**Android ASR SDK 集成进入helloworld 项目**

# 简介：

本文针对遇见集成Android识别SDK有困难的用户，以helloworld项目作为示例，给出的步骤。

本文将集成ActivityMiniRecog至一个新建的helloworld项目。

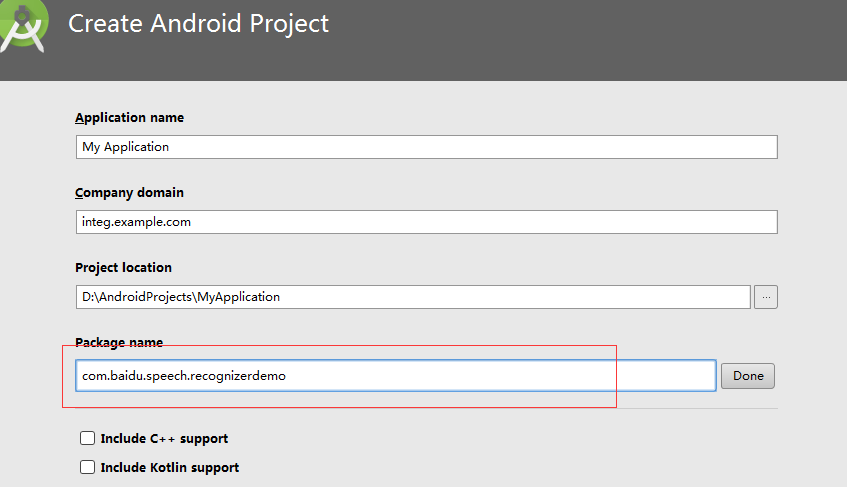
参考文档：<https://cloud.baidu.com/doc/SPEECH/ASR-Android-SDK.html#.E9.9B.86.E6.88.90.E6.8C.87.E5.8D.97>

前提： 将**官方demo**的源代码通过android studio在android真机上运行起来并且**测试成功**。Demo暂不支持模拟器。

目标用户：Android初学者或者想立即集成的用户。

|  |  |
| --- | --- |
| Android Studio | 3.0.1 |
| 官方Demo | 3.0.6 |
|  |  |
| 测试机型 | OPPO R7 - ColorOS 1.4  Android 4.3 |

