Android SDK instructions

1. SDK structure

Service components and permissions

First add the service in the manifest file:

```
<service android:name="net.posprinter.service.PosprinterService"/>
```

Service components must be registered。
Need to add permissions:

```
<uses-feature android:required="true" android:name="android.hardware.usb.host" />

    <uses-permission android:name="android.hardware.usb.UsbAccessory" />

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<!--    <uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED"/>-->
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Tools:

The SDK encapsulates bill instructions, label instructions, 76-pin printing instructions and image processing methods.
Note instruction class: DataForSendToPrinterPOS58, DataForSendToPrinterPOS80
Label instruction class: DataForSendToPrinterTSC;
76 dot matrix printer instruction class: DataForSendToPrinterPOS76;
The methods in these classes are all static methods and can be called directly by the class name. The function of the methods in these tool classes is to process various types of printer commands into byte[] for sending directly to the printer.

The corresponding search method of instruction and programming manual:
For example: open the index.html webpage of the SDK webpage command, then click on the DataForSendToPrinterPOS58 class, choose a method at random, and click on it. For example, CancelChineseCharModel, you can see the description and return value type, and then find the logo of this instruction: FS, and then copy the logo to the programming manual to find the corresponding description and parameter values.

2. How to use

2.1. Bind service to get IMbinder object

```
public static IMyBinder binder;

ServiceConnection conn=new ServiceConnection() {

    @Override

    public void onServiceConnected(ComponentName name, IBinder service) {

        binder= (IMyBinder) service;

    }


    @Override

    public void onServiceDisconnected(ComponentName name) {


    }

};

Intent intent=new Intent(this, PosprinterService.class);

bindService(intent,conn,BIND_AUTO_CREATE);
```

The Binder class is a class for managing printers. The binder encapsulates the methods of connecting, sending, and receiving data.

2.2, the main method of binder (printer binder)

Connect the printer: There are three connection methods, Bluetooth, network and USB.
Connection method:

```
/**

 *Connect to the network port

 *@param UiExecute The implementation class of the execute interface, the implementation *method is
executed in the UI thread

 *@param ethernetIP Printer ip address

 *@param ethernetPort Printer port number

 **/

void ConnectNetPort(String ethernetIP,int ethernetPort,TaskCallback callback);

/**

 *Connect the Bluetooth port

 *@param UiExecute The implementation class of the execute interface, the implementation *method is
executed in the UI thread

 *@param bluetoothID Bluetooth name
```

```
 **/
void ConnectBtPort(String bluetoothID,TaskCallback callback);
/**
 *Connect to the USB port
 *@param UiExecute The implementation class of the execute interface, the implementation *method is executed in the UI thread
 *@param context Context
 *@param usbPathName usbPathName
**/
void ConnectUsbPort[Context context,String usbPathName,TaskCallback callback);
```

send data：

```
//Send data to the printer
//@param dataThe data received by the printer is byte[], byte array
void WriteSendData(TaskCallback callback,ProcessData processData);
```