

百度语音合成 SDK Android 版 开发手册 V2.0



百度在线网络技术(北京)有限公司

(版权所有, 翻版必究)

目录

目录.....	2
1 概念解释.....	4
2 简介.....	4
2.1 兼容性.....	4
2.2 开发包说明.....	4
2.3 总体框图.....	4
3 集成指南.....	5
3.1 添加 BDTTSCClient 到工程.....	5
3.2 添加语音合成资源文件.....	5
3.3 权限声明.....	5
3.4 Proguard 配置.....	6
4 语音合成功能代码.....	6
4.1 Tts 初始化与授权等参数设置.....	6
4.2 参数设置.....	6
4.3 授权验证接口.....	8
4.4 合成并播放.....	8
4.5 流程控制.....	9
4.6 状态监听.....	9
4.7 只合成不播放.....	9
4.8 批量合成播放接口.....	9
4.9 获取合成数据.....	9
4.10 切换离线发音人接口.....	9
4.11 加载离线英文模型文件接口.....	10
5 下载模型文件.....	10
5.1 根据条件查询服务器模型信息.....	10
5.2 根据条件查询本地模型信息.....	10
5.3 查询服务器可用模型信息.....	11
5.4 查询本地可用模型信息.....	11
5.5 查询服务器模型文件信息.....	12
5.6 查询本地模型文件信息.....	12
5.7 模型文件信息.....	12
5.8 获取本地前端模型文件路径.....	13
5.9 获取本地后端模型文件信息.....	13
5.10 下载模型文件.....	14
5.11 判断模型信息是否可用.....	14

5.12 判断模型文件是否可用	14
5.13 获得引擎参数	15
5.14 停止下载	15
5.15 下载监听器	16
5.15.1 下载开始回调	16
5.15.2 下载进度回调	16
5.15.3 下载结束回调	17
5.16 Conditions 条件对象	17
5.16.1 添加模型 id 查询条件	17
5.16.2 添加语言查询条件	18
5.16.3 添加性别查询条件	18
5.16.4 添加说话人查询条件	18
5.16.5 添加领域查询条件	19
5.16.6 添加质量查询条件	19
6 错误码列表	20
7 日志管理	22
FAQ	23

1 概念解释

语音合成是实现人机语音交互，建立一个有听和讲能力的交互系统所必需的关键技术。随着语音技术的发展，百度自主研发了语音合成系统（TTS），功能是接受用户发送的文本，生成语音发送给用户。

对本文中将提到的概念约定如下：

- **语音合成**：将文本合成为语音，即声音文件。
- **合成引擎**：将文本合成为语音的核心模块。
- **TTS**：Text To Speech，即“从文本到语音”。
- **BDTTSCClient**：语音合成 SDK 简称，详见下条。
- **语音合成 SDK**：即本开发包，文中简称为 BDTTSCClient。BDTTSCClient 是一个封装了网络收发、音频播放功能的语音合成解决方案。借助 BDTTSCClient 可以快速地在应用程序中集成语音合成功能。

2 简介

百度语音合成客户端 Android 版 SDK（以下简称 BDTTSCClient）是一种面向 Android 移动设备的语音合成解决方案，以 JAR 包 + 动态链接库的形式发布。目前版本已支持 SDK 内部直接播放合成语音和从 SDK 获取语音数据，并支持男女声、语速、音调、音量、音频码率设置，后续版本将支持更多参数设置（--最新信息请参见语音合成 WEB 文档--）。

2.1 兼容性

系统：支持 Android 2.3（API Level 9）及以上，需要开发者通过 minSdkVersion 来保证支持系统的检测。

机型：手机和平板皆可。

构架：支持 arm64-v8a armeabi armeabi-v7a mips x86 x86_64。

2.2 开发包说明

表 1 开发包说明表

文件(夹)名	说明
doc/Baidu_TTS_SDK_Android_Manual.pdf	本文档
doc/javaDoc	Javadoc
libs	语音合成 SDK Lib 库，包括各平台的 so 库及 Jar 包
BaiduTtsSample	开发示例(Eclipse project)
data	语音合成资源文件
license	授权文件

