**Check Visibility of Web Elements Using Various Types WebDriver Commands**

**How to check visibility of web elements using various types of looping and conditional commands in WebDriver:**

Previously in the series, we discussed about [WebDriver’s Select class](http://www.softwaretestinghelp.com/selenium-select-class-selenium-tutorial-13/)which is primarily used to handle web elements like dropdowns and selecting various options under the dropdowns.

Moving ahead in the [Selenium series](http://www.softwaretestinghelp.com/selenium-tutorial-1/), we would be discussing about the various types of looping and conditional commands in WebDriver like isSelected(), isEnabled() and isDispalyed(). These methods are used to determine the visibility scope for the web elements.

So let us start with a brief introduction – WebDriver has a W3C specification that details out the information about the different visibility preferences based out on the types of the web elements upon which the actions are to be performed.

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, dropboxes, checkboxes, radio buttons, labels etc.

* isDisplayed()
* isSelected()
* isEnabled()

For an improved understanding, let us discuss the aforementioned methods with code examples.

As a specimen, we would be using the “google.com” as an application under test and the “Learning\_Selenium” project created in the previous tutorials for script generation.

**Scenario to be automated**

1. Launch the web browser and open the application under test – http://google.com
2. Verify the web page title
3. Verify if the “Google Search” button is displayed
4. Enter the keyword in the “Google Search” text box by which we would want to make the request
5. Verify that the “Search button” is displayed and enabled
6. Based on visibility of the Search button, click on the search button

### ****WebDriver Code****

**Step 1:** Create a new java class named as “VisibilityConditions” under the “Learning\_Selenium” project.

**Step 2:** Copy and paste the below code in the “VisibilityConditions.java” class.

**Below is the test script that is equivalent to the above mentioned scenario:**

|  |  |
| --- | --- |
| 1 | import org.openqa.selenium.By; |
| 2 | import org.openqa.selenium.WebDriver; | |

|  |  |
| --- | --- |
| 3 | import org.openqa.selenium.WebElement; |
| 4 | import org.openqa.selenium.firefox.FirefoxDriver; | |

|  |  |
| --- | --- |
| 5 |  |
| 6 | public class VisibilityConditions { | |

|  |  |
| --- | --- |
| 7 |  |
| 8 | /\*\* | |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | \* @param args | | |
| 10 | | \*/ |

|  |  |
| --- | --- |
| 11 |  |
| 12 | public static void main(String[] args) { | |

|  |  |
| --- | --- |
| 13 |  |
| 14 | // objects and variables instantiation | |

|  |  |  |
| --- | --- | --- |
| 15 | WebDriver driver = new FirefoxDriver(); | |
| 16 | String appUrl = "https://google.com"; |

|  |  |
| --- | --- |
| 17 |  |
| 18 | // launch the firefox browser and open the application url | |

|  |  |  |
| --- | --- | --- |
| 19 | driver.get(appUrl); | |
| 20 |  |

|  |  |
| --- | --- |
| 21 | // maximize the browser window |
| 22 | driver.manage().window().maximize(); | |

|  |  |
| --- | --- |
| 23 |  |
| 24 | // declare and initialize the variable to store the expected title of the webpage. | |

|  |  |  |
| --- | --- | --- |
| 25 | String expectedTitle = "Google"; | |
| 26 |  |

|  |  |  |
| --- | --- | --- |
| 27 | // fetch the title of the web page and save it into a string variable | |
| 28 | String actualTitle = driver.getTitle(); |

|  |  |
| --- | --- |
| 29 |  |
| 30 | // compare the expected title of the page with the actual title of the page and print the result | |

|  |  |  |
| --- | --- | --- |
| 31 | if (expectedTitle.equals(actualTitle)) | |
| 32 | { |

|  |  |  |
| --- | --- | --- |
| 33 | System.out.println("Verification Successful - The correct title is displayed on the web page."); | |
| 34 | } |

|  |  |  |
| --- | --- | --- |
| 35 | else | |
| 36 | { |

|  |  |  |
| --- | --- | --- |
| 37 | System.out.println("Verification Failed - An incorrect title is displayed on the web page."); | |
| 38 | } |

