

uniapp 安卓芯烨云打印机云打印UTS原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
初始化	2.1
批量添加打印机	2.2
设置打印机语音类型	2.3
清空打印队列	2.4
查询订单是否打印成功	2.5
查询打印机某天的订单统计数	2.6
获取单个打印机状态	2.7
批量获取打印机状态	2.8
金额播报	2.9
打印内容	2.10
标签打印	2.11
字符串替换	2.12
格式化订单数据	2.13

使用方法

介绍

安卓芯烨云打印机云打印UTS原生插件支持打印文本，图片，条形码，二维码，标签打印等，打印功能比较灵活可以自定义打印排版，插件UTS开发，支持uniapp和uniapp x

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=25096>

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-xpyPrinter"
```

示例文件用到的打印示例

```
/**  
 * 此处仅为演示用例  
 * 演示用例为芯烨云sdk的示例程序内容
```

```

*/
import * as module from "@uni_modules/leven-uts-xpyPrinter"
export class PrintUtils {
  constructor() {}

  /**
   * 打印不带金额播报，小票打印字体对齐样例，不支持金额播报
   * 注意：对齐标签L C R CB 请勿嵌套使用，嵌套使用内层标签有效，外层失效；
   * 同一行请勿使用多个对齐标签，否则只有最后一个对齐标签有效
   *
   * <BR>：换行符（同一行有闭合标签（如 </C>）则应放到闭合标签前面，连续两个换行符<BR><BR>
   * <L></L>：左对齐
   * <C></C>：居中对齐
   * <R></R>：右对齐
   * 注意：同一行内容不能使用多种对齐方式，可通过补空格方式自定义对齐样式。
   * 58mm的机器，一行打印16个汉字，32个字母
   * 80mm的机器，一行打印24个汉字，48个字母
   *
   * <N></N>：字体正常大小
   * <HB></HB>：字体变高一倍
   * <WB></WB>：字体变宽一倍
   * <B></B>：字体放大一倍
   * <CB></CB>：字体放大一倍居中
   * <HB2></HB2>：字体变高二倍
   * <WB2></WB2>：字体变宽二倍
   * <B2></B2>：字体放大二倍
   * <BOLD></BOLD>：字体加粗
   * <IMG></IMG>：打印LOGO图片，需登录开放平台在【打印机管理=》设备管理】下通过设置LOGO
   * 空标签，如 <IMG></IMG> 即可，具体可参考样例。
   * 图片宽度设置：可以通过 <IMG> 标签名称自定义，如 <IMG60> 表示宽度为60，
   * 不需要指定高度。<IMG> 标签不指定宽度默认为40，最小值为20，最大值为100
   * <QR></QR>：二维码（标签内容是二维码值，最大不能超过256个字符，单个订单最多只能打印-
   * 二维码宽度设置：可以通过 <QR> 标签名称自定义，如 <QR180> 表示宽度为18
   * 不需要指定宽度。<QR> 标签不指定宽度默认为110，最小值为90，最大值为180
   * <BARCODE></BARCODE>：条形码（标签内容是条形码值）
   */
  printFontAlign() {
    let list = [];
    list.push("不加标签：默认字体大小<BR>")
    list.push("<BR>")
    list.push("L标签：<L>左对齐<BR></L>")
    list.push("<BR>")
    list.push("R标签：<R>右对齐<BR></R>")
    list.push("<BR>")
    list.push("C标签：<C>居中对齐<BR></C>")
    list.push("<BR>")
    list.push("N标签：<N>字体正常大小<BR></N>")
    list.push("<BR>")
    list.push("HB标签：<HB>字体变高一倍<BR></HB>")
    list.push("<BR>")
  }
}

```

