

# uniapp 安卓商米打印机UTS原生插件

## 目 录

|          |         |
|----------|---------|
| 使用说明     | 1       |
| 使用方法     | 1.1     |
| 更新日志     | 1.2     |
| 状态说明     | 1.3     |
| 插件方法     | 2       |
| 初始化      | 2.1     |
| 获取打印机列表  | 2.2     |
| 选择打印机    | 2.3     |
| 打印文件     | 2.4     |
| 获取打印机信息  | 2.5     |
| 打印机状态查询  | 2.6     |
| 发送ESC命令  | 2.7     |
| 发送TSPL命令 | 2.8     |
| 开启钱箱     | 2.9     |
| 获取钱箱状态   | 2.10    |
| 释放资源     | 2.11    |
| 热敏小票接口   | 2.12    |
| 行设置      | 2.12.1  |
| 添加文本     | 2.12.2  |
| 打印文本     | 2.12.3  |
| 列排列打印    | 2.12.4  |
| 打印条形码    | 2.12.5  |
| 打印二维码    | 2.12.6  |
| 打印图片     | 2.12.7  |
| 打印分割线    | 2.12.8  |
| 打印结束输出   | 2.12.9  |
| 事务模式-开关  | 2.12.10 |
| 事务模式-提交  | 2.12.11 |
| 标签小票接口   | 2.13    |
| 画布设置     | 2.13.1  |
| 绘制文本     | 2.13.2  |
| 绘制条形码    | 2.13.3  |
| 绘制二维码    | 2.13.4  |
| 绘制图像     | 2.13.5  |

|                   |        |
|-------------------|--------|
| 绘制特殊图形 .....      | 2.13.6 |
| 打印绘制的内容 .....     | 2.13.7 |
| LCD客显控制 .....     | 2.14   |
| 客显屏的状态 .....      | 2.14.1 |
| 显示指定大小的文本内容 ..... | 2.14.2 |
| 显示多行文本内容 .....    | 2.14.3 |
| 显示位图 .....        | 2.14.4 |
| 显示价格内容 .....      | 2.14.5 |

# 使用方法

---

## 介绍

安卓商米打印机UTS原生插件是商米打印服务SDK开发的插件，插件已包含SDK的全部功能，欢迎使用

## 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=22728>

## 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-sunmiPrinter"
```

## 示例程序

示例程序页面较多，请通过插件市场插件页面导入示例项目

# 更新日志

---

2025-03-21

首次发布

# 状态说明

## 打印机状态

| 状态码                 | 描述                       |
|---------------------|--------------------------|
| UNKNOWN             | 未知状态                     |
| COMM                | 打印机通信异常                  |
| READY               | 打印机已准备好，可正常打印            |
| OFFLINE             | 打印机离线/故障                 |
| ERR_STEP            | 字车故障（需要激光打印机支持）          |
| ERR_COVER           | 打印机未关闭纸舱盖                |
| ERR_CUTTER          | 印机切刀错误（需要切刀支持）           |
| ERR_DRUM_LOSS       | 鼓组件未安装（需要激光打印机支持）        |
| ERR_MOTOR_HOT       | 马达过热                     |
| ERR_PAPER_JAM       | 打印机堵纸中                   |
| ERR_PAPER_OUT       | 打印机缺纸中                   |
| ERR_DRUM_EMPTY      | 鼓组件已用完（需要激光打印机支持）        |
| WARN_CARTRIDGE      | 墨盒即将用尽（需要激光打印机支持）        |
| ERR_CARTON_LOSS     | 纸盒未安装（需要激光打印机支持）         |
| ERR_DUPLEX_LOSS     | 双面单元未安装（需要激光打印机支持）       |
| ERR_PRINTER_HOT     | 打印头过热                    |
| WARN_PICK_PAPER     | 当前打印纸待取走（需要具有取值传感器的设备支持） |
| ERR_CARTON_EMPTY    | 纸盒缺纸（需要激光打印机支持）          |
| ERR_DRUM_MISMATCH   | 鼓组件不匹配（需要激光打印机支持）        |
| ERR_CARTRIDGE_LOSS  | 墨盒未安装（需要激光打印机支持）         |
| ERR_PAPER_MISMATCH  | 当前打印纸不匹配打印机              |
| WARN_SPECIAL_PAPER  | 多功能纸盒即将用尽（需要激光打印机支持）     |
| WARN_THERMAL_PAPER  | 热敏打印纸纸将尽                 |
| ERR_CARTON_MISMATCH | 纸盒不匹配（需要激光打印机支持）         |

状态说明

| 状态码                    | 描述                     |
|------------------------|------------------------|
| ERR_CARTRIDGE_EMPTY    | 墨盒已用完（需要激光打印机支持）       |
| WARN_STANDARD_PAPER    | 标准打印纸纸盒即将用尽（需要激光打印机支持） |
| ERR_COVER_INCOMPLETE   | 打印机未完全关闭纸舱盖            |
| ERR_CARTRIDGE_MISMATCH | 墨盒不匹配（需要激光打印机支持）       |

打印机类型

| 状态码                | 描述      |
|--------------------|---------|
| thermal printer    | 普通热敏打印机 |
| black mark printer | 黑标热敏打印机 |
| label printer      | 标签热敏打印机 |
| needle printer     | 针式打印机   |
| laser printer      | 激光打印机   |

# 初始化

## 方法名

init

使用插件前需先初始化

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{  
  "data": {  
    "defaultPrinter": {  
      "printerId": "打印机id",  
      "status": "打印机状态",  
      "hardwareVersion": "打印机硬件版本号,根据具体型号打印机返回值略有不同",  
      "name": "打印机名称",  
      "version": "打印机固件版本号",  
      "distance": "打印机已经打印的距离",  
      "cutter": "打印机使用切刀次数",  
      "hot": "打印机发生过热次数",  
      "density": "打印机浓度",  
      "type": "打印机类型",  
      "paper": "打印机当前支持的纸张可打印区域"  
    }  
  },  
  "message": "",
```