# 准备一个helloworld项目及官网注册应用



注意，红框画出的是**包名。 在项目测试成功后，**您可以将您自己的包名填写在**官网申请的应用**里**。**并且该应用开通了**语音识别**能力**。**

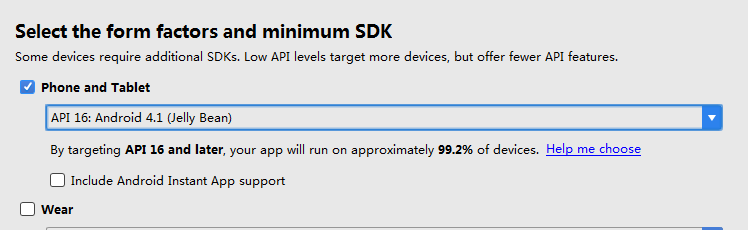
目前您可以“**暂时”**用demo的示例的应用信息。

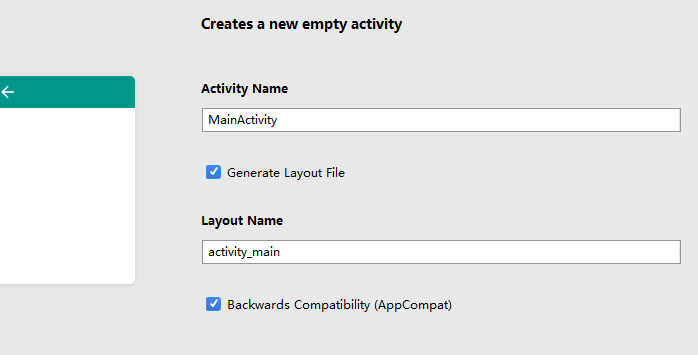
**APP\_ID** **10308735**

**API\_KEY** **tgPciIeIhrHVBu6DiNn7YTdD**

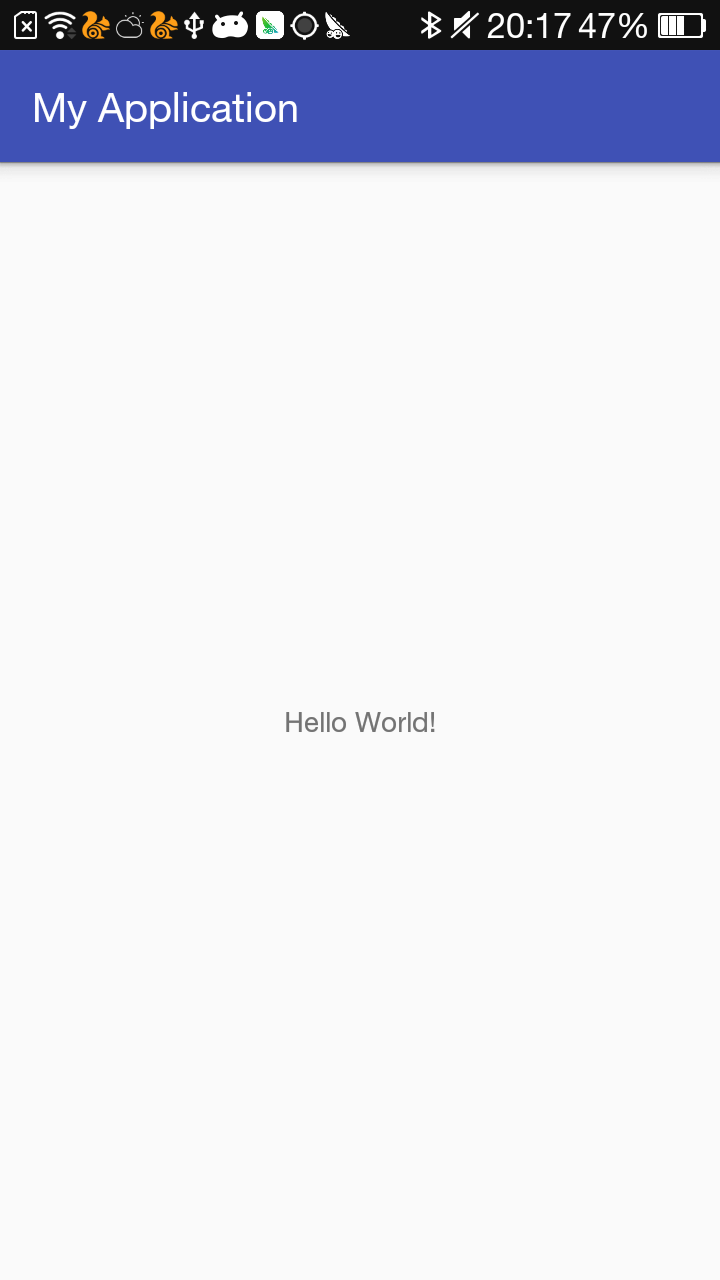
**SECRET\_KEY c743fcb3b3d30ddb58b9bd0b816d4e25**