```

list.push("WB标签：<WB>字体变宽一倍<BR></WB>")
list.push("<BR>")
list.push("B标签：<B>字体放大一倍<BR></B>")
list.push("<BR>")
list.push("HB2标签：<HB2>字体变高二倍<BR></HB2>")
list.push("<BR>")
list.push("WB2标签：<WB2>字体变宽二倍<BR></WB2>")
list.push("<BR>")
list.push("B2标签：<B2>字体放大二倍<BR></B2>")
list.push("<BR>")
list.push("BOLD标签：<BOLD>字体加粗<BR></BOLD>")
// 嵌套使用对齐和字体
list.push("<BR>")
list.push("<C>嵌套使用：<BOLD>居中加粗</BOLD><BR></C>")
// 打印条形码和二维码
list.push("<BR>")
list.push("<C><BARCODE>9884822189</BARCODE></C>")
list.push("<C><QR>https://www.xpyun.net</QR></C>")
module.print({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //打印内容
  list: list,
  //打印份数
  copies: 1,
  //声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用户申请退单
  voice: 2,
  //打印模式：
  //值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不
  //值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，
  mode: 0
}, res => {

})
}

/**
 * 3.小票打印字体对齐样例，支持金额播报
 * 注意：对齐标签L C R CB 请勿嵌套使用，嵌套使用内层标签有效，外层失效；
 * 同一行请勿使用多个对齐标签，否则只有最后一个对齐标签有效
 */
printFontAlignVoiceSupport() {
  let list = [];
  list.push("不加标签：默认字体大小<BR>")
  list.push("<BR>")
  list.push("L标签：<L>左对齐<BR></L>")
  list.push("<BR>")
  list.push("R标签：<R>右对齐<BR></R>")
  list.push("<BR>")
  list.push("C标签：<C>居中对齐<BR></C>")

```

```

list.push("<BR>")
list.push("N标签：<N>字体正常大小<BR></N>")
list.push("<BR>")
list.push("HB标签：<HB>字体变高一倍<BR></HB>")
list.push("<BR>")
list.push("WB标签：<WB>字体变宽一倍<BR></WB>")
list.push("<BR>")
list.push("B标签：<B>字体放大一倍<BR></B>")
list.push("<BR>")
list.push("HB2标签：<HB2>字体变高二倍<BR></HB2>")
list.push("<BR>")
list.push("WB2标签：<WB2>字体变宽二倍<BR></WB2>")
list.push("<BR>")
list.push("B2标签：<B2>字体放大二倍<BR></B2>")
list.push("<BR>")
list.push("BOLD标签：<BOLD>字体加粗<BR></BOLD>")
// 嵌套使用对齐和字体
list.push("<BR>")
list.push("<C>嵌套使用：<BOLD>居中加粗</BOLD><BR></C>")
// 打印条形码和二维码
list.push("<BR>")
list.push("<C><BARCODE>9884822189</BARCODE></C>")
list.push("<C><QR>https://www.xpyun.net</QR></C>")

module.print({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //打印内容
  list: list,
  //打印份数
  copies: 1,
  //声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用户申请退单了
  voice: 2,
  //打印模式：
  //值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不在线
  //值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，则返回设备不在线
  mode: 0,
  //支付方式：
  //取值范围41~55：
  //支付宝 41、微信 42、云支付 43、银联刷卡 44、银联支付 45、会员卡消费 46、会员卡充值
  //仅用于支持金额播报的芯烨云打印机。
  payType: 41,
  //支付与否：
  //取值范围59~61：
  //退款 59 到账 60 消费 61。
  //仅用于支持金额播报的芯烨云打印机。
  payMode: 60,
  //支付金额：
  //最多允许保留2位小数。
  //仅用于支持金额播报的芯烨云打印机。

```

