

语音合成

在线 SDK 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

在线 SDK 文档

SDK 更新说明

Android SDK

基础语音合成

实时语音合成

流式文本语音合成

iOS SDK

基础语音合成

实时语音合成

流式文本语音合成

HarmonyOS NEXT

基础语音合成

实时语音合成

流式文本语音合成

FLUTTER

基础语音合成

在线 SDK 文档

SDK 更新说明

最近更新时间：2025-07-24 15:03:32

SDK 版本说明

腾讯云语音合成 SDK（Android）

版本：V2.0.14

更新时间：2025-06-30

SDK 介绍：腾讯云语音合成客户端 SDK，支持基础语音合成、流式语音合成、实时语音合成。

更新日志：

版本	更新内容
2.0.14	优化工程
2.0.13	新增实时语音合成
2.0.12	新增流式文本语音合成
2.0.11	优化合成回调异常问题
2.0	新版 SDK 在稳定性、功能健全性、接口自由度等方面均有所优化，推荐使用
1.5.3	为了让您获得更好的体验，推荐使用 V2.0 版本

服务提供方：腾讯云计算（北京）有限责任公司

合规使用说明：《[语音合成 SDK 合规使用指南](#)》

个人信息处理规则：《[语音合成 SDK 个人信息保护规则](#)》

腾讯云语音合成 SDK（iOS）

版本：V2.0.16

更新时间：2025-06-30

SDK 介绍：腾讯云语音合成客户端 SDK，支持基础语音合成、流式语音合成、实时语音合成。

更新日志：

版本	更新内容
2.0.16	优化合成效率
2.0.14	解决 WebSocket 类冲突问题

2.0.12	新增实时语音合成
2.0.11	新增流式文本语音合成
2.0.10	- 优化字级别时间戳不准确问题 - 修复快速切换无法正常播放问题
2.0	新版 SDK 在稳定性、功能健全性、接口自由度等方面均有所优化，推荐使用
1.5.1	为了让您获得更好的体验，推荐使用 V2.0 版本

服务提供方：腾讯云计算（北京）有限责任公司

合规使用说明：[《语音合成 SDK 合规使用指南》](#)

个人信息处理规则：[《语音合成 SDK 个人信息保护规则》](#)

腾讯云语音合成 SDK（Flutter）

版本：V0.0.6

更新时间：2025-04-08

SDK 介绍：腾讯云语音合成客户端 SDK，支持基础语音合成。

更新日志：

版本	更新内容
0.0.6	修复 Android 平台偶现引擎报错
0.0.3	- 增加报错信息 - 修复 iOS 回调失败问题

服务提供方：腾讯云计算（北京）有限责任公司

合规使用说明：[《语音合成 SDK 合规使用指南》](#)

个人信息处理规则：[《语音合成 SDK 个人信息保护规则》](#)

腾讯云语音合成 SDK（HarmonyOS NEXT）

版本：V1.3.1

更新时间：2025-06-30

SDK 介绍：腾讯云语音合成客户端，SDK 支持基础语音合成、流式语音合成、实时语音合成。

更新日志：

版本	更新内容
1.3.1	优化工程

1.3.0	demo 增加长文本基础语音合成示例代码
1.2.0	新增实时语音合成 SDK
1.1.0	新增流式文本语音合成 SDK
1.0.0	新增基础语音合成 SDK

服务提供方：腾讯云计算（北京）有限责任公司

合规使用说明：[《语音合成 SDK 合规使用指南》](#)

个人信息处理规则：[《语音合成 SDK 个人信息保护规则》](#)

腾讯云离在线语音合成 SDK（Android）

版本：V2.0.14

更新时间：2025-06-30

SDK 介绍：腾讯云离在线语音合成客户端 SDK，支持离线模式和在线模式的基础语音合成、流式语音合成、实时语音合成。

更新日志：

版本	更新内容
2.0.14	修复了一些已知问题
2.0.13	新增实时语音合成基础功能
2.0.12	离线增加流式数据返回接口

服务提供方：腾讯云计算（北京）有限责任公司

合规使用说明：[《语音合成 SDK 合规使用指南》](#)

个人信息处理规则：[《语音合成 SDK 个人信息保护规则》](#)

腾讯云离在线语音合成 SDK（iOS）

版本：V2.0.16

更新时间：2025-06-30

SDK 介绍：腾讯云离在线语音合成客户端 SDK，支持离线模式和在线模式的基础语音合成、流式语音合成、实时语音合成。

更新日志：

版本	更新内容
2.0.16	优化合成流程
2.0.14	优化日志组件以及网络组件

2.0.13	解决 WebSocket 类冲突问题
2.0.12	解决 WebSocket 类冲突问题

服务提供方：腾讯云计算（北京）有限责任公司

合规使用说明：[《语音合成 SDK 合规使用指南》](#)

个人信息处理规则：[《语音合成 SDK 个人信息保护规则》](#)

Android SDK

基础语音合成

最近更新时间：2024-11-07 11:17:42

❗ 说明

- 当前页面为新版（V2.0.0及以上）SDK 开发文档。新客户可直接按当前文档接入新版 SDK。
- 旧版（V1.5.3版本及以下）开发文档已于2022年9月5日下线。正在使用旧版 SDK 的客户，可前往 [控制台](#) 查看开发文档。
- 新版 SDK 在稳定性、功能健全性、接口自由度等方面都有所优化。我们将继续支持旧版（V1.5.3版本及以下）SDK，但建议正在使用旧版 SDK 的客户及时升级到新版，以获取更好的使用体验。
- Android SDK 支持模拟器。

开发相关

开发准备

- 支持 Android 4.1 以上版本 API LEVEL 16，支持手机与平板。
- 合成实时流式语音，需要手机能够连接网络（3/4/5G 或 Wi-Fi 网络等）。
- 建议使用最新版本 Android Studio 进行开发。
- 服务端 [API 文档](#)。

下载安装 SDK

- 语音合成 [Android SDK](#)。
- 解压后即是示例代码工程，目录 `sdk` 下的 aar 文件即 SDK 包。
- 用 Android Studio 打开此工程查看语音合成示例代码。

环境配置

- 添加实时语音识别 SDK aar 包 放在 libs 目录下，在 App 的 build.gradle 文件中添加以下代码。

```
implementation(name: 'libqcloudtts-release', ext: 'aar')
```

- 在 AndroidManifest.xml 添加如下权限：

```
< uses-permission android:name="android.permission.INTERNET"/>
```



```
< uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />
```

接口说明

获得 TTS 合成器实例

```
//获得实例  
TtsController mTtsController = TtsController.getInstance();  
  
//销毁实例  
TtsController.release();
```

初始化引擎

```
//第二个参数固定传入 TtsMode.ONLINE  
//TtsResultListener: 合成监听器，用于获取合成结果  
mTtsController.init(Context context, TtsMode.ONLINE, TtsResultListener  
    listener)
```

合成监听器，用于获取合成结果

实例化 *TtsResultListener* 时，默认需要重写 *onSynthesizeData()* 和 *onError()* 方法。

onSynthesizeData()方法签名说明

参数	说明
byte[] bytes	语音数据
String utteranceId	语句 ID
String text	文本
int engineType	引擎类型；0：在线，1：离线；当前是纯在线 SDK，请忽略此参数

onError()方法签名说明

参数	说明
TtsError error	错误信息，无错误返回 null
String	语句 ID（如果有则返回）

utteranceId	
String text	文本（如果有则返回）

示例

```
TtsResultListener listener = new TtsResultListener() {

    @Override
    public void onSynthesizeData(byte[] bytes, String utteranceId,
String text, int engineType) {
        // 您可以在这里将音频保存或者送入播放接口播放，可调用播放器入参接口入参
    }

    @Override
    public void onError(TtsError error, String utteranceId, String text)
    {
        // 您可以在这里添加错误后处理
    }

    @Override
    public void onOfflineAuthInfo(QCloudOfflineAuthInfo offlineAuthInfo)
    {
        // 离在线SDK保留接口，请忽略，如果您后续升级为离线SDK或者离在线SDK，此接口将用于返回授权信息
    }
}
```

合成文本入参接口

接口	说明
synthesize(String text, String utteranceId)	text 为需要合成的文本；utteranceId 为标记该文本的 ID，将随合成结果返回宿主层
synthesize(String text)	text 为需要合成的文本

