

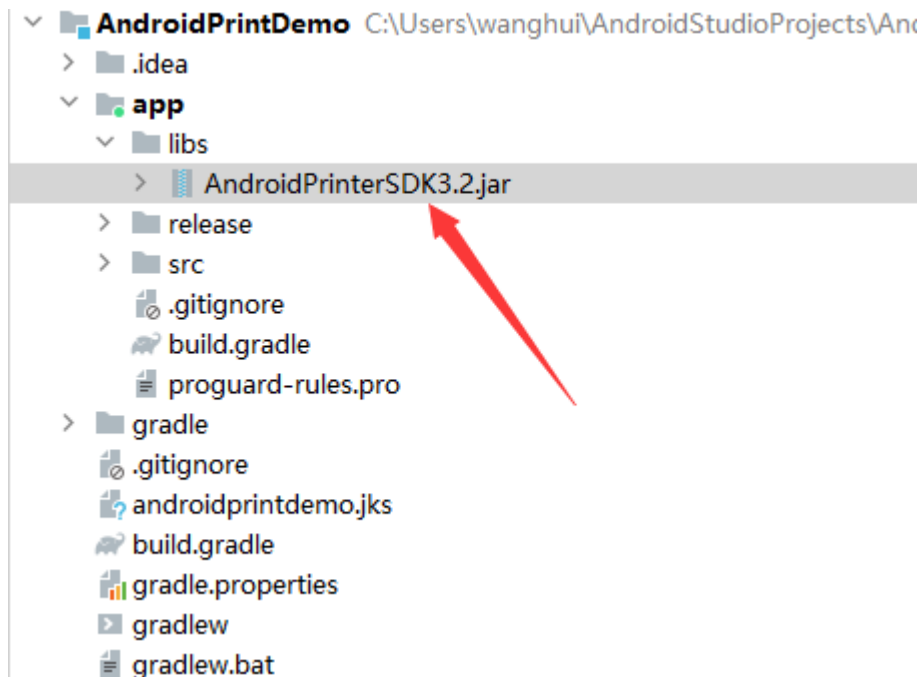
Android integration documentation

Introduction:

The receipt printer adopts the ESC/POS commands common in the printer industry, and the SDK encapsulates the device discovery, connection, and data sending functions of Bluetooth, USB, and WIFI, which can be integrated more quickly. Commonly used ESC/POS commands have been encapsulated, such as font size, bold, underline, center; barcode printing; picture printing, etc.

First Step:

Import the "AndroidPrinterSDK3.2.jar" jar package into the libs directory and import it in build.gradle.



```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'com.google.android.material:material:1.2.1'
    //implementation project(':android-printer-sdk')
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
}
```

Second Step:

According to the interface type of the printer, select Bluetooth, USB or WIFI for integration, and copy the corresponding files. The following files are in the printed case and are for reference only. Developers can implement it by themselves.

1.Files to be copied for integrated Bluetooth communication:



BluetoothDeviceList.java



BluetoothOperation.java

2.Files to be copied with integrated USB connection:



UsbOperation.java



UsbDeviceList.java

3.Files to be copied for integrated WIFI connection:



IpAddressEdit.java



WifiOperation.java

4.The final master file that needs to be copied



MainActivity.java



IPrinterOpertion.java

After copying the above, follow the code comments to discover, connect and send data.

Third Step:

After the second step above is completed, communication with the device is already possible. The following is to control the content sent, as shown in the figure below:

```
// 初始化打印机
// Initialize the printer
mPrinter.init();

// 横线和纵向都放大字体一倍
// Double the font size for both horizontal and vertical lines
mPrinter.setCharacterMultiple(1, 1);

// 打印文本
// Print text
mPrinter.printText("Print Text");

// 打印并换行
// Print and newline
mPrinter.setPrinter(Command.PRINT_AND_NEWLINE);

// 恢复默认字体大小
// Restore the default font size
mPrinter.setCharacterMultiple(0, 0);

// 字体加粗并设置下划线
// Bold font and underline
mPrinter.setPrintModel(false, true, false, false, true);

// 打印文字
// Print Text
mPrinter.printText("Print Text");

// 打印并走纸 n 行
// Print and feed paper n lines
mPrinter.setPrinter(Command.PRINT_AND_WAKE_PAPER_BY_LINE, 2);
```

```
public static void printText(Resources resources, PrinterInstance mPrinter) {  
    // 初始化打印机  
    // Initialize the printer  
    mPrinter.init();  
    // 横线和纵向都放大字体一倍  
    // Double the font size for both horizontal and vertical lines  
    mPrinter.setCharacterMultiple( x: 1, y: 1);  
    // 打印文本  
    // Print text  
    mPrinter.printText( content: "Print Text");  
    // 打印并换行  
    // Print and newline  
    mPrinter.setPrinter(Command.PRINT_AND_NEWLINE);  
    // 恢复默认字体大小  
    // Restore the default font size  
    mPrinter.setCharacterMultiple( x: 0, y: 0);  
    // 字体加粗并设置下划线  
    // Bold font and underline  
    mPrinter.setPrintModel( smallFont: false, isBold: true, isDoubleHeight: false, isDoubleWidth: false, isUnderLine: true);  
    // 打印文字  
    // Print Text  
    mPrinter.printText( content: "Print Text");  
    // 打印并走纸n行  
    // Print and feed paper n lines  
    mPrinter.setPrinter(Command.PRINT_AND_WAKE_PAPER_BY_LINE, value: 2);  
}
```