Department of
Computer Science and Engineering

Title: Software testing tools part II (Selenium Webdriver -II).

Software Testing & Quality Assurance Lab

CSE 434

Green University of Bangladesh

1 Objective(s)

- To understand Locators in Selenium.
- Find Element and FindElements in Selenium WebDriver.

2 Locators

Locator is a command that tells Selenium IDE which GUI elements (say Text Box, Buttons, Check Boxes etc) its needs to operate on. Identification of correct GUI elements is a prerequisite to creating an automation script. But accurate identification of GUI elements is more difficult than it sounds. Sometimes, you end up working with incorrect GUI elements or no elements at all! Hence, Selenium provides a number of Locators to precisely locate a GUI element

2.1 Locating by ID:

This is the most common way of locating elements since ID's are supposed to be unique for each element. Target Format: id=id of the element.

2.2 Locate Element By Name using Filters:

Filters can be used when multiple elements have the same name. Filters are additional attributes used to distinguish elements with the same name.
Target Format: name=name-of-the-element filter=value-of-filter

2.3 Locating by Link Text:

This type of CSS locator in Selenium applies only to hyperlink texts. We access the link by prefixing our target with "link=" and then followed by the hyperlink text.
Target Format: link=link-text

2.4 Locating by CSS Selector:

CSS Selectors in Selenium are string patterns used to identify an element based on a combination of HTML tag, id, class, and attributes. Locating by CSS Selectors in Selenium is more complicated than the previous methods, but it is the most common locating strategy of advanced Selenium users because it can access even those elements that have no ID or name.
CSS Selectors in Selenium have many formats, but we will only focus on the most common ones.

- Tag and ID
- Tag and class
- Tag and attribute
- Tag, class, and attribute
- Inner text

3 Find Element and FindElements

Interaction with a web page requires a user to locate the web element. Find Element command is used to uniquely identify a (one) web element within the web page. Whereas, Find Elements command is used to uniquely identify the list of web elements within the web page. There are multiple ways to uniquely identify a web element within the web page such as ID, Name, Class Name, Link Text, Partial Link Text, Tag Name and XPATH.

3.1 FindElement command syntax:

Selenium Find Element command takes in the By object as the parameter and returns an object of type list WebElement in Selenium. By object in turn can be used with various locator strategies such as find element by ID Selenium, Name, Class Name, XPATH etc. Below is the syntax of FindElement command in Selenium web driver.
W ebElementelementN ame = *driver.f indElement*(*By.LocatorStrategy*("*LocatorV alue*"));
Locator Strategy can be any of the following values.

- ID

- Selenium find element by Name
- Class Name
- Tag Name
- Link Text
- Partial Link Text
- XPATH

Locator Value is the unique value using which a web element can be identified. It is the responsibility of developers and testers to make sure that web elements are uniquely identifiable using certain properties such as ID or name.
Example:
W ebElementloginLink = driver.f indElement(By.linkT ext(”Login”));

3.2 FindElements command syntax:

FindElements in Selenium command takes in By object as the parameter and returns a list of web elements. It returns an empty list if there are no elements found using the given locator strategy and locator value. Below is the syntax of find elements command.

List < W ebElement > elementN ame = driver.f indElements(By.LocatorStrategy(”LocatorV alue”));

Example:
List < W ebElement > listOfElements = driver.f indElements(By.xpath(”//div”));

4 Example 1: How to use Find Element command

Scenario:

- Open the AUT.
- Find and click radio button.

```
package com.sample.stepdefinitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class NameDemo {

public static void main(String[] args) {
// TODO Auto−generated method stub

System.setProperty("webdriver.chrome.driver", "D:\\3rdparty\\chrome\\ chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.manage().window().maximize();

driver.get("http://demo.guru99.com/test/ajax.html");

// Find the radio button for NO using its ID and click on it
driver.findElement(By.id("no")).click();

//Click on Check Button
driver.findElement(By.id("buttoncheck")).click();
```

}

}

5 Example 2: How to use Find Element command

Scenario:

- Open the URL for Application Under Test
- Find the text of radio buttons and print it onto the output console

```
package com.sample.stepdefinitions;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class NameDemo {

public static void main(String[] args) {

      System.setProperty("webdriver.chrome.driver", "X://chromedriver.exe"); WebDriver driver =
      new ChromeDriver();
      driver.get("http://demo.guru99.com/test/ajax.html");
      List<WebElement> elements = driver.findElements(By.name("name"));
      System.out.println("Number of elements:" +elements.size());

      for (int i=0; i<elements.size();i++){
          System.out.println("Radio button text:" + elements.get(i).getAttribute(" value"));
      }
   }
}
```

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Open a different URL for Application Under Test
2. Find the text of radio buttons and print it onto the output console

7 Lab Exercise (Submit as a report)

1. Find Element command returns the web element that matches the first most element within the web page. 2. Find Elements command returns a list of web elements that match the criteria.

3. Find Element by XPath in Selenium command throws NoSuchElementException if it does not find the element matching the criteria.

4. Find Elements command returns an empty list in Selenium if there are no elements matching the criteria

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.

9 Resources

https://www.guru99.com/find-element-selenium.html