Selenium 4 Beginners

Rex Jones II

Selenium Switch Methods

Table of Contents

Selenium 4 Beginners

Rex Jones II

Introduction

Hello everybody and Welcome To Selenium 4 Beginners. In this video, I will cover Number 5 Switch Methods for Selenium.

The Transcript, Presentation, and Code will be available on github at https://github.com/RexJonesII/Selenium4Beginners and http://tinyurl.com/Switch-Methods-Selenium

Selenium Method Categories

Switch Methods are the last group of Selenium Method Categories. We already covered Browser Methods, WebElement Methods, Navigation Methods, and Wait Methods. The Switch Methods are a group of methods that switch to alerts, windows, and frames. Selenium will not allow us to directly interact with a WebElement inside an alert, window, or frame unless we execute a switch statement.

Tutorial Plan

We are going to discuss the Type of Elements That Require Switching, Selenium Methods Used For Switching, then Demo the Selenium Switch Methods.

Type of Elements That Require Switching

Let's start with the Type of Elements That Require Switching. First, we have Frames, then Alerts, and last is Windows.

Frames

A frame is a tag within a frameset tag. The frame defines one window within a frameset. Let me show you an example by using W3Schools. you see the 3 frame tags that's located inside the frameset tag. When I click Try It For Yourself, we see 3 frames on the right side. Frame A, Frame B, and Frame C. On the left side, we see the same 3 frame tags within a frameset tag.

Watch what happens when I change the value for src within a frame tag. I am going to change Frame B to https://www.w3schools.com then click Run. A w3schools.com window shows up within Frame B after entering the URL. Also, look what happens after right clicking the frame. An option named View Frame Source shows up. Frame A and Frame C shows the same option View Frame Source. However, we do not see View Frame Source when right clicking outside out of Frames A, B, and C.

Our demo tutorial will focus on iframes which is very similar to what we saw with frames. Iframe is short for inline frame which is used to embed documents within a document. Try It Yourself. The iframe tag uses an src attribute just like the frame tag. iFrame also has an option named View Frame Source when I right click.

However, there's a difference between frame and iframe. An iframe is not located within a frameset tag and the window for an iframe cannot be resized. When I hover over the window, the icon for resizing does not show up. Let's go to frames and notice we can resize the window.

Most applications, I come across have an iframe tag because they are designed for interactive applications. Iframes is a better option for adding other websites within a window. One reason an iframe is better is because developers can keep track of less HTML documents while using iframes. With automation, whether we come across an iframe or frame, we must always switch before performing an action.

Here's an example. The frame is the blue Log Into Application box. We must switch to the frame before entering data in the Username and Password fields. Also, before clicking the Submit button. Most of the times, it's difficult to notice a frame by looking at the web page. An exception shows up if we do not switch to the frame.

Alerts

The next element that require switching is alerts. Alerts are better known as pop up boxes. It's a box that contains a message. In this example, the alert has 2 buttons. An OK and CANCEL button. We also see a title that states Pop-Up / Alert Box with some pretend data between the title and 2 buttons. There are 3 types of alerts with a purpose to: number 1 send information, number 2 get confirmation, and number 3 receive input from a user. We are forced to switch and interact with the alert when it pops up. We are not able to interact with the web page. That's why the web page is gray and disabled.

Windows

The last element required for switching is Windows. It can be a new window or tab. This example shows a new window on top of the web page. We see a title that states Switch To This New Window Then Perform An Action. All windows have an ID called a window handle. A window handle is unique and alphanumeric that allows us to switch to the window. We can perform any action after switching to the window.

Selenium Methods Used For Switching

There are different Selenium methods used for switching.

Switch Method Types For Frames

Here's the methods used for switching to a frame.

frame(WebElement element) selects a frame by WebElement. WebElement is the Data Type and element is the frame element we will switch to.

frame(String nameOrId) selects a frame by its name or ID. String is the Data Type. Name and ID are the HTML attributes. We can switch by the name of the frame or id of the frame.

frame(int index) selects a frame by its index. int is the integer Data Type. All indexes start at zero.

defaultContent() select the first frame on the page or the main document if the page contains an iframe.

parentFrame() is important if the focus has changed from the parent context. By default, Selenium has focus set on the parent.

