softech solutions

Elements of JMeter

Lecture Notes

A Complete Guide for Beginners

Softech Solutions Inc.

www.technosolution.tech

saney.alam@techsolutions.tech

Elements of JMeter

TABLE OF CONTENTS

ELEMENTS OF JMETER

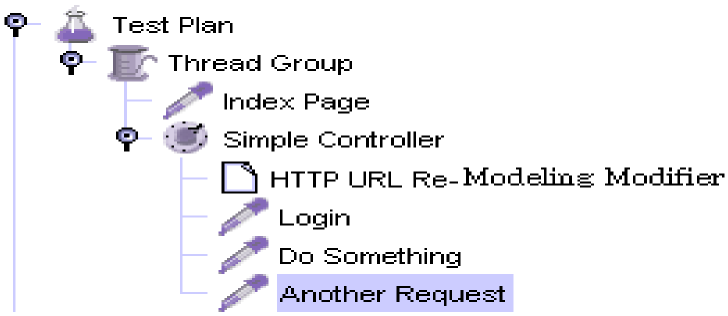
The different part or component of JMeter is called Element which co-relates with each other but designed for different-different purpose. Before start working on JMeter, it is best to know all components or Elements of JMeter with full detail description. From this article you will get to know the Elements of JMeter and why they are useful in JMeter.

Given below figure presents you some common elements in JMeter,

- Test Plan
- Thread Group
- Controllers
- Listeners
- Timers
- Configuration Elements
- Pre-Processor Elements
- Post-Processor Elements
- Execution order of Test Elements

TEST PLAN

A test plan is the top level body of JMeter, explains sequence of steps execute at run time. A final test plan made up of one or more Thread Groups, Sampler, logic controllers, listeners, timers, assertions, and configuration elements. Each Sampler can be preceded by one or more Pre-processor element, followed by Post-processor element, and/or Assertion element. It saves in Java Management Extensions (JMX) format.

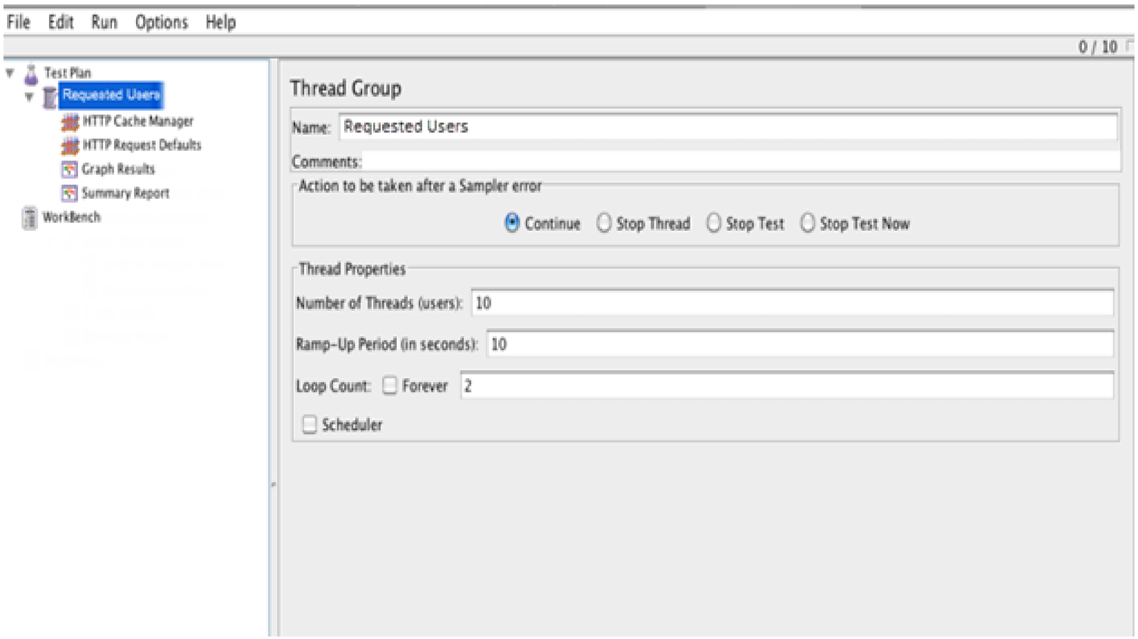


JMX is an open text based format, used across dissimilar systems. This format makes test plan to be open in a text editor and can use the “find and replace” function to speedily change a CMS name, web server name, or document ID that might appear hundreds of times throughout the test plan with very little determination.

THREAD GROUP

Thread Group is an initial stage of test plan. The name, Thread Groups represent a group of Threads. Under this group, each thread simulates one real user requests to the server.

A control on a thread group facilitates to set the number of threads on each group. Also can control Ramp Up Time and the number of test iterations.



