

uniapp 安卓热敏打印机UTS原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
状态码	1.3
插件方法	2
USB方式	2.1
获取所有的usb设备	2.1.1
获取所有USB打印机设备	2.1.2
usb连接	2.1.3
画布打印	2.2
画线	2.2.1
画边框	2.2.2
画文字	2.2.3
画区域块文字	2.2.4
画条形码	2.2.5
画区域块条形码	2.2.6
画二维码	2.2.7
画图	2.2.8
区域画图	2.2.9
打印	2.2.10
TSPL指令打印	2.3
获取TSPL纸张宽度	2.3.1
设置TSPL纸张宽高	2.3.2
页设置TSPL	2.3.3
TSPL标签打印线	2.3.4
TSPL标签打印边框	2.3.5
TSPL标签打印文本	2.3.6
TSPL标签区域打印文本	2.3.7
TSPL标签打印二维码	2.3.8
TSPL标签打印条码	2.3.9
TSPL标签打印图片	2.3.10
TSPL标签区域图片打印	2.3.11
TSPL标签打印	2.3.12
TSPL标签翻转区域打印	2.3.13

TSPL标签擦除区域	2.3.14
TSPL标签设置	2.3.15
TSPL设置标签间的缝隙大小	2.3.16
TSPL标签蜂鸣器	2.3.17
TSPL标签发送指令	2.3.18
TSPL标签开启钱箱	2.3.19
通用方法	2.4
关闭连接	2.4.1
发送字节数据	2.4.2
打印文本	2.4.3
打印图片带压缩	2.4.4
打印图片带等级	2.4.5
打印灰色图片	2.4.6
针打图形	2.4.7
打印表格	2.4.8
打印条码和二维码	2.4.9
获取纸张宽度	2.4.10
设置纸张宽度	2.4.11
页设置	2.4.12
初始化打印机	2.4.13
设置字体样式	2.4.14
设置打印属性	2.4.15
获取编码集	2.4.16
设置编码集	2.4.17
打开钱箱	2.4.18
切纸	2.4.19
打印机是否缺纸	2.4.20
打印机纸张是否将用尽	2.4.21
获取打印机状态	2.4.22
获取打印结果状态	2.4.23
十六进制字符串转字节数组	2.4.24
字符串转16进制	2.4.25

使用方法

介绍

安卓热敏打印机UTS原生插件支持大部分热敏打印机，插件提供了USB连接方式，画布打印，TSPL指令打印，通用打印等，可以打印出想打印的内容，插件功能比较多，具体使用可参考示例项目

联系作者

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=22445>

示例文件下载

1. 插件首页下载（推荐）

通过“使用 HBuilderX 导入示例项目”可直接创建uniapp项目

2. 本地下载，[点击这里](#)即可下载

演示程序

使用方法

加群下载演示程序



uniapp leven 系列...

群号: 106179987



用法

在需要使用插件的页面加载以下代码

```
import * as module from "@uni_modules/leven-uts-printer"
```

页面内容

页面内容比较复杂，请参考示例项目

更新日志

2025-03-05

首次发布

状态码

打印机连接状态

状态码	描述
-100	未知错误
-101	打印机连接失败
-102	打印机连接关闭
-103	未找到打印机设备
-104	没有usb权限
-105	usb插入
-106	usb拔出

USB方式

获取所有的usb设备

方法名

getAllUsbDevices

用法

- 用法如下：

```
module.getAllUsbDevices(res => {  
  console.log(res)  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "deviceName": "/dev/bus/usb/003/002",  
        "deviceClass": 239,  
        "productName": "RXGD-CAMERA",  
        "vendorId": 3034,  
        "productId": 12337,  
        "deviceId": 3002  
      },  
      {  
        "deviceName": "/dev/bus/usb/001/005",  
        "deviceClass": 0,  
        "productName": "USB OPTICAL MOUSE ",  
        "vendorId": 10077,  
        "productId": 2982,  
        "deviceId": 1005  
      },  
      {  

