

语音识别安卓SDK

目录

- 1. 文档说明
 - 2. 版本说明
 - 3. SDK说明&功能简介
 - 语音识别
 - 在线识别
 - 离线命令词
 - 唤醒词
 - 语义
 - 录音环境
 - 4.输入参数
 - 识别输入参数
 - 识别输入事件
 - ASR_START 输入事件参数
 - ASR_KWS_LOAD_ENGINE 输入事件参数
 - LANGUAGE 参数
 - 语音识别LANGUAGE
 - 语音自训练平台
 - 具体功能参数
 - 鉴权信息
 - 设置语言、模型、自定义词库、语义
 - 长语音
 - 设置静音时长进行断句
 - 开启/关闭根据静音时长切分句子
 - 离线命令词
 - 唤醒输入参数
 - WAKEUP_START 输入事件参数
 - 具体功能参数
 - 语义输入参数
 - 具体功能参数
- 5.输出参数
 - 识别输出参数
 - 唤醒输出参数

- 6. Demo运行
 - 6.1 配置包名和签名
 - 6.2 修改鉴权信息
- 7. SDK集成
- 8. 相关授权文件
- 9. 语音相关接口流程
 - 9.1在线识别调用流程
 - 9.2 离线命令词调用流程
 - 9.3识别错误码
 - "4004"鉴权子错误码
 - 服务端错误码请参照 [点击查看](#)
 - 9.4唤醒词调用流程
 - 9.5唤醒错误码
- 10. 代码混淆
- 11. 权限

1. 文档说明

文档名称	语音识别集成文档
所属平台	Android
提交日期	2026-05-12
概述	本文档是百度语音开放平台Android SDK的用户指南，描述了 短语音识别、离线自定义命令词识别、语音唤醒、语义解析与对话管理 等相关接口的使用说明。SDK内部均为采用流式协议，即用户边说边处理。区别于Restapi需

2. 版本说明

1	名称	版本号
2	语音识别	3.5.2
3	系统支持	android 5.1+
4	架构支持	支持armeabi-v7a, armeabi-v8a架构
5	硬件要求	要求设备上有麦克风
6	网络	支持移动网络（不包括2G、3G）、WIFI等网络环境

7	开发环境	建议使用最新版本Android Studio 进行开发
---	------	-----------------------------

3. SDK说明&功能简介

1	文件名称	版本号	MD5值	资源描述
2	bdasr_aipd_V3_20260327_8dd3e2f2a6.aar	8dd3e2f2a6	0fee003cb5a02ad179679edbd71c0914	jar 库 ; 终
3				

语音识别Android SDK 功能 主要分为语音识别 及 语义理解与对话管理：

- 1. 语音识别： 将录音转为文字。目前在线识别支持普通话、英文、粤语和四川话。
- 2. 语义理解与对话管理： 提取语音识别出的文字的意图与关键信息，并做出回应。

语音识别

语音识别，可以分为在线识别，离线命令词，及唤醒词

- 1. 在线识别： 即联网使用的识别功能，支持自定义词库及自训练平台。目前在线识别支持普通话、英文、粤语和四川话，通过在请求时配置不同的language参数，选择对应模型。默认为麦克风输入，可以设置参数为pcm格式16k采样率，16bit，小端序，单声道的音频流输入。
- 2. 离线命令词： 断网时识别固定的预定义短语（定义在bsg文件中），SDK强制优先使用在线识别。断网时激活，只能识别预定义的短语。联网时，强制使用在线识别。固定短语的语法需要从控制台“离线词&本地语义”模块预定义并下载为baidu_speech_grammar.bsg文件
- 3. 唤醒词： 识别预定义的“关键词”， 这个“关键词”必须在一句话的开头。本地功能，不需要网络。唤醒词即识别“关键词”，当SDK的识别引擎“听到”录音中的关键词后，立即告知用户。与android系统的锁屏唤醒完全无关。关键词和离线命令词一样，需要预定义并下载为WakeUp.bin文件

在线识别

在线是指手机联网时(3G 4G 5G wifi)，在线识别可以分为：

- 1. 在线普通识别： 流式识别出识别用户输入的录音音频流，支持普通话、英文、粤语和四川话。限制60s时长。
- 2. 在线长语音识别： 在线普通识别的基础上，没有60s时长的限制。在线识别可以测试 DEMO中的第一个按钮“在线识别”。

自定义词库

设置方法：

登录百度云-管理中心“管理应用”“选择应用”“语音识别词库设置”右侧“进行设置”；

设置效果：

1. 可以自定义识别词，提升准确率。
2. 仅在普通话输入法模型下生效。
3. 自定义词库适合短句，保证词库中一模一样的短句可以被识别出。
4. 词库中的分词优先级较高。

举例 词库定义了1个短句 :1.“摆渡船来了”百度内部处理的可能的分词结果：摆渡船来了。

以下录音的结果

1. 原始音频：摆渡船来了 =》识别结果： 摆渡船来了 【保证结果】
2. 原始音频：摆渡船来了么 =》识别结果： 百度传来了么 【可能结果，不保证】
3. 原始音频：摆渡船来 =》识别结果： 百度传来 【可能结果，不保证】
4. 原始音频：百度传来了喜讯 =》识别结果： 摆渡船来了喜讯 【不保证，词库内的分词优先级高】

最好在1万行以内。

副作用：如果用户的测试集中包含大量非自定义词表的query，整体上准确率下降。

语音自训练平台模型训练

自训练平台可以认为是自定义词库的升级版本，可以更加直观地查看训练效果。同样使用您自定义的文本进行结果优化。

具体功能及使用说明请参考文档“[自训练平台手册](#)”

离线命令词

离线命令词，联网时，强制使用在线识别，在断网时或在线请求超时，使用离线命令词功能。离线命令词功能不支持任意语句的识别，只能识别预定义的固定短语。

唤醒词

唤醒词即识别预定义的“关键词”。与在线长语音识别不同，长语音识别会返回所有识别结果，唤醒词只会识别出您预先定义的关键词。与android本身的锁屏唤醒没有任何关系。

唤醒词是本地功能，正常使用时无需联网。在**<http://ai.baidu.com/tech/speech/wake>**页面下方可以自行定义bin文件。百度语音提供了近15个预定义唤醒词，效果有优化。也可以自定义唤醒词，效果不如预定义唤醒词。bin文件中最多可以有10个唤醒词，其中自定义唤醒词不超过3个，并且2个字的预定义唤醒词不超过3个。进行唤醒词操作前必须要有相对静音。

语义

语义包括理解与对话管理，可用于提取语音识别出的文字的意图与关键信息，并做出回应。语音识别Android SDK提供了本地语义的方式：

本地语义：

- 在任意网络下都可使用，android sdk内部根据设置的bsg文件的定义进行离线命令词语义解析

录音环境

百度短语音识别（含唤醒）要求安静的环境，真人的正常语速的日常用语，并且不能多个人同时发音。

音频格式要求:

默认为麦克风输入，可以设置参数为pcm格式16k采样率，16bit，小端序，单声道的音频流输入。

以下场景讲会导致识别或者唤醒效果变差，错误，甚至没有结果:

1. 吵杂的环境
2. 有背景音乐，包括扬声器在播放百度合成的语音。

以下场景的录音可能没有正确的识别结果:

1. 音频里有技术专业名称或者用语（技术专业名称请到自训练平台改善）
2. 音频里是某个专业领域的对话，非日常用语。比如专业会议，动画片等

建议先收集一定数量的真实环境测试集，按照测试集评估及反馈。语音识别没有降噪功能无法过滤背景音等杂声。

4.输入参数

识别输入参数

场景:

- 在线识别：百度语音服务器将录音识别出文字，包括长语音
- 离线命令词：离线识别出预定义的固定短语
- 本地语义：在识别出文字的基础上（包括离线命令词识别），对文字做语义分析。任意网络状况。

使用网络状况:

- 离线：涵盖离线命令词，及离线命令词识别后的本地语义解析。

共支持4个语种，语种请在 ASR_START输入事件中的LANGUAGE参数中设置

- 中文普通话（全部场景）
- 中文四川话（离线命令词及语义不支持）
- 粤语（离线命令词及语义不支持）
- 英语（离线命令词及语义不支持）

识别输入事件

,以下参数均为SpeechConstant类的常量, 如SpeechConstant.ASR_START, 实际的String字面值可以参见SpeechConstant类或自行打印

1	事件名	类型	值	场景
2	ASR_START	String (JSON结构的字符串)	json内的参数 见下文 “ASR_START 参数”	全部
3	ASR_STOP			全部
4	ASR_CANCEL			全部
5	ASR_KWS_LOAD_ENGINE	String (JSON结构的字符串)	json内的参数 见下文 “ASR_KWS_LOAD_ENGINE 参数”	
6	ASR_KWS_UNLOAD_ENGINE			离线命令词

ASR_START 输入事件参数

1	事件参数	类型/值	场景	常用程度	描述
2	APP_ID	String	全部	推荐	开放平台创建应用后 AuthUtil 工具类中填入自己的 权信息。 (新版创建AipeEver 参考9.1在线识别调用
3	APP_KEY	String	全部	推荐	开放平台创建应用后 AuthUtil 工具类中填入自己的 权信息。 (新版创建AipeEver 参考9.1在线识别调用
4	SECRET	String	全部	推荐	开放平台创建应用后 AuthUtil 工具类中填入自己的 权信息。 (新版创建AipeEver 参考9.1在线识别调用
	LANGUAGE	String	在线	常用	默认"cmn-Hans-CN"

