Assignment

Performance Testing:
It is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

Types of Performance Testing:

Load testing: Load testing simulates a real-world load on the system to see how it performs under stress. It helps identify bottlenecks and determine the maximum number of users or transactions the system can handle.

Stress testing: Stress testing is a type of load testing that tests the system's ability to handle a high load above normal usage levels. It helps identify the breaking point of the system and any potential issues that may occur under heavy load conditions.

Spike testing: Spike testing is a type of load testing that tests the system's ability to handle sudden spikes in traffic. It helps identify any issues that may occur when the system is suddenly hit with a high number of requests.

Soak testing: Soak testing is a type of load testing that tests the system's ability to handle a sustained load over a prolonged period. It helps identify any issues that may occur after prolonged usage of the system.

Endurance testing: This type of testing is similar to soak testing, but it focuses on the long-term behaviour of the system under a constant load.

Performance Testing Tool:

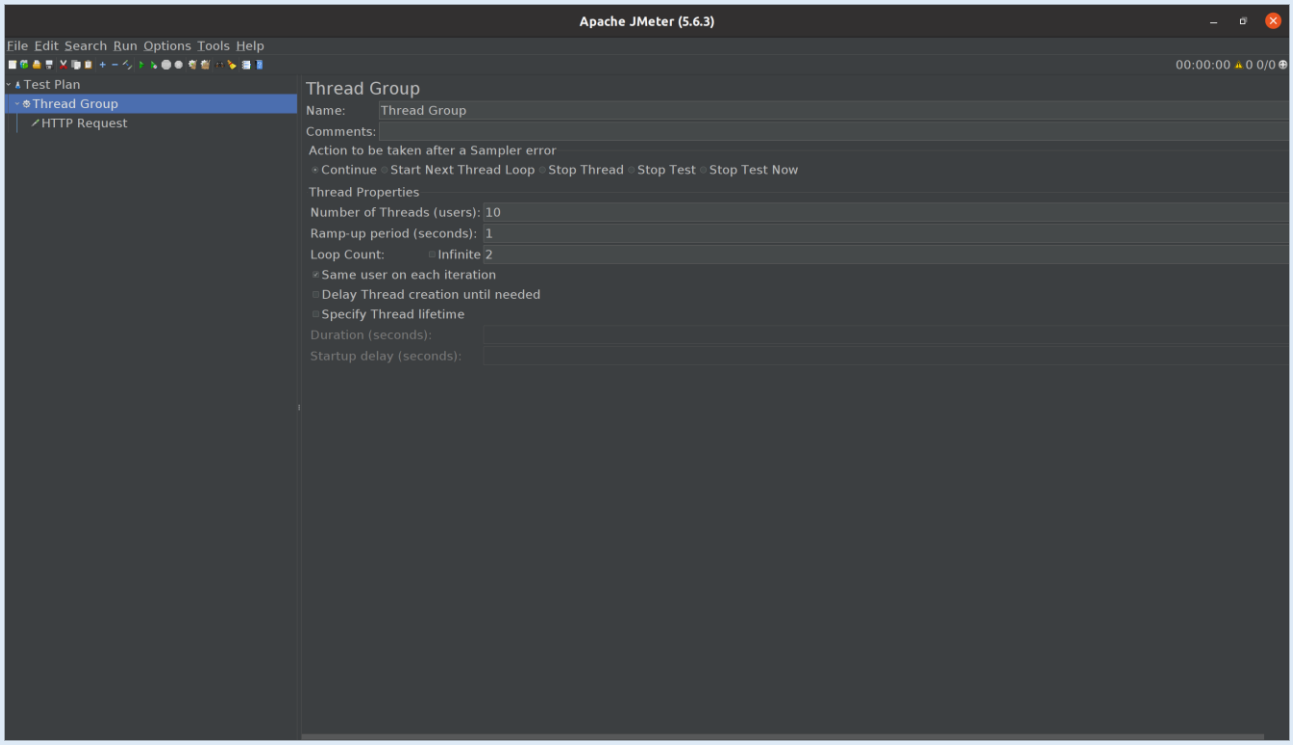
Apache Jmeter:
It is an open-source, purely Java-based software. The software is used to perform performance testing, functional testing, and load testing of web applications. It is used to test load testing functional behavior and measuring performance.

Test Plan:
A test plan describes a series of steps JMeter will execute when run. A complete test plan will consist of one or more Thread Groups, logic controllers, sample generating controllers, listeners, timers, assertions, and configuration elements.

Purpose of each component added in the Test Plan:

Thread Group :

Thread group is the basic element of the Jmeter Test plan. As said in the name Thread group is a group of threads that are executing the same scenario. This is considered the beginning point of a test plan. Thread group holds other test elements like controllers, samplers, config elements also the listeners. Each thread in the thread group will independently execute all the elements under the thread group while running the test plan. Here is the control panel of the thread group element.



Thread Properties

- **Number of Threads** – This is the count of virtual users that we are expecting to connect to the server. For example, if we give 10, Jmeter will simulate 10 virtual users that connect to the server and perform the same steps given. By default, it is set to 1 thread.
- **Ramp-Up Period** – This can be defined as the time in which JMeter can bring the number of threads mentioned above into the running state. This is given in seconds. By default, it is set to 1 second.
- **Loop Count** – This indicates how many times each thread was supposed to perform the task. If the number of threads is 10 and the loop count is 2 then the same task will be performed 20 times. If this value is set to infinity the task

will continue to run until the test was stopped. By default is value is one iteration.