```


获取所有的usb设备

```
        "deviceName": "/dev/bus/usb/001/004",
        "deviceClass": 0,
        "productName": "TOT2D PRODUCT DEFHID+KBW",
        "vendorId": 44953,
        "productId": 32772,
        "deviceId": 1004
    },
    {
        "deviceName": "/dev/bus/usb/001/003",
        "deviceClass": 0,
        "productName": "REG0 PRINTER",
        "vendorId": 1155,
        "productId": 22304,
        "deviceId": 1003
    },
    {
        "deviceName": "/dev/bus/usb/001/006",
        "deviceClass": 239,
        "productName": "Android",
        "vendorId": 11388,
        "productId": 24581,
        "deviceId": 1006
    }
]
},
"message": "",
"code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.list	Array[Object]	usb设备列表
data.list.productId	Integer	产品id
data.list.vendorId	Integer	供应商id
data.list.productName	String	产品名称
data.list.deviceClass	Integer	设备序号
data.list.devicesName	String	设备名称
data.list.deviceId	Integer	设备id

获取所有的usb设备

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

获取所有USB打印机设备

方法名

getAllUsbPrinterDevices

用法

- 用法如下：

```
module.getAllUsbPrinterDevices(res => {  
  console.log(res)  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "deviceName": "/dev/bus/usb/001/003",  
        "deviceClass": 0,  
        "productName": "REGO PRINTER",  
        "vendorId": 1155,  
        "productId": 22304,  
        "deviceId": 1003  
      }  
    ]  
  },  
  "message": "",  
  "code": 0  
}
```

- 回调说明：

获取所有USB打印机设备

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.list	Array[Object]	usb打印机设备列表
data.list.productId	Integer	产品id
data.list.vendorId	Integer	供应商id
data.list.productName	String	产品名称
data.list.deviceClass	Integer	设备序列号
data.list.devicesName	String	设备名称
data.list.deviceId	Integer	设备id
code	Integer	返回类型，0.成功，其他：失败

usb连接

方法名

usbConnect

所有的连接监听也在此方法中，包括关闭连接、断开连接等

用法

- 用法如下：

```
module.usbConnect({
  deviceName: "/dev/bus/usb/001/003"
}, res => {
  console.log(res)
});
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
deviceName	String	是	无	设备名称

回调

- 示例

```
{
  "data": {},
  "message": "usb连接成功",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

usb连接

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：请参考【 状态码 】

画布打印

画线

方法名

drawLine

用法

```
module.drawLine({
  lineWidth: 2,
  startX: 0,
  startY: 0,
  endX: 100,
  endY: 100,
  isSolidLine: false
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
lineWidth	Integer	是	无	线宽度
startX	Integer	是	无	x起始位置
startY	Integer	是	无	y起始位置
endX	Integer	是	无	x结束位置
endY	Integer	是	无	y结束位置
isSolidLine	Boolean	是	无	是否是实线

回调示例

```
{
  "message": "",
  "code": 0,
```


画线

```
"data": {}  
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画边框

方法名

drawBorder

用法

```
module.drawLine({
  lineWidth: 2,
  topLeftX: 0,
  topLeftY: 0,
  bottomRightX: 0,
  bottomRightY: 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
lineWidth	Integer	是	无	线宽度
topLeftX	Integer	是	无	左上x位置
topLeftY	Integer	是	无	左上y位置
bottomRightX	Integer	是	无	右下x位置
bottomRightY	Integer	是	无	右下y位置

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画文字

方法名

drawText

用法

```
module.drawText({
  "textX": 0,
  "textY": 0,
  "text": "123456",
  "fontSize": "Size_16",
  "rotate": "Rotate_0",
  "bold": 0,
  "reverse": 0,
  "underline": 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
textX	Integer	是	无	文本x起点位置
textY	Integer	是	无	文本y起点位置
text	String	是	无	文本内容
fontSize	String	否	Size_16	字体大小，可选值：Size_16，Size_24，Size_32，Size_48，Size_64，Size_72，Size_96

参数名	参数类型	是否必填	默认值	描述
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
bold	Integer	否	0	是否加粗，0：不加粗，1：加粗
reverse	Integer	否	0	是否翻转，0：否，1：是
underline	Integer	否	0	是否加下划线，0：否，1：是

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画区域块文字

方法名

drawTextForArea

用法

```
module.drawTextForArea({
    "areaStartX": 0,
    "areaStartY": 0,
    "areaEndX": 100,
    "areaEndY": 100,
    "xAlign": "CENTER",
    "yAlign": "CENTER",
    "text": "123456",
    "fontSize": "Size_64",
    "bold": 0,
    "reverse": 0,
    "underline": 0,
    "deleteline": 0,
    "rotate": "Rotate_0"
}, res => {
    resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
areaStartX	Integer	是	无	打印起始x位置
areaStartY	Integer	是	无	打印起始y位置
areaEndX	Integer	是	无	打印结束x位置
areaEndY	Integer	是	无	打印结束y位置