2.3 总体框图



图 1 语音合成系统架构图

3 集成指南

3.1 添加 BDTTSCient 到工程

将开发包中的 `libs` 目录整体拷贝到工程目录，`libs` 目录包括了各平台的 `SO` 库，开发者视应用需要可以进行删减。`galaxy_lite.jar` 是百度 Android 公共基础库，如果项目中还集成了其它百度 SDK，如 `Push SDK`，在打包过程中出现类似如下的错误信息：

```
[2013-10-22 11:02:57 - Dex Loader] Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Configuration;  
[2013-10-22 11:02:57 - VoiceRecognitionDemo] Conversion to Dalvik format failed: Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Configuration;
```

请将此 Jar 包移除。

如果 Eclipse ADT 版本插件低于 17，需要手工添加依赖库，添加方法为：Project => Properties => Java Build Path => Libraries => Add JAR...。

3.2 添加语音合成资源文件

将开发包中的 `data` 目录下的 `dat` 文件放到某一可读路径下，一遍设置资源文件参数时使用。

3.3 权限声明

BDTTSCient 需要一些权限需要在 `AndroidManifest.xml` 文件，增加如下权限：

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

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

3.4 Proguard 配置

请不要混淆 sdk 的 jar 包。

4 语音合成功能代码

4.1 Tts 初始化与授权等参数设置

```
// 获取 tts 实例
speechSynthesizer = SpeechSynthesizer.getInstance();
// 设置 app 上下文（必需参数）
speechSynthesizer.setContext(Context);
// 设置 tts 监听器
speechSynthesizer.setSpeechSynthesizerListener(SpeechSynthesizerListener);
// 文本模型文件路径，文件的绝对路径（离线引擎使用）
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_TEXT_MODEL_FILE,
TEXT_MODEL_FILE_FULL_PATH_NAME);
// 声学模型文件路径，文件的绝对路径（离线引擎使用）
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_SPEECH_MODEL_FILE,
SPEECH_MODEL_FILE_FULL_PATH_NAME);
// 本地授权文件路径，如未设置将使用默认路径。设置临时授权文件路径，
LICENCE_FILE_NAME 请替换成临时授权文件的实际路径，仅在使用临时 license 文件时需要进行
设置，如果在[应用管理]中开通了离线授权，不需要设置该参数，建议将该行代码删除（离线引擎）
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_LICENCE_FILE,
LICENSE_FILE_FULL_PATH_NAME);
// 请替换为语音开发者平台上注册应用得到的 App ID (离线授权)
speechSynthesizer.setAppId("your_app_id");
// 请替换为语音开发者平台注册应用得到的 apikey 和 secretkey (在线授权)
speechSynthesizer.setApiKey("your_api_key", "your_secret_key");
// 授权检测接口
AuthInfo authInfo = speechSynthesizer.auth(TtsMode);
// 引擎初始化接口
speechSynthesizer.initTts(TtsMode);
```

4.2 参数设置

在调用 speak 方法开始合成并朗读之前，可以对参数进行配置（未设置的参数将使用默认值），

代码如下（完整参数列表详见 JavaDoc）。

```
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_VOLUME, "5");
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_SPEED, "5");
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PITCH, "5");
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_SPEAKER, SPEAKER_FEMALE);
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_MIX_MODE,
MIX_MODE_DEFAULT);
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUDIO_ENCODE,
AUDIO_ENCODE_AMR);
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUDIO_RATE,
AUDIO_BITRATE_AMR_15K85);
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_VOCODER_OPTIM_LEVEL, "0");
```