初始化

```
"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.printerId       | String  | 打印机id                                |
| data.status          | String  | 打印机状态，请参考【 <a href="#">状态说明</a> 】    |
| data.hardwareVersion | String  | 打印机硬件版本号,根据具体型号打印机返回值略有不同            |
| data.name            | String  | 打印机名称                                |
| data.version         | String  | 打印机固件版本号                             |
| data.distance        | String  | 打印机已经打印的距离                           |
| data.cutter          | String  | 打印机使用切刀次数                            |
| data.hot             | String  | 打印机发生过热次数                            |
| data.density         | String  | 打印机浓度                                |
| data.type            | String  | 打印机类型，具体请参考【 <a href="#">打印机类型</a> 】 |
| data.paper           | String  | 打印机当前支持的纸张可打印区域                      |
| code                 | Integer | 返回类型，0.成功，其他：失败                      |

# 获取打印机列表

## 方法名

getPrinterList

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "printerId": "打印机id",  
        "status": "打印机状态",  
        "hardwareVersion": "打印机硬件版本号,根据具体型号打印机返回值略有不同",  
        "name": "打印机名称",  
        "version": "打印机固件版本号",  
        "distance": "打印机已经打印的距离",  
        "cutter": "打印机使用切刀次数",  
        "hot": "打印机发生过热次数",  
        "density": "打印机浓度",  
        "type": "打印机类型",  
        "paper": "打印机当前支持的纸张可打印区域"  
      }  
    ]  
  },  
  "message": "",
```

获取打印机列表

```
"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] | 打印机列表集合                              |
| data.list.printerId       | String        | 打印机id                                |
| data.list.status          | String        | 打印机状态，请参考【 <a href="#">状态说明</a> 】    |
| data.list.hardwareVersion | String        | 打印机硬件版本号,根据具体型号打印机返回值略有不同            |
| data.list.name            | String        | 打印机名称                                |
| data.list.version         | String        | 打印机固件版本号                             |
| data.list.distance        | String        | 打印机已经打印的距离                           |
| data.list.cutter          | String        | 打印机使用切刀次数                            |
| data.list.hot             | String        | 打印机发生过热次数                            |
| data.list.density         | String        | 打印机浓度                                |
| data.list.type            | String        | 打印机类型，具体请参考【 <a href="#">打印机类型</a> 】 |
| data.list.paper           | String        | 打印机当前支持的纸张可打印区域                      |
| code                      | Integer       | 返回类型，0.成功，其他：失败                      |

# 选择打印机

## 方法名

changePrinter

## 用法

- 用法如下：

```
module.changePrinter({
  //打印机ID
  printerId: "打印机ID"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

| 参数名       | 参数类型   | 是否必填 | 默认值 | 参数描述                |
|-----------|--------|------|-----|---------------------|
| printerId | String | 是    | 无   | 打印机id，可以从获取打印机列表中取到 |

## 回调

- 示例

```
{
  "data": {},
  "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 |

选择打印机

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

# 打印文件

## 方法名

printFile

## 用法

- 用法如下：

