《VSS Android SDK 使用说明》

# 1.VSS Android SD\_V1.0简介

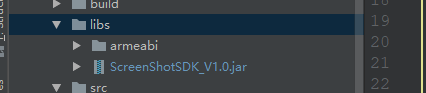
VSS Android SDK是一套基于Android 4.0及以上版本设备的应用程序接口，它为用户提供了一套完善的开发框架，以jar包与so库的形式提供。封装了远程协助服务以及许多OpenAPI，移动应用只需引入jar包和so库并修改少量代码，不需要理解信息传输流程，即可快速实现远程协助等功能，并可调用平台提供的OpenAPI。

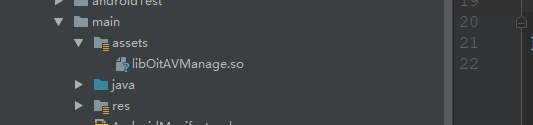
主要提供的功能有：远程协助服务，可视化帮助服务，自动化备份服务，信息定位服务，双向语音服务，语音向导服务。具体接入方法可参考以下内容。

# 2.创建工程及引用SDK源码文件

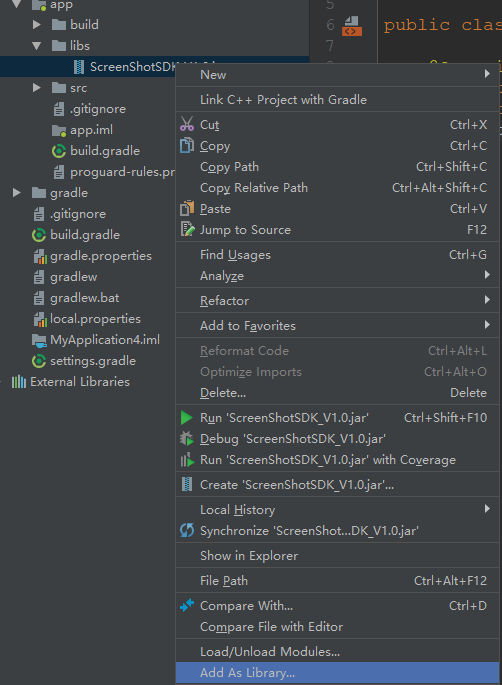
以下为AndroidStduio为开发IDE进行范例说明：

1. 创建一个工程，并把ScreenShotSDK\_V1.0.jar放入libs目录，libOitAVManage.so放入assets目录下，如下图所示：**(注：libs文件夹中需有不为空的armeabi或其他32位架构的arm包，不能有64位的arm包，so暂不支持64位cpu架构。)**





(2)将ScreenShotSDK\_V1.0.jar加入编译路径中。 具体的操作步骤为：选中ScreenShotSDK\_V1.0.jar，右键菜单中选择Add as library，如下图所示：



