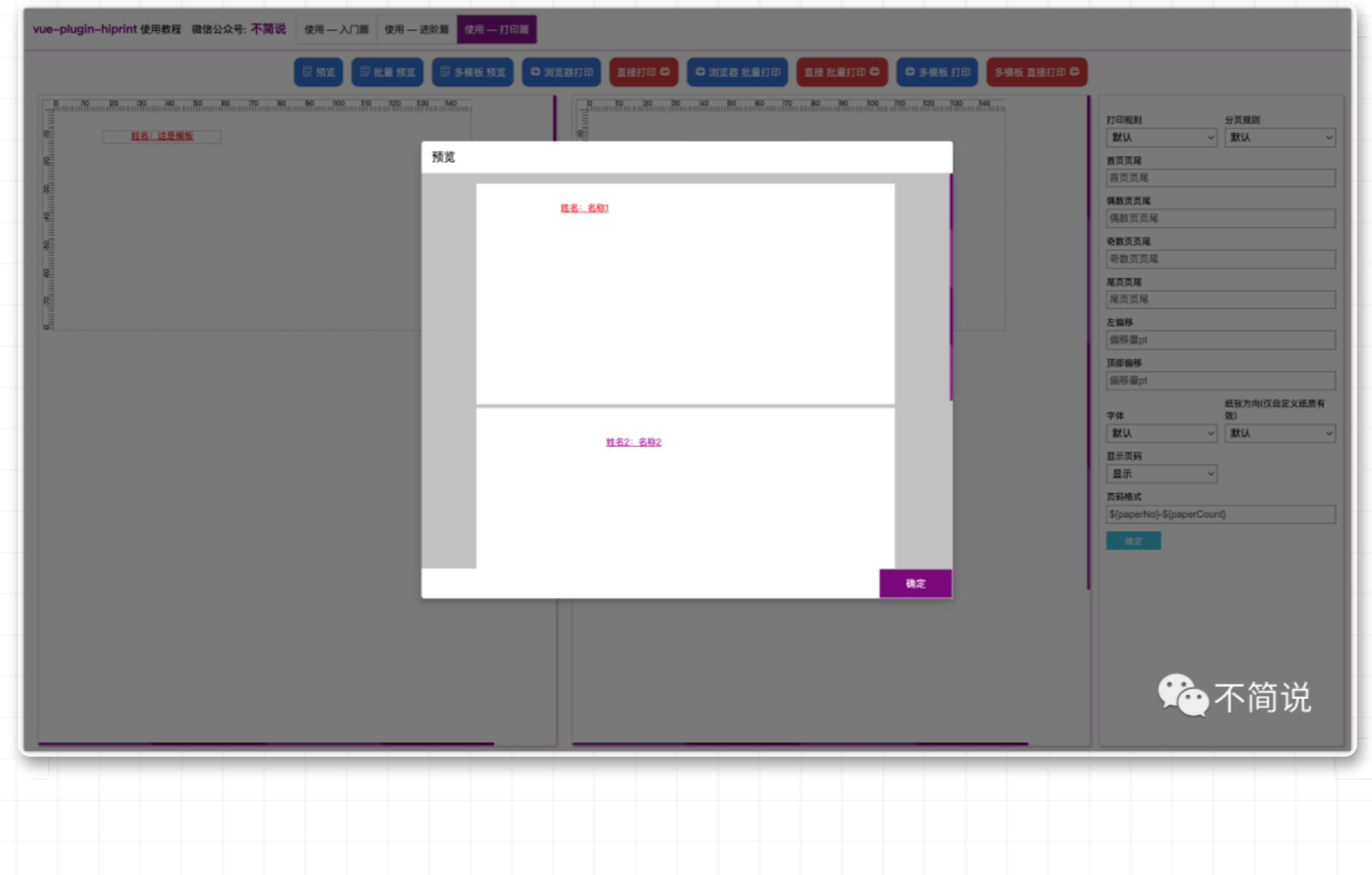




收录于合集 #vue-plugin-hiprint 4 #hiprint 4

相信你看过了 [\[vue-plugin-hiprint\] 使用-进阶篇](#) 已经知道了 模板对象结构，如何查看可用API。本打印篇 将讲解 打印相关API 及 注意事项。