参数名	参数类型	是否必填	默认值	描述
xAlign	String	是	无	x方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
yAlign	String	是	无	y方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
text	String	是	无	文本内容
fontSize	String	否	Size_16	字体大小，可选值：Size_16，Size_24，Size_32，Size_48，Size_64，Size_72，Size_96
bold	Integer	否	0	是否加粗，0：不加粗，1：加粗
reverse	Integer	否	0	是否翻转，0：否，1：是
underline	Integer	否	0	是否加下划线，0：否，1：是
deleteline	Integer	否	0	是否加删除线，0：否，1：是
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画条形码

方法名

drawBarcode

用法

```
module.drawBarcode({
  "startX": 1,
  "startY": 150,
  "text": "12345678",
  "barcodeType": "CODE128",
  "lineWidth": 2,
  "height": 75,
  "rotate": "Rotate_0"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
text	String	是	无	文本内容
barcodeType	String	是	无	条码类型
lineWidth	Integer	是	无	线宽度
height	Integer	否	0	高度
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270

barcodeType可选值：UPC_E，CODABAR，ITF，CODE39，CODE93，CODE128，EAN8Plus2，EAN8Plus5，EAN13Plus2，EAN13Plus5，JAN3_EAN13，JAN8_EAN8，JAN128，MSI，Postnet，QR，UPC_A，UPCAPlus2，UPCAPlus5，UPCEPlus2，UPCEPlus5

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画区域块条形码

方法名

用法

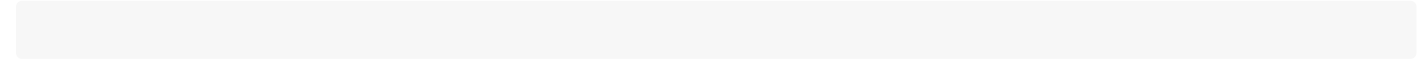
参数说明

参数名	参数类型	是否必填	默认值	描述
areaStartX	Integer	是	无	打印起始x位置
areaStartY	Integer	是	无	打印起始y位置
areaEndX	Integer	是	无	打印结束x位置
areaEndY	Integer	是	无	打印结束y位置
xAlign	String	是	无	x方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
yAlign	String	是	无	y方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
text	String	是	无	文本内容
barcodeType	String	是	无	条码类型
lineWidth	Integer	是	无	线宽度
height	Integer	否	0	高度

参数名	参数类型	是否必填	默认值	描述
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270

barcodeType可选值：UPC_E，CODABAR，ITF，CODE39，CODE93，CODE128，EAN8Plus2，EAN8Plus5，EAN13Plus2，EAN13Plus5，JAN3_EAN13，JAN8_EAN8，JAN128，MSI，Postnet，QR，UPC_A，UPCAPlus2，UPCAPlus5，UPCEPlus2，UPCEPlus5

回调示例



回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画二维码

方法名

drawQrCode

用法

```
module.drawQrCode({
  "startX": 1,
  "startY": 150,
  "text": "12345678",
  "rotate": "Rotate_0",
  "ver": 2,
  "level": 75,
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
text	String	是	无	文本内容
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
ver	Integer	是	无	-
level	Integer	否	0	等级

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

画图

方法名

drawGraphic

用法

```
module.drawGraphic({
  "type": "drawable",
  "path": "goodwork",
  "startX": 0,
  "startY": 150,
  "zoomImage": true,
  "newWidth": 100
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
zoomImage	boolean	否	false	是否缩放图片
newWidth	Integer	否	无	缩放图片后的宽度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

区域画图

方法名

drawGraphicForArea

用法

```
module.drawGraphicForArea({
    "areaStartX": 0,
    "areaStartY": 0,
    "areaEndX": 100,
    "areaEndY": 100,
    "bmpSizeX": 100,
    "bmpSizeY": 100,
    "xAlign": "CENTER",
    "yAlign": "CENTER",
    "type": "drawable",
    "path": "goodwork",
    "zoomImage": true,
    "newWidth": 100
}, res => {
    resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
areaStartX	Integer	是	无	打印起始x位置
areaStartY	Integer	是	无	打印起始y位置
areaEndX	Integer	是	无	打印结束x位置
areaEndY	Integer	是	无	打印结束y位置
bmpSizeX	Integer	是	无	图片的x位置
bmpSizeY	Integer	是	无	图片的y位置

