

郑小凯的个人博客

知识总是要一点点积累的

使用selenium把网页保存为PDF

📅 2019-03-05 | 📅 2019-03-09 | 📁 python , 网络爬虫与数据分析 | 💬 0 | 👁




背景

前面通过selenium爬取了微信公众号“新世相”的所有文章链接，详见[使用Selenium获取微信公众号的所有文章](#)。那么接下来就该获取具体文章了。由于网页是含有图片的，想想还是通过浏览器把网页打印成PDF保存好了，同时保存一份不含图片的文本文件，可以用于后续分析。

那么怎么使用selenium打印PDF呢？

思路

在网上找了找解决方案，主要有如下几种：

- 利用第三方包：pdfkit，可参考：<https://www.cnblogs.com/silence-cc/p/9463227.html> 
- 使用chrome的 `-print-to-pdf` 模式，将请求到 html 导出为pdf，可参考：
<http://osask.cn/front/ask/view/1029784> 
- 使用js命令 `'window.print()'` 来调用浏览器打印，可参考：
<https://gitee.com/shinemic/codes/09y87ph6vf2c5zamwls3q48> 

这里我们选用第三种，相对来说适应性比较好，也方便查看进展，如果想隐藏页面，只需要加入 `-headless` 选项即可。

实现如下：

- 配置chromedriver的options

```
1  appState = {
2      "recentDestinations": [
3          {
4              "id": "Save as PDF",
5              "origin": "local"
6          }
7      ]
8  }
```



```
7         ],
8         "selectedDestinationId": "Save as PDF",
9         "version": 2
10    }
11    profile = {
12        'printing.print_preview_sticky_settings.appState': json.dumps(appState),
13        'savefile.default_directory': './articles'
14    }
15    chrome_options = webdriver.ChromeOptions()
16    chrome_options.add_experimental_option('prefs', profile)
17    chrome_options.add_argument('--kiosk-printing')
```

这里 `savefile.default_directory` 用来指定保存文章的路径，需自行配置。

◦ 保存pdf

```
1  driver.get(url)
2  time.sleep(5)
3  # 保存PDF
4  temp_title = driver.title
5  driver.execute_script('window.print();')
```

这里chrome打印网页时默认文件名为网页的 `title`，所以这里先保存一下

`temp_title=driver.title`。

◦ 改名

```
1  os.rename('./articles/' + temp_title + '.pdf', './articles/' + title + '.pdf')
```

由于如果打开同一个网站的多个页面并保存pdf，那么很可能就会出现由于网站title相同而覆盖的情况，所以每次保存完毕后，改一下pdf的文件名。

注意：当网页异常等情况可能出现title为空的情况，那么这里改名的时候就会报异常错误，需要进行异常处理。

实现

根据上述思路，在打开网页、导出pdf、改名之后加上sleep，防止异常。实现如下：



```
1  def get_articles():
```

```
2     appState = {
3         "recentDestinations": [
4             {
5                 "id": "Save as PDF",
6                 "origin": "local"
7             }
8         ],
9         "selectedDestinationId": "Save as PDF",
10        "version": 2
11    }
12    profile = {
13        'printing.print_preview_sticky_settings.appState': json.dumps(appState),
14        'savefile.default_directory': './articles'
15    }
16    chrome_options = webdriver.ChromeOptions()
17    chrome_options.add_experimental_option('prefs', profile)
18    chrome_options.add_argument('--kiosk-printing')
19    driver = webdriver.Chrome(executable_path='./chromedriver', options=chrome_opt
20    driver.implicitly_wait(60)
21    count = 1
22    with open('articles.csv', newline='') as csvfile:
23        spamreader = csv.reader(csvfile, delimiter=';')
24        for line in spamreader:
25            try:
26                title = line[0].split(';')[1]
27                url = line[1]
28                print("下载第" + str(count) + "篇, 标题: " + title)
29                driver.get(url)
30                time.sleep(5)
31                # 保存PDF
32                temp_title = driver.title
33                driver.execute_script('window.print();')
34                time.sleep(10)
35                os.rename('./articles/' + temp_title + '.pdf', './articles/' + tit
36                # 保存txt
37                content = driver.find_element_by_id('js_article').text
38                with open('./text/' + title + '.txt', 'w') as f:
39                    f.write(content)
40                count += 1
41            except Exception as e:
42                logging.exception(e)
43    driver.quit()
44    return
```



完整代码参考: https://github.com/keejo125/web_scraping_and_data_analysis/tree/master/weixin

如果大家有更好的方法，也欢迎分享。

打赏

[# python](#) [# 爬虫](#)



[← 使用Selenium获取微信公众号的所有文章](#)

[使用matplotlib绘制时间序列图表 >](#)

| 昵称 | 邮箱 | 网址(http://) |
|---|----|-------------|
| <div>随便说点什么~</div> <div>表情 预览</div> <div> 回复</div> | | |

Code 403: 访问被api域名白名单拒绝，请检查你的安全域名设置。

Powered By [Valine](#)
v1.3.10

© 2020 郑小凯

由 [Hexo](#) 强力驱动 v3.8.0 | 主题 – [NexT.Gemini](#) v6.6.0

