目錄

概述

English Document

该文档用于引导Yolanda客户接入云康宝的智能设备。

该SDK支持设备请参考设备类型

对接步骤

1、商务申请

跟Yolanda商务人员签订合作协议，我司会提供开发所需的 `appid` 和 `配置文件`

我们会提供一个表格，只要按照要求填写表格，发给我们的人员处理，通常很快就会给到上述两样东西

> 千万不要在正式APP使用 `123456789` 这个测试的appid，它不稳定，我们随时可能封杀它，或变更测试用的appid

2、下载研究SDK

下载相应平台的SDK，使用上面申请到的appid和配置文件运行demo，研究下demo和文档是如何使用SDK的。

强烈建议研究demo后，再接入到自己的APP，另外可以先看看FAQ，仔细阅读其中的问题，也许你的疑惑就会解开。

> 建议开发人员 `star` 我们的SDK项目，这样可以及时收到更新通知

3、设计体脂秤功能

设计如何在现有APP中接入体脂秤。上一步研究好SDK后，也能为这一步设计体脂秤避免不少雷区。

通常体脂秤的使用步骤为：

1. 添加设备
   > 这个并不是跟安卓或iOS系统配置，蓝牙4.0是可以支持不配对使用的。我们的体脂秤是属于自定义协议，通常只有我们的APP或接入了我们SDK的APP才能使用。不能添加至系统，用系统蓝牙连接。
2. 测量
   > 这里的测量其实包括蓝牙扫描和蓝牙连接，在测量过程中，最好展示一些测量动画，提示用户正在测量，避免使用过程太枯燥
3. 保存和展示数据
   > 收到锁定测量数据后，就把数据保存在本地并进行展示。为了达到离线测量，可以先把数据保存本地，然后再上传到服务器，如果当时没有网络，则可以等后续有网络再上传数据到服务器。展示数据的方式可以参考我们的APP，当然如果有自己的风格和分析方式，也可以用自己的。

4、开发APP功能

根据设计好的功能，把SDK中接入APP中。

这里要特别注意安卓的定位权限和定位服务开关。iOS则主要13版本后，需要蓝牙权限。

5、自测功能

开发完成后，可以先使用我们的提供测试用例SDK测试用例，验证称重功能是否能够使用。

> 上述附件可以鼠标右键点击下载

如果可以的话，建议把测试版本也发给Yolanda，我们会安排测试人员辅助客户的APP测试。

我们会把测试的结果以及我们对APP的一些看法提到给客户。

6、提交上线

测试完成，就提交上线。上线成功最好也通知Yolanda。这样Yolanda会登记上线结果，继续跟进体验。

SDK 下载链接

SDK文件已经不再单独下载链接，下文是我们提供的Demo，Demo项目中有最新的SDK文件。

iOS

Android

集成方式

Android集成方式

iOS集成方式

开发流程

使用SDK前，需要先调用初始化，方法为：QNBleApi.initSdk

代码示例：

```
//安卓示例代码
String configFilePath = "file:///android_asset/test123456789.qn";
QNBleApi.getInstance(context).initSdk("test123456789", configFilePath, new QNResultCal
    @Override
    public void onResult(int code, String msg) {
        Log.d("BaseApplication", "初始化文件" + msg);
    }
});
```

```
NSString *file = [[NSBundle mainBundle] pathForResource:@"123456789" ofType:@"qn"];
[[QNBleApi sharedBleApi] initSdk:@"123456789" firstDataFile:file callback:^(NSError *e
}];
```

初始化方法是任何设备都需要调用的方法，下面是不同的设备类型的工作流程。

如果不清楚自己手上是何种设备，可以参考常见问题中的如何判断设备类型

Android 快速集成

Android Studio

Android 使用gradle集成

- 在你工程的根目录下的 **build.gradle**添加**jitpack**支持

```
    allprojects {
        repositories {
            //其它仓库配置
            maven { url 'https://jitpack.io' }
    }
}
```

- 在你的module的根目录下的**build.gradle**添加依赖最新版本查看地址

```
<!--这里的版本号，x.x.x 可以指定为任意release版本-->
<!--如果希望一直使用最新版本可以替换 x.x.x 为 master-SNAPSHOT -->

dependencies {
        ...
        compile 'com.github.YolandaQingniu:qnscalesdkX:x.x.x'
}
```

本地依赖

- 下载最新的jar和so库，导入下载的.zip压缩包中的 `jar和so库`
- 在清单文件中申请蓝牙权限、位置权限、网络权限（离线SDK不需要）

```
    xml
   <!--蓝牙权限-->
   <uses-permission android:name="android.permission.BLUETOOTH" />
   <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
   <!--6.0及之后需要动态申请-->
   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
   <!--用来存储日志-->
   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <!--如果是在线的sdk需要网络权限-->
   <uses-permission android:name="android.permission.INTERNET" />
   <!--qnscalesdk:1.1.3-beta3 之前(包含1.1.3-beta3)的版本需要增加此权限，之后的版本不需
  <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
  <!--qnscalesdk:1.1.3-beta3 之后(1.1.3-beta4开始)的版本需要增加此权限，之前的版本不需要
   <uses-permission android:name="android.permission.WAKE_LOCK" />
```

- 需要在**AndroidManifest.xml**注册SDK中的组件：

```xml
 <!--qnscalesdk:1.1.3-beta3之前(包含1.1.3-beta3)的版本配置-->
<service android:name="com.qingniu.qnble.scanner.BleScanService"/>
<service android:name="com.qingniu.scale.measure.ble.ScaleBleService"/>
<service android:name="com.qingniu.scale.measure.broadcast.ScaleBroadcastService"/>
<service android:name=".measure.broadcast.ScaleFoodBroadcastService" />


<!--qnscalesdk:1.1.3-beta3 之后(1.1.3-beta4开始)的版本配置-->
<service android:name="com.qingniu.qnble.scanner.BleScanService" android:permission="a
<service android:name="com.qingniu.scale.measure.ble.ScaleBleService" android:permissi
<service android:name="com.qingniu.scale.measure.broadcast.ScaleBroadcastService" andr
<service android:name="com.qingniu.scale.wsp.ble.ScaleWspBleService" android:permissic
<service android:name="com.qingniu.scale.measure.broadcast.ScaleFoodBroadcastService"
<service android:name=".measure.broadcast.ScaleFoodBroadcastService" android:permissic
<service android:name="com.qingniu.heightscale.ble.HeightScaleBleService" android:perm
```

- SDK中使用到了v4包的资源，开发者项目中需要引入v4包的资源

混淆配置(proguard-rules)

```
-keep class com.qingniu.scale.model.BleScaleData{*;}
```

其它

配置文件的集成方式可以参考demo。放在 `assets` 目录，然后在初始化时使用："file:///android_asset/文件名.qn"来传文件路径

iOS 快速集成

SDK的运行需要appid以及配置文件，商家在接入时可先使用Yolanda提供的测试appid和测试配置文件，正式发布时必须向Yolanda官方获取正式appid和配置文件

安装方式

cocoapods安装:

- 先安装Cocoapods；
- 通过 pod repo update 更新QNSDK的cocoapods版本；
- 在Podfile对应的target中，添加 `pod 'QNSDK'` ，并执行pod install；
- 在项目中使用CocoaPods生成的.xcworkspace运行工程；
- 在你的代码文件头引入头文件 `#import <QNSDK/QNDeviceSDK.h>`

Carthage安装:

- 安装 Carthage；
- 打开 Cartfile, 添加 `github "https://github.com/YolandaQingniu/sdk-ios-demo.git"` ；
- 打开命令行, cd 到你的 project 目录, 输入 carthage update；
- 将 Carthage/Build/ 目录下的 `QNSDK.framework` 拖到你的项目工程配置的 Build Phases -> Linked Binary and Libraries 里面；
- 在你的代码文件头引入头文件 `#import <QNSDK/QNDeviceSDK.h>`

手动安装:

- 下载SDK安装包至工程
- 引入SDK路径 【TARGETS】-> 【Build Setting】->【Search Paths】->【LibrarySearch Paths】中添加SDK路径
- 配置链接器 【TARGETS】-> 【Build Setting】-> 【Linking】-> 【Other Linker Flags】中添加 `-ObjC` 、 `-all_load` 、 `-force_load [SDK路径]` 其中之一

工程的配置

- 在Info.plist中有对 【Privacy - Bluetooth Peripheral Usage Description】 【Privacy - Bluetooth Always Usage Description】 键 进行蓝牙的使用说明

注意事项

- SDK适配8.0及以上系统
- iOS10.0及以上系统必须Info.plist中配置蓝牙的使用说明，否则无法使用系统的蓝牙功能
- iOS13.0及以上系统必须Info.plist中配置蓝牙的使用授权说明，否则会发现奔溃

- 必须为SDK配置链接器，否则SDK无法正常运行

WSP 蓝牙 WiFi 秤

WSP 蓝牙 WiFi 秤是我司推出的一款支持自动识别用户测量，脱离 app 直接在秤显示屏上展示测量结果，并将测量结果自动上传的智能设备。

该设备避免了用户在测量过程中需要使用蓝牙连接设备才能进行详细测量的步骤，提升用户体验。只要使设备连接网络，并向设备注册一次需要测量的用户信息，后续该用户便可脱离 app 使用。已注册用户在无需使用手机的情况下，上秤测量，测量完成后秤会自动识别该测量数据属于哪个注册用户，识别成功后会在设备上显示该次测量的各项指标，同时将测量数据上次到对应的服务器。

秤端用户简述

目前 WSP 设备最多允许 8 个用户，秤端的用户需要通过蓝牙连接的方式向设备注册用户。当注册满 8 个用户后，再次向设备注册用户时，秤端会返回注册失败的状态。因此建议您注册用户之前，先判断是否需要删除秤中的无效用户，以避免无效用户占据资源。

当向设备注册用户时，需要给设备下发该用户秘钥（秘钥一般由服务器对该设备生成用户的唯一标识），注册成功时，设备会记录该用户秘钥同时返回该用户的唯一序列号。因此注册该用户成功后，app 需要将设备返回该用户的序列号与秘钥保存起来。当通过蓝牙连接更新用户资料或用户通过蓝牙连接测量时，需要先使用用户序列号与秘钥访问秤端的指定用户成功后才能进行。同时序列号也是作为设备上传测量数据属于哪个注册用户的标识，以及作为删除用户的必要条件。

- 注册用户

  秤端能缓存已经注册的用户信息，通过已经注册的用户信息识别未连接蓝牙时的测量数据属于哪个用户，从而使用识别到的用户信息获取详细测量数据

- 访问用户

  对秤进行操作时都需要先访问已注册的用户，访问成功后才能进相应的操作，比如修改用户信息、连接蓝牙进行测量等

- 删除用户

  当已经向秤注册用户后，后续用户解绑该秤，需删除秤端的用户，避免资源占用

访客模式测量

访客模式是秤体提供的一个公共通道的测量方式，该方式无需注册用户直接使用，访客模式使用的是临时用户，该用户的存在时间为从连接到断开，当设备断开连接时临时用户即销毁，同时在访客模式下不会产生与接收存储数据

