**Android ASR SDK 集成合成DEMO中**

# 简介：

本文针对遇见集成Android识别与合成 有困难的用户，以android合成SDK作为示例，给出的步骤。

本文将集成ActivityAbstractRecog内demo封装的功能至官方android合成demo项目。

android合成demo，可以将文本合成为音频并且播放。

我们添加的功能为：识别成功后，将识别出来的文本，合成之后播放出来。

参考文档：<http://ai.baidu.com/docs#/ASR-Android-SDK/55389ffa>

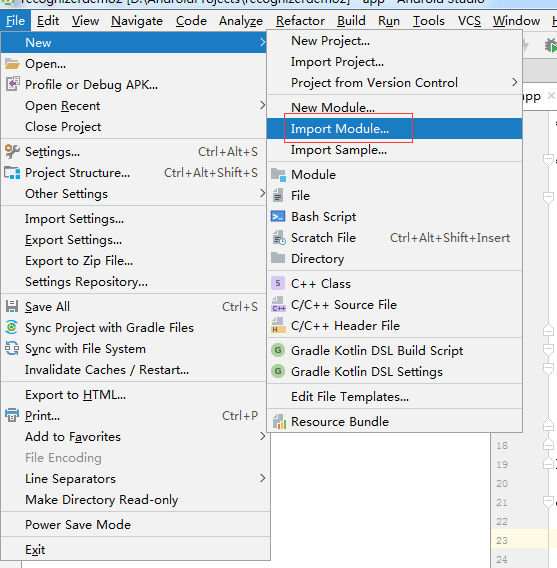
前提： 将**官方识别demo和合成demo两个项目**的源代码通过android studio在android真机上运行起来并且**测试成功**。Demo暂不支持模拟器。

|  |  |
| --- | --- |
| Android Studio | 3.5 |
| 官方识别Demo | 3.1.1 |
| 官方合成 | 2.5.5 |
| 测试机型 | Oneplus 5 |

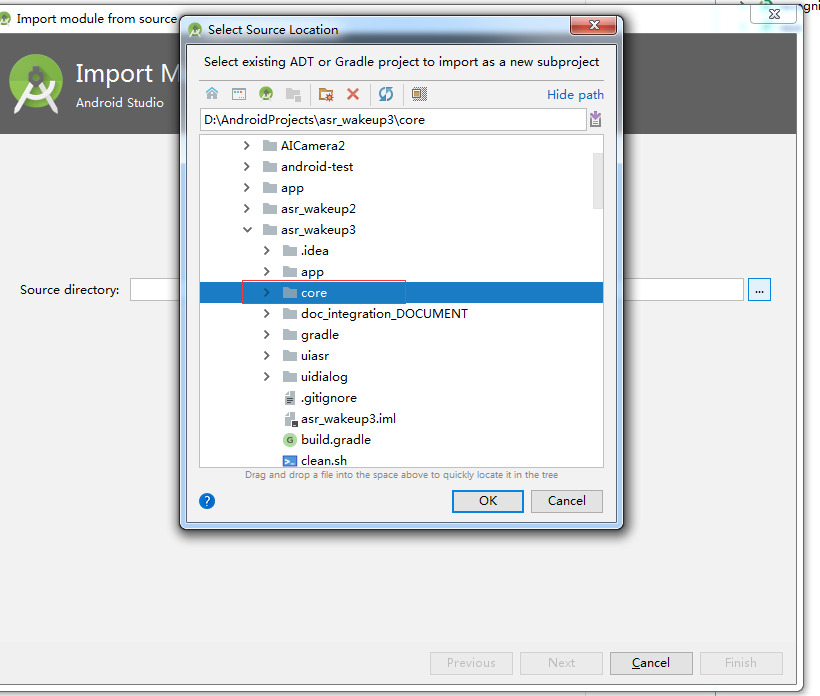
运行离在线合成demo：

# 导入core 模块

导入asr demo的core module



选择demo的core目录



此时查看项目目录下的settings.gradle, 可以看见core module已经被加载

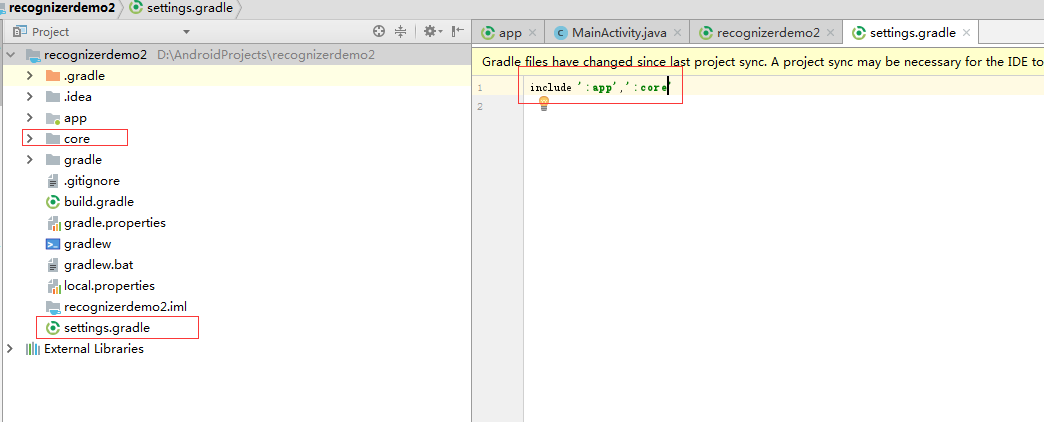
===================================================================

