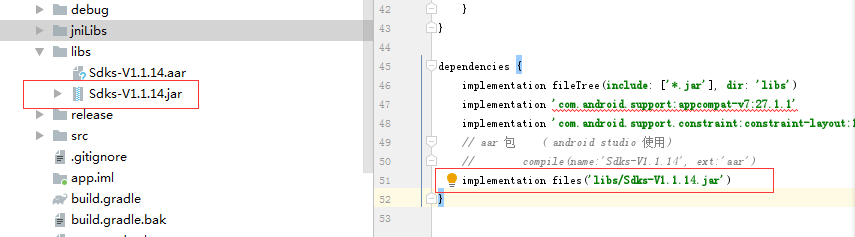
**倍泰体脂秤蓝牙SDK使用说明**

**编辑蓝牙SDK接口说明，仅供倍泰体脂秤开发者开发应用层使用**

# [1. SDK配置（Android studio）说明](#_Toc31860)

## [1.1Android studio配置](#_Toc18631)

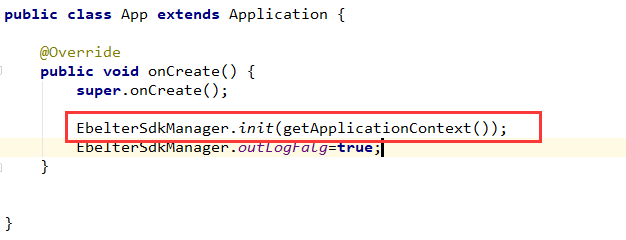
### 第一步: 把libs文件夹下的jar文件(或者aar文件)拷贝到工程目录app/libs目录下



### 第二步: 在项目module下的build.gradle文件下添加如下配置

![AL](B}KBDY@6OHVCL%77$]U](data:image/png;base64,)

### 第三步：在Application的onCreate方法中初始化一下sdk就能正常使用体脂秤了



注意：

minSdkVersion 必须大于等于18 *//否则不支持BLE*

targetSdkVersion 等于22 *//否则需要动态申请权限*

targetSdkVersion 大于22时需要自己手动添加动态权限申请

## [1.2代码示例](#_Toc10619)

Android SDK的最小版本和目标版本的范围：

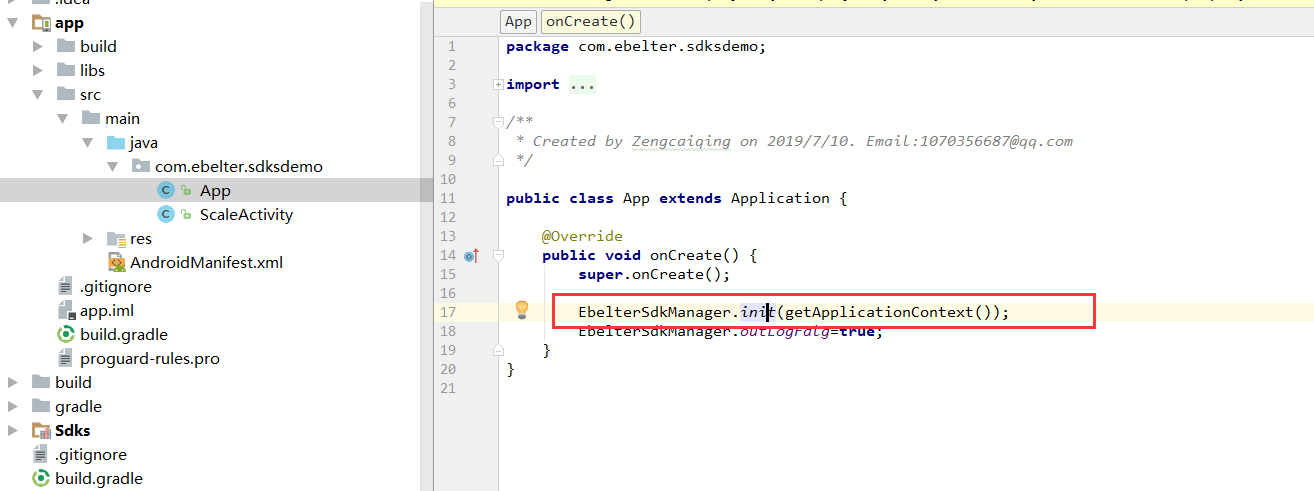


# [2. 蓝牙体脂秤引擎的使用说明](#_Toc22351)

## [2.1蓝牙引擎模块初始化](#_Toc11865)

在Application里面初始化一下EbelterSdkManager

EbelterSdkManager.*init*(getApplicationContext());



## [2.2代码示例](#_Toc2456)

### 第一步 初始化一个ScaleManager对象出来

*//-----------01***mScaleManager**=**new** ScaleManager(**this**);

