

html-pdf-node 打印插件

文档地址: <https://www.npmjs.com/package/html-pdf-node>

插件安装:

```
npm i html-pdf-node
```

```
npm i puppeteer
```

```
<style>
  .page-break-before {
    page-break-before: always; //分页符
  }
  .page-break-avoid {
    page-break-inside: avoid; //避免在元素内部插入分页符
  }
</style>
<script>
  // 分页逻辑
  function addPageBreak() {
    let doc = document.querySelectorAll('.echarts-page'); //获取所有可能需要分页的元素
    let index_h = document.getElementById('indexHome').offsetHeight; //封面高度
    let a4_h = 952; //A4纸: 794×1123, 浏览器打印内容高度: 588*1040, pdf插件打印内容高度: 595*952

    let prev_top = 0; //上个分页节点位置
    doc.forEach(item => {
      let t = item.getBoundingClientRect().top; //当前节点所在位置
      h = item.offsetHeight; //当前节点内容高度
      let surplus = (t + h - index_h - prev_top) % a4_h; //取余

      if (Math.abs(surplus) < h) { //当前节点被分页截断
        item.classList.add("page-break-before"); //添加分页

        let classList = item.className;
        if (classList.indexOf('echarts-page-table') !== -1) { //分页标签在表格上面, 若表格可
          被分页截断, 执行以下代码
            let tab = (h - Math.abs(surplus)) % a4_h; //取余
            if (Math.abs(tab) >= 100) { //打印时底部留白区域范围
              item.classList.remove("page-break-before");
            } else {
              prev_top = t - index_h;
            }
          } else {
            prev_top = t - index_h;
          }
        }
      }
    });
  }
</script>
```

html-pdf-node 打印脚本: pdf-print.js

```
var html_to_pdf = require('html-pdf-node');
var fs = require("fs");
```

```

let options = {
  args: [
    "--no-sandbox",
    "--disable-setuid-sandbox",
    "--allow-file-access-from-file",
  ],
  format: "A4", //pdf插件打印内容高度: 595* 952, A4纸: 794×1123
  // landscape: true, //是否横向打印
  // scale: 2, //缩放0.1-2之间
  margin: {
    top: "15mm",
    bottom: "15mm",
    left: 0,
    right: 0,
  },
  displayHeaderFooter: true, //是否展示页眉页尾

  /**
   * 页眉页尾可配置参数
   * @params title标题 date时间 url路径 pageNumber页码 totalPages总页码
   */
  // 页眉模板
  headerTemplate: `
<div style="color: #111111;margin: 0 auto;width:12cm;font-size: 2.5mm;border-bottom: 0.1mm
solid #d6d6d6;padding-bottom: 0.5mm;">
  <span class='title'></span>
</div>
`,

  // 页尾模板
  footerTemplate: `
<div style="color: #111111;width: 12cm; margin: 0 auto; border-top: 0.1mm solid
#d6d6d6; font-size: 2.5mm; padding-top: 0.5mm; text-align: right;">
  <span class="pageNumber"></span><span class="totalPages"></span>
</div>
`,
  printBackground: true, //打印背景图形
};

function printPDF(param) {
  let file = param.file || [{ url: "https://example.com", name: 'example.pdf' }];

  // 生成PDF文件并保存至当前文件夹
  file.map((item, i) => {
    html_to_pdf.generatePdfs([item], Object.assign({}, options, item.options)).then(output => {
      // console.log("PDF Buffer: ", output); // PDF Buffer: [{url: "https://example.com", name:
      "example.pdf", buffer: <PDF buffer>}]

      // fs.writeFile('文件路径', '要写入的内容', ['编码'], '回调函数');
      fs.writeFile(item.name, output[0].buffer, "binary", (err) => {
        if (err) {
          console.error("文件保存失败", err);
        } else {
          console.log(`${item.name}文件已保存!`);
        }
      });
    });
  });

});

});
}

```

```
//执行打印
printPDF({
  file: [{
    url: "https://example.com",
    options: {
      displayHeaderFooter: false, //是否展示页眉页尾
    },
    name: 'test_bak.pdf'
  }, {
    url: "F:/项目文件/企业预警通app/mobile/hf/bondReports/html/pages/test.html",
    name: 'test.pdf'
  }]
});

// export default {
//   printPDF
// }
```

cmd终端执行打印pdf命令

```
node .\pdf-print.js
```