Table of Contents

Overview

中文文档入口在这里

This document is used to guide Yolanda customers to access Yunkangbao's smart devices.

The SDK supports devices, please refer to Device Type

Docking steps

1. Business application

Sign a cooperation agreement with Yolanda business staff, our company will provide the `appid` and `configuration file` required for development

We will provide a form, as long as you fill out the form according to the requirements and send it to our staff for processing, usually the two things above will be given soon

> Never use the test appid of `123456789` in the official app, it is unstable, we may block it at any time, or change the appid for test

2. Download the research SDK

Download the SDK for the corresponding platform, run the demo using the appid and configuration file applied for above, and study how the demo and documentation use the SDK.

It is strongly recommended that you study the demo before connecting to your own APP. In addition, you can take a look at FAQ, and read the questions carefully, maybe your doubts will be solved.

> Suggest developers `star` Our SDK project, so that we can receive update notifications in time

3. Design body fat scale function

Design how to connect the body fat scale in the existing APP. After researching the SDK in the previous step, we can also design a body fat scale for this step to avoid many minefields.

The general steps of using body fat scale are:

1. Add device
   > This is not configured with Android or iOS systems, Bluetooth 4.0 can support unpaired use. Our body fat scale is a custom protocol, usually only our APP or APP connected to our SDK can be used. Cannot be added to the system, use the system Bluetooth connection.
2. Measurement

> The measurement here actually includes Bluetooth scanning and Bluetooth connection. During the measurement process, it is best to show some measurement animations to remind the user that the measurement is being performed to avoid the boring process

3. Save and display data

> After receiving the locked measurement data, the data is saved locally and displayed. In order to achieve offline measurement, you can save the data locally and then upload it to the server. If there is no network at the time, you can wait for the network to upload the data to the server. The way to display data can refer to our APP, of course, if you have your own style and analysis method, you can also use your own.

4. Develop APP function

According to the designed function, connect the SDK to the APP.

Pay special attention to the location permission and location service switch of Android. iOS requires Bluetooth permission after the main 13 version.

5. Self-test function

After the development is completed, you can use our provided test case SDK test case to verify whether the weighing function can be used.

> The above attachments can be downloaded by right-clicking

If possible, it is recommended to send the test version to Yolanda, we will arrange testers to assist customers in APP testing.

We will mention the results of the test and some of our views on the APP to our customers.

6. Submit online

After the test is completed, it will be submitted online. It's best to inform Yolanda when you go online successfully. In this way, Yolanda will register the online results and continue to follow up the experience.

SDK download link

The SDK file is no longer a separate download link. The following is the Demo provided by us. The Demo project has the latest SDK file.

iOS

Android

Integration

Android integration

iOS integration method

Development Process

Before using the SDK, you need to call initialization first, the method is: QNBleApi.initSdk

Code example:

```
//Android sample code
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", "Initialization file" + msg);
    }
});
```

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

The initialization method is the method that any device needs to call. The following is the workflow of different device types.

If you do n't know what kind of device you have, you can refer to How to determine the type of device

Android quick integration

Android Studio

Android using gradle integration

- In the root directory of your project **build.gradle** Add to **jitpack** stand by

```
  allprojects {
      repositories {
//Other warehouse configuration
          maven { url 'https://jitpack.io' }
  }
}
```

- In the root directory of your module **build.gradle** Add dependency View Address of Latest Version

```
<!--The version number here, x.x.x can be specified as any release version-->
<!--If you want to always use the latest version, you can replace x.x.x with mast
dependencies {
      ...
      compile 'com.github.YolandaQingniu:qnscalesdk:x.x.x'
}
```

Local dependency

- Download the latest jar and so libraryImport the downloaded `jar and so library`
- Apply for Bluetooth permission in the manifest file、Location permissions、Network permissions (not required for offline SDK)

```
  xml
 <!--Bluetooth permission-->
 <uses-permission android:name="android.permission.BLUETOOTH" />
 <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
 <!--6.0 and later need to apply dynamically-->
 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
 <!--Used to store logs-->
 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
 <!--If it is an online sdk, network permission is required-->
 <uses-permission android:name="android.permission.INTERNET" />
<!--qnscalesdk: Versions before 1.1.3-beta3 (including 1.1.3-beta3) need to add t
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<!--qnscalesdk: Versions after 1.1.3-beta3 (starting from 1.1.3-beta4) need to ad
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

- Need to be in **AndroidManifest.xml** Register the components in the SDK:

