
Android SDK 使用说明

USB 指纹读头

2024-12-20

各个 Android 版本上对权限有不同的要求，具体使用时请参考 Demo。鉴于 Android Studio 的版本，Demo 可能编译不过。请重新新建 Project，然后将代码与设置 Copy 到新的 Project，再编译新的 Project 一般都能通过。

SDK 在 Android 上调用有多种方式，请根据自己的项目特点，选择其中一种。

SDK 支持 Android6 以后各个版本，无需第三方库

一, API 说明

1, UsbReader 类(用于消息 Handler 方式)

```
//初始化比对库
public boolean InitMatch()
//比对两个指纹数据
public int MatchTemplates(byte[] piFeatureA,byte[] piFeatureB)

//注册 USB 监控器
public void ResumeRegister()
//停止 USB 监控器
public void PauseUnRegister()
//请求权限
public void requestPermission()

//打开设备
public int OpenDevice()
//关闭设备
public int CloseDevice()

//开始采集图像（异步函数，从消息中返回结果）
public int CaptureImage()
//开始采集比对特征点模板（异步函数，从消息中返回结果）
public int GenerateTemplate()
//开机登记特征点模板（异步函数，从消息中返回结果）
public int EnrolTemplate()

//获取指纹 RAW 图像
public int GetRawImage(byte[] lprawdate,int[] lpsize)
//获取指纹 bitmap 图像
public int GetBmpImage(byte[] lpbmpdate,int[] lpsize)
```

```
//获取比对特征点模板数据
public int GetTemplateByGen(byte[] lpTemplate,int[] lpsize)
//获取登记特征点模板数据
public int GetTemplateByEnl(byte[] lpTemplate,int[] lpsize)
```

2, AsyncUsbReader 类 (用于回调函数方式)

```
//打开设备
public int OpenDevice(Context parentContext)
//关闭设备
public int CloseDevice()

//设备采集图像 (回调函数中返回结果)
public void GetImage()
//从设备上传图像 (回调函数中返回结果)
public void UpImage()
//根据图像生成特征点 (回调函数中返回结果)
public void GenTemplate(int bufferId)
//合并特征点 (回调函数中返回结果)
public void RegModel()
//从设备上传特征点模板数据 (回调函数中返回结果)
public void UpTemplate()

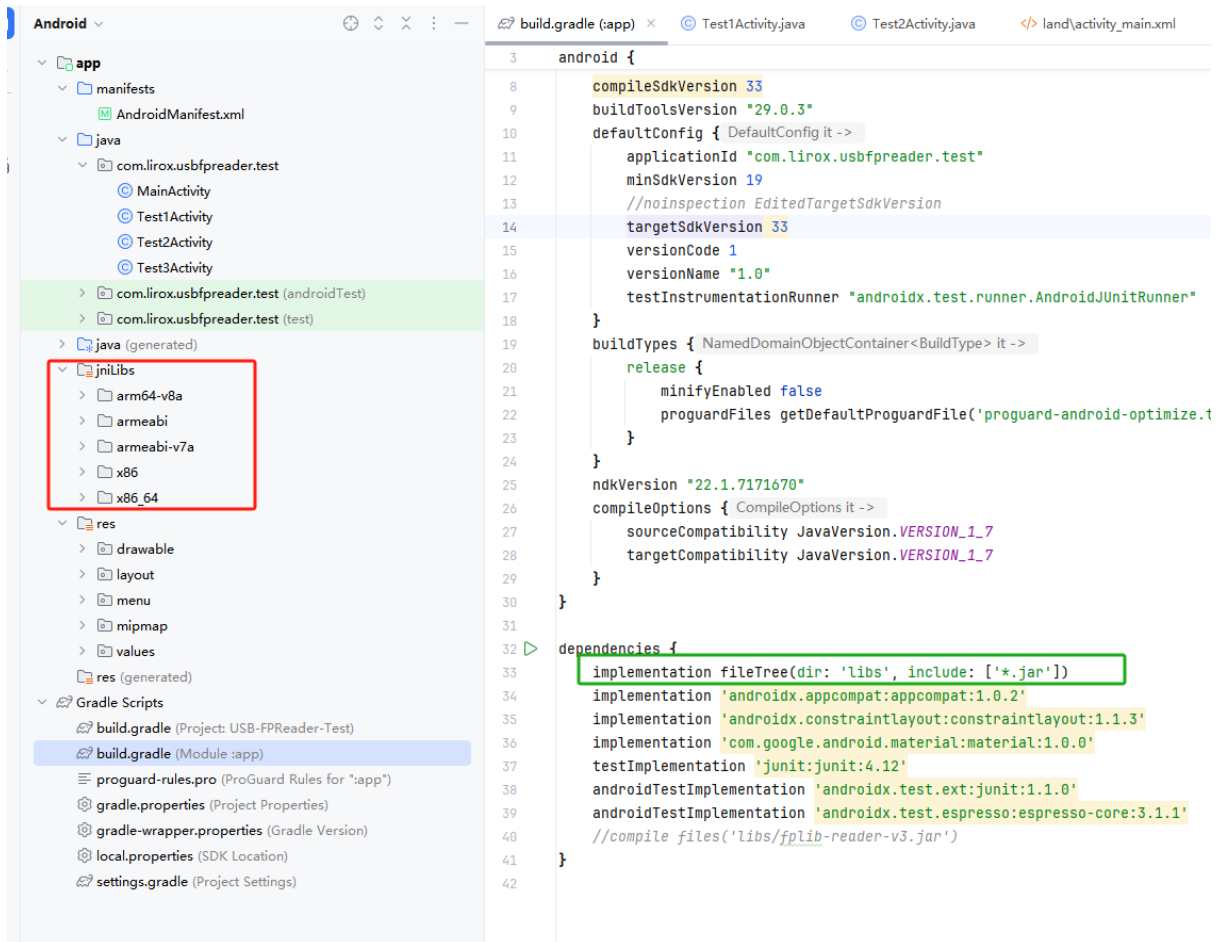
//设置回调函数
public void setOnGetImageListener(OnGetImageListener onGetImageListener)
public void setOnUpImageListener(OnUpImageListener onUpImageListener)
public void setOnGenTemplateListener(OnGenTemplateListener
onGenCharListener)
public void setOnRegModelListener(OnRegModelListener onRegModelListener)
public void setOnUpTemplateListener(OnUpTemplateListener onUpCharListener)
```

