                                                                                                                                                                                                                                                            |
| Version        | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                                                                                                                                                                                                                                                                                                                     |
| Region         | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。                                                                                                                                                                                                                                                                                                                                                                                          |
| EngSerViceType | 是  | String | 引擎模型类型。<br>电话场景： <ul style="list-style-type: none"><li>• 8k_zh：中文电话通用；</li><li>• 8k_en：英文电话通用；</li></ul><br>非电话场景： <ul style="list-style-type: none"><li>• 16k_zh：中文通用；</li><li>• 16k_zh-PY：中英粤；</li><li>• 16k_zh_medical：中文医疗；</li><li>• 16k_en：英语；</li><li>• 16k_yue：粤语；</li><li>• 16k_ja：日语；</li><li>• 16k_ko：韩语；</li><li>• 16k_vi：越南语；</li><li>• 16k_ms：马来语；</li><li>• 16k_id：印度尼西亚语；</li><li>• 16k_fil：菲律宾语；</li></ul> |

| 参数名称           | 必选 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------|----|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |    |         | <ul style="list-style-type: none"> <li>• 16k_th: 泰语;</li> <li>• 16k_pt: 葡萄牙语;</li> <li>• 16k_tr: 土耳其语;</li> <li>• 16k_ar: 阿拉伯语;</li> <li>• 16k_es: 西班牙语;</li> <li>• 16k_hi: 印地语;</li> <li>• 16k_fr: 法语;</li> <li>• 16k_de: 德语;</li> <li>• 16k_zh_dialect: 多方言, 支持23种方言 (上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话);</li> </ul> 示例值: 16k_en |
| SourceType     | 是  | Integer | 语音数据来源。0: 语音 URL; 1: 语音数据 (post body)。<br>示例值: 1                                                                                                                                                                                                                                                                                                                                               |
| VoiceFormat    | 是  | String  | 识别音频的音频格式, 支持wav、pcm、ogg-opus、speex、silk、mp3、m4a、aac、amr。<br>示例值: wav                                                                                                                                                                                                                                                                                                                          |
| Url            | 否  | String  | 语音的URL地址, 需要公网环境浏览器可下载。当 SourceType 值为 0时须填写该字段, 为 1 时不填。音频时长不能超过60s, 音频文件大小不能超过3MB。<br><br>注意: 推荐使用 <a href="#">腾讯云COS</a> 来存储音频、生成URL并提交请求, 此种方式会走内网下载音频, 极大降低整体请求时延; 并且不会产生外网和流量下行费用, 可节约成本 (COS桶权限需要设置公有读私有写, 或URL设置时效访问签名)<br>示例值: http://test.voice.com/test.wav                                                                                                                         |
| Data           | 否  | String  | 语音数据, 当SourceType 值为1 (本地语音数据上传) 时必须填写, 当SourceType 值为0 (语音 URL上传) 可不写。要使用base64编码(采用python语言时注意读取文件应该为string而不是byte, 以byte格式读取后要decode()。编码后的数据不可带有回车换行符)。音频时长不能超过60s, 音频文件大小不能超过3MB (Base64后)。                                                                                                                                                                                               |
| DataLen        | 否  | Integer | 数据长度, 单位为字节。当 SourceType 值为1 (本地语音数据上传) 时必须填写, 当 SourceType 值为0 (语音 URL上传) 可不写 (此数据长度为数据未进行base64编码时的数据长度)。<br>示例值: 6400                                                                                                                                                                                                                                                                       |
| WordInfo       | 否  | Integer | 是否显示词级别时间戳。0: 不显示; 1: 显示, 不包含标点时间戳; 2: 显示, 包含标点时间戳。默认值为 0。<br>示例值: 0                                                                                                                                                                                                                                                                                                                           |
| FilterDirty    | 否  | Integer | 是否过滤脏词 (目前支持中文普通话引擎)。0: 不过滤脏词; 1: 过滤脏词; 2: 将脏词替换为 *。默认值为 0。<br>示例值: 0                                                                                                                                                                                                                                                                                                                          |
| FilterModal    | 否  | Integer | 是否过滤语气词 (目前支持中文普通话引擎)。0: 不过滤语气词; 1: 部分过滤; 2: 严格过滤。默认值为 0。<br>示例值: 0                                                                                                                                                                                                                                                                                                                            |
| FilterPunc     | 否  | Integer | 是否过滤标点符号 (目前支持中文普通话引擎)。0: 不过滤; 1: 过滤句末标点; 2: 过滤所有标点。默认值为 0。<br>示例值: 0                                                                                                                                                                                                                                                                                                                          |
| ConvertNumMode | 否  | Integer | 是否进行阿拉伯数字智能转换。0: 不转换, 直接输出中文数字; 1: 根据场景智能转换为阿拉伯数字。默认值为1。<br>示例值: 0                                                                                                                                                                                                                                                                                                                             |
| HotwordId      | 否  | String  | 热词id。用于调用对应的热词表, 如果在调用语音识别服务时, 不进行单独的热词id设置, 自动生效默认热词; 如果进行了单独的热词id设置, 那么将生效单独设置的热词id。                                                                                                                                                                                                                                                                                                         |

| 参数名称            | 必选 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|----|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |    |         | 示例值：966ecea2a*537c8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| CustomizationId | 否  | String  | 自学习模型 id。如设置了该参数，将生效对应的自学习模型。<br>示例值：966ecea**d537c8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| HotwordList     | 否  | String  | 临时热词表：该参数用于提升识别准确率。<br>单个热词限制：“热词 权重”，单个热词不超过30个字符（最多10个汉字），权重1~11或者100，如：“腾讯云 5”或“ASR 11”；<br>临时热词表限制：多个热词用英文逗号分割，最多支持128个热词，如：“腾讯云 10,语音识别 5,ASR 11”；<br>参数 hotword_list（临时热词表）与 hotword_id（热词表）区别：<br>hotword_id：热词表。需要先在控制台或接口创建热词表，获得对应hotword_id传入参数来使用热词功能；<br>hotword_list：临时热词表。每次请求时直接传入临时热词表来使用热词功能，云端不保留临时热词表。适用于有极大量热词需求的用户；<br>注意：<br>• 如果同时传入了 hotword_id 和 hotword_list，会优先使用 hotword_list；<br>• 热词权重设置为11时，当前热词将升级为超级热词，建议仅将重要且必须生效的热词设置到11，设置过多权重为11的热词将影响整体字准确率。<br>• 热词权重设置为100时，当前热词开启热词增强同音替换功能（仅支持8k_zh,16k_zh），举例：热词配置“蜜制 100”时，与“蜜制”同拼音（mizhi）的“秘制”的识别结果会被强制替换成“蜜制”。因此建议客户根据自己的实际情况开启该功能。建议仅将重要且必须生效的热词设置到100，设置过多权重为100的热词将影响整体字准确率。 |
| InputSampleRate | 否  | Integer | 支持pcm格式的8k音频在与引擎采样率不匹配的情况下升采样到16k后识别，能有效提升识别准确率。仅支持：8000。如：传入 8000，则pcm音频采样率为8k，当引擎选用16k_zh，那么该8k采样率的pcm音频可以在16k_zh引擎下正常识别。注：此参数仅适用于pcm格式音频，不传入值将维持默认状态，即默认调用的引擎采样率等于pcm音频采样率。<br>示例值：0                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

### 3. 输出参数

| 参数名称          | 类型                    | 描述                                                                                     |
|---------------|-----------------------|----------------------------------------------------------------------------------------|
| Result        | String                | 识别结果。<br>示例值：测试输出结果                                                                    |
| AudioDuration | Integer               | 请求的音频时长，单位为ms<br>示例值：1500                                                              |
| WordSize      | Integer               | 词时间戳列表的长度<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：4                                        |
| WordList      | Array of SentenceWord | 词时间戳列表<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：array                                       |
| RequestId     | String                | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

### 4. 示例

#### 示例1 通过语音Url来调用接口

用户通过语音Url的方式（SourceType为0）请求一句话识别服务，请求模型为电话 8k中文（EngSerViceType = 8k\_zh），音频格式为wav

#### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: SentenceRecognition
X-TC-Timestamp: 1599140162
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=0615f73a69c6b054affd69e4b1cbb68fbe84ae8024a9347df4aa5054933adac8
```

<公共请求参数>

```
{
  "UsrAudioKey": "test",
  "SubServiceType": 2,
  "Url": "http://tes.cos.ap-guangzhou.myqcloud.com/test.wav",
  "ProjectId": 0,
  "EngSerViceType": "8k_zh",
  "VoiceFormat": "wav",
  "SourceType": 0
}
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "41ed9283-0c09-46fb-917b-0b83fa95f0be",
    "Result": "腾讯云语音识别欢迎您。",
    "AudioDuration": 2430,
    "WordSize": 4,
    "WordList": [
      {
        "Word": "腾讯云",
        "StartTime": 120,
        "EndTime": 810
      },
      {
        "Word": "语音识别",
        "StartTime": 810,
        "EndTime": 1530
      },
      {
        "Word": "欢迎",
        "StartTime": 1530,
        "EndTime": 1890
      },
      {
        "Word": "您",
        "StartTime": 1890,
```

```
"EndTime": 2250
}
]
}
}
```

## 示例2 通过语音数据来调用接口

用户通过上传语音数据（Data）的方式（SourceType为1）请求一句话识别服务，请求模型为电话 8k中文（EngSerViceType = 8k\_zh），音频格式为wav

### 输入示例

```
POST / HTTP/1.1
Host: asr.tencentcloudapi.com
Content-Type: application/json; charset=utf-8
X-TC-Version: 2019-06-14
X-TC-Region: ap-shanghai
X-TC-Action: SentenceRecognition
X-TC-Timestamp: 1599140479
Authorization: TC3-HMAC-SHA256 Credential=*****/2020-09-03/asr/tc3_request, SignedHeaders=content-type;host, Signature=aa5e2b8b16ced1ac9f877c9a92dab641dcd940837f5869a9e933688c181094de
<公共请求参数>
```

```
{
  "UsrAudioKey": "test",
  "SubServiceType": 2,
  "ProjectId": 0,
  "EngSerViceType": "8k_zh",
  "VoiceFormat": "wav",
  "Data": "eGNmYXNkZmFzZmFzZGZhc2RmCg==",
  "SourceType": 1
}
```

### 输出示例

```
{
  "Response": {
    "RequestId": "41ed9283-0c09-46fb-917b-0b83fa95f0be",
    "Result": "腾讯云语音识别欢迎您。",
    "AudioDuration": 2430,
    "WordSize": 4,
    "WordList": [
      {
        "Word": "腾讯云",
        "StartTime": 120,
        "EndTime": 810
      },
      {
```

```
"Word": "语音识别",
"StartTime": 810,
"EndTime": 1530
},
{
"Word": "欢迎",
"StartTime": 1530,
"EndTime": 1890
},
{
"Word": "您",
"StartTime": 1890,
"EndTime": 2250
}
]
}
}
```

## 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.ErrorRecognize | 识别失败。 |

| 错误码                                              | 描述                        |
|--------------------------------------------------|---------------------------|
| FailedOperation.ServiceIsolate                   | 账号因为欠费停止服务，请在腾讯云账户充值。     |
| FailedOperation.UserHasNoAmount                  | 资源包耗尽，请购买资源包或开通后付费        |
| FailedOperation.UserHasNoFreeAmount              | 资源包耗尽，请开通后付费或者购买资源包       |
| FailedOperation.UserNotRegistered                | 服务未开通，请在腾讯云官网语音识别控制台开通服务。 |
| InternalServerError                              | 内部错误。                     |
| InternalServerError.ErrorConfigure               | 初始化配置失败。                  |
| InternalServerError.ErrorCreateLog               | 创建日志失败。                   |
| InternalServerError.ErrorDownFile                | 下载音频文件失败。                 |
| InternalServerError.ErrorFailNewprequest         | 新建数组失败。                   |
| InternalServerError.ErrorFailWritetodb           | 写入数据库失败。                  |
| InternalServerError.ErrorFileCannotopen          | 文件无法打开。                   |
| InternalServerError.ErrorGetRoute                | 获取路由失败。                   |
| InternalServerError.ErrorMakeLogpath             | 创建日志路径失败。                 |
| InternalServerError.ErrorRecognize               | 识别失败。                     |
| InvalidParameter.ErrorContentlength              | 请求数据长度无效。                 |
| InvalidParameter.ErrorParamsMissing              | 参数不全。                     |
| InvalidParameter.ErrorParsequest                 | 解析请求数据失败。                 |
| InvalidParameterValue                            | 参数取值错误。                   |
| InvalidParameterValue.ErrorInvalidAppid          | AppId无效。                  |
| InvalidParameterValue.ErrorInvalidClientip       | ClientIp无效。               |
| InvalidParameterValue.ErrorInvalidEngservice     | EngServiceType无效。         |
| InvalidParameterValue.ErrorInvalidProjectid      | ProjectId无效。              |
| InvalidParameterValue.ErrorInvalidRequestid      | RequestId无效。              |
| InvalidParameterValue.ErrorInvalidSourcetype     | SourceType无效。             |
| InvalidParameterValue.ErrorInvalidSubservicetype | SubserviceType无效。         |
| InvalidParameterValue.ErrorInvalidUrl            | Url无效。                    |
| InvalidParameterValue.ErrorInvalidUseraudiokey   | UsrAudioKey无效。            |
| InvalidParameterValue.ErrorInvalidVoiceFormat    | 音频编码格式不支持。                |
| InvalidParameterValue.ErrorInvalidVoicedata      | 音频数据无效。                   |
| InvalidParameterValue.ErrorVoicedataTooLong      | 音频时长超过限制。                 |



# 自学习相关接口

## 创建自学习模型

最近更新时间：2025-04-25 01:09:10

### 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户使用该接口可以创建自学习模型，以供识别调用。

注意：调用该接口后，模型会自动训练。新建模型成功后，调用ModifyCustomizationState接口修改为上线状态，即可在识别请求中使用对应模型ID。

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 | <a href="#">公共参数</a> ，本接口取值：CreateCustomization。                               |
| Version   | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                        |
| Region    | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。                                             |
| ModelName | 是  | String | 自学习模型名称，需在1-20字符之间<br>示例值：出行模型                                                 |
| TextUrl   | 是  | String | 文本文件的下载地址，服务会从该地址下载文件，目前仅支持腾讯云cos<br>示例值： <code>https://xxx.xx.com/text</code> |
| ModelType | 是  | String | 自学习模型类型，填写8k或者16k<br>示例值：8k                                                    |

### 3. 输出参数

| 参数名称      | 类型     | 描述                                                                                    |
|-----------|--------|---------------------------------------------------------------------------------------|
| ModelId   | String | 模型ID<br>示例值：769762ae-xxxxxxxxxx-ec3cda3142e                                           |
| RequestId | String | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。 |

### 4. 示例

#### 示例1 创建自学习模型

### 输入示例

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

{
  "ModelName": "自学习模型",
  "ModelType": "16k",
  "TextUrl": "http://www.download.com/test.txt"
}
```

### 输出示例

```
{
  "Response": {
    "RequestId": "a23740a1-4bb5-4154-9f1f-ea36b1bfe2ea",
    "ModelId": "413c479df72d11eaacd3c81fbecfd0bd"
  }
}
```

## 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。    |
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| AuthFailure.UnauthorizedOperation          | 未授权操作。         |
| FailedOperation.ErrorDownFile              | 下载音频文件失败。      |
| InternalError                              | 内部错误。          |
| InvalidParameter                           | 参数错误。          |
| InvalidParameter.FileEncode                | 文件编码错误。        |
| InvalidParameterValue                      | 参数取值错误。        |
| LimitExceeded.CustomizationFull            | 自学习模型创建个数已到限制。 |

# 查询自学习模型列表

最近更新时间：2025-04-25 01:09:09

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

查询自学习模型列表

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  | <a href="#">公共参数</a> ，本接口取值：GetCustomizationList。 |
| Version | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。           |
| Region  | 否  | String  | <a href="#">公共参数</a> ，本接口不需要传递此参数。                |
| Limit   | 否  | Integer | 分页大小，默认1000<br>示例值：10                             |
| Offset  | 否  | Integer | 分页offset，默认0<br>示例值：0                             |

## 3. 输出参数

| 参数名称       | 类型                             | 描述                                                                                     |
|------------|--------------------------------|----------------------------------------------------------------------------------------|
| Data       | Array of <a href="#">Model</a> | 自学习模型数组<br>注意：此字段可能返回 null，表示取不到有效值。                                                   |
| TotalCount | Integer                        | 自学习模型总量<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：10                                         |
| RequestId  | String                         | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 查询自学习模型列表

查询自学习模型列表

输入示例

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

#### 输出示例

```
{
  "Response": {
    "RequestId": "5f44d72be3a308c9b62158b0",
    "TotalCount": 2,
    "Data": [
      {
        "ModelName": "dd",
        "DictName": "example.txt",
        "ModelId": "a07acb16e60311ea9a76b49691037310",
        "ModelType": "8k",
        "ServiceType": "realtime",
        "ModelState": -1,
        "AtUpdated": "2020-08-24 20:16:25",
        "TagInfos": null
      },
      {
        "ModelName": "ddd",
        "DictName": "example.txt",
        "ModelId": "98bda0fee60311ea9a76b49691037310",
        "ModelType": "16k",
        "ServiceType": "realtime",
        "ModelState": -1,
        "AtUpdated": "2020-08-24 20:16:15",
        "TagInfos": null
      }
    ]
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

- Tencent Cloud SDK 3.0 for Python: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)

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

## 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                               | 描述          |
|-----------------------------------|-------------|
| AuthFailure                       | CAM签名/鉴权错误。 |
| AuthFailure.UnauthorizedOperation | 未授权操作。      |
| FailedOperation                   | 操作失败。       |
| InternalError                     | 内部错误。       |

# 更新自学习模型

最近更新时间：2025-05-09 01:09:53

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过该接口可以更新自学习模型，如模型名称、模型类型、模型语料。

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 | <a href="#">公共参数</a> ，本接口取值：ModifyCustomization。                 |
| Version   | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                          |
| Region    | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。                               |
| ModelId   | 是  | String | 要修改的模型ID<br>示例值：fsamdfiuh****xfdsgdsh                            |
| ModelName | 否  | String | 要修改的模型名称，长度需在1-20个字符之间<br>示例值：通用模型                               |
| ModelType | 否  | String | 要修改的模型类型，为8k或者16k<br>示例值：8k                                      |
| TextUrl   | 否  | String | 要修改的模型语料的下载地址，目前仅支持腾讯云cos<br>示例值：https://www.modle.com/model.txt |

## 3. 输出参数

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

## 4. 示例

### 示例1 修改自学习模型

修改

输入示例