WSP 设备协议简述

wsp 设备的协议是在WSP 官方协议的基本上进行优化修改而成。wsp 协议交互比较复杂，以下是该设备协议涉及的部分

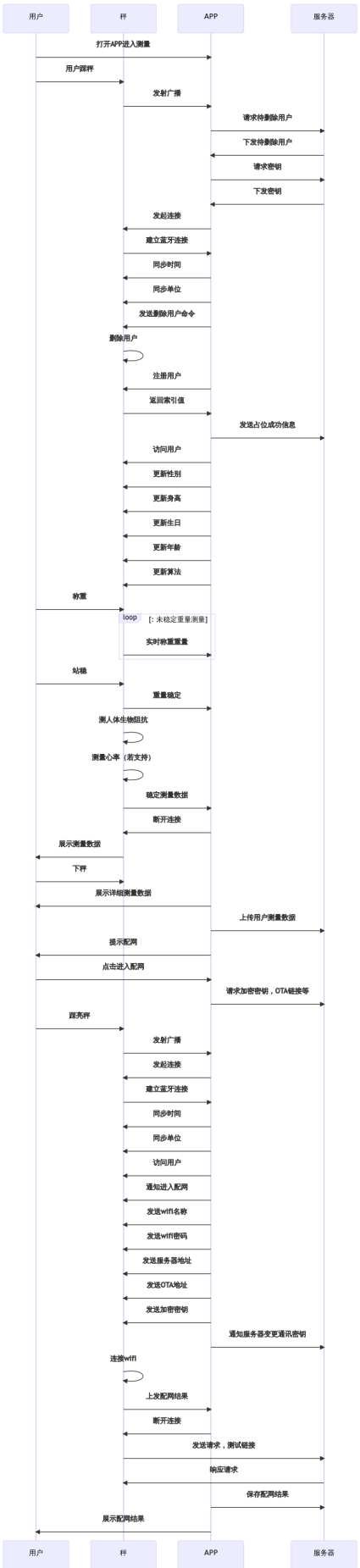
协议涉及的方面：

- 用户方面
  - 注册用户
  - 访问用户
  - 更新用户资料
    - 年龄
    - 身高
    - 性别
  - 删除用户
- 测量方面
  - 测量状态
  - 实时体重
  - 存储数据
  - 测量数据
- 配网方面
  - WiFi 名称
  - WiFi 密码
  - 数据上传地址
  - OTA 升级地址
  - 通讯秘钥
  - 配网结果
- 其他
  - 单位

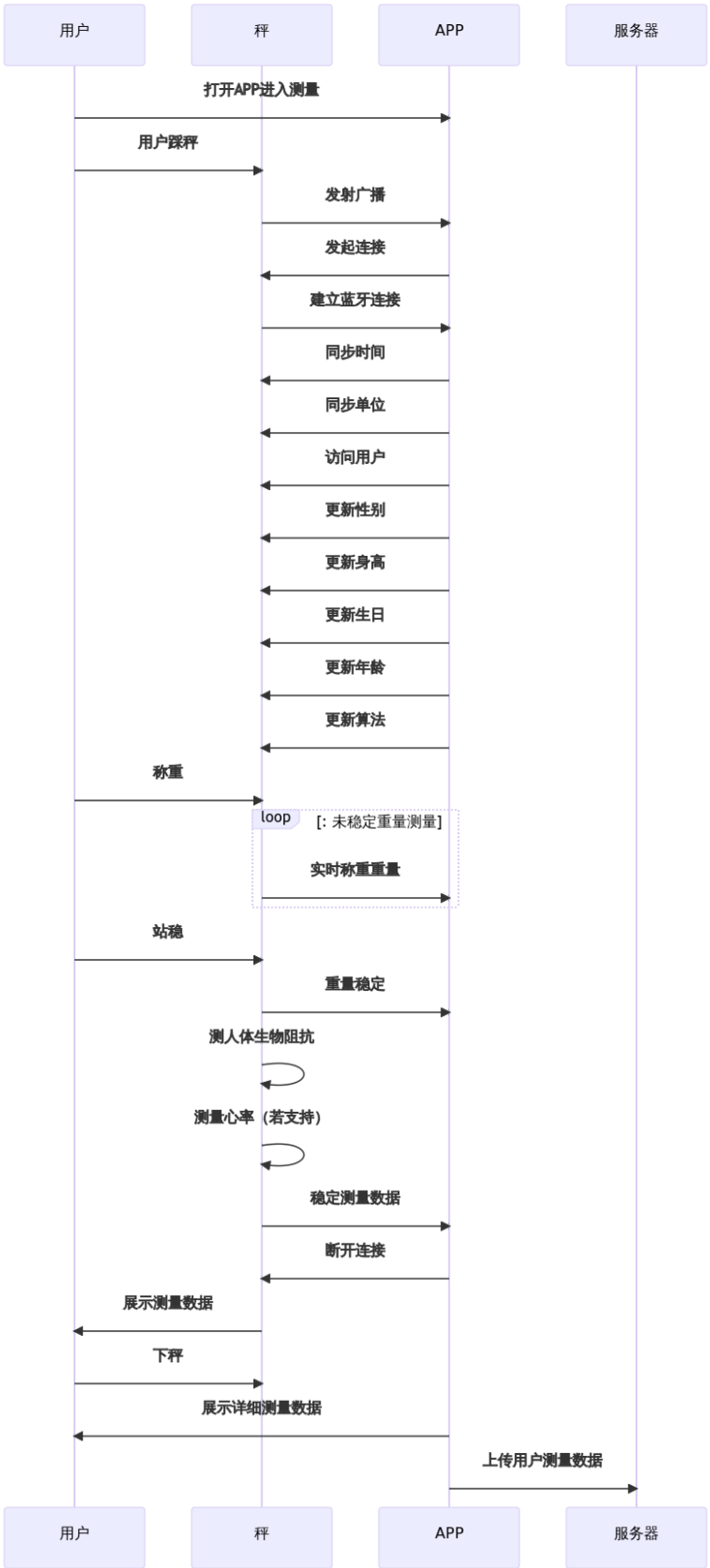
该设备的涉及的方面比较多，并且协议之间的通讯过程以及使能方式都要严格的要求，我司目前已经将其封装成 SDK 方便接入使用。

不同情况下的工作流程

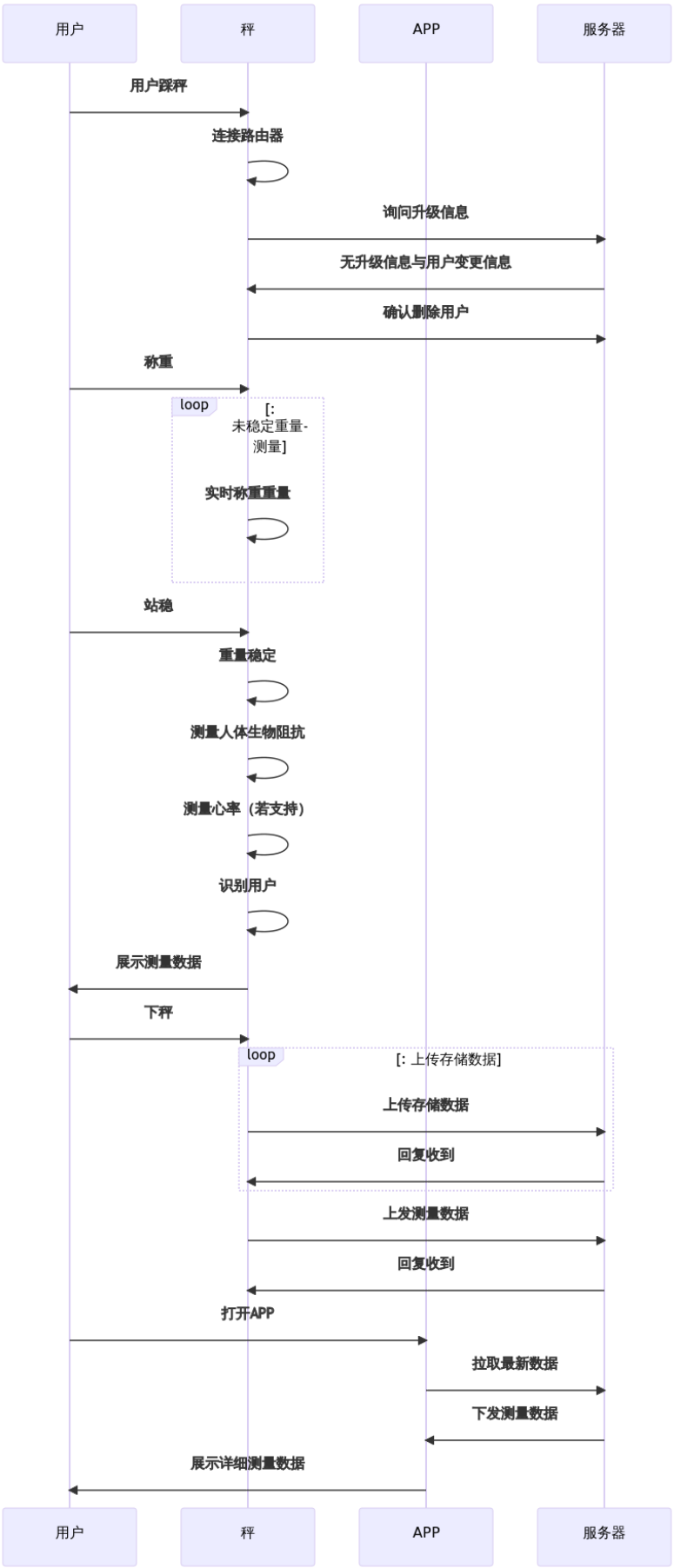
- 首次使用绑定配网


用户
秤
APP
服务器
打开APP进入测量
用户踩秤
发射广播
请求待删除用户
下发待删除用户
请求密钥
下发密钥
发起连接
建立蓝牙连接
同步时间
同步单位
发送删除用户命令
删除用户
注册用户
返回索引值
发送占位成功信息
访问用户
更新性别
更新身高
更新生日
更新年龄
更新算法
称重
loop
[: 未稳定重量测量]
实时称重重量
站稳
重量稳定
测人体生物阻抗
测量心率（若支持）
稳定测量数据
断开连接
展示测量数据
下秤
展示详细测量数据
上传用户测量数据
提示配网
点击进入配网
请求加密密钥，OTA链接等
踩亮秤
发射广播
发起连接
建立蓝牙连接
同步时间
同步单位
访问用户
通知进入配网
发送wifi名称
发送wifi密码
发送服务器地址
发送OTA地址
发送加密密钥
通知服务器变更通讯密钥
连接wifi
上发配网结果
断开连接
发送请求，测试链接
响应请求
保存配网结果
展示配网结果
用户
秤
APP
服务器

