

Load Testing with JMeter

What is Load Testing:

Load testing is a kind of performance testing that helps determining how the application behaves when multiple users access it simultaneously.

Some Load testing tools:

Apache JMeter, Load Runner, Web Load, Load UI, Neo Load.

What is JMeter:

JMeter is one of the JAVA tools which is used to perform Load Testing client/server applications. ApacheJMeter is open source software, a 100% pure JAVA desktop application designed to load test functional behavior and measure performance of the application.

Why use JMeter:

Now it is one of the most preferred tools because it supports multiple load injectors managed by a single controller. It is also highly portable and supports all the JAVA based apps. Now JMeter has lots of advantages when it comes to performance testing. It supports in windows, Linux, Mac OS, Ubuntu.

So, main points are:

Open source, built in java platform, highly extensible and platform independent, user friendly, comprehensive GUI which helps to configure element and comprehensive documentation.

JMeter installation:

JMeter is an open source testing tool and there are no such findings, but we have to keep certain things in mind before installing this particular tool.

Prerequisites to install JMeter:

Because JMeter is pure Java desktop application, it requires a fully compliant JVM 6 or higher. You need to download the latest version of Java SE development kit.

Install:

In order to download JMeter we need to go the official website that is http://jmeter.apache.org/download_jmeter.cgi

From here we will be downloading our latest version of Apache JMeter.

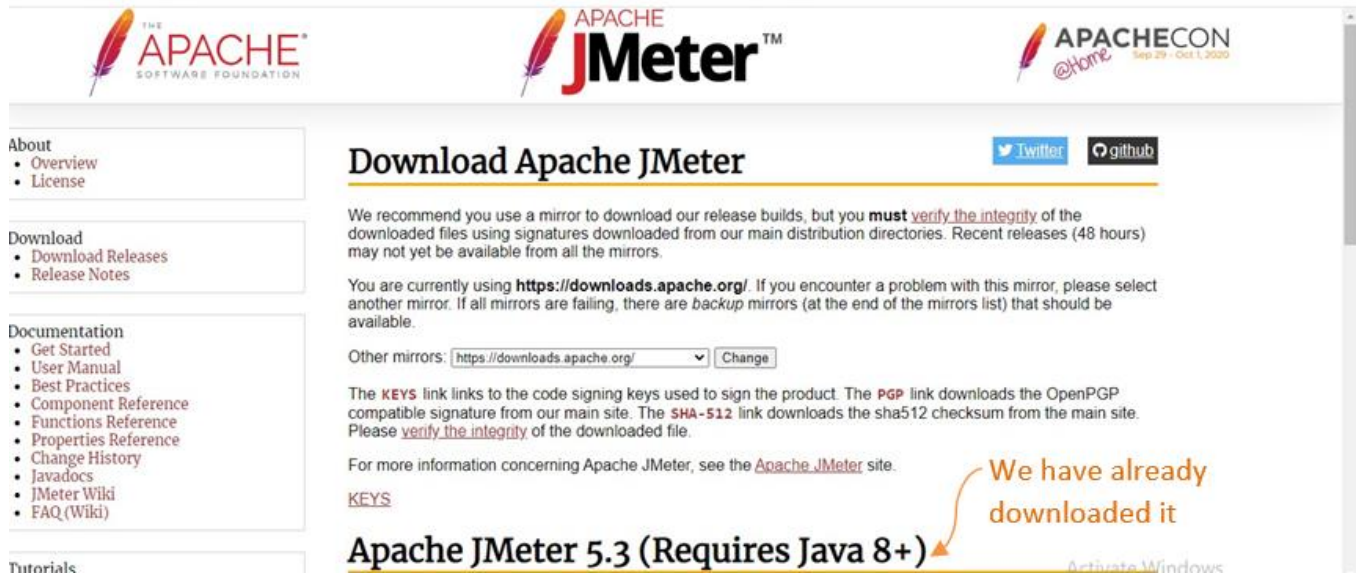


Figure1:Apache JMeter official website

Now we will download the zip file, and this is very simple process.



Figure 2: Download JMeter file

Now open and run the file. This is Zip file. So, we need to extract this file. After extracting we will go inside the **apache-jmeter-5.3** folder.

Inside the folder there is another folder that is **bin** folder.

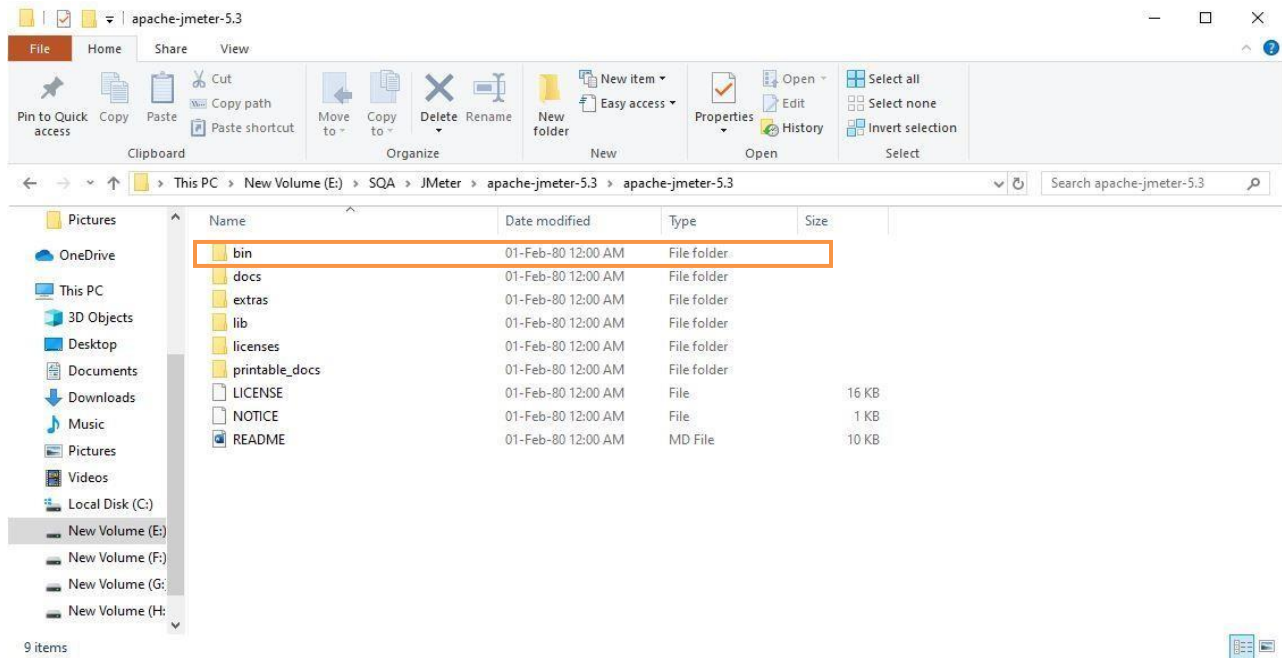


Figure 3: Bin folder inside Apache JMeter

Once We enter the bin folder here, we can see there is an application file named as **Jmeter**.

