# **Puty Android 打印 SDK 接入文档**

## **1. 概述**

本文档旨在指导开发者如何接入 Puty 打印 SDK，并实现连接和打印功能。通过本 SDK，开发者可以轻松地与打印机设备进行交互，完成标签打印等操作。

## **2. 环境准备**

开发工具：Android Studio

设备要求：支持蓝牙的 Android 设备

权限要求：在 AndroidManifest.xml 中添加以下权限

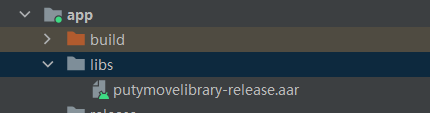
（已在AAR包AndroidManifest中导入，用户接入无需再次导入）：

<!-- Android 11（API 级别 30）及以下版本蓝牙权限-->  
<!-- 用于连接到已配对的蓝牙设备 -->  
<uses-permission android:name="android.permission.BLUETOOTH" />  
<!-- 用于发现和配对蓝牙设备 -->  
<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
<!--用于蓝牙扫描，因为扫描结果可以用于定位-->  
<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
  
<!--Android 12（API 级别 31）及以上版本蓝牙权限-->  
<!-- 用于扫描附近的蓝牙设备 -->  
<uses-permission  
 android:name="android.permission.BLUETOOTH\_SCAN"  
 android:usesPermissionFlags="neverForLocation"  
 tools:targetApi="s" />  
<!--用于使当前设备可被其他蓝牙设备检测到-->  
<uses-permission android:name="android.permission.BLUETOOTH\_ADVERTISE" />  
<!--用于与已配对的蓝牙设备通信-->  
<uses-permission android:name="android.permission.BLUETOOTH\_CONNECT" />

## **3. 导入 SDK**

### **3.1 添加 AAR 包**

将 putymovelibrary-release.aar 文件放入项目的 libs 文件夹中；



### 3.2 添加依赖

在 build.gradle 文件中添加以下依赖：

android **{**

repositories **{** flatDir **{** dirs 'libs'  
 **}  
 }**

}

dependencies **{** implementation fileTree(include: ['\*.jar'], dir: 'libs')  
 implementation 'androidx.appcompat:appcompat:1.1.0'  
 implementation 'androidx.recyclerview:recyclerview:1.0.0'  
 implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0'  
 //二维码库  
 implementation 'cn.bingoogolapple:bga-qrcode-zxing:1.3.6'  
 //Kotlin协程  
 implementation "org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.3"  
 //打印sdk  
 implementation(name: 'putymovelibrary-release', ext: 'aar')  
**}**

## **4. 初始化 SDK**

//初始化打印机  
PrinterInstance.*init*(Context context);

## **打印**

### **5.1单张图片无预览打印示例：**

注意：执行打印时，如果检测到未连接，会自动弹出搜索连接对话框

//测试图片  
Bitmap bitmap = *loadBitmapFromAssets*(this, "label\_40\_90.png");  
//打印位图 density 浓度 1~15 0:浓度跟随机器 pagerType 纸张类型：-1:跟随机器 0:连续纸 1:黑标纸 2:间隙纸 3:定位孔紙 offsetX 水平偏移 offsetY 垂直偏移  
LabelPrint.*printBitmap*(this, bitmap, PrintDirection.*DIRECTION\_90*, 0, -1, 0, 0, new PrintStatusCallback() {  
 @Override  
 public void onPrintError(String error) {  
 Toast.*makeText*(MainActivity.this, error, Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 public void onPrintCompleted() {  
 Toast.*makeText*(MainActivity.this, "打印完成", Toast.*LENGTH\_SHORT*).show();  
 }  
});

### **5.2单张图片预览打印示例：**

//测试图片  
Bitmap bitmap = *loadBitmapFromAssets*(this, "label\_40\_90.png");  
//设置数据源  
PrintPreviewActivity.*setBitmap*(bitmap);  
Bundle bundle = new Bundle();  
//设置打印模式  
bundle.putInt(PrintMode.*MODE\_KEY*, PrintMode.*BITMAP\_PRINT*);

//设置打印方向  
bundle.putInt(PrintDirection.*PRINT\_DIRECTION\_KEY*, PrintDirection.*DIRECTION\_90*);  
//打开打印预览页面  
startActivity(PrintPreviewActivity.class, bundle);