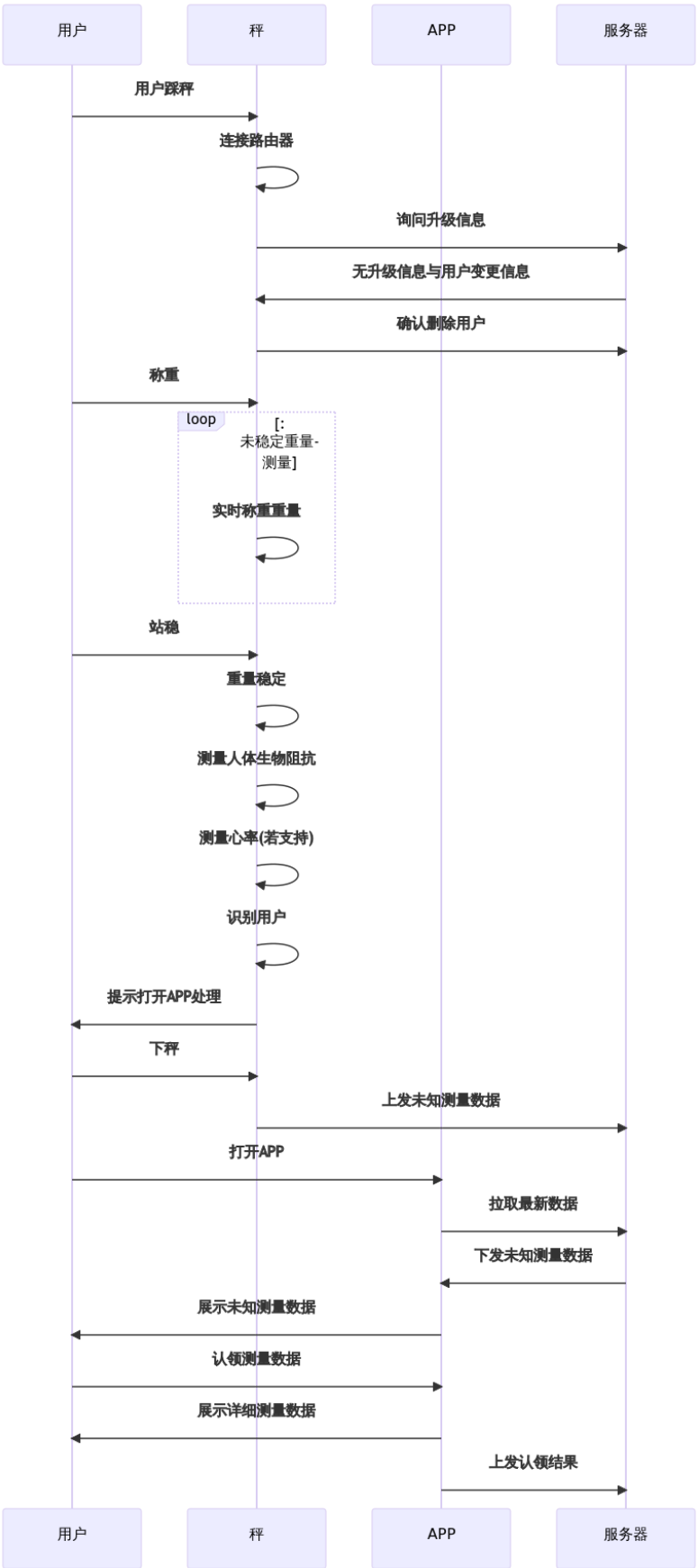
- 连接蓝牙的测量



- 普通测量


用户
秤
APP
服务器
用户踩秤
连接路由器
询问升级信息
无升级信息与用户变更信息
确认删除用户
称重
loop
[:
未稳定重量-
测量]
实时称重重量
站稳
重量稳定
测量人体生物阻抗
测量心率（若支持）
识别用户
展示测量数据
下秤
loop
[: 上传存储数据]
上传存储数据
回复收到
上发测量数据
回复收到
打开APP
拉取最新数据
下发测量数据
展示详细测量数据
用户
秤
APP
服务器

- 未识别用户的测量


用户
秤
APP
服务器
用户踩秤
连接路由器
询问升级信息
无升级信息与用户变更信息
确认删除用户
称重
loop
[:
未稳定重量-
测量]
实时称重重量
站稳
重量稳定
测量人体生物阻抗
测量心率(若支持)
识别用户
提示打开APP处理
下秤
上发未知测量数据
打开APP
拉取最新数据
下发未知测量数据
展示未知测量数据
认领测量数据
展示详细测量数据
上发认领结果
用户
秤
APP
服务器

一、初始化

使用QNBleApi.initSdk进行初始化

二、扫描

1. 设置蓝牙扫描回调监听类

在扫描前，需要设置监听类，方法为QNBleApi.setBleDeviceDiscoveryListener，该方法只需要调用一次即可，在确定不需要扫描记得设置为 null/nil

android 示例：

```
QNBleApi.getInstance(context).setBleDeviceDiscoveryListener(new QNBleDeviceDiscoveryLi
    @Override
    public void onDeviceDiscover(QNBleDevice device) {
        //该方法回调设备对象，可以设备对象进行处理
    }

    @Override
    public void onStartScan() {
        //开始扫描的回到
    }

    @Override
    public void onStopScan() {
        //结束扫描的回到
    }

    @Override
    public void onScanFail(int code) {
        //扫描失败的回调，会有错误码返回，具体可以参考 错误码详情页
    }
});
```

iOS 示例：

```
//设置代理
QNBleApi *bleApi = [QNBleApi sharedBleApi];
bleApi.discoveryListener = self;

//实现代理方法
- (void)onDeviceDiscover:(QNBleDevice *)device {
    //该方法会在发现设备后回调
}

- (void)onStopScan {
    //停止扫描时回调
}

- (void)onStartScan {
    //开始扫描时回调
}
```

2. 启动扫描

确认蓝牙已打开，安卓这边还需要检查下 `定位权限` 和 `定位开关` 。如果确认蓝牙已打开，定位权限已授权，定位服务开关已打开，则可以开始蓝牙扫描

> 安卓 6.0 以后，对 targetSdkVersion>=23 以上的 APP，进行蓝牙扫描需要获取定位权限，详细说明查看关于 定位服务开关不是强制性的，但是某些手机不打开这个开关，也无法扫描到设备，跟各家的手机系统相关
>
> iOS13 系统增加了蓝牙使用权限，需要检查是否有使用权限，确认已授权并且蓝牙已打开的情况下，开始扫描

扫描方法为QNBleApi.startBleDeviceDiscovery，扫描到的设备数据，会在上面设置的扫描接口中QNBleDeviceDiscoveryListener回调。

另外有关扫描的一些特性设置，可以在QNConfig进行设置，需要设置的内容已基本覆盖。

> 通常 APP 会有个专门用于测量的界面，我们一般是在界面显示之后进行蓝牙扫描，界面消失的时候停止扫描。

android 示例：

```
QNBleApi.getInstance(context).startBleDeviceDiscovery(new QNResultCallback() {
            @Override
            public void onResult(int code, String msg) {
                //该方法并不回到设备，而是表示扫描是否启动成功
                if (code != CheckStatus.OK.getCode()) {
                   ToastMaker.show(ScanActivity.this,code+":"+msg);
                }
            }
        });
```

iOS 示例:

```
//启动扫描
[[QNBleApi sharedBleApi] startBleDeviceDiscovery:^(NSError *error) {
    //该处回调表示启动扫描方法是否成功
    if (error) {
        NSLog([NSString stringWithFormat:@"启动扫描方法失败,原因: %@",error]);
    }
}];
```

三、连接

收到回调设备后，可以判断是否为需要连接的设备（这个属于 APP 的业务逻辑），如果是的话就进行连接。

连接之前，需要先设置连接状态回调和测量数据回调。

设置连接状态回调方法为：QNBleApi.setBleConnectionChangeListener，该方法同设置扫描回调方法一样，多次连接也只需要设置一次，如果确定不再需要连接的时候，则设置为 null/nil

设置蓝牙状态回调后，还需要设置测量数据的回调接口，方法为：QNBleApi.setDataListener，同样也是设置一次即可，不再使用时，请设置为 null/nil

android 示例：

```
QNBleApi.getInstance(context).setDataListener(new QNScaleDataListener() {
    @Override
    public void onGetUnsteadyWeight(QNBleDevice device, double weight) {
        //该方法是收到了不稳定的体重数据，一次测量中，该方法会回调多次，直到数据稳定，拿到完整
    }

    @Override
    public void onGetScaleData(QNBleDevice device, QNScaleData data) {
        //该方法是收到了完整的测量数据
    }

    @Override
    public void onGetStoredScale(QNBleDevice device, List<QNScaleStoreData> storedData
        //该方法是收到了秤端存储数据，存储数据的处理方式可以参考demo，也可以自己定义
    }

    //测量过程中的连接状态
    @Override
    public void onScaleStateChange(QNBleDevice device, int status) {
        setBleStatus(status);
    }
});
```

iOS 示例：

```
- (void)onGetUnsteadyWeight:(QNBleDevice *)device weight:(double)weight {
    //该方法是收到了不稳定的体重数据，一次测量中，该方法会回调多次，直到数据稳定，拿到完整数据
}

- (void)onGetScaleData:(QNBleDevice *)device data:(QNScaleData *)scaleData {
    //该方法是收到了完整的测量数据
}


- (void)onGetStoredScale:(QNBleDevice *)device data:(NSArray <QNScaleStoreData *> *)st
    //该方法是收到了秤端存储数据，存储数据的处理方式可以参考demo，也可以自己定义
}

- (void)onScaleStateChange:(QNBleDevice *)device scaleState:(QNScaleState)state {
    //测量过程中的连接状态
}

- (void)wspRegisterUserComplete:(QNBleDevice *)device user:(QNUser *)user {
    //注册用户成功的回调
}

- (void)onScaleEventChange:(QNBleDevice *)device scaleEvent:(QNScaleEvent)scaleEvent {
    //设备事务的回调
}
```

另外，调用连接前，最好把之前的扫描给停止（我们发现部分手机同时蓝牙扫描和蓝牙连接，会降低连接成功的失败率）。停止扫描后，延迟个 200~500ms 再调用连接，会提升连接的成功率。 停止蓝牙扫描的方法为：QNBleApi.stopBleDeviceDiscovery。

android 示例：

```
QNBleApi.getInstance(context).stopBleDeviceDiscovery(new QNResultCallback() {
    @Override
    public void onResult(int code, String msg) {
        if (code == CheckStatus.OK.getCode()) {
            isScanning = false;
        }
    }
});
```

iOS 示例：

```
[[QNBleApi sharedBleApi] stopBleDeviceDiscovery:^(NSError *error) {
    //该处回调表示停止扫描方法是否成功
    if (error) {
        NSLog([NSString stringWithFormat:@"停止扫描方法失败,原因: %@",error]);
    }
}];
```

