

标贝科技语音合成服务

Android SDK 使用说明文档（2.2.0）

Name	Date	Reason For Changes	Version
	2019.11.27	重构流式在线合成 sdk。	2.0.0
	2019.12.16	SDK 支持长文本合成，新增播放器功能。	2.1.0
	2020.9.3	SDK 统一 demo、版本号、错误码，规范集成方式	2.2.0

标贝（北京）科技有限公司 DataBaker(Beijing)technology co.,LTD

北京市海淀区西小口路 66 号中关村东升科技园 B-2 号楼 A203 室, 010-58465943

目录

1. Android Studio 集成 jar（参考 demo）	3
2. SDK 关键类.....	4
3. 调用说明.....	4
3. 参数说明.....	5
4.1 基本参数说明.....	5
4.2 BakerCallback 回调类方法说明.....	5
4.3 BakerMediaCallback 回调类方法说明.....	6
4.4 失败时返回的 code 对应表.....	6
4.4.1 失败时返回的 msg 格式.....	6
4.4.2 对应 code 值：	6

1. Android Studio 集成 jar（参考 demo）

1.1 在工程 model 下 build.gradle 文件中添加以下代码。建议使用此方式

```
dependencies {  
    implementation 'com.databaker.synthesizer:synthesizer:2.2.+'  
}
```

或者将 jar 包添加至工程 module 下，lib 文件夹里。

1.2 在 module 的 build.gradle 文件里，添加以下代码。注意：SDK 内使用了 okhttp、gson 所以需要开发者添加以下两个库。okhttp 请使用 4.x 版本，以免出现版本兼容 bug。

```
dependencies {  
    implementation 'com.squareup.okhttp3:okhttp:4.2.2'  
    implementation 'com.google.code.gson:gson:2.8.6'  
}
```

1.3 在主 Module 的 AndroidManifest.xml 文件中添加网络权限。若使用离线合成 SDK，在安卓 6.0 系统及以上版本需要申请写 SDK 权限。

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

1.4 在主 Module 的 AndroidManifest.xml 文件中的 application 节点添加以下属性。

```
android:usesCleartextTraffic="true"
```

Eclipse 环境也遵循相关集成 jar 包的方式即可。

1.5 关于混淆

SDK 中用到了 okhttp 和 gson，所以需要将这两个包的混淆代码添加上。具体混淆代码可以去官方文档上查阅。如果项目中已经有这两个包的混淆代码，不必重复添加。请加上我们 SDK 其他类的混淆代码，如下：

```
-keep class com.databaker.synthesizer.bean.** { *; }  
-keep public class com.databaker.synthesizer.BakerConstants{*;}  
-keep public class com.databaker.synthesizer.BakerSynthesizer{*;}  
-keep public class com.databaker.synthesizer.BakerCallback{*;}  
-keep public class com.databaker.synthesizer.SynthesizerCallback{*;}  
-keep public class com.databaker.synthesizer.BakerMediaCallback{*;}  
-keep public class com.databaker.synthesizer.BaseMediaCallback{*;}
```

2. SDK 关键类

1. **BakerSynthesizer**: 语音合成关键业务处理类，全局只需一个实例即可。
2. **BakerCallback**: 合成结果源数据回调类。在获得合成音频源数据，或发生错误等情况发生时触发此回调。如果您的应用场景中需要直接处理返回的字节类型源数据，您可以实现该类，并在回调方法中加入自己的处理逻辑。设置参数时请将此 **callback** 提交给 **BakerSynthesizer** 实例。
3. **BakerMediaCallback**: 如果想直接使用 SDK 中的播放器来处理文本合成播放任务。您可以实现该类，此回调类中包含了播放器的各种状态回调，您可以在这些回调方法中实现自己的其他业务逻辑。设置参数时请将此 **callback** 提交给 **BakerSynthesizer** 实例。
4. **BakerConstants**: 参数、错误码等常量类。

3. 调用说明

- 1.1 初始化 **BakerSynthesizer** 类，得到 **BakerSynthesizer** 的实例。
- 1.2 SDK 中提供了 2 个回调类。如果想要自己处理合成返回的字节类型源数据，则可以定义 **BakerCallback** 实现类。如果想直接将合成文本数据交给 SDK 中的播放器处理，则可以定义 **BakerMediaCallback** 实现类。[如果选择了定义 **BakerCallback** 实现类，SDK 中不会执行播放器等一整套业务代码，不用担心由此带来的各类附加资源开销。](#)
- 1.3 设置 **BakerSynthesizer** 合成参数，包括必填参数和非必填参数。
- 1.4 调用 **BakerSynthesizer.start()** 方法开始与云端服务连接。
- 1.5 **callback** 中的 **onPrepared()** 意义是合成的第一帧数据已取得。所以您可以在此回调方法中[调用 **bakerSynthesizer.bakerPlay\(\)** 开启播放任务。](#)
- 1.6 在 **callback** 其他回调方法中按照您的业务需求实现对应逻辑。
- 1.7 如果需要发起新的请求，可以重复第 3-6 步。
- 1.8 在业务完全处理完毕，或者页面关闭时，调用 **bakerSynthesizer.onDestroy()**；结束 **websocket** 服务，释放资源。