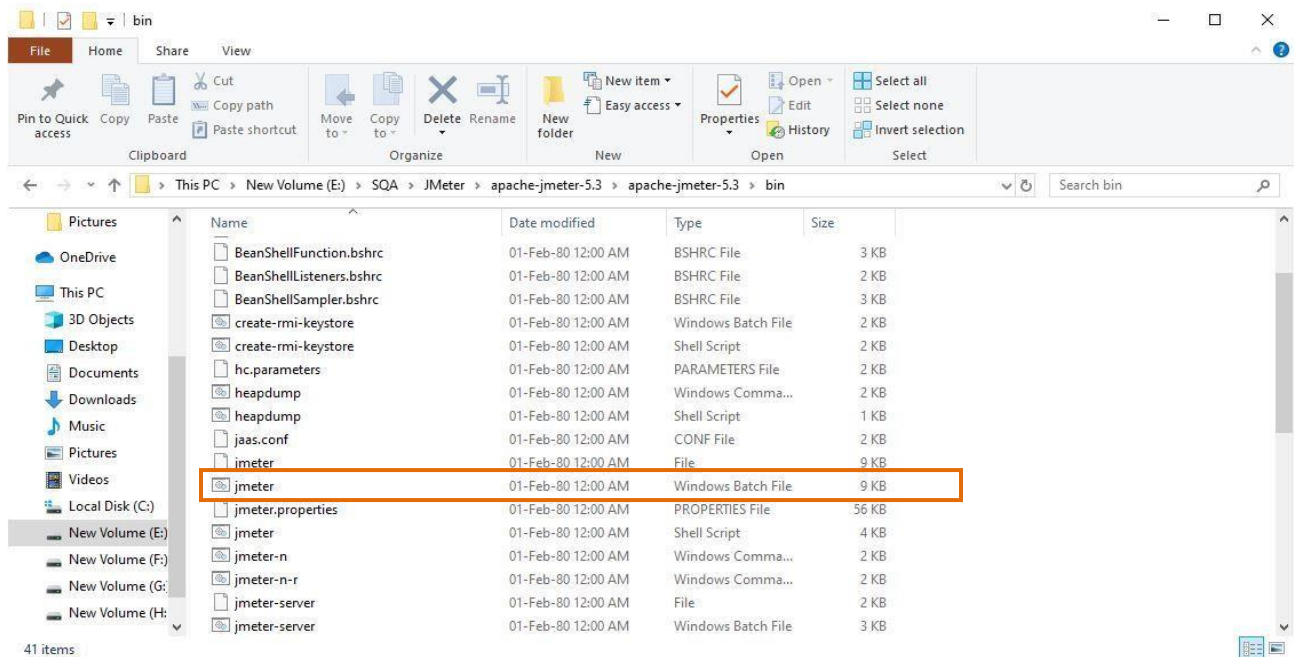
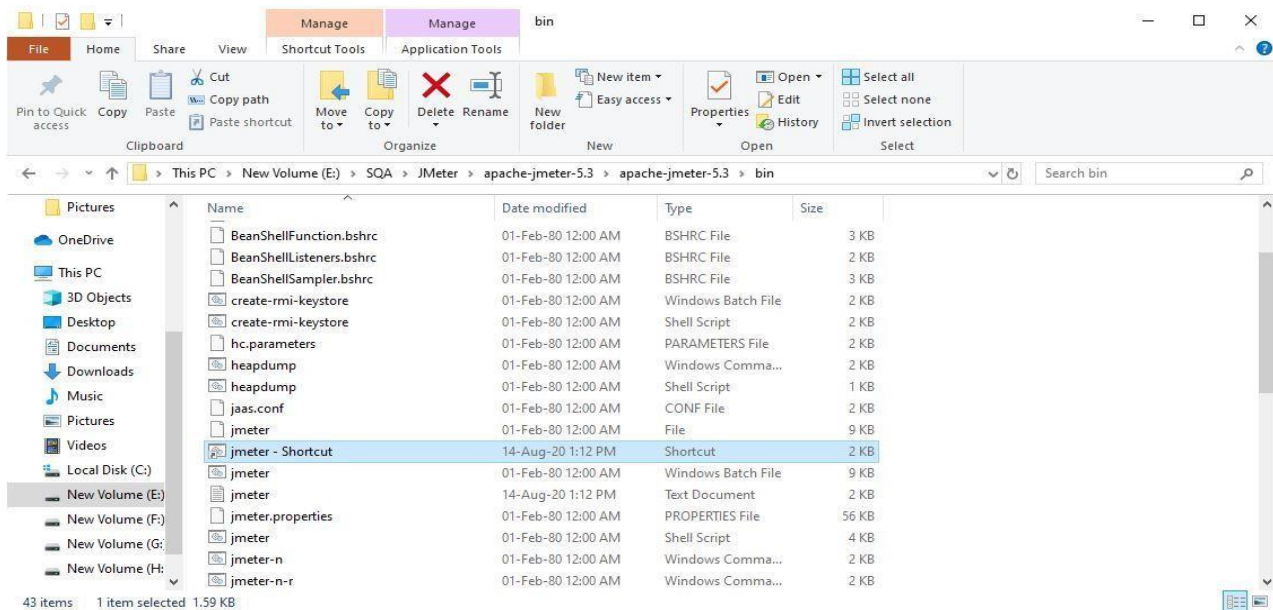


Figure 4: jmeter inside the bin folder

Now we will create a shortcut of this file and copy the shortcut file to our desktop to run it easily



Figurer 5: Shortcut of Jmeter file

Now double click on this file, wait a bit, then our apache jmeter will

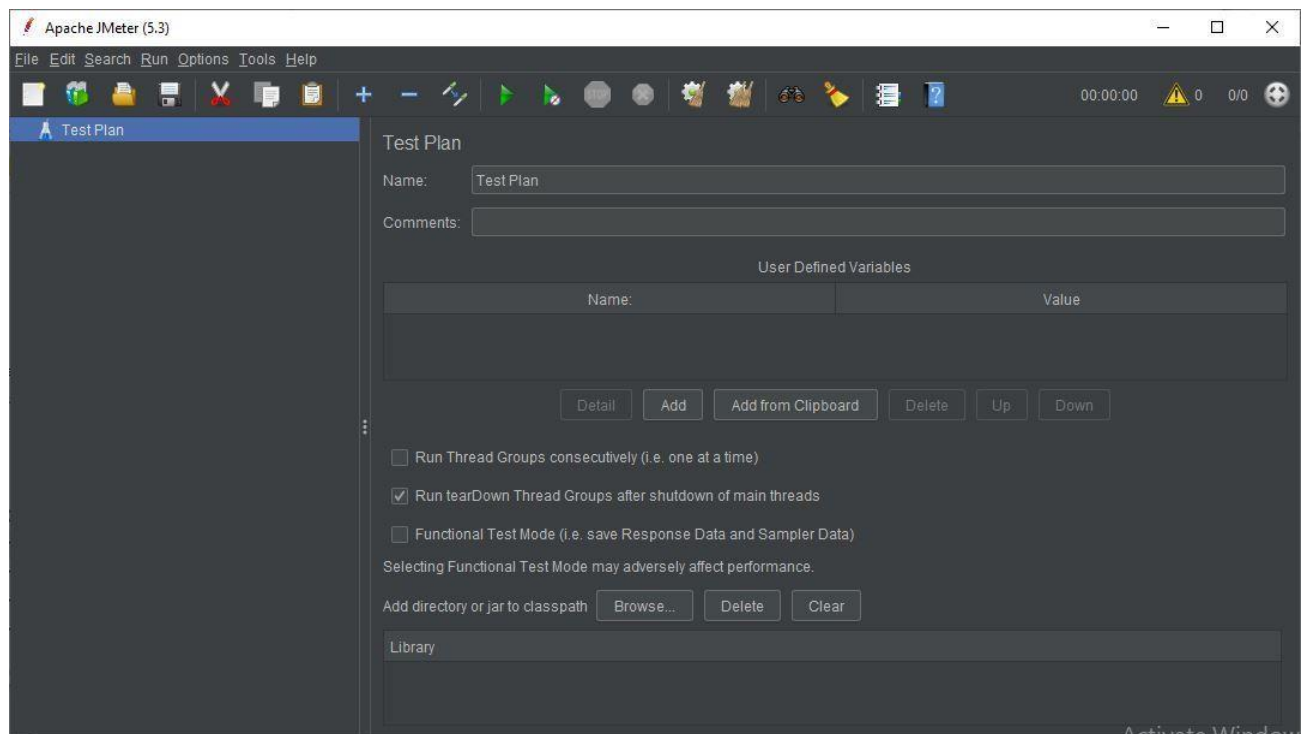


Figure 6: Apache Jmeter user interface

Now, our installation is successful

Short description of JMeter Homepage:

It is divided into two work frames. In the left work frame, we add the threads, groups, test plan name, listener, sampler or whatever we are performing. All the element of JMeter are here.