# 3.ScreenShotSDK\_V1.0初始化配置

使用ScreenShotSDK\_V1.0时，主要有以下四个步骤：

Step1: 给应用的AndroidManifest增加SDK需要的配置；

Step2: 应用MainActivity里加载so文件。

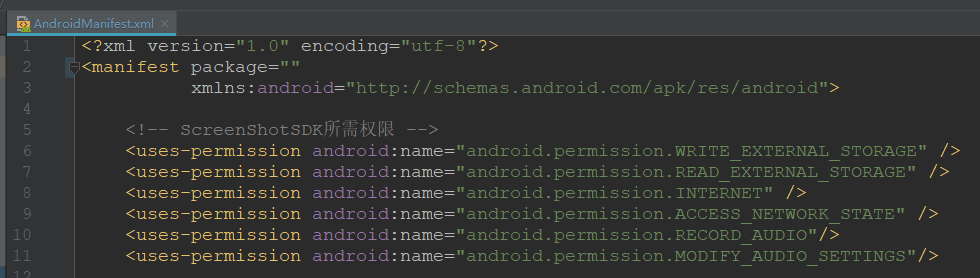
Step3: SDK初始化；

Step4: 实现SDK接口回调；

Step5: 调用OpenAPI。

## 3.1配置AndroidManifest：

在应用的AndroidManifest.xml增加以下权限配置



<!-- ScreenShotSDK所需权限 -->

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

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

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

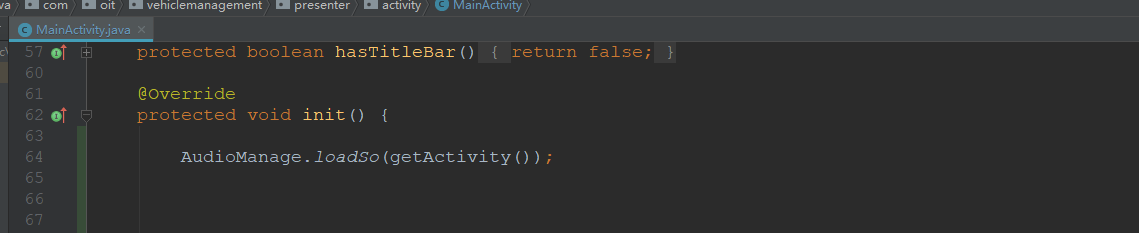
<uses-permission android:name="android.permission.RECORD\_AUDIO"/>

<uses-permission android:name="android.permission.MODIFY\_AUDIO\_SETTINGS"/>

**（注：当api>6.0的时候，麦克风权限需动态授权，否则远程辅助功能无法开启。）**

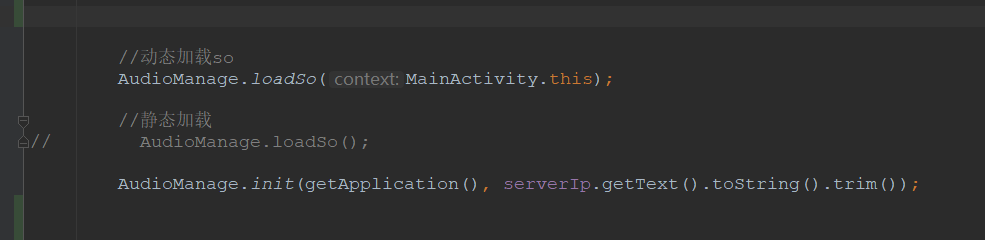
## 3.2配置加载so库：

在MainActivity的onCreate()方法中加载so库。**(注：不可在application中加载so。)**



## 3.3SDK初始化

在MainActivity的onCreat()方法中加载完SO库之后，调用AudioManage.Init(Application application, String lvsip)方法初始化SDK，两项参数分别为application实例，负载均衡服务器IP。**（注：此方法需在onCreate方法中执行。）**

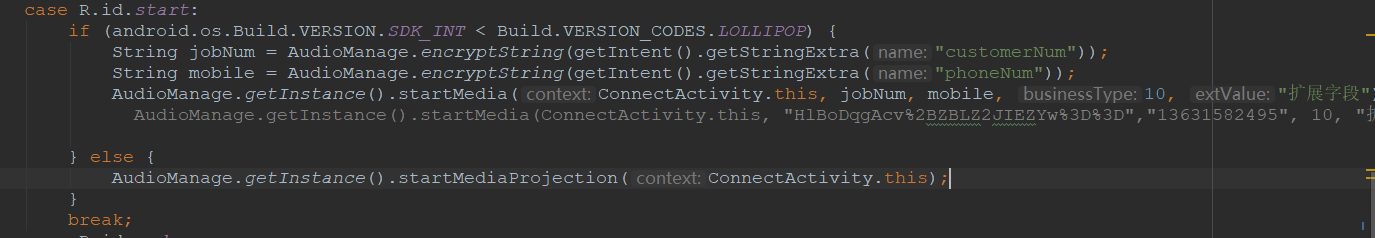


# 4.调用OpenAPI

SDK中AudioManage类是对外提供功能接口的主要实现类，所有接口都通过AudioManage类的实例来调用。

## 4.1开启视频辅助功能

AudioManage.getInstance().startMedia(Context context, String jobNum, String mobile, int businessType, String extValue)方法开启视频辅助，当api>5.0的时候使用AudioManage.getInstance().startMediaProjection(Context context)方法开启视频辅助，在onActivityForResult()回调中再执行上一方法。



