

# 语音合成 离线 SDK 文档



## Copyright Notice

©2013–2025 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

## Trademark Notice

 Tencent Cloud

This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies ("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

## Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

## Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

# Contents

离在线 SDK 文档

  开通授权

  Android SDK 精品

  iOS SDK 精品

  Android SDK 基础

  iOS SDK 基础

  离在线语音合成 SDK 个人信息保护规则

# 离在线 SDK 文档

## 开通授权

Last updated: 2023-12-27 18:15:51

### 语音合成离在线 SDK 测试申请操作指引

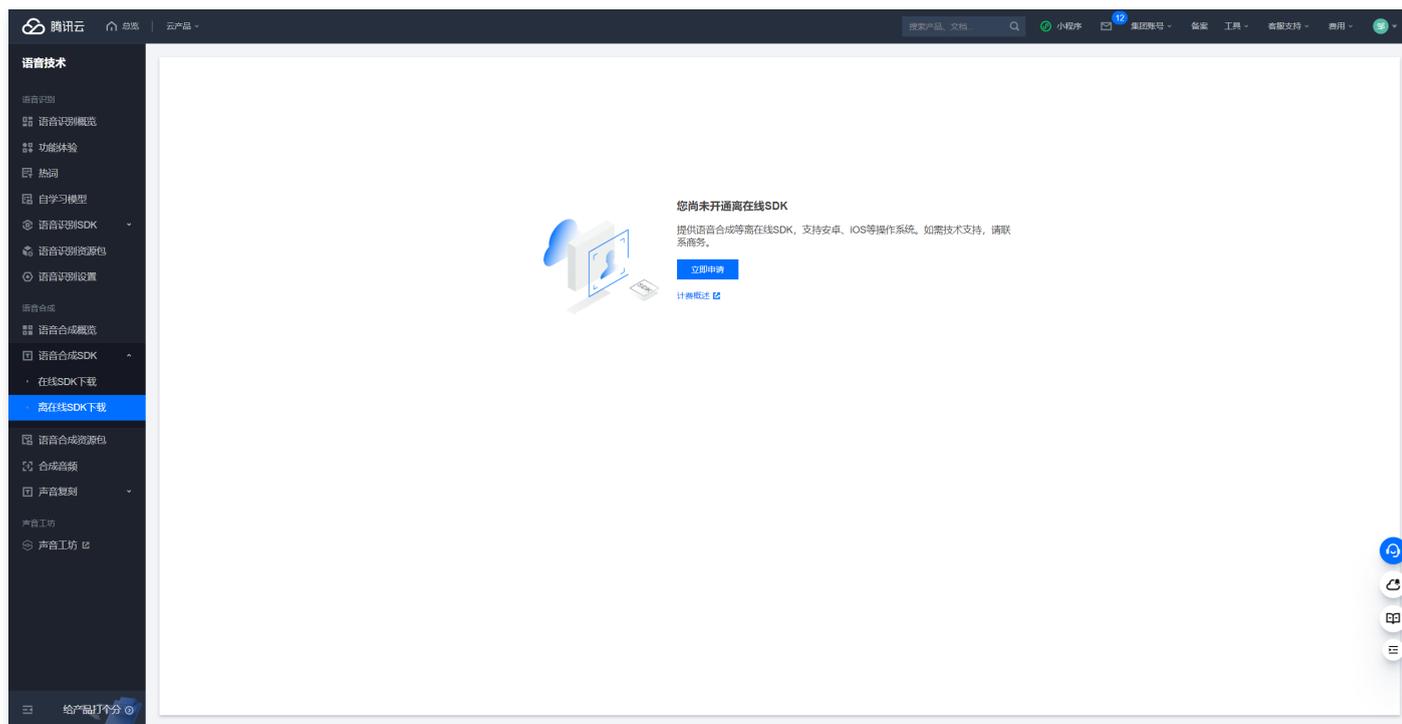
#### 测试准备

1. 准备好企业的腾讯云账号
2. 访问链接，登录腾讯云语音技术控制台。

#### 说明:

目前 ASR、TTS、声音工坊控制台已合并为语音技术控制台。

3. 选择离在线 SDK 管理，若提示“没有开通语音合成服务”，先点击开通语音合成服务；提示“未进行企业认证”，请先进行企业认证，如下图所示：



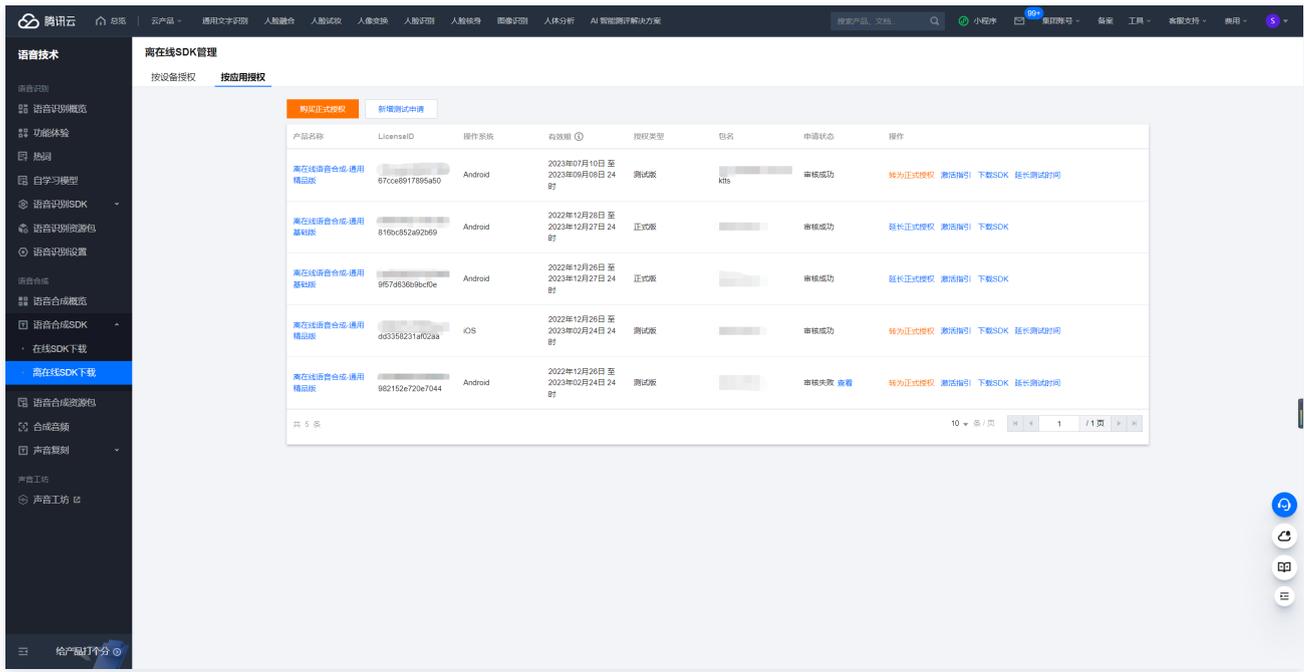
#### 新增测试申请

1. 单击上图所示立即申请，填写语音合成 SDK 需求表。
2. 填写语音合成 SDK 需求表后请告知对接的商务，进行需求审核（需求审核的目的是保证上线产品与客户需求匹配、可提供产品最终使用）。
3. 在离在线 SDK 管理页点击选择按设备授权或按应用授权，单击新增测试申请，审核通过后即可测试试用。