（使用新版本gradle无此问题，可忽略下面A-D）

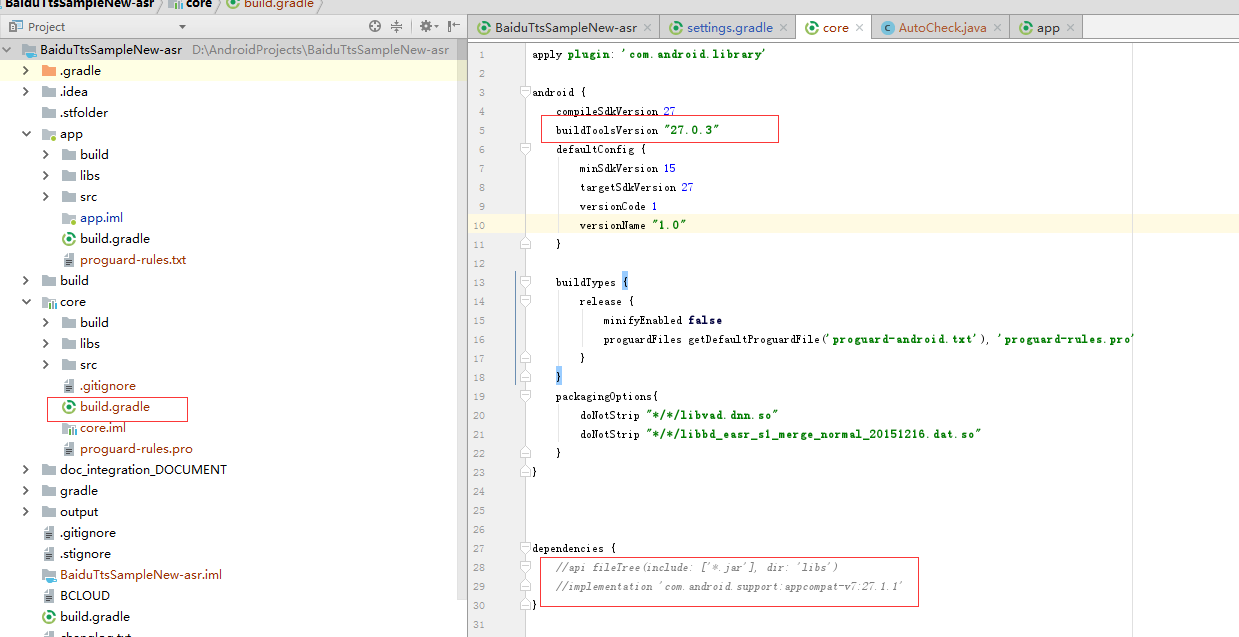
此时，我们会发现导入的core mudule和项目本身的gradle不一致。我们按照如下方式修改core module。

