5.6 Selenium 等待 HTML 元素

任务：

在浏览器加载网页的过程中，网页的有些元素时常会有延迟的现象，在 HTML 元素还没有准备好的情况下去操作这个 HTML 元素必然会出现错误，这个时候 Selenium 需要等待 HTML 元素，我们来讨论如果使用 Selenium 等待延迟的 HTML 元素并最终爬取元素的数据。

5.6.1 创建延迟模拟网站

1、创建网页模版

在 templates 中创建一个 phone.html，这个文件使用 Ajax 从服务器获取手机的品牌放在一个<select>中，注意<select>中的<option>开始是不存在的，只有获取数据后才产生，模版文件如下：

```
<script>
    function loadMarks()
    {
        var http=new XMLHttpRequest();
        http.onreadystatechange=function()
        {
          if (http.readyState==4 && http.status==200)
          {
              var xmark=document.getElementById("xmark");
              var xcolor=document.getElementById("xcolor");
              marks=eval("("+http.responseText+")");
              for(var i=0;i<marks.length;i++)
              xmark.options.add(new Option(marks[i],marks[i]));
              document.getElementById("submit").disabled=false;
              document.getElementById("msg").innerHTML="品牌";
          }
        };
        http.open("get","/marks",true);
        http.send(null);
    }
    loadMarks();
</script>
<body>
<form name="frm" action="/">
  <div><span id="msg"></span><select id="xmark" ></select></div>
<input type="submit" value="提交" id="submit" disabled="true">
</form>
</body>
```

2、创建网站服务器

网站服务器在访问地址"/"时首先提交 phone.html 网页，然后网页中根据 JavaScript 代码

会执行 loadPhones 函数，再次访问服务器"/phones"时发送手机的品牌 marks 与颜色 colors 数据，数据按 JSON 字符串格式发送。为了模拟延迟过程使用 time.sleep(1)延迟 1 秒后发送数据，程序如下：

```
import flask
import json
import time
app=flask.Flask(__name__)
@app.route("/")
def index():
    return flask.render_template("phone.html")
@app.route("/marks")
def loadMarks():
    time.sleep(1)
    marks=["华为","苹果","三星"]
    return json.dumps(marks)
app.run()
```

运行服务器并使用浏览器浏览结果，延迟一会后出现如图 5-8-1 所示的网页界面。

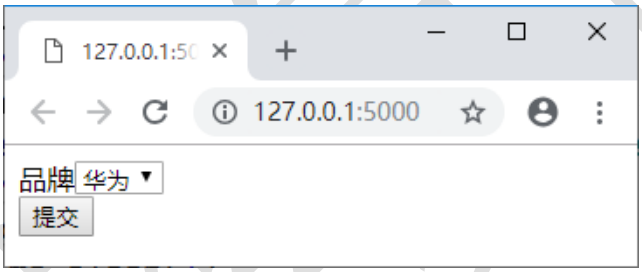

图 5-6-1 延迟模拟网站

5.6.2 编写爬虫程序

我们编写一个爬虫程序去爬取手机的所有品牌与所有颜色，并选择其中的一个品牌与颜色进行提交。

这个爬虫程序编写如下：

```
from selenium import webdriver
import time
driver = webdriver.Chrome()
driver.get("http://127.0.0.1:5000")
marks=driver.find_elements_by_xpath("//select/option")
print("品牌数量:",len(marks))
for mark in marks:
    print(mark.text)
form=driver.find_element_by_xpath("//form")
print(form.get_attribute("innerHTML").strip())
time.sleep(5)
driver.close()
```

执行这个程序结果：

```
品牌数量: 0
```

品牌<select id="xmark"></select>
<input type="submit" value="提交" id="submit" disabled="true">

