

实时音视频

AI 智能语音服务



腾讯云

【 版权声明 】

©2013–2026 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內容。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

AI 智能语音服务

功能介绍

开通服务

配合 TRTC 使用

后台接入 AI 转录/翻译

SDK 接入 AI 转录/翻译

视频通话接入 AI 转录/翻译

多人会议接入 AI 转录/翻译

AI 转录/翻译迁移说明

AI 转录/翻译回调说明

直接使用

实时语音识别

一句话识别

录音文件识别

语音合成

AI 智能语音服务

功能介绍

最近更新时间：2026-05-09 17:58:31

方案概述

基于腾讯云 RTC 全球传输网络，AI 智能语音服务提供语音转文本（ASR）、实时翻译和文字转语音（TTS）能力。目前支持两种使用模式：**直接调用 ASR/TTS 能力（不依赖 TRTC 房间），或配合 TRTC 房间使用。**通过语音识别引擎（ASR）可将语音转换为文本内容，快速构建直播实时字幕、会议内容实时记录、视频通话与语聊内容转写、录音文件离线转写等多种场景下的语音识别应用。语音识别转文本后支持开启多语言翻译能力，实现跨语言内容翻译转写。

基本架构

● 配合 TRTC 使用：

- **实时语音转文本：**如下方架构图所示，音频流经过 TRTC 全球网络节点传输到服务端进行音频处理后，由语音识别引擎（ASR）将语音转写为文本。系统支持多语种、热词权重配置、VAD 检测以及实时流式识别。对于需生成字幕或实时内容记录的场景，在此流程即可直接输出最终转写结果回调。
- **实时翻译链路（可选择开启）：**针对跨语言理解的场景，可在语音转写的基础上开启翻译功能。系统会将转写后的文本通过 LLM 翻译引擎进行翻译处理，并同时输出原文转写内容以及多语种翻译结果。



● 直接使用：

- **语音转文本：**可通过 API 直接推送音频流或上传音频文件进行识别，不依赖 TRTC，支持实时语音识别和录音文件识别两种模式。直接使用不支持开启文本翻译。

- 文字转语音 (TTS) 可通过 API 直接调用，将输入文本合成为语音输出。当前支持中文、英语、日语、韩语及粤语，同时提供声音克隆能力，可基于少量音频样本生成专属音色。

能力支持

能力	直接使用	配合 TRTC 使用
语音转文本 — 实时语音识别	✓	✓
语音转文本 — 录音文件识别	✓	-
语音转文本 — 一句话识别	✓	-
实时翻译	-	✓
文字转语音	✓	✓

可参考下文按能力分类的详细说明，根据使用场景选择适合的模型。

一、语音转文本 (ASR)

1. 实时语音识别

支持 **直接使用** 和 **配合 TRTC 使用** 两种模式，支持的语音引擎和语种见下表。

功能	版本类型	特性说明	语种与模型型号
实时语音识别	基础版语音引擎	基础通用语音识别模型。在近场、非复杂噪声的声学环境下具备良好的响应速度与识别准确率。	<ul style="list-style-type: none"> "zh": 8k 采样率中文识别模型，主要用于电话音频。
	标准版语音引擎	基于大模型引擎，语音识别性能大幅增强，在噪声回音大、人声远小等复杂音频环境的识别准确率显著提升。常见应用场景如会议、直播、语聊、游戏等实时字幕以及实时转写记录等。高度契合 RTC 实时互动相关场景。	<ul style="list-style-type: none"> "8k_zh_large": 8k 中文大模型引擎，针对电话音频支持较好。 "16k_zh_large": 16k 大模型引擎，同时支持中文、英文、多种中文方言口音等语言的识别。 "16k_zh_en": 最新 16k 中英大模型引擎，同时支持中文、英语、多种中文方言口音的识别，对中英混说场景识别更优。
	高级版语音引擎	覆盖小语种、方言的精准识别需求。	越南语、日语、韩语、印度尼西亚语、泰语、葡萄牙语、土耳其语、阿拉伯语、西班牙语、印地语、法语、马来语、菲律宾语、德语、意大利语、俄语、瑞典语、丹麦语、挪威语、中国粤语

具体的语种配置参数枚举列表，请参考 [高级版语种参数说明](#)。

如果您有更多其他语言需求，请 [联系我们](#)。

说明：

- 以上语言引擎在配合 TRTC 使用时全部支持。在直接使用时，仅支持 "zh" 和 "16k_zh_en"两种语种，如您有更多其他语言需求，请 [联系我们](#)。
- 上文表格中列出的“模型型号”参数值对应配合 TRTC 使用时转录服务端 API 接口 AsrParam 中的 Lang 字段，完整的参数说明请参阅 [服务端转录 API 说明](#)。
- 通过含 UI 的场景方案接入或者 SDK 客户端接入转录功能无需填写模型型号，仅需填写语种参数。ASR 默认中英使用标准版 16k_zh_en 模型。

2. 录音文件识别

仅支持直接使用模式，通过 REST API 上传音频文件进行离线识别。

功能	语种与模型型号
录音文件识别	<ul style="list-style-type: none"> • "zh": 8k 采样率中文识别模型，主要用于电话音频。 • "16k_zh_en": 最新 16k 中英大模型引擎，同时支持中文、英语、多种中文方言口音的识别，对中英混说场景识别更优。

3. 一句话识别

仅支持直接使用模式，通过 REST API 上传 60 秒以内的短音频，同步返回识别结果。适用于语音指令、语音搜索等短语音场景。

功能	语种与模型型号
一句话识别	<ul style="list-style-type: none"> • "zh": 8k 采样率中文识别模型，主要用于电话音频。 • "16k_zh_en": 最新 16k 中英大模型引擎，同时支持中文、英语、多种中文方言口音的识别，对中英混说场景识别更优。

二、实时翻译

仅支持配合 TRTC 房间使用，不支持单独接入。

功能	版本类型	特性说明	语种与模型型号
实时翻译	实时翻译引擎	基于最新大语言模型技术的深度语义理解能力，系统能够自主适应目标语言表达习惯，实现自然流畅翻译效果。相较于传统翻译，翻译结果的机械感显著降低，在非正式标	支持15种语言翻译：中文、英语、西班牙语、葡萄牙语、法语、德语、俄语、阿拉伯语、日语、韩语、越南语、马来

准的口语对话交流、泛娱互动等复杂场景下的效果提升明显。

语、印度尼西亚语、意大利语、泰语。

三、文字转语音 (TTS)

支持直接使用，以及配合 [AI 实时对话](#) 使用。

- **支持语言**：中文、英语、日语、韩语及粤语
- **声音克隆**：基于少量音频样本生成专属音色

计费方式

AI 智能语音费用包含：语音转文本费用、实时翻译费用（如开启）、文字转语音费用。若配合 TRTC 使用，还会产生音频通话费用。具体能力解锁所需套餐版本和费用详情请参见 [AI 智能语音服务计费说明](#)。

接入方式

TRTC AI 智能语音解决方案支持以下接入方式：

- **配合 TRTC 使用**：
 - **无 UI 接入**：支持通过 [服务端接入](#)，也可通过 TRTC SDK [客户端快速接入](#)，简化开发集成流程。
 - **场景（含 UI）方案接入**：目前已经支持 [视频通话 AI 实时字幕和翻译](#) 方案和 [多人会议 AI 实时字幕](#) 方案。助力快速上线含 UI 的场景化实时转录功能。
- **直接使用**：
 - 不依赖 TRTC 房间，通过 API 直接调用 ASR / TTS 能力，适用于已自有音频采集链路或仅需单点能力的场景：

能力	接入方式	说明
实时语音识别	WebSocket 接入	通过 WebSocket 推送音频流进行实时流式识别。
录音文件识别	HTTP POST 接入	上传音频文件，异步返回识别结果。
一句话识别	HTTP POST 接入	上传短音频（≤60s），同步返回识别结果。
文字转语音	HTTP POST 接入	输入文本，返回合成后的语音音频，支持流式和非流式。

应用场景

应用场景	使用方式与价值
在线教育	通过 AI 转录对讲师与学员发言实时转写，课上提供实时字幕辅助理解，同步实时沉淀课程笔记与关键发言，可用于实时总结，课后复习、回放、存档。开启翻译功能后，可支持多语课堂或跨国教学场景。

社交娱乐	在社交娱乐场景中，通过 AI 转录能够对在线直播、游戏语聊、互动问答等场景，对发言内容进行实时转写，为观众提供准确字幕增强理解。同时支持输出多语言翻译内容，让不同地区用户能够更自然地参与互动。
办公协作	在线会议、线上面试、商务沟通、研讨会等场景下，通过 AI 转录生成实时字幕与内容记录，会中提升理解效率，会后可用于内容纪要整理。开启翻译功能后，可同时输出多语言转写，支持跨国团队无障碍协作。
客服中心	在线客服、智能外呼、呼叫中心等场景中，通过实时转写帮助客服准确理解用户内容。通话结束后沉淀完整文本记录，用于质检、合规与服务分析。开启翻译功能后，支持客服快速处理跨语言咨询，提高响应效率。
有声内容与语音播报	通过 TTS 将文本内容转换为自然语音，适用于新闻播报、有声读物、通知朗读、无障碍播报等场景。支持多音色切换和声音克隆，满足个性化语音输出需求。

开通服务

最近更新时间：2026-05-09 17:58:31

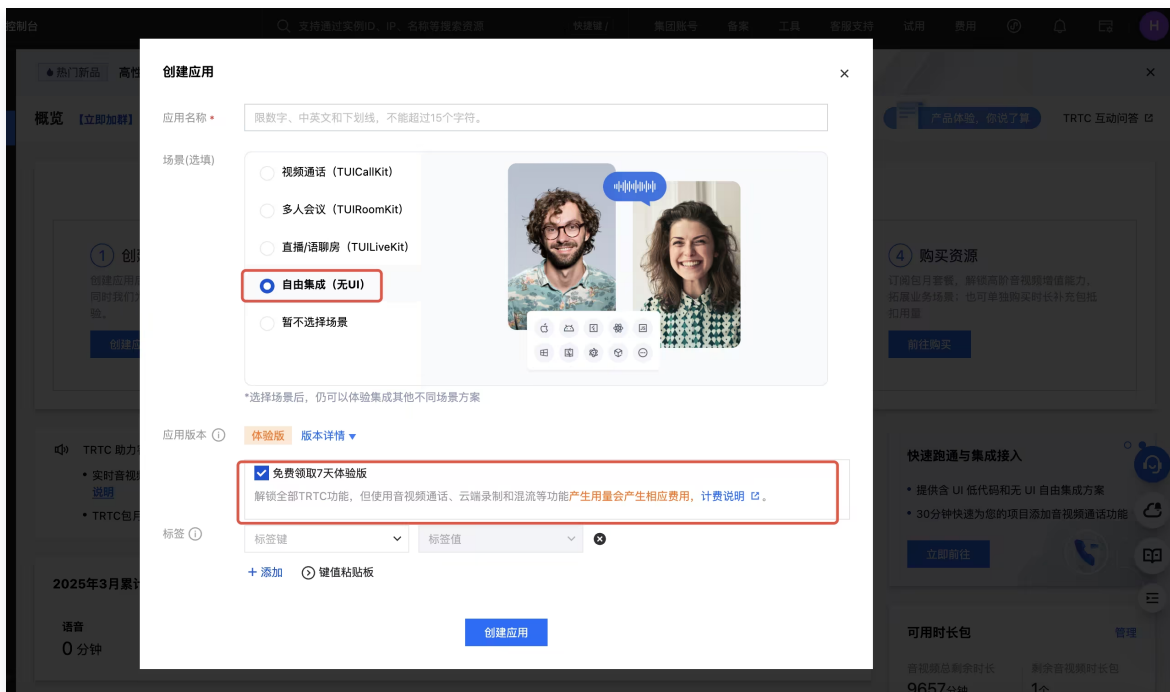
本文介绍如何 [开通体验版](#)、[开通正式版](#) 和 [续费正式版](#) 服务，您可以根据 [AI 智能语音计费说明](#) 了解所需要的服务及费用，并参考下述指引免费体验开通和购买开通正式版 AI 智能语音服务。

开通体验版

1. 登录 [实时音视频 TRTC 控制台](#)，单击**创建应用**。



2. 在弹窗中输入您的应用名称，选择**自由集成（无 UI）** 场景，勾选**免费领取7天体验版**，单击**创建应用**。



3. 进入应用管理页面，成功开通**自由集成（无 UI）** 的7天试用服务。领取完成后，可以看到体验版的基本信息，包括服务状态、版本信息和功能详情、到期时间。这里的 SDKAppID 以及 SDK 密钥（SDKSecretKey）会在后续步骤中使用到。



4. 体验版开通完成后，您可以根据 [功能介绍](#) 文档了解 AI 智能语音功能进行体验。测试过程中所产生的音视频时长等资源消耗（例如语音转文本服务、实时翻译服务、TTS 服务）仍然会产生后付费产品费用（具体可参见 [AI 智能语音计费说明](#)）。

⚠ 注意：

新账号首次可前往 [试用中心](#) 免费领取10000分钟音视频时长，可抵扣音视频通话时长，但语音转文本服务、实时翻译服务、TTS 服务仍需后付费。

5. 若您当前 SDKAppID 领取的体验版已到期，您可单击[版本详情](#)，在版本详情弹窗中单击[体验续期](#)直接使用第二次体验机会，此时体验版有效期将延长7天。

⚠ 注意：

通过[体验续期](#)延长体验版有效期，将消耗1次免费体验机会。

开通正式版

[AI 智能语音服务](#) 主要包含两方面的费用：[音频通话费用](#)、以及[语音转文字和文字转语音费用](#)；若开启翻译功能则会增加实时翻译费用。

一、配合 TRTC 使用

配合 TRTC 使用语音转文字以及实时翻译能力，需要购买 [AI 智能语音识别套餐包](#)来开启能力，**不支持直接后付费使用**。计费详情可参见 [AI 智能语音计费说明](#) 。具体开通指引如下：

1. 登录 [实时音视频购买页](#)。
2. 根据实际业务需求选择配置。接入方案选择无 UI 自定义开发，以及选择需要购买 AI 智能语音套餐的 SDKAppID，购买后无法修改。



3. 套餐类型选择 AI 智能语音套餐，并选择您需要开通的套餐版本。



4. 购买完成后，您可以回到 [实时音视频 TRTC 控制台](#)，在 [时长包管理](#) 页面查看具体的 AI 智能语音套餐的使用情况。开通完成后，您可以根据 [快速接入](#) 文档进行接入。

二、直接使用

如您选择 [直接调用](#) 语音转文字和文字转语音服务，支持通过后付费的方式直接使用，或者购买 AI 智能语音套餐包获取更多用量折扣，套餐包购买流程和上文一致。

续费正式版

您可对 AI 智能语音套餐进行续费，具体操作如下：

1. 登录 [续费管理控制台](#)，选择您想要续费的 AI 智能语音套餐，单击续费。



2. 按照指示完成续费即可。

续费

已选中1个资源

资源ID/资源名	产品描述	当前到期时间	续费时长 ⓘ	续费后到期时间 ⓘ	单价 ⓘ	续费金额
资源ID	-	2025-04-10	1个月 ▾	2025-05-10	0.00元/月	00元

自动续费 账户余额足够时，资源到期后按续费周期自动续费

优惠信息 当前暂无可用优惠券。

价格 0.00元 最终实付金额以支付订单时为准

[下一步](#) [取消](#)

配合 TRTC 使用 后台接入 AI 转录/翻译

最近更新时间：2026-05-09 17:58:31

本文将介绍通过服务端配置接入最新 2.0 版本 TRTC AI 转录的实时语音转文本和实时翻译功能。

TRTC AI 转录/翻译 2.0 对 1.0 的使用配置和回调方式进行了优化。若 1.0 需要升级到 2.0 版本，可跳转 [AI 转录/翻译 2.0 迁移说明](#) 查看相关说明文档。

说明：

- 2026 年 3 月 17 日起新用户请使用本文档接入 2.0 新版接口。
- 已接入旧版 AI 转录与翻译接口 ([StartAITranscription](#)) 的用户可跳转 [后台接入 AI 转录与翻译 \(旧\) 说明](#) 查看相关说明文档。

