

标贝科技离线语音合成服务

Android SDK 使用说明文档（2.0）

Name	Date	Reason For Changes	Version
	2020.4.24	创建文档，编写使用说明。	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 aar（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 aar（参考 demo）

1.1 集成 aar 包有多种方式，本文提供一种作为参考，采用其他正确集成方式都可行。将 aar 包拷贝至 libs 目录下，并执行下一步（1.2）。注意新增*.aar 格式。

1.2 在主 module 的 build.gradle 文件里，添加以下代码。

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar', '*.aar'])  
    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" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

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 aar 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.baker.synthesis.offline.bean.** { *; }  
-keep public class com.baker.synthesis.offline.BakerConstants{*;}  
-keep public class com.baker.synthesis.offline.authorization.AuthorizationCallback{*;}  
-keep public class com.baker.synthesis.offline.OffLineSynthesizerEngine{*;}  
-keep public class com.baker.synthesis.offline.OfflineBakerSynthesizer{*;}  
-keep public class com.baker.synthesis.offline.BakerCallback{*;}  
-keep public class com.baker.synthesis.offline.BakerMediaCallback{*;}  
-keep public class com.baker.synthesis.offline.SynthesizerCallback{*;}  
-keep public class com.baker.synthesis.offline.BaseMediaCallback{*;}
```

2. SDK 关键类

1. **OfflineBakerSynthesizer**: 离线合成关键业务处理类，全局只需要一个实例即可。
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**: 参数等常量类。

3. 调用说明

- 1.1 设备首次使用离线合成服务，需要联网，进行服务授权激活。[Demo](#) 中的 [aar](#) 内包含 5 个示例发音人，若定制化离线服务，可指定发音人，降低 [aar](#) 文件大小。
- 1.2 初始化 **OfflineBakerSynthesizer** 类，得到 **OfflineBakerSynthesizer** 的实例。
- 1.3 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。](#)
- 1.4 设置 **OfflineBakerSynthesizer** 合成参数，包括必填参数和非必填参数。
- 1.5 调用 **start()** 方法开始合成服务。
- 1.6 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 **bakerPlay\(\)** 开启播放任务。](#)
- 1.7 在 callback 其他回调方法中按照您的业务需求实现对应逻辑。
- 1.8 如果需要发起新的请求，可以重复第 4-7 步。
- 1.9 在业务完全处理完毕，或者页面关闭时，调用 **onDestroy()** 结束服务，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。

- **bakerPlay()** 播放音频，常在 **onPrepared()** 回调方法里调用此方法执行播放。
- **bakerPause()** 暂停播放。
- **bakerStop()** 停止播放。
- **isPlaying()** 当前播放状态，boolean 型，true=正在播放中，false=暂停或停止播放。

- `getCurrentPosition()` 当前播放进度。
- `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
<code>setText</code>	合成文本	是	设置要转为语音的合成文本
<code>setBakerCallback</code>	数据回调方法	是	设置返回数据的 <code>callback</code>
<code>setVoice</code>	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
<code>setSpeed</code>	语速	否	设置播放的语速，在 50~200 之间（只支持整型值），不传时默认为 105
<code>setVolume</code>	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

4.2 BakerCallback 回调类方法说明

参数	参数名称	说明
<code>onSynthesisStarted</code>	开始合成	开始合成
<code>onPrepared</code>	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。
<code>onBinaryReceived</code>	流式持续返回数据的接口回调	<code>data</code> 合成的音频数据。 <code>audioType</code> 音频类型，如 <code>audio/pcm</code> 。 <code>interval</code> 音频 <code>interval</code> 信息，可能为空。 <code>endFlag</code> 是否时最后一个数据块， <code>false</code> ：否， <code>true</code> ：是。
<code>onSynthesisCompleted</code>	合成完成。	当 <code>onBinaryReceived</code> 方法中 <code>endFlag</code> 参数= <code>true</code> ，即最后一条消息返回后，会回调此方法。
<code>onTaskFailed</code>	合成失败	返回 <code>msg</code> 内容格式为： <code>{"code":40000,"message":"...", "trace_id":"1572234229176271"}</code> <code>trace_id</code> 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":40000,"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 9xxxx 表示 SDK 相关错误，1xxxx 参数相关错误，2xxxx 合成引擎相关错误，3xxxx 授权及其他错误
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
90001	合成 SDK 初始化失败
90002	合成文本内容为空
90003	参数格式错误
90004	返回结果解析错误
90005	合成失败，失败信息相关错误。
90006	SDK 中播放器相关错误。
90007	离线 SDK 初始化相关错误
90008	离线 SDK 授权错误
90009	传入文本过长
90010	文本编码不支持
90011	引擎内部错误（离线合成引擎错误码）
90012	离线授权成功
90013	离线授权未知错误
90014	引擎正在合成

90015	引擎没有初始化
90016	内存不足
90017	引擎参数错误
90018	引擎运行时错误
90019	模型文件错误
90020	动态库链接错误
90021	正在初始化
90022	释放离线合成 SDK 错误
90023	无 SD 卡读写权限错误
10001	access_token 参数获取失败或未传输
10002	domain 参数值错误
10003	language 参数错误
10004	voice_name 参数错误
10005	audiotype 参数错误
10006	rate 参数错误
10007	idx 错误
10008	single 错误
10009	text 参数错误
10010	文本太长
20000	获取资源错误
20001	断句失败
20002	分段数错误
20003	分段后的文本长度错误
20004	获取引擎链接错误
20005	RPC 链接失败错误
20006	引擎内部错误
20007	操作 redis 错误
20008	音频编码错误
30000	鉴权错误 (access_token 值不正确或已经失效)
30001	并发错误
30002	内部配置错误
30003	json 串解析错误
30004	获取 url 失败
30005	获取客户 IP 地址失败
30006	任务队列错误
40001	请求不是 json 结构
40002	缺少必须字段
40003	版本错误
40004	字段值类型错误
40005	参数错误
50001	处理超时
50002	内部 rpc 调用失败
50004	其他内部错误