3, 其它函数

略

二, 使用说明

以 Demo 为例说明 SDK 的使用方法



- 1, 将各个平台的 so 库复制到 jniLibs 目录
- 2, 将 fplib-reader-v3.jar 复制到 libs 目录



- 3, USB 权限配置必不可少

1, Hander 消息方式 (Test1Activity)

```
//声明变量
private UsbReader fpModule;

//在 onCreate 中调用
fpModule=new UsbReader();
fpModule.InitMatch();
fpModule.SetContextHandler(this,mHandler);
fpModule.ResumeRegister();
fpModule.requestPermission();

//在 onDestroy 中调用
fpModule.PauseUnRegister();
```

调用 fpModule.EnrolTemplate();或 fpModule.GenerateTemplate()后, 在 mHandler 中得到结果, 截图如下

```

> @SuppressWarnings("HandlerLeak")
> private Handler mHandler = handleMessage(msg) -> {
>     switch (msg.what){
>         case Constants.FPM_DEVICE:{...}
>         break;
>         case Constants.FPM_PLACE:
>             tvStatus.setText("Place Finger");
>             break;
>         case Constants.FPM_LIFT:
>             tvStatus.setText("Lift Finger");
>             break;
>         case Constants.FPM_IMGCHECK:
>             tvStatus.setText("Lift Finger");
>             break;
>         case Constants.FPM_CAPTURE:{
>             if(msg.arg1==1){
>                 tvStatus.setText("Capture Image OK");
>             }else{
>                 tvStatus.setText("Capture Image Fail");
>             }
>             isworking=false;
>         }
>         break;
>         case Constants.FPM_GENCHAR:{...}
>         break;
>         case Constants.FPM_ENRFPT:{...}
>         break;
>         case Constants.FPM_NEWIMAGE:{...}
>         break;
>         case Constants.FPM_TIMEOUT:
>             tvStatus.setText("Time Out");
>             isworking=false;
>             break;
>     }
> };

```

2, Callback 回调方式 (Test2Activity)

//声明变量

```
private AsyncUsbReader vFingerprint;
```

在 onCreate 中调用

```
FPInit();
```

```
ResumeRegister();
```

```
requestPermission();
```

执行 vFingerprint.GetImage();开启采集指纹特征点

FPInit 函数用来设置回调函数。

1 个用法

```
private void FPInit(){
    vFingerprint= UserManager.getInstance().getAsyncUsbReader();

    vFingerprint.setOnGetImageListener(new OnGetImageListener(){...});

    vFingerprint.setOnUpImageListener(new OnUpImageListener() {...});

    vFingerprint.setOnGenTemplateListener(new OnGenTemplateListener() {...});

    vFingerprint.setOnRegModelListener(new OnRegModelListener() {...});

    vFingerprint.setOnUpTemplateListener(new OnUpTemplateListener() {...});
}

@Override
protected void onPause() {...}

@Override
protected void onResume() {...}

@Override
protected void onDestroy(){
    super.onDestroy();
    if(isopening)
        vFingerprint.CloseDevice();
    PauseUnRegister();
}
```