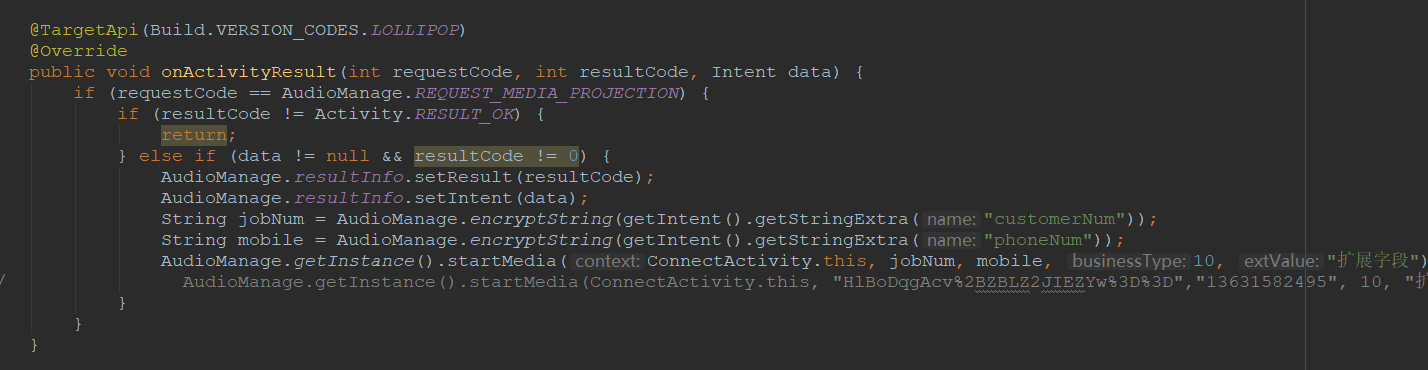
注：(1)当手机版本高于6.0的时候，需动态申请麦克风权限，不然双向语音无法开启。

(2) startMediaProjection()方法中涉及到手机录屏权限的用户授权，若申请权限失败（取消授权），重新调用startMediaProjection()将会重新提示权限的授权请求。

（3）执行startMedia()方法后，若未连接坐席端成功，需执行endMedia（）方法，方可再执行startMedia()，不然没有效果。

## 4.2实现回调

当api>6.0的时候，执行开启视频辅助功能方法后，会传入onActivityResult的回调，用以接收返回视频信息采集的调用结果，此回调写在调用使用onStartMedia()方法的activity页面。