```
xml
<!--qnscalesdk: version configuration before 1.1.3-beta3 (including 1.1.3-beta3)-->
<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: Version configuration after 1.1.3-beta3 (starting from 1.1.3-beta4)-->
<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
```

- The resources of the v4 package are used in the SDK, and the resources of the v4 package need to be introduced in the developer project

Obfuscated configuration (proguard-rules)

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

Other

The integration method of configuration file can refer to demo. Put it in the `assets` directory and use it during initialization:" file:///android_asset/filename.qn "to pass the file path

iOS quick integration

SDK operation requires appid and configuration files. Merchants can first use the test appid and test configuration files provided by Yolanda. When officially released, they must obtain official appid and configuration files from Yolanda.

Installation method

cocoapods installation:

- Install Cocoapods first；
- Update the cocoapods version of QNSDK via pod repo update；
- In the target corresponding to Podfile, add `pod'QNSDK'`, And execute pod install；
- Use .xcworkspace generated by CocoaPods in the project to run the project；
- Introduce the header file in your code file header#import`

Carthage installation:

- Install Carthage；
- Open Cartfile, add `github" https://github.com/YolandaQingniu/sdk-ios-demo.git"`；
- Open the command line, cd to your project directory, enter carthage update；
- Drag the `QNSDK.framework` under Carthage/Build/directory to the Build Phases of your project engineering configuration->Linked Binary and Libraries；
- Introduce the header file in your code file header#import`

Manual installation:

- Download the SDK installation package to the project
- Introduce SDK path【TARGETS】-> 【Build Setting】->【Search Paths】->【LibrarySearch Paths】Add SDK path
- Configure linker【TARGETS】-> 【Build Setting】-> 【Linking】-> 【Other Linker Flags】Add `-ObjC`、`-all_load`、One of `-force_load [SDK path]`

Project configuration

- There are right in Info.plist【Privacy - Bluetooth Peripheral Usage Description】【Privacy - Bluetooth Always Usage Description】Key for Bluetooth instructions

Precautions

- SDK adapts to 8.0 and above systems

- iOS10.0 and above systems must use Bluetooth instructions in Info.plist, otherwise the Bluetooth function of the system cannot be used
- iOS13.0 and above systems must configure Bluetooth usage authorization instructions in Info.plist, otherwise it will be crashed
- The linker must be configured for the SDK, otherwise the SDK will not work properly

WSP Bluetooth WiFi scale

WSP Bluetooth WiFi scale is a smart device launched by our company that supports automatic recognition of user measurements, directly displays the measurement results on the scale display without the app, and automatically uploads the measurement results.

This device avoids the step in which users need to use Bluetooth to connect to the device in order to make detailed measurements during the measurement process, and improves the user experience. As long as the device is connected to the network and the user information to be measured is registered with the device once, the user can subsequently use the app without the app. Registered users can measure on the scale without using a mobile phone. After the measurement is completed, the scale will automatically identify which registered user the measurement data belongs to. After successful recognition, the indicators of the measurement will be displayed on the device and the measurement data Last time to the corresponding server.

Brief description of scale user

At present, the WSP device allows up to 8 users, and users on the scale end need to register users with the device through Bluetooth connection. When 8 users have been registered and the user is registered with the device again, the scale will return to the status of registration failure. Therefore, before registering users, it is recommended to determine whether you need to delete the invalid users in the scale to avoid invalid users occupying resources.

When registering a user with the device, the user key needs to be issued to the device (the key is generally generated by the server for the user's unique identifier). When the registration is successful, the device will record the user key and return the user's unique sequence number. Therefore, after successfully registering the user, the app needs to save the serial number and secret key that the device returns to the user. When updating user data via Bluetooth connection or measuring via Bluetooth connection, the user ’s serial number and secret key must be used to access the designated user on the scale side successfully. At the same time, the serial number is also used as an identification of the registered user to which the uploaded measurement data belongs, and as a necessary condition for deleting the user.

- registered user

The scale can cache the registered user information, and identify the user to which the measurement data when the Bluetooth is not connected by the registered user information, so as to obtain detailed measurement data using the identified user information

- Visit users

When operating the scale, you need to access the registered user first, and then you can enter the corresponding operation after successful access, such as modifying user information、Connect to Bluetooth for measurement, etc.

- delete users

After the user has been registered with the scale, subsequent users unbind the scale, and the user on the scale side needs to be deleted to avoid resource occupation

Visitor Mode Measurement

Guest mode is a common channel measurement method provided by the scale body. This method does not require registered users to use it directly. Guest mode uses temporary users. The existence time of this user is from connection to disconnection. Temporary user when the device is disconnected That is to destroy, while in the guest mode will not generate and receive stored data

WSP Device Protocol Brief

The protocol of the wsp device is basically optimized and modified in WSP Official Agreement. The wsp protocol interaction is more complicated. The following are the parts involved in the device protocol

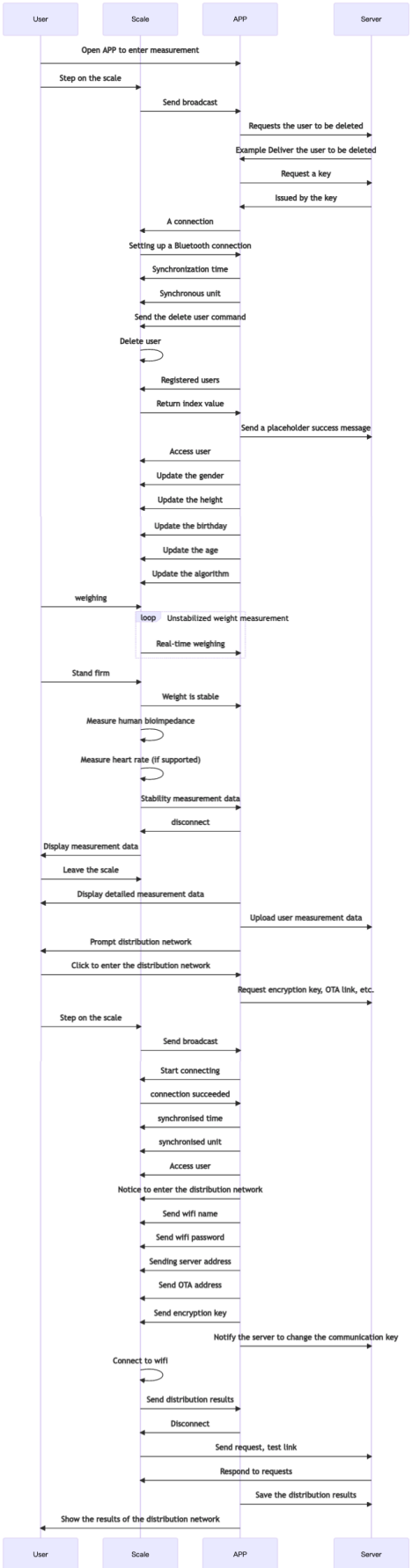
Aspects covered by the agreement:

- User
- registered user
- Visit users
- Update user information
- Age
- Height
- Gender
- delete users
- Measurement
- Measurement status
- Real-time weight
- Storing data
- Measurement data
- Distribution network
- WiFi name
- WiFi password
- Data upload address
- OTA upgrade address
- Communication key
- Distribution results
- Other
- Unit

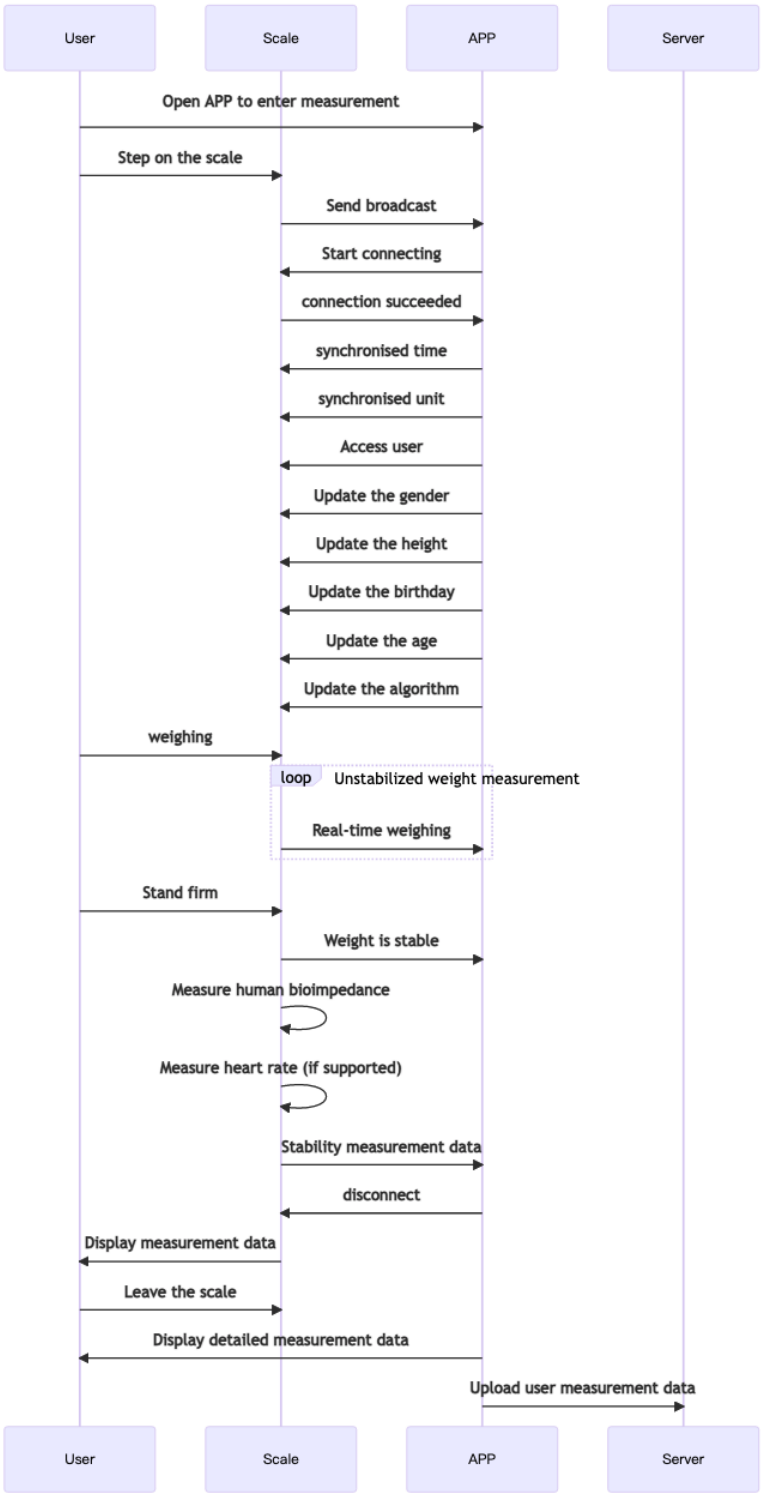
The device involves many aspects, and the communication process between protocols and the enabling method are strictly required. Our company has packaged it into SDK for easy access.

Workflow in different situations

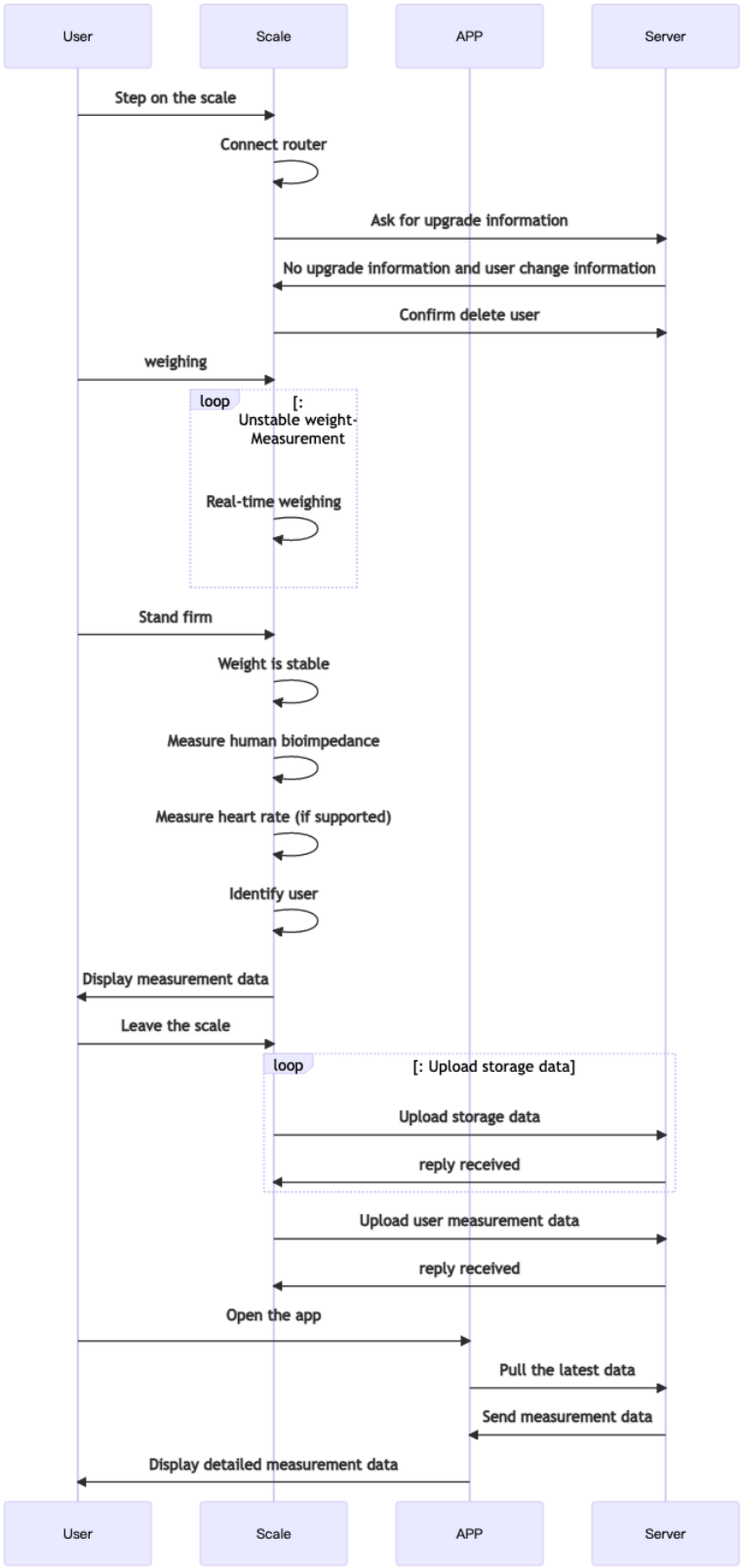
- The first time to use the binding distribution network


User
Scale
APP
Server
Open APP to enter measurement
Step on the scale
Send broadcast
Requests the user to be deleted
Example Deliver the user to be deleted
Request a key
Issued by the key
A connection
Setting up a Bluetooth connection
Synchronization time
Synchronous unit
Send the delete user command
Delete user
Registered users
Return index value
Send a placeholder success message
Access user
Update the gender
Update the height
Update the birthday
Update the age
Update the algorithm
weighing
loop
Unstabilized weight measurement
Real-time weighing
Stand firm
Weight is stable
Measure human bioimpedance
Measure heart rate (if supported)
Stability measurement data
disconnect
Display measurement data
Leave the scale
Display detailed measurement data
Upload user measurement data
Prompt distribution network
Click to enter the distribution network
Request encryption key, OTA link, etc.
Step on the scale
Send broadcast
Start connecting
connection succeeded
synchronised time
synchronised unit
Access user
Notice to enter the distribution network
Send wifi name
Send wifi password
Sending server address
Send OTA address
Send encryption key
Notify the server to change the communication key
Connect to wifi
Send distribution results
Disconnect
Send request, test link
Respond to requests
Save the distribution results
Show the results of the distribution network
User
Scale
APP
Server

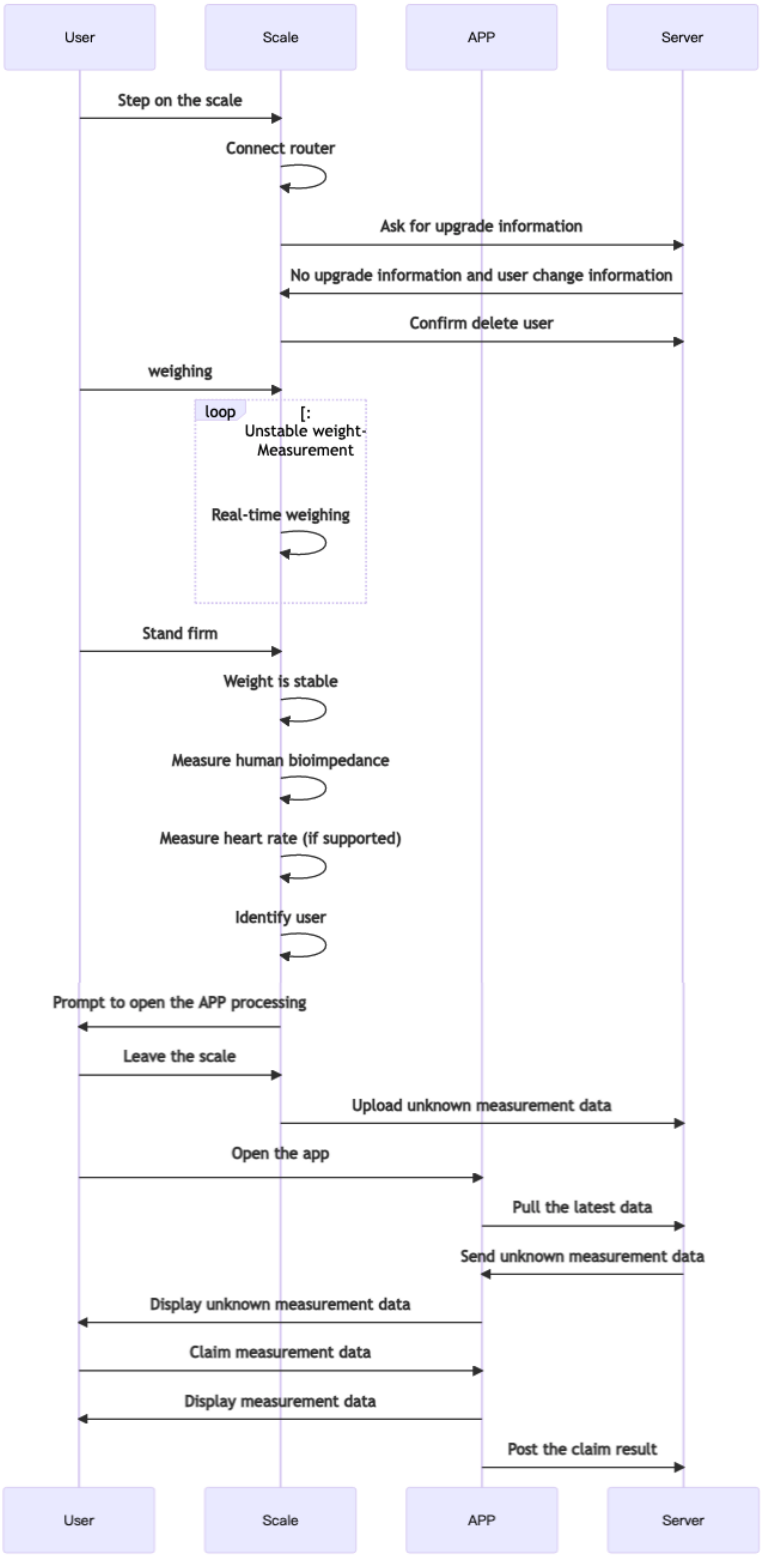
- Measurement with Bluetooth connection



- Normal measurement



- Measurement of unidentified users



One、initialization

Use QNBleApi.initSdk to initialize

2. scanning

1. Set the Bluetooth scan callback monitor class

Before scanning, you need to set the listener class. The method is QNBleApi.setBleDeviceDiscoveryListener. This method only needs to be called once. When you do not need to scan, remember to set to null/nil

android example:

