

Group B
ENSE 375 - Project Step 4
April 5, 2021

JMeter Process

Below is the step-by-step process we took to install and experiment using JMeter with Jenkins.

1. Install the Performance Plugin on Jenkins by going into Manage Jenkins -> Manage Plugins and searching Performance under the available plugins tab

Updates

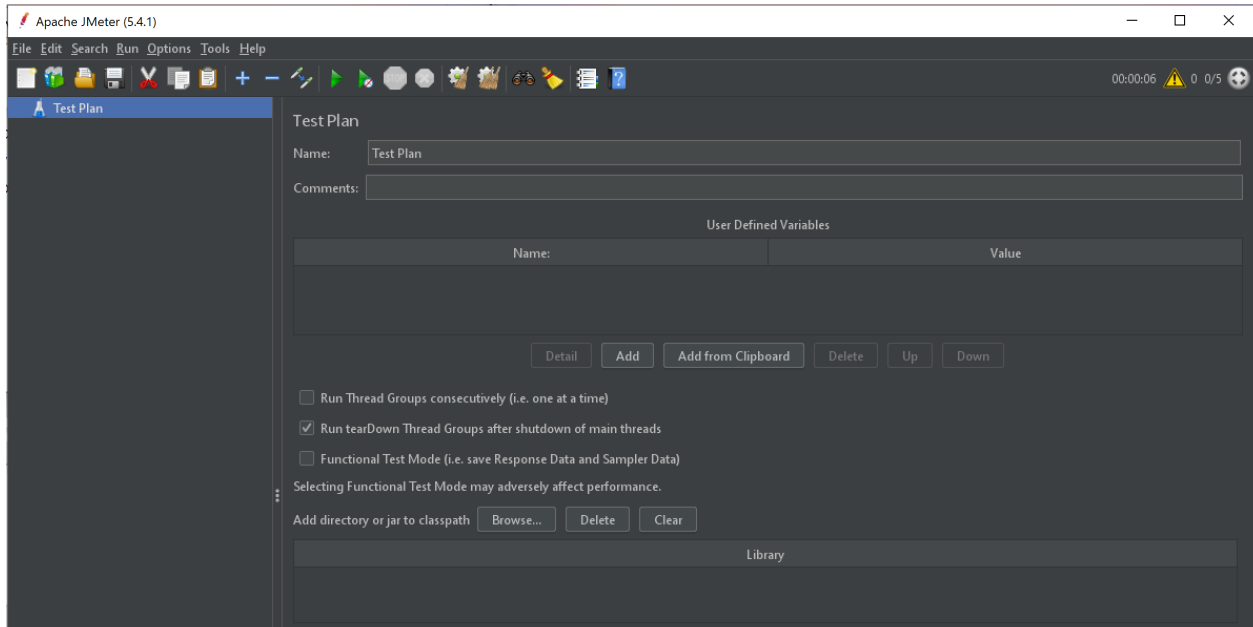
Available

Installed

Advanced

Enabled	Name ↓	Version	Previously installed version	Uninstall
	Performance Plugin			
<input checked="" type="checkbox"/>	<div>This plugin allows tracking performance KPIs, based on results read from popular testing tools (Apache JMeter, JUnit, Taurus).</div>	3.19		<div>Uninstall</div>

2. Download JMeter from http://jmeter.apache.org/download_jmeter.cgi
3. Unzip the JMeter folder and open the jmeter.bat file. You should see an interface similar to the one below



4. Add a thread group to the test plan by right clicking on the test plan and choosing Add -> Threads(Users) -> Thread Group
5. Set the following Thread Group settings
 - Number of Threads: The number of users (threads) you are testing.
 - Ramp-up Period: How quickly (in seconds) you want to add users. For example, if set to zero, all users will begin immediately.
 - Loop Count: How many times the test should repeat.

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: ☐ Infinite

6. Add a HTTP request to a website (We used the BlazeMeter demo website - blazedemo.com) to the thread group by right clicking the thread group and choosing Add -> Sampler -> HTTP Request
7. Set the following HTTP Request settings
 - Name: Provide a custom name or simply leave it named the default "HTTP Request".
 - Server Name or IP: The server's DNS name or IP address. In this case - blazedemo.com.