Switch Method Types For Alerts/Pop-Ups

Next, is the methods for Alerts. Imagine a pop-up box that has 2 buttons like one of our previous slides. The accept method accepts the alert by clicking the OK button and the dismiss method clicks the Cancel button. If you want to get information from the pop-up box, use the getText method while the sendKeys method types data into the pop-up box.

Switch Method Types For Windows

Last, we have 3 methods for switching to a window. The alphanumeric window handle is assigned to each window. The getWindowHandle method gets the window handle of the current window and getWindowHandles gets the window handle of all current windows. The switchTo().window() method switches focus between the windows.

Demo Frames

Now, let's dive into our demo of Selenium Switch Methods.

frame()

Let's walkthrough our Application Under Test https://paytm.com/ . First, we load the application then click the Log In/Sign Up hyperlink. Next, we click the link at the bottom "Login/Signup with mobile number and password". Finally, we enter our number or email. This Test Script will Fail and throw an exception if we do not switch to a frame but which step requires a switch. We are going to switch after clicking the Log In/Sign Up hyperlink.

Let's go to Eclipse and create our Automation Test Script. We setup our Test with System.setProperty on Chrome using driver = new ChromeDriver(); then maximize the window driver.manage().window().maximize;

The Test method is switchToFrames. Now load the AUT. driver.get, https://paytm.com . Add a dynamic Wait statement. This here is an Explicit Wait Statement for 5 seconds. Why add a wait statement?

We are going wait until 24x7 Help shows up. I am going to reload the page and you will see how we wait until 24x7 Help and these other elements show up. Finish the wait statement. You can watch video number 22 to see the benefits of an Explicit Wait. Now we click the Log In/Sign Up hyperlink. driver.findElement By xpath dot click. The xpath value will contain text for Log In/Sign Up then wait for this dialog to load. We are going to wait for the bottom hyperlink because that's the link we will click.

This is a customized xpath value for Log In/Sign Up. I will create a video explaining how to create customized xpath values. We add the wait statement then click the hyperlink at the bottom.

Last, we enter the phone number or email. driver.findElement By id dot sendKeys Get the value for id Inspect the element. We see the value for id is input_0 send (123) 456-7890.

Let's Run.

Nothing happens after clicking the Log In/Sign Up hyperlink. Our test failed.

Let's go back to our AUT and see why it failed. Right click and we see View Frame Source. That means this dialog is a frame or an iframe. It is like the example from our Introduction when looking at frames in W3 Schools. Notice the difference when I right click outside the iframe. We do not see View Frame Source. That lets us know we must switch before performing a command.

Go back to Eclipse and switch to frame. driver.switchTo. The description states "Sends future commands to a different frame or window". In this intellisense, we see different methods for switching to a frame or window. We are going to switch to a frame.

Here's the frame methods, we reviewed in our introduction. frame(index) selects a frame by it's zero based index. frame(String) selects a frame by its name or ID. frame(WebElement) selects a frame using it's element. Select frame.

Now, let's inspect an element within the iframe dialog. We are going to inspect the bottom link. The blue highlighted part is the link but let's scroll up to the iframe tag. This iframe tag has an src attribute that contains login. CTRL + F to find the value. Let's write our customized xpath value.

2 forward slashes

Iframe

Opening and Closing brackets

contains

Opening and Closing parenthesis

At symbol

src which is the attribute name

comma

2 single quotes

login is written inside the single quotes

The yellow highlight indicates we have located the iframe.

Copy the value

Now, let's switch to the iframe. driver dot findElement By xpath then paste the value.

Let's Run.

We see phone number (123) 456-7890.

switchToFrames Passed.

frame (WebElement)

Another way of writing the same Test Script is to declare a WebElement. I am going to Copy and Paste this code. Write WebElement, name the object reference login, next we find the iframe and assign it to login.

