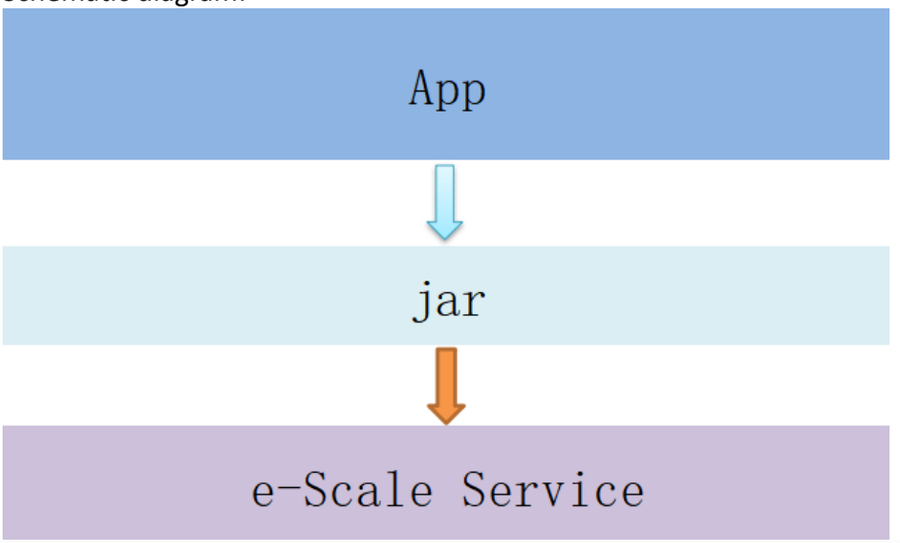
Electronic Scale 2.0 Developer document

1.Framework

It adopts built-in electronic scale service and external reference SDK jar package.
The functions and compatibility of the electronic scale are adapted by internal services, and the developer only needs to call the standard interface of the jar package.
The iterative maintenance of the built-in electronic scale service can be independently upgraded by the platform without additional development by the customer;
Schematic diagram:



2. Electronic scale information

2.1 Electronic scale parameter

Host Model	Swan 2 (Scale)
Measuring range	6kg≤Max≤15kg
n(Grade)	3000 (III)
Certified division value	2/5g
Working temperature (℃)	-10~+40℃

2.2 Quantitative correlation

#1. Zero

Every time the smart electronic scale is re-powered, the electronic scale will automatically record the initial zero point as a reference for subsequent weighing. The zero point range for a power-on restart is usually 10% of full scale. When the weight exceeds 10% of the full scale, the electronic scale will not be able to find the zero position. For example, for a 15kg electronic scale, if the weight of the pan exceeds 1.5kg when the machine is turned on, the electronic scale will not be able to find the zero position. If the weight is less than 1.5kg, the electronic scale will default to start weighing from zero.
We recommend that you empty all the items on the pan before powering it on.

#2. Manually clear/zero Settings
In daily weighing, if you need to return to zero, you can manually clear the zero back to the initial zero position. The range of manual zeroing is 2% of full scale. For a 15kg electronic scale, the range of manual zeroing is 298g. For example, after you clean the pan, the weight may show up as -0.004kg. When you click the Clear button, the weight display will return to "0.000kg".
Note: In the case of tare, the Clear button will not work.

#3. Peel/peel
During the weighing process, if the weight of the product packaging needs to be removed, it can be performed by peeling. The weight of the peel will affect the weighing range. For example, if a 15kg electronic scale is set with a tare of 5kg, then the remaining weighing range will be only 10kg.
Weighing and peeling: First put the package on the scale, click the peeling button, the weight of the package will be automatically recorded as tare weight;
Preset tare if you know the package weight, you can manually enter the tare value;
For multi-range electronic scales, the maximum peeling weight is usually Max1-e1. For example, for a 6/15kg electronic scale, the maximum tare is -5.998kg.
Note: The value of preset tare must be set accurately. First, the value of the preset tare must be less than the maximum tare; Secondly, the preset tare value must be set according to the correct index value. For example, for a 6/15kg (n=3000) electronic scale, the preset tare value of 6.005kg or 0.019kg is not correct.

#4. Net weight
Net weight refers to the value of the weight that the consumer should pay for the product. If the product comes with packaging, it is necessary to remove the packaging as tare first.
#5. Gross weight
Gross weight = tare + net weight

3、Electronic scale service usage

3.1 Downloading the SDK

Link Electronic scale sdk downloads

3.2 Initialize SDK

After importing the jar package


```
implementation files('libs/iMinEscale_SDK_V1.0.0.13_20250217.jar')
```

Connect the imin electronic scale service, when the service is connected, you can use the various functions in the SDK

```
ScaleManager scaleManager = ScaleManager.getInstance(context);
//iMin Scale service
scaleManager.connectService(newScaleManager.ScaleServiceConnection() {

    @Override
    public void onServiceConnected() {
        //Service binding
    }
    @Override
    public void onServiceDisconnect(){
        //Service unbundling
    }
}
```