功能说明

通过 TRTC AI 转录与翻译功能，您可以将房间内每位用户的音频流实时转换为文本，并支持多语言翻译。

- 转录功能：由语音识别引擎（ASR）将语音转写为文本。系统支持多语种、热词权重配置、VAD 检测以及实时流式识别。
- 翻译功能：可在语音转写的基础上开启翻译功能。系统会将转写后的文本通过 LLM 翻译引擎进行翻译处理，并同时输出原文转写内容以及翻译结果，该功能为可选项。

注意：

- 单个应用默认支持 100 路任务并发，所有任务的 ASR 并发总数上限为 200 路，翻译并发总数上限为 100 路。如需更高并发数，请 [提交工单](#) 联系我们进行评估。
- 接口超时时间为 6 秒。
- 接口 QPS 限制 20，如需提高请 [提交工单](#) 进行评估。

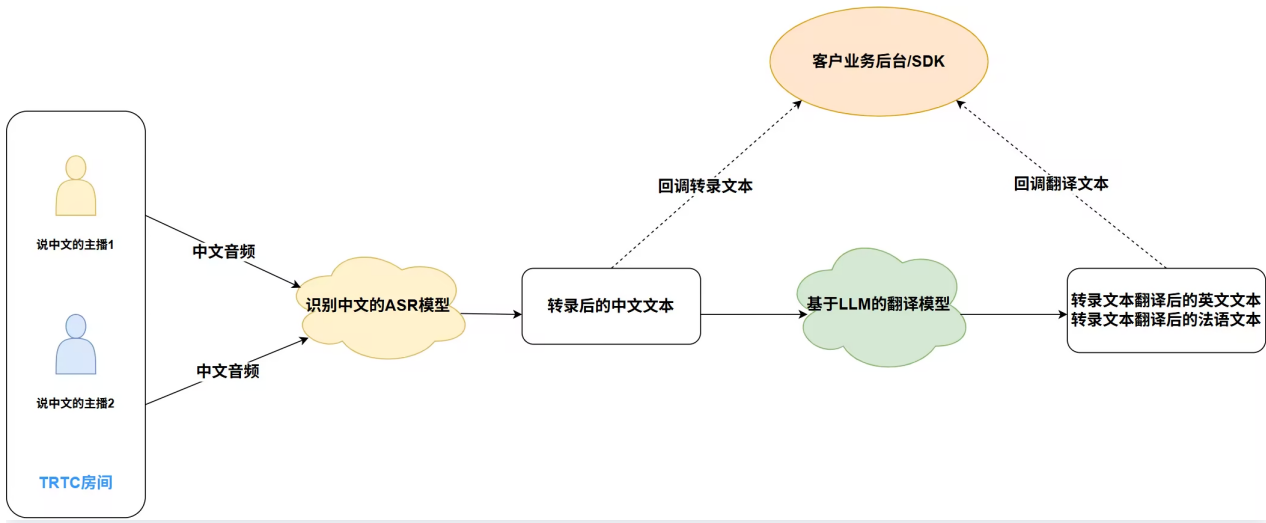
场景说明

房间内的所有用户均使用同一种语言

在如图所示的 TRTC 房间中，有多位主播使用中文进行交流。您可以发起任务订阅所有用户的中文音频流，并选择中文 ASR 模型将音频实时转录为中文文本。转录结果将通过回调方式返回。同时，您可以设置翻译目标语言，翻译模型会将转录后的中文文本翻译成指定语言，并通过回调发送翻译结果。

说明：

支持回调结果到您的业务后台 SDK，回调接入方式请参考 [回调接口](#)。

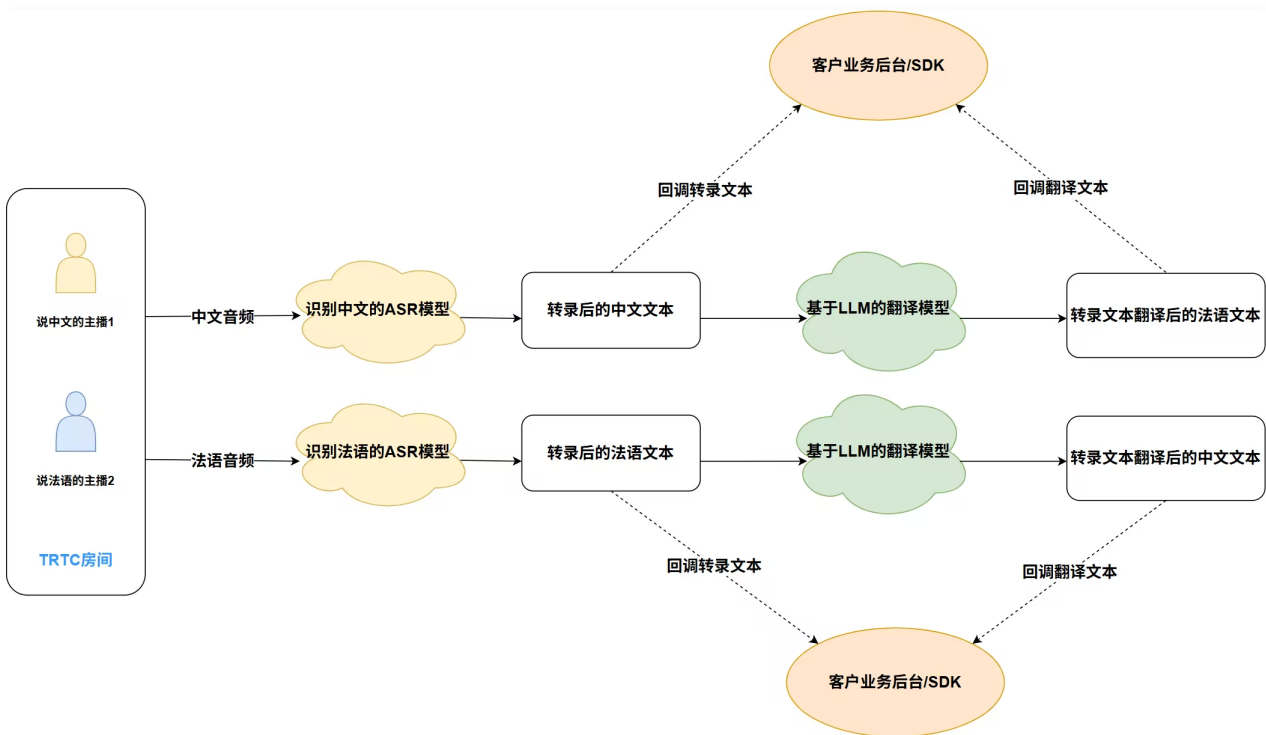


房间内的用户使用多种语言

在如图所示的 TRTC 房间中，两位主播分别使用中文和法语进行交流。由于单个任务的 ASR 模型仅支持特定语言的识别，需要发起以下两项任务：

1. 任务一：订阅主播 1 的中文音频，选择中文 ASR 模型进行实时转录，生成中文文本并通过回调返回。可设置翻译目标语言为法语，系统将自动转录文本翻译为法语，翻译结果同样通过回调返回。
2. 任务二：订阅主播 2 的法语音频，选择法语 ASR 模型进行实时转录，生成法语文本并通过回调返回。可设置翻译目标语言为中文，系统将自动转录文本翻译为中文，翻译结果同样通过回调返回。

通过以上两项任务，主播 1 和主播 2 即可实现实时的跨语言交流。



接入说明

前提条件

- 登录 [TRTC 控制台](#)，开通 TRTC 服务并 [创建应用](#)。
- 前往 [控制台 > 功能配置 > 增值功能](#) 开启 AI 智能识别的语音转文本、实时翻译功能。



说明：

- 您可以通过以下任意一种方式获取功能开启权限：领取包月套餐体验版、购买 [AI 智能识别套餐包](#)，或订阅 TRTC 包月套餐（部分版本），具体版本支持的功能详情可参见 [AI 智能识别套餐包计费说明](#)、[TRTC 包月套餐计费说明](#)，您可根据您的需要进行订阅。
- 语音转文本和实时翻译功能会根据调用量和模型引擎版本产生费用，详情请参见 [费用详情](#)。

步骤1：集成 TRTC SDK

导入 TRTC SDK 到项目中，进入 TRTC 房间并开启本地麦克风音频采集与发布，相关指引可参考：

- [iOS 无 UI 集成指引](#)
- [Android 无 UI 集成指引](#)
- [Web&H5 无 UI 集成指引](#)
- [小程序 无 UI 集成指引](#)
- [Flutter 无 UI 集成指引](#)

步骤2：RESTful API 发起任务

通过您的服务器调用 REST API（[CreateCloudTranscription](#)）可以启动转录任务，请重点关注任务 ID（TaskId）参数；这个参数是本次任务的唯一标识，您需要保存这个任务 ID 作为后续针对这个任务接口操作的输入参数。任务具体发起步骤如下：

1. 设置基本参数

您需要指定发起转录翻译任务的基本信息，例如您的应用 ID（sdkappid）、房间信息（RoomId）、房间类型（RoomIdType）。这些参数均为必填的。

2. 设置进房订阅参数

使用 AI 转录/翻译服务时，将会有机器人作为虚拟观众加入房间，订阅需要识别的音频流。您需要通过 [TranscriptionParams](#) 指定进房机器人的参数和转录订阅的用户参数。进房机器人的参数包括 ID（UserId）、签名（UserSig）、任务空闲等待时间（MaxIdleTime）。您可以通过白名单（SubscribeUids）指定需要进行转录翻译任务的主播，通过黑名单（UnSubscribeList）指定不需要进行转录翻译任务的主播，通过 `SendCustomMode` 参数指定接收转录和翻译文本的方式。

3. 配置 ASR 参数

通过 `AsrParam` 的 `Lang` 参数选择适合您场景的 ASR 模型引擎用于实现语音转文本。基于不同的场景我们提供多种 ASR 引擎，具体版本与模型类对应参数可查看下方表格。

说明：

- 例如您需要使用 16k 中英大模型引擎，在 `Lang` 参数填入 `16k_zh_en` 即可。
- `AsrParam` 的 `VadSilenceTime` 参数用于配置语音识别 Vad 的时间。
- ASR 引擎分为基础版、标准版和高级版，不同版本计费价格不同，具体计费规则参考 [费用详情](#)。

版本类型		特性说明	语种与模型型号
ASR 引擎	基础版语言引擎	基础语音识别模型。在近场、非复杂噪声的声学环境下具备良好的响应速度与识别准确率。	<ul style="list-style-type: none"> • "zh": 8k 采样率中文识别模型，主要用于电话音频。
	标准版语言引擎	基于大模型引擎，语音识别性能大幅增强，在噪声回音大、人声远小等复杂音频环境的识别准确率显著提升。常见应用场景如会议、直播、语聊、游戏等实时字幕以及实时转写记录等。高度契合 RTC 实时互动相关场景。	<ul style="list-style-type: none"> • "8k_zh_large": 8k 中文大模型引擎，针对电话音频支持较好。 • "16k_zh_large": 16k 大模型引擎，同时支持中文、英文、多种中文方言口音等语言的识别。 • "16k_zh_en": 最新 16k 中英大模型引擎，同时支持中文、英语、多种中文方言口音的识别，对中英混说场景识别更优。
	高级版语言引擎	覆盖小语种、方言的精准识别需求。	<ul style="list-style-type: none"> • "vi": 越南语 • "ja": 日语 • "ko": 韩语 • "id": 印度尼西亚语 • "th": 泰语 • "pt": 葡萄牙语

- "tr": 土耳其语
- "ar": 阿拉伯语
- "es": 西班牙语
- "hi": 印地语
- "fr": 法语
- "ms": 马来语
- "fil": 菲律宾语
- "de": 德语
- "it": 意大利语
- "ru": 俄语
- "sv": 瑞典语
- "da": 丹麦语
- "no": 挪威语
- "zh-yue": 中国粤语

如果您有更多其他语言需求，请 [联系我们](#) 进行评估。

4. 开启翻译（可选）

如果您需要对转录后的文本进行实时翻译，通过设置翻译参数 `TranslationParam` 开启相关功能。在翻译参数中指定您需要翻译的目标语言。目前支持配置以下翻译目标语言作为输出结果：

翻译目标语言代码	对应语种
"zh"	中文
"en"	英语
"es"	西班牙语
"pt"	葡萄牙语
"fr"	法语
"de"	德语
"ru"	俄语
"ar"	阿拉伯语
"ja"	日语
"ko"	韩语
"vi"	越南语

"ms"	马来语
"id"	印度尼西亚语
"it"	意大利语
"th"	泰语

⚠ 注意:

- 翻译配置不是必填项，若您只需要语音转文本功能，则无需填写翻译参数，不影响 AI 转录整体功能使用。
- 实时翻译功能输入源语种和翻译后输出语种目前支持15种语言：中文、英语、西班牙语、葡萄牙语、法语、德语、俄语、阿拉伯语、日语、韩语、越南语、马来语、印度尼西亚语、意大利语、泰语。若前置 ASR 转录语种不在上述范围，翻译功能无法开启。如需其他语种支持可以 [联系我们](#)。
- 由于受不同语境或语种差异的影响，AI 翻译的译文内容适用于辅助参考，不应作为唯一专业意见或结论。

步骤3：接收转录翻译的结果回调

方式一：通过服务端回调接收

语音转文本服务同时提供了服务端事件回调，便于您的服务接收实时对话的消息，查看 [AI 转录与翻译 2.0 回调事件](#)。

方式二：通过客户端 SDK 接收回调

您可以参考以下代码，从 TranscriberStore 的响应式数据中监听回调并刷新 UI。

Android

请在您的 build.gradle 中添加 依赖，然后执行 Gradle Sync。

```
implementation 'io.trtc.uikit:atomicx-core:4.0.0.110'  
implementation  
"com.tencent.liteav:LiteAVSDK_Professional:13.1.0.19861"  
implementation "com.tencent.imsdk:imsdk-plus:8.7.7201"
```

然后按以下代码监听消息列表的回调，更新到相应 UI 上，messages 是 TranscriberMessage 的列表。

Kotlin

```
// 假如以 RecyclerView 显示消息列表，则通过 submitList 发送消息列表即可。  
AITranscriberStore.shared.transcriberState.realtimeMessageList.collect  
{ messages ->  
    adapter.submitList(messages.toList())
```

```
}
```

TranscriberMessage 参数说明

参数名	类型	描述
<code>segmentId</code>	<code>String</code>	用户的每句话都对应一个唯一的 <code>segmentId</code> 。
<code>speakerUserId</code>	<code>String</code>	说话用户的 ID。
<code>speakerUserName</code>	<code>String</code>	说话用户的昵称。
<code>sourceText</code>	<code>String</code>	用户的语音转文本。
<code>translationTexts</code>	<code>Map<TranslationLanguage, String></code>	用户语音文本对应的翻译文本，可翻译成多种语言。
<code>timestamp</code>	<code>Long</code>	当前句子的时间戳。
<code>isCompleted</code>	<code>Boolean</code>	当前句子是否已经完成。

iOS

请在您的 Podfile 文件中添加 `pod 'AtomicXCore'` 依赖，然后执行 `pod install`。

```
target 'xxxx' do
  pod 'AtomicXCore'
end
```

然后按以下代码监听消息列表的回调，更新到相应 UI 上，`messages` 是 `TranscriberMessage` 的列表。

Swift

```
// 假如以 UITableView 显示消息列表，更新消息后，reloadData 即可。
AITranscriberStore.shared.state
    .subscribe(StatePublisherSelector(keyPath: \.realtimeMessageList))
    .receive(on: RunLoop.main)
```

```
.sink { [weak self] in self?.updateMessages($0) }
.store(in: &cancellables)
```

TranscriberMessage 参数说明

参数名	类型	描述
segmentId	String	用户的每句话都对应一个唯一的 segmentId。
speakerUserId	String	说话用户的 ID。
speakerUserName	String	说话用户的昵称。
sourceText	String	用户的语音转文本。
translations	[TranslationLanguage: String]	用户语音文本对应的翻译文本，可翻译成多种语言。
timestamp	Int64	当前句子的时间戳。
isCompleted	Bool	当前句子是否已经完成。

