

# SPIDER-DAY05

## 1. selenium爬虫

### 1.1 selenium概述

- 1 【1】定义
- 2     1.1) 开源的Web自动化测试工具
- 3
- 4 【2】用途
- 5     2.1) 对Web系统进行功能性测试, 版本迭代时避免重复劳动
- 6     2.2) 兼容性测试 (测试web程序在不同操作系统和不同浏览器中是否运行正常)
- 7     2.3) 对web系统进行大数量测试
- 8
- 9 【3】特点
- 10     3.1) 可根据指令操控浏览器
- 11     3.2) 只是工具, 必须与第三方浏览器结合使用
- 12
- 13 【4】安装
- 14     4.1) Linux: `sudo pip3 install selenium`
- 15     4.2) Windows: `python -m pip install selenium`

### 1.2 PhantomJS概述

- 1 | phantomjs为无界面浏览器(又称无头浏览器), 在内存中进行页面加载, 高效

### 1.3 环境安装

#### ■ 环境安装

- 1 【1】环境说明 ('以下三种环境任意安装其中一种即可')
- 2     环境一: selenium + PhantomJS
- 3     环境二: selenium + chromedriver + Chrome ('我们安装此版本')
- 4     环境三: selenium + geckodriver + Firefox
- 5
- 6 【2】提前下载驱动

```
7 2.1) chromedriver : 下载对应版本
8 http://npm.taobao.org/mirrors/chromedriver/
9 2.2) geckodriver
10 https://github.com/mozilla/geckodriver/releases
11 2.3) phantomjs
12 https://phantomjs.org/download.html
13
14 【3】 添加到系统环境变量
15 2.1) windows: 将解压后的可执行文件拷贝到Python安装目录的Scripts目录中
16 windows查看python安装目录(cmd命令行): where python
17 2.2) Linux : 将解压后的文件拷贝到/usr/bin目录中
18 sudo cp chromedriver /usr/bin/
19
20 【3】 Linux中需要修改权限
21 sudo chmod 777 /usr/bin/chromedriver
22
23 【4】 验证
24 4.1) Ubuntu | Windows
25 from selenium import webdriver
26 webdriver.Chrome()
27 webdriver.Firefox()
28
29 4.2) Mac
30 from selenium import webdriver
31 webdriver.Chrome(executable_path='/Users/xxx/chromedriver')
32 webdriver.Firefox(executable_path='/User/xxx/geckodriver')
```

## 2. selenium详解

### 2.1 代码演示

```
1 """示例代码一: 使用 selenium+浏览器 打开百度"""
2
3 # 导入selenium的webdriver接口
4 from selenium import webdriver
5 import time
6
7 # 创建浏览器对象
8 driver = webdriver.Chrome()
9 driver.get('http://www.baidu.com/')
10 # 5秒钟后关闭浏览器
11 time.sleep(5)
12 driver.quit()
```

```

1  """示例代码二：打开百度，搜索赵丽颖，点击搜索，查看"""
2
3  from selenium import webdriver
4  import time
5
6  # 1.创建浏览器对象 - 已经打开了浏览器
7  driver = webdriver.Chrome()
8  # 2.输入：http://www.baidu.com/
9  driver.get('http://www.baidu.com/')
10 # 3.找到搜索框,向这个节点发送文字：赵丽颖
11 driver.find_element_by_xpath('//*[@id="kw"]').send_keys('赵丽颖')
12 # 4.找到 百度一下 按钮,点击一下
13 driver.find_element_by_xpath('//*[@id="su"]').click()

```

## 2.2 浏览器对象方法

```

1  【1】 driver.get(url=url)    - 地址栏输入url地址并确认
2  【2】 driver.quit()         - 关闭浏览器
3  【3】 driver.close()        - 关闭当前页
4  【4】 driver.page_source     - HTML结构源码
5  【5】 driver.page_source.find('字符串')
6      从html源码中搜索指定字符串,没有找到返回：-1,经常用于判断是否为最后一页
7  【6】 driver.maximize_window() - 浏览器窗口最大化

