JavaScriptExecutor

1. It is used to boost up the performance of selenium
2. When nothing is working or id is dynamic in nature, we can use JavaScriptExecutor
3. JavaScriptExecutor is a class, we have to cast driver with JavaScriptExecutor

JavascriptExecutor js = ((JavascriptExecutor)driver);

1. executeScript is a method is used to execute JavaScriptCode

js.executeScript("….”);

* Highlight element in Selenium using JavaScriptExecutor:
* +Must have feature Highlight element Selenium for better execution.
* In Automation, testing sometimes element highlighter plays very important role. It helps us to track our execution flow which step is being processed. Some tools like QTP, Sahi etc. you will get this inbuilt feature.
* For Selenium, we have to write small code, which simply highlights element based on our parameter values. let’s get started and see Highlight element Selenium using CSS values. In Selenium, we can use JavascriptExecutor (interface) to execute Javascript code into webdriver.
* 1. Highlight an Element 2. DrawBorder around an Element 3. Generate custom Alert during Test Execution 4. Click on an Element by using JaveScript 5. Refresh browser by using JavaScript 6. Get Title by using JavaScript 7. Get InnerText of page by using JavaScript 8. Scroll Into View and Scroll Page Down

**package** com.NormalPrograms;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.Alert;

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

**import** org.openqa.selenium.JavascriptExecutor;

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

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

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

**public** **class** JavaScriptExecutor {

**public** **static** **void** main(String[] args)

{

WebDriver driver = **null**;

String url = "https://www.facebook.com/";

System.*setProperty*("webdriver.chrome.driver", "F:\\Javaworkspace\\Practicing\_Programs\\driverFiles\\chromedriver.exe");

driver = **new** ChromeDriver();

//driver = new HtmlUnitDriver();

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

driver.manage().timeouts().implicitlyWait(40, TimeUnit.***SECONDS***);

driver.manage().timeouts().pageLoadTimeout(40, TimeUnit.***SECONDS***);

driver.manage().deleteAllCookies();

driver.get(url);

driver.findElement(By.*id*("email")).sendKeys("hello");

driver.findElement(By.*id*("pass")).sendKeys("hello");

//<input value="Log In" aria-label="Log In" data-testid="royal\_login\_button" type="submit" id="u\_0\_8">

WebElement login = driver.findElement(By.*xpath*("//input[@value='Log In' and @data-testid='royal\_login\_button']"));

System.***out***.println("login identified");

*flash*(login, driver);

System.***out***.println("flash executed");

*drawcolor*(login, driver);

System.***out***.println("drawcolor executed");

*generatealert*(driver, "There is an issue with this page");

System.***out***.println("generatealert executed");

Alert lt = driver.switchTo().alert();

lt.accept();

*clickon*(login, driver);

System.***out***.println("clickon executed");

*refresh*(driver);

System.***out***.println("refresh executed");

System.***out***.println(*getTitle*(driver));

System.***out***.println("getTitle executed");

System.***out***.println(*getPageInnerText*(driver));

System.***out***.println("getPageInnerText executed");

*scrollpageDown*(driver);

System.***out***.println("scrolldown executed");

driver.navigate().back();

WebElement lang =driver.findElement(By.*linkText*("Sign Up"));

*scrollpageIntoView*(lang, driver);

System.***out***.println("scrollintiview executed");

}

//Highlighting an Element

**public** **static** **void** flash(WebElement element,WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

String bgcolor = element.getCssValue("backgroundColor");

**for**(**int** i=0;i<10;i++)

{

*changecolor*("rgb(0,200,0)", element, driver);

*changecolor*(bgcolor, element, driver);

}

}

**public** **static** **void** changecolor(String color,WebElement element, WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

js.executeScript("arguments[0].style.backgroundColor = '"+color+"'",element);

**try**

{

Thread.*sleep*(20);

}

**catch**(InterruptedException e)

{

}

}

// DrawBorder around an Element

**public** **static** **void** drawcolor(WebElement element,WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

js.executeScript("arguments[0].style.border = '3px solid red'",element);