包名Package name：**com.baidu.speech.recognizerdemo**





Android studio自动下载依赖项目后，运行，手机上出现以下界面：



# 复制 AndroidManifest.xml

从官方demo的AndroidManifest.xml中找到如下信息，然后复制到您自己的同名文件中。

此处需要您复制a)权限 b) 官网申请的应用信息 c) SDK的Service



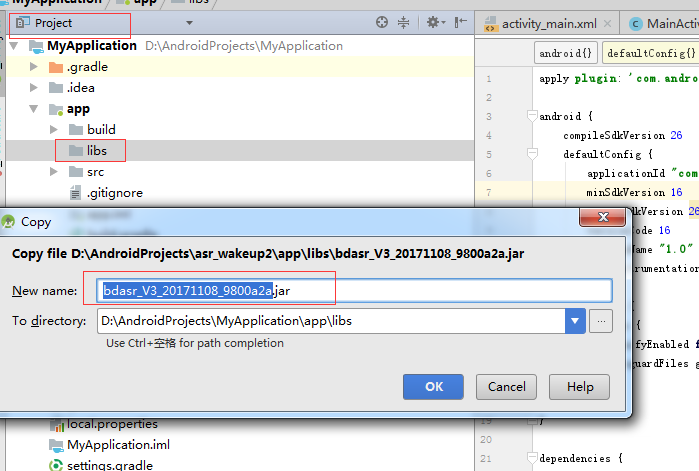
紫色的请**测试成功后**替换成您自己的申请的应用信息。

红色是您需要的添加的部分

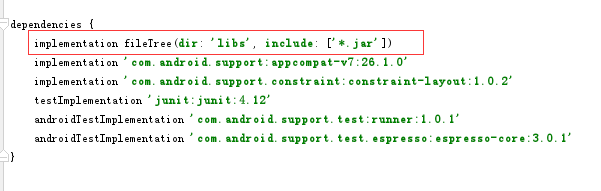


# 复制jar 文件

将官方demo中的app\libs\bdasr\_V3\_20171108\_9800a2a.jar 复制进您项目的同名目录。



此处是android 默认的jar libs目录，在build.gradle中可以确认：

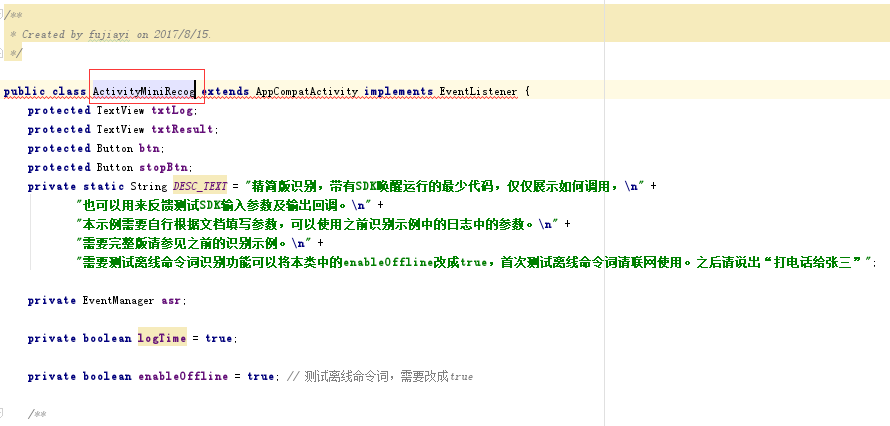


# 替换MainActivity

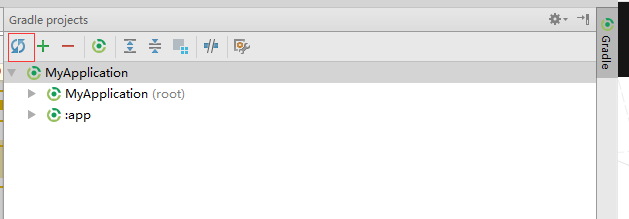
找到官方demo中的ActivityMiniRecog

