

# Android SDK使用指南

## 版本记录

V1.0	更新画布功能	2016-3-20
V1.1	更新白板功能，包括录制、截屏等功能	2016-6-28
V1.15	增加P2P功能，包括音频、笔迹、图片同步等	2016-7-23
V1.2	增加蓝牙设备交互，增加单画布离线笔记支持	2016-9-28

## 目录

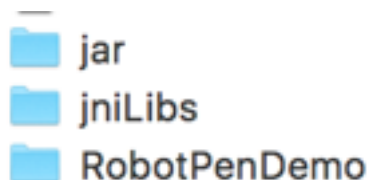
一、SDK文件夹相关文件说明	2
二、SDK使用步骤说明	2
三、相关类介绍	3

Android SDK 针对用户会使用到的笔、白板以及在使用笔和白板过程中会用到的必要功能进行了封装。

名词定义：

- 笔：配套硬件中的智能笔。
- 笔迹：APP中将多个笔的坐标点串联而成的记录。
- 白板：APP中可见的操作区域，包括一个或多个画布加操作区域。
- 画布：APP中可以显示笔迹的区域。
- 场景：配套硬件中可以供智能笔绘制的板子。

## 一、SDK文件夹相关文件说明



- jar文件夹内SDK需要的必要的jar文件。
  - **ROBOTPENSdk** 核心服务，用于与笔建立连接、通信。
  - **ROBOTPENFILE** 文件服务，用于处理图、文等文件类信息。
  - **ROBOTPENREMOTE** 笔的交互服务，用于处理笔迹等信息。
  - **ROBOTPENMODEL** SDK中使用到的对象。
  - **ROBOTPENUTIL** SDK中使用到的常用方法。
  - **ROBOTDB** SDK中离线笔记本地存储的数据库操作。
- jniLibs文件夹内是与设备交互的so文件，建议放到项目的application中。

```
static {  
    System.loadLibrary("avutil-54");  
    System.loadLibrary("swresample-1");  
    System.loadLibrary("swscale-3");  
    System.loadLibrary("postproc-53");  
    System.loadLibrary("avcodec-56");  
    System.loadLibrary("avformat-56");  
    System.loadLibrary("avfilter-5");  
    System.loadLibrary("RecordImageUtil");  
}
```
- RobotPenDemo是实用SDK，针对笔和白板的使用交互的简单示例。

## 二、SDK使用步骤说明

1. 将jar、so文件放置到项目对应的文件夹下；
2. 修改配置文件：

- 修改AndroidManifest.xml文件添加权限和服务。
  - 权限：

```
<!-- 访问网络权限 -->
<uses-permission android:name="android.permission.INTERNET" />
<!-- 获取音频权限 -->
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<!-- 写入存储权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<!-- 蓝牙权限 -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```
  - 服务：