```
https://asr.tencentcloudapi.com/?Action=ModifyCustomization
&ModelId=413c479df72d11eaacd3c81fbecfd0bd
&ModelName=修改名称
&ModelType=16k
&TextUrl=http://www.download.com/test.txt
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "b3808ad3-d8dd-4b65-96c9-7d6f1f81b6b6"
  }
}
```

## 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             |
|--------------------------------------------|----------------|
| AuthFailure                                | CAM签名/鉴权错误。    |
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |



| 错误码                               | 描述          |
|-----------------------------------|-------------|
| AuthFailure.UnauthorizedOperation | 未授权操作。      |
| FailedOperation.ErrorDownFile     | 下载音频文件失败。   |
| InternalError                     | 内部错误。       |
| InvalidParameter                  | 参数错误。       |
| InvalidParameter.FileEncode       | 文件编码错误。     |
| InvalidParameter.ModelState       | 该模型状态不允许删除。 |
| InvalidParameterValue             | 参数取值错误。     |
| InvalidParameterValue.ModelId     | 模型不存在。      |

# 删除自学习模型

最近更新时间：2025-05-09 01:09:53

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

用户通过该接口可以删除自学习模型

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 | <a href="#">公共参数</a> ，本接口取值：DeleteCustomization。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。          |
| Region  | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。               |
| ModelId | 是  | String | 要删除的模型ID<br>示例值：sdsafjasj*sdfasfjj               |

## 3. 输出参数

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

## 4. 示例

### 示例1 删除自学习模型

删除

输入示例

```
https://asr.tencentcloudapi.com/?Action=DeleteCustomization
&ModelId=f3520776f31211ea9d26c81fbecfd0bd
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "7cc30534-edde-4969-b281-7a30c5cbf806"
  }
}
```

## 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。    |
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| AuthFailure.UnauthorizedOperation          | 未授权操作。         |
| FailedOperation                            | 操作失败。          |
| InternalError                              | 内部错误。          |
| InternalError.TagRequestError              | 请求标签接口出错       |
| InvalidParameter                           | 参数错误。          |
| InvalidParameter.ModelState                | 该模型状态不允许删除。    |
| InvalidParameterValue                      | 参数取值错误。        |

| 错误码                           | 描述      |
|-------------------------------|---------|
| InvalidParameterValue.ModelId | 模型不存在。  |
| LimitExceeded                 | 超过配额限制。 |

# 修改自学习模型状态

最近更新时间：2025-04-25 01:09:09

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

通过该接口，用户可以修改自学习模型状态，上下线自学习模型

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  | <a href="#">公共参数</a> ，本接口取值：ModifyCustomizationState。 |
| Version | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。               |
| Region  | 否  | String  | <a href="#">公共参数</a> ，本接口不需要传递此参数。                    |
| ModelId | 是  | String  | 自学习模型ID                                               |
| ToState | 是  | Integer | 想要变换的模型状态，-1代表下线，1代表上线                                |

## 3. 输出参数

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

## 4. 示例

### 示例1 修改自学习模型状态

修改状态

输入示例

```
https://asr.tencentcloudapi.com/?Action=ModifyCustomizationState
&ModelId=aa6f402f263f12ea856fc81fbecfd0sd
&ToState=1
&<公共请求参数>
```

## 输出示例

```
{
  "Response": {
    "ModelId": "aa6f402f263f12ea856fc81fbecfd0sd",
    "RequestId": "3290cfb2-91a7-4989-91de-d1d624a77c12"
  }
}
```

## 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。    |
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| AuthFailure.UnauthorizedOperation          | 未授权操作。         |
| InternalError                              | 内部错误。          |
| InvalidParameter                           | 参数错误。          |
| InvalidParameterValue                      | 参数取值错误。        |
| InvalidParameterValue.ModelId              | 模型不存在。         |

| 错误码                           | 描述          |
|-------------------------------|-------------|
| InvalidParameterValue.ToState | 非法的模型状态。    |
| LimitExceeded.OnlineFull      | 上线模型个数已到限制。 |

# 下载自学习模型语料

最近更新时间：2025-05-09 01:09:53

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过该接口可以下载自学习模型的语料

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 | <a href="#">公共参数</a> ，本接口取值：DownloadCustomization。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。            |
| Region  | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。                 |
| ModelId | 是  | String | 自学习模型ID<br>示例值：sdsafjasj*sdfasfjj                  |

## 3. 输出参数

| 参数名称        | 类型     | 描述                                                                                     |
|-------------|--------|----------------------------------------------------------------------------------------|
| DownloadUrl | String | 下载地址<br>示例值：https://www.model.com/model.txt                                            |
| RequestId   | String | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 下载自学习模型

下载

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=DownloadCustomization
&ModelId=bc4a37d9f18511ea9a1ec81fbecfd0bd
&<公共请求参数>
```

#### 输出示例



```
{
  "Response": {
    "RequestId": "864d6a32-3b1c-462d-b019-ebdd65b3a644",
    "DownloadUrl": "https://customization-download-temp-1256237915.cos.ap-guangzhou.myqcloud.com/bc4a37d9f18511ea9a1ec81fbecfd0bd_0.dict"
  }
}
```

## 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。    |
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| AuthFailure.UnauthorizedOperation          | 未授权操作。         |
| FailedOperation                            | 操作失败。          |
| FailedOperation.ErrorDownFile              | 下载音频文件失败。      |
| InternalServerError                        | 内部错误。          |
| InvalidParameter                           | 参数错误。          |

| 错误码                           | 描述      |
|-------------------------------|---------|
| InvalidParameterValue.ModelId | 模型不存在。  |
| LimitExceeded                 | 超过配额限制。 |

# 获取自学习模型信息

最近更新时间：2025-05-09 01:09:53

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

通过自学习模型id获取自学习模型详细信息

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 | <a href="#">公共参数</a> ，本接口取值：GetModelInfo。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。   |
| Region  | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。        |
| ModelId | 是  | String | 模型id<br>示例值：sdsafjasj*sdfasfjj            |

## 3. 输出参数

| 参数名称      | 类型                    | 描述                                                                                     |
|-----------|-----------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">Model</a> | 模型信息                                                                                   |
| RequestId | String                | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 通过模型id获取模型信息

通过模型id获取模型信息

输入示例

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

```
{
  "ModelId": "5a15aa3ae*****da0d56c92bf48cdf2"
}
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "ModelName": "通用模型",
      "DictName": "c6824a6d-05ec-431c-95c9",
      "ModelId": "5a15aa3ae*****da0d56c92bf48cdf2",
      "ModelType": "8k",
      "ServiceType": "",
      "ModelState": 1,
      "AtUpdated": "2023-04-27 14:40:09",
      "TagInfos": null
    },
    "RequestId": "dsafsswdadas*****dssadas"
  }
}
```

## 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。    |
| InternalServerError.FailAccessDatabase | 访问数据库失败。 |
| InvalidParameter                       | 参数错误。    |
| InvalidParameterValue.ModelId          | 模型不存在。   |

# 热词相关接口

## 创建热词表

最近更新时间：2025-04-25 01:09:11

### 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过本接口进行热词表的创建。

- 默认最多可创建30个热词表。
- 每个热词表最多可添加1000个词，每个词最长10个汉字或30个英文字符，不能超出限制。
- 热词表可以通过数组或者本地文件形式上传。
- 本地文件必须为UTF-8编码格式，每行仅添加一个热词且不能包含标点和特殊字符。
- 热词权重取值范围为[1,11]之间的整数或者100，权重越大代表该词被识别出来的概率越大。
- 注意: 热词权重设置为11时，当前热词将升级为超级热词，建议仅将重要且必须生效的热词设置到11，设置过多权重为11的热词将影响整体字准率。

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                           | <a href="#">公共参数</a> ，本接口取值：CreateAsrVocab。                                                                                                                                                                                                                  |
| Version       | 是  | String                           | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                                                                                                                                                      |
| Region        | 否  | String                           | <a href="#">公共参数</a> ，本接口不需要传递此参数。                                                                                                                                                                                                                           |
| Name          | 是  | String                           | 热词表名称，长度在1-255之间<br>示例值：避风港                                                                                                                                                                                                                                  |
| Description   | 否  | String                           | 热词表描述，长度在0-1000之间<br>示例值：热词                                                                                                                                                                                                                                  |
| WordWeights.N | 否  | Array of <a href="#">HotWord</a> | 词权重数组，包含全部的热词和对应的权重。每个热词的长度不大于10个汉字或30个英文字符，权重为[1,11]之间整数或者100，数组长度不大于1000<br>注意：<br>– 热词权重设置为11时，当前热词将升级为超级热词，建议仅将重要且必须生效的热词设置到11，设置过多权重为11的热词将影响整体字准率。<br>– 热词权重设置为100时，当前热词开启热词增强同音替换功能（仅支持8k_zh,16k_zh），举例：热词配置“蜜制 100”时，与“蜜制”同拼音（mizhi）的“秘制”的识别结果会被强制替换 |

| 参数名称          | 必选 | 类型     | 描述                                                                                                                                                     |
|---------------|----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |    |        | 成“蜜制”。因此建议客户根据自己的实际情况开启该功能。建议仅将重要且必须生效的热词设置到100，设置过多权重为100的热词将影响整体字准确率。<br>示例值：1                                                                       |
| WordWeightStr | 否  | String | 词权重文件（纯文本文件）的二进制base64编码，以行分隔，每行的格式为word weight，即以英文符号 为分割，左边为词，右边为权重，如：你好 5。<br>当用户传此参数（参数长度大于0），即以此参数解析词权重，WordWeights会被忽略<br>示例值：避风港 10 (进行bas64编码) |

### 3. 输出参数

| 参数名称      | 类型     | 描述                                                                                    |
|-----------|--------|---------------------------------------------------------------------------------------|
| VocabId   | String | 词表ID，可用于获取词表信息<br>示例值：adstdadsdsad                                                    |
| RequestId | String | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。 |

### 4. 示例

#### 示例1 创建热词表

用户通过上传词权重数组方式创建热词表

##### 输入示例

```
https://asr.tencentcloudapi.com/?Action=CreateAsrVocab
&Name=词表名称
&Description=词表描述
&WordWeights.0.Word=智聆
&WordWeights.0.Weight=1
&WordWeights.1.Word=滨海大厦
&WordWeights.1.Weight=6
&WordWeights.2.Word=存储桶
&WordWeights.2.Weight=8
&WordWeights.3.Word=核保
&WordWeights.3.Weight=10
&<公共请求参数>
```

##### 输出示例

```
{
  "Response": {
    "VocabId": "aa6f402f263f12ea856fc81fbecfd0sd",
    "RequestId": "b3808ad3-d8dd-4b65-96c9-7d6f1f81b6b6"
  }
}
```

### 5. 开发者资源

## 腾讯云 API 平台

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

## API Inspector

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

## SDK

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

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

## 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                                          | 描述                    |
|----------------------------------------------|-----------------------|
| FailedOperation                              | 操作失败。                 |
| FailedOperation.ServiceIsolate               | 账号因为欠费停止服务，请在腾讯云账户充值。 |
| InternalServerError                          | 内部错误。                 |
| InternalServerError.FailAccessDatabase       | 访问数据库失败。              |
| InvalidParameter                             | 参数错误。                 |
| InvalidParameter.ErrorContentlength          | 请求数据长度无效。             |
| InvalidParameter.ErrorParamsMissing          | 参数不全。                 |
| InvalidParameter.ErrorParsequest             | 解析请求数据失败。             |
| InvalidParameter.InvalidVocabState           | 非法的词表状态。              |
| InvalidParameterValue                        | 参数取值错误。               |
| InvalidParameterValue.ErrorInvalidAppid      | AppId无效。              |
| InvalidParameterValue.ErrorInvalidClientip   | ClientIp无效。           |
| InvalidParameterValue.InvalidParameterLength | 非法的参数长度。              |
| InvalidParameterValue.InvalidVocabId         | 非法的VocabId。           |
| InvalidParameterValue.InvalidWordWeight      | 词权重不合法。               |



| 错误码                                        | 描述                |
|--------------------------------------------|-------------------|
| InvalidParameterValue.InvalidWordWeightStr | 非法的WordWeightStr。 |
| LimitExceeded                              | 超过配额限制。           |
| LimitExceeded.VocabFull                    | 热词表数量已到账号限制。      |
| MissingParameter                           | 缺少参数错误。           |
| UnknownParameter                           | 未知参数错误。           |

# 删除热词表

最近更新时间：2025-04-25 01:09:11

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

用户通过本接口进行热词表的删除。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

## 2. 输入参数

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

| 参数名称    | 必选 | 类型     | 描述                                          |
|---------|----|--------|---------------------------------------------|
| Action  | 是  | String | <a href="#">公共参数</a> ，本接口取值：DeleteAsrVocab。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。     |
| Region  | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。          |
| VocabId | 是  | String | 热词表Id<br>示例值：xcxdsada                       |

## 3. 输出参数

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

## 4. 示例

### 示例1 删除热词表

用户通过该API，传入热词表ID，可以删除对应的热词表

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=DeleteAsrVocab
&VocabId=aa6f402f263f12ea856fc81fbecfd0sd
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "7eccebc3-2d60-4cee-8fa8-7f8001a71ab2"
  }
}
```

## 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.FailAccessDatabase       | 访问数据库失败。  |
| InvalidParameter                             | 参数错误。     |
| InvalidParameter.ErrorContentlength          | 请求数据长度无效。 |
| InvalidParameter.ErrorParamsMissing          | 参数不全。     |
| InvalidParameter.ErrorParsequest             | 解析请求数据失败。 |
| InvalidParameterValue                        | 参数取值错误。   |
| InvalidParameterValue.ErrorInvalidAppid      | AppId无效。  |
| InvalidParameterValue.InvalidParameterLength | 非法的参数长度。  |

| 错误码                                        | 描述                |
|--------------------------------------------|-------------------|
| InvalidParameterValue.InvalidVocabId       | 非法的VocabId。       |
| InvalidParameterValue.InvalidWordWeight    | 词权重不合法。           |
| InvalidParameterValue.InvalidWordWeightStr | 非法的WordWeightStr。 |

# 下载热词表

最近更新时间：2025-04-25 01:09:10

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过本接口进行热词表的下载，获得词表权重文件形式的 base64 值，文件形式为通过 “|” 分割的词和权重，即 `word|weight` 的形式。

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 | <a href="#">公共参数</a> ，本接口取值：DownloadAsrVocab。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。       |
| Region  | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。            |
| VocabId | 是  | String | 词表ID。<br>示例值：ssdasdasd                        |

## 3. 输出参数

| 参数名称          | 类型     | 描述                                                                                     |
|---------------|--------|----------------------------------------------------------------------------------------|
| VocabId       | String | 词表ID。<br>示例值：dasdsadasd                                                                |
| WordWeightStr | String | 词表权重文件形式的base64值。<br>示例值：1                                                             |
| RequestId     | String | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 下载热词表

用户通过该接口可以获得相应热词表里的内容，即 `word|weight` 形式的 base64 值。

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=DownloadAsrVocab
&VocabId=aa6f402f263f12ea856fc81fbecfd0sd
```

&amp;&lt;公共请求参数&gt;

### 输出示例

```
{
  "Response": {
    "VocabId": "aa6f402f263f12ea856fc81fbecfd0sd",
    "WordWeightStr": "5a2X56ym5LiyfDQNCg==",
    "RequestId": "1085dec9-df56-4124-8a6d-acb198177923"
  }
}
```

## 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。      |
| InternalServerError.FailAccessDatabase | 访问数据库失败。   |
| InternalServerError.FailAccessRedis    | 访问Redis失败。 |
| InvalidParameter                       | 参数错误。      |

| 错误码                                  | 描述          |
|--------------------------------------|-------------|
| InvalidParameterValue                | 参数取值错误。     |
| InvalidParameterValue.InvalidVocabId | 非法的VocabId。 |

# 获取热词表

最近更新时间：2025-04-25 01:09:10

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

用户根据词表的ID可以获取对应的热词表信息

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 | <a href="#">公共参数</a> ，本接口取值：GetAsrVocab。 |
| Version | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。  |
| Region  | 否  | String | <a href="#">公共参数</a> ，本接口不需要传递此参数。       |
| VocabId | 是  | String | 热词表ID<br>示例值：ashsdhsdshsjh               |

## 3. 输出参数

| 参数名称        | 类型                                | 描述                                            |
|-------------|-----------------------------------|-----------------------------------------------|
| Name        | String                            | 热词表名称<br>示例值：热词                               |
| Description | String                            | 热词表描述<br>示例值：我是热词                             |
| VocabId     | String                            | 热词表ID<br>示例值：xasddasds                        |
| WordWeights | Array of <a href="#">HotWord</a>  | 词权重列表<br>示例值：1                                |
| CreateTime  | <a href="#">Timestamp ISO8601</a> | 词表创建时间<br>示例值：1                               |
| UpdateTime  | <a href="#">Timestamp ISO8601</a> | 词表更新时间<br>示例值：1                               |
| State       | Integer                           | 热词表状态，1为默认状态即在识别时默认加载该热词表进行识别，0为初始状态<br>示例值：0 |



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

## 4. 示例

### 示例1 获取热词表

用户通过热词表ID获取热词表内容

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=GetAsrVocab
&VocabId=aa6f402f263f12ea856fc81fbecfd0sd
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "a31edb26-3b6a-481e-91f8-d0009065eae",
    "Name": "词表名称",
    "Description": "词表描述",
    "VocabId": "aa6f402f263f12ea856fc81fbecfd0sd",
    "WordWeights": [
      {
        "Word": "智聆",
        "Weight": 1
      },
      {
        "Word": "滨海大厦",
        "Weight": 6
      },
      {
        "Word": "存储桶",
        "Weight": 8
      },
      {
        "Word": "核保",
        "Weight": 10
      }
    ],
    "CreateTime": "2020-01-14T10:24:07+08:00",
    "UpdateTime": "2020-01-14T10:24:07+08:00",
    "State": 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。       |
| InternalServerError.FailAccessDatabase | 访问数据库失败。    |
| InvalidParameter                       | 参数错误。       |
| InvalidParameterValue.InvalidVocabId   | 非法的VocabId。 |

# 列举热词表

最近更新时间：2025-05-09 01:09:54

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过该接口，可获得所有的热词表及其信息。

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  | <a href="#">公共参数</a> ，本接口取值：GetAsrVocabList。 |
| Version | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。      |
| Region  | 否  | String  | <a href="#">公共参数</a> ，本接口不需要传递此参数。           |
| Offset  | 否  | Integer | 分页Offset<br>示例值：0                            |
| Limit   | 否  | Integer | 分页Limit<br>示例值：10                            |

## 3. 输出参数

| 参数名称       | 类型                             | 描述                                                                                     |
|------------|--------------------------------|----------------------------------------------------------------------------------------|
| VocabList  | Array of <a href="#">Vocab</a> | 热词表列表                                                                                  |
| TotalCount | Integer                        | 热词列表总数<br>示例值：100                                                                      |
| RequestId  | String                         | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 列举热词表

