

# 8 Ways to Improve JMeter Scripts

By: Leticia Almeida |  
@lalmeida88

Abstracta

As a professional software testing services company, for some time now we have been working to provide our clients JMeter scripts that enable them to run performance tests quickly and easily with the help of BlazeMeter (with whom we have officially partnered in 2015 and from whom we have received extensive product training).

# INTRODUCTION

We know that these scripts we build are often used by teams that do not necessarily have training in JMeter but still want to be able to make changes to them and understand what they do. For this, we've been working on building scripts that are intuitive for anyone who is not an expert in JMeter and can without difficulty, read, understand, modify, and run them in BlazeMeter.

From our past experiences, we have built an eight-point checklist which we review before delivering a script to a client. This checklist includes a series of best practices which we have made available for you today.

# Checklist: 8 Ways to Improve JMeter Scripts

## 1. Enable debug logging

Often we enable logging in our test elements to debug any errors that may arise in the process of building a script, but once the script is completed and before delivering to the stakeholder, we should remove these elements to prevent them from impacting the performance of the load generator (the machine where JMeter simulates the load). There are cases where the script itself uses many of the resources of the load generator, but if on top of that, it always writes each request and response on disk, we would be adding unnecessary resource consumption.

## 2. Do not add secondary requests or embedded sources

All requests that are characterized as static and that come as an embedded resource in a primary request (images, css files, javascript, etc.) do not necessarily need to be included in the scripts. This is because in JMeter, within the options of an HTTP Request element corresponding to a primary request, we can enable the Retrieve All Embedded Resources option that generates the following behavior:

- It sends the primary request (the HTTP Request in question)
- When getting the response (HTML file), it parses it and searches for all resources that are embedded in the HTML automatically
- For each recognized element, it executes an HTTP Request

This way it will react more accurately than if all secondary requests were previously pre-programmed in the script. Even though this involves more processing for the load generator, we consider it very beneficial for maintaining and testing the best simulation of a real user.

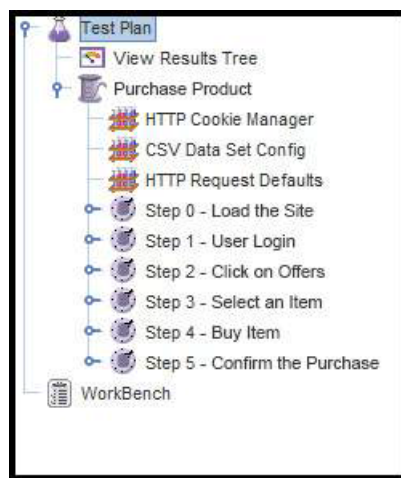
## 3. Add a representative name to the Thread Group

The name that we give the Thread Group element should be representative of the test case or the flow that is being automated. With this, we help the stakeholder to analyze a script that has a name that not only the tester understands, but anyone involved with the test case.

## 4. Group by group actions

All test scripts consist of a series of user actions, such as: Access the site, login, click bids, select an item, buy items, confirm the purchase, etc. All of these actions involve a set of HTTP requests that can be grouped into blocks by using the Transaction Controller to which you can set a name. We recommend that the name be the same action that it encompasses so that the stakeholder can easily trace the steps sequence and the script.

For instance:



Note that we are also following the third checklist item when doing this.