示例

```
//内部有维护队列，可持续添加语句，SDK内将依次合成
TtsError error = null;
//当返回的error不为null时，入参失败
error = mTtsController.synthesize("今天天气不错", "第1句");
```

```
error = mTtsController.synthesize("腾讯云语音合成", "第2句");
error = mTtsController.synthesize("腾讯云AI", "第3句");
error = mTtsController.synthesize("腾讯云AI", "第4句");

//取消未合成的任务并清空内部队列
mTtsController.cancel();
```

TtsController 配置参数方法

接口	说明
setSecretId(String s)	腾讯云安全凭证, 获取地址
setSecretKey(String s)	腾讯云安全凭证, 获取地址同上
setToken(String s)	若 STS 临时证书鉴权时需要设置 Token, 请参见 获取联合身份临时访问凭证
setOnlineVoiceSpeed(float f)	语速, 范围: [-2, 6], 分别对应不同语速: -2代表0.6倍、-1代表0.8倍、0代表1.0倍 (默认)、1代表1.2倍、2代表1.5倍、6代表2.5倍。 如果需要更细化的语速, 可以保留小数点后 2 位, 例如0.5/1.25/2.81等。 参数值与实际语速转换, 可参考 代码示例
setOnlineVoiceVolume(float f)	设置在线所合成音频的音量
setOnlineVoiceType(int i)	设置在线所合成音频的音色 ID, 完整的音色 ID 列表请参见 基础语音合成
setOnlineVoiceLanguage(int i)	主语言类型: 1-中文 (默认), 2-英文
setOnlineCodec(String s)	在线模式编码格式, 非业务必要不建议更改: 默认 mp3, 目前支持 mp3、wav、pcm, 如更改为 pcm 不支持播放
setConnectTimeout(int i)	连接超时, 范围: [500,30000], 单位ms, 默认15000ms
setReadTimeout(int i)	读取超时, 范围: [2200,60000], 单位ms, 默认30000ms

示例

```
mTtsController.setSecretId("*****");
```

```
mTtsController.setSecretKey("*****");
mTtsController.setOnlineVoiceSpeed(0.0); //配置语速
mTtsController.setOnlineVoiceVolume(1.0); //配置音量
mTtsController.setOnlineVoiceType(1001); //配置音色id
mTtsController.setOnlineVoiceLanguage(1); //配置主语言
mTtsController.setOnlineCodec("mp3"); //配置合成格式
mTtsController.setConnectTimeout(15 * 1000); //连接超时时间
mTtsController.setReadTimeout(30 * 1000); //读取超时时间
```

播放接口

初始化播放器

如果 SDK 的内置播放器无法满足您的需求，您也可以使用自己实现的播放器替换。demo 中也额外提供了一份播放器源码，您可以修改播放器逻辑，源代码位于 MediaPlayerDemo.java，与 SDK 内置播放器一致。

```
//使用SDK中提供的播放器
QCloudMediaPlayer mediaPlayer = new QCloudMediaPlayer(new
QCloudPlayerCallback() {

    @Override
    public void onTTSPPlayStart() {
        Log.d(TAG, "开始播放");
    }

    @Override
    public void onTTSPPlayWait() {
        Log.d(TAG, "播放完成，等待音频数据");
    }

    @Override
    public void onTTSPPlayResume() {
        Log.d(TAG, "恢复播放");
    }

    @Override
    public void onTTSPPlayPause() {
        Log.d(TAG, "暂停播放");
    }

    @Override
    public void onTTSPPlayNext(String text, String utteranceId) {
        Log.d(TAG, "开始播放: " + utteranceId + "|" + text);
    }
});
```

```

    }

    @Override
    public void onTTSPPlayStop() {
        Log.d(TAG, "播放停止, 内部队列已清空");
    }

    @Override
    public void onTTSPPlayError(QPlayerError error) {
        Log.d(TAG, "播放器发生异常:" + error.getmCode() + ":" +
error.getmMessage());
    }

    /**
     * @param currentWord 当前播放的字符（此为预估值）
     * @param currentIndex 当前播放的字符在所在的句子中的下标（此为预估值）
     */
    @Override
    public void onTTSPPlayProgress(String currentWord, int currentIndex)
    {
        Log.d(TAG, "onTTSPPlayProgress: " + currentWord + "|" +
currentIndex);
    }
});

```

播放器入参

enqueue()方法签名说明

参数	说明
byte[] bytes	返回音频流，通过传入字节数组播放
File audio	返回音频文件，通过传入文件播放
String text	音频对应的文本
String utteranceId	文本 ID

示例

```
// 通过音频数据入参
```

```
QPlayerError err = mediaPlayer.enqueue(byte[] bytes,String text,String
utteranceId);
```

//通过音频文件入参

```
QPlayerError err = mediaPlayer.enqueue(File audio,String text,String
utteranceId);
```

暂停、恢复或停止播放

```
mediaPlayer.PausePlay();
mediaPlayer.ResumePlay();
mediaPlayer.StopPlay();
```

客户端错误码

ID	错误码	说明
-100	TTS_ERROR_CODE_UNINITIALIZED	SDK 未初始化
-101	TTS_ERROR_CODE_GENERATE_SIGN_FAIL	签名失败
-102	TTS_ERROR_CODE_NETWORK_CONNECT_FAILED	网络异常
-103	TTS_ERROR_CODE_DECODE_FAIL	Response 解析失败
-104	TTS_ERROR_CODE_SERVER_RESPONSE_ERROR	后端返回失败错误码，详细错误信息请查看后端错误码
-105	TTS_ERROR_CODE_QUEUE_IS_FULL	合成队列已满
-106	TTS_ERROR_CODE_CANCEL_FAILURE	取消失败，请稍后重试
-900	TTS_ERROR_CODE_OFFLINE_NO_SUPPORT	当前 SDK 不支持离线合成能力，请使用 TtsMode.ONLINE

服务端错误码

请参考 [语音合成 API 文档](#)。

实时语音合成

最近更新时间：2025-08-21 15:06:12

1. 接入准备

1.1 SDK 获取

实时 TTS Android SDK 以及 Demo 的下载地址：[接入 SDK 下载](#)。

1.2 接入须知

- 该 SDK 需要手机能够连接网络，且 Android 设备 API Level 为16及以上版本。
- 运行 Demo 必须设置 AppID、SecretID、SecretKey，可在 [API 密钥管理](#) 中获取。

2. SDK 集成说明

2.1 添加 SDK

将下载的 SDK 文件复制到项目的 libs 文件夹中，并在项目的 build.gradle 文件中添加以下代码：

```
dependencies {  
    implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')  
    // 实时TTS SDK  
    implementation files("libs/realtime_tts-release-VERSION.aar")  
    // 实时TTS SDK内部依赖的okhttp库（必选）  
    implementation 'com.squareup.okhttp3:okhttp:4.9.3'  
    // 实时TTS SDK内部依赖的gson库（必选）  
    implementation 'com.google.code.gson:gson:2.8.9'  
}
```

2.2 添加权限

在项目的 AndroidManifest.xml 文件中添加 SDK 需要的权限，例如：

```
<uses-permission android:name="android.permission.INTERNET" />
```

3. SDK 接口使用说明

本章节主要介绍 SDK 接口调用详细步骤以及接口调用时机（具体使用方法可参考 SDK 交付 Demo 内的示例代码）。

3.1 构造配置项

```
RealTimeSpeechSynthesizerRequest request = new
RealTimeSpeechSynthesizerRequest();
/***** 配置项含义也可参考官网文档:
https://cloud.tencent.com/document/product/1073/94308 *****/
request.setVolume(volume); // 音量大小, 范围[-10, 10], 对应音量大小。默认为0,
代表正常音量, 值越大音量越高。
request.setSpeed(speed); // 语速, 范围: [-2, 6], 分别对应不同语速: -2: 代表0.6
倍; -1: 代表0.8倍; 0: 代表1.0倍(默认); 1: 代表1.2倍; 2: 代表1.5倍; 6: 代表2.5
倍
request.setCodec("pcm"); // 返回音频格式: pcm: 返回二进制pcm音频(默认); mp3:
返回二进制mp3音频
request.setSampleRate(SAMPLE_RATE); // 音频采样率: 24000: 24k(部分音色支持);
16000: 16k(默认); 8000: 8k
request.setVoiceType(voiceType); // 音色ID
request.setEnableSubtitle(true); // 是否开启时间戳功能, 默认为false。
request.setEmotionCategory("neutral");// 控制合成音频的情感, 仅支持多情感音色使
用
request.setEmotionIntensity(100); // 控制合成音频情感程度, 取值范围为
[50,200], 默认为 100; 只有 EmotionCategory 不为空时生效。
request.setSessionId(UUID.randomUUID().toString());//sessionId, 需要保持全
局唯一(推荐使用 uuid), 遇到问题需要提供该值方便服务端排查
request.setText("你好"); //合成语音的源文本, 按UTF-8编码统一计算。中文最大支持
600 个汉字(全角标点符号算一个汉字); 英文最大支持 1800 个字母(半角标点符号算一个字
母)
request.set(String_key, Object_value);//设置自定义扩展参数(可选, 非必须设置)
/***** 配置项含义也可参考官网文档:
https://cloud.tencent.com/document/product/1073/94308 *****/
```