参数名	参数类型	是否必填	默认值	描述
xAlign	String	是	无	x方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
yAlign	String	是	无	y方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
zoomImage	boolean	否	false	是否缩放图片
newWidth	Integer	否	无	缩放图片后的宽度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

区域画图

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印

方法名

print

用法

```
module.print({
  "rotate": "Rotate_0",
  skip: 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
skip	Integer	否	0	-

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

打印

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL指令打印

获取TSPL纸张宽度

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.width	Integer	纸张宽度
code	Integer	返回类型，0.成功，其他：失败

设置TSPL纸张宽高

参数名	参数类型	是否必填	默认值	描述
width	Integer	否	无	宽度
height	Integer	否	无	高度

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

页设置TSPL

方法名

pageSetupTSPL

用法

```
module.pageSetupTSPL({
  paperSize: "SIZE_58mm",
  pageWidth: 384,
  pageHeight: 540
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
paperType	String	是	无	纸张类型，可选值： SIZE_58mm， SIZE_80mm， SIZE_100mm
pageWidth	Integer	是	无	纸张宽度
pageHeight	Integer	是	无	纸张高度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印线

方法名

drawLineTSPL

用法

```
module.drawLineTSPL({
  "startX": 0,
  "startY": 0,
  "lineLength": 100,
  "lineHeight": 2
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	x起始位置
startY	Integer	是	无	y起始位置
lineLength	Integer	是	无	线长度
lineHeight	Integer	是	无	线高度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印边框

方法名

drawBorderTSPL

用法

```
module.drawBorderTSPL({
  lineWidth: 2,
  topLeftX: 0,
  topLeftY: 0,
  bottomRightX: 0,
  bottomRightY: 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
lineWidth	Integer	是	无	线宽度
topLeftX	Integer	是	无	左上x位置
topLeftY	Integer	是	无	左上y位置
bottomRightX	Integer	是	无	右下x位置
bottomRightY	Integer	是	无	右下y位置

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印文本

方法名

drawTextTSPL

用法

```
module.drawTextTSPL({
  "startX": 0,
  "startY": 0,
  "isSimplifiedChinese": true,
  "xMultiplication": 1,
  "yMultiplication": 1,
  "rotate": "Rotate_0",
  "text": "123456"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	x起始位置
startY	Integer	是	无	y起始位置
isSimplifiedChinese	Boolean	否	true	是否是简体中文
xMultiplication	Integer	否	1	x放大倍数
yMultiplication	Integer	否	1	y放到倍数
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
text	String	是	无	文本内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签区域打印文本

方法名

drawTextTSPLForArea

用法

```
module.drawTextTSPLForArea({
  "areaStartX": 0,
  "areaStartY": 0,
  "areaEndX": 100,
  "areaEndY": 100,
  "xAlign": "CENTER",
  "yAlign": "CENTER",
  "isSimplifiedChinese": true,
  "xMultiplication": 1,
  "yMultiplication": 1,
  "rotate": "Rotate_0",
  "text": "123456"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
areaStartX	Integer	是	无	打印起始x位置
areaStartY	Integer	是	无	打印起始y位置
areaEndX	Integer	是	无	打印结束x位置
areaEndY	Integer	是	无	打印结束y位置
xAlign	String	是	无	x方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐

参数名	参数类型	是否必填	默认值	描述
yAlign	String	是	无	y方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
isSimplifiedChinese	Boolean	否	true	是否是简体中文
xMultiplication	Integer	否	1	x放大倍数
yMultiplication	Integer	否	1	y放到倍数
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
text	String	是	无	文本内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印二维码

方法名

用法

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
barcodeType	String	是	无	二维码类型，可选值：QR，DMATRIX，PDF417
level	Integer	是	无	等级
width	Integer	是	无	宽度
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
text	String	是	无	文本内容

