



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

[vue-plugin-hiprint]^[1]默认分支是 npm 包及 demo 的融合代码；对于部分新手来说有点难以下手。本篇开始以新的 demo 开始讲述如何使用 vue-plugin-hiprint。

demo项目通过 create-vue^[2]创建，默认 vue3，如果你有哪里不清楚的，欢迎各位 留言 反馈，demo链接见文末。

图 1. 安装

```
npm i vue-plugin-hiprint
```

引入「打印样式」print-lock.css

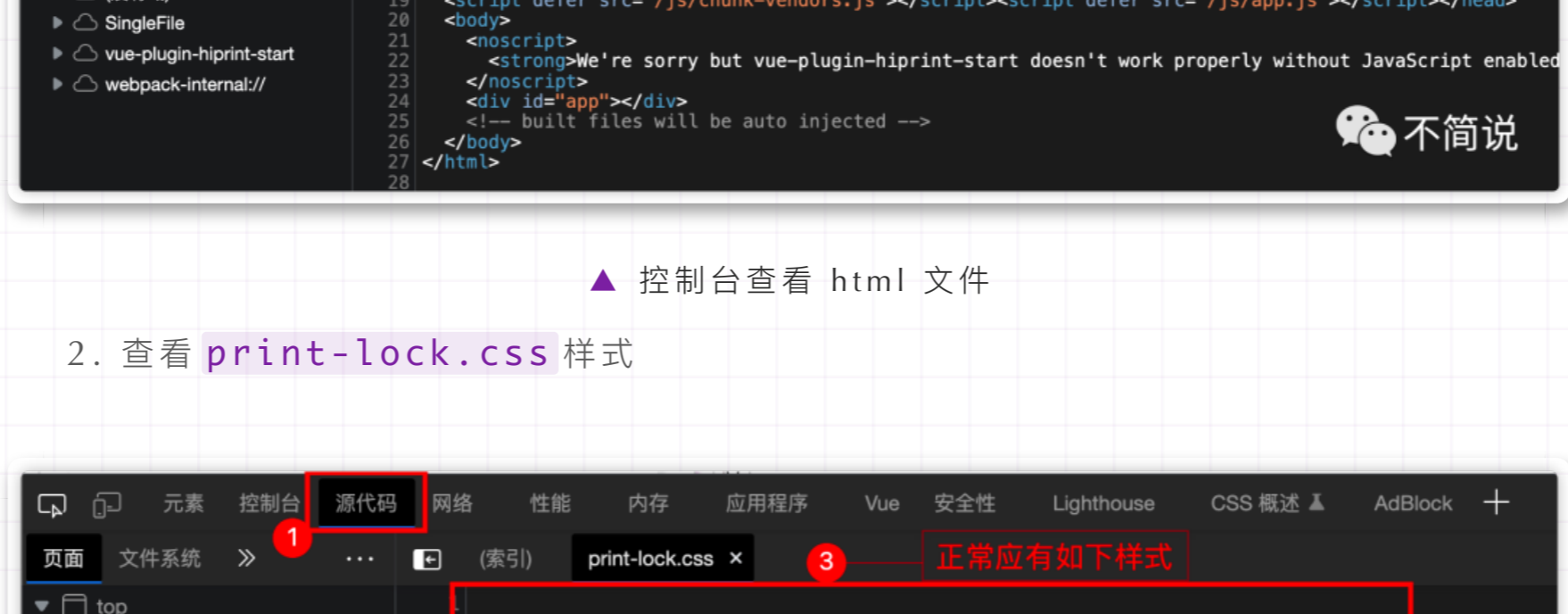
- 1. 从 node_modules/vue-plugin-hiprint/dist/ 目录下复制一份 print-lock.css 到静态资源目录 public 下
- 2. 在 public/index.html 中添加样式如下图：

```
<link rel="stylesheet" type="text/css" media="print" href="/base/hiprint-lock.css" />
```

这个打印样式是必须引入的，否则会打印重复，多出一页。

如何检查是否引入成功呢？

- 1. 浏览器 控制台 查看



- 2. 查看 print-lock.css 样式

```
body print { margin: 0; }
body print { margin-left: 20px; padding: 10px; }
```