```

        money: 20.15
    }, res => {

    })
}

/**
 * 3.小票打印综合排版样例，不支持金额播报
 * 58mm打印机一行可打印32个字符
 *
 * 注意：此示例用到原生插件中的字符串重复打印和格式化订单数据
 */
async printComplexReceipt() {
    let list = [];
    list.push("<C><B>芯烨云小票</B><BR></C>");
    list.push("<BR>");
    //示例中用到的原生插件的字符串重复打印
    let repeatSpace16 = await this.repeat(" ", 16);
    let repeatSpace2 = await this.repeat(" ", 2);
    list.push("菜名" + repeatSpace16 + "数量" + repeatSpace2 + "单价" + repeatSpace2);
    list.push(await this.repeat("-", 32) + "<BR>");
    list.push(await this.formatPrintOrderItem("可乐鸡翅", 2, 9.99));
    list.push(await this.formatPrintOrderItem("水煮鱼特辣", 1, 108.0));
    list.push(await this.formatPrintOrderItem("豪华版超级无敌龙虾炒饭", 1, 99.9));
    list.push(await this.formatPrintOrderItem("炭烤鳕鱼", 5, 19.99));
    list.push(await this.repeat("-", 32) + "<BR>");
    list.push("<R>合计：327.83元<BR></R>");
    list.push("<BR>");
    list.push("<L>");
    list.push("客户地址：珠海市香洲区xx路xx号<BR>");
    list.push("客户电话：1363*****88<BR>");
    list.push("下单时间：2020-9-9 15:07:57<BR>");
    list.push("备注：少放辣 不吃香菜<BR>");
    list.push("<C>");
    list.push("<QRCODE s=6 e=L l=center>https://www.xpyun.net</QRCODE>");
    list.push("</C>");
    module.print({
        //打印机设备编号
        sn: "xxxxxxxxxxxxxxxx",
        //打印内容
        list: list,
        //打印份数
        copies: 1,
        //声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用户申请退单了
        voice: 2,
        //打印模式：
        //值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不在线
        //值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，则返回设备不在线
        mode: 0
    }, res => {

```

```

    })
  }

  /**
   * 3.小票打印综合排版样例，支持金额播报
   * 58mm打印机一行可打印32个字符
   */
  async printComplexReceiptVoiceSupport() {
    let list = [];
    list.push("<C><B>芯烨云小票</B><BR></C>");
    list.push("<BR>");
    //示例中用到的原生插件的字符串重复打印
    let repeatSpace16 = await this.repeat(" ", 16);
    let repeatSpace2 = await this.repeat(" ", 2);
    list.push("菜名" + repeatSpace16 + "数量" + repeatSpace2 + "单价" + repeatSpace2);
    list.push(await this.repeat("-", 32) + "<BR>");
    list.push(await this.formatPrintOrderItem("可乐鸡翅", 2, 9.99));
    list.push(await this.formatPrintOrderItem("水煮鱼特辣", 1, 108.0));
    list.push(await this.formatPrintOrderItem("豪华版超级无敌龙虾炒饭", 1, 99.9));
    list.push(await this.formatPrintOrderItem("炭烤鳕鱼", 5, 19.99));
    list.push(await this.repeat("-", 32) + "<BR>");
    list.push("<R>合计：327.83元<BR></R>");
    list.push("<BR>");
    list.push("<L>");
    list.push("客户地址：珠海市香洲区xx路xx号<BR>");
    list.push("客户电话：1363*****88<BR>");
    list.push("下单时间：2020-9-9 15:07:57<BR>");
    list.push("备注：少放辣 不吃香菜<BR>");
    list.push("<C>");
    list.push("<QR CODE s=6 e=L l=center>https://www.xpyun.net</QR CODE>");
    list.push("</C>");
    module.print({
      //打印机设备编号
      sn: "xxxxxxxxxxxxxxxx",
      //打印内容
      list: list,
      //打印份数
      copies: 1,
      //声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用户申请退单了
      voice: 2,
      //打印模式：
      //值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不在线
      //值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，返回设备不在线
      mode: 0,
      //支付方式：
      //取值范围41~55：
      //支付宝 41、微信 42、云支付 43、银联刷卡 44、银联支付 45、会员卡消费 46、会员卡充值
      //仅用于支持金额播报的芯烨云打印机。
      payType: 41,
    });
  }
}

```

```

//支付与否：
//取值范围59~61：
//退款 59 到账 60 消费 61。
//仅用于支持金额播报的芯烨云打印机。
payMode: 60,
//支付金额：
//最多允许保留2位小数。
//仅用于支持金额播报的芯烨云打印机。
money: 20.15
}, res => {

