Practice Programming Assignment: Performance Testing with JMeter

Try Again · 0/100 points earned
You must earn 80/100 points to pass.

Deadline Pass this assignment by Jan 28, 11:59 PM IST

Instructions | My submissions | Discussions

Task

This assignment provides an example of how JMeter can be used with Selenium tests to provide measures for timing and assessing the successes and failures of your tests. Attached is a working solution for the Functional Testing with Selenium assignment. You are tasked with running these tests or others of your choosing within JMeter. It's noted here that there will be quiz questions focused on the provided project. The assignment is given here:

Performance_Testing_JMeter_Assign
ZIP File

Setup

In order for Selenium to connect to a web application, it will need to make use of a browser driver to send information to and receive information from the application. Browsers are constantly being updated and improved and this means that functionality can change at a moment's notice. Firefox alongside GeckoDriver is fairly lenient in what versions you decide to use; however, Chrome requires that the major version of the ChromeDriver always matches the major version of the Chrome browser. Browser updates can happen silently, so keep these points in mind if problems arise.

This assignment was designed for a Windows environment. If you are using a different environment, please let us know if any problems occur so that we can resolve issues as soon as possible. Please look below for any variations or alterations necessary to ensure your code works correctly in a different environment.

To set up your environment for either Firefox or Chrome, follow the below instructions.

Firefox:

1. Download the recommended version of Firefox with the Firefox Installer found here https://www.mozilla.org/en-US/firefox/new/

2. Download the latest release of GeckoDriver from https://github.com/mozilla/geckodriver/releases for your environment.

3. Place GeckoDriver in the "/drivers" directory of this project. The environment is set to understand the Windows installation. If you are not using Windows, you will need to alter Line 26 of "src/main/java/web_testing_app/WebTesting.java" to the name of the file in your "/drivers" directory.

Chrome:

1. Download the recommended version of Chrome from https://www.google.com/chrome/.

2. Download the recommended version of ChromeDriver from https://chromedriver.chromium.org/downloads for your version of Chrome.

3. Place ChromeDriver in the "/drivers" directory of this project. The environment is set to understand the Windows installation. If you are not using Windows, you will need to alter Line 31 of "src/main/java/web_testing_app/WebTesting.java" to the name of the file in your "/drivers" directory.

Once your environment is set up to run the tests, uncomment the creation of "driver" at the top of each test in the assignment so that the tests use the correct browser. There shouldn't be any necessary variations in how you write your code based on using Firefox versus Chrome. Since we do not control the creation of the Google Calculator and Google Speed Test, it is possible that your tests will not work for unforeseeable and unknown reasons. Please let us know if you notice any failures that indicate a change in these programs. Let it be noted here that the Google Speed Test failing to complete is a normal occurrence when the connection is slow and we hope you consider the effects of this flaky occurrence on the test results.

This assignment is an alternate example of the previous Web Testing Assignment. Please follow these steps to install the JMeter application and run your test classes on this application.

1. Create a new Java application run configuration in Eclipse for Performance_Testing_JMeter_App. Give it the name, "Performance_Testing_JMeter_App" and set the Main class to "performance_testing_app.Main". Run this configuration to ensure that your web driver exists.

2. Install JMeter to a directory of your choice. Remember this location for the following steps.

3. Right-click on the project and export the example WebTestingAssignment project as a runnable jar with your created launch configuration. Select "Copy required libraries into a sub-folder next to the generated JAR".

4. Copy the dependencies generated by this export to the directory <JMeter>/lib/gradle.

5. Export the example project as a static JAR with these selections. Please check the boxes "Export generated class files and resources" and "Export Java source files and resources".

6. Place this generated JAR file in the directory <JMeter>/lib/junit.

7. Now from Eclipse, go to Run → External Tools → External Tools Configuration… Create a new program and name it "JMeter Sample". This program should point to the location "<JMeter>/bin/jmeter.bat".

8. Run the JMeter application through Eclipse and create a new Thread Group. Give it the name "Example Thread Group".

9. Create a new JUnit Request in this thread group. Check the box to search for JUnit 4 annotations. Call this request "setUp" and choose the test method setUp. This is the instantiation function call. Make a similar JUnit Request for performanceTestSpeedTest and performanceTestCalculator. Finally, add a request for tearDownAll in this manner.

10. Create a Response Time Listener for the Thread Group and set the Thread Group to have a Loop Count of say 5. This will loop our thread calls 5 times to give us a good average of the amount of time it takes for our request to finish.

11. Run the thread group and view the graph created by your tests. The start of the graph for each request can be ignored as each of these requests is not running at the same time and JMeter attempts to find an "average" value for points that do not exist.

Deliverable

There is currently no deliverable for this assignment. Please complete the Functional Testing with Selenium Quiz found in this lesson and linked below.

How will this be graded?

You must complete each problem as it is stated in Specification.txt and on your Assignment.java. Hints are given to help you solve the problem given a limited understanding of the web application itself.

After you follow the above steps and gain an understanding of what this assignment is accomplishing, complete the below quiz.

https://www.coursera.org/learn/web-mobile-testing/quiz/p6TTq/performance-testing-with-jmeter

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submissions" tab.

GeckoDriver (Firefox): https://github.com/mozilla/geckodriver/releases

ChromeDriver (Chrome): https://chromedriver.chromium.org/downloads