PC 端 Web SDK 原生接口说明文档 V4.0.6

文档修改记录

序号	版本号	修改内容	修改者	修改日期
1	v3.1.2	• 文档建立	周功成	2021/4/7
2	v3.1.4	• 更新 3.1.4 内容	石坤	2021/11/15
3	v3.1.8	• 兼容新机型 • 修复旋转后裁切异常 bug	石坤	2022/8/8
4	v3.1.8	更新接口封装	张彬	2023/6/7
5	v3.2.1	• 支持 K3 及 K3W 机型 • 增加 WIFI 相关接口	张彬	2023/10/12
6	v3.2.2	• 支持 M2 机型 • 增加 M2 相关说明	张彬	2023/10/30
7	v3.2.5	• 支持 B3S_P 机型 • 支持 B21S 机型 • 支持 B31 机型 • 更新图像库	张彬	2024/9/12
8	v4.0.3	• 支持 M3、K2、B21Pro 系列机型 • 完善错误码 • 新增绘制带 logo 二维码接口	张彬	2025/4/29

		• 提高 Websocket 通讯速度 5.Demo 支持黑标间隙纸		
9	v4.0.6	• 新增 closePrinter接口 • 修复Wifi搜索接口BUG • 修复历史遗留 WIFI连接BUG	张彬	2025/9/13

DEMO 目录结构

代码块

```
Demo/
├── drawParameter/                    # 绘制接口JSON参数的示例数据
├── img/                             # 测试图片
├── js/                              # DEMO引用JS
│   ├── api/                         # 存放接口文件的文件夹
│   │   └── jcPrinterSdk_api_third.js   # 接口文件
│   ├── printData/                   # 存放打印数据的文件夹
│   │   ├── text.js                  # 文本打印及预览示例数据
│   │   ├── barcode.js               # 一维码打印及预览示例数据
│   │   ├── qrCode.js                # 二维码打印及预览示例数据
│   │   ├── line.js                  # 线条打印及预览示例数据
│   │   ├── graph.js                 # 图形打印及预览示例数据
│   │   ├── img.js                   # 图片打印及预览示例数据
│   │   ├── combination.js           # 组合打印及预览示例数据
│   │   ├── batch.js                 # 批量打印及预览示例数据
│   │   ├── fixed_asset.js           # 固定资产及预览示例数据
│   │   ├── inspection_testing.js    # 检验检测打印及预览示例数据
│   │   └── warehousing_manufacturing.js   # 仓储物流打印及预览示例数据
│   └── iJcPrinterSdk_third.js       # 接口调用示例
├── index.css                        # 示例DEMO页面样式
└── index.html                       # 示例DEMO页面html代码
```

产品目的

JCAPI 接口为调用者提供易用的方法完成标签绘图、打印操作。本接口中提供了标贴的绘制方法，包括：文字、一维码、二维码，图形、线条、图像绘制，同时还能进行绘制对象的旋转，调用者还可以调用方法获得绘制完成的标签图片用于标签预览，打印。方便用户在二次开发中调用接口，缩短开发周期，加快开发

打印机支持

支持打印机型号
B1
B203
B21 /B21_Pro/B21S
B3S / B3S_P
B31
B4
B11
K2
K3/K3W
B50/B50W
B32/Z401/B32R
M2
M3

准备工作

- 安装精臣打印服务（jcPrinterSdk.exe）
  - 前置：关闭杀毒软件（如 360，易误报）
  - 关键：**必须默认路径安装（C 盘）**
  - 注意：勿禁用服务开机启动

- 安装对应机型驱动

机型系列	系统要求
B50/B11	Win7/10/11 均需装驱动
其他机型	Win10/11 无需装，仅 Win7 需装

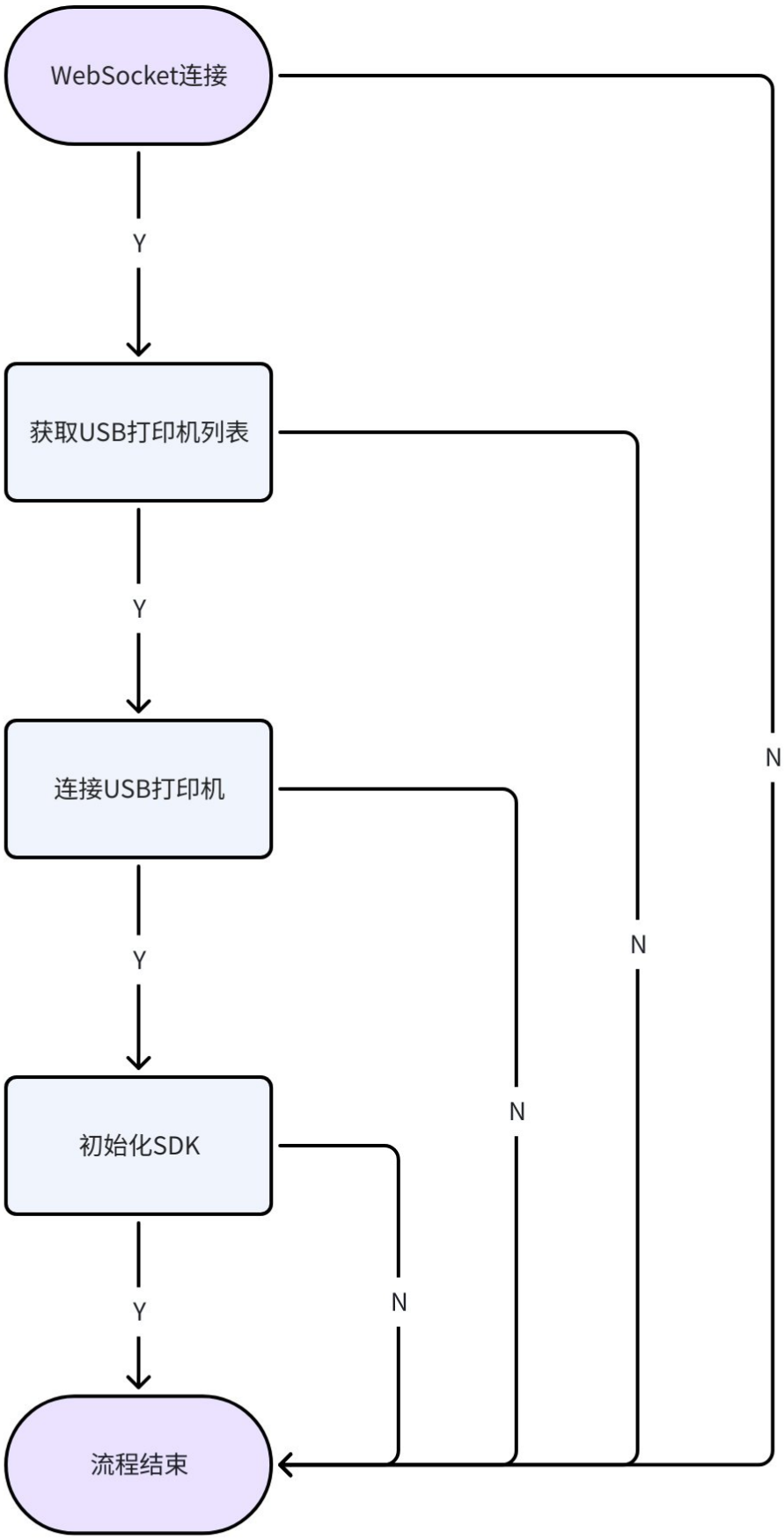
- 设备连接（2 种方式，不支持蓝牙）
  - USB 连接
    - 系统：仅支持 Windows
    - 驱动：可能需装（参考第 2 步）
    - 特别：**不支持驱动打印**（已用驱动打印需下载专用驱动
  - WIFI 连接
    - 机型：**仅支持 K3W 机型**
    - 系统：仅支持 Windows
    - 驱动：无需安装

一、初始化及打印调用流程、打印流程

1.1 初始化流程

1.1.1 USB 打印初始化流程

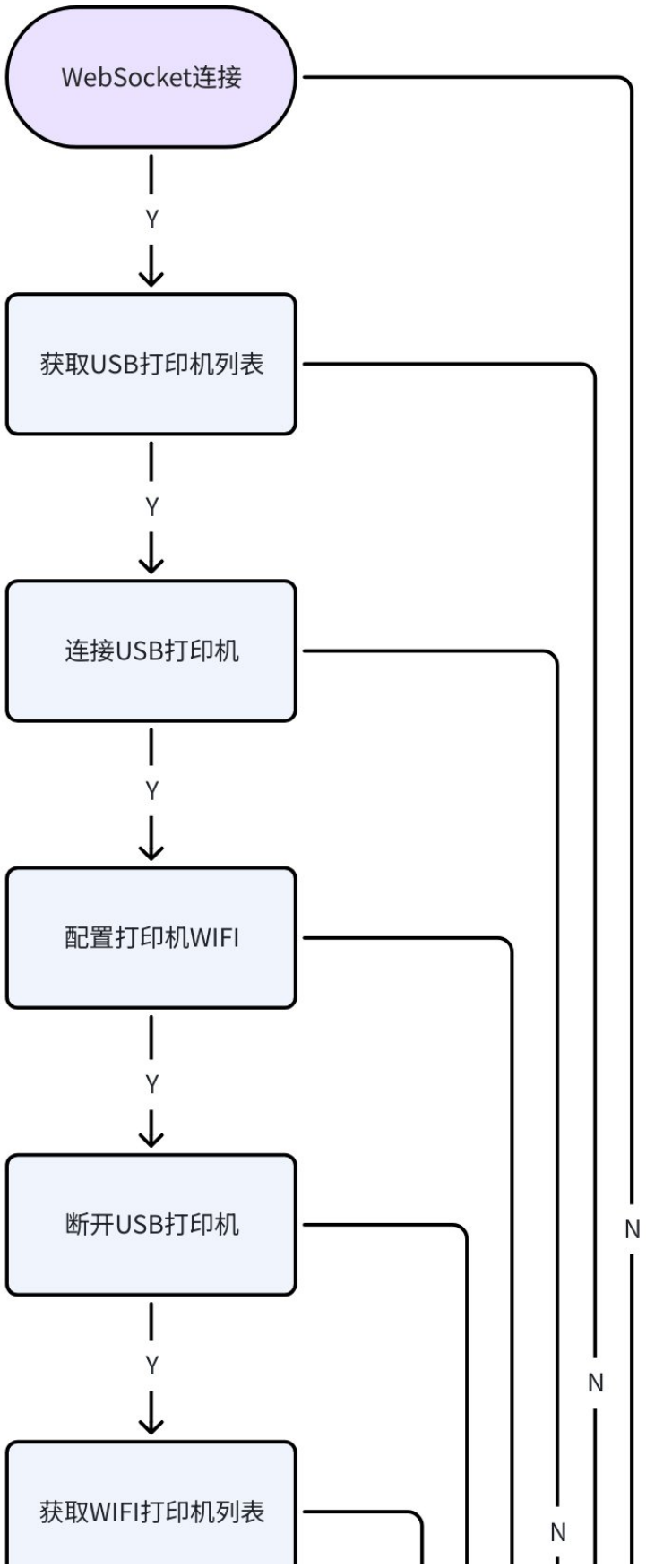
- WebSocket 建议页面加载时进行初始化，在 WebSocket 初始成功后回调中进行获取打印机、选择打印机、初始化 SDK 等操作
- 因为所有接口均为异步操作，调用下一接口需要在当前接口回调中执行
- 记录打印机列表获取状态、连接状态、初始化状态，打印机需要检查对应的状态

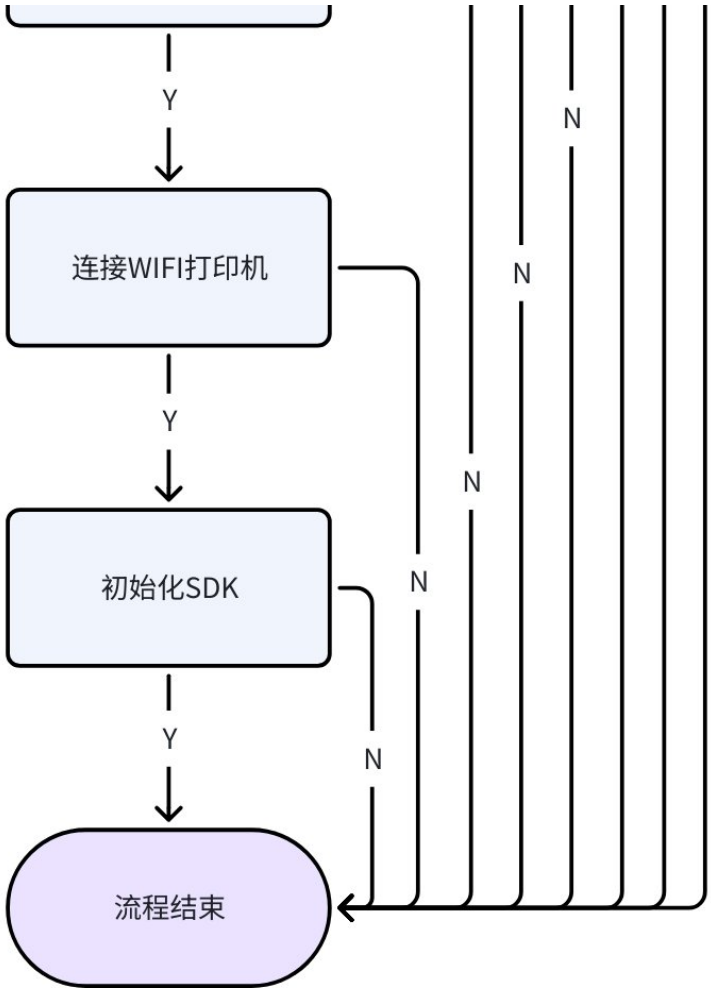


1.1.2 WIFI 打印初始化流程

- WebSocket 建议页面加载时进行初始化，在 WebSocket 初始成功后回调中进行获取打印机、选择打印机、初始化 SDK 等操作

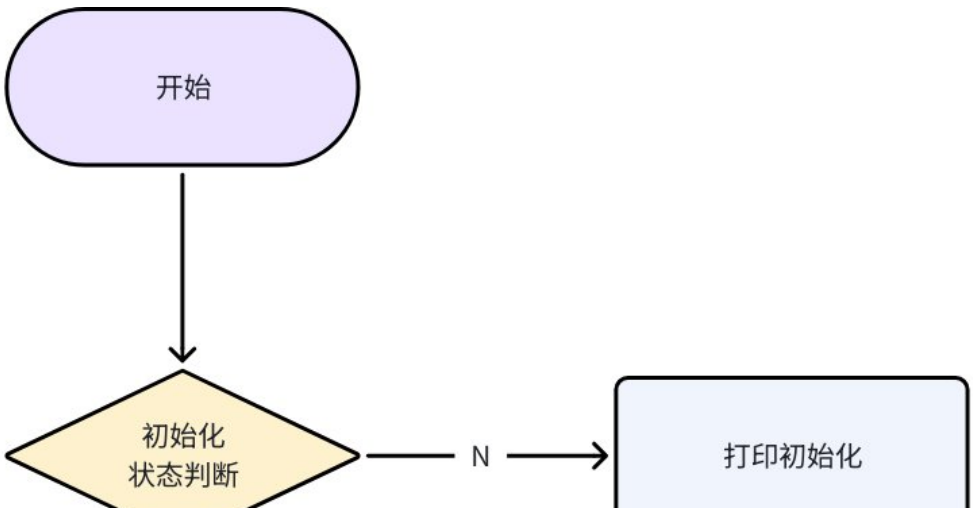
- 因为所有接口均为异步操作，调用下一接口需要在当前接口回调中执行
- 记录打印机列表获取状态、连接状态、初始化状态，打印机需要检查对应的状态
- **打印机 WIFI 配置成功后，后续直接搜索连接，无需多次进行配置（省略 USB 打印机获取、打印机连接、打印机网络配置）**

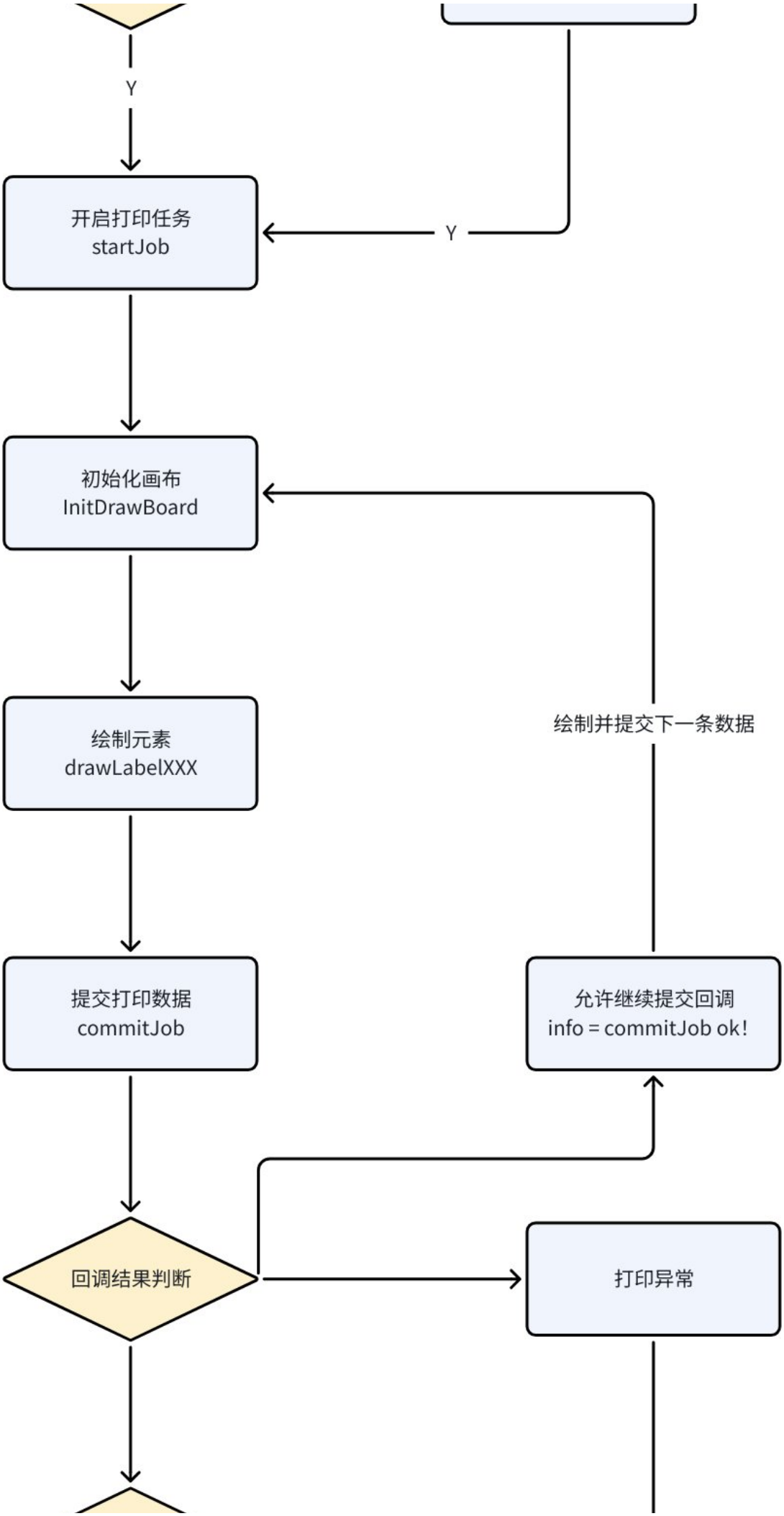


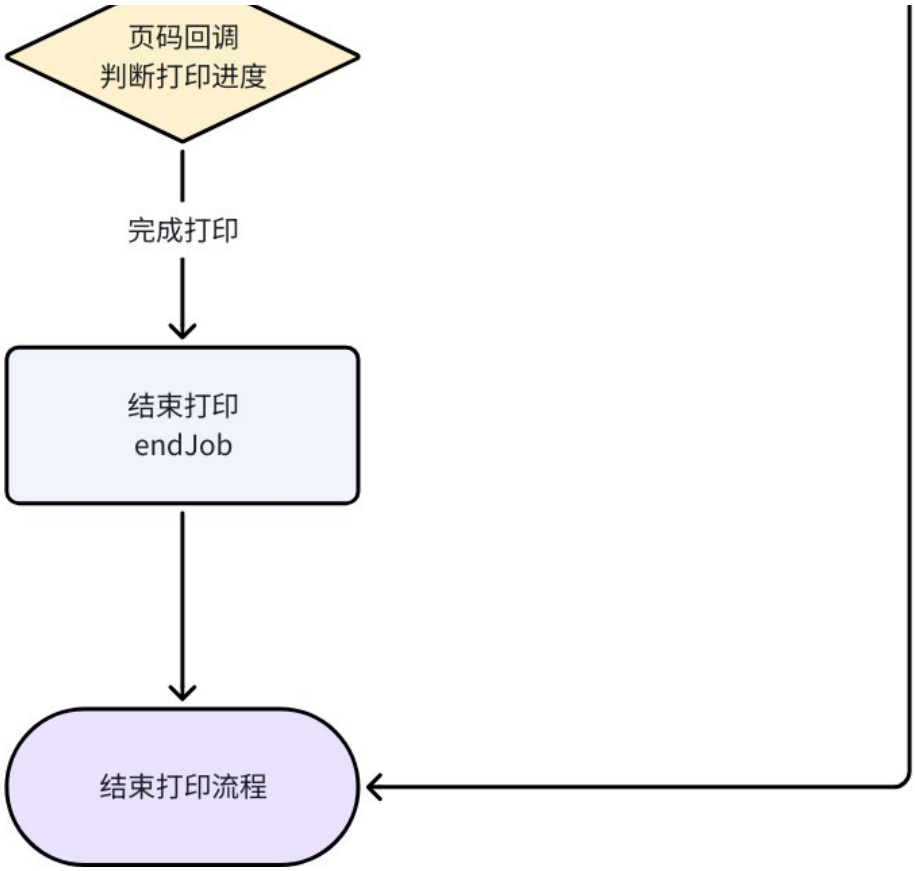


1.2 打印流程

- 打印前建议判断 WebSocket 是否初始化成功、SDK 是否初始化成功（包含初始化 SDK，获取打印机、选择打印机三个流程）
- 因为所有接口均为异步操作，除 WebSocket 初始化调用是在单独的回调接口中判断是否初始化成功外，其他接口应在回调中解析数据进行判断后再进行下一接口调用
- commitJob 会有多种回调，包含异常取消、页码回调、提交成功回调，可参考流程图及 DEMO 进行处理




