Java Script Executor

In general, we do click on an element using click() method in Selenium.

For example:

*driver.findElement(By.id("Id Value")).click();*

Sometimes web controls don’t react well against selenium commands and we may face issues with the above statement (click()). To overcome such kind of situation, we use JavaScriptExecutor interface.

It provides mechanism to execute Javascript through Selenium driver. It provides “executescript” & “executeAsyncScript” methods, to run JavaScript in the context of the currently selected frame or window.

There is no need to write separate script to execute JavaScript within the browser using Selenium WebDriver script. Just use predefined interface named ‘Java Script Executor’. We need to import the below package in the script.

Package:

import org.openqa.selenium.JavascriptExecutor;

Syntax:

*JavascriptExecutor js = (JavascriptExecutor) driver;*

*js.executeScript(Script,Arguments);*

Script – The JavaScript to execute

Arguments –The arguments to the script(Optional). May be empty.

Returns –One of Boolean, Long, String, List, WebElement, or null.

Let’s see some scenarios we could handle using this Interface:

**To type Text in Selenium WebDriver without using sendKeys() method**

js.executeScript("document.getElementById('some id').value='someValue';");

js.executeScript("document.getElementById(Name).value=Bix;");

**To click a Button in Selenium WebDriver using JavaScript**

js.executeScript("document.getElementById('enter your element id').click();");

//or

js.executeScript("arguments[0].click();", loginButton);

**To handle Checkbox**

js.executeScript("document.getElementById('enter element id').checked=false;");

**To generate Alert Pop window in selenium**

js.executeScript("alert('Welcome To BIX);");

**To refresh browser window using Javascript**

js.executeScript("history.go(0)");

**To get innertext of the entire webpage in Selenium**

String sText = js.executeScript("return document.documentElement.innerText;").toString();

System.out.println(sText);

**To get the Title of our webpage**

String sText = js.executeScript("return document.title;").toString();

System.out.println(sText);

**To get the domain**

String sText = js.executeScript("return document.domain;").toString();

System.out.println(sText);

**To get the URL of a webpage**

String sText = js.executeScript("return document.URL;").toString();

System.out.println(sText);

**To click on a SubMenu which is only visible on mouse hover on Menu**

js.executeScript("$('ul.menus.menu-secondary.sf-js-enabled.sub-menu li').hover()");

**To navigate to different page using Javascript**

js.executeScript("window.location = 'https://www.google.com");

**To find hidden element in selenium using JavaScriptExecutor**

js.executeScript("arguments[0].click();", element);

**To perform Scroll on application using Selenium**

//Vertical scroll - down by 500 pixels

js.executeScript("window.scrollBy(0,500)");

for scrolling till the bottom of the page we can use the code like

js.executeScript("window.scrollBy(0,document.body.scrollHeight)");

**To perform scroll till element**

je.executeScript("arguments[0].scrollIntoView(true);",webelement);