按设备授权	有效期60天	默认授权测试设备数 5台
按应用授权	有效期60天	默认授权测试应用数 1个

#### 注意:

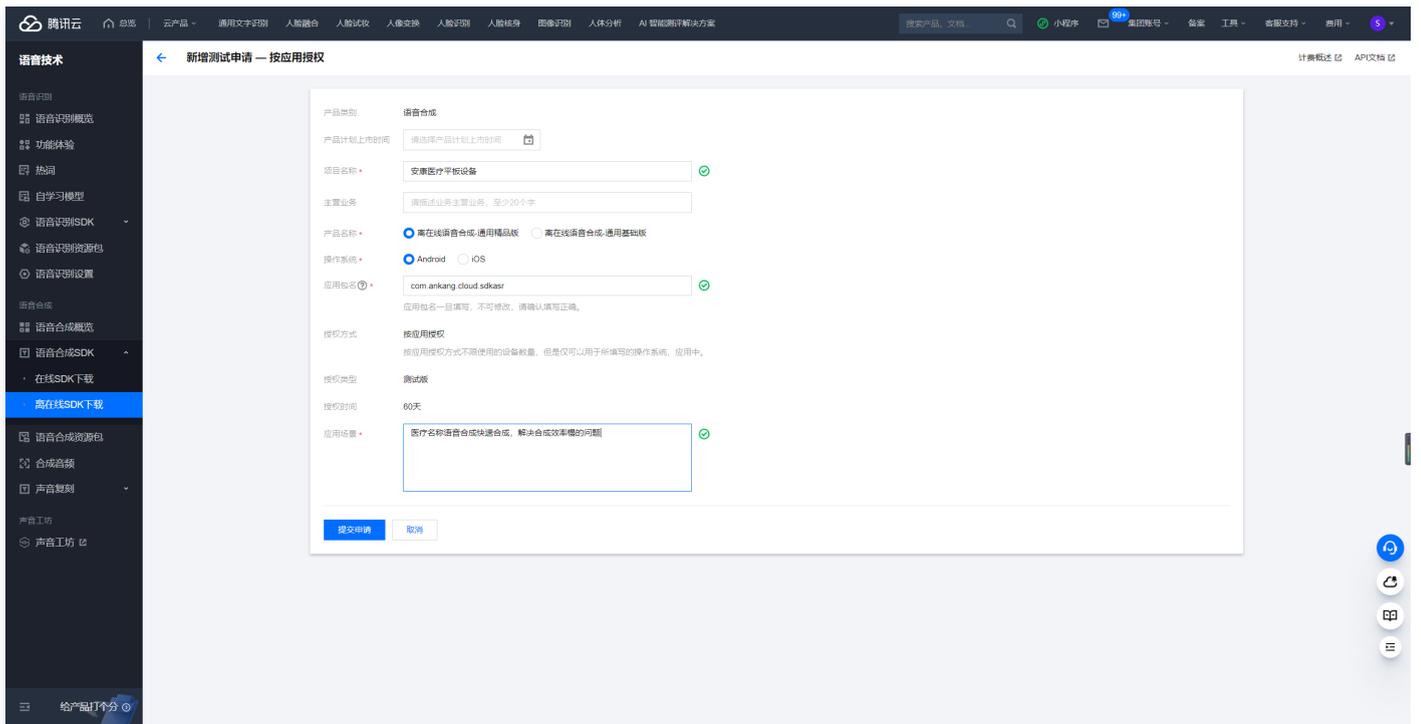
此处用语音合成【按应用授权】演示操作流程（按设备授权或语音合成可同样参考）

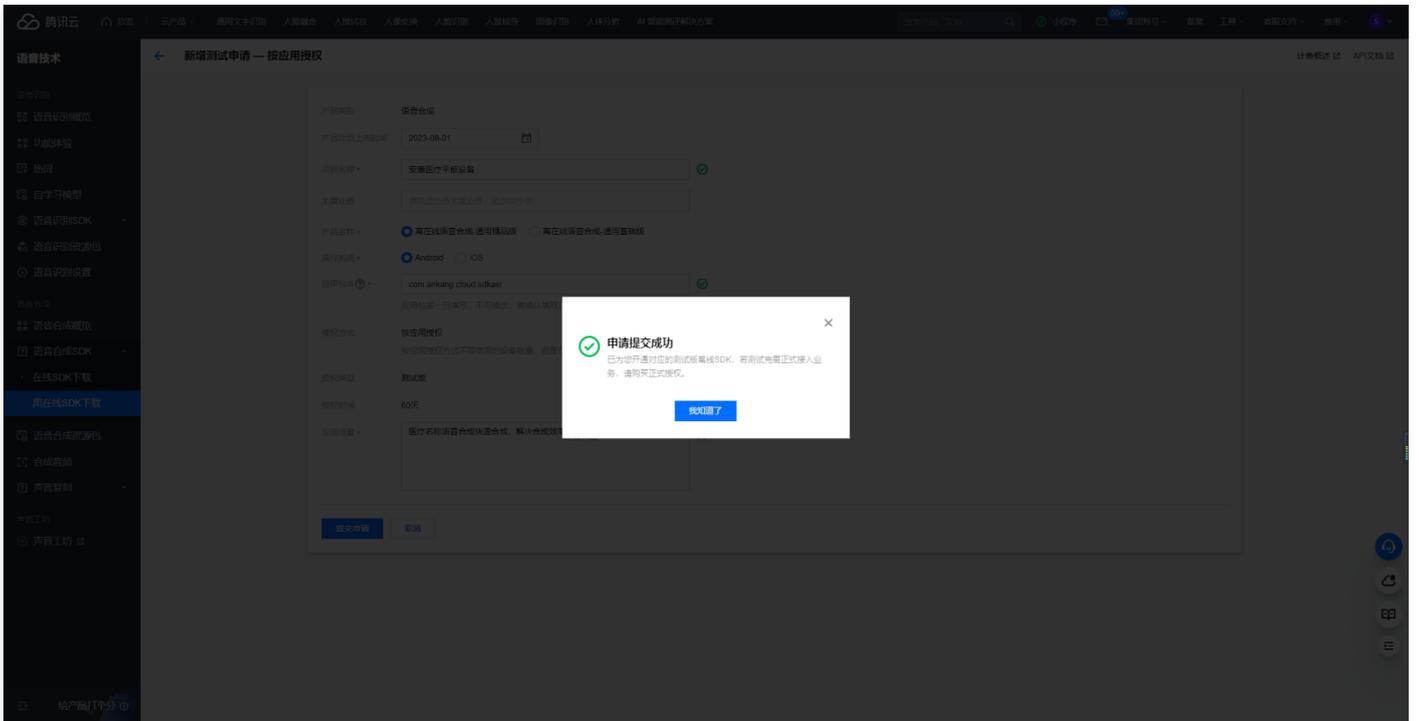
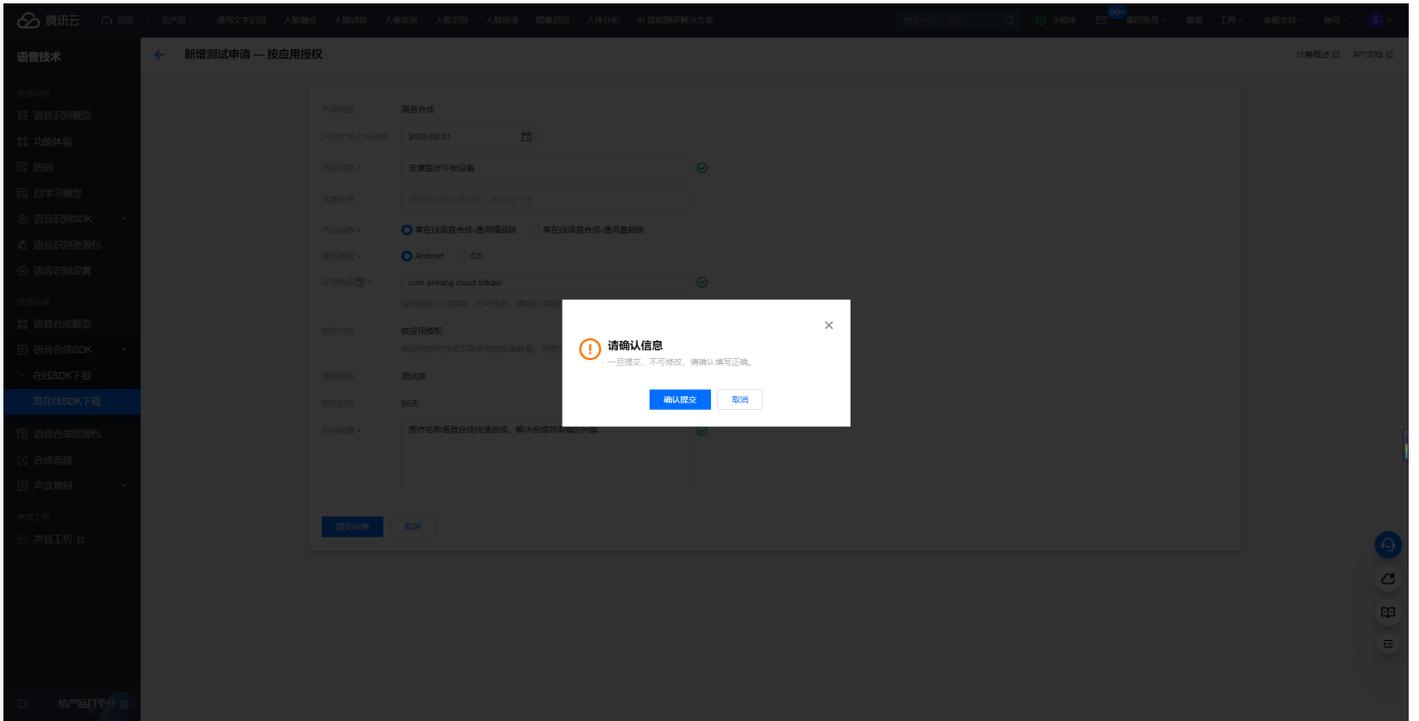


4. 在申请页面填写SDK测试所需要的信息，填写应用包名命名方式建议参考以下（注意：按设备授权直接选择数量）；

Android-SDK 应用包名：com.tencent.cloud.sdkttts

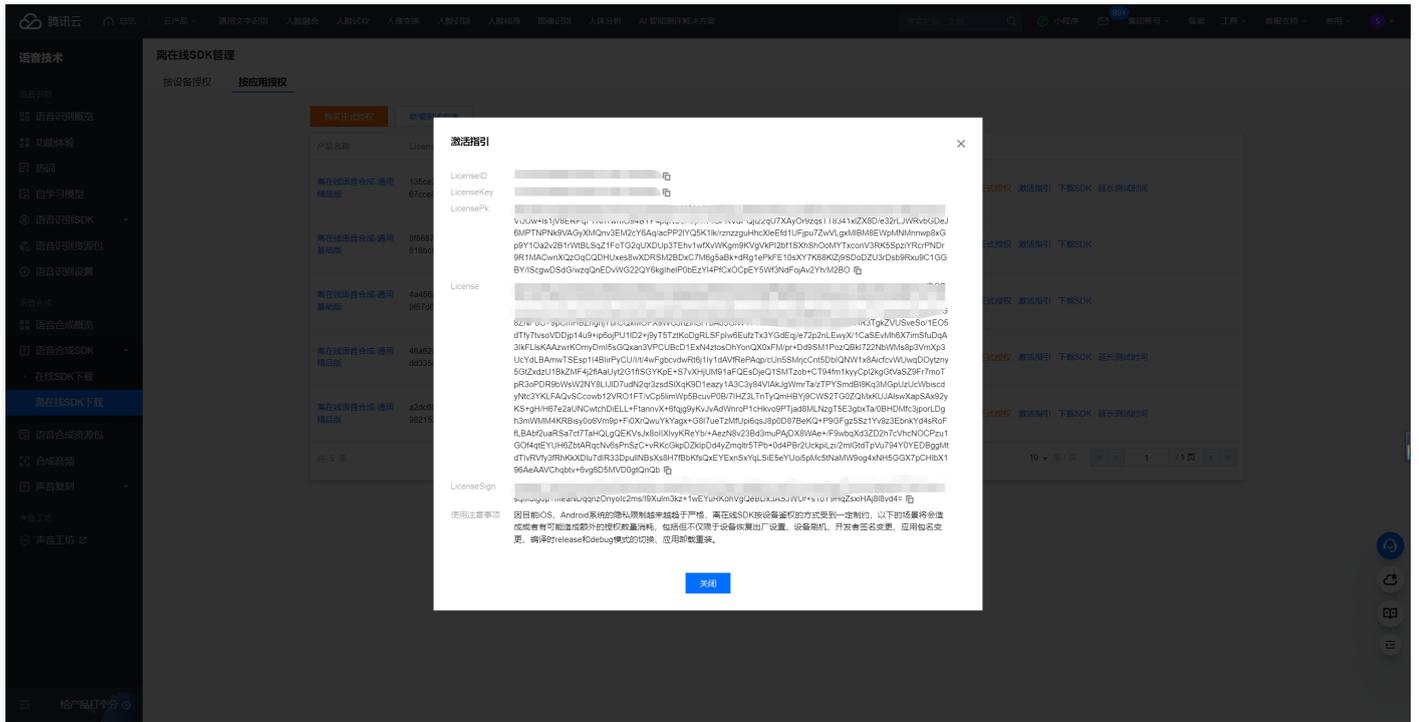
iOS-SDK 应用包名：com.tencent.cloud.iai.sdkttts







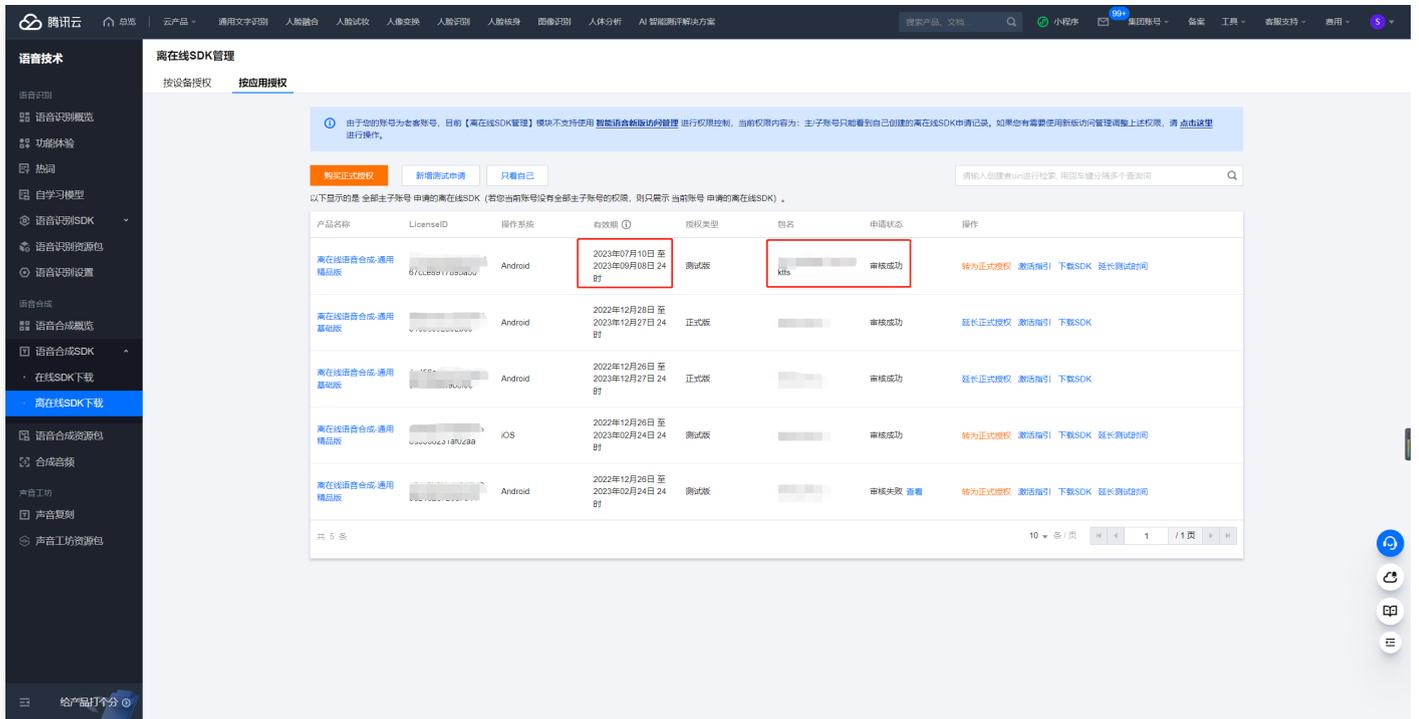
### 2. 单击激活指引;