### **5.3图片批量无预览打印示例：**

List<Bitmap> bitmaps = new ArrayList<>();  
for (int i = 0; i < 2; i++) {  
 //生成测试位图  
 Bitmap bitmap = *loadBitmapFromAssets*(this, "label\_40\_90.png");  
 bitmaps.add(bitmap);  
}  
//打印位图 density 浓度 1~15 0:浓度跟随机器 pagerType 纸张类型：-1:跟随机器 0:连续纸 1:黑标纸 2:间隙纸 3:定位孔紙 offsetX 水平偏移 offsetY 垂直偏移  
LabelPrint.*printBitmaps*(this, bitmaps, PrintDirection.*DIRECTION\_90*, 0, -1, 0, 0, new PrintStatusCallback() {  
 @Override  
 public void onPrintError(String error) {  
 Toast.*makeText*(MainActivity.this, error, Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 public void onPrintProgress(int pageIndex) {  
 if (bitmaps.size() > 1) {  
 Toast.*makeText*(MainActivity.this, "打印第" + pageIndex + "张", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 @Override  
 public void onPrintCompleted() {  
 Toast.*makeText*(MainActivity.this, "打印完成", Toast.*LENGTH\_SHORT*).show();  
 }  
});

### **5.4图片批量预览打印示例：**

//生成测试位图  
Bitmap bitmap1 = *loadBitmapFromAssets*(this, "label\_40x90.png");  
Bitmap bitmap2 = *loadBitmapFromAssets*(this, "label\_40x90.png");  
List<Bitmap> bitmaps = new ArrayList<>();  
bitmaps.add(bitmap1);  
bitmaps.add(bitmap2);  
//设置数据源  
PrintPreviewActivity.*setBitmaps*(bitmaps);  
Bundle bundle = new Bundle();  
//设置打印模式  
bundle.putInt(PrintMode.*MODE\_KEY*, PrintMode.*BITMAPS\_PRINT*);

//设置打印方向  
bundle.putInt(PrintDirection.*PRINT\_DIRECTION\_KEY*, PrintDirection.*DIRECTION\_90*);  
//打开打印预览页面  
startActivity(PrintPreviewActivity.class, bundle);

### 5.5标签无预览打印示例：

注意：执行打印时，如果检测到未连接，会自动弹出搜索连接对话框

List<Label> labels = label\_50\_30();  
//打印标签 density 浓度 1~15 0:浓度跟随机器 pagerType 纸张类型：-1:跟随机器 0:连续纸 1:黑标纸 2:间隙纸 3:定位孔紙 offsetX 水平偏移 offsetY 垂直偏移  
LabelPrint.*printLabels*(this, labels, 0, -1, 0, 0, new PrintStatusCallback() {  
 @Override  
 public void onPrintError(String error) {  
 Toast.*makeText*(MainActivity.this, error, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onPrintProgress(int pageIndex) {  
 if (labels.size() > 1) {  
 Toast.*makeText*(MainActivity.this, "打印第" + pageIndex + "张", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 @Override  
 public void onPrintCompleted() {  
 Toast.*makeText*(MainActivity.this, "打印完成", Toast.*LENGTH\_SHORT*).show();  
 }  
});

### 5.6标签预览打印示例：

Bundle bundle = new Bundle();  
//设置打印模式  
bundle.putInt(PrintMode.*MODE\_KEY*, PrintMode.*LABEL\_PRINT*);  
//设置数据源  
PrintPreviewActivity.*setLabels*(label\_50\_30());  
//打开打印预览页面  
startActivity(PrintPreviewActivity.class, bundle);

### 5.7报文无预览打印示例：

//报文打印 density 浓度 1~15 0:浓度跟随机器 pagerType 纸张类型：-1:跟随机器 0:连续纸 1:黑标纸 2:间隙纸 3:定位孔紙 offsetX 水平偏移 offsetY 垂直偏移  
LabelPrint.*printLabelsByXmlData*(this, xml\_40\_90, 0, -1, 0, 0, new PrintStatusCallback() {  
 @Override  
 public void onPrintError(String error) {  
 Toast.*makeText*(MainActivity.this, error, Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 public void onPrintCompleted() {  
 Toast.*makeText*(MainActivity.this, "打印完成", Toast.*LENGTH\_SHORT*).show();  
 }  
});

### 5.8报文预览打印示例：

Bundle bundle = new Bundle();  
//设置打印模式  
bundle.putInt(PrintMode.*MODE\_KEY*, PrintMode.*XML\_DATA\_PRINT*);  
//设置数据源  
PrintPreviewActivity.*setXmlData*(xml\_40\_90);  
//打开打印预览页面  
startActivity(PrintPreviewActivity.class, bundle);

