**Usage of Selenium Select Class for Handling Dropdown Elements on a Web Page**

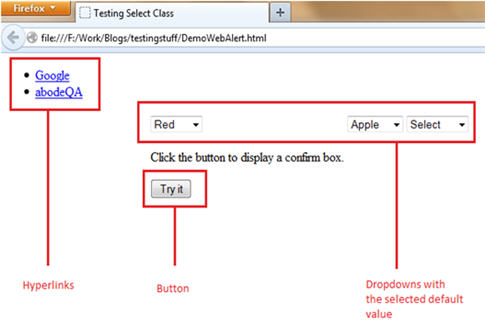
we would concentrate on **handling the various types of web elements available on the web pages**. Therefore, in this tutorial, we would consider **“dropdowns” and exercise their handling strategies**.

Before moving towards problem statement and its resolution, let us take a moment to introduce and create an understanding regarding the application under test. As a sample, we have created a **dummy HTML page** consisting of multiple and assorted web elements.

The elementary web elements those constitute the web page are:

* Hyperlink
* Button
* Dropdown

**Please take a reference of the following webpage aforementioned above:**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-select-class-1.jpg)

### Explanation of Application under Test

We have designed the web page in a way to include a few fundamental types of web elements.

* **Hyperlink**: The two hyperlinks namely “Google” and “abodeQA” have been provided that re-directs the user to “https://www.google.co.in/” and “http://www.abodeqa.com/” respectively on the click event.
* **Dropdown**: The three dropdowns have been created for selecting colors, fruits and animals with a value already set to default.
* **Button**: A “try it” button has been created to show up the pop up box having Ok and Cancel button upon click event.

**Subsequent is the HTML code used to create the above mentioned webpage:**

|  |  |  |
| --- | --- | --- |
| 1 | <!DOCTYPE html> | |
| 2 | <html> |

|  |  |  |
| --- | --- | --- |
| 3 | <head><title> Testing Select Class </title> | |
| 4 | <body> |

|  |  |
| --- | --- |
| 5 | <div id="header"> |
| 6 | <ul id="linkTabs"> | |

|  |  |
| --- | --- |
| 7 | <li> |
| 8 | <a href="https://www.google.co.in/">Google</a> | |

|  |  |  |
| --- | --- | --- |
| 9 | </li> | |
| 10 | | <li> |

|  |  |  |
| --- | --- | --- |
| 11 | <a href="http://abodeqa.wordpress.com/">abodeQA</a> | |
| 12 | </li> |

|  |  |
| --- | --- |
| 13 | </ul> |
| 14 | </div> | |

|  |  |  |
| --- | --- | --- |
| 15 | <div class="header\_spacer"></div> | |
| 16 | <div id="container"> |

|  |  |  |
| --- | --- | --- |
| 17 | <div id="content" style="padding-left: 185px;"> | |
| 18 | <table id="selectTable"> |

|  |  |  |
| --- | --- | --- |
| 19 | <tbody> | |
| 20 | <tr> |

|  |  |
| --- | --- |
| 21 | <td> |
| 22 | <div> | |

|  |  |
| --- | --- |
| 23 | <select id="SelectID\_One"> |
| 24 | <option value="redvalue">Red</option> | |

|  |  |
| --- | --- |
| 25 | <option value="greenvalue">Green</option> |
| 26 | <option value="yellowvalue">Yellow</option> | |

|  |  |  |
| --- | --- | --- |
| 27 | <option value="greyvalue">Grey</option> | |
| 28 | </select> |

|  |  |  |
| --- | --- | --- |
| 29 | </div> | |
| 30 | </td> |

|  |  |
| --- | --- |
| 31 | <td> |
| 32 | <div> | |

|  |  |
| --- | --- |
| 33 | <select id="SelectID\_Two"> |
| 34 | <option value="applevalue">Apple</option> | |

|  |  |  |
| --- | --- | --- |
| 35 | <option value="orangevalue">Orange</option> | |
| 36 | <option value="mangovalue">Mango</option> |

|  |  |  |
| --- | --- | --- |
| 37 | <option value="limevalue">Lime</option> | |
| 38 | </select> |

|  |  |  |
| --- | --- | --- |
| 39 | </div> | |
| 40 | </td> |

|  |  |
| --- | --- |
| 41 | <td> |
| 42 | <div> | |

|  |  |
| --- | --- |
| 43 | <select id="SelectID\_Three"> |
| 44 | <option value="selectValue">Select</option> | |

|  |  |  |
| --- | --- | --- |
| 45 | <option value="elephantvalue">Elephant</option> | |
| 46 | <option value="mousevalue">Mouse</option> |

|  |  |  |
| --- | --- | --- |
| 47 | <option value="dogvalue">Dog</option> | |
| 48 | </select> |

|  |  |  |
| --- | --- | --- |
| 49 | </div> | |
| 50 | </td> |

|  |  |  |
| --- | --- | --- |
| 51 | </tr> | |
| 52 | <tr> |

|  |  |
| --- | --- |
| 53 | <td> |
| 54 | <!DOCTYPE html> | |

|  |  |
| --- | --- |
| 55 | <html> |
| 56 | <body> |

|  |  |  |
| --- | --- | --- |
| 57 | <p>Click the button to display a confirm box.</p> | |
| 58 | <button onclick="myFunction()">Try it</button> |

|  |  |
| --- | --- |
| 59 | <script> |
| 60 | function myFunction() | |

|  |  |
| --- | --- |
| 61 | { |
| 62 | confirm("Press a button!"); | |

|  |  |
| --- | --- |
| 63 | } |
| 64 | </script> | |

|  |  |
| --- | --- |
| 65 | </body> |
| 66 | </html> |

|  |  |
| --- | --- |
| 67 | </td> |
| 68 | </tr> |

|  |  |
| --- | --- |
| 69 | </tbody> |
| 70 | </table> |

|  |  |
| --- | --- |
| 71 | </div> |
| 72 | </div> |

|  |  |
| --- | --- |
| 73 | </body> |
| 74 | </html> |

### ****Scenario to be automated****

* Launch the web browser and open the webpage
* Click on the “Google” hyperlink
* Navigate back to the original web page
* Select the “Green” in color dropdown
* Select the “Orange” in the fruit dropdown
* Select the “Elephant” in the animal dropdown

### ****WebDriver Code using Selenium Select Class****

Please take a note that, for script creation, we would be using “Learning\_Selenium” project created in the former tutorial.