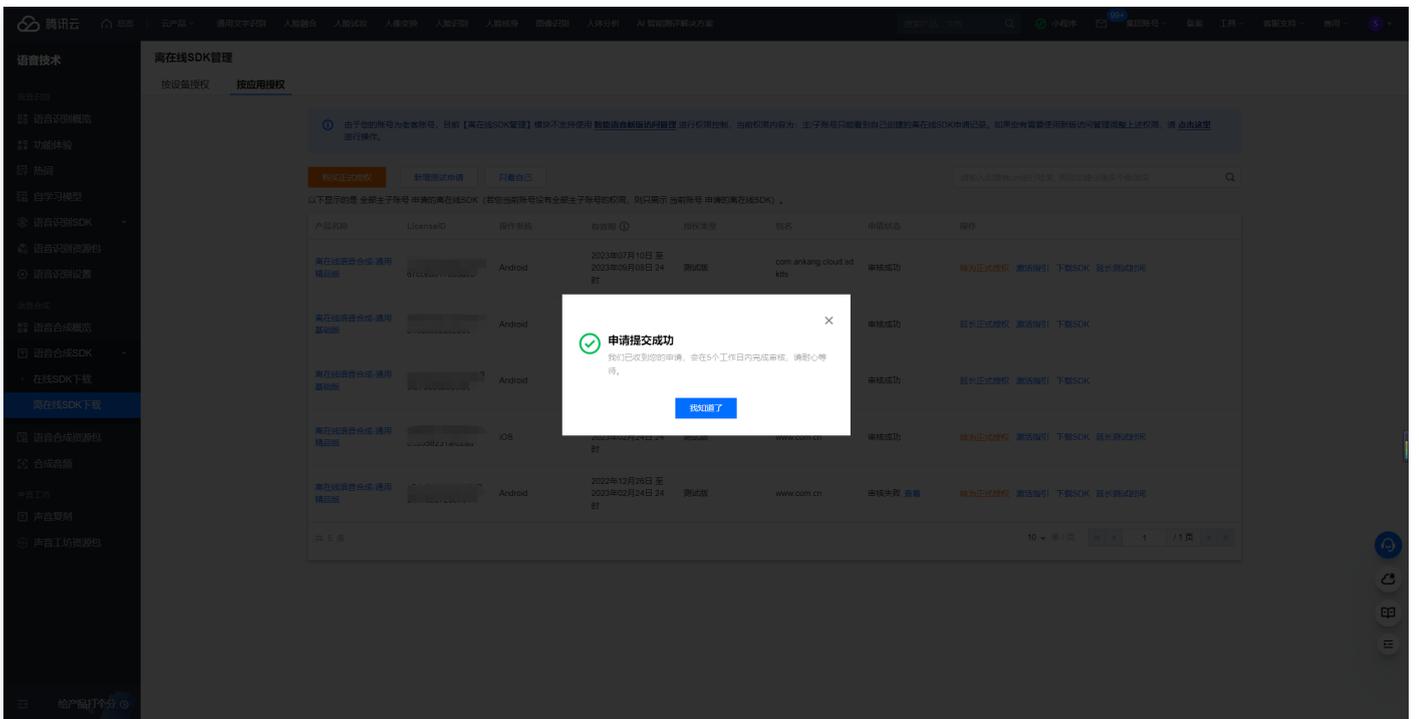
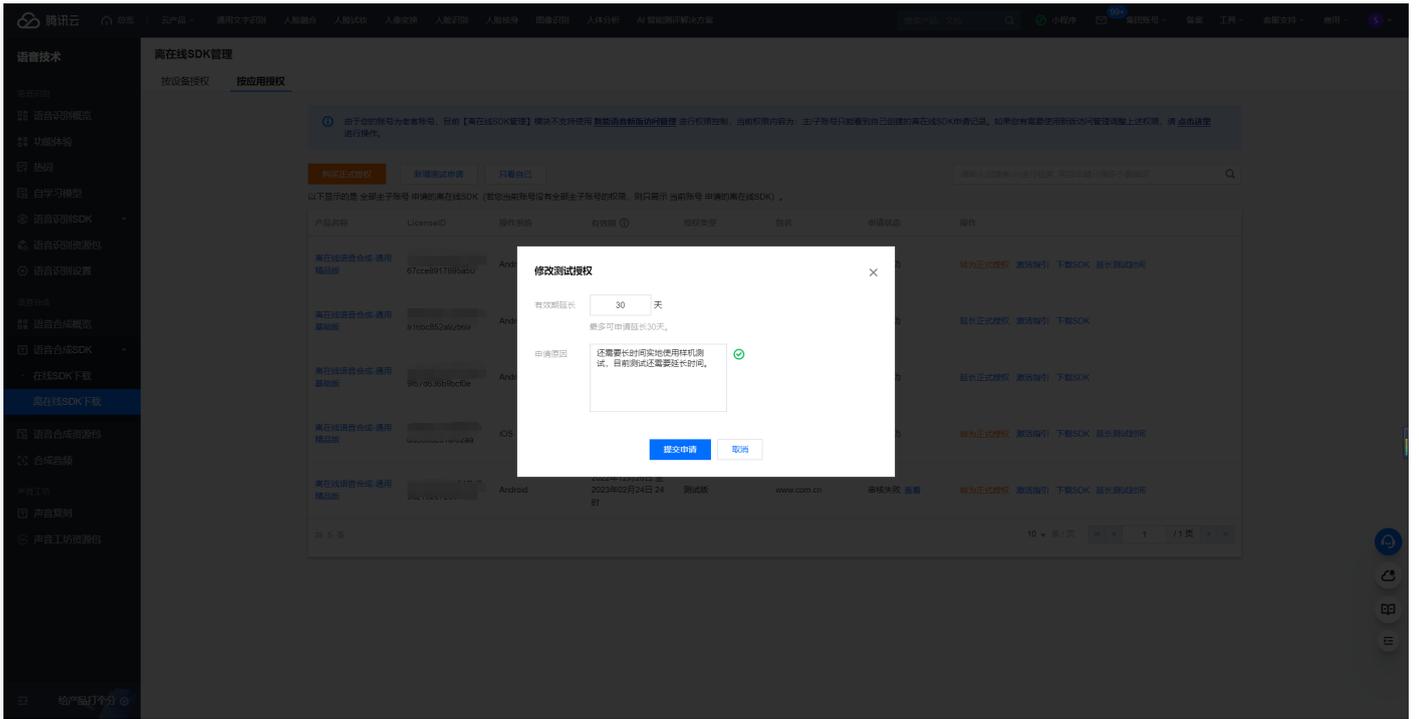


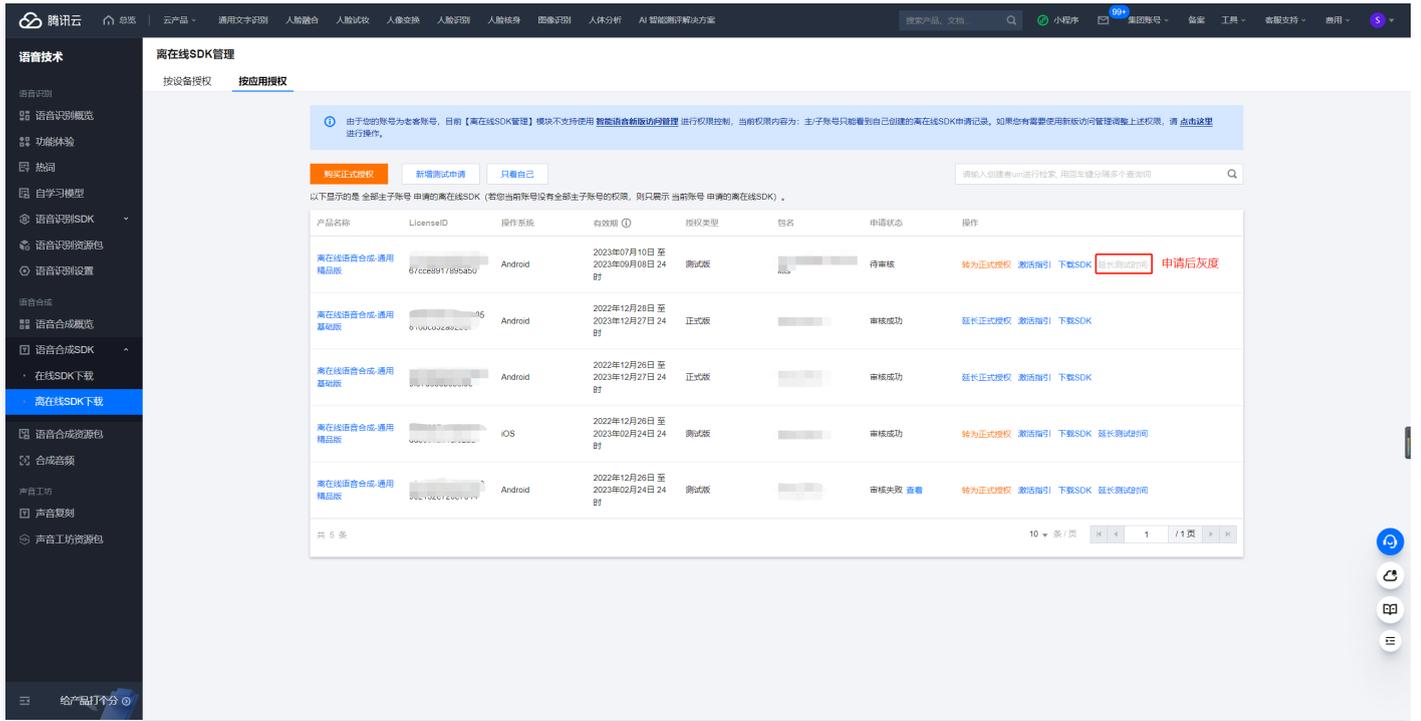
### 3. 按照下载 SDK压缩包内文档指引，集成 SDK 到 App 后进行密钥输入进行激活。