3.2 账号信息配置

```
// 账号信息获取可参考 0.2接入须知
Credential credential = new Credential(appId, secretId, secretKey,
token);
```

3.3 构造合成器

```
// 网络连接的代理, 全局唯一即可
private static final SpeechClient proxy = new SpeechClient();
...
RealTimeSpeechSynthesizerListener listener = new
RealTimeSpeechSynthesizerListener() {
```



```
@Override
public void onSynthesisStart(SpeechSynthesizerResponse response) { //合成开始}

@Override
public void onSynthesisEnd(SpeechSynthesizerResponse response) { //合成结束}

@Override
public void onAudioResult(ByteBuffer buffer) { //合成的音频数据}

@Override
public void onTextResult(SpeechSynthesizerResponse response) { //合成文本的信息}

@Override
public void onSynthesisCancel() { //合成取消}

@Override
public void onSynthesisFail(SpeechSynthesizerResponse response) { //合成失败}
};
// 构造合成器
RealTimeSpeechSynthesizer synthesizer = new
RealTimeSpeechSynthesizer(proxy, credential, request, listener);
```

3.4 开始合成

```
synthesizer.start();
```

3.5 取消合成

```
// 会直接断开websocket链接
synthesizer.cancel();
```

4. 接口详情

4.1 RealTimeSpeechSynthesizer

4.1.1 构造接口

```
public RealTimeSpeechSynthesizer(SpeechClient client,
```

```
        Credential credential,  
        RealTimeSpeechSynthesizerRequest  
request,  
        RealTimeSpeechSynthesizerListener  
listener) throws SynthesizerException
```

功能：实时 TTS SDK 的核心类的构造接口，注意异常捕获。

参数：

参数类型	参数名称	参数含义
SpeechClient	client	网络连接的代理，全局唯一即可
Credential	credential	鉴权相关信息的实体类
RealTimeSpeechSynthesizerRequest	request	语音合成的配置项
RealTimeSpeechSynthesizerListener	listener	语音合成的关键事件回调

4.1.2 启动合成器接口

```
public void start() throws Exception
```

功能：启动合成器，注意异常捕获。

4.1.3 取消合成接口

```
public void cancel()
```

功能：取消合成，直接关闭 WebSocket 连接，不向服务端发送结束通知。

4.1.4 SDK版本号

```
public static String version()
```

4.2 RealTimeSpeechSynthesizerRequest

SDK 配置类，配置项含义请参见 [实时语音合成](#)。

类型	名称	含义	默认值
----	----	----	-----

String	text	合成语音的源文本，按 UTF-8 编码统一计算。中文最大支持 600 个汉字（全角标点符号算一个汉字）；英文最大支持 1800 个字母（半角标点符号算一个字母）	""
Float	volume	音量大小，范围[-10, 10]，对应音量大小。默认为0，代表正常音量，值越大音量越高。	0
Float	speed	语速，范围：[-2, 6]，分别对应不同语速： -2：代表0.6倍 -1：代表0.8倍 0：代表1.0倍（默认） 1：代表1.2倍 2：代表1.5倍 6：代表2.5倍	0
String	codec	返回音频格式： - pcm：返回二进制 pcm 音频（默认） - mp3：返回二进制 mp3 音频	pcm
Integer	sampleRate	音频采样率： 24000：24k（部分音色支持） 16000：16k（默认） 8000：8k	16000
Integer	voiceType	音色 ID，取值可参见 基础语音合成音色列表 。	0
Boolean	enableSubtitle	是否开启时间戳功能，默认为 false。	false
String	emotionCategory	控制合成音频的情感，仅支持多情感音色使用。取值：neutral(中性)、sad(悲伤)、happy(高兴)、angry(生气)、fear(恐惧)、news(新闻)、story(故事)、radio(广播)、poetry(诗歌)、call(客服)、撒娇(sajiao)、厌恶(disgusted)、震惊(amaze)、平静(peaceful)、兴奋(exciting)、傲娇(aojiao)、解说(jieshuo)	""
Integer	emotionIntensity	控制合成音频情感程度，取值范围为 [50,200]，默认为 100；只有 EmotionCategory 不为空时生效。	100
String	sessionId	需要保持全局唯一（推荐使用 UUID），遇到问题需要提供该值方便服务端排查	""
String	fastVoiceType	一句话版声音复刻音色 ID，使用一句话版声音复刻音色时需填写。	""

4.3 Credential

密钥信息实体类，可在 [API 密钥管理](#) 中获取。

类型	名称	含义	默认值
String	appid	appid	""
String	secretId	secretId，在控制台申请。	""
String	secretKey	secretKey，在控制台申请。	""
String	token	用于临时授权场景。	""

4.4 RealTimeSpeechSynthesizerListener

合成监听类

```
public abstract class RealTimeSpeechSynthesizerListener {  
    //合成开始  
    public abstract void onSynthesisStart(SpeechSynthesizerResponse  
response);  
    //合成结束  
    public abstract void onSynthesisEnd(SpeechSynthesizerResponse  
response);  
    //合成的音频数据  
    public abstract void onAudioResult(ByteBuffer data);  
    //合成文本的信息  
    public abstract void onTextResult(SpeechSynthesizerResponse  
response);  
    //合成取消  
    public abstract void onSynthesisCancel();  
    //合成失败  
    public abstract void onSynthesisFail(SpeechSynthesizerResponse  
response);  
}
```

4.5 SpeechSynthesizerResponse

合成 Response

类型	名称	含义
String	sessionId	由客户端在握手阶段生成并赋值在调用参数中。

Integer	code	状态码，0代表正常，非0值表示发生错误。
Integer	end	该字段返回1时表示文本全部合成结束，客户端收到后需主动关闭 WebSocket 连接。
String	message	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。
String	requestId	音频流唯一 ID，由服务端在握手阶段自动生成。
String	message Id	本 message 唯一 ID。
SpeechSynthesizerResult	result	语音合成文本结果。

4.6 SpeechSynthesizerResult

合成结果

类型	名称	含义
SpeechSynthesizerSubtitle[]	subtitles	词列表信息。

4.7 SpeechSynthesizerSubtitle

合成词信息

类型	名称	含义
String	text	文本信息
Integer	beginTime	文本对应 TTS 语音开始时间戳。
Integer	endTime	文本对应 TTS 语音结束时间戳。
Integer	beginIndex	该字在整句中的开始位置。
Integer	endIndex	该字在整句中的结束位置。
String	phoneme	该字的音素。

5. 错误码

错误码

CLIENT_CANNOT_BE_NULL(-500, "client cannot be null")
CREDENTIAL_CANNOT_BE_NULL(-501, "credential cannot be null")
REQUEST_CANNOT_BE_NULL(-502, "request cannot be null")
LISTENER_CANNOT_BE_NULL(-503, "listener cannot be null")
APPID_IS_EMPTY(-504, "appid cannot be empty")
SECRETID_IS_EMPTY(-505, "secretId cannot be empty")
SECRETKEY_IS_EMPTY(-506, "secretKey cannot be empty")
START_SYNTHESIZER_FAIL(-507, "fail to start synthesizer")
SEND_TEXT_FAIL(-508, "fail to send text")
CONNECT_SERVER_FAIL(-509, "fail to connect server")
INCORRECT_STATE(-510, "") // error msg 视情况而定

流式文本语音合成

最近更新时间：2025-08-21 15:06:12

1. 接入准备

1.1 SDK 获取

流式 TTS Android SDK 以及 Demo 的下载地址：[接入 SDK 下载](#)。

1.2 接入须知

- 该 SDK 需要手机能够连接网络，且 Android 设备 API Level 为16及以上版本。
- 运行 Demo 必须设置 AppID、SecretID、SecretKey，可在 [API 密钥管理](#) 中获取。