实践教程

为了保障 AI 转录和翻译的高可用性，在集成 RESTful API 的同时请关注以下几点。

- 调用 CreateCloudTranscription 请求后，请关注 HTTP response，如果请求失败，那么需要根据具体的状态码采取相应的重试策略。错误码是由“一级错误码”和“二级错误码”组合而成，例如：InvalidParameter.SdkAppId。具体情况说明可参考下表：

返回错误码	问题说明	解决措施
InvalidParameter.xxxxx	输入的参数有误。	请根据具体提示检查参数填写。
InternalServerError.xxxxx	遇到服务端错误。	可以使用相同的参数重试多次，直到返回正常，拿到 taskId 为止。 建议使用退避重试策略，如第一次3s 重试，第二次6s 重试，第三次12s 重试，以此类推。

FailedOperation.RestrictConcurrenc	并发转录任务数超过了后台预留的资源（默认是 100路）。	联系您的商务经理或 提交工单 评估提高限频。
------------------------------------	------------------------------	--

- 调用 CreateCloudTranscription 接口时，指定的 UserId/UserSig 是转录作为单独的机器人用户加入房间的 ID，请不要和 TRTC 房间内的其他用户重复。同时，TRTC 客户端加入的房间类型必须和转录接口指定的房间类型保持一致，例如 SDK 创建房间用的是字符串房间号，那么转录任务的房间类型也需要相应设置成字符串房间号。
- 转录状态查询，您可以通过以下几种方式来得到转录相应的任务信息：
 - 成功发起 CreateCloudTranscription 任务后 15s 左右，调用 DescribeCloudTranscription 接口查询转录任务对应的信息，如果查询到状态为 Idle 说明转录机器人没有拉到上行的音频流，请检查房间内是否有主播上行。
 - 转录的任务信息会通过回调发送给您。具体回调信息请参见 [接收回调事件](#)。

API 请求频率限制实践教程

腾讯云 API 服务对每个用户的请求频率设置了上限，以保障系统稳定性和资源公平分配。当用户请求频率超过预设阈值时，系统会返回频率限制错误。默认转录接口的 QPS 为 20次/秒。可联系商务人员或者通过 [提交工单](#) 来申请提升限频。通常情况下，QPS 的设置值与在线报备的最高并发的比值是 1:20，例如 2000 路并发在线的转录任务，可以提升 QPS 到 100，实际需要请根据业务实现方式进行合理评估。

如遇到限频错误，短期内可以按照以下方法尝试快速调整：

- 降低请求频率至限制范围内。
- 业务实现请求队列。
- 添加适当的请求间隔时间。

长期方案可以按照以下方法进行调整：

- 实现指数退避重试机制，如第一次 3s 重试，第二次 6s 重试，第三次 12s 重试，以此类推直到重试成功。
- 优化业务逻辑，转录提前进房等待，减少并发 API 调用。

具体限频报错的示例，如下：

```
{
  "Response": {
    "Error": {
      "Code": "RequestLimitExceeded",
      "Message": "您当前每秒请求21次，超过每秒频率上限20，请稍后重试。",
    },
    "RequestId": "79c5cdbc-8a97-4d8b-be91-5c5e9143ad93"
  }
}
```

SDK 接入 AI 转录/翻译

最近更新时间：2026-05-09 17:58:31

本文将介绍如何使用 TRTC SDK 中的 `AITranscriberManager` 接口在客户端快速集成 AI 实时转录（语音转文字）与实时翻译功能。

方案介绍

TRTC 的 AI 实时转录与翻译功能支持将房间内的音频流实时转换为文本，并可同步翻译为多种目标语言。通过 SDK 提供的 `AITranscriberManager`，您可以直接在客户端发起转录任务、接收识别结果并进行控制。相较于服务端接入方式，通过 SDK 接入无需自建后台调用云端 API，极大地简化了开发流程。

前提条件

- 登录 [TRTC 控制台](#)，开通 TRTC 服务并 [创建应用](#)。
- 前往 [控制台 > 功能配置 > 增值功能](#)，开启 AI 智能识别的语音转文字、实时翻译功能。



说明：

- 开启前需领取[包月套餐体验版](#)、购买 [AI 智能识别套餐包](#) 或订阅 TRTC 包月套餐的部分版本才可使用，具体版本支持的功能详情可参见 [TRTC 包月套餐](#)，您可根据您的需要进行订阅。
- 语音转文字和实时翻译功能会根据调用量产生费用，详情请参见 [费用详情](#)。

接入流程

步骤1：集成 TRTC SDK

导入 TRTC SDK 到项目中，进入 TRTC 房间并开启本地麦克风音频采集与发布。目前已经支持 iOS、Android、Windows、Mac 以及 Web 客户端直接发起转录和翻译任务，相关指引可参考：

- [iOS 无 UI 集成指引](#)
- [Android 无 UI 集成指引](#)
- [Windows C++ 无 UI 集成指引](#)
- [Mac 无 UI 集成指引](#)
- [Web 无 UI 集成指引](#)

📌 说明:

- iOS、Android、Windows、Mac 端 SDK 导入后请继续阅读后续步骤。
- Web 端后续接入步骤说明请参考 [Web SDK 开启实时语音转录和翻译](#)。

步骤2: 获取 AITranscriberManager 实例

`AITranscriberManager` 是管理 AI 转录功能的核心类，您需要通过 `TRTCCloud` 获取其实例。

Android

```
import com.tencent.liteav.transcriber.AITranscriberManager;

TRTCCloud mTRTCCloud = TRTCCloud.sharedInstance(context);
AITranscriberManager aiTranscriberManager =
mTRTCCloud.getAITranscriberManager();
```

iOS&Mac

```
TRTCCloud *trtcCloud = [TRTCCloud sharedInstance];
AITranscriberManager *manager = [trtcCloud getAITranscriberManager];
```

Windows

```
liteav::ITRTCCloud* trtcCloud =
liteav::ITRTCCloud::getTRTCShareInstance();
liteav::AITranscriberManager* manager = trtcCloud-
>getAITranscriberManager();
```

步骤3: 设置事件监听

通过设置监听器（Listener），您可以接收到房间内参与转录的用户的转录状态变化、实时转录翻译信息以及错误信息。

Android

```
AITranscriberManager.AITranscriberListener listener = new
AITranscriberManager.AITranscriberListener() {
    @Override
    public void onRealtimeTranscriberStarted(String roomId, String
transcriberRobotId) {
        // 转录已开启
    }

    @Override
    public void onReceiveTranscriberMessage(String roomId,
AITranscriberManager.TranscriberMessage message) {
        // 处理实时转录与翻译消息
    }

    @Override
    public void onRealtimeTranscriberStopped(String roomId, String
transcriberRobotId, int reason) {
        // 转录已停止
    }

    @Override
    public void onRealtimeTranscriberError(String roomId, String
transcriberRobotId, int error, String errorInfo) {
        // 处理实时转录服务错误信息
    }
};

aiTranscriberManager.addListener(listener);
```

iOS&Mac

```
- (void)onRealtimeTranscriberStarted:(NSString *)roomId
transcriberRobotId:(NSString *)transcriberRobotId {
    // 转录已开启
}
```

```
- (void)onReceiveTranscriberMessage:(NSString *)roomId message:
(TranscriberMessage *)message {
    // 处理实时转录与翻译消息
}

- (void)onRealtimeTranscriberStopped:(NSString *)roomId
transcriberRobotId:(NSString *)transcriberRobotId reason:(NSInteger)reason
{
    // 转录已停止
}

- (void)onRealtimeTranscriberError:(NSString *)roomId transcriberRobotId:
(NSString *)transcriberRobotId error:(NSInteger)error errorInfo:(NSString
*)errorInfo {
    // 处理实时转录服务错误信息
}

[manager addListener:self];
```

Windows

```
class MyTranscriberListener : public liteav::AITranscriberListener {
public:
    void onRealtimeTranscriberStarted(const char* roomId, const char*
transcriberRobotId) override {
        // 转录已开启
    }

    void onReceiveTranscriberMessage(const char* roomId, const
liteav::TranscriberMessage& message) override {
        // 处理实时转录与翻译消息
    }

    void onRealtimeTranscriberStopped(const char* roomId, const char*
transcriberRobotId, int reason) override {
        // 转录已停止
    }

    void onRealtimeTranscriberError(const char* roomId, const char*
transcriberRobotId, int error, const char* errorInfo) override {
```

```
// 处理实时转录服务错误信息
}
};

MyTranscriberListener* listener = new MyTranscriberListener();
manager->addListener(listener);
```

TranscriberMessage 转录消息详情

字段名	类型	说明
<code>segmentId</code>	String	消息段的唯一标识 ID。可用于消息去重或排序。
<code>speakerUserId</code>	String	说话用户的 ID。
<code>sourceText</code>	String	识别出的源语言文本（Unicode 编码）。
<code>translations</code>	Map/List	翻译后的目标语言文本。
<code>timestamp</code>	long	消息生成的 UTC 时间戳，单位：毫秒。
<code>isCompleted</code>	bool	转录是否结束。 <code>true</code> ：表示该句话已说完，是最终结果。 <code>false</code> ：表示该句话还在进行中，是中间结果（流式更新）。

步骤4：启动转录任务

构建 `TranscriberParams` 参数对象，配置转录机器人 ID、源语言、目标翻译语言等信息，然后调用 `startRealtimeTranscriber` 开启服务。

Android

```
AITranscriberManager.TranscriberParams params = new
AITranscriberManager.TranscriberParams();
params.transcriberRobotId = "my_robot"; // 可选：指定机器人 ID
params.sourceLanguage = "zh"; // 源语言：中文
params.translationLanguages = Arrays.asList("en", "ja"); // 可选：不填时仅发起转录，不进行翻译
params.userIdsToTranscribe = Arrays.asList("userA"); // 可选：不填时表示转录房间内所有用户
```

```
aiTranscriberManager.startRealtimeTranscriber(params);
```

iOS&Mac

```
TranscriberParams *params = [[TranscriberParams alloc] init];
params.transcriberRobotId = @"my_robot"; // 可选：指定机器人 ID
params.sourceLanguage = @"zh"; // 源语言：中文
params.translationLanguages = @[@"en", @"ja"]; // 可选：不填时仅发起转录，不进行翻译
params.userIdsToTranscribe = @[@"userA"]; // 可选：不填时表示转录房间内所有用户

[manager startRealtimeTranscriber:params];
```

Windows

```
liteav::TranscriberParams params;
params.transcriberRobotId = "my_robot"; // 可选：指定机器人 ID
params.sourceLanguage = "zh"; // 源语言：中文
const char* targetLangs[] = {"en", "ja"};
params.translationLanguages = targetLangs; // 可选：不填时仅发起转录，不进行翻译
params.translationLanguagesCount = 2;
const char* transcribeUsers[] = {"userA"};
params.userIdsToTranscribe = transcribeUsers; // 可选：不填时表示转录房间内所有用户
params.userIdsToTranscribeCount = 1;

manager->startRealtimeTranscriber(params);
```

TranscriberParams 转录参数详情

参数字段	类型	必填	说明
<code>transcriberRobotId</code>	String	否	<p>转录机器人的唯一标识 ID。</p> <p>单个转录任务场景下，如果不指定，SDK 会默认生成一个 ID，格式为 <code>transcriber_\${roomId}_robot_\${userId}</code>。</p> <p>若同时发起多个转录任务，需要指定的机器人 ID。</p>

<code>sourceLanguage</code>	String	是	源语言代码。 指定源音频的语言种类，请填写标准的语言代码（例如“zh”）。 支持列表见下方 支持的语言代码 。
<code>translationLanguages</code>	List/Array	否	翻译目标语言代码列表。 如果需要翻译，请在此设置目标语言代码（例如“en”）。 支持列表见下方 支持的语言代码 。
<code>userIdsToTranscribe</code>	List/Array	否	指定转录的用户 ID 列表。 如果不填，默认转录房间内所有用户的音频。

SDK 会通过回调通知该接口的结果：

- 调用成功后，您会收到 `onRealtimeTranscriberStarted` 回调，表示转录任务已成功开启，此时您将可以通过 `onReceiveTranscriberMessage` 回调接收实时转录翻译信息。
- 调用失败后，您会收到 `onRealtimeTranscriberError` 回调，表示转录任务开启失败，此时您可以根据具体错误码做出相应措施（详情见下方 [服务端错误码详情](#)）。

步骤5：停止转录任务

当业务结束或不再需要转录时，调用 `stopRealtimeTranscriber` 停止任务以释放资源。请传入启动时使用的机器人 ID。（若启动时未指定机器人 ID 会默认生成一个，填写为空即可停止该机器人任务。）

Android

```
aiTranscriberManager.stopRealtimeTranscriber("my_robot");
```

iOS&Mac

```
[manager stopRealtimeTranscriber:@"my_robot"];
```

Windows

```
manager->stopRealtimeTranscriber("my_robot");
```

SDK 会通过回调通知该接口的结果：

- 调用成功后，您会收到 `onRealtimeTranscriberStopped` 回调，表示转录任务已成功停止，此时您将不会再接收到新的转录消息。
- 调用失败后，您会收到 `onRealtimeTranscriberError` 回调，表示转录任务停止失败，此时您可以根据具体错误码做出相应措施（详情见下方 [服务端错误码详情](#)）。

支持的语言代码

源语言 (sourceLanguage)

语言代码	语言名称
zh	中文
en	英语
zh-yue	中国粤语
vi	越南语
ja	日语
ko	韩语
id	印度尼西亚语
th	泰语
pt	葡萄牙语
tr	土耳其语
ar	阿拉伯语
es	西班牙语
hi	印地语
fr	法语
ms	马来语
fil	菲律宾语
de	德语
it	意大利语
ru	俄语
sv	瑞典语
da	丹麦语

no	挪威语
----	-----

⚠ 注意:

- 客户端发起实时转录功能目前支持 22 种语言：中文、英语、中国粤语、越南语、日语、韩语、印度尼西亚语、泰语、葡萄牙语、土耳其语、阿拉伯语、西班牙语、印地语、法语、马来语、菲律宾语、德语、意大利语、俄语、瑞典语、丹麦语、挪威语。如需其他语种支持可以 [联系我们](#)。
- 为了简化接入流程并提供最佳的效果，通过客户端 `AITranscriberManager` 发起的中文和英文实时转录默认使用标准版语言引擎中的最新 `16k_zh_en` 中英大模型语音识别，相关计费说明参考 [标准版语言引擎费用](#)。

翻译目标语言 (translationLanguages)

语言代码	语言名称
zh	中文
en	英语
es	西班牙语
pt	葡萄牙语
fr	法语
de	德语
ru	俄语
ar	阿拉伯语
ja	日语
ko	韩语
vi	越南语
ms	马来语
id	印度尼西亚语
it	意大利语
th	泰语

⚠ 注意：

- 实时翻译功能输入源语种和翻译后输出语种目前支持15种语言：中文、英语、西班牙语、葡萄牙语、法语、德语、俄语、阿拉伯语、日语、韩语、越南语、马来语、印度尼西亚语、意大利语、泰语。若前置ASR 转录语种不在上述范围，翻译功能无法开启。如需其他语种支持可以 [联系我们](#)。
- 由于受不同语境或语种差异的影响，AI 翻译的译文内容适用于辅助参考，不应作为唯一专业意见或结论。

服务端错误码详情

错误码	含义	建议处理措施
2000	参数错误。	建议检查请求参数是否合法。
2002	任务不存在。	若是调用 stop 接口时返回，可忽略。
2026	转录服务（ASR/翻译服务）未开通。	请前往控制台开通相关服务。
3000	内部错误。	建议重试。
4003	任务正在退出。	若是调用 stop 接口时返回，可忽略。
5000	资源过载。	建议采用退避策略重试。
5001	并发受限。	请联系产品侧提升并发限制。
-102009	主播不在房间内。	建议确认主播状态后延迟重试。
-102005	房间不存在。	建议确认房间状态后延迟重试。