In the right work frame, we add all the values of the number of threads or users or in which website we are performing our test plan. This consists of the configuration.

Test plan is what to and how to test.

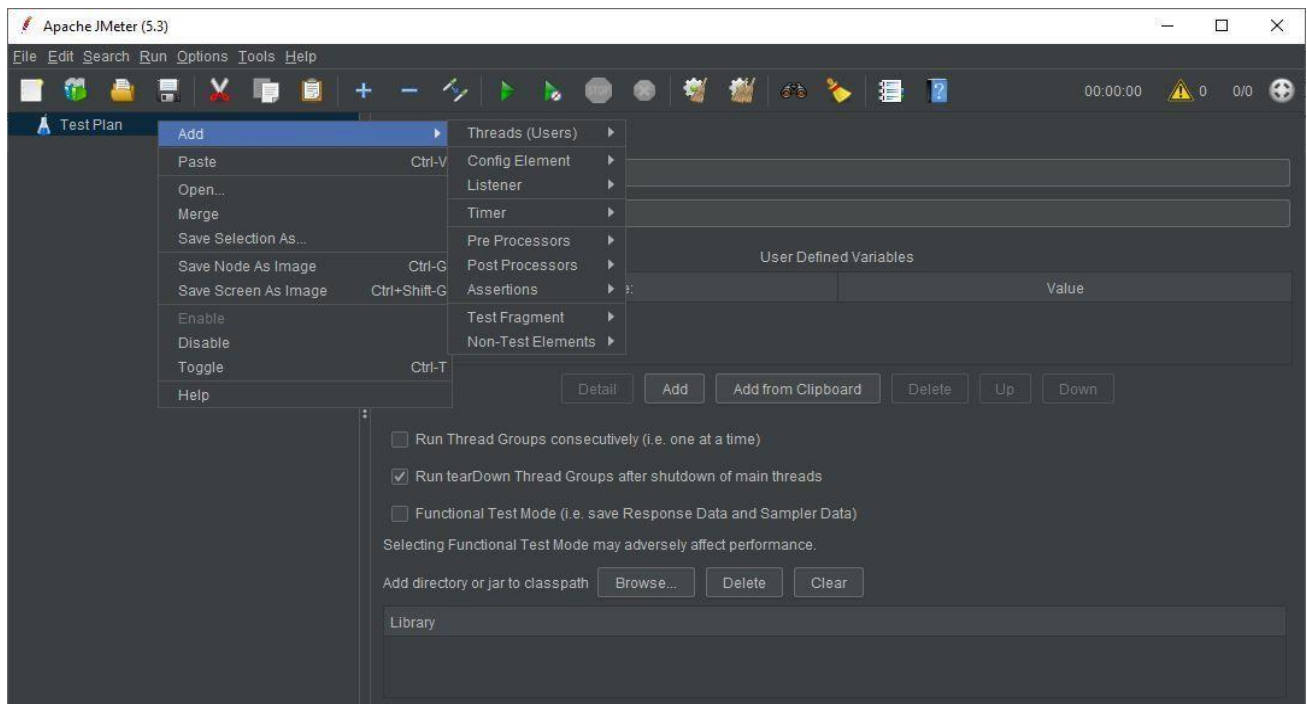


Figure 7: Apache JMeter GUI

Elements of JMeter:

Before starting any test let's discuss some important elements of JMeter. There are 4 important elements:

- Thread group
- Samplers
- Listeners
- Configuration
- 5. Assertion

1. Thread groups:

Thread group is a collection of threads. Each thread represents one user using the application under the test.

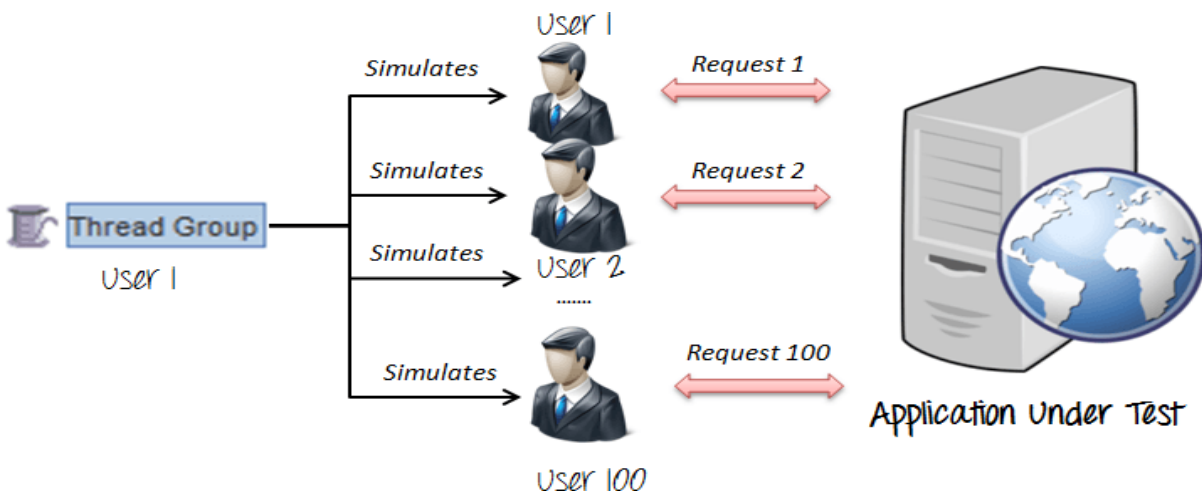


Figure 8: Thread groups

Basically, each thread simulates one real user request to the server and the controls for a thread group allow you to set the number of threads for each group.

For example, if you set the number of threads as 100, JMeter will create and simulate 100 user requests to the server under the test.

2.Samplers:

As we know that Jmeter supports testing HTTP, FTP, JDBC and many other protocols. thread groups create request to server, but which type of request should be created? Thread groups know it from samplers.

Now for the **FTP** request, if you want to create an FTP request to the server then you need to use an FTP request sampler to do this task.

Now **HTTP** request will send **HTTP/HTTPS** request to the server. JMeter sends an http request to google website and retrieves HTML files or image from this website.

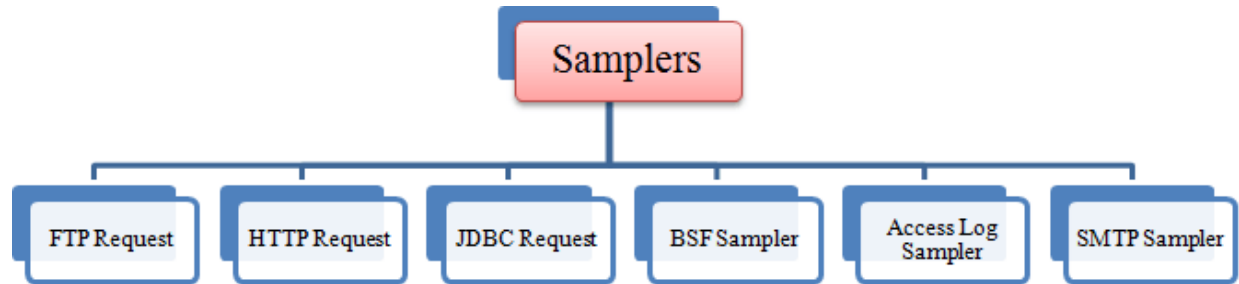


Figure 9: Samplers

Now the **JDBC** request. This sampler lets you execute database performance testing. It sends JDBC request to a Database.

