IDO-BLE SDK

快速接入

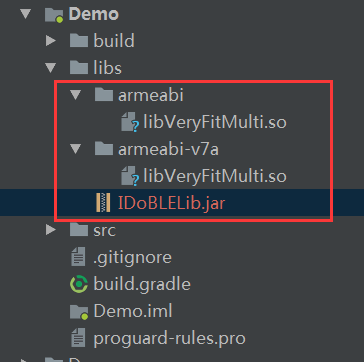
## 修改记录

|  |  |
| --- | --- |
| 版本 | V1 |
| 时间 | 2017.11.06 |
| 修改者 | 周治金 |
| 修改内容 | 创建文档 |

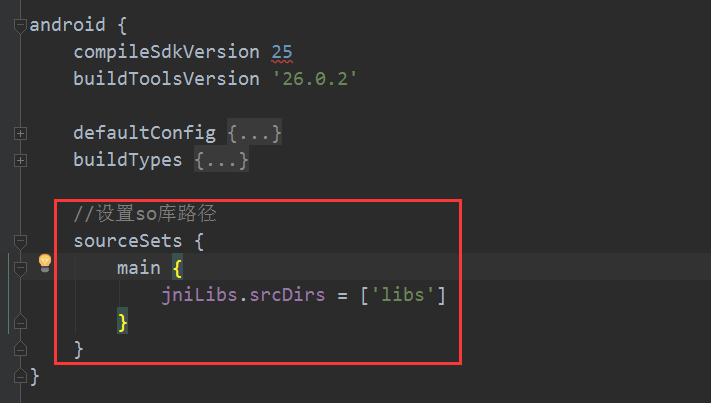
## 一 环境配置

1. 启动android studio，创建或打开你的APP工程

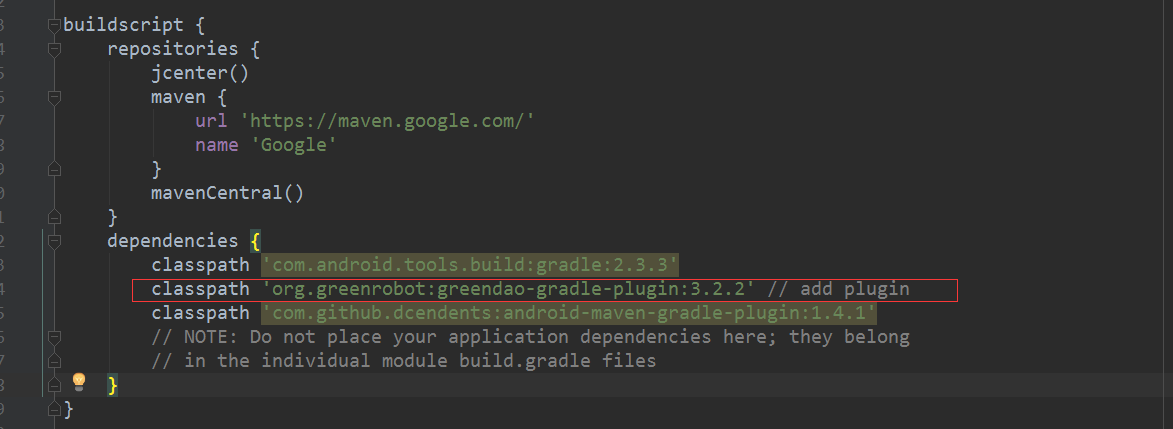
2. 将我们提供给你的so库、Jar包拷入到libs目录中，如下图(红色线条标记区域)：



3. 在moudle的build.gradle中加入以下配置(红色线条标记区域)，用来设置so库的路径：



在project的build.gradle中加入以下配置(红色线条标记区域)



classpath 'org.greenrobot:greendao-gradle-plugin:3.2.2'

开源库：

//SDK--need  
 compile 'org.greenrobot:greendao:3.2.2'

compile 'com.alibaba:fastjson:1.2.35'

compile 'com.squareup.okhttp3:okhttp:3.4.1'

compile 'com.squareup.okio:okio:1.9.0'

compile 'com.google.code.gson:gson:2.8.0'

compile 'com.android.support:support-core-utils:27.0.2'

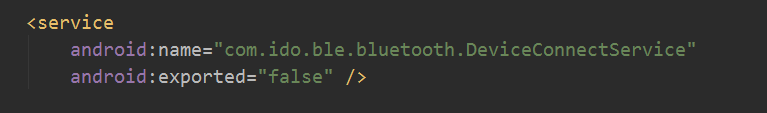
compile 'no.nordicsemi.android:dfu:1.6.1'

