



Puppeteer初探--爬取并生成《ES6标准入门》PDF

原 javascript node.js 青菜叶子 2017年08月18日发布

首先介绍Puppeteer

Puppeteer是一个node库，他提供了一组用来操纵Chrome的API（默认headless也就是无UI的chrome，也可以配置为有UI）

有点类似于PhantomJS，但Puppeteer是Chrome官方团队进行维护的，前景更好。


使用Puppeteer，相当于同时具有Linux和Chrome的能力，应用场景会非常多。就爬虫领域来说，远比一般的爬虫工具功能更丰富，性能分析、自动化测试也不在话下，今天先探讨爬虫相关

[Puppeteer官方文档请猛戳这里](#)

Puppeteer 核心功能

1. 利用网页生成PDF、图片
2. 爬取SPA应用，并生成预渲染内容（即“SSR”服务端渲染）
3. 可以从网站抓取内容
4. 自动化表单提交、UI测试、键盘输入等
5. 帮你创建一个最新的自动化测试环境（chrome），可以直接在此运行测试用例
6. 捕获站点的时间线，以便追踪你的网站，帮助分析网站性能问题

OK，基本熟悉之后，接下来进行爬虫教学：

1. 使用 `puppeteer.launch()` 运行puppeteer，他会return一个promise，使用then方法获取browser实例，[Browser API猛戳这里](#)
2. 拿到browser实例后，通过 `browser.newPage()` 方法，可以得到一个page实例，[猛戳 Page API](#)
3. 使用 `page.goto()` 方法，跳转至[ES6标准入门](#)
4. 在 `page.evaluate()` 方法中注册回调函数，并分析dom结构，从下图可以进行详细分析，并通过 `document.querySelectorAll('ol li a')` 拿到文章的所有链接 
5. 拿到所有链接之后，依次爬取各个页面（也可以promise all同时抓取多个页面），使用 `page.pdf()` 方法打印当前页面
6. 核心代码如下



首页



问答



专栏



讲堂



更多

```
puppeteer.launch().then(async browser => {
  let page = await browser.newPage();

  await page.goto('http://es6.ruanyifeng.com/#README');
  await timeout(2000);

  let aTags = await page.evaluate(() => {
    let as = [...document.querySelectorAll('ol li a')];
    return as.map((a) =>{
      return {
        href: a.href.trim(),
        name: a.text
      }
    });
  });

  await page.pdf({path: `./es6-pdf/${aTags[0].name}.pdf`});
  page.close()

  // 这里也可以使用promise all, 但cpu可能吃紧, 谨慎操作
  for (var i = 1; i < aTags.length; i++) {
    page = await browser.newPage()
    var a = aTags[i];
    await page.goto(a.href);
    await timeout(2000);
  }
});
```

完整代码访问 Github

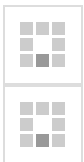
<https://github.com/zhentao/p...>

效果如下，这里简述几个需要注意的问题：

如果在page go之后马上进行pdf抓取，此时页面还未完成渲染，只能抓到loading图（如下），所以用timeout做了简单点处理



最终爬取效果如下，PDF的尺寸、预览效果、首页重复就不做过多整理，预览效果如下,如果想要自己处理，可以设置一下chrome尺寸，打印页数



最后声明，生成的PDF很粗糙，应该不会对阮老师产生什么影响，如有问题可以第一时间联系我....

赞 | 15

收藏 | 37

你可能感兴趣的文章

[Node.js介绍](#) 29 收藏, 2.1k 浏览

[初探Node.js](#) 1 收藏, 82 浏览

[node.js初级基础一](#) 6 收藏, 574 浏览

10 条评论

默认排序 时间排序



viko16 · 2017年08月25日

使用 `page.waitForSelector` 比 `timeout(2000)` 更好一些

<https://github.com/GoogleChro...>

👍 赞 +1 回复

嗯, 这里api看的不够细致, 有些waitfor不错, 有些场景timeout也挺好

— **青菜叶子** 作者 · 2017年08月25日

@**青菜叶子** 什么时候用timeout会比waitfor更适合些呢

— **typescript** · 2017年10月14日

[添加回复](#)



菜鸟一号 · 2017年09月12日

大神 我使用puppeteer 有一个按钮是from表单提交, 使用click,不能提交咋办

👍 赞 +1 回复

1 有什么报错、提示么?

— **青菜叶子** 作者 · 2017年09月12日

试一下`page.mouse.click()`

— **妄与空** · 2017年10月17日

[添加回复](#)



走走_停停 · 1月29日

👍 赞 回复



valleykid · 6 天前

安装puppeteer时如果无法下载Chromium，可以使用[puppeteer-cn](#)代替的。这个包先检查你本地chrome是否大于59，再决定是否安装Chromium，并且使用国内源安装，速度很快且保证成功。

👍 赞 回复



青菜叶子 作者 · 6 天前

如果安装不了，可以把npm切换到淘宝源试试，`npm --registry https://registry.npm.taobao.org`，不建议换库

👍 赞 回复

哈哈，你误解我意思了。puppeteer下载Chromium的默认路径是google服务器，GFW的原因可能会下载失败。puppeteer-cn还是依赖了puppeteer的，只是改写了其中的[launch方法](#)，并主动检测本地chrome是否符合headless条件，不符合会使用国内源安装Chromium。

npm切到淘宝源可以加速puppeteer的安装，但是无法改变Chromium的默认下载路径，还是存在问题的。

— valleykid · 5 天前

[添加回复](#)



文明社会，理性评论

发布评论

滴滴云

1核 1G 20GB
SSD云服务器限时特惠
包月仅需0.9元

立即抢购



青菜叶子

131 声望

关注作者

发布于专栏

系列文章

[Puppeteer再探--自动把SF文章推荐到掘金](#) 1 收藏, 565 浏览

[Puppeteer终探--前端监控](#) 7 收藏, 460 浏览

Copyright © 2011-2018 SegmentFault. 当前呈现版本 17.06.16

浙ICP备 15005796号-2 浙公网安备 33010602002000号 杭州堆栈科技有限公司版权所有

CDN 存储服务由 又拍云 赞助提供

[移动版](#) [桌面版](#)