BSF samplers allows you to write BSF scripting language.

Access log sampler allows you to read access log and generate http request. The log could be images, HTML, CSS etc.

If you want to test the mail server, you can use **SMTP** sampler. This sampler is used to send an email messages using the SMTP protocol.

2. Listener:

Listener show the result of the test execution. They can show result in different format such as Tree, table, graph, log file.



Figure 10: Listener

Graph result listener display the server response time on a graph view. View result in tree show result of the user request in basic HTML format. Table result show the summary of the test result in table format.

Log shows the summary of test result in the text file.

3. Configuration:

File important element is the configuration element. It basically sets in default and variable for later used by the samplers.

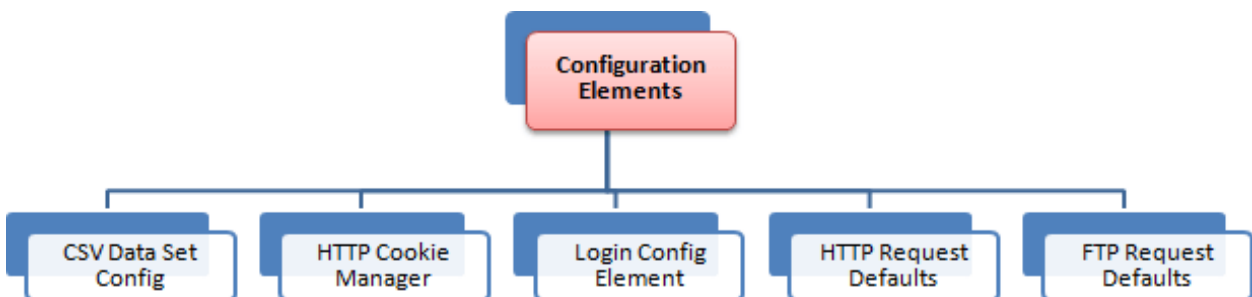


Figure 11: Configuration elements

CSV dataset config: Suppose you want to test e website for 100 users signing in with the different credentials, you did not need to record this file for 100 times. You can parameterize the script to enter different log in credentials. This log in information could be stored in a text file. JMeter has an element that allows you to read different parameter from that text file. It is the CSV data set config which is used to read lines from a file and split them into variable.

HTTP cookie manager: The cookie manager automatically stores the cookie and will use it for all future request for that website.

HTTP request default: It lets you set the default value for HTTP request controller uses. When we need to send 100 HTTP request to the server google.com then, we can add a single HTTP default with the server name or IP field that is equal to google.com.

Login config element: It lets you add or override the username and password settings in samplers. For example, if you simulate one user log in to the website facebook.com with user and password, you can use the log in config element to add the username and password setting in a user request.

4. Assertion:

Assertion help verifies that your server under test returns the **expected** results. Types of assertion:

1. Response assertion: The response assertion lets you add pattern strings to be compared against various fields of the server response.

For example, you send a user request to the website <http://www.google.com> and get the server response. You can use Response Assertion to verify if the server response **contains** expected pattern string (e.g. "OK").

2. Duration assertion: The Duration Assertion tests that each server response was received within a **given amount** of time. Any response that takes longer than the given number of milliseconds (specified by the user) is marked as a failed response.

3. HTML assertion: The HTML Assertion allows the user to check the HTML syntax of the response data. It means the response data must be met the HTML syntax.

Test with JMeter in GUI:

Give name of the test plan.