Controller:

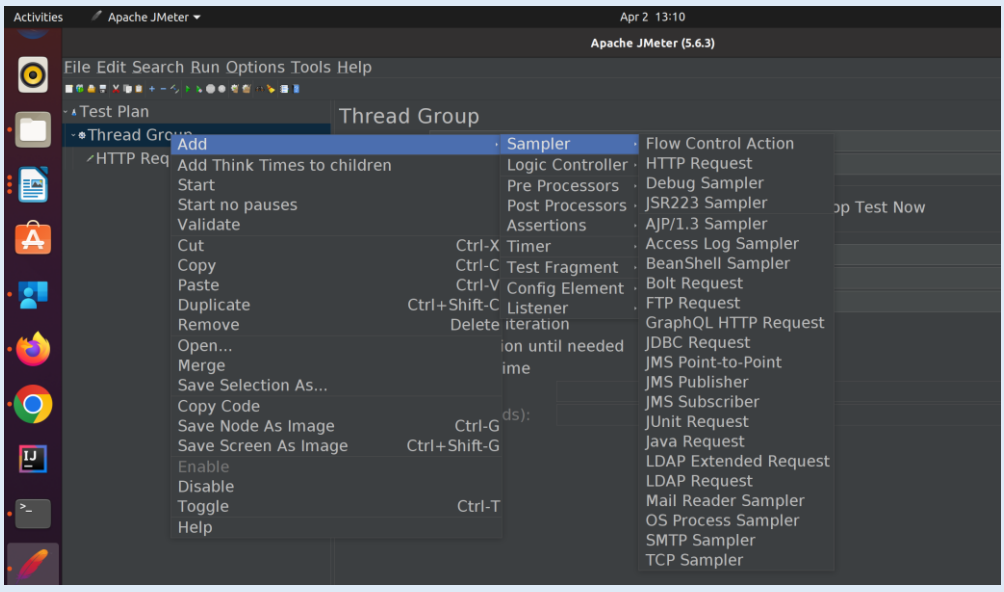
There are two Types of Controller in jmeter:

- **Sampler:**
  Samplers are the components which allow JMeter to send specific types of requests to a server. It simulates a user's request for a page to the target server.

  Samplers are a must to add component to a test plan as only it can let JMeter know what type of request need to go to a server. Requests could be HTTP, HTTP(s), FTP, TCP, SMTP, SOAP etc.

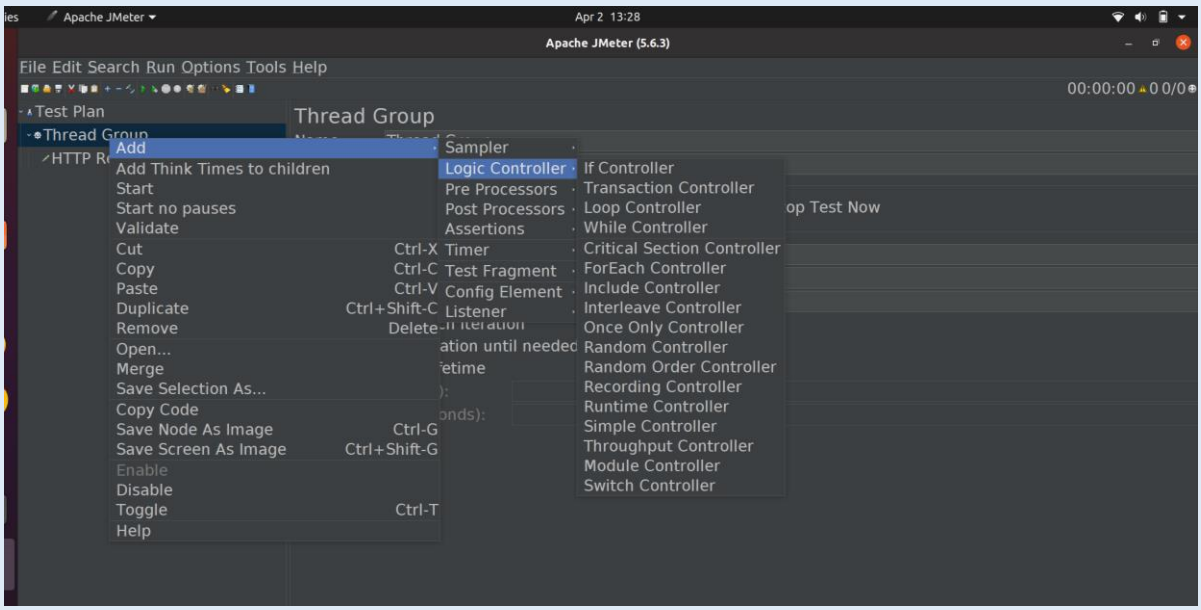
  There are Types of Samplers such as FTP Request ,HTTP Request (can be used for SOAP or REST Webservice also) ,JDBC Request ,Java object request ,JMS request ,JUnit Test request ,LDAP Request ,Mail request,OS Process request ,TCP request

  In this Assignment We Use HTTP Request Sampler.



- **Logic Controller :**
  Logic Controllers help you to control the flow the order of processing of samplers in a thread. It can also change the order of requests coming from their child elements.
  Some Logical Controller Jmeter Provide Runtime Controller :IF Controller ,Transaction Controller ,Recording Controller ,Simple Controller ,While Controller ,Switch Controller ,ForEach Controller ,Module Controller ,Include