## 延长测试时间

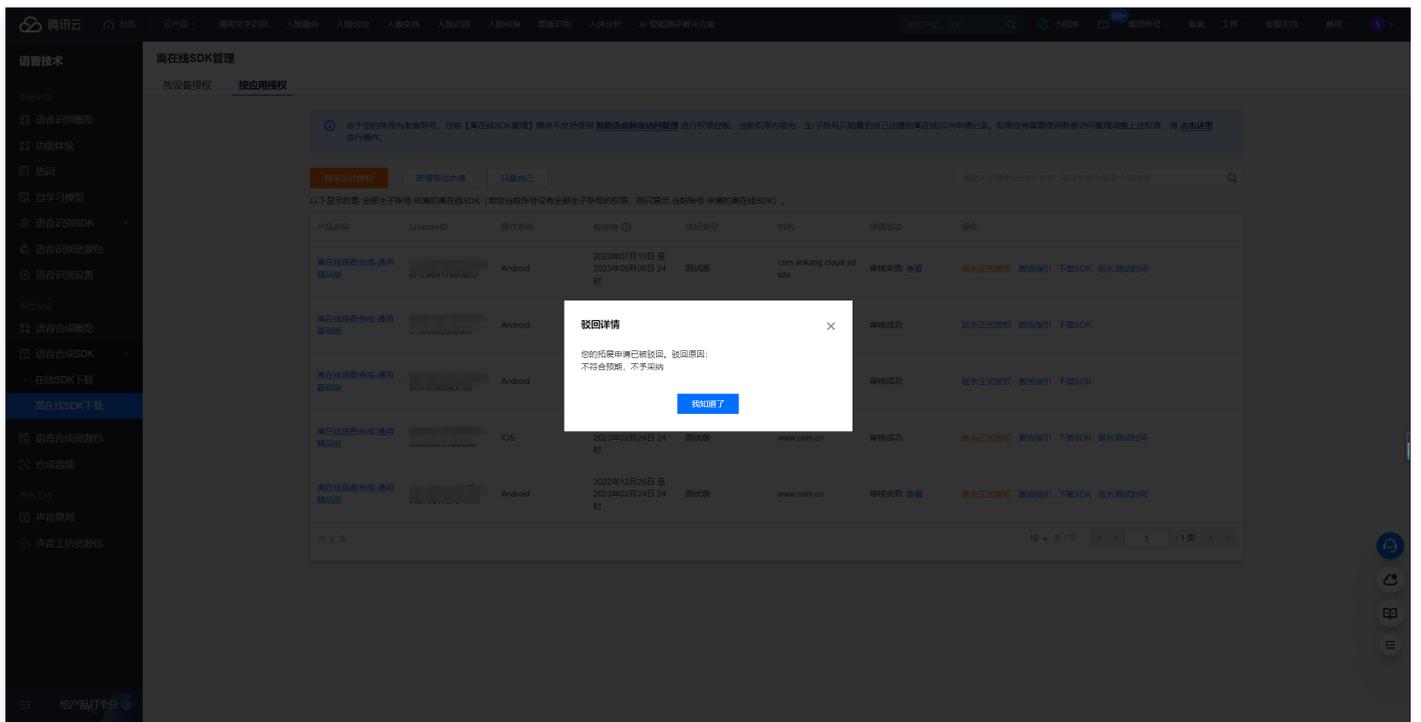
#### 1. 按应用授权单击延长测试时间，或按设备授权单击修改测试授权；





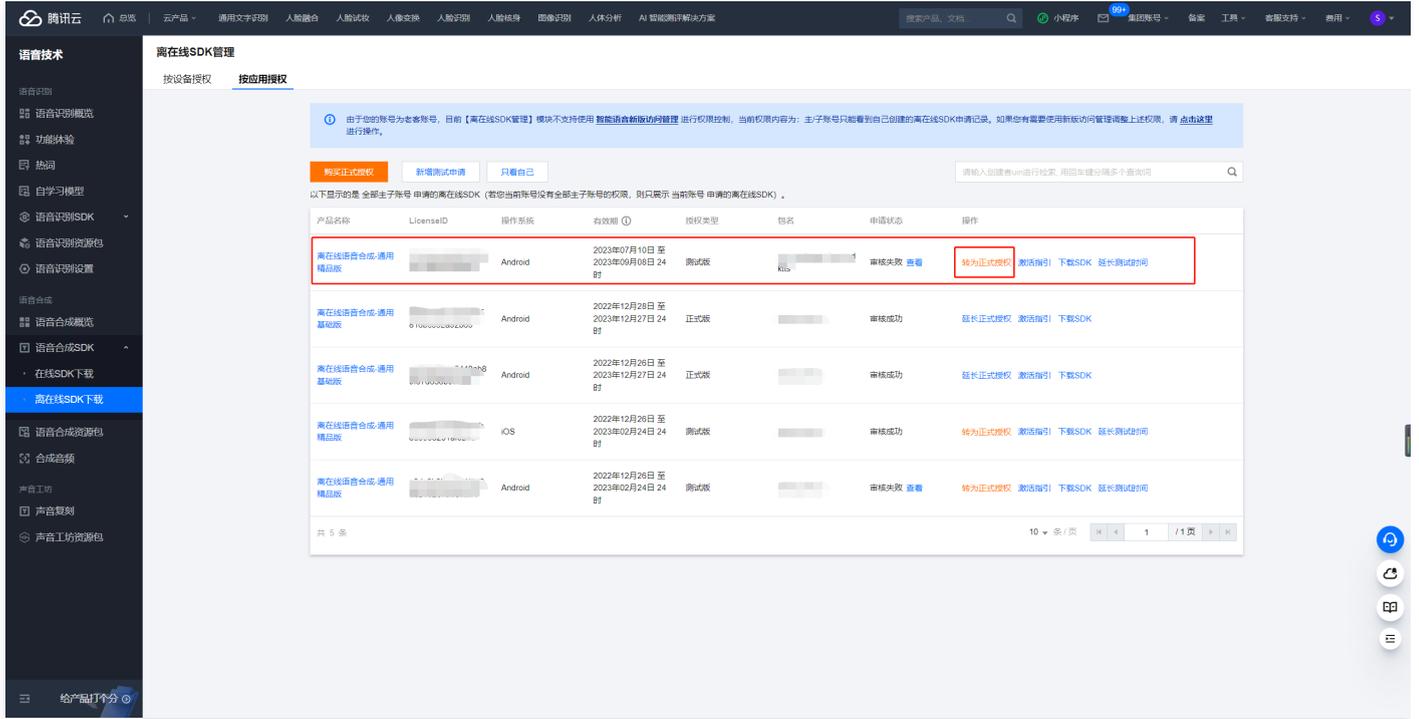


2. 待审核通过后，可延长测试时间，若审核失败，单击查看原因；

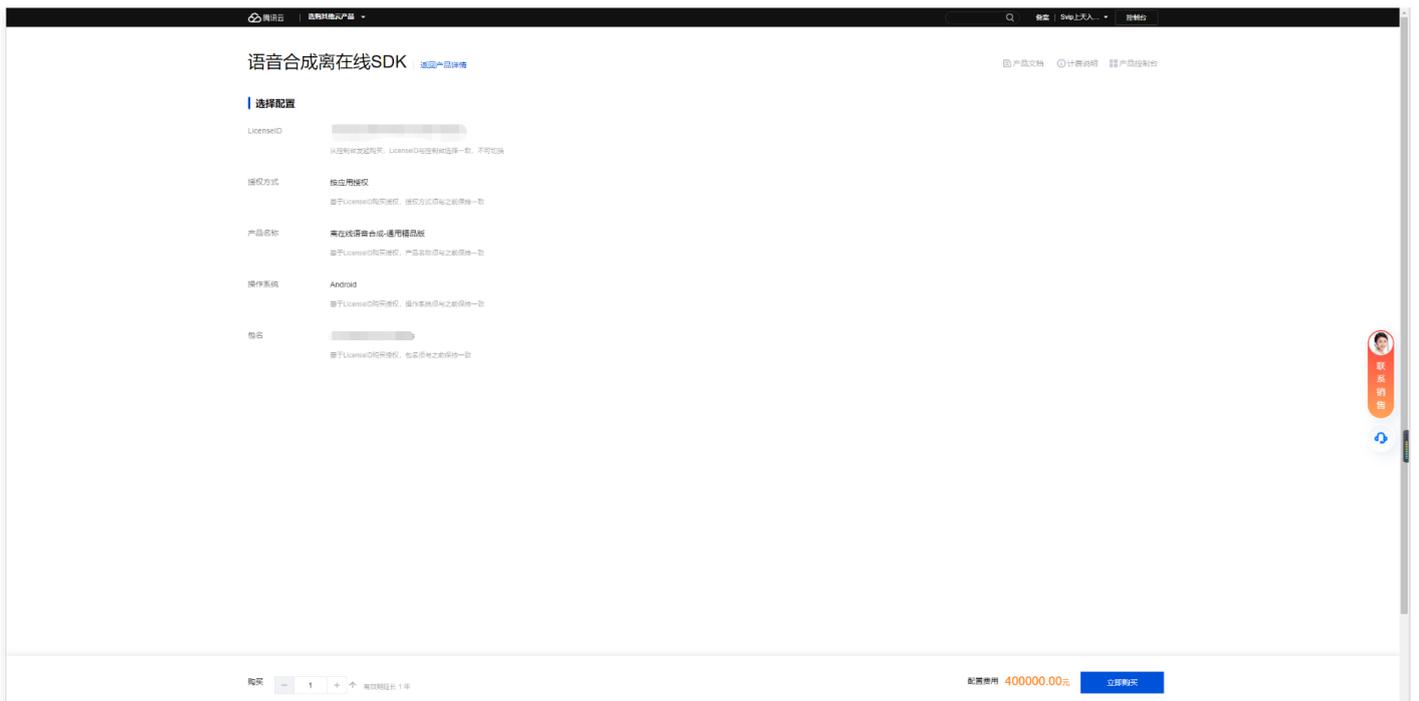


## 转为正式授权

1. 在**离线 SDK 管理**页单击**转为正式授权**跳到**购买页**（如不需要测试、或不需要转为正式授权，可直接单击**购买正式授权**，翻到页后看**购买正式授权流程**）；



2. 购买页选择**购买XX个**，提交**订单并支付成功**，单击**进入控制台**；



腾讯云 Q 帮助 | 登录 | 购买 | 续费 | 续费 | 续费

### 确认产品信息 [返回修改配置](#)

**下单说明** 请从产品信息后提交订单。如有优惠券可在支付时选择使用。最终支付金额以支付订单时为准。

#### 产品清单

预付费产品 (1)							应付合计	400000.00元
产品名称	配置	类型	单价	数量	时长	订单金额		
标准语音合成-离线在线SDK	LicenseID: [redacted] 授权方式: 按位授权 <a href="#">展开</a>	新购	400000.00元	x1	一次性购买	400000.00元		

#### 选择优惠券

##### 代金券

使用代金券抵扣 **0.00元** + 兑换  
暂无可用代金券

实付金额 **400000.00元** [无支付](#)



腾讯云 Q 帮助 | 登录 | 购买 | 续费 | 续费 | 续费

### 确认产品信息 [返回修改配置](#)

**下单说明** 请从产品信息后提交订单。如有优惠券可在支付时选择使用。最终支付金额以支付订单时为准。

#### 产品清单

预付费产品 (1)							应付合计	400000.00元
产品名称	配置	类型	单价	数量	时长	订单金额		
标准语音合成-离线在线SDK	LicenseID: [redacted] 授权方式: 按位授权 <a href="#">展开</a>	新购	400000.00元	x1	一次性购买	400000.00元		

#### 选择优惠券

##### 代金券

使用代金券抵扣 **0.00元** + 兑换  
暂无可用代金券

#### 选择支付方式

**余额支付** 支付: 0.00元

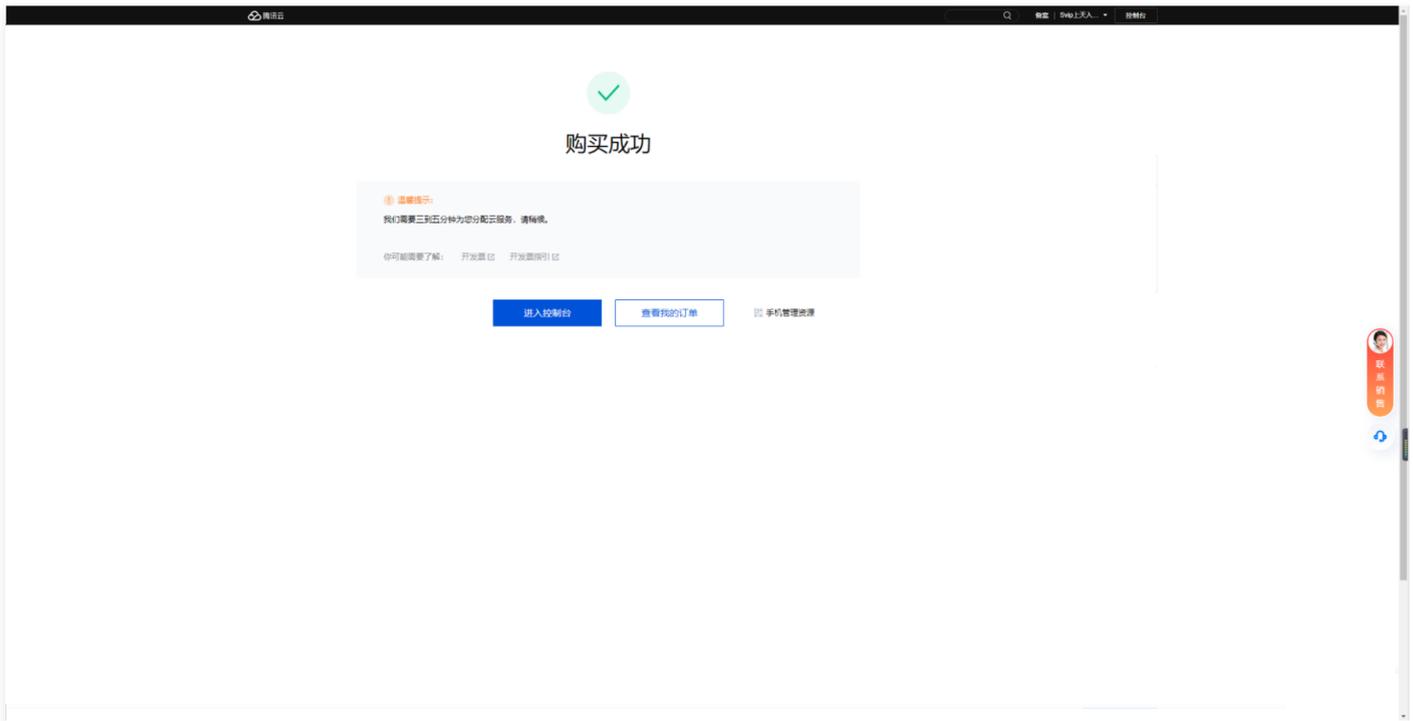
账户可用余额: [redacted] 余额不足不支持  
如需充值请点击左侧“充值”按钮或联系客服。

**在线支付** 支付: 0.00元

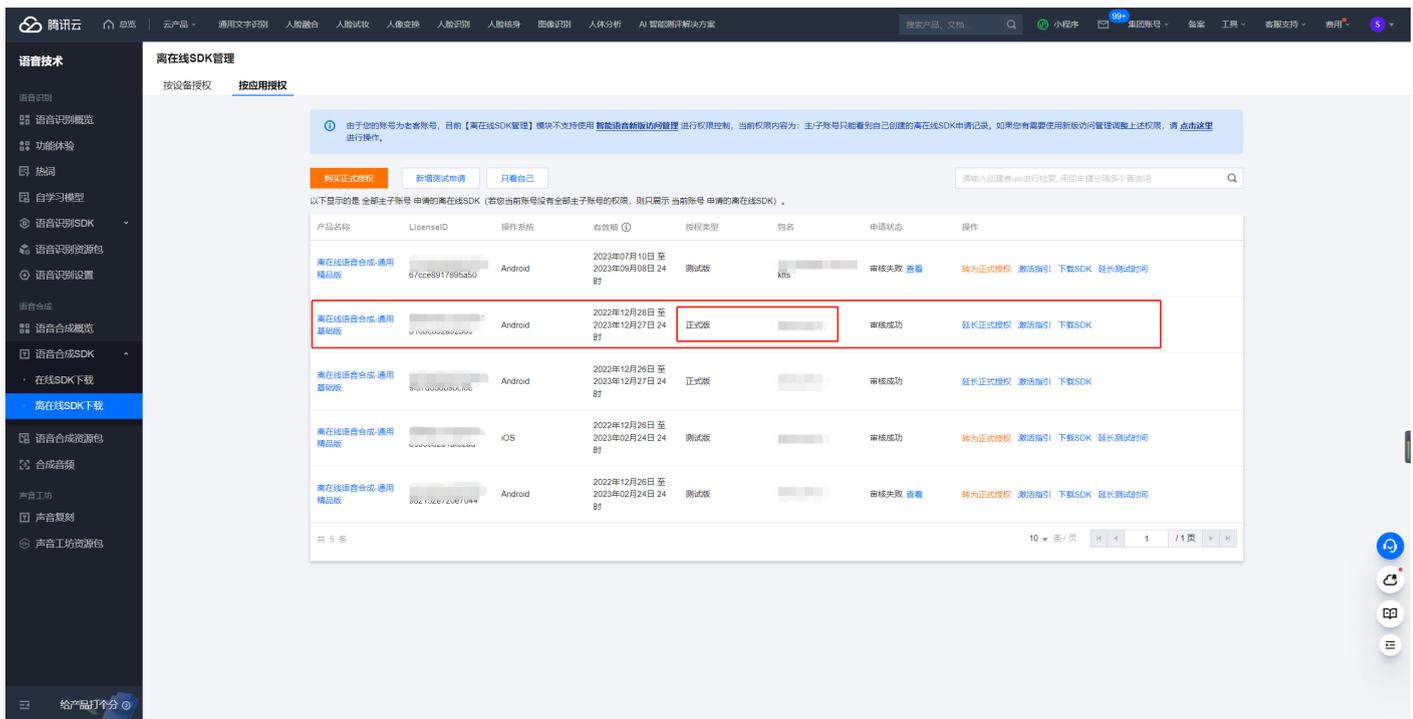
支持 多种支付方式  
如需使用网银信用卡支付，请在微信支付后添加信用卡支付。[查看详情不支持银行](#)

实付金额 **400000.00元** [无支付](#)