由此可见这个爬虫程序没有爬取到手机品牌与颜色的数据，原因是服务器有延迟，这些数据还没有在网页中生成。

5.6.3 Selenium 强制等待

Selenium 使用 time.sleep(seconds)来实现强制等待 seconds 秒，这种方式是最简单粗暴的方式，不管当前操作是否完成，是否可以进行下一步操作，都必须等 seconds 秒的时间。缺点是不能准确把握需要等待的时间（有时操作还未完成，等待就结束了，导致报错；有时操作已经完成了，但等待时间还没有到，浪费时间），如果在用例中大量使用，会浪费不必要的等待时间，影响测试用例的执行效率。例如爬虫程序在加载网页后强制等待 1.5 秒：

```
from selenium import webdriver
import time
driver = webdriver.Chrome()
driver.get("http://127.0.0.1:5000")
#设置强制等待 1.5 秒
time.sleep(1.5)
marks=driver.find_elements_by_xpath("//select/option")
print("品牌数量:",len(marks))
for mark in marks:
    print(mark.text)
form=driver.find_element_by_xpath("//form")
print(form.get_attribute("innerHTML").strip())
time.sleep(5)
driver.close()
```

执行结果：

品牌数量: 3
华为
苹果
三星
品牌<select id="xmark"><option value="华为">华为</option><option value="苹果">苹果</option><option value="三星">三星</option></select>
<input type="submit" value="提交" id="submit">

由此可见在经过等待 1.5 秒后程序从服务器获取了手机品牌数据并创建了<select>中的各个<option>元素，因此程序爬取到了手机的品牌数据。但是如果设置的强制等待时间不够长，还是爬取不到需要的数据。

5.6.4 Selenium 隐性等待

Selenium 使用 implicitly_wait(seconds)设置隐性等待指定的秒数，即网页在加载时最长等待 seconds 秒，例如爬虫程序在访问网页设置隐性加载时间为 1.5 秒：

```
from selenium import webdriver
import time
driver = webdriver.Chrome()
#设置隐性加载时间 1.5 秒
```

```
driver.implicitly_wait(1.5)
driver.get("http://127.0.0.1:5000")
marks=driver.find_elements_by_xpath("//select/option")
print("品牌数量:",len(marks))
for mark in marks:
    print(mark.text)
form=driver.find_element_by_xpath("//form")
print(form.get_attribute("innerHTML").strip())
time.sleep(5)
driver.close()
```

执行的结果：

```
品牌数量: 3
华为
苹果
三星
<div>品牌<select id="xmark"><option value="华为">华为</option><option value="苹果">苹果</option><option value="三星">三星</option></select></div>
<input type="submit" value="提交" id="submit">
```

由此可见在经过等待后程序从服务器获取了手机品牌数据并创建了<select>中的各个<option>元素，因此程序爬取到了手机的品牌数据。同样如果设置的隐性等待时间不够长，还是爬取不到需要的数据。

5.6.5 Selenium 显示等待

1、循环等待

实际上这个爬虫程序能否爬到数据的关键是<select>中是否已经出现了<option>元素，我们可以设置一个循环来判断是否有<option>元素，程序修改如下：

```
from selenium import webdriver
import time
driver = webdriver.Chrome()
try:
    driver.get("http://127.0.0.1:5000")
    waitTime=0
    while waitTime<10:
        marks = driver.find_elements_by_xpath("//select/option")
        if len(marks)>0:
            break
        time.sleep(0.5)
        waitTime+=0.5
    if waitTime>=10:
        raise Exception("Waiting time out")
    marks=driver.find_elements_by_xpath("//select/option")
    print("品牌数量:",len(marks))
    for mark in marks:
        print(mark.text)
```

```
        form=driver.find_element_by_xpath("//form")
        print(form.get_attribute("innerHTML").strip())
    except Exception as err:
        print(err)
    time.sleep(5)
    driver.close()
```

这个程序中使用 waitTime 变量来构造一个循环，它最长等待 10 秒，每间隔 0.5 秒就检查一次<select>中是否有<option>存在，如果找到了<option>元素就退出等待循环，不然就继续等待直到<option>出现为止，如果 10 秒内还没有出现据抛出异常。

这个爬虫程序的结果：

品牌数量: 3

华为

苹果

三星

品牌<select id="xmark"><option value="华为">华为</option><option value="苹果">苹果</option><option value="三星">三星</option></select>

<input type="submit" value="提交" id="submit">

2、显示等待

Selenium 的显示等待与循环等待有点类似，它是专门等待指定的元素的。Selenium 使用 WebDriverWait 类来实现显示等待，在使用显示等待之前先引入 WebDriverWait、EC 以及 By 等类：

```
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
```