连接设备的方法需要传Yolanda蓝牙设备对象(QNBleDevice)以及Yolanda用户资料对象（QNUser），其中QNBleDeviceQNUser的创建方法为：QNBleApi.buildUser，具体使用方法可以参考方法说明。

前面的操作确认 OK 后，终于可以进行连接了，连接的方法为：QNBleApi.connectWspDevice。连接状态回调方法为上述QNBleConnectionChangeListener

android 示例：

```
QNWspConfig qnWspConfig = new QNWspConfig();
    //.......进行配置设置
    mQNBleApi.connectWspDevice(device, qnWspConfig, new QNResultCallback() {
        @Override
        public void onResult(int code, String msg) {
            QNLogUtils.log("WspScaleActivity", "wifi 配置code:" + code + ",msg:" + msg
        }
    });
```

iOS 示例：

```
    //设置wsp设备的操作配置
    QNWspConfig *config = [[QNWspConfig alloc] init];
    [[QNBleApi sharedBleApi] connectWspDevice:device config:config callback:^(NSError

    }];
```

称重过程

称重过程的数据以及状态会在上述提到的QNScaleDataListener中进行回调

测量结束

收到稳定数据后（即收到QNScaleDataListener.onGetScaleData）即表示测量完成，当测量完成后，设备会自动断开连接。

至此，设备的基本流程已经走完，APP 可以在收到稳定数据后自行保存数据和展示数据。数据标准判断可以我司的方式-SDK 指标标准描述。

QNBleApi

入口类

方法

initSdk

根据 appid 初始化 SDK，校验 appid 与配置文件。该方法也会加载配置文件中的相关配置。

该方法只需调用一次即可，不要多次调用。建议安卓放在 Application.onCreate 中调用，iOS 放在 AppDelegate 下 didFinishLaunchingWithOptions 这个方法调用。这个方法只执行了轻量级操作，耗时操作都是异步执行，并不会影响 APP 的启动速度。

通常情况下，该方法会发送一个请求到我司服务器，用以更新配置文件中的配置，比如更新算法，更新指标，增加新型号等。这样可以达到在线升级配置的效果。如果明确不希望有此特性，可以跟我司商务沟通关闭该功能（即 SDK 改为离线模式）。另外，如果没有网络，该请求失败后，也不会有影响，SDK 会继续沿用上一次的配置。

> 注意： demo 中的 `123456789` appid 为测试 id, 测试 id 具有不稳定性与随机变化性。接入商与Yolanda达成协议后可以获取独立稳定的 appid。切记测试 Id 不可用于上线应用市场

参数

名称	类型	说明
appId	String	公司提供给客户的 app_id，通过本地初始化数据包进行校验，安卓 iOS 共用同一个 appid
firstDataFile	String	配置文件的文件路径，可以传 uri 的形式，也可以传该文件的全路径，该文件包含了
callback	QNResultCallback	安卓是接口，IOS 用 block 或闭包）回调方法，返回初始化的结果，错误码参考附表。

setSysBleStateListener

设置系统蓝牙状态回调

参数

名称	类型	说明
listener	QNBleStateListener	监听器，包含一个方法,监听系统蓝牙状态。

setBleDeviceDiscoveryListener

设置扫描回调对象，有扫描结果时，会在设置的监听方法中返回，SDK 只支持一个回调对象，重复设置会覆盖之前的结果。不需要使用时，需要重新设置为 null，否则可能出现内存泄露。

参数

名称	类型	说明
listener	QNBleDeviceDiscoveryListener	监听器，包含一个方法,扫描到设备时会在这个对象中进行回调。

getCurSystemBleState

获取当前系统蓝牙状态

返回值

类型：int

当前系统蓝牙状态

startBleDeviceDiscovery

开始扫描蓝牙设备，只返Yolanda的设备。遇到错误或蓝牙关闭则自动停止。

扫描结果在 QNBleDeviceDiscoveryListener 中回调

扫描的一些配置行为请参考getConfig 和 QNConfig

参数

名称	类型	说明
callback	QNResultCallback	回调对象，返回此次调用启动扫描是否成功

stopBleDeviceDiscovery

停止扫描蓝牙设备，该方法调用系统的停止扫描方法，如果未启动扫描，也需要调用下停止扫描，保证调用该方法后，不会再有扫描对象回调。

参数

名称	类型	说明
callback	QNResultCallback	回调对象，返回此次调用停止扫描是否成功

connectWspDevice

连接Yolanda Wsp 设备。连接的过程，会在 QNBleConnectionChangeListener 中回调，测量数据会在 QNScaleDataListener 进行回调

参数

名称	类型	说明
device	QNBleDevice	需要连接的蓝牙设备。
config	QNWspConfig	连接 wsp 设备时的配置项
callback	QNResultCallback	返回连接操作是否调用成功（并非是连接成功）

disconnectDevice

断开已连接的Yolanda蓝牙设备。断开连接的过程，会在 QNBleConnectionChangeListener 中回调

参数

名称	类型	说明
device/mac	QNBleDevice/String	需要断开连接的蓝牙设备或 mac 地址，这两个参数只要穿一个即可
callback	QNResultCallback	返回连接操作是否调用成功（并非是连接成功）

setDataListener

设置数据传输监听器

参数

名称	类型	说明
listener	QNScaleDataListener 或 QNWspScaleDataListener	测量数据监听接口，所有数据都会在里面回调

setLogListener

设置日志信息的监听

参数

名称	类型	说明
listener	QNLogListener	日志信息的监听

setBleOTAListener

设置升级固件的监听

参数

名称	类型	说明
listener	QNBleOTAListener	升级固件的监听

getConfig

获取 SDK 的当前设置情况

返回值

类型: QNConfig

SDK 设置对象，其中的设置方式，也是通过 QNConfig 完成

convertWeightWithTargetUnit

根据提供的 kg 数值的体重，转化为指定单位的数值

> 不支持 st，如果传 st 会直接返回 lb 的数值，即传 lb 或 st 返回一样的值，APP 需要直接转化英石的单位，1st=14lb，比如 145.2lb 会显示成 10 st 5.2 lb.

参数

名称	类型	说明
kgWeight	double	千克数值的体重
unit	int	0 为 kg，默认值 1 为 lb,磅，所有秤都能够支持这个单位 2 为 斤，秤端如果不支持，则会显示 kg

返回值

类型：double

返回指定单位的数值

generateScaleData

生成测量数据，如果参数异常，该方法会返回 NUL

该方法只支持 wsp 设备使用

参数

名称	类型	说明
user	QNUser	该条数据的所属用户
modeId	String	型号标识
weight	Double	体重，单位为 kg
resistance	int	50 阻抗值
secResistance	int	500 阻抗值
heartRate	int	心率值，若无则赋值 0
measureTime	Date	测量时间
hmac	String	相关数据签名

返回值

类型：QNScaleData

错误时，返回 NULL

restoreFactorySettingsCallback

WSP设备恢复出厂设置

参数

名称	类型	说明
callback	QNResultCallback	返回恢复出厂设置操作是否调用成功

QNResultCallback

统一的方法调用回调方法

onResult

参数

名称	类型	说明
errorCode	int	错误码 为0 则调用成功，其它则是错误，具体参考附表
errorMsg	string	错误信息，成功时，返回 ok

QNBleDevice

Yolanda蓝牙设备对象，目前只是指体脂称，后续可能会延续到手环

属性

名称	类型	说明
mac	String	MAC地址，sdk会根据扫描到的广播信息解析，同一台设备，安卓和IOS返回的结果一样。每台设备唯一
name	String	型号名称，SDK会识别扫描到的设备型号，如果识别不出来，则返回一个默认的型号名称，Scale
modeId	String	型号标识
bluetoothName	String	蓝牙名称，这个是硬件设备广播名称，不同于name。APP以name为准。
RSSI	int	信号强度，是一个负数，一般在-30~-100 之间，数值越大，表示信号强度越大，同一个设备可能回调多次，每次回调时，可能信号强度的值不一样。
isScreenOn	Boolean	是否已开机，秤的屏幕亮着，则是开机，否则未开机。
isSupportWifi	Boolean	是否支持Wifi功能，如果支持，则可以调用配网等操作
maxUserNum	int	(WSP设备专属)wsp秤最大支持注册用户数
registeredUserNum	int	(WSP设备专属)wsp秤已注册用户数
isSupportEightElectrodes	boolean	(WSP设备专属)是否支持八电极设备

QNUser

Yolanda 用户对象

属性

名称	类型	说明
userId	String	用户唯一标识，不能为空，每个用户唯一，与离线数据相关。如果秤端支持用户名显示，该字段会下发秤端显示
height	int	身高，单位 cm，最低 40，最高 240
gender	String	性别，男: male，女: female
birthday	Date	生日，用以计算年龄，日期精度精确到天，计算的年龄范围是 3~80，超过这个范围的，等于上限或下限。
athleteType	int	是否为运动员模式（0 为普通算法,1 为运动员算法），该字段只在年龄大于或等于 18 岁时才生效
clothesWeight	double	单位: KG 衣物重量，设置该值时，测量体重值返回的将是减去衣物重量的值。 衣物重量不可超过体重的一半,否则将会忽略超过的值
index	int	WSP 设备专用，秤端该用户索引，该值由向秤注册用户成功时，秤端返回。需要保存到服务器
secret	int	WSP 设备专用，秤端该用户秘钥, 注册 WSP 用户时，一般由服务器生成下发，取值范围 (0,9999)，后续需要保存在服务器 ，在访问用户时使用。
measureFat	int	WSP 设备专用, 控制秤端是否进行测量脂肪
indicateDis	int	WSP 设备专用, 控制秤端是否进行显示测量后除体重与BMI以外的指标

QNBleStateListener

系统蓝牙状态监听

状态类型

名称	值	说明
unknown	0	未知状态，初始化系统蓝牙时，该状态会首次回调
resetting	1	系统蓝牙正在重置
unsupported	2	系统不支持蓝牙功能
unauthorized	3	未授权
poweredOff	4	系统蓝牙关闭
poweredOn	5	系统蓝牙打开

方法

onBleSystemState

系统蓝牙状态变化的回调

参数

名称	类型	说明
state	int	系统状态

QNScaleDataListener

数据监听器，测量的相关数据在这里回调

方法

onGetUnsteadyWeight

收到了秤上传的实时重量，可以用来同步显示秤端的数值

参数

名称	类型	说明
device	QNBleDevice	上传实时数据的设备。
weight	double	连接 APP 进行测量时，秤上面的数值变化，APP 会有同步数据回调，只做 UI 的展示，不作为最后数据保存。

onGetScaleData

获取到了实时的稳定测量数据，在连接 APP 的情况下，进行测量，数据会进入到这个回调

参数

名称	类型	说明
device	QNBleDevice	上传数据的设备。
data	QNScaleData	Yolanda测量数据，包含了体重，BMI、体脂率等数据。

onGetStoredScale

获取到了存储数据，这种数据是用户在测量的时候没有连接 APP，数据会存储到秤端，然后等下次连接的时候，秤会把存储的数据上传到 APP 中，然后就会进入到这个回调。