```
module.printFile({
  path: "文件路径"
}, res => {
  if (res.code == 0) {
    //打印成功
  } else {
    //打印失败
  }
})
```

- 参数说明

| 参数名         | 参数类型    | 是否必填 | 默认值   | 参数描述                              |
|-------------|---------|------|-------|-----------------------------------|
| path        | String  | 是    | 无     | 文件路径，支持网络地址                       |
| fileCollate | Boolean | 否    | false | 置是否逐份打印，如果是pdf文件且要打印文件份数超过一张时逐份打印 |
| fileCopies  | Integer | 否    | 1     | 指定文件打印份数                          |

| 参数名        | 参数类型    | 是否必填 | 默认值      | 参数描述                                                                         |
|------------|---------|------|----------|------------------------------------------------------------------------------|
| duplex     | String  | 否    | SINGLE   | 指定单双面打印，需要打印机硬件支持，<br>DOUBLE_LONG：长边打印，<br>DOUBLE_SHORT：短边打印，<br>SINGLE：单页打印 |
| fileEnd    | Integer | 否    | 无        | 设置文件打印最终页数                                                                   |
| fileRotate | String  | 否    | ROTATE_0 | 指定打印的方向，可选值：<br>ROTATE_90，<br>ROTATE_180，<br>ROTATE_270                      |
| fileStart  | Integer | 否    | 无        | 设置文件打印起始页数                                                                   |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 获取打印机信息

## 方法名

getPrinterInfo

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{  
  "data": {  
    "hardwareVersion": "打印机硬件版本号,根据具体型号打印机返回值略有不同",  
    "name": "打印机名称",  
    "version": "打印机固件版本号",  
    "distance": "打印机已经打印的距离",  
    "cutter": "打印机使用切刀次数",  
    "hot": "打印机发生过热次数",  
    "density": "打印机浓度",  
    "type": "打印机类型",  
    "paper": "打印机当前支持的纸张可打印区域"  
  },  
  "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.hardwareVersion | String  | 打印机硬件版本号,根据具体型号打印机返回值略有不同            |
| data.name            | String  | 打印机名称                                |
| data.version         | String  | 打印机固件版本号                             |
| data.distance        | String  | 打印机已经打印的距离                           |
| data.cutter          | String  | 打印机使用切刀次数                            |
| data.hot             | String  | 打印机发生过热次数                            |
| data.density         | String  | 打印机浓度                                |
| data.type            | String  | 打印机类型，具体请参考【 <a href="#">打印机类型</a> 】 |
| data.paper           | String  | 打印机当前支持的纸张可打印区域                      |
| code                 | Integer | 返回类型，0.成功，其他：失败                      |

# 打印机状态查询

## 方法名

getPrinterStatus

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{
  "data": {
    "status": "打印机状态"
  },
  "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.status | String  | 打印机状态，请参考【 <a href="#">状态说明</a> 】 |
| code        | Integer | 返回类型，0.成功，其他：失败                   |

# 发送ESC命令

## 方法名

sendEscCommand

## 用法

- 用法如下：

```
module.sendEscCommand({
  //发送的字节数组
  bytes: [0x1d, 0x33]
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

| 参数名   | 参数类型        | 是否必填 | 默认值 | 参数描述    |
|-------|-------------|------|-----|---------|
| bytes | Array[byte] | 是    | 无   | 发送的字节数组 |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 发送TSPL命令

## 方法名

sendTsplCommand

## 用法

- 用法如下：

```
module.sendTsplCommand({
  //发送的内容
  text: "SIZE 40 mm, 30 mm\r\n"
}, res => {
  console.log(JSON.stringify(res))
})
```

- 参数说明

| 参数名  | 参数类型   | 是否必填 | 默认值 | 参数描述    |
|------|--------|------|-----|---------|
| text | String | 是    | 无   | 发送的指令内容 |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 开启钱箱

## 方法名

openCash

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{
  "data": { },
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 获取钱箱状态

## 方法名

isOpenCash

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{
  "data": {
    "status": true
  },
  "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.status | Boolean | 钱箱开启状态，true：开启，false：未开启 |
| code        | Integer | 返回类型，0.成功，其他：失败          |

# 释放资源

## 方法名

release

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{
  "data": { },
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

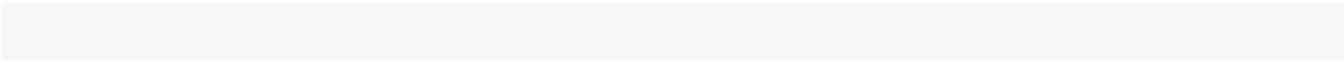
# 热敏小票接口

---



# 行设置

- 用法如下：



- 参数说明

| 参数名         | 参数类型    | 是否必填 | 默认值   | 参数描述                                              |
|-------------|---------|------|-------|---------------------------------------------------|
| width       | Integer | 否    | 无     | 设置行内可打印区域宽度，可设置最大范围为纸张宽度减去左边距当可设置范围超出时仅设置最大值      |
| align       | String  | 否    | 无     | 设置之后每行的对齐方式，可选值：LEFT，CENTER，RIGHT                 |
| height      | Integer | 否    | 无     | 设置每行高度，行高范围在0~255像素当行高小于打印内容时以行内最大高度内容为准          |
| posX        | Integer | 否    | 0     | 设置行左边距                                            |
| posY        | Integer | 否    | 0     | 设置行右边距，在行打印中不可用                                   |
| renderColor | String  | 否    | BLACK | 设置颜色，可选值：BLACK，RED，仅在支持双色打印的商米设备上生效，同时双色打印需要打印纸支持 |

行设置

- 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 |

# 添加文本

## 方法名

addText

向打印机发送文本内容，addText()不会直接打印，通过多次调用addText()并设置不同的TextStyle使一行内容呈现不同效果，而printText()将直接打印出发送的文本内容  
当最后调用addText()接口需要添加“ \n” 字符才可以保证打印输出

