

The **Apache JMeterTM**is pure Java **open source** software, which was first developed by Stefano Mazzocchi of the Apache Software Foundation, designed to load test functional behavior and measure performance.

JMeter is to analyze and measure the performance of web application or variety of services. Performance testing means testing a web application against heavy load, multiple and concurrent user traffic.

JMeter originally is used for testing Web Application or FTP application. Nowadays, it is used for functional test, database server test etc.

Open source:

JMeter is totally free , allow developer use the source code for the development

Friendly GUI : JMeter is extremely easy to use and doesn’t take time to get familiar with it

Platform independent : JMeter is java desktop application .so it can run on multiple platforms

Full multi-threading framework: JMeter allows concurrent and simultaneous sampling of different function by separate thread group

Visualize test result : test result can be display in different format such as chart ,table, tree and log file

Highly extensible : you can write your own test .JMeter also supports visualization plugins allow you extend your testing

Multiple testing strategy: JMeter support many testing strategies such as load testing , distributed testing , and functional testing

Support multi-protocol :JMeter does not only support web application testing , but also evauate database server performance all basic protocols such as HTTP,JDBC,LDAP,SOAP,JMS ,and FTP are supported by JMeter

Record and play back –record the user activity on the browser and simulate them in the web application using JMeter

**JMeter workflow**

**The components of JMeter**

The different components of JMeter are called Elements and each are designed for different purpose

**Elements in JMeter**

Thread Group

Samplers

Logic controller

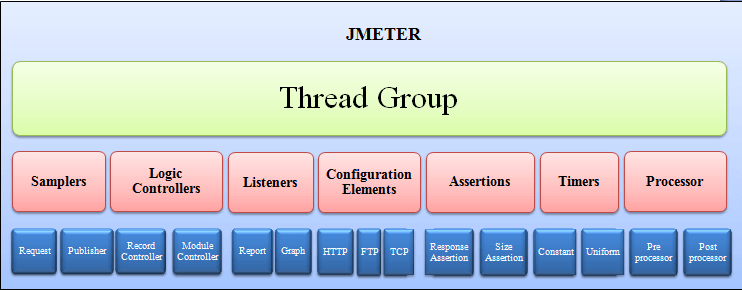
Listeners

Configuration Elements

Assertions

Timers

Processor



**Important elements are**

Thread Group

Samplers

Listeners

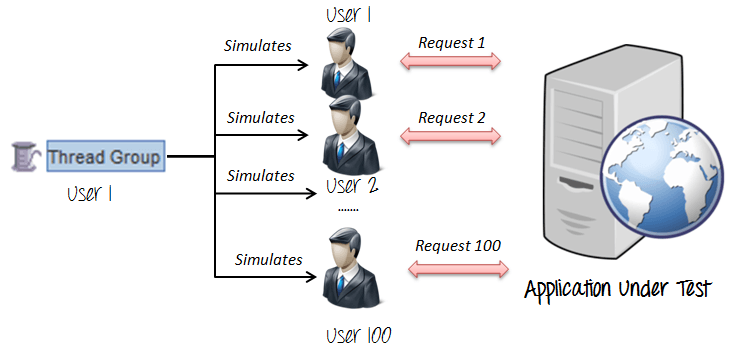
Configuration

**THREAD GROUP**

Thread groups is a collection of threads each thread represents one user using the application under test Basically each thread simulates one real user request to the server

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 thread as 100: JMeter will create and simulate 100 user requests to the server under test



**SAMPLERS**

**JMeter** supports testing HTTP, FTP, JDBC and many other protocols. Sampler helps to thread group to identify which type of request (HTTP,FTP ,etc)

**FTP request :**

Let sampler will help to send an FTP “download file “or “upload file” request to an FTP server

**HTTP request :**

HTTP sampler allow to send HTTP/HTTPS request to a web server

Example:

Google website to get or retrieve images or files

**JDBC request :**

JDBC sampler used to execute database performance testing . it send a JDBC request (SQL Query) to a database.

**BSF sampler :**

BSF sampler will help to write a sampler in JMeter using a BSF scripting language.

**ACCESS LOG sampler**

Access log sampler will allows you to read access logs and generate http request . the log could be image,css,html etc .

**SMTP sampler**

SMTP sampler is used to send email messages using SMTP protocol

**LISTENERS**

Listeners help to shows the result of the test execution .it will show the result in different format such as

Tree, table, graph or log file

**Graph :**

Display the server response times on graph

**Result Tree:**

Display result of user request in basic HTML format

**Table Result :**

Display summary of test result in table format

**Log :**

Display summary of test result in text file

**Latency,sampletime,Throughput**

Latency = 922 ms means that it takes 922 ms as the first response of the request from when the request has been made.

Sample Time(or Response Time) = 1232 ms means that it takes 1232 ms to process the request from when it has been made.

Throughput – indicates the number of transactions per second an application can handle,the amount of transactions produced over time during a test.

Throughput = (number of requests) / (total time).

**CONFIGURATION ELEMENTS :**

A configuration element works closely with a Sampler. Although it does not send requests (except for [HTTP(S) Test Script Recorder](https://jmeter.apache.org/usermanual/component_reference.html#HTTP(S)_Test_Script_Recorder)), it can add to or modify requests.

A configuration element is accessible from only inside the tree branch where you place the element. For example, if you place an HTTP Cookie Manager inside a Simple Logic Controller, the Cookie Manager will only be accessible to HTTP Request Controllers you place inside the Simple Logic Controller .

* **CSV data set config:**

Jmetre has a element that allows you to read different parameter from the text file it is called csv data set config.,which is use to read line from the files ,and split them into variables.it is also called parameter testing.

* **HTTP COOKIE MANAGER**

Http cookies manager also has the same features of web browsers,when http request and response containing a cookies ,the cookie manager automatically store that cookie and will use it for all future request to that particular website.

* **HTTP REQUEST DEFAULT**

This element allow to set default values that http request control use

Example: In a single http request we can send 100 or more http request by default with the server name or ip address.

**TIMERS**

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](https://jmeter.apache.org/usermanual/test_plan.html#scoping_rules).

If you choose to add more than one timer to a Thread Group, JMeter takes the sum of the timers and pauses for that amount of time before executing the samplers to which the timers apply. Timers can be added as children of samplers or controllers in order to restrict the samplers to which they are applied.

To provide a pause at a single place in a test plan, one can use the [Flow Control Action](https://jmeter.apache.org/usermanual/component_reference.html#Flow_Control_Action) Sampler.

**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.

For instance, you can assert that the response to a query will contain some particular text. The text you specify can be a Perl-style regular expression, and you can indicate that the response is to contain the text, or that it should match the whole response.

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.

## 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. **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.

Jmeter - Web Test Plan

**Start JMeter**

Add Thread Group

1. Right click on test plan
2. Select  Add > Threads (Users) > Thread Group.

Add Sampler

1. Right click on thread group
2. Select Add > Sampler > HTTP request
3. **Name** − We will change the name to reflect the action what we want to achieve.
4. **Server Name or IP** − Here, we have to type the web server name. (http:// part is not written this is only the name of the server or its IP)
5. **Protocol** − We will keep this blank, which means we want HTTP as the protocol.
6. **Path** − We will type path as / (slash). It means we want the root page of the server.

Add Listener

Add View Results Tree Listener under the Thread Group (User) node. It will ensure that the results of the Sampler will be available to view in this Listener node element.

1. Right-click the Thread Group (Users)
2. Choose Add > Listener > View Results Tree option

Run the Test Plan

View the Output