参数

名称	类型	说明
device	QNBleDevice	上传数据的设备。
storedDataList	List<QNScaleStoreData>	存储数据列表，旧款秤顶多存储 20 条数据，新款则可以存储 40 条数据。QNScaleStoreData 传入用户资料(即 QNUser) 可以生成 QNScaleData

onGetElectric

获取电量

目前仅支持充电款的秤。当电量少于 20%时，可认为电量较低

名称	类型	说明
device	QNBleDevice	上传数据的设备。
electric	int	电量 单位 %

onScaleStateChange

测量状态变化后的方法回调，该方法不是蓝牙连接状态的回调

参数

名称	类型	说明
device	QNBleDevice	蓝牙设备对象
scaleState	int	参考秤状态码

onScaleEventChange

秤事件的回调

参数

名称	类型	说明
device	QNBleDevice	蓝牙设备对象
scaleEvent	int	参考秤事件

QNLogListener

日志信息的监听

方法

onLog

日志信息输出

参数

名称	类型	说明
log	String	日志信息

QNWspScaleDataListener

wsp 设备数据监听器，该设备相关测量数据在这里回调，该 listener 继承于 QNScaleDataListener

方法

wspRegisterUserComplete

wsp 设备注册用户成功，注册用户成功后设备会返回该用户在秤端的序列号

参数

名称	类型	说明
device	QNBleDevice	蓝牙设备对象
user	QNUser	用户对象

QNConfig

SDK配置对象，由它来控制扫描的行为，通过它设置SDK的一些行为

属性

名称	类型	说明
onlyScreenOn	Boolean	是否只返回已开机（亮屏）的设备，默认为false，即返回开机和不开机的秤。
allowDuplicates	Boolean	同一个设备是否返回多次，一般来说，设备会一直发射蓝牙广播，扫描时系统会对同一个设备回调多次，SDK会对同一个设备做处理，让一次扫描过程中，一个设备只返回一次，商家可以设置这个参数，SDK则直接每次都进行返回。默认false（该字段对onBroadcastDeviceDiscover监听无效）
duration	int	扫描持续时间,单位 ms，默认为0，即一直进行扫描，除非APP调用了stopBleDeviceDiscovery.不为0时，最小值为 3000ms 则会延时duration ms的时间后，自动停止扫描。（若有广播秤的接入需要注意该字段的设置,见广播秤工作原理）
connectOutTime	long	(安卓专属)连接设备限定时间,单位ms，默认为10000，即调用连接开始10秒之内，如果无法完成整个连接过程，就会停止连接并回调错误;设置的值小于等于0时，表示不进行回调;设置的值小于3000且大于0时，表示超时时间为3000ms.如果在连接超时之前，已经停止了连接，即使期间没有成功连接设备也不会有错误回调.
unit	int	0 为 kg，默认值 1 为 lb,磅，所有秤都能够支持这个单位 2 为 斤，秤端如果不支持，则会显示kg 3 为 st+lb，英石，秤端如果不支持，则会显示lb，的数值 4 为st, 秤端如果不支持，则会显示lb的数值
showPowerAlertKey	bool	默认为false,为true时，当SDK初始化，并且蓝牙处于关闭状态，系统会弹框提示蓝牙状态，详情请参考Apple Developer Documenttation => CoreBluetooth => CBCentralManager => Central Manager Initialization Options

名称	类型	说明
setNotCheckGPS	bool	(安卓专属)默认为false,为true时，SDK进行扫描时不会进行GPS权限做判断，继续执行扫描
enhanceBleBoradcast	bool	是否需要在扫描时启动强化广播秤信号(对普通秤无效，该选项只针对广播秤)

unit

端显示的单位，不设置的话，SDK默认为kg，设置后会保存本地，如果当前已经有连接的设备，会尽量实时更新秤端的单位显示。

> SDK会一直返回kg的数值。APP需要自己进行数值的转换。

方法

save

保存修改后的设置

参数

名称	类型	说明
callback	QNResultCallback	返回该设置操作是否成功

QNWiFiConfig

WiFi信息类

属性

名称	类型	说明
ssid	String	wifi名称(长度必须少于32个字节)
pwd	String	wifi密码(有密码时，密码长度必须大于或等于8个字节且小于64个字节 无密码时，可以传nil)
serveUrl	String	数据传输地址，`WSP设备设置有效`，`其他设备可设为null`，格式要求http://hostname:port/path/, 最大长度为 128 个字节

方法

checkSSIDVail

检验WiFi名称的有效性

返回值

类型：Boolean

checkPWDVail

检验WiFi密码的有效性

返回值

类型：Boolean

QNWspConfig

WSP 设备配置类，由它来控制 WSP 操作

配置说明:

- wifiConfig = null && curUser != null时，SDK仅进行注册访问等操作
- wifiConfig != null && curUser == null时，SDK仅对设备进行配网操作
- wifiConfig != null && curUser != null时，SDK会先进行配网，配网成功后进行注册访问等操作；若配网失败，则不会有注册访问等操作

属性

名称	类型	说明
wifiConfig	QNWiFiConfig	是否配网，当 value 值为 null 时，不进行配网操作；当 value 值为 QNWiFiConfig 对象时，根据对象中的值进行配网设置
deleteUsers	[int]	需要删除的用户 index 集合
curUser	QNUser	当前测量用户，当只需配网无需测量时，可设置为null，当curUser为null时、isRegist、isChagne、isVisitor设置均无效
isRegist	Boolean	是否需要注册用户，与 isChange 属性，只允许其中一个为 true
isChange	Boolean	是否需要修改用户信息，与 isRegist 属性，只允许其中一个为 true
isVisitor	Boolean	是否使用访客进行测量，使用访客进行测量时不必设置 isRegist 与 isChange, 同时QNUser无需设置 index、secret
dataUrl	String	数据传输地址，只有 wifiConfig 有值时才起作用，否则可设为 null，格式要求 `http://hostname:port/path/`，最大长度为 128 个字节
otaUrl	String	OTA 升级地址，只有 wifiConfig 有值时才起作用，否则可设为 null，格式要求 `protocol://hostname[:port]/path/` 最大长度为 128 个字节
encryption	String	通讯秘钥，必须为 16 字节，只有 wifiConfig 有值时才起作用，否则可设为 null
isReadSN	Boolean	是否读取SN码，默认不读取
longitude	String	经度，例如："+134.5"、"90.5"、"-87.8"。整数部分最多支持三个数字，对于支持天气查询的秤，需要设置经纬度，用于做天气的查询，如果为空，则不设置
latitude	String	纬度，规则同longitude
isDelayScreenOff	Boolean	是否延迟显示屏熄屏时间(大约延时60s)，默认false

名称	类型	说明
OTAConfig	QNBleOTAConfig	是否蓝牙OTA, 当 value 值为 null 时，不进行OTA ； 当 value 值为 QNOtaConfig 对象时，根据对象中的值进行OTA

QNBleConnectionChangeListener

蓝牙连接变化监听接口

onConnecting

正在进行连接设备，在调用连接设备后，会马上回调

参数

名称	类型	说明
device	QNBleDevice	状态变化的蓝牙设备对象

onConnected

设备已连接成功 会回调设备对象

onServiceSearchComplete

设备的服务搜索完成，正常情况下会在 onConnected后面调用

onDisconnecting

正在断开连接，调用断开连接时，会马上回调

onDisconnected

蓝牙断开了连接

onConnectError

出现了连接错误

参数

名称	类型	说明	
device	QNBleDevice		蓝牙设备对象
errorCode	int	错误码参考附表-错误码，IOS的第二个参数使用系统错误对象	

QNBleDeviceDiscoveryListener

扫描设备监听回调类

方法

onDeviceDiscover

参数

名称	类型	说明
device	QNBleDevice	扫描到的蓝牙设备

onBroadcastDeviceDiscover

广播秤设备相关信息的回调

SDK会返回能扫描到周围的所有广播秤的设备信息，app需要自行处理这些信息

> 同时为了兼容已接入广播秤的客户，原onDeviceDiscover方法中回调广播秤设备信息将保持不变。强烈建议有接入广播秤的客户使用该API

参数

名称	类型	说明
device	QNBleBroadcastDevice	扫描到广播秤

onKitchenDeviceDiscover

厨房秤设备相关信息的回调

SDK会返回能扫描到周围的所有厨房秤的设备信息，app需要自行处理这些信息

参数

名称	类型	说明
device	QNBleKitchenDevice	扫描到厨房秤

onStartScan

成功启动扫描后触发该方法

onStopScan

停止扫描后的回调

自动停止(定时)或调用stopScan停止后，都会触发该回调

onScanFail

没有扫描成功则回调(只有Android才会有这个回调)

参数

名称	类型	说明
code	int	参考扫描状态码

QNScaleData

Yolanda体脂称测量数据，包含了体重，BMI、体脂率等数据。

属性

名称	类型	说明
user	QNUser	测量的用户，包含 userId,性别，生日，身高等。
measureTime	Date	测量的时间，精确到秒
hmac	String	数据的特征标识
height	Double	身高
heightMode	int	身高模式 （1 为体脂模式，0 为体重模式）

方法

getItem

获取单个的指标，如果没有指定的 type，则返回 NULL

参数

名称	类型	说明
type	int	指标类型，参考附表

返回值

类型：QNScaleItemData

单个的指标对象，包含这个指标的值，英文名称，是否达标等。

getItemValue

获取单个指标的值，只会返回这个指标的值

参数

名称	类型	说明
type	int	指标类型，参考附表

返回值

类型：double

返回浮点数类型，有些指标的值实际上是 int 类型，APP 需要自己转化。

getAllItem

以列表的形式返回该条测量所有的指标，如果该测量数据无效，则除了体重和 BMI 的指标外，其它的指标数据的值都为 0

返回值

类型：List<QNScaleItemData>

指标列表，顺序会按照 TYPE 排序，APP 如果需要重新定义顺序，则需要自己排序。

setFatThreshold

体脂变化控制

用于控制体脂的变化，缓解体脂相差过大的问题，wsp 倘若调用此方法进行体脂变化控制后有可能产生变化控制后的指标与秤端显示的数据不一致

> 使用注意事项:
> 1.该方法只有在身体资料以及体重没有发生太大的变化才能使用；
> 2.接入方必须保存数据的特征标识 `hmac` ,以便进行体脂变化控制；
> 3.必须在调用该方法进行控制后再获取相关的指标数值，否则会出现数据错乱

参数

名称	类型	说明
hmac	String	上次有效测量数据(即测到体脂测量数据)的数据特征标识,不可为 nil
threshold	Double	体脂的最大误差控制，该值只是控制大概范围，不会精确控制。 比如值为 2,那么当此次测量体脂与上次相差大于 2 时，则该次测量的体脂会控制在与上次测量的体脂相差 2 左右
callback	QNResultCallback	是否设置成功