用户通过该接口，可获得所有的热词表及其信息。

输入示例

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

#### 输出示例

```
{
  "Response": {
    "RequestId": "e4c745a1-27a0-4143-972e-2ced14645ab4",
    "TotalCount": 1,
    "VocabList": [
      {
        "Name": "词表1",
        "Description": "业务1",
        "VocabId": "e92bd0d632d211eab508446a2eb5fd98",
        "WordWeights": [
          {
            "Word": "智聆",
            "Weight": 1
          }
        ],
        "TagInfos": [
          "tag"
        ],
        "CreateTime": "2020-01-09T19:26:42+08:00",
        "UpdateTime": "2020-01-14T10:43:07+08:00",
        "State": 0
      },
      {
        "Name": "词表2",
        "Description": "业务2",
        "VocabId": "f10b9bf4367411eab508446a2eb5fd98",
        "WordWeights": [
          {
            "Word": "滨海大厦",
            "Weight": 6
          }
        ],
        "TagInfos": [
          "tag"
        ],
        "CreateTime": "2020-01-14T10:24:07+08:00",
        "UpdateTime": "2020-01-14T10:24:07+08:00",
        "State": 0
      },
      {
        "Name": "词表3",
        "Description": "业务3",
        "VocabId": "540acf7332d311eab508446a2eb5fd98",
        "WordWeights": [
```

```
{
  "Word": "存储桶",
  "Weight": 8
},
{
  "TagInfos": [
    {
      "tag": "tag"
    }
  ],
  "CreateTime": "2020-01-09T19:29:41+08:00",
  "UpdateTime": "2020-01-09T19:29:42+08:00",
  "State": 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             |
|--------------------------------------------|----------------|
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| AuthFailure.UnauthorizedOperation          | 未授权操作。         |

| 错误码                                    | 描述         |
|----------------------------------------|------------|
| FailedOperation                        | 操作失败。      |
| InternalServerError                    | 内部错误。      |
| InternalServerError.FailAccessDatabase | 访问数据库失败。   |
| InternalServerError.FailAccessRedis    | 访问Redis失败。 |
| InternalServerError.TagRequestError    | 请求标签接口出错   |
| InvalidParameter                       | 参数错误。      |
| InvalidParameterValue                  | 参数取值错误。    |
| MissingParameter                       | 缺少参数错误。    |

# 设置热词表状态

最近更新时间：2025-04-25 01:09:10

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

用户通过该接口可以设置热词表的默认状态。初始状态为0，用户可设置状态为1，即为默认状态。默认状态表示用户在请求识别时，如不设置热词表ID，则默认使用状态为1的热词表。

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  | <a href="#">公共参数</a> ，本接口取值：SetVocabState。 |
| Version | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。    |
| Region  | 否  | String  | <a href="#">公共参数</a> ，本接口不需要传递此参数。         |
| VocabId | 是  | String  | 热词表ID。<br>示例值：dsadsdas                     |
| State   | 是  | Integer | 热词表状态，1：设为默认状态；0：设为非默认状态。<br>示例值：1         |

## 3. 输出参数

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

## 4. 示例

### 示例1 设置热词表状态

用户通过该接口可以将热词表设置为默认状态（State=1），或者取消默认状态（State=0）。

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=SetVocabState
&VocabId=aa6f402f263f12ea856fc81fbecfd0sd
```

```
&State=1
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "VocabId": "aa6f402f263f12ea856fc81fbecfd0sd",
    "RequestId": "9910af1f-d479-4cb5-a303-b0a4c1c03137"
  }
}
```

## 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。      |
| InternalServerError.FailAccessDatabase | 访问数据库失败。   |
| InternalServerError.FailAccessRedis    | 访问Redis失败。 |
| InvalidParameter                       | 参数错误。      |



| 错误码                                     | 描述          |
|-----------------------------------------|-------------|
| InvalidParameterValue                   | 参数取值错误。     |
| InvalidParameterValue.InvalidVocabId    | 非法的VocabId。 |
| InvalidParameterValue.InvalidVocabState | 非法的词表状态。    |

# 更新热词表

最近更新时间：2025-05-09 01:09:54

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过本接口进行对应的词表信息更新。

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                           | <a href="#">公共参数</a> ，本接口取值：UpdateAsrVocab。                                                                                                                             |
| Version       | 是  | String                           | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                                                                 |
| Region        | 否  | String                           | <a href="#">公共参数</a> ，本接口不需要传递此参数。                                                                                                                                      |
| VocabId       | 是  | String                           | 热词表ID<br>示例值：aa6f402f263f12ea856fc81fbecfd0sd                                                                                                                           |
| Name          | 否  | String                           | 热词表名称，长度在1-255之间<br>示例值：新的词表名称                                                                                                                                          |
| WordWeights.N | 否  | Array of <a href="#">HotWord</a> | 词权重数组，包含全部的热词和对应的权重。每个热词的长度不大于10个汉字或30个英文字符，权重为[1,11]之间整数或100，数组长度不大于1000（注意：如果仅更新热词表名称或者描述字段，请求不用带本参数）                                                                 |
| WordWeightStr | 否  | String                           | 词权重文件（纯文本文件）的二进制base64编码，以行分隔，每行的格式为word weight，即以英文符号 为分割，左边为词，右边为权重，如：你好 5。<br>当用户传此参数（参数长度大于0），即以此参数解析词权重，WordWeights会被忽略（注意：如果仅更新热词表名称或者描述字段，请求不用带本参数）<br>示例值：蜜汁 10 |
| Description   | 否  | String                           | 热词表描述，长度在0-1000之间<br>示例值：新的词表描述                                                                                                                                         |

## 3. 输出参数

| 参数名称      | 类型     | 描述                                                                                     |
|-----------|--------|----------------------------------------------------------------------------------------|
| VocabId   | String | 热词表ID<br>示例值：aa6f402f263f*a856fc81fbecfd0sd                                            |
| RequestId | String | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 更新热词表

用户通过该API传入已经创建的热词表ID，可以更新其相应的内容

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=UpdateAsrVocab
&VocabId=aa6f402f263f12ea856fc81fbecfd0sd
&Name=新的词表名称
&Description=新的词表描述
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "VocabId": "aa6f402f263f12ea856fc81fbecfd0sd",
    "RequestId": "3290cfb2-91a7-4989-91de-d1d624a77c12"
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

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

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                                          | 描述                    |
|----------------------------------------------|-----------------------|
| FailedOperation                              | 操作失败。                 |
| FailedOperation.ServiceIsolate               | 账号因为欠费停止服务，请在腾讯云账户充值。 |
| InternalError                                | 内部错误。                 |
| InternalError.FailAccessDatabase             | 访问数据库失败。              |
| InvalidParameter                             | 参数错误。                 |
| InvalidParameterValue                        | 参数取值错误。               |
| InvalidParameterValue.InvalidParameterLength | 非法的参数长度。              |
| InvalidParameterValue.InvalidVocabId         | 非法的VocabId。           |
| InvalidParameterValue.InvalidWordWeight      | 词权重不合法。               |
| InvalidParameterValue.InvalidWordWeightStr   | 非法的WordWeightStr。     |
| LimitExceeded.VocabFull                      | 热词表数量已到账号限制。          |
| MissingParameter                             | 缺少参数错误。               |
| UnknownParameter                             | 未知参数错误。               |

# 说话人验证相关接口

## 说话人注册

最近更新时间：2025-04-25 01:09:08

### 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

说话人注册接口用于注册一个指定音频，生成一个唯一的说话人id，后续可通过说话人验证接口验证其它音频和已有的说话人ID匹配度，注册时可指定说话人昵称，方便标识说话人ID，说话人昵称可重复配置。

（注：一个appid最多可以注册1000个说话人ID，一个说话人ID仅支持一条音频注册，后续可通过更新接口进行更新）

使用须知

支持的输入格式：编码文件(PCM, WAV)、16 bit采样位数、单声道（mono）。

支持的音频采样率：16000 Hz。

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  | <a href="#">公共参数</a> ，本接口取值：VoicePrintEnroll。                                      |
| Version     | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                            |
| Region      | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                                                    |
| VoiceFormat | 是  | Integer | 音频格式 0: pcm, 1: wav<br>示例值：0                                                       |
| SampleRate  | 是  | Integer | 音频采样率，目前支持16000，单位：Hz，必填<br>示例值：16000                                              |
| Data        | 是  | String  | 音频数据, base64 编码, 音频时长不能超过30s，数据大小不超过2M<br>示例值：UklGRka1CQBXQVZFZm10IBAAAAABAAEAgD4E |
| SpeakerNick | 否  | String  | 说话人昵称 不超过32字节<br>示例值：小明                                                            |
| GroupId     | 否  | String  | 分组id, 仅支持大小写字母和下划线的组合，不超过128个字符<br>示例值：group_A                                     |

### 3. 输出参数

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

## 4. 示例

### 示例1 注册说话人

通过一段音频数据进行说话人注册

#### 输入示例

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

{
  "VoiceFormat": 0,
  "SampleRate": 16000,
  "SpeakerNick": "ross",
  "Data": "UklGRka1CQBXQVZFZm10IBAAAAABAAEAgD4E"
}
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "VoicePrintId": "db24a3e3-7926-5xxxxxxf01",
      "SpeakerNick": "ross"
    },
    "RequestId": "dbxxxxxxe3-7926-5xxxxxxf01"
  }
}
```

## 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.NotExistentVoicePrintId     | 不存在的话人id          |
| InternalServerError                         | 内部错误。             |
| InternalServerError.ErrorRecognize          | 识别失败。             |
| InternalServerError.FailAccessDatabase      | 访问数据库失败。          |
| InternalServerError.FailedVoicePrintDecode  | 说话人音频解码失败         |
| InternalServerError.FailedVoicePrintEnroll  | 说话人注册接口失败         |
| InternalServerError.VoicePrintAudioFailed   | 说话人音频处理失败         |
| InternalServerError.VoicePrintDecodeFailed  | 传入音频解码失败          |
| InternalServerError.VoicePrintEnrollFailed  | 说话人ID注册失败         |
| InternalServerError.VoicePrintVerifyFailed  | 说话人验证失败           |
| InvalidParameter                            | 参数错误。             |
| InvalidParameterValue.ErrorVoicedataTooLong | 音频时长超过限制。         |
| InvalidParameterValue.NoHumanVoice          | 音频内容没有人声或有效人声小于1秒 |
| LimitExceeded.VoicePrintFull                | 说话人ID创建数量达到上限     |
| MissingParameter                            | 缺少参数错误。           |

# 说话人认证

最近更新时间：2025-05-09 01:09:52

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

本接口用于校验传入音频与已注册音频的匹配程度，通过指定说话人ID（`VoicePrintId`）和一段音频进行音频和说话人的匹配度判断

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  | <a href="#">公共参数</a> ，本接口取值： <code>VoicePrintVerify</code> 。                 |
| Version      | 是  | String  | <a href="#">公共参数</a> ，本接口取值： <code>2019-06-14</code> 。                       |
| Region       | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                                              |
| VoiceFormat  | 是  | Integer | 音频格式 0: pcm, 1: wav<br>示例值：0                                                 |
| SampleRate   | 是  | Integer | 音频采样率，目前支持16000，单位：Hz，必填<br>示例值：16000                                        |
| Data         | 是  | String  | 音频数据, base64 编码, 音频时长不能超过30s，数据大小不超过2M<br>示例值： <code>ADKFG****ALDHASG</code> |
| VoicePrintId | 是  | String  | 说话人id, 说话人唯一标识<br>示例值： <code>dasdagfd****sdsada</code>                       |

## 3. 输出参数

| 参数名称      | 类型                                   | 描述                                                                                     |
|-----------|--------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">VoicePrintVerifyData</a> | 说话人验证数据                                                                                |
| RequestId | String                               | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 说话人验证

验证音频和说话人id是否一致

输入示例



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

{
  "VoiceFormat": 0,
  "SampleRate": 16000,
  "VoicePrintId": "KJHGUGKGGFGYFVDA",
  "Data": "UklGRiRTBwBXQVZFZm10IBAA"
}
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "VoicePrintId": "KJHGUG***GGFGYFVDA",
      "Score": "60.0",
      "Decision": 1
    },
    "RequestId": "dasdasdadsdas****dasdada"
  }
}
```

## 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.NotExistentVoicePrintId     | 不存在的话人id            |
| FailedOperation.UserHasNoFreeAmount         | 资源包耗尽，请开通后付费或者购买资源包 |
| InternalServerError                         | 内部错误。               |
| InternalServerError.ErrorGetRoute           | 获取路由失败。             |
| InternalServerError.ErrorRecognize          | 识别失败。               |
| InternalServerError.FailAccessDatabase      | 访问数据库失败。            |
| InternalServerError.FailedVoicePrintDecode  | 说话人音频解码失败           |
| InternalServerError.FailedVoicePrintVerify  | 说话人验证接口失败           |
| InternalServerError.VoicePrintAudioFailed   | 说话人音频处理失败           |
| InternalServerError.VoicePrintDecodeFailed  | 传入音频解码失败            |
| InternalServerError.VoicePrintEnrollFailed  | 说话人ID注册失败           |
| InternalServerError.VoicePrintVerifyFailed  | 说话人验证失败             |
| InvalidParameter                            | 参数错误。               |
| InvalidParameterValue                       | 参数取值错误。             |
| InvalidParameterValue.ErrorVoicedataTooLong | 音频时长超过限制。           |
| InvalidParameterValue.NoHumanVoice          | 音频内容没有人声或有效人声小于1秒   |

# 说话人更新

最近更新时间：2025-05-09 01:09:52

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

本接口用于更新和覆盖已注册的音频数据和说话人昵称，更新后原有的音频数据将失效。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

## 2. 输入参数

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

| 参数名称         | 必选 | 类型      | 描述                                                         |
|--------------|----|---------|------------------------------------------------------------|
| Action       | 是  | String  | <a href="#">公共参数</a> ，本接口取值：VoicePrintUpdate。              |
| Version      | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                    |
| Region       | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                            |
| VoiceFormat  | 是  | Integer | 音频格式 0: pcm, 1: wav<br>示例值：0                               |
| SampleRate   | 是  | Integer | 音频采样率 目前仅支持16000 单位Hz<br>示例值：16000                         |
| VoicePrintId | 是  | String  | 说话人id，说话人唯一标识<br>示例值：sdsafjasj*sdfasfjj                    |
| Data         | 是  | String  | 音频数据, base64 编码, 音频时长不能超过30s，数据大小不超过2M<br>示例值：ADDAFASAFASA |
| SpeakerNick  | 否  | String  | 说话人昵称 不超过32字节<br>示例值：小明                                    |

## 3. 输出参数

| 参数名称      | 类型                                 | 描述                                                                                     |
|-----------|------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">VoicePrintBaseData</a> | 说话人基础数据                                                                                |
| RequestId | String                             | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 说话人更新

更新说话人的音频和昵称

#### 输入示例

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

{
  "VoiceFormat": 0,
  "SampleRate": 0,
  "SpeakerNick": "小明",
  "VoicePrintId": "34a0a2b-922f*****90302f155a6d",
  "Data": "UklGRka1CQBxQVZFZm10IBAAAAABAAEAgD4E"
}
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "VoicePrintId": "34a0a2b-922f*****90302f155a6d",
      "SpeakerNick": "小明"
    },
    "RequestId": "676a22c*****3625b0"
  }
}
```

## 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.NotExistentVoicePrintId     | 不存在的话人id          |
| InternalServerError                         | 内部错误。             |
| InternalServerError.ErrorGetRoute           | 获取路由失败。           |
| InternalServerError.ErrorRecognize          | 识别失败。             |
| InternalServerError.FailAccessDatabase      | 访问数据库失败。          |
| InternalServerError.FailedVoicePrintEnroll  | 说话人注册接口失败         |
| InternalServerError.VoicePrintAudioFailed   | 说话人音频处理失败         |
| InternalServerError.VoicePrintDecodeFailed  | 传入音频解码失败          |
| InternalServerError.VoicePrintEnrollFailed  | 说话人ID注册失败         |
| InternalServerError.VoicePrintVerifyFailed  | 说话人验证失败           |
| InvalidParameter                            | 参数错误。             |
| InvalidParameterValue.ErrorVoicedataTooLong | 音频时长超过限制。         |
| InvalidParameterValue.NoHumanVoice          | 音频内容没有人声或有效人声小于1秒 |
| MissingParameter                            | 缺少参数错误。           |

# 说话人删除

最近更新时间：2025-05-09 01:09:52

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

本接口用于以删除已经注册的说话人信息（删除之后，原有的说话人ID和说话人音频数据都会失效）

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

## 2. 输入参数

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

| 参数名称         | 必选 | 类型      | 描述                                                                                         |
|--------------|----|---------|--------------------------------------------------------------------------------------------|
| Action       | 是  | String  | <a href="#">公共参数</a> ，本接口取值：VoicePrintDelete。                                              |
| Version      | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                    |
| Region       | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                                                            |
| VoicePrintId | 否  | String  | 说话人id，说话人唯一标识<br>示例值：sdsafjasj*sdfasfjj                                                    |
| GroupId      | 否  | String  | 说话人分组ID,仅支持大小写字母和下划线的组合，不超过128个字符<br>示例值：分组1                                               |
| DelMod       | 否  | Integer | 删除模式：<br>0.默认值，删除该条声纹<br>1.从分组中删除该条声纹，声纹本身不删除<br>2.从声纹库中删除分组，仅删除分组信息，不会真正删除分组中的声纹<br>示例值：0 |

## 3. 输出参数

| 参数名称      | 类型                                 | 描述                                                                                     |
|-----------|------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">VoicePrintBaseData</a> | 说话人基本信息                                                                                |
| RequestId | String                             | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 说话人删除

删除保存的说话人信息

输入示例

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

{
  "VoicePrintId": "34a0a2b-922f*****90302f155a6d"
}
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "VoicePrintId": "34a0a2b-922f*****90302f155a6d",
      "SpeakerNick": "小明"
    },
    "RequestId": "676a22c*****3625b0"
  }
}
```

## 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.NotExistentVoicePrintId    | 不存在的话人id  |
| InternalServerError                        | 内部错误。     |
| InternalServerError.ErrorGetRoute          | 获取路由失败。   |
| InternalServerError.ErrorRecognize         | 识别失败。     |
| InternalServerError.FailAccessDatabase     | 访问数据库失败。  |
| InternalServerError.VoicePrintAudioFailed  | 说话人音频处理失败 |
| InternalServerError.VoicePrintDecodeFailed | 传入音频解码失败  |
| InternalServerError.VoicePrintEnrollFailed | 说话人ID注册失败 |
| InternalServerError.VoicePrintVerifyFailed | 说话人验证失败   |
| InvalidParameter                           | 参数错误。     |
| MissingParameter                           | 缺少参数错误。   |



# 说话人比对

最近更新时间：2025-04-25 01:09:08

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

通过比对两段音频内说话人的声纹，得到一个打分，可通过打分判断两段音频声纹相似度, 打分区间[0 – 100]。音频要求：16k采样率，16bit位深，pcm或者wav格式，单声道，总时长不超过30秒的音频，base64编码数据大小不超过2M，音频内容只有一个说话人声音，并且尽可能清晰，这样结果更加准确。

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  | <a href="#">公共参数</a> ，本接口取值：VoicePrintCompare。                                                                                        |
| Version       | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                               |
| Region        | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                                                                                                       |
| VoiceFormat   | 是  | Integer | 音频格式 0: pcm, 1: wav；pcm和wav音频无损压缩，识别准确度更高<br>示例值：0                                                                                    |
| SampleRate    | 是  | Integer | 音频采样率，目前仅支持16k，请填写16000<br>示例值：16000                                                                                                  |
| SrcAudioData  | 是  | String  | 对比源音频数据, 音频要求：base64 编码,16k采样率，16bit位深，pcm或者wav格式，单声道，音频时长不超过30秒的音频，base64编码数据大小不超过2M<br>示例值：UklGRka1CQBXQVZFZm10zOjQzAABJU0ZUDgAAAE  |
| DestAudioData | 是  | String  | 对比目标音频数据, 音频要求：base64 编码,16k采样率，16bit位深，pcm或者wav格式，单声道，音频时长不超过30秒的音频，base64编码数据大小不超过2M<br>示例值：UklGRka1CQBXQVZFZm10zOjQzAABJU0ZUDgAAAE |