```
QNBleApi.getInstance(context).setBleDeviceDiscoveryListener(new QNBleDeviceDiscoveryLi
    @Override
    public void onDeviceDiscover(QNBleDevice device) {
//This method calls back the device object, which can be processed by the device objec
    }

    @Override
    public void onStartScan() {
//Back to start scanning
    }

    @Override
    public void onStopScan() {
//return to the end of the scan
    }

    @Override
    public void onScanFail(int code) {
//The callback for the scan failure will be returned with an error code. For details,
    }
});
```

iOS example:

```
//set proxy
QNBleApi *bleApi = [QNBleApi sharedBleApi];
bleApi.discoveryListener = self;

//Implement the proxy method
- (void)onDeviceDiscover:(QNBleDevice *)device {
//This method will call back after the device is found
}

- (void)onStopScan {
//Callback when scanning is stopped
}

- (void)onStartScan {
//Callback when starting scanning
}
```

2. Start the scan

Confirm that Bluetooth is turned on, and Android needs to check the `location permission` and `location switch`. If you confirm that Bluetooth is turned on, the location permission is authorized, and the location service switch is turned on, you can start Bluetooth scanning

> After Android 6.0, targetSdkVersion>= 23 App above, you need to obtain positioning permission for Bluetooth scanning, please check About The location service switch is not mandatory, but some phones do not turn on this switch, and the device cannot be scanned, which is related to each mobile phone system.
>
> iOS13 system has added Bluetooth usage permission, you need to check whether there is usage permission, confirm that authorized and Bluetooth is turned on, start scanning

The scanning method is QNBleApi.startBleDeviceDiscovery,-the-scanned-device-data-will-be-called-back-in-the-qnbledevicediscoverylistener in the scanning interface set above.

In addition, some feature settings related to scanning can be set in QNConfig, and the content to be set has been basically covered.

> Usually APP will have an interface dedicated to measurement. We generally perform Bluetooth scanning after the interface is displayed, and stop scanning when the interface disappears.

android example:

```
QNBleApi.getInstance(context).startBleDeviceDiscovery(new QNResultCallback() {
            @Override
            public void onResult(int code, String msg) {
//This method does not return to the device, but indicates whether the scan was starte
                if (code != CheckStatus.OK.getCode()) {
                    ToastMaker.show(ScanActivity.this,code+":"+msg);
                }
            }
        });
```

iOS example:

```
//start scan
[[QNBleApi sharedBleApi] startBleDeviceDiscovery:^(NSError *error) {
//The callback here indicates whether the scan method is successfully started
    if (error) {
        NSLog([NSString stringWithFormat:@"Failed to start scanning method, reason: %@
    }
}];
```

three、connection

After receiving the callback device, you can determine whether it is a device that needs to be connected (this is the business logic of APP), and if so, connect it.

Before connecting, you need to set the connection status callback and measurement data callback.

The method for setting the connection status callback is: QNBleApi.setBleConnectionChangeListener, this method is the same as setting the scan callback method, multiple connections only need to be set once, if it is

determined that no longer need to connect Time, set to null/nil

After setting the Bluetooth status callback, you also need to set the callback interface of the measurement data. The method is: QNBleApi.setDataListener, which can also be set once, please set it when it is no longer used Null/nil

android example:

```
QNBleApi.getInstance(context).setDataListener(new QNScaleDataListener() {
    @Override
    public void onGetUnsteadyWeight(QNBleDevice device, double weight) {
//This method received unstable weight data. During one measurement, this method will
    }

    @Override
    public void onGetScaleData(QNBleDevice device, QNScaleData data) {
//The method is to receive the complete measurement data
    }

    @Override
    public void onGetStoredScale(QNBleDevice device, List<QNScaleStoreData> storedData
//This method is to receive the stored data on the scale side, the processing method c
    }

//Connection status during measurement
    @Override
    public void onScaleStateChange(QNBleDevice device, int status) {
        setBleStatus(status);
    }
});
```

iOS example:

```
- (void)onGetUnsteadyWeight:(QNBleDevice *)device weight:(double)weight {
//This method received unstable weight data. During one measurement, the method will c
}

- (void)onGetScaleData:(QNBleDevice *)device data:(QNScaleData *)scaleData {
//The method is to receive the complete measurement data
}

- (void)onGetStoredScale:(QNBleDevice *)device data:(NSArray <QNScaleStoreData *> *)st
//This method is to receive the stored data on the scale side, the processing method c
}

- (void)onScaleStateChange:(QNBleDevice *)device scaleState:(QNScaleState)state {
//Connection status during measurement
}

- (void)wspRegisterUserComplete:(QNBleDevice *)device user:(QNUser *)user {
//callback for registered user success
}

- (void)onScaleEventChange:(QNBleDevice *)device scaleEvent:(QNScaleEvent)scaleEvent {
//callback of device transaction
}
```

In addition, before calling the connection, it is best to stop the previous scan (we found that some mobile phones have Bluetooth scanning and Bluetooth connection at the same time, which will reduce the failure rate of successful

connection). After stopping the scan, delaying the call for 200 ~ 500ms before calling the connection will increase the success rate of the connection. The method to stop Bluetooth scanning is: QNBleApi.stopBleDeviceDiscovery.

android example:

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

iOS example:

```
[[QNBleApi sharedBleApi] stopBleDeviceDiscovery:^(NSError *error) {
//The callback here indicates whether the method to stop scanning is successful
    if (error) {
        NSLog([NSString stringWithFormat:@"Failed to stop scanning method, reason: %@"
    }
}];
```

The method of connecting the device needs to pass the Yolanda Bluetooth device object (QNBleDevice) and the Yolanda user profile object (QNUser), where QNBleDeviceQNUser is created as: QNBleApi.buildUser, the specific method of use can be Refer to the method description.

After confirming OK in the previous operation, you can finally connect. The connection method is: QNBleApi.connectWspDevice.-the-connection-status-callback-method-is-the-above-qnbleconnectionchangelistener

android example:

```
 QNWspConfig qnWspConfig = new QNWspConfig();
    //.......Make configuration settings
    mQNBleApi.connectWspDevice(device, qnWspConfig, new QNResultCallback() {
        @Override
        public void onResult(int code, String msg) {
            QNLogUtils.log("WspScaleActivity", "wifi Configuration code:" + code + ",m
        }
    });
```

iOS example:

```
//Set the operation configuration of wsp device
    QNWspConfig *config = [[QNWspConfig alloc] init];
    [[QNBleApi sharedBleApi] connectWspDevice:device config:config callback:^(NSError

