CS570 – Software Testing

**HOP09 – Selenium WebDriver – Retrieve Cookies**

11/30/2020 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:
  + Retrieve session Cookies for later uses.
  + Store session Cookies to a file.

**Resources**

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

In this week’s HOP, we will learn how to retrieve, store Cookies then use it automate login action. But why do we need to handle Cookies?

Each cookie is associated with a name, value, domain, path, expiry, and the status of whether it is secure or not. In order to validate a client, a server parses all of these values in a cookie.

When Testing a web application using selenium web driver, you may need to create, update or delete a cookie.

For example, when automating Online Shopping Application, you many need to automate test scenarios like place order, View Cart, Payment Information, order confirmation, etc.

If cookies are not stored, you will need to perform login action every time before you execute above listed test scenarios. This will increase your coding effort and execution time.

The solution is to store cookies in a File. Later, retrieve the values of cookie from this file and add to it your current browser session. As a result, you can skip the login steps in every Test Case because your driver session has this information in it.

The application server now treats your browser session as authenticated and directly takes you to your requested URL.

Click [HERE](https://www.guru99.com/cookie-testing-tutorial-with-sample-test-cases.html) to learn more about Cookies and Cookie Testing.

1. Move startUsingSeleniumWebDriver project we worked on last week to Module 9 folder.
2. In VSCode, open startUsingSeleniumWebDriver folder under the Hands-on Practice folder, the path should be similar to:

CS570-hop-Hands-on-practice/Module9/startUsingSeleniumWebDriver

**Login into application and store the authentication cookie generated.**

1. Open this URL, let’s investigate the website before we write our script:

<https://testpages.herokuapp.com/styled/cookies/adminlogin.html>

Graphical user interface, text, application, email

Description automatically generated

1. Read the notes provided on the website, and try to login using Admin and AdminPass as username and password. kkki
2. Under webdriver folder, create a new file called “RetrieveCookies.java”

Graphical user interface, text, application

Description automatically generated

1. Type the following into your RetrieveCookies.java:

Text

Description automatically generated

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

mvn test -Dtest=RetrieveCookies.java

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

You should see that username and password are entered for you, the end result should be the admin view:

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

Description automatically generated

Go back to VSCode, look under your project folder, you should now see a file named “Cookies.data”, that stores all the Cookie information of your login:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

With the Cookie information, we can later use them to gain logged in view, and able to skip logging in every time we test our app. Let’s see how this can be done in the next HOP.

No challenge for this section ☺

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