## 3. 输出参数

| 参数名称      | 类型                                    | 描述                                                                                     |
|-----------|---------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">VoicePrintCompareData</a> | 音频声纹比对结果，包含相似度打分                                                                       |
| RequestId | String                                | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

## 示例1 音频声纹比对

比对两段音频的声纹相似度

### 输入示例

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

{
  "VoiceFormat": 0,
  "SampleRate": 16000,
  "SrcAudioData": "AAAAADADAAAA",
  "DestAudioData": "AAAAADADAAAA"
}
```

### 输出示例

```
{
  "Response": {
    "Data": {
      "Score": "100.0",
      "Decision": "1"
    },
    "RequestId": "dsadasdasdasdasd"
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

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

- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

## 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                                         | 描述                |
|---------------------------------------------|-------------------|
| InternalServerError                         | 内部错误。             |
| InternalServerError.VoicePrintAudioFailed   | 说话人音频处理失败         |
| InternalServerError.VoicePrintDecodeFailed  | 传入音频解码失败          |
| InternalServerError.VoicePrintVerifyFailed  | 说话人验证失败           |
| InvalidParameter                            | 参数错误。             |
| InvalidParameter.InvalidParameter           | 参数错误。             |
| InvalidParameterValue.ErrorInvalidVoicedata | 音频数据无效。           |
| InvalidParameterValue.NoHumanVoice          | 音频内容没有人声或有效人声小于1秒 |

# 说话人注册数量统计

最近更新时间：2025-05-09 01:09:52

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

统计并返回注册的说话人id总数

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  | <a href="#">公共参数</a> ，本接口取值：VoicePrintCount。              |
| Version  | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                   |
| Region   | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                           |
| GroupId  | 否  | String  | 分组ID,仅支持大小写字母和下划线的组合，不超过128个字符<br>示例值：sdasf***dasdaffsasd |
| CountMod | 否  | Integer | 统计模式<br>0: 统计所有声纹数量<br>1: 统计指定分组下的声纹数量<br>示例值：0           |

## 3. 输出参数

| 参数名称      | 类型                                  | 描述                                                                                     |
|-----------|-------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">VoicePrintCountData</a> | 统计数据                                                                                   |
| RequestId | String                              | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 示例

返回注册总数

输入示例

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

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

{ }
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "Total": 1
    },
    "RequestId": "676a22c*****3625b0"
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

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

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                 | 描述    |
|---------------------|-------|
| InternalServerError | 内部错误。 |

| 错误码              | 描述    |
|------------------|-------|
| InvalidParameter | 参数错误。 |

# 说话人验证1:N

最近更新时间：2025-04-25 01:09:08

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

说话人验证1:N接口，可以通过传入一段说话人音频，并且指定已存在的`groupId`，和返回`topN`，接口返回`groupId`内所有声纹和传入音频声纹比对打分`TopN`的结果。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

## 2. 输入参数

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

| 参数名称        | 必选 | 类型      | 描述                                                                                     |
|-------------|----|---------|----------------------------------------------------------------------------------------|
| Action      | 是  | String  | <a href="#">公共参数</a> ，本接口取值：VoicePrintGroupVerify。                                     |
| Version     | 是  | String  | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                |
| Region      | 否  | String  | <a href="#">公共参数</a> ，此参数为可选参数。                                                        |
| VoiceFormat | 是  | Integer | 音频格式 0: pcm, 1: wav<br>示例值：0                                                           |
| SampleRate  | 是  | Integer | 音频采样率，目前支持16000，单位：Hz，必填<br>示例值：16000                                                  |
| Data        | 是  | String  | 音频数据, base64 编码, 音频时长不能超过30s，数据大小不超过2M<br>示例值：UklGRka1CQBXQVZFZm10zOjQzAABJU0ZUDgAAAE* |
| GroupId     | 是  | String  | 分组id, 支持数字，字母，下划线，长度不超过128<br>示例值：home_group                                           |
| TopN        | 是  | Integer | 返回打分结果降序排列topN, TopN大于0， 小于可创建声纹最大数量<br>示例值：1                                          |

## 3. 输出参数

| 参数名称      | 类型                              | 描述                                                                                     |
|-----------|---------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">VerifyTopResult</a> | TopN 返回结果;系统建议打分70分以上为同一个人音色，评分也取决于音频质量、长度等其他原因影响，您可以按照业务需求适当提高或降低分数要求                 |
| RequestId | String                          | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

## 示例1 说话人验证1:N

提供一段音频数据和提前注册的一组说话人数据进行比对，返回topN

### 输入示例

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

{
  "VoiceFormat": 1,
  "SampleRate": 16000,
  "Data": "UklGRsb7AQBxQVZFZm10IBAAAAAB",
  "GroupId": "test",
  "TopN": 2
}
```

### 输出示例

```
{
  "Response": {
    "Data": {
      "VerifyTops": [
        {
          "Score": "100.0",
          "SpeakerId": "张三",
          "VoicePrintId": "03c7c4-34e1-4cc2-97d4-f031bc6538d0"
        },
        {
          "Score": "89.0",
          "SpeakerId": "李四",
          "VoicePrintId": "16a936-c0c9-44b8-81e6-19515a9515ef"
        }
      ]
    },
    "RequestId": "affe4cc1-7158-ad-aefc-371fce09c092"
  }
}
```

## 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.UserHasNoFreeAmount         | 资源包耗尽，请开通后付费或者购买资源包 |
| InternalServerError                         | 内部错误。               |
| InternalServerError.FailAccessDatabase      | 访问数据库失败。            |
| InternalServerError.FailedVoicePrintDecode  | 说话人音频解码失败           |
| InternalServerError.FailedVoicePrintVerify  | 说话人验证接口失败           |
| InternalServerError.VoicePrintAudioFailed   | 说话人音频处理失败           |
| InternalServerError.VoicePrintDecodeFailed  | 传入音频解码失败            |
| InternalServerError.VoicePrintVerifyFailed  | 说话人验证失败             |
| InvalidParameter                            | 参数错误。               |
| InvalidParameter.FailedVoicePrintDecode     | 音频解码失败，请检查音频格式是否正确  |
| InvalidParameter.InvalidParameter           | 参数错误。               |
| InvalidParameterValue                       | 参数取值错误。             |
| InvalidParameterValue.ErrorVoicedataTooLong | 音频时长超过限制。           |
| InvalidParameterValue.NoHumanVoice          | 音频内容没有入声或有效入声小于1秒   |

# 关键字检索相关接口

## 创建关键词表

最近更新时间：2025-04-25 01:09:12

### 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过本接口进行关键字词表的创建。

- 默认每个用户最多可创建30个关键字词表。
- 每个关键词词表最多可添加100个词，每个词最多5个汉字或15个字符。
- 词表通过本地文件形式上传。
- 本地文件必须为UTF-8编码格式，每行仅添加一个词且不能包含标点和特殊字符。

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 | <a href="#">公共参数</a> ，本接口取值： <code>CreateAsrKeyWordLib</code> 。                                                                                      |
| Version     | 是  | String | <a href="#">公共参数</a> ，本接口取值： <code>2019-06-14</code> 。                                                                                               |
| Region      | 是  | String | <a href="#">公共参数</a> ，详见产品支持的 <a href="#">地域列表</a> 。                                                                                                 |
| Name        | 是  | String | 词表名称，长度在1~20之间<br>仅限中英文数字-_<br>示例值：测试词表                                                                                                              |
| KeyWordFile | 是  | String | 词文件（纯文本文件）的二进制base64编码，以行分隔<br>格式要求：TXT<br>每行只有一个词，不满足格式则报错无法上传<br>每个词限制 <b>5个汉字，15个字符</b> ，单个词库最多不超过100个词<br>注意不要有空行，尤其是最后一行<br>示例值：避风港(进行base64编码) |

### 3. 输出参数

| 参数名称      | 类型                               | 描述                                                                                     |
|-----------|----------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">KeyWordLibIdData</a> | 词表ID数据<br>示例值：空                                                                        |
| RequestId | String                           | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 创建关键词表

用户通过上传文件方式创建关键词表

#### 输入示例

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

{
  "Name": "词表名",
  "KeyWordFile": "<文件base64>"
}
```

#### 输出示例

```
{
  "Response": {
    "Data": {
      "KeyWordLibId": "aa6f402f263f1*****fc81fbecfd0sd"
    },
    "RequestId": "b3808ad3-d8dd-4b65-96c9-7d6f1f81b6b6"
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

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

## 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                                          | 描述                    |
|----------------------------------------------|-----------------------|
| FailedOperation                              | 操作失败。                 |
| FailedOperation.ServiceIsolate               | 账号因为欠费停止服务，请在腾讯云账户充值。 |
| InternalServerError                          | 内部错误。                 |
| InternalServerError.FailAccessDatabase       | 访问数据库失败。              |
| InvalidParameter                             | 参数错误。                 |
| InvalidParameter.ErrorContentlength          | 请求数据长度无效。             |
| InvalidParameter.ErrorParamsMissing          | 参数不全。                 |
| InvalidParameter.KeyWordLibNameExist         | 关键词库名字已存在             |
| InvalidParameterValue                        | 参数取值错误。               |
| InvalidParameterValue.ErrorInvalidAppid      | AppId无效。              |
| InvalidParameterValue.InvalidParameterLength | 非法的参数长度。              |
| LimitExceeded                                | 超过配额限制。               |
| MissingParameter                             | 缺少参数错误。               |
| UnknownParameter                             | 未知参数错误。               |

# 删除关键词表

最近更新时间：2025-04-25 01:09:12

## 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

用户通过本接口进行关键词表的删除。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

## 2. 输入参数

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

| 参数名称         | 必选 | 类型     | 描述                                                   |
|--------------|----|--------|------------------------------------------------------|
| Action       | 是  | String | <a href="#">公共参数</a> ，本接口取值：DeleteAsrKeyWordLib。     |
| Version      | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。              |
| Region       | 是  | String | <a href="#">公共参数</a> ，详见产品支持的 <a href="#">地域列表</a> 。 |
| KeyWordLibId | 是  | String | 关键词表ID<br>示例值：ed9d5b5a9fa14f4a86dc71360e6a3835       |

## 3. 输出参数

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

## 4. 示例

### 示例1 删除关键词表

用户通过该API，传入词表ID，可以删除对应的关键词表

#### 输入示例

```
https://asr.tencentcloudapi.com/?Action=DeleteAsrKeyWordLib
&KeyWordLibId=aa6f402f263f12ea856fc81fbecfd0sd
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "7eccebc9-2d60-4cee-8fa8-7f8001a71ab2"
  }
}
```

## 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.FailAccessDatabase  | 访问数据库失败。  |
| InvalidParameter                        | 参数错误。     |
| InvalidParameter.ErrorContentlength     | 请求数据长度无效。 |
| InvalidParameter.ErrorParamsMissing     | 参数不全。     |
| InvalidParameter.ErrorParsequest        | 解析请求数据失败。 |
| InvalidParameterValue                   | 参数取值错误。   |
| InvalidParameterValue.ErrorInvalidAppid | AppId无效。  |

# 列举关键词表

最近更新时间：2025-05-09 01:09:55

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过该接口，可获得所有的关键词表及其信息。

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          | <a href="#">公共参数</a> ，本接口取值：GetAsrKeyWordLibList。    |
| Version        | 是  | String          | <a href="#">公共参数</a> ，本接口取值：2019-06-14。              |
| Region         | 是  | String          | <a href="#">公共参数</a> ，详见产品支持的 <a href="#">地域列表</a> 。 |
| Offset         | 否  | Integer         | 分页Offset<br>示例值：0                                    |
| Limit          | 否  | Integer         | 分页Limit<br>示例值：10                                    |
| SpecifyNames.N | 否  | Array of String | 词库名称或者UIN检索<br>示例值：腾讯云                               |
| OnlySelf       | 否  | Boolean         | 只看用户自己创建的<br>示例值：0                                   |

## 3. 输出参数

| 参数名称      | 类型                                 | 描述                                                                                     |
|-----------|------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">KeyWordLibListData</a> | 关键词列表返回数据                                                                              |
| RequestId | String                             | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 列举关键词表

用户通过该接口，可获得所有的关键词表及其信息。

输入示例

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

### 输出示例

```
{
  "Response": {
    "Data": {
      "KeyWordLibList": [
        {
          "KeyWordLibId": "aa6f402f263f*****56fc81fbecfd0sd",
          "Name": "示例词表",
          "KeyWordList": [
            "避风港"
          ],
          "CreateTime": "2024-08-24T13:28:48+00:00",
          "UpdateTime": "2024-08-24T13:28:48+00:00"
        }
      ],
      "TotalCount": 1
    },
    "RequestId": "b3808ad3-d8dd-4b65-96c9-7d6f1f81b6b6"
  }
}
```

## 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.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| AuthFailure.UnauthorizedOperation          | 未授权操作。         |
| FailedOperation                            | 操作失败。          |
| InternalError                              | 内部错误。          |
| InternalError.FailAccessDatabase           | 访问数据库失败。       |
| InvalidParameter                           | 参数错误。          |
| InvalidParameterValue                      | 参数取值错误。        |
| MissingParameter                           | 缺少参数错误。        |

# 更新关键词表

最近更新时间：2025-04-25 01:09:11

## 1. 接口描述

接口请求域名：`asr.tencentcloudapi.com`。

用户通过本接口进行对应的关键词表信息更新。

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 | <a href="#">公共参数</a> ，本接口取值：UpdateAsrKeyWordLib。                                                                                           |
| Version      | 是  | String | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                                                                                    |
| Region       | 是  | String | <a href="#">公共参数</a> ，详见产品支持的 <a href="#">地域列表</a> 。                                                                                       |
| KeyWordLibId | 是  | String | 关键词表ID<br>示例值：aa6f402**2ea856fc81fbecfd0sd                                                                                                 |
| Name         | 是  | String | 词表名称，长度在1-20之间<br>仅限中英文数字-_<br>示例值：新的词表名称                                                                                                  |
| KeyWordFile  | 是  | String | - 词文件（纯文本文件）以行分隔，进行二进制base64编码<br>- 格式要求：TXT 每行只有一个词，不满足格式则报错无法上传<br>- 每个词最多5个汉字或15个字符，单个词库最多不超过100个词<br>- 此参数为空则只更新词表名称<br>示例值：<文件base64> |

## 3. 输出参数

| 参数名称      | 类型                               | 描述                                                                                     |
|-----------|----------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">KeyWordLibIdData</a> | 关键词表ID数据                                                                               |
| RequestId | String                           | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

## 4. 示例

### 示例1 更新关键词表

用户通过该API传入已经创建的关键词表ID，可以更新其相应的内容

## 输入示例

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

{
  "KeyWordLibId": "aa6f402*****2ea856fc81fbecfd0sd",
  "Name": "词表名",
  "KeyWordFile": "<文件base64>"
}
```

## 输出示例

```
{
  "Response": {
    "Data": {
      "KeyWordLibId": "aa6f402f*****ea856fc81fbecfd0sd"
    },
    "RequestId": "3290cfb2-91a7-4989-91de-d1d624a77c12"
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

### SDK

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

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

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

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

| 错误码                                          | 描述                    |
|----------------------------------------------|-----------------------|
| FailedOperation                              | 操作失败。                 |
| FailedOperation.ServiceIsolate               | 账号因为欠费停止服务，请在腾讯云账户充值。 |
| InternalError                                | 内部错误。                 |
| InternalError.FailAccessDatabase             | 访问数据库失败。              |
| InvalidParameter                             | 参数错误。                 |
| InvalidParameter.KeyWordLibNameExist         | 关键词库名字已存在             |
| InvalidParameterValue                        | 参数取值错误。               |
| InvalidParameterValue.InvalidParameterLength | 非法的参数长度。              |
| MissingParameter                             | 缺少参数错误。               |
| UnknownParameter                             | 未知参数错误。               |

# 业务用量相关接口

## 通过日期查询用量

最近更新时间：2025-04-25 01:09:12

### 1. 接口描述

接口请求域名：asr.tencentcloudapi.com。

查询用户用量

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          | <a href="#">公共参数</a> ，本接口取值：GetUsageByDate。                                       |
| Version       | 是  | String          | <a href="#">公共参数</a> ，本接口取值：2019-06-14。                                           |
| Region        | 是  | String          | <a href="#">公共参数</a> ，详见产品支持的 <a href="#">地域列表</a> 。                              |
| BizNameList.N | 是  | Array of String | 需要查询的业务类型名字列表<br>- asr_rt 实时识别<br>- asr_rec 录音文件识别<br>示例值：["asr_rec"]             |
| StartDate     | 是  | String          | 查询开始时间<br>开始时间包含当天，支持 YYYY-MM-DD 日期以国内时区为准<br>开始时间到结束时间需要在3个月以内<br>示例值：2024-09-01 |
| EndDate       | 是  | String          | 查询结束时间<br>结束时间包含当天，支持 YYYY-MM-DD 日期以国内时区为准<br>开始时间到结束时间需要在3个月以内<br>示例值：2024-09-06 |

### 3. 输出参数

| 参数名称      | 类型                                  | 描述                                                                                     |
|-----------|-------------------------------------|----------------------------------------------------------------------------------------|
| Data      | <a href="#">UsageByDateInfoData</a> | 用量次数                                                                                   |
| RequestId | String                              | 唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。 |

### 4. 示例

## 示例1 用量查询输入示例

需要查询8月份录音识别和实时识别的用量

### 输入示例

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

{
  "BizNameList": [
    "asr_rt",
    "asr_rec"
  ],
  "StartDate": "2024-08-01",
  "EndDate": "2024-08-31"
}
```

### 输出示例

```
{
  "Response": {
    "Data": {
      "UsageByDateInfoList": [
        {
          "BizName": "asr_rt",
          "Count": 0,
          "Duration": 0
        },
        {
          "BizName": "asr_rec",
          "Count": 47,
          "Duration": 351
        }
      ]
    },
    "RequestId": "877e8723-792b-4b4e-a8c0-57bece96b7f0"
  }
}
```

## 5. 开发者资源

### 腾讯云 API 平台

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

### API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

| 错误码                   | 描述      |
|-----------------------|---------|
| InternalServerError   | 内部错误。   |
| InvalidParameter      | 参数错误。   |
| InvalidParameterValue | 参数取值错误。 |
| LimitExceeded         | 超过配额限制。 |

# 实时语音识别相关接口

## 实时语音识别（WebSocket）

最近更新时间：2025-07-09 18:19:21

 **注意：**

此接口为实时语音识别接口，在参数风格、错误码等方面有区别于云 API 接口，请知悉。

### 接口描述

本接口服务采用 WebSocket 协议，对实时音频流进行识别，同步返回识别结果，达到“边说边出文字”的效果。

在使用该接口前，需要 [开通语音识别服务](#)，并进入 [API 密钥管理页面](#) 新建密钥，生成 AppID、SecretID 和 SecretKey，用于 API 调用时生成签名，签名将用来进行接口鉴权。