Y
开启打印任务
startJob
Y
初始化画布
InitDrawBoard
绘制元素
drawLabelXXX
绘制并提交下一条数据
提交打印数据
commitJob
允许继续提交回调
info = commitJob ok!
回调结果判断
打印异常



- 流程注意事项：
  - 默认初始化检查不通过发起初始化必定成功，开启任务，绘制过程成功（异常状态多为参数问题和打印机问题，参数问题开发阶段可以完全解决）

二、页面初始化相关接口

2.1 初始化打印服务

代码块

```
/**
 * 初始化打印服务，连接打印服务
 *
 * @param {function} onServiceConnected - 当打印服务连接建立时调用的回调函数。
 * @param {function} onNotSupportedService - 当打印服务服务不支持时调用的回调函数。
 * @param {function} onServiceDisconnected - 当打印服务连接断开时调用的回调函数。
 * @param {function} onPrinterDisConnect - 当打印机离线时调用的回调函数。
 * @return {undefined} 该函数没有返回值。
 *
 * @description
 * 1. 所有接口的调用前提是先调用该接口进行打印服务连接。
 * 2. 调用成功后会停止初始化打印服务，如果未调用成功，会间隔3秒调用一次，直到成功连接为止。
 * 3. 建议在页面加载时调用该接口，回调成功后依次调用初始化SDK、获取打印机、选择打印机等接口。
```

```
 */
function getInstance(onServiceConnected, onNotSupportedService,
onServiceDisconnected, onPrinterDisConnect)
```

代码块

```
//页面加载时调用
getInstance(() => {
                console.log('打印服务连接成功');
        }, () => {
                console.log('当前浏览器不支持打印服务');
        }, () => {
                console.log('打印服务连接断开');
        }, () => {
                console.log('打印离线');
        });
```

2.2 初始化 SDK initSdk

代码块

```
/**
 * 初始化SDK，在打印服务连接成功后调用此接口。
 * 在调用SDK的绘制接口之前，必须先调用此接口。
 *
 * @param {object} json - 包含必要参数的JSON对象,格式如下:
 *  {
 *    "fontDir": string, //字体文件目录，默认为""，暂不生效
 * }
 * @param {function} callbackFunction - 发送消息后执行的回调函数。
 * @return {undefined} 该函数没有返回值。
 */
function initSdk(json, callbackFunction)
```

代码块

```
//初始化SDK参数JSON
{
  "fontDir": ""
}
//初始化成功返回JSON
{
  "apiName": "initSdk",
    "resultAck": {
      "errorCode": 0,
```

```
        "info": "initSdkApi ok!",
        "result": 0
    }
}


//调用流程
var initSdkParam = {
  "fontDir": ""
};

//进行初始化
initSdk(initSdkParam, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;
  //判断错误码
    if (errorCode === 0) {
        console.log('初始化成功');
    } else {
        console.log('初始化失败');
  //提示错误信息
        return alert(info);
    }

//初始化成功进行下一步操作

});
```

2.3 获取 USB 连接的打印机列表 getAllPrinters

代码块

```
/**
 * 获取所有打印机信息。
 *
 * @param {function} callbackFunction - 获取信息后执行的回调函数。
 * @return {undefined} 该函数没有返回值。
 *
 * @description
 * 需要在打印服务连接成功后调用此函数，建议在打印服务连接成功的回调函数中调用。
 * 注意：此函数只能获取 USB连接的打印机列表。
 */
function getAllPrinters(callbackFunction)
```

代码块

```
{
        "apiName": "getAllPrinters",
        "resultAck": {
                "errorCode": 0,
                "info": "{\"e623012991\":\"31\"}",//打印机名称及类型
                "result": "true"
        }
}


//调用流程
getAllPrinters(function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }

    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;
    //获取失败退出
    if (errorCode !== 0) {
        return;
    } else {
        //获取成功解析数据，数据包含序列号及端口
        allPrinters = JSON.parse(info)
        let allPrintersName = Object.keys(allPrinters);
        let allPrintersValue = Object.values(allPrinters);
    }
});
```

2.4 获取 WIFI 连接的打印机列表 scanWifiPrinter

代码块

```
/**
 * 搜索Wifi打印机
 *
 * @param {function} callbackFunction - 获取信息后执行的回调函数。
 * @return {undefined} 该函数没有返回值。
 *
 * @description
 * 需要在打印服务连接成功后调用此函数，建议在打印服务连接成功的回调函数中调用。
 */
```

```
function scanWifiPrinter(callbackFunction) {
  //刷新设备时，关闭设备
  //closePrinter();
  var jsonObj = { apiName: 'scanWifiPrinter' };

  sendMsg(jsonObj, callbackFunction);
}
```

代码块

```
//返回结果
{
        "apiName": "scanWifiPrinter",
        "resultAck": {
                "errorCode": 0,
                "info": "[{
      "deviceName":"K3W-E828013369",
                        "IP":"192.168.1.10",
                        "tcpPort":"9200",
                        "avaliableClient":"0"
          }]",
    "result": "true"
        }
}


//调用流程
scanWifiPrinter(function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }

    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;
    //获取失败退出
    if (errorCode !== 0) {
        return;
    } else {

        let deviceAndPortArray = info.map(function(item) {
           return {
               deviceName: item.deviceName,
               tcpPort: item.tcpPort
           };
        });
    }
```

```
});
```

2.5 连接 USB 打印机列表中的打印机 selectPrinter

代码块

```
/**
 * 发送消息以选择打印机。
 *
 * @param {string} printerName - 打印机名称。
 * @param {number} port - 端口号。
 * @param {function} callbackFunction - 消息发送后的回调函数。
 * @return {undefined} 无返回值。
 *
 * @description
 * 需要在打印服务连接成功后调用此函数，建议在getAllPrinters调用成功的回调接口中调用该接口，保证传入的打印机名称和端口的打印机状态正常。
 * 注意：此函数仅能连接 USB 打印机列表中的打印机。
 */
function selectPrinter(printerName, port, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "selectPrinter",
        "resultAck": {
                "callback": {
                        "name": "onConnectSuccess",
                        "printerName": "e623012991"
                },
                "errorCode": 0,
                "info": "select printer ok!",
                "result": true
        }
}

selectPrinter("e623012991", 31, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }

    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    var arrParse = JSON.parse(JSON.stringify(data));
    //选择失败退出
```

```
    if (errorCode !== 0) {
        return;
    } else  {
        //选择成功，打印机连接成功
    }
});
```

2.6 连接 WIFI 打印机列表中的打印机 connectWifiPrinter

代码块

```
/**
 * 发送消息以选择打印机。
 *
 * @param {string} printerName - 打印机名称。
 * @param {number} port - 端口号。
 * @param {function} callbackFunction - 消息发送后的回调函数。
 * @return {undefined} 无返回值。
 *
 * @description
 * 需要在打印服务连接成功后调用此函数，建议在scanWifiPrinter调用成功的回调接口中调用该接口，保证传入的打印机名称和端口的打印机状态正常。
 * 注意：此函数仅能连接 WIFI 打印机列表中的打印机。
 */
function connectWifiPrinter(printerName, port, callbackFunction)
```

代码块

```
//示例返回成功数据
{
        "apiName": "selectPrinter",
        "resultAck": {
                "callback": {
                        "name": "onConnectSuccess",
                        "printerName": "e623012991"
                },
                "errorCode": 0,
                "info": "select printer ok!",
                "result": true
        }
}
//示例返回失败数据
{
  "apiName":"connectWifiPrinter",
  "resultAck":{
    "callback":{
```

```
        "name":"onDisConnect",
        "printerName":"K3_W-F612010061"
      },
      "errorCode":0,
      "info":"success",
      "result":false
    }
  }

  connectWifiPrinter("K3W-E828013369", 9200, function (error, data) {
      //通讯超时或websocket打印服务异常
      if (error) {
          return alert(error.message);
      }
            //此版文报存在问题，errorCode连接成功与连接失败一致，暂时先用result判断
      const { result } = JSON.parse(JSON.stringify(data)).resultAck;

      if (result) {
        console.log('连接成功')
      } else {
        console.log('连接失败')
                            }
  });
```

2.7 断开打印机连接 closePrinter

代码块

```
  /**
   * 发送消息以断开打印机。
   *
   * @param {function} callbackFunction - 消息发送后的回调函数。
   * @return {undefined} 无返回值。
   *
   * @description
   * 可以断开USB和WIFI连接的打印机
   */
  function closePrinter(callbackFunction) {
    var jsonObj = {
      apiName: 'closePrinter',
    };
    sendMsg(jsonObj, callbackFunction);
  }
```

2.8 配置打印机的 WIFI 信息 configurationWifi

代码块

```
/**
 * 配置打印机的Wifi网络
 *
 * @param {string} wifiName - wifi网络的名称。
 * @param {string} wifiPassword - wifi网络的密码。
 * @param {function} callbackFunction - wifi配置完成后要调用的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 注意:仅支持2.4G频段网络，且需要在连接成功后配置。需在USB连接成功后配置
 */
function configurationWifi(wifiName, wifiPassword, callbackFunction)
```

代码块

```
//示例返回数据
{
  "apiName":"configurationWifi",
  "resultAck":{
    "errorCode":0,
    "info":"configuration wifi printer ok!",
    "result":true
  }
}

let name = document.getElementById('wifi_name');
let password = document.getElementById('wifi_password');
if(name.value.trim()!== ""){
  configurationWifi(name.value, password.value, function (error, data) {
    if (error) {
      return alert(error.message);
    }

    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;

    if (errorCode === 0) {
      return alert("网络配置成功，请断开USB线缆后使用WIFI搜索连接打印机（PC需要和打印机在同一网络）");
    } else {
      return alert(info);
    }


  });
```

```
}
```

2.9 获取打印机的 WIFI 相关配置 getWifiConfiguration

代码块

```
/**
 * 获取打印机的wifi配置。
 *
 * @param {function} callbackFunction - 在获取WiFi配置后将调用的回调函数。
 * @return {undefined} 此函数不返回任何值。
 */
function getWifiConfiguration(callbackFunction)
```

代码块

```
//示例返回成功数据
 {
   "apiName":"getWifiConfiguration",
     "resultAck":{
     "errorCode":0,
     "info":"{
       \n\t\"wifiName\" : \"Test\"\n
       }\n",
      "result":"{
       \n\t\"wifiName\" : \"Test\"\n
       }\n"
   }
 }
//示例返回失败数据
 {
   "apiName":"getWifiConfiguration",
     "resultAck":{
     "errorCode":23,
     "info":"select printer connect first!",
     "result":false
   }
 }

getWifiConfiguration(function (error, data) {
    if (error) {
      return alert(error.message);
    }

```

```
    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;

    if (errorCode === 0) {
      const infoObj = JSON.parse(info)

      return alert("获取配置信息成功-Wifi名称为:" + infoObj.wifiName);
    } else {
      return alert(info);
    }
});
```

三、打印机状态回调

3.1 上盖状态变化回调

代码块

```
/**
 * 打印机上盖状态变化回调函数。
 *
 * @param {type} coverStatus - 打印机上盖状态，0为上盖开启,1为上盖关闭
 *
 */
function onCoverStatusChange(coverStatus)
```

3.2 电量变化回调

代码块

```
/**
 * 打印机电量变化回调函数。
 *
 * @param {type} powerLever - 电量等级，取值范围1-4，满格电为4
 *
 */
function onElectricityChange(powerLever)
```

四、绘制打印数据相关接口

4.1 创建画板 InitDrawingBoard

代码块

```
/**
```

```
 * 初始化绘制画板
 *
 * @param {Object} json - 包含初始化绘制画板所需数据的JSON对象。格式如下:
 * {
 *   "width": number, // 画板的宽度，单位为mm
 *   "height": number, // 画板的高度，单位为mm
 *   "rotate": number, // 画板的旋转角度，仅支持0、90、180、270
 *   "path": string, // 字体文件的路径，默认为""，暂不生效
 *   "verticalShift": number, // 垂直偏移量，暂不生效
 *   "HorizontalShift": number // 水平偏移量，暂不生效
 * }
 * @param {Function} callbackFunction - 消息发送后要执行的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 增加接口说明:
 * 1.在调用绘制接口之前，必须先初始化SDK。
 * 2.绘制元素前，必须先初始化画板，否则会引起崩溃!
 * 3.初始化画板时会清空画板上次绘制的内容!
 */
function InitDrawingBoard(json, callbackFunction)
```

代码块

```
{
        "apiName": "InitDrawingBoard",
        "resultAck": {
                "errorCode": 0,
                "info": "init draw board success!",
                "result": 0
        }
}


var InitDrawingBoardParam={
  "width":48,
  "height":30,
  "rotate":0,
  "path":"ZT001.ttf",
  "verticalShift":0,
  "HorizontalShift":0};

//SDK初始化成功后调用
InitDrawingBoard(InitDrawingBoardParam, function (error, data) {
```

```
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }


    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //画布初始化失败，退出流程
    if (errorCode !== 0) {
        return
    }

    //画布初始化成功，进行控件绘制

});
```

4.2 绘制文本 DrawLableText

代码块

```
/**
 * 绘制标签文本。
 * @param {object} json - 包含标签文本信息的JSON对象。
 *   JSON格式要求如下:
 *   - x: x轴坐标，单位mm
 *   - y: y轴坐标，单位mm
 *   - height: 文本高度，单位mm
 *   - width: 文本宽度，单位mm
 *   - value: 文本内容
 *   - fontFamily: 字体名称，暂不生效，使用默认字体思源黑体
 *   - rotate: 旋转角度，0:0，1:90，2:180，3:270
 *   - fontSize: 字号，单位mm
 *   - textAlignHorizonral: 水平对齐方式：0:左对齐 1:居中对齐 2:右对齐
 *   - textAlignVertical: 垂直对齐方式：0:顶对齐 1:垂直居中 2:底对齐
 *   - letterSpacing: 字母之间的标准间隔，单位mm
 *   - lineSpacing: 行间距（倍距），默认1
 *   - lineMode: 1:宽高固定，内容大小自适应，预设宽高过大时字号放大，预设宽高过小时字号缩小，
 *     保证内容占据满预设宽高（字号/字符间距/行间距 按比例缩放）
 *     2:宽度固定，高度自适应
 *     4:宽高固定,超出内容直裁切
 *     6:宽高固定，内容超过预设的文本宽高自动缩放
 *     建议设置为6
 *   - fontStyle: 字体样式[加粗，斜体，下划线，删除下划线（预留）]
 * @param {function} callbackFunction - 绘制完成后执行的回调函数。
 * @description 绘制标签文本前必须先初始化画板
```

```
 */
function DrawLableText(json, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "DrawLableText",
        "resultAck": {
                "errorCode": 0,
                "info": "draw bar code success!",//此处返回信息有误，下个版本修复
                "result": 0
        }
}

var DrawLableTextParam = {
        "x": 20.0,
        "y": 10.0,
        "height": 10,
        "width": 50,
        "value": "精臣SDK",
        "fontFamily": "宋体",
        "rotate": 0,
        "fontSize": 4.0,
        "textAlignHorizonral": 0,
        "textAlignVertical": 0,
        "letterSpacing": 1.0,
        "lineSpacing": 1.0,
        "lineMode": 0,
        "fontStyle": [false, false, false, false]
}

DrawLableText(DrawLableTextParam, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //文本绘制失败，退出绘制
    if (errorCode !== 0) {
        return
    }

    //进行下一步操作
});
```

4.3 一维码绘制 DrawLableBarCode

代码块

```
/**
 * 绘制一维码条形码。
 *
 * @param {Object} json - 包含一维码条形码信息的JSON对象。格式如下:
 * {
 *   "x": number, // x轴坐标，单位mm
 *   "y": number, // y轴坐标，单位mm
 *   "height": number, // 一维码宽度，单位mm
 *   "width": number, // 一维码高度，单位mm（包含文本高度）
 *   "value": string, // 一维码内容
 *   "codeType": number, // 条码类型:
 *                       // 20: CODE128
 *                       // 21: UPC-A
 *                       // 22: UPC-E
 *                       // 23: EAN8
 *                       // 24: EAN13
 *                       // 25: CODE93
 *                       // 26: CODE39
 *                       // 27: CODEBAR
 *                       // 28: ITF25
 *   "rotate": number, // 旋转角度，0: 0，1: 90，2: 180，3: 270
 *   "fontSize": number, // 文本字号，单位mm，字号为0则文本不显示
 *   "textHeight": number, // 文本高度，单位mm，高度为0则文本不显示
 *   "textPosition": number // 一维码文字识别码显示位置:
 *                          // 0: 下方显示
 *                          // 1: 上方显示
 *                          // 2: 不显示
 * }
 * @param {Function} callbackFunction - 消息发送后要执行的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 1. 绘制元素前，必须先初始化画板
 */
function DrawLableBarCode(json, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "DrawLableBarCode",
        "resultAck": {
                "errorCode": 0,
```

```
                "info": "draw bar code success!",
                "result": 0
        }
}

var DrawLableBarCodeParam = {
  "x": 20.0,
  "y": 10.0,
  "height": 10,
  "width": 50,
  "value": '12345678',
  "codeType": 20,
  "rotate": 0,
  "fontSize": 4.0,
  "textHeight": 0,
  "textPosition": 0,
}

DrawLableBarCode(DrawLableBarCodeParam,function(error,data){
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //一维码绘制失败，退出绘制
    if (errorCode !== 0) {
        return
    }

    //进行下一步操作
  });
```