					其中输入法模型是指
5	LM_ID	int	在线	常用	自训练平台上线后的
6	DECODER	int	全部	常用	离在线的并行策略
7		0 （默认）	在线		纯在线(默认)
8		2	离线		离在线融合(在线优先)
9	VAD	String	全部	高级	语音活动检测， 根据60s音频。长语音请识别
10		VAD_DNN (默认)		高级	新一代VAD，各方面
11		VAD_TOUCH		不常用	关闭语音活动检测。说话松手停止的场景音
12	BDS_ASR_ENABLE_LONG_SPEECH	boolean	全部	常用	是否开启长语音。即和VAD=touch联用！
13		true			开启长语音
14		false （默认）			关闭长语音，无法识别
15	VAD_ENDPOINT_TIMEOUT	int	全部	高级	静音超时断句及长语音
16		0	在线	常用	开启长语音。即无静音联用！
17		800ms	全部	高级	开启长语音。即开启
18	IN_FILE	String： 文件 路径 资源路径或回调方法名	全部	高级	该参数支持设置为： a. pcm文件，系统跟 b. pcm文件, JAVA资源 超过3分钟 c. InputStream数据流 #com.test.Factory.create InputStream的方法create 为： public static InputStream 每次read中sleep，返回 d.wav文件，建议8k采样 为16k pcm文件 后设置
19					9.1 音频格式转换
	OUT_FILE	String ： 文件 路径	全部	高级	保存识别过程产生的
20	AUDIO_MILLS	int： 毫秒	全部	高级	录音开始的时间点。 (System.currentTimeMillis)
21	NLU	String	本地语义	高级	本地语义解析设置。 生效，无论网络状况
22					

23		disable (默认)		高级	禁用
24		enable		高级	启用
25		enable-all		不常用	在enable的基础上,
26	ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH	String: 文件路径 支持assets路径	本地语义	高级	用于支持本地语义解NLU参数。注意bsg
27	SLOT_DATA	String (JSON格式)	本地语义	高级	与ASR_OFFLINE_E中动态扩展本地语义格式参见代码示例或
28	DISABLE_PUNCTUATION	boolean	在线	不常用	在选择"cmn-Hans-C
29		true			禁用标点
30		false (默认)			不禁用标点, 无法使
31	PUNCTUATION_MODE	int	在线	不常用	在选择"cmn-Hans-C要设置DISABLE_PL
32		0 (默认)	在线		打开后处理标点
33		1	在线		关闭后处理标点
34		2	在线		删除句末标点
35		3	在线		将所有标点替换为空
36	ACCEPT_AUDIO_DATA	boolean	全部	高级	是否需要语音音频数
37		true			需要音频数据回调
38		false (默认)			不需要音频数据回调
39	ACCEPT_AUDIO_VOLUME	boolean	全部	高级	是否需要语音音量数
40		true (默认)			需要音量数据回调
41		false			不需要音量数据回调
42	SOUND_START	int: 资源ID	全部	不常用	说话开始的提示音
43	SOUND_END	int: 资源ID	全部	不常用	说话结束的提示音
44	SOUND_SUCCESS	int: 资源ID	全部	不常用	识别成功的提示音
45	SOUND_ERROR	int: 资源ID	全部	不常用	识别出错的提示音
46	SOUND_CANCEL	int: 资源ID	全部	不常用	识别取消的提示音
47	SAMPLE_RATE	int	全部	基本不用	采样率, 固定及默认

48		16000（默认）			
49	ASR_OFFLINE_ENGINE_LICENSE_FILE_PATH	String：文件路径，支持assets路径	离线命令词	基本不用	临时授权文件路径。网创建应用时下载通使用正式授权时请确和唤醒词功能需要使

ASR_KWS_LOAD_ENGINE 输入事件参数

1	事件参数	类型	值	场景	常用程度	描述
2	SLOT_DATA	String	JSON格式	本地语义	高级	与ASR_OFFLINE_E离线命令词bsg文件demo
3	DECODER	int	2			固定值：2，离在线
4	ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH	String	文件路径，支持assets路径			用于支持离线命令词其它说明见NLU参数语义解析设置，在织官网的应用里设置“

LANGUAGE 参数

在线参数， 请根据语言， 来选择LANGUAGE。

- 语言：目前支持中文普通话， 四川话， 粤语， 和英语四个
- 输入法模型：适用于较长的句子输入。默认有标点; 开启标点后， 不支持本地语义。
- 自训练平台模型： 在输入法模型的基础上， 可以自行上传词库和句库， 生成您自己的训练模型。

语音识别LANGUAGE

1	LANGUAGE	语言	模型	是否有标点
2	"cmn-Hans-CN"	普通话	语音近场识别模型	有标点（逗号）
3	"en-GB"	英语		无标点
4	"yue-Hans-CN"	粤语		有标点（逗号）
5	"sichuan-Hans-CN"	四川话		有标点（逗号）

语音自训练平台

语音自训练平台上，训练实时语音识别基础模型，可以在Android SDK上添加训练上线的模型ID。注意必须填写上线模型给出的LANGUAGE。示例 获取专属模型参数LANGUAGE:"cmn-Hans-CN" modelid:1235，可在SDK参数中填写 LANGUAGE="cmn-Hans-CN"; 同时 LM_ID 设置为1235。

具体功能及使用说明请参考文档“自训练平台手册”

1	LANGUAGE	语言	模型	是否有标点
2	"cmn-Hans-CN"	普通话	输入法模型	逗号
3				

具体功能参数

鉴权信息

描述:APP_ID, APP_KEY, SECRET3个鉴权信息决定是否拥有识别、唤醒、离线命令词等权限

请求示例：设置APP_ID, APP_KEY, SECRET

</>Java

```
1 {"appid":155905xx,"secret": "y7S7hAI894BB3LF1yHYmvQEusxxxxxx", "key": "q2uPyBe6LmWTZ1vb0gxxxxxx", "accept-audio-volume":false}
```

设置语言、模型、自定义词库、语义

描述:LANGUAGE参数决定调用哪个识别模型，即决定了语言，模型。

- 1. 语言共有4种：中文普通话、英语、粤语及四川话。
- 2. 输入法模型：分别适用于长句输入/短句输入。
- 3. 在输入法模型下自定义词库生效。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>LANGUAGE,语种

请求示例：英语 输入法模型

</>Java

```
1 {"LANGUAGE": "en-GB", "accept-audio-volume":false}
```

长语音

****描述:****开启长语音识别功能，此时VAD参数不能设为touch；长语音可以识别数小时的音频。注意选输入法模型。

BDS_ASR_ENABLE_LONG_SPEECH= true 或 VAD_ENDPOINT_TIMEOUT = 0 都可以开启长语音

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>长语音及VAD时长设置

请求参数示例： 开启长语音，不能与VAD=touch联用

</>

Java

```
1 {"enable.long.speech":true,"accept-audio-volume":false}
2 或
3 {"accept-audio-volume":false,"vad.endpoint-timeout":0}
```

设置静音时长进行断句

****描述：**普通识别的录音限制60s。连续xxxms静音后，SDK认为一句话结束。

VAD_ENDPOINT_TIMEOUT=0，作用是静音断句的时长设置，使用长语音功能时不能使用这个参数。

VAD_ENDPOINT_TIMEOUT =0

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>长语音及VAD时长设置

请求示例： 连续800ms静音，表示一句话结束

</>

Java

```
1 {"accept-audio-volume":false,"vad.endpoint-timeout":800}
```

开启/关闭根据静音时长切分句子

描述：默认开启根据静音时长切分句子。通过设置VAD参数的值，当设置VAD=dnn时，表示开启VAD，此时通过设置的静音时长进行断句；开启长语音功能时，VAD必然是开启的状态；当设置VAD=touch时，表示关闭VAD，不能使用长语音功能，限制录音时长60s，在60s内的只能点击停止按钮通过发送停止事件才能停止识别。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>VAD是否开启dnn或touch

请求示例： 开启VAD，根据静音时长切分句子

</>

Java

```
1 {"accept-audio-volume":false,"vad":"dnn"}
```

请求示例： 关闭VAD，60s内音频，SDK等待调用stop事件结束录音

</>

Java

```
1 {"accept-audio-volume":false,"vad":"touch"}
```

离线命令词

设置纯在线功能和离在线融合识别

DECODER 参数

- DECODER = 0，表示禁用离线功能，只使用在线功能；
- DECODER = 2，表示启用离线功能，但是SDK强制在线优先。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 离线命令词功能=>开始

请求示例：DECODER=2，表示断网时，使用离线识别固定短语；

</>

Java

```
1 {"accept-audio-volume":false,"decoder":2}
```

设定离线命令词文件路径

离线命令词只能识别bsg文件中预定义的固定短语，其中bsg文件必须在项目中设定路径，参数 **ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH**作用就是定义离线命令词识别的bsg文件路径;该参数需要在**ASR_KWS_LOAD_ENGINE**加载离线资源输入事件中初始化；

ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH设置bsg文件路径

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 离线命令词功能

请求示例：定义assets目录下的离线命令词文件

</>

Java

```
1 {"accept-audio-  
data":false,"grammar":"assets:\\\\baidu_speech_grammar.bsg","decoder":2}
```

扩展离线命令词的词条部分

离线命令词bsg文件的左侧词条部分内容可通过**SLOT_DATA**参数进行动态覆盖，覆盖后原先的bsg文件中的左侧词条部分失效。和**ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH**一起使用。

SLOT_DATA扩展词条

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击全部识别=>设置=>纯在线或在线+离线命令词=>离线命令词及在线识别混合
=>离线命令词及本地语义

=>扩展词条

请求示例：动态修改离线命令词的词条部分

</>Java

```
1 {"accept-audio-  
  data":false,"grammar":"asset:\\\\baidu_speech_grammar.bsg","decoder":2, "slot-  
  data":{"name":["妈妈","老伍"],"appname":["手百","度秘"]}}
```

唤醒输入参数

以下参数均为SpeechConstant类的常量，如SpeechConstant.WAKEUP_START

1	事件名	类型	值	描述
2	WAKEUP_START	json格式的字符串	json内的参数见下文“WAKEUP_START参数”	开始识别唤醒词
3	WAKEUP_STOP			停止识别唤醒词