1. 如果settings.gradle里没有“:core”,请按截图自行添加。

include **':app'**,**':core'**

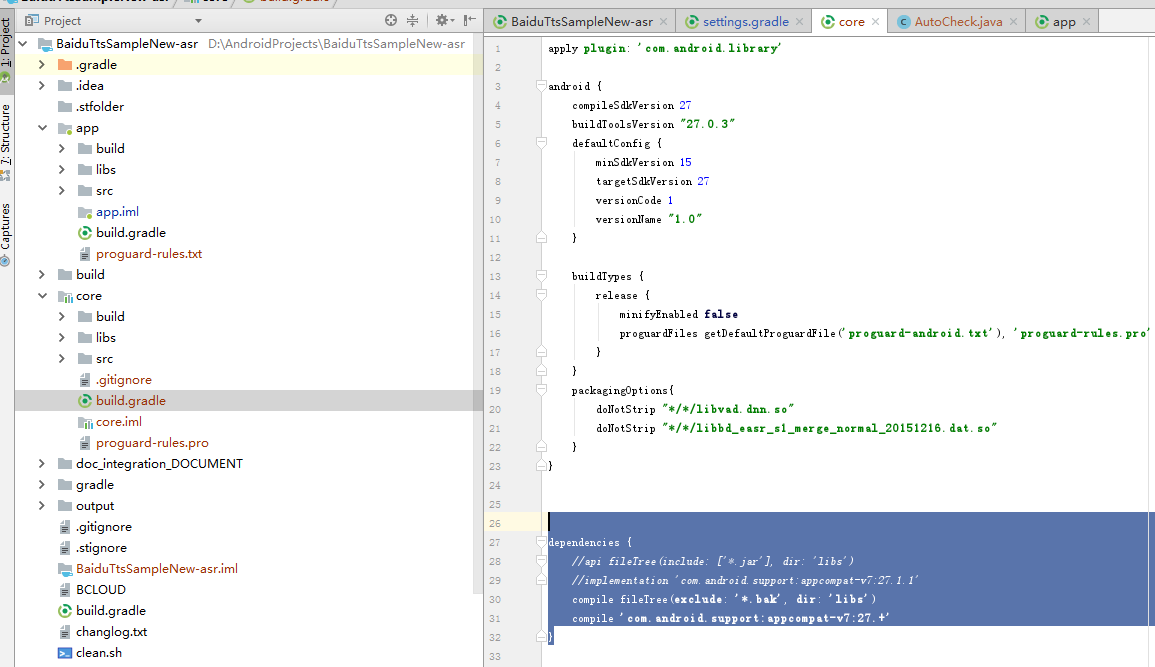


1. 修改core/build.gradle, 按照下图修改：

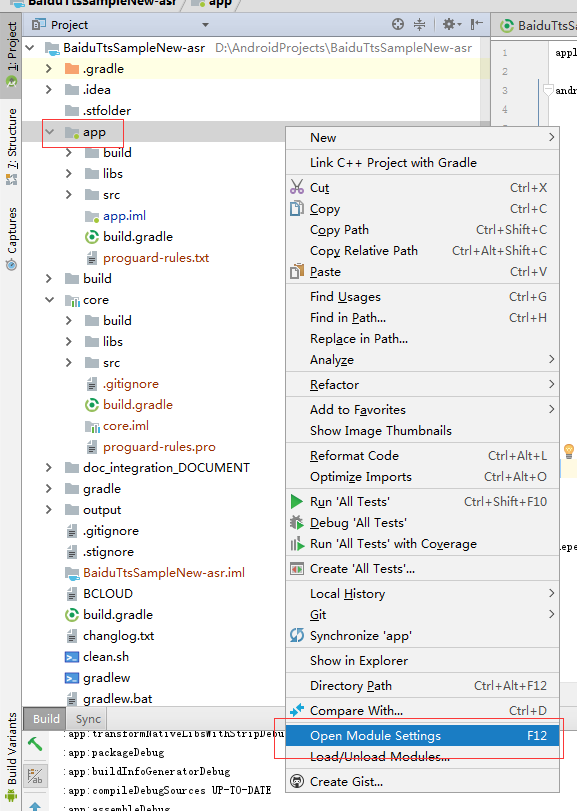


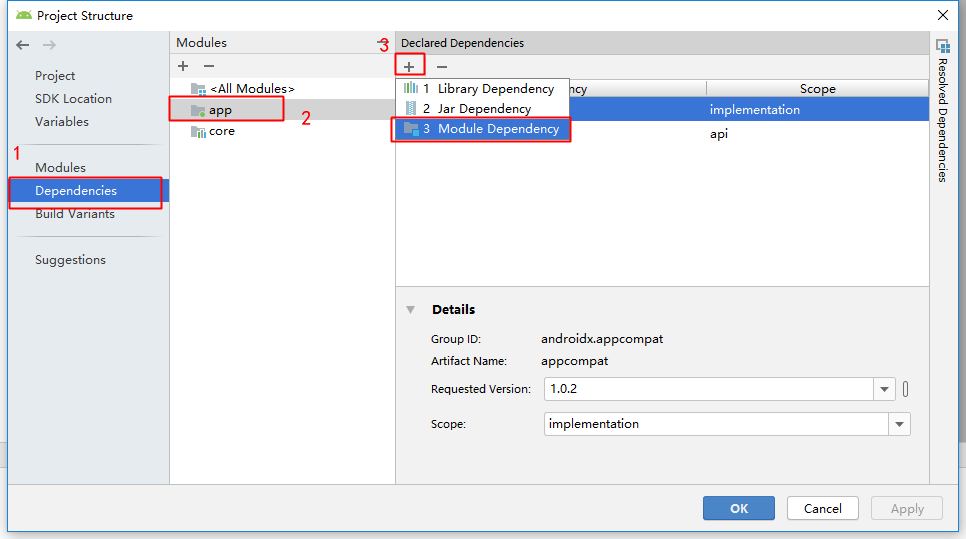
1. 刷新后可以看见core目录的图标，变为上面的模块图标
2. 此时编译会报类不存在，从app/buid.gradle 中复制如下2行到core/build.gradle中

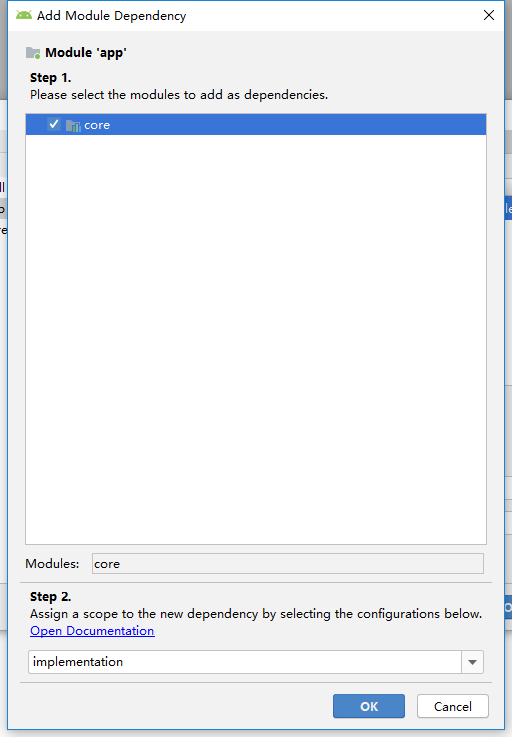
api fileTree(**exclude**: **'\*.bak'**, **dir**: **'libs'**)  
implementation **'androidx.appcompat:appcompat:1.0.2'**