## 用法

- 用法如下：

```
module.addText({
  text: "123"
}, res=>{})
```

- 参数说明

| 参数名             | 参数类型    | 是否必填 | 默认值   | 参数描述                             |
|-----------------|---------|------|-------|----------------------------------|
| text            | String  | 是    | 无     | 打印的文本                            |
| enableBold      | boolean | 否    | false | 是否加粗                             |
| posX            | Integer | 否    | 0     | 相对本行已打印内容的距离                     |
| posY            | Integer | 否    | 0     | 相对本行Y方向已打印内容的距离，在行打印文本时不可用       |
| textSize        | Integer | 否    | 24    | 设置文本字符大小                         |
| align           | String  | 否    | LEFT  | 可选值：LEFT，CENTER，RIGHT，在行打印文本时不可用 |
| enableAntiColor | boolean | 否    | false | 是否文本反白                           |
| enableInvert    | boolean | 否    | false | 是否文本倒置                           |

| 参数名                 | 参数类型    | 是否必填 | 默认值      | 参数描述                                              |
|---------------------|---------|------|----------|---------------------------------------------------|
| enableInvert        | boolean | 否    | false    | 是否文本斜体                                            |
| enableStrikethrough | boolean | 否    | false    | 是否文本删除线                                           |
| enableUnderline     | boolean | 否    | false    | 是否文本下划线                                           |
| font                | String  | 否    | 无        | 设置自定义字体                                           |
| height              | String  | 否    | 无        | 高度，在行打印文本时不可用                                     |
| renderColor         | String  | 否    | BLACK    | 设置颜色，可选值：BLACK，RED，仅在支持双色打印的商米设备上生效，同时双色打印需要打印纸支持 |
| rotate              | String  | 否    | ROTATE_0 | 指定打印的方向，可选值：ROTATE_90，ROTATE_180，ROTATE_270       |
| textHeightRatio     | Integer | 否    | 0        | 纵向放大倍数，取值范围：[0,7]                                 |
| textSpace           | Integer | 否    | 0        | 文本字间距，取值范围：[0,100]                                |
| textWidthRatio      | Integer | 否    | 0        | 横向放大倍数，取值范围：[0,7]                                 |
| width               | String  | 否    | 无        | 宽度，在行打印文本时不可用                                     |

## 回调

- 示例

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

添加文本

```
    "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 打印文本

## 方法名

printText

## 用法

- 用法如下：

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

- 参数说明

| 参数名                 | 参数类型    | 是否必填 | 默认值   | 参数描述                             |
|---------------------|---------|------|-------|----------------------------------|
| text                | String  | 是    | 无     | 打印的文本                            |
| enableBold          | boolean | 否    | false | 是否加粗                             |
| posX                | Integer | 否    | 0     | 相对本行已打印内容的距离                     |
| posY                | Integer | 否    | 0     | 相对本行Y方向已打印内容的距离，在行打印文本时不可用       |
| textSize            | Integer | 否    | 24    | 设置文本字符大小                         |
| align               | String  | 否    | LEFT  | 可选值：LEFT，CENTER，RIGHT，在行打印文本时不可用 |
| enableAntiColor     | boolean | 否    | false | 是否文本反白                           |
| enableInvert        | boolean | 否    | false | 是否文本倒置                           |
| enableInvert        | boolean | 否    | false | 是否文本斜体                           |
| enableStrikethrough | boolean | 否    | false | 是否文本删除线                          |

| 参数名             | 参数类型    | 是否必填 | 默认值      | 参数描述                                              |
|-----------------|---------|------|----------|---------------------------------------------------|
| enableUnderline | boolean | 否    | false    | 是否文本下划线                                           |
| font            | String  | 否    | 无        | 设置自定义字体                                           |
| height          | String  | 否    | 无        | 高度，在行打印文本时不可用                                     |
| renderColor     | String  | 否    | BLACK    | 设置颜色，可选值：BLACK，RED，仅在支持双色打印的商米设备上生效，同时双色打印需要打印纸支持 |
| rotate          | String  | 否    | ROTATE_0 | 指定打印的方向，可选值：ROTATE_90，ROTATE_180，ROTATE_270       |
| textHeightRatio | Integer | 否    | 0        | 纵向放大倍数，取值范围：[0,7]                                 |
| textSpace       | Integer | 否    | 0        | 文本字间距，取值范围：[0,100]                                |
| textWidthRatio  | Integer | 否    | 0        | 横向放大倍数，取值范围：[0,7]                                 |
| width           | String  | 否    | 无        | 宽度，在行打印文本时不可用                                     |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |



# 列排列打印

## 方法名

printTexts

## 用法

- 用法如下：