    }];
```

Weighing process

The data and status of the weighing process will be recalled in the above-mentioned QNScaleDataListener

End of measurement

After receiving stable data (that is, QNScaleDataListener.onGetScaleData), the measurement is completed. When the measurement is completed, the device will automatically disconnect.

At this point, the basic process of the device has been completed, and the APP can save the data and display the data after receiving the stable data. Data standards can be judged by our company-SDK indicator standard description.

QNBleApi

Entrance class

Method

initSdk

Initialize the SDK according to appid and verify the appid and configuration file. This method will also load the relevant configuration in the configuration file.

This method only needs to be called once, not multiple times. It is recommended that Android be called in Application.onCreate, and iOS should be called under AppDelegate didFinishLaunchingWithOptions. This method only performs lightweight operations, and the time-consuming operations are all performed asynchronously, and will not affect the startup speed of the APP.

Normally, this method sends a request to our server to update the configuration in the configuration file, such as updating the algorithm, updating the index, and adding new models. This can achieve the effect of online upgrade configuration. If you clearly do not want to have this feature, you can close this function with our business communication (that is, change the SDK to offline mode). In addition, if there is no network, after the request fails, there will be no impact, the SDK will continue to use the last configuration.

> Note: The `123456789` appid in the demo is the test id, which has instability and random variability. After reaching an agreement with Yolanda, the access provider can obtain an independent and stable appid. Remember to test Id can not be used in the online application market

Parameters

name	Types of	Explanation
appId	String	The app_id provided by the company to the customer is verified by the local initialization data packet, and Android iOS shares the same appid
firstDataFile	String	The file path of the configuration file can be transmitted in the form of uri or the full path of the file. The file contains
callback	QNResultCallback	Android is the interface, IOS uses the block or closure) callback method, and returns the initialization result. Refer to the attached table for error codes.

setSysBleStateListener

Set system Bluetooth status callback

Parameters

name	Types of	Explanation
listener	QNBleStateListener	The listener contains a method to monitor the Bluetooth status of the system.

setBleDeviceDiscoveryListener

Set the scan callback object. When there is a scan result, it will be returned in the set monitoring method. The SDK only supports one callback object. Repeated settings will overwrite the previous results. When it is not needed, it needs to be reset to null, otherwise a memory leak may occur.

Parameters

name	Types of	Explanation
listener	QNBleDeviceDiscoveryListener	The listener contains a method that will callback in this object when the device is scanned.

getCurSystemBleState

Get the current system Bluetooth status

return value

Type: int

Current System Bluetooth Status

startBleDeviceDiscovery

Start scanning for Bluetooth devices and only return to Yolanda's device. It automatically stops when it encounters an error or Bluetooth is turned off.

The scan result is called back in QNBleDeviceDiscoveryListener

For some configuration behavior of scanning, please refer to getConfig-and-qnconfig

Parameters

name	Types of	Explanation
callback	QNResultCallback	Callback object, returns whether the call to start scanning is successful

stopBleDeviceDiscovery

To stop scanning Bluetooth devices, this method calls the system's stop scanning method. If scanning is not started, you also need to call stop scanning to ensure that there will be no more callbacks for scanning objects after calling this method.

Parameters

name	Types of	Explanation
callback	QNResultCallback	Callback object, return whether the call to stop scanning is successful

connectWspDevice

Connect the Yolanda Wsp device. The connection process will be called back in QNBleConnectionChangeListener, and the measurement data will be called back in QNScaleDataListener

Parameters

name	Types of	Explanation
device	QNBleDevice	Bluetooth device to be connected.
config	QNWspConfig	Configuration items when connecting to wsp device
callback	QNResultCallback	Returns whether the connection operation is successfully called (not the connection is successful)

disconnectDevice

Disconnect the connected Yolanda Bluetooth device. The process of disconnecting will be called back in QNBleConnectionChangeListener

Parameters

name	Types of	Explanation
device/mac	QNBleDevice/String	Need to disconnect the Bluetooth device or mac address, these two parameters only need to wear one
callback	QNResultCallback	Returns whether the connection operation is successfully called (not the connection is successful)

setBleConnectionChangeListener

Set the Bluetooth connection change listener. The SDK only supports one callback object. Repeated settings will overwrite the previous results.

Parameters

name	Types of	Explanation
listener	QNBleConnectionChangeListener	Connect Change Listener

setDataListener

Set up data transmission listener

Parameters

name	Types of	Explanation
listener	QNScaleDataListener or QNWspScaleDataListener	Measurement data monitoring interface, all data will be called back in it

setLogListener

Set the monitoring of log information

Parameters

name	Types of	Explanation
listener	QNLogListener	Log information monitoring

getConfig

Get the current settings of SDK

return value

Types of: QNConfig

SDK setting object, the setting method is also done through QNConfig

convertWeightWithTargetUnit

Convert the weight into the specified unit according to the weight provided in kg

> St is not supported, if st is passed, the value of lb will be returned directly, that is, if lb or st is returned, the APP needs to directly convert the unit of British stone, 1st = 14lb, for example, 145.2lb will be displayed as 10 st 5.2 lb.

Parameters

name	Types of	Explanation
kgWeight	double	Weight in kilograms
unit	int	0 is kg, the default value 1 is lb, pound, all scales can support this unit 2 is kg. If the scale does not support it, kg will be displayed

return value

Type: double

Returns the value in the specified unit

generateScaleData

Generate measurement data, if the parameter is abnormal, the method will return NUL

This method only supports wsp devices

Parameters

name	Types of	Explanation
user	QNUser	The user to which the data belongs
modeId	String	Model identification
weight	Double	Weight in kg
resistance	int	50 impedance value
secResistance	int	500 impedance value
heartRate	int	Heart rate value, or 0 if none
measureTime	Date	measure time
hmac	String	Related data signature

return value

Type: QNScaleData

On error, return NULL

QNResultCallback

Unified method call callback method

onResult

Parameters

name	Types of	Explanation
errorCode	int	If the error code is 0, the call is successful, and the others are errors. For details, please refer to the attached table
errorMsg	string	Error message, if successful, return ok

QNBleDevice

Yolanda Bluetooth device object, currently only refers to body fat claims, follow-up may continue to the bracelet

Attributes

name	Types of	Explanation
mac	String	The MAC address and SDK will be parsed according to the scanned broadcast information. On the same device, Android and IOS will return the same result. Unique for each device
name	String	The model name, the SDK will identify the scanned device model, if it cannot be identified, it will return a default model name, Scale
modeId	String	Model identification
bluetoothName	String	Bluetooth name, this is the broadcast name of the hardware device, different from name. APP is based on name.
RSSI	int	The signal strength is a negative number, generally between -30 and -100. The larger the value, the greater the signal strength. The same device may be recalled multiple times. Each time the callback is performed, the signal strength value may be different.
isScreenOn	Boolean	Whether it is turned on, the screen of the scale is on, it is turned on, otherwise it is not turned on.
isSupportWifi	Boolean	Whether to support Wifi function, if it supports, you can call the operation such as distribution network

QNUser

Yolanda user object

Attributes

name	Types of	Explanation
userId	String	User unique identifier, cannot be empty, each user is unique and related to stored data
height	int	Height, unit cm, minimum 40, maximum 240
gender	String	Sex: Male:male: female
birthday	Date	The birthday is used to calculate the age. The date is accurate to the day. The calculated age range is 3 ~ 80. If it exceeds this range, it is equal to the upper or lower limit.
athleteType	int	Whether it is the athlete mode (0 is the ordinary algorithm, 1 is the athlete algorithm), this field only takes effect when the age is greater than or equal to 18
clothesWeight	double	unit: KG Clothes weight, when this value is set, the weight value returned will be the value minus the clothes weight. The weight of clothing cannot exceed half of the weight, otherwise the excess value will be ignored
index	int	For WSP equipment only, the user index of the scale side, this value is returned by the scale side when the user is successfully registered with the scale
secret	int	Special for WSP equipment, the user key of the scale, this value is generally issued by the server, and the value range is (0,9999)
indicateConfig	QNIndicateConfig	The user sets the scale display indicator control object (this object is only valid for WSP scale)

QNBleStateListener

System Bluetooth status monitoring

Status type

name	value	Explanation
unknown	0	Unknown state, this state will be called back for the first time when initializing Bluetooth
resetting	1	System Bluetooth is resetting
unsupported	2	The system does not support Bluetooth
unauthorized	3	unauthorized
poweredOff	4	System Bluetooth is off
poweredOn	5	System Bluetooth is on

Method

onBleSystemState

Callback of system Bluetooth status change

Parameters

name	Types of	Explanation
state	int	system status

QNScaleDataListener

Data listener, related measurement data is called back here

Method

onGetUnsteadyWeight

Received the real-time weight uploaded by the scale, which can be used to synchronously display the value on the scale

Parameters

name	Types of	Explanation
device	QNBleDevice	Device for uploading real-time data.
weight	double	When the APP is connected for measurement, the value on the scale changes, and the APP will have a synchronous data callback, which is only displayed in the UI, not saved as the last data.

onGetScaleData

The real-time stable measurement data is obtained. When the APP is connected and the measurement is performed, the data will enter this callback

Parameters

name	Types of	Explanation
device	QNBleDevice	The device that uploaded the data.
data	QNScaleData	Yolanda measurement data, including weight, BMI、Data such as body fat percentage.

onGetStoredScale

Obtained storage data, this kind of data is that when the user does not connect to the APP when measuring, the data will be stored to the scale side, and then when the next time is connected, the scale will upload the stored data to the APP, and then will enter this Callback.

Parameters

name	Types of	Explanation
device	QNBleDevice	The device that uploaded the data.
storedDataList	List<QNScaleStoreData>	Storage data list, the old scale can store at most 20 data, the new one can store 40 data. QNScaleStoreData can generate QNScaleData by passing in user data (that is, QNUser)

onGetElectric

Get battery

Currently only supports charging scales. When the battery is less than 20%When the power is low

name	Types of	Explanation
device	QNBleDevice	The device that uploaded the data.
electric	int	Power unit%

onScaleStateChange

Method callback after measuring state change, this method is not a callback for Bluetooth connection status

Parameters

name	Types of	Explanation
device	QNBleDevice	Bluetooth device object
scaleState	int	Refer to Scale Status Code

onScaleEventChange

Callback for scale events

Parameters

name	Types of	Explanation
device	QNBleDevice	Bluetooth device object
scaleEvent	int	Refer to Scale Events

QNLogListener

Log information monitoring

Method

onLog

Log information output

Parameters

name	Types of	Explanation
log	String	Log information

QNConfig

SDK configuration object, which controls the scanning behavior, and sets some SDK behaviors through it

Attributes