## 5. Check the inclusion of requests to external servers

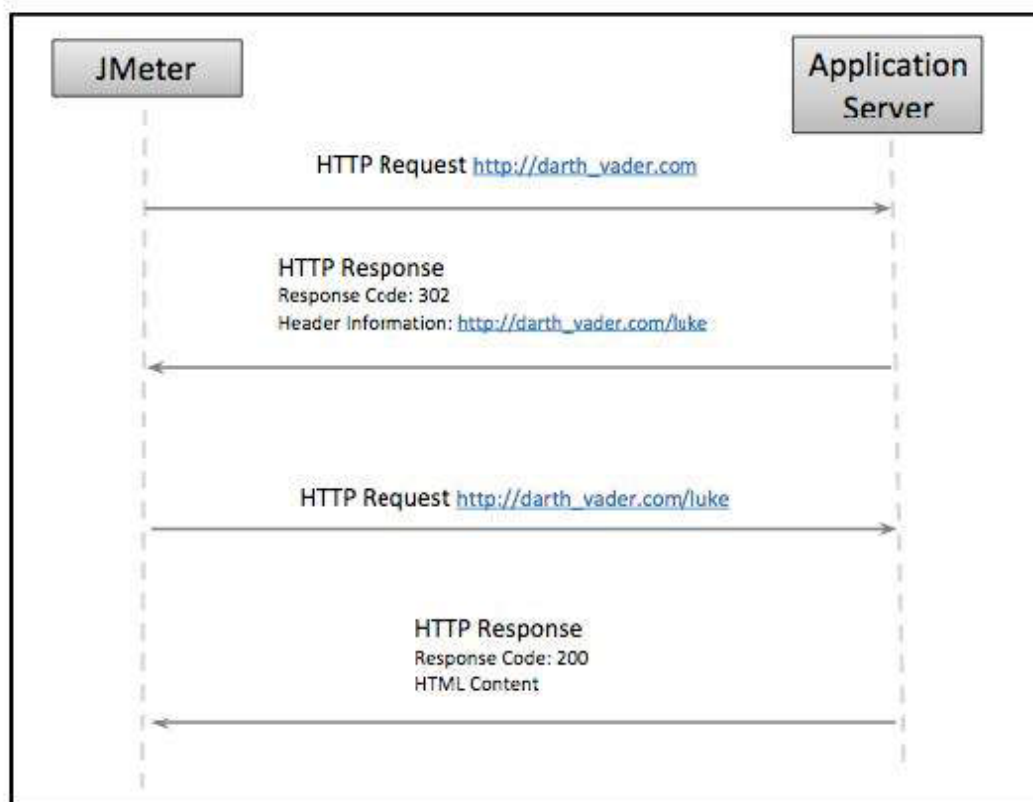
Sometimes, while performing an action on the application, requests are triggered that may not necessarily be to the host of the application. In these, it's always appropriate to reaffirm with the stakeholder if they are interested in analyzing the performance of the host (ie, if the host is also part of the SUT - system under test), because in the case that it is not, the HTTP request can be omitted and does not generate traffic to the server, avoiding requests in our script that are not needed.

Example: banner ads bring things from Facebook, Google Analytics, Twitter, etc.

## 6. Do not include redirects

Some requests make us redirect to a different element, resource or object, indicating that what we are looking for is available in a different place. In this kind of response, when you run an HTTP request, its response code has a 301 or 302 in the header of the response with a parameter called Location with the URL to which the request should be redirected. In JMeter it is not necessary to generate an HTTP Request for these requests because in the HTTP Request order (which we'll call the parent), we can enable the Follow Redirects option to do this task for us.

Here is an example of a redirect (Darth Vader being the parent URL and Luke being the new one):



This makes it easier to handle redirects, and makes the script easier to understand, stopping us from worrying about the twists and turns that can occur after a request. Otherwise, it would be parsing the header of the response by using regular expressions and creating a HTTP Request manually.

## 7. Include validations in all primary HTTP requests

Validations or response assertions in JMeter are the way we validate that the response we are getting from the server is correct. The tool only checks the response code, and if it is not an error 5XX or 4XX, then it will say that it was correct, but might miss many other problems that may occur. In which case it is necessary to make a minimal validation that assures us that we are following the flow we want.

Knowing what to validate in a response is an art, there is no manual to address this issue, but we have written a post on the subject that you can find in the additional resources section of this checklist called, "JMeter response assertions: How to know what to validate in an HTTP response."

**The important thing here is that a response assertion has to be included in every HTTP Request.**

While on the subject of validations, I also would like to add a note about redirects.

When presented with a redirect, the parent request (Darth Vader) must validate what would be the last response, which was triggered by the redirect response chain and not the one the parent requested. This is a detail that JMeter validates automatically, but it's better if we know the behavior under these situations.

## 8. Use HTTP request defaults

All the HTTP Request elements have a place to set up the server and port, among others. For a matter of practicality, it's never good to leave this value hardcoded.

This is because if the script is extensive and we change these values during execution, since it's very common that we automate in a test environment and then execute in different environments (test, pre-production, production, etc), it's a cumbersome task to go request by request changing these values. To give you an idea, in my last project, we had 14 scripts with 40 to 50 requests each.