### 接口要求

集成实时语音识别 API 时，需按照以下要求。

| 内容   | 说明                                                                                                                                                                                                         |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语言种类 | 支持中文普通话、粤语、英语、韩语、日语、泰语、印度尼西亚语、越南语、马来语、菲律宾语、葡萄牙语、土耳其语、阿拉伯语、西班牙语、印地语、法语、德语、上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话。可通过接口参数 engine_model_type 设置对应语言类型。 |
| 支持行业 | 通用、金融、游戏、教育、医疗。                                                                                                                                                                                            |
| 音频属性 | <ul style="list-style-type: none"><li>采样率：16000Hz 或 8000Hz</li><li>采样精度：16bits</li><li>声道：单声道（mono）</li></ul>                                                                                              |
| 音频格式 | pcm、wav、opus、speex、silk、mp3、m4a、aac                                                                                                                                                                        |
| 请求协议 | wss 协议                                                                                                                                                                                                     |
| 请求地址 | wss://asr.cloud.tencent.com/asr/v2/<appid>?{请求参数}                                                                                                                                                          |
| 接口鉴权 | 签名鉴权机制，详情请参见 <a href="#">签名生成</a> 。                                                                                                                                                                        |
| 响应格式 | 统一采用 JSON 格式。                                                                                                                                                                                              |
| 数据发送 | <ul style="list-style-type: none"><li>建议每 40ms 发送 40ms 时长（即 1:1 实时率）的数据包，对应 PCM 大小为：8k 采样率 640 字节，16k 采样率 1280 字节。</li><li>音频发送速率过快超过 1:1 实时率或者音频数据包之间发送间隔超过 6 秒，可能导致引擎出错，后台将返回错误并主动断开连接。</li></ul>        |
| 并发限制 | 默认单账号限制并发数为 200 路，如您有提高并发限制的需求，请 <a href="#">前往购买</a> 。                                                                                                                                                    |

### 接口调用流程

接口调用流程分为两个阶段：握手阶段和识别阶段。两阶段后台均返回 text message，内容为 JSON 序列化字符串，以下是格式说明：

| 字段名  | 类型      | 描述                      |
|------|---------|-------------------------|
| code | Integer | 状态码，0 代表正常，非 0 值表示发生错误。 |



|            |         |                                                                                                                                                                                           |
|------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| message    | String  | 错误说明，发生错误时显示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。                                                                                                                                      |
| voice_id   | String  | 音频流全局唯一标识，一个 WebSocket 连接对应一个，用户自己生成（推荐使用 UUID），最长128位。<br><b>注意：</b> 每次建立 WebSocket 连接都必须重新生成 voice_id，任何情况下中断了识别或者 WebSocket 连接(如连接失败、正常结束、识别返回错误码等)，voice_id 作废，重新发起识别必须使用新的 voice_id。 |
| message_id | String  | 本 message 唯一 ID。                                                                                                                                                                          |
| result     | Result  | 最新语音识别结果。                                                                                                                                                                                 |
| final      | Integer | 该字段返回1时表示音频流全部识别结束。                                                                                                                                                                       |

其中识别结果 Result 结构体格式为：

| 字段名            | 类型         | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| slice_type     | Integer    | <p>识别结果类型：</p> <ul style="list-style-type: none"> <li>0：一段话开始识别。</li> <li>1：一段话识别中，voice_text_str 为非稳态结果（该段识别结果还可能变化）。</li> <li>2：一段话识别结束，voice_text_str 为稳态结果（该段识别结果不再变化）。</li> </ul> <p>根据发送的音频情况，识别过程中可能返回的 slice_type 序列有：</p> <ul style="list-style-type: none"> <li>0-1-2：一段话开始识别、识别中（可能有多次1返回）、识别结束。</li> <li>0-2：一段话开始识别、识别结束。</li> <li>2：直接返回一段话完整的识别结果。</li> </ul> <p><b>注意：</b>如果需要0和2配对返回，需要设置 filter_empty_result=0（slice_type=0时，识别结果可能为空，默认是不返回空识别结果的）。一般在外呼场景需要配对返回，通过 slice_type=0 来判断是否有人声出现。<br/>示例值：0</p> |
| index          | Integer    | 当前一段话结果在整个音频流中的序号，从0开始逐句递增。<br>示例值：0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| start_time     | Integer    | 当前一段话结果在整个音频流中的起始时间。<br>示例值：60                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| end_time       | Integer    | 当前一段话结果在整个音频流中的结束时间。<br>示例值：2700                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| voice_text_str | String     | 当前一段话文本结果，编码为 UTF8。<br>示例值：ASR 语音识别结果。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| word_size      | Integer    | 当前一段话的词结果个数。<br>示例值：1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| word_list      | Word Array | <p>当前一段话的词列表，Word 结构体格式为：</p> <ul style="list-style-type: none"> <li>word：String 类型，该词的内容。</li> <li>start_time：Integer 类型，该词在整个音频流中的起始时间。</li> <li>end_time：Integer 类型，该词在整个音频流中的结束时间。</li> <li>stable_flag：Integer 类型，该词的稳态结果，0表示该词在后续识别中可能发生变化，1表示该词在后续识别过程中不会变化。</li> </ul> <p>• 示例值：[{"word":"我","start_time":380,"end_time":680,"stable_flag":1}]</p>                                                                                                                                                                    |

## 握手阶段

### 请求格式

握手阶段，客户端主动发起 WebSocket 连接请求，请求 URL 格式为：

```
wss://asr.cloud.tencent.com/asr/v2/<appid>?{请求参数}
```

其中 <appid> 需替换为腾讯云注册账号的 AppID，可通过 [API 密钥管理页面](#) 获取，{请求参数}格式为：

```
key1=value2&key2=value2...
```

参数说明：

| 参数名称              | 必填 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|----|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| secretid          | 是  | String  | 腾讯云注册账号的密钥 secretid，可通过 <a href="#">API 密钥管理页面</a> 获取。<br>示例值：*****Qq1zhZMN8dv0*****                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| timestamp         | 是  | Integer | 当前 UNIX 时间戳，单位为秒。如果与当前时间相差过大，会引起签名过期错误。<br>示例值：1745932688                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| expired           | 是  | Integer | 签名的有效期截止时间 UNIX 时间戳，单位为秒。expired 必须大于 timestamp 且 expired – timestamp 小于90天。<br>示例值：1746019088                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| nonce             | 是  | Integer | 随机正整数。用户需自行生成，最长10位。<br>示例值：8743357                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| engine_model_type | 是  | String  | <p>引擎模型类型</p> <p>识别引擎采用分级计费方案，标记为“大模型版”的引擎适用大模型计费方案，请参见 <a href="#">计费概述（在线版）</a>。</p> <p>电话场景：</p> <ul style="list-style-type: none"><li>8k_zh：中文电话通用。</li><li>8k_en：英文电话通用。</li><li>8k_zh_large：中文电话场景专用大模型引擎【大模型版】。通过显著提升模型参数规模与语言建模能力，实现对电话音频中复杂场景（如口音干扰、背景噪声）的高精度识别，识别准确率较常规版本大幅提升。</li></ul> <p><b>注意：</b>如您有电话通讯场景识别需求，但发现需求语种仅支持 16k，可将 input_sample_rate 参数设置为8000，使用下方 16k 引擎，亦能获取识别结果。但 16k 引擎并非基于电话通讯数据训练，无法承诺此种调用方式在电话场景下的识别效果，需由您自行验证识别结果是否可用。</p> <p>非电话场景：</p> <ul style="list-style-type: none"><li>16k_zh_large：普方英大模型引擎【大模型版】。当前模型同时支持中文、英文、<a href="#">多种中文方言</a> 等语言的识别，模型参数量极大，语言模型性能增强，针对噪声大、回音大、人声小、人声远等低质量音频的识别准确率极大提升；</li><li>16k_multi_lang：多语种大模型引擎【大模型版】。当前模型同时支持英语、日语、韩语、阿拉伯语、菲律宾语、法语、印地语、印尼语、马来语、葡萄牙语、西班牙语、泰语、土耳其语、越南语、德语的识别，可实现15个语种的自动识别(句子/段落级别)；</li><li>16k_zh_en：中英大模型引擎【大模型版】。当前模型同时支持中文、英语识别，模型参数量极大，语言模型性能增强，针对噪声大、回音大、人声小、人声远等低质量音频的识别准确率极大提升；</li><li>16k_zh：中文通用；</li><li>16k_zh-PY：中英粤；</li><li>16k_zh-TW：中文繁体；</li><li>16k_zh_edu：中文教育；</li><li>16k_zh_medical：中文医疗；</li><li>16k_zh_court：中文法庭；</li><li>16k_yue：粤语；</li><li>16k_en：英文通用；</li></ul> |

|                  |   |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|---|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |   |         | <ul style="list-style-type: none"> <li>16k_en_game: 英文游戏;</li> <li>16k_en_edu: 英文教育;</li> <li>16k_ko: 韩语;</li> <li>16k_ja: 日语;</li> <li>16k_th: 泰语;</li> <li>16k_id: 印度尼西亚语;</li> <li>16k_vi: 越南语;</li> <li>16k_ms: 马来语;</li> <li>16k_fil: 菲律宾语;</li> <li>16k_pt: 葡萄牙语;</li> <li>16k_tr: 土耳其语;</li> <li>16k_ar: 阿拉伯语;</li> <li>16k_es: 西班牙语;</li> <li>16k_hi: 印地语;</li> <li>16k_fr: 法语;</li> <li>16k_de: 德语;</li> <li>16k_zh_dialect: 中文普通话 + 多方言混合引擎, 除普通话外支持23种方言 (上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话);</li> </ul> <p>示例值: 16k_zh</p> |
| voice_id         | 是 | String  | <p>音频流全局唯一标识, 一个 WebSocket 连接对应一个, 用户自己生成 (推荐使用 uuid), 最长128位。</p> <p><b>注意:</b> 每次建立 WebSocket 连接都必须重新生成 voice_id, 任何情况下中断了识别或者 WebSocket 连接 (如连接失败、正常结束、识别返回错误码等), voice_id 作废, 重新发起识别必须使用新的 voice_id</p> <p>示例值: TBgVFCsxdn3i2DGt</p>                                                                                                                                                                                                                                                                                                                                                           |
| voice_format     | 否 | Int     | <p>语音编码方式, 可选, 默认值为4。1: pcm; 4: speex(sp); 6: silk; 8: mp3; 10: opus (<a href="#">opus 格式音频流封装说明</a>); 12: wav; 14: m4a (每个分片须是一个完整的 m4a 音频); 16: aac</p> <p>示例值: 1</p>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| needvad          | 否 | Integer | <p>0: 关闭 vad, 1: 开启 vad, 默认为0。</p> <p>为保证识别效果, 如果语音分片长度超过60秒, 会强制在60s断一次, 建议客户音频超过60s时, 开启 vad (人声检测切分功能), 提升切分效果。</p> <p>示例值: 1</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| hotword_id       | 否 | String  | <p>热词表 id。如不设置该参数, 自动生效默认热词表; 如果设置了该参数, 那么将生效对应的热词表。</p> <p>示例值: da3f5f5555cf11eda6da525400aec391</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| customization_id | 否 | String  | <p>自学习模型 id。如设置了该参数, 将生效对应的自学习模型。</p> <p>示例值: 39bc8e96504511edbd76446a2eb5fd98</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| filter_dirty     | 否 | Integer | <p>是否过滤脏词 (目前支持中文普通话引擎)。默认为0。0: 不过滤脏词; 1: 过滤脏词; 2: 将脏词替换为 “* ”</p> <p>示例值: 0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| filter_modal     | 否 | Integer | <p>是否过滤语气词 (目前支持中文普通话引擎)。默认为0。0: 不过滤语气词; 1: 部分过滤; 2: 严格过滤。</p> <p>示例值: 0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| filter_punc      | 否 | Integer | <p>是否过滤句末的句号 (目前支持中文普通话引擎)。默认为0。0: 不过滤句末的句号; 1: 过滤句末的句号。</p> <p>示例值: 0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                     |   |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------|---|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filter_empty_result | 否 | Integer | <p>是否回调识别空结果，默认为1。0：回调空结果；1：不回调空结果。</p> <p><b>注意：</b>如果需要 slice_type=0 和 slice_type=2 配对回调，需要设置 filter_empty_result=0。一般在外呼场景需要配对返回，通过 slice_type=0 来判断是否有人声出现。</p> <p>示例值：1</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| convert_num_mode    | 否 | Integer | <p>是否进行阿拉伯数字智能转换（目前支持中文普通话引擎）。0：不转换，直接输出中文数字，1：根据场景智能转换为阿拉伯数字，3：打开数学相关数字转换。默认值为1</p> <p>示例值：1</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| word_info           | 否 | Int     | <p>是否显示词级别时间戳。0：不显示；1：显示，不包含标点时间戳，2：显示，包含标点时间戳。支持引擎 8k_en、8k_zh、8k_zh_finance、16k_zh、16k_en、16k_ca、16k_zh-TW、16k_ja、16k_wuu-SH、8k_zh_large、16k_zh_large，默认为0</p> <p>示例值：0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| vad_silence_time    | 否 | Integer | <p>语音断句检测阈值，静音时长超过该阈值会被认为断句（多用在智能客服场景，需配合 needvad = 1 使用），取值范围：240-2000（默认1000），单位 ms，此参数建议不要随意调整，可能会影响识别效果，目前仅支持 8k_zh、8k_zh_finance、16k_zh、8k_zh_large、16k_zh_large 引擎模型</p> <p>示例值：1000</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| max_speak_time      | 否 | Integer | <p>强制断句功能，取值范围 5000-90000（单位:毫秒），默认值60000。在连续说话不间断情况下，该参数将实现强制断句（此时结果变成稳态，slice_type=2）。如：游戏解说场景，解说员持续不间断解说，无法断句的情况下，将此参数设置为10000，则将在每10秒收到 slice_type=2 的回调。（目前仅支持8k_zh、16k_zh、8k_zh_large、16k_zh_large 引擎）</p> <p>示例值：60000</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| noise_threshold     | 否 | Float   | <p>噪音参数阈值，默认为0，取值范围：[-1,1]，对于一些音频片段，取值越大，判定为噪音情况越大。取值越小，判定为人声情况越大。</p> <p><b>慎用：可能影响识别效果</b></p> <p>示例值：0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| signature           | 是 | String  | <p>接口签名参数</p> <p>示例值：*****g1JfeBi%2FYnTjyjekxfDA%3D</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| hotword_list        | 否 | String  | <p>临时热词表：该参数用于提升识别准确率。</p> <ul style="list-style-type: none"> <li>单个热词限制：“热词 权重”，单个热词不超过30个字符（最多10个汉字），权重[1-11]或者100，如：“腾讯云 5”或“ASR 11”。</li> <li>临时热词表限制：多个热词用英文逗号分割，最多支持128个热词，如：“腾讯云 10,语音识别 5,ASR 11”。</li> <li>参数 hotword_id（热词表）与 hotword_list（临时热词表）区别： <ul style="list-style-type: none"> <li>hotword_id：热词表。需要先在控制台或接口创建热词表，获得对应hotword_id传入参数来使用热词功能。</li> <li>hotword_list：临时热词表。每次请求时直接传入临时热词表来使用热词功能，云端不保留临时热词表。适用于有极大量热词需求的用户。</li> </ul> </li> </ul> <p><b>注意：</b></p> <ul style="list-style-type: none"> <li>如果同时传入了 hotword_id 和 hotword_list，只有 hotword_list 生效。</li> <li>热词权重设置为11时，当前热词将升级为超级热词，建议仅将重要且必须生效的热词设置到11，设置过多权重为11的热词将影响整体字准确率。</li> <li>热词权重设置为100时，当前热词开启热词增强同音替换功能（仅支持 8k_zh,16k_zh），举例：热词配置“蜜制 100”时，与“蜜制”同拼音（mizhi）的“秘制”的识别结果会被强制替换成“蜜制”。因此建议客户根据自己的实际情况开启该功能。建议仅将重要且必须生效的热词设置到100，设置过多权重为100的热词将影响整体字准确率。</li> <li>热词不能包含空格，如：ASR 腾讯云</li> </ul> <p>示例值：语音助理 10</p> |

|                     |   |         |                                                                                                                                                                                                                                                                                                                                                |
|---------------------|---|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| input_sample_rate   | 否 | Integer | <p>支持 PCM 格式的 8k 音频在与引擎采样率不匹配的情况下升采样到 16k 后识别，能有效提升识别准确率。仅支持：8000。如：传入 8000，则 PCM 音频采样率为 8k，当引擎选用 16k_zh，那么该 8k 采样率的 PCM 音频可以在 16k_zh 引擎下正常识别。</p> <p><b>注意：</b>此参数仅适用于 PCM 格式音频，不传入值将维持默认状态，即默认调用的引擎采样率等于 PCM 音频采样率。<br/>示例值：8000</p>                                                                                                         |
| emotion_recognition | 否 | Integer | <p><b>增值付费功能情绪识别能力（目前仅支持 16k_zh, 8k_zh）</b><br/>0：不开启。<br/>1：开启情绪识别，但不在文本展示情绪标签。<br/>2：开启情绪识别，并且在文本展示情绪标签。<br/>默认值为 0<br/>支持的情绪分类为：高兴、伤心、愤怒。</p> <p><b>注意：</b><br/>1. 本功能为增值服务，需将参数设置为 1 或 2 时方可按对应方式生效。<br/>2. 如果传入参数值 1 或 2，需确保账号已购买情绪识别资源包，或账号开启后付费；若当前账号已开启后付费功能，并传入参数值 1 或 2，将自动计费。<br/>3. 参数设置为 0 时，无需购买资源包，也不会消耗情绪识别对应资源。<br/>示例值：0</p> |
| replace_text_id     | 否 | String  | <p>替换词汇表 ID，适用于热词和自学习场景也无法解决的极端 case 词组，会对识别结果强制替换。具体可参考（<a href="#">词汇替换</a>）；强制替换功能可能会影响正常识别结果，请谨慎使用。</p> <p><b>注意：</b><br/>本功能配置完成后，预计在 10 分钟后生效。</p>                                                                                                                                                                                       |

## signature 签名生成

1. 对除 signature 之外的所有参数按字典序进行排序，拼接请求 URL（不包含协议部分：wss://）作为签名原文，这里以 `appid=125922***`，`secretid=*****Qq1zhZMN8dv0*****` 为例拼接签名原文，则拼接的签名原文为：

```
asr.cloud.tencent.com/asr/v2/125922***?
engine_model_type=16k_zh&expired=1673494772&needvad=1&nonce=1673408372&secretid=*****Qq1zhZMN8dv0
*****&timestamp=1673408372&voice_format=1&voice_id=c64385ee-3e5c-4fc5-bbfd-7c71adb35b0
```

