Selenium WebDriver Basic

A powerful framework for UI Automation Testing

Selenium WebDriver Introduction

What is Selenium WebDriver?

It is a powerful framework (actually the first choice) for UI Automation Testing on webpage.

Why do we choose Selenium WebDriver?

(1) Open Source, it is free!
(2) Multiple Languages support, such as Java, Python, Ruby, C#...
(3) Multiple Operating System support, Windows, MacOs, Linux
(4) Provide a Unified Interface for Multiple Browser: Chrome, Firefox, Edge, etc. (Polymorphism!)
(5) Support parallel testing
(6) Integration with other frameworks: TestNG, Cucumber, JUnit etc.

Limitations

(1) No support for window based applications

Environment Setup

- **Selenium-java**
  - Selenium webdriver API written in java
- **Webdrivermanager**
  - Download the driver from the internet
- **TestNG**
  - Serves as the backbone of the entire testing framework

```
<dependencies>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>3.141.59</version>
    </dependency>
    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>4.0.0</version>
    </dependency>
    <dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>7.4.0</version>
    </dependency>
</dependencies>
```

Webdriver

Browser Commands	Conditional Commands	Wait commands
• close() • quit()	• isDisplayed() • isEnabled() • isSelected()	• Implicitwait • Explicitwait
Get commands	**Navigation Commands**	**Switch commands**
• get() • getTitle() • getPageSource() • getCurrentUrl() • getText()	• navigate().forward() • navigate().back() • navigate().to() • navigate().refresh()	• switchTo().frame() • switchTo().alert() • switchTo().defaultContent() • switchTo().window() • Driver.WindowHandles()

Open Webpage

In Selenium webdriver, there are two options to open a link:

- driver.navigate().to(url)
- driver.get(url)

What's the difference between these two methods?

- No difference, choose one based on your own preference
- But Navigation Class provides more options, for example forward(), back(), refresh()

Element Locators (Very Important!)

There are so many elements in the webpage (sometimes thousands), how to find the exact element we want?

Using Locators!

- Id
- Name
- LinkText
- CSS
- XPath
  - Absolute XPath
  - Relative XPath

XPath

What is XPath?

XPath is an expression language designed to support the query XML documents.

- **Absolute XPath:**
  - Starts with “/”, syntax: /html/body/div
  - It always starts with the root of the entire document.
- **Relative XPath:**
  - Starts with “//”, syntax: //tagname[@attribute='value']]/ . . .
  - It can start with any node

Waits

Background:

Sometimes, elements in the web page take a long load due to the network, or there are some dynamic elements in the webpage (controlled by javascript), in that case, we need to wait for the elements to meet our expectations.

Types of Wait:

- Implicit Wait
  - Specify the maximum time that the webdriver will wait until it throws NoSuchElementException
- Explicit Wait
  - Fluent Wait
  - WebDriverWait (A subclass of FluentWait)

Basic Web Elements

- ✓ Textbox
- ✓ Web Links
- ✓ Button
- ✓ Dropdown
- ✓ Radio button
- ✓ Check box
- ✓ Frame/IFrame
- ✓ File uploader
- ✓ Popup window
- ❑ Table

Actions Class

Selenium has also provided a set of APIs to handle with some special mouse events:

- doubleClick()
- clickAndHold()
- release()
- moveToElement(target)
- contextClick()
- dragAndDrop(source, target)
- dragAndDrop(source, x-offset, y-offset)

Special Topic - Find All the broken/valid links in a webpage