

## 5.4 Selenium 实现用户登录

### 任务:

Selenium 查找的 HTML 元素是一个 `WebElement` 对象，这个对象不但可以获取元素的属性值，而且还能执行一些键盘输入与鼠标点击的动作，我们设计一个实现用户登录并爬取数据的程序来演示这个过程。

### 5.4.1 创建用户登录网站

#### 1、创建网页模版

很多网站都需要用户登录后才能访问到其它网页，在 `templates` 中创建一个 `login.html` 的用户登录网页模版文件，在启动时显示登录界面，用户输入名称（假定是“xxx”）及密码（假定是“123”）后就完成登录，然后转去“/show”的页面，显示出手机的记录。`login.html` 内容如下：

```
<body>
<form id="frm" action="/" method="post">
<div>用户<input type="text" name="user"></div>
<div>密码<input type="password" name="pwd"></div>
<div><input type="submit" name="login" value='登录'></div>
</form>
</body>
```

#### 2、创建服务器程序

服务器程序首先提交一个 `login.html` 的网页，用户输入名称与密码后提交，服务器获取用户名称与密码后判断是否正确，如果不正确就继续显示登录页面，如果正确就转“/show”页面显示手机记录，服务器程序如下：

```
import flask
app=flask.Flask(__name__)

@app.route("/",methods=["GET","POST"])
def login():
    user=flask.request.values.get("user") if "user" in flask.request.values else ""
    pwd=flask.request.values.get("pwd") if "pwd" in flask.request.values else ""
    if user=="xxx" and pwd=="123":
        return flask.redirect("/show")
    else:
        return flask.render_template("login.html")

@app.route("/show",methods=["GET","POST"])
def show():
    s="<table border='1'>"
    s=s+"<tr><td>品牌</td><td>型号</td><td>价格</td></tr>"
    s=s+"<tr><td>华为</td><td>P9</td><td>3800</td></tr>"
    s=s+"<tr><td>华为</td><td>P10</td><td>4200</td></tr>"
```

```
s=s+"<tr><td>苹果</td><td>iPhone6</td><td>5800</td></tr>"
s=s+"</table><p>"
s=s+"<a href='/>退出</a>"
return s
```

app.run()

其中语句 `return flask.redirect("/show")` 是转 `/show` 网页。

### 3、浏览器浏览

启动服务器程序后浏览 web 地址 `"http://127.0.0.1:5000"`, 结果如图 5-4-1 与 5-4-2 所示, 用户登录成功后就可以看到手机的记录。

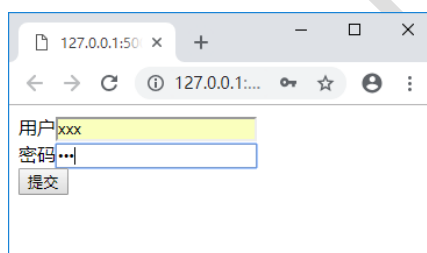


图 5-4-1 用户登录

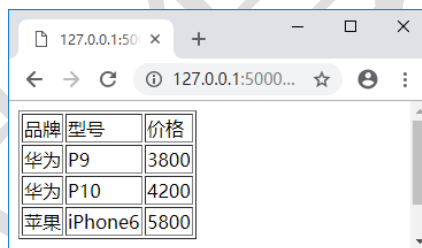


图 5-4-2 登录成功

显然如果要爬取这个 Web 网站的所有手机记录, 就必须模拟用户登录过程, 使得访问到 `/show` 的页面。

#### 5.4.2 Selenium 使用元素动作

Selenium 查找 HTML 时返回一个 `WebElement` 对象, 这个对象的功能十分强大, 不但可以使用它获取元素的属性值, 而且它还有动作, 主要的动作是键盘输入动作与鼠标点击动作。

##### 1、键盘输入动作

有些元素例如 `<input type="text">` 文本输入框是用户可以输入文字的, `WebElement` 对象 `element` 可以模拟用户的键盘输入动作, 主要动作有:

- (1) 函数 `clear()` 模拟清除 `element` 元素中的所有文字;
- (2) 函数 `send_keys(string)` 模拟键盘在元素中输入字符串 `string`;

