

标贝科技语音识别服务

Android SDK 使用说明文档（1.2.3）

Name	Date	Reason For Changes	Version
	2020.02.25	创建文档，编写使用说明。	1.0.0
	2020.08.05	增加文件识别	1.1.0
	2020.09.18	SDK 统一 demo、版本号、错误码，规范集成方式	1.2.0
	2020.10.12	错误码统一升级	1.2.2
	2020.10.22	新增通过字节流方式作为识别数据源	1.2.3

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
4. 参数说明.....	5
4.1 基本参数说明.....	5
4.2 BakerRecognizerCallback 回调类方法说明.....	6
4.3 失败时返回的 code 对应表.....	6

1. Android Studio 集成 aar（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 {
    //识别库
    implementation 'com.baker.speech.asr:bakerasr:1.2.+'
    //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/.aar 包添加至工程 module 下，lib 文件夹里。

集成 .jar/.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。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.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

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.speech.asr.basic.**{*;}
-keep class com.baker.speech.asr.bean.**{*;}
-keep class com.baker.speech.asr.utils.**{*;}
-keep public class com.baker.speech.asr.AsrConstants{*;}
-keep public class com.baker.speech.asr.AsrErrorConstants{*;}
-keep public class com.baker.speech.asr.BakerRecognizer{
    public <methods>;
}
-keep public class com.baker.speech.asr.vadJni{*;}
-keep                                     public                                     class
com.baker.speech.asr.event.EventManagerMultiNet$BakerWebSocketListener{*;}
```

2. SDK 关键类

1. **BakerRecognizer**: 语音识别关键业务处理类，全局只需一个实例即可。
2. **BakerRecognizerCallback**: 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。设置参数时请将此 callback 提交给 **BakerRecognizer** 实例。

3. 调用说明

- 1.1 初始化 **BakerRecognizer** 类，得到 **BakerRecognizer** 的实例。

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 **BakerRecognizer** 类，传入由我们授权邮件里面给到的账户相关的 **clientId** 和 **clientSecret**。

```
BakerRecognizer bakerRecognizer = BakerRecognizer.getInstance  
(VoiceSpeechActivity.this, clientId, clientSecret);
```

如果是私有化部署请采用以下示例代码初始化 **BakerRecognizer** 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。

```
BakerRecognizer bakerRecognizer = BakerRecognizer.getInstance  
(VoiceSpeechActivity.this);  
bakerRecognizer.setUrl("ws://xxx"); //设置私有化部署的识别服务器地址
```

- 1.2 设置 **BakerRecognizer** 识别所需参数，包括必填参数和非必填参数。请参考 demo 和注释信息。

1.3 调用 **BakerRecognizer.startRecognize()**方法开启语音识别，该方法返回 **int** 值，返回 **0** 代表启动成功。识别结果反馈在 **callback** 类的各个回调方法中，各回调方法的介绍请参考以下 **BakerRecognizerCallback** 回调类方法说明。正常识别一般无需调用 **BakerRecognizer.stopRecognition()**方法结束识别，SDK 中的 VAD 会自动检测，当符合条件时会自动结束识别，比如声音停顿时间超过设置的语音后 endpoint 静音检测时间。

文件识别功能: **BakerRecognizer.startRecognize(String filePath)**在此方法中的参数传入文件路径即可，支持原始 PCM 文件。

开发者提供识别数据源: **BakerRecognizer.startRecognizeWithByte()**使用此方法开启语音识别。

多次调用 **BakerRecognizer.sendPcmBuffer(buffer)**方法发送数据。

当 `BakerRecognizer.startRecognizeWithByte()` 方法开启识别的时候 `sendPcmBuffer()` 方法有效。`buffer` 的大小务必固定 1024(最后一包除外)；传递完数据后可以自行调用 `BakerRecognizer.stopRecognition()` 结束识别或者当识别过程中静音段超时会自动结束识别。

`sendPcmBuffer()`；返回结果说明：0=正常，1=`buffer` 是空，2=`buffer` 超过 1024。

1.4 如果需要发起新的识别请求，可以重复第 1.2-1.3 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>setCallback</code>	设置识别回调	是	设置识别回调类
<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>setVadSos</code>	设置语音前端点	否	静音超时时间，即用户多长时间不说话则当做超时处理，参数请勿随意设置，因为此处设置的参数是按照帧计算的。
<code>setVadEos</code>	设置语音后端点	否	后端点静音检测时间，即用户停止说话多长时间即认为不再输入，自动停止录音，参数请勿随意设置，因为此处设置的参数是按照帧计算的。
<code>setVadWait</code>	设置语音最大识别时长	否	设置语音最大识别时长，最长 60s=1800，参数请勿随意设置，因为此处设置的参数是按照帧计算的。
<code>setVadPause</code>	设置语句间停顿间隔	否	设置语句间停顿间隔，默认 2s=45，参数请勿随意设置，因为此处设置的参数是按照帧计算的。
<code>setSample</code>	设置识别声音的采样率	否	设置采样率 目前支持 16000 默认 16000
<code>addPct</code>	是否添加标点	否	是否添加标点 <code>true</code> =返回标点， <code>false</code> =不返回标点，默认 <code>false</code>
<code>enableItn</code>	是否执行归一化处理	否	是否执行归一化处理， <code>true</code> =需要处理， <code>false</code> =不需要处理，默认= <code>false</code> 。
<code>setRecognizeType</code>	识别类型	否	识别模式， 0= 一句话识别，sdk 端做 vad 1=长语音识别，服务端做 vad，默认为 0。

4.2 BakerRecognizerCallback 回调类方法说明

参数	参数名称	说明
onReadyOfSpeech	准备就绪	准备就绪回调
onBeginOfSpeech	识别开始	当识别到有声音时，会唤起此回调方法。即 mic 输出声音流时。
onResult	流式识别持续返回数据的接口回调	此回调方法会被多次唤起。持续返回实时的识别数据信息。List<String> nbest 参数是一个识别结果集合，最优识别结果理论上是 nbest(0)，List<String> uncertain 参数是一个预测的可能出现的语音识别内容的结果集，供选择使用。boolean isLast 当前结果是否是最终识别数据，false: 否，true: 是。
onEndOfSpeech	识别正常结束。	正常识别结束时，会唤起此回调方法。
onVolumeChanged	识别音量信息	float volume 是取值 0-1 之间的浮点数，代表声音的量能。byte[] data 是声音的二进制信息，也可以自己根据源数据流计算处理声音量能信息。
onError	识别出现错误	int code 是错误码，错误码具体请参考以下错误码对照表。String message 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。

4.3 失败时返回的 code 对应表

错误码	含义
13180001	缺少 ClientId
13180002	缺少 Secret
13180003	token 获取失败
13180004	没有录音权限
13180005	录音出现未知错误
13180006	本地网络不可用
13180007	webSocket failure
13180008	没有说话
13180009	声音太短
13180010	录音被电话打断

13180011	本地网络未知错误
13180012	服务器检测到识别超过 60s
13180013	vad 设置参数错误
13180014	vad 发送数据错误
13180015	vad 状态错误
13180016	超过 vad 识别最长时限
13180017	vad 数据返回错误
13180018	buffer_closed
13180019	返回结果解析错误
13180020	返回结果解析错误;返回 null
13180021	传递的数据为空
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