配置app依赖core

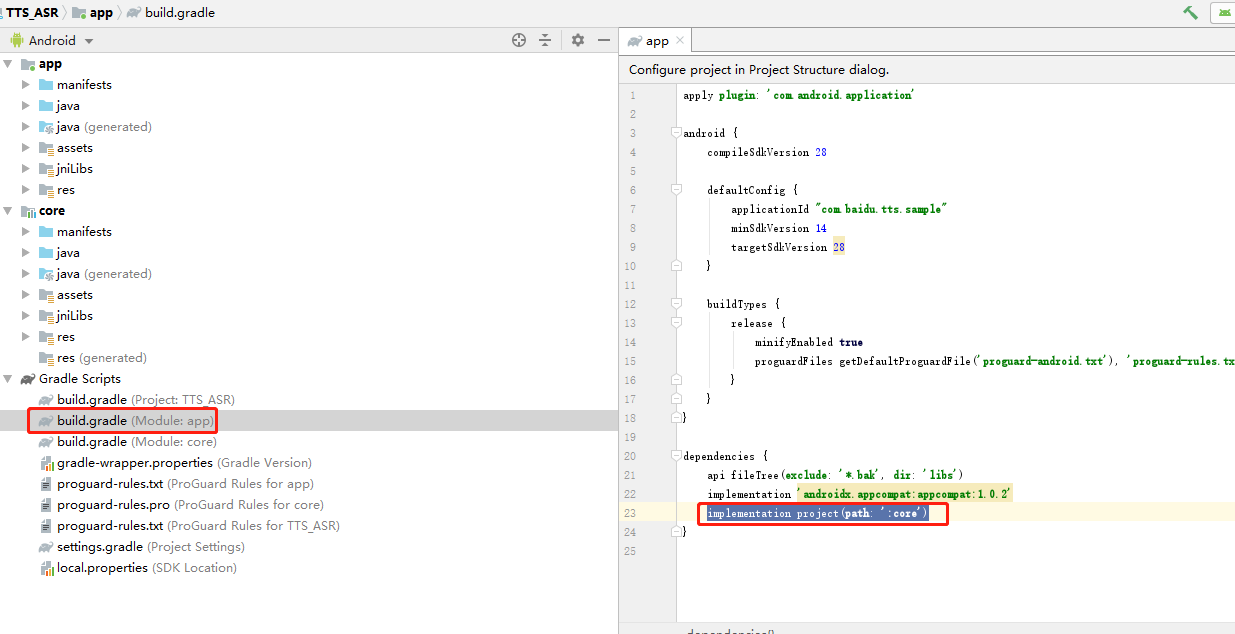






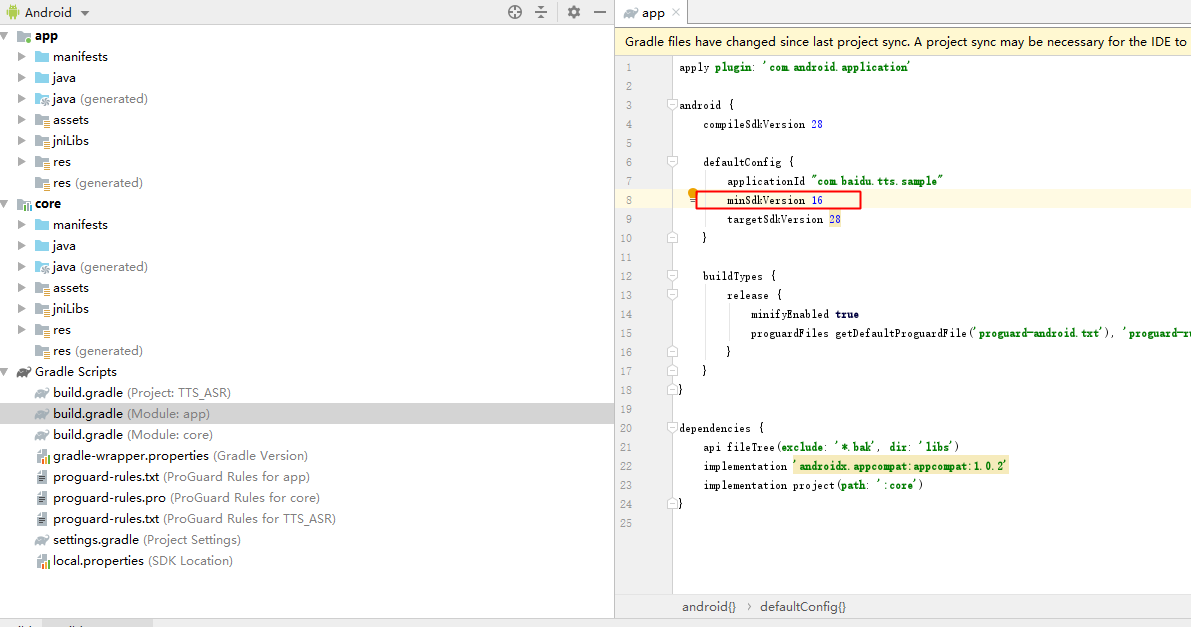
此时观察build.gradle文件，发现自动添加了如下一行

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



此时编译，会发现minSdkVersion不对，

修改minSdkVersion 16



此时再次编译运行，项目可以跑起来

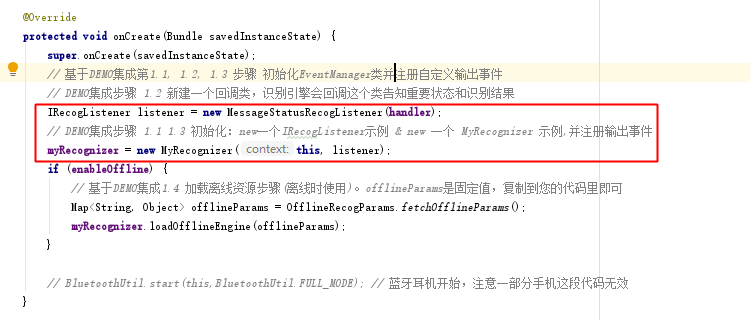