4.4.1 二维码绘制 DrawLableQrCode

代码块

```
/**
 * 绘制二维码。
 *
 * @param {Object} json - 包含二维码信息的JSON对象。格式如下:
 * {
 *   "x": number, // x轴坐标，单位mm
 *   "y": number, // y轴坐标，单位mm
 *   "height": number, // 二维码高度，默认宽高一致
 *   "width": number, // 二维码宽度，单位mm
 *   "value": string, // 二维码内容
```

```
 *   "codeType": number, // 条码类型:
 *                       // 31: QR_CODE
 *                       // 32: PDF417
 *                       // 33: DATA_MATRIX
 *                       // 34: AZTEC
 *   "rotate": number, // 旋转角度，仅支持0、90、180、270
 * }
 * @param {Function} callbackFunction - 消息发送后要执行的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 1. 绘制元素前，必须先初始化画板
 */
function DrawLableQrCode(json, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "DrawLableQrCode",
        "resultAck": {
                "errorCode": 0,
                "info": "draw qr code success!",
                "result": 0
        }
}

var DrawLableQrCodeParam = {
        "x": 20.0,
        "y": 10.0,
  "height": 10,
  "width": 10,
  "value": "精臣SDK",
  "rotate": 0,
  "codeType": 31,
}

DrawLableQrCode(DrawLableQrCodeParam, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //二维码绘制失败，退出绘制
    if (errorCode !== 0) {
        return
```

```
    }

    //进行下一步操作
});
```

4.4.2 二维码绘制 DrawLableQrCodeWithLogo

代码块

```
/**
 * 绘制带logo的二维码。
 * @param {*} json - 包含二维码信息的JSON对象。格式如下:
 * {
 *   "x": number, // x轴坐标，单位mm
 *   "y": number, // y轴坐标，单位mm
 *   "height": number, // 二维码高度，默认宽高一致
 *   "width": number, // 二维码宽度，单位mm
 *   "value": string, // 二维码内容
 *   "codeType": number, // 条码类型:
 *                       // 31: QR_CODE
 *                       // 32: PDF417
 *                       // 33: DATA_MATRIX
 *                       // 34: AZTEC
 *   "rotate": number, // 旋转角度，仅支持0、90、180、270
 *  "correctLevel": 2,//纠错级别，取值范围1-4，默认2
 *  ""logoBase64": ": string,//logo的base64编码(不含数据头，如)
 *  ""logoPosition": ": 0,//logo的位置，取值范围0-4，默认0:居中，3右下·
 *    "logoScale": 0.25,//logo占据二维码的比例
 * }
 * @param {*} callbackFunction
 */
function DrawLableQrCodeWithLogo(json, callbackFunction)
```

4.5 线条绘制 DrawLableLine

代码块

```
/**
 * 绘制线条。
 *
 * @param {Object} json - 包含线条信息的JSON对象。格式如下:
 * {
 *   "x": number, // x轴坐标，单位mm
 *   "y": number, // y轴坐标，单位mm
 *   "height": number, // 线高，单位mm
```

```
 *   "width": number, // 线宽，单位mm
 *   "lineType": number, // 线条类型: 1:实线 2:虚线类型,虚实比例1:1
 *   "rotate": number, // 旋转角度，仅支持0、90、180、270
 *   "dashwidth": number // 线条为虚线宽度，【实线段长度，空线段长度】
 * }
 * @param {Function} callbackFunction - 消息发送后要执行的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 1. 绘制元素前，必须先初始化画板
 */
function DrawLableLine(json, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "DrawLableLine",
        "resultAck": {
                "errorCode": 0,
                "info": "draw line success!",
                "result": 0
        }
}


var DrawLableLineParam = {
        "x": 2.0,
        "y": 2.0,
  "height": 2,
  "width": 50,
  "rotate": 0,
  "lineType": 2,
  "dashwidth": [1,1],
}

DrawLableLine(DrawLableLineParam, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //线条绘制失败，退出绘制
    if (errorCode !== 0) {
        return
    }
```

```
32
33      //进行下一步操作
34  });
```

4.6 绘制图形 DrawLableGraph

代码块

```
1   /**
2    * 绘制图形。
3    *
4    * @param {Object} json - 包含绘制图形信息的JSON对象。格式如下:
5    * {
6    *   "x": number, // x轴坐标，单位mm
7    *   "y": number, // y轴坐标，单位mm
8    *   "height": number, // 图形高度，单位mm
9    *   "width": number, // 图形宽度，单位mm
10   *   "rotate": number, // 旋转角度，仅支持0、90、180、270
11   *   "cornerRadius": number, // 圆角半径，单位mm，暂不生效
12   *   "lineWidth": number, // 线宽，单位mm
13   *   "lineType": number, // 线条类型: 1:实线 2:虚线类型,虚实比例1:1
14   *   "graphType": number, // 图形类型: 1:圆，2:椭圆，3:矩形 4:圆角矩形
15   *   "dashwidth": number // 线条为虚线宽度，【实线段长度，空线段长度】
16   * }
17   * @param {Function} callbackFunction - 消息发送后要执行的回调函数。
18   * @return {undefined} 此函数不返回任何值。
19   *
20   * @description
21   * 1. 绘制元素前，必须先初始化画板
22   */
23  function DrawLableGraph(json, callbackFunction)
```

代码块

```
1   //返回数据示例
2   {
3           "apiName": "DrawLableGraph",
4           "resultAck": {
5                   "errorCode": 0,
6                   "info": "draw graph success!",
7                   "result": 0
8           }
9   }
10
11  var DrawLableGraphParam = {
12    "x": 2.0,
```

```
    "y": 5.0,
    "height": 30,
    "width": 40,
    "rotate": 0,
    "graphType": 3,
    "cornerRadius": 0,
    "lineWidth": 4,
    "lineType":2,
    "dashwidth": [1,1],
}

DrawLableGraph(DrawLableGraphParam, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //图形绘制失败，退出绘制
    if (errorCode !== 0) {
        return
    }
    //进行下一步操作
});
```

4.7 绘制图像 DrawLableImage

代码块

```
/**
 * 绘制图片。
 *
 * @param {Object} json - 包含绘制图片信息的JSON对象。格式如下:
 * {
 *   "x": number, // x轴坐标，单位mm
 *   "y": number, // y轴坐标，单位mm
 *   "height": number, // 图片高度，单位mm
 *   "width": number, // 图片宽度，单位mm
 *   "rotate": number, // 旋转角度，仅支持0、90、180、270
 *   "imageProcessingType": number, // 图像处理算法，默认0
 *   "imageProcessingValue": number, // 算法参数，默认127
 *   "imageData": number, // 图片base64数据，不含数据头
 *                        // 如原始数据为“”
 *                        // 传入的数据需要去除头部，数据为，“iVBORw0KGgoAAAANSU”
 * }
 * @param {Function} callbackFunction - 消息发送后要执行的回调函数。
 * @return {undefined} 此函数不返回任何值。
```

```
 *
 * @description
 * 增加接口说明:
 * 1. 绘制元素前，必须先初始化画板
 */
function DrawLableImage(json, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "DrawLableImage",
        "resultAck": {
                "errorCode": 0,
                "info": "draw image success!",
                "result": 0
        }
}

var DrawLableImageParam = {
        "x": 2.0,
        "y": 2.0,
         "height": 10,
         "width": 50,
         "rotate": 0,
         "imageProcessingType": 0,
         "imageProcessingValue":127,

"imageData":"iVBORw0KGgoAAAANSUhEUgAAAZAAAACgCAYAAAAisjrVAAAAAXNSR0IArs4c6QAAAA
RnQU1BAACxjwv8YQUAAAAgY0hSTQAAeiYAAICEAAD6AAAAgOgAAHUwAADqYAAAOpgAABdwnLpRPAAAA
AlwSFlzAAAOxAAADsQBlSsOGwAAROVJREFUeF7tnQmYFcW1gFtEDe64Am5BIyhxe/jcRTQquBDc17gb
FfdEfUYlKu6JccWnKIoYiIIGNVEB81TcggsEFQFN3BWjKIILaIIG+9V/pmuoqam+3ffevqFPz76b76
Z27e6urrvvXXqLHVqia+++ipeuPD7qF27JaM45m+7aIkllpCiKIqiKHEcS/n++++NbGhn/i40ssLIid
mzZ8dLLdU+at9+qegHP/hBvQBRFEVRFIsVIP/+97+j//znu2jevPnREvPnnfxf0vvvvfRS0VJLLZVVUxRF
ZR0vvvvuu2jBgm9RONqp8FAURVFyY2XGEgsXfh9pjy1pcWbhwoahwoW8+fPj/71r38Z0Z1eo/Ih27dOkSLbPM
MmqOUxRFqQHffPOvaAkz+MbJ6yDffvttNGrUqGj8+PHyvw8D9HLLLLSLcvnYzSSy8txxnYGdBD907
dOzr66KOjlVZaqX6gf+yxx6L7778/mjt3rrQ3b948aRuJt9pqq0Vvv/12NHv2bLHFffsnll/L+CiusIH
Xd69xzzzz3R/vvvL0JEURRFKRYm7JkChIH5N7/5TTR8/HA5wYEVll11VXHHCwxdffCEDOoJkxRVXD
GoCBx10UDRo0Pb9Sy+9NLr44ot44otF+Ky++urRj3/8Y/kfkk0+KUNh5552lXbSMtdZaKxxKxoYT0
2muvRUOHDo1++MMLcUC4A5zTz31VLTLLrrskrxqSMRw2Ik9bpeo0NeXeXylEHpgGq+aEE06
IBw8enLyqY/LkjH3jV199NTmSj7uuuNXrvxbTqDhNTNmSj+uuuy7eYYcdgNpJEfqeOONN2Ijo0Jhw4YlRxZx4403xuYDio
0QSY4oitISMONMs5ctt9wy6U/vb3xq8/4Mf/CBef/31k3v3xxqxd8/4MfX/CBef/31G9XLg9/WUkstm2++i
evUhRx89dVX0YwMyMyIjEBqUv/71r9H/7Hsh9HXH7778fTZs2LDDOigaA1TJkypcE5lA8//LDef+Gz
YMGC6NNNPPxUzlQVzFXXRiCguaD2Y1zBn4QNRFEXJC+PQ119/nbsAAOvXbA/8W4BpygGOfvwEpqOrNjRyGHD
J8jBy5JyN8E5YNOFvZ9UkVxuJqTCP49Ip0h9HIvYj4ypykKMUuOpmb+KRvGmP4ZqIrDSGGIJlmQXZhtRa
2K11577fiJJ56IJk2aKXQ4TRz3qaL2ZwHdoQ7YaTu4IaqSMxYd4RLM1lCiQWBVYm5jc9Jc9aJQ1iMEC
eqEirZyeCWr4Fvd1OGHBTX2Zc7LlO/MfPvw+gtSZ9NhmSBlOC3RgmuJE7M0M5SqrhgGXoNH/KB7m+yiJjn5+
```