## 4.3结束视频辅助

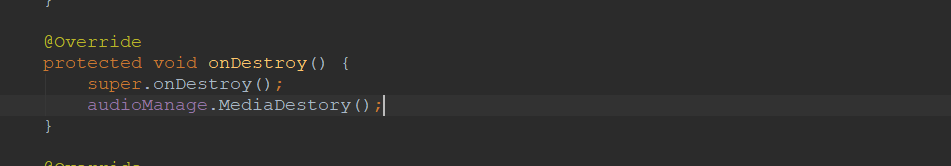
断开视频辅助连接



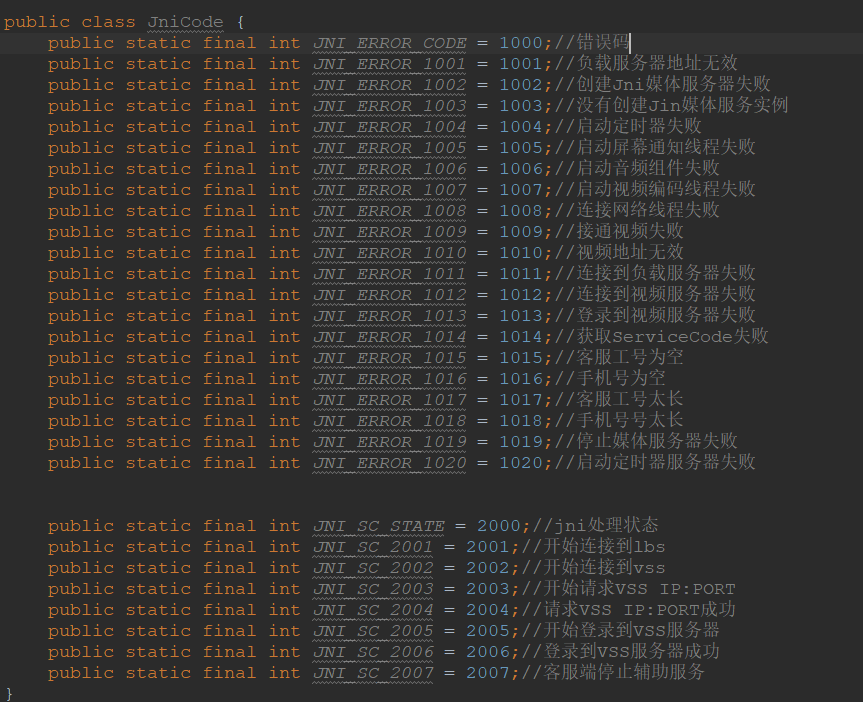
AudioManage.getInstance().endMedia();

## 4.4audioManage实例内容销毁

在初始化的MainActivity的onDestory()方法中销毁AudioManage资源内容。



## 4.5视频辅助过程中状态码以及错误码：



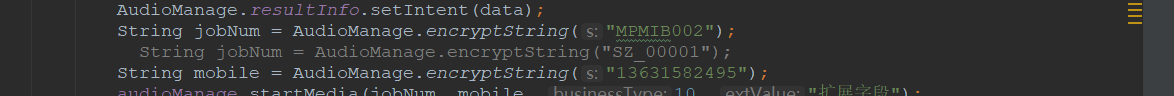
## 4.6.sdk提供客户号，手机号加密函数：

String jobNum = AudioManage.encryptString("MPMIB002");