WAKEUP_START 输入事件参数

1	事件参数	类型	常用程度	描述
2	APP_ID	String	推荐	开放平台创建应用后分配的鉴权信息，填写后会覆盖 An 上线后请使用此参数填写鉴权信息。
3	APP_KEY	String	推荐	开放平台创建应用后分配的鉴权信息，填写后会覆盖 An 上线后请使用此参数填写鉴权信息。
4	SECRET	String	推荐	开放平台创建应用后分配的鉴权信息，填写后会覆盖 An 上线后请使用此参数填写鉴权信息。
5	WP_WORDS_FILE	String	常用	唤醒词bin文件路径，支持android asset目录（如assets:
6	IN_FILE	String: 文件路径 资源路径 或回调方 法名	全部	该参数支持设置为： a. pcm文件，系统路径，如：/sdcard/test/test.pcm；音 b. pcm文件，JAVA资源路径，如：res:///com/baidu.test/ c. InputStream数据流，#方法全名的字符串，格式如：“ 返回InputStream的方法create16kInputStream()”，注 create16KInputStream()。超过3分钟的录音文件，请在 d.wav文件，建议8k采样率，设置前需要调用convertWa 9.1 音频格式转换
7	ACCEPT_AUDIO _DATA	boolean	基本不用	默认关闭。开启后，会有音频回调（CALLBACK_EVEN

8	WP_ENGINE_LIC NSE_FILE_PATH	string	基本不用	不填写，在联网时会获取自动获取离线正式授权。有特殊 (如assets:///mylicense.dat)
9	SAMPLE_RATE	int	基本不用	16000（默认值，且唤醒仅支持16k采样）

具体功能参数

设置唤醒词文件路径

唤醒词功能是本地功能使用时无需联网，只能唤醒bin文件中预定义的关键词，bin文件需要使用WP_WORDS_FILE设置路径，支持android asset目录（如assets:///wakeUp.bin）；

WP_WORDS_FILE设置bin文件路径

DEMO中测试方法：（具体可参见识别安卓SDK测试文档）

demo测试：精简版唤醒

请求示例：使用唤醒功能加载唤醒词bin文件；

</>Java

```
1 {"kws-file":"assets:\\\\WakeUp.bin","accept-audio-volume":false}
```

语义输入参数

- 请求参数说明

1	字段	是否必选	类型	备注
2	bot_session_list	是	string (json)	至少有1个元素
3	bot_session_list[].bot_session_id	是	string	技能的session信息，由系统自动生成，当用户意图已经切换且下一轮会话不需

具体功能参数

本地语义

必须设置ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数生效，无论网络状况，都可以有本地语义结果。

NLU= enable表示开启本地语义

DEMO中测试方法：（具体可参见识别安卓SDK测试文档）

Demo测试：在线和本地语义=>设置=>本地语义解析enable && 本地语义文件 && 扩展词条

请求示例：设置本地语义的bsg文件路径，例如在asset目录下；

</>Java

```
1 {"accept-audio-  
  data":false,"grammar":"assets:\\\\baidu_speech_grammar.bsg","LANGUAGE":"cmn-  
  Hans-CN","slot-data":{"name":["妈妈","老伍"],"appname":["手百","度  
  秘"]},"nlu":"enable"}
```

- 设置本地语义文件路径

使用本地语义需要设置bsg文件的路径，本地语义共用离线命令词的参数和文件。使用本地语义功能的时候，不需要作为ASR_KWS_LOAD_ENGINE的输入参数。但是需要作为**ASR_START**事件的输入参数。

ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数设置bsg文件路径。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

Demo测试：在线和本地语义=>设置=>本地语义文件

请求示例：设置本地语义的bsg文件路径，例如在asset目录下；

```
</> Java |  
  
1 {"accept-audio-  
  data":false,"grammar":"assets:\\\\baidu_speech_grammar.bsg","LANGUAGE":"cmn-  
  Hans-CN","nlu":"enable"}
```

- 扩展本地语义文件的词条部分内容

bsg文件的左侧词条部分内容可通过**SLOT_DATA**参数进行动态覆盖，覆盖后原先的bsg文件中的左侧词条部分失效。和**ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH**一起使用。

SLOT_DATA参数动态覆盖bsg文件词条内容

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

Demo测试：在线和本地语义设置本地语义文件，扩展词条

请求示例：动态覆盖bsg文件左侧词条内容

```
</> Java |  
  
1 {"accept-audio-  
  data":false,"grammar":"assets:\\\\baidu_speech_grammar.bsg","LANGUAGE":"cmn-  
  Hans-CN","slot-data":{"name":["妈妈","老伍"],"appname":["手百","度秘"]},"  
  nlu":"enable"}
```

5.输出参数

识别输出参数

语音回调事件统一由 `public void onEvent(String name, String params, byte[] data, int offset, int length)` 该方法回调。其中name是回调事件， params是回调参数。（data， offset， length） 缓存临时数据， 三者一起， 生效部分为 data[offset] 开始， 长度为length。

1	事件名 (name)	事件参数	类型	值	描述
2	CALLBACK_EVENT _ASR_READY				引擎准备就绪
3	CALLBACK_EVENT _ASR_BEGIN				检测到第一帧语音数据
4	CALLBACK_EVENT _ASR_END				检测到最后一帧语音数据
5	CALLBACK_EVENT _ASR_PARTIAL	params	json		识别结果
6		params[results_recognition]	String[]		解析后的识别结果
7		params[result_type]	String	partial_result	临时识别结果
8		params[result_type]	String	final_result	最终结果，长语音识别
9		params[result_type]	String	nlu_result	语义结果，长语音识别
10		(data, offset, length)	String		语义结果，长语音识别
11	CALLBACK_EVENT _ASR_FINISH	params	String(json格式)		一句话识别结果
12		params[error]	int		错误领域
13		params[sub_error]	int		错误码
14		params[desc]	String		错误描述
15	CALLBACK_EVENT _ASR_LONG_SPEECH				长语音额度
16	CALLBACK_EVENT _ASR_EXIT				识别结束
17	CALLBACK_EVENT _ASR_AUDIO	(data, offset, length)	byte[]		PCM音频
18	CALLBACK_EVENT _ASR_VOLUME	params	json		当前音量
19		params[volume]	float		当前音量

20		params[volume-percent]	int		当前音量
21	CALLBACK_EVENT _ASR_LOADED				离线模型
22	CALLBACK_EVENT _ASR_UNLOADED				离线模型
23					

唤醒输出参数

语音回调事件统一由 `public void onEvent(String name, String params, byte[] data, int offset, int length)` 方法回调
其中name是回调事件， params是回调参数。（data， offset， length）缓存临时数据，三者一起，生效部分为 data[offset] 开始，长度为length。

1	事件名	事件参数	类型	描述	
2	CALLBACK_EVENT _WAKEUP_STARTED			引擎开始运行	
3	CALLBACK_EVENT _WAKEUP_AUDIO	(data,offset,length)	byte[]	PCM音频片段回调， 需要输入ACCEPT_AUDIO_DATA参数激活。保存的pcm文件的采样率是16000， 16bits， 单声道， 小端序。	
4	CALLBACK_EVENT _WAKEUP_SUCCESS			唤醒成功	
5		errorCode		错误码,错误码为0表示唤醒成功， 唤醒出错会在CALLBACK_EVENT_WAKEUP_ERROR事件中	
6		errorDesc		错误描述,此处固定为 success	
7		word	String	具体的唤醒词	
8	CALLBACK_EVENT _WAKEUP_ERROR	params	String(json格式)	错误描述的回调	
9		params[desc]	int	错误描述	
10	CALLBACK_EVENT _WAKEUP_STOPED			唤醒已关闭	

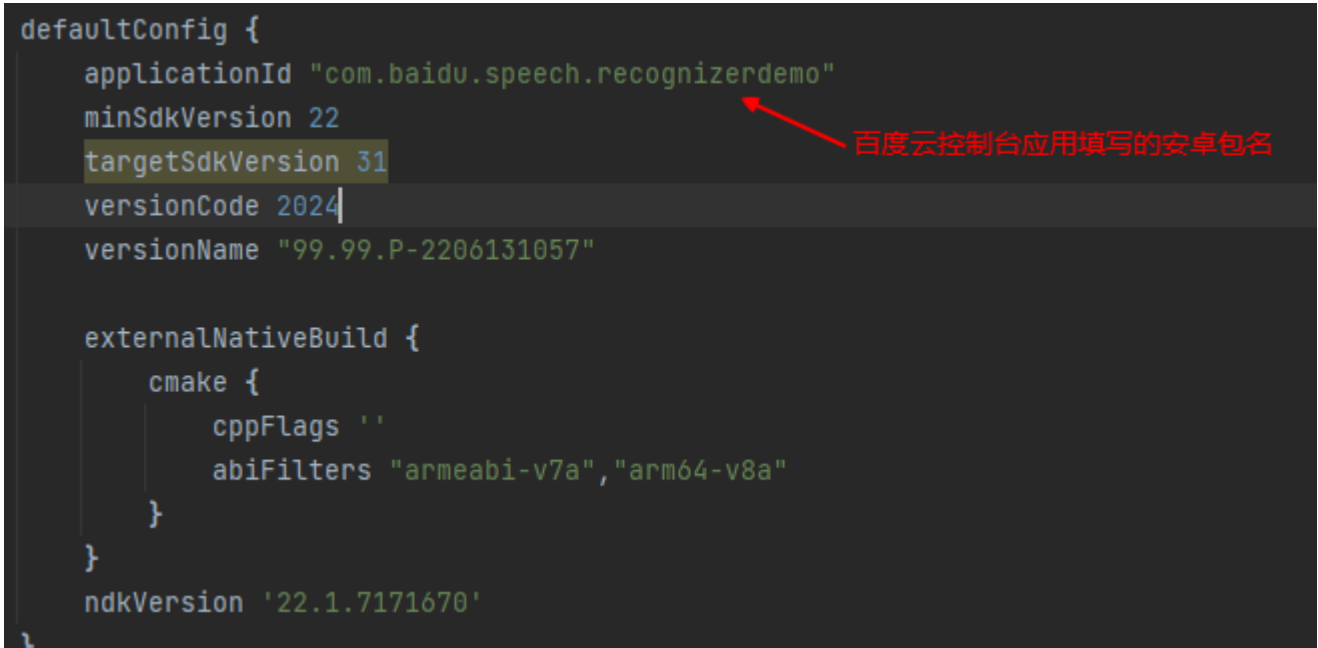
6. Demo运行

6.1 配置包名和签名

从百度云控制台下载Demo之后，需要在build.gradle中配置好包名。

```
defaultConfig {
    applicationId "com.baidu.speech.recognizerdemo"
    minSdkVersion 22
    targetSdkVersion 31
    versionCode 2024
    versionName "99.99.P-2206131057"

    externalNativeBuild {
        cmake {
            cppFlags ''
            abiFilters "armeabi-v7a","arm64-v8a"
        }
    }
    ndkVersion '22.1.7171670'
}
```