h2i5ZdfXjQLc259Yda/7bbbRrfeemuD42Ywj1ZeeeXo2WefbXCcctFFF0XrrLNO6qJFroPfI6/9dIMN
NpAZxKxZsxpoLYqitDwYL/wxoYhiJqTJFUpD8E7ofDcIJ21pA37ZUthgnzzU6jlQ8j6LasktQBjMl1x
ySYmqooMWBApqHQ5097it59dHXcQERXu+gMAMNWfOnPrILfe8UnTs2DHaZJNNon/+85/JEUVRWiq1Mj
Nj9i4KxqkQBAS5bLbZZsl/i8g78a2lub3IZ1GK3ALEwk3bgf3555+Pjj32WImm6tOnT4MHhx2Qem6IL
iCE0qKyaBtfCxoFdWxEVxZEYvXo0SMyqmJ9iK+iKC0Td2LIGMJYUGnp3Llz0lKxAzLjT8+ePRuUjTba
SCbM7vWJBnVh4vvZZ58lr0rjT5CreRbuc4BaCieXsgUIAgDpfOedd0Y77LCDmLZOP/10MTu5MPgjDBA
w/fv3l/Bbyr777hvdddddjRzhwMBP+C9qoP/BlII+bLPNNrKG5L333sutuSiK0ry0b98++a88GCM23H
DDaI011kiOFMvGG2/cyIHO+JIF/VpllVWSV+VR6bOAWj2HLMoWIKhGCIXjjz9e/rL+AjXO1xYOOOCA6
IEHHhDz0qRJkyRaizJx4kSxz4VURKK5kMLdunWTxI5I1jSYCeyxxx4S5YXA2HvvvWXNyJgxY4LCSVGU
lkdeK4MPE0xKmq+iFjAm5aHUuFWKSp8FNOVzcCm7xziIbrvttuiNN96Ihg0bFi277LLJO43Zb7/9RCD
g3EbwUDAxnXvuueIk90HKsyiRmQWaThpI6q5du0rdm2++Ofr8889FAh988MGymJCQYkVRWi+E2TIG1Q
osGcccc0yDwoQ5C0xHdmF1W6BsAdKpUycZ/BnkKyVknnr55Zej8847LzruuOOinXbaqVEd+xqNBmnLX
7Qg0p7ccMMN4oAfMGCAmMhIjzJ9+nSpryhK6wNtYP31109eFc/HH38c/f73v29QHnnkkUaRU360ExNf
ok/bCmULEPwdDNLWp1Fu4Vz8J9bMhLpHfqsjjzxSvhRIenwaPiw8JDrr/PPPj9Zee23Jf3XmmWdG662
3nmhEL730kgiWgQMHSj1mC2+99VZytqIorQnMPdWYfLJIC8X1/av+okHr1G4r5P4EGJRZDU6kE34M69
OopDCwY9ZC3UMNRbBgliJnDH4M/4tBVAMzADQfPthf/OIXYg5DCGH2ImLif//3f+UY57MiHhX30EMPF
cGSFpKnKMriCdFOmK6LILTq/MUXX0zebQhjoFuP8cyFQKA25YM1EjUX77zzTmwG8dgIkuRI5cyYMSM2
Qqh+RbkZ+KWUgq13jRaSvGrI66+/LqvajUBKjsTxzJkz4y+++CJ5pShKc8AQ45ciVmmzatxMJnO15a9
E90lbdb7eeuvFw4cPry+DBg0K1nOLmfzGHTp0qKhflT4LSt62iqSsleg4rY866qig87tcWLOx/fbb15
uqWN2ZtcITkxXrPUIQn00ctOt4x8zFGhRFUVonRfka0iKYyOztOtHJoJEFGkhzRUQ1B7UzIiqKomRgF
7wRRWmLXe1d6hgTRLJfYFKqljSfBdGcrgBhPRum9lL9YpKLmb2SfvnPwrYN/jH72h4r4jlUQuZ+IErb
AV8Ruc2IeCsVnq0oeQkNzltuuaX4UoE0SEcccUQj5zTnlTrGX76vrEH7wx/+IMfwSfz3f/+3/O/iXi9
E2nkhsJpg8XDx+4oWQmqlcvsVehb2+fnH3NeAVcdeD9Ku6Z9XDYXtB6K0Dj755JO4d+/eYlv++9//nh
wtDvPDis2sLt5vv/3Eb6W0fsw406hsvfXWybvFwp5Boev5/gGftPNCZdttt03Oyk+aP6JWzwHS7qlI8
IE0iQbCGo0333wzU81ac801xcfyzjvvJEdKs+6664oqmxdmLIMHD47OOussido69dRTk3fyY2fooa1y
89wns4cf/ehHjXw+zFpYC4NviHbcVa9uVEfe42gQLOIMJXvLggWgRLSRuqFITYR7JEszedPIIMDOkkr
rxs6gXZjB29ky71cTjsvwxfcKmMGzHMDHn+kTdUVdC9k08O/mwW+L35qZbCWvFvWH1E7du3eXY2nagP
scoJpn4T4HSHsWRQ73TaaBvPfeew2kYFphj5Azzzwz+F6onHzyyckV8kPEhfmgYvOhJkfyQyTYYYcdJ
tf+7W9/mxxdxMcffxwb1bVRP/1y2223JWcswnz4sgfK5ptvHt97773J0UUYQdBorxMi4jju88orr8Rb
bLFFvRYxZ84cea5GONWXn/zkJ/Hee+8d77TTTvXH2AmSnR3NF1t2f+zZs2f9e2awj/v37y/nufXPOee
csqLd5s6dK7u2FRHNp7R8Qt//pi6+BhKKumL3VDfqKi3iKk9bFHffkDQNpDlKkZQVhVUNLPYz1ytZyJ
1lBidZVX7hhRdGhxxySLAexc4A3H2Gkb5//OMfRYqXKmbgkzaYSYTed8tDDz2UtF4HkWD33HNPNHbsW
FmwyJ7sU6dOTd6NZLbOPicXXHBBoz5TZs6cKe2GosN4H80D7Yt7RxOzheRszEy4nts/IuM4zvu2Ln0w
wkPyitGebZuYeVbX2sI+LdwHKRvsMVbVch6OQjQ7kmHa9/7xj3/I8yAHmT1GIf0+Mfl8fm7f0gp9pT4
ztND7frnqqqvkHhSlKPwoKV7zmySS0zrM86QtAb8tfo9kyciKKm0tNIkJi0uwVwcL/A488MDk6CIYhP
/rv/5LUgUYLSS65pprovvuu08EAqF0PldffXVkNIBo3LhxkoUXuAYLDhkY06DO008/Lfmzzjjjj KhXr
17JO41h8CKlCpEeaZA52A0l/NOf/iTmH/r3P//zP8nRRWASwnzD39122y052pBPPvlETHlpGO1ChB8R
G/QxDUxQRGikwUKpfv36iXqNKTCLxx9/PNp9991l4RRZmH1QmV2zgAv9RGAghMhkYBkxYoRErIgqHID
zUP0RysriSanvaFOByRjTssU3KWH+RXgwubGTU34fpC3x8dsK1WM5AZM7Wy+treagyOGe322TCRCEBx
lzGbQRDDa3Fe8NGTIkMmqjZO9l5ssgj3bBjPenP/2p1APqMuPFXsmHzup0tIJy+POf/yyzZdrea6+9k
qOl8X0b/CgQbHzpWKHPandmyo8++mh0zjnnRL/73e+kngt9Z3ZP330hh1AgZPDkk08WgYRdFTttCGbv
FO4/zV7Ke0SL4K9BUyJ5pQ8fvn2ODOSlfug8cz47PpuRI0cGU81kgTZDG9ttt1107bXXRr/85S+jDz7
4QCYKSuuFyU5zQ2r2rAglfktMYq0AYTJERJSP31aoHr5WNworra3mgHsviiaPwnr33XfFto4d3bJgwY
L4Zz/7Wfzzn/9cbPz4AqhnPuD4oosuSmotwkhzseVdeumlUrdcnn/++djMEMrygRjtSa7pFqNJiR0fm
74ZiOPu3bvHZnaenFEafBTjx4+PjcCMjWBtsII+C+pzfd8fUglGE4pXXHFF+VsKbMJc02hYyZHyMJph
bASu7Hdvnz9/zz77bPG5KEpT4vsklllmGfF/+P4NpTT4QJpUgICZXYuDl1BRnOukNGHpPwOqhQH1kks
uEUfUW2+9lRyNZdBE2PBhc66F/0lt4H4piirLLbechLe6IDxw4PMALQhCQucmTZok9xQqONxefvllqW

```
e0iOTMhpjZSzxgwAC5ttHS4mWXXbZB4ZkQSrj66qs3eo8fAufhBPf7nIbRlmKjukt4bYhHHnlE2mTwr
wTu2WhVsdEy5fUzzzwjnxWCEHiOZuYnjn5FaQp8AWK08Lhr164qQMqE8a9ZFhLiD8F3gQ8BZxPhbqNH
j673J9ClV199VcwdZN/FqQ7/93//F/Xt21dSChgBI8eAjJiontj9cSRjlsHchKnHhXazfCA4kDGLce7
XX38tfoStttqq3uSGn4XwOFJJ46vBnIMphteornkh5Tw7M/oQDHDZZZeJMx0TXblwrhm0xdSEfyEL7p
Okkw8//LD4KLDdWqzPBr/OvffeW3aKBkyV7NFy/fXXSwJM4Pnj/+EvAQ1cHzMlzvkZM2Y0uL6y+INDO
guCbNzfM4EdgwYNSl6F4XfC755QWHtu2nl++3z3QiHk/PbtmJGnDxa/fZ+8/SqStGuGxpxKaXITlg/m
C9OP2AiG5EhDzCAo7xPWiumKWTahvsz205g3b54kfkwDMwwmlLFjxyZHGvPxxx/HRngkrxrCzH6PPfa
ITznllAYaCJx33nnxAQcckLwKc/rpp8dG2KQu1EMDMR+8mJYI0SWUNm8hBBjNbq+99hKzUTkMHDhQnv
XgwYPlHjEp8hoNpVyM8JawX86fPHlycrQuVPmpp54SDQSNzOWCCy6Q+jxDpfXAZ5pV1llnnaR2HWmhs
X5Bs3XPTTvPbz8PeftAyWq/yH7lJe2aRcL412y5sJjt2lTIF198cX0eGBccTziBDzvsMJm1MjtlAaDV
Bnxw1HIO2sCECROSo4tg1mId4aHFfrzPYj5SwqMhoIGUwndiE3mFs33TTTcNRhbts88+ov0QCmwXGZU
Cx3YlpRIuv/xy0VzQYIgC47NhFkNQQDkQSEBIJCn7CR12nZX0zYYw+6HMV1xxRXTLLbdEv/nNb+TzQx
tR2ga+1uk7uENgZWAG756bdl4lWm2ePliy2k8bryrpV17K6X9VJMKkSRk3bpxIQ2z9RpCIJO7WrVtw1
vzQQw/Jwj/qhxbvubDQj5mzGdiDDm1m96SkRwMxg3hydBHMkO1iPmzzIS3EaiA4gIH+o43MmjVLXgP3
tcIKK8S33nqrpJk3g6L033yomZoBfTRqbXzkkUcmR8oD7aVcDWT27NnxDTfcIAEO9JPPg78bbLBBPHr
06FR/jQWNEEc8mhXn0Ra8//77MhOyPiA+EzMBkM+HZ2OPU/AdEZTA8zICuL4faIrlBBkoLQs+x5ZYOn
XqlPSwPG0jb3Hb53sfqhMq7nkhqu1rkTSLEx2TCDdy4okn1g9Mr776arz88suL2cMew8RjNAGpe/DBB
0vUFf/jWB41apTU8cF8dcIJJ4iJBEEQ4v777xcB8vDDDydHGkLEFc60kIkKuAa5nIgk69Onj/TptNNO
S95dxN133y3vUejPk08+mbxTGgSI0chktTbObQb1vIX6OATzCBAG5Ztuuinu0aNHfT+JeuP+ged/zDH
H1L/HivXQCnkw2pZ8Pttss40IVGA/Fntu3oIznfsHhBCfP+ZNFSCLL6HPudKy7rrrxnfccYeUCy+8ML
NOqXpuBgcmL6E6edvyizU/W/gt3nnnnbnaCmWWcEnra6j4/acUSZMJEC7ETHKNNdaQG0MDCcFgwcDLQ
Eg9BiQ7oFl++ctf1j+gfffdN3700UclsgusAAnNcCnPPvts/Otf/1reZ6buv89M4Q9/+EO84YYbBgUI
g9uDDz5YH/HF9S1EiyHYrFChoHEcfvjhMqvgmvhvmNETplyKm2++uWINBOFz0EEHJa/qoN9EPxFJha/
E9o/oN+7Tf8Y+PBe0LnsepW/fvpKShXDcckKK2RQM4Zi1gZjSOnC/M9UWN5HheynpkUgG6oJVIVRv++
23T2rUWRVCddzrQdo13cJ3Gz9kVkRXWr+yzkvra6j4z6JomkyAYFo6/vjjZebOA/DBbHHEEUfU3zg5l
myYZwic3EhwZtvMVO1AT04mHpr7ECstrO3wTVgMxI8//ni84447xk8//bQ4hN0ZPIUQWvpEqK4LYcr7
779/g7qdO3eOzzrrLAnt5Z7Jx4Nj0L6PGSykaaSVlVZaqf5cwo8RxDyPqVOnxtdcc40cRzthXcebb76
Z9Kw8XnrppfjKK68UkyPtYQq0AjwPPD/OQ6AprR/7fSyiMHGz8NsL1eE7f/TRR9eXfv36ZdYjJ1yojn
s98MN/QwVzO5NFVxAwYXT7REnrV6XnhYr/LChF0mQCJA9sczts2LDkVfNy0kkniakN00wW1157bXzLL
beUvX0uiwhJROj6Tlo7zLoQQAhgzJaKwuQxNPj5s/885G0rVI+FrlnXzCNAEDoM3K4gKMdvkee8PM8m
7VkUCQKk2aKwfNgQ5bjjjkteNS+33nqrRB8R6ZEFqeFJQeJHFWXB2gpSnpTKe9XaIAqLdBGkciFSTVF
YuxXCjE/Jf/lJa8uP8PTrsTUD1wtFgpYL33GKS1oUVhZp5+V5NqyRaQpajABRFKX1Qe4lv7hJN9MGbc
LsQ+fawsJh9rtx2yIvXQhy0Ln411y4cKGcmzUwk6MuC5YC+PXynFcOWc+G8tdkiUSt0S1tFUWpGf5sH
Ng3/MMPP5T/GeyqWbPgtkXSTzIq+Ky66qoNkrL611xyySUlg8WGG25Yn0wxRKh9hJO74rtdu3ai0WBR
sW3lOc+ypbNhVbXPhnUy/jXyZAbIC2vdVIAorQK+xqSh+Oabb2QwKDftilIbQgKE/WpYsAuVDpJ2ISG
phGxbefGvyYDfqVMnSf1TSoCEII0R/bDwvVtxxRUlW3eptvzzLEUKkG233Va0tFqBAFETltIqwCzBvg
tsSkbhB1o0ZCZgJnnKKafIKn2lMlxTU6V+B7SG9u3by99qQcjRFhOPcmFLB5+Q0PRhM7YQbh82q2A7a
pdKfS/loBpIBjwe9tUguSF/0+DDsruQkSwyC2Y9zEDKcaLTLqldSLLIPhoHHXRQ8k4+sPUyQw/toZL3
PpnxbbDBBrLrmguDAn06/vjj5TU/SMAmTLHY4/TF/erZ43ZAYSY2atQoMSuUwyeffBKdcMIJ8lkMGzY
sOVoMJP+84447ohtvvFH2lenZs2fyjpIGn6OPu6dGaK8MvqPuPuNpIDzQZuyMndk2CTp90C7cAbvamX
0WCBC+G6U0kLz3TVtoWzyzNPI+LyhyuJd0TaZBpQSscRgzZgxPPbOwsHD33XcPvhcqV199dXKV/LB4j
3T4rIUpBzMwy3oTQmhZI0KiShfSkbBGgxDEUF9tIRMAiyl9zJdYQpOJqWfdiQvx7KGFkSTIJPWMD+0c
csghsnCLzAT0m/MJcaSYH3/cq1cvSca51VZbNTjOayOYYyOYJSWKPc5977bbbvF2221XX5/zCdcuZyH
kX/7ylwaL0JTiyZPFgEWwrLnic7SkrdL2V3ebgT1Yr4jCQkK2Wth6662Tq+WH9XJ+eySQzQrbDZ2XVo
qE37yasDJAU2AHQ/O8Usvs2bNlD2WSPpIo8Ve/+pXsUBiqSyGlMltjurOgzz77TLZ7ZcZRqpx00kmyg
yGqc+h9t5Ai3WJna6Sypm9m4JZdH+22m7xPKDLmH2aHoX6z5TCJDkMaDFoGmgsp4DfffHPRyCjMnkjd
z6wP9dztnxG40cSJE2W7W1sfJ+T+++8v2hbaCNdFW2G2Q7JMCmGYzz33XPTPTkk09Gc+fOrT9OdAp9QDv
```

```
i+RjBIMfRHEjOyLa806dPr69PXcId2VGS5JZu39IK2wlw7dB7oVJk+uy2Ar8nnh2/PVv4brjHzMDaaG
dMvjuh8/L6w9xz/euVOubC95XvFN8vvvO2uNFigNbA79gtvqOda/Hdx7dnIaIrdF6evlKKRk1YOeALQ
XZZvtj+PgIMbOyVwZoOTCbrrLOOmDiIDGG/kC5duiQ160CVvvLKK+WDN9qERGsAg+VHH30kA1waDJBs
xcs6CtaesKYiDRx5DOShwd7CAOv+AC699NLohhtukIzHhx9+eHK0Dq59++23R4MHDxYhkZZNmGdV6os
6YsQIaafU3vXAj6bUD592KOxDjwDKgvsaOnSo/NgwwbnQ5zfeeCMyM9/kyCL4ESJc+Wu0D9mOFzCtsS
8/z5nzQ9AvPt88+7Ioi0jbryMEpk87wObZ5wNqbcIK4UaLQZq5LYTRsjLNdHkpcrhXE1ZOzGxCEpGtv
fbakosLs4qFFZ+kVSEVC2lazEArSQrJp+Wn68BUQs4tcmKde+65cdqeI2mY2YikESHJIOp6Hlj9TZ8w
A9EfCueSgRcwhZmBuD4lCznEQmDKw8RlM+7a0rVrV1kxSw4uTGOo76RgQZX3i/khS5oHzgu9TzHCR1K
ykJqFZ0j2ZB/uiZXsJNnkuqXAdMc9kyqGPUe4j0rgHskHhimNLMHs60J7yuKHb8LiO5dnJXoIcru5ba
UVzM4u7JMTqucWfiuYsFwzXdp5lfS9WtSElRNm8TiHWaHOLntGiCTvRGJCeffdd2UT/VVWWUVm3926d
ZMwPvY8cWEWhIZBYWaKGl4OqOjMZtEa0HTywMwZUw/ayk477SRl6623Fu0Ixzdaz4MPPigmtbfffltM
ZCG4r1133VXuadq0adETTzwhJqHx48fLzpHEm9Me5iXUd7Qbv3D/7PnBKvTQ+xRUdExOmPTQCo466qi
kB4vgnni+vXv3jsaOHSszyjTQDjC9cZ9GcFekxnOffOaHHHKIPD/WFdAWn/2pp56a1FIWF2yghoXvCF
pKJd8N38KQhht5BnyH80A9NzILq0KISvpeCIkwUXJCLqeNN95YnLgkTBw6dKjMLnAIA7Nj9sFgPxGct
jgELa+99po4mSk2BxYzYlLM81HUojAzMaqmXAvMoBcfdthhsreKOxtHy2JfdDQkq6n4hfe4/1I7PpKk
kfvm2ubL30C7wPGJxkLadrQM9z20E9vnffbZR/qZhRFW8S9+8QtxkIcyHHPfaGxkQyYQolz4LPl8zeS
gPoM0ec/YL4bPkizGaCSHHnqoZIJWGmM/0+Ys/h4bfHZ2F09K9+7dg+flLXyv3fZC5Wc/+1ly9Tqqce
T71yOYIFTPv++iQQNRAVIBpJ0nygf1EjPLGWec0WAAwTRFlBMbMt11111yDLMXgzamGTdpJGapOXPmy
OBMhA+p0zGX8D8p8N1C5lsGq7XWWiu+/PLLG73PIIeZibbYf4RNnkhSaU1u/KV9BmgiwBAglQow9v8I
QeQU29LeeOONyZHywLTH88TslofXX39doqlQ8xEoLnYjL0xNPOdyIeU/eyoQFQb0iVT7CBCbDBKhS2Q
XkV7lbOLVVvC/N81RsvbYqGTvGrdUYj6qRoD410vrf9Z9V4sKkCpgps2g9aMf/Uhm5T5k58U3woybsF
d8KAgddjLELp8GvgnsqiHy+kDwaxi1PHm1iG+//baRALEgAEktT/ulYMbdv3//BpqVC9oXe7KwJwIzd
77EeQszQUJwzzzzzNwCBKiLDwqtBsHNs9l1111FwNvBvxzYSoC+4xdCIFpCAgTs9ZkZ8r6yiNDA1tTF
9SGE4PcbOi9vyWo/RDn7evgFf54LY0aoXq3DzdUHUgWTJ08W2zvRWexzbp5n8k4dhMQOGDBAokIIS2W
/dGz2ZoYdXD1LuB7t7bHHHrInu426cMEvQPgfdfET+LBS2gygksLgoosuqouSKMFSTpQTEUP0kzDkgw
8+OCIs14UwRNo1A6f4T9Zee+3kndJgw81TqoFIKO77pptuklXihA2vvPLKEjVH1uNyYE927pNILXxVb
roJrsNz4vOjfQvHCdclMg6fSJ8+fcTnoiwe+N/1cpk6dWryXxiyFpCDyi12wW0l8Ft022IJQQgWF7r1
KIVTJ0uUvDCLxzTC5k/M1tmznVkzs3p/1owWgNmKx0xJ24oX0C6Y1f70pz+V2bi/GA+YtbCjIjZPzGg
+zAiYceMT8H0cYDUQNvbClg9PPPGEmMOszwHfDDsacg36S/TWcccdJ/03X8BGbfqggWDCYlOtSsCEVa
4Ggk2bc9AA8Keg6dFfZmqYAkvBc8f0heZgBIFoLY888oi8x7a+rg+I3S+JwrOmSXucnS75vPAjsdGRv
T4zQN4rpXG2dux33y2VmHzyQERk6HpZGkI55qRK+o5Juqi20u4xbykSNWGVCb4NI+0lvI4VzBa2qWW7
XjaisjCQI2T40DDpsOKa//GZsJuhDwMZ9nTMQzjGPvjgg+SdReDDsCYs67T3QSCwC6FvogIGMgQCQor
95zHRcC+hwZ7wWPulw2TDlr15QICw+6SZpctgXE5hh0lMX3kECM8HYWNXzuNbYpX+/Pnz5X0GdjcrAF
kCeDYh3n77bREM7Nhm/Sh8fvbcvIVdLAGTAuZLPkvCLlWANCz4i2rBK6+8ErxelgAhGCZ0XqhU0vc0A
VVJW2n3mLcUCQJEFxJmgHpLaCqmHTOoiumGhYNGS0hq1IH5iBz8JPSjHrBBFovyzMxYXmOCwsRCKDCg
xlKHRU2EEZqZtOS6YtW0GYBk1bfL559/Ht1///2RmeVGZpCNttpqq+SdOghHnDRpkoTlYoY644wzGoT
3kVaaxXQXXHCBmI2MNiOLB4GFjYT7ssjPrprF/IZ5hvuibTMgRoceeqj0t1T4IgvsWIFOH8vl/PPPl+
eFCc6CuY4+GKEgecD4PIBQXkxG3Ke/ItiF58FCTzcz6YEHHhj169dPwpv9z7IUY8eOja644gpZja5kE
zJPmgG93kSLSdZfpZ0Hhi3CwjEl2jxRaQsE3euBf03+P/LII5NXpSHHmv3+5SVvv/IwceLEaMcdd0xe
lU+Rwz0mchUgGSBAGNCvu+46SXzIwGZmy8m7dau5+YIwGE+YMEGOsXaBev6KZwuCgtXqrJxlACelB3Z
1vkysACehYbWwLoOUKgzkFgZikgDST9Z7YCNFYM2aNSupEUkqE5IRmtm7rHcAviicR59feOEFOQasd+
GHt9FGG8n16Lfrd0F4hfw9afDDtuCfwY+ET+jEE0+UxIoIN65Hahn8FLZ/eWHAwXfF6nWjNYrQJhsAg
tL1B6XB+hY3mSKpYZTSZAmQaldW59lbxB+o814T/5e/nwbfOXdvkTwUKUDS9jzJS5F+EPmtI0CU6sDX
Yb4kEkLb3ODDwBxz9913ByOxXKhrNKTYDKjJkXxgGiOkmHOx+7cVMI9h2sIXlNek19Yx40yjYgbO5N1
8K7JDBZMnkXLuCm8zGAfruteDvNfEFFwEaRFX+MgWZ9SEpShKTcnSQNJm51mQlQFTJ9pvVlv+TD/vNc
ks4WsbaCWXXHJJ8qouwsrXUnzQGsiG4JOnfZ+062WdVwtUA1GUFgAr3lmHQxAFDv3WhBlnGhVXI0jTG
vIUAjXytOVrINVckzxoLmkRVpUWv32ftOtlnVcL0EB0HYiieDCzssWuESB/kj3273//W44BPrDQ8XJg
```