参数默认值列表如下：

参数名	默认值	备注
PARAM_VOLUME	"5"	中级音量，范围[0-9]
PARAM_SPEED	"5"	中速，范围[0-9]
PARAM_PITCH	"5"	中调，范围[0-9]
PARAM_SPEAKER (切换在线的发音人)	SPEAKER_FEMALE	SPEAKER_FEMALE, SPEAKER_MALE, SPEAKER_MALE_HIGH, SPEAKER_MALE_EMOTION
PARAM_MIX_MODE (MIX 模式专有参数)	MIX_MODE_DEFAULT	MIX_MODE_DEFAULT(mix 模式下，wifi 使用在线合成，非 wifi 使用离线合成); MIX_MODE_HIGH_SPEED_NETWORK(mix 模式下，wifi,4G,3G 使用在线合成，其他使用离线合成); MIX_MODE_HIGH_SPEED_SYNTHESIZE(mix 模式下，在线返回速度如果慢（超时，一般为 1.2 秒）直接切换离线，适用于网络环境较差的情况)
PARAM_AUDIO_ENCODE (在线合成参数)	AUDIO_ENCODE_AMR	AUDIO_ENCODE_BV , AUDIO_ENCODE_AMR , AUDIO_ENCODE_OPUS

PARAM_AUDIO_RATE (在线合成参数)	AUDIO_BITRATE_AMR_15K85	AUDIO_BITRATE_BV_16K , AUDIO_BITRATE_AMR_6 K6 , AUDIO_BITRATE_AMR_8 K85 , AUDIO_BITRATE_AMR_1 2K65 , AUDIO_BITRATE_AMR_1 4K25 , AUDIO_BITRATE_AMR_1 5K85 , AUDIO_BITRATE_AMR_1 8K25 , AUDIO_BITRATE_AMR_1 9K85 , AUDIO_BITRATE_AMR_2 3K05 , AUDIO_BITRATE_AMR_2 3K85 , AUDIO_BITRATE_OPUS_8 K , AUDIO_BITRATE_OPUS_1 6K , AUDIO_BITRATE_OPUS_1 8K , AUDIO_BITRATE_OPUS_2 0K , AUDIO_BITRATE_OPUS_2 4K , AUDIO_BITRATE_OPUS_3 2K
PARAM_VOCODER_OPTI M_LEVEL (离线合成参数)	0	合成引擎速度优化等级，取 值范围[0, 2]，值越大速度越 快（离线引擎）

4.3 授权验证接口

```
AuthInfo authInfo = speechSynthesizer.auth(TtsMode ttsMode)
```

获取授权的结果信息。可通过 `authinfo.isSuccess()` 判断授权是成功还是失败。

4.4 合成并播放

```
speechSynthesizer.speak("百度一下");
```


该接口比较耗时，采用排队策略，调用后将自动加入合成队列，并按调用顺序进行合成和播放

```
speechSynthesizer.speak(String text, String utteranceId)
```

该接口功能和 `speak(String text)` 功能相同，参数 `utteranceId` 用来指定此次文本合成的标识，`utteranceId` 可以用数字区分（"1","2"...），或者全局唯一标识 `uuid`，总之用户可以选择自己可以用来区分的值即可。所有回调接口都会返回该标识 `utteranceId`，以方便用户做区分（不能用文本或者文本的 md5 区分是因为可能有相同文本存在）。

4.5 流程控制

通过 `pause()`, `resume()`, `stop()`, `release()` 对流程进行控制。

服务的生命周期控制方法。

4.6 状态监听

为了更好地实现用于界面，`BDTTSCClient` 提供了 `SpeechSynthesizerListener` 监听接口用于对合成器的状态进行通知，监听器的回调方法参见 `JavaDoc`。

完整的开发示例请参见开发包所附示例 eclipse project——`BaiduTtsSample`。

4.7 只合成不播放

`BDTTSCClient` 还支持仅获取合成数据，开发者可以在获取数据后先本地缓存，在合适的时候再进行播放，以避免较差网络情况下播放出现卡顿。参数设置同 4.1.2。

```
speechSynthesizer.synthesize("百度一下");
```

该接口比较耗时，采用排队策略，调用后将自动加入合成队列，并按调用顺序进行合成。

```
speechSynthesizer.synthesize(String text, String utteranceId)
```

该接口功能和 `synthesize(String text)` 功能相同，参数 `utteranceId` 用来指定此次文本合成的标识，`utteranceId` 可以用数字区分（"1","2"...），或者全局唯一标识 `uuid`，总之用户可以选择自己可以用来区分的值即可。所有回调接口都会返回该标识 `utteranceId`，以方便用户做区分（不能用文本或者文本的 md5 区分是因为可能有相同文本存在）。