## 打印数据生成

### 6.1 标签打印数据

*/\*\*  
 \* 创建50x30标签模板  
 \** ***@return*** *\*/*private List<Label> label\_50\_30() {  
 List<Label> labels = new ArrayList<>();  
 //创建标签 name:标签名 w:标签宽度，单位：毫米 h:标签高度，单位：毫米；  
 Label label = new Label("50\*30", 50, 30);  
 //预览图背景  
 //label.backGroundImage = getImageFromAssetsFile(context, "45\_100.png");  
 //标签打印方向 0：水平打印 1:旋转90打印  
 label.printDirect = 0;  
  
 //创建一维码  
 Barcode1DElement barCodeEl = new Barcode1DElement(this, "123456789", 2 \* 8, 2 \* 8, 20 \* 8, 10 \* 8);  
 //一维码字号  
 barCodeEl.fontSize = 24;  
 //文本加粗  
 barCodeEl.fontBlod = 1;  
 label.Elements.add(barCodeEl);  
  
 //创建二维码元素 单位：像素 注意：10f \* 8中的10表示10毫米  
 Barcode2dElement qrCodeEl = new Barcode2dElement(this,"H250331010957", 32f \* 8, 2f \* 8, 12 \* 8, 12 \* 8);  
 label.Elements.add(qrCodeEl);  
  
 //创建线条元素  
 LineElement lineEl = new LineElement(this, 2 \* 8f, 16 \* 8, 46 \* 8, 0.5f \* 8);  
 //旋转角度  
 lineEl.rate = 0;  
 //lineEl.lineHeight = 1 \* 8;  
 label.Elements.add(lineEl);  
  
 //创建文本元素 content:文本内容 w:文本宽度 h：文本高度 单位：像素 注意：37\*8 中的37表示37毫米  
 TextElement contentEl = new TextElement(this, "床前明月光；疑是地上霜。\n举头望明月，低头思故乡。", 2f \* 8, 20f \* 8, 46 \* 8, 16 \* 8);  
 //对齐方式 0：左对齐 1：居中对齐 2:右对齐 3：分散对齐  
 contentEl.textMode = 1;  
 //是否加粗 0：不加粗 1：加粗  
 contentEl.fontBlod = 0;  
 //设置字号 单位像素  
 contentEl.fontSize = 28;  
 //添加元素到标签  
 label.Elements.add(contentEl);  
 //添加标签到标签集合  
 labels.add(label);  
  
 return labels;  
}

### 6.2报文打印数据

*/\*\*  
 \* ODF设备/光交箱/光分纤箱/光终端盒标签  
 \*/*private final String xml\_40\_90 = "<?xml version=\"1.0\" encoding=\"utf-8\" ?>" +  
 "<Data>" +  
 "<Print>" +  
 "<EntityTypeId>1003</EntityTypeId>" +  
 "<Text>机房信息</Text>" +  
 "<Text>机房名称:济南四里村4层南传输</Text>" +  
 "<Text>是否两核两关:是</Text>" +  
 "<Text>机房类型:移网核心机房</Text>" +  
 "<Text>机房负责人(空间责任人):王中楹</Text>" +  
 "<Text>联系方式:15615311791</Text>" +  
 "<Text>动力资源责任人:张兴强</Text>" +  
 "<Text>动力资源责任人联系方式:15615319852</Text>" +  
 "<Code>123456789</Code>" +  
 "</Print>" +  
 "</Data>";

## **其他常用API**

### 判断蓝牙设备是否已连接

boolean isConnected = LabelPrint.*isConnected*(String macAddress);

### 7.2连接蓝牙设备

public static void connectPrinter(final Context context, String macAddress, final OnBluetoothConnectListener onBluetoothConnectListener)；

### 7.2断开蓝牙设备连接

LabelPrint.*closeConnection*();

### 7.3取消&中断打印

LabelPrint.*cancelPrint*();

### 7.4创建预览图

/\*\*  
 \* 根据标签创建预览图  
 \* @param label 打印标签  
 \* @param degrees 旋转角度 常用值 0,-90  
 \* @return 预览图  
 \*/

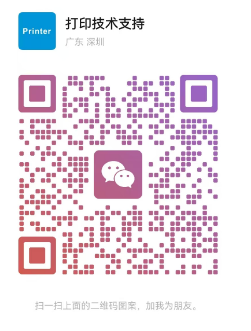
Bitmap bitmap = LabelPrint.*createPreviewBitmap*(Label label, int degrees);

## **9. 示例代码**

完整的示例代码请参考 Demo：PutySDK

## **10. 联系支持**

如需进一步支持，请联系技术支持团队：



微信：YF3028370006

电话：+86 136 3297 6564