（com.baidu.android.voicedemo.activity.mini 目录下）

除第一行外，将ActivityMiniRecog全部的内容复制到MainActivity中

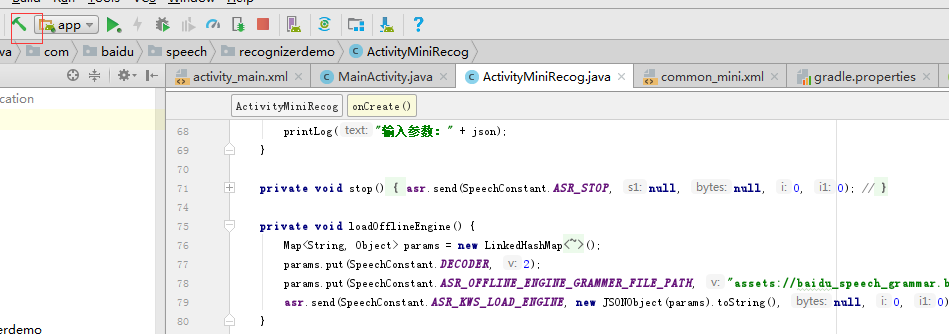


将红框中的ActivitityMiniRecog改为MainActivity



点击上图刷新按钮，可以发现缺少common\_mini layout文件

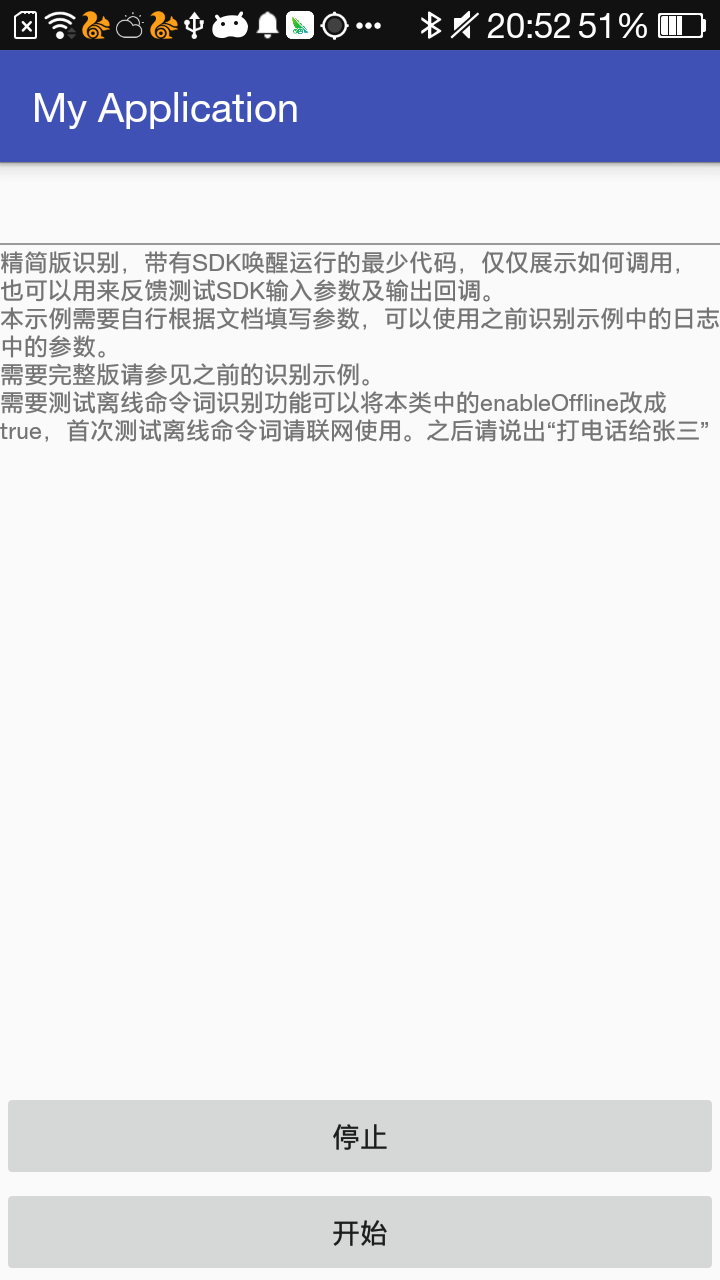
复制官方demo中的 res\layout\common\_mini.xml 文件，至项目的同名目录



测试编译项目，此时应该没有报错。

# 复制jni库的so文件

此时项目可以运行，可以看见如下界面：



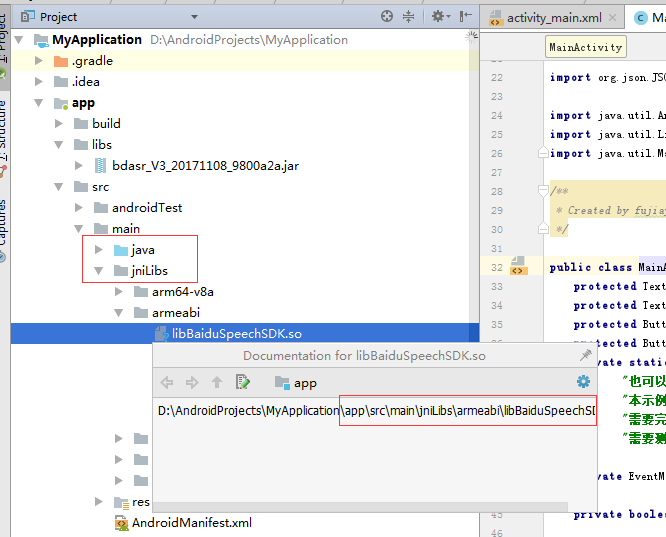
**由于缺少so文件**，点击开始后，有如下报错：

:{"sub\_error":5001,"error":5,"origin\_result":{"sub\_error":5001,"error":5,"sn":"","desc":"Can not load so library"},"desc":"Can not load so library"}