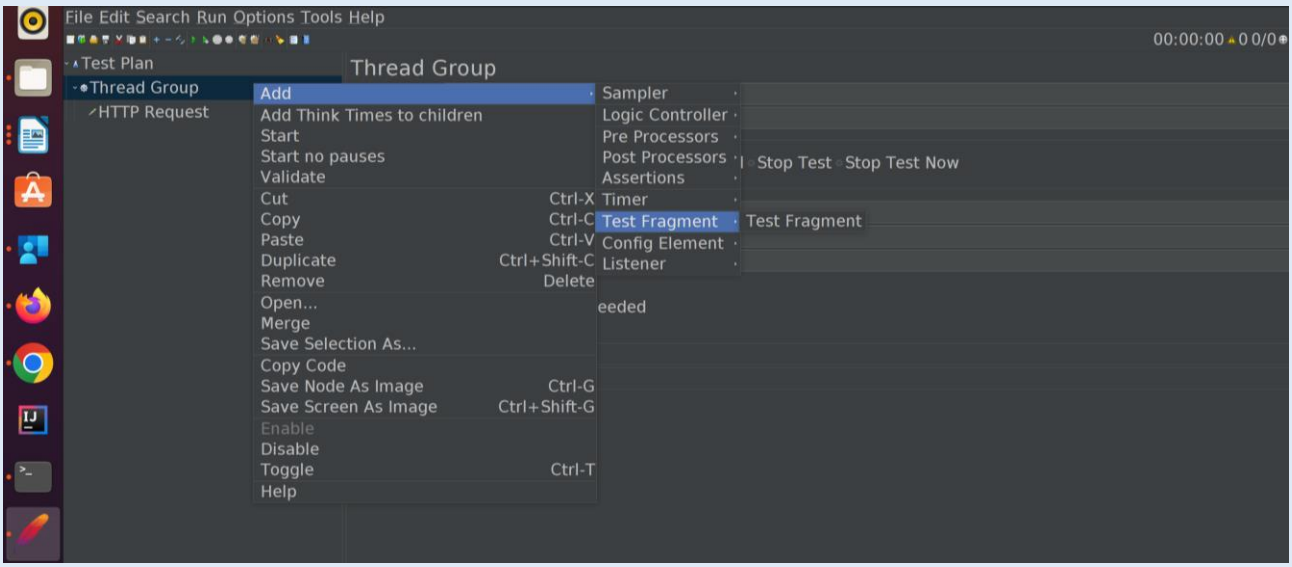
Controller ,Loop Controller ,Once Only Controller ,Interleave Controller ,Random Controller ,Random Order Controller ,Throughput Controller.



Test Fragments:

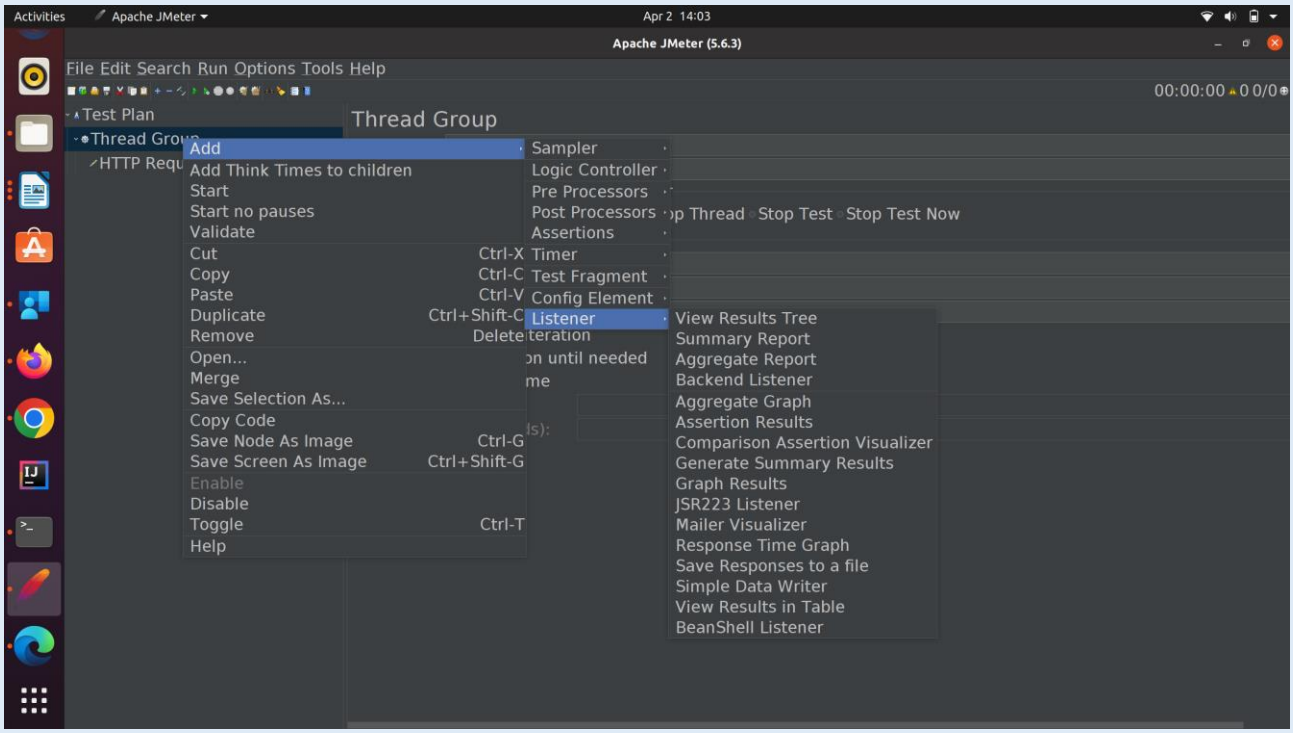
The Test Fragment element is a special type of controllers that exists on the Test Plan tree at the same level as the Thread Group element. It is distinguished from a Thread Group in that it is not executed unless it is referenced by either a Module controller or an Include controller.

This element is purely for code re-use within Test Plans.