视频通话接入 AI 转录/翻译

最近更新时间：2026-05-09 17:58:31

本文将介绍如何使用 TRTC 音视频通话 SDK 组件（TUICallKit）快速构建具备 AI 实时通话字幕和翻译能力的视频通话应用。

方案介绍

TUICallKit 是基于通话场景深度定制的一款产品，支持9人同时在线通话，提供微信视频聊天同款 UI 组件，仅需3步，最快1天上线，可节省90%开发时间。支持小程序、iOS、Android、Web、uni-app、Flutter 等平台全球互通。

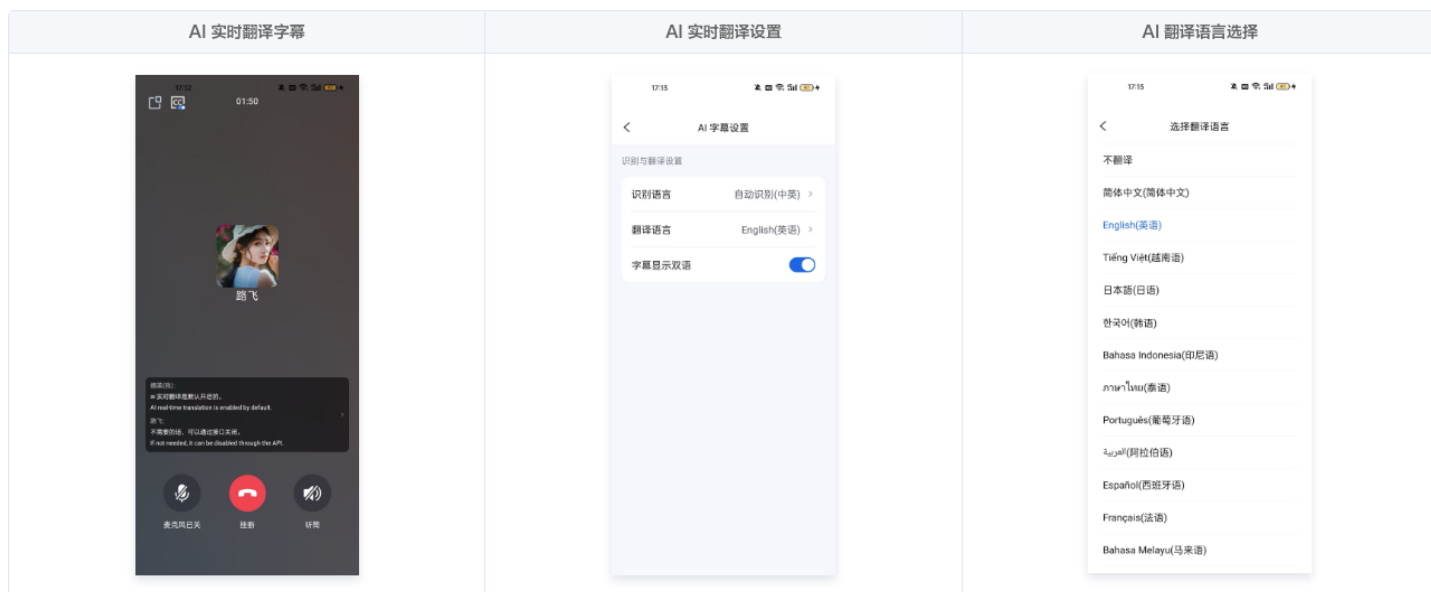
目前，TUICallKit 已经深度集成 TRTC AI 实时语音转文字和翻译引擎，仅需简单配置即可支持在通话界面中直接开启字幕显示和多语种翻译。

说明：

有关 TUICallKit 其他功能服务的详情介绍请参见 [视频通话组件介绍](#)。



效果演示



开通与计费

使用 TUICallKit 组件构建实时通话 AI 字幕和翻译功能包含两大独立计费模块：TUICallKit 使用费用、AI 实时转录相关费用。

⚠ 注意：

- TUICallKit 由实时音视频 TRTC 和即时通信 IM 提供底层技术支持，因此 TUICallKit 费用包含购买即时通信 IM 的费用及开通实时音视频 TRTC 包月套餐包费用两部分。TUICallKit 开通使用详情请参考 [TUICallKit 开通](#)。
- AI 实时转录费用主要由两个部分组成：RTC 音频通话费用、AI 语音转文本费用；若需要翻译功能则在前两者基础上额外增加实时翻译费用。以上能力都需要通过购买 AI 智能识别 或 TRTC 包月套餐包解锁，超量支持后付费，具体能力解锁所需套餐版本和费用详情请参见 [AI 智能识别计费](#)。

接入流程

步骤1：开通并接入视频通话 SDK

根据 [开通服务](#) 指引完成视频通话 SDK 组件（TUICallKit）开通，并参考以下各平台指引接入组件：

- [Android 快速接入](#)
- [iOS 快速接入](#)
- [Web&H5 \(Vue2/Vue3\) 快速接入](#)
- [Web&H5 \(React\) 快速接入](#)
- [uni-app \(uts 插件\) 快速接入](#)
- [uni-app \(小程序\) 快速接入](#)
- [uni-app \(客户端\) 快速接入](#)
- [微信小程序 快速接入](#)
- [Flutter 快速接入](#)

- [React Native 快速接入](#)

步骤2: 开启 AI 实时转录翻译功能

参见 [AI 智能识别计费](#) 按需购买解锁 AI 转录和翻译功能，并根据 [视频通话接入 AI 实时转录翻译](#) 指引在 TUICallKit 中开启 AI 实时转录翻译功能。

多人会议接入 AI 转录/翻译

最近更新时间：2026-05-09 17:58:31

本文将介绍如何使用 TRTC 多人会议 SDK 组件（TUIRoomKit）快速构建具备 AI 实时字幕的多人会议应用。

方案介绍

TUIRoomKit 是一款定位企业会议、网络研讨会、在线教育等多人音视频会话场景的含 UI 低代码组件，支持小程序、iOS、Android、Web、uni-app、Flutter 等平台全球互通。提供房间管理、成员管理、屏幕分享等会控功能，支持标清、高清、超高清等多种画质。通过集成该组件，仅需3步1天内就可以为您的 App 添加多人音视频会话功能。目前 TUIRoomKit 支持在 Web 和 Android 端开启 AI 实时字幕功能。

说明：

有关 TUIRoomKit 其他功能服务的详情介绍请参见 [多人会议组件介绍](#)。



开通与计费

使用 TUIRoomKit 组件实现在线会议实时 AI 字幕功能包含两大独立计费模块：TUIRoomKit 使用费用、AI 实时转录相关费用。

注意：

- TUIRoomKit 由实时音视频 TRTC 和即时通信 IM 提供底层技术支持，因此 TUIRoomKit 费用包含购买即时通信 IM 的费用及开通实时音视频 TRTC 包月套餐包费用两部分。TUIRoomKit 开通使用详情请参考 [开通服务](#)。

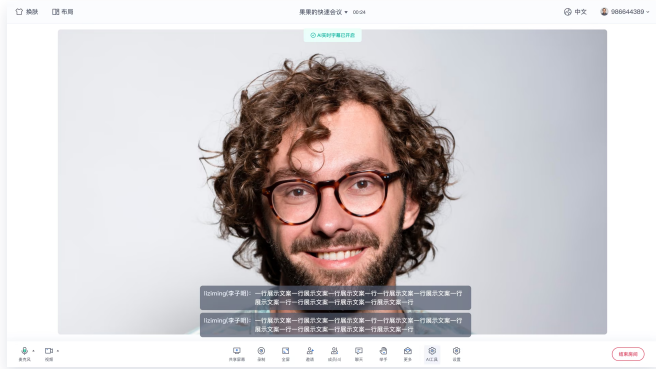
- AI 实时转录费用主要由两个部分组成：RTC 音频通话费用、AI 语音转文本费用。以上能力都需要通过购买 AI 智能识别 或 TRTC 包月套餐包解锁，超量支持后付费，具体能力解锁所需套餐版本和费用详情请参见 [AI 智能识别计费说明](#)。目前 TUIRoomKit 暂不支持实时翻译功能。

功能概述

接入 TUIRoomKit 后，您可以通过点击底部栏的“AI 助手”开启 AI 实时字幕功能，该功能可实现：

- AI 实时字幕：将会议过程中的讨论内容以字幕形式显示。
- AI 实时会议记录：将会议过程中的讨论内容以文字形式记录下来。

AI 实时字幕



AI 实时会议记录



接入流程

步骤1：开通接入多人会议 SDK

根据 [开通服务](#) 指引完成多人会议 SDK 组件（TUIRoomKit）服务开通。Web 与 Android SDK 接入相关指引参见：

- [Web&H5 \(Vue2/Vue3\) 快速接入](#)
- [Android 快速接入](#)

说明：

TUIRoomKit 的 AI 实时字幕功能目前仅支持 Web、Android 端使用。若您需要在 TUIRoomKit 其他平台（小程序、iOS、uni-app、Flutter 等）使用 AI 实时字幕功能请 [联系我们](#)。

步骤2：开通接入 AI 实时字幕

根据 [AI 智能识别计费](#) 指引购买套餐包解锁 AI 语音转文本功能，并在多人会议 SDK 组件（TUIRoomKit）启用 AI 实时字幕，开启方式请参考以下指引：

- [Web 开启方式](#)
- [Android 开启方式](#)

AI 转录/翻译迁移说明

最近更新时间：2026-05-09 17:58:31

TRTC AI 转录/翻译 2.0 对 1.0 的使用配置和回调方式进行了优化。若需要升级到 2.0 版本，请参见本文对于接入传参的差异说明。

发起任务的差异

通用参数的差异

2.0版本去掉了 SessionId 参数，可以不用传入 SessionId。

转录机器人参数的差异

2.0版本的机器人进房参数更简洁。差异如下：

- 回调转录翻译结果的方式：2.0 版本通过 SendCustomMode 参数控制回调转录翻译结果的方式，推荐设置 SendCustomMode 为 1 以使用自定义数据的方式回调，自定义数据回调方式相较于 1.0 版本自定义消息方式可靠性更高。
- 订阅用户的管理方式：1.0版本使用 TranscriptionMode、TargetUserId、TargetUserIdList 这三个参数管理订阅的用户，2.0 版本可以使用 SubscribeList 和 UnSubscribeList 参数通过订阅黑白名单的机制来管理订阅用户。

配置项	旧版本 1.0 RecognizeConfig 参数	新版本 2.0 AsrParam 参数	迁移建议
机器人信息	<ul style="list-style-type: none">• UserId• UserSig• MaxIdleTime	<ul style="list-style-type: none">• UserId• UserSig• MaxIdleTime	保持原始参数设置。
回调转录翻译结果	<ul style="list-style-type: none">• IMAdminUserId• IMAdminUserSig	SendCustomMode	推荐通过设置 SendCustomMode 为 1 来使用自定义数据方式进行回调。若需要继续使用自定义消息方式回调可设置 SendCustomMode 为 2。
订阅用户的管理	<ul style="list-style-type: none">• TranscriptionMode• TargetUserId• TargetUserIdList	<ul style="list-style-type: none">• SubscribeList• UnSubscribeList	推荐通过 SubscribeList 设置要订阅的用户，通过 UnSubscribeList 设置不订阅的用户。

			不设置这两个参数时默认订阅所有用户。使用黑白名单时，同一个用户同时在黑白名单时，以黑名单为主。
声纹、断句配置	<ul style="list-style-type: none"> • VoicePrint • TurnDetection 	不支持	不设置这两个参数。

语音识别参数的差异

对于语音识别（ASR）功能，1.0版本使用 RecognizeConfig 参数作为语音识别的配置参数，2.0版本使用 AsrParam 参数作为语音识别的配置参数，具体的配置子项参数名完全一致。

注意：2.0版本模糊识别支持的语言仅支持高级版 ASR 语种，详情参见 [高级版语种列表](#)。

翻译参数的差异

配置项	旧版本 1.0 TranslationConfig 参数	新版本 2.0 TranslationParam 参数	迁移建议
翻译目标语言	TargetLanguages	TargetLang	使用 TargetLang 配置项替代 TargetLanguages 配置项。
TTS 配置	TTSTConfig	不支持	如果需要使用 TTS 实现翻译同声传译的请继续使用 1.0 版本接口。
翻译术语表	Terminology	Terminology	-

查询任务的差异

2.0版本查询转录任务时只用填写 SdkAppId 和 TaskId，返回结果是任务的开始时间和状态。

参数	旧版本 1.0 DescribeAITranscription 参数	新版本 2.0 DescribeCloudTranscription 参数	迁移建议
输入参数	<ul style="list-style-type: none"> • TaskId • SdkAppId • SessionId 	<ul style="list-style-type: none"> • TaskId • SdkAppId 	只用填写 TaskId 和 SdkAppId 参数

返回参数	<ul style="list-style-type: none"> • StartTime • Status • TaskId • SessionId 	<ul style="list-style-type: none"> • StartTime • Status • TaskId 	-
------	--	---	---

停止任务的差异

相较于 1.0 版本，2.0版本停止任务需要填写 SdkAppId 参数，其余用法保持一致。

AI 转录/翻译回调说明

最近更新时间：2026-05-09 17:58:31

本文针对 **TRTC 后台接入 AI 转录/翻译 2.0** 的回调事件进行具体说明。

配置信息

实时音视频 TRTC 控制台支持自助配置回调信息，选择 **AI 转录 2.0 回调**，配置完成后即可接收事件回调通知。详细操作指引请参见 [回调配置](#)。

⚠ 注意：

您需要提前准备以下信息：

- 必要项：接收回调通知的 HTTP / HTTPS 服务器地址。
- 可选项：计算签名的 **密钥 key**，由您自定义一个最大32个字符的 key，以大小写字母及数字组成。

超时重试

事件回调服务器在发送消息通知后，5秒内没有收到您的服务器的响应，即认为通知失败。首次通知失败后会立即重试，后续失败会以10秒的间隔继续重试，直到消息存续时间超过1分钟，不再重试。

事件回调消息格式

事件回调消息以 HTTP / HTTPS POST 请求发送给您的服务器，其中：

- 字符编码格式：UTF-8。
- 请求：body 格式为 JSON。
- 应答：HTTP STATUS CODE = 200，服务端忽略应答包具体内容，为了协议友好，建议客户应答内容携带 JSON: {"code":0}。
- 包体示例：下述为“启动 AI 转录与翻译 2.0”事件的包体示例。

```
{
  "EventGroupId": 14,
  "EventType": 1401,
  "CallbackTs": 1687770730166,
  "EventInfo": {
    "EventMsTs": 1622186275757,
    "TaskId": "xxx",
    "RoomId": "1234",
    "RoomIdType": 0,
    "RobotId": "trtc_partner_test_1",
    "Payload": {
```

```
"Status": 0
}
}
}
```

参数说明

回调消息参数

- 事件回调消息的 header 中包含以下字段：

字段名	含义
Content-Type	application/json。
Sign	签名值。
SdkAppId	创建应用时控制台分配的 SdkAppId。

- 事件回调消息的 body 中包含以下字段：

字段名	类型	含义
EventGroupId	Number	事件组 ID，转录事件组事件固定为14。
EventType	Number	回调通知的事件类型。
CallbackTs	Number	事件回调服务器向您的服务器发出回调请求的 Unix 时间戳，单位为毫秒。
EventInfo	JSON Object	事件信息 。

事件组 ID

字段名	值	含义
EVENT_GROUP_CLOUD_TRANSCRIPTION	14	转录事件组。

事件类型

字段名	值	含义
EVENT_TYPE_CLOUD_TRANSCRIPTION_START	1401	转录开始状态回调。
EVENT_TYPE_CLOUD_TRANSCRIPTION_END	1402	转录任务结束状态回调。

ON_STOP		
EVENT_TYPE_CLOUD_TRANSCRIPTION_ASR_MSG	1403	回调转录 ASR 的完整句子。
EVENT_TYPE_CLOUD_TRANSCRIPTION_TRANSLATE_MSG	1404	回调转录翻译的完整句子。