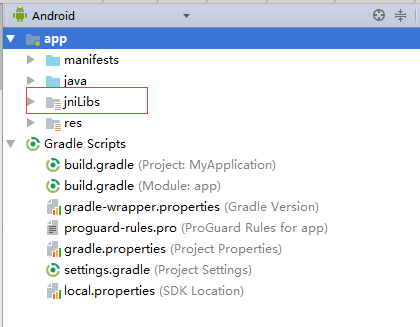
我们继续修改项目：

**复制官方demo中 app\src\main\jniLibs 至项目的同名目录**，如果对android studio不熟悉，这一步可以在windows 资源管理器操作

目录层级如下：



**可以切换视图确认下：**



# 联网测试

请确认手机可以联网，

运行项目后，点击“开始”按钮(注意只要点一次)，然后大声说“开始测试”。中途有权限申请，请同意。

注意，如果您不知道测试方法的，请先运行官方demo，确保官方demo先**测试成功。**



至此集成成功，之后我们需要根据需求定制参数。

# 选择纯在线或者支持离线命令词。

离线命令词是指SDK断网时可以识别bsg文件中预先固定的短语。

Bsg文件可以在<http://speech.baidu.com/asr>这里下载。

1. 纯在线：如果您不需要离线命令词功能，那么SDK仅在联网时有识别结果。

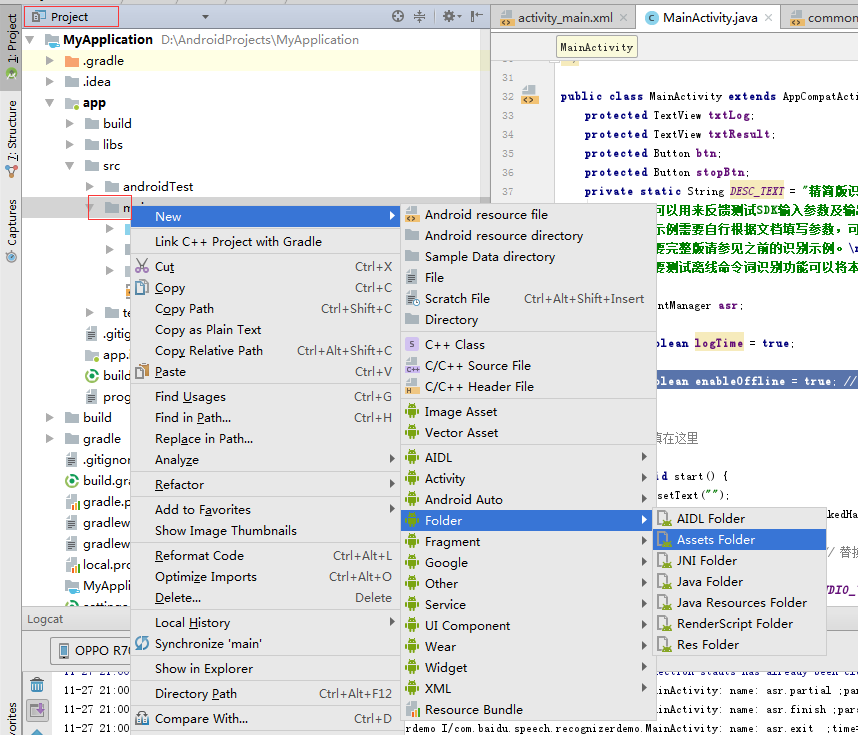
此时请修改为纯在线

**private boolean enableOffline** = **false**;

1. 离线命令词：如果您需要离线命令词功能，那么请下载您自定义的bsg文件。

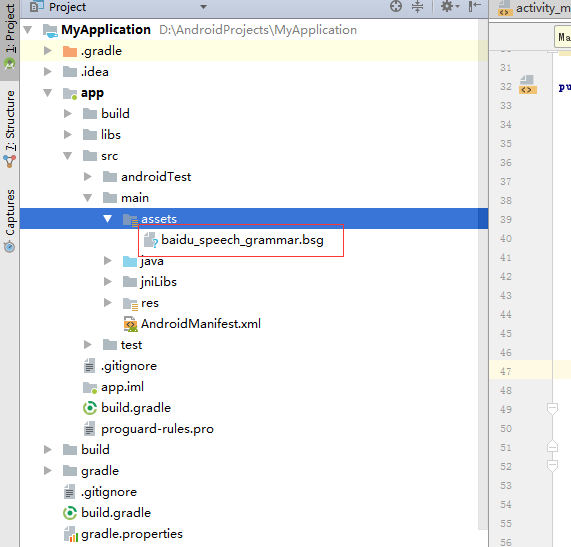
以官方demo提供的bsg文件为例：

新建assets目录：