源代码：https://github.com/CcSimple/vue-plugin-hiprint-start



噯 1. 前言

首先我们应该清楚两种 打印模式：

1. 拖拽 生成的html，创建 iframe 然后调用 浏览器 提供的 print API
2. 将 生成的html，通过 socket.io 发送到 直接打印客户端 再调用 electron 提供的 print API

提供的 打印 API 如下：

1. 浏览器打印：
 - 模板对象：print、printByHtml
 - hiprint对象：print
2. 借助打印客户端：
 - 模板对象：print2、printByHtml2
 - hiprint对象：print2

其中 hiprint对象提供的 print、print2 适用于多模板打印；模板对象适用于 单/多面板 打印。

噯 2 填充数据

关于 填充数据/数据源 的问题，很多小伙伴都问过我。先看一张图：

名称	类型	描述
title	string	标题栏内容 field 存在: title为标题, 打印结果为 title;data, field 不存在: title为内容, 打印结果为 title
field	string	字段名称 printData 的 key
testData	string	测试数据 打印设计时作为data 交互打印时请再去管理控制台

如上图，其中打印设计时指的就是 设计器 区域。设置字段名即可在调用 print、print2、getHtml 时填充对应的数据。如果不想显示 冒号: (标题+数据)这种，设置参数 hideTitle:true 即可。

上代码：

```
let printData = {name:'打印/预览时显示的名称'};
let hiprintTemplate = new hiprint.PrintTemplate({xxx});
// 弹出浏览器打印窗口, 并填充 字段名: name 的元素
hiprintTemplate.print(printData);
let hiprintTemplate2 = new hiprint.PrintTemplate({xxx});
// hiprint对象打印
hiprint.print({
  templates: [
    { template: hiprintTemplate, data: printData },
    { template: hiprintTemplate2, data: printData },
  ],
});
```

OK，看了上面的 核心代码 及 描述，相信你已经很清楚如何 填充数据 了。

噯 3 获取预览html

预览功能 其实是 拿到生成的html 然后去渲染。所以 getHtml 是需要 先填充数据的。

示例：

```
let printData = {name:'名称1'};
let hiprintTemplate = new hiprint.PrintTemplate({xxx});
// printData 如果是数据, 那就是数据预览了
let html = hiprintTemplate.getHtml(printData);
// 渲染html
console.log(`${'#preview_content'}.length : `, `${'#preview_content'}.length`);
$('#preview_content').html(html);
```

预览就是这么简单的 😊，但是也需要 注意：如果你是 弹窗预览，那么一定要先能够获取到 \$('#preview_content') 这个容器才行哟。

噯 4 浏览器打印

关于 浏览器打印，由于各家浏览器 内核 不一样。打印效果 有所不一。这里默认使用的是「Microsoft Edge」基于 Chromium。

经测试 基于 Chromium 的浏览器可正常打印。如：Edge、Chrome 等。

上代码：

```
let printData = {name:'名称1'};
let printData2 = {name:'名称2'};
let hiprintTemplate = new hiprint.PrintTemplate({xxx});
// 打印
hiprintTemplate.print(printData);
// 批量打印
hiprintTemplate.print([printData, printData2]);
```

浏览器打印 更多参数：

```
// printData 为数据 预览 批量打印
hiprintTemplate.print(printData, {
  // 将渲染面板偏移设置
  leftOffset: -10, // 左偏移
  topOffset: -10 // 上偏移
}, {
  callback: () => {
    // 无法捕获, 用户是否点击了 打印/取消 按钮
    console.log('浏览器打印窗口已打开')
  },
  styleHandler: () => {
    // 这里拼接返回html->head标签内的css/style
    // 1. 例如: 使用hiprint官网的样式
    let css = '<link href="http://hiprint.io/Content/hiprint/css/print-lock.css" media="print">';
    // 2. 重写样式: 所有文本红色
    css += '<style>hiprint-printElement-text{color:red !important;}</style>';
    return css
  }
});
```