To prevent this situation, we used the HTTP Request Defaults configuration element. In this element we are able to configure the server and port default to wherever we want the requests to go and thus, if we leave server and port values of all HTTP Requests as blank values, the request will be directed to those specified in the HTTP Request Defaults. It is a simple way for the stakeholder to know how to set these parameters.

---

Thanks for reading our checklist. We are still adding items as we go so we encourage you to share your experiences and help us improve it!

We hope it has been helpful for you.

...



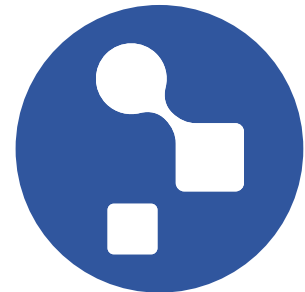
## About the author



Leticia Almeida is passionate about improving the performance of IT systems. As a tester for Abstracta, she has several years of experience in performance testing, where she has taken on several roles analyzing the performance of applications built upon multiple technologies. An expert in load/performance testing with Apache JMeter, Leticia has participated in many projects for our clients like Barnes and Noble and BBVA Financial. She manages to be as efficient as possible when testing, saving time and money for clients. For one in particular, she managed to reduce the time to execute a test that would normally take one hour to just ten minutes.

## About Abstracta

Abstracta is a world leader in quality assurance and testing focused on improving the performance of software applications. With offices in Latin America and Silicon Valley, Abstracta has over nine years of expertise working not only with leading-edge proprietary and open source testing tools, but developing specialized tools for financial, retail and technology including companies such as BBVA financial group, Verifone, GeneXus software, and the largest retail bookseller in the United States.



Formed by passionate computer engineers and PhDs, Abstracta is always looking for innovative ideas and solutions to give customers the best solutions possible. Abstracta develops its own testing tools and are is an active contributor to the open source software testing community. Abstracta has extensive experience using performance testing tools such as Jmeter, OpenSTA, HP LoadRunner, WatiN, LoadUI and more.

## ADDITIONAL RESOURCES

1. 7 Performance Testing Strategy Fallacies Undermining your Testing Strategy via StickyMinds

<http://www.stickyminds.com/article/7-performance-testing-fallacies-undermining-your-test-strategy>

2. How to Use JMeter Assertions in Three Easy Steps via BlazeMeter

<https://blazemeter.com/blog/how-use-jmeter-assertions-3-easy-steps>

3. Introduction to Taurus: An Alternative to JMeter via Abstracta US

<http://abstracta.us/2015/09/07/introduction-to-aurus-an-alternative-tojmeter/>

4. JMeter Performance and Tuning Tips via UBIK Ingenierie

[http://www.ubik-ingenierie.com/blog/jmeter\\_performance\\_tuning\\_tips/](http://www.ubik-ingenierie.com/blog/jmeter_performance_tuning_tips/)

5. JMeter Response Assertions: How to Know What to Validate in a Response Request via Abstracta US

<http://abstracta.us/2016/02/19/jmeter-response-assertions-how>

## FOR MORE INFORMATION

Contact the author

 [leticia.almeida@abstracta.com.uy](mailto:leticia.almeida@abstracta.com.uy)

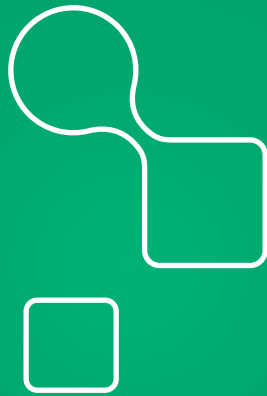
 <https://uy.linkedin.com/in/almeidaleticia/en>

For information about professional testing services,  
please visit [www.abstracta.us](http://www.abstracta.us)  
or contact Sofía Palamarchuk at [sofia@abstracta.us](mailto:sofia@abstracta.us)

For more testing articles and tips visit:  
<http://abstracta.us/knowledge-center>

**Abstracta**  
44 Tehama St,  
San Francisco, CA 94105  
(408) 757-0005

©Abstracta, All rights reserved.



ABSTRACTA.US



©Abstracta