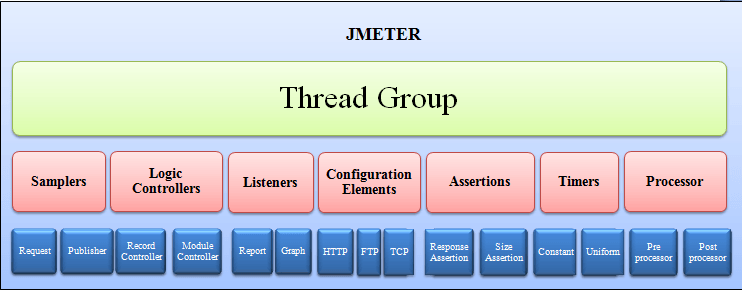
**Elements of JMeter**

The different components of JMeter are called Elements. Each Element is designed for a specific purpose.

The figure below gives some common elements in JMeter.



Remaining components will be discussed as when they are used in the succeeding tutorials. The elements discussed in this tutorial are

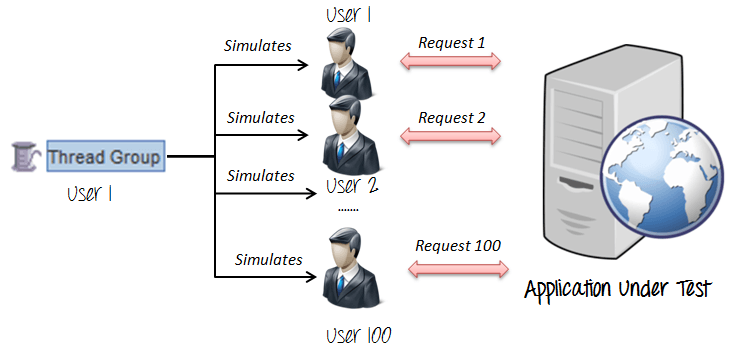
* Thread Group
* Samplers
* Listeners
* Configuration

## 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.

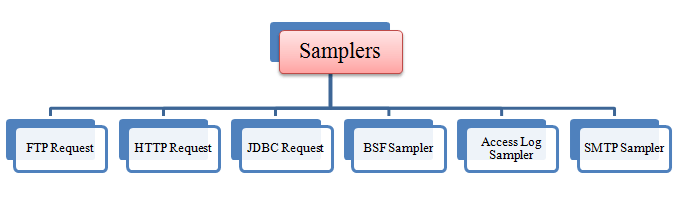
For example, if you set the number of threads as 100; JMeter will create and simulate 100 user requests to the server under test



## Samplers

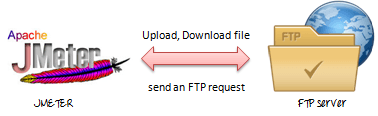
As we know already that JMeter supports testing HTTP, FTP, JDBC and many other protocols.

The user request could be FTP Request, HTTP Request, JDBC Request…Etc.

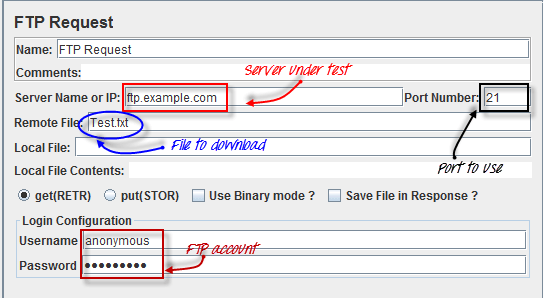


## FTP request:

Let’s imagine you want to performance test an FTP server. You can use an FTP request sampler in JMeter to do this task. This controller lets you send an FTP “download file” or “upload file” request to an FTP server.



For example, if you want to download a file “Test.txt” from an FTP server under test, you need to configure some parameters in JMeter as the figure below

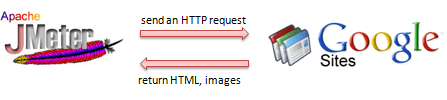


JMeter will send FTP command to FTP server ftp.example.com, and then download a file Test.txt from that server.

## HTTP request:

This sampler lets you send an HTTP/HTTPS request to a web server.

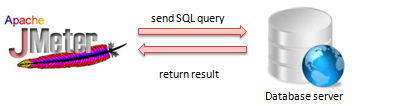
Consider the example below. JMeter sends an HTTP request to Google website and retrieve HTML files or image from this website.



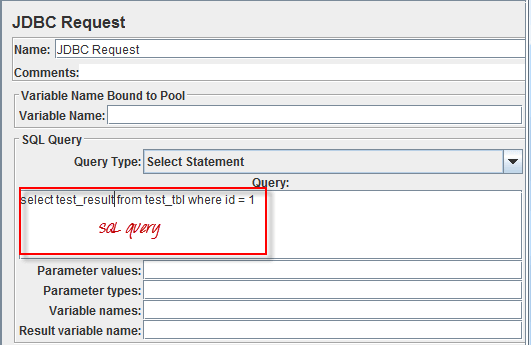
In the tutorial JMeter Performance Testing, we will explain more about this HTTP request.