EVENT_TYPE_CLOUD_TRANSCRIPTION_START

字段名	类型	含义
EventMsTs	Number	事件发生的 Unix 时间戳，单位为毫秒。
TaskId	String	任务 ID。
RoomId	String	TRTC 的房间 ID。
RoomIdType	Integer	<ul style="list-style-type: none">0: 表示数字房间号。1: 表示字符串房间号。
RobotId	String	机器人 ID。
Payload.Status	Number	<ul style="list-style-type: none">0: 启动转录任务。1: 启动转录任务失败。

```
{
  "EventGroupId": 14,
  "EventType": 1401,
  "CallbackTs": 1687770730166,
  "EventInfo": {
    "EventMsTs": 1622186275757,
    "TaskId": "xxx",
    "RoomId": "1234",
    "RoomIdType": 0,
    "RobotId": "trtc_partner_test_1",
    "Payload": {
      "Status": 0
    }
  }
}
```

EVENT_TYPE_CLOUD_TRANSCRIPTION_STOP

字段名	类型	含义
EventMsTs	Number	事件发生的 Unix 时间戳，单位为毫秒。
TaskId	String	任务 ID。
RoomId	String	TRTC 的房间 ID。
RoomIdType	Integer	<ul style="list-style-type: none"> 0: 表示数字房间号。 1: 表示字符串房间号。
RobotId	String	机器人 ID。
Payload.LeaveCode	Integer	<ul style="list-style-type: none"> 0: 转录模块正常调用停止转录退出。 1: 转录机器人被客户踢出房间。 2: 客户解散房间。 3: 服务器将转录机器人踢出。 4: 服务器解散房间。 99: 代表房间内除了转录机器人没有其他用户流，超过指定时间退出。 101: 同一用户重复进入相同房间导致机器人退出。

```

{
  "EventGroupId": 14,
  "EventType": 1402,
  "CallbackTs": 1687770730166,
  "EventInfo": {
    "EventMsTs": 1622186275757,
    "TaskId": "xxx",
    "RoomId": "1234",
    "RoomIdType": 0,
    "RobotId": "trtc_partner_test_1",
    "Payload": {
      "LeaveCode": 0
    }
  }
}

```

EVENT_TYPE_CLOUD_TRANSCRIPTION_ASR_MSG

字段名	类型	含义
-----	----	----

EventMsTs	Number	事件发生的 Unix 时间戳，单位为毫秒。
TaskId	String	任务 ID。
RoomId	String	TRTC 的房间 ID。
RoomIdType	Integer	<ul style="list-style-type: none"> 0: 表示数字房间号。 1: 表示字符串房间号。
RobotId	String	机器人 ID。
Payload	JSON Object	<p>为 JSON 对象:</p> <pre> { "UserId": "Trtc_User_0", "Text": "xxxx", "StartTimeMs": 108, "EndTimeMs": 10568, "RoundId": "40c9e724-3268-4b66-a9ff-41ed44d8edb6", "StartUtcMs": 1761568438912, "EndUtcMs": 1761568449372 } </pre>

```

{
  "EventGroupId": 14,
  "EventType": 1403,
  "CallbackTs": 1687770730166,
  "EventInfo": {
    "EventMsTs": 1761568449890,
    "TaskId": "xxx",
    "RoomId": "1234",
    "RoomIdType": 0,
    "RobotId": "trtc_partner_test_1",
    "Payload": {
      "UserId": "Trtc_User_0",
      "Text": "Oh yeah? What's the ultimate predator? What's the ultimate predator? What's the enemy you harbor in your own heart? Who hates you? That's the ultimate predator.",
      "StartTimeMs": 108,

```

```

        "EndTimeMs": 10568,
        "RoundId": "40c9e724-3268-4b66-a9ff-41ed44d8edb6",
        "StartUtcMs": 1761568438912,
        "EndUtcMs": 1761568449372
    }
}
}

```

EVENT_TYPE_CLOUD_TRANSCRIPTION_TRANSLATE_MSG

字段名	类型	含义
EventMsTs	Number	事件发生的 Unix 时间戳，单位为毫秒。
TaskId	String	任务 ID。
RoomId	String	TRTC 的房间 ID。
RoomIdType	Integer	<ul style="list-style-type: none"> 0: 表示数字房间号。 1: 表示字符串房间号。
RobotId	String	机器人 ID。
Payload	JSON Object	为 JSON 对象： <pre> { "UserId": "Trtc_User_0", "Text": "Presume, was exactly the same way. ", "TranslateMsg": [{ "Language": "fr", "Text": "Je suppose, c'était exactement la même chose. " }, { "Language": "zh", "Text": "大概也是一模一样的。" }], "StartTimeMs": 108, </pre>

```
"EndTimeMs": 10568,  
"RoundId": "40c9e724-3268-  
4b66-a9ff-41ed44d8edb6",  
"StartUtcMs": 1761568438912,  
"EndUtcMs": 1761568449372  
}
```

```
{  
  "EventGroupId": 14,  
  "EventType": 1404,  
  "CallbackTs": 1687770730166,  
  "EventInfo": {  
    "EventMsTs": 1761568449890,  
    "TaskId": "xxx",  
    "RoomId": "1234",  
    "RoomIdType": 0,  
    "RobotId": "trtc_partner_test_1",  
    "Payload": {  
      "UserId": "Trtc_User_0",  
      "Text": "presume, was exactly the same way. ",  
      "TranslateMsg": [  
        {  
          "Language": "fr",  
          "Text": "Je suppose, c'était exactement la même chose."  
        },  
        {  
          "Language": "zh",  
          "Text": "大概也是一模一样的。"  
        }  
      ],  
      "StartTimeMs": 108,  
      "EndTimeMs": 10568,  
      "RoundId": "40c9e724-3268-4b66-a9ff-41ed44d8edb6",  
      "StartUtcMs": 1761568438912,  
      "EndUtcMs": 1761568449372  
    }  
  }  
}
```

```
}
```

计算签名

签名由 HMAC SHA256 加密算法计算得出，您的事件回调接收服务器收到回调消息后，通过同样的方式计算出签名，相同则说明是腾讯云的实时音视频的事件回调，没有被伪造。签名的计算如下所示：

```
//签名 Sign 计算公式中 key 为计算签名 Sign 用的加密密钥。  
Sign = base64 ( hmacsha256 (key, body) )
```

⚠ 注意：

body 为您收到回调请求的原始包体，不要做任何转化，示例如下：

```
body="  
{  
  \n\t"EventGroupId":\t1,\n\t"EventType":\t103,\n\t"CallbackTs":  
  \t1615554923704,\n\t"EventInfo":\t{\n\t\t"RoomId":\t12345,\n\t\t"EventTs":\t1608441737,\n\t\t"UserId":\t"test",\n\t\t"Unique  
  Id":\t1615554922656,\n\t\t"Role":\t20,\n\t\t"Reason":\t1\n\t}  
}"
```

签名校验示例

Java

```
import javax.crypto.Mac;  
import javax.crypto.spec.SecretKeySpec;  
import java.util.Base64;  
//# 功能：第三方回调sign校验  
//# 参数：  
//# key：控制台配置的密钥key  
//# body：腾讯云回调返回的body体  
//# sign：腾讯云回调返回的签名值sign  
//# 返回值：  
//# Status：OK 表示校验通过，FAIL 表示校验失败，具体原因参考Info  
//# Info：成功/失败信息  
  
public class checkSign {
```

```
public static String getResultSign(String key, String body) throws
Exception {
    Mac hmacSha256 = Mac.getInstance("HmacSHA256");
    SecretKeySpec secret_key = new SecretKeySpec(key.getBytes(),
"HmacSHA256");
    hmacSha256.init(secret_key);
    return
Base64.getEncoder().encodeToString(hmacSha256.doFinal(body.getBytes()));
}

public static void main(String[] args) throws Exception {
    String key = "123654";
    String body = "{\n" + "\t\"EventGroupId\":\t2,\n" +
"\t\"EventType\":\t204,\n" + "\t\"CallbackTs\":\t1664209748188,\n" +
"\t\"EventInfo\":\t{\n" + "\t\t\"RoomId\":\t8489,\n" +
"\t\t\"EventTs\":\t1664209748,\n" + "\t\t\"EventMsTs\":\t1664209748180,\n"
+ "\t\t\"UserId\":\t\"user_85034614\", \n" + "\t\t\"Reason\":\t0\n" +
"\t}\n" + "}";
    String Sign = "kkoFeO3Oh2ZHnJtg8tEAQhtXK16/KI05W3BQff8IvGA=";
    String resultSign = getResultSign(key, body);

    if (resultSign.equals(Sign)) {
        System.out.println("{\"Status': 'OK', 'Info': '校验通过'}");
    } else {
        System.out.println("{\"Status': 'FAIL', 'Info': '校验失败'}");
    }
}
}
```

Python

```
# -*- coding: utf8 -*-
import hmac
import base64
from hashlib import sha256

# 功能：第三方回调sign校验
# 参数：
# key：控制台配置的密钥key
# body：腾讯云回调返回的body体
# sign：腾讯云回调返回的签名值sign
```

```
# 返回值:
# Status: OK 表示校验通过, FAIL 表示校验失败, 具体原因参考Info
# Info: 成功/失败信息

def checkSign(key, body, sign):
    temp_dict = {}
    computSign = base64.b64encode(hmac.new(key.encode('utf-8'),
body.encode('utf-8'), digestmod=sha256).digest()).decode('utf-8')
    print(computSign)
    if computSign == sign:
        temp_dict['Status'] = 'OK'
        temp_dict['Info'] = '校验通过'
        return temp_dict
    else:
        temp_dict['Status'] = 'FAIL'
        temp_dict['Info'] = '校验失败'
        return temp_dict

if __name__ == '__main__':
    key = '123654'
    body = "{\n" + "\t\"EventGroupId\":\t2,\n" +
"\t\"EventType\":\t204,\n" + "\t\"CallbackTs\":\t1664209748188,\n" +
"\t\"EventInfo\":\t{\n" + "\t\t\"RoomId\":\t8489,\n" +
"\t\t\"EventTs\":\t1664209748,\n" + "\t\t\"EventMsTs\":\t1664209748180,\n"
+ "\t\t\"UserId\":\t\"user_85034614\", \n" + "\t\t\"Reason\":\t0\n" +
"\t}\n" + "}"
    sign = 'kkoFe030h2ZHnJtg8tEAQhtXK16/KI05W3BQff8IvGA='
    result = checkSign(key, body, sign)
    print(result)
```

PHP

```
<?php

class TlsEventSig {

    private $key = false;
    private $body = false;
```

```
public function __construct( $key, $body ) {
    $this->key = $key;
    $this->body = $body;
}

private function __hmacsha256() {
    $hash = hash_hmac( 'sha256', $this->body, $this->key, true );
    return base64_encode( $hash );
}

public function genEventSig() {
    return $this->__hmacsha256();
}
}

$key="789";
$data="
{\n\t\"EventGroupId\":\t1,\n\t\"EventType\":\t101,\n\t\"CallbackTs\":\t160
8086882372,\n\t\"EventInfo\":\t{\n\t\t\"RoomId\":\t20222,\n\t\t\"EventTs\"
:\t1608086882,\n\t\t\"UserId\":\t\"222222_phone\"\n\t}\n}";

$api = new TlsEventSig($key, $data);
echo $api->genEventSig();
```

Golang

```
package main
import "fmt"
import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/base64"
)

func main () {
    var data = "
{\n\t\"EventGroupId\":\t1,\n\t\"EventType\":\t101,\n\t\"CallbackTs\":\t160
8086882372,\n\t\"EventInfo\":\t{\n\t\t\"RoomId\":\t20222,\n\t\t\"EventTs\"
:\t1608086882,\n\t\t\"UserId\":\t\"222222_phone\"\n\t}\n}
"
    var key = "789"
```

```
//JSRUN引擎2.0，支持多达30种语言在线运行，全仿真在线交互输入输出。  
fmt.Println(hmacsha256(data,key))  
}  
  
func hmacsha256(data string, key string) string {  
    h := hmac.New(sha256.New, []byte(key))  
    h.Write([]byte(data))  
    return base64.StdEncoding.EncodeToString(h.Sum(nil))  
}
```

直接使用

实时语音识别

最近更新时间：2026-05-09 17:58:31

接口描述

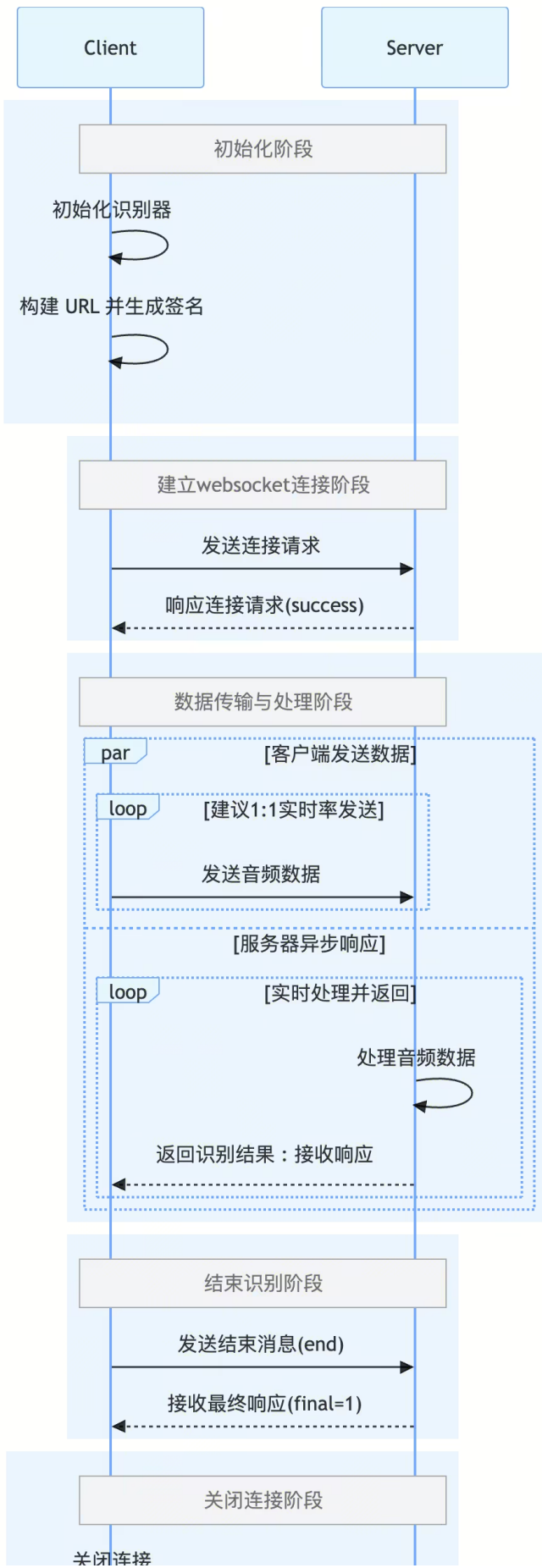
本接口服务采用 WebSocket 协议，对实时音频流进行识别，同步返回识别结果，达到"边说边出文字"的效果。

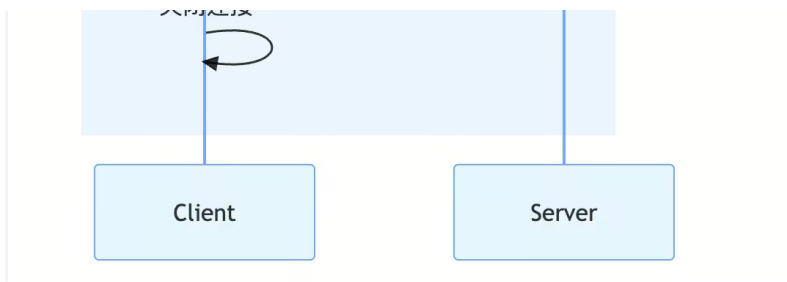
接口要求

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、英文。
支持行业	通用。
音频属性	<ul style="list-style-type: none">采样率：16000Hz 或 8000Hz。采样精度：16bits。声道：单声道（mono）。
音频格式	pcm。
请求协议	WSS 协议。
请求地址	<code>wss://域名/asr/v2/?\<appid\>?\{请求参数\}</code> 。
响应格式	JSON。
数据发送	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。音频发送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。