HTTP Request

Name:

Comments:

▲ ▼

Basic Advanced

Web Server

Protocol [http]: Server Name or IP:

HTTP Request

▼ Path:

8. Add timer to HTTP Request by right clicking the HTTP Request and choosing Add -> Timer -> Constant Timer. The constant timer will determine how many milliseconds to wait between requests. The default is 300.
9. Add a listener to the thread group by right clicking the thread group and choosing Add -> Listener-> View Results Tree
10. Save the test plan and it will be saved as a .jmx file
11. Run the test plan by clicking the green arrow at the top. The result can be viewed on the listener.
12. Add another HTTP request to the thread group by right clicking the thread group and choosing Add -> Sampler -> HTTP Request. This one will test accessing the blazedemo.com/reserve.php page by entering "Boston" and "London" as parameters into the From and To selection menus.

HTTP Request

Name:

Comments:

▲ ▼ ...

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: Port Number:

HTTP Request

▼ Path: Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

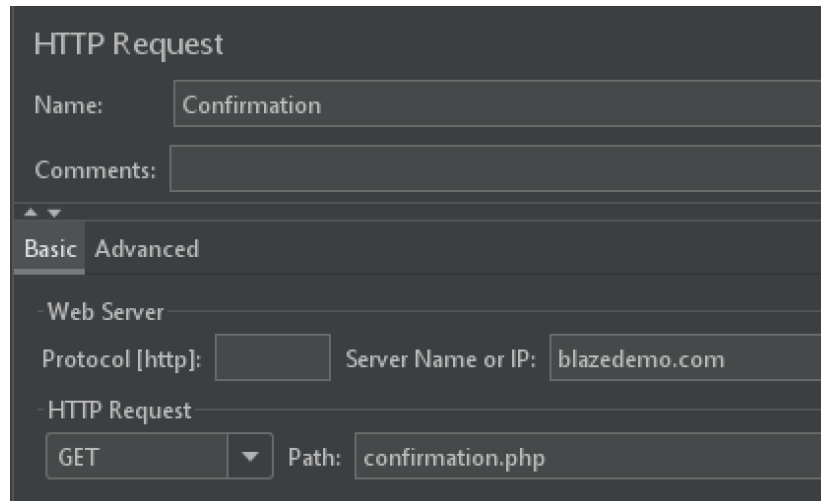
Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
fromPort	Boston	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
toPort	London	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

13. Add a listener to the thread group by right clicking the thread group and choosing Add -> Listener-> View Results in Table

14. Add another HTTP request to the thread group by right clicking the thread group and choosing Add -> Sampler -> HTTP Request. This one will test accessing the `blazedemo.com/confirmation.php` page as if the user had selected and purchased a flight.



The screenshot shows the 'HTTP Request' configuration window. The 'Name' field is set to 'Confirmation'. The 'Comments' field is empty. The 'Basic' tab is selected, showing the 'Web Server' section with 'Protocol [http:]' set to 'http' and 'Server Name or IP' set to 'blazedemo.com'. The 'HTTP Request' section shows the method set to 'GET' and the 'Path' set to 'confirmation.php'.

15. Add timer to HTTP Request by right clicking the HTTP Request and choosing Add -> Timer -> Constant Timer.
16. Add an assertion to the HTTP Request by right clicking the HTTP Request and choosing Add -> Assertions -> Response Assertion. An assertion will let you check for errors. In this case, we will test if the webpage displays the expected text. The confirmation page should display the following text.

Travel The World home



Thank you for your purchase today!

Setting the following assertion settings will verify that the text is present on the screen.

Response Assertion

Name:

Comments:

Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only ☐ JMeter Variable Name to use

Field to Test

☒ Text Response ☐ Response Code ☐ Response Message ☐ Response Headers

☐ Request Headers ☐ URL Sampled ☐ Document (text) ☐ Ignore Status

☐ Request Data

Pattern Matching Rules

☐ Contains ☐ Matches ☐ Equals ☒ Substring ☐ Not ☐ Or

Patterns to Test

Patterns to Test	
1	Thank you for your purchase today!