name	Types of	Explanation
onlyScreenOn	Boolean	Whether to return only the devices that have been turned on (bright screen), the default is false, that is, return to the scale that is turned on and not turned on.
allowDuplicates	Boolean	Whether the same device returns multiple times, in general, the device will always send Bluetooth broadcasts, the system will call back to the same device multiple times during scanning, and the SDK will process the same device, so that during a scan, one device will only return Once, the merchant can set this parameter, and the SDK returns directly every time. Default false (this field is invalid for onBroadcastDeviceDiscover-monitoring)
duration	int	Scan duration, unit ms, the default is 0, that is to scan, unless the APP calls stopBleDeviceDiscovery. When it is not 0, the minimum value is 3000ms, the scan will be stopped automatically after a delay of duration ms. (If you have access to a broadcast scale, you need to pay attention to the setting of this field, see Working Principle of Broadcast Scale)
connectOutTime	long	(Only for Android) The limited time to connect the device, the unit is ms, the default is 10000, that is, within 10 seconds of starting the connection, if the entire connection process cannot be completed, the connection will be stopped and callback error;When the set value is less than or equal to 0, it means no callback;When the set value is less than 3000 and greater than 0, it means that the timeout period is 3000ms. If the connection has been stopped before the connection times out, even if the device is not successfully connected during the period, there will be no error callback.
unit	int	0 is kg, the default value 1 is lb, pound, all scales can support this unit 2 is kg. If the scale does not support it, kg will be displayed 3 is st, British stone, if the scale end does not support, it will display the value of lb,

name	Types of	Explanation
showPowerAlertKey	bool	The default is false, when it is true, when the SDK is initialized and Bluetooth is turned off, the system will pop up a box to prompt Bluetooth status. For details, please refer to Apple Developer Documenttation => CoreBluetooth => CBCentralManager => Central Manager Initialization Options
setNotCheckGPS	bool	(Android only) The default is false. When it is true, the SDK will not judge GPS permission when scanning, and continue to perform scanning
enhanceBleBoradcast	bool	Whether it is necessary to start the enhanced broadcast scale signal during scanning (invalid for ordinary scales, this option is only for broadcast scales)

unit

The unit displayed on the terminal, if not set, the SDK defaults to kg. After setting, it will be saved locally. If there is currently a connected device, the unit display on the scale will be updated in real time as much as possible.

> The SDK will always return the value of kg. APP needs to convert values by itself.

Method

save

Save modified settings

Parameters

name	Types of	Explanation
callback	QNResultCallback	Returns whether the setting operation is successful

QNWiFiConfig

WiFi information

Attributes

name	Types of	Explanation
ssid	String	wifi name (length must be less than 32 bytes)
pwd	String	wifi password (when there is a password, the password length must be greater than or equal to 8 bytes and less than 64 bytes (No password, you can pass nil)
serveUrl	String	Data transmission address, `WSP device setting is valid` , `Other devices can be set to null` , the format requires http://hostname:port/path/, the maximum length is 128 bytes

Method

checkSSIDVail

Check the validity of the WiFi name

return value

Type: Boolean

checkPWDVail

Check the validity of the WiFi password

return value

Type: Boolean

QNWspScaleDataListener

wsp device data listener, the relevant measurement data of the device is called back here, the listener inherits from QNScaleDataListener

Method

wspRegisterUserComplete

The user registration of the wsp device is successful. After the user registration is successful, the device returns the serial number of the user on the scale

Parameters

name	Types of	Explanation
device	QNBleDevice	Bluetooth device object
user	QNUser	User object

QNWspConfig

WSP device configuration class, which controls WSP operation

Attributes

name	Types of	Explanation
wifiConfig	QNWiFiConfig	Whether to configure the network, when the value is null, no network operation is performed；When the value is the QNWiFiConfig object, configure the network according to the value in the object
deleteUsers	[int]	User index collection to be deleted
curUser	QNUser	Current measurement user
isRegist	Boolean	Whether user registration is required, with the isChange attribute, only one of them is allowed to be true
isChange	Boolean	Whether user information needs to be modified, and the isRegist attribute, only one of which is true
isVisitor	Boolean	Whether to use visitors for measurement, it is not necessary to set isRegist and isChange when using visitors for measurement, and QNUser does not need to set index、secret
dataUrl	String	Data transmission address, only works when wifiConfig has value, otherwise it can be set to null, the format requires `http://hostname:port/path/` , the maximum length is 128 bytes
otaUrl	String	OTA upgrade address, only works when wifiConfig has value, otherwise it can be set to null, the format requires `protocol://hostname[:port]/path/` , the maximum length is 128 bytes
encryption	String	The communication key must be 16 bytes, and only works if wifiConfig has a value, otherwise it can be set to null
longitude	String	Longitude is first converted into a 7-digit string, for example: `+134.54` , note that both + and. Are required. For scales that support weather query, the latitude and longitude need to be set for weather query, if it is empty, it will not be displayed
latitude	String	Latitude, first converted to a 7-bit string, for example: `-022.41` , note that both- and. Are required. For scales that support weather query, the latitude and longitude need to be set for weather query, if it is empty, it will not be displayed

QNBleConnectionChangeListener

Bluetooth connection change monitoring interface

onConnecting

Connected device is in progress, after calling the connected device, it will call back immediately

Parameters

name	Types of	Explanation
device	QNBleDevice	Bluetooth device object with state change

onConnected

The device is successfully connected Will call back the device object

onServiceSearchComplete

The device's service search is completed, and it will be called after onConnected under normal circumstances

onDisconnecting

Disconnecting, when calling disconnect, it will call back immediately

onDisconnected

Bluetooth disconnected

onConnectError

A connection error has occurred

Parameters

name	Types of	Explanation	
device	QNBleDevice		Bluetooth device object
errorCode	int	Error code reference attached table-error code, the second parameter of IOS uses system error object	

QNBleDeviceDiscoveryListener

Scanning device listening callback class

Method

onDeviceDiscover

Parameters

name	Types of	Explanation
device	QNBleDevice	Scanned Bluetooth device

onBroadcastDeviceDiscover

Callback of broadcast scale equipment related information

The SDK will return the device information that can scan all the surrounding broadcast scales, and the app needs to process this information by itself

> At the same time, in order to be compatible with customers who have access to the broadcast scale, the device information of the callback broadcast scale in the original onDeviceDiscover method will remain unchanged. It is strongly recommended that customers who have access to the broadcast scale use this API

Parameters

name	Types of	Explanation
device	QNBleBroadcastDevice	Scan to broadcast scale

onKitchenDeviceDiscover

Callback of information about kitchen scale equipment

The SDK will return the device information that can scan all the kitchen scales around, and the app needs to process this information by itself

Parameters

name	Types of	Explanation
device	QNBleKitchenDevice	Scan to kitchen scale

onStartScan

This method is triggered after a successful scan start

onStopScan

Callback after stopping scanning

This callback will be triggered after automatic stop (timed) or stop by calling stopScan

onScanFail

Callback if no scan is successful (only Android will have this callback)

Parameters

name	Types of	Explanation
code	int	Refer to Scan Status Code

QNScaleData

Yolanda body fat scale measurement data, including body weight, BMI、Data such as body fat percentage.

Attributes

Name	Type	Description
user	QNUser	Measured users, including userId, gender, birthday, height, etc.
measureTime	Date	Measured time, accurate to seconds
hmac	String	Characteristic identification of data
height	Double	height
heightMode	int	heightMode (1 is body fat mode, 0 is body weight mode)

Method

getItem

Get a single indicator, or NULL if no specified type

Parameters

name	Types of	Explanation
type	int	Index type, refer to the attached table

return value

Type: QNScaleItemData

A single indicator object contains the value of this indicator, the English name, whether it meets the standard, etc.

getItemValue

Get the value of a single indicator, only return the value of this indicator

Parameters

name	Types of	Explanation
type	int	Index type, refer to the attached table

return value

Type: double

Returns the floating-point number type. The value of some indicators is actually of type int, and the APP needs to convert it by itself.

getAllItem

Return all the measurement indicators in the form of a list. If the measurement data is invalid, the values of other indicator data are all 0 except weight and BMI

return value

Type: List<QNScaleItemData>

Index list, the order will be sorted according to TYPE, if APP needs to redefine the order, you need to sort by yourself

setFatThreshold

Body fat change control

It is used to control the change of body fat and alleviate the problem of excessive body fat difference. Wsp If this method is called for body fat change control, it may happen that the index after the change control is inconsistent with the data displayed on the scale side

> Precautions for use:
> 1. This method can be used only when there is not much change in body data and weight；
> 2. The access party must save the characteristic identifier of the data `hmac` in order to control body fat changes；
> 3. The relevant index value must be obtained after calling this method for control, otherwise data will be garbled

Parameters

name	Types of	Explanation
hmac	String	The data feature identifier of the last valid measurement data (i.e. body fat measurement data) cannot be nil
threshold	Double	The maximum error control of body fat, this value is only the approximate range of control, and will not be accurately controlled. For example, if the value is 2, then when the difference between the measured body fat and the last time is greater than 2, the measured body fat will be controlled to be about 2 different from the last measured body fat
callback	QNResultCallback	Whether the setting is successful

QNScaleStoreData

Yolanda stores data objects. When the measurement is not connected to the APP, after connecting to the APP after the end, you will receive such data

Attributes

Name	Type	Description
weight	double	The weight corresponding to this stored data
measureTime	Date	Measurement time
mac	String	mac address
isDataComplete	Boolean	Whether the stored data is complete, there is no need to call the setUser method when the data is complete, when the data is not complete, you need to use the setUser method to set the user, regardless of whether the data is completed, you need to use the generateScaleData method to generate data
height	Double	Height data

Method

setUser

Set up measurement users

name	Types of	Explanation
user	QNUser	User profile, used to produce measurement data

return value

Type: Boolean is set successfully

generateScaleData

Generate measurement data based on stored data and user data. If user data is not set, this method returns NULL

return value

Type: QNScaleData

On error, return NULL

buildStoreData

Used to build stored data

> Only support obtaining relevant information about storage data from Yolanda Cloud for construction

Parameters

name	Types of	Explanation
weight	double	Weight in kilograms
measureTime	Date	measure time
mac	String	Scale side unique identification
hmac	String	Relevant data information, use signature information in the query user profile interface in the Bluetooth scale with Wifi function
callback	QNResultCallback	Whether the construction is successful

return value

Type: QNScaleStoreData