Listeners:

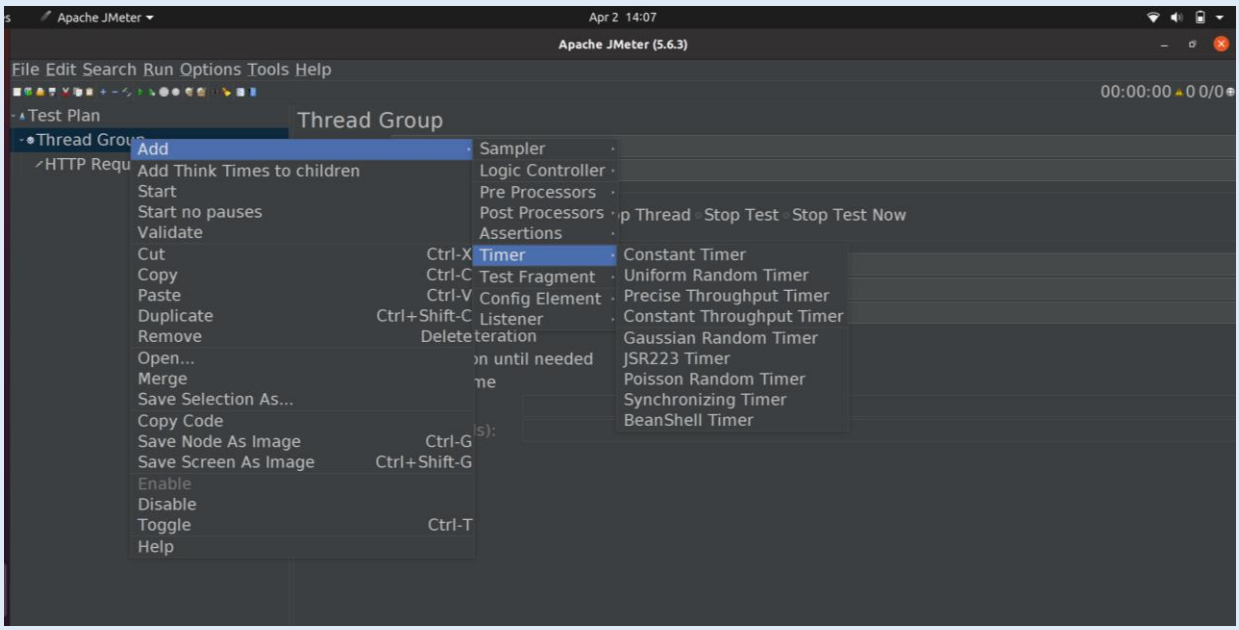
Listeners provide access to the information JMeter gathers about the test cases while JMeter runs. The Graph View listener plots the response times on a graph. The "View Results Tree" Listener shows details of sampler requests and responses, and can display basic HTML and XML representations of the response. Other listeners provide summary or aggregation information. Additionally, listeners can direct the data to a file for later use. Every listener in JMeter provides a field to indicate the file to store data to. There is also a Configuration button which can be used to choose which fields to save, and whether to use CSV or XML format.



Timer:

By default, a JMeter thread executes samplers in sequence without pausing. We recommend that you specify a delay by adding one of the available timers to your Thread Group. If you do not add a delay, JMeter could overwhelm your server by making too many requests in a very short amount of time.

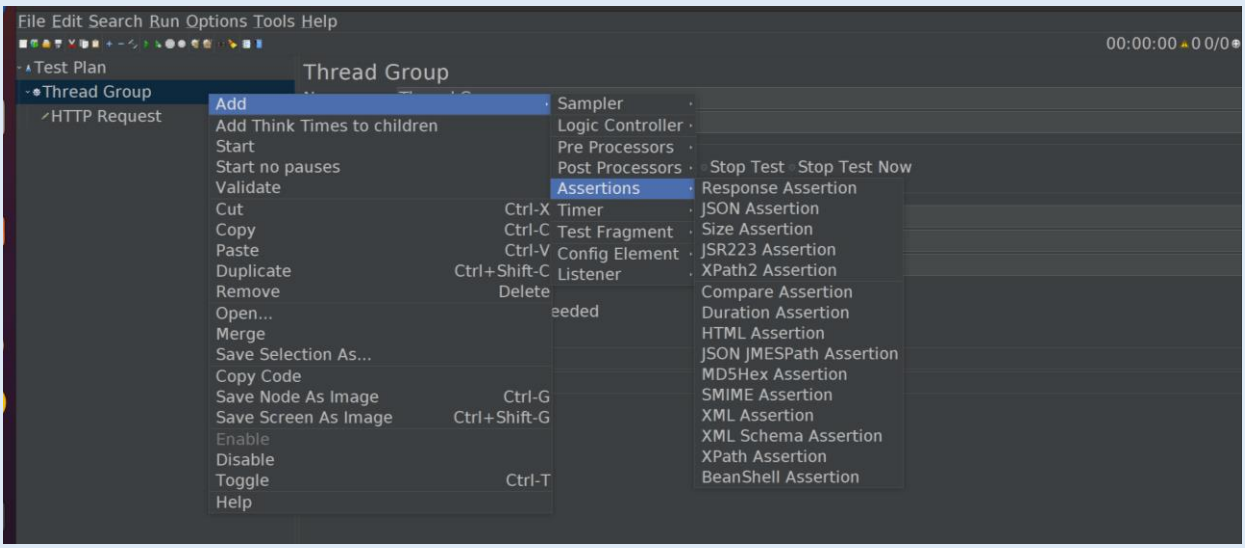
A timer will cause JMeter to delay a certain amount of time before each sampler which is in its scope.



Assertions:

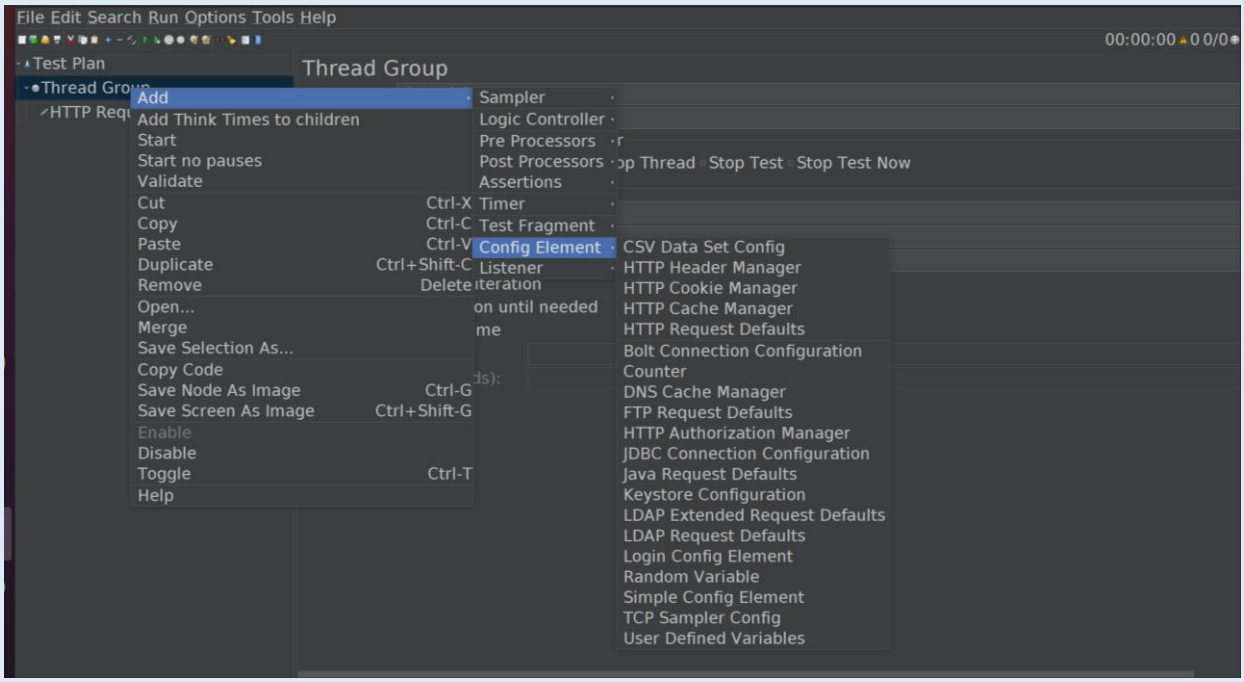
Assertions allow you to assert facts about responses received from the server being tested. Using an assertion, you can essentially "test" that your application is returning the results you expect it to.

You can add an assertion to any Sampler. For example, you can add an assertion to a HTTP Request that checks for the text, "</HTML>". JMeter will then check that the text is present in the HTTP response. If JMeter cannot find the text, then it will mark this as a failed request.



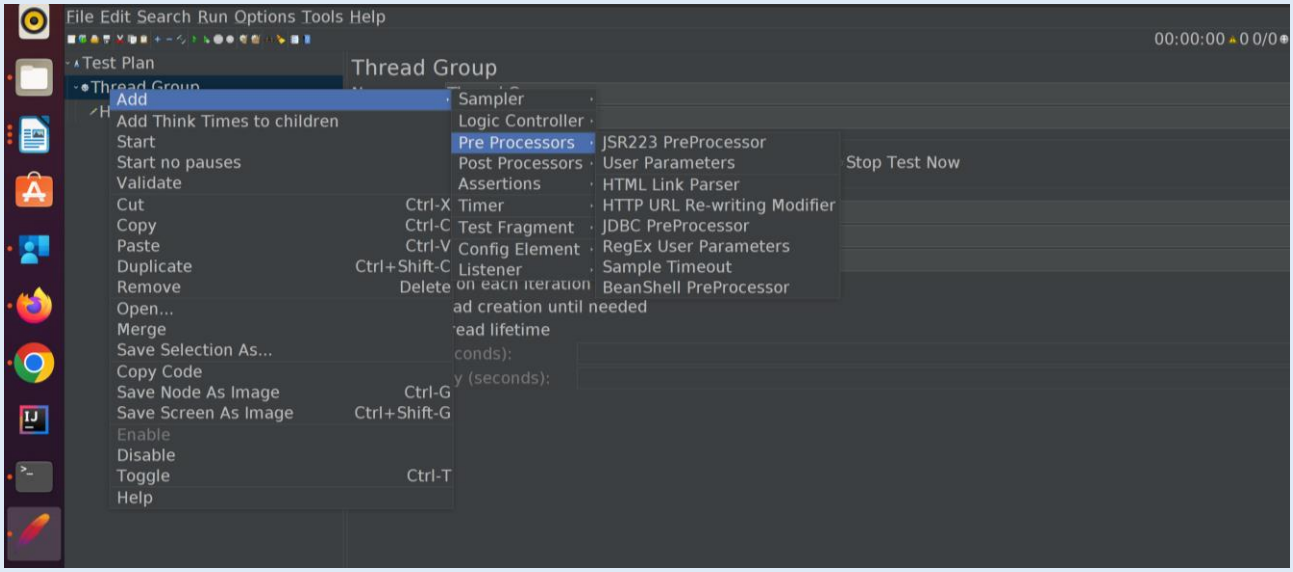
Configuration Elements:

It is a simple element where we can collect the configuration values of all samplers like webserver's hostname or database url etc. A conf element is accessible from only inside the tree branch where we place the element.
A configuration element is accessible from only inside the branch where you place the element.