})
}

/**
 * 4. 标签打印综合排版样例
 * 如何确定坐标：坐标原点位于左上角，x轴是从左往右，y轴是从上往下；
 * 根据测试，x轴最大值=标签纸宽度*8，y轴最大值=标签纸高度*8，
 * 如标签纸尺寸为40*30，x轴最大值=40*8=320，y轴最大值=30*8=240
 * 实际排版效果需要用户按实际纸张尺寸和需求自行排版
 * <p>
 * 打印内容内（标签除外）大于号和小于号需要经过转译才能正常打印。其中，“<”用“&lt”表示，“>”用“&gt”表示。
 *
 * <PAGE></PAGE>：
 * 分页，用于支持打印多张不同的标签页面（最多10张），不使用该标签表示所有元素只打印在一个页面。
 *
 * <SIZE>width,height</SIZE>：
 * 设置标签纸宽高，width 标签纸宽度(不含背纸)，height 标签纸高度(不含背纸)，单位mm，如40mm*30mm。
 *
 * <TEXT x="10" y="100" w="1" h="2" r="0">文本内容</TEXT>：
 * 打印文本，其中：
 * 属性 x 为水平方向起始点坐标（默认为0）
 * 属性 y 为垂直方向起始点坐标（默认为0）
 * 属性 w 为文字宽度放大倍率1-10（默认为1）
 * 属性 h 为文字高度放大倍率1-10（默认为1）
 * 属性 r 为文字旋转角度(顺时针方向，默认为0)：
 * 0 0度
 * 90 90度
 * 180 180度
 * 270 270度
 *
 * <BC128 x="10" y="100" h="60" s="1" n="1" w="1" r="0">1234567</BC128>：
 * 打印code128一维码，其中：
 * 属性 x 为水平方向起始点坐标（默认为0）
 * 属性 y 为垂直方向起始点坐标（默认为0）
 * 属性 h 为条形码高度（默认为48）
 * 属性 s 是否人眼可识：0 不可识，1 可识（默认为1）
 * 属性 n 为窄 bar 宽度，以点(dot)表示（默认为1）
 * 属性 w 为宽 bar 宽度，以点(dot)表示（默认为1）
 * 属性 r 为文字旋转角度（顺时针方向，默认为0）：

```



```

* 0 0度
* 90 90度
* 180 180度
* 270 270度
*
* <BC39 x="10" y="100" h="60" s="1" n="1" w="1" r="0">1234567</BC39> :
* 打印code39一维码，其中：
* 属性 x 为水平方向起始点坐标（默认为0）
* 属性 y 为垂直方向起始点坐标（默认为0）
*
* * 属性 h 为条形码高度（默认为48）
* * 属性 s 是否人眼可识：0 不可识，1 可识（默认为1）
* * 属性 n 为窄 bar 宽度，以点(dot)表示（默认为1）
* * 属性 w 为宽 bar 宽度，以点(dot)表示（默认为2）
* * 属性 r 为文字旋转角度(顺时针方向，默认为0)：
* * 0 0度
* * 90 90度
* * 180 180度
* * 270 270度
*
*
* * <QR x="20" y="20" w="160" e="H">二维码内容</QR> :
* * 打印二维码，其中：
* * 属性 x 为水平方向起始点坐标（默认为0）
* * 属性 y 为垂直方向起始点坐标（默认为0）
* * 属性 w 为二维码宽度（默认为160）
* * 属性 e 为纠错等级：L 7% M 15% Q 25% H 30%（默认为H）
* * 标签内容是二维码值，最大不能超过256个字符
* * 注意：单个订单最多只能打印一个二维码
* <IMG x="16" y="32" w="100"> :
* 打印LOGO图片，需登录开放平台在【打印机管理=》设备管理】下通过设置LOGO功能进行上传。此
* 写入空标签，若指定了<PAGE>标签，<IMG>标签应该放到<PAGE>标签里面， <IMG>，如 <IMG>
* 体可参考样例。其中：
* * 属性 x 为水平方向起始点坐标（默认为0）
* * 属性 y 为垂直方向起始点坐标（默认为0）
* * 属性 w 为logo图片最大宽度（默认为50），最小值为20，最大值为100。logo图片的高度
*/

```