1.3 添加 SDK

将下载的 SDK 文件(*.aar)复制到项目的 libs 文件夹中，并在项目的 build.gradle 文件中添加以下代码：

```
dependencies {  
    implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')  
    // 流式TTS SDK  
    implementation files("libs/stream_tts-release-VERSION.aar")  
    // 流式TTS SDK内部依赖的okhttp库（必选）  
    implementation 'com.squareup.okhttp3:okhttp:4.9.3'  
    // 流式TTS SDK内部依赖的gson库（必选）  
    implementation 'com.google.code.gson:gson:2.8.9'  
}
```

1.4 添加权限

在项目的 AndroidManifest.xml 文件中添加 SDK 需要的权限，例如：

```
<uses-permission android:name="android.permission.INTERNET" />
```

2. SDK 接口使用说明

本章节主要介绍 SDK 接口调用详细步骤以及接口调用时机（具体使用方法可参考 SDK 交付 Demo 内的示例代码）

2.1 构造配置项

```
FlowingSpeechSynthesizerRequest request = new
FlowingSpeechSynthesizerRequest();
/***** 配置项含义也可参考官网文档:
https://cloud.tencent.com/document/product/1073/108595 *****/
request.setVolume(volume); // 音量大小, 范围[-10, 10], 对应音量大小。默认为0,
代表正常音量, 值越大音量越高。
request.setSpeed(speed); // 语速, 范围: [-2, 6], 分别对应不同语速: -2: 代表0.6
倍; -1: 代表0.8倍; 0: 代表1.0倍(默认); 1: 代表1.2倍; 2: 代表1.5倍; 6: 代表2.5
倍
request.setCodec("pcm"); // 返回音频格式: pcm: 返回二进制pcm音频(默认); mp3:
返回二进制mp3音频
request.setSampleRate(SAMPLE_RATE); // 音频采样率: 24000: 24k(部分音色支持);
16000: 16k(默认); 8000: 8k
request.setVoiceType(voiceType); // 音色ID
request.setEnableSubtitle(true); // 是否开启时间戳功能, 默认为false。
request.setEmotionCategory("neutral"); // 控制合成音频的情感, 仅支持多情感音色使
用
request.setEmotionIntensity(100); // 控制合成音频情感程度, 取值范围为
[50, 200], 默认为 100; 只有 EmotionCategory 不为空时生效。
request.setSessionId(UUID.randomUUID().toString()); // sessionId, 需要保持全
局唯一(推荐使用 uuid), 遇到问题需要提供该值方便服务端排查
request.set(String_key, Object_value); // 设置自定义扩展参数(可选, 非必须设置)
/***** 配置项含义也可参考官网文档:
https://cloud.tencent.com/document/product/1073/108595 *****/
```

2.2 账号信息配置

```
// 账号信息获取可参考 0.2接入须知
Credential credential = new Credential(appId, secretId, secretKey,
token);
```

2.3 构造合成器

```
// 网络连接的代理, 全局唯一即可
private static final SpeechClient proxy = new SpeechClient();
...
FlowingSpeechSynthesizerListener listener = new
FlowingSpeechSynthesizerListener() {
    @Override
    public void onSynthesisStart(SpeechSynthesizerResponse response) { //
合成开始}
```



```
@Override
public void onSynthesisEnd(SpeechSynthesizerResponse response) { //合成结束}

@Override
public void onAudioResult(ByteBuffer buffer) { //合成的音频数据}

@Override
public void onTextResult(SpeechSynthesizerResponse response) { //合成文本的信息}

@Override
public void onSynthesisCancel() { //合成取消}

@Override
public void onSynthesisFail(SpeechSynthesizerResponse response) { //合成失败}
};
// 构造合成器
FlowingSpeechSynthesizer synthesizer = new
FlowingSpeechSynthesizer(proxy, credential, request, listener);
```

2.4 启动合成器

```
synthesizer.start();
```

2.5 发送文本

```
// 输入待合成的文本text
synthesizer.process(text);
```

2.4 停止合成器

```
// 会向服务端发送断链请求，服务端完成已有的合成任务后向客户端返回断链确认，客户端再断开websocket链接
synthesizer.stop();
```

2.7 取消合成

```
// 会直接断开websocket链接  
synthesizer.cancel();
```

3. 接口详情

3.1 FlowingSpeechSynthesizer

3.1.1 构造接口

```
public FlowingSpeechSynthesizer(SpeechClient client,  
                                Credential credential,  
                                FlowingSpeechSynthesizerRequest request,  
                                FlowingSpeechSynthesizerListener  
listener) throws SynthesizerException
```

功能：流式 TTS SDK 的核心类的构造接口，注意异常捕获。

参数：

参数类型	参数名称	参数含义
SpeechClient	client	网络连接的代理，全局唯一即可
Credential	credential	鉴权相关信息的实体类
FlowingSpeechSynthesizerRequest	request	语音合成的配置项
FlowingSpeechSynthesizerListener	listener	语音合成的关键事件回调

3.1.2 启动合成器接口

```
public void start() throws Exception
```

功能：启动合成器，注意异常捕获。

3.1.3 发送文本接口

```
public void process(String text) throws Exception
```

功能：发送待合成文本，注意异常捕获。

参数：

参数类型	参数名称	参数含义
String	text	待合成文本

3.1.4 结束合成接口

```
/**
 * 结束合成:发送结束合成通知,接收服务端确认
 */
public void stop() throws Exception
/**
 * 结束合成:发送结束合成通知,接收服务端确认, 超时未返回则抛出异常
 */
public void stop(long milliseconds) throws Exception
```

3.1.5 取消合成接口

```
public void cancel()
```

功能: 取消合成, 直接关闭 WebSocket 连接, 不向服务端发送结束通知。

3.1.6 SDK 版本号

```
public static String version()
```

3.2 FlowingSpeechSynthesizerRequest

SDK 配置类, 配置项含义请参见 [流式文本语音合成](#)。

类型	名称	含义	默认值
Float	volume	音量大小, 范围[-10, 10], 对应音量大小。默认为0, 代表正常音量, 值越大音量越高。	0
Float	speed	语速, 范围: [-2, 6], 分别对应不同语速: -2: 代表0.6倍 -1: 代表0.8倍 0: 代表1.0倍 (默认) 1: 代表1.2倍 2: 代表1.5倍 6: 代表2.5倍	0

String	codec	返回音频格式： - pcm：返回二进制 pcm 音频（默认） - mp3：返回二进制 mp3 音频	pcm
Integer	sampleRate	音频采样率： 24000：24k（部分音色支持） 16000：16k（默认） 8000：8k	16000
Integer	voiceType	音色 ID，取值可参见 官网文档	0
Boolean	enableSubtitle	是否开启时间戳功能，默认为 false	false
String	emotionCategory	控制合成音频的情感，仅支持多情感音色使用。取值： neutral(中性)、sad(悲伤)、happy(高兴)、angry(生气)、fear(恐惧)、news(新闻)、story(故事)、radio(广播)、poetry(诗歌)、call(客服)、撒娇(sajiao)、厌恶(disgusted)、震惊(amaze)、平静(peaceful)、兴奋(exciting)、傲娇(aojiao)、解说(jieshuo)	""
Integer	emotionIntensity	控制合成音频情感程度，取值范围为 [50,200]，默认为 100；只有 EmotionCategory 不为空时生效。	100
String	sessionId	需要保持全局唯一（推荐使用 UUID），遇到问题需要提供该值方便服务端排查	""

3.3 Credential

密钥信息实体类，可在 [API 密钥管理](#) 中获取。

类型	名称	含义	默认值
String	appid	appid，在控制台申请。	""
String	secretId	secretId，在控制台申请。	""
String	secretKey	secretKey，在控制台申请。	""
String	token	用于临时授权场景。	""

3.4 FlowingSpeechSynthesizerListener

合成监听类

```
public abstract class FlowingSpeechSynthesizerListener {
```

```
//合成开始
public abstract void onSynthesisStart(SpeechSynthesizerResponse
response);
//合成结束
public abstract void onSynthesisEnd(SpeechSynthesizerResponse
response);
//合成的音频数据
public abstract void onAudioResult(ByteBuffer data);
//合成文本的信息
public abstract void onTextResult(SpeechSynthesizerResponse
response);
//合成取消
public abstract void onSynthesisCancel();
//合成失败
public abstract void onSynthesisFail(SpeechSynthesizerResponse
response);
}
```