Since we already found iFrame, the last step is to switch to the iframe but this time we will use the object reference login.

This is just another way of writing the same statement for frame (WebElement). The previous way used 1 line to find and switch to an iframe. This way uses 2 lines to find iframe then switch to iframe. Let's change the phone number to an email so when it runs We see a different value. ABC@Email.com.

Both methods switchToFrames and switchToFrames_WebElement Passed using frame(WebElement). That's how switch to a frame using WebElement. Thank You and Next, I will show you how to switch to a frame using frame (String).

frame (String)

Next, is frame using (String) which selects a frame by its name or id.

Let's walkthrough the Application Under Test (AUT). First, we are going to load Chase then add a wait statement for this Welcome box. Finally, we enter a username. (Go to Eclipse)

The Method Name is switchToFrames_String. Let's the load application driver.get("https://www.chase.com"). Next, we find the Username element driver.findElement By id sendKeys @Test3400. Inspect the element.

Copy the value for id. Paste the value, add a Wait Statement before entering the Username, a dynamic explicit wait for 5 seconds. Next wait until userId-input-field is visible then run. There's no information in the Username text field. There's no information because the Welcome Sign In box has View Frame Source. View Frame Source indicates this box is a frame or iframe. Let's inspect. Scroll up to iframe tag. In addition, to the src attribute, notice the iframe has an id and name attribute with a value of logonbox. The previous switch example used a value within src but we are going to use logonbox for name and id.

Let's go back to Eclipse and perform our switch. As expected, we see a failure. This time, let's use a String for switching. driver.switchTo.frame (String)

The description shows select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID. This means, we can use the name attribute or id attribute but name ranks higher than id. However, the iframe we are switching to has the same name and id. Therefore, it does not apply for this scenario.

Add logongbox within double quotes then run. This time username has information and switchToFrames_String Passed. That's it and Thank You for watching Switch to a frame using String. Next, is switch to a frame using index.

frame (index)

Next, is frame using index which selects a frame by an index.

Let's walkthrough the Application Under Test HTML Source Inline Frames. Scroll down to the Interaction section and we see 2 boxes. Click Me and tick… tick.. If we click the box named Click Me then the box named tick tick changes to BOOM! These boxes will be used for our test. Inspect both boxes and we see iframe. We need the index for Click Me.

CTRL + F, 2 forward slashes, iframe, and there are 12 frames. Search, Search. The Click Me iframe is 3 of 12. Therefore, the index is 2 because indexes start at 0.

Go to Eclipse. The method is switchToFrames_Index. Let's load the page driver.get Next, we are going to scroll down the page using a JavascriptExecutor. Import the JavaScriptExecutor class.

0 and 1250 are coordinates. The first number 0 is an x-axis which scrolls horizontally (left or right). A negative number scrolls to the left while a positive number scrolls to the right. In our scenario, I modified the y-axis from 0 to 1250 to scroll down the page. A negative number scrolls up the page. Now, we switch to the frame by writing driver.switchTo().frame. The description for index states "Select a frame by its (zero-based) index". Our index is 2 because we need the 3rd option - 3 of 12.

The last step is to click the hyperlink named Click Me. Before executing this Test Method, let's see the exception by commenting out the switch method then Run. Fail. NoSuchElementException. Unable to locate element. Which element? The Click Me hyperlink.

Let's uncomment the switch statement and Run again. We see BOOM! replaced tick… tick… and the Test Method PASSED.

Summary

We covered 3 ways for switching to a frame. The most reliable way is to use frame WebElement. Next is frame name or id. We can use frame index but it's the least reliable way for switching to a frame. Personally, I do not recommend index because the application can change and add more frames or iframes. As a result, your index can change and your Test Script fail

That's it and Thank You for watching How To Switch To A Frame.

The Transcript, Presentation, and Code will be available on github at https://github.com/RexJonesII/Selenium4Beginners and http://tinyurl.com/Switch-Methods-Selenium