```

async printLabel() {
  let list = [];
  //第一个标签
  list.push("<PAGE>");
  list.push("<SIZE>40,30</SIZE>"); //设定标签纸尺寸
  list.push("<TEXT x=\"8\" y=\"8\" w=\"1\" h=\"1\" r=\"0\">");
  list.push("#001" + await this.repeat(" ", 4));
  list.push("一号桌" + await this.repeat(" ", 4));
  list.push("1/3");
  list.push("</TEXT>");
  list.push("<TEXT x=\"8\" y=\"96\" w=\"2\" h=\"2\" r=\"0\">");
  list.push("黄金炒饭");
  list.push("</TEXT>");
  list.push("<TEXT x=\"8\" y=\"200\" w=\"1\" h=\"1\" r=\"0\">");
  list.push("王女士" + await this.repeat(" ", 4));
}

```

```

list.push("136****3388");
list.push("</TEXT>");
list.push("</PAGE>");
//第二个标签
list.push("<PAGE>");
list.push("<TEXT x=\"8\" y=\"8\" w=\"1\" h=\"1\" r=\"0\">");
list.push("#001" + await this.repeat(" ", 4));
list.push("一号桌" + await this.repeat(" ", 4));
list.push("2/3");
list.push("</TEXT>");
list.push("<TEXT x=\"8\" y=\"96\" w=\"2\" h=\"2\" r=\"0\">");
list.push("凉拌青瓜");
list.push("</TEXT>");
list.push("<TEXT x=\"8\" y=\"200\" w=\"1\" h=\"1\" r=\"0\">");
list.push("王女士" + await this.repeat(" ", 4));
list.push("136****3388");
list.push("</TEXT>");
list.push("</PAGE>");
//第三个标签
list.push("<PAGE>");
list.push("<TEXT x=\"8\" y=\"8\" w=\"1\" h=\"1\" r=\"0\">");
list.push("#001" + await this.repeat(" ", 4));
list.push("一号桌" + await this.repeat(" ", 4));
list.push("3/3");
list.push("</TEXT>");
list.push("<TEXT x=\"8\" y=\"96\" w=\"2\" h=\"2\" r=\"0\">");
list.push("老刘家肉夹馍");
list.push("</TEXT>");
list.push("<TEXT x=\"8\" y=\"200\" w=\"1\" h=\"1\" r=\"0\">");
list.push("王女士" + await this.repeat(" ", 4));
list.push("136****3388");
list.push("</TEXT>");
list.push("</PAGE>");
//第四个标签 打印条形码
list.push("<PAGE>");
list.push("<TEXT x=\"8\" y=\"8\" w=\"1\" h=\"1\" r=\"0\">");
list.push("打印条形码：");
list.push("</TEXT>");
list.push("<BC128 x=\"16\" y=\"32\" h=\"32\" s=\"1\" n=\"2\" w=\"2\" r=\"0\">");
list.push("12345678");
list.push("</BC128>");
list.push("</PAGE>");
//第五个标签 打印二维码，宽度最小为128 低于128会无法扫描
list.push("<PAGE>");
list.push("<TEXT x=\"8\" y=\"8\" w=\"1\" h=\"1\" r=\"0\">");
list.push("打印二维码宽度128：");
list.push("</TEXT>");
list.push("<QR x=\"16\" y=\"32\" w=\"128\">");
list.push("https://www.xpyun.net");
list.push("</QR>");

```

```

list.push("</PAGE>");
module.printLabel({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //打印内容
  list: list,
  //打印份数
  copies: 1,
  //声音播放模式, 0 为取消订单模式, 1 为静音模式, 2 为来单播放模式, 3为有用户申请退单
  voice: 2,
  //打印模式:
  //值为 0 或不指定则会检查打印机是否在线, 如果不在线 则不生成打印订单, 直接返回设备不
  //值为 1不检查打印机是否在线, 直接生成打印订单, 并返回打印订单号。如果打印机不在线, 订
  mode: 0
}, res => {

