# I. Introduction SDK

### copy witstecbletagsdk.aar package to app/libs directory.

app/build.gradle add the following code to the arr file to the project dependency

|  |
| --- |
| 1. repositories { 2. flatDir { 3. dirs'libs' 4. } 5. } 6. dependencies { 7. implementation (name :'waitstecbletagsdk\_v1.0.0', ext :'ar') 8. } |

# II. APPLICATION

### Application for APP First Start Permission

Add permission to declare to apply to AndroidManifest.xml file

Attention: In addition to the declaration in the AndroidManifest.xm, you also need to dynamically apply for these permissions when entering the main interface, otherwise you can not use the function properly

|  |
| --- |
| 1. <uses-permission android：name= android.permission.ACCESS\_NETWORK\_STATE<uses-permission android：name= 2. <uses-permission android：name= EPaperSdk.bleConnectDeviceMsgManager.release3<uses-permission android：name= 3. <uses-permission android：name= EPaperSdk.bleConnectDeviceMsgManager.release2<uses-permission android：name= 4. <uses-permission android：name= EPaperSdk.bleConnectDeviceMsgManager.release1<uses-permission android：name= 5. <uses-permission android：name= EPaperSdk.bleConnectDeviceMsgManager.release0<uses-permission android：name= 6. <uses-permission android：name= android.permission.READ\_EXTERNAL\_STORAGE<uses-permission android：name= |

### Bluetooth/position switch check

every time before calling the SDK interface, you need to check if the bluetooth and position switch are on at the same time. if there is no need to guide the user to turn on the bluetooth and position switch, if all are on state, you can do a series of operations.

|  |
| --- |
| 1. // Check whether the location is open 2. if (ScanUtil.isLocationEnabled (MainActivity.this)){ 3. // Get Bluetooth Manager 4. BluetoothAdapter mBluetoothAdapter =BluetoothAdapter.getDefaultAdapter(); 5. // Check if Bluetooth is open 6. if (! mBluetoothAdapter.isEnabled()){ 7. // If not, request Bluetooth 8. Intent enableBtIntent =new Intent (BluetoothAdapter.ACTION\_REQUEST\_ENABLE); 9. startActivityForResult (enableBtIntent ,1); 10. **else {}** 11. // Bluetooth operation 12. } 13. **else {}** 14. // Enter the Open Position Setup Interface 15. startActivityForResult (new intent (Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS),0); 16. } |

# SDK interface calls

### Initialization SDK

Note: Add initialization SDK, to the Application onCreate method

|  |
| --- |
| 1. EPaperSdk.init (this); |

### Write-offs

Call the following interface to log out of the application

1. EPaperSdk.destroy();

### Log Debugging Switch

The SDK log switch can be turned on in debug mode to see more log information to help locate the problem quickly. it is recommended to turn off the log switch in release mode.

## Scan device interface calls

1. EPaperSdk.setDebugMode (true);

Note: Open Bluetooth, search nearby device name "WITSTEC" Bluetooth device, and get MAC address and signal strength

### Startup scan

Directions: Start Bluetooth scanning

|  |
| --- |
| 1. EPaperSdk.BleScanManager.startScanNow (); |

* **Scanned results returned**

Description: The search results are returned once in 2.5 seconds, the return results are filtered, only return the electronic signature called "WITSTEC" device,

|  |
| --- |
| 1. // Obtain scan results 2. EPaperSdk.BleScanManager.getScanResult (new BleScanCallbackCompat(){ 3. @ Override @ 4. public void onScanResult (List <ScanData>deviceList){ 5. // Return device list List 6. } 7. }); |

### ScanData Object Parameter Description

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Type** | **Note** |
| address | String | Bluetooth mac address |
| name | String | Bluetooth Device Name |
| rssi | Int | Equipment signal values |

* **Stop scanning**

Description: Stop searching for nearby devices, no longer return scan results,

|  |
| --- |
| 1. EPaperSdk.BleScanManager.stopCycleScan (); |

## Connection Device Interface Call

Description: Connect the device, get the device details, disconnect, check if connected, reconnect.