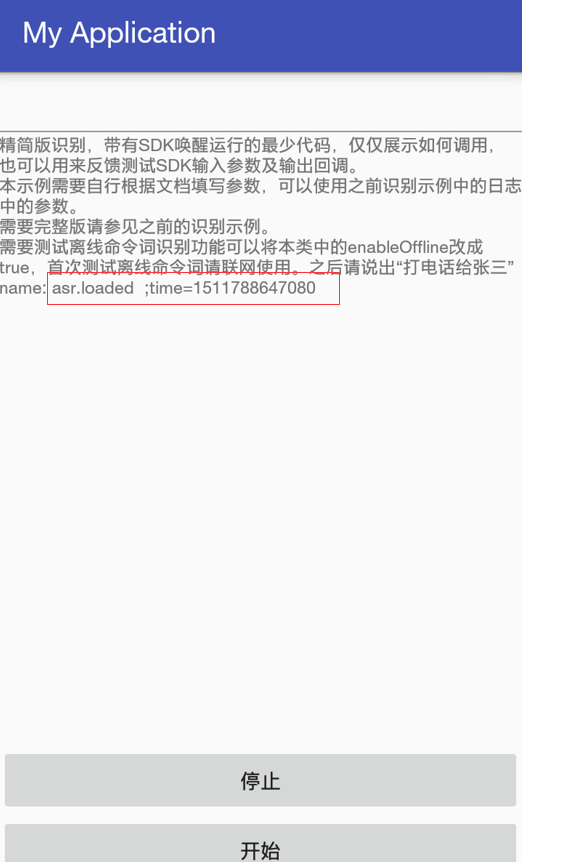


成功后如图

复制demo中的assets app\src\main\assets\baidu\_speech\_grammar.bsg至同名目录中

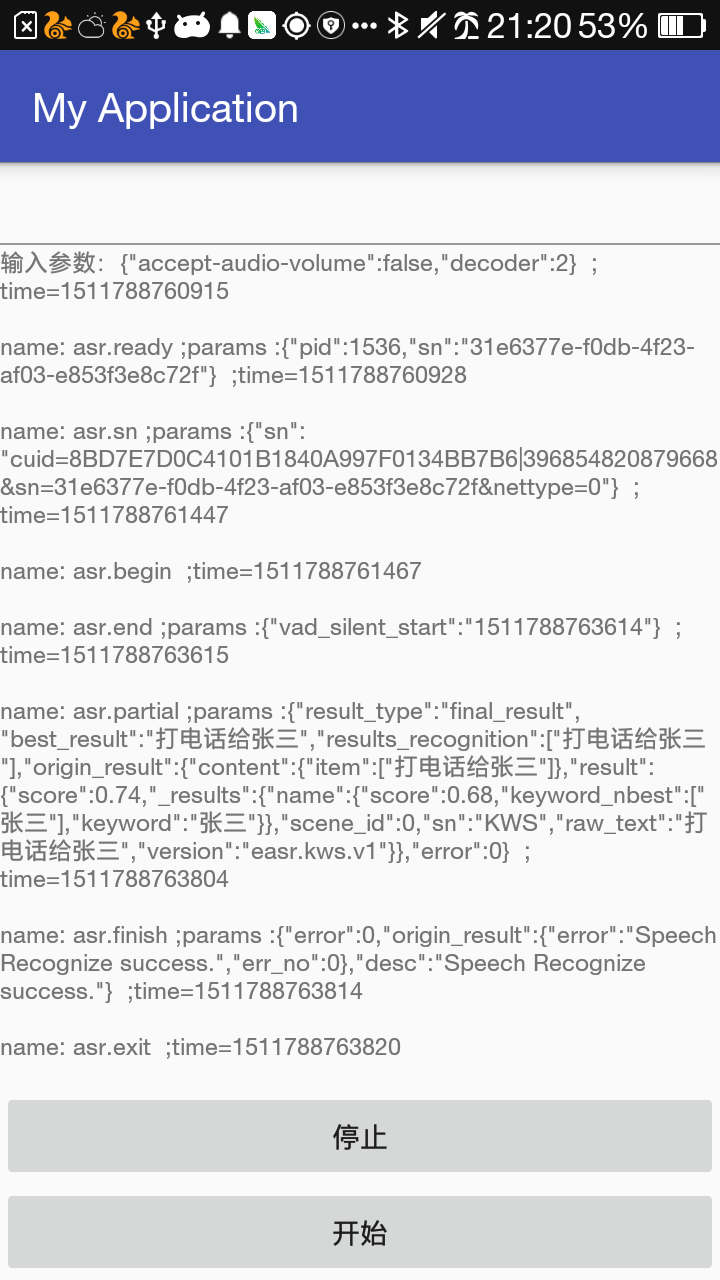


运行项目。



**出现红框加载的文字说明离线引擎加载成功。否则为失败**

离线命令词首次使用需要联网，请联网测试一次识别成功后断网，说“打电话给张三”。注意离线引擎识别率低于在线的，请发音清晰。



此时，官方demo集成全部完毕。

# 个性化识别参数

打开demo，选一个您要的识别场景应用，这里以“在线识别”界面为例。 进入界面后，点击“设置”按钮。

经过一轮测试，准备选择“保存音频文件”及VAD时长2000ms作为在线识别的参数。



点击开始录音后，可以看见如下输入参数：



这个json在logcat中也可以找到。

“vad.endpoint-timeout” 是定义在SpeechConstant中的常量。

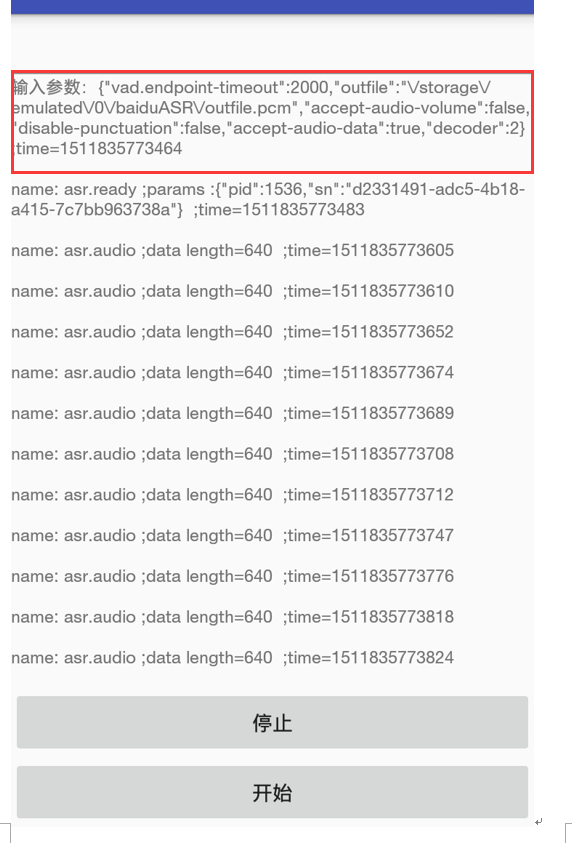
项目中填写同样的参数，生成一样的json。

Start方法代码如下：

**private void** start() {  
 **txtLog**.setText(**""**);  
 Map<String, Object> params = **new** LinkedHashMap<String, Object>();  