})
}

//原生插件中的字符串重复打印, 此处做同步处理
repeat(str, repeat) {
  return new Promise(resolve => {
    module.repeat({
      str: str,
      repeat: repeat
    }, res => {
      let result = "";
      if (res.code == 0) {
        result = res.data.content;
      }
      resolve(result)
    })
  })
}

//商品列格式化处理, 此处做同步处理
formatPrintOrderItem(foodName, quantity, price) {
  return new Promise(resolve => {
    module.formatPrintOrderItem({
      foodName: foodName,
      quantity: quantity,
      price: price
    }, res => {
      let result = "";
      if (res.code == 0) {
        result = res.data.content;
      }
      resolve(result)
    })
  })
}

```

使用方法

```
}  
}
```

示例文件

请到插件首页通过 [使用 HBuilderX 导入示例项目](#)

更新日志

2025-09-09

首次发布

初始化

方法名

init

在使用其他接口之前必须先初始化

用法

- 用法如下：

```
module.init({
  /**必填*：开发者ID ：芯烨云后台注册账号（即邮箱地址或开发者ID），开发者用户注册成功之后，登
  userName: "xxxxxxxxxxxxxxxx",
  /**必填*：开发者密钥 ：芯烨云后台注册账号后自动生成的开发者密钥，开发者用户注册成功之后，登
  userKey: "xxxxxxxxxxxxxxxx"
}, 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
userName	String	是	无	开发者ID ：芯烨云后台注册账号（即邮箱地址或开发者ID），开发者用户注册成功之后，登录芯烨云后台，在【个人中心=》账号信息】下可查看开发者ID

初始化

参数名	参数类型	是否必填	默认值	参数描述
userKey	String	是	无	开发者密钥：芯 烨云后台注册账 号后自动生成的 开发者密钥，开 发者用户注册成 功之后，登录芯 烨云后台，在 【个人中心=》账 号信息】下可查 看开发者密钥

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

批量添加打印机

方法名

addPrinters

用法

- 用法如下：

```
module.addPrinters({
  //打印机列表
  list: [{
    //打印机设备编号
    sn: "xxxxxxxxxxxxxxxx",
    //打印机名称
    name: "xxxxxxxxxxxxxxxx"
  }]
}, 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
list	Array[Object]	是	无	打印机列表
list.sn	String	是	无	打印机设备编号
list.name	String	是	无	打印机名称

回调

- 示例

```
{
  "data": {
    "successList": [],
    "failList": []
  },
  "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.successList	Array[String]	添加成功的打印机列表
data.failList	Array[String]	添加失败的打印机列表
code	Integer	返回类型，0.成功，其他：失败

设置打印机语音类型

方法名

setVoiceType

用法

- 用法如下：

```
module.setVoiceType({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //语音类型,0真人语音（大） 1真人语音（中） 2真人语音（小） 3 滴滴声 4 静音
  voiceType: 4
}, 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
sn	String	是	无	打印机设备编号
voiceType	Integer	是	无	语音类型,0真人语音（大） 1真人语音（中） 2真人语音（小） 3 滴滴声 4 静音

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

清空打印队列

方法名

delPrinterQueue

用法

- 用法如下：

```
module.delPrinterQueue({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
}, 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
sn	String	是	无	打印机设备编号

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

查询订单是否打印成功

方法名

queryOrderState

用法

- 用法如下：

```
module.queryOrderState({
  //订单编号
  orderId: "xxxxxxxxxxxxxxxxx"
}, 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
orderId	String	是	无	订单ID

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

查询订单是否打印成功

参数名	参数类型	参数描述
data.status	Boolean	true:成功，false：失败
code	Integer	返回类型，0.成功，其他：失败

查询打印机某天的订单统计数

方法名

queryOrderStatis

用法

- 用法如下：

```
module.queryOrderStatis({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //查询日期，格式yyyy-MM-dd，如：2020-09-09
  date: "2025-08-20"
}, 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
sn	String	是	无	打印机设备编号
date	String	是	无	查询日期，格式 yyyy-MM-dd， 如：2020-09-09