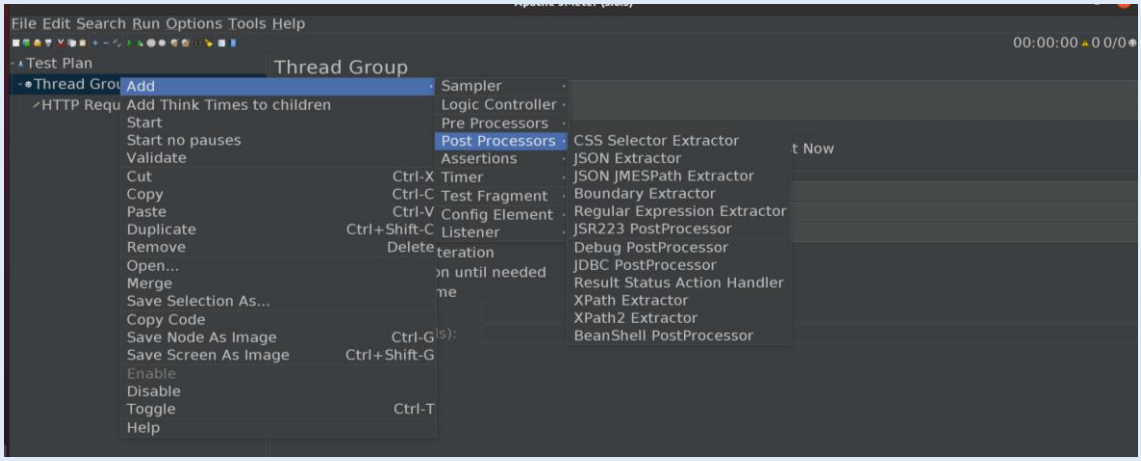
Pre-Processor Elements

A Pre-Processor executes some action prior to a Sampler Request being made. If a Pre-Processor is attached to a Sampler element, then it will execute just prior to that sampler element running. A Pre-Processor is most often used to modify the settings of a Sample Request just before it runs, or to update variables that aren't extracted from response text. See the scoping rules for more details on when Pre-Processors are executed.



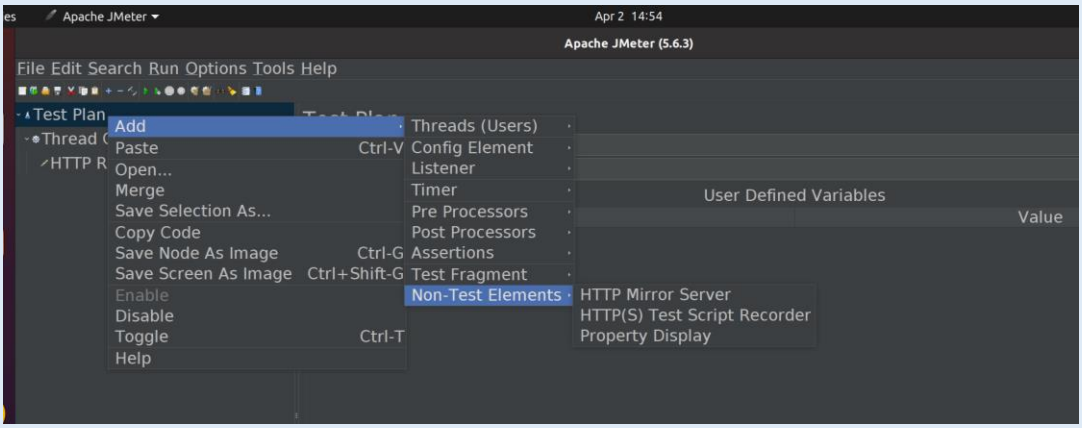
Post-Processor Elements

A Post-Processor executes some action after a Sampler Request has been made. If a Post-Processor is attached to a Sampler element, then it will execute just after that sampler element runs. A Post-Processor is most often used to process the response data, often to extract values from it. See the scoping rules for more details on when Post-Processors are executed.



Non-Test Elements:

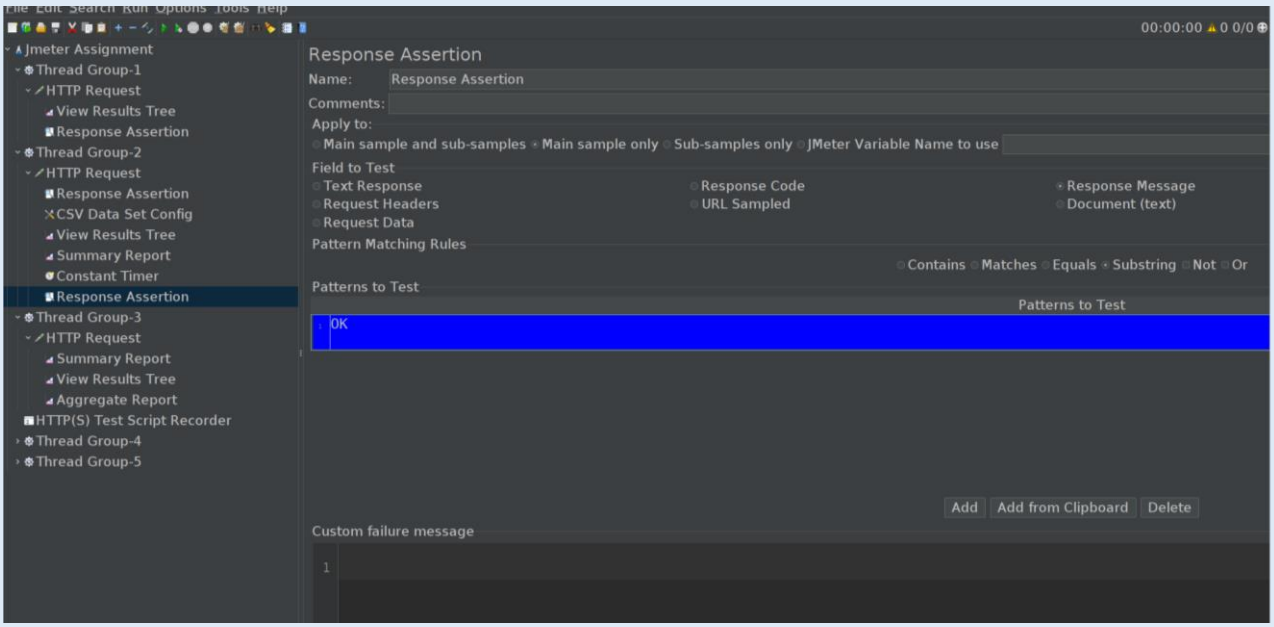
These elements do not directly participate in the test execution, but have a significant contribution to JMeter scripting. HTTP(S) Test Script Recorder is specially used to record the test script.
JMeter elements are scope specific, so you need to add them in the script logically. Some of the elements can be a parent as well as a child element for others, but some elements do not follow the hierarchy like 'Timer' does not have any child element.