6.2 修改鉴权信息

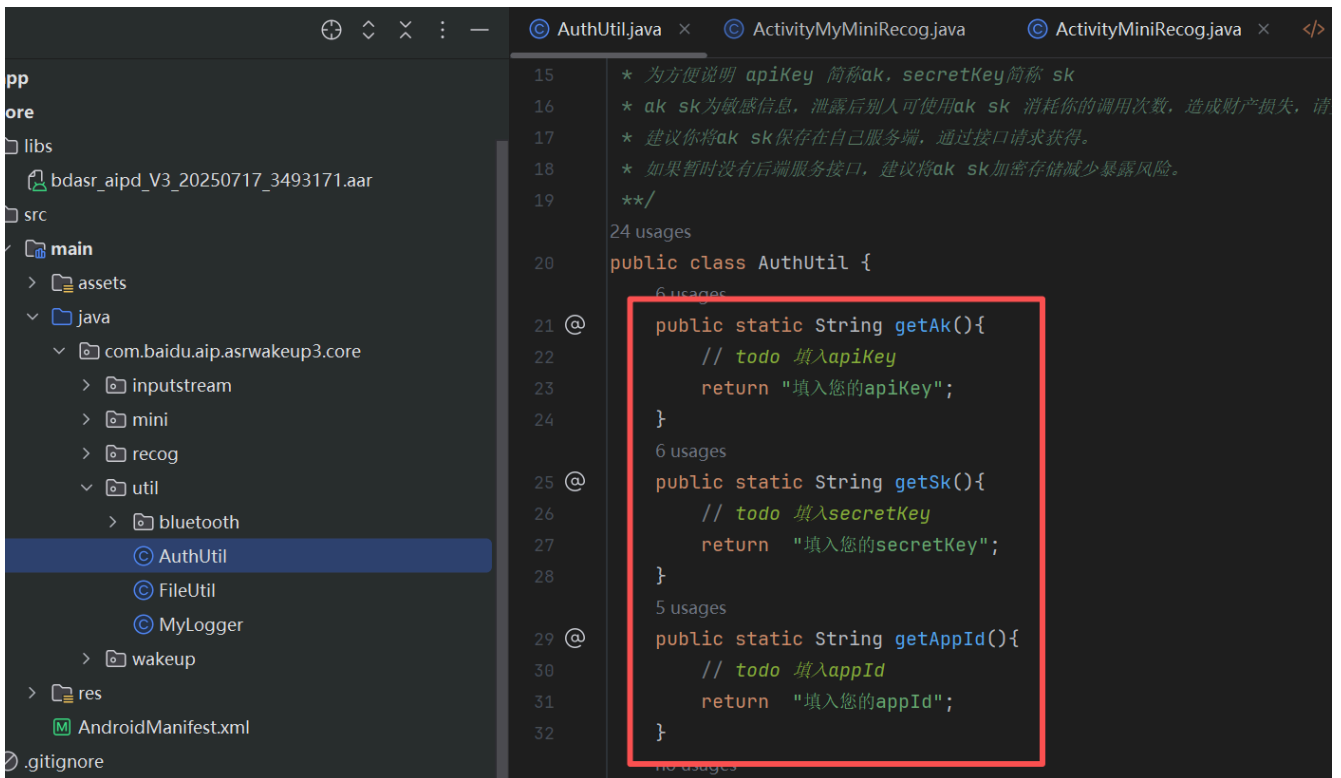
sdk分为4种激活方式，实际使用种选择其中一种方式进行激活，4种激活方法详细描述如下：

1	激活方式	说明
2	Appkey SecretKey激活	在百度云网站申请自己的语音合成应用后，会有appKey、appSecret鉴权信息活一次，不退出app永久有效
3	永久iamKey激活	需在官网申请的永久iamKey，使用该方法进行在线鉴权，每次启动app需激活
4	accessToken激活	需在使用appkey和secretKey获取的access_token，使用access_token进行授权新的token从新授权
5	临时iamKey激活	与accessToken类似，需获取临时iamKey，该key存在时限，时间到期后需要

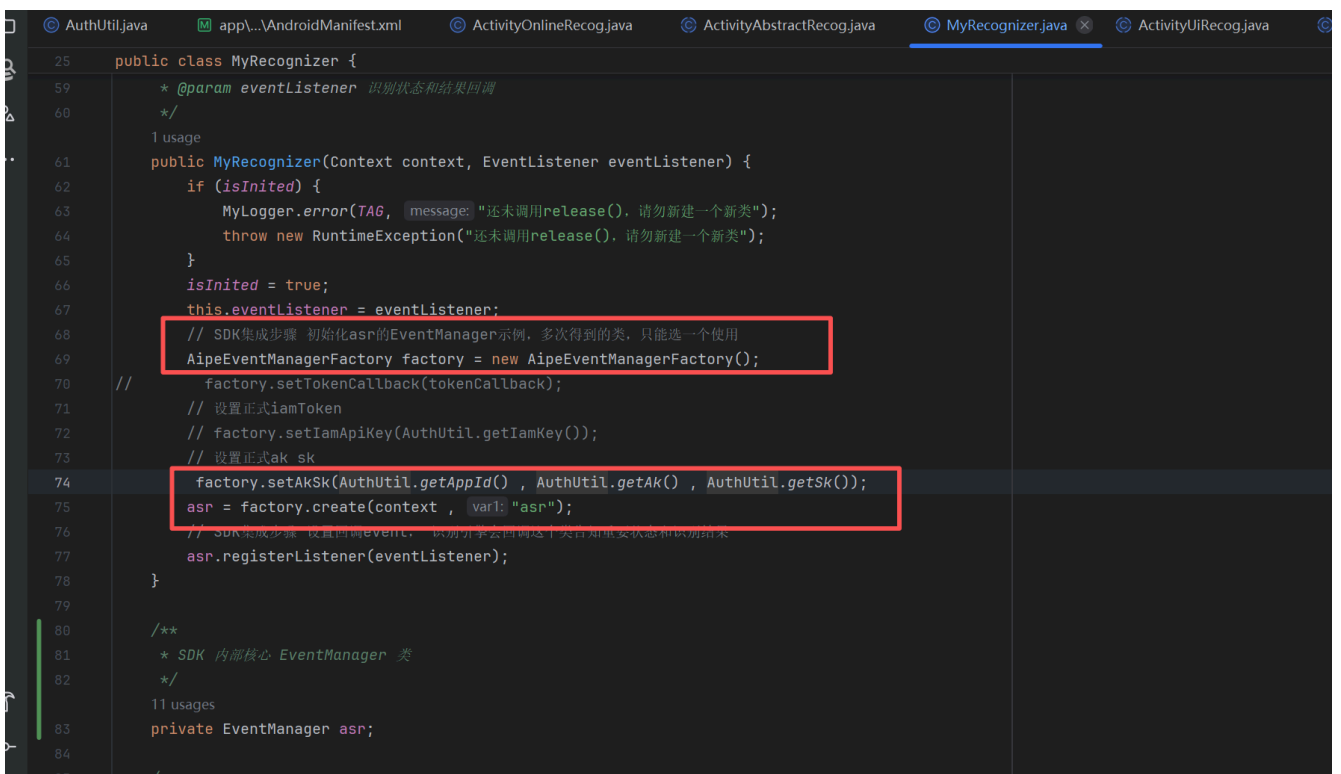
1.AppKey SecretKey激活：

百度云网站上申请自己语音识别应用，会有appId、appKey、appSecret及android包名 4个鉴权信息，申请地址为[百度云应用列表](#)

申请成功后在AuthUtil类中填写appid、 apiKey和 secretKey 进行授权：



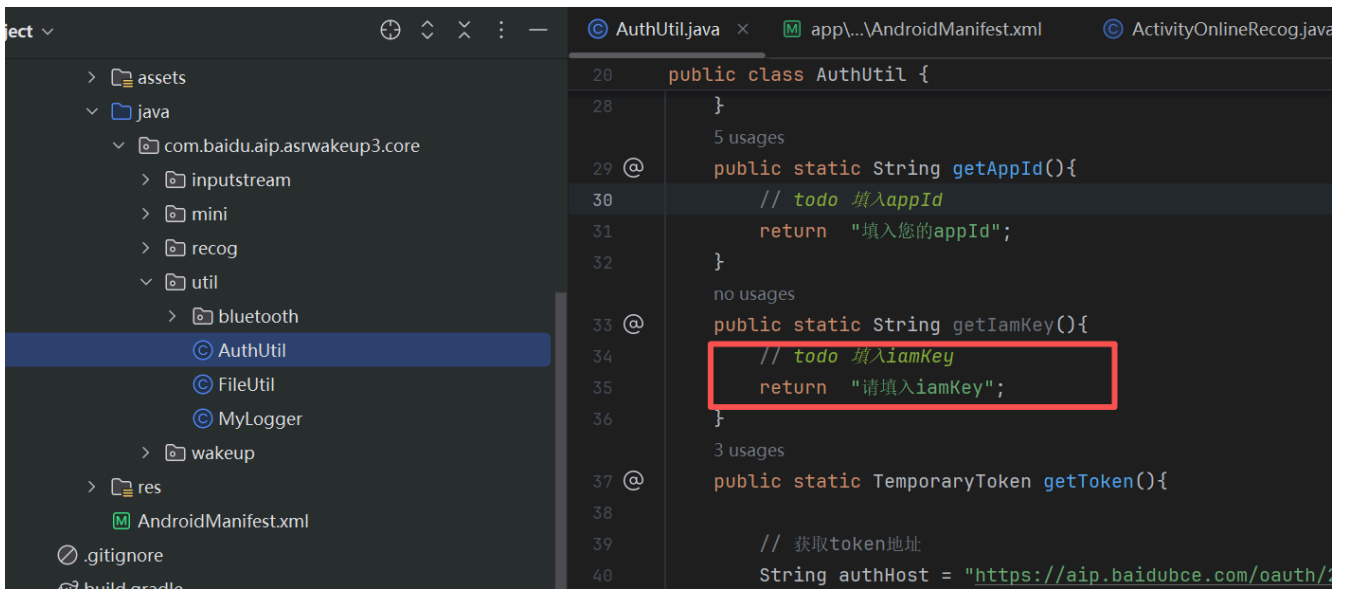
ak、sk、appid修改成功后, 创建AipeEventManagerFactory传入ak、sk、appid并创建核心EventManager 类