```
let style = {
  align: "CENTER"
};
module.printTexts({
  texts: ["第一列", "第二列", "第三列"],
  colsWidth: [1, 1, 1],
  styles: [style, style, style]
}, res=>{})
```

- 参数说明

| 参数名               | 参数类型           | 是否必填 | 默认值   | 参数描述                       |
|-------------------|----------------|------|-------|----------------------------|
| texts             | Array[String]  | 是    | 无     | 打印的文本列表                    |
| colsWidth         | Array[Integer] | 是    | 无     | 打印的列宽度                     |
| styles            | Array[Object]  | 是    | 无     | 打印的列样式                     |
| styles.enableBold | boolean        | 否    | false | 是否加粗                       |
| styles.posX       | Integer        | 否    | 0     | 相对本行已打印内容的距离               |
| styles.posY       | Integer        | 否    | 0     | 相对本行Y方向已打印内容的距离，在行打印文本时不可用 |
| styles.textSize   | Integer        | 否    | 24    | 设置文本字符大小                   |

| 参数名                        | 参数类型    | 是否必填 | 默认值      | 参数描述                                              |
|----------------------------|---------|------|----------|---------------------------------------------------|
| styles.align               | String  | 否    | LEFT     | 可选值：LEFT，CENTER，RIGHT，在行打印文本时不可用                  |
| styles.enableAntiColor     | boolean | 否    | false    | 是否文本反白                                            |
| styles.enableInvert        | boolean | 否    | false    | 是否文本倒置                                            |
| styles.enableInvert        | boolean | 否    | false    | 是否文本斜体                                            |
| styles.enableStrikethrough | boolean | 否    | false    | 是否文本删除线                                           |
| styles.enableUnderline     | boolean | 否    | false    | 是否文本下划线                                           |
| styles.font                | String  | 否    | 无        | 设置自定义字体                                           |
| styles.height              | String  | 否    | 无        | 高度，在行打印文本时不可用                                     |
| styles.renderColor         | String  | 否    | BLACK    | 设置颜色，可选值：BLACK，RED，仅在支持双色打印的商米设备上生效，同时双色打印需要打印纸支持 |
| styles.rotate              | String  | 否    | ROTATE_0 | 指定打印的方向，可选值：ROTATE_90，ROTATE_180，ROTATE_270       |
| styles.textHeightRatio     | Integer | 否    | 0        | 纵向放大倍数，取值范围：[0,7]                                 |
| styles.textSpace           | Integer | 否    | 0        | 文本字间距，取值范围：[0,100]                                |
| styles.textWidthRatio      | Integer | 否    | 0        | 横向放大倍数，取值范围：[0,7]                                 |

| 参数名          | 参数类型   | 是否必填 | 默认值 | 参数描述          |
|--------------|--------|------|-----|---------------|
| styles.width | String | 否    | 无   | 宽度，在行打印文本时不可用 |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 打印条形码

## 方法名

printBarCode

## 用法

- 用法如下：

```
module.printBarCode({
  code: "123"
}, res=>{})
```

- 参数说明

| 参数名       | 参数类型    | 是否必填 | 默认值   | 参数描述                  |
|-----------|---------|------|-------|-----------------------|
| code      | String  | 是    | 无     | 条码内容                  |
| align     | String  | 否    | LEFT  | 可选值：LEFT，CENTER，RIGHT |
| posX      | Integer | 否    | 0     | 相对本行已打印内容的距离          |
| posY      | Integer | 否    | 0     | 相对本行Y方向已打印内容的距离       |
| height    | Integer | 否    | 无     | 生成条码指定缩放高度            |
| barHeight | Integer | 否    | 162   | 设置条码高度，取值范围：[1,255]   |
| dotWidth  | Integer | 否    | 2     | 设置码块宽度，取值范围：[1,16]    |
| readable  | boolean | 否    | false | 设置HRI位置               |

| 参数名       | 参数类型   | 是否必填 | 默认值      | 参数描述                                                    |
|-----------|--------|------|----------|---------------------------------------------------------|
| rotate    | String | 否    | ROTATE_0 | 指定打印的方向，可选值：<br>ROTATE_90，<br>ROTATE_180，<br>ROTATE_270 |
| symbology | String | 否    | CODE128  | 设置条码类型，<br>请参考下方说明                                      |
| width     | String | 否    | 无        | 生成条码指定缩放宽度                                              |

symbology可选值：UPCA，UPCE，EAN13，EAN8，CODE39，ITF，CODABAR，CODE93，CODE128，EAN128，CODE128A，ITF\_C，CODE39\_S，CODE39\_C，EAN13\_2，UPCA\_2，UPCA\_5，EAN13\_5，EAN8\_2，EAN8\_5，UPCE\_2，UPCE\_5

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 打印二维码

## 方法名

printQrCode

## 用法

- 用法如下：

```
module.printQrCode({
  code: "123"
}, res=>{})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值      | 参数描述                                        |
|--------|---------|------|----------|---------------------------------------------|
| code   | String  | 是    | 无        | 二维码内容                                       |
| width  | String  | 否    | 无        | 生成二维码指定缩放宽度                                 |
| height | Integer | 否    | 无        | 生成二维码指定缩放高度                                 |
| align  | String  | 否    | LEFT     | 可选值：LEFT，CENTER，RIGHT                       |
| posX   | Integer | 否    | 0        | 相对本行已打印内容的距离                                |
| posY   | Integer | 否    | 0        | 相对本行Y方向已打印内容的距离                             |
| rotate | String  | 否    | ROTATE_0 | 指定打印的方向，可选值：ROTATE_90，ROTATE_180，ROTATE_270 |

| 参数名        | 参数类型    | 是否必填 | 默认值 | 参数描述                 |
|------------|---------|------|-----|----------------------|
| errorLevel | String  | 否    | L   | 设置二维码纠错等级，可选值请参考下方说明 |
| dotSize    | Integer | 否    | 4   | 设置二维码块大小             |