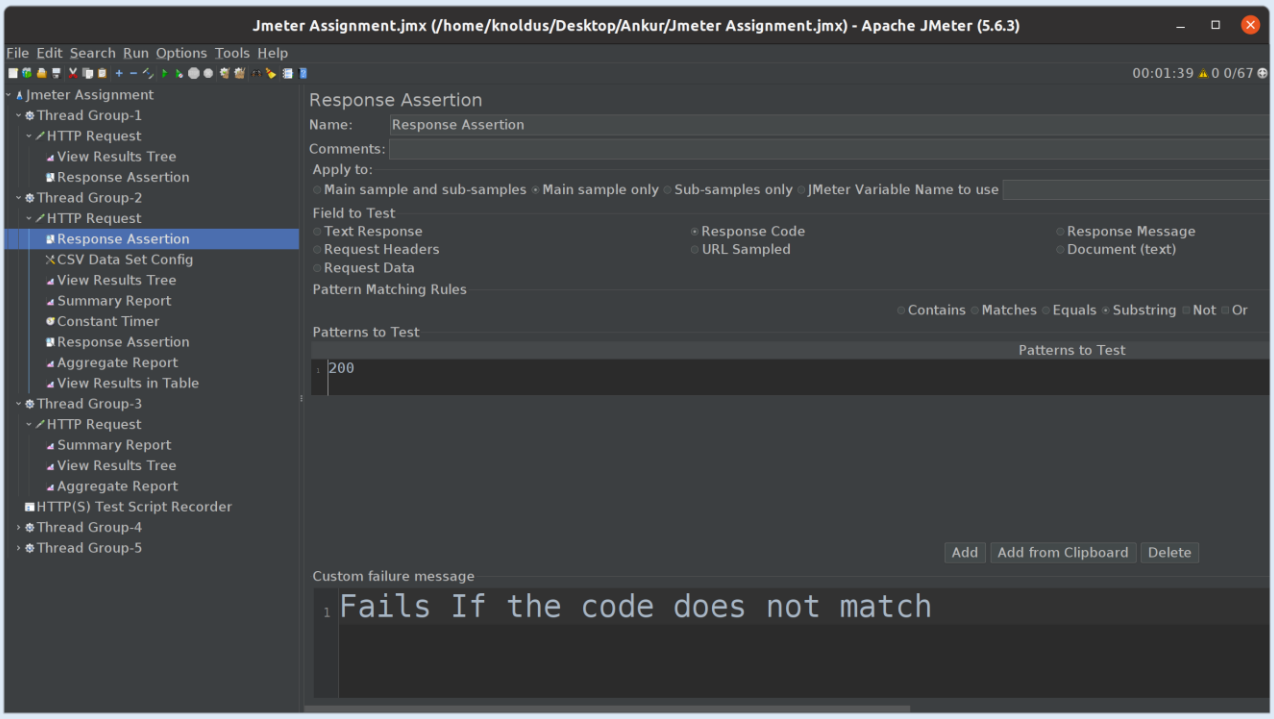
Operation perform on Jmeter:

Assertions:

Response Message:
This Assertion we used to verify response message of request.for example i check OK message as a responce.



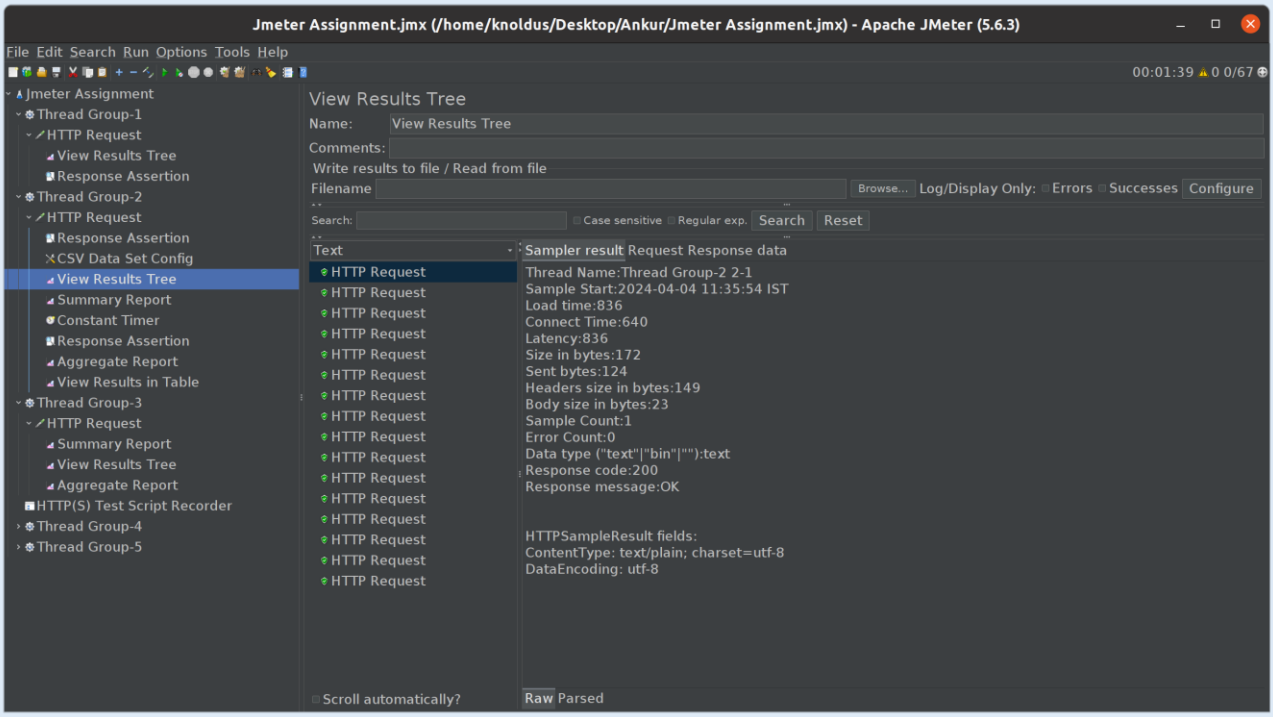
Response Code:
This Assertion we used to verify response code of request.for example i check Status code 200 as a responce and if responce code is defrent is shows error message.