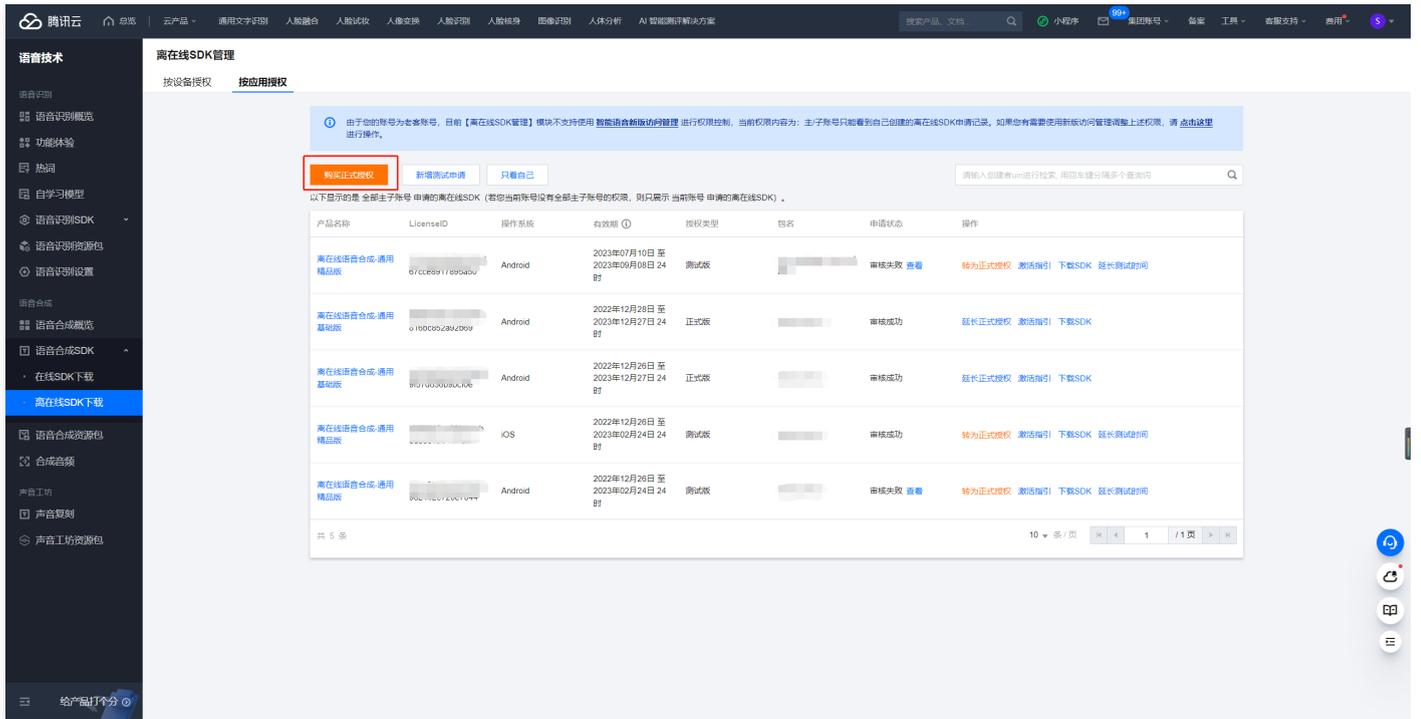




### 3. 查看按应用授权页面，确定转为正式授权购买成功。



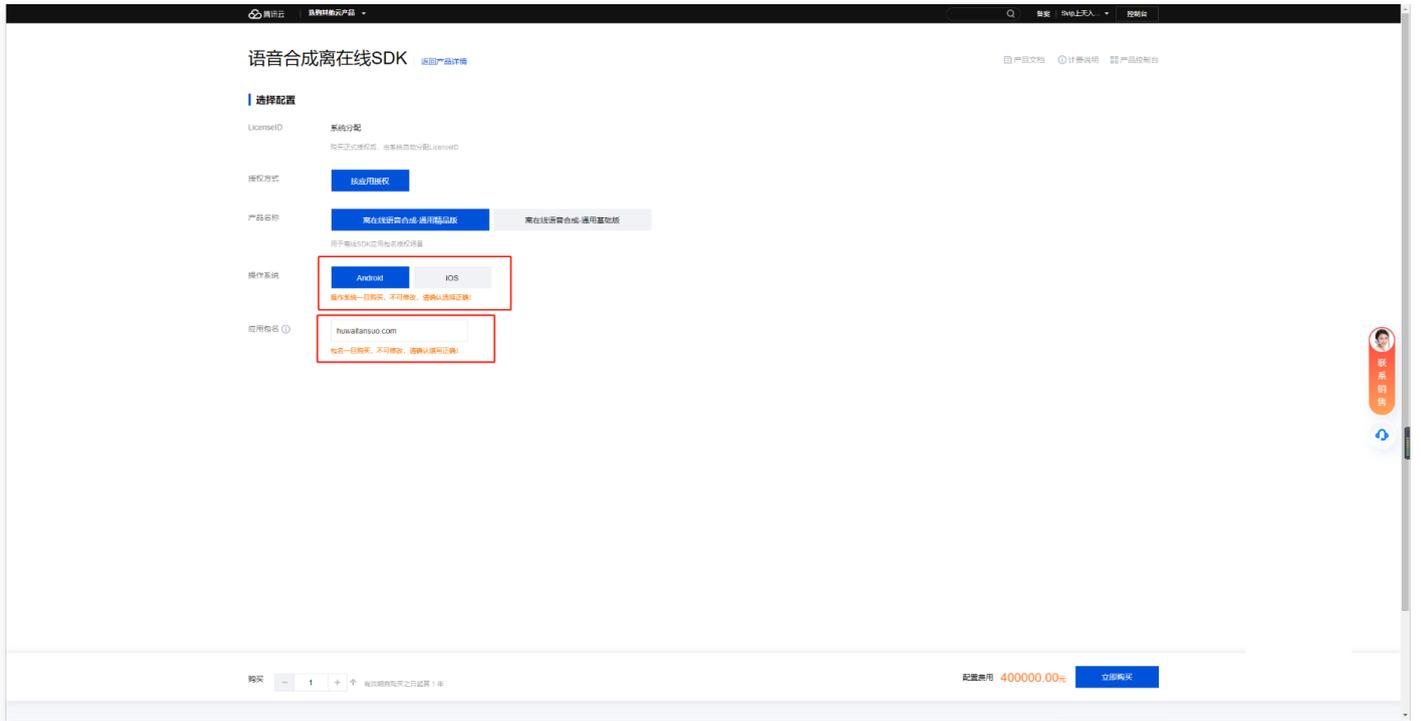
1. 在离在线 SDK 管理页，单击按应用授权，再单击购买正式授权跳到购买页；



2. 选择购买XX个，填写应用包名命名方式建议参考以下（按设备授权直接选择数量），提交订单并支付成功，单击进入控制台；

Android-SDK 应用包名: com.tencent.cloud.sdkttts

iOS-SDK 应用包名: com.tencent.cloud.iai.sdkttts



腾讯云
帮助 0元上云 购物车

## 确认产品信息 [返回修改配置](#)

[下载说明](#) 请确认产品信息后提交订单，如有优惠券可在支付时选择使用，**最终支付金额以支付订单时为准。**

### 产品清单

预付费产品 (1)							应付合计	400000.00元
产品名称	配置	类型	单价	数量	时长	订单金额		
标准语音合成-离线控制SDK	LicenseID: 系统分配 授权方式: 按应得版权 购买应用数量: 1个 <a href="#">展开</a>	新购	400000.00元	x1	一次性购买	400000.00元		

### 选择优惠券

**代金券**

使用代金券抵扣 0.00元 + 兑换

暂无可用代金券

应付金额 **400000.00元** 去支付

联系客服

腾讯云
帮助 0元上云 购物车

## 支付订单 帮助文档 订单管理

[重要说明](#) 所有消费（包括购买、开通、续费）均可开票，订单支付并开发票成功后，可前往 [费用中心](#) > [发票管理](#) 开票。

### 待支付订单

预付费订单 (1)							应付合计	400000.00元
订单号	产品名称	配置	类型	数量	时长	订单金额		
2023091122038918844601	标准语音合成-离线控制SDK	LicenseID: 系统分配 <a href="#">授权方式...详情</a>	新购	x1	一次性购买	400000.00元		

### 选择优惠券

**代金券**

使用代金券抵扣 0.00元 + 兑换

暂无可用代金券

### 选择支付方式

**余额支付** 支付: 289958.80元

您当前可用余额 178779.13元，余额不足支付。  
 如需使用余额支付，请在购物车勾选余额支付，或前往[我的钱包](#)充值。

**在线支付** 支付: 9.00元

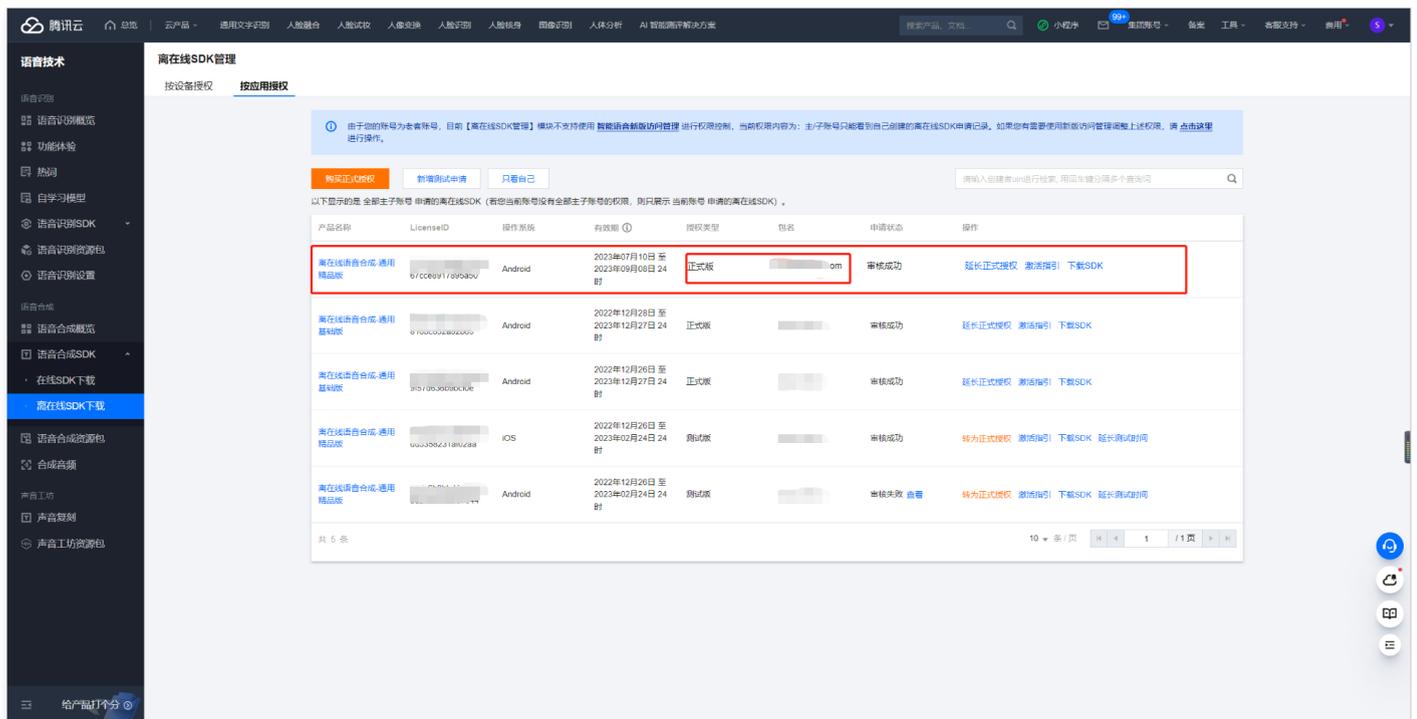
支持 多种支付方式。  
 如需使用银行卡支付，请在购物车勾选银行卡支付，或前往[我的钱包](#)充值。

余额支付
应付金额 **400000.00元**
确认支付

联系客服



### 3. 查看按应用授权页面，确定购买正式授权购买成功。



# Android SDK 精品

Last updated: 2024-11-15 16:01:02

## 说明:

- 当前页面为 v2.x.x 版本 SDK 开发文档。
- 离线 SDK 包含在线 SDK 功能，在线部分请参考 v2.x.x 版本在线 SDK 开发文档。

## 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。
- 按设备授权方式需要手机能够连接网络（3/4/5G 或 Wi-Fi 网络等）。
- 建议使用 Android Studio 进行开发。

### 下载安装 SDK

- 请参考 [开通授权](#)，根据指引开通并下载 SDK 文件。
- 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 打开工程查看语音合成示例代码。

### 环境配置

添加实时语音合成 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: 'qcloudtts-release', ext: 'aar')
```

```
<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

```
mTtsController.init(Context context, TtsMode mode, TtsResultListener listener)
```

### TTS合成器工作模式参数

如果您下载的压缩包为"xxx\_offline.zip"，则支持以下三种模式，如果为"xxx\_online.zip"，则仅支持在线模式。

参数	说明
TtsMode.OFFLINE	离线
TtsMode.ONLINE	在线
TtsMode.MIX	混合

**在线模式参数配置(不使用则不需要配置)****TtsController 配置参数方法**

接口	说明
setAppId(long l)	配置腾讯云 AppId
setSecretId(String s)	配置腾讯云 SecretId
setSecretKey(String s)	配置腾讯云 SecretKey
setToken(String s)	STS 临时证书鉴权时需要设置 Token
setOnlineVoiceSpeed(float f)	设置在线所合成音频的语速，范围：[-2, 2]，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 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
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 <a href="#">基础语音合成</a>
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，默认15000ms(15s)，范围[500,30000] 单位ms，Mix 模式下建议调小此值，以获得更好的体验
setReadTimeout(int i)	读取超时，默认30000ms(30s)，范围[2200,60000] 单位ms，Mix 模式下建议调小此值，以获得更好的体验

**示例：**

```

mTtsController.setAppId(0L);
mTtsController.setSecretId("*****");
mTtsController.setSecretKey("*****");
mttsController.setToken(null);
mTtsController.setOnlineVoiceSpeed(mVoiceSpeed);
mTtsController.setOnlineVoiceVolume(1.0);
mTtsController.setOnlineVoiceType(1001);
mTtsController.setOnlineVoiceLanguage(1);
mTtsController.setOnlineCodec("mp3");
mTtsController.setConnectTimeout(15 * 1000);
mTtsController.setReadTimeout(30 * 1000);

```

**离线模式参数配置（不使用则不需要配置）****TtsController 配置参数方法**● **按设备授权**

接口	说明
refreshAuth	是否强制联网刷新授权（false：仅第一次联网激活下载授权文件，true：联网刷新授权文件，无网络下将激活失败）
secret_id	腾讯云 SecretId（可能与在线模式不是同一个账号，需要输入购买离线 SDK 对应的账号的 SecretId）
secret_key	腾讯云 SecretKey（可能与在线模式不是同一个账号，需要输入购买离线 SDK 对应的账号的 SecretKey）
lic_key	密钥对应的 licKey，请在腾讯云官网页面获取，或者由腾讯云商务线下下发
lic_pk	密钥对应的 licPk，请在腾讯云官网页面获取，或者由腾讯云商务线下下发