3.5 SpeechSynthesizerResponse

合成 Response

类型	名称	含义
String	session Id	由客户端在握手阶段生成并赋值在调用参数中。
Integer	code	状态码，0代表正常，非0值表示发生错误。
Integer	end	该字段返回1时表示文本全部合成结束，客户端收到后需主动关闭 WebSocket 连接。
String	message	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。
String	request Id	音频流唯一 ID，由服务端在握手阶段自动生成。
String	message Id	本 message 唯一 ID。
SpeechSynthesizerResult	result	语音合成文本结果。

3.6 SpeechSynthesizerResult

合成结果

类型	名称	含义
SpeechSynthesizerSubtitle[]	subtitles	词列表信息。

3.7 SpeechSynthesizerSubtitle

合成词信息

类型	名称	含义
String	text	文本信息
Integer	beginTime	文本对应 TTS 语音开始时间戳。
Integer	endTime	文本对应 TTS 语音结束时间戳。
Integer	beginIndex	该字在整句中的开始位置。
Integer	endIndex	该字在整句中的结束位置。
String	phoneme	该字的音素。

4. 错误码

错误码
CLIENT_CANNOT_BE_NULL(-400, "client cannot be null")
CREDENTIAL_CANNOT_BE_NULL(-401, "credential cannot be null")
REQUEST_CANNOT_BE_NULL(-402, "request cannot be null")
LISTENER_CANNOT_BE_NULL(-403, "listener cannot be null")
APPID_IS_EMPTY(-404, "appId cannot be empty")
SECRETID_IS_EMPTY(-405, "secretId cannot be empty")
SECRETKEY_IS_EMPTY(-406, "secretKey cannot be empty")
START_SYNTHESIZER_FAIL(-407, "fail to start synthesizer")
SEND_TEXT_FAIL(-408, "fail to send text")
CONNECT_SERVER_FAIL(-409, "fail to connect server")

```
INCORRECT_STATE(-410, "") // error msg 视情况而定
```

iOS SDK

基础语音合成

最近更新时间：2025-08-06 11:08:32

❗ 说明

- 当前页面为新版（V2.0.0及以上）SDK 开发文档。新客户可直接按当前文档接入新版 SDK。
- 旧版（V1.5.1版本及以下）开发文档已于2022年9月5日下线。正在使用旧版 SDK 的客户，可前往 [控制台](#) 查看开发文档。
- 新版 SDK 在稳定性、功能健全性、接口自由度等方面都有所优化。我们将继续支持旧版（V1.5.1版本及以下）SDK，但建议正在使用旧版 SDK 的客户及时升级到新版，以获取更好的使用体验。

开发相关

开发准备

- 支持 iOS 9.0 及以上版本，不支持 bitcode 版本。
- 合成实时流式语音，需要手机能够连接网络（3/4/5G 或 Wi-Fi 网络等）。
- 语音合成 [iOS SDK](#)。
- 服务端 [API 文档](#)。

导入 SDK

直接导入

SDK 文件夹内的 QCloudTTS.framework 即为 SDK，引入 `<QCloudTTS/QCloudTTSEngine.h>` 及 `<QCloudTTS/QCloudMediaPlayer.h>` 即可调用 SDK 相关接口。

CocoaPods 导入

```
pod 'QCloudTTS'
```

接口说明

获得 TTS 合成器实例

```
// 获得实例
QCloudTTSEngine *ttsEngine = [QCloudTTSEngine sharedInstance];

// 销毁实例
```



```
[QCloudTTSEngine instanceRelease];
```

初始化引擎

```
-(void) engineInit:(TtsMode)mode Delegate:(id<QCloudTTSEngineDelegate>
_Nonnull) delegate;
//TtsMode: 固定填入 TTS_MODE_ONLINE
//QCloudTTSEngineDelegate: TTS合成器代理，用于获取合成结果
```

QCloudTTSEngineDelegate: TTS 合成器代理，用于获取合成结果

onSynthesizeData 方法签名说明

参数	说明
NSData data	语音数据
String utteranceId	语句 ID
String text	文本
int engineType	引擎类型，0：在线，1：离线；当前是纯在线 SDK，请忽略此参数

onError 方法签名说明

参数	说明
TtsError error	错误信息，无错误返回 null
String utteranceId	语句 ID（如果有则返回）
String text	文本（如果有则返回）

示例

```
-(void) onSynthesizeData:(NSData *_Nullable)data UtteranceId:(NSString
*_Nullable)utteranceId Text:(NSString *_Nullable)text EngineType:
(NSInteger)type{
    // 您可以在这里将音频保存或者送入播放接口播放，可调用播放器入参接口入参
}

-(void) onError:(TtsError *_Nullable)error UtteranceId:(NSString
*_Nullable)utteranceId Text:(NSString *_Nullable)text{
    // 您可以在这里添加错误后处理
```

```
}

- (void)onOfflineAuthInfo:(QCloudOfflineAuthInfo *
_Nonnull)OfflineAuthInfo {
    // 离在线SDK保留接口，请忽略，如果您后续升级为离线SDK或者离在线SDK，此接口将用于
    返回授权信息
}
```

合成文本入参接口

接口	说明
<code>-(TtsError *_Nullable) synthesize:(NSString *_Nonnull)text UtteranceId:(NSString *_Nullable)utteranceId;</code>	- text 为需要合成的文本 - utteranceId 为标记该文本的 ID，将随合成结果返回宿主层，业务不需要可传 nil

示例

```
// 内部有维护队列，可持续添加语句，SDK内将依次合成
TtsError error = nil;
error = [ttsEngine synthesize:@"今天天气不错".text UtteranceId:@"第1句"];
error = [ttsEngine synthesize:@"腾讯云语音合成".text UtteranceId:@"第2句"];
error = [ttsEngine synthesize:@"腾讯云AI".text UtteranceId:@"第3句"];
error = [ttsEngine synthesize:@"腾讯云".text UtteranceId:nil];

// 取消未合成的任务并清空内部队列
[ttsEngine cancel];
```

QCloudTTSEngine 配置参数方法

接口	说明
<code>setOnlineAuthParam:(NSInteger)appId SecretId:(NSString *_Nonnull)secretId SecretKey:(NSString *_Nonnull)secretKey Token:(NSString *_Nullable)token</code>	appId: 配置腾讯云 appId、SecretId、SecretKey、Token (token, 可为空, 如果使用 STS 临时证书 鉴权, secretId 和 secretKey 均入参临时的, 同时需要入参对应的 token)
<code>setOnlineVoiceSpeed:(float)voiceSpeed;</code>	语速, 范围: [-2, 6], 分别对应不同语速: -2代表0.6倍 -1代表0.8倍

	• 0代表1.0倍（默认） • 1代表1.2倍 • 2代表1.5倍 • 6代表2.5倍 如果需要更细化的语速，可以保留小数点后2位，例如 0.5/1.25/2.81等。 参数值与实际语速转换，可参考 代码示例。
setOnlineVoiceVolume:(float)voiceVolume;	设置在线所合成音频的音量，默认0
setOnlineVoiceLanguage:(int)primaryLanguage;	主语言类型：1-中文（默认）2-英文
setOnlineCodec:(NSString* _Nonnull)code;	在线模式编码格式， 非业务必要不建议更改 ：默认 mp3 ，目前支持 mp3、wav、pcm，如更改为 pcm 不支持播放
setOnlineVoiceType:(int)voiceType;	设置在线所合成音频的音色 ID，完整的音色 ID 列表参见 基础语音合成
setTimeoutIntervalForRequest:(int)timeout;	连接超时，范围：[500,30000]，单位ms，默认 15000ms
setTimeoutIntervalForResource:(int)timeout;	读取超时，范围：[2200,60000]，单位ms，默认 30000ms

示例

```
[ttsEngine setOnlineAuthParam:123 SecretId:@"AK****Hp7"
SecretKey:@"D9***c2" Token:nil];
[ttsEngine setOnlineVoiceSpeed:0.0]; //配置语速
[ttsEngine setOnlineVoiceType:1001]; //配置音色id
[ttsEngine setOnlineVoiceLanguage:1]; //配置主语言
[ttsEngine setOnlineVoiceVolume:0]; //配置音量
[ttsEngine setTimeoutIntervalForRequest:15 *1000];
[ttsEngine setTimeoutIntervalForResource:30 *1000];
```