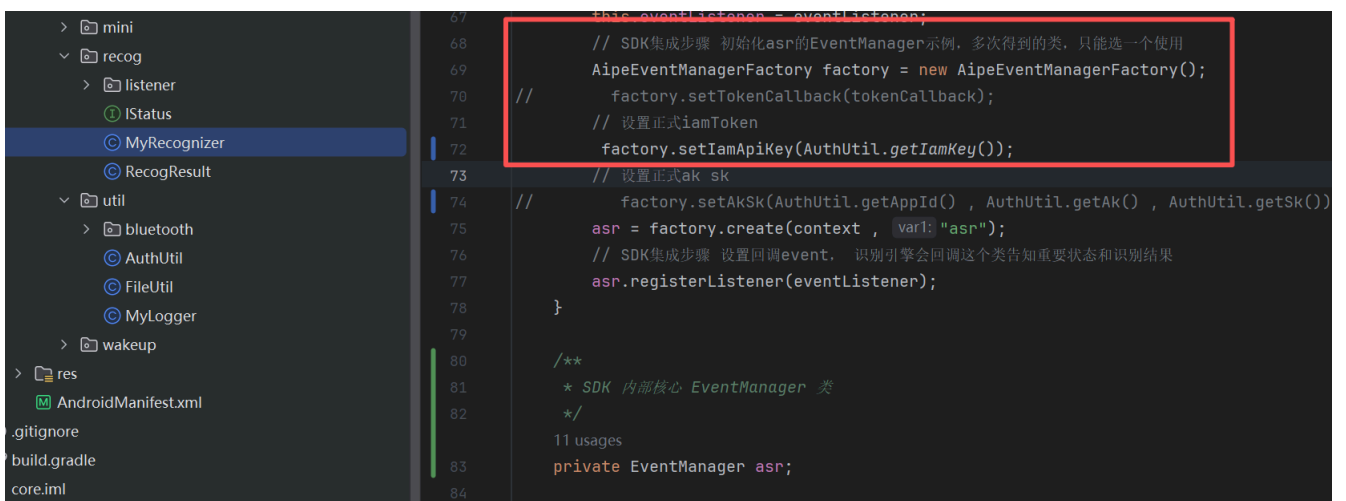
2.永久iamKey激活:

“永久iamKey激活”和“appKey secretKey激活”类似。

将官网申请的永久iamKey放到项目中



iamKey修改成功后，创建AipeEventManagerFactory传入iamKey创建核心EventManager类



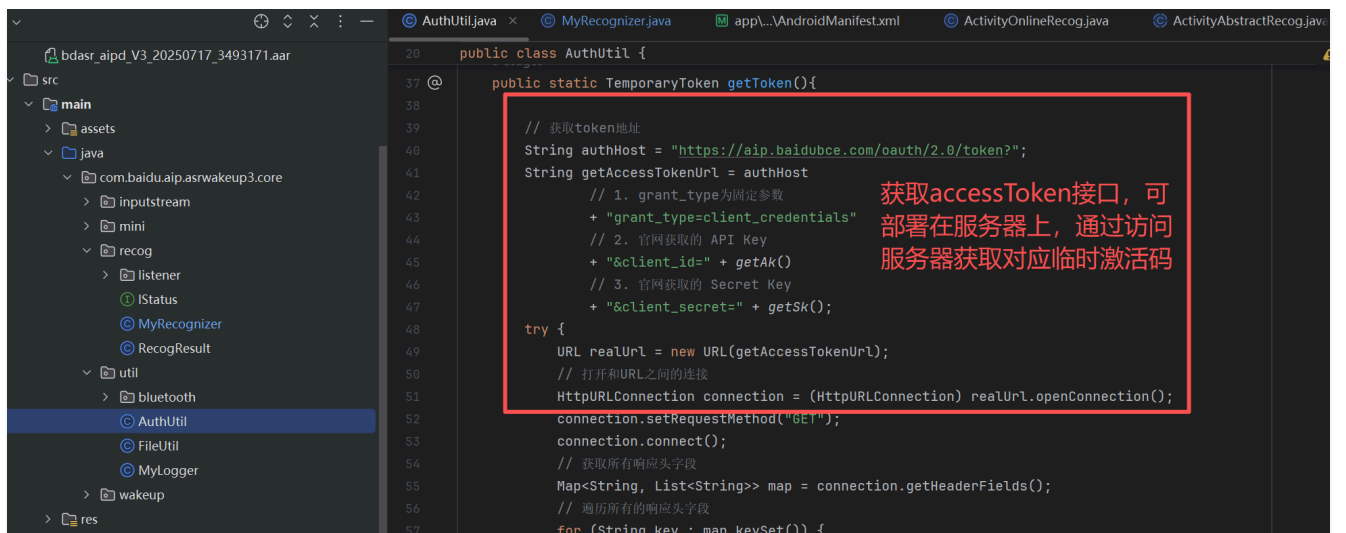
3.临时accessToken和临时iam key激活方式

accessToken和临时iamKey为相同接口，接口会随时间失效，客户需要在callback中创建http请求获取到对应token返回给sdk,sdk会在过期前3分钟通过callback重新向客户端获取最新token。

accessToken获取地址：<https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu>

iamKey获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Hm5us339w

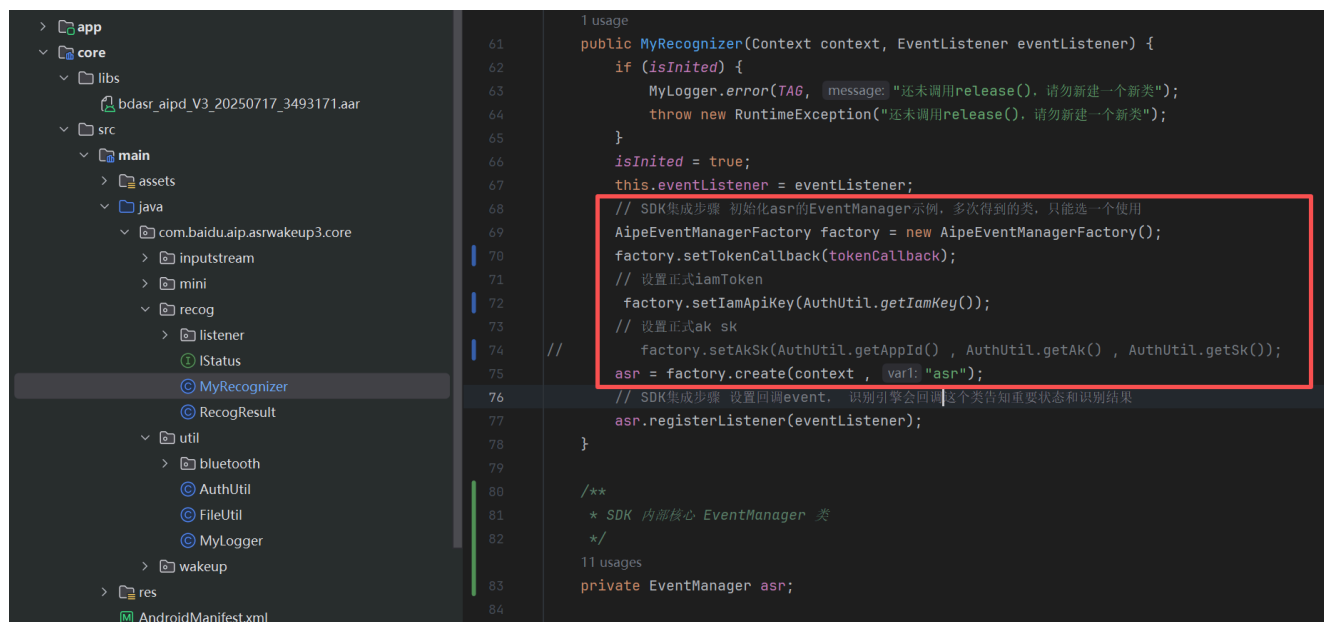
获取临时accessToken/iamKey:



创建token监听接口

```
6    }
7
8    1 usage
9    TokenCallback tokenCallback = new TokenCallback() {
10        @Override
11        public TemporaryToken getToken() {
12            return AuthUtil.getToken();
13        }
14    };
15    /**
16     * 初始化 提供 EventManagerFactory 需要的Context 和EventListener
```

修改成功后，创建AipeEventManagerFactory传入token监听接口创建核心 EventManager 类



```
61    1 usage
62    public MyRecognizer(Context context, EventListener eventListener) {
63        if (isInitiated) {
64            MyLogger.error(TAG, "message: "还未调用release(), 请勿新建一个新类");
65            throw new RuntimeException("还未调用release(), 请勿新建一个新类");
66        }
67        isInitiated = true;
68        this.eventListener = eventListener;
69        // SDK集成步骤 初始化asr的EventManager示例，多次得到的类，只能选一个使用
70        AipeEventManagerFactory factory = new AipeEventManagerFactory();
71        factory.setTokenCallback(tokenCallback);
72        // 设置正式iamToken
73        factory.setIamApiKey(AuthUtil.getIamKey());
74        // 设置正式ak sk
75        factory.setAkSk(AuthUtil.getAppId(), AuthUtil.getAk(), AuthUtil.getSk());
76        asr = factory.create(context, "asr");
77        // SDK集成步骤 设置回调event， 识别引擎会回调这个类告知重要状态和识别结果
78        asr.registerListener(eventListener);
79    }
80    /**
81     * SDK 内部核心 EventManager 类
82     */
83    11 usages
84    private EventManager asr;
```