errorLevel纠错等级说明  
L：纠错等级L 7%  
M：纠错等级M 15%  
Q：纠错等级Q 25%  
H：纠错等级H 30%

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 打印图片

## 方法名

printBitmap

## 用法

- 用法如下：

```
module.printBitmap({
  path: "test",
  type: "drawable"
}, res=>{})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值  | 参数描述                                                               |
|--------|---------|------|------|--------------------------------------------------------------------|
| path   | String  | 是    | 无    | 打印图片的路径，不支持网络图片                                                    |
| type   | String  | 是    | 无    | 图片类型，<br>path：本地文件绝对路径，<br>drawable：drawable文件，<br>assets：assets文件 |
| width  | String  | 否    | 无    | 图片缩放宽度                                                             |
| height | Integer | 否    | 无    | 图片缩放高度                                                             |
| align  | String  | 否    | LEFT | 可选值：LEFT，CENTER，RIGHT                                              |
| posX   | Integer | 否    | 0    | 相对本行已打印内容的距离                                                       |
| posY   | Integer | 否    | 0    | 相对本行Y方向已打印内容的距离                                                    |



| 参数名       | 参数类型    | 是否必填 | 默认值          | 参数描述                                               |
|-----------|---------|------|--------------|----------------------------------------------------|
| algorithm | String  | 否    | BINARIZATION | 设置图片转换方式，可选值：BINARIZATION(二值化算法)，DITHERING(抖动灰度算法) |
| value     | Integer | 否    | 0            | 设置额外参数                                             |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 打印分割线

## 方法名

printDividingLine

## 用法

- 用法如下：

```
module.printDividingLine({
  dividingLine: "EMPTY",
  offset: 30
}, res=>{})
```

- 参数说明

| 参数名          | 参数类型    | 是否必填 | 默认值 | 参数描述                                                       |
|--------------|---------|------|-----|------------------------------------------------------------|
| dividingLine | String  | 是    | 无   | 线类型，可选<br>值：EMPTY(空白<br>线即空行)，<br>SOLID(实线)，<br>DOTTED(点线) |
| offset       | Integer | 是    | 无   | 分割线的显示高<br>度                                               |

## 回调

- 示例

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

- 回调说明：

打印分割线

| 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 |

# 打印结束输出

## 方法名

autoOut

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{
  "data": { },
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 事务模式-开关

## 方法名

enableTransMode

## 用法

- 用法如下：

```
module.enableTransMode({
  enable: true
}, res=>{})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值 | 参数描述     |
|--------|---------|------|-----|----------|
| enable | Boolean | 是    | 无   | 事务模式开启状态 |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 事务模式-提交

## 方法名

printTrans

## 用法

- 用法如下：

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

- 参数说明

无

## 回调

- 示例

```
{
  "data": { },
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 标签小票接口

---

# 画布设置

## 方法名

initCanvas

## 用法

- 用法如下：

```
module.initCanvas({
  align: "CENTER"
}, res=>{})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值 | 参数描述                                         |
|--------|---------|------|-----|----------------------------------------------|
| width  | Integer | 否    | 无   | 设置行内可打印区域宽度，可设置最大范围为纸张宽度减去左边距当可设置范围超出时仅设置最大值 |
| align  | String  | 否    | 无   | 设置之后每行的对齐方式，可选值：LEFT，CENTER，RIGHT            |
| height | Integer | 否    | 无   | 设置每行高度，行高范围在0~255像素当行高小于打印内容时以行内最大高度内容为准     |
| posX   | Integer | 否    | 0   | 设置行左边距                                       |



| 参数名         | 参数类型    | 是否必填 | 默认值   | 参数描述                                              |
|-------------|---------|------|-------|---------------------------------------------------|
| posY        | Integer | 否    | 0     | 设置行右边距，<br>在行打印中不可用                               |
| renderColor | String  | 否    | BLACK | 设置颜色，可选值：BLACK，RED，仅在支持双色打印的商米设备上生效，同时双色打印需要打印纸支持 |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 绘制文本

## 方法名

renderText

## 用法

- 用法如下：

```
module.renderText({
  text: "123"
}, res=>{})
```

- 参数说明

| 参数名                 | 参数类型    | 是否必填 | 默认值   | 参数描述                             |
|---------------------|---------|------|-------|----------------------------------|
| text                | String  | 是    | 无     | 打印的文本                            |
| enableBold          | boolean | 否    | false | 是否加粗                             |
| posX                | Integer | 否    | 0     | 相对本行已打印内容的距离                     |
| posY                | Integer | 否    | 0     | 相对本行Y方向已打印内容的距离，在行打印文本时不可用       |
| textSize            | Integer | 否    | 24    | 设置文本字符大小                         |
| align               | String  | 否    | LEFT  | 可选值：LEFT，CENTER，RIGHT，在行打印文本时不可用 |
| enableAntiColor     | boolean | 否    | false | 是否文本反白                           |
| enableInvert        | boolean | 否    | false | 是否文本倒置                           |
| enableInvert        | boolean | 否    | false | 是否文本斜体                           |
| enableStrikethrough | boolean | 否    | false | 是否文本删除线                          |

| 参数名             | 参数类型    | 是否必填 | 默认值      | 参数描述                                              |
|-----------------|---------|------|----------|---------------------------------------------------|
| enableUnderline | boolean | 否    | false    | 是否文本下划线                                           |
| font            | String  | 否    | 无        | 设置自定义字体                                           |
| height          | String  | 否    | 无        | 高度，在行打印文本时不可用                                     |
| renderColor     | String  | 否    | BLACK    | 设置颜色，可选值：BLACK，RED，仅在支持双色打印的商米设备上生效，同时双色打印需要打印纸支持 |
| rotate          | String  | 否    | ROTATE_0 | 指定打印的方向，可选值：ROTATE_90，ROTATE_180，ROTATE_270       |
| textHeightRatio | Integer | 否    | 0        | 纵向放大倍数，取值范围：[0,7]                                 |
| textSpace       | Integer | 否    | 0        | 文本字间距，取值范围：[0,100]                                |
| textWidthRatio  | Integer | 否    | 0        | 横向放大倍数，取值范围：[0,7]                                 |
| width           | String  | 否    | 无        | 宽度，在行打印文本时不可用                                     |