ls4OigQD4PNx/UxFgX+JTM5777137s2HWjLkhyKfXFPj70BYdKbl0A6HLmnXyzqvVqgJqxXD4EfkF4k
PQ7skMnBRWByHKQBHMlFcwIDpfzWoi1Oc5JIkZCQpYl7Ye4PAAtKq48AmMWE5EAX2/vvvi+Pbj07jPj
ETEJSAQ5WFlm7f6TcmD/YJwWFO9JYLgz8mBvpEgIRNWEc7tMezI3273XESZzwBA1yTRZ84+4E+sr8Kw
RYM1jzTcsBJz2JErkviTnexYrXQ57vvvlsWtPJ9KCfyrBp49j5FmLCIusOhTTRgVlu+CauaSCY+U7ub
IpQTwZUHv32ftOtlnWfhWRQFvxsVIK0YBg0EAuGqoZXrLoT9IkjuvPPO5EhpCGUlHDgvfM0QHhRCcRn
I84KAwPZLVBkhuQyCZMdlNoZgYMBlUyi2+EXIlIIQbO7VBc0B4cM9MYCzMZiFcN3zzjtPVrpjZ7bYqD
T+unuuEKVFHwnfXnfddSWCj/5awUx/mT0THcdnwmcEVjgTDs4PkzZ5TX2EGq+pz3vAZ0U02mWXXdbg+
qV44IEH5F4effTRRkK4VtRKgDCRQMDymZQrQCqNZOK7leXvSIN+uudW01Y1FDncqwBRZMAibHefffaR
QioPBlT+hhgzZowM3ggBQn7hyy+/lDQirF0pCtakhIQMg99ZZ50lAyCbW/GXrzDpJAhJZn1HKLUD60g
IkWbTL0KUXRjEWVtDuDCaFwM/MHgzOGMeQLMg/JP32UyLEGY0ElL5o3kA7ZA+hfDt4cOHS8p9BB9pTu
zOj/zoSIdP6DTPD0FgYbC163Zok/vieqTZZ3Ds2bNnvVZDPdLo9+rVS3aXZFZdNAjDalJuQK0ECEKYz
wnhWq4AqRTMjO4kohwIO3fXIVXTVjUULUBoUGnFEHLLqnVWafuwsp50JWymY77ckk32oosukhXcofTo
rNJmlfg222wj6UMsrKInFQrpPNIKKVTIskuaDzbcCtWxxajp4qDLixm4JUmlGexjo2EkRxeBg5r7J1V
JKGSZcGdCLbknnlcahC0bwZGaSBII9aUdVtLTrxBm9it7ZoeyDYQgUwBJO0ng6WMEjCTfDD1HyvTp08
U5b2a7jBxSSMVC5mZWYJM+J3SevYdqsdd0ixnQk3frEleG6uQpbPDGc7EYIRGs516vGkgFFGo/T3H7C
Wl9rXUpEg3jbQOgTTC7xvzDzHjIkCFiowf8I8zimR3zl1kdi+wwTVGPTZMszIQxE7FSnJnw5Zdf3mCV
ex5IOsmKcFb15108xqpfbPYkJ2Q2i0bACnz8FPgn0IgwFzGjZ8bsmp8s9J3Fm2gutGXBF4K5yW4vTFg
o5jDq+3DMCBkpaAHuYlIL/cO0YjUJNBoSVLrwc6PfZCugzvXXX99gsacL18Shzor+vfbaS86h/XJhK1
5Mh/hr8IOgsfBc2Z7X1YBqQZYGgk3/iCOOkP/LAbMmnxW+HGv7z6uBVLt4MQtMj/ho7Ap54H/XR5HWV
76D7nk+BENUEwRR5HCvGkgbgpTV5LQiDxaaBzADNQOc5NdCi2DGjLaCBuLm+gLzZZF9TMg5ZbeZNT9i
+ZsH6l533XWyvwZ7leQFzYU07eb7Wl/Qgpjps7EVKejZXtbdj6UU1CNUlpk5ienKgeeEllNqQ6088Cz
Z4IqFcDZxYwjukXtlSwBS1FcCGpcZqGTPF3K2MRPm+ZOU84ADDqj/LtQK93OzpSiNwCevBsKzDNUrop
DMk3xuWYkS0/qadV61e5cUiWogbQxm8cz2sL8yW2dWzhaz+DOYpQEORtK2kPOJaClm23xF2MqWMFNmS
DjaiYDB5o9vwfpCioYZGtqT3dKTGTlpYugTGpXVgEhPgrOaSC/CakNfaWaFaA30G3t0KB090Vbk1sIx
zvNwNSxm/mhua6yxhmy1i+/IBQc3z5Ntf3luWeGdzN7QnEaMGCHanh8VxL2QU4zPieeNBlIO+FDsZ4t
/Cl8J1yJ1DlomviOcuDw7Ir7w69SCkAbC7L8WPhu0Kqtdu/gaSKV+lzzYFDw8T34baaT1YYcddih5Hl
YDGw1YCUUO93yHVYC0Mdjj4uijj5Z9PfixkRwRc5Q1ozAAE5F02mmnST3MWZgKGMwomKGsyYGvDiYyB
BMDFj8cVHe+5AgXFwbE++67T6KgyBvG3uIu1uxD8kbaZICmPRzYDP70gT5jxuGHhwDByU3ySqLMcOTn
hYHTJrx0YQDCpMT1+VsOmBbY64Xnd/XVV+cKwcUcRoTWs88+K8+F0GjgXglIIJDhqquuknstB6O5yPm
EJvN5sT8Jz5NEnAgQzH7k8cLsx6SAwZXr+8k5iyAkQEIrsIsgbcV3UwoQvpNMNjANuveIw9xdKc5vLB
QkkvVs0u4xL0ULEBpU2hjkwCIPD+aokDkJpzJO5y5dusSDBw+OzWAamwExHjhwYFIjDA5RHMRGU0iON
ARnMCYs9swIgXmJlOtpzmLMcJjgzA+xkbP7iiuuiI3QS16FOeSQQ2RLYIIBQuBIx9HPfZMHib3d8xae
J85pUvd//vnnSYv5wMHN82WbYJ6fEa6ybfGECROSGvnBJIaZiv6494k5DNMjfSSlvwup543gF1OgGaC
So8VgxplmL74JyzcfmYmLBJJkmY9C5N2fI89K8bxtVdLPWoAJS1eit0GYwbC+gdnMlVdeKX9d0CJYrE
SqeWaohMfiDMaBHoIZM469c845R3YAdGd7FruGwXzvZIbsw6wfBzfZQjHt8DoNq624EL6LQ5bQVj+bM
WngcbrPnDlTtINSJgD6Zx20mMzyFupzXiWgDWHGYk0Hs0/uD8f+LrvsktTIhmeLNoM2hnaHici9T8yA
zG7R5nwzD1oJz5w+sJvj448/nrzTOpg7d27yXx2+doz2jPZJCHa5hBbohsgT/JC3rUr6WSvUhNWGwMz
DAIrNG8HBYIuQAOz2ZuYq/wORQkRmETUFDDKs/whF7TBwsh6CRWpmdltvInFBgHBd7PIMVKTUcOFHzE
DPYEo6D9fHAdaEhRkAHwYRTvhpuB4mMQZH+sH5XJ81HwiNkSNHRuPGjRPfBia4kEnFwiBCtBJmCMx65
YIAZTV5XhMW98xgTXQO98LAztoGTCz4UEi5z+JIf+W8hZ8ugwl+Dp4Lz4AIN/wpRJchSO3PG9Ma608e
e+wx8SPZ/Vx4HpgNETоIFyK0MK/wXeBZ9O3bt6LIL0s5C0ZrBf4Iouws+PN4thYmTHxeZCpw6+XBbys
Nvw8himyrKVATVhuCtQJE3Sy77LLx0KFDk6N1UU5EZ5mBQ16b2bSs0+jQoUNsBhcxq1xzzTXymnUUmL
R8zOAu6c2NFiDrJNKim9hGF1MBpq4QmKh69eoVNFFxjRkzZsR9+vSJN910UzHTsAukERBJjUWYmXi9u
o85Km8EE+tgMOPwjDAhYXbIWzp37iz9MYItM8U9pikj4GIjjKWPPH/SxNt1IzxLoq/sPbDF8bXXXism
Nh92P8TEyD3b6xrto1HkWlah38Bah+HDh8trnjfPXVFCaBRWK4dZO45aoniYfaMVsCqa9SAuaCbjx4+
X1d/sWAiYrTCpuDN2tBAc0HxldtttN9ldEW2BGRyzXWbgbFbF6mU/yonZCs5iIkzIOcWaBBfMQGgxzH
7RHnwNBA2GVeRcA9MMazrOPvtsMb9h+sJkg6ZBigjWitAGs3hm+ERHsV6AFeiYaLg2GxGhabhgNkMzQ

zPgeZULGhrOdDQt+9zQEB5++GGZ+bMeg10lgb5grqOfpbBrVEhDgoYEONuJ9uIz6N27t2gOeeAZobmh
yeU9R1HSUA2klcPs0Qyo4vxmXYed4Vp4zR7LRgjUz0TZZzw007XgmGU9hxnA4rFjx8qMlesY9Tvu27d
vfTvVFNZ2mC9ncsU6mO2YATg2AkRWSbPinCAA9zz2JMcJT10XZuRoWG7dNddcUzQd2txxxx1FgzCCSb
Qu3keb4Fje4vaFc1m7QX+mTJkiGhyaGc5q1tn4/csL60/uuOMOWaeDk5z78j/TNNBO0ADRruiDolSLa
iCKaBP4Owi9ZX/w5gYfCTZ/ZuiltplFW0C7YF0HvpxyYCUyocc4rNGe2gJoVWgz5BILaYiKUi5oICpA
FEVRlLJBgGgYr6IoilIRKkAURVGUilABoiiKolSEChClTYIzmVXzhNyGXluyjtvit5NW8l7PL2nnpfX
DJ+16Pmnt++dltZfVTtHX8Uvedv2Sdf3mOi+tnl/Szktrp2pwotcCmm4Jxd8UJ1SHUsnmOaF2KG5bof
eboygNYWEez+Wwww4LvrZkHbfFbyet5L2eX9LOS+uHT9r1fNLa98/Lai+rnaKv45e87fol6/rNdV5aP
b+knZfWTjUQxtvqNRB/wZRNDW7p0aOHLCirZGGV3xaL01hA5rbl11FaBnZDKDL+gv/aknXc4reTRt7r
+aSdl9YPn7Tr+aS175+X1V5WO0Vfxydvuz5Z12+u89Lq+aSdl9ZOtbQ5ExY5gVxIYEYOJSNQkyORrNZ
mHwa/+EnYfFjR7bajKIrSmmlzAsQXAgz6Pscee2x0wgknNCqlMsRCKMusoihKa6XJBQgz9FoU8h/lIX
Quxc0ym2Z28lOI+1gTVh5CfSiq5H0WiqIo1dDkAsTfCrQoKt2LIURaH7PSXmSZuFxq9RygyGehKIqSR
rOasJitV1PYe6EWkGWWfEFuYU8E//q+ECAzLBpAJfhtl1Nq9RwURVFK0awCpNIIJfZurjRyKg/sI+07
0O2e4aVYY401ZECvhGqitTQ1t6IozcFi6URn74iWiJqOFEVpSyyWAoQNcWo5WG+//faS2twteXwWbPh
TqQlLURRlcWOxFCA9e/ZstJtckbBfBHsnuMWNcrLFj3bq1KlTxSYsRVGUxY3FUoCwVWstcUN6S4EQcf
FfK4qitGYWSwHCDnpFDdahVef+anWL71z3682bN0+FiKIobYbFUoDkiYjKS2jVOYLhhRdeiKZOnVpfO
nToEJ122mmN6rloNJSiKG2JxVKAdOzYMfmvetLCZ7fZZptos802qy95hENb2V9bURQFmlWAENm04oor
RiuvvHJ9gaxjOK8xFdUyEmvzzTdvFIVFJstS/VpllVXkWCX98p+FxT8WejYaPqwoSnPQrAKEFd/9+/e
P+vXrV19Y9Z11bM0115RjtZzxEyrsR2HtueeeJfuFoKm0X/6zoB2/ff+1Paaaj6IozUGTCxDXZIRTeu
TIkQ0KzumsY2PGjJFjbnLDpgifHTVqVIN++P0aPXp0o36l4ZvO/GdBO377/mt7zL+ehhIritIUNLkAC
aUJKaKkRU75+FFX//rXv5J3svEXE/pt3XbbbfI3T1LFWj0HSt5noSiKUg1NLkD8SKaiSt5B04+6sov/
Jk6cGL300ktSiLjKg9/WgAED5G/WviFQq+dAUQGiKEpT0Kw+kObANx3xGgGyvZO+JO8qd78tK4xquUp
eURSlpdDqR7qsCCVWteMgd81TvqnK4rfl15s1a1aj6LC0thRFURZ3aiZAbBRRc5esCKWZM2c2Wj0e2g
8k1JZf76ijjmpUL62tpi6KoiiFYwbPNsVyyy2HtKgv3bt3j9u1axfPnz8/qaG0BYYMGSKf/4ABA4KvL
VnHbfHbSSt5r+eXtPPS+uGTdj2ftPb987Lay2qn6Ov4JW+7fsm6fnOdl1bPL2nnpbVTDd98803c5oz1
voMZhzcmJ/M8kiOKoihKHHmomQKxDOqsYKZacUYfrzE4rm2yyifx1z007z2/fD7F9//33k/8WkacPtvj
t++TtV5GErqkoilI0NRMg7iruUsXPMRXai8Mv7EjIX/fctPP89tE0QsVN4Z6nD7Zk5cjK268iCV1TUR
SlaJrdhOWvovZDY0Mw2BMq656bdl6eVeE+efpgyWo/LXNwJf3KSzn9VxRFqZRmFyCsm3BLWtgrAsOWK
VOmiN/CP8+tQwG3ji1fffWVvAeh99PaChE6320fHwvH/Lb8cyjueSFC54RKOf1XFEWplMVmZJkwYYKY
Zihps/qnn346mjRpUn1JW1G+1FJLJf+lz9bd65W6pkvnzp1lAHfNU9OmTYsmT56cq19ZZq1yNAv/WSi
KohTNYiNAevfuHW299dZS0hYH7rjjjtGWW25ZX9IG5DypStzrUfL4LEL9wuHv9qlUv7JAOOXFfxaKoi
hFs9gIkPnz5yf/NQ7FtYT28Ajh1kur414vL59++qk45F1CEVFp13Sp9DyL/ywURVGKpmYCxI9yChU2h
goRqutGSYXep63QHh5Z9TbaaCPxEfj13OvlxebCcsH85faJQvtZpJ2Xp4SehaIoStE0qwaSZoqqhLQU
6r4m4V+TXFiQNbu39UrB4O1T6f7t1URShfqhKIpSNDUTIKF9KihF7JUxZMiQ6Pbbb2/QVloK9azB9LP
PPsslyPKkaJ83b16j6+U5rxzy7COi6dwVRWkKaiZAQvtUUPIMqFl7ZZCg8MQTT2zQFmYau5+HW7Jm8k
Rk5XFOh9r3nfEh53ie88ohzz4iCBB3fxOKoihK0TS5CauINQk9e/ZstL7BdxrbknW9lVZaSQSIG9obI
tS+LzCWXXbZRsIodF5an7L6UA6+E15RFKVomlyAuOaicqKKXPKYwfLCgI/ZKSvtewjfXJXXp5N2324f
1I+hKEpLp2YCJLQnBSVrr4z27dsn76bzxhtvNBqsEQShkrW6m7YYrEPnZhVfEMyZMyfXwJ92327bdq/
2o48+ulFdt6Q9L7ctiqIoStHUTICEnLsUNwdUyCGcJ2qJxXm+GSjN11Hpor1KsGG8WX6X0H37fhEbRn
zXXXc1quuWavwpiqIo1dDkJqwsFixYILNq/AG2gHuM/9FA3Nk+juPQeXlNQf55WX2wx1zQdrieLyB8k
1tIKHDfLva1q+Wknef2yfYrq6+KoijV0uIECIMt6dr5awu4x1588UU55sLx0Hl58c/L6oM95kIYL/iR
Un7k2bHHHtvgfYofehsKxU07z+2T7Vc1z0JRFCUPLU6AMIPPW7JWp1OyVpT36NFDzGGhc7NK2kp6n7z
mNhfbtqs9pJ0X6ptfFEVRiqbFCZBa4zu+mZ1XOsDmjbrKW8/FnlNJdJiiKEpTUDMB4kcBNVfxo7D8CK
g333xTBEjo3KyCMMqKkqJUIgSsoMu6Hr4Ot44tWdFniqIo1dLqNRA/Cst3cFeSNNElK0qKUuTug/710
qLW1HGuKEqtaXMmLJ9q/QOVpH1HQ6g1Gt6rKEqtafMCpNZUu6+Hj58aJa0t3Q9EUZRa0+QCxEYFFV3y
RkRVQ+i6WSawavb18Etonw+Oh9D9QBRFqTVNLkCqmX2XopJIJyjHnFSJuSot9LaS56DrORRFaUk0qwk
rjwM6VEL7geTFv2ZowV4aabP9pqLovUUURVGqoVkFSGhldZ5y6qmnNtoPJC/+NdMEyAsvvBBNnTq1Qc

mzADAvWWnmQ0ybNq1Rnyjuvh9pRakMm9TS1z7tcYt93z/u49fLateSdl5aP3zSrueT1r5/XlZ7ea9nq
fY6PrZ+Vrs+WddvrvPS6vn49WpNswqQSgfkrbbaSgbgSgbhvNfcZpttos0226xBqeR6aSaySqKkSCLp
94niOsvTilIZP/zhD6Ntt9022nTTTZMjddjjttj3/eN+8etltWtL2nlp/fBJu55PWvv+eVnt5b2epdr
r+MXWz2rXL1nXb67z0ur5xa9Xc+IaQdOhYiRkUiOOzWAerJNVllxySfmbpy23DuS9phmY4y222KJB+f
rrr5NW6thuu+0a1fGLESAVt++Tdr2s85TGDBkyRD6HAQMGBF8rxaLPu2lpiuf9zTffxM2qgVRKpZpLO
fhRTBTfUR+KsPKLec5J7Ybkad8n7XqVBhAoiqJUw2IpQH784x9XZE6ChQsXJv+Vjx+yW+0qdp+s9tLe
L7ofiqIoeVgsBcjbb79dcURUNZFMfhqUciK48hDaaMotadfLOo+iKIpSOGLMqgE0HSpF+EDWXXdd8S1
U4gN55ZVX4pdeeqns0qFDh2D7ecrEiRMLa6vSojRbfJNiz7vpqUpnvdi6wNZbbXVUqObsvBTfOQt1W
yN66czqdT81pb56KOPxN8zY8aMJg9VLIcPPvhAwqYJr/YzIrPJ18cffxxNmTJF/FlsjMb9fPnll0mNM
J988kk0adIkuf9PP/203ueFRvr+++9LW375vr+++9LW375+9//nqpt0wd8cG49v29+8a/d0rDP/eWXX5bvShru51Pq
ub/33nvyLF577bWSi36LqEdIILlYV/5lPnjxZziva0lEoiTApHJoOFVcjCL2fp6y55prylv/xAk5byayC
VUqm2RHH7CdW0VVlZ3Pjuu+/iZ599Nu7Xr1+D0VVlZ3nmmVt76/Nu7Xr1+D9hoo43ip59+Oqlvl8OOH34YX3
LJJfHSSy/doK9EC/7617+OjQCEqerNmzYPPoeanwNzYpPOeWUnLnHNObfnHNOe/ZXXrDd5Z6yy+/fHzTT
TfFCxYskPdmzpwzpwZH3vssY3ao6ywwgqI1LvzzZkrW9uueqqq2Iz2CUtlkeRz9vCc3j8
8cclOtHtJ2W33XaLJ0yYEC9cuCsFDqPfjgg3G3bt0a1TvvvPPizz77Tnqj7p///Od4ww2wOd4ww03bFCHz2fgwIE
1qzd9+vR47733blDHlt69e8fvvvuu1CuHWjxvhtzSQJfjHNFw4aRoCs0czgMr/LAi0e3+Xg+l4tOyyy4
rt36ZKx5EcmgG414O0fhUBqdWZzbFnB/t0WNx+Qlpf/fN8fv/738tMsRJq9DHXFD0oyWyxY8eO00v+LL
75YZsBXXnmlrNNpKdCn119/PfrnP8n/8Z1/PfrnP/8Z9enTR47Nnj07GpGGjhhhRhjGJ51OUuq+MPvuuu6/MOI1w
iA4++ODkaBQ98sgjgKRng5PvbpUsX0RZ4ffjhh0hou6fjQGIxREmxgzZkxkhENyZkPeeOMN6cO4cePkGRp
hlbyTDf0688wzo+7eXXXXXaZrD9oKXDfZ599tmgVjz76aNS63c5zv/tmgVjz76aNS1a1c5zv/coxmUo3PPTPTd64oknIiPOo3
XXXTcaOnRotP7668tnRp0777tnRp0777xTPqczzjgj+stf/f/hJdcMEF0XrmMvSrZK8xkRdrjmQ0bNiw5VmnMH
z++sHrnn3++fM+NcBZTo0o0VJncUAWMyJAaoFpP1iQWrUgbvVbAyvRDH7JgBho6L09Z
3GEm3qtXL5L5ZlBvfjjii8nRlosRIPGhhx4x4ad+rUKR4+fHHhZs//Wz9cM8KKtofFMnTpV3nCtsnBK6vlaFVr
BT37yk3ivvfnKn3nmmdgMgRHj1ijKG+Pa/0b9+bgpctHNl0+1CM8KKtofFMnTpV3gc0uT/96U/xpptuGu
+3337xY489+JrNRj148GDPpZvxcvrr4C3nmnPvXv/41OdpyQLO//fbbRSs1E4r0ksvjksyo
muied1xxx3xOuusE5sBzOuzmDiNY5N6MAJfnYoS8rM/if5g2bZo8eM//sb2nnnnmeM/MCYf
8YEHHihjjB/3ZZ599NRt755135LyWP4QGq1T0UtNYu8mGea/FeaFm3TbIGgdVWqeTUH//jHPyRNzsS
JE6NBgwaJxmBmJNe9vs+6/RSa+6+/z0MN2fOFB+SGSSA+YQP/pQqV7q+VGkSSDaB+QJE0TT6NmzZZzR27NjIBZ+mE
lK/WZhaLXs6b/lllvKa+z/zGTx8d8+KaBsvhn0a2hVtPxA1dOdutqhxjvytvzb0vAXl9X2a2dWmutFN
w03NEqNgbYzcOBAscffl0Lg+zUP//5z6WPgCBVRtP2fcVUlfpLM3QmZL7aVy/zWr2M8zGcoqnyMbjfLnurgAhREIZWWmyEdG/R9KaGVZ
xUxGxCeBNsL9Y63AujFixIoW7dusqob8dcIg8gM8PId/OKLLwqtBBgBn4Kqm7bPsLJCYXQ5L3SN7
3318+ixZLnSwpHpoOldAK7CIK0jt0vabUQNq3by9//XtkFuTin2dL1rNJu8c8ZZXFn0qRJ8S677BsvJsc
ceseflBJUdbHmPGjBFNoW/fvvGrHC2NERwyQ2bmPGrUqlhPnz7ymW2FdHz7ymW2yySaicW299dhxWmutFZvBU
Oz4ZuCX2XUazL7xE15//fWi9mlVPNPFLm9B60EDOv744+++744xxVxv4X4X4X4yAeol1mlVPNPWWER+B78NrKfAM
jjjiiNgMvPHDDz8sx+gr/qi11147PvHEE0VLszbzf/6uyzz5bjRhjJc+fZXXz
xxfX3jSaGhsNnOm7cuELr4f8IMWfOnLh///////7xeuutFFTTyJS1YZ7Fz7PJBUhTlywBsvHGG8ft2ft2rVL3i
0PHJBuW2mFH2QWedtqS/DcUO9xpt94442xmVnHZrYtJom//e1vjvNjyaWp6DtYjWVxSu/ngh4SZwszS5XPERISQo
++Y2/iL85l6RhuIn3vuufj5558X58558X09Sdd94pDt9VV11VHKscQ0AY7UOm127dhXhXBmTdtdcTs0+RWxmO+33dhXhxGFOhhIKK8VG84hXW221
2Gg0Mgjy3eY6tEWo+EMPPSSDoIG05M19fGsuMbqq68uAxnxCsCDexDHhonw4LoIGvs7oY+33HJL3KVL
YY46pd/S2NPjcH330URHWhPUjJDEHYWrjOdJ/M3uPjQYlJj++KwzWfBZ8Vscdd5wEPRjtpH4Q57lsv/
328lwYtJm0/O53v4vXXGKIxxr2dZ0gc+R/p/7733St+ZkGLOevPNN6V/LY6V6XW/9+CkGL0yu/LrkPPR3gdt/n
x9mqF4e/LbSSp62Q+eFSiX9XFxhgCi1Zgab9ueff57UbjWmTJi2mOojxQb6cQA50fkUBhg7rnnnnvjL
L79MWmwMWgJ177rrruRI+Pl07NgxxgBE4/gFrGBsAdDL9dlIAD+4lNA08WfhzBrydBfhAF
amntfyyyzjPhCrPaBL4FoJrcOmhyTEldAEq3FoI/vya2L5ku01/fff194PXxUI0aMaPA+Zf311xcf1a
effipttUR4/jWLwtp5552T/5oX7Mqu32XPPFPfdsYO8luoUoCGzN5fpn/LbS8PsQosi2FEVRag3jVc0Ei
KIoitJ6QYDokmhFURSlIlSAKIqiKBWhAkWoAFEUURVEqQgWWIoiiKUhEswU7+VRRF

```
UZR8IDra/ec/le8RriiKorRNUD7aLVjwb8lGqSiKoih5QGaIAPnuu+8k5TBphRVFURSlFMgKZMbXX8+P/h/2RoQk7TZ4uwAAAABJRU5ErkJggg==",
}

DrawLableImage(DrawLableImageParam, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode } = JSON.parse(JSON.stringify(data)).resultAck;
    //图像绘制失败，退出绘制
    if (errorCode !== 0) {
        return
    }

    //进行下一步操作
});
```