业务流程图





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。
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。 <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;"> <p>⚠ 注意： 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> </div>
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	识别结果类型： <ul style="list-style-type: none"> 0：一段话开始识别。

		<ul style="list-style-type: none"> • 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)。 • 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) 根据发送的音频情况, 识别过程中可能返回的 slice_type 序列有: <ul style="list-style-type: none"> ○ 0-1-2: 一段话开始识别、识别中 (可能有 1 返回)、识别结束。 ○ 0-2: 一段话开始识别、识别结束。 ○ 2: 直接返回一段话完整的识别结果。
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。
start_time	Integer	当前一段话结果在整个音频流中的起始时间。
end_time	Integer	当前一段话结果在整个音频流中的结束时间。
voice_text_str	String	当前一段话文本结果, 编码为 UTF8。
word_size	Integer	当前一段话的词结果个数 (word_list 数组的长度与 word_size 的值相等)。
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"> • word: String 类型, 该词的内容。 • start_time: Integer 类型, 该词在整个音频流中的起始时间。 • end_time: Integer 类型, 该词在整个音频流中的结束时间。 • 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。

握手阶段

请求格式

握手阶段, 客户端发起 WebSocket 连接请求, 请求 URL 格式为:

正式环境: `wss://asr.cloud-rtc.com/asr/v2/<appid>?{请求参数}`

在 WebSocket header 中填写以下字段:

key	必填	value 含义
X-TRTC-SdkAppId	是	TRTC 应用 ID, 获取方式见 应用基本信息 。

X-TRTC-UserSig	是	TRTC 签名，计算及使用请参见 用户鉴权 ，其中的 UserID 等于 URL 参数中的 voice_id。
----------------	---	---

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

```
key1=value1&key2=value2...
```

参数说明：

参数名	必填	类型	描述
secretid	是	String	腾讯云注册账号的密钥 secretid，可通过 API 密钥管理 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。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。
nonce	是	Integer	随机正整数。用户需自行生成，最长 10 位。示例值：8743357。
engine_model_type	是	String	引擎模型类型，目前支持： <ul style="list-style-type: none"> 8k_zh：中文通用，常用于电话场景。 16k_zh_en：中英文通用。
voice_id	是	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。 <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;"> <p>⚠ 注意： 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> </div>
voice_format	否	Integer	语音编码方式，目前支持：1: PCM。
needvad	否	Integer	<ul style="list-style-type: none"> 0: 关闭 vad。 1: 开启 vad。 默认为 0。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。
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。
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。 <ul style="list-style-type: none"> 0: 不过滤脏词。 1: 过滤脏词。 2: 将脏词替换为 "*"
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。 <ul style="list-style-type: none"> 0: 不过滤语气词。 1: 部分过滤。 2: 严格过滤。
filter_punc	否	Integer	是否过滤句末的句号（目前支持中文普通话引擎）。默认为 0。 <ul style="list-style-type: none"> 0: 不过滤句末的句号。 1: 过滤句末的句号。
convert_num_mode	否	Integer	是否进行阿拉伯数字智能转换（目前支持中文普通话引擎）： <ul style="list-style-type: none"> 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。
word_info	否	Integer	是否显示词级别时间戳： <ul style="list-style-type: none"> 0: 不显示。 1: 显示，不包含标点时间戳。
vad_silence_time	否	Integer	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。
max_speak_time	否	Integer	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。
signature	是	String	接口签名参数，用于兼容第三方服务。如果不涉及，值与 X-TRTC-UserSig 一致。
hotword_list	否	String	临时热词表：该参数用于提升识别准确率。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，例如：“腾讯云 5”或“ASR 11”。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，例如：“腾讯云 10,语音识别 5,ASR 11”。

⚠ 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。
- 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。

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率。热词不能包含空格，例如：ASR 腾讯云。

language	否	String	该字段当前版本不生效，传入后将被忽略，不影响请求结果。
sample_rate	否	String	采样速率，可选值：8k 或 16k。

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": "faab2b17-ba43-4ea4-be58-a03180ba26dd"
}
```

识别阶段

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。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。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": "faab2b17-ba43-4ea4-be58-a03180ba26dd",
  "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": "faab2b17-ba43-4ea4-be58-a03180ba26dd",
  "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": "faab2b17-ba43-4ea4-be58-a03180ba26dd",
  "message_id": "8d9e6337-2c8a-49e2-ae9-9179d71ab7a8",
  "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": "faab2b17-ba43-4ea4-be58-a03180ba26dd",
  "message_id": "8d9e6337-2c8a-49e2-ae9-9179d71ab7a8"
}
```

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。 <div style="border: 1px solid #00a0e3; padding: 5px; margin-top: 10px;">注意： 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。</div>
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 6 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	<p>因机器负载过高、网络抖动等导致失败，请重新发起新识别。</p> <p>⚠ 注意： 该问题通常为偶发，少量出现可忽略，发起新识别即可。</p>
5001	<p>因机器负载过高、网络抖动等导致失败，请重新发起新识别。</p> <p>⚠ 注意： 该问题通常为偶发，少量出现可忽略，发起新识别即可。</p>
5002	<p>因机器负载过高、网络抖动等导致失败，请重新发起新识别。</p> <p>⚠ 注意： 该问题通常为偶发，少量出现可忽略，发起新识别即可。</p>

一句话识别

最近更新时间：2026-05-09 17:58:31

接口描述

本接口用于对 60 秒之内的短音频文件进行识别。

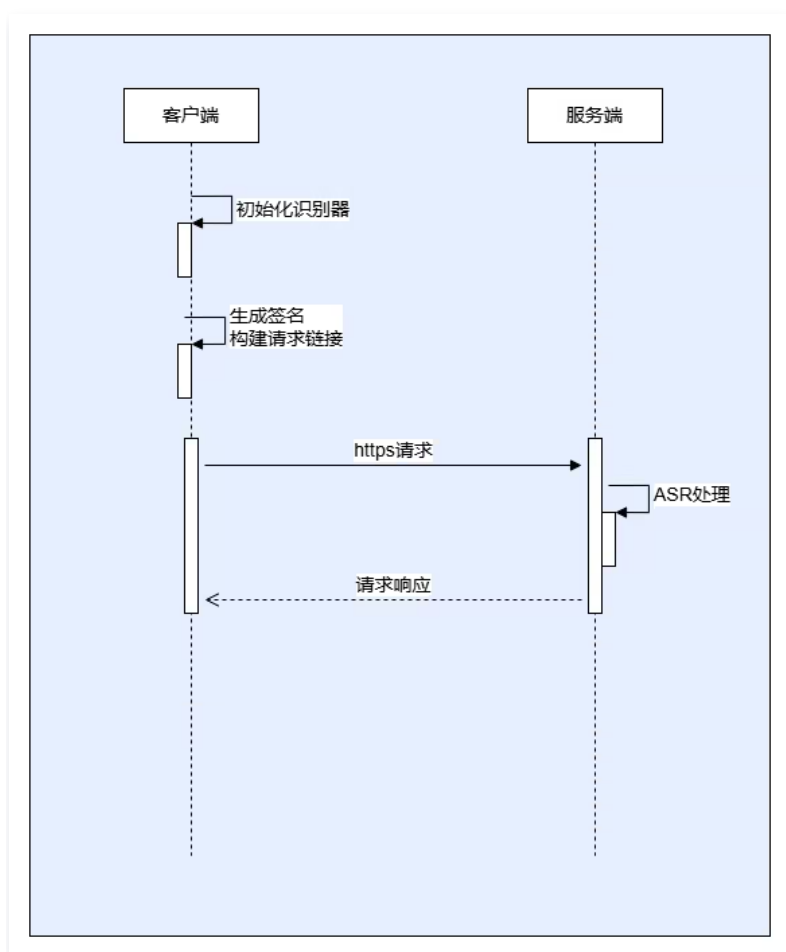
- 音频格式支持 wav、pcm、ogg-opus、mp3、m4a。
- 请求方法为 HTTP POST，Content-Type 为 "application/json; charset=utf-8"。

接口要求

集成一句话识别 API 时，需按照以下要求。

内容	说明
语言种类	支持中文普通话、英文。
支持行业	通用。
音频属性	<ul style="list-style-type: none">• 采样率：16000Hz（pcm 格式例外，可通过 InputSampleRate 参数支持 8000Hz，详见 输入参数 说明）。• 采样精度：16bits。• 声道：单声道（mono）。
音频格式	wav、pcm、ogg-opus、mp3、m4a。
请求协议	https 协议。
请求地址	<code>https://域名/v1/SentenceRecognition?请求参数</code> 。
响应格式	JSON。
数据发送	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。
并发限制	默认单账号限制并发数为 30 次每秒。

业务流程图



接口调用流程

请求格式

客户端发起 HTTP POST 请求，请求 URL 格式为：

正式环境：`https://asr.cloud-rtc.com/v1/SentenceRecognition?{\请求参数}`

在请求 header 中填写以下字段：

key	必填	value 含义
X-TRTC-SdkAppId	是	TRTC 应用 ID，获取方式见 应用基本信息 。示例值：1400188366。
X-TRTC-UserSig	是	TRTC 签名，计算及使用请参见 用户鉴权 ，其中的 UserID 等于 URL 参数中的 RequestId。
X-Business-Name	否	默认不填。内部业务与平台沟通后填写，不可随意填写。

请求参数

请求参数格式为：`AppId=value1&RequestId=value2`

参数说明:

参数名	必填	类型	描述
AppId	是	String	腾讯云注册账号的 AppId，可通过 API 密钥管理 获取。
RequestId	是	String	全局请求唯一 ID (uuid)，该参数用来生成请求 UserSig。

输入参数

一句话识别的请求消息体，格式为 json。

参数说明:

参数名	必填	类型	描述
EngServiceType	是	String	引擎模型类型，目前支持： 16k_zh_en：中英文通用。
SourceType	是	Integer	语音数据来源： <ul style="list-style-type: none">0：语音 URL。1：语音数据 (post body)。
VoiceFormat	是	String	识别音频的音频格式，支持 wav、pcm、ogg-opus、mp3、m4a。
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。
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。必须使用 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 编码后)。
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)
WordInfo	否	Integer	是否显示词级别时间戳： <ul style="list-style-type: none">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。
FilterDirty	否	Integer	<p>是否过滤脏词（中文普通话）：</p> <ul style="list-style-type: none"> 0: 不过滤脏词。 1: 过滤脏词。 2: 将脏词替换为 *。 <p>默认值为 0。</p>
FilterModal	否	Integer	<p>是否过滤语气词（中文普通话）：</p> <ul style="list-style-type: none"> 0: 不过滤语气词。 1: 部分过滤。 2: 严格过滤。 <p>默认值为 0。</p>
FilterPunc	否	Integer	<p>是否过滤标点符号（中文普通话）：</p> <ul style="list-style-type: none"> 0: 不过滤。 1: 过滤句末标点。 2: 过滤所有标点。 <p>默认值为 0。</p>
ConvertNumMode	否	Integer	<p>是否进行阿拉伯数字智能转换：</p> <ul style="list-style-type: none"> 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。 <p>默认值为 1。</p>
HotwordList	否	String	<p>临时热词表：该参数用于提升识别准确率。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，例如：“热词 5”或“ASR 11”；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，例如：“热词 10,语音识别 5,ASR 11”。</p> <div style="border: 1px solid #00aaff; padding: 10px; margin-top: 10px;"> <p>⚠ 注意：</p> <ul style="list-style-type: none"> 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。 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”。 </div>
InputSampleRate	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，</p>

那么该 8k 采样率的 pcm 音频可以在 16k_zh 引擎下正常识别。

⚠ 注意:

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。

			那么该 8k 采样率的 pcm 音频可以在 16k_zh 引擎下正常识别。
language	否	String	该字段当前版本不生效，传入后将被忽略，不影响请求结果。

响应格式

响应消息为 json 格式，所有返回字段都位于顶层 Response 对象之内。

参数说明：

参数名	类型	描述
Response	Object	顶层响应对象，包含以下字段。
Response.Result	String	识别结果。
Response.AudioDuration	Integer	请求的音频时长，单位为 ms。
Response.WordSize	Integer	词时间戳列表的长度。 ⚠ 注意: 此字段可能返回 null，表示取不到有效值。
Response.WordList	Array of SentenceWord	词时间戳列表。 ⚠ 注意: 此字段可能返回 null，表示取不到有效值。
Response.StartTimestamps	Array of Integer	每个词的起始时间戳列表（单位 ms）。
Response.EndTimestamps	Array of Integer	每个词的结束时间戳列表（单位 ms）。
Response.RequestId	String	请求的 RequestId，定位问题时需要提供该次请求的 RequestId。

SentenceWord 定义为：

名称	类型	描述
Word	String	词结果。
StartTime	Integer	词在音频中的开始时间。
EndTime	Integer	词在音频中的结束时间。

示例

用户通过 SourceType=1 请求一句话识别，请求如下

请求网址：

```
https://asr.cloud-rtc.com/v1/SentenceRecognition?
AppId=12345&RequestId=2f5e8263-a23b-4e65-be22-6b7c19bd4064
```

输入参数：

```
{
  "EngSerViceType": "16k_zh_en",
  "SourceType": 1,
  "VoiceFormat": "wav",
  "Data": "xxxxxxxx",
  "DataLen": 8
}
```

响应结果：

```
{
  "Response": {
    "Result": "我们能接受的范围内。",
    "RequestId": "2f5e8263-a23b-4e65-be22-6b7c19bd4064",
    "AudioDuration": 1840
  }
}
```

错误码

数值	说明
----	----

400	音频发送失败。
413	音频超限。
400	音频为空。
400	请求消息体格式错误。
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是否与调用参数一致。
5000	<p>因机器负载过高、网络抖动等导致失败，请重新发起新识别。</p> <div style="border: 1px solid #00aaff; padding: 5px;"><p>⚠ 注意： 该问题通常为偶发，少量出现可忽略，发起新识别即可。</p></div>
5001	<p>因机器负载过高、网络抖动等导致失败，请重新发起新识别。</p> <div style="border: 1px solid #00aaff; padding: 5px;"><p>⚠ 注意： 该问题通常为偶发，少量出现可忽略，发起新识别即可。</p></div>
5002	<p>因机器负载过高、网络抖动等导致失败，请重新发起新识别。</p> <div style="border: 1px solid #00aaff; padding: 5px;"><p>⚠ 注意： 该问题通常为偶发，少量出现可忽略，发起新识别即可。</p></div>