# 集成功能逻辑

再回顾下，我们添加的功能为：识别成功后，将识别出来的文本，合成之后播放出来。

识别触发的按钮，我们借用界面上的“使用说明”按钮。

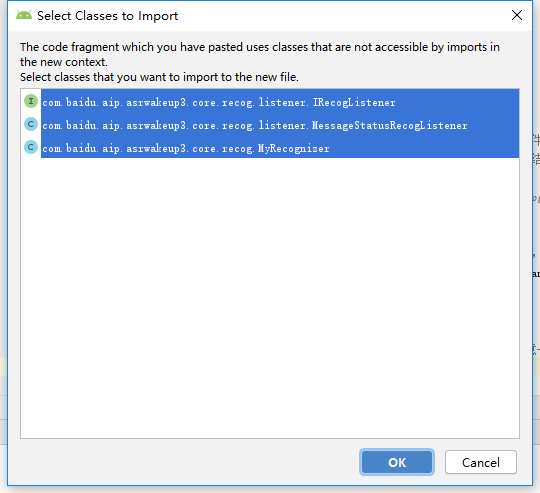
打开识别demo项目中ActivityAbstractRecog.java，我们复制一些有用的代码。

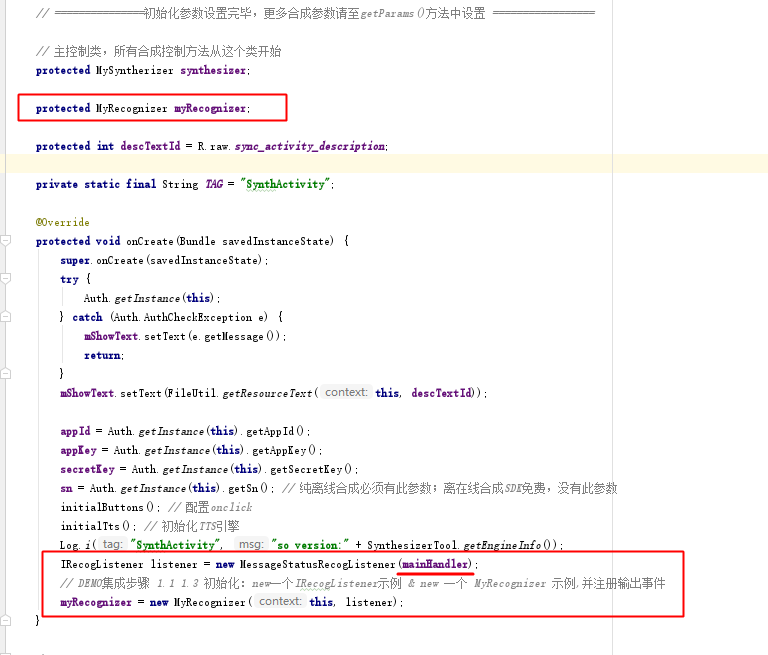
## A．初始化识别引擎



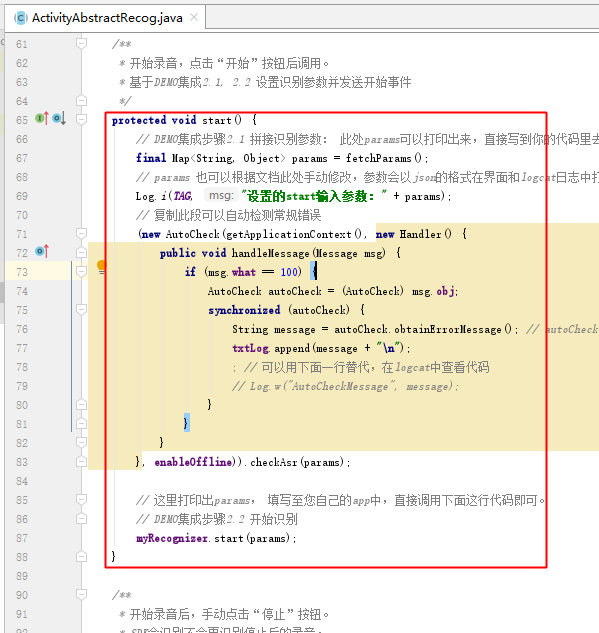
将这段复制到合成的项目中的SynthActivity.java 的onCreate 中