17. Save and run the test plan. The results are as follows:

View Results in Table

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	04:07:56.072	Thread Grou...	BlazeDemo	876	✓	3447	236	278	167
2	04:07:56.279	Thread Grou...	BlazeDemo	891	✓	3447	236	181	59
3	04:07:56.469	Thread Grou...	BlazeDemo	741	✓	3447	236	189	61
4	04:07:56.948	Thread Grou...	Reserve	319	✓	5975	318	110	0
5	04:07:56.670	Thread Grou...	BlazeDemo	690	✓	3447	236	205	68
6	04:07:56.876	Thread Grou...	BlazeDemo	643	✓	3447	236	196	60
7	04:07:57.211	Thread Grou...	Reserve	309	✓	5975	318	109	0
8	04:07:57.171	Thread Grou...	Reserve	417	✓	5975	318	128	0
9	04:07:57.360	Thread Grou...	Reserve	344	✓	5975	318	155	0
10	04:07:57.568	Thread Grou...	Confirmation	339	✓	4485	268	111	0
11	04:07:57.520	Thread Grou...	Reserve	393	✓	5975	318	108	0
12	04:07:57.826	Thread Grou...	Confirmation	365	✓	4485	268	113	0
13	04:07:58.008	Thread Grou...	Confirmation	301	✓	4485	268	109	0
14	04:07:57.900	Thread Grou...	Confirmation	662	✓	4485	268	116	0
15	04:07:58.220	Thread Grou...	BlazeDemo	456	✓	3447	236	119	0
16	04:07:58.220	Thread Grou...	Confirmation	495	✓	4485	268	119	0
17	04:07:58.500	Thread Grou...	BlazeDemo	311	✓	3447	236	112	0
18	04:07:58.677	Thread Grou...	Reserve	324	✓	5975	318	111	0
19	04:07:58.625	Thread Grou...	BlazeDemo	405	✓	3447	236	136	0
20	04:07:58.869	Thread Grou...	BlazeDemo	369	✓	3447	236	118	0

☐ Scroll automatically? ☐ Child samples? No of Samples 45 Latest Sample 575 Average 403 Deviation 142

```

44 2021-04-05 04:07:56,568 INFO o.a.j.t.JMeterThread: Thread started: Thread Group 1-5
45 2021-04-05 04:08:01,656 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-1
46 2021-04-05 04:08:01,656 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-1
47 2021-04-05 04:08:01,913 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-3
48 2021-04-05 04:08:01,913 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-3
49 2021-04-05 04:08:02,433 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-4
50 2021-04-05 04:08:02,433 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-4

```

Using JMeter test plan with Jenkins:

1. Add the "jmeter.save.saveservice.output_format=xml" line to the bottom of the user.properties file (this is located in /bin folder of your JMeter installation)
2. Create a freestyle project with Jenkins and in the Build section, add an "Execute Windows batch command" step. In the command area, enter the following command to pull your saved test plan .jmx file from your JMeter \bin folder as well as to save your Test.jtl file to the same folder.

Build



Execute Windows batch command



Command

```
C:\Users\kfide\Downloads\apache-jmeter-5.4.1\apache-jmeter-5.4.1\bin
\jmeter.bat -Jjmeter.save.saveservice.output_format=xml -n -t C:\Users
\kfide\Downloads\apache-jmeter-5.4.1\apache-jmeter-5.4.1\bin
\JMeterPluginTest.jmx -l C:\Users\kfide\Downloads\apache-jmeter-
5.4.1\apache-jmeter-5.4.1\bin\Test.jtl
```

See [the list of available environment variables](#)


Advanced...

Add build step ▼


3. Then add a “Publish Performance test result report” post build action which will use the Performance plugin. Set the source data to the path of your “Test.jtl” file and leave all other settings as default. This can be seen below.

Post-build Actions


Publish Performance test result report

Source data files (autodetects format): 

C:\Users\kfide\Downloads\apache-jmeter-5.4.1\apache-jmeter-5.4.1\bin\Test.jtl

Regex for included samplers 

☒ Show Trend Graphs

Select evaluation mode 

☐ Expert Mode ☒ Standard Mode

Standard Mode

Select mode:

☐ Relative Threshold ☒ Error Threshold

Use Error thresholds on single build:

Unstable

Failed

Advanced...