录音文件识别

最近更新时间：2026-05-09 17:58:31

接口描述

本接口可对较长的录音文件进行识别

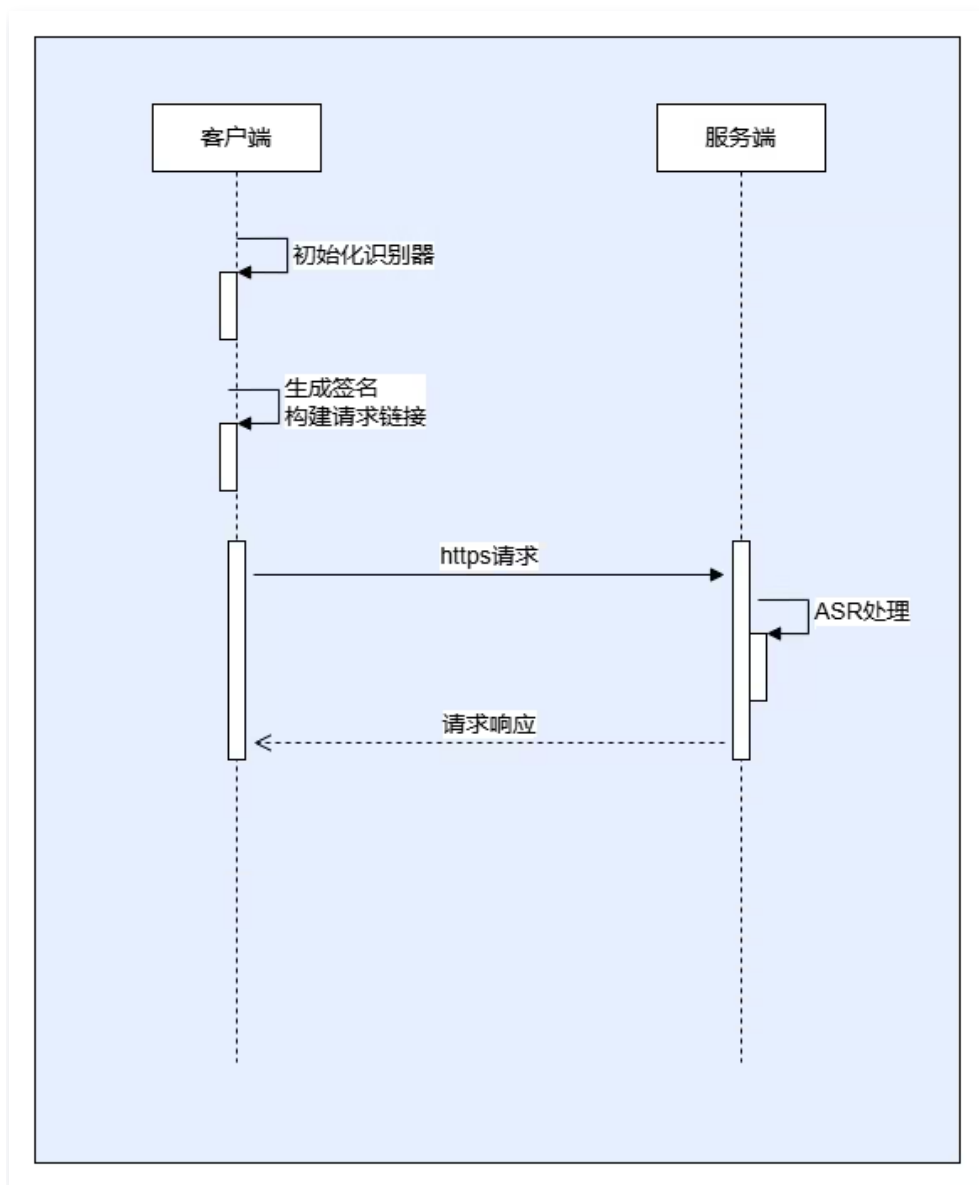
- 音频格式支持 wav、ogg-opus、mp3、m4a。
- 请求方法为 HTTP POST，Content-Type 为 "application/json; charset=utf-8"。
- 音频提交方式：本接口支持音频 URL、本地音频文件两种请求方式。
- 音频限制：音频 URL 时长不能大于 12 小时，文件大小不超过 1GB；本地音频文件不能大于 5MB。
- 如何获取识别结果：支持回调或轮询的方式获取结果。
- 识别结果有效时间：识别结果在服务端保存 24 小时。

接口要求

集成录音文件识别 API 时，需按照以下要求。

内容	说明
语言种类	支持中文普通话、英文。
支持行业	通用。
音频属性	采样率：16000Hz 采样精度：16bits 声道：单声道（mono）。
音频格式	wav、ogg-opus、mp3、m4a。
请求协议	https 协议。
请求地址	https://域名/v1/CreateRecTask?{\请求参数}。
响应格式	JSON。
数据发送	支持本地语音文件上传和语音 URL 上传两种请求方式。音频 URL 时长不能大于 12 小时，文件大小不超过 1GB；本地音频文件不能大于 5MB。
并发限制	默认单账号限制并发数为 20 次每秒。

业务流程图



接口调用流程

请求格式

客户端发起 HTTP POST 请求，请求 URL 格式为：

正式环境：`https://asr.cloud-rtc.com/v1/CreateRecTask?{请求参数}`

在请求 header 中填写以下字段：

key	必填	value 含义
X-TRTC-SdkAppId	是	TRTC 应用 ID，获取方式请参见 应用基本信息 。
X-TRTC-UserSig	是	TRTC 签名，计算及使用请参见 用户鉴权 ，其中的 UserID 等于 URL 参数中的 RequestId。

请求参数

请求参数格式为：`AppId=value1&RequestId=value2`

参数说明

参数名	必填	类型	描述
AppId	是	String	腾讯云注册账号的 appid，可通过 API 密钥管理 获取。
RequestId	是	String	全局请求唯一 ID（UUID），该参数用来生成请求 UserSig。

输入参数

录音文件识别的请求消息体，格式为 json。

参数说明

参数名	必填	类型	描述
EngineModelType	是	String	引擎模型类型，目前支持： <ul style="list-style-type: none">16k_zh：中文通用。16k_zh_en：中英文通用。
ResTextFormat	是	Integer	识别结果返回样式： <ul style="list-style-type: none">0：基础识别结果（仅包含有效人声时间戳，无词粒度的详细识别结果）。1：基础识别结果之上，增加词粒度的详细识别结果（包含词级别时间戳、语速值，不含标点）。2：基础识别结果之上，增加词粒度的详细识别结果（包含词级别时间戳、语速值和标点）。
ChannelNum	是	Integer	当前只支持配置为 1。
SourceType	是	Integer	语音数据来源： <ul style="list-style-type: none">0：语音 URL。1：语音数据（post body）。
CallbackUrl	否	String	回调 URL，用户自行搭建的用于接收识别结果的服务 URL。回调格式和内容详见： 录音识别回调说明 。

⚠ 注意：

- 如果用户使用轮询方式获取识别结果，则无需提交该参数。

			<ul style="list-style-type: none"> 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。
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 12 小时，音频文件大小不能超过 1GB。
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。必须使用 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）。音频文件大小不能超过 5MB（Base64 编码后）。
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）。
FilterDirty	否	Integer	是否过滤脏词（中文普通话）： <ul style="list-style-type: none"> 0：不过滤脏词。 1：过滤脏词。 2：将脏词替换为 *。 默认值为 0。
FilterModal	否	Integer	是否过滤语气词（中文普通话）： <ul style="list-style-type: none"> 0：不过滤语气词。 1：部分过滤。 2：严格过滤。 默认值为 0。
FilterPunc	否	Integer	是否过滤标点符号（中文普通话）： <ul style="list-style-type: none"> 0：不过滤。 1：过滤句末标点。 默认值为 0。
ConvertNumMode	否	Integer	是否进行阿拉伯数字智能转换： <ul style="list-style-type: none"> 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。
HotwordList	否	String	临时热词表：该参数用于提升识别准确率。 <ul style="list-style-type: none"> 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，例如：“热词 5”或“ASR 11”。

- 临时热词表限制：多个热词用英文逗号分隔，最多支持 128 个热词，例如：“热词|10,语音识别|5,ASR|11”。

⚠ 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。
- 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”。

			<ul style="list-style-type: none"> 临时热词表限制：多个热词用英文逗号分隔，最多支持 128 个热词，例如：“热词 10,语音识别 5,ASR 11”。 <div style="border: 1px solid #00a0e3; padding: 10px; margin-top: 10px;"> <p>⚠ 注意：</p> <ul style="list-style-type: none"> 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。 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”。 </div>
language	否	String	暂不支持。

响应格式

响应消息为 json 格式，所有返回字段都位于顶层 Response 对象之内。

参数说明

参数名	类型	描述
Response	Object	顶层响应对象，包含以下字段。
Response.RequestId	String	请求的 RequestId，定位问题时需要提供该次请求的 RequestId。
Response.Data	Object	响应数据对象，包含以下字段。
Response.Data.RecTaskId	String	<p>该参数由服务端生成。录音文件识别的请求返回结果，包含结果查询需要的 RecTaskId。</p> <div style="border: 1px solid #00a0e3; padding: 10px; margin-top: 10px;"> <p>⚠ 注意：</p> <p>RecTaskId 有效期为 24 小时，不同日期可能出现重复 RecTaskId，请不要依赖 RecTaskId 作为您业务系统里的唯一 ID。</p> </div>

示例

用户通过 `SourceType=1` 请求录音文件识别，请求如下：

请求网址:

```
https://asr.cloud-rtc.com/v1/CreateRecTask?
AppId=1234***5&RequestId=7737***81dd-4b60-a08b-b34ed46243a7
```

输入参数

```
{
  "EngineModelType": "16k_zh_en",
  "ResTextFormat": 0,
  "ChannelNum": 1,
  "SourceType": 1,
  "Data": "xxxxxxxx",
  "DataLen": 8
}
```

响应结果

```
{
  "Response": {
    "RequestId": "24fe80e8-5b0c-4232-866f-4a402ee3127b",
    "Data": {
      "RecTaskId":
      "TT1Ie0V0voyHKBrWqqUZw57smgjSR3K1eazaxrmIuEZOeyw+xUgmJ+Y5eli2U9Y2dcpUF4wsM
      OqPjQAPxw8bvcUoWTON2M6wjcbZcHtrpt0Dm"
    }
  }
}
```

录音文件回调说明

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，使用 HTTP 的 POST 方法，将所有内容放入 Body 中，Content-Type 为 `application/x-www-form-urlencoded`。

回调 body 说明

回调 body 示例

```
code=0&requestId=DYz1ieNlIyb5nZT+R9kQsCAocpoDD6l1qvqam7XBk78PDQ..&appid=12516***&text=%5B0&audioTime=8.420000&message=&resultDetail=
```

回调参数说明

名称	类型	描述
code	Integer	任务状态码，0 为成功，其他：失败。
message	String	失败原因文字描述，成功时此值为空。
requestId	String	任务唯一标识，与录音识别请求中返回的 RecTaskId 一致。
appid	Integer	请求的 AppId。
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	包含详细识别结果，如创建任务时 ResTextFormat 为 0，则不包含该字段。
audioTime	Float	语音总时长（秒）。

详细识别结果

SentenceDetail

单句的详细识别结果，包含单个词的时间偏移

名称	类型	描述
FinalSentence	String	<p>单句最终识别结果。</p> <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;"> <p>⚠ 注意： 此字段可能返回 null，表示取不到有效值。</p> </div>
SliceSentence	String	单句中识别结果，使用空格拆分为多个词。

		<p>注意： 此字段可能返回 null，表示取不到有效值。</p>
StartMs	Integer	<p>单句开始时间（毫秒）。</p> <p>注意： 此字段可能返回 null，表示取不到有效值。</p>
EndMs	Integer	<p>单句结束时间（毫秒）。</p> <p>注意： 此字段可能返回 null，表示取不到有效值。</p>
WordsNum	Integer	<p>单句中词个数。</p> <p>注意： 此字段可能返回 null，表示取不到有效值。</p>
Words	Array of SentenceWords	<p>单句中词详情。</p> <p>注意： 此字段可能返回 null，表示取不到有效值。</p>
SpeechSpeed	Float	<p>单句语速，单位：字数/秒。</p> <p>注意： 此字段可能返回 null，表示取不到有效值。</p>

SentenceWords

识别结果中词文本，以及对应时间偏移

名称	类型	描述
Word	String	<p>词文本。</p> <p>注意：</p>

		此字段可能返回 null，表示取不到有效值。
StartTime	Integer	在句子中的开始时间偏移量。 ⚠ 注意： 此字段可能返回 null，表示取不到有效值。
EndTime	Integer	在句子中的结束时间偏移量。 ⚠ 注意： 此字段可能返回 null，表示取不到有效值。

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。

录音文件识别结果查询

接口描述

- 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。采用回调方式时详见录音文件回调说明

⚠ 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 RecTaskId 作为业务唯一 ID，不同日期可能出现重复 RecTaskId。

- 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"。
- 默认接口请求频率限制：50 次/秒。

内容	说明
请求协议	https 协议。
请求地址	https://域名/v1/DescribeTaskStatus?{请求参数}。
响应格式	JSON。
并发限制	默认单账号限制并发数为 50 次每秒。

请求格式

客户端发起 HTTP POST 请求，请求 URL 格式为：

正式环境：`https://asr.cloud-rtc.com/v1/DescribeTaskStatus?{请求参数}`

在请求 header 中填写以下字段：

key	必填	value 含义
X-TRTC-SdkAppId	是	TRTC 应用 ID，获取方式请参见 应用基本信息 。
X-TRTC-UserSig	是	TRTC 签名，计算及使用请参见 用户鉴权 ，其中的 UserID 等于 URL 参数中的 RequestId。

请求参数

请求参数格式为：`AppId=value1&RequestId=value2`

参数说明

参数名	必填	类型	描述
AppId	是	String	腾讯云注册账号的 AppId，可通过 API 密钥管理 获取。
RequestId	是	String	全局请求唯一 ID (UUID)，该参数用来生成请求 UserSig。

输入参数

录音文件查询接口的请求消息体格式为 json。

参数名	必填	类型	描述
RecTaskId	是	String	<p>从 CreateRecTask 接口获取的 RecTaskId，用于获取任务状态与结果。</p> <div style="border: 1px solid #00a88f; padding: 5px; margin-top: 10px;"> <p>⚠ 注意： RecTaskId 有效期为 24 小时，超过 24 小时的 RecTaskId 请不要再查询。</p> </div>

输出参数

录音文件查询接口请求消息体响应格式为 json，所有返回字段都位于顶层 Response 对象之内。

参数名	类型	描述
Response	Object	顶层响应对象。
Response.Data	TaskStatus	录音文件识别的请求返回结果。
Response.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。

TaskStatus 结构如下：

名称	类型	描述
RecTaskId	String	任务标识。RecTaskId 数据类型为 String。
Status	Integer	<p>任务状态码：</p> <ul style="list-style-type: none"> ● 0：任务等待。 ● 1：任务执行中。 ● 2：任务成功。 ● 3：任务失败。
StatusStr	String	<p>任务状态：</p> <ul style="list-style-type: none"> ● waiting：任务等待。 ● doing：任务执行中。 ● success：任务成功。 ● failed：任务失败。
Result	String	识别结果。

ErrorMsg	String	失败原因说明。
ResultDetail	Array of Sentence Detail	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 不为 0 时该字段不为空）。 <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;">注意： 此字段可能返回 null，表示取不到有效值。</div>
AudioDuration	Float	音频时长（秒）。 <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;">注意： 此字段可能返回 null，表示取不到有效值。</div>

查询示例

请求网址

```
https://asr.cloud-rtc.com/v1/DescribeTaskStatus?RequestId=2f5e8263-a23b-4e65-be22-6b7c19bd4064&AppId=12345
```

输入参数

```
{
  "RecTaskId": "DYz1ieN1Iyb5nZT+R9kQsCAocpoDD6l1qvqam7XBk78PDQ.."
}
```

响应结果

```
{
  "Response": {
    "RequestId": "2f5e8263-a23b-4e65-be22-6b7c19bd4064",
    "Data": {
      "RecTaskId":
      "DYz1ieN1Iyb5nZT+R9kQsCAocpoDD6l1qvqam7XBk78PDQ..",
      "Status": 2,
      "StatusStr": "success",
      "AudioDuration": 2.38,
      "Result": "[0:0.200,0:1.380] 你好。\\n",
      "ResultDetail": [
```

```
{
  "FinalSentence": "你好。",
  "SliceSentence": "你好",
  "StartMs": 200,
  "EndMs": 1380,
  "SpeechSpeed": 2,
  "WordsNum": 1,
  "Words": [
    {
      "Word": "你好",
      "StartTime": 200,
      "EndTime": 1380
    }
  ]
},
  "ErrorMsg": ""
}
```

语音合成

最近更新时间：2026-05-09 17:58:31

接口描述

腾讯云对话式 TTS 是基于 TRTC（实时音视频）打造的新一代低延迟语音合成能力，主打实时对话场景：超低延迟、拟人度高、支持声音克隆与多语种，能让 AI “听起来像真人”。访问更多实时音视频服务可前往 [TRTC 产品页](#)。

体验入口

内容	说明
全部音色体验页	输入文本即可在线合成，持续更新，含试听文件。

TTS 模型与音色说明

推荐模型：flow_02_turbo

内容	说明
定位	对话场景主推 TTS 模型。
首包延迟	低至 300ms，适合实时对话、实时配音。
拟人度	音质高、自然度高，具备口语化表达。
语言支持	中文、英文、日语、粤语。
完整精品音色清单	覆盖中文、英文、日语、粤语四种语言，支持角色配音、有声书、客服、播报等多种场景。

