Selenium Easy

For Complete Web Automation Testing Tutorials using Selenium Webdriver

Home >> Selenium Tutorials >> Perform operations on new window

Perform operations on new window

There are cases where we need to open new window and perform operations or there may be cases where after clicking on any button / link, it opens new window and need to perform operations on the new window.

Let us look into such example:

Test case: We need to open 'http://linkedin.com' and click on 'Help Center' link at the bottom which will open new window.
1. Verify the title of the new window
2. Verify text 'Welcome' on the page.
3. Search for a Question with text "Frequently Asked Questions" and verify the result.

```
package com.pack;

import java.util.Set;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.annotations.Test;

public class WindowExamples {
        public static WebDriver driver;

        @Test
        public void verifySearchInNewWindow() throws InterruptedException {
                driver = new FirefoxDriver();
                driver.navigate().to("http://linkedin.com/");
                driver.manage().window().maximize();
                String mainHandle = driver.getWindowHandle();

                //Wait for the element to be present
                WebDriverWait wait = new WebDriverWait(driver, 5);
                wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelect
or(".cust-svc-link")));
                driver.findElement(By.linkText("Help Center")).click();

                //Switch to new window and verify the title
                waitForNewWindowAndSwitchToIt(driver);
                String newTitle = getCurrentWindowTitle();
                Assert.assertEquals(newTitle, "LinkedIn Help Center", "New window tit
le is not matching");

                //Verify the text present on the page
                String textOnpage=driver.findElement(By.cssSelector(".welcome")).getT
ext().trim();
                Assert.assertEquals(textOnpage, "Welcome!");

                //Verify search text on the page
                String searchText="Frequently Asked Questions";
                WebElement searchInputBox=driver.findElement(By.id("kw"));
                searchInputBox.sendKeys(searchText);

                WebElement searchButton = driver.findElement(By.cssSelector(".butto
n.leftnoround.blue"));
                searchButton.click();

                WebElement resultedElement = driver.findElement(By.cssSelector(".rn_E
lement2"));
                String resultedText = resultedElement.getText().trim();
                System.out.println(resultedText);
                Assert.assertTrue(resultedText.contains(searchText), "Search successf
ull");

                closeAllOtherWindows(driver, mainHandle);
        }
```

To execute the above test we have created a methods which can be reused with multiple tests.

Below method is used to get the main window handle. we will the driver as parameter.

```
        public static String getMainWindowHandle(WebDriver driver) {
                return driver.getWindowHandle();
        }
```

Below method is used to get the current window title

```
        public static String getCurrentWindowTitle() {
                String windowTitle = driver.getTitle();
                return windowTitle;
        }
```

Below method is used to close all the other windows except the main window.

```
        public static boolean closeAllOtherWindows(WebDriver driver, String openWindowHandle) {
                Set<String> allWindowHandles = driver.getWindowHandles();
                for (String currentWindowHandle : allWindowHandles) {
                        if (!currentWindowHandle.equals(openWindowHandle)) {
                                driver.switchTo().window(currentWindowHandle);
                                driver.close();
                        }
                }

                driver.switchTo().window(openWindowHandle);
                if (driver.getWindowHandles().size() == 1)
                        return true;
                else
                        return false;
        }
```

Below method is used to wait for the new window to be present and switch to it.

```
         public static void waitForNewWindowAndSwitchToIt(WebDriver driver) throws InterruptedException {
                String cHandle = driver.getWindowHandle();
                String newWindowHandle = null;
                Set<String> allWindowHandles = driver.getWindowHandles();

                //Wait for 20 seconds for the new window and throw exception if not found
                for (int i = 0; i < 20; i++) {
                    if (allWindowHandles.size() > 1) {
                        for (String allHandlers : allWindowHandles) {
                            if (!allHandlers.equals(cHandle))
                                newWindowHandle = allHandlers;
                        }
                        driver.switchTo().window(newWindowHandle);
                        break;
                    } else {
                        Thread.sleep(1000);
                    }
                }
                if (cHandle == newWindowHandle) {
                    throw new RuntimeException(
                            "Time out - No window found");
                }
            }
}
```

Note: When ever we work on multiple windows, switching plays major role. We should switch to the desired window to perform operations and again switch back to default window to work on main window.