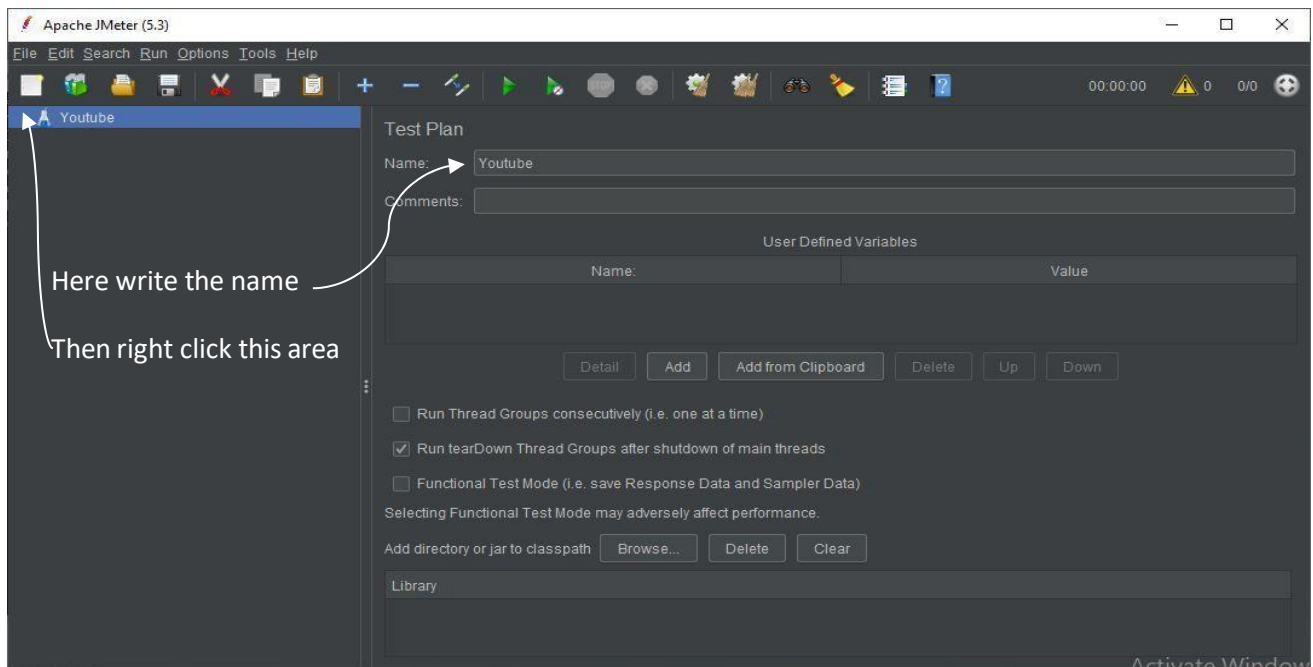


Figure 12: Test name

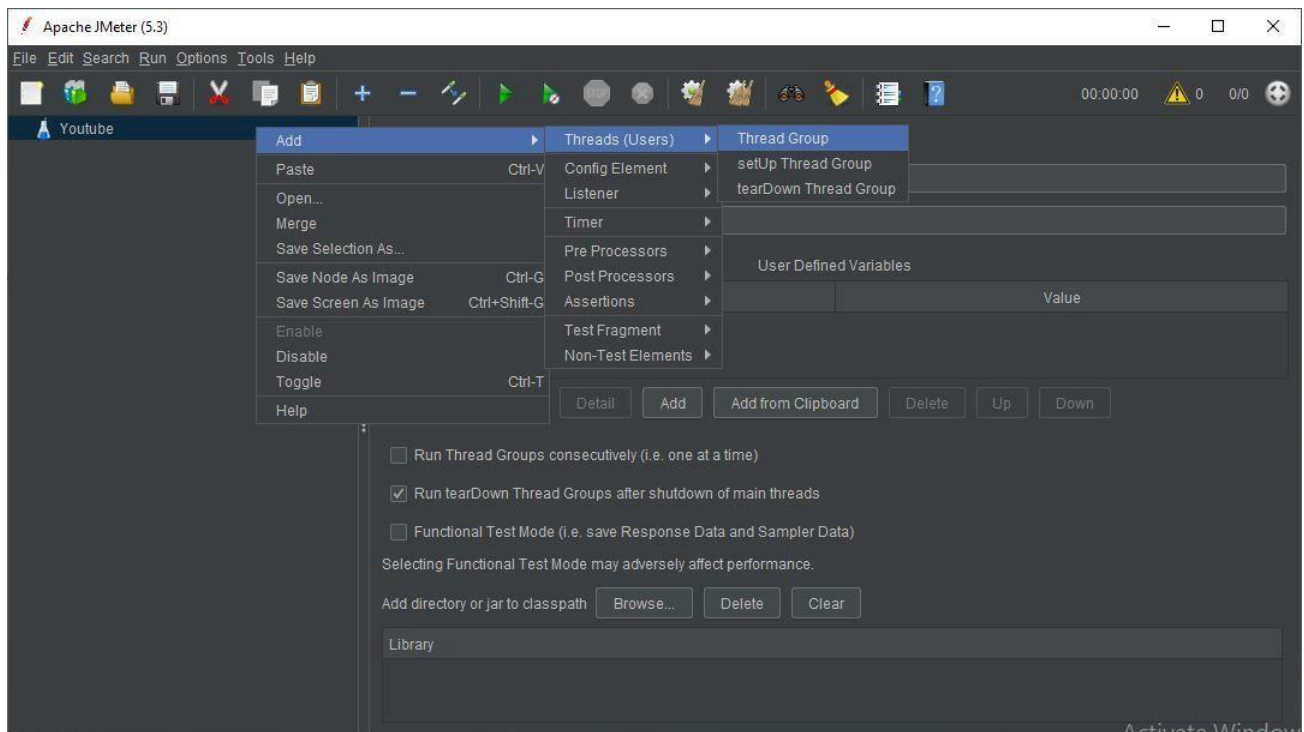


Figure 13: Add thread group
Add a thread group and rename the group as user

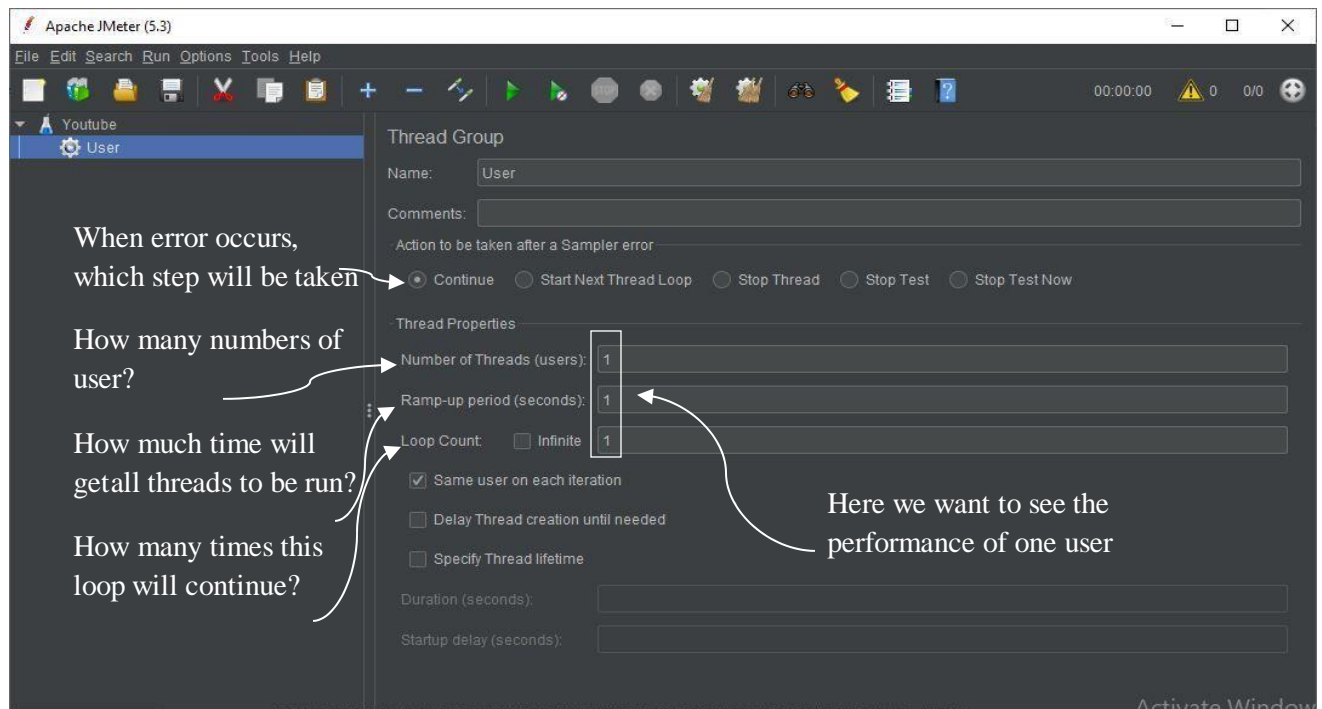


Figure 14: Configure thread group

Let's add an HTTP request sampler. This sampler let you send http or https request to the server.

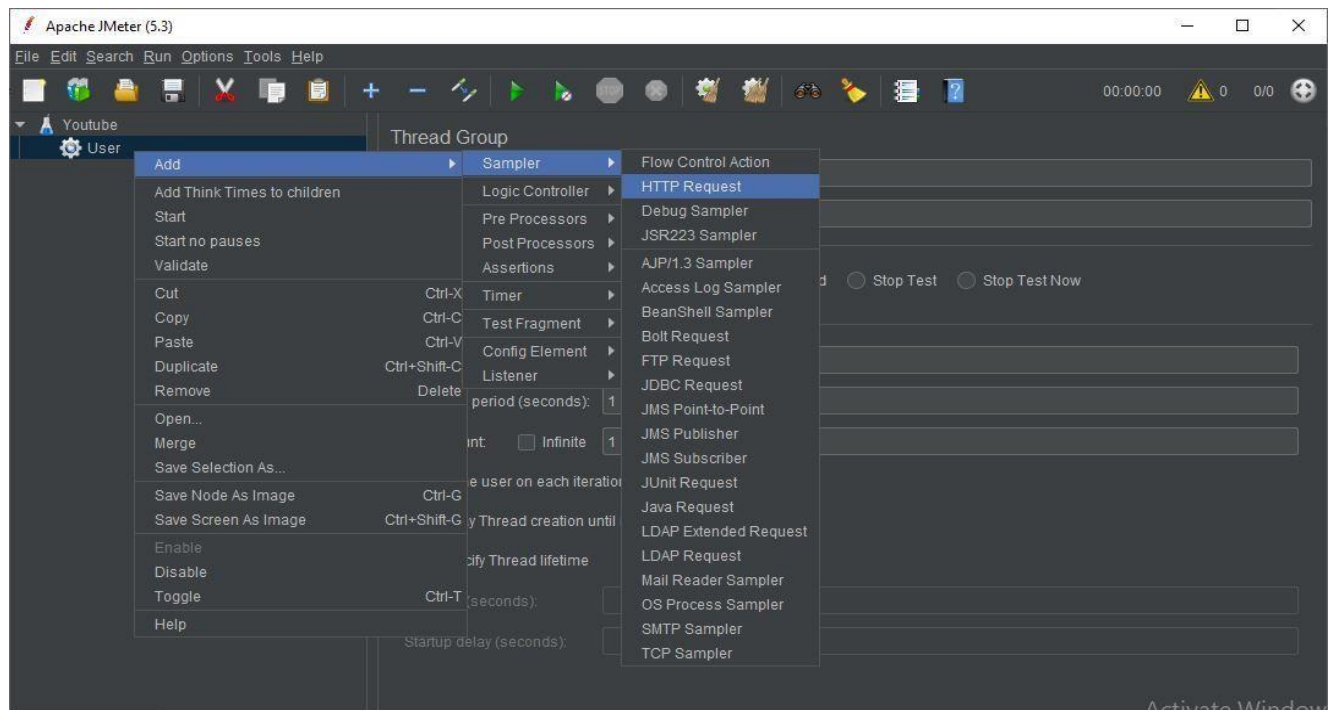


Figure 15: Add sampler (HTTP request)

We want to test youtube homepage. So, go to youtube and copy homepage URL. The URL is <https://www.youtube.com/>

but we are going to add www.youtube.com on server name or IP part.