然后将Handler使用合成demo中mainHandle的，修改后如下





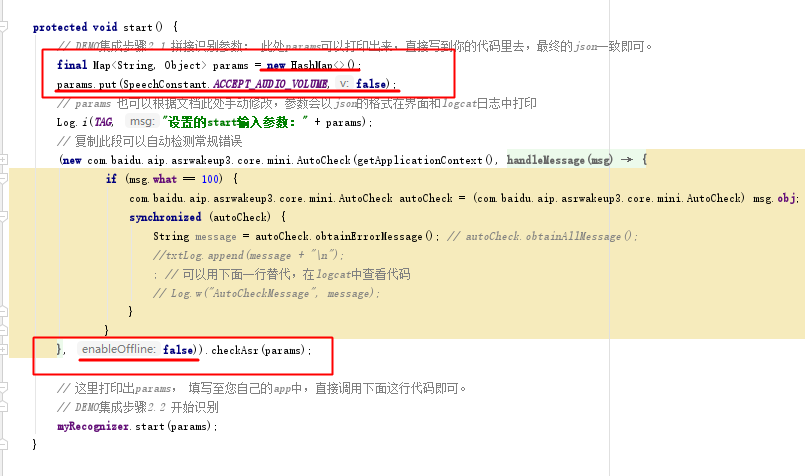
## B. 点击“使用说明”按钮开始识别



复制识别demo中的start方法，其中**final** Map<String, Object> params

需要改成您自己手写。测试识别demo，params是被打印出来的，我们手写代码添加上去。

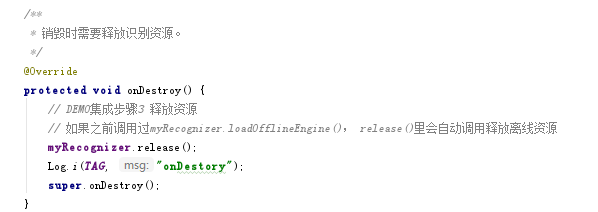
复制及简单修改后的代码如下：



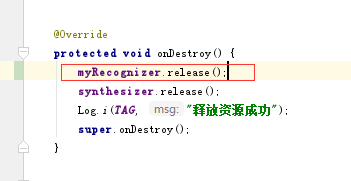
替换下“使用说明”按钮的点击



## C. 退出时销毁资源

识别demo中，

复制到我们的代码里



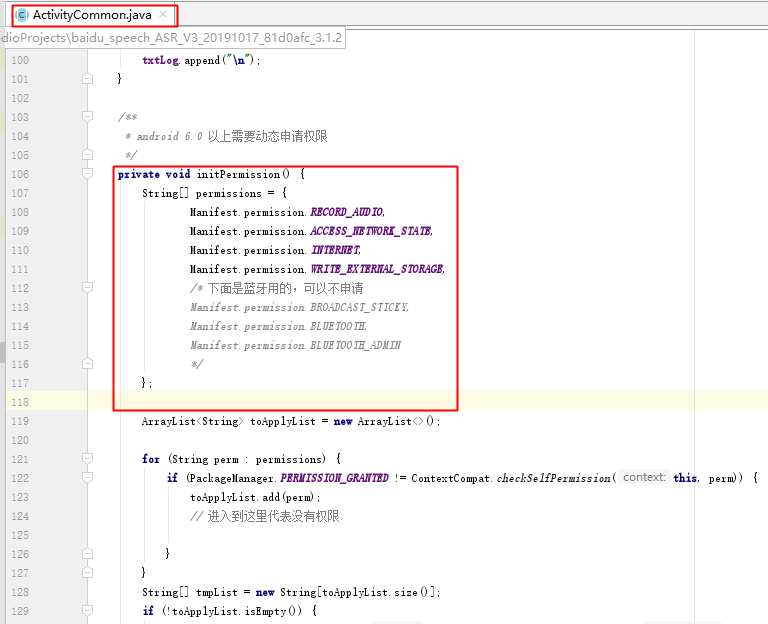
此时识别集成完毕，可以启动demo，点击“使用说明”按钮测试下识别。///

/

启动测试识别时，我们会发现缺少RECORD\_AUDIO权限。原因是合成的demo没有动态申请该权限。



识别demo中，我们发现ActivityCommon.java 有如下申请权限的代码：



在合成demo的BaseActivity.java中也有类似代码，合并权限申请的数组后：



再次启动，点击“使用说明”按钮后，说“开始识别”，logcat中（不是界面中），会出现如下日志：

2019-12-05 13:13:49.010 31510-31510/com.baidu.tts.sample I/RecogEventAdapterwf: name:asr.partial; params:{"results\_recognition":["开始识别"],"result\_type":"final\_result","best\_result":"开始识别","origin\_result":{"corpus\_no":6766819018412961033,"err\_no":0,"result":{"word":["开始识别"]},"sn":"b92c5dcc-cc93-4679-b468-8fce4b51fdeb","voice\_energy":32784.91796875},"error":0}