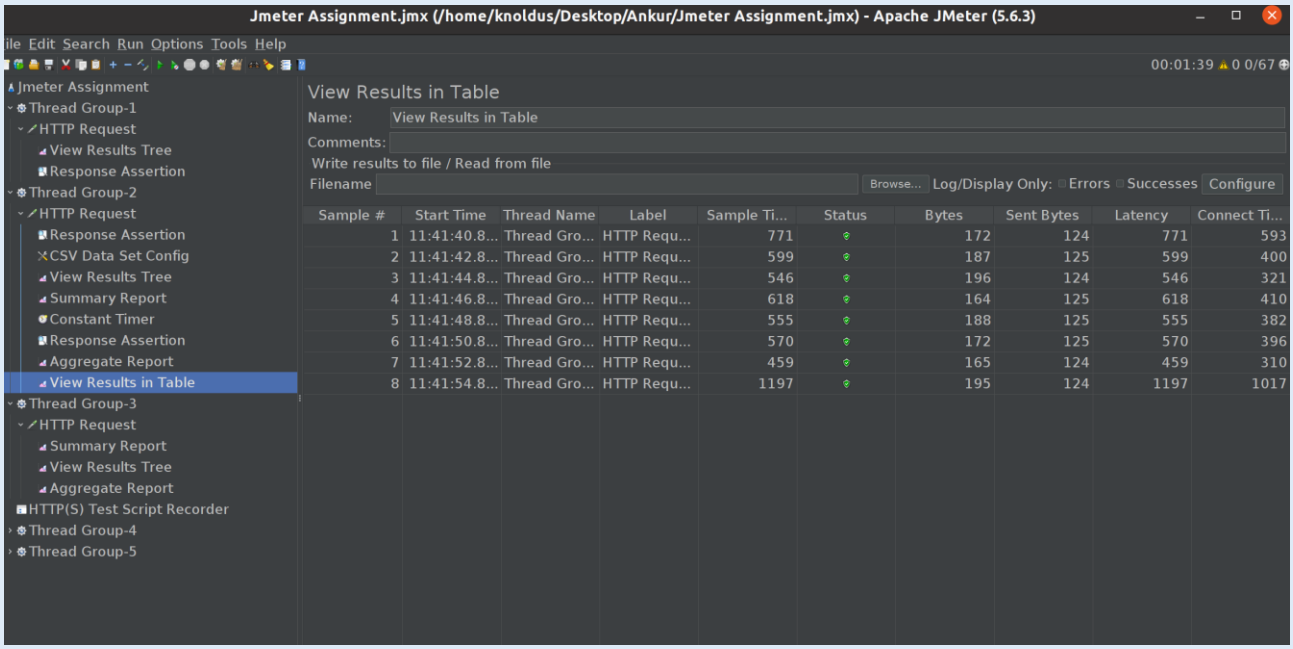
Reports:

1. View Results Tree:

It is used when we need to see the requests in tree format.

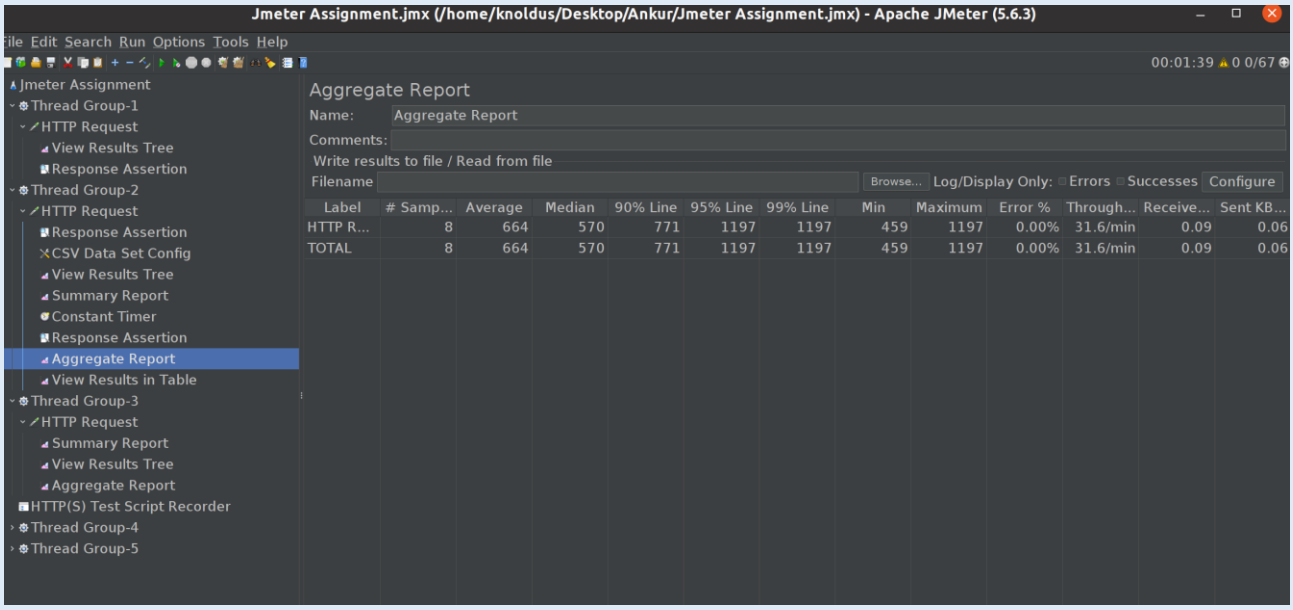


2. View Results In Table:
It is used when we we want to see each requests in table repesentation.



3. Aggregate Reports:

We need to see requests with there avarage time ,throughtput etc.



4. Summary Reports:
It shows the conclution of all the requests we are request to the servers.