```

## 2.3 定位节点

```

1  【1】 单元素查找('结果为1个节点对象')
2      1.1) 【最常用】 driver.find_element_by_id('id属性值')
3      1.2) 【最常用】 driver.find_element_by_name('name属性值')
4      1.3) 【最常用】 driver.find_element_by_class_name('class属性值')
5      1.4) 【最万能】 driver.find_element_by_xpath('xpath表达式')
6      1.5) 【匹配a节点时常用】 driver.find_element_by_link_text('链接文本')
7      1.6) 【匹配a节点时常用】 driver.find_element_by_partical_link_text('部分链接文本')
8      1.7) 【最没用】 driver.find_element_by_tag_name('标记名称')
9      1.8) 【较常用】 driver.find_element_by_css_selector('css表达式')
10
11 【2】 多元素查找('结果为[节点对象列表]')
12     2.1) driver.find_elements_by_id('id属性值')
13     2.2) driver.find_elements_by_name('name属性值')
14     2.3) driver.find_elements_by_class_name('class属性值')
15     2.4) driver.find_elements_by_xpath('xpath表达式')
16     2.5) driver.find_elements_by_link_text('链接文本')
17     2.6) driver.find_elements_by_partical_link_text('部分链接文本')
18     2.7) driver.find_elements_by_tag_name('标记名称')
19     2.8) driver.find_elements_by_css_selector('css表达式')

```

## 2.4 节点对象操作

- 1 【1】 node.send\_keys('') - 向文本框发送内容
- 2 【2】 node.click() - 点击
- 3 【3】 node.get\_attribute('属性名') - 获取节点的属性值
- 4 【4】 node.text - 获取当前节点及子节点和后代节点的文本内容

## 2.5 猫眼电影爬虫

```
1 """
2 抓取猫眼电影top100的 电影名称、主演、上映时间
3 URL地址: https://maoyan.com/board/4
4 """
5 from selenium import webdriver
6 import time
7
8 url = 'https://maoyan.com/board/4'
9 driver = webdriver.Chrome()
10 driver.get(url)
11
12 def get_data():
13     # 基准xpath: [<selenium xxx li at xxx>,<selenium xxx li at>]
14     li_list = driver.find_elements_by_xpath('//*[@id="app"]/div/div/div[1]/dl/dd')
15     for li in li_list:
16         item = {}
17         # info_list: ['1', '霸王别姬', '主演: 张国荣', '上映时间: 1993-01-01', '9.5']
18         info_list = li.text.split('\n')
19         item['number'] = info_list[0]
20         item['name'] = info_list[1]
21         item['star'] = info_list[2]
22         item['time'] = info_list[3]
23         item['score'] = info_list[4]
24
25         print(item)
26
27 while True:
28     get_data()
29     try:
30         driver.find_element_by_link_text('下一页').click()
31         time.sleep(2)
32     except Exception as e:
33         print('恭喜你!抓取结束')
34         driver.quit()
35         break
```

## 3. selenium高级

## 3.1 设置无界面模式

```
1 from selenium import webdriver
2
3 options = webdriver.ChromeOptions()
4 # 添加无界面参数
5 options.add_argument('--headless')
6 driver = webdriver.Chrome(options=options)
```

## 3.2 鼠标操作

```
1 """
2 鼠标操作三步走：
3 1、创建鼠标事件类对象
4 2、指定鼠标行为
5 3、执行
6 """
7 from selenium import webdriver
8 # 导入鼠标事件类
9 from selenium.webdriver import ActionChains
10
11 driver = webdriver.Chrome()
12 driver.get('http://www.baidu.com/')
13
14 # 移动到 设置，perform()是真正执行操作，必须有
15 element = driver.find_element_by_xpath('//*[@id="u1"]/a[8]')
16 ActionChains(driver).move_to_element(element).perform()
17
18 # 单击，弹出的Ajax元素，根据链接节点的文本内容查找
19 driver.find_element_by_link_text('高级搜索').click()
```

## 3.3 切换页面

### ■ 适用网站+应对方案

```
1 【1】适用网站类型
2 页面中点开链接出现新的窗口，但是浏览器对象driver还是之前页面的对象，需要切换到不同的窗口进行
  操作
3
4 【2】应对方案 - driver.switch_to.window()
5
6 # 获取当前所有句柄（窗口） - [handle1,handle2]
7 all_handles = driver.window_handles
8 # 切换browser到新的窗口，获取新窗口的对象
9 driver.switch_to.window(all_handles[1])
```

### ■ 民政部网站案例-selenium

```

1  """
2  适用selenium+Chrome抓取民政部行政区划代码
3  http://www.mca.gov.cn/article/sj/xzqh/2019/
4  """
5  from selenium import webdriver
6
7  class GovSpider(object):
8      def __init__(self):
9          # 设置无界面
10         options = webdriver.ChromeOptions()
11         options.add_argument('--headless')
12         # 添加参数
13         self.driver = webdriver.Chrome(options=options)
14         self.one_url = 'http://www.mca.gov.cn/article/sj/xzqh/2019/'
15
16     def get_incr_url(self):
17         self.driver.get(self.one_url)
18         # 提取最新链接节点对象并点击
19
20     self.driver.find_element_by_xpath('//td[@class="arlisttd"]/a[contains(@title,"代
21     码")]').click()
22     # 切换句柄
23     all_handlers = self.driver.window_handles
24     self.driver.switch_to.window(all_handlers[1])
25     self.get_data()
26
27     def get_data(self):
28         tr_list = self.driver.find_elements_by_xpath('//tr[@height="19"]')
29         for tr in tr_list:
30             code = tr.find_element_by_xpath('./td[2]').text.strip()
31             name = tr.find_element_by_xpath('./td[3]').text.strip()
32             print(name, code)
33
34     def run(self):
35         self.get_incr_url()
36         self.driver.quit()
37
38 if __name__ == '__main__':
39     spider = GovSpider()
40     spider.run()