2. 对签名原文使用 SecretKey 进行 HMAC-SHA1 加密，之后再使用 base64 编码。例如对上一步的签名原文，`SecretKey=*****SkqpeHqgmSz*****`，使用 HMAC-SHA1 算法进行加密并做 base64 编码处理：

```
Base64Encode(HmacSha1("asr.cloud.tencent.com/asr/v2/125922***?
engine_model_type=16k_zh&expired=1673494772&needvad=1&nonce=1673408372&secretid=*****Qq1zhZMN8dv0
*****&timestamp=1673408372&voice_format=1&voice_id=c64385ee-3e5c-4fc5-bbfd-7c71adb35b0",
"*****SkqpeHqgmSz*****"))
```

得到 signature 签名值为：

```
G8jDQBRg1JfeBi/YnTjyjekxfDA=
```

3. 将 signature 值进行 `urlencode`（必须进行 URL 编码，否则将导致鉴权失败偶现）后拼接得到最终请求 URL 为：

```
wss://asr.cloud.tencent.com/asr/v2/125922***?
engine_model_type=16k_zh&expired=1592380492&filter_dirty=1&filter_modal=1&filter_punc=1&needvad=1
&nonce=1592294092123&secretid=A*****0r&timestamp=1592294092&voice_for
mat=1&voice_id=c64385ee-3e5c-4fc5-bbfd-7c71adb35b0&signature=G8jDQBRg1JfeBi%2FYnTjyjekxfDA%3D
```

## Opus 音频流封装说明

压缩 FrameSize 固定640，即一次压缩640 short，否则解压会失败。传到服务端可以是多帧的拼接组合，每一帧需满足下面格式。每一帧压缩数据封装如下：

| OpusHead ( 4字节 ) | 帧数据长度 ( 2字节 ) | Opus 一帧压缩数据                |
|------------------|---------------|----------------------------|
| Opus             | 长度 len        | 对应 len 长的 Opus encode data |

### 请求响应

客户端发起连接请求后，后台建立连接并进行签名校验，校验成功则返回 code 值为0的确认消息表示握手成功；如果校验失败，后台返回 code 为非0值的消息并断开连接。

```
{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN"}
```

### 识别阶段

握手成功之后，进入识别阶段，客户端上传语音数据并接收识别结果消息。

#### 上传数据

在识别过程中，客户端持续上传 binary message 到后台，内容为音频流二进制数据。建议每 40ms 发送 40ms 时长（即1:1实时率）的数据包，对应 PCM 小为：8k 采样率640字节，16k 采样率1280字节。音频发送速率过快超过1:1实时率或者音频数据包之间发送间隔超过6秒，可能导致引擎出错，后台将返回错误并主动断开连接。

音频流上传完成之后，客户端需发送以下内容的 text message，通知后台结束识别。

```
{"type": "end"}
```

### 接收消息

客户端上传数据的过程中，需要同步接收后台返回的实时识别结果，结果示例：

```
{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN","message_id":"RnKu9FODFHK5FPpsrN_11_0",
,"result":{"slice_type":0,"index":0,"start_time":0,"end_time":1240,"voice_text_str":"实时",
"word_size":0,"word_list":[]}}
```

```
{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN","message_id":"RnKu9FODFHK5FPpsrN_33_0",
,"result":{"slice_type":2,"index":0,"start_time":0,"end_time":2840,"voice_text_str":"实时语音识别",
"word_size":0,"word_list":[]}}
```

后台识别完所有上传的语音数据之后，最终返回 final 值为1的消息并断开连接。

```
{"code":0,"message":"success","voice_id":"CzhjngBkv8lk5pRUxhpX","message_id":"CzhjngBkv8lk5pRUxhpX_241",
"final":1}
```

识别过程中如果出现错误，后台返回 code 为非0值的消息并断开连接。

```
{"code":4008,"message":"后台识别服务器音频分片等待超时",
"voice_id":"CzhjngBkv8lk5pRUxhpX","message_id":"CzhjngBkv8lk5pRUxhpX_241"}
```

## 开发者资源

### SDK

- [Tencent Cloud Speech SDK for Go](#)
- [Tencent Cloud Speech SDK for Java](#)
- [Tencent Cloud Speech SDK for C++](#)
- [Tencent Cloud Speech SDK for Python](#)
- [Tencent Cloud Speech SDK for JS](#)
- [Tencent Cloud Speech SDK for C#](#)

## SDK 调用示例

- [Golang 示例](#)
- [Java 示例](#)
- [C++ 示例](#)
- [Python 示例](#)
- [JS 示例](#)
- [C# 示例](#)

## 错误码

| 数值   | 说明                                                                         |
|------|----------------------------------------------------------------------------|
| 4000 | 音频数据发送过多，请1秒内最多发送3秒音频数据。<br><b>注意：</b> 实时识别的效果是“边说边出文字”，1秒内发送的音频数据总时长应为1秒。 |
| 4001 | 参数不合法，具体详情参考 message。                                                      |
| 4002 | 鉴权失败。                                                                      |
| 4003 | AppID 服务未开通，请在控制台开通服务。                                                     |
| 4004 | 资源包耗尽，请开通后付费或者购买资源包。                                                       |
| 4005 | 账户欠费停止服务，请及时充值。                                                            |
| 4006 | 账号当前调用并发超限。                                                                |
| 4007 | 音频解码失败，请检查上传音频数据格式与调用参数一致。                                                 |
| 4008 | 客户端超过15秒未发送音频数据。                                                           |
| 4009 | 客户端连接断开。                                                                   |
| 4010 | 客户端上传未知文本消息。                                                               |
| 5000 | 因机器负载过高、网络抖动等导致失败，请重新发起新识别。<br><b>注意：</b> 该问题通常为偶发，少量出现可忽略，发起新识别即可。        |
| 5001 | 因机器负载过高、网络抖动等导致失败，请重新发起新识别。<br><b>注意：</b> 该问题通常为偶发，少量出现可忽略，发起新识别即可。        |
| 5002 | 因机器负载过高、网络抖动等导致失败，请重新发起新识别。<br><b>注意：</b> 该问题通常为偶发，少量出现可忽略，发起新识别即可。        |
| 6001 | 境外调用请前往 <a href="#">腾讯云国际站</a> 开通服务。国内站用户请检查是否使用境外代理，如果使用请关闭。              |



# 录音文件识别极速版相关接口

## 录音文件识别极速版

最近更新时间：2025-07-09 18:19:29

 **注意：**  
此接口为录音文件识别极速版接口，在参数风格、错误码等方面有区别于云 API 接口，请知悉。

### 1. 接口描述

本接口支持使用者通过 HTTPS POST 方式上传一段音频并在极短时间内同步返回识别结果（通常30分钟音频可在10秒内完成识别），可满足音视频字幕、准实时质检等场景下对语音文件识别时效性的要求。

在使用该接口前，需要在语音识别控制台开通服务，并进入 [API 密钥管理页面](#) 新建密钥，生成 AppID、SecretID 和 SecretKey，用于 API 调用时生成签名，签名将用来进行接口鉴权。

### 2. 接口要求

集成录音识别极速版 API 时，需按照以下要求。

| 内容   | 说明                                                                                                                                                                                                   |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 支持语言 | 支持中文普通话、粤语、英语、韩语、日语、泰语、印度尼西亚语、越南语、马来语、菲律宾语、葡萄牙语、土耳其语、阿拉伯语、西班牙语、印地语、法语、德语、上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话。可通过接口参数 engine_type 设置对应语言类型。 |
| 音频格式 | wav、pcm、ogg-opus、speex、silk、mp3、m4a、aac、amr                                                                                                                                                          |
| 使用限制 | 支持 100MB 以内且时长不超过2小时的音频文件，如音频时长大于2小时，建议使用 <a href="#">录音文件识别</a> 。                                                                                                                                   |
| 请求协议 | HTTPS                                                                                                                                                                                                |
| 请求地址 | <code>https://asr.cloud.tencent.com/asr/flash/v1/&lt;appid&gt;?{请求参数}</code>                                                                                                                         |
| 接口鉴权 | 签名鉴权机制，详情请参见 <a href="#">签名生成</a> 。                                                                                                                                                                  |
| 响应格式 | 统一采用 JSON 格式。                                                                                                                                                                                        |
| 并发限制 | 普通版本单个账号（AppId 维度）免费并发数为20，大模型版本免费并发数为5，如您有提高并发限制的需求，请 <a href="#">前往购买</a> 。                                                                                                                        |

### 3. 接口调用说明

本接口为 HTTPS 同步调用，包含请求 Header、请求 URL、请求 Body、响应 Response。

- 请求 Header：包含 Host、签名串等信息。
- 请求 URL：包含请求的所有参数。
- 请求 Body：包含音频原始数据。
- 响应 Response：返回识别结果，为 JSON 结构。

以下为详细说明。

#### 3.1 请求 Header 说明

包括 Host、Authorization、Content-Type、Content-Length 四个参数。

| 参数名称 | 必选 | 类型     | 描述                                  |
|------|----|--------|-------------------------------------|
| Host | 是  | String | 语音识别服务域名，固定为 asr.cloud.tencent.com。 |



|                |   |         |                                            |
|----------------|---|---------|--------------------------------------------|
| Authorization  | 是 | String  | 用户的签名字符串，用于鉴权。详情请参见 <a href="#">签名生成</a> 。 |
| Content-Type   | 是 | String  | application/octet-stream                   |
| Content-Length | 是 | Integer | 请求长度，此处对应语音数据字节数，单位：字节。                    |

### 3.2 请求参数说明

URL 中的请求参数可包含如下参数：

| 参数名称        | 必选 | 类型      | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|----|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appid       | 是  | Integer | 用户在腾讯云注册账号的 appid，可以进入 <a href="#">API 密钥管理页面</a> 获取。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| secretid    | 是  | String  | 用户在腾讯云注册账号 appid 对应的 secretid，可以进入 <a href="#">API 密钥管理页面</a> 获取。<br>示例值：*****Qq1zhZMN8dv0*****                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| engine_type | 是  | String  | <p>引擎模型类型</p> <p>识别引擎采用分级计费方案，标记为“大模型版”的引擎适用大模型计费方案，<a href="#">点击这里</a> 查看产品计费说明。</p> <p>电话场景：</p> <ul style="list-style-type: none"><li>8k_zh：中文电话通用；</li><li>8k_en：英文电话通用；</li><li><b>8k_zh_large</b>：中文电话场景专用大模型引擎【大模型版】。通过显著提升模型参数规模与语言建模能力，实现对电话音频中复杂场景（如口音干扰、背景噪声）的高精度识别，识别准确率较常规版本大幅提升。</li></ul> <p>非电话场景：</p> <ul style="list-style-type: none"><li>16k_zh：中文通用；</li><li><b>16k_zh_large</b>：普方英大模型引擎【大模型版】。当前模型同时支持中文、英文、<a href="#">多种中文方言</a> 等语言的识别，模型参数量极大，语言模型性能增强，针对噪声大、回音大、人声小、人声远等低质量音频的识别准确率极大提升，<a href="#">点击这里</a> 对比中文普通话常规版本与普方英大模型版本的识别效果；</li><li><b>16k_multi_lang</b>：多语种大模型引擎【大模型版】。当前模型同时支持英语、日语、韩语、阿拉伯语、菲律宾语、法语、印地语、印尼语、马来语、葡萄牙语、西班牙语、泰语、土耳其语、越南语、德语的识别，可实现15个语种的自动识别(句子/段落级别)；</li><li>16k_zh-PY：中英粤；</li><li>16k_yue：粤语；</li><li>16k_en：英语；</li><li>16k_ja：日语；</li><li>16k_ko：韩语；</li><li>16k_vi：越南语；</li><li>16k_ms：马来语；</li><li>16k_id：印度尼西亚语；</li><li>16k_fil：菲律宾语；</li><li>16k_th：泰语；</li><li>16k_pt：葡萄牙语；</li><li>16k_tr：土耳其语；</li><li>16k_ar：阿拉伯语；</li><li>16k_es：西班牙语；</li><li>16k_hi：印地语；</li><li>16k_fr：法语；</li><li>16k_de：德语；</li><li>16k_zh_dialect：多方言，支持23种方言（上海话、四川话、武汉话、贵阳话、昆明话、西安话、郑州话、太原话、兰州话、银川话、西宁话、南京话、合肥话、南昌话、</li></ul> |

|                     |   |         |                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|---|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |   |         | 长沙话、苏州话、杭州话、济南话、天津话、石家庄话、黑龙江话、吉林话、辽宁话)；<br>示例值：16k_zh                                                                                                                                                                                                                                                                                                                                  |
| voice_format        | 是 | String  | 音频格式。支持 wav、pcm、ogg-opus、speex、silk、mp3、m4a、aac、amr。<br>示例值：wav                                                                                                                                                                                                                                                                                                                        |
| timestamp           | 是 | Integer | 当前 UNIX 时间戳，如果与当前时间相差超过3分钟，会报签名失败错误。<br>示例值：1745932688                                                                                                                                                                                                                                                                                                                                 |
| speaker_diarization | 否 | Integer | 是否开启说话人分离（目前支持中文普通话引擎），默认为0，0：不开启，1：开启。<br>示例值：0                                                                                                                                                                                                                                                                                                                                       |
| hotword_id          | 否 | String  | 热词表 id。如不设置该参数，自动生效默认热词表；如设置了该参数，那么将生效对应的热词表。<br>示例值：da3f5f5555cf11eda6da525400aec391                                                                                                                                                                                                                                                                                                  |
| customization_id    | 否 | String  | 自学习模型 id。如设置了该参数，将生效对应的自学习模型。<br>示例值：39bc8e96504511edbd76446a2eb5fd98                                                                                                                                                                                                                                                                                                                  |
| filter_dirty        | 否 | Integer | 是否过滤脏词（目前支持中文普通话引擎），默认为0。0：不过滤脏词；1：过滤脏词；2：将脏词替换为*。<br>示例值：0                                                                                                                                                                                                                                                                                                                            |
| filter_modal        | 否 | Integer | 是否过滤语气词（目前支持中文普通话引擎），默认为0。0：不过滤语气词；1：部分过滤；2：严格过滤。<br>示例值：0                                                                                                                                                                                                                                                                                                                             |
| filter_punc         | 否 | Integer | 是否过滤标点符号（目前支持中文普通话引擎），默认为0。0：不过滤，1：过滤句末标点，2：过滤所有标点。<br>示例值：0                                                                                                                                                                                                                                                                                                                           |
| convert_num_mode    | 否 | Integer | 是否进行阿拉伯数字智能转换，默认为1。0：全部转为中文数字；1：根据场景智能转换为阿拉伯数字。<br>示例值：1                                                                                                                                                                                                                                                                                                                               |
| word_info           | 否 | Integer | 是否显示词级别时间戳，默认为0。0：不显示；1：显示词时间戳，不包含标点；2：显示词时间戳，包含标点；3：标点符号分段，包含每段时间戳，特别适用于字幕场景（包含词级时间、标点、语速值）。<br>示例值：0                                                                                                                                                                                                                                                                                 |
| first_channel_only  | 否 | Integer | 是否只识别首个声道，默认为1。0：识别所有声道；1：识别首个声道。<br><br><b>注意：</b> 如识别所有声道，则会按照 声道数*音频时长 来计费，例如时长10秒的双声道文件，如识别所有声道，会按照20秒计费。<br>示例值：1                                                                                                                                                                                                                                                                 |
| sentence_max_length | 否 | Integer | 单标点最多字数，取值范围：[6, 40]。默认为0，不开启该功能。该参数可用于字幕生成场景，控制单行字幕最大字数。仅支持 8k_zh、16k_zh 引擎。（目前仅支持 8k zh/16k zh 引擎。）<br>示例值：0                                                                                                                                                                                                                                                                         |
| hotword_list        | 否 | String  | 临时热词表：该参数用于提升识别准确率。<br><ul style="list-style-type: none"> <li>单个热词限制："热词 权重"，单个热词不超过30个字符（最多10个汉字），权重[1-11]或者100，如：“腾讯云 5”或“ASR 11”。</li> <li>临时热词表限制：多个热词用英文逗号分割，最多支持128个热词，如：“腾讯云 10, 语音识别 5, ASR 11”。</li> <li>参数 hotword_id（热词表）与 hotword_list（临时热词表）区别： <ul style="list-style-type: none"> <li>hotword_id：热词表。需要先在控制台或接口创建热词表，获得对应 hotword_id 传入参数来使用热词功能。</li> </ul> </li> </ul> |

- hotword\_list: 临时热词表。每次请求时直接传入临时热词表来使用热词功能，云端不保留临时热词表。适用于有极大量热词需求的用户。

**注意：**

- 如果同时传入了 hotword\_id 和 hotword\_list，会优先使用 hotword\_list；
- 热词权重设置为11时，当前热词将升级为超级热词，建议仅将重要且必须生效的热词设置到11，设置过多权重为11的热词将影响整体字准确率。
- 热词权重设置为100时，当前热词开启热词增强同音替换功能（仅支持 8k\_zh,16k\_zh, 8k\_zh\_large, 16k\_zh\_large），举例：热词配置“蜜制100”时，与“蜜制”同拼音（mizhi）的“秘制”的识别结果会被强制替换成“蜜制”。因此建议客户根据自己的实际情况开启该功能。建议仅将重要且必须生效的热词设置到100，设置过多权重为100的热词将影响整体字准确率。

示例值：语音助理|10,ASR 腾讯云|10

|                   |   |         |                                                                                                                                                                                                                                        |
|-------------------|---|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| input_sample_rate | 否 | Integer | <p>支持 PCM 格式的 8k 音频在与引擎采样率不匹配的情况下升采样到 16k 后识别，能有效提升识别准确率。仅支持：8000。如：传入 8000，则 PCM 音频采样率为 8k，当引擎选用16k_zh，那么该8k采样率的 PCM 音频可以在 16k_zh 引擎下正常识别。</p> <p><b>注意：</b>此参数仅适用于 PCM 格式音频，不传入值将维持默认状态，即默认调用的引擎采样率等于 PCM 音频采样率。</p> <p>示例值：8000</p> |
|-------------------|---|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 3.3 请求 Body 说明

请求 Body 中包含音频原始数据，最大不能超过100MB。

### 3.4 响应结果说明

HTTPS 请求返回的响应为 JSON 结构，具体结构说明如下：

| 参数             | 类型      | 描述                                                                                                                                                                                                                             |
|----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| code           | Integer | 0: 正常，其他，发生错误。<br>示例值：0                                                                                                                                                                                                        |
| message        | String  | code 非0时，message 中会有错误消息。<br>示例值：音频解析失败。                                                                                                                                                                                       |
| request_id     | String  | 请求唯一标识，请您记录该值，以便排查错误。<br>示例值：6810d6f958de50265f4837b0                                                                                                                                                                          |
| audio_duration | Integer | 音频时长，单位为毫秒。<br>示例值：520                                                                                                                                                                                                         |
| flash_result   | Result  | 声道识别结果列表。<br>示例值：[{"text":"野望。","channel_id":0,"sentence_list":[{"text":"野望。","start_time":0,"end_time":520,"speaker_id":0,"emotional_energy":0,"speech_speed":0,"word_list":[{"word":"野望","start_time":10,"end_time":500}]}]} |