4.8 标签预览 generateImagePreviewImage

代码块

```
/**
 * 生成图像预览图像。
 *
 * @param {number} displayScale - 图像显示比例，表示 1mm 的点数，可调整预览图大小。
 *                                例如，200dpi 的打印机可设置为 8，300dpi 的打印机可设置为 11.81。
 * @param {Function} callbackFunction - 图像生成后要执行的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 增加方法说明:
 * 1. 在调用此函数之前，必须确保图像数据已准备好，否则无法生成预览。
 */
function generateImagePreviewImage(displayScale, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "generateImagePreviewImage",
        "resultAck": {
                "errorCode": 0,
                "info": "{\n\t\"ImageData\" :
\"iVBORw0KGgoAAAANSUhEUgAAAZAAAADwCAIAAAChXqV1AAAgAElEQVR4AezBeaznd33f++fr8/2db
```

WbOeMYzY+MZb+DxgAk2lN5QGiVRC71SkyqtKhVVAZEAoUsQ6U1RK4SaRKJqdRM1EamKyI3C4pKNViit
qlQJ/aMNNA2JaEoANxhsg5fx2B4vs585y+/7eV7zTY80R/POVUa3FhrpPB5R2bVr167rQVR27dq163o
QlV27du26HkRl165du64HUdm1a9eu60FUdu3atet6EJVdu3btuh5EZdeuXbuuB1HZtWvXrutBVHbt2r
XrehCVXbt27boeRGXXrl27rgdR2bVr167rQVR27dq163oQlV27du26HkRl165du64HUbkWSaioVJJQU
akkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQq
SaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRW
VShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZ
KEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQU
akkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQq
UbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRW
VShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSU
JFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQU
akkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUk
VFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRW
VShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSU
JFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqUbkWSaioV
JJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUk
VFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEiko
lCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSU
JFpZKEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSahcv
rwxn2/O5/OtrS1Aba0BR44cofLCC88lCzCXEVsyQDftxgMHqZw9e5ZJEjUT9YYbbqBy9uzZJCqQBEgC
7N+/n8rZs2eBJCqQhMkNN9xA5dy5c733JGomTg4ePEjl7JnTI7MkdIHIONNx8ciNq1See+45rqAC6k0
33URFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqSa
ioVJJQUakkoaKy0wgb5y/svWE/lbNnz6pMMmFyww03ULl8+TLQJkDvfRgGYDabUVHZpgJJ1NYaFRVQg
SRM1NYaFZVKEioqoAJJABVorVGZz+dJeu/qMAxA711dXFykcvHiRcBtgAocOHCAygsvvKDyoiwwdtPN
XL3p8M1UVCpJqKhUklBRqSSholJJQkWlkoSKSiUJFXXtt3/n3GvvWZ4Nw8rOYdrMqzz7zqz/bDb33/5t/8m
1Tuv//+PmGnd7/73VTuv//+JOrevXs3NjbOnj65MiR7/7cyn8MMkVObzOd/+rHm2qaeTYhW1wxbJmtJg
CeeeILKPffcA7TWkrTWkrTWkvyP//E/qHz3d393ay/JBbSXKrCbCyHXbfKT4uNG3u/xFW7KrZVl/9x+5c7z3J
k08+eddddy0uLiZRW2tUZtZwiuZRW2tUVCbqbQOI7qwsICkISKSiUJFZVKEioqlSRUVCpJqKhUklBRqUTlWiSholJ
JQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqUTlWiSholJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpU
kVFQqSaioVJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSicq1SEJFpZK
CRWVShIqKpUkVFQqSaioVJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSicq1SEJFpZKEikolCRWVShIqKpUkVFQqUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKpUkVFQqSahcv

4+cODAMAxMVKD3vrm5ef78+eeff35tbU2dzWaAylVUrkUSKip/OjUJExVIojJJoiZhogJJADUJOyXpv
QOZqExUtrXWVP4UKtuSACqTJGoSIAnghEkSKkm4SpLe+1/4C3/hM5/5zEc/+tHXve513/zmN7/whS98
6EMfYqImOXPmzMGDB9m1U1R2ffucevIJMsxms49//OOPP/74Rz7yka985Sv33nvvF77whbe//e2z2cz
J1tbWpUuXxnFcWVkBxnFUx3HsvavjOKrz+Xwcx957EhVQgd67ylWScIUkKjslUYEkbFOBJPyZqUmAJC
oTNQmgAkkAlUkSlaskUZMwUZPwZ5CEbWoStqmtNa6gJvEKgNpaG4YhiQqoSZgk4SpJuEISrqCurq4+9
NBDn/rUp17/+v/j137tV37u536Oyebm5rlz544cOcKuK0Rl17fJxceefGEpfX3z4x//+Hvf+9719fXb
b7/92LFjFy9e3LNnzxve8IZ77713ZWVlc3Pz6aefPnny5NGjRw8fPPgyovXegT+bzee/9/Pnzjz32mHr
s2LG9e/eq8/m89z5OLl68+PDDDz/55JPr6+tAktaayv8mKpCE/9+SMFGTqEn4M3OSROUKKldQk6hAEp
U/gyTDMLTWAJUrJOEKSdgpCTslUVtrN91006c//ff/bWvffVv/+2/3Xs/c+bMoUOH2HWFq/OCf+ZmfAXrvs9ksOwSVKx9L/+O6+8B6iJFFq
Oz6Nnn69DMf+Cf/5BW33/GTP/mTm5ub//Sf/tOf+ZmfAXrvs9ksyd69e0+cOPGGN7xhfX19YWHh7rvv
XllZUcdxVMeJurW1tbm5eerUqYeeeujQoUO33npknEyn883J88999999wDDzwzDPPzOdzJklms1lrLQm
QBEgCJGGShCskYVvvPQmg8qdQkzBRgSMVLYlUZMwUZOoSdimJlGTsE3lKiqgAr13tjkBnAAq21T+bI
ZhaK0BKldIwiQOJOyVhpyRM1EyAX/zFX/qDP/iDf/SP3velL33pTW960/r/r6+6dPtTTVEEUBlogJOgGEEYXvay133Xd33X3X3X3XffvbCw0Ht37Ovr6w9+/Wu/Wu/8zu/JJs/+ZjWeBJf
QOt/BUB/zUx/8QlGhFZ5bxW1ZlqszZk2Zr4+k2dOOf33Y2NVN57NEbvRXOJ0tRcGrN22k/bY/AVk81mJuv0tCNuj8/4u8/RdC6AV57//g4eVUA8QHvUoIp5ZDAKz1o0Nb/HRYXsYxhs4YJknpWjJc2T4bE9kH7e9Xsl7uamxs5w29ZUtt9XbvWHrFh3P0+/YzsyXOHaOeRzWI4Dqmjja1Xh47tqhYYc26U1Pj7kJoi9p94ZVyS9/dbdVwTqDUGYDrmOvgxXCBYFc7eBNcqjsTMGDiUSqNsOafgqqUfJM6CxfdiI8s3CzqOUhi19T2VkXM9N4Z7/lyn+jjfJ/Y8tJ2Wdf/k7zmO98ug==
MMpEU7ddbmCEdZQPVNqwzo3uSXScd76Mf3vc2DbK5fvrJW8CgZBpe3/Ctv2aeSpzcyqRBXFZaAiUJgaVsOVkUFcVuWQCSJglRZOCeWnk+V+fTcgvzvAEVk04IuMTUJIZnaxsoDdoTYPVyAF3mKjcZ4e8VjvAZbqcxsm5lOU1cD8m9kSD9/yBpfO9w4Ry6Dws/UNXeVz1FFXpN5OJycJLMUn3hj1xNpsUQSrJj7vx1YNPHpjyvxm73vyxdwMe/yJlV/wKSQXiBN9BSJ+BClk3sZo5+qSA/ZjHyX9jL0v2pWZmLBqVG51pajzRyM5P41PA9p8vdCd9EqrHk4ov7Zj1ZT4f5a/9B7fNXzJ4W/dXsXsAO2sn3Q1N2kmtEbcr89Q6dPlS7yVcMdeuozA8TJ+E2MOQRi86xoopTT5DUbZTjD6btHUHvrW8a6//Gul0mhfnyUMFTYvcGVjxK2vXr3hG4dTvl7sv4H+zCy1hNfQ==

cePrpp0+cOEHlda973ec///nl5eX/9t/+25kzZ376p3/661//epLTp09TOXDgQO99Pp8DSZiM47i+vk
6ltUal905lGAaukmQ+n1OZzWZU5vM5lYWFBSpbW1tUFhcXqWxublJZWlpiWxK2ra+vU1lZWQFaa2oma
mvtwoULVA4dOsSktZZEzeSZZ56hogKnTp3at2/fT/zET7z1rW9dXFwchuF1r3sdLyWV61lUrkUSKiqV
JFRUKkmoqFSSUFGpJKGiUklCRaWShIrKTmcmd911F5UHHnggSe/9fe9734c//OETJ0586EMfetvb3nb
zzTdTueGGG9RxHNUkbltfX6cyDANXSMJkPp9Tmc1mSbjK1tYWlYWFBbYlYdvm5iaVxcVFdkoCbGxsUF
leXqayvr5OZc+ePeyUBLh06RKVffv2AUmYZNvZs2ep3HzzzUySAK01JqdOnaLysz/7s1/6Opf+wT/48
fl8M0lrbTabtdZe97rX8VJSuZ5F5VokoaJSSUJFpZKEiopkolCRWVShIqKpUkVFQqSaio7HT69OnLly/f
eeedVB566CFgHMckvffWu99Pp/fe++9VG688cbW2jiOvXc1iZMLQq5Qmc1mQBJ22traorKwsJRUWwUAEULn
C1tYWlcXFRSqbm5tUlpaWqGxsbFBZWVnhKknW1tao7Nu3TwWSEmYXLhwgcr+/fuZZZAJk8sILL1A5ev
RoJkBrLcmRI0f+83/+z/v376fy3//uc6cQTKtB7d5JkHMfjx49TUR977LEPPfOAAv/ALv6DecMMN
wIkTJx5++++GEqS0tLTpJwhc3NTSrLy8uAyrYkwPr6OpWlpSW1tQaobNvY2KCytLTEJAlXWF9fp7KyspK
EKyQBLl26RGVsR8GmRzxnz5+ncvDgQSYq0FrLo5PyKa7Nu3TwWSAEmYXLhwgcr+/fuZZAJk8sILL1A5ev
RoJkBrLcmRI0f+83/+z/v376fy3//uc6cQTKtB7d5JkHMfjx49TUR977LEPPfOADv/ALv6DecMMN
wIkTJx5++GEqS0tLTpJwhc3NTSrLy8uAyrYkwPr6OpWlpSW1tQaobNvY2KCytLTEJAlXWF9fp7KyspK
EKyQBLl26RGV1dZWdkgDnz5+ncvDgQSYq0FrL5LnnnqPyR3/0R6997WvVz3/+869z/+9asffvjh97//A/
P55uc+9zkqf/iHf8i21lomrbUkwzAACwsLd911F7t2isq1SEJFpZKEikolCRWVShIqKpUkVFQqSaioV
JJQUdnpiccfF+644w4qTp04NwwqTz31NwwDM5/NbbrkF+Ot//a+//YEf/6l9R+YEf
+IHW2jAMKpPWWrtrcBrTWgbUvy0Y9+lMpP/dRPzWazdzxPGazhckwDLPZbahYTabvfvd76bye7bye7
3e6212Wy2sLAwmywsLAzDcMcdd1CZz+cqqqWYEPxXovasrKytUvvGNb3gNXb4wTtU9U4U4L777qPPy4IMPttaGYdja2qLSWmvbjcYH2HYe
only9fvnDhwl/5K3+F+c5//I//8cte9rKfTS2syXU333zmUfl1i1/8oportNaStNaYQCSH
D9+nF07ReVaJKGiUklCRaWShIpKJQkVlUoSKioqlVJFRUVtUoSKiqVJFRUKkmoqOz0jW98E2Y2Fx+fbbjlE5c+YMNHUYht57
a20Yoq6urlJ5/vnnnx3EExnHsvQNqa+3o0aNnUnnnvvuUXeck6iZqEmOHjXmoXOHIkSNUnvuuuuUSY
qoCYBjhw5QuZWZ55JAiRRgdaaEzQnTp2+urUGjO933bbVS+9rWvJVEBbL377bZdVbS+9rWvJVEBbJ0mAV73fVTS+9rWvJVE
PoBOi933bbbVS+9rWvJVEBJ0mAV73qVVS+8pWvMFGTtNaybRgGQkjrUSKiqVJFRUKkm
FSSUFGpJKGiUklCRaWShIrKTg899NDKysp73/veYDexxPBgSDaaWmJytrausrEbUlWV/dSef75
55kkUZOwKFDh6g8/vjjc77vOrG1N9fvGo5tvrVn8OKPFNSTTIqb1qEJT6XNbT4+7nnQ0FNMvWk21HLIJ+QAl1+3MAdzy4KKemMm0TzAbHIKHkO6H/34w0ppa1lo+Wi2u5
7hZ6Pp+XiLH2b9qqCI8q7AkqwGU3dK5hZMCk9yTyqn7lrx6r2Rn36Rm7GXHe5u9+Fp+ESOCcXQKYPuRn5
wTb3llluoPPPMM2rvPQmgkluueUWKqdOnaCJcRp9N7nl1lupPPPUPD7uzWvJl/9Q7z6ssk7iBbP5k0Vzr4ewK
RUQE0yDENr5EXw8lccZ9dOUbkWSaioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKjt9EK1x9QBRNJlL7elTsJ
irjxlprDbAFUJMAbVii0t2MqICaDE5mC0tU5lsbahIVaKICs+UVKhvra6aruaBFBba4tLK1TW1Ze
qEASJ3v27KGytp0SWXgnNUYO/evVQuXbVm7u0SJ3I+p7Qm2qX1LWBcuvFudrf0aMh4Kgw7nDIR7m4RtzY0jWE4Z
bGqAmAfbv30flzKt/ZJkR4Gd0Wqkog6vxxwtZ2e6GpCoZQqW+GaCMXqEtPRcnHatBLt/XQBkAlbcUG
SholJJQkWlkoSKSiUJFZWdHv3mI4uLi1Ruft9nR76Kv9dTB9JbvB1sbGxQWVpaYrsk8s/OkK0gTiUcVPFXCpJpQKhUklBRqSShos
m4kAdQkKpM2LFHp4wYTNYmaP9EWqYzzdRaG4xFWTWYaHv7hA9fW7z6ueLPkLE+eF/wWAW68D5NFbQ8t0u4K2dhTKvi4AEcrBB
w2yZyrixxiSJkyTqbHkvlfXLFwu0uLvfe++++A1SeOnWyxWJeWyxTpYr4M3jw7tQbA4aRn76dZTz1+2ybbO5QeIrMgZ8
hQx5XoFQFXBXuLP0aTAfN1DR0ZK9uRmV1XkATkAZWpQcaANJGhtQl2cUT1pU56+kTA15fBHqK9Dm+hAJrTRY1N
AMJNDtXR1my1Tsm4CaRGXsmQ1A2iIV++Y4jq21+XzeWhuH4dOdYvWmFF43rR1Y4uE0y3CJKWOhUdHdi2M7gYdawXMYa7ecRX
7qcbpIeBtJ7twWYLaxQcWtdZZJEBdRhaQ+VvnlZBZIAKpNhaQ+VjfWLTdR5pAukO+K+1YNUTj35hNon
6jiOvXcgSWsNaK3HdcecrJ2LVTVK5FEioqlSRUFVFQqSaiqVJFRUKkmoaqKSykyk6PFvxFY7f
eTmW+fsmWWRoJLbwoAZIFKn7LmERNBuiA2tqMim7htyQhsfckahuWqPRXo/feWlN778MwRDoOs2Uq43
y9tab23ofZrI9jEnWYLVOZb10G1GEYeu9JAHW2sEKljxtM1CRqXtTNwjKVvnkZSNJ7b8Ng70za4gqVc
WNNba2pQO+doTUZlvZQ2Vy7AKjAiC9Kou7dd4DKU6dOqr33cRzdMNN9lnjsDdJ17FS0nlehaVa5GEikol
CRWVShIqKpUkVFQqSaioVJJQUakkoaKy08lH2uNY3fcSWW+finbbAGSkBctUNFRBZJwhWSgMo5bLfb
eW2tA7+RFrSUDlXG+3lpTe+9sSzLMqn0Ybn9qe9GwRMWt9R4a6d
hao4sCWVimMs7Xm6hJfFLE7UtrlDpm5fVJCMmcTKQYWkPlY31i036i8KLVKD3vm/1IJVTTz4BjOPYe
1d772rv/e4Tr+KlpHI9i8q1SEJFpZKEikolCRWVShIqKpUkVFQqSaioVJJQUakkoaJSSUFGpJKGiUklCRaWShIrKTmcmd912F5UHHnggSe/9fe9734c//OETJ0586EMfetvb3nb
9UvZxtD4E3nRAhUdVbYlAXrvTXUZ0trFMb5OmNvwzDvY+W
39EQAACAASURBVJJhNptvbSWZLaxQGefrQJLee17IeS17UZdIWV6g43+jjCLTWSIB5H2dtyGyJilvrKttUJs
PSHip98zKgAgGSEQfSFleobF2+mKT3zmRu70h33+pBKqeefELtvY/jqPbe1d77iVfew0tt0wJ5XoWlWuRh
IpKJQkVlVoSKioVJJQUakkoaJSSUJFpZKEikolCRWVShIqKjs9e9

