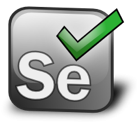
CS570 – Software Testing

**HOP07 – Selenium WebDriver – Handling Alerts & Popup Windows**

11/15/2020 Developed by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)



**Caution**

* If you already finished this module through any CityU School of Technology & Computing (STC) courses, just skim this module and skip it.
* Some version numbers may not match with the newly released ones. If so, stay with the most recent ones.
* This tutorial targets Windows OS and Mac users.
* We cannot explain every step. This cookbook always needs your own creative judgement. Try to solve the problem on your own, after a few tries, if you cannot solve the issue, contact TA for help.

**Learning Outcomes**

* Deeper understanding of Selenium WebDriver.
* Continue performing web app testing using script, specifically able to:
  + Handling different alert boxes and popup windows
  + Send data to alert boxes

**Resources**

* Javapoint.come | Selenium WebDriver - <https://www.javatpoint.com/selenium-webdriver>
* Guru-99 | Selenium tutorial - <https://www.guru99.com/selenium-tutorial.html>

An alert dialog box is a special dialog box that is displayed in a graphical user interface when something unexpected occurred or notification/information is provided, that requires immediate user action.

Alert boxes normally require users to confirm or exit out of the message. Other alert boxes require user inputs. In this HOP, we will learn how we can handle different alerts using Selenium.

1. Move startUsingSeleniumWebDriver project we worked on last week to Module 7 folder.
2. In VSCode, open startUsingSeleniumWebDriver folder under the Hands-on Practice folder, the path should be similar to:

CS570-hop-Hands-on-practice/Module7/startUsingSeleniumWebDriver

**Handling Alerts and Popup Windows**

1. Ope this URL, let’s investigate the website before we write our script: <https://testpages.herokuapp.com/styled/alerts/alert-test.html>

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Try to click different buttons to see what each button does.

**Simple Alert**

The first button represents a simple alert where some information/warning is shown, coming with a confirm button:

Graphical user interface, text, application

Description automatically generated

1. Under webdriver folder, create a new file called “HandleAlert.java”

Text

Description automatically generated

1. Type the following into your HandleAlert.java:

Text

Description automatically generated

Line 20: to find the Element ID, right click on the website, click “Inspect” (for Mac Users) or “View Page Source” (for Windows Users), select element you want to inspect to see the detail:

Graphical user interface, text, application

Description automatically generated

1. Run the test to see result, using the following command:

mvn test -Dtest=HandleAlert.java

(Make sure you are in the right path when running the command. You should be in the startUsingSeleniumWebDriver folder path)

You should see the “Show alert box” button and the alert popup box clicked as expected, afterwards, the webdriver should be closed and quit automatically:

Graphical user interface, text, application, chat or text message

Description automatically generated

**Dialog Box**

Dialog box gives us more options, we can either dismiss or accept the message shown to us:

Graphical user interface, text, application

Description automatically generated

1. Modify your HandleAlert.java file to match the following:

Text

Description automatically generated

As you can see, to dismiss an alert message, we can simply use the built-in dismiss() function, which will automatically looks for the “Cancel” option in the dialog box for us.

This time, in line 47, we pause the execution for 5000 milliseconds, so we have a little more time to investigate our result.

After the “Cancel” button was clicked, scroll down on the website, you should see the following message confirming your script run as expected:

**Graphical user interface, text, application, chat or text message

Description automatically generated**

1. Run the test to see result, using the following command:

mvn test -Dtest=HandleAlert.java

**Send data to alert box**

1. This time we will input some data into the popup window. Modify your HandleAlert.java file to match the following:

**Text

Description automatically generated**

1. Run the test to see result, using the following command:

mvn test -Dtest=HandleAlert.java

**Nothing really happened for the “Show Prompt box” script, as we only clicked on the button, and did not change the user input, nor accept or dismiss the alert box.**

**Challenge: Now, let’s change the content of the user input from the default content “change me” to “YOUR NAME clicked” (Change “YOUR NAME” to your real name”), then click “OK” to confirm.**

**Graphical user interface, text, application

Description automatically generated**

After running your script, you should expect similar result as following (instead of “Kim clicked”, it should be “Your Name clicked”)

Graphical user interface, text, application, chat or text message

Description automatically generated

*Hint: After capturing the text in the alert box, use sendKeys() method to send text to the alert box.*

**Submit your work:**

In VSCode terminal, Type the following command:

* git add . (to copy all changes you have made)
* git commit -m “Submission for Module 7 – Your Name” (To add a message to your submission)
* git push origin master (to upload your work to Github)