## 示例:

```
String secret_id = "AKID*****9SXO";
String secret_key = "f5*****05xa";
String lic_key = "AI8*****064";
String lic_pk = "hg9*****2BO";
mTtsController.setOfflineAuthParamDoOnline(true, secret_id, secret_key, lic_key, lic_pk);
```

## ● 按应用授权

接口	说明
lic	授权密钥，请在腾讯云官网页面获取，或者由腾讯云商务线下载
lic_sign	该密钥对应的 licPk，请在腾讯云官网页面获取，或者由腾讯云商务线下载
lic_pk	该密钥对应的 licSign，请在腾讯云官网页面获取，或者由腾讯云商务线下载

## 示例:

```
String lic = "PLK*****YEoe01EEUoTDH/R/mk=";
String lic_sign = "PoTpR*****g0ucFkOQOU=";
String lic_pk = "hg9*****2BO";
mTtsController.setOfflineAuthParamDoOffline(lic, lic_sign, lic_pk);
```

## TtsController 配置参数方法

接口	说明
setOfflineVoiceVolume(float voiceVolume)	离线音量 > 0
setOfflineVoiceSpeed(float voiceSpeed)	离线语速[0.5,2.0]
setOfflineVoiceType(String s)	<p>离线音色名称，名称配置位于音色资源目录\voices\config.json 中，可自行删除或增加音色，音色需要授权，对应音色如未授权合成时会报错，可联系腾讯云商务购买。</p> <p>默认如下(离线音色名称 - 音色名称)</p> <p>M206 - 智皓</p> <p>db1 - 智聆</p> <p>db3 - 智萌</p> <p>db7 - 智甜</p> <p>f0 - 智莉</p> <p>f2 - 智芸</p> <p>femozhifou - 智蓉</p> <p>fn - 智瑜</p> <p>kefu - 智美</p> <p>kefu2 - 智娜</p> <p>kefu3 - 智琪</p> <p>m0 - 智云</p> <p>m25 - 智华</p> <p>memozhifou - 智靖</p> <p>newsman - 智宁</p> <p>pb - 智逍遥</p> <p>xiaowei - 智言</p>
setOfflineResourceDir(String resourceDir)	设置离线模型资源所在路径

## 示例

```
mTtsController.setOfflineResourceDir("/sdcard/res");
```

```
mTtsController.setOfflineVoiceSpeed(1.0f);
mTtsController.setOfflineVoiceVolume(1.0f);
mTtsController.setOfflineVoiceType("pb");
```

### 混合模式参数配置（不使用则不需要配置）

混合模式提供一种策略会在离线合成和在线合成之间进行切换。

在线模式下根据网络情况会切换自动离线模式（SDK判断网络较差时会切换），离线模式下每隔一段时间会尝试切回在线模式。

### TtsController 授权配置

混合模式需要在线和离线均有授权,授权配置参照在线模式与离线模式的方法（两个都需要配置）。

### TtsController 配置参数方法

在线模式配置和离线模式配置会影响混合模式下处于相应状态时合成行为。

以下为混合模式特有配置会影响策略的选择：

接口	说明
setCheckNetworkIntervalTime(int i)	Mix 模式下，已经连接网络，但出现网络错误或者后台错误后的检测间隔时间，用于从离线模式自动切换回在线模式，默认值5分钟，单位为秒

示例：

```
mTtsController.setCheckNetworkIntervalTime(5 * 60);
```

### 合成文本入参接口

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

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;
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();
```

### 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

### 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）

### onOfflineAuthInfo()方法签名说明

参数	说明
offlineAuthInfo.getExpireTime() ( )	授权到期时间
offlineAuthInfo.getError().get Code()	0为授权成功，其他为失败
offlineAuthInfo.getResponse( )	使用在线拉取授权文件的方式鉴权时，服务器返回的 Response 数据（仅授权失败时需要关注）

### 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
        //注：实际业务上判断一下，如果是混合模式下返回在线合成的后端错误码，应当忽略，可不作处理，SDK内会调用离线合成继续工作
    }

    @Override
    public void onOfflineAuthInfo(QCloudOfflineAuthInfo offlineAuthInfo) {
        //offlineAuthInfo 返回离线sdk授权信息，包含错误码、到期时间、当前设备ID
        //注意，如果使用离线模式，需要收到此回调后再调用合成接口，否则可能会因授权失败导致合成失败!!!
    }

    @Override
    public void onChunk(ByteBuffer chunk) {
        // 离线合成返回的流式数据，格式为16k 单通道pcm
    }
}
```

### 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 内播放器一致。

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

    @Override
    public void onTTSPlayStart() {
        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

接口	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);
```

Last updated: 2024-11-15 16:01:02

**说明:**

- 当前页面为 v2.x.x 版本 SDK 开发文档。
- 离在线 SDK 包含在线 SDK 功能,在线部分请参考 v2.x.x 版本在线 SDK 开发文档。

## 开发相关

### 开发准备

- 按设备授权方式需要手机能够连接网络（3/4/5G 或 Wi-Fi 网络等）。
- 不支持 iOS 模拟器。
- 不支持 iOS11.0 以下系统。

### 下载安装SDK

- 请参考 [开通授权](#)，根据指引开通并下载 SDK 文件。

### 导入 SDK

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。

### 配置隐私协议

根据 [苹果隐私政策](#) 规定，集成 SDK 的应用需要在 Xcode 项目的 PrivacyInfo.xcprivacy 中补全条款，若项目中没有，需要根据 [官方说明](#) 使用 Xcode 15 及以上的版本，新建一个 PrivacyInfo.xcprivacy 文件，参考以下方式，添加 SDK 依赖的隐私条款：

1. 在 Xcode 中选择 PrivacyInfo.xcprivacy。
2. 将 Privacy Accessed API Types 条款添加到 PrivacyInfo.xcprivacy 中，具体配置如下图：

▼ App Privacy Configuration	Dictionary	(1 item)	
▼ Privacy Accessed API Types	◇ Array	(2 items)	
▼ Item 0	Dictionary	(2 items)	
▼ Privacy Accessed API Reasons	◇ Array	(1 item)	
Item 0	String	35F9.1: Measure time on-device, per documentation	◇
Privacy Accessed API Type	◇ String	System Boot Time	◇
▼ Item 1	Dictionary	(2 items)	
▼ Privacy Accessed API Reasons	◇ Array	(1 item)	
Item 0	String	E174.1: Write or delete file on-device, per documentation	◇
Privacy Accessed API Type	◇ String	File Timestamp	◇

### 接口说明

#### 获得 TTS 合成器实例

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

//销毁实例
[QCloudTTSEngine instanceRelease];
```

#### 初始化引擎

```
-(void) engineInit:(TtsMode)mode Delegate:(id<QCloudTTSEngineDelegate> _Nonnull) delegate;
```

**TtsMode:** TTS合成器工作模式参数（如果您下载的压缩包为"xxx\_mixed\_\*.zip"，则支持以下三种模式，如果为"xxx\_online.zip"，则仅支持在线模式）

枚举	说明
TTS_MODE_OFFLINE	离线

TTS_MODE_ONLINE	在线
TTS_MODE_MIX	混合

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

### 在线模式参数配置(不使用则不需要配置)

#### QCloudTTSEngine 配置参数方法

接口	说明
setOnlineAuthParam: (NSInteger)appId SecretId: (NSString* _Nonnull)secretId SecretKey: (NSString* _Nonnull)secretKey Token: (NSString* _Nullable)token	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 )
setOnlineVoiceSpeed:(float)voiceSpeed;	设置在线所合成音频的语速,语速, 范围: [-2, 2], 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, 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。
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 列表见 <a href="#">基础语音合成</a>
setTimeoutIntervalForRequest:(int)timeout;	连接超时默认15000ms(15s) 范围[500,30000] 单位ms, Mix模式下建议调小此值, 以获得更好的体验
setTimeoutIntervalForResource:(int) timeout;	读取超时默认30000ms(30s) 范围[2200,60000] 单位ms, Mix模式下建议调小此值, 以获得更好的体验

### 示例

```
[ttsEngine setOnlineAuthParam:123 SecretId:@"AK***Hp7" SecretKey:@"D9***c2" Token:nil];
[ttsEngine setOnlineVoiceSpeed:1.0];
[ttsEngine setOnlineVoiceType:1001];
[ttsEngine setOnlineVoiceLanguage:1];
[ttsEngine setOnlineVoiceVolume:0];
[ttsEngine setTimeoutIntervalForRequest:15 *1000];
[ttsEngine setTimeoutIntervalForResource:30 *1000];
```

### 离线模式参数配置(不使用则不需要配置)

#### TtsController 离线授权配置

- 按设备授权

参数	说明
lic_pk	密钥对应的 licPk,请在腾讯云官网页面获取, 或者由腾讯云商务线下下发
lic_key	密钥对应的 licKey,请在腾讯云官网页面获取, 或者由腾讯云商务线下下发
secret_id	腾讯云 secretId (可能与在线模式不是同一个账号, 需要输入购买离线SDK对应的账号的secretId)
secret_key	腾讯云 secretKey (可能与在线模式不是同一个账号, 需要输入购买离线SDK对应的账号的secretKey)

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
RefreshAuth	是否强制联网刷新授权(NO:仅第一次联网激活下载授权文件; YES:联网刷新授权文件，无网络下将激活失败)

**示例**

```
NSString* secret_id = @"*****";
NSString* secret_key = @"*****";
NSString* lic_key = @"*****";
NSString* lic_pk = @"*****";
[tts setOfflineAuthParamDoOnline:lic_pk LicKey:lic_key SecretId:secret_id SecretKey:secret_key Token:nil RefreshAuth:NO];
```

**● 按应用授权**

接口	说明
lic	授权密钥，请在腾讯云官网页面获取，或者由腾讯云商务线下下发
lic_sign	该密钥对应的 licSign，请在腾讯云官网页面获取，或者由腾讯云商务线下下发
lic_pk	该密钥对应的 licPk，请在腾讯云官网页面获取，或者由腾讯云商务线下下发

**示例**

```
NSString* lic_pk = @"hg93NdFoJA*****v2Yh/M2B0";
NSString* lic = @"ZZ+MfsW*****HTXbt4=";
NSString* lic_sign = @"xK/7*****w6CKE=";
[tts setOfflineAuthParamDoOffline:lic LicPk:lic_pk LicSign:lic_sign];
```

**TtsController 配置参数方法**

接口	说明
setOfflineVoiceVolume:(float) voiceVolume	离线音量 > 0
setOfflineVoiceSpeed:(float) voiceSpeed	离线语速[0.5,2.0]
setOfflineResourceDir:(NSString * _Nonnull) resourceDir	设置离线模型资源所在路径
setOfflineVoiceType:(NSString * _Nonnull) voiceType	<p>离线音色名称，名称配置位于音色资源目录\voices\config.json 中，可自行删除或增加音色，音色需要授权，对应音色如未授权合成时会报错，可联系腾讯云商务购买。</p> <p>默认如下(离线音色名称 - 音色名称)</p> <ul style="list-style-type: none"> <li>M206 - 智皓</li> <li>db1 - 智聆</li> <li>db3 - 智萌</li> <li>db7 - 智甜</li> <li>f0 - 智莉</li> <li>f2 - 智芸</li> <li>femozhifou - 智蓉</li> <li>fn - 智瑜</li> <li>kefu - 智美</li> <li>kefu2 - 智娜</li> <li>kefu3 - 智琪</li> <li>m0 - 智云</li> <li>m25 - 智华</li> <li>memozhifou - 智靖</li> <li>newsman - 智宁</li> </ul>

pb - 智逍遥  
xiaowei - 智言

### 示例

```
NSString *resource_dir = [[NSBundle mainBundle] pathForResource:@"tts_resource" ofType:nil];
[ttsEngine setOfflineResourceDir:resource_dir];
[ttsEngine setOfflineVoiceType:@"pb"];
[ttsEngine setOfflineVoiceSpeed:1.0];
[ttsEngine setOfflineVoiceVolume:1.0];
```

### 混合模式参数配置(不使用则不需要配置)

- 混合模式提供一种策略会在离线合成和在线合成之间进行切换。
- 在线模式下根据网络情况会切换自动离线模式(SDK 判断网络较差时会切换),离线模式下每隔一段时间会尝试切回在线模式。

### TtsController 授权配置

混合模式需要在线和离线均有授权,授权配置参照在线模式与离线模式的方法(两个都需要配置)。

### TtsController 配置参数方法

- 在线模式配置和离线模式配置会影响混合模式下处于相应状态时合成行为。
- 以下为混合模式特有配置会影响策略的选择。

接口	说明
setCheckNetworkIntervalTime:(int) checkNetworkIntervalTime;	Mix模式下,已经连接网络,但出现网络错误或者后台错误后的检测间隔时间,用于从离线模式自动切换回在线模式,默认值5分钟