回调示例

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印条码

方法名

drawBarCodeTSPL

用法

```
module.drawBarCodeTSPL({
  "startX": 1,
  "startY": 150,
  "barcodeType": "CODE128",
  "height": 10,
  "isReadable": false,
  "rotate": "Rotate_0",
  "narrowWidth": 1,
  "wideWidth": 2
  "text": "123456"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
barcodeType	String	是	无	条码类型
height	Integer	是	无	条码高度
isReadable	Boolean	是	无	是否可读
rotate	String	否	Rotate_0	旋转角度，可选值：Rotate_0，Rotate_90，Rotate_180，Rotate_270
narrowWidth	Integer	是	无	窄宽

参数名	参数类型	是否必填	默认值	描述
wideWidth	Integer	是	无	宽宽
text	String	是	无	文本内容

barcodeType可选值：UPC_E，CODABAR，ITF，CODE39，CODE93，CODE128，EAN8Plus2，EAN8Plus5，EAN13Plus2，EAN13Plus5，JAN3_EAN13，JAN8_EAN8，JAN128，MSI，Postnet，QR，UPC_A，UPCAPlus2，UPCAPlus5，UPCEPlus2，UPCEPlus5

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印图片

方法名

drawBitmapTSPL

用法

```
module.drawBitmapTSPL({
  "type": "drawable",
  "path": "goodwork",
  "startX": 0,
  "startY": 150,
  "zoomImage": true,
  "newWidth": 100,
  "mode":0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
zoomImage	boolean	否	false	是否缩放图片
newWidth	Integer	否	无	缩放图片后的宽度
mode	Integer	否	无	-

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签区域图片打印

方法名

drawBitmapTSPLForArea

用法

```
module.drawBitmapTSPLForArea({
    "areaStartX": 0,
    "areaStartY": 0,
    "areaEndX": 100,
    "areaEndY": 100,
    "xAlign": "CENTER",
    "yAlign": "CENTER",
    "type": "drawable",
    "path": "goodwork",
    "zoomImage": true,
    "newWidth": 100,
    "mode":0
}, res => {
    resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
areaStartX	Integer	是	无	打印起始x位置
areaStartY	Integer	是	无	打印起始y位置
areaEndX	Integer	是	无	打印结束x位置
areaEndY	Integer	是	无	打印结束y位置
xAlign	String	是	无	x方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐

参数名	参数类型	是否必填	默认值	描述
yAlign	String	是	无	y方向对齐方式， 可选值：NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
zoomImage	boolean	否	false	是否缩放图片
newWidth	Integer	否	无	缩放图片后的宽度
mode	Integer	否	无	-

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签打印

方法名

printTSPL

用法

```
module.printTSPL({
  "numbers": 1
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
numbers	Integer	否	0	打印份数

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签翻转区域打印

方法名

reverseAreaTSPL

用法

```
module.reverseAreaTSPL({
  "startX": 1,
  "startY": 150,
  "width": 2
  "height": 10
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
width	Integer	是	无	宽度
height	Integer	是	无	高度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签擦除区域

方法名

eraseAreaTSPL

用法

```
module.eraseAreaTSPL({
  "startX": 1,
  "startY": 150,
  "width": 2
  "height": 10
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
startX	Integer	是	无	起始x位置
startY	Integer	是	无	起始y位置
width	Integer	是	无	宽度
height	Integer	是	无	高度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签设置

方法名

setPrinterTSPL

用法

```
module.setPrinterTSPL({
  "command": "SHIFT",
  "value": 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
command	String	是	无	命令
value	Integer	是	无	值

command 可取值：DIRECTION：方向，FEED：-，REPRINT：重新打印，SPEED：速度，DENSITY：密度，OFFSET：偏移量，SHIFT：移动，FORMFEED：走纸，HOME：重置，PRINTKEY：打印密钥

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL设置标签间的缝隙大小

方法名

setGAPTSP

用法

```
module.setGAPTSP({
  value: 0,
  offset: 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
offset	String	是	无	间隙偏移
value	Integer	是	无	间隙宽度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

TSPL设置标签间的缝隙大小

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

TSPL标签蜂鸣器

参数名	参数类型	是否必填	默认值	描述
level	String	是	无	等级
interval	Integer	是	无	蜂鸣时长

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签发送指令

方法名

sendStrToPrinterTSPL

用法

```
module.sendStrToPrinterTSPL({
  content: "REFERENCE 10,20\r\n"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
content	String	是	无	发送内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

TSPL标签开启钱箱

方法名

openCashBoxTSPL

用法

```
module.openCashBoxTSPL({
  t1: 0,
  t2: 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
t1	Integer	是	无	1号钱箱，取值范围：[0,255]
t2	Integer	是	无	2号钱箱，取值范围：[0,255]

