Lab 5: Web Applications Testing

Software Testing 2021

2021/04/08

Selenium

About selenium

Selenium automates browsers.

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

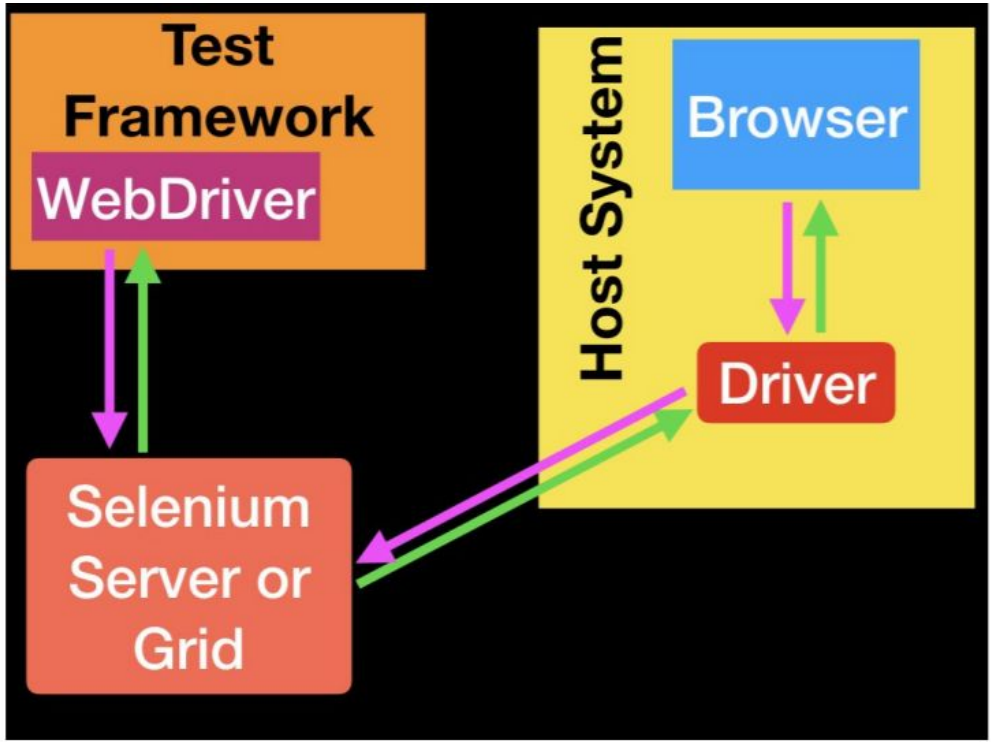
Boring web-based administration tasks can also be automated as well.

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.

Selenium webdriver



Download selenium library

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for **other languages exist**, these are the core ones that are supported by the main project hosted on GitHub.

LANGUAGE	STABLE VERSION	RELEASE DATE	BETA VERSION	BETA RELEASE DATE	LINKS
Ruby	3.142.6	October 04, 2019	4.0.0beta2	March 16, 2021	Download Beta Download Changelog API Docs
Java	3.141.59	November 14, 2018	4.0.0-beta-2	March 16, 2021	Download Beta Download Changelog API Docs
Python	3.141.0	November 01, 2018	4.0.0.b2	March 16, 2021	Download Beta Download Changelog API Docs
C#	3.14.0	August 02, 2018	4.0.0-beta2	March 17, 2021	Download Beta Download Changelog API Docs
JavaScript	3.6.0	October 06, 2017	4.0.0-beta.2	March 16, 2021	Download Beta Download Changelog API Docs

- https://www.selenium.dev/downloads/

Download webdriver

- check your browser version
- driver version should be same as browser version

Browser	Supported OS	Maintained by	Download	Issue Tracker
Chromium/Chrome	Windows/macOS/Linux	Google	Downloads	Issues
Firefox	Windows/macOS/Linux	Mozilla	Downloads	Issues
Edge	Windows 10	Microsoft	Downloads	Issues
Internet Explorer	Windows	Selenium Project	Downloads	Issues
Safari	macOS El Capitan and newer	Apple	Built in	Issues
Opera	Windows/macOS/Linux	Opera	Downloads	Issues

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

Add webdriver to PATH

- add webdriver directory to PATH

or

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
...
WebDriver driver = new ChromeDriver();
System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");
```

Example

Browser navigation

Navigate to:

```
//Convenient
driver.get("https://selenium.dev");

//Longer way
driver.navigate().to("https://selenium.dev");
```

Pressing the browser's forward, back button:

```
//Pressing the browser's back button:
driver.navigate().back();
//Pressing the browser's forward button:
driver.navigate().forward();
```

Refresh the current page:

```
driver.navigate().refresh();
```

Quitting the browser at the end of a session:

```
driver.quit();
```

Windows and Tabs

Get window handle:

```
driver.getWindowHandle();
```

Switching windows or tabs:

```
//Store the ID of the original window
String originalWindow = driver.getWindowHandle();
...
//Click the link which opens in a new window
driver.findElement(By.linkText("new window")).click();
...
//Loop through until we find a new window handle
for (String windowHandle : driver.getWindowHandles()) {
    if(!originalWindow.contentEquals(windowHandle)) {
        driver.switchTo().window(windowHandle);
        break;
    }
}
```

Create new window (or) new tab and switch:

