Rex
Jones II

Selenium Switch Methods

Table of Contents

Introduction

Hello and Welcome, My name is Rex Jones II. In this video, we are going to switch to a window using 3 Selenium Switch Methods. Those 3 window methods are getWindowHandle(), getWindowHandles(), and switchTo().window.

The getWindowHandle() method gets the current window handle and the getWindowHandles() method gets all the window handles. Window Handle is a unique alphanumeric id assigned to each window. We use that unique id for the 3rd method switchTo().window(). The switchTo().window method switches focus between the windows.

Demo

Let's use Tools QA for the AUT. We are going to get the window handle of this main window, click the New Browser Window button, and get the window handles of both windows, then switch to the 2nd window.

In Eclipse, we already have a setup() and teardown() method. Now, let's switchWindows(). The first step is to get the window handle of the main window. driver.getWindowHandle(). We see it returns the current window handle and can be used to switch to this window at a later date. Notice the Data Type is String. Therefore, we assign the window handle to String and the object is mainHandle. Print the id:
sysout System.***out***.println("Main Window ID: " + mainHandle + "\n");

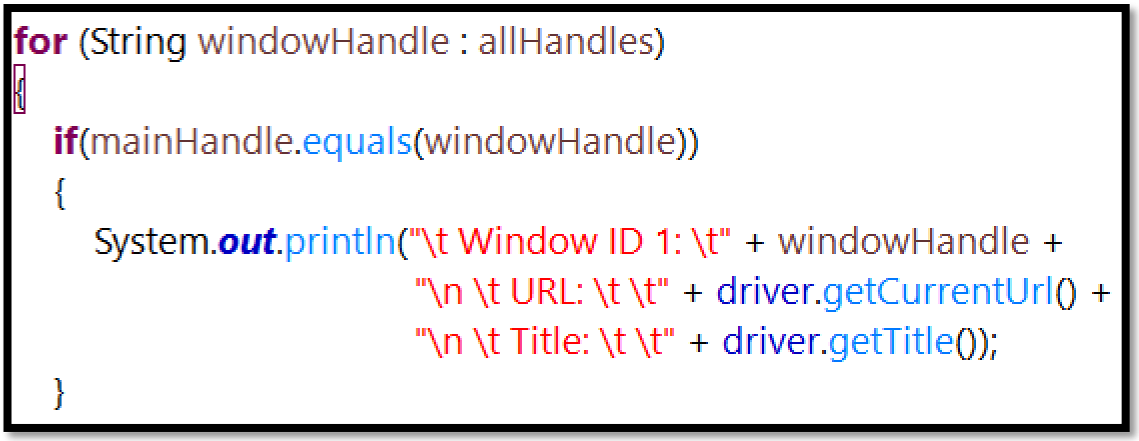
Skip a line and Inspect the button. The value for id is button1. Go back to Eclipse.

Let's scroll down the page before clicking the button.

```
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.scrollBy(0, 250)");
```

Import the classes. driver.findElement(By.*id*("button1")).click(); So far, we have the main window id and clicked the button. Next, we get the window handle of both windows. driver.getWindowHandles(). This method returns a set of window handles which can be used to iterate over all open windows. This data type is Set <String>. Assign the id to Set <String> and the object is allHandles. Import the class. Let's verify we have 2 open windows by printing the size.
Sysout System.***out***.println("Windows Open After Click: " + allHandles.size()); Size will return the number of windows.

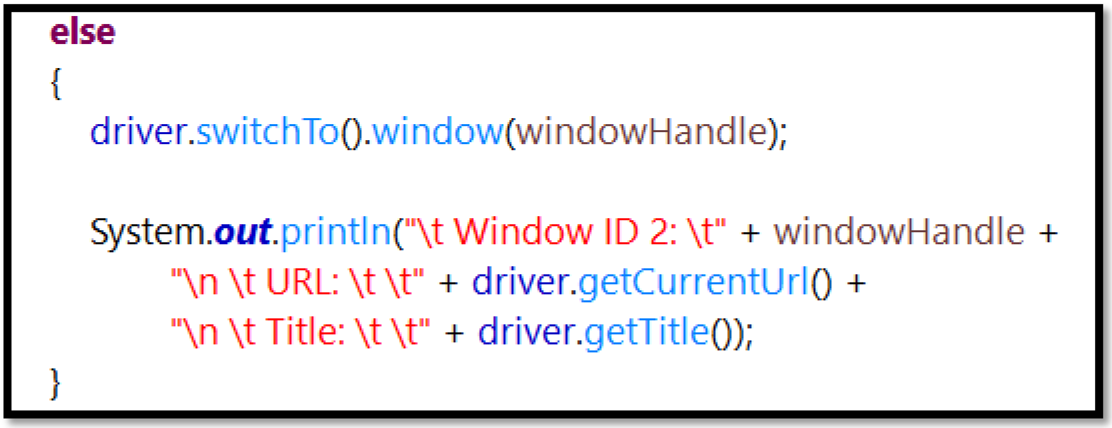
Now, we are going to loop through both windows then switch to the window. Let's use an enhanced for loop. for (CTRL + SPACE) You can watch the previous video to see an enhanced for loop.



```
for (String windowHandle : allHandles)
{
  if(mainHandle.equals(windowHandle))
  {
    System.out.println("\t Window ID 1: \t" + windowHandle +
                        "\n \t URL: \t \t" + driver.getCurrentUrl() +
                        "\n \t Title: \t \t" + driver.getTitle());
  }
```

Print the Window Handle for Window 1 get the URL, and get the Title.
Now, it's time to switch to the window.

else
driver.switchTo().window(windowHandle);
Copy and Paste the print statements then change 1 to 2.



```
else
{
  driver.switchTo().window(windowHandle);

  System.out.println("\t Window ID 2: \t" + windowHandle +
        "\n \t URL: \t \t" + driver.getCurrentUrl() +
        "\n \t Title: \t \t" + driver.getTitle());
}
```

Let's Run. The Main Window has an ID. As expected, there are 2 windows open after clicking the button. Notice, the Main Window and Window 1 have the same ID. They have the same ID because they are the same window. However, Window 1 and Window 2 have different ID's. The ID's are unique and will show a different ID every time we execute. The URL and Title are shown for Window 1 and Window 2. That's it and Thank You for watching How To Switch To A Window.