然后编译运行



7. SDK集成

bdasr_V3_xxxxxxx_xxxx.jar 库

将core/libs/bdasr_V3_xxxxx_xxxx.jar 复制到您的项目的同名目录中。

复制NDK 架构目录

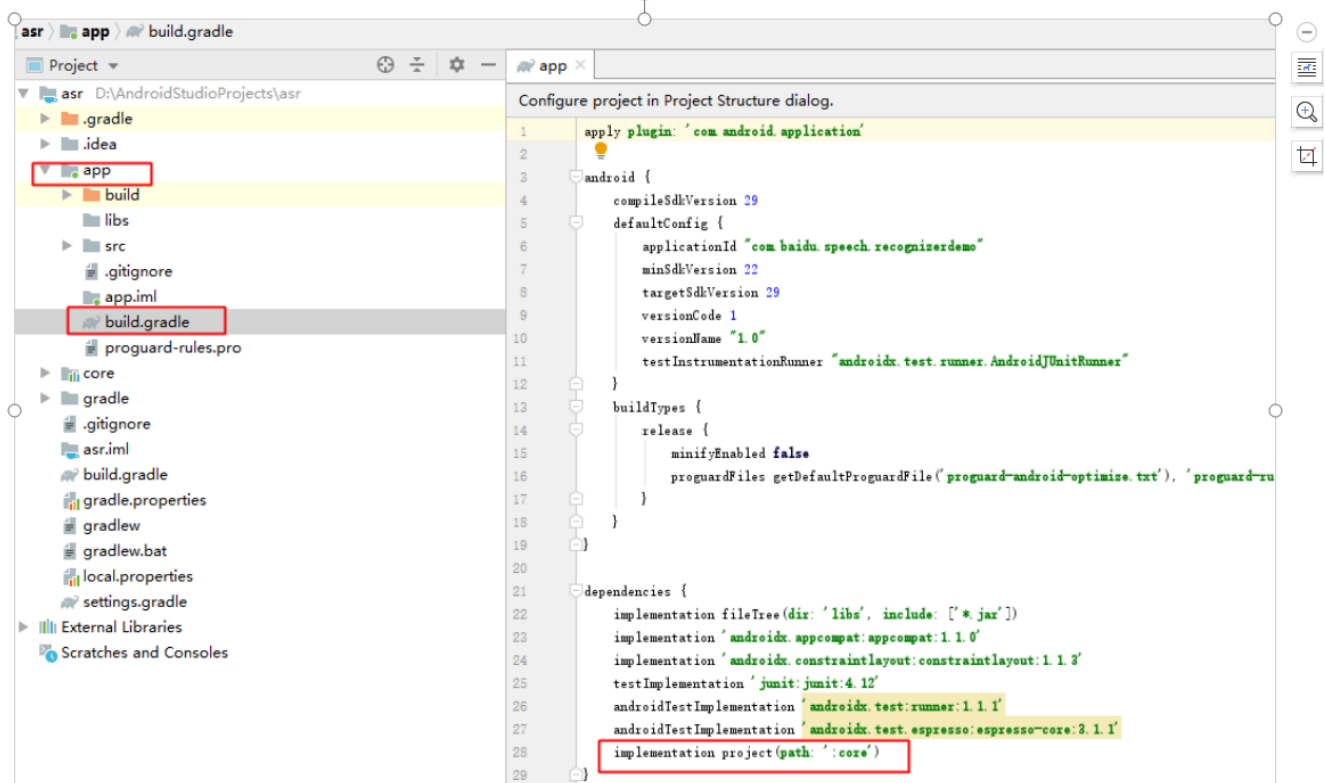
1. 将 core/src/main/jniLibs 下armeabi等包含so文件的5个目录，复制合并到您的项目的同名或者存放so的目录中。如果build.gradle中定义过jniLibs.srcDirs，则复制合并到这个目录。

2. 如与第三方库集成, 至少要保留armeabi目录。如第三方库有7个架构目录, 比语音识别SDK多出2个目录 mips和mips64, 请将mips和mips64目录删除, 剩下5个同名目录合并。
3. 如第三方库仅有armeabi这一个目录, 请将语音识别SDK的额外4个目录如armeabi-v7a删除, 合并armeabi目录下的so。即目录取交集, so文件不可随意更改所属目录。
4. 打包成apk文件, 按照zip格式解压出libs目录可以验证。
5. 运行时 `getApplicationInfo().nativeLibraryDir` 目录下查看是否有完整so文件。特别是**系统app**需要手动push so文件到这个目录下。

build.gradle 文件及包名确认

1. 根目录下build.gradle确认下gradle的版本。
2. app/build.gradle 确认下 `applicationId` 包名是否与官网申请应用时相一致 (离线功能需要)。demo的包名是"com.baidu.speech.recognizerdemo"。
3. 确认 `compileSdkVersion` `buildToolsVersion` 及 `targetSdkVersion`
4. 一、导入demo的core module, 并配置app依赖core

`implementation project(path: ':core')`



二、修改app/java/com.baidu.speech.recognizerdemo/MainActivity.java, 修改app/java/com.baidu.speech.recognizerdemo/MainActivity.java:

```
import com.baidu.aip.asr.wakeup3.core.mini.ActivityMiniRecog;  
  
public class MainActivity extends ActivityMiniRecog {  
  
}
```

```
package com.baidu.speech.recognizerdemo;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import com.baidu.aip.asr.wakeup3.core.mini.ActivityMiniRecog;  
  
public class MainActivity extends ActivityMiniRecog {  
  
}
```

具体参考官方demo内 doc_integration_DOCUMENT文件夹下有ASR-INTEGRATION-helloworld-V3.01.docx文件，helloworld 集成sdk的完整图示实例。

8. 相关授权文件

请将百度云控制台创建应用时获取的语音(APPID)、API/SECRET KEY 并填写包名。
在线识别与唤醒都需要进行相关验证后方可使用：

1	引擎类型	验证方法
2	在线识别	开放平台使用API/SECRET KEY + APPID 进行验证
3	离线识别	使用APPID+包名首次联网自动下载授权文件进行验证
4	唤醒引擎	与离线识别验证方法一致

9. 语音相关接口流程

9.1在线识别调用流程

1 初始化

1.1创建AipeEventManagerFactory工厂对象，设置激活参数，创建EventManager对象

SDK中，eventManager通过工厂创建语音识别的事件管理器。注意识别事件管理器只能维持一个，请勿同时使用多个实例。即创建一个新的识别事件管理器后，之前的那个置为null，并不再使用。

</>

Java

```

1    // 临时iam key 和 accessToken 接口回调，sdk会在剩余时间小于3分钟时重新调用该接口
    获取最新token
2    TokenCallback tokenCallback = new TokenCallback() {
3        @Override
4        public TemporaryToken getToken() {
5            return AuthUtil.getToken();
6        }
7
8    };
9    /**
10     * SDK 内部核心 EventManager 类
11     */
12    private EventManager asr;
13    public void init() {
14        AipeEventManagerFactory factory = new AipeEventManagerFactory();
15        // 临时iamToken 或 临时accessToken激活
16        // factory.setTokenCallback(tokenCallback);
17        // 正式iamToken激活
18        // factory.setIamApiKey(AuthUtil.getIamKey());
19        // 设置正式ak sk激活
20        factory.setAkSk(AuthUtil.getAppId() , AuthUtil.getAk() ,
        AuthUtil.getSk());
21    }
22    asr = factory.create(context , "asr");
23    }

```

xt类

详见ActivityMiniRecog类中“基于sdk集成1.1 初始化EventManager对象”

1.2 自定义输出事件类

SDK中，需要实现EventListener的输出事件回调接口。该类需要处理SDK在识别过程中的回调事件。可以参考DEMO中对SDK的调用封装。

```

</> Java |
1    EventListener yourListener = new EventListener() {
2        @Override
3        public void onEvent(String name, String params, byte [] data, int offset,
        int length) {
4            if(name.equals(SpeechConstant.CALLBACK_EVENT_ASR_READY)){
5                // 引擎就绪，可以说话，一般在收到此事件后通过UI通知用户可以说话了
6            }
7            if(name.equals(SpeechConstant.CALLBACK_EVENT_ASR_PARTIAL)){
8                // 一句话的临时结果，最终结果及语义结果

```

```
9      }  
10     // ... 支持的输出事件和事件支持的事件参数见“输入和输出参数”一节  
11   }  
12 };
```

详见ActivityMiniRecog类中“基于sdk集成1.2 自定义输出事件类”

1.3 注册自己的输出事件类

```
</> Java |  
  
1 asr.registerListener(yourListener);
```

详见ActivityMiniRecog类中“基于sdk集成1.3 注册自己的输出事件类”

DEMO中，以上两步合并为

```
</> Java |  
  
1 IRecogListener listener = new MessageStatusRecogListener(handler);  
2  
3 // 可以传入IRecogListener的实现类，也可以如SDK，传入EventListener实现类
```

```
</> Java |  
  
1 //如果传入IRecogListener类，在RecogEventAdapter为您做了大多数的json解析。  
2  
3 MyRecognizer myRecognizer = new MyRecognizer(this, listener);  
4 //this是Activity或其它Context类
```

详见ActivityAbstractRecog类中“基于DEMO集成第1.1, 1.2, 1.3 步骤 初始化EventManager类并注册自定义输出事件”

1.4 音频格式转换

- 音频格式转换器：将 WAV 格式音频文件转换为 PCM 格式，不改变采样率
- 音频重采样器：将任意采样率的 WAV 格式音频文件重采样到 16k 采样率

```
</> Plain Text |  
  
1 /**  
2  * 将 WAV 文件转换为 16k PCM 文件  
3  *  
4  * @param wavFilePath 输入的 WAV 文件路径  
5  * @param pcmFilePath 输出的 PCM 文件路径  
6  * @param wavSampleRate WAV 文件的采样率  
7  * @return true 转换成功, false 转换失败  
8  */
```

```
9 public static boolean convertWavTo16kPcm(String wavFilePath, String pcmFilePath,
    int wavSampleRate)
```