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

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

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

**package** com.webdriver.selectdemo;

**import** org.junit.After;

**import** org.junit.Before;

**import** org.junit.Test;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.Select;

/\*import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.Select;

\*/

/\*\*

\* **@author** Smita B Kumar

\*

\*/

**public** **class** HandlingDropDown{

WebDriver driver;

/\*\*

\* Set up browser settings and open the application

    \*/

@Before

**public** **void** setUp(){

System.*setProperty*("webdriver.gecko.driver", "D:\\Selenium\\Software\\geckodriver-v0.13.0-win64\\geckodriver.exe");

driver =**new** FirefoxDriver();

driver.manage().window().maximize();

}

/\*\*

\* Test to select the dropdown values

\* **@throws** InterruptedException

\*/

@Test

**public** **void** testSelectFunctionality() **throws** InterruptedException{

//goto the google

driver.findElement(By.*linkText*("Google")).click();

//navigate back to the previous webpage

driver.navigate().back();

Thread.*sleep*(5000);

//select the first operator using "select by value"

Select selectByValue=**new** Select(driver.findElement(By.*id*("SelectID\_One")));

selectByValue.selectByValue("greenvalue");

Thread.*sleep*(5000);

// select the second dropdown using "select by visible text"

Select selectByVisibleText =**new** Select(driver.findElement(By.*id*("SelectID\_Two")));

selectByVisibleText.selectByVisibleText("Lime");

Thread.*sleep*(5000);

// select the third dropdown using "select by index"

Select selectByIndex =**new** Select(driver.findElement(By.*id*("SelectID\_Three")));

selectByVisibleText.selectByIndex(2);

Thread.*sleep*(5000);

}

/\*\*Tear down the setup after test completes

\* \*/

@After

**public** **void** tearDown(){

driver.quit();

}

### }

### ****Code Walkthrough****

**Import Statements**

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

* **import**org.openqa.selenium.support.ui.Select – Import this package prior to the script creation. The package references to the Select class which is required to handle the dropdown.

**Object Instantiation for Select class**

Select selectByValue = new Select(driver.findElement(By.id(“SelectID\_One”)));

We create a reference variable for Select class and instantiate it using Select class and the identifier for the drop down.

The identifier or the locator value for the drop down can be found using the techniques discussed in the initial tutorials (by using Selenium IDE and firebug).

**Take a notice that the identifier for a dropdown can be found as below:**

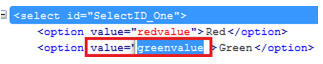
**Step 1:** Most or almost all the dropdowns elements are defined in the <Select> tag having multiple values (values that can be set into the dropdown) that are defined under the <option> tags.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-select-class-2.jpg)

**Setting the value in the dropdown using selectByValue()method**

selectByValue.selectByValue(“greenvalue”);

In the above java command, we select the value “green” in the drop down using the selectByValue() method and parameterizing it with the text present in the value attribute.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-select-class-3.jpg)

**Setting the value in the dropdown using selectByVisibleText() method**

selectByValue.selectByVisibleText(“Lime”);

In the above java command, we select the value “Lime” in the drop down using the selectByVisibleText() method and parameterizing it with the text present on the user interface or the text present between the opening and closing <option> tags.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-select-class-4.jpg)

**Setting the value in the dropdown using selectByIndex()method**

selectByValue.selectByIndex(“2”);

In the above java command, we select the third value in the drop down using the selectByIndex() method and parameterizing it with the index value of the element which is desired to be selected in the dropdown.

Take a note that the index value starts with “0”.

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

In this tutorial, we tried to make you acquainted with the WebDriver’s Select class that is used to handle dropdown elements present on the web page. We also briefed you about the methods that can be used to populate the value in the dropdown.

**Here is the article summary:**

* WebDriver’s Select class is used to handle the dropdown elements present on a web page.
* Prior to the actual scripting, we need to import a package to be able to create a WebDriver script for handling a dropdown and making the Select class accessible.
  + **import**org.openqa.selenium.support.ui.Select;
* We create a reference variable for Select class and instantiate it using Select class and the identifier for the drop down.
  + Select selectByValue = new Select(driver.findElement(By.id(“SelectID\_One”)));
* The identifier or the locator value for the drop can be found using Selenium IDE and firebug.
* Ideally there are three ways to select the desired value in the dropdown amongst the listed one.
  + selectByValue()
  + selectByVisibleText()
  + selectByIndex()
* The following java command is used to select the “green” color in the dropdown. Take a notice the value in the dropdown is selected using the selectByValue()
  + selectByValue(“greenvalue”);
* The following java command is used to select the “Lime” fruit in the dropdown. Take a notice the value in the dropdown is selected using the selectByVisibleText()
  + selectByVisibleText(“Lime”);
* The following java command is used to select the third value amongst all the available options enlisted for the dropdown. Take a notice the value in the dropdown is selected using theselectByIndex()
  + selectByIndex(“2”);