函数中传递客户号或手机号明文，返回加密后的密文。

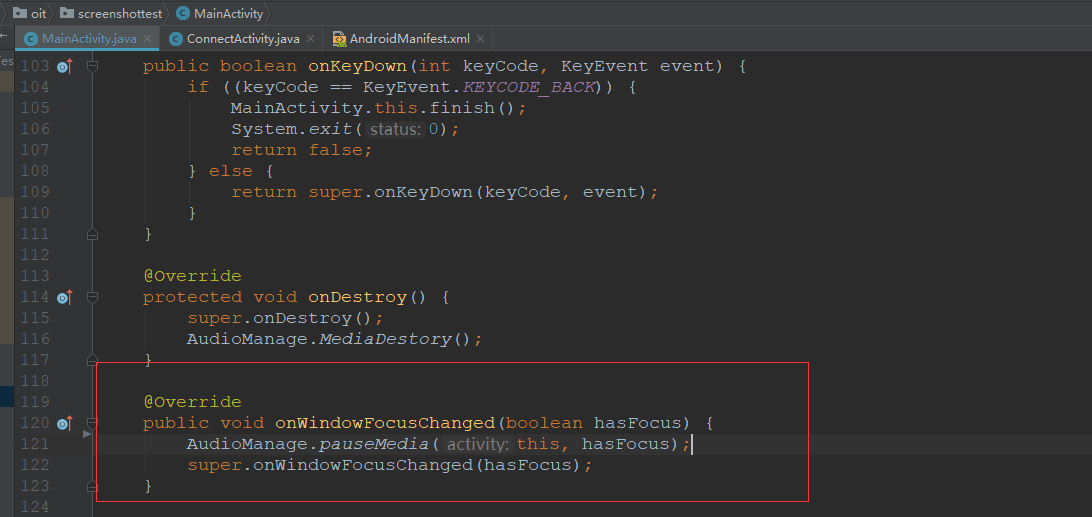
然后调用startMedia()函数中客户号，手机号参数为加密的密文。

**（注：若应业务流程需求，加密过程修改为后台接口加密则不使用sdk方法对客户号手机号加密。）**



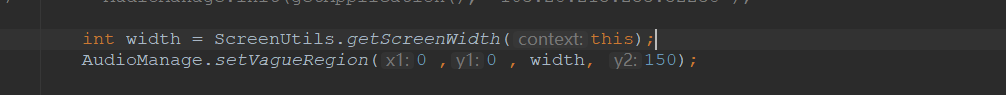
## 4.7使不录制通知栏内容：

若需不录制通知栏内容，需在有此需求的Acitvity中的onWindowFocusChanged(boolean hasFocus)方法中监听全屏焦点，实现AudioManage.pauseMedia(Context context, Boolean hasFocus);方法。**（注：若执行了此操作，会使得原生弹出框弹出时也会暂停录制视频）**



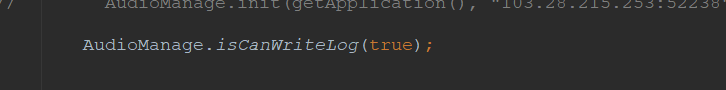
## 4.8使固定区域模糊化：

使用AudioManage.setVagueRegion(int x1, int y1, int x2, int y2)函数设定模糊化区域，x1, y1,参数为模糊区域左上角坐标，x2, y2,参数为模糊区域右下角坐标。此方法在可在任意时刻调用。四项参数为0时则表示模糊区域为0。



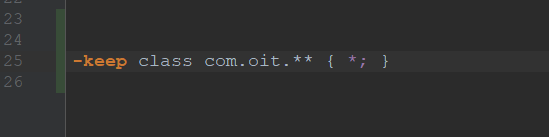
## 4.9新增日志输出文件方法

使用AudioManage.isCanWriteLog(true);方法设置可输入log日志到文件，路径为ScreenShot/Log。在使用初始化方法之后调用即可。默认为false，不使用则不会输出日志文件。



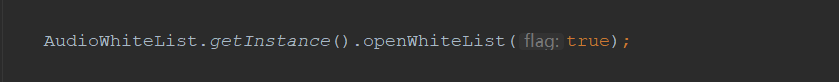
## 4.10混淆规则

剔除com.oit包下内容即可。

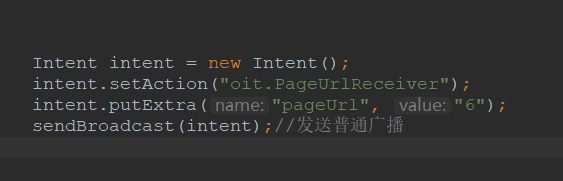


## 4.11新增白名单功能（2018-6-27）

使用AudioWhiteList.getInstance().openWhiteList(true);方法开启白名单功能，默认为false。



通过以下代码发送跳转页面链接



Intent intent = new Intent();intent.setAction("oit.PageUrlReceiver");intent.putExtra("pageUrl", "6");sendBroadcast(intent);//发送普通广播

**（后修改为发送本地广播）**

目前白名单内容为：

额度初审：'/xinyidai/pages/limitFinal.html'

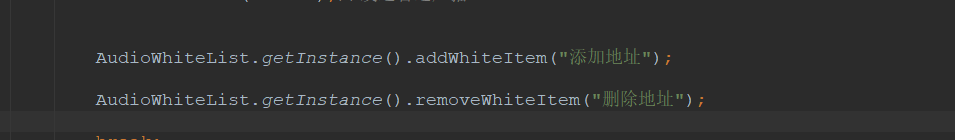
收还款信息页： '/xinyidai/pages/confirm.html'

工作信息页： '/xinyidai/pages/workInfo.html'

家庭信息：'/xinyidai/pages/familyInfo.html'

联系人信息：'/xinyidai/pages/linkInfo1.html'

若需添加白名单地址或删除白名单地址则调用以下两个函数即可；



## 4.12支持4.0系列手机（2018-9-10）

需在在清单文件的application标签中添加android:largeHeap="true" <application android:largeHeap="true" >标签