图 2. 构建可拖拽元素（自定义显示样式）

这里以默认的 defaultElementProvider 做示例。

如果想知道如何自定义可拖拽元素provider，请查看上一篇文章 [\[vue-plugin-hiprint\] 如何自定义可拖拽元素provider](#)

2.1 编写页面

```
<div class="flex-row justify-center flex-wrap">
  <div class="title">基础元素</div>
  <!-- tid 与 defaultElementProvider 对应 -->
  <!-- class="ep-draggable-item" -->
  <div class="ep-draggable-item item" tid="defaultModule.text">
    <i class="iconfont sv-text" />
    <span>文本</span>
  </div>
  <div class="ep-draggable-item item" tid="defaultModule.image">
    <i class="iconfont sv-image" />
    <span>图片</span>
  </div>
  <div class="ep-draggable-item item" tid="defaultModule.table">
    <i class="iconfont sv-table" />
    <span>表格</span>
  </div>
</div>
```

- 必须先在 init 对应 provider
- tid 与 defaultElementProvider 中对应
- 元素dom 必须包含 class="ep-draggable-item"

2.2 初始化provider并构建可拖拽元素

```
<script setup>
import { onMounted } from "vue";
import { hiprint, defaultElementProvider } from "vue-plugin-hiprint";
// 初始化 provider
hiprint.init({
  providers: [defaultElementProvider()],
});
// ...
onMounted() => {
  buildLeftElement();
};
const buildLeftElement = () => {
  // eslint-disable-next-line no-undef
  hiprint.PrintElementManager.buildByHtml(`${C}ep-draggable-item`);
};
</script>
```

图 3. 构建设计器

设计器 就是打印设计的画布，将元素拖入其中即可开始 拖拽设计 你的 打印模板 啦~

首先插入一段 html 如下：

```
<div class="flex-5 center">
  <!-- 设计器内 草稿 -->
  <div id="hiprint-printTemplate"></div>
</div>
```

然后在对应时机去 构建 (design)

```
onMounted() => {
  buildDesigner();
};
// ...
let hiprintTemplate;
const buildDesigner = () => {
  // eslint-disable-next-line no-undef
  hiprint.PrintElementManager.empty(); // 先清空，避免重复构建
  hiprintTemplate = new hiprint.PrintTemplate({
    settingContainer: "#PrintElementOptionSetting", // 元素参数容器
  });
  // 构建 并挂载到 容器中
  hiprintTemplate.design("#hiprint-printTemplate");
};
</script>
```

注意：必须要在 onMounted 中去构建设计器，因为要先查找该 容器。

其中 hiprintTemplate.design("#hiprint-printTemplate") 就是去执行构建，传入的参数是 jquery^[1] 选择器。同理 settingContainer 也是插入 jquery 选择器。