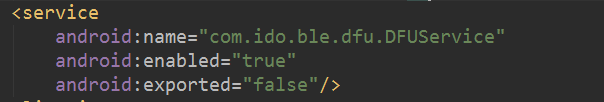
4. 在moudle的AndroidManifest.xml中加入以下配置：

权限：

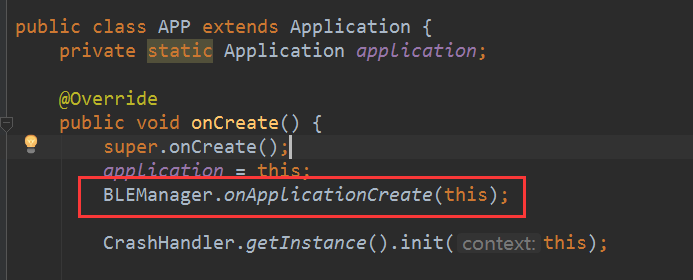
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
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

servicce：

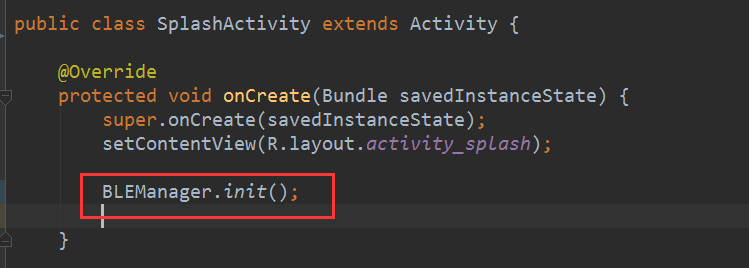




5. 在你Application页面中加入以下代码，如下图(红色线条标记区域)：



6. 在你的第一个Activity页面加入SDK初始代码，如下图(红色线条标记区域)：

****

**如果是6.0或以上的系统，你应先获取权限，然后再进行初始化操作**

## 二 初始化流程

**场景I：首次使用APP**

1. 扫描设备

2. 连接设备

3. 绑定设备

4. 设置用户信息

5. 设置单位

6. 设置每日目标

7. 同步配置信息

**场景II：解绑**

1. 扫描设备

2. 连接设备

3. 绑定设备

4. 同步配置信息

**注意：**

红色标记的步骤，必须执行，且需按指定步骤;

黑色字体标记的步骤，可不做配置；

具体例子，可参考我们提供的 Demo

## 三 回调机制

回调机制主要用来监听“连接状态”，“app与手环交互状态反馈”；

比如:

扫描设备时，通过设置扫描callback可以知道扫描到了哪些设备；

绑定时，通过绑定callback可以知道是否绑定成功；

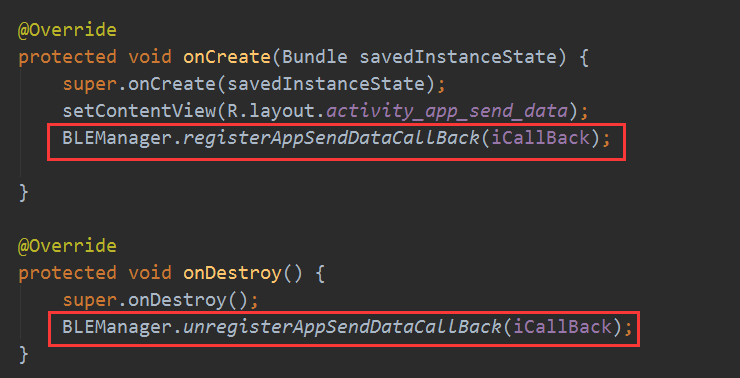
设置参数时，通过设置callback可以知道是否设置成功等...

