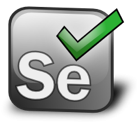
CS570 – Software Testing

**HOP05 – Testing Web App using scripts & Selenium WebDriver - Continued**

11/01/2020 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.

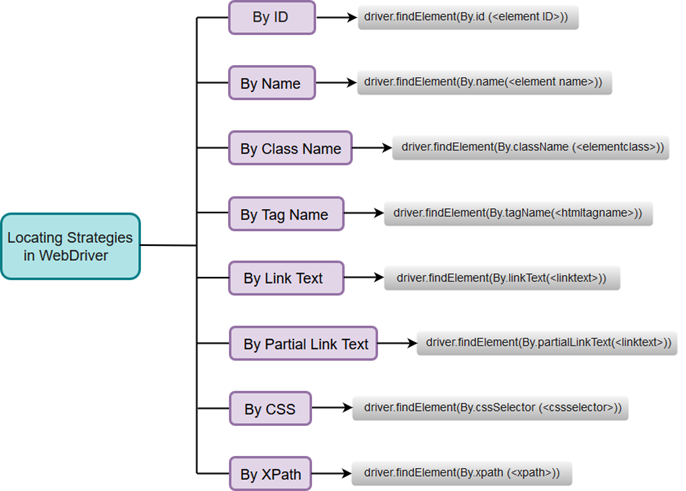
**Learning Outcomes**

* Deeper understanding of Selenium WebDriver.
* Continue performing web app testing using script, specifically able to:
  + Automate locating and filling out forms
  + Automate locating and choosing values from drop-down menu

**Resources**

* Javapoint.come | Selenium WebDriver - <https://www.javatpoint.com/selenium-webdriver>

Last week, we have learned how to locate elements of website using WebDriver, this week, let’s continue. Below are several strategies we can do so:



Last week, we have located elements using ID. This week, let’s locate by Name.

1. Move startUsingSeleniumWebDriver project we created last week to Module 5 folder.
2. In VSCode, open startUsingSeleniumWebDriver folder under the Hands-on Practice folder, the path should be similar to:

CS570-hop-Hands-on-practice/Module5/startUsingSeleniumWebDriver

**Locating and Filling text box**

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

A picture containing text

Description automatically generated

1. Type the following into your LocateByName.java:

Text

Description automatically generated

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

mvn test -Dtest=LocateByName.java

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

You should see:

A picture containing table

Description automatically generated

1. We have filled out the form, let’s add a function to click the “Submit” button:

Text

Description automatically generated

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

mvn test -Dtest=LocateByName.java

The end result should be a blank form, as after clicking the “Submit” button, the form should be refreshed and cleared out:

Graphical user interface

Description automatically generated

**Selecting Values from Drop down menu:**

Notice that we have a drop-down menu in the form we are working on. Let’s write a script to select items from this drop down menu.

1. In the browser, right click, then select “Inspect” (for Mac users) or “View Page Source” (for Windows Users). Hoover over the page you want to click, you will see the attribute information:

Graphical user interface, application

Description automatically generated

1. Edit your LocateByName.java to match the following screen. We will select value called “Selection Item 6” from the drop-down menu:

Text

Description automatically generated

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

mvn test -Dtest=LocateByName.java

You should see:

Graphical user interface

Description automatically generated

1. Finally, close and quit WebDriver using the following scripts:

Graphical user interface, text, application

Description automatically generated

1. Re-run the test using mvn test -Dtest=LocateByName.java, after all tests are run, Chrome browser should be closed automatically.

**REQUIRED:**

**Comment each line of code to explain the program.**

**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 5 – Your Name” (To add a message to your submission)
* git push origin master (to upload your work to Github)