播放接口

初始化播放器

如果 SDK 的内置播放器无法满足您的需求，您也可以使用自己实现的播放器替换。Demo 中也额外提供了一份播放器源码，您可以修改播放器逻辑，源代码位于 MediaPlayerDemo.m，与 SDK 内置播放器一致。

```
//使用SDK中提供的播放器
```

```
QCloudMediaPlayer player = [[QCloudMediaPlayer alloc] init];
```

```
//-----QCloudPlayerDelegate-----
```

```
//播放开始
```

```
-(void) onTTSPplayStart{  
    NSLog(@"onTTSPplayStart");  
}
```

```
//队列所有音频播放完成，音频等待中
```

```
-(void) onTTSPplayWait{  
    NSLog(@"onTTSPplayWait");  
}
```

```
//恢复播放
```

```
-(void) onTTSPplayResume{  
    NSLog(@"onTTSPplayResume");  
}
```

```
//暂停播放
```

```
-(void) onTTSPplayPause{  
  
    NSLog(@"onTTSPplayPause");  
}
```

```
//播放中止
```

```
-(void) onTTSPplayStop{  
    NSLog(@"onTTSPplayStop");  
  
}
```

```
//播放器异常
```

```
-(void) onTTSPplayError:(QCPLError* _Nullable)playError{  
  
    NSLog(@"playError.code==%,playError.message==%", @(playError.mCode), playError.message);  
}
```

```
//即将播放播放下一句，即将播放音频对应的句子，以及这句话utteranceId
```

```
-(void) onTTSPplayNextWithText:(NSString* _Nullable)text UtteranceId:  
(NSString* _Nullable)utteranceId{  
    NSLog(@"text==%,utteranceId==%", text, utteranceId);  
}
```

```
// 当前播放的字符,当前播放的字符在所在的句子中的下标.
// currentWord 当前读到的单个字,是一个估算值不一定准确
// currentIndex 当前播放句子中读到文字的下标
-(void)onTTSPplayProgressWithCurrentWord:(NSString* _Nullable)currentWord
CurrentIndex:(NSInteger)currentIndex{

    NSLog(@"CurrentWord==%@,currentIndex==%@",currentWord,@(currentIndex));
}
```

播放器入参

参数	说明
data	入参音频流,通过传入字节数组播放
url	入参音频文件,通过传入文件播放
String text	音频对应的文本
String utteranceId	文本 ID

示例

```
//通过音频数据入参
QCPlayerError err = [_player enqueueWithData:data Text:text
UtteranceId:utteranceId];

//通过音频文件入参
NSString *str = [self filePathWithName:@"tmp.mp3"];
NSURL * url = [NSURL URLWithString:str];
QCPlayerError err = [_player enqueueWithFile:url Text:text
UtteranceId:utteranceId];
```

客户端错误码

ID	错误码	说明
-1000	TTS_ERROR_CODE_UNINITIALIZE_D	SDK 未初始化
-1001	TTS_ERROR_CODE_GENERATE_SIGN_FAIL	签名失败

-102	TTS_ERROR_CODE_NETWORK_CONNECT_FAILED	网络异常
-103	TTS_ERROR_CODE_DECODE_FAIL	Response 解析失败
-104	TTS_ERROR_CODE_SERVER_RESPONSE_ERROR	后端返回失败错误码，详细错误信息请查看后端错误码
-105	TTS_ERROR_CODE_QUEUE_IS_FULL	合成队列已满
-106	TTS_ERROR_CODE_CANCEL_FAILURE	取消失败，请稍后重试
-900	TTS_ERROR_CODE_OFFLINE_NO_SUPPORT	当前 SDK 不支持离线合成能力，请使用 TtsMode.ONLINE

服务端错误码

更多信息请参见 [语音合成 API 文档](#)。

实时语音合成

最近更新时间：2025-08-06 11:08:32

开发环境

1. 支持 iOS 9.0 及以上版本。
2. 需要手机能够连接网络。

导入 SDK

直接导入

将 SDK 文件夹内的 QCloudRealTTS.xcframework 添加到项目的链接库中即可。

CocoaPods 导入

```
pod 'QCloudRealTTS'
```

接口说明

QCloudRealTTSController

实时语音合成控制器

```
@protocol QCloudRealTTSController <NSObject>
/*
 * 取消合成任务
 */
-(void)cancel;

@end
```

QCloudRealTTSConfig

实时语音合成配置

```
@interface QCloudRealTTSConfig : NSObject

@property (nonnull) NSString* appID; // 腾讯云 appid
@property (nonnull) NSString* secretID; // 腾讯云 secretID
@property (nonnull) NSString* secretKey; // 腾讯云 secretKey
```

```

@property (nonnull) NSString* token; // 临时token,不为空字符时生效,使用临时
token时,secretId,secretKey需为临时密钥
@property int connectTimeout; // > 0 生效,单位为ms,默认为0

/*
 * 设置传入后台的api的参数,参数可参考文档
https://cloud.tencent.com/document/product/1073/94308 说明
 * @param key 参数名称
 * @param value 参数值,参数值为nil会删除已设置的key
 */
- (QCloudRealTTSConfig*)setApiParam:(nonnull NSString*)key value:
(nullable NSString*)value;
- (QCloudRealTTSConfig*)setApiParam:(nonnull NSString*)key ivalue:
(NSInteger) value;
- (QCloudRealTTSConfig*)setApiParam:(nonnull NSString*)key fvalue:
(float) value;
- (QCloudRealTTSConfig*)setApiParam:(nonnull NSString*)key bvalue:
(BOOL) value;

/*
 * 创建实时语音合成控制器
 * @param listener 用于回调合成任务的接口及中间信息
 */
- (id<QCloudRealTTSController>)build:
(id<QCloudRealTTSListener>) listener;
@end

```

QCloudRealTTSListener

实时语音合成结果回调

```

@protocol QCloudRealTTSListener <NSObject>

/*
 * 合成任务结束
 */
- (void)onFinish;

/*
 * 合成任务出错
 * @param error 错误信息
 */
- (void)onError:(nonnull NSError*)error;

```



```
@optional
/*
 * 合成日志
 * @param value 日志信息
 * @param level 日志等级
 */
-(void)onLog:(nonnull NSString*)value level:(int)level;
/*
 * 服务端返回的音频数据,可参考文档
https://cloud.tencent.com/document/product/1073/94308 说明
 * @param data 服务端返回的音频数据
 */
-(void)onData:(nonnull NSData*)data;
/*
 * 服务端返回的json数据,可参考文档
https://cloud.tencent.com/document/product/1073/94308 说明
 * @param msg 服务端返回的json数据
 */
-(void)onMessage:(nonnull NSString*)msg;

@end
```

错误码

合成任务的 NSError 仅通过 QCloudStreamTTSListener的onError 回调出来。

```
enum : NSInteger {
    REALTTSPARAMETERERROR = 2000, // 参数错误,SDK配置项设置有问题,一般为授权信息没有设置
    REALTTSWEBSOCKETERROR = 2001, // websocket错误,网络问题
    REALTTSCANCELERROR = 2002, // 取消错误,成功调用cancel返回此错误
    REALTTSSERVERERROR = 2003, // 服务端返回错误,可通过取userInfo中的Message获取详细信息
};
```

流式文本语音合成

最近更新时间：2025-08-06 11:08:32

开发环境

1. 支持 iOS 9.0 及以上版本。
2. 需要手机能够连接网络。
3. 语音合成 [iOS SDK](#)。

导入 SDK

直接导入

将 SDK 文件夹内的 QCloudStreamTTS.xcframework 添加到项目的链接库中即可。

CocoaPods 导入

```
pod 'QCloudStreamTTS'
```

接口说明

QCloudStreamTTSTransformer

流式文本合成控制器

```
@protocol QCloudStreamTTSTransformer <NSObject>

/*
 * 发送合成文本, 需要在onReady后调用, 否则发送的文本不会处理
 * @param text 需要合成的文本
 */
-(void) synthesis:(nonnull NSString*) text;

/*
 * 取消合成任务
 */
-(void) cancel;

/*
 * 停止合成任务
 */
-(void) stop;
```

```
@end
```