```
// Opens a new tab and switches to new tab
driver.switchTo().newWindow(WindowType.TAB);

// Opens a new window and switches to new window
driver.switchTo().newWindow(WindowType.WINDOW);
```

Closing a window or tab:

```
//Close the tab or window
driver.close();

//Switch back to the old tab or window
driver.switchTo().window(originalWindow);
```

Find element

Locating one element:

```
WebElement cat = driver.findElement(By.id("meow"));
```

Narrow the scope of element:

```
WebElement cat = driver.findElement(By.id("meow"));
WebElement cat_0_0 = cat.findElement(By.id("meow_meow"));
```

Locating multiple elements:

```
List<WebElement> cats = driver.findElements(By.cssSelector("#meow li"));
```

```
<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

Waits

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

<script>
  var initialised = false;
  window.addEventListener("load", function() {
    var newElement = document.createElement("cat");
    newElement.textContent = "meow";
    document.body.appendChild(newElement);
    initialised = true;
  });
</script>
```

Waits

```
driver.get("file:///race_condition.html");
WebElement element = driver.findElement(By.tagName("cat"));
assertEquals(element.getText(), "meow");
```

Waits

Explicit wait:

```
//wait until timeout or condition meet
WebElement foo = new WebDriverWait(driver, Duration.ofSeconds(3))
        .until(driver -> driver.findElement(By.name("cat")));
assertEquals(foo.getText(), "meow");
```

Expected condition:

```
new WebDriverWait(driver, Duration.ofSeconds(3))
.until(ExpectedConditions.elementToBeClickable(By.xpath("//a/h3")));
```

Implicit wait:

```
//wait until time out
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

Others

getText():

click():

sendkey():

Other information can be found in document.

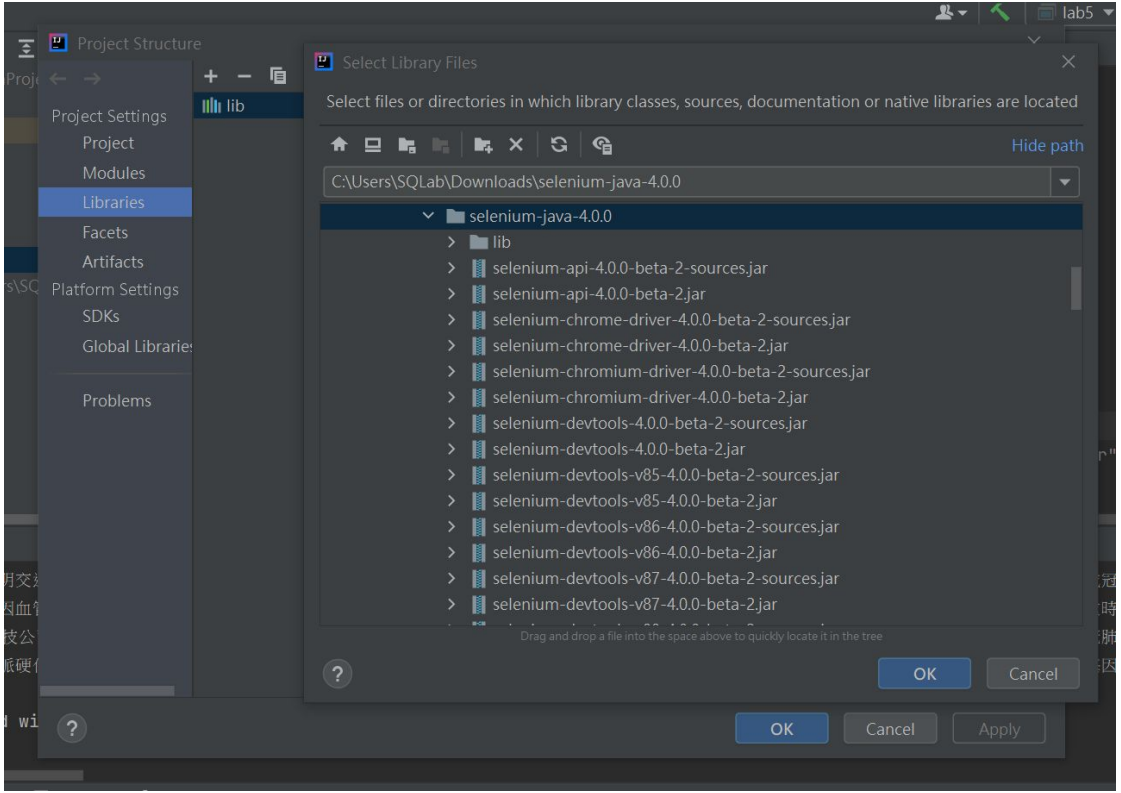
- https://www.selenium.dev/documentation/en/webdriver/

Lab

Lab 5

1. Download selenium-java library from official website, and download the corresponding driver for your browser.
   a. https://www.selenium.dev/downloads/
   b. https://www.selenium.dev/documentation/en/webdriver/driver_requirements/
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 NEWS ➜ 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***.java to E3

Import selenium-java



Reference

Reference

https://www.selenium.dev/