当执行完 design 后，我们将看到如下：

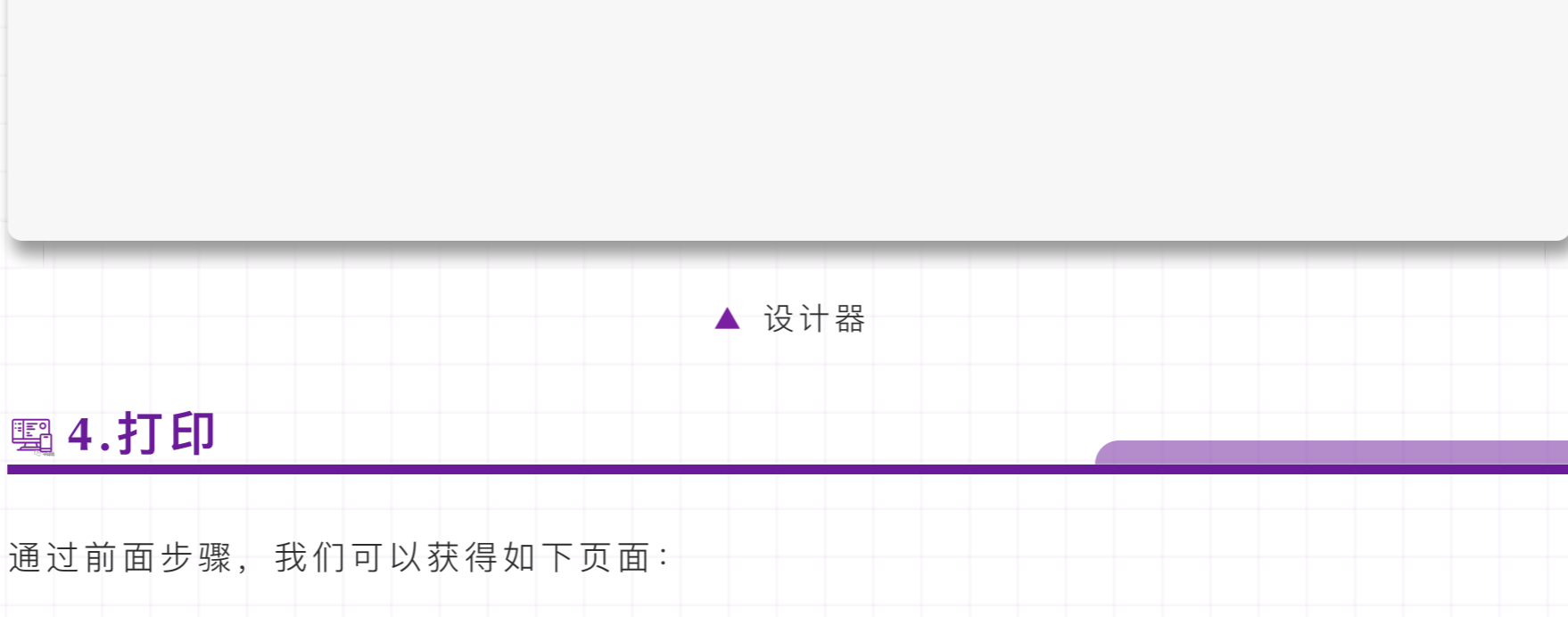


图 4. 打印

通过前面步骤，我们可以获得如下页面：



ok，入上图我们已经 插入了 两个文本进入了 设计器 区域

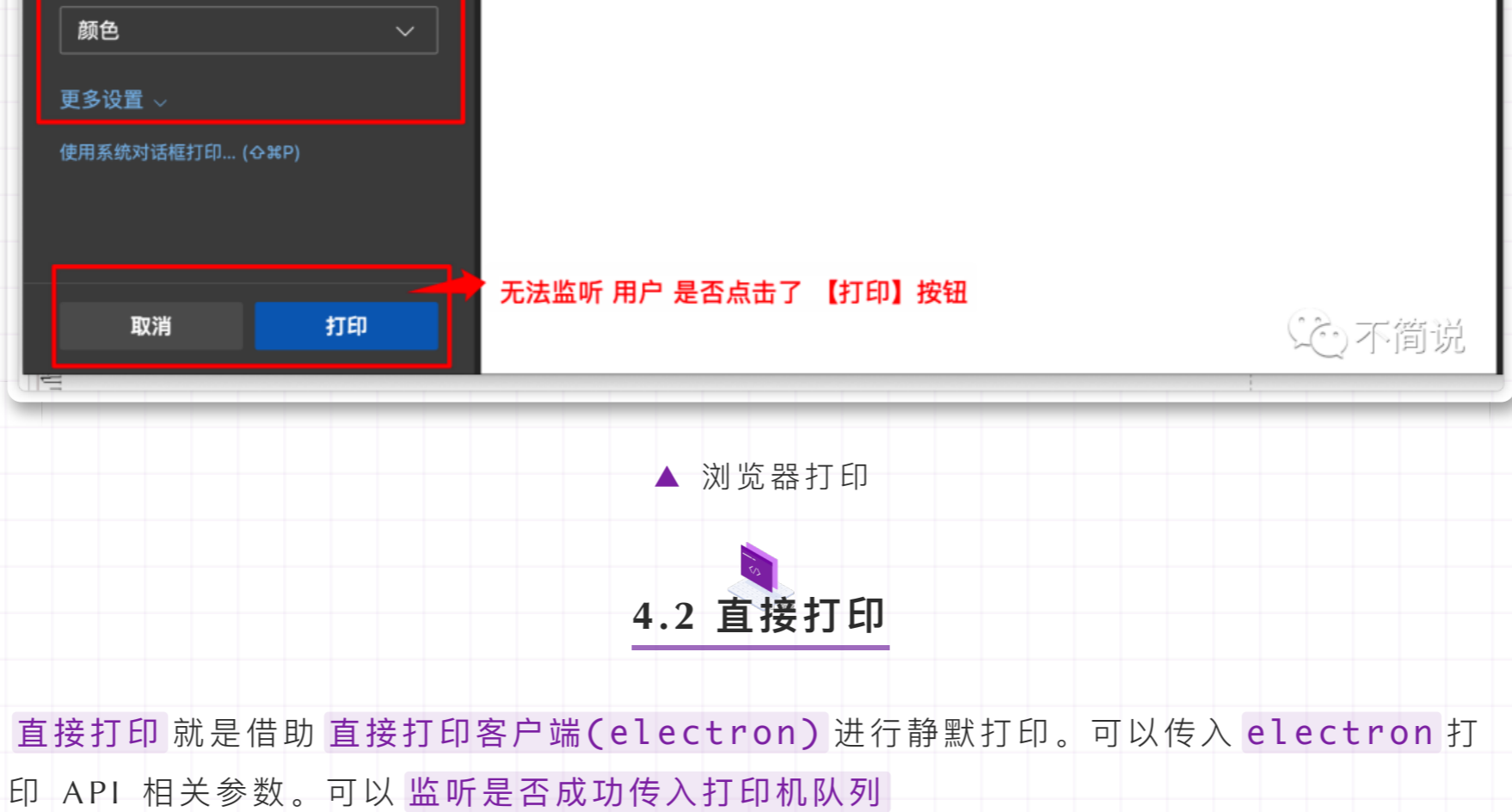
我们来尝试打印！

4.1 浏览器打印

调用浏览器打印功能进行打印：

```
let printData = {name: 'CSimple'};
let options = {leftOffset: -1, topOffset: -1};
let ext = {
  callback: () => {
    console.log('浏览器打印窗口已打开');
  },
  styleHandler: () => {
    return '<style>hiprint-primElement-text{color:red !important;}</style>';
  }
};
hiprintTemplate.print(printData, options, ext);
```

浏览器没有提供相关 API 所以无法监听 用户 是否点击了打印 按钮 如下图：



4.2 直接打印

直接打印 就是借助 直接打印客户端(Electron) 进行静默打印，可以传入 electron 打印 API 相关参数。可以 监听 是否成功传入打印机队列

示例：

```
const print2 = () => {
  if (hiprint.hiwebSocket.opened) {