</>

Plain Text

```
1 if (new File(TEST_WAV_FILE_PATH).exists()) {
2     if (!AudioProcessingUtils.convertWavTo16kPcm(TEST_WAV_FILE_PATH,
        TEST_PCM_FILE_PATH, 8000)) {
3         Toast.makeText(this, "WAV 转换 PCM 失败", Toast.LENGTH_SHORT).show();
4         return;
5     }
6 }
7 if (!new File(TEST_PCM_FILE_PATH).exists()) {
8     Toast.makeText(this, "未找到灌测音频: " + TEST_PCM_FILE_PATH + ", 使用麦克风",
        Toast.LENGTH_SHORT).show();
9 } else {
10     Toast.makeText(this, "找到灌测音频: " + TEST_PCM_FILE_PATH,
        Toast.LENGTH_SHORT).show();
11     /* 传入反射方法 audio() */
12     params.put("infile", String.format("#%s.audio()",
        ActivityLongSpeech.class.getName()));
13 }
14 // ...
15 public static InputStream audio() throws FileNotFoundException {
16     return new FileAudioInputStream(TEST_PCM_FILE_PATH);
17 }
```

如果需要独立使用该接口，即在初始化 SDK (AipeEventManagerFactory.create(...)) 之前调用该接口，需要先运行以下代码以加载音频转换依赖的 Native 库：

</>

Plain Text

```
1 System.loadLibrary("BaiduSpeechSDK");
```

2 开始识别

开始事件的参数可以参见“输入和输出参数”。

2.1 设置识别输入参数

SDK中，您需要根据文档或者demo确定您的输入参数。DEMO中有UI界面简化选择和测试过程。demo中，在点击“开始录音”按钮后，您可以在界面或者日志中看见ASR_START事件的json格式的参数。

</>

Java

```
1 // asr.params(反馈请带上此行日志):{"accept-audio-data":false,"disable-
    punctuation":false,"accept-audio-volume":true,"LANGUAGE":"en-GB"}
```

```
2
3 //其中{"accept-audio-data":false,"disable-punctuation":false,"accept-audio-
  volume":true,"LANGUAGE":"en-GB"}为ASR_START事件的参数
4
5 String json = "{\"accept-audio-data\":false,\"disable-
  punctuation\":false,\"accept-audio-volume\":true,\"LANGUAGE\":\"en-GB\"}"
```

详见ActivityMiniRecog类中“基于SDK集成2.1 设置识别参数”

2.2 发送start开始事件

```
</> Java |
1 asr.send(SpeechConstant.ASR_START, json, null, 0, 0);
```

详见ActivityMiniRecog类中“基于SDK集成2.2 发送开始事件”

DEMO中，您需要传递Map<String,Object>的参数，会将Map自动序列化为json

```
</> Java |
1 Map<String, Object> params;
2 ... // 设置识别参数
3 // params ="accept-audio-data":false,"disable-punctuation":false,"accept-audio-
  volume":true,"LANGUAGE":"en-GB"
4 myRecognizer.start(params);
```

详见ActivityAbstractRecog类中“基于DEMO集成2.1, 2.2 设置识别参数并发送开始事件”

3 收到回调事件

3.1开始回调事件

回调事件在您实现的EventListener中获取。OnEvent方法中，name是输出事件名，params该事件的参数，(data,offset, length)三者一起组成额外数据。如回调的音频数据，从data[offset]开始至data[offset + length] 结束，长度为length。

```
</> Java |
1 public void onEvent(String name, String params, byte [] data, int offset, int
  length);
```

详见ActivityMiniRecog类中“基于SDK集成3.1 开始回调事件”

DEMO中，回调事件在您实现的IRecogListener中获取。

详见RecogEventAdapter类中“基于DEMO集成3.1 开始回调事件”

4 控制识别

4.1 控制停止识别

```
</> Java |  
  
1 asr.send(SpeechConstant.ASR_STOP, null, null, 0, 0);  
2 //发送停止录音事件，提前结束录音等待识别结果
```

详见ActivityMiniRecog类中"基于SDK集成4.1 发送停止事件"

4.2 控制取消识别

```
</> Java |  
  
1 asr.send(SpeechConstant.ASR_CANCEL, null, null, 0, 0);  
2 //取消本次识别，取消后将立即停止不会返回识别结果
```

详见ActivityMiniRecog类中"基于SDK集成4.2 发送取消事件"

DEMO 中:

```
</> Java |  
  
1 myRecognizer.stop();  
2 // 发送停止录音事件，提前结束录音等待识别结果
```

详见ActivityAbstractRecog类中"基于DEMO集成4.1 发送停止事件"

```
</> Java |  
  
1 myRecognizer.cancel();  
2 // 取消本次识别，取消后将立即停止不会返回识别结果
```

详见ActivityAbstractRecog类中"基于DEMO集成4.2 发送取消事件"

5 事件管理器退出

5.1 在线不需要卸载离线命令词

先启动取消，避免有还在运行的识别。

之后需要将之前的listener卸载，不卸载的话，可能有内存溢出

asr.send(SpeechConstant.ASR_CANCEL, null, null, 0, 0); // 取消识别

5.2 释放资源

```
</> Java |  
  
1 asr.unregisterListener(this);
```

详见ActivityMiniRecog类中基于SDK集成5.2 退出事件管理器"

DEMO中,

```
</>

1 myRecognizer.release();
2 // 含有离线引擎卸载
```

详见ActivityAbstractRecog类中基于DEMO的5.2 退出事件管理器"

9.2 离线命令词调用流程

离线命令词的bsg文件设置：

在语音控制台的左侧功能栏中，进入“离线词&语义设置”模块，根据页面上的引导自行定义词条和语法，并生成bsg文件。其中右侧“说法”部分，为固定语法，下载后不可更改。左侧“词条”部分，代码中可以动态定义覆盖。



离线命令词功能可以测试DEMO中的第二个按钮“离线命令词识别”

调用流程：

离线命令词功能需要首先实现之前的在线识别功能的代码。离线引擎加载需要在EventManager初始化之后，识别事件之前。

在SDK中，

```
</> Java

1 HashMap map = new HashMap();
2 map.put(SpeechConstant.DECODER, 2);
3 // 0:在线 2.离在线融合(在线优先)
4 map.put(SpeechConstant.ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH,
    "/sdcard/yourpath/baidu_speech_grammar.bsg");
```

```
5 //设置离线命令词文件路径
```

// 下面这段可选，用于生成SLOT_DATA参数，
用于动态覆盖ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH文件的词条部分

```
</> Java |  
  
1 JSONObject json = new JSONObject();  
2 json.put("name", new JSONArray().put("王自强").put("叶问")).put("appname", new  
3 JSONArray().put("手百").put("度秘"));  
4 map.put(SpeechConstant.SLOT_DATA, json.toString());  
5 // SLOT_DATA 参数添加完毕
```

1.1 -1.3步骤同在线

1.4 加载离线资源

```
</> Java |  
  
1 asr.send(SpeechConstant.ASR_KWS_LOAD_ENGINE, new  
2 JSONObject(map).toString());
```

//加载离线引擎，使用离线命令词时使用，请在整个离线识别任务结束之后卸载离线引擎

详见ActivityMiniRecog类中”基于SDK离线命令词1.4 加载离线资源(离线时使用)”

//离线引擎加载完毕事件后，开始你的识别流程，此处开始你的识别流程，注意离线必须断网生效或者SDK无法连接百度服务器时生效，只能识别bsg文件里定义的短语。

2.1-4.2 步骤同在线

5.1 卸载离线资源

//不再需要识别功能后，卸载离线引擎。再次需要识别功能的话，可以重复以上步骤。即依旧需要EventManager初始化之后，识别事件之前加载离线引擎。

```
</> Java |  
  
1 asr.send(SpeechConstant.ASR_KWS_UNLOAD_ENGINE, null, null, 0, 0);
```

详见ActivityMiniRecog类中”基于SDK集成5.1 卸载离线资源步骤(离线时使用)”

在demo中，

```
</> Java |  
  
1 HashMap<String, Object> map = new HashMap<String, Object>();  
2 map.put(SpeechConstant.DECODER, 2);  
3 // 0:在线 2.离在线融合(在线优先)  
4 map.put(SpeechConstant.ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH,  
5 "/sdcard/yourpath/baidu_speech_grammar.bsg");
```

```
5 //设置离线命令词文件路径
```

// 下面这段可选，用于生成SLOT_DATA参数，
用于动态覆盖ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH文件的词条部分

</>Java|

```
1 JSONObject json = new JSONObject();
2 json.put("name", new
3 JSONArray().put("王自强").put("叶问")).put("appname", new
4 JSONArray().put("手百").put("度秘"));
5 map.put(SpeechConstant.SLOT_DATA, json.toString());
6 // SLOT_DATA 参数添加完毕
```

</>Java|

```
1 myRecognizer.loadOfflineEngine(map);
2 //加载离线引擎，使用离线命令词时使用，请在整个离线识别任务结束之后卸载离线引擎
```

详见ActivityAbstractRecog类中“基于DEMO集成1.4 加载离线资源步骤(离线时使用)”

//离线引擎加载完毕事件后，开始你的识别流程，注意离线必须断网生效或者SDK无法连接百度服务器时生效，只能识别bsg文件里定义的短语。

//不再需要识别功能后，卸载离线引擎。再次需要识别功能的话，可以重复以上步骤，新建MyRecognizer类即可。

</>Java|

```
1 myRecognizer.release();
```

详见ActivityAbstractRecog类中“基于DEMO5.1 卸载离线资源(离线时使用)”

9.3识别错误码

1	错误领域	描述	错误码	错误描述及可能原因
2	1	网络超时		出现原因可能为网络已经连接但质量比较差，建议检测网络状态
3			1000	DNS连接超时
4			1001	网络连接超时
5			1002	网络读取超时
6			1003	上行网络连接超时
7			1004	上行网络读取超时
8			1005	下行网络连接超时