注意：若使用 SDK 中播放器执行合成音频播放任务，有以下方法可调用。

- **bakerSynthesizer.bakerPlay()** 播放音频，常在 **onPrepared()** 回调方法里调用此方法执行播放。
- **bakerSynthesizer.bakerPause()** 暂停播放。
- **bakerSynthesizer.bakerStop()** 停止播放。
- **bakerSynthesizer.isPlaying()** 当前播放状态，**boolean** 型，**true**=正在播放中，**false**=暂停或停止播放。
- **bakerSynthesizer.getCurrentPosition()** 当前播放进度。
- **bakerSynthesizer.getDuration()** 文本合成音频的总长度。

3. 参数说明

4.1 基本参数说明

参数	参数名称	是否必填	说明
setText	合成文本	是	设置要转为语音的合成文本
setBakerCallback	数据回调方法	是	设置返回数据的 callback
setVoice	发音人	否	设置发音人声音名称，默认：标准合成_模仿儿童_果子
setLanguage	合并文本语言类型	否	合成请求文本的语言，目前支持 ZH(中文和中英混)和 ENG(纯英文，中文部分不会合成)，默认：ZH
setSpeed	语速	否	设置播放的语速，在 0~9 之间（支持浮点值），不传时默认为 5
setVolume	音量	否	设置语音的音量，在 0~9 之间（只支持整型值），不传时默认值为 5
setPitch	音调	否	设置语音的音调，取值 0-9，不传时默认为 5 中语调
setAudioType	返回数据文件格式	否	可不填，不填时默认为 4 audiotype=4：返回 16K 采样率的 pcm 格式 audiotype=5：返回 8K 采样率的 pcm 格式 audiotype=6：返回 16K 采样率的 wav 格式 audiotype=6&rate=1：返回 8K 的 wav 格式

注意：如果调整了参数中的采样率或码率，记得注意(Demo 中示例的)播放器的采样率也要同步调整。

4.2 BakerCallback 回调类方法说明

参数	参数名称	说明
onSynthesisStarted	开始合成	开始合成
onPrepared	准备就绪	第一帧数据返回时的回调，此时可以使用数据执行播放。
onBinaryReceived	流式持续返回数据的接口回调	data 合成的音频数据，已使用 base64 加密，客户端需进行 base64 解密。 audioType 音频类型，如 audio/pcm。 interval 音频 interval 信息，可能为空。 endFlag 是否时最后一个数据块，false: 否，true: 是。
onSynthesisCompleted	合成完成。	当 onBinaryReceived 方法中 endFlag 参数=true，即最后一条消息返回后，会回调此方法。
onTaskFailed	合成失败	返回 msg 内容格式为： {"code":1000000,"message":"...", "trace_id":"..."}

		1572234229176271"} trace_id 是引擎内部合成任务 ID。
--	--	---

4.3 BakerMediaCallback 回调类方法说明

参数	参数名称	说明
onPrepared	准备就绪	第一帧数据返回时的回调，此时可以使用数据执行播放。
onCacheAvailable	数据缓存进度回调	percentsAvailable 已缓存的百分比，整数型，取值范围 0-100。
playing	开始播放回调	播放状态切换：开始播放时的回调。
noPlay	暂停或停止播放回调	播放状态切换：暂停或停止播放时的回调。
onCompletion	播放结束	当数据播放完成时的回调。
onError	各类失败时的回调	返回 msg 内容格式为： {"code":1000000,"message":"...","trace_id":"1572234229176271"} trace_id 是引擎内部合成任务 ID。

4.4 失败时返回的 code 对应表

4.4.1 失败时返回的 msg 格式

参数名称	类型	描述
code	int	错误码 10xxxxx 表示服务端相关错误，20xxxxx、30xxxxx 表示 SDK 相关错误
message	string	错误描述
trace_id	string	引擎内部合成任务 id

4.4.2 对应 code 值：

错误码	含义
1010001	access_token 参数获取失败或未传输
1010002	domain 参数值错误
1010003	language 参数错误
1010004	voice_name 参数错误
1010005	audiotype 参数错误
1010006	rate 参数错误
1010007	idx 错误
1010008	single 错误
1010009	text 参数错误
1010010	文本太长
1020000	获取资源错误
1020001	断句失败
1020002	分段数错误

1020003	分段后的文本长度错误
1020004	获取引擎链接错误
1020005	RPC 链接失败错误
1020006	引擎内部错误
1020007	操作 redis 错误
1020008	音频编码错误
1030000	鉴权错误 (access_token 值不正确或已经失效)
1030001	并发错误
1030002	内部配置错误
1030003	json 串解析错误
1030004	获取 url 失败
1030005	获取客户 IP 地址失败
1030006	任务队列错误
1040001	请求不是 json 结构
1040002	缺少必须字段
1040003	版本错误
1040004	字段值类型错误
1040005	参数错误
1050001	处理超时
1050002	内部 rpc 调用失败
1050004	其他内部错误
2000001	缺少 ClientId
2000002	缺少 Secret
2000003	token 获取失败
2000004	参数格式错误
2000005	合成文本内容为空
2000006	发音人参数错误
2000007	返回结果解析错误
2000008	合成失败, 失败信息相关错误
3000001	播放器相关错误
3000002	合成文本内容转码错误
3000003	返回结果解析错误;返回 null