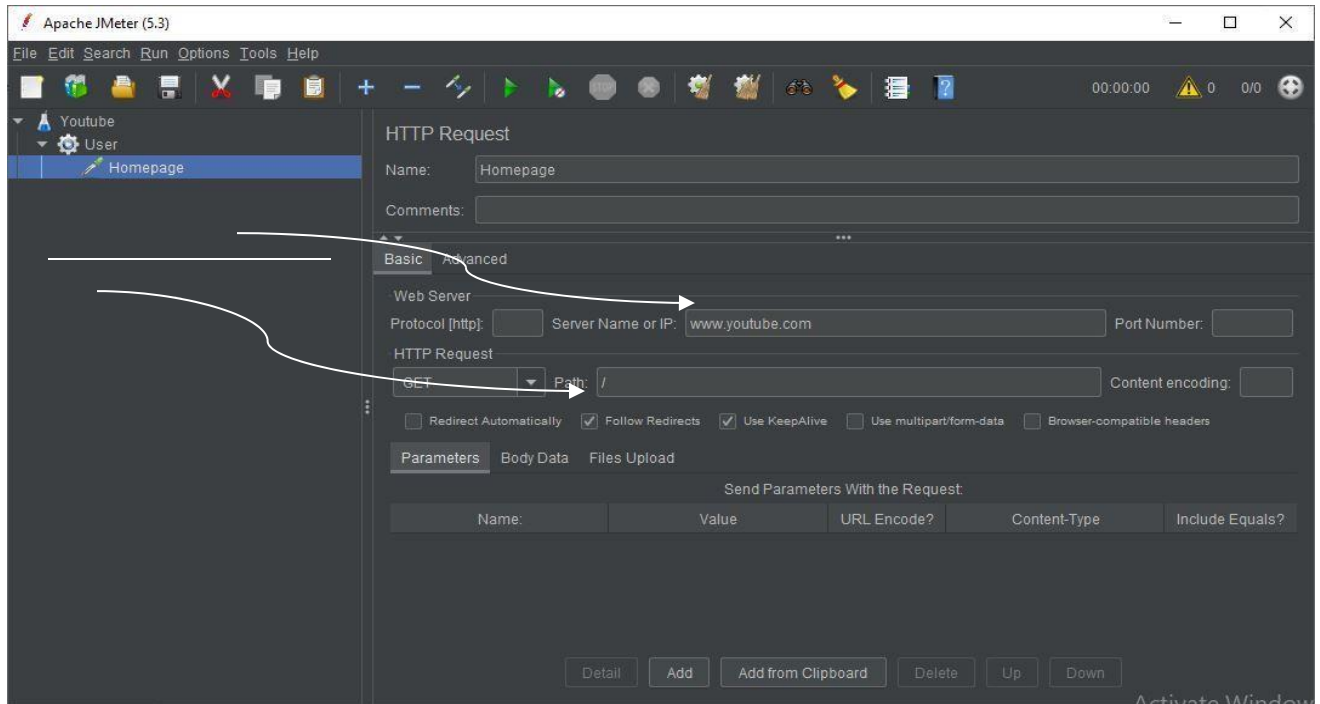


Figure 16: Add HTTP request

Now we need to add listener. Initially we select table format to see

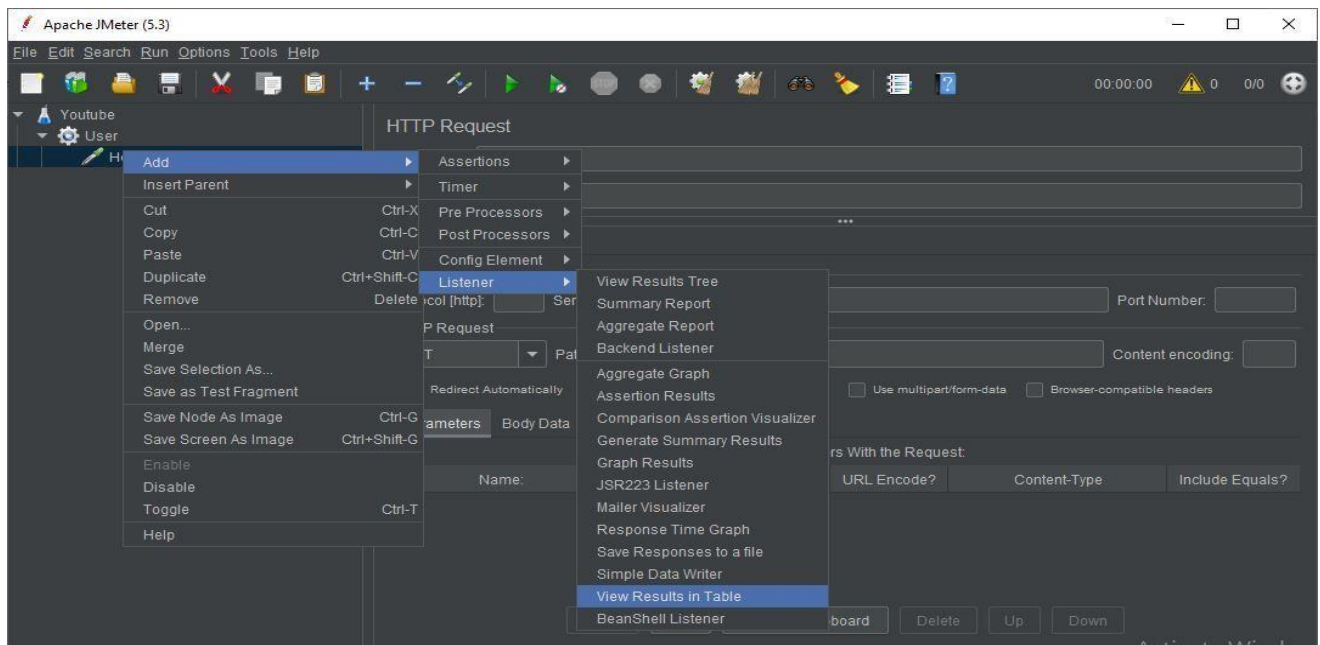


Figure 17: Add listener

So, before running the test plan we need to save it. Then click on run button.