}

// Generate custom Alert during Test Execution

**public** **static** **void** generatealert(WebDriver driver,String message)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

js.executeScript("alert('"+message+"')");

}

// Click on element by Js

**public** **static** **void** clickon(WebElement element,WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

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

}

// Refresh browser by Js

**public** **static** **void** refresh(WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

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

}

//Get Title by using JS

**public** **static** String getTitle(WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

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

**return** title;

}

// Get InnerText of page by JS

**public** **static** String getPageInnerText(WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

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

**return** title;

}

// Scroll down Page

**public** **static** **void** scrollpageDown(WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

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

}

// Scroll Page IntoView

**public** **static** **void** scrollpageIntoView(WebElement element,WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

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

}

}

JavaScriptExecutorConcept.java by Naveen

/\*

\* @author Naveen Khunteta

\*/

package SeleniumSessions;

import java.io.File;

import java.io.IOException;

import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class JavaScriptExecutorConcept {

public static void main(String[] args) throws IOException {

System.setProperty("webdriver.chrome.driver", "/Users/naveenkhunteta/Downloads/chromedriver");

WebDriver driver = new ChromeDriver(); //launch chrome

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

driver.manage().deleteAllCookies(); //delete all the cookies

//dynamic wait

driver.manage().timeouts().pageLoadTimeout(40, TimeUnit.SECONDS);

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

driver.get("https://www.freecrm.com"); //enter URL

driver.findElement(By.name("username")).sendKeys("naveenk");

driver.findElement(By.name("password")).sendKeys("test@1234");

//driver.findElement(By.xpath("//input[contains(@type,'submit')]")).click();

//executeScript -- to execute JavaScript code

WebElement loginBtn = driver.findElement(By.xpath("//input[contains(@type,'submit')]")); //login button

flash(loginBtn, driver); //highlight the element

drawBorder(loginBtn, driver); //draw a border

//take screenshot:

// Take screenshot and store as a file format

File src = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

// now copy the screenshot to desired location using copyFile //method

FileUtils.copyFile(src, new File("/Users/naveenkhunteta/Documents/workspace/MorningSessions/src/SeleniumSessions/element.png"));

//generate Alert

//generateAlert(driver, "There is an issue with Login button on Login Page");

//click on any element by using JS executor

clickElementByJS(loginBtn, driver);

//refresh the page:

//1. by using selenium:

driver.navigate().refresh();

//2. by using JS executor:

refreshBrowserByJS(driver);

//get the tile of the page by JS:

System.out.println(getTitleByJS(driver));

//get the page text:

System.out.println(getPageInnerText(driver));

//scroll page down:

//scrollPageDown(driver);

WebElement forgotPwdLink = driver.findElement(By.xpath("//a[contains(text(),'Forgot Password?')]"));

scrollIntoView(forgotPwdLink, driver);

}

public static void flash(WebElement element, WebDriver driver) {

JavascriptExecutor js = ((JavascriptExecutor) driver);

String bgcolor = element.getCssValue("backgroundColor");

for (int i = 0; i < 10; i++) {

changeColor("rgb(0,200,0)", element,driver);//1

changeColor(bgcolor, element,driver);//2

}

}

public static void changeColor(String color, WebElement element, WebDriver driver) {

JavascriptExecutor js = ((JavascriptExecutor) driver);

js.executeScript("arguments[0].style.backgroundColor = '"+color+"'", element);

try {

Thread.sleep(20);

} catch (InterruptedException e) {

}

}

public static void drawBorder(WebElement element, WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

js.executeScript("arguments[0].style.border='3px solid red'", element);

}

public static void generateAlert(WebDriver driver, String message){

JavascriptExecutor js = ((JavascriptExecutor) driver);

js.executeScript("alert('"+message+"')");

}

public static void clickElementByJS(WebElement element, WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

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

}

public static void refreshBrowserByJS(WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

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

}

public static String getTitleByJS(WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

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

return title;

}

public static String getPageInnerText(WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

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

return pageText;

}

public static void scrollPageDown(WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

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

}

public static void scrollIntoView(WebElement element, WebDriver driver){

JavascriptExecutor js = ((JavascriptExecutor) driver);

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

}

}

Open with