

## 标贝科技长语音识别服务

### Android SDK 使用说明文档（1.0.0）

Name	Date	Reason For Changes	Version
	2020.10.15	创建文档，编写使用说明。	1.0.0

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
4. 参数说明.....	5
4.1 基本参数说明.....	5
4.2 LongTimeAsrCallBack 回调类方法说明.....	5
4.3 失败时返回的 code 对应表.....	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.baker.sdk.longtime.asr:bakerlongtimeasr:1.0.0'  
//识别库会依赖下面两个库  
implementation 'com.squareup.okhttp3:okhttp:4.2.2'  
implementation 'com.google.code.gson:gson:2.8.6'
```

}

或者将.jar/包添加至工程 module 下，lib 文件夹里。

注意：SDK 内使用了 okhttp、gson 所以需要开发者添加以这两个库，若工程内已存在，无需重复添加。okhttp 请使用 4.x 版本，以免出现版本兼容 bug。

**1.2** 在主 Module 的 AndroidManifest.xml 文件中添加网络权限。

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

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

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

## 1.4 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.sdk.longtime.asr.bean.** { *; }  
-keep public class com.baker.sdk.longtime.asr.LongTimeAsr{*;}  
-keep public class com.baker.sdk.longtime.asr.listener.LongTimeAsrCallBack{*;}  
-keep public class com.baker.sdk.longtime.asr.base.BakerLongTimeAsrConstants{*;}  
-keep class com.baker.sdk.http.**{*;}  
-keep class com.baker.sdk.basecomponent.**{*;}  

```

## 2. SDK 关键类

1. LongTimeAsr: 长语音识别关键业务处理类，全局只需一个实例即可。
2. LongTimeAsrCallBack: 实时识别结果源数据回调类。您可以在这些个回调方法中获取到当前识别的各类结果和状态，或者错误信息。您可以在回调方法中处理识别结果。初始化 SDK 时 请将此 callback 提交给 LongTimeAsr 实例。

## 3. 调用说明

- 3.1 实例化 LongTimeAsr 类，得到 LongTimeAsr 的实例。

```
LongTimeAsr longTimeAsr = new LongTimeAsr();
```

- 3.2 初始化 SDK。

如果是调用我们公有云公共识别接口，请采用以下示例代码初始化 LongTimeAsr 类,传入由我们授权邮件里面给到的账户相关的 `clientId` 和 `clientSecret`。

```
longTimeAsr.initSdk(context, clientId, clientSecret, LongTimeAsrCallBack);
```

如果是私有化部署请采用以下示例代码初始化 LongTimeAsr 类。并注意需要在参数中设置私有化部署的识别服务器地址，仅私有化部署时需要设置该参数。

```
longTimeAsr.initSdk(context, LongTimeAsrCallBack);
```

```
longTimeAsr.setUrl("ws://xxxx"); //设置私有化部署的识别服务器地址
```

- 3.3 长语音识别支持 2 种识别场景，第一种是 SDK 内调用手机麦克风录音、上传完成识别返回识别结果。此种方式需要接入方提前获取应用的录音权限。第二种场景是接收字节流，SDK 收到数据后，上传服务器完成识别，返回识别结果。

第一种场景：

- A. 需要应用取得录音权限。
- B. 通过 LongTimeAsr 实例，设置各类参数。
- C. `longTimeAsr.startAsr()`; 开启识别。
- D. 在 LongTimeAsrCallBack 回调中接收状态、识别结果，或错误信息等。
- E. 通过 `longTimeAsr.stopAsr()`; 此方法结束识别。

第二种场景：

- A. 通过 LongTimeAsr 实例，设置各类参数。
- B. `longTimeAsr.start()`; 通知 SDK 准备识别。
- C. 在 LongTimeAsrCallBack 的 `onReady()` 回调方法中得到 SDK 已准备好后。

- D. 多次调用 `longTimeAsr.send(byte[] data)`;方法向 SDK 输入待识别语音字节流。此方法单次接收固定长度（5120）字节数组。最后一片数据可以是其他长度字节数组。
- E. 在 `LongTimeAsrCallBack` 回调中接收状态、识别结果，或错误信息等。
- F. `longTimeAsr.end()`;结束识别。
- G. 此种方式，每次在调用 `longTimeAsr.send(byte[] data)`时，前后 2 次间隔时间不能超过 60 秒，私有化部署根据服务器设定具体时间为准。

3.4 在确定不再使用识别后，调用 `longTimeAsr.release()`释放 SDK 资源。

3.5 如果需要发起新的识别请求，可以重复第 3.3 步。

## 4. 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>initSdk</code>	初始化 SDK	是	必须调用此方法初始化 SDK，详见 3.2
<code>setUrl</code>	识别服务器地址	否	仅私有化部署，需要设置服务器 URL。
<code>setSampleRate</code>	设置语音录音采样率	否	采样率支持 16000/8000，默认 16000。
<code>setAddPct</code>	设置是否返回标点	否	是否在短静音处添加标点，默认 <code>true</code>
<code>setDomain</code>	模型名称	否	模型名称，必须填写公司购买的语言模型，默认为 <code>common</code>
<code>setAudioFormat</code>	音频编码格式	否	输入字节流方式识别时，支持设置音频编码格式 <code>pcm</code> 或 <code>wav</code>

### 4.2 LongTimeAsrCallBack 回调类方法说明

参数	参数名称	说明
<code>onReady</code>	准备就绪	授权验证通过，且与服务器连接成功，准备就绪回调
<code>onRecording</code>	实时识别持续返回数据的接口回调	当识别有效，且识别到内容后，会唤起此回调方法，多次回调。回调内容是当前句的实时识别内容，回调中的字符串结果会持续变化至最新校正后的结果。布尔值代表当前句识别是否结束，为 <code>true</code> 时代表当前句识别最终结果，下一次回调是下一句的识别内容。 <b>注意：需要开发者根据此回调逐句追加显示或处理识别结果。</b>
<code>onError</code>	识别出现错误	<code>String code</code> 是错误码，错误码具体请参考以下错误码对照表。 <code>String message</code> 是具体的错误信息日志。识别过程中出现各种错误情况都会回调此方法。一旦出现错误，此次识别任务中断。

### 4.3 失败时返回的 code 对应表

错误码	含义
14180001	缺少 ClientId
14180002	缺少 Secret
14180003	token 获取失败
14180004	没有录音权限
14180005	本地网络不可用
14180006	没有调用 start 方法
14180007	录音尚未初始化
14180008	正在录音
14180009	发送的数据为空
14180010	response is null
14180011	gson to object error
14180012	websocket 发送消息出错
14180013	websocket onFailure error
14180014	录音意外中断
90000	文本数据
30001	HTTP 请求参数错误
30002	服务内部错误
30003	识别结果解析出错
30004	应用包名未知
30005	语音质量问题
30006	输入语音过长
30007	连接识别引擎失败
30008	会话 id 不存在
30009	Rpc 调用非法
30010	redis rpop 操作返回空
30011	redis rpop 值不合法
30012	rpc 调用识别引擎失败
30013	Redis rpop 操作失败
30014	redis lpush 操作失败
30015	单个语音分片过长
30016	回调 url 失败
40001	json 解析失败
40002	json 字段不全
40003	版本错误
40004	json 字段值类型错误
40005	参数错误
40006	idx 超时
40007	idx 顺序错误
40008	token 校验失败

40009	token 处于未激活状态
40010	token 已过期
40011	使用量已超过购买量
40012	qps 错误
50001	处理超时
50002	内部 rpc 调用失败
50003	服务端繁忙
50004	其他内部错误