# 完成识别后播放功能

先让logcat中的识别结果在界面上显示下，查看合成demo的handler的处理，我们发现只有msg.what =PRINT的才会打印， PRINT值 = 0。

改下识别listener发Message的逻辑。



再次运行，我们发现界面上有识别的信息了。

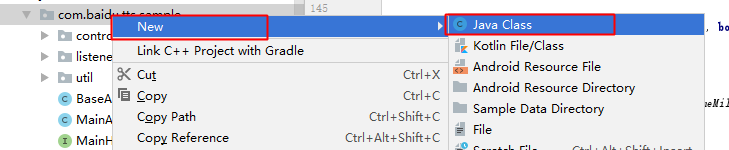
了解合成的demo，我们发现合成播放是调用了

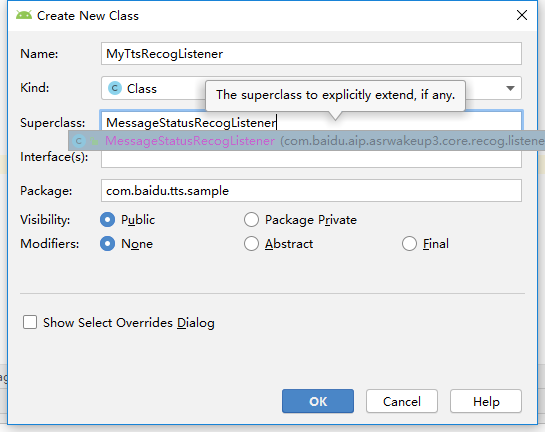
**synthesizer**.speak(text);

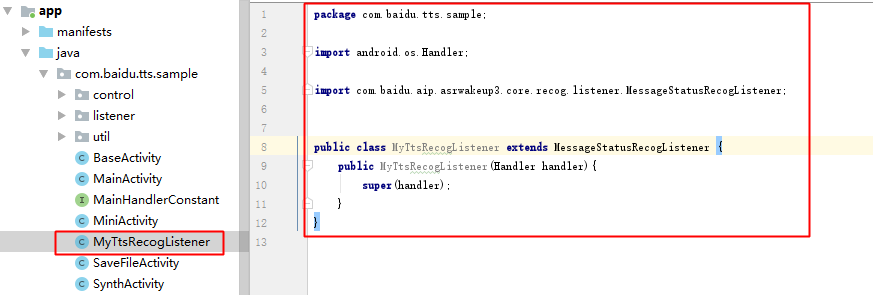
因此需要在识别结束的回调中调用这个播放的方法。

为此，我们需要一个自己的IRecogListener

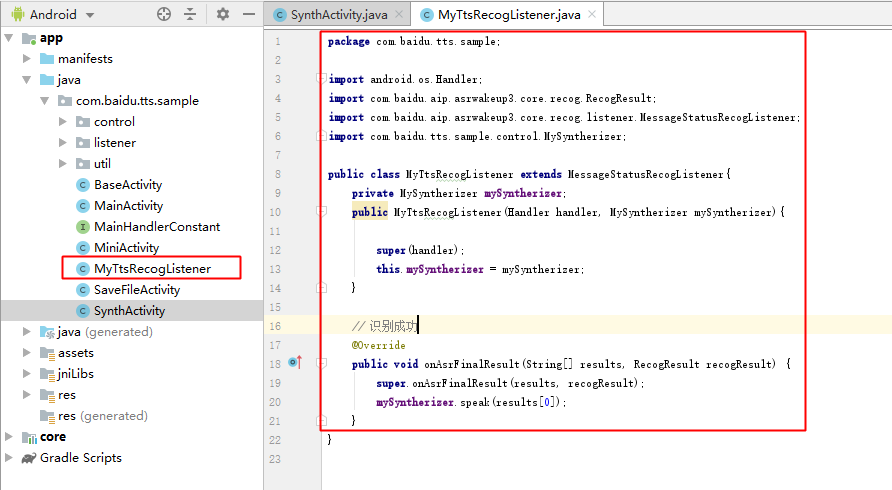
新建一个MyTtsRecogListener类



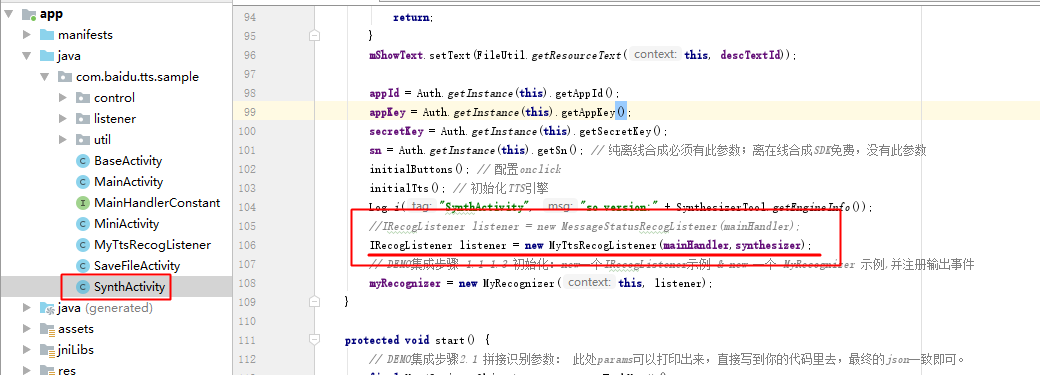




初始化的时候传入 synthesizer， 这样可以在识别成功的时候调用synthesizer.speak(text);