## 回调

- 示例

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

绘制文本

- 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 |

# 绘制条形码

## 方法名

renderBarCode

## 用法

- 用法如下：

```
module.renderBarCode({
  code: "123"
}, res=>{})
```

- 参数说明

| 参数名       | 参数类型    | 是否必填 | 默认值   | 参数描述                  |
|-----------|---------|------|-------|-----------------------|
| code      | String  | 是    | 无     | 条码内容                  |
| align     | String  | 否    | LEFT  | 可选值：LEFT，CENTER，RIGHT |
| posX      | Integer | 否    | 0     | 相对本行已打印内容的距离          |
| posY      | Integer | 否    | 0     | 相对本行Y方向已打印内容的距离       |
| height    | Integer | 否    | 无     | 生成条码指定缩放高度            |
| barHeight | Integer | 否    | 162   | 设置条码高度，取值范围：[1,255]   |
| dotWidth  | Integer | 否    | 2     | 设置码块宽度，取值范围：[1,16]    |
| readable  | boolean | 否    | false | 设置HRI位置               |

| 参数名       | 参数类型   | 是否必填 | 默认值      | 参数描述                                        |
|-----------|--------|------|----------|---------------------------------------------|
| rotate    | String | 否    | ROTATE_0 | 指定打印的方向，可选值：ROTATE_90，ROTATE_180，ROTATE_270 |
| symbology | String | 否    | CODE128  | 设置条码类型，请参考下方说明                              |
| width     | String | 否    | 无        | 生成条码指定缩放宽度                                  |

symbology可选值：UPCA，UPCE，EAN13，EAN8，CODE39，ITF，CODABAR，CODE93，CODE128，EAN128，CODE128A，ITF\_C，CODE39\_S，CODE39\_C，EAN13\_2，UPCA\_2，UPCA\_5，EAN13\_5，EAN8\_2，EAN8\_5，UPCE\_2，UPCE\_5

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 绘制二维码

## 方法名

renderQrCode

## 用法

- 用法如下：

```
module.renderQrCode({
  code: "123"
}, res=>{})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值      | 参数描述                                                        |
|--------|---------|------|----------|-------------------------------------------------------------|
| code   | String  | 是    | 无        | 二维码内容                                                       |
| width  | String  | 否    | 无        | 生成二维码指定<br>缩放宽度                                             |
| height | Integer | 否    | 无        | 生成二维码指定<br>缩放高度                                             |
| align  | String  | 否    | LEFT     | 可选值：LEFT，<br>CENTER，<br>RIGHT                               |
| posX   | Integer | 否    | 0        | 相对本行已打印<br>内容的距离                                            |
| posY   | Integer | 否    | 0        | 相对本行Y方向已<br>打印内容的距离                                         |
| rotate | String  | 否    | ROTATE_0 | 指定打印的方<br>向，可选值：<br>ROTATE_90，<br>ROTATE_180，<br>ROTATE_270 |

| 参数名        | 参数类型    | 是否必填 | 默认值 | 参数描述                 |
|------------|---------|------|-----|----------------------|
| errorLevel | String  | 否    | L   | 设置二维码纠错等级，可选值请参考下方说明 |
| dotSize    | Integer | 否    | 4   | 设置二维码块大小             |

errorLevel纠错等级说明  
L：纠错等级L 7%  
M：纠错等级M 15%  
Q：纠错等级Q 25%  
H：纠错等级H 30%

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |



# 绘制图像

## 方法名

renderBitmap

## 用法

- 用法如下：

```
module.renderBitmap({
  path: "test",
  type: "drawable"
}, res=>{})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值  | 参数描述                                                               |
|--------|---------|------|------|--------------------------------------------------------------------|
| path   | String  | 是    | 无    | 打印图片的路径，不支持网络图片                                                    |
| type   | String  | 是    | 无    | 图片类型，<br>path：本地文件绝对路径，<br>drawable：drawable文件，<br>assets：assets文件 |
| width  | String  | 否    | 无    | 图片缩放宽度                                                             |
| height | Integer | 否    | 无    | 图片缩放高度                                                             |
| align  | String  | 否    | LEFT | 可选值：LEFT，CENTER，RIGHT                                              |
| posX   | Integer | 否    | 0    | 相对本行已打印内容的距离                                                       |
| posY   | Integer | 否    | 0    | 相对本行Y方向已打印内容的距离                                                    |

| 参数名       | 参数类型    | 是否必填 | 默认值          | 参数描述                                               |
|-----------|---------|------|--------------|----------------------------------------------------|
| algorithm | String  | 否    | BINARIZATION | 设置图片转换方式，可选值：BINARIZATION(二值化算法)，DITHERING(抖动灰度算法) |
| value     | Integer | 否    | 0            | 设置额外参数                                             |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 绘制特殊图形