回调

- 示例

```
{
  "data": {
    "printed": 10,
    "waiting": 0
  },
  "message": "",
  "code": 0
}
```

查询打印机某天的订单统计数

- 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.printed	Integer	已经打印的数量
data.waiting	Integer	等待打印的数量
code	Integer	返回类型，0.成功，其他：失败

获取单个打印机状态

方法名

queryPrinterStatus

用法

- 用法如下：

```
module.queryPrinterStatus({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxxxx"
}, 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
sn	String	是	无	打印机设备编号

回调

- 示例

```
{
  "data": {
    "status": 1
  },
  "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

获取单个打印机状态

参数名	参数类型	参数描述
data.status	Integer	0：离线，1：在线正常，2：在线缺纸
code	Integer	返回类型，0.成功，其他：失败

备注：离线的判断是打印机与服务器失去联系超过 30 秒

批量获取打印机状态

方法名

queryPrintersStatus

用法

- 用法如下：

```
module.queryPrintersStatus({
  //打印机设备编号数组
  list: ["xxxxxxxxxxxxxxxxx"],
}, 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
list	String	是	无	打印机设备编号列表

回调

- 示例

```
{
  "data": {
    "status": [1]
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

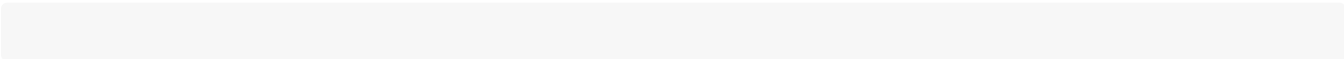
批量获取打印机状态

参数名	参数类型	参数描述
data	Object	数据对象
data.status	Array[Integer]	0：离线，1：在线正常，2：在线缺纸
code	Integer	返回类型，0.成功，其他：失败

备注：离线的判断是打印机与服务器失去联系超过 30 秒

金额播报

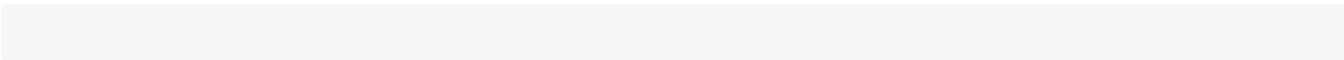
- 用法如下：



- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
sn	String	是	无	打印机设备编号
payType	Integer	是	无	取值范围 41~55；支付宝 41、微信 42、云 支付 43、银联刷 卡 44、银联支付 45、会员卡消费 46、会员卡充值 47、翼支付 48、 成功收款 49、嘉 联支付 50、壹钱 包 51、京东支付 52、快钱支付 53、威支付 54、 享钱支付 55
payMode	Integer	是	无	支付状态；取值 范围59~61；退 款 59 到账 60 消 费 61，仅用于支 持金额播报的芯 烨云打印机
money	float	是	无	支付金额；最多 允许保留2位小 数，仅用于支持 金额播报的芯烨 云打印机

- 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
data.status	String	播报状态
code	Integer	返回类型，0.成功，其他：失败