|  |  |
| --- | --- |
| 39 |  |
| 40 | // verify if the “Google Search” button is displayed and print the result | |

|  |  |  |  |
| --- | --- | --- | --- |
| 41 | booleansubmitbuttonPresence=driver.findElement(By.id("gbqfba")).isDisplayed(); | | |
| 42 | | System.out.println(submitbuttonPresence); |

|  |  |
| --- | --- |
| 43 |  |
| 44 | // enter the keyword in the “Google Search” text box by which we would want to make the request | |

|  |  |  |
| --- | --- | --- |
| 45 | WebElement searchTextBox = driver.findElement(By.id("gbqfq")); | |
| 46 | searchTextBox.clear(); |

|  |  |  |
| --- | --- | --- |
| 47 | searchTextBox.sendKeys("Selenium"); | |
| 48 |  |

|  |  |
| --- | --- |
| 49 | // verify that the “Search button” is displayed and enabled |
| 50 | boolean searchIconPresence = driver.findElement(By.id("gbqfb")).isDisplayed(); | |

|  |  |  |
| --- | --- | --- |
| 51 | boolean searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled(); | |
| 52 |  |

|  |  |  |
| --- | --- | --- |
| 53 | if (searchIconPresence==true && searchIconEnabled==true) | |
| 54 | { |

|  |  |
| --- | --- |
| 55 | // click on the search button |
| 56 | WebElement searchIcon = driver.findElement(By.id("gbqfb")); | |

|  |  |  |
| --- | --- | --- |
| 57 | searchIcon.click(); | |
| 58 | } |

|  |  |
| --- | --- |
| 59 |  |
| 60 | // close the web browser | |

|  |  |
| --- | --- |
| 61 | driver.close(); |
| 62 | System.out.println("Test script executed successfully."); | |

|  |  |
| --- | --- |
| 63 |  |
| 64 | // terminate the program | |

|  |  |  |
| --- | --- | --- |
| 65 | System.exit(0); | |
| 66 | } |

|  |  |
| --- | --- |
| 67 | } |

**Code Walkthrough**

------------

Following are the ways in which we ascertain the presence of web elements on the web page.

**boolean**submitbuttonPresence=driver.findElement(By.id(“gbqfba”)).isDisplayed();

### isDispalyed()

isDisplayed() is the method used to verify presence of a web element within the webpage. The method is designed to result a Boolean value with each success and failure. The method returns a “true” value if the specified web element is present on the web page and a “false” value if the web element is not present on the web page.

Thus the above code snippet verifies for the presence of submit button on the google web page and returns a true value if the submit button is present and visible else returns a false value if the submit button is not present on the web page.

**boolean**searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();

The method deals with the visibility of all kinds of web elements not just limiting to any one type.

### isEnabled()

isEnabled() is the method used to verify if the web element is enabled or disabled within the webpage. Like isDisplayed() method, it is designed to result a Boolean value with each success and failure. The method returns a “true” value if the specified web element is enabled on the web page and a “false” value if the web element is not enabled (state of being disabled) on the web page.

Thus the above code snippet verifies if the submit button is enabled or not and returns a Boolean value depending on the result.

The isEnabled() method is significant in scenarios where we want to ascertain that only if “Condition A” is fulfilled, then the element(principally button) is enabled. Refer the following illustration for the same.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Webdriver-commands-11.jpg)

In the above figure, Register button button is enabled only when the agreement checkbox is selected.

Akin to above methods, we have a method referenced as “isSelected()” which tests if the specified web element is selected or not.

**boolean**searchIconSelected = driver.findElement(By.id(“male”)).isSelected();

### isSelected()

isSelected() is the method used to verify if the web element is selected or not. isSelected() method is pre-dominantly used with radio buttons, dropdowns and checkboxes. Analogous to above methods, it is designed to result a Boolean value with each success and failure.

Thus the above code snippet verifies if the male radio button is selected or not and returns a Boolean value depending on the result. Refer the following image for the same.

### ****Conclusion****

In this tutorial, we tried to make you acquainted with the WebDriver’s looping and conditional operations. These conditional methods often deal with almost all types of visibility options for web elements.