## 方法名

renderArea

## 用法

- 用法如下：

```
module.renderArea({
  shape: "RECT_FILL",
  width: 50,
  height: 50
}, res=>{{})
```

- 参数说明

| 参数名       | 参数类型    | 是否必填 | 默认值       | 参数描述                          |
|-----------|---------|------|-----------|-------------------------------|
| shape     | String  | 否    | RECT_FILL | 打印的图形，可参考下方说明                 |
| posX      | Integer | 否    | 0         | 设置起始x坐标                       |
| posY      | Integer | 否    | 0         | 设置起始y坐标                       |
| width     | String  | 否    | 50        | 设置图形的宽度，形状为线段时无效，形状为圆形时表示圆形直径 |
| height    | Integer | 否    | 50        | 设置图形的高度，形状为线段时无效，形状为圆形时无效     |
| endX      | Integer | 否    | 50        | 设置线段的终点x坐标                    |
| endY      | Integer | 否    | 50        | 设置线段的终点y坐标                    |
| thickness | Integer | 否    | 1         | 描边宽度                          |

shape说明

- RECT\_FILL：填充矩形 矩形区域为黑块
- RECT\_WHITE：擦除矩形 矩形区域为白块
- RECT\_REVERSE：反白矩形 矩形区域将反白
- BOX：空心矩形
- CIRCLE：空心圆形
- OVAL：空心椭圆
- PATH：线段

# 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 打印绘制的内容

## 方法名

printCanvas

## 用法

- 用法如下：

```
module.printCanvas({
  count: 2
}, res=>{})
```

- 参数说明

| 参数名   | 参数类型    | 是否必填 | 默认值 | 参数描述   |
|-------|---------|------|-----|--------|
| count | Integer | 否    | 1   | 欲打印的次数 |

## 回调

- 示例

```
{
  "data": {},
  "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            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# LCD客显控制

---

# 客显屏的状态

## 方法名

lcdConfig

## 用法

- 用法如下：

```
module.lcdConfig({
  command: "INIT"
}, res=>{})
```

- 参数说明

| 参数名     | 参数类型   | 是否必填 | 默认值 | 参数描述          |
|---------|--------|------|-----|---------------|
| command | String | 是    | 无   | 命令，可选值请参考下方说明 |

command命令类型

- INIT：初始化LCD
- CLEAR：清除LCD屏显内容
- WAKE：唤醒LCD
- SLEEP：休眠LCD

## 回调

- 示例

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

- 回调说明：

客显屏的状态

| 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 |



# 显示指定大小的文本内容

## 方法名

showText

## 用法

- 用法如下：

```
module.showText({
  text: "123"
}, res=>{})
```

- 参数说明

| 参数名  | 参数类型    | 是否必填 | 默认值   | 参数描述                       |
|------|---------|------|-------|----------------------------|
| text | String  | 是    | 无     | 需要显示的内容, 可显示的内容长度为128/size |
| size | Integer | 否    | 32    | 指定显示字体大小, 取值范围：[6,40]      |
| fill | Boolean | 否    | false | 是否拉伸显示内容高度整个屏幕             |

## 回调

- 示例

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

- 回调说明：

显示指定大小的文本内容

| 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 |

# 显示多行文本内容

## 方法名

showTexts

## 用法

- 用法如下：

```
module.showTexts({
  texts: ["第一列", "第二列", "第三列"],
  align: [1, 1, 1]
}, res=>{})
```

- 参数说明

| 参数名   | 参数类型           | 是否必填 | 默认值 | 参数描述                    |
|-------|----------------|------|-----|-------------------------|
| texts | Array[String]  | 是    | 无   | 打印每行文本的内容，当内容为空时表示此行无内容 |
| align | Array[Integer] | 是    | 无   | 打印文本内容的行间距比例            |

## 回调

- 示例

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

- 回调说明：

显示多行文本内容

| 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 |

# 显示位图

## 方法名

showBitmap

## 用法

- 用法如下：

```
module.showBitmap({
  path: "test",
  type: "drawable"
}, res=>{})
```

- 参数说明

| 参数名  | 参数类型   | 是否必填 | 默认值 | 参数描述                                                               |
|------|--------|------|-----|--------------------------------------------------------------------|
| path | String | 是    | 无   | 打印图片的路径，不支持网络图片                                                    |
| type | String | 是    | 无   | 图片类型，<br>path：本地文件绝对路径，<br>drawable：drawable文件，<br>assets：assets文件 |

## 回调

- 示例

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

显示位图

- 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 |

# 显示价格内容

## 方法名

showDigital

## 用法

- 用法如下：

```
module.showDigital({
  digital: "123"
}, res=>{})
```

- 参数说明

| 参数名     | 参数类型   | 是否必填 | 默认值 | 参数描述                                     |
|---------|--------|------|-----|------------------------------------------|
| digital | String | 是    | 无   | 显示价格内容，字符串可支持7位0-9数字和A-Z字符以及在7位字符中插入的“.” |

## 回调

- 示例

```
{
  "data": {},
  "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 |

显示价格内容

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