4.8 批量合成播放接口

```
int batchSpeak(java.util.List<SpeechSynthesizeBag> speechSynthesizeBags)
```

该接口可以批量传入多个文本并进行排队合成并播放（如果没有设置 `utteranceId`，则使用 `list` 的索引值作为 `utteranceId`）。

4.9 获取合成数据

通过 `SpeechSynthesizerListener` 的 `onSynthesizeDataArrived` 方法获取数据，代码如下：

```
public void onSynthesizeDataArrived(String utteranceId, byte[] audioData, int progress);
```

合成数据分多次返回。

4.10 切换离线发音人接口

```
public int loadModel(String speechModelPath, String textModelPath)
```

该接口用来切换离线的发音人(发音人模型文件)。传入对应的要切换的资源文件的绝对路径即可，如果 `speechModelPath` 不同，则传入对应的路径；如果 `textModelPath` 不同，则传入对应的路径，如果相同，则传 `null` 即可。该方法必须在 `initTTS` 成功后才可调用。

4.11 加载离线英文模型文件接口

```
public int loadEnglishModel(String englishTextModelPath, String englishSpeechModelPath)
```

该接口用来支持离线的英文合成，可以实现英文单词的合成功能。传入对应的英文 `text` 模型文件路径和英文的 `speech` 模型文件路径即可。该方法必须在 `initTTS` 成功后才可调用。

5 下载模型文件

5.1 根据条件查询服务器模型信息

□ 说明

根据条件查询服务器模型信息。

□ 方法

```
public BasicHandler<ModelBags> getServerModels (Conditions conditions)
```

□ 参数

名称	描述
Conditions	查询条件

□ 返回

`BasicHandler<ModelBags>` 查询任务返回的结果，支持取消 `BasicHandler.cancel()`。

5.2 根据条件查询本地模型信息

□ 说明

根据条件查询本地模型信息。

□ 方法

```
public BasicHandler<ModelBags> getLocalModels(Conditions conditions)
```

□ 参数

名称	描述
----	----

Conditions	查询条件
------------	------

☐ 返回

BasicHandler<ModelBags> 查询任务返回的结果，支持取消 BasicHandler.cancel()。

5.3 查询服务器可用模型信息

☐ 说明

根据条件查询服务器可用的模型信息。

☐ 方法

public BasicHandler<ModelBags> getServerModelsAvailable(AvailableConditions conditions)

☐ 参数

名称	描述
AvailableConditions	查询可用条件

☐ 返回

BasicHandler<ModelBags> 查询任务返回的结果，支持取消 BasicHandler.cancel()。

5.4 查询本地可用模型信息

☐ 说明

根据条件查询本地可用的模型信息。

☐ 方法

public BasicHandler<ModelBags> getLocalModelsAvailable(AvailableConditions conditions)

☐ 参数

名称	描述
AvailableConditions	查询可用条件

☐ 返回

BasicHandler<ModelBags> 查询任务返回的结果，支持取消 BasicHandler.cancel()。

5.5 查询服务器模型文件信息

□ 说明

根据条件查询服务器的模型文件信息。

□ 方法

```
public BasicHandler<ModelBags> getServerModelFileInfos(Set<String> fileIds)
```

□ 参数

名称	描述
fileIds	需要查询的文件的 id,对应 model 信息中的 textDataId 和 speechDataId

□ 返回

BasicHandler<ModelBags> 查询任务返回的结果，支持取消 BasicHandler.cancel()。

5.6 查询本地模型文件信息

□ 说明

根据条件查询本地的模型文件信息。

□ 方法

```
public BasicHandler<ModelBags> getLocalModelFileInfos(Set<String> fileIds)
```