回到识别初始化的地方，替换掉原来的Listener

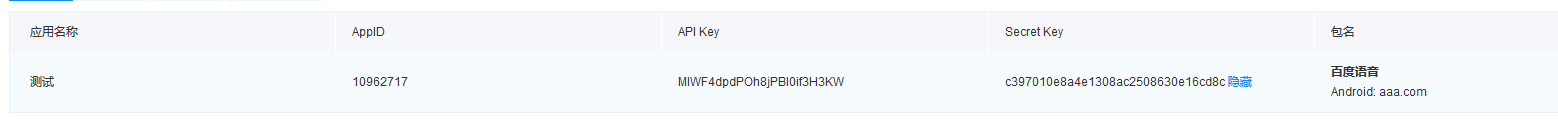


再次启动测试。至此集成及开发结束。

# 修改demo的应用信息

到这里，集成和测试完毕。之后上线时需要把demo中的应用信息替换成您自己。

您在网页中需要找到如下信息



如您的Android包名自己填写的是aaa.com

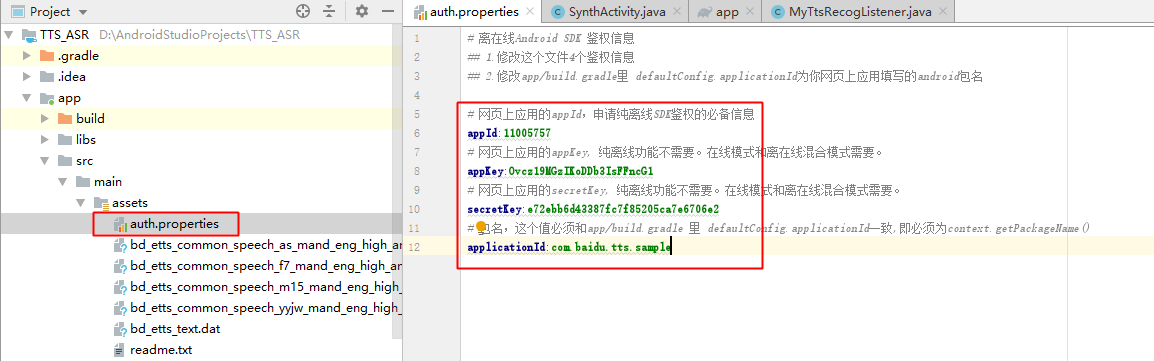
识别鉴权信息在core\src\main\AndroidManifest.xml

替换您的appId appKey secretKey

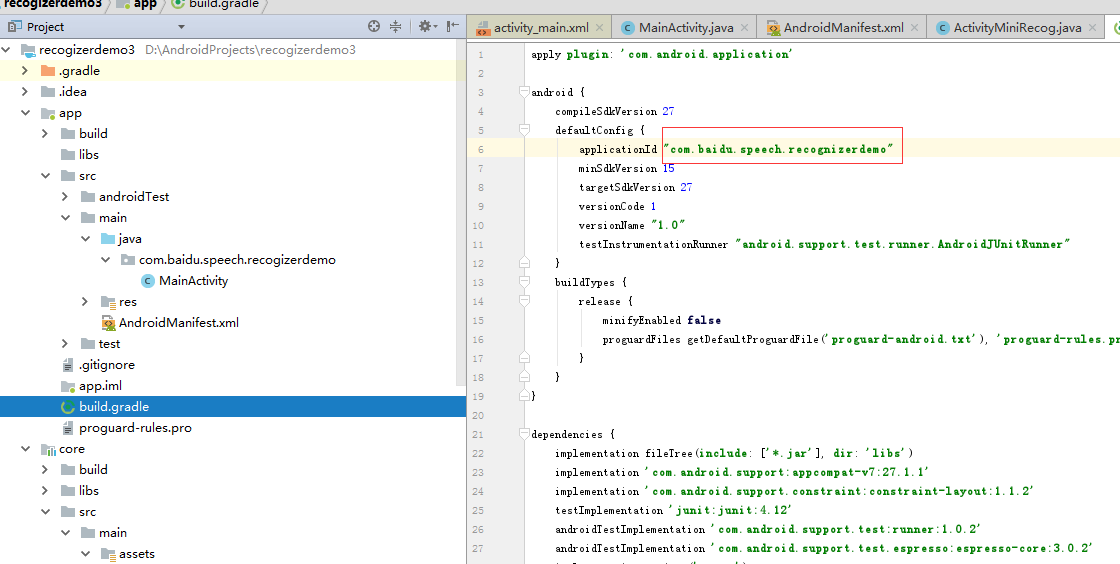


合成鉴权在您的app\src\main\assets\auth.properties文件里

替换您的appId appKey secretKey 和applicationId（即包名）：



在app\build.gradle，也需要替换成您上面填写的包名



# 完整项目下载

链接：https://pan.baidu.com/s/1xgLJTEZj0r8qiMkr0gNOfA

提取码：xn6i