



链滴

Python+Selenium 实现浏览器自动化操作

作者: [mnizht](#)

原文链接: <https://ld246.com/article/1575888466352>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Python+Selenium 实现浏览器自动化操作

python适合写脚本，正好黑客派的签到老是忘记，所以.....

先网上搜一下，selenium这个工具看起来挺好用，安装的话，有pip、npm安装，或者下载安装的方，这个根据个人情况自己定就好。

因为我用的Chrome浏览器，所以驱动用了chromedriver,需要注意自己的浏览器版本和下载的驱动对应（一般版本号前两位一样的都是可以的）

完成自动签到就是定时在签到页面点一下签到按钮（好像也可以直接调那个签到的接口，不过咱也不道接口是啥，况且这里主要是想试一下浏览器的自动化操作，就不考虑直接调接口了）

签到代码

```
# 签到地址
checkin_url = 'https://hacpai.com/activity/checkin'

driver.get(checkin_url)
# for cookie in driver.get_cookies():
#     print("%s -> %s" %(cookie['name'],cookie['value']))
#
# print('=====')
# cookie可以先本地实际登陆一下，然后从浏览器中或者抓包工具取到
driver.add_cookie({
    'name': 'symphony',
    'value': '7e9f0c6f4ce59cc9ab1c3c72536bec63bacfb08a23d93dcd2029ab9afe41fc9de4
c2c424a977cc5b87921fb09da53e3dea781d9e3cbb07944b2b6af9cd7a9c59ddd74c88188f2190
cfa12b7b5a402e20cbd65b647a74bb497e6af00c8a58f3a79484d7556e0e18fa86dd394cc49230
666930819e12863f0b7118fa814ff59'
})

time.sleep(1)
# 重新发送请求，由于添加了cookie，此时应该是登录状态
driver.get(checkin_url)

# 拿到 签到按钮
btn = driver.find_element_by_css_selector('.btn.green')
btn.click()
```

然后，就成功了

然后，ip被封了

- 。
- 。
- 。

当时试着试着发现打不开黑客派网站了，但是使用手机移动网络却可以，联系了一下D哥，

嗯-----，果然是ip被封了。。。说是短时间内多次访问，，，还好说了一下就给解封了，感谢D哥

还是不搞这个了

从京东爬取商品评价

于是又想着从网上爬点数据试试，于是几经折腾，终于完成了从京东爬取商品评价的功能。

(当时正赶上各种因为pc犯法的事件频发，整的我也是担心哪天就被叫去喝茶了)

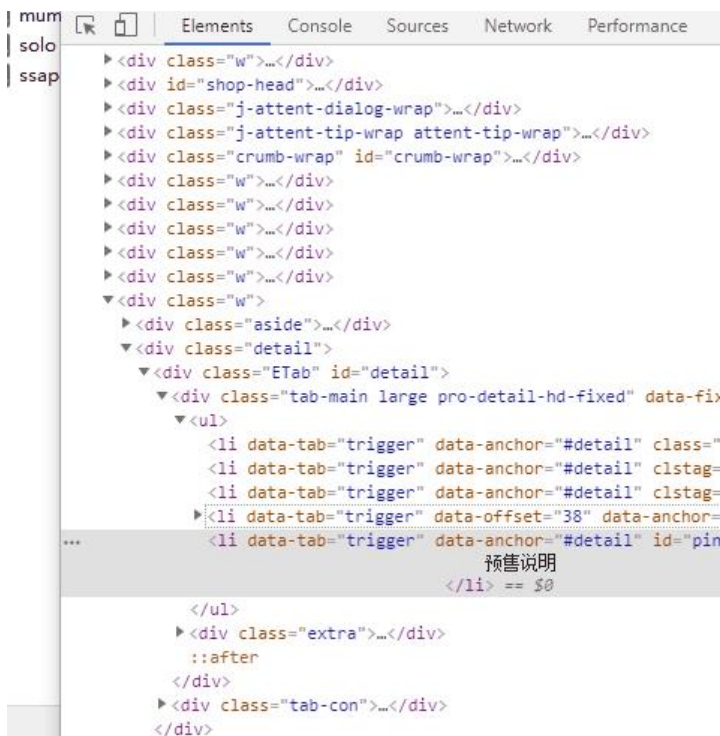
只因写了一段爬虫，公司200多人被抓!