aMMCFfscOjCOY7apbViiMs7XkwBqEibqMFumMt+6nKG10RHbbHDsQJI2LFEZ5+tAa42uvfeQri2zhRU
q863LTdTWmppEBdriCpWtzTVgEJIRgXSTDEt7qIwba6213nsStfc+m81678PSHirrly8ATXpQASd79x
2gcurJJ9Q+GcfRCXD87lfyUlK5nkXlWiSholJJQkWlkoSKSiUJFZVKEioqlSRUVCpJqKjs9Nij31hcX
Dx67DYq48Ya0IahYyZqXtQWqfQ+B5JARw1hUFubUenjhtpaG8cRSKImGWbLVOZbl4fZzN7Z5mS2sEJl
vnW5SQ9N1MyGdHsYZstU5luXgYEAPaSbRG2LK1Tsm31rztB670mAdJO0xRUqmxuXmgzD0HtPgsq3DEt
7qGxdvpgESKICakuG5b1Uti5fHBFQARVQ9+47QOWpUyf7VZIcv/uVvJRUrmdRuRZJqKhUklBRqSShol
JJQkWlkoSKSiUJFZVKEioqV1Aff+ybi4uzo8fuoNI3LwdIenhRa03Ni9oilS9/6YskaBIgE+A7XnMfl
f/5wJdba2prTU3Se09yz6tfQ+XrX/uqmqT3nonaWrv7xKuoPPT1B9UkY+9J0NYacPeJV1F5+KGvAa21
3nsSJ0mO3/1KKg99/cEkTNRMgLuOn6Dy0NcfHIZhPo6BJL33YTaz9+N3v5LKNx55CGitAWprLZPbbr+
TysULZ1prQBIVUIG9+w5QOfXkE2rvfRzH3rvaewfuPvEqXkoq17OoXIskVFQqSaioVJJQUakkoaJSSU
JFpZKEikolCRWVK1y6dGnv3r2nTp06duwYlXFjrbUG2AIkUV944YXDR26h8sY3vrG1NgzDfD5XoUNL8
vnPf57Kd37ndwIq4KT3rn7pS1+icu+997bWgHEcl5eX19fXk6gPPPAAlde//vUbGxtqktaaCiR+5Sv/
k8prXvMalUkSYDabbWxsfPWrX6Xy+te/Xh0nS0tLTpJ88YtfpPLa174WUJmoSYAvf/nLVN74xjf03od
h0Lyo/S989rP/lcr3fu93t9bUcRxba+rW1tab3/zmf/7P/28qJ594LMk4jn2bmuT43a/kpaRyPYvKtU
hCRaWShIpKJQkVlUoSKiqVJFRUKkmoqFSSUFG5wtbW5lOnnlxYWDh67DYq8/VLwzD0kERtw4DpvQ+zR
Sr33/9xJmPf6r0PbQFoA+/44XdT+djHfmk2DLPZrE+S9N6T/PA73kXllz95P+BkNpvN5/Pee2vth9/x
Lir/+v6PA22ytbWVRG2t/dAPv5PKJz7+0WEYWmu99yQqkOTtP/QOKr/yy/+6984kiQr03t/xzh+h8iu
//K9ba1tbW7PZbBxHoLU2n8/f8c4fofLLn7wf6L0nGccxCZN3vPNHqPzi//OR2WwGbG5u/sZv/Mb6xk
ZrbRzH3/3d36Ny6skngPl8Po6jk967euKV9/BSUrmeReVaJKGiUklCRaWShIpKJQkVlUoSKiqVJFRUK
kmoqFzh61//2r69e5IcPXYblXFjLRNb2JYXtUUq4zja5733+Xw+jiPjfBzH1tr+G49QuXj2eRVQW2sB
EmDvDTdSuXTuBTWJmgRIMo7j6sHDVC6efT4JV1CBfQcOUbl49nmVSRKVyerBw1QunHkuyTiOrTVABZK
sHjxM5fwLz6pA77211ntPAhw4fDOVF04/lURNogIqcOjmo1TOPPt0EqC1No4jMJvNhmHYs/8glSdPPq
723sdxVHvvToZhANSl2dKtd97Brp2ici2SUFGpJKGiUklCRaWShIpKJQkVlUoSKiqVJFRUrvD8889eu
nRpZXn5pptvoTJurLXWAFuAJEzSFqn0cSutofwvnQRNW6TSx40kQO+9tTafz4dhUIfZMpX51uUkgNok
yYjAbGGFyrixBqiZzO0DsWW2sEJlvnUZGIgKqK01IAvLVPrWmpqEidpaAzJbodI3LzPpvScBWgJkcYX
KuLEGqElUJrYsLO6hsrl2AVBHfBHgt2R1/wEqp558om8bxxHovQNJWmvqwvLibcfuYNdOUbkWSaioVJ
JQ8f9tD/5ibT0LO7//vs+79t7nj41tbIxtQpikabAh7QUNMGkhqtSZqGqVaJRqRpOOhokmE1VVe1Gph
Y4qFUGmTG/SXuRmmuk0N71opVZRLiJ1rqYalJIATjMNJIMZwPgPAYIdjPH5u9f7fLv8Og897n7Z56zT
/dK9lp/PRzMHyBw1c4DMUTMHyBw1c4DMUTMHyBw1t3jhhW9dvfLywcHRY297e+aMN66WUgKSDSCJpHC
YOXW8ASQlYF0nAZJQDjNnXF8vpWRjrAHJhjqsLmROvXkNqOQ1rsdhGGqtw9GlzLl548pqtVqv16WUJL
VWNqqrC5czZ339yjAM6jiONEnK4cXMqTevAev1GkgCqGXj8GLm1JvXMlGBJECtdTi6lDnjjauV1LiSW
mspRQWGo0uZc+3qy5moSWqtmdxz7wOZ8/xzzySptY7jWGt1koRJksuXL7/1kcfSvR5qtgFkjpo5QOao
mQNkjpo5QOaomQNkjpo5QOaomQNkjppbfP1Pn1/fGFfD8Njb3p45442rpZRsgATIhHKYOdabNQKpJgG
cDKsLmbM+vpYEyOsNqwuZsz6+BqhAMd/DwYXMGW9cTcJGKeN6PQxDJerq4GLmrI+vARnrGEspVDeA4e
hS5tSb14CxJFWqQCbl8GLmrK9fATJRK9kopawOLmbO+voVlUkSJ8DqwuXMuXb15TROaq1J7n3TmzPn+
eeeSVIbJ7VWoJSS5PI99771rY+kez3UbAPIHDVzgMxRMwfIHDVzgMxRMwfIHDVzgMxRMwfIHDW3eO65
Z4ZSgMfe9vbMWV+/UkoBAhJAZaMcZo4eJ0VNohZQ2SirzBnX15OUYK2shlqrmmR1cDFzxhtXVSZjZKN
aycHhpcw5vnkVSFLMRq21lMLGwYXMWV+/ApRSnCQZYzGrC5czZ7x+BTABxjgENclwdClz1tevAEnUTI
Akw9GlzFlfv5KmErUYYHXhcuZcv/ZdNROqY9xIcs+9D2TO155/Vh3Hsdaq1lrVJEyS3Hvfg2956M3pX
g812wAyR80cIHPUzAEyR80cIHPUzAEyR80cIHPUzAEyR80tvvLVL184OKSUxx77ocwZb1wFklCKZANI
QjnMHOvaBIzmVUVHNsph5ozr68VYGMdxRVljCcVwcCFzPL6ujrGUQrXWWkpRh6NLmTPeuJoEqMRJMRu
rC5czZ7xxVR1KqTpGNqobBxfvyZz19StJABVIoiZZXbicOeONqyqgMqm1AsPRpcxZX78CqJmoSYDVhc
uZc+3qy5k4SaLUWt903wOZ8/xzzyQZx7E2ahKglKLe/8D9Dz74cLrXQ802gMxRMwfIHDVzgMxRMwfIH
DVzgMxRMwfIHDVzgMxRc4uvPv3lg4NhWB0++ujbMufmjStA2TCVMElCOcwc680aCyuV1EC1YspwlDl1
vKEmqbUySQKU4Shz6s1rSSwU40aBapJyeDFzxhtXcwsVSDIcXcqc8cbVJGoppdYKZDIcXcqcmzeuDGH
DCeDk4OI9mbO+fiUJoGZSSlGHo0uZc3ztFSCJClTymsOjy5lz7erLSWwyqTVvuu+BzPna88+q6/Varb
U6qbWWSa31zQ899OYHHkz3eqjZBpA5auYAmaNmDpA5auYAmaNmDpA5auYAmaNmDpA5am7x3NNfYZVy4

ehtD/9Q5ow3rpZSKgGSAJlQDjPHutaRUqKZ1FpLKZTDzBnX14EkGGslCajl8GLmeHw9G3A8roEkQ3jV
wYXMWV+/UkqptWYC1FqHYSiHFzNnvHG1ErUYNpJaa1bD6uBi5ty4/gqQZgi1VmB14XLmjDeullKO6wi
k4FiHsFEOL2bOjeuvZDKESkq1FopZXbicOdeuvpxEkuprktRa33Tfg5nzteefrbdIUmvNBEjy8EMP33
v//eleDzXdD9xXn/7ywcGwWh2uDodSykAZNgKrYQhDKUkoJZBEAiSREJKSV9VsaBpJqmmATGqtmVCtZ
EMtG+bPDSVjVTMUCTAer7/zne889dS/eObZ5+PxQw899M4nHn/LW95yMKyGUMexDEMSYG1Vh1BKWa/X
maxWq3EcgSQQkKTWmqFQLVAVSKMCKpBEBVQgiQQoJCaVqAdlGMexlKIC0WPrASXJSIAkQ1BJqg7DEEh
hHMchBNZ1XFFqrUmASlSqpRR1jEmollKSjDEJ1QJVK9mgmskYM5GkulFrTVJrLWW1Xq+d1FqTjOOoju
PopDZqkn/5xx/PktTsMtR0P3BfffrLh4erYTgYDspAWa1WQyilrIbBpJTCa0pJItkAMpEQjEnIoIaah
GCtgAKmGceRiVprTcL3GMkLf/bif/vf/Nrf//v/1fHNmxnXpZQxAk6GkGSMSVaF0Qz0DwZ889/wn/49P
/vW/+teAJKwGN9ZjmYzjmAQ4ruOKopKYALVWoJRSawWSAGNMUgzJOqbgWFcUFVArGUISQK21AtFaeFV
1KOUYhxoSSqnkk5/85JNP/p8f/shH1sfHOmasbMQUnNREMwxDKauvfOUrv/Vbv/Whv/HvP/zww2NMQh
XIxEk0MMZSCtUxZkJ1jEncICWotdYkVYYxnG0qRN1HEe1TtQ6SfLj73wiS1Kzy1DT/cB95ctPXbh4e
TUMZcVAGSZAaZKUUpIASYA0EoIxCSETJVRCNImaxAmvqW4A2QBr5WD1j/7Rb3zoQrTxXJOI5qrTXJj7/ziSxJzS5DTfcD98UvfuGey
5dLKavVahgok4MylCZJKSUgAXIryIavSiEJRgVUQM1ELSYQX5WE1VCP10CG8l9+9GMf/9hHXY9JKimG
xER9/Cf+lS8/9YX1/9YX1ej3G3/md3/ngBz+YBPi5n/u5/+0f/+NrV6+WUtiw1nDz5vrX/+E//E/+4/9oWJX
18QhYSFLMhpMCJpUMwULGWmCsNUNhXf6Ho4MbDAdr3zJc+fmsbtb1wXpIcSMNoCYBnAAWitkAysHq7/
zyf/Df/YN/UOtaLeZ7Di5e+vjHP/P/7Nb3zj0qVLbY16qLUCfvlVdQBD9VW1OpQhZGMov/
hef+MTfo8pqqLUCSf7J//8P3//tTh7IMw5AE+JW/94mPzCsqHwFQrSkXAdxxVy1Ga9hzw/8CCpAmjIJxVRSzEYlG8Dn/ujzv/epT//ep
tQ2LQ0hCYnJjPf72
b//2K6+8cjwZx/H4+Hi9Xt+4cQP44Ac/+wR/8Xz/zM3+J6qsKmahJVCATNYm
aBFBrrTDUWnVMoiRZr9fAOI7qOI5qrVWtr5cEeOfj78qS1Owy1HQ/cC9tfPvFYRgODw9LHpTAMpZRhGE
opwDAM6kEZMpRiNlgNadQkQBIVSKJm4qSYSooZYzEqkEqESt5DVUk5RSMpT/5X/9ze+89NLf+lt/0U4D
MMYqY6RoWBeozI5Pj7+6o+/6Ec/+tM//8pfbIDv+M0//+zM/8pbIRSVGTVKIyUZOwUU2ztpZS1DTFjCUlFLNRCeAGwaiA
CiRRgSQqoCYBktRah2FQj46OfvM3f+vg4OAv/+/V/i2oBk0qKqeRWtdYhJAFevvLKr//6f/8Lv/ALjz7
61rJhKlGTAMWolbxGzQRQk6hJ8VWjmhS1ToD1eg3UWtrfNTBOktRax3x3F0kkR94l0/kSWp2WWo6f7/8K
UvffFgtTo4OCilDAPDBFitVQUmlJKKlJKvj+MRE2iFqMmqbVaLXm9dQhjDEJkEQFVECCttQ7DcHx8
OXKlVrrhQsXjo6OSjVBgrqqAxAo6OgFrrwcFBknFcF9lQk1Ty/6ICaQA1CVBrBZK4QUpQSylqCqkmUZMAOUEF1EzYqCYp
pYwxE8lQQ3JsHYbhc5/73/73730/qq/9a3r129+97m8+97vTe9N+4CbB97znzrnPQ888ACTUopRSsl
KsWADJFGBWitQa4Qa4UkJPFVJKm1JlGWqdqHWijuNYNM0CPPGun8iS1Oxy1XpWlZlnmKUZMAKpDECZBEzRxAzQRQc4shHNeRiQqoSUlSa83EE2qt
6jiOap2o6/U6iVonSdRiro6OLh48fCegfZztAlqTmjeRLX/riUMpqtSqlHBwMQCnl4OASk
FKBOglKImATJR83pqJmqtNcmly/flTMcTlo5/TmNC9/50Ugifqm+x7MnO++/GeAeu+b3pwz9f
J3XkwC1Fp5TTVJJSqgJgHUNICaCZBEzUQF1LyqJNGRSa01iQDpklprEie1e1VhWotaq1VnUcxzqm
eT/USmrCxcu/OiP/+HBwcHFgJBpAlqXmD+aPP/+HBwcHh4WEpZbmVIzGEMlrFzegJYkmlkJVRTbMMkkbEhR
oiaptQJJ1J1ETNxj33PpAz9lEmFTO67/6GcqZe+/SO134e
ahIkKqPlzJalqGkBNSiY6JgGS1FphUDNxAqi11iRqHHAD67iHMdaq5N5aa5qqlHpwePHmx3GHXsg9i3feBmm