然后构造一个定位元素的 locator 的对象，例如通过 XPath 的方法定位<select>中的<option>元素：

```
locator=(By.XPATH,"//select/option")
```

最后使用 WebDriverWait 构造一个实例，调用 until 方法：

```
WebDriverWait(driver,10, 0.5).until(EC.presence_of_element_located(locator))
```

这条语句的含义是等待 locator 指定的元素出现，最长等待 10 秒，每间隔 0.5 秒就出现检查一次。如果在 10 秒内出现了该元素就是结束等待，否则就继续等待。如果超过 5 秒还没有等待到 locator 要求的元素就抛出一个异常。例如使用显示等待的爬程序如下：

```
from selenium import webdriver
import time
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
try:
    driver.get("http://127.0.0.1:5000")
    locator = (By.XPATH, "//select/option")
    WebDriverWait(driver, 10,0.5).until(EC.presence_of_element_located(locator))
    marks=driver.find_elements_by_xpath("//select/option")
```

```
        print("品牌数量:",len(marks))
        for mark in marks:
            print(mark.text)
        form=driver.find_element_by_xpath("//form")
        print(form.get_attribute("innerHTML").strip())
    except Exception as err:
        print(err)
    time.sleep(5)
    driver.close()
```

执行的结果：

品牌数量: 3

华为

苹果

三星

品牌<select id="xmark"><option value="华为">华为</option><option value="苹果">苹果</option><option value="三星">三星</option></select>

<input type="submit" value="提交" id="submit">

显然程序等待到了<option>元素的出现，爬取到了手机品牌数据。这种等待的优点就是等待判断准确，不会浪费多余的等待时间，在实际中使用可以提高执行效率。

5.6.5 Selenium 显示等待形式

显示等待有很多种形式，读者可以查看 Selenium 的文档说明，下面是一些常用的形式：

1、EC.presence_of_element_located(locator)

这种形式是等待 locator 指定的元素出现，也就是 HTML 文档中建立起了这个元素。

2、EC.visibility_of_element_located(locator)

这种形式是等待 locator 指定的元素可见，注意元素出现时未见得可见，例如：

<select id="xmark" style="display:none">...</select>

那么元素<select>是出现的但是不可见。

3、EC.element_to_be_clickable(locator)

这种形式是等待 locator 指定的元素可以被点击，例如在爬虫程序中等待<input type="submit">按钮可用被点击：

```
locator = (By.XPATH, "//input[@type='submit']")
WebDriverWait(driver, 10,0.5).until(EC.element_to_be_clickable(locator))
```

或者等待<option>是否可以被点击：

```
locator = (By.XPATH, "//select/option")
WebDriverWait(driver, 10,0.5).until(EC.element_to_be_clickable(locator))
```

使用这两种方法都可以爬取到手机品牌数据。

但是注意使用：

```
locator = (By.XPATH, "//select")
WebDriverWait(driver, 10,0.5).until(EC.element_to_be_clickable(locator))
```

是等待<select>是否可以点击，这个元素就是没有<option>时也是可以点击的，因此用这个等待是爬取不到手机的品牌数据的。

4、EC.element_located_to_be_selected(locator)

这种形式是等待 locator 指定的元素可以被选择，可以被选择的元素一般是<select>中的<option>、<input type="checkbox">以及<input type="radio">等元素。例如爬虫程序中使用下列的等待：

```
locator = (By.XPATH, "//select/option")
WebDriverWait(driver, 10,0.5).until(EC.element_located_to_be_selected(locator))
```

同样能爬取到手机的品牌数据。

但是使用下列是不行的：

```
locator = (By.XPATH, "//input[@type='submit']")
WebDriverWait(driver, 10,0.5).until(EC.element_located_to_be_selected(locator))
```

因为这样的<input type='submit'>是怎么样也不可以选择的。

5、EC.text_to_be_present_in_element(locator,text)

这种形式是等待 locator 指定的元素的文本中包含指定的 text 文本，例如爬虫程序中使用下列的等待：

```
locator = (By.ID, "msg")
WebDriverWait(driver, 10,0.5).until(EC.text_to_be_present_in_element(locator,"品"))
```

即等待.....元素中的文本包含"品"字，由于在<option>出现后设置文本是"品牌"，因此爬虫程序可以爬取到手机品牌数据。