### Connection equipment

Description: Enter the mac address to connect the device, and the device details will be returned after the connection device is successful.

1. EPaperSdk.connectMsgManager.connection (mac,new BleConnectionDeviceInfoCallback(){
2. @ Override @
3. public void onConnectionChange (StatusCode statusCode){
4. // Connection Status Callback, Update Connection Status
5. }
6. @ Override @
7. public void onConnectionError (ErrorCode errorCode){
8. // Call when connection encounters exception, return exception error code
9. }
10. @ Override @
11. public void onConnectionSuccess (DeviceInfomsg){
12. // The connection gets the device information successfully and returns the detailed device information
13. }
14. });

**The state StatusCode callback in the void onConnectionChange (statusCode) method**

|  |  |
| --- | --- |
| **State code** | **Note** |
| CONNECTION\_START | Connection equipment |
| CONNECTION\_SUCCESS | Device connection successfully |
| CONNECTION\_GET\_MSG\_START | Equipment information is being obtained |
| CONNECTION\_GET\_MSG\_SUCCESS | Access to device information |

The state ErrorCode callback in the void onConnectionError (errorCode) method

|  |  |
| --- | --- |
| **State code** | **Note** |
| ERROR\_BLE\_CONNECTION\_TIMEOUT | Connection Device Timeout |
| ERROR\_CONNECTION\_GET\_DEVICE\_MSG\_TIMEOUT | Failed to obtain device information |

**void onConnectionSuccess description of object parameters in DeviceInfo callback method**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Type** | **Note** |
| address | String | Bluetooth mac address |
| name | String | Bluetooth Device Name |
| version | String | Device Firmware Version |
| power | Int | Percentage of surplus equipment |
| deviceType | DeviceType | Type of equipment size |

**DeviceType enumeration object description**

|  |  |
| --- | --- |
| **Parameters** | **Note** |
| DEVICE\_042 | 4.2 inch equipment |
| DEVICE\_075 | 7.5 inch equipment |
| DEVICE\_029 | 2.9 inch equipment |

* **Disconnect**

Directions: Disconnect the device connection and need to reconnect after disconnection.

1. EPaperSdk.connectMsgManager.disConnection ();

* **Gets the device connection state**

Description: Return device connection status true/false.

1. boolean isConnection =EPaperSdk.connectMsgManager.isConnection ();

* **Reconnect, no callback**

description: incoming mac address, reconnect to the device.