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

通用方法

关闭连接

方法名

close

用法

```
module.close(res => {
  console.log(res)
})
```

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

发送字节数据

方法名

sendBytesData

用法

```
module.sendBytesData({
  content: bytes
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
content	Array[byte]	是	无	字节数据数组

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印文本

方法名

printText

用法

```
module.printText({
  content: "123"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
content	String	是	无	打印内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印图片带压缩

方法名

printImageWithCompressed

用法

```
module.printImageWithCompressed({
  "type": "drawable",
  "path": "goodwork",
  "alignType": "NONE",
  "left": 0,
  "isCompressed": false
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
alignType	String	否	NONE	对齐方式，NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
left	Integer	否	0	边距

参数名	参数类型	是否必填	默认值	描述
isCompressed	Boolean	否	false	是否压缩

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印图片带等级

方法名

printImageWidthLevel

用法

```
module.printImageWidthLevel({
    "type": "drawable",
    "path": "goodwork",
    "alignType": "NONE",
    "left": 0,
    "level": 0
}, res => {
    resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
alignType	String	否	NONE	对齐方式，NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
left	Integer	否	0	边距

参数名	参数类型	是否必填	默认值	描述
level	Integer	否	0	等级

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印灰色图片

方法名

printColorImg2Gray

用法

```
module.printColorImg2Gray({
  "type": "drawable",
  "path": "goodwork",
  "alignType": "NONE",
  "left": 0,
  "isCompressed": false
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
alignType	String	否	NONE	对齐方式，NONE：自动，END：尾对齐，CENTER：居中对齐，START：开始对齐
left	Integer	否	0	边距

打印灰色图片

参数名	参数类型	是否必填	默认值	描述
isCompressed	Boolean	否	false	是否压缩

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

针打图形

方法名

printImageStylus

用法

```
module.printColorImg2Gray({
    "type": "drawable",
    "path": "goodwork",
    "multiple": 0,
    "pointNum": 0
}, res => {
    resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	图片类型，path：本地文件绝对路径，drawable：drawable文件，assets：assets文件
path	String	是	无	图片路径或图片名称
multiple	Integer	否	0	参数为0倍高倍宽，为1只倍高
pointNum	Integer	否	0	针点数量

回调示例

针打图形

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印表格

方法名

printTable

用法

```
module.printTable({
  column: "品名;数量;单价;金额",
  regular: ";",
  columnWidth: [14, 6, 6, 6],
  rows: [
    "保鲜袋;1;10.00;10.00",
    "铁丝挂钩;2;5.00;10.00",
    "雨伞;3;5.00;15.00"
  ]
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
column	String	是	无	列名称（表头）
regular	String	否	;	列的分隔符
columnWidth	Array[Integer]	是	无	列宽
rows	Array[String]	是	无	行内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打印条码和二维码

方法名

printBarCode

用法

```
module.printBarCode({
  type: "CODE128",
  width: 2,
  height: 150,
  level: 2,
  content: "123456"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
type	String	是	无	码类型
width	Integer	是	无	宽度
height	Integer	是	无	高度
level	Integer	是	无	等级
content	String	是	无	内容

type参数可取值：CODE39，CODE128，CODE93，ITF，UPC_E，CODABAR，DATAMATRIX，JAN8，JAN13，JAN128，PDF417，QRCODE，UPC_A

回调示例

```
{
  "message": "",
  "code": 0,
```

```
"data": {}  
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

获取纸张宽度

方法名

getPaperWidth

用法

```
module.getPaperWidth(res => {  
  console.log(res)  
})
```

回调示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "width": 384  
  }  
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.width	Integer	纸张宽度
code	Integer	返回类型，0.成功，其他：失败

设置纸张宽度

方法名

用法

参数说明

参数名	参数类型	是否必填	默认值	描述
width	Integer	是	无	宽度

回调示例

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

页设置

方法名

pageSetup

用法

```
module.pageSetup({
  paperType: "Size_58mm",
  pageWidth: 384,
  pageHeight: 540
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
paperType	String	是	无	纸张类型，可选值： Size_58mm， Size_80mm， Size_100mm
pageWidth	Integer	是	无	纸张宽度
pageHeight	Integer	是	无	纸张高度