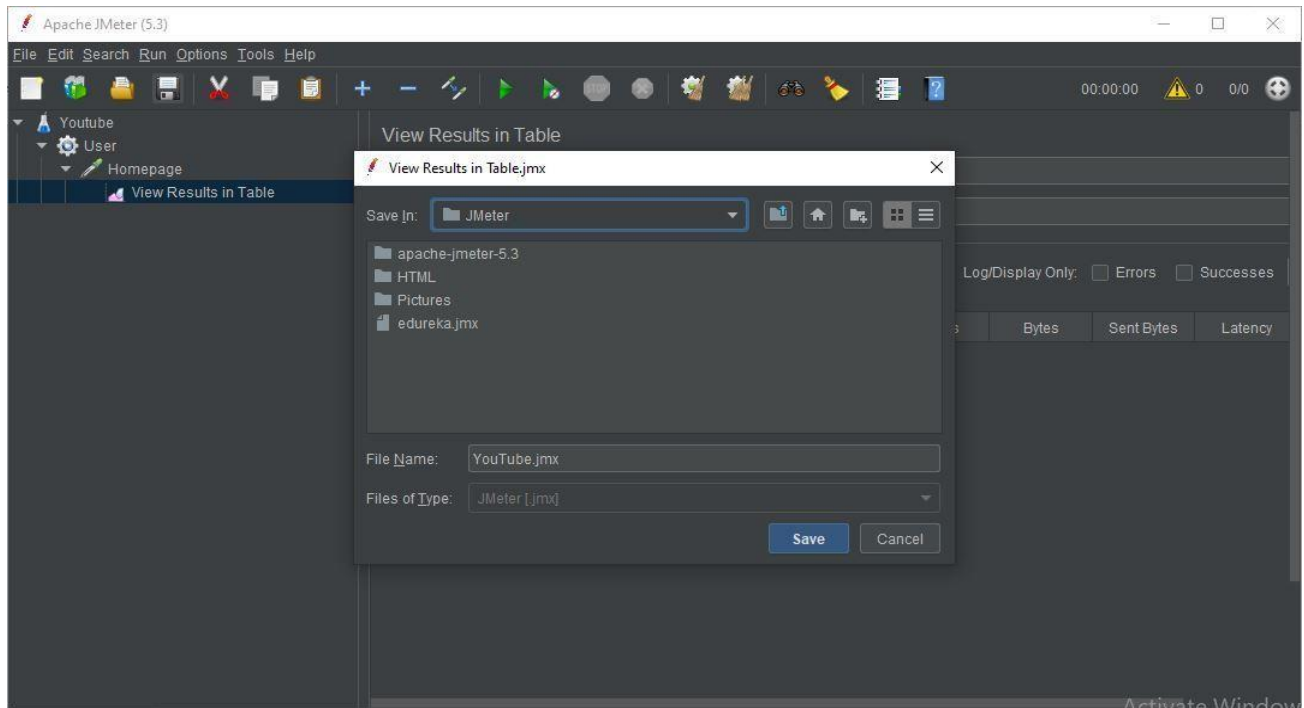


Figure 18: Save Test plan

This file extension is .jmx

I have saved this in E:\SQA\JMeter location of my pc. You can save it any location. But remember this location because we need to go there.

Now run the test plan then see the result.

The screenshot shows the JMeter results table with the following data:

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	13:45:24.875	User 1-1	Homepage	2309	OK	445032	234	364	254

Annotations in the image:

- 'When the process starts' points to the 'Start Time' column.
- 'We save the level name as Homepage OK' points to the 'Label' column.
- 'Total time' points to the 'Sample Time(ms)' column.
- 'Status is OK' points to the 'Status' column.
- 'Time duration between before sending request and get the 1st response' points to the 'Latency' column.

Figure 19: Report

We can also see the result in different format using other listeners.

Now we will check the result using 5 threads. So, go to the Thread(user) configuration and edit it. Then run the test.

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	14:04:54.867	User 1-1	Homepage	667	✓	446217	234	129	42
2	14:04:55.065	User 1-2	Homepage	714	✓	444641	234	154	54
3	14:04:55.469	User 1-4	Homepage	682	✓	445412	234	152	59
4	14:04:55.272	User 1-3	Homepage	936	✓	452842	234	144	51
5	14:04:55.665	User 1-5	Homepage	944	✓	451755	234	130	45

Figure 20: Result of 5 threads

Now we add more HTTP request of youtube's other pages where we can go from the homepage insame way but this time path will different.

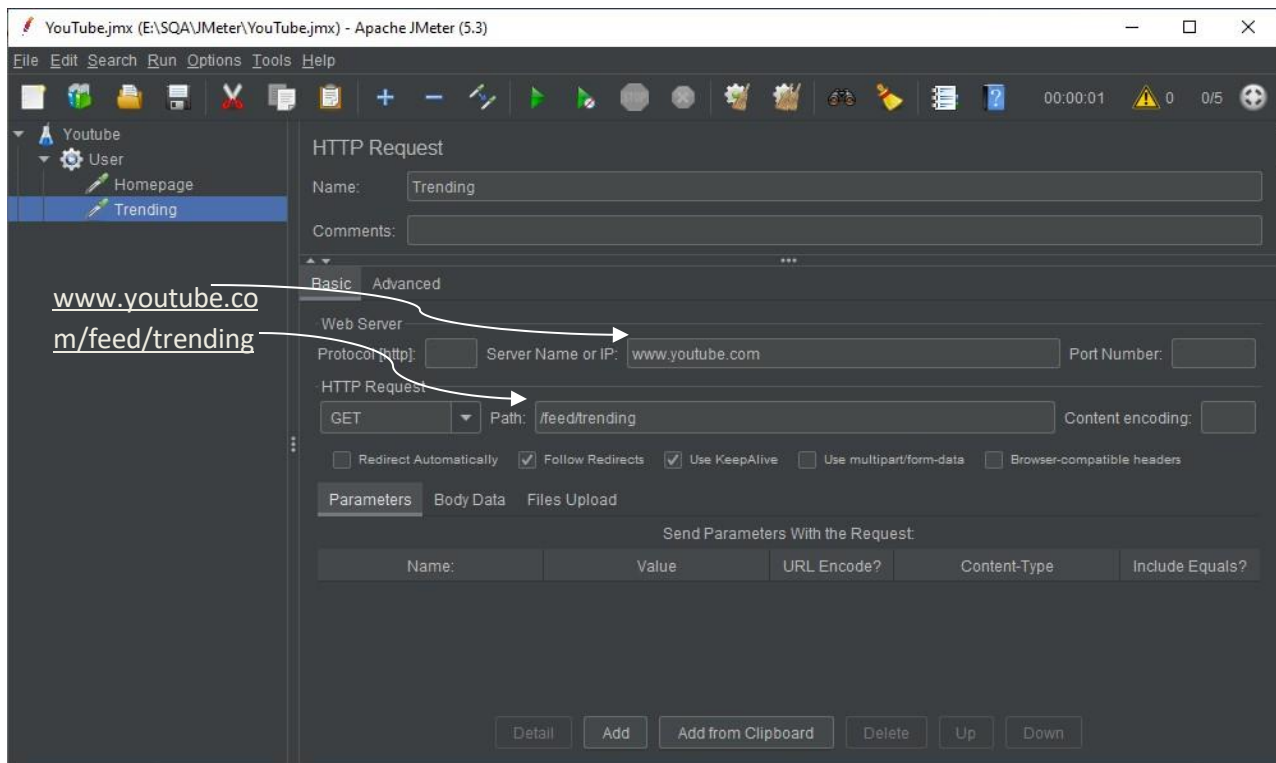


Figure 21: Add another HTTP request

We add YouTube trending website request. Now also add two or three HTTP request of you tube insame way. Then save it and add table format as listener.