0AWZKaN6TPf/4PDw6Hw9XRMAyr1QostwDKBFCSmkRJKqAmUfKqmkSpdQ088OaHc6Ze+NY3mSQ1Sa0Ve
PChR9K8+MI3kgKoDz70cOa8+MKfZvLgQw/nTL3wrW8mAZOoeVVJo2NeVdKoSYAkar4PNUkpRa21AknG
cQSSqEm8BTCOYxIn46TWmkSttapJ1FjW403K4cVL5fLFe97+wz+S7vtDzTaALEnNG9U4jp/7w392dHR
0cHCQ5ODgoJQCDMOQZBgGoJSiMslETaICtVZArbUCSd76yGNpvvH1ryVRSym11iSlFDUJkESttQKllC
QqoD7y6NvS/Ok3v57Jw299NHO++Y0/AdQkb33ksTTf+PrXmDgB1EcefVuaP/3m1zNRmdQJkyQqoAJJa
q2A+uhjP5TmG1//mgqogFpKcZJbqEmA3KLWCqiAmjlqEhVQkzhJoiZRa61J1Fqrk1qrWmtVa61Jaq2Z
uD7m0uEFLj7x7nelux3UbAPIktS84T355GcoXjy6BKxWqyTDMNCUUgAVUAEnQBo1k3f8hR9N88xXv5J
ETQOoQCZAEhXILX74HT+S5rlnv5oE+KG3vyNznnv2q5moP/yOH0nz3LNfVQE1CaD+8Dt+JM2zzzydhE
mtNZNSSpJxHJmogJNSipN3/IUfTfPsM09nojJxkkQFMlGBWiuTWiuQRgVUQE3CpNaqJlEBNQngnFprk
lqrTa3V1yQqcHR0dP3G+N6ffE+6O4OabQBZkpruFp/9/c9Q3SilDMPApJSShEkmgBMgiVpKUd/9E/9q
mj/6/B8CaibqMAy11lJKJiqQRk1SSnn8iXeneeoLf6wC73z8XZnzhX/+R5kA73z8XWme+sIfAyqgZvL
Ox9+V5otP/fM0ak5QgUwANYn6+BPvTvPUF/5YTQKoSYBaayml1poT1CRqKUUFMnFSSqm1JgHUTJwAmT
hJU2tVk6hAErXWal7F5Morxx/86Z9Ktz3UbAPIktR030et9Q8+85nvXruyXq+BYRiAJKUUoJSiJgGSA
Oq/8YGfTvO7n/qdTIDMAZIAahrgfe//qTRPfvbTapL3vu8vZs6Tn/10EkD9yfe+P82Tn/10EkAF1FLK
e/6196b5/Sc/A6iZqEAmKpBGzS1+8r3vT/PkZz+dBMgt1ExUQAUyUfP9qUmATNRMADWvByRRk5j4moz
jzVK9ds/Fb7/nfX813f83qOn2F5BGTQPkVGoaII2aCZA5auYAadRMgDRqGiCnUtMAadTsMiBLUrPLUN
PtLyCNmgbIqdQ0QBo1EyBz1MwB0qiZAGnUNEBOpaYB0qjZZUCWpGaXoabbX0AaNQ2QU6lpgDRqJkDmq
JkDpFEzAdKoaYCcSk0DpFGzy4AsSc0uQ023v4A0ahogp1LTAGnUnACkUdMAadScAKRR0wA5lZoGSKNm
lwFZkppdhppufwFp1DRATqWmAdKoOQFIo6YB0qg5AUijpgFyKjUNkEbNLgOyJDW7DDXd/gLSqGmAnEp
NA6RRcwKQRk0DpFFzApBGTQPkVGoaII2aXQZkSWp2GWq6/QWkUdMAOZWaBkij5gQgjZoGSKPmBCCNmg
bIqdQ0QBo1uwzIktTsMtR0+wtIo6YBcio1DZBGzQRIo+Z2gDRqJkAaNQ2QU6lpgDRqdhmQJanZZajZB
pAlqenODpBGTQPkVGoaII2aCZBGze0AadRMgDRqGiCnUtMAadR0+ws12wCyJDXd2QHSqGmAnEpNA6RR
MwHSqLkdII2aCZBGTQPkVGoaII2abn+hZhtAlqSmOztAGjUNkFOpaYA0aiZAGjW3A6RRMwHSqGmAnEp
NA6RR0+0v1GwDyJLUdGcHSKPmrgBp1EyANGrmAJmjZgKkUXNXgDRquv2Fmm0AWZKa7uwAadTcFSCNmg
mQRs0cIHPUTIA0au4KkEZNt79Qsw0gS1LTnR0gjZq7AqRRMwHSqJkDZI6aCZBGzV0B0qjp9hdqtgFkS
Wq6swOkUXNXgDRqJkAaNXOAzFEzAdKouStAGjXd/kLNNoAsSU13doCcKTUTII2aBkijpgHSqJkAOVNq
uv2Fmm5/ATlTaiZAGjUNkEZNA6RRMwFyptTsMiBLUrPLUNPtLyBnSs0ESKOmAdKoaYA0aiZAzpSaXQZ
kSWp2GWq6/QXkTKmZAGnUNEAaNQ2QRs0EyJlSs8uALEnNLkNN120PyN1S080BsiQ1uww1Xbc9IHdLTT
cHyJLU7DLUdN32gNwtNd0cIEtSs8tQ0+0vICeouQWQiZo5QE5Q0wCZo+Z2gJxKTQPkBDX7AsiS1Owy1
HT7C8gcNRMgjZo5QE5Q0wCZo+ZUQG5HTQPkBDX7AsiS1Owy1GwDyJLUdGcHyBw1EyCNmjlATlDTAJmj
5lRAbkdNA+QENd0bA2q2AWRJarqzA2SOmgmQRs0cICeoaYDMUXMqILejpgFygprujQE12wCyJDXd2QH
SqLljQBo1DZAT1DRAGjV3DEijpgHSqDkBSKOm21+o2QaQJanpzg6QRs0dA9KoaYCcoKYB0qi5Y0AaNQ
2QRs0JQBo13f5CzTaALElNd3aANGruGJBGTQPkBDUNkEbNHQPSqGmANGpOANKo6fYXarYBZElqurMDp
FFzx4A0ahogJ6hpgDRq7hiQRk0DpFFzApBGTbe/ULMNIEtS050dII2aBsgJahogp1LTAJmjpgFyx9Sc
CkijpntjQM02gCxJTXd2gDRqGiAnqGmAnEpNA2SOmgbIHVNzKiCNmu6NATXd/gLSqGmAnKCmAXIqNQ2
QOWoaIHdMzamANGr2BZAlqdllqOn2F5BGTQPkBDUNkFOpaYDMUdMAuWNqTgWkUbMvgCxJzS5DTbe/gM
xRMwHSqLkdIKdS0wBp1JwApFHTAGnUNEAmavYRkCWp2WWo6fYXkDlqJkAaNbcD5FRqGiCNmhOANGoaI
I2aBshEzT4CsiQ1uww13f4CMkfNBEij5naAnEpNA6RRcwKQRk0DpFHTAJmo2UdAlqRml6Gm219A5qiZ
AGnU3A6QU6lpgDRqTgDSqGmANGoaIBM1+wjIktTsMtR0bwxAGjV3DMgdUzMHSKNmAqRR0wCZo2YCZI6
aXQZkSWp2GWq6NwYgjZo7BuSOqZkDpFEzAdKoaYDMUTMBMkfNLgOyJDW7DDXbALIkNd0ygDRq7hiQO6
ZmDpBGzQRIo6YBMkfNBMgcNd3+Qs02gCxJTbcMIBM12wByx9TMATJR0wBp1DRA5qiZAJmjpttfqNkGk
CWp6XYBkEbNHCCNmhOANGoaII2aBshETQOk3t2PMgAABxxJREFUUdPtL9RsA8iS1HS7AEijZg6QRs0J
QBo1DZBGTQNkoqYB0qjp9hdqtgFkSWq6XQCkUTMHSKPmBCCNmgZIo6YBMlHTAGnUdPsLNdsAsiQ13dk
BcqbUTIDMUXM7QO6KmgmQOWq6/YWabQBZkpru7AA5U2omQOaouR0gd0XNBMgcNd3+Qs02gCxJTXd2gJ
wpNRMgc9TcDpC7omYCZI6abn+hpttfQM6UmgmQOWpuB8hdUTMBMkfNLgOyJDW7DDXd/gLSqLkrQBo1J
wC5K2rmAGnUNEAmavYRkCWp2WWo6fYXkEbNXQHSqDkByF1RMwdIo6YBMlGzj4AsSc0uQ023v4A0au4K
kEbNCUDuipo5QBo1DZCJmn0EZElqdhlquv0FpFFzV4A0ak4AclfUzAHSqGmATNTsIyBLUrPLUNPtLyC

```
NmgbIqdQ0QBo1JwBp1DRAGjUNkBPU3DEgjZp9AWRJanYZarr9BaRR0wA5lZoGSKPmBCCNmgZIo6YBco
KaOwakUbMvgCxJzS5DTbe/gDRqGiCnUtMAadScAKRR0wBp1DRATlBzx4A0avYFkCWp2WWo2QaQJanpz
g6QRk0D5FRqGiCNmhOANGoaII2aBsgJau4YkEZN98aAmm0AWZKa7uwAadQ0QE6lpgHSqJkA2YaaUwGZ
o6YBcio13f5CzTaALElNd3aANGoaIKdS0wBp1EyAbEPNqYDMUdMAOZWabn+hZhtAlqSmOztAGjUNkFO
paYA0aiZAtqHmVEDmqGmAnEpNt79Qsw0gS1LTnR0gjZoGyKnUNEAaNQ2QO6bmVEDmqGmAnEpNt79Qsw
0gS1LTnR0gjZoGyKnUNEAaNScAadTMAdKouStAJmoaII2abn+hZhtAlqSmOztAGjUNkFOpaYA0ak4A0
qiZA6RRc1eATNQ0QBo13f5CzTaALElNd3aANGoaIKdS0wBp1JwApFEzB0ij5q4AmahpgDRquv2Fmm5/
AWnU3BUgjZoGyAlqbgfICWrmADlBzT4CsiQ1uww13f4C0qi5K0AaNRMgc9TcDpAT1MwBcoKafQRkSWp
2GWq6/QWkUXNXgDRqJkDmqLkdICeomQPkBDX7CMiS1Owy1HT7C0ij5q4AadRMgMxRcztATlAzB8gJav
YRkCWp2WWo6fYXkDOlZgJkjpoGyBw1JwC5K2r2BZAlqdllqOn2F5AzpWYCZI6aBsgcNScAuStq9gWQJ
anZZajp9heQM6VmAmSOmgbIHDUnALkravYFkCWp2WWo6fYXkDOlZgJkjpoGyBw1JwC5K2r2BZAlqdll
qNkGkCWp6bqum4OabQBZkpqu67o5qNkGkCWp6bqum4OabQBZkpqu67o5qNkGkCWp6bqum4OabQBZkpq
u67o5qNkGkCWp6bqum4OabQBZkpqu67o5qOm67nwAsiQ1uww1XdedD0CWpGaXoabruvMByJLU7DLUdF
13PgBZkppdhpqu684HIEtSs8tQ03Xd+QBkSWp2GWq6rjsfgCxJzS5DTdd15wOQJanZZajZBpAlqem6r
puDmm0AWZKaruu6OajZBpAlqem6rpuDmm0AWZKaruu6OajZBpAlqem6rpuDmm0AWZKaruu6OajZBpAl
qem6rpuDmm0AWZKaruu6Ojpuu58ALIkNbsMNV3XnQ9AlqRml6Gm67rzAciS1Owy1HRddz4AWZKaXYa
aruvOByBLUrPLUNN13fkAZElqdhlquq47H4AsSc0uQ03XdecDkCWp2WWo2QaQJanpuq6bg5ptAFmSmq
7rujmo2QaQJanpuq6bg5ptAFmSmq7rujmo2QaQJanpuq6bg5ptAFmSmq7rujmo2QaQJanpuq6bg5ptA
FmSmq7rujmo6brufACyJDW7DDVd150PQJakZpehpuu68wHIktTsMtR0XXc+AFmSml2Gmq7rzgcgS1Kz
y1DTdd35AGRJanYZarquOx+ALEnNLkNN13XnA5AlqdllqNkGkCWp6bqum4OabQBZkpqu67o5qNkGkCW
p6bqum4OabQBZkpqu67o5qNkGkCWp6bqum4OabQBZkpqu67o5qNkGkCWp6bqum4OabQBZkpqu67o5qO
m67nwAsiQ1uww1XdedD0CWpGaXoabruvMByJLU7DLUdF13PgBZkppdhpqu684HIEtSs8tQ03Xd+QBkS
Wp2GWq6rjsfgCxJzS5DTdd15wOQJanZZajZBpAlqem6rpuDmm0AWZKaruu6OajZBpAlqem6rpuDmm0A
WZKaruu6OajZBpAlqem6rpuDmm0AWZKaruu6OajZBpAlqem6rpuDmm0AWZKaruu6Ojpuu58ALIkNbs
MNV3XnQ9AlqRml6Gm67rzAciS1Owy1HRddz4AWZKaXYaaruvOByBLUrPLUNN13fkAZElqdhlquq47H4
AsSc0uQ03XdecDkCWp2WX/N3y+mOLHwCY+AAAAAElFTkSuQmCC\",\n\t\"errorCode\" :
0,\n\t\"errorInfo\" : \"No error!\"\n}\n",
                "result": 0
        }
}

var generateImagePreviewImageParam = {
        "displayScale":8
}

generateImagePreviewImage(generateImagePreviewImageParam['displayScale'],
                          function (error, data) {
    //通讯超时或websocket打印服务异常
          if (error) {
        return alert(error.message);
    }
    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;

    //预览图生成失败，退出流程
    if (errorCode !== 0) {
        return
    }
```

```
27
28      //解析处理数据
29      var obj = JSON.parse(info);
30      var data = obj.ImageData;
31
32
33  });
```

五、打印接口说明

5.1 开始打印

代码块

```
1   /**
2    * 启动打印任务。
3    *
4    * @param {number} printDensity - 打印浓度，根据不同打印机型号取值范围不同，具体如下:
5    *   - B1、B21、B21S、B21_Pro、B203、B3S、B3S_P、B31、B4、K2、K3、K3W、M2、M3: 取值范围 1~5，默认为 3。
6    *   - B50、B11、B50W、B32、Z401、B32R: 取值范围 1~15，默认为 8。
7    * @param {number} paperType - 纸张类型，可选值:
8    *   1: 间隙纸
9    *   2: 黑标纸
10   *   3: 连续纸
11   *   4: 定孔纸
12   *   5: 透明纸
13   *   6: 标牌
14   *   10: 黑标间隙纸
15   * @param {string} printMode - 打印模式，可选值:
16   *   1: 热敏
17   *   2: 热转印
18   *   注意，不同打印机型号支持的打印模式有限制，具体如下:
19   *   - B1、B21、B21S、B21_Pro、B203、B3S、B3S_P、B31、B4、K2、K3、K3W、B11 仅支持热敏。
20   *   - B50、B50W、B32、Z401、B32R、M2、M3 仅支持热转印。
21   * @param {number} count - 总打印份数，表示所有页面的打印份数之和。
22   *   例如，如果你有3页需要打印，第一页打印3份，第二页打印2份，第三页打印5份，那么count的值应为10（3+2+5）。
23   * @param {Function} callbackFunction - 打印任务启动后要执行的回调函数。
24   * @example
25   * //返回数据示例
26   * {
27   *     "apiName": "startJob",
28   *     "resultAck": {
29   *         "errorCode": 0,
```

```
 *         "info": "startJob ok!",
 *         "result": 0
 *     }
 * }
 * @description 返回结果中的 errorCode 含义如下:
 *              - 0: 成功
 *              - -1: 失败，info 表示原因
 *              - -2: 打印机忙碌，info 表示原因
 *              - -3: 打印机接收到不支持的参数，主要是浓度、纸张类型、打印模式，info 表示具体原因
 * @return {undefined} 此函数不返回任何值。
 *
 */
function startJob(printDensity, printLabelType, printMode, count, callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "startJob",
        "resultAck": {
                "errorCode": 0,
                "info": "startJob ok!",
                "result": 0
        }
}

startJob(3, 1, 1, 10, function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;
    //开启打印任务失败，提示错误信息，退出打印流程
    if (errorCode !== 0) {
        return alert(info);
    }

    //进行下一步操作
});
```

5.2 提交打印任务

代码块

```
/**
 * 提交打印任务，并执行回调函数。
 *
 * @param {string} [printData=null] - 打印数据的 JSON 字符串。
 * @param {string} printerImageProcessingInfo - 打印机图像处理信息的 JSON 字符串，包含打印份数信息，格式如下：
 * {
 *   "printerImageProcessingInfo": {
 *     "printQuantity": 1 // 用于指定当前页的打印份数。例如，如果需要打印3页，第一页打印3份，第二页打印2份，第三页打印5份，则在3次提交数据时，printerImageProcessingInfo 中的 "printQuantity" 值分别应为 3，2，5。
 *   }
 * }
 * @param {function} callbackFunction - 提交作业后的回调函数。
 * @return {undefined} 此函数不返回任何值。
 *
 * @description
 * 需要先开启打印任务，完成绘制后再提交打印任务
 */
function commitJob(printData = null, printerImageProcessingInfo, callbackFunction)
```

代码块

```
//数据提交成功返回数据示例
{
        "apiName": "commitJob",
        "resultAck": {
                "errorCode": 0,
                "info": "commitJob ok!",
                "result": 0
        }
}

//打印进度返回示例1：此回调的含义为第一页第一份打印完成
{
    "apiName": "commitJob",
    "resultAck": {
        "errorCode": 0,
        "info": "",
        "onPrintEPCCodeCompleted": "",
        "onPrintPageCompleted": 1,//打印完成份数回调
        "onPrintPageLengthCompleted": 60,
        "printCopies": 1,//打印完成份数回调(新增)
        "printPages": 1,//打印完成页数回调(新增)
        "printQuantity": 1,//打印完成页数回调
```

```
        "time": 0
    }
}

//打印进度返回示例1：此回调的含义为第一页第二份打印完成
{
    "apiName": "commitJob",
    "resultAck": {
        "errorCode": 0,
        "info": "",
        "onPrintEPCCodeCompleted": "",
        "onPrintPageCompleted": 2,//打印完成份数回调
        "onPrintPageLengthCompleted": 60,
        "printCopies": 2,//打印完成份数回调(新增)
        "printPages": 1,//打印完成页数回调(新增)
        "printQuantity": 1,//打印完成页数回调
        "time": 0
    }
}

//打印进度返回示例1：此回调的含义为第二页第一份打印完成
{
    "apiName": "commitJob",
    "resultAck": {
        "errorCode": 0,
        "info": "",
        "onPrintEPCCodeCompleted": "",
        "onPrintPageCompleted": 1,//打印完成份数回调
        "onPrintPageLengthCompleted": 60,
        "printCopies": 1,//打印完成份数回调(新增)
        "printPages": 2,//打印完成页数回调(新增)
        "printQuantity": 2,//打印完成页数回调
        "time": 0
    }
}


var jsonObj = {"printerImageProcessingInfo": {"printQuantity":1}};

commitJob(null, JSON.stringify(jsonObj), function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode, info, printCopies, printPages } =
JSON.parse(JSON.stringify(data)).resultAck;
    var resultInfo = "commitJob ok";
```

```
69      //异常导致打印终止
70      if (errorCode !== 0) {
71          return alert(info);
72      }
73
74
75      //所有页数据的所有份数打印完成
76      if (printPages === list.length && printCopies ===
    jsonObj.printerImageProcessingInfo.printQuantity) {
77          //结束打印任务
78          endJob(function (error, data) {
79              if (error) {
80                  alert(error.message);
81              } else {
82                  const arrParse = JSON.parse(JSON.stringify(data));
83                  if (String(arrParse.resultAck.info).indexOf("endJob ok") > -1)
    {
84
85                  }
86              }
87
88          });
89          return;
90      }
91
92      //当前页数据提交完成，但是未完所有页数据提交，继续发送下一页数据
93      if (String(arrParse.resultAck.info).indexOf(resultInfo) > -1 && x <
    list.length - 1) {
94          console.log("发送下一页打印数据:  ")
95          x++;
96          printTag(list, x);
97      }
98  });
```

5.3 结束打印任务

代码块

```
1   /**
2    * 结束打印任务
3    *
4    * @param {function} callbackFunction - 结束任务后的回调函数
5    * @description
6    * 收到最后一页最后一份打印页面后调用该函数结束打印任务
7    */
8   function endJob(callbackFunction)
```

代码块

```
//返回数据示例
{
        "apiName": "endJob",
        "resultAck": {
                "errorCode": 0,
                "info": "endJob ok!",
                "result": 0
        }
}

endJob(function (error, data) {
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;

    if (errorCode !== 0) {
        return alert(info);
    }

    alert('打印成功');
});
```

5.4 取消打印任务

代码块

```
/**
 * 取消当前的打印任务，并执行回调函数。
 *
 * @param {function} callbackFunction - 取消打印任务后的回调函数。
 * @return {undefined} 此函数不返回任何值。
 */
function cancelJob(callbackFunction)
```

代码块

```
cancelJob(function(error,data){
    //通讯超时或websocket打印服务异常
    if (error) {
        return alert(error.message);
    }
```

```
    const { errorCode, info } = JSON.parse(JSON.stringify(data)).resultAck;
    if (errorCode !== 0) {
        return alert(info);
    }

    alert('取消打印成功');
});
```

六、回调说明

代码块

```
/**
 * {
 *   "apiName": string, // 调用的 API 名称
 *   "resultAck": {
 *     "errorCode": number, // 错误代码，0 表示成功，其他值表示错误
 *     "info": string, // 信息字符串，描述操作结果
 *     "result": number // 结果代码，通常与 errorCode 一致
 *   }
 * }
 */
{
  "apiName": "commitJob",
  "resultAck": {
    "errorCode": 0,
    "info": "commitJob ok!",
    "result": 0
  }
}
```

七、错误码相关说明

7.1 错误码说明描述

代码块

```
* 0-无错误
//打印机返回部分
* 1-盒子打开
* 2-缺纸
* 3-电量不足
* 4-电池异常
* 5-手动停止
```

```
 * 6-数据错误
 * 7-温度过高
 * 8-走纸异常
 * 9-正在打印
 * 10-未检测到打印头
 * 11-环境温度过低
 * 12-打印头松动
 * 13-未检测到碳带
 * 14-不匹配的耗材
 * 15-用完的碳带
 * 16-不支持的纸张类型
 * 17-设置纸张类型失败
 * 18-设置打印模式失败
 * 19-设置浓度失败
 * 20-写入rfid失败
 * 21-边距参数错误
 * 22-超时错误
 * 23-断开连接
 * 24-画板参数设置错误
 * 25-旋转角度参数错误
 * 26-json参数错误
 * 27-出纸异常（关闭上盖检测）
 * 28-检查纸张类型
 * 29-碳带与打印模式不匹配
 * 30-设置浓度不支持
 * 31-不支持的打印模式
 * 32-标签材质设置异常，请重新设置
 * 33-不支持该标签材质，请更换或重新设置
 * 34-不支持RFID写入
 * 50-非法标签
 * 51-非法碳带和标签

//内部使用
//E_UNKNOW_ERROR = 255,
```