1. EPaperSdk.bleConnectDeviceMsgManager.connection (mac);

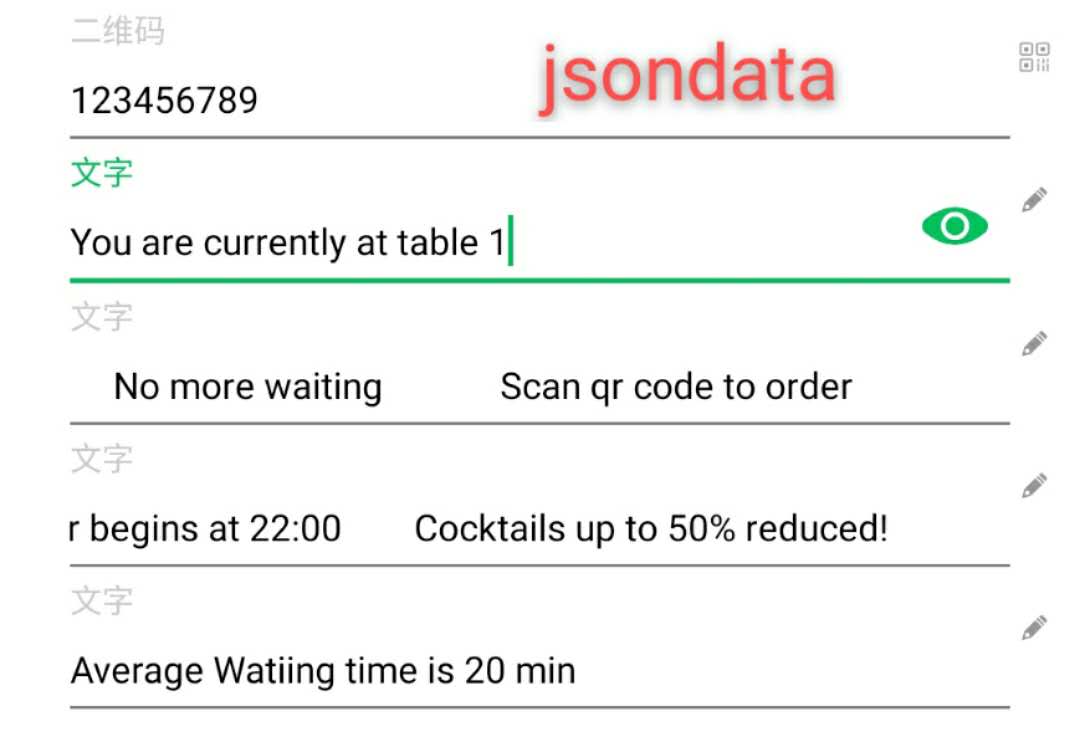
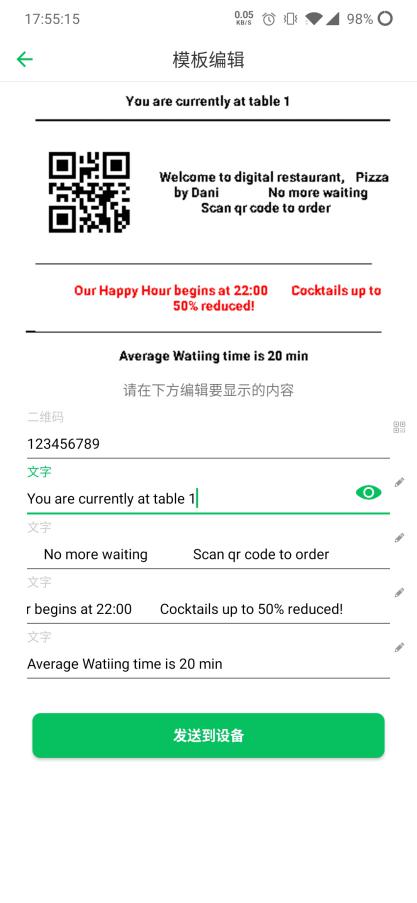
* **Release Connection Resources**

Description: Disconnect the device and release the resource

1. EPaperSdk.bleConnectDeviceMsgManager.release ();

* **Send template images to device interface calls**

Description: Because it is custom version SDK, so built-in template files, no need to import template files. For ease of understanding, the following is an example diagram of the display effect of a built-in template file, with five input boxes, for one QR code, four text. All you need to do is pass in the values in the input box as Json.



### Json Format Description

Json field description: red font is immutable value, green font is changeable value. and the values inside are sequential, data the first data corresponds to the QR code data, the second input corresponds to the first text input box,... from top to bottom, and so on.

Json data examples

{

"id"："001",

"mac"："mac address ",

"data"：[{

"type"："qrcode",

"content"：" the first input box for the value of the QR code"

},{

"type"："text",

"content"：" the value of the first text in the second input box"

},{

"type"："text",

"content"：" the value of the second text in the third input box"

},{

"type"："text",

"content"：" the value of the third text in the fourth input box"

},{

"type"："text",

"content"：" the value of the fourth text in the fifth input box"

}]

}

|  |  |  |
| --- | --- | --- |
| **Json parameters** | **Type** | **Note** |
| Id | String | Fixed to 001 |
| Mac | String | Equipment mac address |
| type | String | Type of input box (" qrcode "," textqrcode ") |
| context | String | Value of input box |

### Connection Device Send Template Picture

Description: connect the device to send the template to the electronic price tag, after the template is sent, the electronic price tag shows the template content, please make sure the battery power is more than 30% before sending.