### 第二步 给ScaleManager对象设置各种监听器。

**a手机蓝牙开关监听器IBlueStationListener**

**b蓝牙体脂秤连接状态监听器IConnectStationCallback**

**c蓝牙体脂秤测量监听器IScaleMeasureCallback**

**//-----------02  
mScaleManager.addBluetoothStationListener(mBlueStationListener);//Add a mobile phone bluetooth status listener  
mScaleManager.setConnectStationCallback(mConnectStationCallback);//Add bluetooth body fat scale connection status listener  
mScaleManager.setScaleMeasureCallback(mScaleMeasureCallback);//Add a measurement listening interface**

### 第三步 给ScaleManager设置连接的MAC地址（不设置默认为null，将随机连接附近蓝牙名称符合条件的设备）或蓝牙名称关键字（不设置时将连接蓝牙名称中包含“body fat b”字符串的设备），如果设置了地址或蓝牙名称关键字，引擎只会连接你刚才设置的蓝牙设备.

#### 根据蓝牙地址连接

**mScaleManager.setMustConnectBlueAddress(“xx:xx:xx:xx:xx:xx”); //此方法有返回值 true说明设置成功 false说明设置失败**

**mScaleManager.startWork();**

#### 根据蓝牙名称连接

**mScaleManager.setDeviceBleName(isHeartScale,bleNameKeyStr); // 参数1 是否为心率秤 参数 2 蓝牙名称关键字**

**mScaleManager.startWork();**

#### 默认扫描周围设备

**mScaleManager.startWork();**

### 第四步 设置用户信息，每次信息有变化时都要及时更改ScaleUserInfo用户信息

**//-----------04 Set user information  
//Set the user information, and remember to execute once the user information changes  
ScaleUserInfo userInfo = ScaleUserInfo.getScaleUserInfo();  
String** **userId="0123456789a"; //userId最长为11个字符，且每个字符是16进制中的数字**

**//0123456789abcdef**

**int sex=0; //0代表男 1代表女  
int height=168; //168代表168cm高  
int roleType=0; //0代表普通人 1代表运动员**

**int age=30; //30岁  
float weight = 50.5f; //第一次测量给一个你大体的体重值  
userInfo.setUserId(userId);  
userInfo.setAge(sex);  
userInfo.setHeight(height);  
userInfo.setRoleType(roleType);  
userInfo.setAge(age);  
userInfo.setWeight(weight);  
mScaleManager.updateUserInfo(userInfo);**

**第五步 让引擎开始工作起来，开始连接设备，在监听器里面接收相对应的信息**

*//-----------05***mScaleManager.startWork();**

## 2.3 附加功能说明

### 如果想让引擎暂停工作可以调用

**mScaleManager.stopWork()，暂停之后继续调用mScaleManager.startWork()可以让引擎继续工作**

### 引擎退出的时候记得要释放资源 如下：

**@Override  
protected void onDestroy() {  
 super.onDestroy();  
 if (mScaleManager!=null){  
 mScaleManager.removeBluetoothStationListener(mIBlueStationListener);  
 mScaleManager.exit();  
 mScaleManager=null;  
 }  
}**

### ScaleManager里面还有两个kg和lb单位互转的方法

**public static float kg2lb(float valueKg){  
 return ScaleProduct.kg2lb(valueKg);  
}  
public static float lb2kg(float valueLb){  
 return ScaleProduct.lb2kg(valueLb);  
}**

### 4、请求某个用户的离线数据

**mScaleManager.requestOffMesureData(userId,height,sex,age,roleType);**

### 5、清空体脂秤里面的所有用户信息

**mScaleManager.deleteAllUserInfo();**

### 判断是否是蓝牙地址的方法

**ScaleManager.checkBluetoothAddress(address);//如果返回true说是蓝牙地址 返回false说明不是蓝牙地址**

### 7、扫描蓝牙设备BlueScan类及使用方法

#### // 01初始化一个扫描器

**BlueScan mBlueScan=new BlueScan(this);**

//02设置扫描回调器**mBlueScan.setScanCallback(new IScanCallback() {  
 @Override  
 public void beforeScan() {  
 //扫描开始  
 }  
 @Override  
 public void scanOneDevice(BluetoothDevice device, int rssi, byte[] scanRecord) {  
 //扫描到设备的回调方法  
 String deviceName=device.getName();  
 String deviceAddress=device.getAddress();  
 LogUtils.i(TAG,"扫描到设备--deviceName = "+deviceName+"--deviceAddress = "+deviceAddress);  
 }  
  
 @Override  
 public void overScan() {  
 //扫描结束  
 }  
});**

//03开始扫描  
**mBlueScan.startScanBluetoothDevices();**

# 3.接口及类的说明

## 3.1 ScaleUserInfo类

**用户信息类，此类是单例，用来记录用户信息的，测量用户信息有变化要及时更改此类的相对应的数据，并及时调用mScaleManager.updateUserInfo(userInfo)方法。大体字段如下：**

**private String userId;//userId最长为11个字符,且每个字符是16进制中的数字**

**//0123456789abcde  
private int sex; //0代表男 1代表女  
private int height; //168代表168cm高  
  
private int roleType; //0代表普通人 1代表运动员  
private int age; //年龄  
private float weight;//体重，第一次测量填写跟自己体重差不多的就可以了**