QCloudStreamTTSTTSConfig

流式文本合成配置

```
@interface QCloudStreamTTSTTSConfig : NSObject

@property (nonnull) NSString* appID; // 腾讯云 appid
@property (nonnull) NSString* secretID; // 腾讯云 secretID
@property (nonnull) NSString* secretKey; // 腾讯云 secretKey
@property (nonnull) NSString* token; // 临时token, 不为空字符时生效, 使用临时token时, secretId, secretKey需为临时密钥
@property int connectTimeout; // 大于0生效, 单位为ms, 默认为0

/*
 * 设置传入后台的api的参数, 参数可参考文档
https://cloud.tencent.com/document/product/1073/108595 说明
 * @param key 参数名称
 * @param value 参数值, 参数值为nil会删除已设置的key
 */
- (QCloudStreamTTSTTSConfig*)setApiParam:(nonnull NSString*)key value:(nullable NSString*)value;
- (QCloudStreamTTSTTSConfig*)setApiParam:(nonnull NSString*)key ivalue:(NSInteger)value;
- (QCloudStreamTTSTTSConfig*)setApiParam:(nonnull NSString*)key fvalue:(float)value;
- (QCloudStreamTTSTTSConfig*)setApiParam:(nonnull NSString*)key bvalue:(BOOL)value;

/*
 * 创建流式语音合成控制器
 * @param listener 用于回调合成任务的接口及中间信息
 */
- (id<QCloudStreamTTSTTSController>)build:(id<QCloudStreamTTSTTSListener>)listener;

@end
```

QCloudStreamTTSTTSListener

```
@protocol QCloudStreamTTSTTSListener <NSObject>
```

```
@required
/*
 * 合成任务结束
 */
-(void)onFinish;
/*
 * 合成任务出错
 * @param error 错误信息
 */
-(void)onError:(nonnull NSError*)error;

@optional
/*
 * 合成日志
 * @param value 日志信息
 * @param level 日志等级
 */
-(void)onLog:(nonnull NSString*)value level:(int)level;
/*
 * 服务端返回的音频数据,可参考文档
https://cloud.tencent.com/document/product/1073/108595 说明
 * @param data 服务端返回的音频数据
 */
-(void)onData:(nonnull NSData*)data;
/*
 * 服务端返回的json数据,可参考文档
https://cloud.tencent.com/document/product/1073/108595 说明
 * @param msg 服务端返回的json数据
 */
-(void)onMessage:(nonnull NSString*)msg;
/*
 * 服务端返回ready事件
 */
-(void)onReady;

@end
```

错误码

合成任务的 NSError 仅通过 QCloudStreamTTSListener 的 onError 回调出来。

```
enum : NSInteger {
```

```
STREAMTTSPARAMETERERROR = 2000, // 参数错误, SDK配置项设置有问题, 一般为授权  
信息没有设置  
STREAMTTWEBSOCKETERROR = 2001, // websocket错误, 网络问题  
STREAMTTCANCELERROR = 2002, // 取消错误, 成功调用cancel返回此错误  
STREAMTTSERVERERROR = 2003, // 服务端返回错误, 可通过取userInfo中的  
Message获取详细信息  
};
```

HarmonyOS NEXT

基础语音合成

最近更新时间：2024-10-14 15:58:41

接入准备

SDK 获取

语音合成 HarmonyOS NEXT SDK 以及 Demo 的下载地址：[接入 SDK 下载](#)。

接入须知

- 开发者在调用前请先查看[基础语音合成 相关接口](#)，了解接口的使用要求和使用步骤。
- 该接口需要手机能够连接网络（3G、4G、5G 或 Wi-Fi 等）
- 运行 Demo 必须设置 SecretID、SecretKey，可在[API 密钥管理](#)中获取。

SDK 导入

1. 下载并解压 HarmonyOS NEXT SDK 压缩包，压缩包包含 demo、sdk 和 doc，其中 sdk/qcloudtts_v1.x.x.har 为语音合成的 SDK。
2. 在工程中添加依赖库，oh-package.jsons 中添加一下依赖

```
"dependencies": {  
  "QCloudTTS": "file:../../sdk/qcloudtts_v1.0.0_96b6073.har"  
}
```

请根据实际路径替换配置。

快速接入

以下为 demo 中的代码片段，完整代码请参考 sdk/demo 工程。

配置合成任务

```
let builder = new QCloud.TTS.Builder()  
builder.secretID = this._secret_id  
builder.secretKey = this._secret_key  
builder.token = this._token  
builder.setApiParam(QCloud.TTS.kText, this._text)  
builder.setApiParam(QCloud.TTS.kVolume, this._volume)  
builder.setApiParam(QCloud.TTS.kSpeed, this._speed)
```

```
builder.setApiParam(QCloud.TTS.kVoiceType, this._voice_type)
builder.setApiParam(QCloud.TTS.kEnableSubtitle, this._subtitle)
```

配置合成任务相关参数

启动合成任务

使用上面的Builder可以创建并启动合成任务并等待结果

```
let ret = await builder.build().result
```

主要接口类和方法说明

Builder类说明

Builder 用于创建合成任务

方法

i. setApiParam(key: string, value: string | number | null)

设置请求后台接口时的参数,后台支持参数请参考[基础语音合成 相关接口](#)

名称	类型	描述
key	string	请求后台的参数名称
value	string number null	请求后台的参数值，null时将删除已设置的参数，number 会转为 string 后设置

ii. getApiParam(key: string): string | number | boolean | null

获取已设置的请求参数

名称	类型	描述
key	string	参数名称

iii. build(): Controller

创建合成任务同时返回合成任务控制器

属性

名称	类型	描述
secretID	string	腾讯云 secretId
secretKey	string	腾讯云 secretKey

token	string	腾讯云临时 token，为空字符串时表示不使用临时授权，临时授权参考 获取联合身份临时访问凭证
region	string	请求 region，默认值为 ap-guangzhou 详见产品支持的 地域列表
connect Timeout	number	连接超时设置，单位为ms，默认值为10000
readTimeout	number	读取超时设置

Controller类说明

Controller 用于控制合成任务的流程
属性

名称	类型	描述
task	Promise<string>	合成任务服务端返回结果
result	Promise<QCloud.TTS.Server.ServerResponse>	合成任务服务端返回结果解析后的结构，请参考 基础语音合成 相关接口 输出参数的说明## Listener类说明

错误码

合成任务抛出类型为SDKError时的错误码

错误码	名称	描述
-1	UNKNOWN_ERROR	未知错误
1	REQUEST_ERROR	请求失败，具体原因请查看 message 和 inner_error
2	SERVER_ERROR	服务端返回错误，具体信息查看 message

实时语音合成

最近更新时间：2025-01-06 12:01:52

接入准备

SDK 获取

语音合成 HarmonyOS NEXT SDK以及 Demo 的下载地址：[接入SDK下载](#)。

接入须知

- 开发者在调用前请先查看[实时语音合成 相关接口](#)，了解接口的使用要求和使用步骤。
- 该接口需要手机能够连接网络（3G、4G、5G 或 Wi-Fi 等）
- 运行 Demo 必须设置 AppID、SecretID、SecretKey，可在 [API 密钥管理](#) 中获取。

SDK 导入

1. 下载并解压 HarmonyOS NEXT SDK 压缩包，压缩包包含 demo、sdk 和 doc，其中 sdk/realtts_v1.x.x.har 为实时语音合成的 SDK。
2. 在工程中添加依赖库，oh-package.jsons 中添加一下依赖。

```
"dependencies": {  
  "realtts": "file:../../sdk/realtts_v1.0.0_96b6073.har"  
}
```

请根据实际路径替换配置

快速接入

以下为 demo 中的代码片段,完整代码请参考 sdk/demo 工程。

配置合成任务

```
let builder = new QCloud.RealTTS.Builder()  
builder.appID = this._app_id  
builder.secretID = this._secret_id  
builder.secretKey = this._secret_key  
builder.token = this._token  
builder.setApiParam(QCloud.RealTTS.kText, this._text)  
builder.setApiParam(QCloud.RealTTS.kVoiceType, this._voice_type)  
builder.setApiParam(QCloud.RealTTS.kVolume, this._volume)  
builder.setApiParam(QCloud.RealTTS.kSpeed, this._speed)
```

```
builder.setApiParam(QCloud.RealTTS.kCodec, 'pcm')
```

配置合成任务相关参数

启动合成任务

使用上面的Builder可以创建并启动合成任务并等待结果

```
this._ctl = builder.build({})  
await this._ctl.task
```

主要接口类和方法说明

Builder类说明

Builder 用于创建合成任务

方法

1. setApiParam(key: string, value: string | number | null)