Interface calls:

1. // Incoming parameter description: mac= device MAC address, inputStrJson mac= input box content in the format of the Json string
2. EPaperSdk.templateManager.connection (mac,inputStrJson,new BleTemplateCallback(){
3. // Connection Status Callback, Update Connection Status
4. @ Override @
5. public void onConnectionChange (StatusCode statusCode){
6. }
7. // Call when connection encounters exception, return exception error code
8. @ Override @
9. public void onConnectionError (ErrorCode errorCode){
10. }
11. // After sending and refreshing the electronic price tag is called, you can save the inputStrJson data to the local, facilitate the next automatic update input box content.
12. @ Override @
13. public void onSuccess (String inputStrJson){
14. }
15. });

**The state StatusCode callback in the void onConnectionChange (statusCode) method**

|  |  |
| --- | --- |
| **State code** | **Note** |
| CONNECTION\_START | Connection equipment |
| CONNECTION\_SUCCESS | Device connection successfully |
| TEMPLATE\_START\_SEND | Start sending template images to the device |
| TEMPLATE\_SEND\_LOADING | Is sending a template image to the device |
| TEMPLATE\_SEND\_SUCCESS | Send template pictures to device successfully |
| TEMPLATE\_REFRESH\_DEVICE | Refresh electronic price tag screen content |
| TEMPLATE\_REFRESH\_DEVICE\_SUCCESS | Electronic price tag |

The state ErrorCode callback in the void onConnectionError (errorCode) method

|  |  |
| --- | --- |
| **State code** | **Note** |
| ERROR\_BLE\_CONNECTION\_TIMEOUT | Connection Device Timeout |
| ERROR\_TEMPLATE\_SEND\_TIMEOUT | Send template picture timeout |

### Refresh template images

Note: This interface is called when the value of the input box entering the sending template interface changes, and the function is to preview the template display effect

1. Bitmap bitmap=EPaperSdk.templateManager.refreshTemplate (inputStrJson);

Return Bitmap, use ImageView control display, the specific use of example code as follows.

1. btn\_imageView.setImageBitmap (EPaperSdk.bleConnectDeviceMsgManager.release5(inputStrJson));

### Gets the input box list

Description: This interface is called when entering the sending template interface. The function is to parse the data in json format into a collection of objects, to generate input boxes, to determine the number of input boxes, to return the collection of value objects of input boxes, and to generate an input box for one object. .

1. List<BleTemplateItemData>imageBleMsgList =EPaperSdk.bleConnectDeviceMsgManager.release4();

BleTemplateItemData Object Value Description

|  |  |  |
| --- | --- | --- |
| **Request parameters** | **Type** | **Note** |
| id | String | Template Id |
| type | String | Type of input |
| context | String | Value of text type |
| qrCodeContext; | String | Value of two-dimensional code type |

## Status Code and Error Code

## StatusCode status tables

|  |  |
| --- | --- |
| **State value** | **Note** |
| CONNECTION\_START | Connection equipment |
| CONNECTION\_SUCCESS | Connection device successfully |
| CONNECTION\_GET\_MSG\_START | Equipment information is being obtained |
| CONNECTION\_GET\_MSG\_SUCCESS | Access to device information |
| TEMPLATE\_START\_SEND | Start sending templates to electronic price tag devices |
| TEMPLATE\_SEND\_LOADING | A template is being sent to an electronic price tag device |
| TEMPLATE\_SEND\_SUCCESS | Send template to electronic price tag device successfully |
| TEMPLATE\_REFRESH\_DEVICE | Refresh the electronic tag template picture |
| TEMPLATE\_REFRESH\_DEVICE\_SUCCESS | Refresh the electronic tag template picture |

* E rrorCode Error Error Code Matching Table

|  |  |
| --- | --- |
| **State value** | **Note** |
| ERROR\_PURVIEW\_NO\_OPEN\_BLE | No Bluetooth switch on |
| ERROR\_PURVIEW\_NO\_OPEN\_LOCATION | No positioning switch on |
| ERROR\_BLE\_CONNECTION\_TIMEOUT | Connection Device Timeout |
| ERROR\_CONNECTION\_GET\_DEVICE\_MSG\_TIMEOUT | Gets device information timeout |
| ERROR\_TEMPLATE\_SEND\_TIMEOUT | Send template images to electronic price tag device timeout |