还好，看来评价这东西也不算隐私。

过程中遇到几个问题：

- 元素偶尔获取不到，代码一样，不知道为什么；通常刷新一次后就可以了
- 点击 商品评价 标签后有时候获取不到评价信息，也不知道为什么；通常刷新一次后就好了
- 评价的记录数限制最大1000条。虽然显示有几万加的评价数，但是翻页最大就是100；网上搜了一，好像其它几个电商平台也是这样限制的。
- 有时候操作太快导致driver还没获取到数据，所以之后的操作会出错。添加time.sleep() 就是为了决电脑或网络卡顿问题（还有防止被封）
- 最烦的就是元素获取不到的问题，其实使用相对路径或绝对路径是最准确的，但是因为页面可能存变数，所以这种方法虽然准但却可能不通用。一开始使用相对路径额时候发现可以直接从Chrome浏览器上copy

gif图用的ScreenToGif软件做的，感觉挺小巧好用的（当时给PyCharm反馈bug，一开始google汉英描述了一大段，人还是不懂。。。还是一个gif图好用）



```

<div class="w">...</div>
<div id="shop-head">...</div>
<div class="j-attent-dialog-wrap">...</div>
<div class="j-attent-tip-wrap attent-tip-wrap">...</div>
<div class="crumb-wrap" id="crumb-wrap">...</div>
<div class="w">...</div>
<div class="w">...</div>
<div class="w">...</div>
<div class="w">...</div>
<div class="w">...</div>
<div class="w">
  <div class="aside">...</div>
  <div class="detail">
    <div class="ETab" id="detail">
      <div class="tab-main large pro-detail-hd-fixed" data-fi>
        <ul>
          <li data-tab="trigger" data-anchor="#detail" class=">
            <li data-tab="trigger" data-anchor="#detail" clstag=
            <li data-tab="trigger" data-anchor="#detail" clstag=
          <li data-tab="trigger" data-offset="38" data-anchor=
          <li data-tab="trigger" data-anchor="#detail" id="pin
            预售说明
          </li> == $0
        </ul>
      <div class="extra">...</div>
      ::after
    </div>
  <div class="tab-con">...</div>
</div>

```

获取评价代码

```

import time

import src.driver.DriverProducer as driverProducer
import xlwings as xw
from selenium import webdriver
from selenium.common.exceptions import ElementClickInterceptedException
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys
from src.autoclick import cookie_data

# 创建Chrome对象,使用本地下载的指定地址的驱动
# 有界面
# driver = driverProducer.on_windows()

# 无界面
# driver = driverProducer.off_windows()

# 创建Chrome对象, 使用安装的驱动, 需要配置环境变量
driver = webdriver.Chrome()

def main():
    # xlwings excel读写库
    # 新建excel
    app = xw.App(visible=True, add_book=False)
    file = app.books.add()
    sheet = file.sheets[0]

    # 评论页面
    evaluate = 'https://item.jd.com/100009083138.html#crumb-wrap'
    driver.get(evaluate)
    # 进入商品评价页面
    sppj()
    # 保存评价
    save_evaluate(sheet)
    # 保存excel文件
    file.save('e://data.xlsx')
    app.quit()

    # 使用完关闭浏览器, 不然Chromedriver.exe 进程会一直在内存中
    driver.quit()

def sppj():
    refresh = 0
    while True:
        try:
            # 获取并点击 商品评价
            # 找到li标签中, 有名为 clstag的属性且值为 "shangpin|keycount|product|shangpinpingjia
            # 1" 的元素
            driver.find_element_by_css_selector('li[clstag="shangpin|keycount|product|shangpinpi
            gjia_1"]').click()

```

```

time.sleep(1)
# 获取并点击 只看当前商品评价
# . 表示类选择器, 查找class= comm-curr-sku 的元素
driver.find_element_by_css_selector('.comm-curr-sku').click()
# 正常到这一步后跳出循环继续
break
except (NoSuchElementException, ElementClickInterceptedException):
    # 有时候第一次进页面点击 商品评价 时, 可能会报这个按钮是不可点击的, 或者点击后没有
    回的评价数据, 暂时不清楚这种情况产生的原因
    if refresh == 3:
        print('无法获取元素,或元素无法点击')
        return
    driver.refresh()
    refresh = refresh + 1

def page_turn():
    # 获取下一页按钮
    try:
        # 这里按钮没有用.click() 方法, 因为没有效果
        driver.find_element_by_css_selector(".ui-pager-next").send_keys(Keys.ENTER)
        return True
    except NoSuchElementException:
        print('没有下一页了。。。。。。')
        return False

def save_evaluate(sheet):
    # 获取评论用户信息
    row = 1
    while True:
        time.sleep(2)
        try:
            users = driver.find_elements_by_css_selector('.comment-item')
            if len(users) <= 0:
                print('没有评价了')
                break
            for user in users:
                user_info = user.find_element_by_css_selector('.user-info').text
                comment_con = user.find_element_by_css_selector('.comment-con').text
                print(user_info)
                print(comment_con)
                sheet.range('A%d' % row).value = user_info
                sheet.range('B%d' % row).value = comment_con
                row = row + 1
            if not page_turn():
                break
        except NoSuchElementException:
            print('获取评价时异常')
            break

if __name__ == '__main__':

```

main()

总结

[源码地址](#)

总的来说，selenium做自动化操作还是挺简单的。

这里还涉及到了对excel的操作，之后可以再深入研究下。

参考

[轻松自动化—selenium-webdriver\(python\)](#)

[自动化测试：盘点Selenium页面元素定位的8种方法](#)