‹ Download file using selenium webdriver Introduction to Page Object Model Framework ›

Tags:

Windows

Selenium Tutorials:

Selenium Tutorials

Selenium Tutorials

- Webdriver Select with Multiple Attribute
- Handling keyboard events and mouse hover events using Webdriver
- Set browser width and height in Selenium Webdriver
- Selenium Automation Framework Example
- UI Map or Object Repository using Properties File
- Test Data in automation framework
- Reporting Structure in Selenium Automation Framework
- Test Configuration File / Executable File
- Uploading a file with Selenium Webdriver
- Upload file using AutoIT
- Uploading a file with sendKeys method
- Read data from Properties file using Java Selenium
- Scrolling web page with Selenium Webdriver using java
- Keyword Driven Framework Example
- Install AutoIt in Windows
- Examples with xpath and Css (ID, Name, Text and Links)
- Testing in multiple browsers using selenium and testng
- Working with Checkbox using Resuable Methods
- Working with Select examples
- File upload using Robots
- Download file using selenium webdriver
- Perform operations on new window
- Introduction to Page Object Model Framework
- Mouse hover using javaScript Executor
- Working with Multiple Checkboxes with Safecheck
- Click element using JavaScriptExecutor
- Drag and Drop using Webdriver Action Class
- WebDriver Waits Examples
- Simple Page Object Model example
- Double Click on element using Webdriver
- Right Click Context Menu example
- Working with AutoComplete Text box
- Working with Date Picker Calendar
- Working with Ajax controls using Webdriver
- Get Attribute values using Webdriver
- Get Css values using Webdriver
- Resizing a web element using movebyoffset
- Validate Downloaded file after clicking on downloaded button/ link

- How to verify entries in Exported CSV file
- Element is not clickable at point SeleniumWebdriverException
- Extract PDF text And Verify Text Present in PDF using WebDriver
- Handling Authentication Window with WebDriver (In Firefox, Chrome and IE)

Recent Post

Handling Authentication Window with WebDriver (In Firefox, Chrome and IE)

Selenium Webdriver Tutorials

Locators for Selenium

Optional annotation in TestNG

Parameterization in TestNG using testng.xml

Automation Framework

- Framework Structure
- Object Repository
- Test Data
- Test Scripts
- Reports
- Configuration and Execution

Tags Cloud

Asserts (1) AutoComplete (1) AutoIt (2) Automation (3) Build.xml (1) Checkboxes (2) Cookies (3) Css (2) Css Examples (2) DataProvider (1) Date Picker (1) Drag and Drop (1) Exceptions (4) File Download (3) Firefox Profile (2) Framework Frameworks (5) iFrames (1) Interview Questions (1) JavascriptExecutor (3) JUnit (2) Keyword Driven (1) Listeners (3) Manual Testing (1) Maven (5) Mouse Hover (2) Page Object Model (2) Page Scroll (1) Parallel Test (3) properties file (1)

Home

- Home
- Tutorials
- About
- Contact
- Privacy Policy
- Site Map

Tutorial Menu

- Selenium Tutorials
- TestNG Tutorial
- JUnit Tutorial
- JXL Tutorial
- Apache POI
- ANT Tutorial
- Maven Tutorial

Selenium Tutorials

- Basic Webdriver Tutorials
- Advanced Webdriver Tutorials
- Selenium Framework Examples
- JavascriptExecutor Examples

Interview Questions

- Testing Concepts
- Quality Standards
- General Questions
- Roles & Responsibilities
- Technology & Environment

We welcome your questions and requests