> This method is specially designed for Bluetooth scales with Wifi function, which is not used in ordinary Bluetooth scales and broadcasting scales.

QNScaleItemData

Yolanda body fat is called a single indicator object and contains the basic data of the report generated by the indicator.

Attributes

name	Types of	Explanation
type	int	The type of the indicator, please refer to the attached table
value	double	The value of this indicator, in order to be universal, this field will directly return double, if the value of this item is int type, the app needs to determine the type by itself and then convert
valueType	int	Type of indicator value, 0 is double, 1 is int
name	String	The name of the indicator is all in English, providing a table to convert the Chinese name

common problem

APP logic design related

Whether to bind or pair the device?

A concept needs to be clarified here. The binding or pairing mentioned in the question refers to adding a paired Bluetooth device in Android/iOS. Usually some standard equipment will be needed, such as Bluetooth headset, stylus, bracelet. These will need to be set in the Bluetooth settings of the mobile phone system. But our scale is not used, we will ask for instructions

Bluetooth 4.0, or BLE, can be used directly without binding a Bluetooth device. The private protocol used by our scales cannot be used directly via the mobile phone system Bluetooth, which is true of almost all body fat scales on the market. The added device in our APP (Light Cow Health) only allows the APP to remember the user's scale, not the Bluetooth pairing of the mobile phone system. The process of adding equipment also guides the user how to use body fat scales. Our small program is made in a simple and rude way. After adding the scale and opening it directly, connect to the scale and measure.

The customer can decide whether to add this logic. It is recommended that if it is made into a small function module of APP, it can be directly made into a scale similar to our small program, directly measured on the scale, and displayed data.

How to make the user respond immediately after stepping on the scale

Our own APP (Light Cow Health), in normal use, is to let the user open the APP, step on the scale, and measure immediately, which can reduce the user's operation steps. In fact, to put it bluntly, after opening the APP, the APP automatically calls the entire set of logic of the SDK, without requiring the user to click to select a device or the like. The specific implementation steps are as follows:

1. After entering the measurement interface, check the Bluetooth status
2. Bluetooth is on, start scanning
3. Scan to the body fat scale to determine whether it is an added device or connect directly without judgment (the location depends on the APP logic)
4. Successful connection, display measurement animation
5. The measurement is completed and the measurement results are displayed

After the measurement is completed, is it necessary to immediately disconnect the Bluetooth connection

There is usually no need to actively disconnect the scale unless the user exits the measurement interface.

If you need to measure the immediate disconnection, it is best to be able to delay 2s disconnection. Some of our early successes, if the measurement is completed immediately disconnected, the scale may appear unexpected.

How to analyze data

Whether a data is standard、Low, etc., this can refer to our SDK indicator standard description.

Our Demo also shows how to analyze the data, but the Demo interface only shows the rating registration, and it does not make a beautiful report like our APP.

APPID is related to configuration file

What is appid and configuration file? What is the role?

The appid is a unique identifier created by Yolanda for customers, and will be registered in our background. A given appid usually does not change. The appid `123456789` in the demo is the appid that we tested and let customers experience. The appid 's characteristics are unstable and may be modified at any time. If you have reached a formal cooperation intention, it is recommended to apply to us for appid for business or sales. The application process is very fast and can be completed in a few minutes.

The configuration file is some encrypted data used with appid, and the model is agreed、algorithm、Indicators and other information. Different customer requirements are all different, so we encapsulate customer customization requirements in this configuration file.

Can different clients use the same APPID

Android and iOS can use one appid, the applet/applet plug-in cannot be the same as the appid of the SDK

Whether the SDK is offline? What data will it send to Yolanda cloud?

In the method initSdk,-the-sdk-will-send-a-request-to-verify-with-our-cloud-that-the-configuration-file-needs-to-be-updated.-if-you-need-to-update,-the-latest-configuration-file-will-be-downloaded-(the-configuration-file-is-very-small,-generally-512-bytes), if you do not need to update, then do nothing.

The SDK supports offline mode. If customers specifically request it, we will set the configuration of this item in the configuration file to `offline`. After setting to offline, the SDK no longer sends out any network requests (this can be verified with the packet capture tool). The configuration file is no longer updated. If you need to add a model or indicator, you need to apply for a configuration file with our company again, and then the developer Replace manually.

Initialization prompt appid error

- Check if the initialization file and the appid used match
- Check if the introduced SDK is up to date

Missing data returned after measurement

If it is only the weight and BMI data, you can refer to there is no data such as body fat rate after the measurement is completed.

If the body fat rate can be measured, but the indicator you want is missing, the indicator is usually not included in the configuration file. You need to communicate with our business/sales to increase this indicator.

After adding indicators, our company will re-send a configuration file, which needs to be replaced.

Send the model ID of the scanned device to our company for confirmation

SDK function related

Is it possible to judge whether the scale is off?

For `ordinary Bluetooth scale` and `dual-mode scale` , when the device is not connected, scan the obtained device object QNBleDevice, there is an attribute `isScreenOn` , which indicates Whether the screen is bright. After connecting the device, it is impossible to directly judge whether the device is bright or not, but for some scales, the connection will be disconnected when the screen is turned off.

> The isScreenOn attribute only indicates that the device is bright when the SDK scans to it, and the subsequent device screen state changes, and QNBleDevice will not be dynamically updated. The new QNBleDevice callback needs to be scanned to determine.

For the 'broadcast scale', the ability to scan to the device is the state of being turned on. If it cannot be scanned, it is usually not turned on.

The connection device has been unsuccessful or will be disconnected soon after success

- Check if the device is connected by others
- Check whether the currently connected device has been paired in the system Bluetooth, if it is already paired, you need to cancel the pairing
- Some mobile phones need to be scanned before they can connect successfully. Scan the device before connecting

SDK returns no location permission error

- Check if it is correct **ACCESS_COARSE_LOCATION** with **ACCESS_FINE_LOCATION** Both have applied, and both permissions have been verified in the SDK

- Whether to compile version 26 and above, if yes, both permissions need to be applied separately (new features in 8.0)

  > For Android 6.0 and above, Google classifies Bluetooth as a part of the positioning function. When using Bluetooth scanning, the user needs to authorize the positioning permission, otherwise when the scan is called, the system will prompt that the positioning permission is required A considerable part of mobile phones (about 1/3 of the appearance, the device cannot be scanned without turning on the location service switch, most of them are native, and the domestic system will be optimized for this, it is not necessary)

The SDK returns the wrong file, confirm that the file location is not abnormal, and confirm that the file is used in the demo without exception

- Check if there is so library added
- Check if the packaged apk file contains so library

Monitor callbacks for data or devices, etc., multiple callbacks at the same time

- First determine whether or not, set multiple monitoring. When monitoring is not used, it must be set to null
- Determine whether it is a measurement of wearing shoes, this may lead to the situation of completing multiple measurements in a short time

After the measurement is completed, there is no data such as body fat rate

1. Check whether the information is correct, mainly birthday (used to calculate age)、Whether the two parameters of height exceed the normal value range
2. Check if any shoes are taken off. When wearing shoes, the biological impedance cannot be measured, so the body fat rate and other indicators cannot be calculated

I can't scan the device, what should I do

1. Check if Bluetooth is turned on、Have you stepped on the scale, is the phone too far away from the scale (more than 10 meters)
2. Check if the scanned device has been connected by others
3. If it is Android, you can check whether there is location permission, and whether there is a location service switch
4. Try to restart Bluetooth, if not, try restarting Bluetooth.

Is there a bracelet SDK?

Our company has smart bracelet devices and SDK. However, the SDK is not completely open to the outside world temporarily. If you have any intention, you can contact our company for business negotiation.

Whether the SDK can only do protocol analysis, Bluetooth scanning and connection are managed by customers themselves

This is achievable, the implementation of ordinary Bluetooth scale and Bluetooth broadcast scale will be different.

> It is technically difficult to implement Bluetooth scanning and connection management by yourself. If you do not have relevant Bluetooth development experience, this operation is not recommended. If you insist on this operation, we will think that the customer has richer Bluetooth development experience.

1. Write your own Bluetooth protocol proxy class to implement each Bluetooth operation method
2. Write Data Monitor Callback Class QNScaleDataListener and use [QNBleApi.setBleDeviceDiscoveryListener to register in SDK.
3. Scan the Bluetooth yourself and pass the scanned information to the SDK. Use the method QNBleApi.buildDevice to create a Yolanda Bluetooth device
4. Create user object, use method QNBleApi.buildUser
5. Create Bluetooth Protocol Processing Class
6. Make a Bluetooth connection
7. The connection is successful, and after finding the service is successful, call prepare method of Bluetooth protocol processing class
8. Process the data of QNScaleDataListener callback
9. After receiving stable data, process it by yourself.

Your official website API《Scale status definition》The status here is 0-9, why is it promising in the demo you provided“-1”'S state (and it will indeed return to this state)

-1 always exists, indicating that the connection is lost by default, that is, the connection is disconnected. Usually received after the scale actively disconnects. We will unify this state to the disconnected state as soon as possible.

other problems

Is it possible to give the calculation method of body fat rate and other indicators?

The algorithm of these indicators belongs to my driver and cannot be provided.

What is the data stored on the scale

When the ordinary Bluetooth scale is not connected to the mobile phone during weighing, the scale will store the data in the scale body. When the Bluetooth is connected next time, the data will be transferred to the APP. This data is

distributed by the APP, and the assigned user can hand over to the SDK to calculate the complete data.

Can provide Bluetooth protocol

In principle, we will not provide Bluetooth protocol, if you really need, you can communicate with our business/sales

Why the data measured by the SDK is different from the data of Light Cow or Light Cow Health

Yolanda has several sets of algorithms, different algorithms that different devices may use.

In addition, we have a health question mechanism in our own APP, which will let the user answer 2 questions and use different algorithms according to the user's answer.

