Selenium 4 Beginners

Rex Jones II

Selenium Browser Methods

Table of Contents

Selenium 4 Beginners

Rex Jones II

Introduction

Hello everybody, Welcome To Selenium 4 Beginners. My name is Rex Allen Jones II. In this video, I will cover the Browser Methods for Selenium. There is a total of 5 Selenium Method Categories. Each category has methods for automating applications. In no particular order, we have a list of the following categories:

Browser Methods

WebElement Methods

Wait Methods

Navigation Methods

Switch Methods

Browser Methods are a group of methods that perform actions on a browser.

WebElement Methods are a group of methods that perform actions on WebElements. A WebElement is anything you see on a browser.

Wait Methods are a group of methods that pause between execution statements. For example, when your web page is loading the statement will pause until the web page finishes loading.

Navigation Methods are a group of methods that either loads a web page, refresh a web page, or move backwards and forwards in your browser's history. You know how – you leave one web page and want to go back to the previous page. Navigation Methods will help us perform actions such as going back to the previous page.

Switch Methods are a group of methods that switch to alerts, windows, and frames. An alert is a pop up that comes on your screen. We have to switch to that pop up and perform an action.

Over the next few weeks, I will demo all of these Selenium Method categories. Here's the Tutorial Plan for Selenium Browser Methods. First, we are going to look at the descriptions. Then demo the methods followed by a Practice Exercise.

Descriptions

Now, let's view the Browser Method descriptions. There's a total of 6 methods within this category. In alphabetical order, the 6 methods are close, get, getCurrentUrl, getPageSource, getTitle, and quit.

close () – closes the current active window

get (String url) – loads a new web page

getCurrentUrl() – gets a string defining the current web page URL

getPageSource() – gets the complete page source of the loaded web page

getTitle() – gets the current page title

quit() – stops running the driver and closes the window

Many times, automation engineers alternate between using close and quit because they are similar methods. I will demonstrate the difference between both methods using LinkedIn as our AUT Application Under Test.

Demo

Let's demo the Selenium Browser Methods. We start by writing WebDriver driver, add our Test Annotation, and name the method demoSeleniumBrowserMethods. Instantiate driver by writing driver = new ChromeDriver (); then import WebDriver, Test annotation and ChromeDriver using shortcut keys CTRL + SHIFT + O.

The first Selenium Browser Method we are going to use is get - driver.get. Do you see how the description states "Loads a new web page"? I want you to notice something while going through these Browser Methods. Get is the only Browser Method that receives a parameter. It receives a parameter of (String url). That means to complete this statement we must enter a URL as a String Data Type. https://www.LinkedIn.com

Make sure to write the entire url including https. The web page will not load if we enter www.LinkedIn.com. Let me show you. See how only a blank web page opens up. Now, watch what happens when I add https:// – (Run) LinkedIn opens up.

The next Selenium Browser Method is getCurrentUrl. We write driver.getCurrentUrl. The description states "get a string representing the current URL". Notice this method does not receive a parameter like the get method but it returns a value. The description also shows "Returns – the URL of the page currently loaded in the browser".

Do you see String – WebDriver? String is the return type and WebDriver is the Interface. When this method returns a value, we have the option of assigning the value to a String. It's not required. For demonstration purposes, let's assign the value. String then name the object reference strCurrentURL

Print strCurrentURL by writing sysout CTRL + SPACE. Within the print statement, I will write, What Is The Current URL? then the object reference strCurrentURL. Now, let's run. Go to the Console and verify the URL.

getTitle is the next method and it's similar to getCurrentURL. We write driver.getTitle and we see the description states "title of the current page". This method also returns a String value. I'm getting ready to assign the value String strTitle - and print the value What Is The Page Title + strTitle - Run. The title is LinkedIn: Log In or Sign Up. Console shows the correct Page Title.

The next Selenium Browser Method is getPageSource. We can view the Page Source by right clicking on a page then selecting View Page Source. Selenium will get all of this HTML which is an acronym for HyperText Markup Language. Write driver.getPageSource(). This time, I will not assign the value to an object reference like getCurrentURL and getTitle. However, I will print the Page Source when it is

returned to the method. We can print the value by cutting driver dot getPageSource then pasting it into the print statement. Run. In the Console, we see the Page Source.

Let me show you how to maximize the browser window before discussing the last 2 Selenium Browser Methods close and quit. Did you notice how the browser window was not maximized when executing our statements. We saw part of the browser window and part of Eclipse IDE. To maximize the browser window,

Type driver dot manage dot window dot maximize. Now let's run and you will see how the window is maximized before loading LinkedIn.

Now, let's talk about close and quit. Do you see these Chrome Browsers in the bottom of my Taskbar. I must click X in the browser's right corner to close all of the windows. Each of those windows showed up in my Taskbar after executing our statements. Both methods close and quit will perform that task of closing a browser window. However, the best way to see a difference between the methods is looking at Task Manager. Go to the Details tab. We see chromedriver.exe 6 times (1, 2, 3, 4, 5, 6).

Go back to Eclipse and start with close. Write driver.close, and we see the description states "close the current window". Let's run. This time the browser window will close without me manually clicking x in the top right corner.

Task Manager now shows chromedriver.exe 7 times (1, 2, 3, 4, 5, 6, 7). Personally, I prefer to use quit and you will see why.

Let's check out the quit method by writing driver.quit and the description states "quits this driver, closing every associated window". The Task Manager still shows chromedriver.exe 7 times (1, 2, 3, 4, 5, 6, 7). chromedriver.exe does not remain in Task Manager because the quit method stops running the driver and that's why I prefer quit. The quit method quits the driver and closes the window while the close method only closes the window.

That's it for Selenium Browser Methods.

Practice Exercise

The practice exercise is create a Test Script for Amazon. Load amazon, get the title, get the current url, then quit the driver. If you want to do extra, feel free to get the page source and close the browser using close method. It may seem simple but if you are new to Selenium or refreshing your skills, every bit of practice helps. Practice Make Improvement.

Here's the Test Case I used to demo Selenium Browser Methods. You can use this Test Case under Practice Test Steps for your Amazon Test Script.

Download this video's presentation, automation code, and Test Cases at
https://tinyurl.com/SeleniumBrowserMethods

Thank You For Watching Selenium Browser Methods