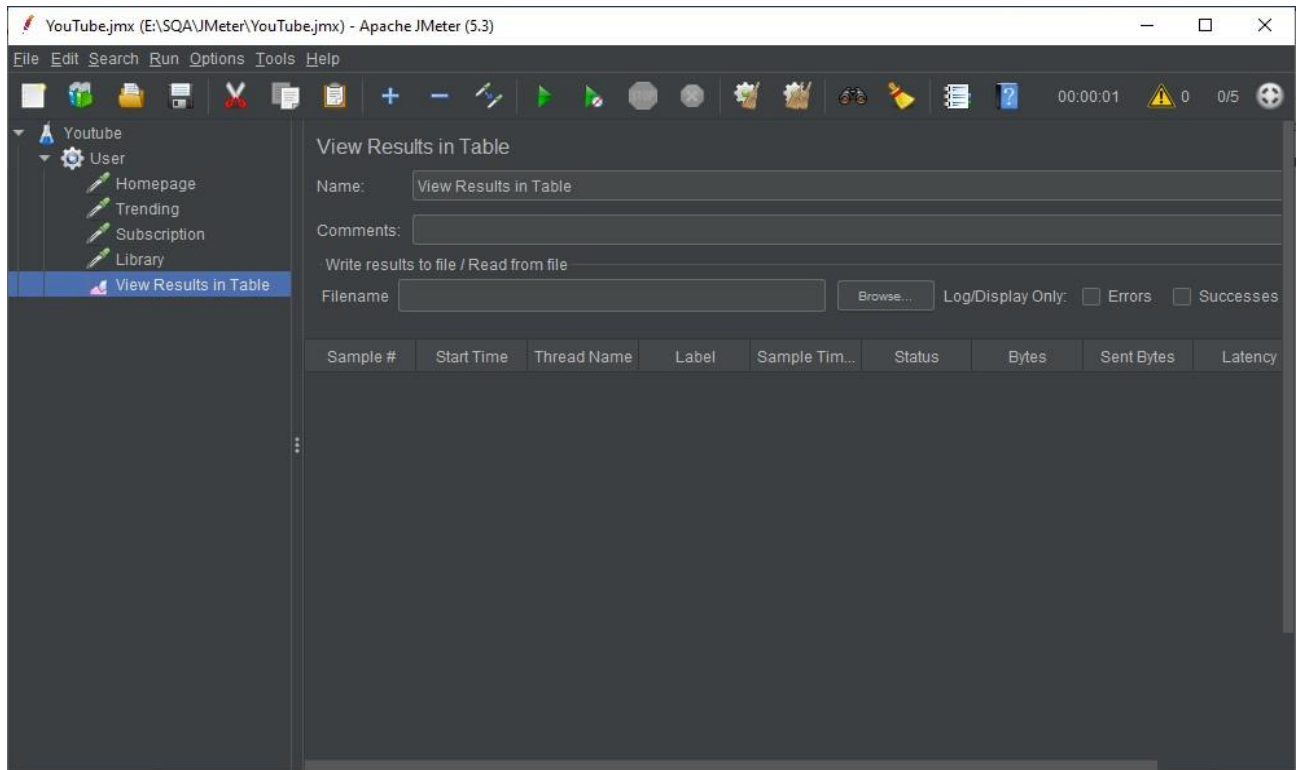


Figure 22: Show all result in table format

Save it and Run.

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	14:29:10.449	User 1-4	Homepage	1104	✓	453310	234	169	59
2	14:29:10.647	User 1-5	Homepage	1355	✓	452526	234	158	57
3	14:29:09.845	User 1-1	Homepage	2168	✓	455029	234	229	125
4	14:29:11.553	User 1-4	Trending	649	✓	724786	260	101	0
5	14:29:12.002	User 1-5	Trending	680	✓	724975	260	125	0
6	14:29:12.013	User 1-1	Trending	669	✓	724858	260	122	0
7	14:29:12.202	User 1-4	Subscription	679	✓	72213	614	102	0
8	14:29:12.881	User 1-4	Library	278	✓	100530	258	100	0
9	14:29:10.249	User 1-3	Homepage	2922	✓	445648	234	1160	1059
10	14:29:12.682	User 1-1	Subscription	585	✓	72321	614	98	0
11	14:29:13.267	User 1-1	Library	319	✓	99995	258	106	0
12	14:29:13.172	User 1-3	Trending	671	✓	724637	260	99	0
13	14:29:12.682	User 1-5	Subscription	1618	✓	72209	614	103	0
14	14:29:10.047	User 1-2	Homepage	4281	✓	452375	234	167	64
15	14:29:14.300	User 1-5	Library	314	✓	100265	258	107	0
16	14:29:14.328	User 1-2	Trending	548	✓	725002	260	99	0
17	14:29:14.876	User 1-2	Subscription	608	✓	72249	614	106	0
18	14:29:15.484	User 1-2	Library	297	✓	100832	258	103	0
19	14:29:13.843	User 1-3	Subscription	3610	✓	72453	614	99	0
20	14:29:17.454	User 1-3	Library	292	✓	99847	258	100	0

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 20
 Latest Sample 292
 Average 1182
 Deviation 1140

Figure 23: Result

Here 5 threads of each makes in total 20 threads. All are successfully run. Now see the result in another format like graph format.

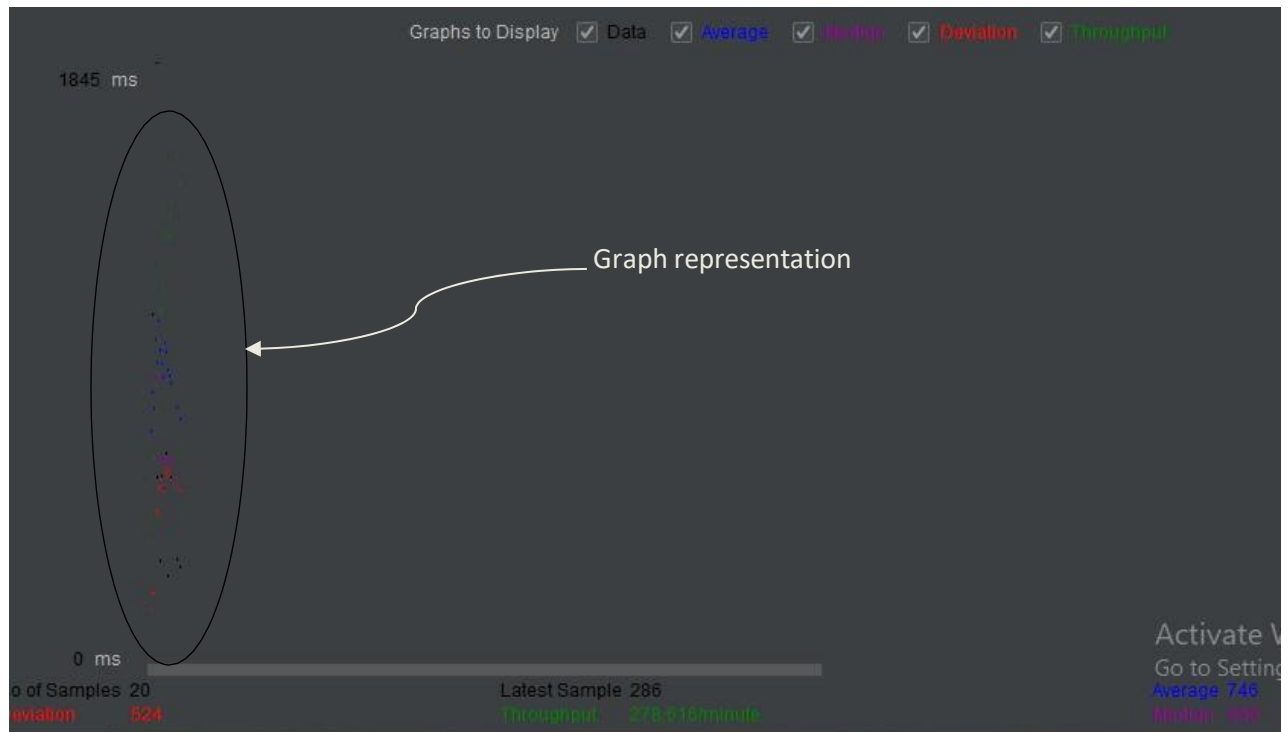


Figure 24: Graphical representation

So now we can see the result in different format using different listeners. But we already feel trouble to see the result in GUI format. We cannot see the result properly. That's why non-GUI Format is come.