Lab 4: Web Applications Testing

Software Testing 2022

2022/03/24

Selenium

About selenium

Selenium automates browsers.

Primarily used for autimating web applications for testing purpose, but is certainly not limited to just that.

Boring web-based administration tasks can also be automated as well.

Selenium webdriver

WebDriver uses browser automation APIs provided by browser vendors to control browser and run tests, as if a real user is operating the browser.

Support multiple browsers and langauges.

Download selenium library with pip

```
sudo apt install python3-pip

pip3 install --upgrade selenium

pip3 install webdriver_manager

pip3 install --upgrade requests
```

Auto install and use webdriver

```python
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))

driver.get("https://google.com")
```

https://pypi.org/project/webdriver-manager/

Example

Browser navigation

Navigate to:

```
driver.get("https://google.com")
```

Refresh the current page:

```
driver.refresh()
```

Quitting the browser at the end of a session:

```
driver.quit()
```

Pressing the browser’s forward, back button:

```
# Pressing the browser's back button
driver.back()
# Pressing the browser's forward button
driver.forward()
```

Windows and Tabs

Get window handle:

```
In [50]: driver.window_handles
Out[50]:
['CDwindow-40DB5FF2B7E673E82B03E1C52E09A5FB',
 'CDwindow-C5BABC52E81C69C12E3EA3B3872CF102',
 'CDwindow-E45204BB0C2F54FA4D87A260E2F00AFA',
 'CDwindow-F44134887F43D2C742D967CAB54B60BE',
 'CDwindow-8C945448D03658C90163E0907934B3DB']
```

Switching windows or tabs:

```
# Store the ID of the original window
original_window = driver.current_window_handle

# Check we don't have other windows open already
assert len(driver.window_handles) == 1

# Click the link which opens in a new window
driver.find_element(By.LINK_TEXT, "new window").click()

# Wait for the new window or tab
wait.until(EC.number_of_windows_to_be(2))

# Loop through until we find a new window handle
for window_handle in driver.window_handles:
    if window_handle != original_window:
        driver.switch_to.window(window_handle)
        break

# Wait for the new tab to finish loading content
wait.until(EC.title_is("SeleniumHQ Browser Automation"))
```

Create new window (or) new tab and switch:

```
# Opens new tab and switch to new tab
driver.switch_to.new_window('tab')

# Opens new window and switch to new window
driver.switch_to.new_window('window')
```

Closing a window or tab:

```
    #Close the tab or window
driver.close()


    #Switch back to the old tab or window
driver.switch_to.window(original_window)
```

Find element

Locating one element:

```
cat = driver.find_element(By.ID,"meow")
```

Narrow the scope of element:

```
cat = driver.find_element(By.ID,"meow")

cat_0_0 = cat.find_element(By.ID,"cat1")
```

Locating multiple elements:

```
In [2]: cats = driver.find_elements(By.CSS_SELECTOR,"#meow li")

In [3]: cats
Out[3]:
[<selenium.webdriver.remote.webelement.WebElement (session="f7d4ff76f061659
69d5975433834aba4", element="a131beff-cbf5-4cf1-8a6c-7fa3a773a80a")>,
 <selenium.webdriver.remote.webelement.WebElement (session="f7d4ff76f061659
69d5975433834aba4", element="33db5768-a0f8-4629-818f-d3aada3fc68a")>,
 <selenium.webdriver.remote.webelement.WebElement (session="f7d4ff76f061659
69d5975433834aba4", element="f777b6a5-39c4-41d5-81fe-d33a7452d0fc")>,
 <selenium.webdriver.remote.webelement.WebElement (session="f7d4ff76f061659
69d5975433834aba4", element="1750e0c0-71d5-4d08-9689-306d59062705")>]
```

```
<ol id=meow>
 <li id=cat1>…
 <li id=cat2>…
 <li id=cat3>…
 <li id=cat4>…
</ol>
```

Element selection strategies

Locator	Description
class name	Locates elements whose class name contains the search value (compound class names are not permitted)
css selector	Locates elements matching a CSS selector
id	Locates elements whose ID attribute matches the search value
name	Locates elements whose NAME attribute matches the search value
link text	Locates anchor elements whose visible text matches the search value
partial link text	Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
tag name	Locates elements whose tag name matches the search value
xpath	Locates elements matching an XPath expression

https://www.selenium.dev/selenium/docs/api/py/webdriver/selenium.webdriver.common.by.html

Waits

```
<!doctype html>
<meta charset=utf-8>
<title>Race Condition Example</title>

<script>
    var initialised = false;
    window.addEventListener("load", function () {
        setTimeout(function () {
            var newElement = document.createElement("p");
            newElement.textContent = "Hello from JavaScript!";
            document.body.appendChild(newElement);
            initialised = true;
        }, 3000);
    });
</script>
```

Waits

```
driver.navigate("file:///race_condition.html")
el = driver.find_element(By.TAG_NAME, "p")
assert el.text == "Hello from JavaScript!"
```

```
selenium.common.exceptions.NoSuchElementException: Message: no such elemen
t: Unable to locate element: {"method":"css selector","selector":"p"}
```

Waits

Explicit wait:

```
driver.get("file:///home/a13579and2468/software_testing_2022/race_condition.html")
el = WebDriverWait(driver,10).until(lambda d: d.find_element(By.TAG_NAME,"p"))
assert el.text == "Hello from JavaScript!"
```

Expected condition:

```
from selenium.webdriver.support import expected_conditions as EC

driver.get("file:///home/a13579and2468/software_testing_2022/race_condition.html")
el = WebDriverWait(driver,10).until(EC.element_to_be_clickable((By.TAG_NAME,'p')))
assert el.text == "Hello from JavaScript!"
```

Implicit wait:

```
driver.get("file:///home/a13579and2468/software_testing_2022/race_condition.html")
driver.implicitly_wait(50)
el = driver.find_element(By.TAG_NAME,"p")
assert el.text == "Hello from JavaScript!"
```


https://www.selenium.dev/documentation/webdriver/waits/

Others

text :

click() :

send_keys() :

Other information can be found in document.

- https://www.selenium.dev/documentation/en/webdriver/

Lab

Lab 4

1. Install python3 and selenium
2. Write a scenario for following requirements:
   a. launch browser and navigate to NYCU home page(https://www.nycu.edu.tw/) ➜ maximize the window ➜ click 消息 ➜ click first news ➜ print the title and content
   b. open a new tab and switch to it ➜ navigate to google(https://www.google.com) ➜ input your student number and submit ➜ print the title of second result ➜ close the browser
3. Upload student_***YourID***.py to E3

Reference

Reference

https://www.selenium.dev/

https://pypi.org/project/webdriver-manager/

https://www.selenium.dev/selenium/docs/api/py/webdriver/selenium.webdriver.common.by.html