QNScaleStoreData

Yolanda存储数据对象，测量时没有连接 APP，结束后再连接 APP，则会接收到此类数据

属性

名称	类型	说明
weight	double	该存储数据对应的体重
measureTime	Date	测量时间
mac	String	mac 地址
isDataComplete	Boolean	存储数据是否完整，当数据完整时无需调用 setUser 方法，当数据未完整时，需使用 setUser 方法设置用户, 无论数据是否完成最终都需使用 generateScaleData 方法生成数据
height	Double	身高数据

方法

setUser

设置测量用户

名称	类型	说明
user	QNUser	用户资料，用来生产测量数据

返回值

类型：Boolean 是否设置成功

generateScaleData

根据存储数据和用户资料生成测量数据，如果未设置用户资料，该方法会返回 NULL

返回值

类型：QNScaleData

错误时，返回 NULL

buildStoreData

用于构建存储数据

> 仅支持从Yolanda云获取相关存储数据相关信息从而进行构建

参数

名称	类型	说明
weight	double	千克数值的体重
measureTime	Date	测量时间
mac	String	秤端唯一标识
hmac	String	相关数据信息，使用带有 Wifi 功能蓝牙秤中的询问用户资料接口中的签名信息
callback	QNResultCallback	是否构建成功

返回值

类型：QNScaleStoreData

> 该方法是专给带有 Wifi 功能的蓝牙秤设计，普通蓝牙秤和广播秤不会用到该方法

QNScaleItemData

Yolanda体脂称单个指标对象，包含了该指标生成报告的基本数据。

属性

名称	类型	说明
type	int	该指标的类型，具体参考附表
value	double	该指标的值，为了通用，该字段会直接返回double，如果该项值是int类型，app需要自己判断类型然后转化
valueType	int	指标值的类型，0 为 double，1 为 int
name	String	该指标的名称，全部为英文名称，提供附表转化中文名称

QNBleOTAListener

称端固件升级监听接口

onOTAStart

开始升级固件

参数

名称	类型	说明
device	QNBleDevice	状态变化的蓝牙设备对象

onOTAUpgrading

固件升级中

参数

名称	类型	说明
device	QNBleDevice	状态变化的蓝牙设备对象

onOTACompleted

固件升级成功

参数

名称	类型	说明
device	QNBleDevice	状态变化的蓝牙设备对象

onOTAFailed

固件升级失败

参数

名称	类型	说明
device	QNBleDevice	状态变化的蓝牙设备对象
errorCode	int	错误码参考附表-错误码，IOS的第二个参数使用系统错误对象

onOTAProgress

出现了连接错误

参数

名称	类型	说明
device	QNBleDevice	蓝牙设备对象
progress	double	当期升级进度(0.0 ~ 1.0)

QNBleOTAConfig

蓝牙固件升级配置项

属性

名称	类型	说明
OTAData	[Data]	OTA数据包
OTAVer	[int]	OTA固件版本

常见问题

APP逻辑设计相关

是否需要绑定或配对设备?

这里需要弄清一个概念，问题中提到的绑定或配对，是指在安卓/iOS中添加配对的蓝牙设备。通常一些标准设备是会需要的，比如蓝牙耳机，手写笔，手环。这些会需要在手机系统的蓝牙设置进行设置。但我们的秤不用，具体我们会在下问说明

蓝牙4.0，也就是BLE，是不需要绑定蓝牙设备就可以直接使用。我们的秤使用的私有协议，无法通过手机系统蓝牙直接使用，市面上面几乎所有的体脂秤都是如此。我们APP（轻牛健康）中的添加设备，仅仅让APP记住用户的秤，并非是手机系统的蓝牙配对。添加设备的过程也是引导用户如何使用体脂称。我们小程序是做成简单粗暴的方式，不用添加秤直接打开后，就上秤连接测量。

客户这边可自行决定是否需要添加该逻辑，建议如果是做成APP的一个小功能模块，是可以直接做成类似我们小程序那样，直接上秤测量，展示数据。

如何做成用户踩秤后，APP马上响应

我们自己的APP（轻牛健康），在正常使用中，是让用户打开APP，踩亮秤，就马上测量，这样可以减少用户的操作步骤。其实说白了，就是打开APP后，APP自动调用SDK的一整套逻辑，无需用户点击选择设备之类的。具体实现步骤如下：

1. 进入测量界面后，检查蓝牙状态
2. 蓝牙已打开，开始扫描
3. 扫描到体脂秤，判断是否为已添加设备或者无需判断就直接连接（该处视APP的逻辑不同而不同）
4. 连接成功，展示测量动画
5. 测量完成，展示测量结果

测量完成后，是否需要立马断开蓝牙连接

通常无需主动断开与秤的连接，除非用户退出测量界面。

如果需要测量立马断开连接，最好也能够延时2s断开。我们早期的一些成功，如果测量完成立马断开，秤可能出现意想不到的现象。

数据如何分析

一个数据是否标准、偏低等，这个可以参考我们的SDK指标标准描述。

我们Demo中也有展示如何对数据进行分析，只不过Demo的界面只展示评级登记，未像我们APP那样做成足够美观的报告。

APPID和配置文件相关

appid与配置文件是什么？分别有什么作用?

appid是Yolanda为客户创建的唯一标识，会在我们后台进行登记，一个给定appid，通常是一直不会变化。demo中的 `123456789` 这个appid是我们测试以及让客户体验用的appid，该appid的特性不稳定，随时可能会进行修改。如果已经达成正式合作意向，建议还是向我们都是商务或销售申请appid，申请过程很快，一般几分钟就可以完成。

配置文件是配合appid使用的一些加密数据，约定了型号、算法、指标等一些信息。不同的客户要求会所有不同，所以我们把客户定制化要求封装在该配置文件中。

不同客户端是否可以使用同一个APPID

安卓和iOS是可以用一个appid，小程序/小程序插件 跟该SDK的appid暂时不能使用同一个

SDK是否为离线的？它会向Yolanda云端发送什么数据?

在方法initSdk中，SDK会发送一个请求跟我们的云端校验，配置文件是否需要更新。如果需要更新，会下载最新的配置文件(配置文件很小，一般为512字节)，如果不需要更新，则不做任何处理。

SDK是支持离线模式的，客户如果特别要求，我们会把配置文件中关于该项的配置设置成 `离线` 。设置成离线后，SDK不再向外发送任何网络请求（这个可以用抓包工具验证）配置文件不再更新，如果需要增加型号或指标，则需要跟我司再次申请配置文件，然后由开发人员手动替换。

初始化提示appid错误

- 检查初始化文件和使用的appid是否匹配
- 检查引入的SDK是否是最新的

测量完成后返回的数据缺失指标

如果是只有体重和BMI的数据可以参考测量完成后没有体脂率等数据.

如果能够测到体脂率，但是缺失自己想要的指标，该情况通常是配置文件中未包含该指标，需要跟我司商务/销售沟通，增加该指标。

我司增加指标后，会重新发送一个配置文件，需要替换该配置文件。

把扫描的设备的model id发给我司进行确认

SDK功能相关

是否能判断秤是否灭屏?

对于 `普通蓝牙秤` 和 `双模秤` 来说，未连接设备时，扫描拿到的设备对象QNBleDevice，有个属性 `isScreenOn` ,该属性指示了屏幕是否亮屏。连接设备后，无法直接判断设备是否亮屏，不过对于一些秤来说，息屏的时候就会断开连接。

> isScreenOn 属性只是表明SDK扫描到那一刻该设备是亮屏的，后续设备屏幕状态变化了，QNBleDevice不会动态更新。需要通过扫描到的新的QNBleDevice回调才能判断。

对于 `广播秤` 来说，能够扫描到设备，就是已开机的状态，扫描不到，通常就是未开机。

连接设备一直无法成功或者成功后很快就断开连接

- 检查设备是否被其他人连接了
- 在系统蓝牙中查看是否当前连接的设备已经被配对,如果已经配对，需要取消配对
- 部分手机需要先扫描才能连接成功，先扫描设备再进行连接

SDK返回无定位权限错误

- 检查是否对**ACCESS_COARSE_LOCATION**和**ACCESS_FINE_LOCATION**都进行了申请，SDK中对2个权限都进行了校验
- 是否编译版本26以及以上，如果是，2个权限都需要单独申请(8.0的新特性)

  > 安卓6.0以上，谷歌把蓝牙归类为定位功能的一部分，使用蓝牙扫描时，需要用户授权定位权限，不然调用扫描时，系统会提示需要定位权限 相当一部分手机（大约有1/3的样子，不打开定位服务开关时，也是无法扫描到设备，原生大多数如此，而国产系统会对此有些优化，就不一定了）

SDK返回错误的文件,确认文件位置无异常，确认文件在demo中使用无异常

- 检查是否有添加so库
- 检查是否打包的apk文件中含有so库

数据或者设备等监听回调，同一时间回调多次

- 先确定是否，设置了多次监听。当监听不使用时，一定要设置为null
- 确定是否是穿鞋测量，这个可能导致短时间内，完成多次测量的情况

测量完成后没有体脂率等数据

1. 检查资料是否正确，主要是生日（用来计算年龄）、身高这两个参数，是否超出正常值范围
2. 是否有脱鞋进行测量。穿鞋测量时，无法测到生物阻抗，所以无法计算体脂率等指标

扫描不到设备，该如何处理

1. 检查是否打开蓝牙、是否有踩亮秤，手机是否跟秤相距过远（超过10米）
2. 检查所扫描的设备，是否已经被其他人连接
3. 如果是Android，则可以检查下是否有定位权限，另外是否有打开定位服务开关
4. 尝试重启蓝牙，再不行，重启蓝牙试试。

是否有手环SDK?

我司有智能手环设备，也是有SDK。不过SDK暂时没有对外完全开放，如有意向，可以联系我司商务洽谈。

SDK是否可以只做协议解析，蓝牙扫描和连接由客户自己管理

这个是可以实现的，普通蓝牙秤和蓝牙广播秤的实现方式会有不同。

> 自己实现蓝牙扫描和连接的管理，有一定的技术难度，如果没有相关蓝牙开发经验，不建议进行此操作。如果执意进行该项操作，我们会为认为客户有较丰富的蓝牙开发经验。

1. 编写自己的蓝牙协议代理类，实现各个蓝牙操作方法
2. 编写数据监听回调类 QNScaleDataListener，并使用[QNBleApi.setBleDeviceDiscoveryListener注册到SDK中。
3. 自己进行蓝牙扫描，并把扫描到的相关信息传给SDK，使用方法QNBleApi.buildDevice，创建Yolanda蓝牙设备对象
4. 创建用户对象，使用方法QNBleApi.buildUser
5. 创建蓝牙协议处理类
6. 进行蓝牙连接
7. 连接成功，发现完服务成功后，调用蓝牙协议处理类的prepare方法
8. 处理QNScaleDataListener回调的数据
9. 收稳定数据后自行处理。

你们官网API的《秤状态定义》里的状态是0-9，为什么在你们提供的Demo里有为“-1”的状态(而且确确实实会返回这个状态)

-1一直存在，表示默认状态失去连接，也就是连接是断开的状态。通常是秤主动断开连接后收到的。我们会尽快把这个状态也统一到断开连接状态。

其他问题

是否可以给到体脂率等指标的计算方法?