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

初始化打印机

方法名

initPrinter

用法

```
module.initPrinter(res => {
  console.log(res)
})
```

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

设置字体样式

方法名

setFont

用法

```
module.setFont({
  "characterType": 0,
  "width": 0,
  "height": 0,
  "bold": 0,
  "underline": 0
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
characterType	Integer	否	0	编码类型，取值范围：[0,1]
width	Integer	否	0	宽度，取值范围：[0,7]
height	Integer	否	0	高度，取值范围：[0,7]
bold	Integer	否	0	是否加粗，0：否，1：是
underline	Integer	否	0	是否加下划线，0：否，1：是

回调示例

设置字体样式

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

设置打印属性

方法名

setPrinter

用法

```
module.setPrinter({
  command: "ALIGN",
  value: 2
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
command	String	是	无	命令
value	Integer	否	0	值

command 可取值：PRINT_AND_WAKE_PAPER_BY_LNCH：打印并唤醒纸，
PRINT_AND_WAKE_PAPER_BY_LINE：打印并换行，PRINT_AND_ENTER：打印回车，
CHARACTER_RIGHT_MARGIN：字符右边距，ALIGN：对齐方式
value：当command为ALIGN时，value=0：左对齐，value=1：居中，value=2：右对齐

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

设置打印属性

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

获取编码集

方法名

getCharsetName

用法

```
module.getCharsetName(res => {
  console.log(res)
})
```

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "charsetName": "GBK"
  }
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.charsetName	String	编码集
code	Integer	返回类型，0.成功，其他：失败

设置编码集

方法名

setCharsetName

用法

```
module.setCharsetName({
  charsetName: "GBK"
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
charsetName	String	是	无	编码集

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

打开钱箱

方法名

openCashBox

用法

```
module.openCashBox({
  cashbox1: true,
  cashbox2: false
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
cashbox1	boolean	否	false	1号钱箱是否开启
cashbox2	boolean	否	false	2号钱箱是否开启

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

打开钱箱

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

切纸

方法名

cutPaper

用法

```
module.cutPaper({
  isHalf: false
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
isHalf	boolean	否	false	是否是半切， true：半切， false：全切

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

切纸

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

打印机是否缺纸

方法名

isPaperOut

用法

```
module.isPaperOut(res => {  
  console.log(res)  
})
```

回调示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "isPaperOut": false  
  }  
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.isPaperOut	boolean	是否缺纸，false：否，true：是
code	Integer	返回类型，0.成功，其他：失败

打印机纸张是否将用尽

方法名

isPaperWillOut

用法

```
module.isPaperWillOut(res => {  
  console.log(res)  
})
```

回调示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "isPaperWillOut": false  
  }  
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.isPaperWillOut	boolean	是否纸将尽，false：否，true：是
code	Integer	返回类型，0.成功，其他：失败

获取打印机状态

方法名

getCurrentStatus

用法

```
module.getCurrentStatus(res => {  
  console.log(res)  
})
```

回调示例

```
{  
  "message": "",  
  "code": 0,  
  "data": {  
    "status": 0  
  }  
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.status	Integer	状态
code	Integer	返回类型，0.成功，其他：失败

status：0：正常，-1：写入数据失败，-2：缺纸，-3：纸将尽，-4：开盖，-5：读数据失败

获取打印结果状态

参数名	参数类型	是否必填	默认值	描述
timeout	Integer	否	100	超时时间，单位： 毫秒

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.status	Integer	状态
code	Integer	返回类型，0.成功，其他：失败

status：0：打印完成，-1：未知异常，-2：正在打印中，-3：缺纸，-4：纸舱盖开盖，-5：与打印机通信失败，-6：通信异常，-7：接收数据格式不正确

十六进制字符串转字节数组

方法名

hexStringToByteArray

用法

```
module.hexStringToByteArray({
  content: "this is a test "
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
content	String	是	无	要转换的内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "bytes": [123, 456, 112]
  }
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

十六进制字符串转字节数组

参数名	参数类型	参数描述
data.bytes	Integer	转换后的byte数组
code	Integer	返回类型，0.成功，其他：失败

字符串转16进制

方法名

stringToHex

用法

```
module.stringToHex({
  content: "this is a test "
}, res => {
  resolve(res)
})
```

参数说明

参数名	参数类型	是否必填	默认值	描述
content	String	是	无	要转换的内容

回调示例

```
{
  "message": "",
  "code": 0,
  "data": {
    "value": "abceef"
  }
}
```

回调说明

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

参数名	参数类型	参数描述
data.value	Integer	转换后的数据
code	Integer	返回类型，0.成功，其他：失败