支持的语言代码（ISO 639-1）

当前推荐模型 `flow_02_turbo` 支持以下 4 种语言（实际支持语言以 [官方 API 文档](#) 为准，后续可能扩展）：

语言	代码
中文	<code>zh</code>
英文	<code>en</code>
日语	<code>ja</code>
粤语	<code>yue</code>

接口选择建议

场景	推荐接口	原因
已知完整文本，要求最低首包延迟	SSE 流式 TTS	一次请求边收边播，比非流式延迟低 50 ms - 100 ms。
离线播报、批量生成、文件导出	非流式 TTS	拿到完整音频文件（pcm/wav/mp3），方便后处理。
需要专属音色	声音克隆	提交样本得到 <code>VoiceId</code> ，再配合上述任意接口使用。

说明：

- **文本由 LLM 流式生成（增量文本）**：建议使用 SSE 流式接口，按句/段分段发送合成请求，每段独立起播，可实现“边想边说”的对话效果。
- **需要落盘做后处理（剪辑/合并/降噪）**：建议使用非流式接口，输出格式优先选择 wav（无损、兼容性好）或 mp3（体积小、适合分发），pcm 适合自定义处理流程。
- **兼顾低延迟与可导出文件**：可先通过 SSE 流式接口实时播放给用户，同时异步调用非流式接口生成完整音频文件用于归档或后处理。

SSE 流式文本转语音 API

本接口通过 Server-Sent Events (SSE) 进行流式语音合成，在对话场景下可边生成边播放，大幅降低首包延迟。比非流式延迟低 50 ms - 100 ms，适合对延迟有要求的实时对话场景。

文档详见 [SSE 流式文本转语音 API](#)，查看完整参数说明、请求/响应示例、错误码。

说明：

默认请求频率限制以 [官方文档](#) 为准，如您有更多 QPS 需求，请 [联系我们](#)。

文本转语音 API（非流式）

本接口提供非流式语音合成能力，适合对延迟不敏感的离线/异步生成场景，一次请求返回完整音频数据（pcm/wav/mp3），便于后处理与文件导出。

文档详见 [文本转语音 API（非流式）](#)，查看完整参数说明、请求/响应示例、错误码。

说明：

默认请求频率限制以 [官方文档](#) 为准，如您有更多 QPS 需求，请 [联系我们](#)。

声音克隆（音色快速复刻）

本接口用于提交少量语音样本（16k 单声道 wav，6 秒 - 180 秒）创建专属克隆音色。生成的 VoiceId 与精品音色 ID 用法一致，可在任意语音合成接口中直接使用。

文档详见 [声音克隆](#)，查看完整参数说明、请求/响应示例、错误码。

说明：

- 该声音克隆服务为限时免费，后续详情见计费说明。
- 用于合成特定音色时，您授权腾讯云使用您提供或上传的音频、人声信息等数据用于合成语音复刻；可随时请求删除已上传内容，删除后不会留存。

TTS 完整精品音色清单

共 77 条精品音色，覆盖中文、英文、日语、粤语四种语言，支持角色配音、有声书、客服、播报等多种场景。以下为完整清单。

中文音色

音色 ID	中文名	英文名	性别	语气	推荐场景
v-male-Bk7vD 3xP	威严 霸总	Commanding CEO - Male	Mal e	Authoritative , Firm	角色配音
v-female-R2s 4N9qJ	温柔 姐姐	Gentle Lady	Fe mal e	Gentle, Warm	角色配音
v-female-m1K pW7zE	傲娇 学姐	Tsundere Girl	Fe mal e	Playful, Proud	角色配音
v-female-U8a T2yLf	夹子 女生	Cutesy Girl	Fe mal e	Cutesy, Bright	角色配音
v-male-s5NqE 0rZ	闲聊 男声	Casual Man	Mal e	Relaxed, Friendly	角色配音
v-male-W1tH9 jVc	自然 男声	Natural Man	Mal e	Neutral, Calm	有声书， 客服
female-kefu- xiaomei	客服 小美	Customer Service - Xiaomei (Sweet Girl)	Fe mal e	Happy, Clear, Formal	播报，客 服

female-kefu-xiaoxin	客服 小心	Customer Service- Xiaoxin (Soft Female)	Female	Neutral, Clear, Formal	客服
female-kefu-xiaoyue	客服 小悦	Customer Service - Xiaoyue (Cheerful Female)	Female	Neutral, Clear, Formal	播报, 客服
male-kefu-xiaoxu	客服 小徐	Customer Service- Xiaoxu (Professional Male)	Male	Happy, Clear, Formal	播报, 客服
v-female-S6n2JxR5	客服 右琪	Youqi - Customer Service	Female	Happy, Clear, Formal	播报, 客服
v-female-S6p4LxQ8	客服 小羊	Xiaoyang - Customer Service	Female	Happy, Clear, Formal	播报, 客服
v-female-H6p3LxP8	客服 小丁	Xiaoding - Customer Service	Female	Happy, Clear, Formal	播报, 客服
v-male-S6m3LxP8	客服 小柒	Xiaoqi - Customer Service	Male	Neutral, Clear, Formal	播报, 客服

英文音色

音色 ID	中文名	英文名	性别	语气	推荐场景
v-female-A7c9QmP2	温柔女老师	Wenrou - Soft Female Teacher	Female	Warm, Patient, Encouraging	中英教育, 情感配音
v-male-X6h4TvP9	暖心男老师	Nuanxin - Warm Male Teacher	Male	Warm, Formal, Encouraging	美音教育, 情感配音
v-male-R3p7LdW8	阳光男老师	Yangguang - Uplifting Male Teacher	Male	Calm, Confident, Encouraging	美音教育, 情感配音
v-male-K8s2QmJ4	权威男老师	Quanwei - Authoritative Male Teacher	Male	Confident, Authoritative, Precise	美音教育, 情感配音

v-male-U5n9 DwC7	理性男 老师	Lixing - Formal Male Teacher	Mal e	Clear, Authoritative, Steady	美音教育, 情 感配音
v-male-J1r4 ZvH6	真诚男 老师	Zhcheng - Sincere Male Teacher	Mal e	Clear, Gentle, Sincere	英音教育, 情 感配音
v-female-P6 q9LmR2	温情女 老师	Wenqing - Warm Female Teacher	Fe mal e	Warm, Caring, Patient	美音教育, 情 感配音
v-female-H7 m4QpL8	理性女 老师	Lixing - Formal Female Teacher	Fe mal e	Caring, Steady, Clear	美音教育, 情 感配音
v-female-N4 s7VkJ3	活泼女 老师	Huopo - Lively Female Teacher	Fe mal e	Warm, Cute, Happy	美音教育, 情 感配音
v-female-T3 c8WdP6	专业女 老师	Zhuanye - Professional Female Teacher	Fe mal e	Calm, Precise, Measured	英音教育, 情 感配音
v-female-R9 k2LmV7	自信女 老师	Zixin - Confident Female Teacher	Fe mal e	Confident, Precise, Measured	英音教育, 情 感配音
v-female-p9 Xy7Q1L	清晰女 旁白	Articulate Narrator-Female	Fe mal e	Neutral, Steady, Clear	有声书, 配音
v-female-Z3 x9LmQ2	理性女 讲解	Analytical Presenter-Female	Fe mal e	Neutral, Steady, Clear	有声书, 配音
v-male-A4b9 KqP2	严谨男 讲师	Scholarly Lecturer-Male	Mal e	Neutral, Steady, Precise	有声书, 配音
v-male-r7K2 pQ9L	权威男 解读	Expert Analyst- Male	Mal e	Neutral, Authoritative, Precise	有声书, 配音
v-male-Q6p8 ZxL3	沉着男 评审	Calm Reviewer- Male	Mal e	Neutral, Authoritative, Precise	有声书, 配音

v-female-T3 s8BqL9	静心女 教练	Mindfulness Coach – Female	Fe mal e	Calm, Soothing, Supportive	有声书, 配音
v-male-P6q7 LzD8	温和男 顾问	Gentle Mentor– Male	Mal e	Calm, Supportive, Clear	有声书, 配音
v-female-M7 k2PxL9	内敛女 播音	Reserved Broadcaster– Female	Fe mal e	Neutral, Clear, Restrained	有声书, 配音
v-female-S5 n9QxJ4	淡然女 配音	Serene Voice Actress	Fe mal e	Calm, Reflective, Reserved	有声书, 配音
v-female-T8 m4WxP7	沉稳女 配音	Composed Voice Actress	Fe mal e	Calm, Neutral, Monotone	有声书, 配音
v-male-D6p3 KxN8	深沉男 评析	Resonant Reviewer – Male	Mal e	Deep, Calm, Introspective	有声书, 配音
v-female-A9 b3KfL2	温情女 主持	Empathic Host– Female	Fe mal e	Warm, Caring, Encouraging	有声书, 配音
v-female-A7 h2MxQ5	真挚女 创作	Sincere Storyteller– Female	Fe mal e	Warm, Gentle, Sincere	有声书, 配音
v-male-G4n7 RxM3	温和男 创作	Gentle Storyteller–Male	Mal e	Warm, Gentle, Affectionate	有声书, 配音
v-male-H3p9 LxK7	暖心男 顾问	Caring Counselor – Male	Mal e	Soft, Reassuring, Compassiona te	有声书, 配音
v-male-R6n2 MxT9	真诚男 主播	Sincere Streamer–Male	Mal e	Warm, Gentle, Supportive	有声书, 配音
v-female-C8 k4NxL6	自信女 演员	Confident Actress	Fe mal e	Gentle, Confident, Reflective	有声书, 配音

v-male-L7m5 QxP4	阳光男 演讲	Uplifting Speaker - Male	Male	Calm, Confident, Encouraging	有声书, 配音
v-male-N4k8 TxR7	理性男 评论	Rational Commentator - Male	Male	Measured, Neutral, Definitive	有声书, 配音
v-female-B7 k5WxN4	理智女 旁白	Intellectual Narrator - Female	Female	Calm, Neutral, Straightforward	有声书, 配音

日语音色

音色 ID	中文名	英文名	性别	语气	推荐场景
v-female-J3 k7NxR2	标准女 播音	Female Broadcaster	Female	Neutral, Measured, Business-like	有声书, 配音
v-female-W6 n8KxL5	专业女 配音	Professional Voice Actress	Female	Neutral, Professional, Deliberate	有声书, 配音
v-female-U4 k9TxM3	沉稳女 配音	Composed Voice Actress	Female	Calm, Neutral, Declarative	有声书, 配音
v-female-S2 k7NxP9	严肃女 播音	Serious Broadcaster - Female	Female	Neutral, Formal, Declarative	有声书, 配音
v-female-Y5 n2KxR8	温和女 旁白	Soft Narrator - Female	Female	Gentle, Clear, Steady	有声书, 配音
v-male-J3n8 DxK2	沉稳男 旁白	Composed Narrator-Male	Male	Calm, Neutral, Formal	有声书, 配音
v-male-M7x2 QrJ5	沉稳男 配音	Chenwen - Calm Male Voice Actor	Male	Precise, Steady, Friendly	播报, 配 音
v-male-F9c6 LhY1	庄重男 配音	Zhuangzhong - Solemn Male Voice Actor	Male	Precise, Deliberate, Formal	播报, 配 音

v-female-B5 q9NvY6	温和女 配音	Wenhe - Gentle Female Voice Actor	Fe mal e	Calm, Warm, Measured	播报, 配 音
v-female-D2 w6HcL4	专业女 旁白	Zhuanye - Professional Female Narrator	Fe mal e	Precise, Formal, Professional	播报, 配 音
v-female-J8 m9NxT2	温柔少 女	Gentle Girl	Fe mal e	Gentle, Warm, Inquisitive	有声书, 配音
v-female-R7 k3MxW5	温柔女 讲师	Gentle Lecturer- Female	Fe mal e	Gentle, Regretful, Vulnerable	有声书, 配音
v-female-H6 p2LxQ8	俏皮女 主播	Playful Streamer- Female	Fe mal e	Playful, Curious, Lighthearted	有声书, 配音
v-male-T4n8 KxM3	沉稳男 主持	Composed Host - Male	Mal e	Calm, Steady, Friendly	有声书, 配音
v-male-M5k7 NxP9	平和男 旁白	Soothing Narrator - Male	Mal e	Calm, Conversational, Reflective	有声书, 配音
v-female-P3 n6LxW4	专业女 讲师	Professional Lecturer - Female	Fe mal e	Precise, Formal, Professional	有声书, 配音
v-male-C9m4 QxL7	庄重男 播音	Formal Broadcaster- Male	Mal e	Calm, Deliberate, Formal	有声书, 配音
v-female-E8 k2NxV6	温和女 讲师	Gentle Lecturer - Female	Fe mal e	Calm, Gentle, Measured	有声书, 配音
v-male-D5k8 MxN2	专业男 讲师	Professional Lecturer - Male	Mal e	Calm, Neutral, Measured	有声书, 配音
v-male-S6n3 KxW9	庄重男 解说	Formal Narrator - Male	Mal e	Formal, Measured, Neutral	有声书, 配音
v-male-L4k9 NxT7	优雅男 讲师	Elegant Lecturer - Male	Mal e	Precise, Neutral, Formal	有声书, 配音

v-male-H5k7 NxM3	儒雅男 播音	Refined Broadcaster- Male	Mal e	Deliberate, Formal, Neutral	有声书, 配音
v-male-B6k3 NxQ9	深沉男 讲师	Deep Voice Lecturer-Male	Mal e	Deliberate, Subdued, Contemplative	有声书, 配音
v-female-K9 m5NxL4	中性女 播音	Neutral Broadcaster - Female	Fe mal e	Neutral, Slightly Formal	有声书, 配音

粤语音色

音色 ID	中文名	英文名	性别	语气	推荐场景
v-female-k3 P8sL0Q	雅致女 解说	Elegant Narrator- Female	Fe mal e	Calm, Neutral, Clear	播报, 配音
v-male-L4s7 PqZ9	沉稳男 解说	Composed Narrator-Male	Mal e	Calm, Neutral, Clear	播报, 配音
v-female-S8 p3JwK6	温和女 播音	Wenhe - Calm Female Announcer	Fe mal e	Neutral, Direct, Calm	有声书, 配 音
v-female-N7 c3LpV5	理性女 向导	Lixing - Precise Female Guide	Fe mal e	Neutral, Precise, Calm	播报, 配音
v-male-P8r6 MdQ3	儒雅男 讲员	Ruya - Polite Male Presenter	Mal e	Neutral, Polite, Inquisitive	有声书, 配 音
v-male-Q2f7 RdP6	稳重男 向导	Wenzhong - Steady Male Guide	Mal e	Neutral, Calm, Clear	有声书, 配 音
v-female-C5 t1QxH9	自信女 讲师	Zixin - Confident Female Instructor	Fe mal e	Natural, Precise, Formal	播报, 配音
v-male-D7p4 XcL2	自然男 播音	Ziran - Natural Male Announcer	Mal e	Natural, Clear, Formal	播报, 配音

说明:

如您有更多其他语言需求，请 [联系我们](#)。

完整示例代码

[官方示例仓库](#) 提供 Python / Node.js 双语言、覆盖以上三种接口的完整示例，建议直接基于示例进行二次集成：

相关服务协议告知

上述 TTS 服务全部基于 TRTC 能力，开通过程会创建 TRTC SDKAppId，相关隐私保护与合规受以下条款约束，使用相关服务即代表您同意相关规定：

- [实时音视频 TRTC SDK 个人信息保护规则](#)
- [实时音视频 TRTC SDK 合规使用指南](#)
- [腾讯云服务协议](#)

使用语音合成（TTS）和声音快速复刻（克隆）功能时，您需要确保：

- **输入素材合法合规**，不含有任何违反法律法规之内容，不应输入未经授权的自然人个人信息、未经授权的知识产权数据、商业秘密数据等。
- **拥有合法授权**：输入素材为您自行所有或已获取对应权利人充分、必要且有效的合法许可及授权，不侵犯任何第三方合法权益（包括但不限于知识产权、肖像权、名誉权、隐私权、商业秘密等）；否则因此产生的违约、侵权纠纷或法律责任由您自行处理与承担。