## 3.2 IBlueStationListener 接口

**手机蓝牙开关状态的监听器，里面有四种状态的回调方法。**

**private final IBlueStationListener mBlueStationListener = new IBlueStationListener() {  
 @Override  
 public void STATE\_OFF() {  
 //手机蓝牙关闭了**

**}  
 @Override  
 public void STATE\_TURNING\_OFF() {  
 //手机蓝牙正在关闭中  
 }  
 @Override  
 public void STATE\_ON() {**

**//手机蓝牙打开了  
 }  
 @Override  
 public void STATE\_TURNING\_ON() {**

**//手机蓝牙正在打开  
 }  
};**

## 3.3 IConnectStationCallback接口

**体脂秤连接状态的监听器，里面有三种状态的回调方法。**

**private final IConnectStationCallback mConnectStationCallback=new IConnectStationCallback() {  
 @Override  
 public void onConnected(ProductStyle productStyle, BluetoothDevice device) {  
 //体脂秤连接成功了  
 }  
 @Override  
 public void onConnecting(ProductStyle productStyle, BluetoothDevice device) {  
 //体脂秤正在连接  
 }  
 @Override  
 public void onDisConnected(ProductStyle productStyle) {  
 //体脂秤断开连接了  
 }  
};**

## 3.4 IScaleMeasureCallback接口

**体脂秤测量接口回调，里面有多种方法**

**private final IScaleMeasureCallback mScaleMeasureCallback=new IScaleMeasureCallback() {  
 @Override  
 public void onScaleWake() {  
 //体脂秤亮屏了  
 }  
 @Override  
 public void onScaleSleep() {  
 //体脂秤息屏了，在这里如果有离线数据的需求，可以去获取离线数据  
 }  
 @Override  
 public void onReceiveMeasureResult(ScaleMeasureResult result) {  
 //测量结果的回调  
 }  
 @Override  
 public void onWeightOverLoad() {  
 //体脂秤超重了的提示回调  
 }  
 @Override  
 public void onReceiveHistoryRecord(OfflineMeasureResult result) {  
 //接收到离线历史数据  
 }  
 @Override  
 public void onLowPower() {  
 //秤的电量低  
 }  
 @Override  
 public void setUserInfoSuccess() {  
 //更新用户信息成功  
 }  
 @Override  
 public void receiveTime(long time) {  
 //接收到秤的时间  
 }  
};**

## 3.4 ScaleMeasureResult类

**用来装载测量结果的信息的，这个类有如下几个字段**

**public String userId; //测量的用户id  
public int age; //年龄  
public int sex; //0男 1女  
public int height; //身高  
public int roleType; //0普通 1运动员  
  
public String measureTime;//测量时间 格式如:1991-06-13 10:08:08  
  
public boolean isOnlyWeightData;//用来表示当前测量的是否是体重数据 如果为true说明是//先测量先回来的体重数据 里面只有体重数据， 如果为false说明是一次完整的测量数据 ，说明//是测量体脂以后返回来的测量信息  
  
/\*阻抗\*/  
public float resistance;//电阻  
  
/\*体脂率\*/  
public float fat; //脂肪率，单位是%  
  
/\*体重\*/  
public float weight; //体重，单位为KG  
  
/\*水分率\*/  
public float waterRate; //水分率，单位为%  
  
/\*基础代谢率\*/  
public float bmr; // 基础代谢率  
  
/\*内脏脂肪等级\*/  
public float visceralFat;//内脏脂肪等级  
  
/\*肌肉量\*/  
public float muscleVolume;//肌肉量，单位是KG   
  
/\*骨骼肌\*/  
public float skeletalMuscle;//骨骼肌，单位是KG  
  
/\*骨量\*/  
public float Bonemineralcontent;//骨盐量，单位是KG  
  
/\*BMI\*/  
public float bmi;//BMI指数  
  
/\*蛋白质\*/  
public float protein; //蛋白质，单位是%  
  
/\*身体得分\*/  
public float bodyScore;//身体得分  
  
/\*身体年龄\*/  
public float bodyAge;//身体年龄**

**/\*心率\*/  
public int heartRate; //有些型号的秤不支持心率功能，这种情况下此字段无效**

**//体重单位  
public String weightUnit ="kg"; //当前体脂秤屏幕上显示的单位，有kg和lb两种，app需根据体脂秤屏幕上显示的单位来设置自己显示的测量结果的单位，当体脂秤的单位是lb时，需调用ScaleManager.kg2lb(float kgValue)接口转换为以lb为单位的值后显示。**