

JMETER AND WEBDRIVER – WHY WOULD YOU WANT TO COMBINE THEM?

What is JMeter good at?



The Apache Foundation states the following about JMeter:

The **Apache JMeter™** desktop application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

So in other words, it's a tool you can use to generate functional load on an application or a platform. This also immediately describes what it is good at: generating load. Yes if implemented well you not only generate dumb-load but also hit the functional application layers with the tool. But the basic function of JMeter is aimed at generating load and measuring the (server) response times during this load.

What does JMeter NOT do?

Despite being capable of generating functional load, JMeter does not render pages nor is it very well equipped to execute embedded JavaScript (it is simply not equipped to do that actually), therefore JMeter will not tell you anything (out of the box) about the render times of pages. Especially not about render times when the server is heavily overloaded by your scripts!

What is WebDriver good at?

SeleniumHQ gives a wonderful description of what Selenium (or nowadays WebDriver) does:



Selenium automates browsers. That's it! What you do with that power is entirely up to you.

Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that.

In short, what WebDriver does is just about anything that happens within your browser. It does render pages, it does execute JavaScript, it retrieves the pages as if an actual human was clicking on a website. So for fully functional automated testing (or checking to stick with the more correct terminology) WebDriver is perfect.

What does WebDriver NOT do?

Well, it is not quite good at generating load. Since WebDriver basically requires a browser (yes, it is possible to run it headless of course) it is very difficult to generate multiple (virtual) users. That would require a bunch of browsers to start up, when talking about 10 users that may seem feasible, however when talking about generating real load (say several 1000's concurrent users) a bunch of browsers becomes a lot more difficult to arrange.

Why combine them then?

The logical question then indeed is, why would you combine them? Below I have set out 3 clear reasons why combining JMeter and WebDriver scripts can be an excellent idea.

- Impact of server-side load on render-times;

When the load on a server increases, the response times of various parts of a web application may increase as well. These increased response times can have implications on the render time of the web application. For example: a web application heavy with Ajax requests is put under load, the server response times increase, this may result in all Ajax-requests becoming slower, therefore making the website extremely unfriendly to the end-user. When you just run a JMeter script, this will hardly be noticed, and if you do notice it, you cannot express the impact it *will* have on the user. You can merely speculate about it.

- Impact of server-side load on functional behavior;

Given that the server is experiencing increased load and therefore the business-logic of the application is working hard to handle all requests effectively, it can be safe to say the underlying database may also be stressed and therefore responding slower than expected. Slower response times of both application-logic and database requests can result in buggy behaviour of the application. For example incomplete data returned, or worse, a time-out on data or application-logic. How does the application deal with that? How are these errors reported to the end-user? Will the application still function normally within the browser when certain aspects of the application platform are malfunctioning? The best answer to this is by testing the functionality thoroughly while the application is under load. An easy way to test this repeatedly and consistently is by automating these functional tests, for example using (part of) the automated regression test while the servers are under increased load.

- Advantage of screenshots of fully rendered pages and possible errors with the application under load;

As a result of the two points mentioned above, it may be extremely useful for both developers/system engineers and your customer to see errors on the pages affected by the increased server load, such as stylesheets not loading or not loading properly, JavaScript not loading, images missing etc.

Screenshots (or screencaptures in movie format) will help make clear to the customer what the problem is and more importantly how big the impact on the end-user will be.

I have listed 3 reasons why combining JMeter and WebDriver can be a good idea. I'd love to hear your suggestions of more reasons to want to combine the two.