Result 结构为：

| 参数            | 类型       | 描述                          |
|---------------|----------|-----------------------------|
| channel_id    | Integer  | 声道标识，从0开始，对应音频声道数。<br>示例值：0 |
| text          | String   | 声道音频完整识别结果。<br>示例值：野望。      |
| sentence_list | Sentence | 句子/段落级别的识别结果列表。             |

示例值: [{"text": "野望。", "start\_time": 0, "end\_time": 520, "speaker\_id": 0, "emotional\_energy": 0, "speech\_speed": 0, "word\_list": [{"word": "野望", "start\_time": 10, "end\_time": 500}]]

Sentence 结构为:

| 参数         | 类型      | 描述                                                                                        |
|------------|---------|-------------------------------------------------------------------------------------------|
| text       | String  | 句子/段落级别文本。<br>示例值: 野望。                                                                    |
| start_time | Integer | 开始时间。<br>示例值: 0                                                                           |
| end_time   | Integer | 结束时间。<br>示例值: 520                                                                         |
| speaker_id | Integer | 说话人 ID (请求中如果设置了 speaker_diarization, 可以按照 speaker_id 来区分说话人)。<br>示例值: 1                  |
| word_list  | []Word  | 词级别的识别结果列表。<br>示例值: [{"word": "我", "start_time": 380, "end_time": 680, "stable_flag": 1}] |

Word 结构为:

| 参数         | 类型      | 描述                |
|------------|---------|-------------------|
| word       | String  | 词级别文本。<br>示例值: 开心 |
| start_time | Integer | 开始时间。<br>示例值: 100 |
| end_time   | Integer | 结束时间。<br>示例值: 500 |

## 3.5 签名生成说明

1. 对所有参数按字典序进行排序, 拼接 POST 串 + 请求 URL 作为签名原文, 这里以 `appid=125922***`, `secretid=*****Qq1zhZMN8dv0*****` 为例拼接签名原文, 如下:

```
POSTasr.cloud.tencent.com/asr/flash/v1/125922***?
engine_type=16k_zh_beta&extra_punc=0&filter_punc=0&first_channel_only=1&hotword_id=584f4d0060d811
ed85da525400aec391&reinforce_hotword=0&secretid=*****Qq1zhZMN8dv0*****&speaker_diarization=0&tim
estamp=1673426168&voice_format=wav&word_info=1
```

2. 对签名原文使用 SecretKey 进行 HMAC-SHA1 加密, 之后再进行 base64 编码。例如对上一步的签名原文, `SecretKey=*****SkqpeHgqmSz*****`, 使用 HMAC-SHA1 算法进行加密并做 base64 编码处理:

```
Base64Encode(HmacSha1("POSTasr.cloud.tencent.com/asr/flash/v1/125922***?
engine_type=16k_zh_beta&extra_punc=0&filter_punc=0&first_channel_only=1&hotword_id=584f4d0060d811
ed85da525400aec391&reinforce_hotword=0&secretid=*****Qq1zhZMN8dv0*****&speaker_diarization=0&tim
estamp=1673426168&voice_format=wav&word_info=1", "*****SkqpeHgqmSz*****"))
```

得到 signature 签名值为:

```
vLBmoXJ80i4YMEuNhpIyrOo+deM=
```

### 3.6 请求示例

用户通过3.5签名生成的签名为 vLBmoXJ8Oi4YMEuNhplyrOo+deM=，语音数据路径为 /data/test.wav，请求录音识别极速版服务，这里以 curl 请求为例来说明。

```
curl --data-binary "@/data/test.wav" -H "Authorization:vLBmoXJ8Oi4YMEuNhplyrOo+deM=" -X POST
"https://asr.cloud.tencent.com/asr/flash/v1/125922****?
convert_num_mode=1&engine_type=16k_zh&filter_dirty=0&filter_modal=0&filter_punc=0&first_channel_only
=1&hotword_id=&secretid=A*****0r&speaker_diarization=0&timestamp=1609560
089&voice_format=wav&word_info=0"
```

### 3.7 返回示例

```
{
  "request_id": "6098aecab9c686fbfd35adb0",
  "code": 0,
  "message": "",
  "audio_duration": 2386,
  "flash_result": [
    {
      "text": "腾讯云智能语音欢迎您。",
      "channel_id": 0,
      "sentence_list": [
        {
          "text": "腾讯云智能语音欢迎您。",
          "start_time": 0,
          "end_time": 2386,
          "speaker_id": 0,
          "word_list": [
            {
              "word": "腾讯云",
              "start_time": 0,
              "end_time": 780,
              "stable_flag": 1
            },
            {
              "word": "智能语音",
              "start_time": 780,
              "end_time": 1590,
              "stable_flag": 1
            },
            {
              "word": "欢迎",
              "start_time": 1590,
              "end_time": 1950,
              "stable_flag": 1
            },
            {
              "word": "您",
              "start_time": 1950,
              "end_time": 2250,
              "stable_flag": 1
            }
          ]
        }
      ]
    }
  ]
}
```

```
}  
]  
}
```

## 4. 开发者资源

### SDK

- [Tencent Cloud Speech SDK for Python](#)
- [Tencent Cloud Speech SDK for Go](#)
- [Tencent Cloud Speech SDK for Java](#)

### SDK 调用示例

- [Python 示例](#)
- [Golang 示例](#)
- [Java 示例](#)

## 5. 错误码说明

| 数值   | 说明                                                                  |
|------|---------------------------------------------------------------------|
| 4001 | 参数不合法，具体详情参考 message。                                               |
| 4002 | 鉴权失败。                                                               |
| 4003 | AppID 服务未开通，请在控制台开通服务。                                              |
| 4004 | 资源包耗尽，请开通后付费或者购买资源包。                                                |
| 4005 | 账户欠费停止服务，请及时充值。                                                     |
| 4006 | 账号当前调用并发超限。                                                         |
| 4007 | 音频解码失败，请检查上传音频数据格式与调用参数一致。                                          |
| 4008 | 客户端数据上传超时。                                                          |
| 4009 | 客户端连接断开。                                                            |
| 4010 | 客户端上传未知文本消息。                                                        |
| 4011 | 音频数据太大。                                                             |
| 4012 | 音频数据为空。                                                             |
| 5001 | 因机器负载过高、网络抖动等导致失败，请重新发起新识别。<br><b>注意：</b> 该问题通常为偶发，少量出现可忽略，发起新识别即可。 |
| 5002 | 音频识别失败，偶现可以忽略，重复出现请提交工单。                                            |
| 5003 | 音频识别超时，偶现可以忽略，重复出现请提交工单。                                            |

# 虚拟号真人判定相关接口

## 虚拟号真人判定（websocket）

最近更新时间：2024-11-07 11:42:34

 **注意：**

此接口为虚拟号真人判定接口，在参数风格、错误码等方面有区别于云 API 接口，请知悉。

### 接口描述

本接口服务采用 websocket 协议，对实时音频流进行识别，能够过滤虚拟号平台转接语音，并在毫秒级快速识别真人用户接听状态并返回结果，帮助客户判断对话机器人的启动时机。

在使用该接口前，需要 [开通语音识别业务](#)，并进入 [API 密钥获取](#) 新建密钥，生成 AppID、SecretID 和 SecretKey，用于 API 调用时生成签名，签名将用来进行接口鉴权。

该接口属于ASR+系列产品，为独立接口，请前往 [计费概述](#) 了解此接口计费规则。

### 接口要求

集成虚拟号真人判定 API 时，需按照以下要求。

| 内容   | 说明                                                                                                             |
|------|----------------------------------------------------------------------------------------------------------------|
| 适用地区 | 仅支持中国大陆地区                                                                                                      |
| 支持行业 | 智能外呼等电话通信行业                                                                                                    |
| 音频属性 | 采样率：8000Hz采样精度：16bits声道：单声道（mono）                                                                              |
| 音频格式 | pcm、wav、opus、speex、silk、mp3、m4a、aac                                                                            |
| 请求协议 | wss 协议                                                                                                         |
| 请求地址 | wss://asr.cloud.tencent.com/asr/virtual_number/v1/<appid>?{请求参数}                                               |
| 接口鉴权 | 签名鉴权机制，详见 <a href="#">签名生成</a>                                                                                 |
| 响应格式 | 统一采用 JSON 格式                                                                                                   |
| 数据发送 | 建议每40ms 发送40ms 时长（即1:1实时率）的数据包，对应 pcm 大小为：8k 采样率640字节音频发送速率过快超过1:1实时率或者音频数据包之间发送间隔超过6秒，可能导致引擎出错，后台将返回错误并主动断开连接 |
| 并发限制 | 默认单账号限制并发数为200路                                                                                                |

### 接口调用流程

接口调用流程分为两个阶段：握手阶段和识别阶段。两阶段后台均返回 text message，内容为 json 序列化字符串，以下是格式说明：

| 字段名        | 类型      | 描述                                                  |
|------------|---------|-----------------------------------------------------|
| code       | Integer | 状态码，0代表正常，非0值表示发生错误                                 |
| message    | String  | 错误说明，发生错误时显示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 |
| voice_id   | String  | 音频流唯一 id，由客户端在握手阶段生成并赋值在调用参数中                       |
| message_id | String  | 本 message 唯一 id                                     |
| result     | Integer | 接听识别结果。 0:真人用户尚未接听 1:真人用户已接听                        |

|       |         |                    |
|-------|---------|--------------------|
| final | Integer | 该字段返回1时表示音频流全部识别结束 |
|-------|---------|--------------------|

## 握手阶段

### 请求格式

握手阶段，客户端主动发起 websocket 连接请求，请求 URL 格式为：

```
wss://asr.cloud.tencent.com/asr/virtual_number/v1/<appid>?{请求参数}
```

其中 <appid> 需替换为腾讯云注册账号的 AppID，可通过 [API 密钥管理平台](#) 获取，{请求参数}格式为：

```
key1=value2&key2=value2...(key 和 value 都需要进行 urlencode)
```

### 参数说明

| 参数名称         | 必填 | 类型      | 描述                                                                                                             |
|--------------|----|---------|----------------------------------------------------------------------------------------------------------------|
| secretid     | 是  | String  | 腾讯云注册账号的密钥 SecretId，可通过 <a href="#">API 密钥管理平台</a> 获取                                                          |
| timestamp    | 是  | Integer | 当前 UNIX 时间戳，单位为秒。如果与当前时间相差过大，会引起签名过期错误                                                                         |
| expired      | 是  | Integer | 签名的有效期截止时间 UNIX 时间戳，单位为秒。expired 必须大于 timestamp 且 expired - timestamp 小于90天                                    |
| nonce        | 是  | Integer | 随机正整数。用户需自行生成，最长10位                                                                                            |
| voice_id     | 是  | String  | 音频流识别全局唯一标识，一个 websocket 连接对应一个，用户自己生成（推荐使用 uuid），最长128位。                                                      |
| voice_format | 否  | Integer | 语音编码方式，可选，默认值为4。1: pcm; 4: speex(sp); 6: silk; 8: mp3; 10: opus; 12: wav; 14: m4a（每个分片须是一个完整的 m4a 音频）; 16: aac |
| signature    | 是  | String  | 接口签名参数                                                                                                         |
| wait_time    | 否  | Integer | 接通等待时长，默认为30秒，最长60秒，识别超过等待时长返回未接通。                                                                             |

## signature 签名生成

1. 对除 signature 之外的所有参数按字典序进行排序，拼接请求 URL 作为签名原文，这里以 Appid=125922\*\*，SecretId=\*\*\*\*\*Qq1zhZMN8dv0\*\*\*\*\* 为例拼接签名原文，则拼接的签名原文为：

```
asr.cloud.tencent.com/asr/virtual_number/v1/125922***?
expired=1673494772&needvad=1&nonce=1673408372&secretid=*****Qq1zhZMN8dv0*****&timestamp=16734083
72&voice_format=1&voice_id=c64385ee-3e5c-4fc5-bbfd-7c71adb35b0
```

2. 对签名原文使用 SecretKey 进行 HmacSha1 加密，之后再进行 base64 编码。例如对上一步的签名原文，SecretKey=SkqpeHgqmSz，使用 HmacSha1 算法进行加密并做 base64 编码处理：

```
Base64Encode(HmacSha1("asr.cloud.tencent.com/asr/virtual_number/v1/125922***?
expired=1673494772&needvad=1&nonce=1673408372&secretid=*****&timestamp=1673408372&voice_format=1&voice_id=c64385ee-3e5c-4fc5-bbfd-7c71adb35b0",
"*****SkqpeHgqmSz*****"))
```

得到 signature 签名值为：



```
G8jDQBRg1JfeBi/YnTjyjekxfDA=
```

3. 将 signature 值进行 urlencode（必须进行 URL 编码，否则将导致鉴权失败偶现）后拼接得到最终请求 URL 为：

```
wss://asr.cloud.tencent.com/asr/virtual_number/v1/125922****?  
expired=1592380492&filter_dirty=1&filter_modal=1&filter_punc=1&needvad=1&nonce=1592294092123&secre  
tid=*****&timestamp=1592294092&voice_format=1&voice_id=RnKu9FODF  
HK5FPpsrN&signature=G8jDQBRg1JfeBi%2FYnTjyjekxfDA%3D
```

## Opus 音频流封装说明

压缩 FrameSize 固定640，即一次压缩640 short，否则解压会失败。传到服务端可以是多帧的拼接组合，每一帧需满足下面格式。  
每一帧压缩数据封装如下：

| OpusHead（4字节） | 帧数据长度（2字节） | Opus 一帧压缩数据                |
|---------------|------------|----------------------------|
| opus          | 长度 len     | 对应 len 长的 opus decode data |

## 请求响应

客户端发起连接请求后，后台建立连接并进行签名校验，校验成功则返回 code 值为0的确认消息表示握手成功；如果校验失败，后台返回 code 为非0值的消息并断开连接。

```
{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN"}
```

## 识别阶段

握手成功之后，进入识别阶段，客户端上传语音数据并接收识别结果消息。

## 上传数据

在识别过程中，客户端持续上传 binary message 到后台，内容为音频流二进制数据。建议每40ms 发送40ms 时长（即1:1实时率）的数据包，对应 pcm 大小为：8k 采样率640字节。音频发送速率过快超过1:1实时率或者音频数据包之间发送间隔超过6秒，可能导致引擎出错，后台将返回错误并主动断开连接。

音频流上传完成之后，客户端需发送以下内容的 text message，通知后台结束识别。

```
{"type": "end"}
```

## 接收消息

客户端上传数据的过程中，需要同步接收后台返回的实时识别结果，如果返回识别结果，则表示已经识别完成，客户端无需再继续给后台发送消息，结果示例：

```
{"code":0,"message":"success","voice_id":"CzhjngBkv8lk5pRUxhpX","message_id":"CzhjngBkv8lk5pRUxhpX_2  
41","final": "1", "result": 1}
```

识别过程中如果出现错误，后台返回 code 为非0值的消息并断开连接。

```
{"code":4008,"message":"后台识别服务器音频分片等待超  
时","voice_id":"CzhjngBkv8lk5pRUxhpX","message_id":"CzhjngBkv8lk5pRUxhpX_241"}
```

## 错误码

| 数值   | 说明                        |
|------|---------------------------|
| 4001 | 参数不合法，具体详情参考 message      |
| 4002 | 鉴权失败                      |
| 4003 | AppID 服务未开通，请在控制台开通服务     |
| 4004 | 无可使用的免费额度                 |
| 4005 | 账户欠费停止服务，请及时充值            |
| 4006 | 账号当前调用并发超限                |
| 4007 | 音频解码失败，请检查上传音频数据格式与调用参数一致 |
| 4008 | 客户端数据上传超时                 |
| 4009 | 客户端连接断开                   |
| 4010 | 客户端上传未知文本消息               |
| 5000 | 后台错误，请重试                  |
| 5001 | 后台识别服务器识别失败，请重试           |
| 5002 | 后台识别服务器识别失败，请重试           |

## 开发者资源

### SDK

- [GitHub – TencentCloud/tencentcloud-speech-sdk-go](#)
- [GitHub – TencentCloud/tencentcloud-speech-sdk-java](#)

# 数据结构

最近更新时间：2025-05-09 01:09:55

## AsyncRecognitionTaskInfo

[音频流异步识别](#)任务信息

被如下接口引用：DescribeAsyncRecognitionTasks。

| 名称     | 类型      | 描述                                            |
|--------|---------|-----------------------------------------------|
| TaskId | Integer | 任务ID<br>示例值：12**135                           |
| Url    | String  | 音频流Url<br>示例值：https://www.audio.com/audio.wav |

## AsyncRecognitionTasks

[音频流异步识别任务列表](#)

被如下接口引用：DescribeAsyncRecognitionTasks。

| 名称    | 类型                                                | 描述                                |
|-------|---------------------------------------------------|-----------------------------------|
| Tasks | Array of <a href="#">AsyncRecognitionTaskInfo</a> | 任务列表<br>注意：此字段可能返回 null，表示取不到有效值。 |

## HotWord

[热词的词和权重](#)

被如下接口引用：CreateAsrVocab, GetAsrVocab, GetAsrVocabList, UpdateAsrVocab。

| 名称     | 类型      | 必选 | 描述                                        |
|--------|---------|----|-------------------------------------------|
| Word   | String  | 是  | 热词<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：秘制 |
| Weight | Integer | 是  | 权重<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：10 |

## KeywordLib

关键词表信息

被如下接口引用：GetAsrKeyWordLibList。

| 名称           | 类型     | 必选 | 描述                                                                      |
|--------------|--------|----|-------------------------------------------------------------------------|
| KeyWordLibId | String | 否  | 关键词表ID<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：aa6f402f263f**6fc81fbecfd0sd |
| Name         | String | 否  | 关键词表名称<br>注意：此字段可能返回 null，表示取不到有效值。                                     |

| 名称          | 类型                | 必选 | 描述                                                           |
|-------------|-------------------|----|--------------------------------------------------------------|
|             |                   |    | 示例值：测试词表                                                     |
| KeyWordList | Array of String   | 否  | 关键词列表<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：["a","b"]          |
| CreateTime  | Timestamp ISO8601 | 否  | 创建时间<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：2024-01-01T12:00:00 |
| UpdateTime  | Timestamp ISO8601 | 否  | 更新时间<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：2024-01-01T12:00:00 |

## KeywordLibIdData

关键词ID

被如下接口引用：CreateAsrKeywordLib, UpdateAsrKeywordLib。

| 名称           | 类型     | 必选 | 描述                                                          |
|--------------|--------|----|-------------------------------------------------------------|
| KeywordLibId | String | 否  | 关键词ID<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：dsadsdadas**adada |

## KeywordLibListData

查询列表返回数据

被如下接口引用：GetAsrKeywordLibList。

| 名称             | 类型                                  | 描述                                            |
|----------------|-------------------------------------|-----------------------------------------------|
| KeywordLibList | Array of <a href="#">KeywordLib</a> | 关键词表列表<br>注意：此字段可能返回 null，表示取不到有效值。           |
| TotalCount     | Integer                             | 关键词列表总数<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：1 |

## KeywordResult

关键字识别结果

被如下接口引用：DescribeTaskStatus。

| 名称             | 类型     | 描述                                                  |
|----------------|--------|-----------------------------------------------------|
| KeywordLibID   | String | 关键词库ID<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：abcdedfh |
| KeywordLibName | String | 关键词库名称<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：测试词库     |

| 名称       | 类型              | 描述                                                             |
|----------|-----------------|----------------------------------------------------------------|
| KeyWords | Array of String | 匹配到的关键词<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：["冰淇淋","月亮","六便士"] |