由于 打印原理 是 生成html 调用浏览器打印。所以如果 批量打印 的数据过多 会造成浏览器 卡死/等待时间长 等问题。

噯 5 直接打印

直接打印 其实就是将 填充完数据的html 通过 socket.io 连接发送到 直接打印客户端

打印客户端源码：https://github.com/CcSimple/electron-hiprint

5.1 获取打印机列表

直接打印 可以指定 打印机；同时本客户端也提供了一些 其他Api；详见源码。

默认 当网页打开时--在 jQuery 的 \$(document).ready 时去 自动连接 打印客户端。所以如果你是 先打开网页，再去打开打印客户端，那么你需要 刷新网页，或者使用 命令重新连接。

判断是否连接及重新连接：

```
// 判断是否已成功连接
if (hiprint.hiwebSocket.opened) {
  // 注意: 连接是异步的
  // 已连接
}
// 重新连接客户端 当然你可以在连接局域网内的客户端
hiprint.hiwebSocket.setHost('http://localhost:17521')
```

获取打印机列表：

```
// 模板对象获取
const printerList = hiprintTemplate.getPrinterList();
// hiprint对象获取
const list = hiprint.hiwebSocket.getPrinterList();
```

注意：这样获取的是 连接成功时「本机的打印机列表」

重新/刷新 获取打印机列表：

```
// 这是异步的
hiprint.refreshPrinterList((list) => {
  console.log('refreshPrinterList')
  console.log(list)
});
```

5.2 打印

```
let printData = {name:'名称1'};
let printData2 = {name:'名称2'};
let hiprintTemplate = new hiprint.PrintTemplate({xxx});
// 直接打印 请使用系统设置的 默认打印机
hiprintTemplate.print2(printData);
// 批量直接打印
hiprintTemplate.print2([printData, printData2]);
// 直接打印回调
// 直接打印到打印机成功
hiprintTemplate.on('printSuccess', function (e) {});
// 直接打印到打印机失败
hiprintTemplate.on('printError', function (e) {});
```

由于客户端原理一样是需要先 渲染html，所以 批量打印 传的数据过多 会造成客户端 卡死/等待时间长 等问题。所以建议使用 队列形式 进行 批量打印。

直接打印 更多参数：

```
// printData 为数据 预览 批量打印
hiprintTemplate.print2(printData, {
  // 将渲染面板偏移设置
  leftOffset: -10, // 左偏移
  topOffset: -10 // 上偏移
});
// 指定打印机, 或 参数
hiprintTemplate.print2(printData, {
  printer: '', // 打印的 名称
  title: "打印任务名称",
  color: false, // 是否打印颜色 默认 true
  copies: 1, // 打印份数 默认 1
});
```

其他参数可查看源码https://github.com/CcSimple/electron-hiprint

噯 6 多模板预览/打印

有了 前面两节 的基础，多模板打印 也就很容易理解了。

示例：

```
let printData = {name:'名称1'};
let printData2 = {name:'名称2'};
let hiprintTemplate = new hiprint.PrintTemplate({xxx});
let hiprintTemplate2 = new hiprint.PrintTemplate({xxx});
// 多模板预览
hiprint.getHtml({
  templates: [
    { template: hiprintTemplate, data: printData },
    { template: hiprintTemplate2, data: printData2 },
  ],
});
// 多模板打印
hiprint.print({
  templates: [
    { template: hiprintTemplate, data: printData },
    { template: hiprintTemplate2, data: printData2 },
  ],
});
// 多模板直接打印
hiprint.print2({
  templates: [
    { template: hiprintTemplate, data: printData, options: {} },
    { template: hiprintTemplate2, data: printData2 },
  ],
  options: { printer: '', landscape: true },
  function (data) {
    console.log('printSuccess');
  },
  function (e) {
    console.log('printError');
  }
});
```

代码一看，瞬间明了。

总结

- 需要理解 打印原理，有问题才能知道大概出在哪
- 需要 结合关键字去探索 相关API 的 更多可能
- 整篇下来难点并不多，新手们需要 多多摸索

打印篇 到此结束，如有不清楚的知识，大家 一定要学会自己查阅相关资料。

欢迎各位码友 转发 及 留言反馈，觉得不错就 点个赞 再走咯~

收录于合集 #vue-plugin-hiprint 4

← 上一篇 · [\[vue-plugin-hiprint\] 使用-进阶篇](#)

喜欢此内容的人还喜欢

- 布隆过滤器
Java程序员进阶
- 一种无文件落地技术(BOF)
猫头鹰安全团队
- Python与量化选基——1. 基金池构建
ForTheLead