其中 `send_keys` 函数不但可以模拟输入一般的文字, 而且还可以模拟输入回车、退格等键盘动作, selenium 提供了一个 `Keys` 类, 其中提供了很多常用的不可见的特殊按键, 主要有 `Keys.BACKSPACE` 退格删除键和 `Keys.ENTER` 回车键。

例如编写下面的程序:

```
from selenium import webdriver
try:
```

```

driver = webdriver.Chrome()
driver.get("http://127.0.0.1:5000")
user = driver.find_element_by_name("user")
pwd = driver.find_element_by_name("pwd")
time.sleep(0.5)
user.send_keys("xxx")
time.sleep(0.5)
pwd.send_keys("123")
except Exception as err:
    print(err)
input("Strike any key to finish...")
driver.close()

```

这个程序使用 `driver = webdriver.Chrome()` 创建浏览器，因此浏览器是可见的，运行服务器后执行该程序，可以清楚地看见在用户输入框与密码框的输入动作，如图 5-4-3 所示。

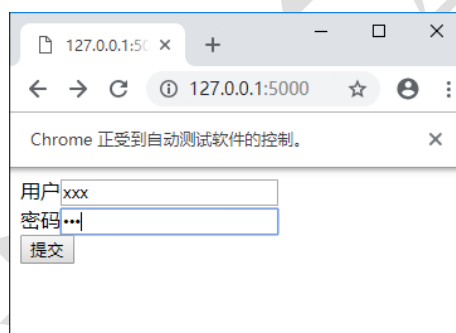


图 5-4-3 实现键盘输入

## 2、鼠标点击动作

很多 HTML 元素都有鼠标点击动作，例如 `<input type="submit">` 提交按钮点击后就提交表单。WebElement 使用 `click()` 函数实现鼠标点击，例如：

```
driver.find_element_by_xpath("input[@type='submit']").click()
```

### 5.4.3 编写爬虫程序

我们使用 Selenium 模拟浏览器的浏览过程与用户的登录过程，然后爬取手机的记录，程序过程如下：

- (1) 创建一个浏览器对象 `driver`，使用这个 `driver` 对象模拟浏览器。
- (2) 访问 `http://127.0.0.1:5000` 网站，获取 `<input type="text" name="user">` 与 `<input type="password" name="pwd">` 元素对象，模拟用户键盘输入名称 "xxx" 与密码 "123"。
- (3) 获取 `<input type="submit" name="login">` 按钮对象，执行 `click()` 点击动作，提交表单。
- (4) 服务器接收提交的 `user` 与 `pwd` 数据，判断是否登录成功，如果登录成功就转 `/show` 页面显示手机记录。
- (5) 爬虫程序爬取数据记录。

爬虫程序如下：

```

from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import time

```

---

```

def login():
    print(driver.current_url)
    user = driver.find_element_by_name("user")
    pwd = driver.find_element_by_name("pwd")
    login = driver.find_element_by_name("login")
    user.send_keys("xxx")
    pwd.send_keys("123")
    login.click()
    time.sleep(0.5)

def spider():
    print(driver.current_url)
    trs=driver.find_elements_by_tag_name("tr")
    for i in range(1,len(trs)):
        tds=trs[i].find_elements_by_tag_name("td")
        if len(tds)==3:
            mark = tds[0].text
            model=tds[1].text
            price=tds[2].text
            print("%-16s%-16s%-16s" %(mark,model,price))

chrome_options = Options()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--disable-gpu')
driver = webdriver.Chrome(chrome_options=chrome_options)
try:
    driver.get("http://127.0.0.1:5000")
    login()
    spider()
except Exception as err:
    print(err)
driver.close()

```

程序先访问 "http://127.0.0.1:5000", 执行 login 登录函数, user.send\_keys("xxx") 与 pwd.send\_keys("123")是向 user 与 pwd 元素发送文本字符串"xxx"与"123", 这个过程等效于用户从键盘输入一样。

login.click()是 login 按钮点击的动作, 执行该命令即完成提交, 提交后网页转去 "http://127.0.0.1:5000/show", 使用 time.sleep(0.5)停顿一会, 开始执行爬虫函数 spider 爬取网页数据。

下面的执行结果明程序成功模拟了用户的登录, 并爬取到了只有登录成功后才能浏览到的数据:

```

http://127.0.0.1:5000/
http://127.0.0.1:5000/show

```

---

华为	P9	3800
华为	P10	4200
苹果	iPhone6	5800

亚森自研