### 示例

```
[ttsEngine setCheckNetworkIntervalTime:5 * 60];
```

### 合成文本入参接口

接口	说明
(TtsError * _Nullable) synthesize:(NSString * _Nonnull)text UtteranceId:(NSString * _Nullable)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,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];
```

### 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
----	----

data	语音数据
utteranceId	语句id
text	文本
engineType	引擎类型 0:在线 1:离线

**onError 方法签名说明**

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

**onOfflineAuthInfo 方法签名说明(仅用在线模式可忽略)**

参数	说明
OfflineAuthInfo	返回离线合成模块授权信息, 使用混合或者离线模式时, 收到此方法成功回调后才可以调用合成接口, 详细说明详见QCloudOfflineAuthInfo.h

**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
    //注: 实际业务上判断一下, 如果是混合模式下返回在线合成的后端错误码, 应当忽略, 可不作处理, SDK内会调用离线合成继续工作
}

- (void)onOfflineAuthInfo:(QCloudOfflineAuthInfo * _Nonnull)OfflineAuthInfo {

    // 离线或者混合模式下, 调用初始化后必须收到此回调, 并且OfflineAuthInfo.err_code为0时, 才可以调用合成接口, 如果err_code
不为0, 请检查参数或者网络, 并重新调用engineInit

    NSString *msg = [NSString
stringWithFormat:@"OfflineAuthInfo:err_code=%d,err_msg=%@\n,deviceId=%@\n,expireTime=%@\n,respone=%@",@(Of
flineAuthInfo.err_code),OfflineAuthInfo.err_msg,OfflineAuthInfo.deviceId,OfflineAuthInfo.expireTime,Offlin
eAuthInfo.respone];

    NSLog(@"%@",msg);

}

```

**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:(QCPlayerError* _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];
```

# Android SDK 基础

Last updated: 2024-11-07 11:17:42

## 说明:

- 当前页面为 v2.x.x 版本 SDK 开发文档。
- 离线 SDK 包含在线 SDK 功能，在线部分请参考 v2.x.x 版本在线 SDK 开发文档。

## 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。
- 按设备授权方式需要手机能够连接网络（3/4/5G 或 Wi-Fi 网络等）。
- 建议使用 Android Studio 进行开发。

### 下载安装 SDK

- 请参考 [开通授权](#)，根据指引开通并下载 SDK 文件。
- 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 打开工程查看语音合成示例代码。

### 环境配置

- 添加实时语音合成 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: 'qcloudtts-release', ext: 'aar')
```

```
<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

```
mTtsController.init(Context context, TtsMode mode, TtsResultListener listener)
```

### TTS 合成器工作模式参数

如果您下载的压缩包为"xxx\_offline.zip"，则支持以下三种模式，如果为"xxx\_online.zip"，则仅支持在线模式。

参数	说明
TtsMode.OFFLINE	离线
TtsMode.ONLINE	在线
TtsMode.MIX	混合

### 在线模式参数配置(不使用则不需要配置)

## TtsController 配置参数方法

接口	说明
setAppId(long l)	配置腾讯云 AppId
setSecretId(String s)	配置腾讯云 SecretId
setSecretKey(String s)	配置腾讯云 SecretKey
setToken(String s)	STS 临时证书鉴权时需要设置 Token
setOnlineVoiceSpeed(float f)	设置在线所合成音频的语速，范围：[-2, 2]，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 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。
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 <a href="#">基础语音合成</a>
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，默认15000ms(15s)，范围[500,30000] 单位ms，Mix 模式下建议调小此值，以获得更好的体验
setReadTimeout(int i)	读取超时，默认30000ms(30s)，范围[2200,60000] 单位ms，Mix模式下建议调小此值，以获得更好的体验

### 示例

```

mTtsController.setAppId(0L);
mTtsController.setSecretId("*****");
mTtsController.setSecretKey("*****");
mTtsController.setToken(null);
mTtsController.setOnlineVoiceSpeed(mVoiceSpeed);
mTtsController.setOnlineVoiceVolume(1.0);
mTtsController.setOnlineVoiceType(1001);
mTtsController.setOnlineVoiceLanguage(1);
mTtsController.setOnlineCodec("mp3");
mTtsController.setConnectTimeout(15 * 1000);
mTtsController.setReadTimeout(30 * 1000);

```

## 离线模式参数配置（不使用则不需要配置）

### TtsController 配置参数方法

#### ● 按设备授权

接口	说明
lic_pk	密钥对应的 licPk，请在腾讯云官网页面获取，或者由腾讯云商务线下发
lic_key	密钥对应的 licKey，请在腾讯云官网页面获取，或者由腾讯云商务线下发
secret_id	腾讯云 SecretId（可能与在线模式不是同一个账号，需要输入购买离线 SDK 对应的账号的 SecretId）
secret_key	腾讯云 SecretKey（可能与在线模式不是同一个账号，需要输入购买离线 SDK 对应的账号的 SecretKey）

#### 示例：

```
String secret_id = "*****";
String secret_key = "*****";
String lic_key = "AI8*****064";
String lic_pk = "hg9*****2B0";
mTtsController.setOfflineAuthParamDoOnline(secret_id, secret_key, lic_key, lic_pk);
```

#### 按应用授权

接口	说明
lic	授权密钥，请在腾讯云官网页面获取，或者由腾讯云商务线下载
lic_sign	该密钥对应的 licPk，请在腾讯云官网页面获取，或者由腾讯云商务线下载
lic_pk	该密钥对应的 licSign，请在腾讯云官网页面获取，或者由腾讯云商务线下载

#### 示例：

```
String lic = "PLK*****YEoeO1EEUoTDH/R/mk=";
String lic_sign = "PoTpR*****g0ucFkOQOU=";
String lic_pk = "hg9*****2B0";
mTtsController.setOfflineAuthParamDoOffline(lic, lic_sign, lic_pk);
```

#### TtsController 配置参数方法

接口	说明
setOfflineVoiceVolume(float voiceVolume)	离线音量 > 0
setOfflineVoiceSpeed(float voiceSpeed)	离线语速[0.5,2.0]
setOfflineVoiceType(String s)	离线音色名称，名称配置位于音色资源目录\voices\config.json 中，demo 中仅提供"pb"、"femalen"两种
setOfflineResourceDir(String resourceDir)	设置离线模型资源所在路径

#### 示例：

```
mTtsController.setOfflineResourceDir("/sdcard/res");
mTtsController.setOfflineVoiceSpeed(1.0f);
mTtsController.setOfflineVoiceVolume(1.0f);
mTtsController.setOfflineVoiceType("pb");
```

#### 混合模式参数配置(不使用则不需要配置)

混合模式提供一种策略会在离线合成和在线合成之间进行切换。

在线模式下根据网络情况会切换自动离线模式（SDK 判断网络较差时会切换），离线模式下每隔一段时间会尝试切回在线模式。

#### TtsController 授权配置

混合模式需要在线和离线均有授权，授权配置参照在线模式与离线模式的方法（两个都需要配置）。

#### TtsController 配置参数方法

在线模式配置和离线模式配置会影响混合模式下处于相应状态时合成行为。

以下为混合模式特有配置会影响策略的选择：

接口	说明
----	----

setCheckNetworkIntervalTime(int i)	Mix 模式下，已经连接网络，但出现网络错误或者后台错误后的检测间隔时间，用于从离线模式自动切换回在线模式，默认值5分钟，单位为秒
------------------------------------	---

**示例：**

```
[ttsEngine setCheckNetworkIntervalTime:5 * 60];
```

**合成文本入参接口**

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

**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;
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();
```

**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

**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）

**onOfflineAuthInfo()方法签名说明**

参数	说明
offlineAuthInfo.getExpireTime()	授权到期时间
offlineAuthInfo.getError().getCode()	0为授权成功，其他为失败
offlineAuthInfo.getResponse()	使用在线拉取授权文件的方式鉴权时，服务器返回的 Response 数据（仅授权失败时需要关注）

## 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
        //注: 实际业务上判断一下, 如果是混合模式下返回在线合成的后端错误码, 应当忽略, 可不作处理, SDK内会调用离线合成继续工作
    }

    @Override
    public void onOfflineAuthInfo(QCloudOfflineAuthInfo offlineAuthInfo) {
        //offlineAuthInfo 返回离线sdk授权信息, 包含错误码、到期时间、当前设备ID
        //注意, 如果使用离线模式, 需要收到此回调后再调用合成接口, 否则可能会因授权失败导致合成失败!!!
    }

    @Override
    public void onChunk(ByteBuffer chunk) {
        // 离线合成返回的流式数据, 格式为16k 单通道pcm
    }

}
```

## 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 onTTSPlayStart() {
        Log.d(TAG, "开始播放");
    }

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

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

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

    @Override
    public void onTTSPlayNext(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

接口	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);

```

# iOS SDK 基础

Last updated: 2024-11-15 16:01:02

## 说明:

- 当前页面为 v2.x.x 版本 SDK 开发文档。
- 离在线 SDK 包含在线 SDK 功能,在线部分请参考 v2.x.x 版本在线 SDK 开发文档。

## 开发相关

### 开发准备

- 按设备授权方式需要手机能够连接网络（3/4/5G 或 Wi-Fi 网络等）。
- 不支持 iOS 模拟器。
- 不支持 iOS9.0 以下系统。

### 下载安装SDK

- 请参考 [开通授权](#)，根据指引开通并下载 SDK 文件。

### 导入SDK

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。

### 配置隐私协议

根据 [苹果隐私政策](#) 规定，集成 SDK 的应用需要在 Xcode 项目的 PrivacyInfo.xcprivacy 中补全条款，若项目中没有，需要根据 [官方说明](#) 使用 Xcode 15 及以上的版本，新建一个 PrivacyInfo.xcprivacy 文件，参考以下方式，添加 SDK 依赖的隐私条款：

1. 在 Xcode 中选择 PrivacyInfo.xcprivacy。
2. 将 Privacy Accessed API Types 条款添加到 PrivacyInfo.xcprivacy 中，具体配置如下图：

App Privacy Configuration	Dictionary	(1 item)
Privacy Accessed API Types	Array	(1 item)
Item 0	Dictionary	(2 items)
Privacy Accessed API Reasons	Array	(1 item)
Item 0	String	E174.1: Write or delete file on-device, per documentation
Privacy Accessed API Type	String	File Timestamp

## 接口说明

### 获得 TTS 合成器实例

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

//销毁实例
[QCloudTTSEngine instanceRelease];
```

### 初始化引擎

```
-(void) engineInit:(TtsMode)mode Delegate:(id<QCloudTTSEngineDelegate> _Nonnull) delegate;
```

**TtsMode:** TTS 合成器工作模式参数（如果您下载的压缩包为"xxx\_mixed\_\*.zip"，则支持以下三种模式，如果为"xxx\_online.zip"，则仅支持在线模式）

枚举	说明
TTS_MODE_OFFLINE	离线

TTS_MODE_ONLINE	在线
TTS_MODE_MIX	混合

**QCloudTTSDelegate:** TTS 合成器代理，用于获取合成结果。

### 在线模式参数配置(不使用则不需要配置)

#### QCloudTTSDelegate 配置参数方法

接口	说明
setOnlineAuthParam:(NSInteger)appId SecretId: (NSString* _Nonnull)secretId SecretKey: (NSString* _Nonnull)secretKey Token: (NSString* _Nullable)token	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 )
setOnlineVoiceSpeed:(float)voiceSpeed;	设置在线所合成音频的语速, 语速, 范围: [-2, 2], 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, 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。
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列表见 <a href="#">基础语音合成</a>
setTimeoutIntervalForRequest:(int)timeout;	连接超时默认15000ms(15s) 范围[500,30000] 单位ms, Mix模式下建议调小此值, 以获得更好的体验
setTimeoutIntervalForResource:(int) timeout;	读取超时默认30000ms(30s) 范围[2200,60000] 单位ms, Mix模式下建议调小此值, 以获得更好的体验

### 示例

```
[ttsEngine setOnlineAuthParam:123 SecretId:@"*****"
SecretKey:@"*****" Token:nil];
[ttsEngine setOnlineVoiceSpeed:1.0];
[ttsEngine setOnlineVoiceType:1001];
[ttsEngine setOnlineVoiceLanguage:1];
[ttsEngine setOnlineVoiceVolume:0];
[ttsEngine setTimeoutIntervalForRequest:15 *1000];
[ttsEngine setTimeoutIntervalForResource:30 *1000];
```

### 离线模式参数配置(不使用则不需要配置)

#### TtsController 离线授权配置

- 按设备授权

参数	说明
lic_pk	密钥对应的 licPk,请在腾讯云官网页面获取, 或者由腾讯云商务线下发
lic_key	密钥对应的 licKey,请在腾讯云官网页面获取, 或者由腾讯云商务线下发
secret_id	腾讯云 secretId ( 可能与在线模式不是同一个账号, 需要输入购买离线 SDK 对应的账号的 secretId )
secret_key	腾讯云 secretKey ( 可能与在线模式不是同一个账号, 需要输入购买离线 SDK 对应的账号的 secretKey )
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

## 示例

```
NSString* secret_id = @"*****";
NSString* secret_key = @"*****";
NSString* lic_key = @"*****";
[tts setOfflineAuthParamDoOnline:lic_pk LicKey:lic_key SecretId:secret_id SecretKey:secret_key Token:nil];
```

## ● 按应用授权

接口	说明
lic	授权密钥，请在腾讯云官网页面获取，或者由腾讯云商务线下载
lic_sign	该密钥对应的 licSign，请在腾讯云官网页面获取，或者由腾讯云商务线下载
lic_pk	该密钥对应的 licPk，请在腾讯云官网页面获取，或者由腾讯云商务线下载

## 示例

```
NSString* lic_pk = @"hg93NdFoJA*****v2Yh/M2B0";
NSString* lic = @"ZZ+MfsW*****HTXbt4=";
NSString* lic_sign = @"xK/7*****w6CKE=";
[tts setOfflineAuthParamDoOffline:lic LicPk:lic_pk LicSign:lic_sign];
```

## TtsController 配置参数方法

接口	说明
setOfflineVoiceVolume:(float) voiceVolume	离线音量 > 0
setOfflineVoiceSpeed:(float) voiceSpeed	离线语速[0.5,2.0]
setOfflineResourceDir:(NSString * _Nonnull) resourceDir	设置离线模型资源所在路径
setOfflineVoiceType:(NSString * _Nonnull) voiceType	离线音色名称，名称配置位于音色资源目录\voices\config.json 中，可自行指定更多的音色，demo中仅提供"pb"、"femalen"两种

## 示例