The algorithm mechanism in the SDK is usually a unified algorithm, immutable. In other words, the same scale uses our APP measurement and SDK measurement, the calculation data and weight are the same, there may be many differences. Therefore, our company does not recommend using our APP and SDK measurement data for comparison. Customers only need to be concerned about whether the given algorithm is suitable and stable.

How to troubleshoot problems in the online version after abnormal Bluetooth connection

First check the user's mobile phone model, if possible, try to use the same model as the customer to test.

Register the log output interface QNLogListener, record related logs (you can save log files), and send them to our developers for analysis.

After Bluetooth is searched and connected, the foot stays on the body fat scale for a few seconds, and then remove the foot, it will return the status code of the measurement completion (no data is returned in the method of returning data)

Why can we search for Bluetooth without any operation (no one is near the body fat scale within 10-30 minutes)

When weighing, if the APP is not connected, the scale will cache this data in it, and it will also broadcast for 5-30 minutes after the screen is closed (it varies depending on the production batch). After the APP is connected, the cached data can be received and displayed to the client. This function is used to solve the

problem that the user does not need a mobile phone when weighing. After weighing, sit on the sofa and turn on the mobile phone and receive the data just now.

Body index constant

constant	value	meaning
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
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
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

constant	value	meaning
TYPE_MUSCLE_CONTROL_INDEX	30	muscle control
TYPE_MUSCLE_MASS_RATE	31	muscle mass rate
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
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
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

Device type comparison table

constant	value	meaning	Bluetooth name	Remark
SCALE_BLE_DEFAULT	100	Ordinary Bluetooth Scale	QN-Scale or QN-Scale1	can be connect measure includin scannin connect data commur
SCALE_BROADCAST	120	Broadcast Scale	QN-S3	Cannot connect data is transmit Bluetoo broadca
SCALE_KITCHEN	130	Kitchen Scale	None	No conr is possil data is transmit Bluetoo broadca
SCALE_WSP	140	WSP Bluetooth scale	QN-Scale	Can be connect measure the use process includes scannin connect data commur
HEIGHT_SCALE	160	Height scale	QN-HS	Can be connect measure with hei data, the process includes scannin connect data commur

Body type comparison table

value	meaning
0	No body type value (generally no body fat rate is measured)
1	Stealth obesity
2	Lack of exercise
3	Lean type
4	Standard type
5	Lean muscle type
6	Obese
7	Fatty
8	Standard Muscle
9	Very muscular

error code

normal operation

Error constant	error code	meaning	Explanation	solution
OK	0	success	Successful operation	no

Initialization related errors

Error constant	error code	meaning	Explanation
ERROR_INVALIDATE_APP_ID	1001	app id is invalidate	APPID is invalid
ERROR_NOT_INIT_SDK	1002	please call the method "initSdk" first	The initSDK method is not called
ERROR_FIRST_DATA_FILE_URI	1003	the first data file uri is error, please provide the correct one	The Uri of the initial data file is incorrect
ERROR_PACKAGE_NAME	1004	the Android Package Name is Error ,Please Check it Or Contact the SDK Provider	Android's package name is incorrect
ERROR_BUNDLE_ID	1005	the IOS APP Bundle Id is Error ,Please Check it Or Contact the SDK Provider	The IOS Bundle Id is incorrect
ERROR_INIT_FILE	1006	The config file content is wrong ,Please provide the correct one	Incorrect initial configuration file

Bluetooth related errors

Error constant	error code	meaning
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
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 constant	error code	meaning
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

Method call error

Error constant	error code	meaning
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

Error constant	error code	meaning
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

Shared scale related errors

Error constant	error code	meaning	Explanation	solu
ERROR_CODER	1301	coder is error	QR code error	Plea con cust serv
ERROR_CODER_INVALID	1302	coder invalid	QR code invalid	Plea rege

Scan status definition

status	value	meaning	solution
FAIL_BLE_IS_OFF	1	Device Bluetooth is off	Turn on Bluetooth
FAIL_BLE_NOT_SUPPORT	2	The device does not support Bluetooth 4.0	Upgrade mobile phone system version or replace mobile phone
FAIL_BLE_INTERNAL_ERROR	3	internal error	Restart scanning
FAIL_BLE_NO_BLUETOOTH	4	Lack of Bluetooth permissions	Apply for Bluetooth permission in the manifest file
FAIL_BLE_NO_LOCATION	5	Missing location permission	Apply for location permission
FAIL_BLE_NONE_DEVICE	6	No Bluetooth devices are scanned	Restart phone Bluetooth

Scale status definition

status	value	meaning	
STATE_DISCONNECTED	0	not connected	
STATE_CONNECTED	1	connected	
STATE_CONNECTING	2	connecting	
STATE_DISCONNECTING	3	Disconnecting	
STATE_START_MEASURE	5	Measuring	
STATE_REAL_TIME	6	Measuring weight	
STATE_BODYFAT	7	Testing bioimpedance	
STATE_HEART_RATE	8	Testing heart rate	
STATE_MEASURE_COMPLETED	9	Measurement completed	
STATE_WIFI_BLE_START_NETWORK	10	WiFi Bluetooth dual-mode device starts to distribute network	
STATE_WIFI_BLE_NETWORK_SUCCESS	11	WiFi Bluetooth dual-mode device networked successfully	
STATE_WIFI_BLE_NETWORK_FAIL	12	WiFi Bluetooth dual-mode device failed to connect	

Scale event definition

event	value	meaning	B pr
EVENT_WIFI_BLE_START_NETWORK	1	WiFi Bluetooth dual-mode device starts to distribute network	co
EVENT_WIFI_BLE_NETWORK_SUCCESS	2	WiFi Bluetooth dual-mode device networked successfully	co
EVENT_WIFI_BLE_NETWORK_FAIL	3	WiFi Bluetooth dual-mode device networking failed	co
EVENT_REGIST_USER_SUCCESS	4	WSP scale exclusive, registered user success	co
EVENT_REGIST_USER_FAIL	5	WSP scale exclusive, registered user failed	co
EVENT_VISIT_USER_SUCCESS	6	WSP scale exclusive, successful access to users	co
EVENT_VISIT_USER_FAIL	7	WSP scale exclusive, failed to access user	co
EVENT_DELETE_USER_SUCCESS	8	Exclusive for WSP scales, successful user deletion	co
EVENT_DELETE_USER_FAIL	9	WSP scale exclusive, failed to access user	co

Test case

Excel version download address

> You can download the file with the right mouse button

Use case number	Use case title	priority	Preconditions	Test proced
SDK_01	Verify that it can be installed and run on the phone normally	high		1. Install th applicatio package on phone 2. Open th app
SDK_02	Verify that the Bluetooth shutdown prompt is normal	in		1. Turn off t phone's Bluetooth 2. Add connectio scale
SDK_03	Verify that the normal data can be measured and the IOS and Android measurement results are consistent	high	Must be measured barefoot	1. Install a open the S applicatio 2. Enter personal physical informatio such as hei 161cm, ag 1991-01-0 male 3. Use th same data Android ar IOS to wei the scale a measure o complete
SDK_04	Verify that the measurement indicators for wearing shoes are normal	high	Shoe measurement	1. Enter yo own perso data 2. Connect scale and measure complete d on the sca
SDK_05	Verify that the number of indicators is normal	high	Must be measured barefoot	1. Edit norr profile 2. Connec scale to measure complete d

Use case number	Use case title	priority	Preconditions	Test proced
SDK_06	Verify that switching between different gender measurement data is normal	high		1. Use ar Android pho edit your personal da and measu the comple data for eac the male a female sca 2. Change IOS phon again: use same data the scale measure piece of da for men ar women
SDK_07	Verify that the data for different height measurements is normal	high	Recommended height input range is 40cm--240cm, minimum is 40cm, maximum is 240cm	1. Enter 40 161cm, 240 respective other body d will not chan 2. Connect scale to th scale and measure complete d with the thr different hei above
SDK_08	Verify correct measurement data for different ages	high	Recommended age range 4 years old-80 years old	Android and use the sar data for measureme 1. Enter th birthday as years old, years old, years old, o informatio remains unchange 2. Connec scale with different ag measure complete d

Use case number	Use case title	priority	Preconditions	Test proced
SDK_09	Verify that switching between kg, lb, kg, and st units is normal.	high	Set the pound unit, the scale does not support the pound, the scale defaults kg, set the st unit, the scale does not support, the scale defaults lb	1. Edit pers data, swit between different ur 2. Conne scales in different ur
SDK_10	Verify that the measured data of the modified data is larger than the measured data before the modification	high		1. Edit pers data and connect to scale The scal measures complete pi of data 2. Modif personal informatio (the modifi information different fr the previo one), weigh the scale a to measure complete d

update record

2020-06-24

- Add WSP extension object
  - Issue user ID and set whether to display certain settings on WSP scale
- Add WSP extension field
  - Increase latitude and longitude

2020-06-09

- Add device type HEIGHT_SCALE
- Increase height data height
- Increase height mode data heightMode
- Store data to increase height data height

2020-03-03

- Add description of visitor mode
- Add control using visitor measurements isVisitor
- Add the identification of whether the stored data is complete isDataComplete
- Add Wsp device monitoring interface QNWspScaleDataListener

2020-02-07

- Add SCALE_WSP device type
- Add index and secret properties of QNUser class
- Add scale event callback
- Add QNWspConfig configuration class
- Add wsp connection method
- Add system Bluetooth status callback monitoring
- Add generateScaleData method
- Add Wsp device process description

2019-10-28

- Add service custom title and content settings setNotificationInfo

2019-10-08

- Add method to create kitchen scale object buildKitchenDevice
- Add QNBleKitchenDevice object
- Add onKitchenDeviceDiscover method
- modify convertWeightWithTargetUnit method the conversion of kitchen scale units
- Increase kitchen scale type

2019-07-19

-Add method to create broadcast scale object buildBroadcastDevice -Add method to create protocol handling class buildProtocolHandler

- Removed Wifi registered device method `registerWiFiBleDevice` -Add Bluetooth protocol proxy object QNBleProtocolDelegate -Add Bluetooth protocol processing object QNBleProtocolHandler -Change QNDataListener to QNScaleDataListener
- Move the `onScaleStateChange` method in QNBleConnectionChangeListener to QNScaleDataListener, which is specially used to indicate the change of weighing state. -[Organize device type, modify to only ordinary Bluetooth scale and broadcast scale]

2019-07-10

-Add Broadcast Scale Object QNBleBroadcastDevice -Add broadcast scale scanning listen callback method onBroadcastDeviceDiscover -Modify config setting class duration、allowDuplicates、unit field description -Add the error code ERROR_NOCOMPLETE_MEASURE of the broadcast scale to get the data -Add and modify the error code of broadcast scale unit ERROR_NOSUPPORT_MODIFY

2019-07-01

-Increase log monitoring

2019-06-03

-Add whether to check GPS permission method

2019-05-09

-Increase data feature identifier `hmac` -Increase body fat change control method `setFatThreshold`

2019-04-24

-Add Device Type `SCALE_WIFI_BLE` -Add WiFi configuration class -Add method to register WiFi Bluetooth dual-mode device `registerWiFiBleDevice` -Additional method of configuring WiFi `connectDeviceSetWifi` -Add attribute `mac` and method `buildStoreData` in stored data object -Add scale status code `STATE_WIFI_BLE_START_NETWORK`、`STATE_WIFI_BLE_NETWORK_SUCCESS`、`STATE_WIFI_BLE_NETWORK_FAIL` -Add method call error code `ERROR_DEVICE_TYPE`、`ERROR_WIFI_PARAMS`、`ERROR_MISS_STATE_LISTENER`

2019-03-22

-Increase clothes weight parameter clothesWeight

2019-03-14

-Modify Android-demo English address -Modify Android-demo Chinese Address -Add Android FAQ -Device class adds device type parameter deviceType

2019-03-14

-Add deviceType attribute -Modify effective time rule validTime -Add Device Type Form

2019-03-05

-Add decodeShareData method parameter validTime -Add error code ERROR_CODER、ERROR_CODER_INVALID

2019-02-25

-Increase Index

2019-01-27

- Add QNUtiils class

2018-10-25

-Callback for increasing power onGetElectric

2018-10-11

-Add athlete type field athleteType -User model added field athleteType -Add error code ERROR_USER_ATHLETE_TYPE

2018-08-27

-Increase connection timeout setting -Add Body Shape Comparison Table

2018-08-23 Add method to build SDK Bluetooth object

2018-08-13 Increase scan timeout setting

2018-06-20 Add device modeID