```

## 3.4 切换frame

### ■ 特点+方法

- 1 【1】 特点
- 2 网页中嵌套了网页，先切换到iframe，然后再执行其他操作
- 3
- 4 【2】 处理步骤
- 5 2.1) 切换到要处理的Frame
- 6 2.2) 在Frame中定位页面元素并进行操作
- 7 2.3) 返回当前处理的Frame的上一级页面或主页面

```

8
9 【3】 常用方法
10 3.1) 切换到frame - driver.switch_to.frame(frame节点对象)
11 3.2) 返回上一级 - driver.switch_to.parent_frame()
12 3.3) 返回主页面 - driver.switch_to.default_content()
13
14 【4】 使用说明
15 4.1) 方法一：默认支持id和name属性值 : switch_to.frame(id属性值|name属性值)
16 4.2) 方法二：
17 a> 先找到frame节点 : frame_node = driver.find_element_by_xpath('xxx')
18 b> 在切换到frame : driver.switch_to.frame(frame_node)

```

## ■ 示例1 - 登录豆瓣网

```

1 """
2 登录豆瓣网
3 """
4 from selenium import webdriver
5 import time
6
7 # 打开豆瓣官网
8 driver = webdriver.Chrome()
9 driver.get('https://www.douban.com/')
10
11 # 切换到iframe子页面
12 login_frame = driver.find_element_by_xpath('//*[id="anony-reg-
13 new"]/div/div[1]/iframe')
14 driver.switch_to.frame(login_frame)
15
16 # 密码登录 + 用户名 + 密码 + 登录豆瓣
17 driver.find_element_by_xpath('/html/body/div[1]/div[1]/ul[1]/li[2]').click()
18 driver.find_element_by_xpath('//*[id="username"]').send_keys('自己的用户名')
19 driver.find_element_by_xpath('//*[id="password"]').send_keys('自己的密码')
20 driver.find_element_by_xpath('/html/body/div[1]/div[2]/div[1]/div[5]/a').click()
21 time.sleep(3)
22
23 # 点击我的豆瓣
24 driver.find_element_by_xpath('//*[id="db-nav-sns"]/div/div/div[3]/ul/li[2]/a').click()

```

## 3.5 selenium总结

### ■ selenium+phantomjs|chrome|firefox小总结

```

1 【1】 设置无界面模式
2 options = webdriver.ChromeOptions()
3 options.add_argument('--headless')
4 driver =
webdriver.Chrome(executable_path='/home/tarena/chromedriver',options=options)
5
6 【2】 browser执行JS脚本
7 driver.execute_script('window.scrollTo(0,document.body.scrollHeight)')
8

```

```
9  【3】 鼠标操作
10     from selenium.webdriver import ActionChains
11     ActionChains(driver).move_to_element('node').perform()
12
13  【4】 切换句柄 - switch_to.frame(handle)
14     all_handles = driver.window_handles
15     driver.switch_to.window(all_handles[1])
16
17  【5】 iframe子页面
18     driver.switch_to.frame(frame_node)
```

#### ■ lxml中的xpath 和 selenium中的xpath的区别

```
1  【1】 lxml中的xpath用法 - 推荐自己手写
2     div_list = p.xpath('//div[@class="abc"]/div')
3     item = {}
4     for div in div_list:
5         item['name'] = div.xpath('..//a/@href')[0]
6         item['likes'] = div.xpath('..//a/text()')[0]
7
8  【2】 selenium中的xpath用法 - 推荐copy - copy xpath
9     div_list = browser.find_elements_by_xpath('//div[@class="abc"]/div')
10    item = {}
11    for div in div_list:
12        item['name'] = div.find_element_by_xpath('..//a').get_attribute('href')
13        item['likes'] = div.find_element_by_xpath('..//a').text
```

## 4. 今日作业

- 1 【1】 使用selenium+浏览器 获取有道翻译结果
- 2 提示:1.text属性
- 3 2.注意time.sleep()
- 4 【2】 使用selenium+浏览器 登录网易163邮箱 : <https://mail.163.com/>