这些指标的算法属于我司机密，无法提供。

什么是秤端存储数据

普通蓝牙秤在称重时未连接手机的情况下，秤会把数据存在秤体内，等下次连接蓝牙时，会把这些数据传给APP，由于SDK无法识别该用户是谁的，所以会把这个数据交由APP来分配，分配用户可以交给SDK来计算完整数据。

可否提供蓝牙协议

原则上我们不会提供蓝牙协议，如果实在需要，可以跟我们的商务/销售沟通

为什么我们使用SDK测量出来的数据与轻牛或轻牛健康的数据会有差异

Yolanda有几套算法，不同的设备可能使用的不同算法。

另外，我们自己APP中有一个健康问题机制，会让用户回答2个问题，并且根据用户回答的情况，而采用不同的算法。

SDK中的算法机制通常为统一的某种算法，不可变的。也就是说，同一个秤使用我司的APP测量与使用SDK进行测量，计算资料和体重一样，也有可能有会不少的差异。所以，我司不建议使用我司APP和SDK的测量数据进行对比，客户只需要关心给到的算法是否合适，是否稳定即可。

线上版本出现蓝牙连接异常后，怎么排查问题

先查清用户的手机型号，如果可以的话，尽量使用跟客户同样的型号来测试。

注册日志输出接口QNLogListener，记录相关日志（可以保存日志文件），发给我司开发人员进行分析。

搜索到蓝牙并连接上后，脚悬空停留在体脂秤上面几秒，然后把脚拿开，会返回测量完成的状态码(返回数据的方法内并没有返回任何数据)

为什么我们在没有任何操作的情况下也能搜索到蓝牙（10-30分钟内没有人靠近体脂秤）

称重的时候，如果未连接APP，秤会把这个数据缓存到里面，息屏后也会广播5-30分钟（视生产批次不同而不尽相同）。此时APP连接后，可以把这些缓存的数据接收到，并显示给客户。该功能用来解决，用户称重时不用手机，称重结束后，坐在沙发上打开手机并接收刚才的数据。

身体指标常量

常量	值	含义
TYPE_WEIGHT	1	weight
TYPE_BMI	2	BMI
TYPE_BODYFAT	3	body fat rate
TYPE_SUBFAT	4	subcutaneous fat
TYPE_VISFAT	5	visceral fat
TYPE_WATER	6	body water rate
TYPE_MUSCLE	7	muscle rate
TYPE_BONE	8	bone mass
TYPE_BMR	9	BMR
TYPE_BODY_SHAPE	10	body type
TYPE_PROTEIN	11	protein

常量	值	含义
TYPE_LBM	12	lean body weight
TYPE_MUSCLE_MASS	13	muscle mass
TYPE_BODY_AGE	14	metabolic age
TYPE_SCORE	15	health score
TYPE_HEART_RATE	16	heart rate
TYPE_HEART_INDEX	17	heart index
TYPE_FAT_MASS_INDEX	21	fat mass
TYPE_OBESITY_DEGREE_INDEX	22	obesity degree
TYPE_WATER_CONTENT_INDEX	23	water content
TYPE_PROTEIN_MASS_INDEX	24	protein mass
TYPE_MINERAL_SALT_INDEX	25	mineral salt

常量	值	含义
TYPE_BEST_VISUAL_WEIGHT_INDEX	26	best visual weight
TYPE_STAND_WEIGHT_INDEX	27	stand weight
TYPE_WEIGHT_CONTROL_INDEX	28	weight control
TYPE_FAT_CONTROL_INDEX	29	fat control
TYPE_MUSCLE_CONTROL_INDEX	30	muscle control
TYPE_MUSCLE_MASS_RATE	31	muscle mass rate
TYPE_FATTY_LIVER_RISK	32	heart index
TYPE_RESISTANCE_50KHZ	33	resistance 50khz
TYPE_RESISTANCE_500KHZ	34	resistance 500khz

常量	值	含义
TYPE_RIGHT_ARM_MUSCLE_WEIGHT_INDEX	101	right upper limb muscle weight
TYPE_LEFT_ARM_MUSCLE_WEIGHT_INDEX	102	left upper limb muscle weight
TYPE_TRUNK_MUSCLE_WEIGHT_INDEX	103	trunk muscle weight

常量	值	含义
TYPE_RIGHT_LEG_MUSCLE_WEIGHT_INDEX	104	lower right muscle weight
TYPE_LEFT_LEG_MUSCLE_WEIGHT_INDEX	105	lower left muscle weight
TYPE_RIGHT_ARM_FAT_INDEX	106	right upper limb fat
TYPE_LEFT_ARM_FAT_INDEX	107	left upper limb fat

常量	值	含义
TYPE_TRUNK_FAT_INDEX	108	trunk fat
TYPE_RIGHT_LEG_FAT_INDEX	109	right leg fat
TYPE_LEFT_LEG_FAT_INDEX	110	left leg fat

设备类型对照表

常量	值	含义	蓝牙名	备注
SCALE_BLE_DEFAULT	100	普通蓝牙秤	QN-Scale或QN-Scale1	可以连接测量，使用过程包括扫描、连接、数据通讯
SCALE_BROADCAST	120	广播秤	QN-S3	不可以进行连接，它的数据通过蓝牙广播数据传输
SCALE_KITCHEN	130	厨房秤	无	不可以进行连接，它的数据通过蓝牙广播数据传输
SCALE_WSP	140	WSP蓝牙秤	QN-Scale	可以连接测量，使用过程包括扫描、连接、数据通讯
HEIGHT_SCALE	160	身高秤	QN-HS	可以连接测量，具有身高数据，使用过程包括扫描、连接、数据通讯

体型对照表

值	含义
0	无体型值(一般是没有测量到体脂率)
1	隐形肥胖型
2	运动不足型
3	偏瘦型
4	标准型
5	偏瘦肌肉型
6	肥胖型
7	偏胖型
8	标准肌肉型
9	非常肌肉型

错误码

操作正常

错误常量	错误码	含义	解释	解决方案
OK	0	success	操作成功	无

初始化相关错误

错误常量	错误码	含义	解释	解决
ERROR_INVALIDATE_APP_ID	1001	app id is invalidate	APPID 无效	使用
ERROR_NOT_INIT_SDK	1002	please call the method "initSdk" first	未调用 initSDK 方法	先调
ERROR_FIRST_DATA_FILE_URI	1003	the first data file uri is error, please provide the correct one	初始数据文件的 Uri 有误	提供
ERROR_PACKAGE_NAME	1004	the Android Package Name is Error ,Please Check it Or Contact the SDK Provider	安卓的 包名 不正确	检查
ERROR_BUNDLE_ID	1005	the IOS APP Bundle Id is Error ,Please Check it Or Contact the SDK Provider	IOS 的 `Bundle Id` 不正确	检查
ERROR_INIT_FILE	1006	The config file content is wrong ,Please provide the correct one	初始配置文件有误	让

蓝牙相关错误

错误常量	错误码	含义
ERROR_BLUETOOTH_CLOSED	1101	the bluetooth is closed,please enable it first
ERROR_LOCATION_PERMISSION	1102	your app need the android authorize the location permission
ERROR_BLE_ERROR	1103	bluetooth internal error occurred
ERROR_CONNECT_WHEN_CONNECTING	1104	bluetooth is doing connect, you should not call connect right now
ERROR_CONNECT_WHEN_HAS_CONNECTED	1105	bluetooth is connected another scale, please disconnect it first
ERROR_BLUETOOTH_UNKNOWN	1106	the bluetooth state is unknown, this state may appear when the bluetooth is not prepared on the IOS

错误常量	错误码	含义
ERROR_BLUETOOTH_RESETTING	1107	the system is resetting the bluetooth, try again later
ERROR_BLUETOOTH_UNSUPPORTED	1108	the phone/pad is not support BLE, try another phone/pad
ERROR_BLUETOOTH_UNAUTHORIZED	1109	bluetooth is unauthorized, ask user authorize
ERROR_BLE_CONNECT_FAIL	1110	failed to connect scale, try reconnect
ERROR_BLE_DISCONNECTING	1111	it is disconnecting th scale, try again later
ERROR_BLE_NONE_SCAN	1112	No bluetooth device has been scanned
ERROR_BLE_CONNECT_OVERTIME	1113	Connect device timeout

方法调用错误

错误常量	错误码	含义
ERROR_ILLEGAL_ARGUMENT	1201	illegal argument,please check the api document
ERROR_MISS_DISCOVERY_LISTENER	1202	miss the discovery listener, please set the listener first
ERROR_MISS_DATA_LISTENER	1203	miss the data listener, please set the listener first
ERROR_USER_ID_EMPTY	1204	the user id argument is null or empty
ERROR_USER_GENDER	1205	the gender argument is wrong, please pass the "male" or "female"
ERROR_USER_HEIGHT	1206	the height argument is wrong, please pass the value within 40 and 240 cm
ERROR_USER_BIRTHDAY	1207	the birthday argument is wrong, please pass the date before today
ERROR_START_SERVICE_BACKGROUND	1208	after **Android O**,not allowed to start service in background
ERROR_USER_ATHLETE_TYPE	1209	the athleteType argument is wrong, only allow 0 or 1
ERROR_USER_SHAPE_GOAL_TYPE	1210	the shape or goal argument is wrong
ERROR_DEVICE_TYPE	1211	the device type is wrong

错误常量	错误码	含义
ERROR_WIFI_PARAMS	1212	the WiFi params is wrong
ERROR_REGISTER_DEVICE	1213	register device is fail
ERROR_NOCOMPLETE_MEASURE	1214	you can get data when measurement complete
ERROR_NOSUPPORT_MODIFY	1215	not supported modifying units
ERROR_WSP_USER_INDEX	1216	the user's index is wrong
ERROR_WSP_USER_SECRET	1217	the user's secret index

共享秤相关错误

错误常量	错误码	含义	解释	解决方案
ERROR_CODER	1301	coder is error	二维码错误	请联系客服处理
ERROR_CODER_INVALID	1302	coder invalid	二维码失效	请重新生成

扫描状态定义

状态	值	含义	解决方案
FAIL_BLE_IS_OFF	1	设备蓝牙处于关闭状态	开启蓝牙
FAIL_BLE_NOT_SUPPORT	2	设备不支持蓝牙4.0	升级手机系统版本或更换手机
FAIL_BLE_INTERNAL_ERROR	3	内部错误	重新启动扫描
FAIL_BLE_NO_BLUETOOTH	4	缺少蓝牙权限	在清单文件中申请蓝牙权限
FAIL_BLE_NO_LOCATION	5	缺少位置权限	申请位置权限
FAIL_BLE_NONE_DEVICE	6	没有扫描到任何蓝牙设备	重启手机蓝牙