Use Relative thresholds for build comparison:

(-) (+)

Unstable % Range

Failed % Range

☐ Compare with previous Build ☒ Compare with Build number

Compare based on 

Expert Mode

Constraint settings

☐ Ignore Failed Builds

☐ Ignore Unstable Builds

☐ Save constraint log to workspace

JUnit output file (optional)

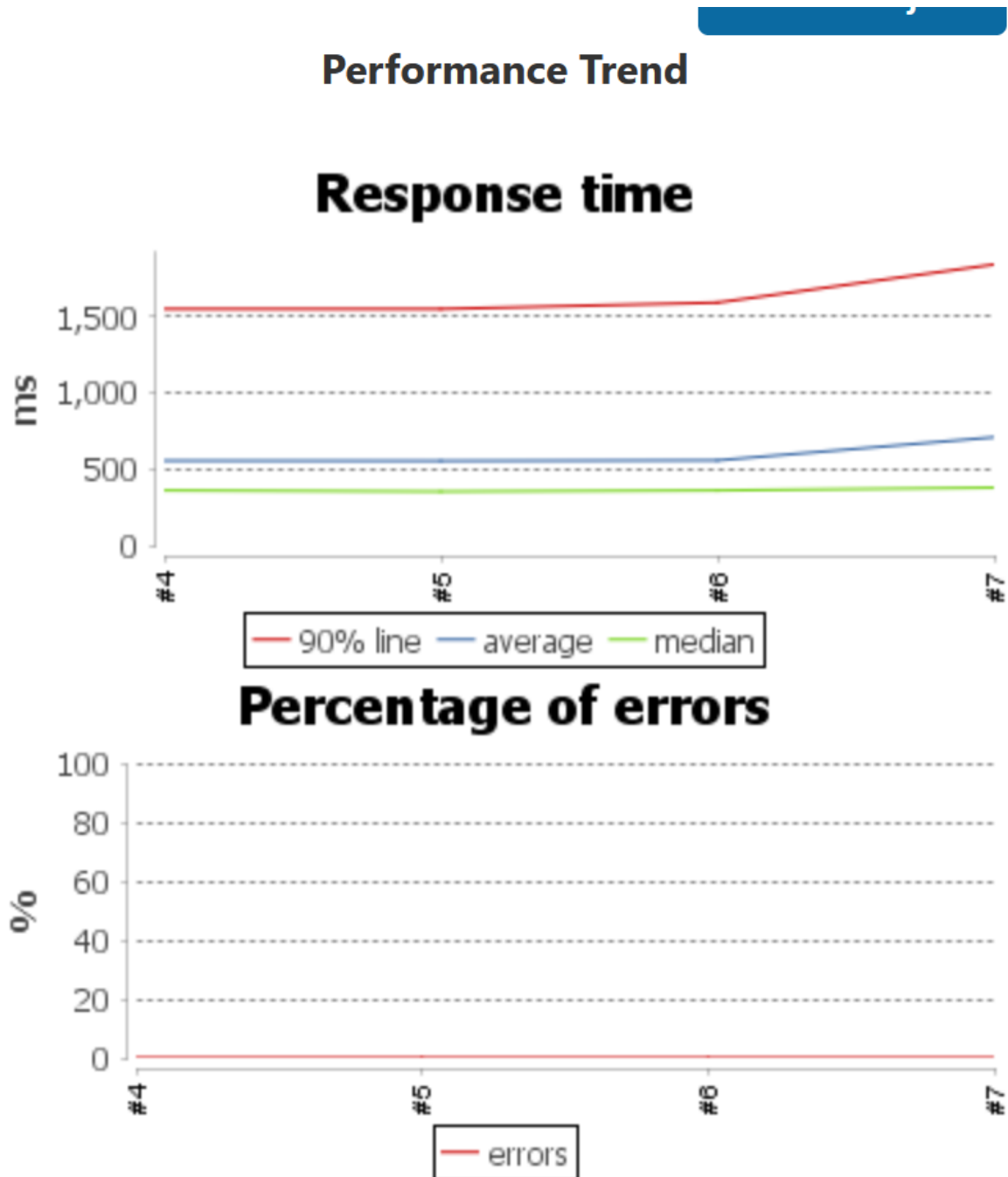
Constraints

Add a new constraint ▾

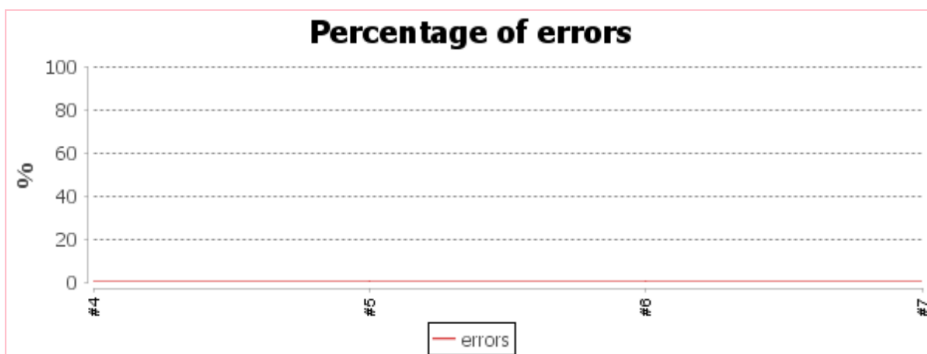
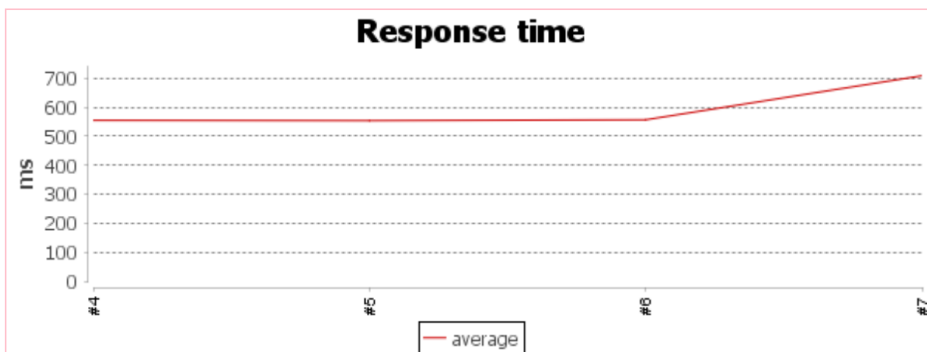
Advanced...

Id action

4. Save your project configuration and “Build Now”. Initially your Performance Trend charts will be empty but after a couple builds you will see results displayed. The charts display average, median and 90 percentile values along with the errors count. You will also receive .xml artifacts. Below is a screenshot of our Performance Trend charts after a few builds.