```
<!-- USB设备服务 -->
<service android:name="cn.robotpen.core.services.UsbPenService"
android:enabled="true" />
<!-- 文件管理服务 -->
<service android:name="cn.robotpen.file.services.FileManageService"
android:enabled="true" />
<!-- 蓝牙设备服务 -->
<service android:name="cn.robotpen.core.services.SmartPenService"
android:enabled="true" />
```
- 修改的key值，包括通讯key和直播key。（只使用本地功能可以忽略此处配置）  
在application中针对QiniuConfig.ACCESS\_KEY  
QiniuConfig.SECRET\_KEY  
AliyunConfig.ACCESS\_KEY  
AliyunConfig.SECRET\_KEY值进行修改。

3. 创建画笔、白板完成集成。

**注：在创建白板的布局时，一定要将白板嵌套在一个RelativeLayout布局中，所以用于装载白板的父节点必须是RelativeLayout。**

### 三、相关类介绍

- 1. PenApplication类  
项目的Application继承PenApplication类。

方法	说明	注意事项
bindPenService(String svrName)	绑定笔服务	绑定服务之前要确定服务已启动
getPenService()	获取笔服务	

方法	说明	注意事项
<b>startPenService()</b>	启动笔服务	
<b>stopPenService()</b>	停止笔服务	
<b>unBindPenService()</b>	解除绑定后台服务	退出APP时需要解绑服务
<b>isServiceRunning(String serviceName)</b>	判断某个服务是否是启动状态	在使用笔服务和文件服务之前必须保证服务是已启动
<b>bindFileService()</b>	绑定文件服务	
<b>getFileService()</b>	获取文件服务	
<b>startFileService()</b>	启动文件服务	
<b>stopFileService()</b>	停止文件服务	
<b>unBindFileService()</b>	解除绑定后台服务	退出APP时需要解绑服务

注：相关的服务名称请参考cn.robotpen.model.symbol.Keys文件。

## 2. PenService类

针对的笔的连接、绘制以及在直播过程中对笔的状态进行了封装。基于蓝牙连接方式的笔和USB连接方式的笔，PenService又派生出SmartPenService和UsbPenService两个子类，可以根据实际情况使用某个子类。

方法	说明	注意事项
<b>getConnectDevice</b>	获取已经连接的笔设备对象	仅支持一个设备
<b>checkDeviceConnect</b>	检查笔设备的连接状态	如果需要针对设备的各个状态进行处理时，需要在此进行实现
<b>disconnectDevice</b>	断开与笔设备的连接	
<b>setOnConnectStateListener</b>	设置笔设备的连接状态变更监听	当设备插入或者拔出时会主动触发
<b>setOnTrailsClientChangeListener</b>	设置笔设备的轨迹监听	主要用于远程交互
<b>setOnPointChangeListener</b>	设置笔设备的坐标变更监听，用于绘制笔的轨迹	笔设备连接成功后，会通过OnPointChangeListener返回PointObject对象
<b>scanDevice</b>	触发主动扫描设备	为true说明已发现笔设备 查找状态会通过OnConnectStateListener返回

方法	说明	注意事项
<b>setAutoFindConfig</b>	设置是否自动发现设备 默认为false	主要用于蓝牙设备 usb设备是自动发现
<b>setBroadcastEnabled</b>		
<b>setScanTime</b>		
<b>setUserId</b>	设置用户ID	
<b>setSceneType</b>	设置场景类型	Keys中INCH_101表示10.1寸的设备板子
<b>setSceneOffset</b>	设置场景偏移	
<b>setLiveGroupId</b>	设置直播组ID	
<b>setLiveTargetId</b>	设置直播对象ID	
<b>getSceneWidth</b>	获取场景的宽度	
<b>getSceneHeight</b>	获取场景的高度	
<b>getSceneOffsetX</b>	获取X轴的偏移	
<b>getSceneOffsetY</b>	获取Y轴的偏移	
<b>startLive</b>	开始直播	
<b>stopLive</b>	停止直播	
<b>sendConnectState</b>	发送连接状态	
<b>sendTrails</b>	发送笔的轨迹	
<b>sendAudio</b>	发送音频	
<b>sendPointInfo</b>	发送笔的点坐标	

### 3. MultipleCanvasView类

提供的针对白板进行操作的封装，主要包括笔迹对象、画布对象、实现了切换背景颜色、切换背景图片、修改笔迹颜色、修改笔迹粗细、针对笔迹的擦除、插入图片和清屏等功能。

MultipleCanvasView可以通过在XML中创建，也可以通过new MultipleCanvasView(Context context,CanvasManagerInterface canvasManage); 创建,需要注意的是MultipleCanvasView必须被包含在一个RelativeLayout中，即MultipleCanvasView的父节点必须是RelativeLayout。

方法	说明	注意事项
<b>show</b>	显示当前画布	

方法	说明	注意事项
<b>hidden</b>	隐藏当前画布	
<b>refresh</b>	刷新当前的白板	
<b>getDrawAreaHeight</b>	获取绘制区域的高度	
<b>getDrawAreaWidth</b>	获取绘制区域的宽度	
<b>drawLine</b>	对PointObject对象进行绘制	
<b>addTrail</b>	添加笔迹	
<b>insertPhoto</b>	插入图片	
<b>setInsertShape</b>	插入图形	
<b>cleanPhoto</b>	清除图片	
<b>cleanScreen</b>	清屏，包括笔迹、图片、图形等全部	
<b>cleanShape</b>	清除图形	
<b>cleanTrail</b>	清除笔迹	
<b>MultipleCanvasView.Canvas ManagelInterface</b>		
<b>getBgColor</b>	获取背景颜色	
<b>getBgPhoto</b>	获取背景图片	
<b>getBgScaleType</b>	获取背景缩放形式	
<b>getDrawAreaParams</b>	获取绘制区域参数	
<b>getPenColor</b>	获取笔颜色	
<b>getPenModel</b>	获取笔模式	
<b>getPenWeight</b>	获取笔的粗细	
<b>getPenService</b>	获取笔服务	
<b>getFileService</b>	获取文件服务	

#### 4. 常用的交互类

- 录制ImageRecordModule类

录制功能将在白板上的操作过程包括笔迹、音频等录制成MP4格式的视频文件存储在本地设备中。在使用录制功能之前要确保上述的so文件已经成功导入到项目中。其次是要设置视频的保存路

径、录制尺寸、录制清晰的等级(默认为2标清)，在录制过程中和录制结束后通过实现 ImageRecordInterface 接口来完成相关的处理。

方法	说明	注意事项
<b>setSavePhotoDir()</b>	设置图片保存路径	
<b>setSaveVideoDir()</b>	设置视频保存路径	
<b>setRecordLevel()</b>	设置录制级别	
<b>setInputSize()</b>	设置录制尺寸	
<b>initImageRecord()</b>	初始化画布录制	
<b>startRecord()</b>	开始录制	
<b>endRecord()</b>	停止录制	
<b>setIsPause()</b>	暂停录制	
<b>saveSnapshot()</b>	保存截图	
<b>releaseImageRes()</b>	释放资源	
<b>getVideoName()</b>	获取当前录制视频的名称	
<b>getIsRecording()</b>	获取是否正在录制	
<b>ImageRecordModule.ImageRecordInterface</b>		
<b>fillImageBuffer()</b>	压制图片	
<b>recordTimeChange()</b>	录制时间变动记录	
<b>recordWarning()</b>	当录制有异常时状态记录	
<b>videoCodeState()</b>	当录制状态有变动时记录	progress>100 代表录制完成

注：录制之前需要设置保存路径和尺寸；  
录制完成的状态为progress>100。