## Model

### 自学习模型信息

被如下接口引用：GetCustomizationList, GetModelInfo。

| 名称          | 类型      | 描述                                                                              |
|-------------|---------|---------------------------------------------------------------------------------|
| ModelName   | String  | 模型名称<br>示例值：通用模型                                                                |
| DictName    | String  | 模型文件名称<br>示例值：model.txt                                                         |
| ModelId     | String  | 模型Id<br>示例值：dassxas*daadsda                                                     |
| ModelType   | String  | 模型类型，“8k”或者“16k”<br>示例值：8k                                                      |
| ServiceType | String  | 服务类型<br>示例值：0                                                                   |
| ModelState  | Integer | 模型状态：<br>-2：模型训练失败；<br>-1：已下线；<br>0：训练中；<br>1：已上线；<br>3：上线中；<br>4：下线中；<br>示例值：0 |
| AtUpdated   | String  | 最后更新时间<br>示例值：2024-01-01T12:00:00                                               |

## SentenceDetail

单句的详细识别结果，包含单个词的时间偏移，一般用于生成字幕的场景。

被如下接口引用：DescribeTaskStatus。

| 名称            | 类型     | 描述                                                                 |
|---------------|--------|--------------------------------------------------------------------|
| FinalSentence | String | 单句最终识别结果<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：你好，今天很开心              |
| SliceSentence | String | 单句中间识别结果，使用空格拆分为多个词<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：你好 今天 很 开心 |
| WrittenText   | String | 口语转书面语结果，开启改功能才有值<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：你好，今天很开心。    |

| 名称              | 类型                                     | 描述                                                                                                                                                                               |
|-----------------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| StartMs         | Integer                                | 单句开始时间（毫秒）<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：0                                                                                                                                 |
| EndMs           | Integer                                | 单句结束时间（毫秒）<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：2000                                                                                                                              |
| WordsNum        | Integer                                | 单句中词个数<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：4                                                                                                                                     |
| Words           | Array of <a href="#">SentenceWords</a> | 单句中词详情<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：[{"OffsetStartMs": 120}]                                                                                                              |
| SpeechSpeed     | Float                                  | 单句语速，单位：字数/秒<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：5.9                                                                                                                             |
| SpeakerId       | Integer                                | 声道或说话人 Id（请求中如果设置了 speaker_diarization 或者 ChannelNum 为双声道，可区分说话人或声道）<br>单声道话者分离时不同的值代表不同的说话人；8k 双声道话者分离时 speakerId 的值为 0 代表左声道，值为 1 代表右声道。<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：0 |
| EmotionalEnergy | Float                                  | 情绪能量值，取值为音量分贝值/10。取值范围：[1,10]。值越高情绪越强烈。<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：1                                                                                                    |
| SilenceTime     | Integer                                | 本句与上一句之间的静音时长<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：0                                                                                                                              |
| EmotionType     | Array of String                        | 情绪类型（可能为空，有 2 种情况 1、没有对应资源包；2、情绪跟语音效果相关，如果情绪不够强烈时可能无法识别）<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：happy                                                                               |
| KeyWordResults  | Array of <a href="#">KeyWordResult</a> | 关键词识别结果列表<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：[{ "KeyWordLibID": "ec9g31d", "KeyWordLibName": "测试词库", "KeyWords": ["冰淇淋", "月亮", "六便士"]} ]                                        |

## SentenceWord

[一句话识别](#) 返回的词时间戳

被如下接口引用：SentenceRecognition。

| 名称        | 类型      | 描述                  |
|-----------|---------|---------------------|
| Word      | String  | 词结果<br>示例值：腾讯云      |
| StartTime | Integer | 词在音频中的开始时间<br>示例值：0 |

| 名称      | 类型      | 描述                     |
|---------|---------|------------------------|
| EndTime | Integer | 词在音频中的结束时间<br>示例值：1024 |

## SentenceWords

识别结果中词文本，以及对应时间偏移

被如下接口引用：DescribeTaskStatus。

| 名称            | 类型      | 描述                                                   |
|---------------|---------|------------------------------------------------------|
| Word          | String  | 词文本<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：晴天           |
| OffsetStartMs | Integer | 在句子中的开始时间偏移量<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：200 |
| OffsetEndMs   | Integer | 在句子中的结束时间偏移量<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：400 |

## Task

[录音文件识别](#)、[实时语音异步识别](#)请求的返回数据

被如下接口引用：CreateAsyncRecognitionTask, CreateRecTask。

| 名称     | 类型      | 描述                                                                                                                            |
|--------|---------|-------------------------------------------------------------------------------------------------------------------------------|
| TaskId | Integer | 任务ID，可通过此ID在轮询接口获取识别状态与结果。TaskId数据类型为uint64。<br>注意：TaskId有效期为24小时，不同日期可能出现重复TaskId，请不要依赖TaskId作为您业务系统里的唯一ID。<br>示例值：400000000 |

## TaskStatus

[获取录音识别结果的返回参数](#)

被如下接口引用：DescribeTaskStatus。

| 名称        | 类型      | 描述                                                                     |
|-----------|---------|------------------------------------------------------------------------|
| TaskId    | Integer | 任务标识。注意：TaskId数据类型为uint64。<br>示例值：1002301                              |
| Status    | Integer | 任务状态码，0：任务等待，1：任务执行中，2：任务成功，3：任务失败。<br>示例值：0                           |
| StatusStr | String  | 任务状态，waiting：任务等待，doing：任务执行中，success：任务成功，failed：任务失败。<br>示例值：waiting |
| Result    | String  | 识别结果。<br>示例值：今天天气真好                                                    |
| ErrorMsg  | String  | 失败原因说明。                                                                |

| 名称            | 类型                                      | 描述                                                                                              |
|---------------|-----------------------------------------|-------------------------------------------------------------------------------------------------|
|               |                                         | 示例值：Failed to download audio file!                                                              |
| ResultDetail  | Array of <a href="#">SentenceDetail</a> | 识别结果详情，包含每个句子中的词时间偏移，一般用于生成字幕的场景。（录音识别请求中 ResTextFormat=1 时该字段不为空）<br>注意：此字段可能返回 null，表示取不到有效值。 |
| AudioDuration | Float                                   | 音频时长(秒)。<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：1.2                                                |

## UsageByDateInfo

用户用量信息

被如下接口引用：GetUsageByDate。

| 名称       | 类型      | 必选 | 描述                                                 |
|----------|---------|----|----------------------------------------------------|
| BizName  | String  | 否  | 业务类型名称<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：1       |
| Count    | Integer | 否  | 识别次数<br>单位：次<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：1 |
| Duration | Integer | 否  | 识别时长<br>单位：秒<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：1 |

## UsageByDateInfoData

用户用量信息

被如下接口引用：GetUsageByDate。

| 名称                  | 类型                                       | 必选 | 描述                                  |
|---------------------|------------------------------------------|----|-------------------------------------|
| UsageByDateInfoList | Array of <a href="#">UsageByDateInfo</a> | 否  | 用量信息列表<br>注意：此字段可能返回 null，表示取不到有效值。 |

## VerifyTop

声纹组对比结果top数据

被如下接口引用：VoicePrintGroupVerify。

| 名称           | 类型     | 必选 | 描述                                             |
|--------------|--------|----|------------------------------------------------|
| Score        | String | 否  | 相似度打分<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：0.70 |
| VoicePrintId | String | 否  | 说话人id<br>注意：此字段可能返回 null，表示取不到有效值。             |



| 名称        | 类型     | 必选 | 描述                                           |
|-----------|--------|----|----------------------------------------------|
|           |        |    | 示例值：dsadcsds-dsads-dsadasd                   |
| SpeakerId | String | 否  | 说话人昵称<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：小明 |

## VerifyTopResult

说话人验证1:N返回结果

被如下接口引用：VoicePrintGroupVerify。

| 名称         | 类型                                 | 必选 | 描述                                             |
|------------|------------------------------------|----|------------------------------------------------|
| VerifyTops | Array of <a href="#">VerifyTop</a> | 否  | 对比打分结果，按照打分降序排列返回<br>注意：此字段可能返回 null，表示取不到有效值。 |

## Vocab

词表内容

被如下接口引用：GetAsrVocabList。

| 名称          | 类型                                | 描述                                                       |
|-------------|-----------------------------------|----------------------------------------------------------|
| Name        | String                            | 热词表名称<br>示例值：通用热词                                        |
| Description | String                            | 热词表描述<br>示例值：适用通用场景                                      |
| VocabId     | String                            | 热词表ID<br>示例值：xsddsffa****dsafgdf                         |
| WordWeights | Array of <a href="#">HotWord</a>  | 词权重列表                                                    |
| CreateTime  | <a href="#">Timestamp ISO8601</a> | 词表创建时间<br>示例值：2024-01-01T12:00:00                        |
| UpdateTime  | <a href="#">Timestamp ISO8601</a> | 词表更新时间<br>示例值：2024-01-01T12:00:00                        |
| State       | Integer                           | 热词表状态，1为默认状态即在识别时默认加载该热词表进行识别，0为初始状态<br>示例值：0            |
| TagInfos    | Array of String                   | 标签数组<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：xxxxxxx****xxxx |

## VoicePrintBaseData

[说话人基础数据](#)，包括说话人id和说话人昵称

被如下接口引用：VoicePrintCount, VoicePrintDelete, VoicePrintEnroll, VoicePrintUpdate。

| 名称           | 类型     | 描述                                                             |
|--------------|--------|----------------------------------------------------------------|
| VoicePrintId | String | 说话人id<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：xasdfgfgf****gvdfdfs |
| SpeakerNick  | String | 说话人昵称<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：腾讯云                  |

## VoicePrintCompareData

音频声纹比对结果，包含比对分数

被如下接口引用：VoicePrintCompare。

| 名称       | 类型      | 描述                                                              |
|----------|---------|-----------------------------------------------------------------|
| Score    | String  | 匹配度 取值范围(0.0 – 100.0)<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：100.0 |
| Decision | Integer | 验证结果 0: 未通过 1: 通过<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：0         |

## VoicePrintCountData

统计返回[说话人注册数量](#)

被如下接口引用：VoicePrintCount。

| 名称             | 类型                                          | 描述                                       |
|----------------|---------------------------------------------|------------------------------------------|
| Total          | Integer                                     | 总数<br>注意：此字段可能返回 null，表示取不到有效值。<br>示例值：0 |
| VoicePrintList | Array of <a href="#">VoicePrintBaseData</a> | 说话人id列表<br>注意：此字段可能返回 null，表示取不到有效值。     |

## VoicePrintVerifyData

[说话人验证数据](#)

被如下接口引用：VoicePrintVerify。

| 名称           | 类型      | 描述                                                 |
|--------------|---------|----------------------------------------------------|
| VoicePrintId | String  | 说话人id<br>注意：此字段可能返回 null，表示取不到有效值。                 |
| Score        | String  | 匹配度 取值范围(0.0 – 100.0)<br>注意：此字段可能返回 null，表示取不到有效值。 |
| Decision     | Integer | 验证结果 0: 未通过 1: 通过<br>注意：此字段可能返回 null，表示取不到有效值。     |

# 错误码

最近更新时间：2024-12-20 01:06:19

## 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 <code>Authorization</code> 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 <a href="#">控制台</a> 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 <a href="#">CAM</a> 文档对鉴权的说明。                                  |
| DryRunOperation                   | DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。                                    |
| FailedOperation                   | 操作失败。                                                                    |
| InternalError                     | 内部错误。                                                                    |
| InvalidAction                     | 接口不存在。                                                                   |
| InvalidParameter                  | 参数错误（包括参数格式、类型等错误）。                                                      |
| InvalidParameterValue             | 参数取值错误。                                                                  |

| 错误码                                               | 说明                                |
|---------------------------------------------------|-----------------------------------|
| InvalidRequest                                    | 请求 body 的 multipart 格式错误。         |
| IpInBlacklist                                     | IP 地址在黑名单中。                       |
| IpNotInWhitelist                                  | IP 地址不在白名单中。                      |
| LimitExceeded                                     | 超过配额限制。                           |
| MissingParameter                                  | 缺少参数。                             |
| NoSuchProduct                                     | 产品不存在                             |
| NoSuchVersion                                     | 接口版本不存在。                          |
| RequestLimitExceeded                              | 请求的次数超过了频率限制。                     |
| RequestLimitExceeded.GlobalRegionUinLimitExceeded | 主账号超过频率限制。                        |
| RequestLimitExceeded.IPLimitExceeded              | IP 限频。                            |
| RequestLimitExceeded.UinLimitExceeded             | 主账号限频。                            |
| RequestSizeLimitExceeded                          | 请求包超过限制大小。                        |
| ResourceInUse                                     | 资源被占用。                            |
| ResourceInsufficient                              | 资源不足。                             |
| ResourceNotFound                                  | 资源不存在。                            |
| ResourceUnavailable                               | 资源不可用。                            |
| ResponseSizeLimitExceeded                         | 返回包超过限制大小。                        |
| ServiceUnavailable                                | 当前服务暂时不可用。                        |
| UnauthorizedOperation                             | 未授权操作。                            |
| UnknownParameter                                  | 未知参数错误，用户多传未定义的参数会导致错误。           |
| UnsupportedOperation                              | 操作不支持。                            |
| UnsupportedProtocol                               | http(s) 请求协议错误，只支持 GET 和 POST 请求。 |
| UnsupportedRegion                                 | 接口不支持所传地域。                        |

## 业务错误码

| 错误码                                        | 说明             |
|--------------------------------------------|----------------|
| AuthFailure                                | CAM签名/鉴权错误。    |
| AuthFailure.CheckResourceResponseCodeError | 用户没有权限进行此查询操作。 |
| FailedOperation.CheckAuthInfoFailed        | 鉴权错误。          |
| FailedOperation.ErrorDownFile              | 下载音频文件失败。      |
| FailedOperation.ErrorRecognize             | 识别失败。          |
| FailedOperation.NoSuchTask                 | 错误的TaskId。     |

| 错误码                                        | 说明                        |
|--------------------------------------------|---------------------------|
| FailedOperation.NotExistentVoicePrintId    | 不存在的话人id                  |
| FailedOperation.ServiceIsolate             | 账号因为欠费停止服务，请在腾讯云账户充值。     |
| FailedOperation.UserHasNoAmount            | 资源包耗尽，请购买资源包或开通后付费        |
| FailedOperation.UserHasNoFreeAmount        | 资源包耗尽，请开通后付费或者购买资源包       |
| FailedOperation.UserNotRegistered          | 服务未开通，请在腾讯云官网语音识别控制台开通服务。 |
| InternalServerError.ErrorConfigure         | 初始化配置失败。                  |
| InternalServerError.ErrorCreateLog         | 创建日志失败。                   |
| InternalServerError.ErrorDownFile          | 下载音频文件失败。                 |
| InternalServerError.ErrorFailNewprequest   | 新建数组失败。                   |
| InternalServerError.ErrorFailWritetodb     | 写入数据库失败。                  |
| InternalServerError.ErrorFileCannotopen    | 文件无法打开。                   |
| InternalServerError.ErrorGetRoute          | 获取路由失败。                   |
| InternalServerError.ErrorMakeLogpath       | 创建日志路径失败。                 |
| InternalServerError.ErrorRecognize         | 识别失败。                     |
| InternalServerError.FailAccessDatabase     | 访问数据库失败。                  |
| InternalServerError.FailAccessRedis        | 访问Redis失败。                |
| InternalServerError.FailedVoicePrintDecode | 说话人音频解码失败                 |
| InternalServerError.FailedVoicePrintEnroll | 说话人注册接口失败                 |
| InternalServerError.FailedVoicePrintVerify | 说话人验证接口失败                 |
| InternalServerError.TagRequestError        | 请求标签接口出错                  |
| InternalServerError.VoicePrintAudioFailed  | 说话人音频处理失败                 |
| InternalServerError.VoicePrintDecodeFailed | 传入音频解码失败                  |
| InternalServerError.VoicePrintEnrollFailed | 说话人ID注册失败                 |
| InternalServerError.VoicePrintVerifyFailed | 说话人验证失败                   |
| InvalidParameter.ErrorContentlength        | 请求数据长度无效。                 |
| InvalidParameter.ErrorParamsMissing        | 参数不全。                     |
| InvalidParameter.ErrorParsequest           | 解析请求数据失败。                 |
| InvalidParameter.FailedVoicePrintDecode    | 音频解码失败，请检查音频格式是否正确        |
| InvalidParameter.FileEncode                | 文件编码错误。                   |
| InvalidParameter.InvalidParameter          | 参数错误。                     |
| InvalidParameter.InvalidVocabState         | 非法的词表状态。                  |
| InvalidParameter.KeyWordLibNameExist       | 关键词库名字已存在                 |

| 错误码                                              | 说明                |
|--------------------------------------------------|-------------------|
| InvalidParameter.ModelState                      | 该模型状态不允许删除。       |
| InvalidParameterValue.ErrorInvalidAppid          | AppId无效。          |
| InvalidParameterValue.ErrorInvalidClientip       | ClientIp无效。       |
| InvalidParameterValue.ErrorInvalidEngservice     | EngSerViceType无效。 |
| InvalidParameterValue.ErrorInvalidProjectid      | ProjectId无效。      |
| InvalidParameterValue.ErrorInvalidRequestid      | RequestId无效。      |
| InvalidParameterValue.ErrorInvalidSourcetype     | SourceType无效。     |
| InvalidParameterValue.ErrorInvalidSubservicetype | SubserviceType无效。 |
| InvalidParameterValue.ErrorInvalidUrl            | Url无效。            |
| InvalidParameterValue.ErrorInvalidUseraudiokey   | UsrAudioKey无效。    |
| InvalidParameterValue.ErrorInvalidVoiceFormat    | 音频编码格式不支持。        |
| InvalidParameterValue.ErrorInvalidVoicedata      | 音频数据无效。           |
| InvalidParameterValue.ErrorVoicedataTooLong      | 音频时长超过限制。         |
| InvalidParameterValue.InvalidParameterLength     | 非法的参数长度。          |
| InvalidParameterValue.InvalidVocabId             | 非法的VocabId。       |
| InvalidParameterValue.InvalidVocabState          | 非法的词表状态。          |
| InvalidParameterValue.InvalidWordWeight          | 词权重不合法。           |
| InvalidParameterValue.InvalidWordWeightStr       | 非法的WordWeightStr。 |
| InvalidParameterValue.ModelId                    | 模型不存在。            |
| InvalidParameterValue.NoHumanVoice               | 音频内容没有人声或有效人声小于1秒 |
| InvalidParameterValue.ToState                    | 非法的模型状态。          |
| LimitExceeded.CustomizationFull                  | 自学习模型创建个数已到限制。    |
| LimitExceeded.OnlineFull                         | 上线模型个数已到限制。       |
| LimitExceeded.VocabFull                          | 热词表数量已到账号限制。      |
| LimitExceeded.VoicePrintFull                     | 说话人ID创建数量达到上限     |