□ 参数

名称	描述
fileIds	需要查询的文件的 id, 对应 model 信息中的 textDataId 和 speechDataId

□ 返回

BasicHandler<ModelBags> 查询任务返回的结果，支持取消 BasicHandler.cancel()。

5.7 模型文件信息

□ 说明

根据 so 库文件信息查询服务器对应的默认模型信息。

□ 方法

```
public BasicHandler<ModelBags> getServerDefaultModels()
```

☐ 参数

无。

☐ 返回

BasicHandler<ModelBags> 查询任务返回的结果，支持取消 BasicHandler.cancel()。

5.8 获取本地前端模型文件路径

☐ 说明

根据 modelId 获取本地前端模型文件的绝对路径。

☐ 方法

```
public String getTextModelFileAbsPath(String modelId)
```

☐ 参数

名称	描述
modelId	模型 id

☐ 返回

前端模型文件的绝对路径。

5.9 获取本地后端模型文件信息

☐ 说明

根据 modelId 获取本地后端模型文件绝对路径。

☐ 方法

```
public String getSpeechModelFileAbsPath (String modelId)
```

☐ 参数

名称	描述
modelId	模型 id

☐ 返回

后端模型文件的绝对路径。

5.10 下载模型文件

☐ 说明

从服务器下载相应的模型文件，支持断点续传。

☐ 方法

public DownloadHandler download(String modelId, OnDownloadListener listener)

☐ 参数

名称	描述
modelId	模型 id
listener	下载监听事件(详细使用请参加 5.15 节)

☐ 返回

DownloadHandler 下载任务返回的结果，支持取消(单个下载取消)DownloadHandler.stop()。

5.11 判断模型信息是否可用

☐ 说明

判断模型文件是否可用。

☐ 方法

public Boolean isModelValid(String modelId)

☐ 参数

名称	描述
modelId	模型 id

☐ 返回

布尔值 是否可用。

5.12 判断模型文件是否可用

☐ 说明

判断模型文件是否可用。

☐ 方法

public Boolean isModelFileValid(String fileId)

☐ 参数

名称	描述
fileId	模型文件 id，对应 model 信息中的 textDataId 和 speechDataId

☐ 返回

布尔值 是否可用。

5.13 获得引擎参数

☐ 说明

判断引擎参数。

☐ 方法

public LibEngineParams getEngineParams()

☐ 参数

无。

☐ 返回

LibEngineParams 引擎参数对象信息。

5.14 停止下载

☐ 说明

停止全部的下载，与 DownHandler.stop（）有区别，DownHandler.stop（）用于停止单个指定的模型（modelId）下载操作。

☐ 方法

public int stop()

☐ 参数

无。

☐ 返回

错误码。

5.15 下载监听器

OnDownloadListener()是下载状态监听器接口，开发者可以通过实现此接口对下载事件做监听。

5.15.1 下载开始回调

☐ 说明

在下载开始时的回调函数。

☐ 方法

public void onStart(String modelId)

☐ 参数

名称	描述
modelId	模型 Id

☐ 返回

无。

5.15.2 下载进度回调

☐ 说明

在下载过程中的进度回调函数。

☐ 方法

public void onProgress(String modelId, long downloadBytes, long totalBytes)

☐ 参数

名称	描述
modelId	模型 Id
downloadBytes	已经下载的字节数
totalBytes	下载文件的总字节数

☐ 返回

无。

5.15.3 下载结束回调

☐ 说明

在下载正常结束或出错停止下载时的回调

☐ 方法

public void onFinish(String modelId, int code)

☐ 参数

名称	描述
modelId	模型 Id
code	错误码(0 为下载完成, -1005 为已下载, 其他为下载失败)

☐ 返回

无。

5.16 Conditions 条件对象

Conditions 为查询下载模型与模型文件的限制条件对象，以下方法都是可选设置的方法，非必须设置的参数方法。

5.16.1 添加模型 id 查询条件

☐ 说明

添加模型 id 查询条件。

☐ 方法

public void appendId(String ModelId)

☐ 参数

名称	描述
ModelId	模型 id（可选参数）

☐ 返回

无。

5.16.2 添加语言查询条件

☐ 说明

添加模型语言查询条件。

☐ 方法

public void appendLanguage (String Language)

☐ 参数

名称	描述
Language	语言（可选参数。可用参数：chn，eng）

☐ 返回

无。

5.16.3 添加性别查询条件

☐ 说明

添加模型性别查询条件。

☐ 方法

public void appendGender (String Gender)