    hiprintTemplate.print2({});
  } else {
    alert('请先连接客户端(刷新网页)，然后再点击 [直接打印]');
  }
};
```

print2 相关参数较多，入门篇 先不一一列举。在接下来的文章中再详细列举。

总结

- 打印样式 print-lock.css 必须正确引入
- 构建 可拖拽元素 一定要先 初始化 相关 provider
- 构建 可拖拽元素 和设计器 一定要在 挂载完 dom 后 (onMounted) 执行

入门篇 到此结束，demo 中的代码会涉及很多基础知识，所以一定要学会自己查阅相关资料。

如有不清楚的可以加我微信，记得备注 start，我好拉你进群：



demo源码: <https://github.com/CSimple/vue-plugin-hiprint-start>

demo预览: <https://ccsimple.github.io/vue-plugin-hiprint-start/>

参考资料

- [1] vue-plugin-hiprint: <https://github.com/CSimple/vue-plugin-hiprint>
- [2] create-vue: <https://github.com/vuejs/create-vue>
- [3] jquery: <https://jquery.com/>

收录于合集 #vue-plugin-hiprint 4

上一篇 [\[vue-plugin-hiprint\] 如何自定义可拖拽元素 provider](#) | 下一篇 [\[vue-plugin-hiprint\] 使用-进阶篇](#)