9			1006	下行网络读取超时
10	2	网络连接失败		出现原因可能是网络权限被禁用，或网络确实未连接，需要开启网络
11			2000	网络连接失败
12			2001	网络读取失败
13			2002	上行网络连接失败
14			2003	上行网络读取失败
15			2004	下行网络连接失败
16			2005	下行网络读取失败
17			2006	下行数据异常
18			2100	本地网络不可用
19	3	音频错误		出现原因可能为：未声明录音权限，或被安全软件限制，或录音设
20			3001	录音机打开失败
21			3002	录音机参数错误
22			3003	录音机不可用
23			3006	录音机读取失败
24			3007	录音机关闭失败
25			3008	文件打开失败
26			3009	文件读取失败
27			3010	文件关闭失败
28			3100	VAD异常，通常是VAD资源设置不正确
29			3101	长时间未检测到人说话，请重新识别
30			3102	检测到人说话，但语音过短
31	4	协议错误		出现原因可能是appid和appkey的鉴权失败
32			4001	协议出错
33			4002	协议出错
34			4003	识别出错
35			4004	鉴权错误，一般情况是LANGUAGE appkey secretkey不正确权限
36	5	客户端调用错误		一般是开发阶段的调用错误，需要开发者检测调用逻辑或对照文档和
37			5001	无法加载so库
38			5002	识别参数有误
39			5003	获取token失败

40			5004	客户端DNS解析失败
41			5005	
42	6	超时		语音过长，请配合语音识别的使用场景，如避开嘈杂的环境等
43			6001	未开启长语音时，当输入语音超过60s时，会报此错误
44	7	没有识别结果		信噪比差，请配合语音识别的使用场景，如避开嘈杂的环境等
45			7001	没有匹配的识别结果。当检测到语音结束，或手动结束时，服务端收结果
46	8	引擎忙		一般是开发阶段的调用错误，出现原因是上一个会话尚未结束，就让任务运行，即便创建多个实例，也只能有一个实例处于工作状态
47			8001	识别引擎繁忙 。当识别正在进行时，再次启动识别，会报busy。
48	9	缺少权限		参见demo中的权限设置
49			9001	没有录音权限 通常是没有配置录音权限： android.permission.RECC
50	10	其它错误		出现原因如：使用离线识别但未将EASR.so集成到程序中；离线授权
51			10001	离线引擎异常
52			10002	没有授权文件
53			10003	授权文件不可用
54			10004	离线参数设置错误
55			10005	引擎没有被初始化
56			10006	模型文件不可用
57			10007	语法文件不可用
58			10008	引擎重置失败
59			10009	引擎初始化失败
60			10010	引擎释放失败
61			10011	引擎不支持
62			10012	离线引擎识别失败 。离线识别引擎只能识别grammar文件中约定好的不如在线。请确保说的话清晰，是grammar中文件定义的，测试成功

"4004"鉴权子错误码

4004的子错误值	错误码描述	原因
-----	-----	-----
4 pv超限	配额使用完毕，请购买或者申请	
6 没勾权限	应用不存或者应用没有语音识别的权限	

|13|并发超限|并发超过限额，请购买或者申请 |
| 101|API key错误| API Key 填错|

服务端错误码请参照 [点击查看](#)

9.4唤醒词调用流程

SDK 的调用过程可以参见DEMO中的ActivityMiniWakeUp类

DEMO的调用过程可以参考DEMO中的ActivityWakeUp类

1 初始化

1.1 初始化EventManager对象

SDK中，通过工厂创建语音唤醒词的事件管理器。注意唤醒词事件管理器同识别事件管理器一样只能维持一个，请勿同时使用多个实例。即创建一个新的唤醒词事件管理器后，之前的那个设置为null。并不再使用。

```
</> Java |  
  
1 EventManager wp= EventManagerFactory.create(this,"wp");  
2 //this是Activity或其它Context类
```

详见ActivityMiniWakeUp类中”基于SDK唤醒词集成1.1 初始化EventManager”

1.2 自定义输出事件类

DEMO中，初始化过程合并到下一步。注意SDK和DEMO调用方式2选1即可。

注册用户自己实现的输出事件类

```
</> Java |  
  
1 EventListener yourListener = new EventListener() {  
2 @Override  
3 public void onEvent(String name, String params, byte [] data, int  
4 offset, int length) {  
5     Log.d(TAG, String.format("event: name=%s, params=%s", name, params));  
6     //唤醒事件  
7     if(name.equals("wp.data")){  
8         try {  
9             JSONObject json = new JSONObject(params);  
10            int errorCode = json.getInt("errorCode");  
11            if(errorCode == 0){  
12                //唤醒成功  
13            } else {  
14                //唤醒失败  
15            }  
16        }  
17    }  
18 }
```

```

16         } catch (JSONException e) {
17             e.printStackTrace();
18         }
19     } else if("wp.exit".equals(name)){
20         //唤醒已停止
21     }
22     }
23 };

```

详见ActivityMiniWakeUp类中“基于SDK唤醒词集成1.2 自定义输出事件类”

1.3 注册自己的输出事件类

</>

Java

```
1 wp.registerListener(yourListener);
```

详见ActivityMiniWakeUp类中“基于SDK唤醒词集成1.3 注册输出事件”

DEMO中，以上两步合并为

</>

Java

```

1 IWakeUpListener listener = new SimpleWakeUpListener();
2 myWakeUp = new MyWakeUp(this,listener);
3 // this是Activity或其它Context类

```

详见ActivityWakeUp类中“基于DEMO唤醒词集成第1.1, 1.2, 1.3步骤”

2 开始唤醒

2.1 设置唤醒输入参数

SDK中，您需要根据文档或者demo确定您的输入参数。DEMO中有UI界面简化选择和测试过程。demo中，在点击“开始”按钮后，您可以在界面或者日志中看见WAKEUP_START事件的json格式的参数。

wakeup.params(反馈请带上此行日志):{"kws-file":"assets:\\\\WakeUp.bin"}
// 其中{"kws-file":"assets:\\\\WakeUp.bin"}为WAKEUP_START事件的参数

</>

Java

```

1 HashMap map = new HashMap();
2 map.put(SpeechConstant.WP_WORDS_FILE, "assets://WakeUp.bin");

```

详见ActivityMiniWakeUp类中“基于SDK唤醒词集成第2.1 设置唤醒的输入参数”
唤醒词文件请去<http://ai.baidu.com/tech/speech/wake/#tech-demo>设置并下载

2.2 发送start开始事件

</>Java|

```
1 wp.send(SpeechConstant.WAKEUP_START, json, null, 0, 0);
```

详见ActivityMiniWakeUp类中”基于SDK唤醒词集成第2.2 发送开始事件开始唤醒”

DEMO中， 您需要传递Map 的参数， 会将Map自动序列化为json

</>Java|

```
1 HashMap<String,Object> params = new
2 HashMap<String,Object>();
3 params.put(SpeechConstant.WP_WORDS_FILE, "assets://WakeUp.bin");
4 myWakeup.start(params);
```

详见ActivityWakeUp类中”基于DEMO唤醒词集成第2.1, 2.2 发送开始事件开始唤醒”

3 收到回调事件

3.1 开始回调事件

SDK中， 回调事件在您实现的EventListener中获取。 OnEvent中， name是输出事件名， params该事件的参数， (data,offset, length)三者一起组成额外数据。 如回调的音频数据， 从data[offset]开始至data[offset + length] 结束， 长度为length。

</>Java|

```
1 public void onEvent(String name, String params, byte [] data, int offset, int
length);
```

详见ActivityMiniWakeUp类中”基于SDK唤醒3.1 开始回调事件”

DEMO中， 回调事件在您实现的IWakeupListener中获取。

详见WakeupEventAdapter类中”基于DEMO唤醒3.1 开始回调事件”

4 控制唤醒

SDK中，

4.1 控制停止唤醒

</>Java|

```
1 wp.send(SpeechConstant.WAKEUP_STOP, null, null, 0, 0);
```

详见ActivityMiniWakeUp类中”基于SDK唤醒词集成第4.1 发送停止事件”

DEMO中，

</>Java|

```
1 myWakeup.stop();
```

详见ActivityWakeUp类中”基于DEMO唤醒词集成第4.1 发送停止事件”

5 事件管理器退出

SDK中无需调用任何逻辑，但需要创建一个新的唤醒词事件管理器的话，之前的事件管理器请设置为null，并不再使用。

DEMO中，

</>Java

```
1 myWakeup.release();
```

详见ActivityWakeUp类中”基于DEMO唤醒词集成第5 退出事件管理器”>

9.5唤醒错误码

1	错误领域	描述	错误码	错误描述
2	10	录音设备出错		
3			1	录音设备异常
4			2	无录音权限
5			3	录音设备不可用
6			4	录音中断
7	11	唤醒相关错误		
8		没有授权文件	11002	
9		授权文件不可用	11003	
10		唤醒异常, 通常是唤醒词异常	11004	
11		模型文件不可用	11005	
12		引擎初始化失败	11006	
13		内存分配失败	11007	
14		引擎重置失败	11008	
15		引擎释放失败	11009	
16		引擎不支持该架构	11010	
17	38	引擎出错		
18			1	唤醒引擎异常
19			2	无授权文件

20			3	授权文件异常
21			4	唤醒异常
22			5	模型文件异常
23			6	引擎初始化失败
24			7	内存分配失败
25			8	引擎重置失败
26			9	引擎释放失败
27			10	引擎不支持该架构
28			11	无识别数据

10. 代码混淆

</>Plain Text |

```
1 -keep class com.baidu.speech.**{*;}

```

11. 权限

1	名称	说明	必
2	必要的权限		
3	android.permission.INTERNET	允许访问网络	是
4	android.permission.ACCESS_NETWORK_STATE	获取网络状态权限	是
5	android.permission.RECORD_AUDIO	允许程序录制声音通过手机或耳机的麦克	是
6	android.permission.WRITE_EXTERNAL_STORAGE	外置卡读写权限	是
7	非必要权限		
8	android.permission.CHANGE_WIFI_STATE	允许程序改变Wi-Fi连接状态	否