```
NSString *resource_dir = [[NSBundle mainBundle] pathForResource:@"tts_resource" ofType:nil];
[ttsEngine setOfflineResourceDir:resource_dir];
[ttsEngine setOfflineVoiceType:@"pb"];
[ttsEngine setOfflineVoiceSpeed:1.0];
[ttsEngine setOfflineVoiceVolume:1.0];
```

## 混合模式参数配置(不使用则不需要配置)

- 混合模式提供一种策略会在离线合成和在线合成之间进行切换。
- 在线模式下根据网络情况会切换自动离线模式(SDK 判断网络较差时会切换),离线模式下每隔一段时间会尝试切回在线模式。

## TtsController 授权配置

混合模式需要在线和离线均有授权,授权配置参照在线模式与离线模式的方法(两个都需要配置)。

## TtsController 配置参数方法

- 在线模式配置和离线模式配置会影响混合模式下处于相应状态时合成行为。
- 以下为混合模式特有配置会影响策略的选择。

接口	说明
----	----

```
setCheckNetworkIntervalTime:(int)
checkNetworkIntervalTime;
```

Mix模式下，已经连接网络，但出现网络错误或者后台错误后的检测间隔时间，用于从离线模式自动切换回在线模式，默认值5分钟

#### 示例

```
[ttsEngine setCheckNetworkIntervalTime:5 * 60];
```

#### 合成文本入参接口

接口	说明
(TtsError *_Nullable) synthesize: (NSString *_Nonnull)text UtteranceId: (NSString *_Nullable)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，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];
```

#### 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
data	语音数据
utteranceId	语句id
text	文本
engineType	引擎类型 0:在线 1:离线

##### onError 方法签名说明

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

##### onOfflineAuthInfo 方法签名说明(仅用在线模式可忽略)

参数	说明
OfflineAuthInfo	返回离线合成模块授权信息，使用混合或者离线模式时，收到此方法成功回调后可以调用合成接口，详细说明详见 QCloudOfflineAuthInfo.h

#### 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
    //注：实际业务上判断一下，如果是混合模式下返回在线合成的后端错误码，应当忽略，可不做处理，SDK内会调用离线合成继续工作
}

- (void) onOfflineAuthInfo: (QCloudOfflineAuthInfo * _Nonnull) OfflineAuthInfo {

    //离线或者混合模式下，调用初始化后必须收到此回调，并且OfflineAuthInfo.err_code为0时，才可以调用合成接口，如果err_code
不为0，请检查参数或者网络，并重新调用engineInit

    NSString *msg = [NSString
stringWithFormat:@"OfflineAuthInfo:err_code=%d,err_msg=%d\n,deviceId=%d\n,expireTime=%d\n,respone=%d",@(Of
flineAuthInfo.err_code),OfflineAuthInfo.err_msg,OfflineAuthInfo.deviceId,OfflineAuthInfo.expireTime,Offlin
eAuthInfo.respone];

    NSLog(@"%@",msg);
}

```

## 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) onTTSPlayStart {
    NSLog(@"onTTSPlayStart");
}

//队列所有音频播放完成，音频等待中
- (void) onTTSPlayWait {
    NSLog(@"onTTSPlayWait");
}

//恢复播放
- (void) onTTSPlayResume {
    NSLog(@"onTTSPlayResume");
}

//暂停播放
- (void) onTTSPlayPause {
    NSLog(@"onTTSPlayPause");
}

```

```

}
//播放中止
-(void)onTTSPplayStop{
    NSLog(@"onTTSPplayStop");
}
//播放器异常
-(void)onTTSPplayError:(QCPPlayerError* _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
QCPPlayerError 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];
QCPPlayerError err = [_player enqueueWithFile:url Text:text UtteranceId:utteranceId];

```

# 离在线语音合成 SDK 个人信息保护规则

Last updated: 2023-07-13 17:52:02

## 引言

离在线语音合成 SDK（以下简称“SDK 产品”）由腾讯云计算（北京）有限责任公司（以下简称“我们”）开发，公司注册地为北京市海淀区知春路49号3层西部309。

《离在线语音合成 SDK 个人信息保护规则》（以下简称“本规则”）主要向开发者及其终端用户（“终端用户”）告知，为了实现 SDK 产品的相关功能，SDK 产品需收集、使用和处理终端用户个人信息的情况。

请开发者及终端用户认真阅读本规则。如果您是开发者，请您确认充分了解并同意本规则后再集成 SDK 产品，如果您不同意本规则及按照本规则履行对应的用户个人信息保护义务，应立即停止接入及使用 SDK 产品；同时，您应仅在征得终端用户的同意后集成 SDK 产品并处理终端用户的个人信息，在获得终端用户同意前不得启用或初始化本 SDK 产品。

## 特别说明

如果您是开发者，您应当：

- 遵守法律、法规收集、使用和处理终端用户的个人信息，包括但不限于制定和公布有关个人信息保护的隐私政策等；
- 在集成 SDK 产品前，告知终端用户 SDK 产品收集、使用和处理终端用户个人信息的情况，并依法征得终端用户同意；
- 在征得终端用户的同意前，除非法律法规另有规定，不应收集任何终端用户的个人信息；
- 向终端用户提供易于操作且满足法律法规要求的用户权利实现机制，并告知终端用户如何查阅、复制、修改、删除个人信息，撤回同意，以及限制个人信息处理、转移个人信息、获取个人信息副本和注销账号；
- 遵守本规则的要求。

如开发者和终端用户对本规则内容有任何疑问或建议，可随时通过本规则第八条提供的方式与我们联系。

## 一、我们收集的信息及我们如何使用信息

### （一）为实现 SDK 产品功能所需收集的个人信息

为实现 SDK 产品的相应功能所必须，我们将向终端用户或开发者收集终端用户在使用与 SDK 产品相关的功能时产生的如下个人信息：

个人信息类型	处理目的	处理方式
AndroidID	离线模式且采用按设备授权的方式时，SDK 需要采集 Android ID 发送到设备后台，获取授权证书用于对设备的唯一性识别与校验	设备端本地加密传输处理
iOS IDFV	离线模式且采用按设备授权的方式时，SDK 需要采集 IDFV 发送到设备后台，获取授权证书用于对设备的唯一性识别与校验	设备端本地加密传输处理
文本信息	为实现 SDK 的文字转语音基础功能，需要上传文本以便转为音频	<ol style="list-style-type: none"> <li>离线模式时：本地化处理，即本 SDK 接收文本信息后在设备端转换成音频，开发者将音频调回；</li> <li>在线模式：加密传输处理；</li> <li>SDK 优先在线模式，在线模式无法适用时可切换为离线模式</li> </ol>

### （二）为实现 SDK 产品功能所需的权限

为实现 SDK 产品的相应功能所必须，我们会通过开发者的应用申请所需权限。

操作系统	权限名称	使用目的	是否可选
Android	网络状态 android.permission.ACCESS_NETWORK_STATE	用来判断使用离线还是在线模式； 在线模式时需要用于获取网络状态，用于优化当前网络通信。	必选
Android	网络访问 android.permission.INTERNET	允许程序访问网络连接，离线模式下按设备授权时向后台发送授权信息获取授权证书； 在线模式下允许程序访问网络连接，用于连接网络，进行数据传输。	必选

iOS	网络状态	用来判断使用离线还是在线模式； 在线模式时需要用于获取网络状态，用于优化当前网络通信。	必选
iOS	网络访问	允许程序访问网络连接，离线模式下按设备授权时向后台发送授权信息获取授权证书； 在线模式下允许程序访问网络连接，用于连接网络，进行数据传输。	必选

请注意，在不同设备和系统中，权限显示方式及关闭方式会有所不同，需同时参考其使用的设备及操作系统开发方的说明或指引。当终端用户关闭权限即代表其取消了相应的授权，我们和开发者将不会继续收集和使用相关权限所对应的个人信息，也无法为终端用户提供需要终端用户开启权限才能提供的对应的功能。

### （三）根据法律法规的规定，以下是征得用户同意的例外情形：

1. 为订立、履行与终端用户的合同所必需。
2. 为履行我们的法定义务所必需。
3. 为应对突发公共卫生事件，或者紧急情况下为保护终端用户的生命健康和财产安全所必需。
4. 为公共利益实施新闻报道、舆论监督等行为，在合理的范围内处理终端用户的个人信息。
5. 依照本法规定在合理的范围内处理终端用户自行公开或者其他已经合法公开的个人信息。
6. 法律行政法规规定的其他情形。

**特别提示：**如我们收集的信息无法单独或结合其他信息识别到终端用户的个人身份，其不属于法律意义上的个人信息。

## 二、信息的公开披露

我们不会将终端用户的个人信息转移给任何公司、组织和个人，但以下情况除外：

1. 事先告知终端用户转移个人信息的种类、目的、方式和范围，并征得终端用户的单独同意。
2. 如涉及合并、分立、解散、被宣告破产等原因需要转移个人信息的，我们会向终端用户告知接收方的名称或者姓名和联系方式，并要求接收方继续履行个人信息处理者的义务。接收方变更原先的处理目的、处理方式的，我们会要求接收方重新取得终端用户的同意。

我们不会公开披露终端用户的个人信息，但以下情况除外：

1. 告知终端用户公开披露的个人信息的种类、目的、方式和范围并征得终端用户的单独同意后。
2. 在法律法规、法律程序、诉讼或政府主管部门强制要求的情况下。

## 三、终端用户如何管理自己的信息

我们非常重视终端用户对其个人信息管理的权利，并竭力帮助终端用户管理个人信息，包括个人信息查询、复制、删除、注销账号以及设置隐私功能等，以保障终端用户的权利。如您是开发者，您应当为终端用户提供实现查询、复制、修改、删除个人信息、撤回同意和注销账号的方式。

基于终端用户的同意而进行的个人信息处理活动，终端用户有权撤回该同意。我们已向开发者提供关闭本 SDK 产品的能力，开发者可以通过不发对

【Android 包含：Secure.ANDROID\_ID、synthesize；iOS 包含：identifierForVendor、synthesize】接口的调用来停止收集和处理终端用户的个人信息。由于我们与终端用户无直接的交互对话界面，终端用户可以直接联系开发者停止使用本 SDK 产品，也可通过本规则第八条提供的方式与我们联系。如您是终端用户，请您理解，特定的业务功能或服务需要您提供服务所需的信息才能得以完成，当您撤回同意后，我们无法继续为您提供对应的功能或服务，也不再处理您相应的个人信息。您撤回同意的决定，不会影响我们此前基于您的授权而开展的个人信息处理。

## 四、信息的存储

### （一）存储信息的地点

我们遵守法律法规的规定，将在中华人民共和国境内收集和产生的个人信息存储在境内。

### （二）存储信息的期限

一般而言，我们仅在为实现目的所必需的最短时间内保留终端用户的个人信息，但下列情况除外：

1. 为遵守适用的法律法规等有关规定。
2. 为遵守法院判决、裁定或其他法律程序的规定。
3. 为遵守相关政府机关执法的要求。

## 五、信息安全

我们为终端用户的个人信息提供相应的安全保障，以防止信息的丢失、不当使用、未经授权访问或披露。

我们严格遵守法律法规保护终端用户的个人信息。

我们将在合理的安全水平内使用各种安全保护措施以保障信息的安全。

例如，我们使用加密技术、匿名化处理等手段来保护终端用户的个人信息。

我们建立严谨的管理制度、流程和组织确保信息安全。

例如，我们严格限制访问信息的人员范围，要求他们遵守保密义务，并进行审查。

若发生个人信息泄露等安全事件，我们会启动应急预案，阻止安全事件扩大，并以推送通知、公告等形式告知开发者。

## 六、未成年人保护

本 SDK 产品主要面向成年人。

如果您是开发者，如果终端用户是未满14周岁的未成年人（“儿童”），您应当向儿童的父母或其他监护人告知本规则，并在征得儿童的父母或其他监护人同意的前提下处理儿童个人信息。如果我们发现开发者未征得儿童监护人同意向我们提供儿童个人信息的，我们将会采取措施尽快删除。

如果您是儿童监护人，当您对您所监护儿童个人信息保护有相关疑问或权利请求时，您可以联系开发者，或通过本规则提供的方式与我们联系。

## 七、变更

我们会适时修订本规则的内容。

如本规则的修订会导致终端用户在本规则项下权利的实质减损，我们将在变更生效前，通过网站公告等方式进行告知。如您是开发者，当更新后的本规则对处理终端用户的个人信息情况有变动的，您应当适时更新隐私政策，并以弹框形式通知终端用户并且征得其同意，**如果终端用户不同意接受本规则，请停止集成 SDK 产品。**

## 八、联系我们

我们设立了专门的个人信息保护团队和个人信息保护负责人，如果开发者和/或终端用户对本规则或个人信息保护相关事宜有任何疑问或投诉、建议时，可以通过以下方式与我们联系：

1. 通过 [腾讯客服](#) 或者 [腾讯云工单/客服](#) 与我们联系或 [在线咨询](#)。
2. 将问题发送至 [Dataprivacy@tencent.com](mailto:Dataprivacy@tencent.com)。
3. 邮寄信件至：中国广东省深圳市南山区海天二路33号腾讯滨海大厦 数据隐私保护部（收）邮编：518054。

我们将尽快审核所涉问题，并在15个工作日或法律法规规定的期限内予以反馈。