Performance Breakdown by URI: Test.jtl



[Response time trends for build: "JmeterTestProject #7"](#)

Comparison with previous build

URI	Samples	Average (ms)	Min(ms)	Median(ms)	Line 90.0(ms)	Max(ms)	Http Code	Errors (%)	Average (KB)	Total (KB)
BlazeDemo	75 ⁺¹⁵	1180 ⁺²⁶⁰	303 ⁰	428 ⁺¹⁷	2300 ⁺⁵⁸	5607 ⁺³¹⁴⁴	200	0.0 % ^{0.0 %}	3.37 ^{0.0}	252.47 ^{+50.49}
Confirmation	75 ⁺¹⁵	492 ⁺¹³¹	297 ⁰	348 ⁺⁵	650 ⁺²³²	1936 ⁺¹³⁵⁰	200	0.0 % ^{0.0 %}	4.38 ^{0.0}	328.49 ^{+65.7}
Reserve	75 ⁺¹⁵	455 ⁺⁶⁶	302 ⁰	372 ⁺¹²	627 ⁺¹¹⁹	1891 ⁺¹²³¹	200	0.0 % ^{0.0 %}	5.83 ^{0.0}	437.62 ^{+87.52}
All URIs	225 ⁺⁴⁵	709 ⁺¹⁵²	297 ⁰	378 ⁺¹⁸	1843 ⁺²⁵⁰	5607 ⁺³¹⁴⁴		0.0 % ^{0.0 %}	4.53	1018.58

Our .jmx, .jtl, and our .xml artifact files have been included in our github directory.