秤状态定义

状态	值	含义	蓝牙属性
STATE_DISCONNECTED	0	未连接	未连接
STATE_CONNECTED	1	已连接	已连接
STATE_CONNECTING	2	正在连接	正在连接
STATE_DISCONNECTING	3	正在断开	已连接
STATE_START_MEASURE	5	正在测量	已连接
STATE_REAL_TIME	6	正在测量体重	已连接
STATE_BODYFAT	7	正在测试生物阻抗	已连接
STATE_HEART_RATE	8	正在测试心率	已连接
STATE_MEASURE_COMPLETED	9	测量完成	已连接
STATE_WIFI_BLE_START_NETWORK	10	WiFi 蓝牙双模设备开始配网	已连接
STATE_WIFI_BLE_NETWORK_SUCCESS	11	WiFi 蓝牙双模设备联网成功	已连接
STATE_WIFI_BLE_NETWORK_FAIL	12	WiFi 蓝牙双模设备联网失败	已连接

秤事件定义

事件	值	含义	蓝牙属性
EVENT_WIFI_BLE_START_NETWORK	1	WiFi蓝牙双模设备开始配网	已连接
EVENT_WIFI_BLE_NETWORK_SUCCESS	2	WiFi蓝牙双模设备联网成功	已连接
EVENT_WIFI_BLE_NETWORK_FAIL	3	WiFi蓝牙双模设备联网失败	已连接
EVENT_REGIST_USER_SUCCESS	4	WSP秤专属，注册用户成功	已连接
EVENT_REGIST_USER_FAIL	5	WSP秤专属，注册用户失败	已连接
EVENT_VISIT_USER_SUCCESS	6	WSP秤专属，访问用户成功	已连接
EVENT_VISIT_USER_FAIL	7	WSP秤专属，访问用户失败	已连接
EVENT_DELETE_USER_SUCCESS	8	WSP秤专属，删除用户成功	已连接
EVENT_DELETE_USER_FAIL	9	WSP秤专属，删除用户失败	已连接
EVENT_SYNC_USER_INFO_SUCCESS	10	WSP秤专属，同步用户信息成功	已连接
EEVENT_SYNC_USER_INFO_FAIL	11	WSP秤专属，同步用户信息失败	已连接

测试用例

Excel 版本下载地址

> 可以鼠标右键下载该文件

用例编号	用例标题	优先级	预置条件	测试步骤	检查点与注意事项	
SDK_01	验证能正常在手机上安装运行	高		1.在手机上安装应用包 2.打开该应用		
SDK_02	验证关闭蓝牙提示正常	中		1.关闭手机蓝牙 2.添加连接秤		
SDK_03	验证填写正常资料能正常测量并验证IOS和安卓测量结果数据一致	高	必须光脚测量	1.安装并打开SDK应用 2.输入个人身体资料，如身高：161cm，年龄1991-01-01，男 3.用安卓和IOS相同的资料分别上秤各测量一条完整		

用例编号	用例标题	优先级	预置条件	测试步骤	检查点与注意事项	
SDK_04	验证穿鞋测量指标显示正常	高	穿鞋测量	1.输入自己个人身体资料 2.连接称并上秤测量一条完整数据		
SDK_05	验证指标个数显示正常	高	必须光脚测量	1.编辑正常个人资料 2.连接秤测量一条完整数据		
SDK_06	验证切换不同性别测量数据正常	高		1.使用安卓手机：编辑个人资料，性别分别为男和女上秤各测量一条完整数据 2.再换IOS手机：用相同资料上秤测量男女各一条数据		
SDK_07	验证不同的身高测量的数据正常	高	建议身高输入的范围在40cm--240cm，最小为40cm，最大为240cm	1.分别输入40cm，161cm,240cm，其他身体资料不变 2.连接秤上秤分别以上面三个不同的身高各测量一条完整数据	以56kg体重为例	

用例编号	用例标题	优先级	预置条件	测试步骤	检查点与注意事项	
SDK_08	验证不同年龄测量数据正确	高	建议年龄范围4岁-80岁	安卓和ios使用相同的资料进行测量； 1.输入生日为5岁，28岁，80岁，其他资料保持不变 2.分别以不同的年龄连接秤各测量一条完整数据		
SDK_09	验证切换kg,lb,斤，st不同的单位测量显示正常	高	设置斤单位，秤不支持斤，秤默认kg,设置st单位，秤不支持，秤默认lb	1.编辑个人资料，切换不同的单位 2.以不同的单位去连接秤		
SDK_10	验证修改后资料测量数据比修改前测量数据差异大	高		1.编辑个人资料，连接秤上秤测量一条完整数据 2.修改个人资料，（修改的资料和前面有差异），再次上秤测量一条完整数据		

更新记录

2021-10-28

- 新增升级固件配置项
- 新增升级固件监听器
- 新增设置升级固件监听器方法
- 新增WSP设备恢复出厂设置方法
- QNUser新增measureFat indicateDis字段

2021-06-16

- 增加st单位支持

2021-04-12

- 增加wsp设备音量设置sound

2021-01-08

- 设备信息增加是否支持八电极字段 isSupportEightElectrodes
- 指标增加八电极相关指标字段 （八电极相关指标不做指标限制，全部输出，阻抗值输出由阻抗配置进行限制）

2021-01-05

- 增加控制wsp设备体重趋势字段
  - 增加 `weightExtend` 参数

2020-11-05

- 增加控制wsp设备延迟熄屏时间字段
  - 增加 `isDelayScreenOff` 参数

2020-10-16

- writeCharacteristicValue
  - 增加 `device` 参数
- readCharacteristicValue
  - 增加 `device` 参数

2020-08-26

- 增加wspReadSnComplete回调
- 增加是否读取sn码的控制isReadSN

2020-08-10

- 增加秤端事件EVENT_SYNC_USER_INFO_SUCCESS、EEVENT_SYNC_USER_INFO_FAIL
- 修改QNWspConfig类longitude、latitude的格式
- 增加wsp位置信息同步状态的回调wspLocationSyncStatus

2020-07-23

- WSP经纬度格式变更

- QNBleDevice对象添加wsp设备特性maxUserNum、registeredUserNum

2020-06-24

- 增加WSP扩展对象
  - 用户ID下发以及设置WSP秤端是否显示某些设置
- 增加WSP扩展字段
  - 增加经纬度

2020-06-09

- 增加设备类型 HEIGHT_SCALE 身高秤
- 增加身高数据 height
- 增加身高模式数据 heightMode
- 存储数据增加身高数据 height

2020-05-14

- 添加测试用例

2020-03-24

- 增加获取当前系统蓝牙状态的方法

2020-03-03

- 增加访客模式的说明
- 增加使用访客测量的控制 isVisitor
- 增加存储数据是否完整的标识 isDataComplete
- 增加 Wsp 设备监听接口 QNWspScaleDataListener

2020-02-07

- 增加 SCALE_WSP 设备类型
- 增加 QNUser 类 index、secret 属性
- 增加秤事件回调
- 增加 QNWspConfig 配置类
- 增加 wsp 的连接方法
- 增加系统蓝牙状态回调监听
- 增加 generateScaleData 方法
- 增加 Wsp 设备流程描述

2019-10-28

- 增加服务自定义标题和内容设置 setNotificationInfo

2019-10-08

- 增加创建厨房秤对象方法 buildKitchenDevice
- 增加 QNBleKitchenDevice 对象
- 增加 onKitchenDeviceDiscover 方法
- 修改 convertWeightWithTargetUnit 方法，增加厨房秤单位的转化
- 增加厨房秤类型

2019-07-19

- 增加创建广播秤对象方法 buildBroadcastDevice

- 增加创建协议处理类的方法 buildProtocolHandler
- 移除 Wifi 注册设备方法 `registerWiFiBleDevice`
- 增加蓝牙协议代理对象 QNBleProtocolDelegate
- 增加蓝牙协议处理对象 QNBleProtocolHandler
- 把 QNDataListener 修改为 QNScaleDataListener
- 把QNBleConnectionChangeListener中的 `onScaleStateChange` 方法发移到 QNScaleDataListener，专门用来表示称重状态的变化。
- 整理设备类型，修改为只有普通蓝牙秤和广播秤

2019-07-10

- 增加广播秤对象 QNBleBroadcastDevice
- 增加广播秤扫描监听回调方法 onBroadcastDeviceDiscover
- 修改 config 设置类 duration、allowDuplicates、unit 字段说明
- 增加广播秤获取数据的错误码 ERROR_NOCOMPLETE_MEASURE
- 增加修改广播秤单位错误码 ERROR_NOSUPPORT_MODIFY

2019-07-01

- 增加日志的监听

2019-06-03

- 增加是否检查 GPS 权限方法

2019-05-09

- 增加数据特征标识 `hmac`
- 增加体脂变化控制方法 `setFatThreshold`

2019-04-24

- 增加设备类型 `SCALE_WIFI_BLE`
- 增加 WiFi 配置类
- 增加注册 WiFi 蓝牙双模设备方法 `registerWiFiBleDevice`
- 增加配置 WiFi 的方法 `connectDeviceSetWifi`
- 增加存储数据对象中属性 `mac` 以及方法 `buildStoreData`
- 增加秤状态码 `STATE_WIFI_BLE_START_NETWORK` 、 `STATE_WIFI_BLE_NETWORK_SUCCESS` 、 `STATE_WIFI_BLE_NETWORK_FAIL`
- 增加方法调用错误码 `ERROR_DEVICE_TYPE` 、 `ERROR_WIFI_PARAMS` 、 `ERROR_MISS_STATE_LISTENER`

2019-03-22

- 增加衣服体重参数 clothesWeight

2019-03-14

- 修改 Android-demo 英文地址
- 修改 Android-demo 中文地址
- 增加 Android 常见问题
- 设备类增加设备类型参数 deviceType

2019-03-14

- 增加 deviceType 属性
- 修改有效时间规则 validTime
- 增加设备类型表单

2019-03-05

- 增加 decodeShareData 方法参数 validTime
- 增加错误码 ERROR_CODER、ERROR_CODER_INVALID

2019-02-25

- 增加指标

2019-01-27

- 增加QNUtiils类

2018-10-25

- 增加电量的回调 onGetElectric

2018-10-11

- 增加运动员模式的字段 athleteType
- 用户模型增加字段 athleteType
- 增加错误码 ERROR_USER_ATHLETE_TYPE

2018-08-27

- 增加连接超时时间设置
- 增加体型对照表

2018-08-23 增加构建 SDK 蓝牙对象方法

2018-08-13 增加扫描超时时间设置

2018-06-20 增加设备 modeID