 String event = **null**;  
 event = SpeechConstant.***ASR\_START***; *// 替换成测试的event* **if** (**enableOffline**){ // 这里是false  
 params.put(SpeechConstant.***DECODER***, 2);  
 }  
 params.put(SpeechConstant.***VAD\_ENDPOINT\_TIMEOUT***,2000);  
 params.put(SpeechConstant.***ACCEPT\_AUDIO\_VOLUME***, **false**);  
 params.put(SpeechConstant.***OUT\_FILE***, **"/storage/emulated/0/baiduASR/outfile.pcm"**);  
 params.put(SpeechConstant.***ACCEPT\_AUDIO\_DATA***, **true**);  
 params.put(SpeechConstant.***DISABLE\_PUNCTUATION***,**false**);  
 *// params.put(SpeechConstant.NLU, "enable");  
 // params.put(SpeechConstant.VAD\_ENDPOINT\_TIMEOUT, 800);  
 // params.put(SpeechConstant.VAD, SpeechConstant.VAD\_DNN);  
 // params.put(SpeechConstant.PROP ,20000);* String json = **null**; *//可以替换成自己的json* json = **new** JSONObject(params).toString(); *// 这里可以替换成你需要测试的json*  
 printLog(**"输入参数："** + json); // 打印出来的json和官方demo的json一致

**asr**.send(event, json, **null**, 0, 0);  
}

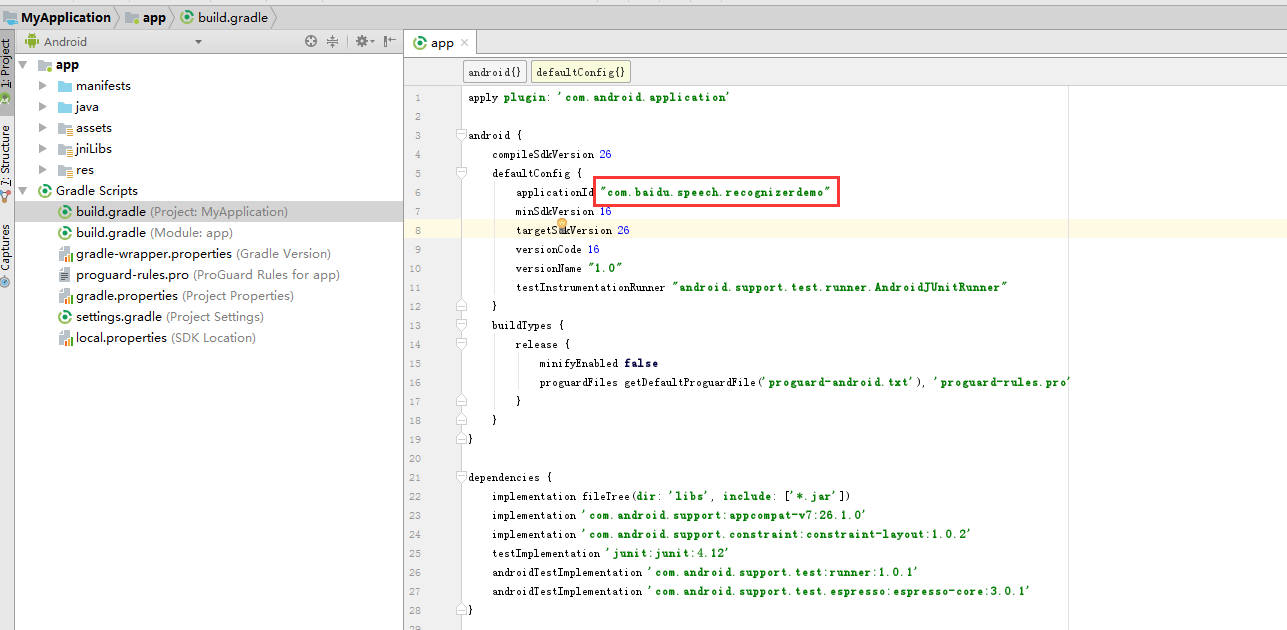
运行项目后，点击“开始”按钮可以看见：



# 替换自己申请应用的信息及包名

此时别忘了测试都是用demo的appId appKey SecretKey及包名信息的。

修改成您自己的包名：



官网申请新应用或者修改原有应用，填写您的包名。

将appId appKey secretKey填写至下图AndroidManifest.xml紫色框的地方。



# 代码下载

链接：https://pan.baidu.com/s/1dEN0T2L 密码：ohjk