

GpService.aidl 说明

包名 : com.gprinter.aidl

文件名 : GpService.aidl

PrinterId : 取值为 0、1、2

说明 : 打印机的序号, Gplink 插件可以同时连接 3 台打印机 : Printer001、Printer002、Printer003, 依次编号为 0、1、2

int openPort(**int** PrinterId, **int** PortType, **String** DeviceName, **int** PortNumber)

功能 : 用来打开端口, 连接过程为异步的, 可以通过注册 "action.connect.status" 广播来接收端口连接的状态;

GpDevice. STATE_NONE = 0; //连接断开

GpDevice. STATE_LISTEN = 1; //监听状态

GpDevice. STATE_CONNECTING = 2; //正在连接

GpDevice. STATE_CONNECTED = 3; //已连接

GpDevice. STATE_INVALID_PRINTER = 4; //无效的打印机

GpDevice. STATE_VALID_PRINTER = 5; //有效的打印机

具体操作可以参考 附带的例程

PrinterId:打印机序号

PortType:端口类型

PortParameters.USB = 2

PortParameters.ETHERNET = 3

PortParameters.BLUETOOTH = 4

DeviceName: 设备名

如果端口为 USB, 则为 USB 设备的 DeviceName

如果端口为 ETHERNET, 则为 IP 地址

如果端口为 BLUETOOTH, 则为蓝牙 Mac 地址

PortNumber : 端口号

如果端口为 USB, 则为 0

如果端口为 ETHERNET, 则为端口号, 打印机一般为 9100 端口

如果端口为 BLUETOOTH, 则为 0

返回值 : 错误值, 如果端口已连接返回 ERROR_CODE.DEVICE_ALREADY_OPEN

错误值说明

```
enum ERROR_CODE {
    SUCCESS, //正常
    FAILED, //失败
    TIMEOUT, // 超时
    INVALID_DEVICE_PARAMETERS, //无效的参数
    DEVICE_ALREADY_OPEN, //端口已经打开
    INVALID_PORT_NUMBER, //无效的端口号
    INVALID_IP_ADDRESS, //无效的 ip 地址
    INVALID_CALLBACK_OBJECT, //无效的回调
    BLUETOOTH_IS_NOT_SUPPORT, //设备不支持蓝牙
    OPEN_BLUETOOTH, //请打开蓝牙
    PORT_IS_NOT_OPEN, //端口未打开
    INVALID_BLUETOOTH_ADDRESS, //无效的蓝牙地址
    PORT_IS_DISCONNECT //端口连接断开
}
```

int closePort(**int** PrinterId)

功能：关闭端口

参数：PrinterId：打印机的序号

返回值：错误值。

参照错误值说明

int getPrinterConnectStatus(**int** PrinterId)

功能：获取打印机连接状态

参数：PrinterId：打印机的序号

返回值：状态值。

GpDevice. STATE_NONE = 0; //连接断开

GpDevice. STATE_LISTEN = 1; //监听状态

GpDevice. STATE_CONNECTING = 2; //正在连接

GpDevice. STATE_CONNECTED = 3; //已连接

int printTestPage(**int** PrinterId)

功能：打印测试页

参数：PrinterId：打印机的序号

返回值：错误值。

参照错误值说明

void queryPrinterStatus(**int** PrinterId, **int** Timesout, **void** requestCode)

功能：查询打印机状态

参数：PrinterId：打印机的序号

Timesout:接收超时时间 ms

因为蓝牙返回数据会有延时，所以此时间需设置为 500-1000ms，根据设备和环境而定

USB 和 WIFI 返回数据较快，一般设置为 100-500ms 左右

返回值：无

接收方法: 以广播的形式返回状态

广播的 action -> GpCom.ACTION_DEVICE_REAL_STATUS

广播中的 extra：

```
int requestCode =  
intent.getIntExtra(GpCom.EXTRA_PRINTER_REQUEST_CODE, -1);  
int status =  
intent.getIntExtra(GpCom.EXTRA_PRINTER_REAL_STATUS, 16);
```

状态值说明：

```
STATE_NO_ERR = 0; //正常  
STATE_OFFLINE = 0x1; //脱机  
STATE_PAPER_ERR = 0x2; //缺纸  
STATE_COVER_OPEN = 0x4; //开盖  
STATE_ERR_OCCURS = 0x8; //过热 错误
```

具体查看 MainActivity 中的广播接收器

int getPrinterCommandType(**int** PrinterId)

功能：查询打印机指令类型

参数：PrinterId：打印机的序号

返回值：打印机的指令类型

```
ESC_COMMAND = 0;
```

TSC_COMMAND = 1;

int sendEscCommand(**int** PrinterId,String b64)

功能：发送 ESC 指令，发送此命令时，确保打印机处于票据模式，否则发送无效

参数：PrinterId：打印机的序号

String b64：发送的票据数据

返回值：错误值 同 ERROR_CODE

int sendTscCommand(**int** PrinterId,String b64)

功能：发送 TSC 指令，发送此命令时，确保打印机处于标签模式，否则发送无效

参数：PrinterId：打印机的序号

String b64：发送的标签数据

返回值：错误值 同 ERROR_CODE

void isUserExperience(**boolean** userExperience)

功能：参与用户体验

参数：userExperience：是否参与

返回值：void