注册callback与注销callback应成对出现，否则将会发生内存泄漏风险；

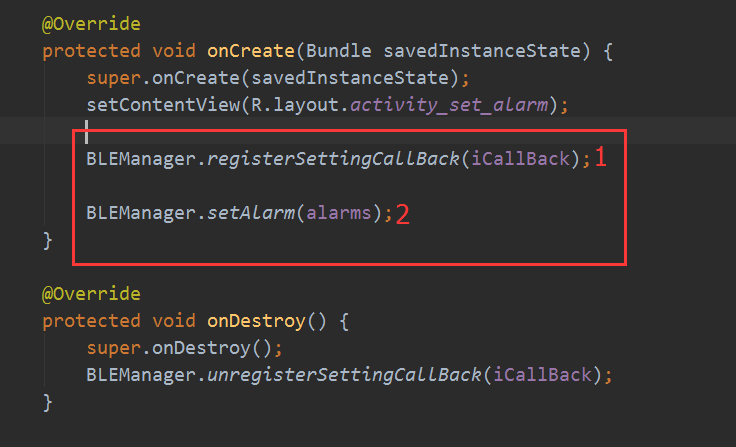
所有回调都是在主线程，所以不要在回调中做任何耗时（阻塞式）的操作！

正确使用方式如：

在onCreate中注册，在onDestroy中注销



在与手环通信之前，你应该先设置callback，然后再去操作；比如设置闹钟，你应先设置闹钟的callback，然后再向手环设备发送闹钟数据，这样你才能正常收到手环设备的反馈结果，来判断是否设置成功，如下图：

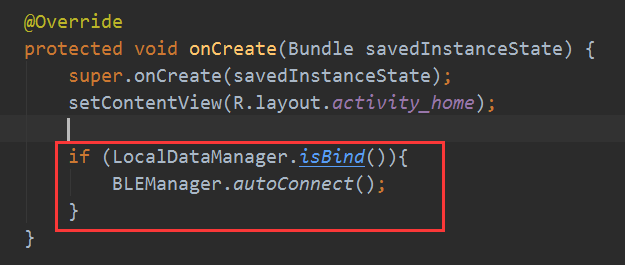


## 四 连接

首次使用App或者解绑后，都应先扫描设备，然后再调用BLEManger.connect()接口去连接设备；

只要前一次是绑定成功的状态，那么你就可以在你的初始化页面调用

BLEManager.autoConnect()去进行自动连接操作，如下图：



SDK具备重连机制，即当App与手环断开连接时，SDK将会进行重连操作，直到与上一次绑定的手环设备连接成功；

**重连机制将会停止在以下任何条件下：**

1. 手机蓝牙开关已关闭

2. App已于手环解绑

3. App进程被杀掉

3. 已连上手环设备

## 五 健康数据

**获取健康数据**：

你无法直接从手环中获取健康数据，只能从SDK的本地数据库中获取健康数据；

所以在获取健康数据之前，你应先调用BLEManager.startSyncHealthData()去同步健康数据，在你收到同步健康数据成功的回调之后，你就可以通过调用LocalDataMangaer.XXX()接口去获取健康数据了；

**同步数据的流程：**

   1. SDK发送同步命令，手环收到该命令之后，会返回所以的健康数据

   2. SDK收到健康数据之后， 会把数据存储在本地数据库DB中，然后App就可以通过SDK 提供的API去获取这些健康数据了

   3. 只要有一次同步是成功的，手环就会自动删除当天之外的所有健康数据；

**数据存储时长:**

 虽然同步之后，手环会删除当天之外的数据，但这些数据已经保存在sdk的数据中，所以不会有任何问题；

如果进行解绑操作，SDK将会清除所有健康数据；

 如果用户卸载App，那么sdk中的健康数据也将一并被清除，那么数据将永远无法找回；

 你可以做数据永久存储操作，即“同步完成之后，从sdk中获取健康数据，并把这些健康数据永久存储在远端服务器中”，那么用户即使卸载了你的应用，你也可以从远端服务器中获取到该用户之前的健康数据；

## 六 本地数据

SDK本地数据管理主要保存两类数据：

1. 健康数据

2. 设置参数数据（比如，设置每日目标，设置成功之后，SDK将会把这个参数保存到本地）

具体可以通过调用LocalDataManger.XXX()去获取你想要的数据

## 七 关键类文件

使用SDK，你将主要使用以下两个类：

1. BLEManager

2. LocalDataManger

## 八 注意事项

1. 在跟手环做交互之前，你都应该先判断，手机是否与设备处于连接状态；（通过BLEManager.isConnected()判断）

2. 如果是android 6.0及以上的系统，你不仅需要在AndroidManifest.xml中申请权限，你还应该还需要动态申请权限

3. 如果你需要判断SDK中的健康数据是否正确，在你的APP的Debug阶段，你可以使用“facebook的Stetho插件”来查看本地数据库数据；具体使用方法，请google搜索‘Stetho’；