打印内容

方法名

print

用法

- 用法如下：

```
module.print({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //打印内容
  list: list,
  //打印份数
  copies: 1,
  //声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用户申请退单了。默认
  voice: 2,
  //打印模式：
  //值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不在线状
  //值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，订单将结
  mode: 0
}, res => {

})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
sn	String	是	无	打印机设备编号
list	Array[String]	是	无	打印内容集合， 具体可参考示例 程序
copies	Integer	否	1	打印份数

参数名	参数类型	是否必填	默认值	参数描述
voice	Integer	否	2	声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用用户申请退单了。默认为 2 来单播放模式
mode	Integer	否	2	打印模式；值为 0 或不指定则会检查打印机是否在线，如果不在 线 则不生成打印 订单，直接返回 设备不在线状态 码；如果在线则 生成打印订单， 并返回打印订单 号；值为 1不检 查打印机是否在线，直接生成打 印订单，并返回 打印订单号。如 果打印机不在 线，订单将缓存 在打印队列中， 打印机正常在线 时会自动打印

参数名	参数类型	是否必填	默认值	参数描述
payType	Integer	否	无	取值范围 41~55；支付宝 41、微信 42、云 支付 43、银联刷 卡 44、银联支付 45、会员卡消费 46、会员卡充值 47、翼支付 48、 成功收款 49、嘉 联支付 50、壹钱 包 51、京东支付 52、快钱支付 53、威支付 54、 享钱支付 55
payMode	Integer	否	无	支付状态；取值 范围59~61；退 款 59 到账 60 消 费 61，仅用于支 持金额播报的芯 烨云打印机
money	float	是	否	支付金额；最多 允许保留2位小 数，仅用于支持 金额播报的芯烨 云打印机

回调

- 示例

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

打印内容

- 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.orderId	String	订单ID
code	Integer	返回类型，0.成功，其他：失败

标签打印

方法名

printLabel

用法

- 用法如下：

```
module.printLabel({
  //打印机设备编号
  sn: "xxxxxxxxxxxxxxxx",
  //打印内容
  list: list,
  //打印份数
  copies: 1,
  //声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用户申请退单了。默认
  voice: 2,
  //打印模式：
  //值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不在线状
  //值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，订单将结
  mode: 0
}, res => {

})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
sn	String	是	无	打印机设备编号
list	Array[String]	是	无	打印内容集合， 具体可参考示例 程序
copies	Integer	否	1	打印份数

参数名	参数类型	是否必填	默认值	参数描述
voice	Integer	否	2	声音播放模式，0 为取消订单模式，1 为静音模式，2 为来单播放模式，3为有用用户申请退单了。默认为 2 来单播放模式
mode	Integer	否	2	打印模式；值为 0 或不指定则会检查打印机是否在线，如果不在线 则不生成打印订单，直接返回设备不在线状态码；如果在线则生成打印订单，并返回打印订单号；值为 1不检查打印机是否在线，直接生成打印订单，并返回打印订单号。如果打印机不在线，订单将缓存在打印队列中，打印机正常在线时会自动打印

参数名	参数类型	是否必填	默认值	参数描述
payType	Integer	否	无	取值范围 41~55；支付宝 41、微信 42、云 支付 43、银联刷 卡 44、银联支付 45、会员卡消费 46、会员卡充值 47、翼支付 48、 成功收款 49、嘉 联支付 50、壹钱 包 51、京东支付 52、快钱支付 53、威支付 54、 享钱支付 55
payMode	Integer	否	无	支付状态；取值 范围59~61；退 款 59 到账 60 消 费 61，仅用于支 持金额播报的芯 烨云打印机
money	float	是	否	支付金额；最多 允许保留2位小 数，仅用于支持 金额播报的芯烨 云打印机

回调

- 示例

```
{
  "data": {
    "orderId": "****"
  },
  "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.orderId	String	订单ID
code	Integer	返回类型，0.成功，其他：失败

字符串替换

方法名

repeat

用法

- 用法如下：

```
module.repeat({
  str: " ",
  repeat: 16
}, res => {
  let result = "";
  if (res.code == 0) {
    result = res.data.content;
  }
  resolve(result)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
str	String	是	无	要替换的字符
repeat	Integer	是	无	替换数量

回调

- 示例

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

- 回调说明：

字符串替换

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

格式化订单数据

方法名

formatPrintOrderItem

用法

- 用法如下：

```
module.formatPrintOrderItem({
  foodName: "可乐鸡翅",
  quantity: 2,
  price: 9.99
}, res => {
  let result = "";
  if (res.code == 0) {
    result = res.data.content;
  }
  resolve(result)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
foodName	String	是	无	名称
quantity	Integer	是	无	质量
price	float	是	无	价格

回调

- 示例

```
{
  "data": {
    "content": "****"
  },
  "message": "",
}
```

格式化订单数据

```
"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.content	String	转换后的内容
code	Integer	返回类型，0.成功，其他：失败