设置请求后台接口时的参数,后台支持参数请参考[实时语音合成 相关接口](#)。

名称	类型	描述
key	string	请求后台的参数名称
value	string number null	请求后台的参数值,null 时将删除已设置的参数,number 会转为 string 后设置

2. getApiParam(key: string): string | number | boolean | null

获取已设置的请求参数

名称	类型	描述
key	string	参数名称

3. build(): Controller

创建合成任务同时返回合成任务控制器

名称	类型	描述
listener	Listener	流式合成过程中的数据及事件回调

属性

名称	类型	描述
appId	string	腾讯云 appId
secretId	string	腾讯云 secretId
secretKey	string	腾讯云 secretKey
token	string	腾讯云临时 token，为空字符串时表示不使用临时授权,临时授权参考 获取联合身份临时访问凭证 。

Controller类说明

Controller 用于控制合成任务的流程

方法

1. cancel(): void

取消合成任务,成功后task抛出取消异常。

属性

名称	类型	描述
task	Promise<void>	合成任务

Listener类说明

Listener 流式合成过程中的数据及事件回调

1. onLog?: (value: string, level: number) => void

日志回调

名称	类型	描述
value	string	日志信息
level	number	日志等级

2. onData?: (data: ArrayBuffer) => void

服务端返回的音频数据,可参考文档 [流式文本语音合成](#) 说明。

名称	类型	描述
data	ArrayBuffer	服务端返回的音频数据

3. onMessage?: (msg: string) => void

服务端返回的json数据,可参考文档 [流式文本语音合成](#)

<https://cloud.tencent.com/document/product/1073/108595> 说明。

名称	类型	描述
msg	ArrayBuffer	服务端返回的json数据

错误码

合成任务抛出类型为 SDKError 时的错误码

错误码	名称	描述
-1	UNKNOWN_ERROR	未知错误
1	REQUEST_ERROR	请求失败,具体原因请查看message和inner_error
2	SERVER_ERROR	服务端返回错误,具体信息查看message
3	SIGNATURE_ERROR	签名错误
4	PARAMETER_ERROR	参数错误,SDK本地参数校验
5	CANCEL_ERROR	调用cancel成功

流式文本语音合成

最近更新时间：2024-12-23 14:18:22

HarmonyOS NEXT SDK 说明

接入准备

SDK 获取

语音合成 HarmonyOS NEXT SDK 以及 Demo 的下载地址：[接入 SDK 下载](#)。

接入须知

- 开发者在调用前请先查看 [流式文本合成相关接口](#)，了解接口的使用要求和使用步骤。
- 该接口需要手机能够连接网络（3G、4G、5G 或 Wi-Fi 等）
- 运行 Demo 必须设置 AppID、SecretID、SecretKey，可在 [API 密钥管理](#) 中获取。

SDK 导入

1. 下载并解压 HarmonyOS NEXT SDK 压缩包，压缩包包含 demo、sdk 和 doc，其中 sdk/streamtts_v1.x.x.har 为流式语音合成的 SDK。
2. 在工程中添加依赖库，oh-package.jsons 中添加一下依赖。

```
"dependencies": {  
  "streamtts": "file:../../sdk/streamtts_v1.0.0_96b6073.har"  
}
```

请根据实际路径替换配置

快速接入

以下为 demo 中的代码片段,完整代码请参考 sdk/demo 工程。

配置合成任务

```
let builder = new QCloud.StreamTTS.Builder()  
builder.appID = this._app_id  
builder.secretID = this._secret_id  
builder.secretKey = this._secret_key  
builder.token = this._token  
builder.setApiParam(QCloud.StreamTTS.kVoiceType, this._voice_type)  
builder.setApiParam(QCloud.StreamTTS.kVolume, this._volume)
```

```
builder.setApiParam(QCloud.StreamTTS.kSpeed, this._speed)
```

配置合成任务相关参数

启动合成任务

使用上面的 Builder 可以创建并启动合成任务并等待任务结束

```
this._ctl = builder.build({})  
await this._ctl.task
```

主要接口类和方法说明

Builder 类说明

Builder 用于创建合成任务

方法

1. setApiParam(key: string, value: string | number | null)

设置请求后台接口时的参数,后台支持参数请参考 [流式文本合成相关接口](#)。

名称	类型	描述
key	string	请求后台的参数名称
value	string number null	请求后台的参数值, null 时将删除已设置的参数, number 会转为 string 后设置

2. getApiParam(key: string): string | number | boolean | null

获取已设置的请求参数

名称	类型	描述
key	string	参数名称

3. build(): Controller

创建合成任务同时返回合成任务控制器

名称	类型	描述
listener	Listener	流式合成过程中的数据及事件回调

属性

名称	类型	描述
appId	string	腾讯云 appId
secretId	string	腾讯云 secretId
secretKey	string	腾讯云 secretKey
token	string	腾讯云临时token,为空字符串时表示不使用临时授权，临时授权参考 安全凭证服务

Controller类说明

Controller 用于控制合成任务的流程

方法

1. synthesis(text: string): void

合成语音,此方法需要在接收到服务端ready事件后调用。

名称	类型	描述
text	string	需要合成的文本

2. cancel(): void

取消合成任务,成功后task抛出取消异常。

3. stop(): void

停止合成任务,成功后task正常返回。

属性

名称	类型	描述
task	Promise<void>	合成任务

Listener类说明

Listener 流式合成过程中的数据及事件回调

1. onLog?: (value: string, level: number) => void

日志回调

名称	类型	描述
value	string	日志信息
level	number	日志等级

2. onData?: (data: ArrayBuffer) => void

服务端返回的音频数据，可参考文档 [流式文本语音合成](#) 说明。

名称	类型	描述
data	ArrayBuffer	服务端返回的音频数据

3. onMessage?: (msg: string) => void

服务端返回的 json 数据，可参考文档 [流式文本语音合成](#) 说明。

名称	类型	描述
msg	ArrayBuffer	服务端返回的 json 数据

4. onReady?: () => void

服务端返回 ready 事件

错误码

合成任务抛出类型为 SDKError 时的错误码

错误码	名称	描述
-1	UNKNOWN_ERROR	未知错误
1	REQUEST_ERROR	请求失败，具体原因请查看 message 和 inner_error
2	SERVER_ERROR	服务端返回错误，具体信息查看 message
3	SIGNATURE_ERROR	签名错误
4	PARAMETER_ERROR	参数错误，SDK 本地参数校验
5	CANCEL_ERROR	调用 cancel 成功

FLUTTER

基础语音合成

最近更新时间：2024-12-12 22:29:02

Flutter SDK

SDK以插件的方式封装了Android和iOS语音合成功能,提供flutter版本的语音合成,本文介绍SDK的安装方法及示例。

开发环境

- dart >= 2.18.4
- flutter >= 3.3.8

获取安装

[下载SDK](#) SDK 内 tts_plugin 目录即为 flutter 插件, 插件内 example 目录下为 demo 示例。

接口说明

接口示例代码为 demo 部分代码,完整代码请参考位于 example 里的 demo 示例。

TTSTControllerConfig

TTSTController 相关配置

参数

```
String secretId = ""; // 腾讯云 secretId
String secretKey = ""; // 腾讯云 secretKey
double voiceSpeed = 0; // 语速,详情见API文档
double voiceVolume = 1; // 音量,详情见API文档
int voiceType = 1001; // 音色,详情见API文档
int voiceLanguage = 1; // 语音,详情见API文档
String codec = "mp3"; // 编码,详情见API文档
int connectTimeout = 15 * 1000; //连接超时, 范围: [500,30000], 单位ms, 默认15000ms
int readTimeout = 30 * 1000; //读取超时, 范围: [2200,60000], 单位ms , 默认30000ms
```

示例

```
var _config = TTSControllerConfig();
_config.secretId = secretId;
TTSController.instance.config = _config;
```

TTSTController

控制语音合成的流程及获取语音合成的结果,该类为单例模式。

方法

```
Future<TTSDData> synthesize(String text, String? utteranceId) async <-->
```

合成

```
cancel() async <--> 停止合成
```

```
release() async <--> 释放资源
```

示例

```
var ret = await TTSController.instance.synthesize(_text, null);
```

TTSDData

合成的音频数据

参数

```
Uint8List data; //音频数据
String text; //合成文本
String? utteranceId; //合成传入标识
```

TTSError

合成过程中的错误,错误码参考[Android SDK](#)与[iOS SDK文档](#)里面客户端错误码。

参数

```
int code = 0; // 错误码
String message = ""; // 错误信息
```