Message warning tips

In order to strengthen the privacy protection policy, Android 11 introduced many changes and restrictions, including changes in package visibility, which will prevent third-party applications from successfully initializing the SDK

So you need to add the following to AndroidManifest.xml:

```
<manifest package=" Developer application package name ">
<queries>
<package android:name="com.imin.peripherservice" />
</queries>
...
</manifest>
```

In particular, it should be noted that this change in Android11 will only affect apps that upgrade targetSdkVersion=30, and apps that have not been upgraded will not be affected for the time being

3.3 Obtain weighing data

The method of quickly acquiring electronic scale data by means of interface callback

```
scaleManager.getData(new ScaleResult(){

    @Override
    public void getResult(int net,int tare，boolean isStable) throws RemoteException {
        //Return weighing result
    }

    @Override
    public void getStatus(boolean isLightWeight, boolean overload, boolean clearZeroErr, boolean calibrationErr) throws
RemoteException {
        //Return to weighing state
```

```
    }

    @Override
    public void getPrice(int net, int tare, int unit, String unitPrice, String totalPrice, boolean isStable, boolean isLightWeight) {
        //Return valuation result
    }

    @Override
    public void error(int errorCode) {
        //crc exception callback
    }
};
```

Weighing result parameter description

Parameter	Type	Implication	Description
net	int	Get weighed net	unit：g
tare	int	Get tare weighed	unit：g
isStable	boolean	Balance steady state	true:stable false：unstable

Description of weighing status parameters

Parameter	Type	Implication	Description
isLightWeight	boolean	Whether the scale is too light	true:too light false:normal
overload	boolean	Whether the scale is overloaded	true:too overloaded false: normal
clearZeroErr	boolean	Whether the scale has zero error	true:error false: normal
calibrationErr	boolean	Whether the scale has calibration errors	true:error false: normal

Weighing price parameter description

Parameter	Type	Implication	Description
net	int	Get weighed net	unit：g
tare	int	Get tare weighed	unit：g
unit	int	Unit of weight	0： g（default） 1 ： 100g 2： 500g 3： kg
unitPrice	String	The unit price set by the current pricing	Default is 0, no unit (customized by the service)
totalPrice	String	The total price calculated by the electronic scale	Current pricing Total price calculated based on weight and set unit price, default 0
isStable	boolean	Balance steady state	true: stable false: unstable
isLightWeight	boolean	Whether the scale is too light	true: underweight false: normal

CRC result

errorcode: 0 indicates that weighing data is normal. Non-0 indicates that weighing data is abnormal

[To the right] The recommended practice for displaying weighing results in business side software is:

The net weight and tare results are prominently displayed and there are signs indicating the current steady state, zero state and net weight state

Zero state: Tare + net weight = 0

Net weight status: Tare ≠ 0

When the weighing is abnormal, the callback result can indicate the current underload, overload and abnormal communication status

Underload status: You need to check whether the current weight is less than -20e. e is the minimum partition value, which can be obtained through other apis

3.4 Other interface names and descriptions

Function	implication	Instructions
String getServiceVersion()	Get the scale service version number	Returns the version number string of the current scale service
String getFirmwareVersion()	Obtain the firmware version number	Return the version number of the current firmware, such as 10034
void zero()	Zero clearing	The zero clearing operation can only clear the result deviation within 300 g
void tare()	Peel/clear the skin	When there is weight on the scale, it is peeled, and when there is no weight, it is peeled
void digitalTare(int i)	Digital peeling	Feed the peeling weight directly to the scale
void cancelGetData()	Unfetch data	Called when exiting the application, used in pairs with getData
int[] readAcceleData()	Read acceleration data	[0][1][2] are X, Y and Z direction data respectively
int readSealState()	Get seal status	0: normal 1: the seal is damaged
int getCalStatus()	Read the status of the calibration button switch	0: is not pressed. 1: is pressed
int[][] getCalInfo()	Read the electronic scale parameter information	The return value is a two-dimensional array of multiple ranges
void restart()	Restart electronic scale	For example, a range of 6/15kg e= 2/5g multi-range electronic scale will return [[6, 2],[15,5]]
void setUnitPrice(String unitPrice)	Set unit price	Set by the electronic scale service when calculating the price, will affect the returned

		pricing results support the calculation of two decimal places
String getUnitPrice()	Gets the currently set unit price	The unit price set by the electronic scale service when calculating the price
void setUnit(int unit)	Sets the unit of weight used in the price calculation	0: Weight by g 1: Weigh by 100g 2: Weigh by 500g 3: Weight by kg
int getUnit()	Gets the weight unit for the current price calculation	The price weight unit used in calculating the price by the electronic scale service will affect the result of calculating the total price
List<String> getCityAccelerations()	Get the city gravity accelerometer	For example, one piece of data: Anhui,97947
int setGravityAcceleration(int index)	Set city gravity acceleration	index corresponds to the footmark of the city's gravity acceleration list; If the return value is 0, the Settings are successful. Other Settings fail