☐ 参数

名称	描述
Gender	性别（可选参数。可用参数：female，male）

☐ 返回

无。

5.16.4 添加说话人查询条件

☐ 说明

添加模型说话人查询条件。

☐ 方法

public void appendSpeaker (String Speaker)

☐ 参数

名称	描述
Speaker	说话人（可选参数。设置的参数必须是完全匹配，否则返回结果将为空）

☐ 返回

无。

5.16.5 添加领域查询条件

☐ 说明

添加模型领域查询条件。

☐ 方法

public void appendDomain (String Domain)

☐ 参数

名称	描述
Domain	领域（可选参数。可用参数：common，custom）

☐ 返回

无。

5.16.6 添加质量查询条件

☐ 说明

添加质量查询条件。

☐ 方法

public void append Quality (String Quality)

☐ 参数

名称	描述
Quality	质量（可选参数。可用参数：high，middle）

☐ 返回

无。

6 错误码列表

错误码值	错误码描述
-1	在线引擎授权失败
-2	在线合成请求失败
-3	在线合成停止失败
-4	在线授权中断异常
-5	在线授权执行时异常
-6	在线授权时间超时
-7	在线合成返回错误信息
-8	在线授权 token 为空
-9	在线引擎没有初始化
-10	在线引擎合成时异常
-11	在线引擎不支持的操作
-12	在线合成请求解析出错
-13	在线合成获取合成结果被中断
-14	在线合成过程异常
-15	在线合成获取合成结果超时
-100	离线引擎授权失败
-101	离线合成停止失败
-102	离线授权下载 License 失败
-103	离线授权信息为空
-104	离线授权类型未知
-105	离线授权中断异常
-106	离线授权执行时异常
-107	离线授权执行时间超时
-108	离线合成引擎初始化失败
-109	离线引擎未初始化
-110	离线合成时异常

-111	离线合成返回值非 0
-112	离线授权已过期
-113	离线授权包名不匹配
-114	离线授权签名不匹配
-115	离线授权设备信息不匹配
-116	离线授权平台不匹配
-117	离线授权的 license 文件不存在
-200	混合引擎离线在线都授权失败
-201	混合引擎授权中断异常
-202	混合引擎授权执行时异常
-203	混合引擎授权执行时间超时
-204	混合引擎初始化 tts 时，离线初始化失败
-300	合成文本为空
-301	合成文本长度过长（不要超过 GBK1024 个字节）
-302	合成文本无法获取 GBK 字节
-400	TTS 未初始化
-401	TTS 模式无效
-402	TTS 合成队列已满（最大限度为 1000）
-403	TTS 批量合成文本过多（最多为 100）
-404	TTS 停止失败
-405	TTS APP ID 无效
-406	TTS 被调用方法参数无效
-500	Context 被释放或为空
-600	播放器为空
-1000	模型管理参数无效
-1001	模型管理请求出错
-1002	模型管理服务器端错误
-1003	模型管理数据库模型信息无效
-1004	模型管理数据库模型文件信息无效
-1005	模型数据已经存在（或已下载）
-1006	无法获取到模型信息

-1007	无法获取到模型文件信息
-1008	模型检查过程异常
-1009	模型文件下载时异常
-9999	未知错误

7 日志管理

为了方便调试，BDTTS+提供了对日志的管理功能。

```
LoggerProxy.printable(boolean isPrint);
```

该接口控制日志的打印功能，如果设置为 `false`，将关闭所有日志；如果设置为 `true`，则开启日志打印功能。默认日志级别为 `debug` 级别

FAQ

1. 如何联系我们？

开发过程中有任何 bug 或反馈意见，请发送邮件至 voice_feedback@baidu.com

2. Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Log

应用同时集成了百度其它 SDK，造成公共库冲突，请删除语音 SDK 中的 galaxy_lite.jar

3. 离线初始化失败。

一般为资源文件的路径不对，请使用可访问的文件绝对路径。

4. Tts sdk 和带 tts 功能的导航 SDK 冲突

该问题目前无法解决，因为导航已经内置了定制的 tts sdk,如果要使用 tts，请使用 rest api 的 tts 方案。