## JDBC request:

This sampler lets you execute Database Performance Testing. It sends a JDBC Request (an SQL query) to a database.



For example, a database server has a field test\_result stored in a table name test\_tbl. You want to query this data from the database server; you can configure JMeter to send a SQL query to this server to retrieve data.



## BSF Sampler:

This sampler allows you to write a sampler using a BSF scripting language.

Here is an example of BSF Sampler in JMeter

Access Log Sampler:

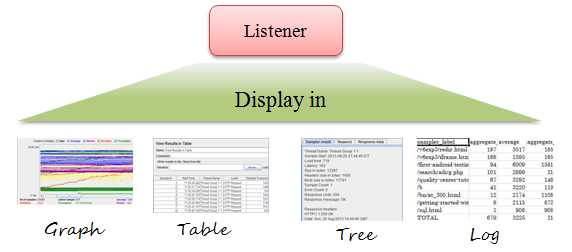
This sampler allows you to read access logs and generate HTTP requests. The log could be image, Html, CSS…

## SMTP Sampler:

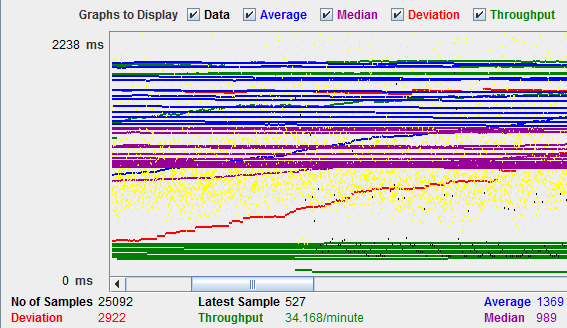
If you want to test a mail server, you can use SMTP sampler. This sampler is used to send email messages using the SMTP protocol.

## Listeners

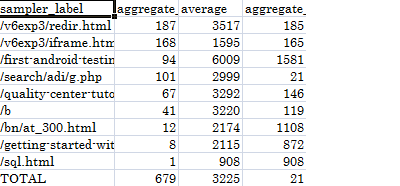
Listeners: shows the results of the test execution. They can show results in a different format such as a tree, table, graph or log file



Graph result listeners display the server response times on a Graph



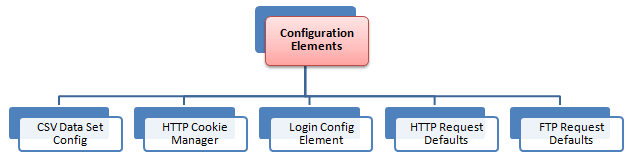
Log show summary of a test results in the text file



## Configuration Elements

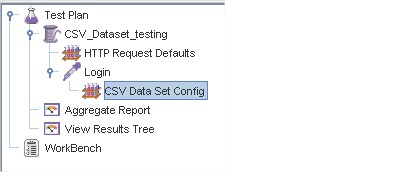
set up defaults and variables for later use by samplers.

The figure below shows some commonly used configuration elements in JMeter

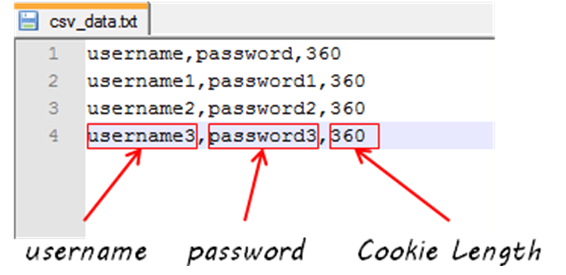


## CSV Data Set Config:

Suppose you want to test a website for 100 users signing-in with different credentials. You do not need to record the script 100 times! You can parameterization the script to enter different login credentials. This login information (e.g. Username, password) could be stored in a text file. JMeter has an element that allows you to read different parameters from that text file. It is “CSV Data Set Config”, which is used to read lines from a file, and split them into variables.



This is an example of CSV Data. It’s a text file which contains user and password to login your target website



## HTTP Cookie Manager

Let’s understand this with an example –

You used your browser (Firefox, IE…Etc) to browse www.google.com

You log in with your user and password.

Your username and password will be stored in your computer as cookies.

Next time, when you visit www.google.com, you don’t need to do log in again because your browser will use your cookies as user data to log in.

HTTP Cookie Manager also has the same feature as a web browser. If you have an HTTP Request and the response contains a cookie, the Cookie Manager automatically stores that cookie and will use it for all future requests to that particular website.

### HTTP request default

This element lets you set default values that your HTTP Request controllers use.

For example,

You are sending 100 HTTP requests to the server google.com

You would have to manually enter server name = google.com for all these 100 requests

Instead, you could add a single HTTP request defaults with the “Server Name or IP” field = google.com

No need to type 100 times!