Thread Group

Name: Requested Users

Comments:

Action to be taken after a Sampler error

● Continue ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

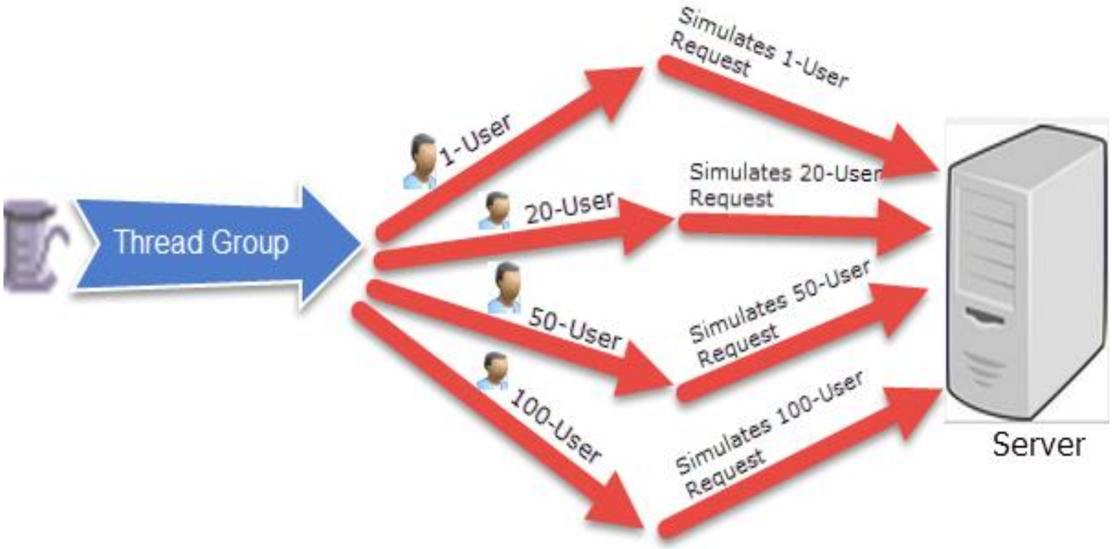
Number of Threads (users): 10

Ramp-Up Period (in seconds): 10

Loop Count: □ Forever 2

□ Scheduler

If you mention 10 numbers of threads then JMeter will generate and simulate 100 user requests to the server under test.


Thread Group
1-User
20-User
50-User
100-User
Simulates 1-User Request
Simulates 20-User Request
Simulates 50-User Request
Simulates 100-User Request
Server

CONTROLLERS

There are two types of controllers:

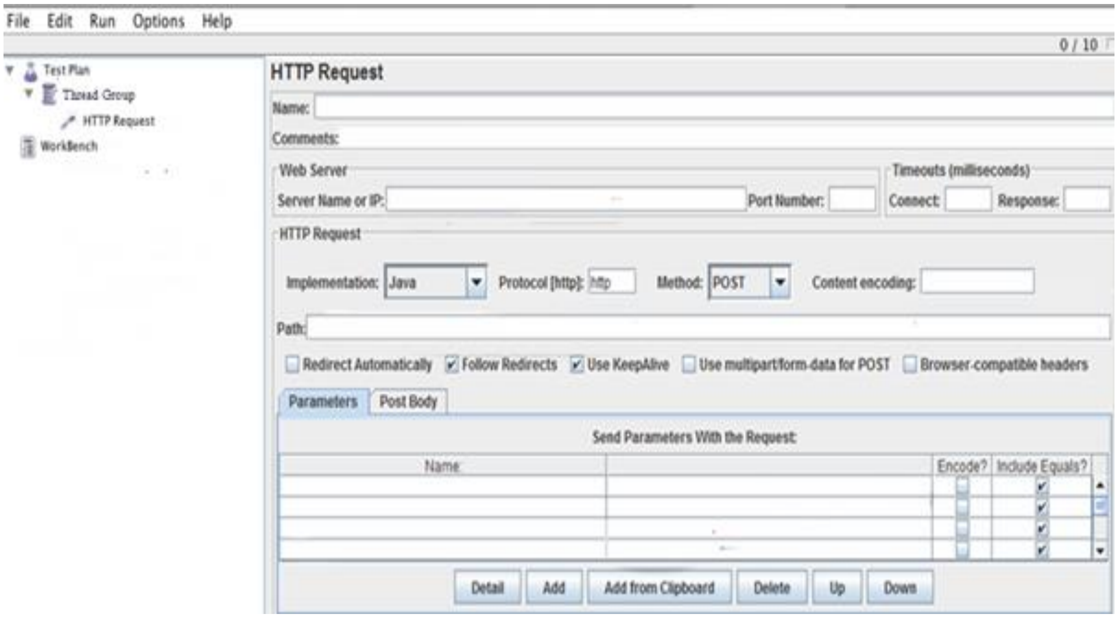
1) Samplers
2) Logical Controllers

SAMPLERS

Samplers facilitate JMeter to deliver explicit types of requests to the server. It simulates a user's request for a page from the target server. So that, to avail POST, GET, DELETE functions on a HTTP service, user can add HTTP Request sampler. Apart from HTTP Request sampler, there are other simpler too,

- HTTP Request
- FTP Request
- JDBC Request
- Java Request
- SOAP/XML Request
- RPC Requests

An HTTP Request Sampler looks like this figure,



Implementation: Java

Protocol [http]: http

Method: POST

Content encoding:

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

LOGICAL CONTROLLERS

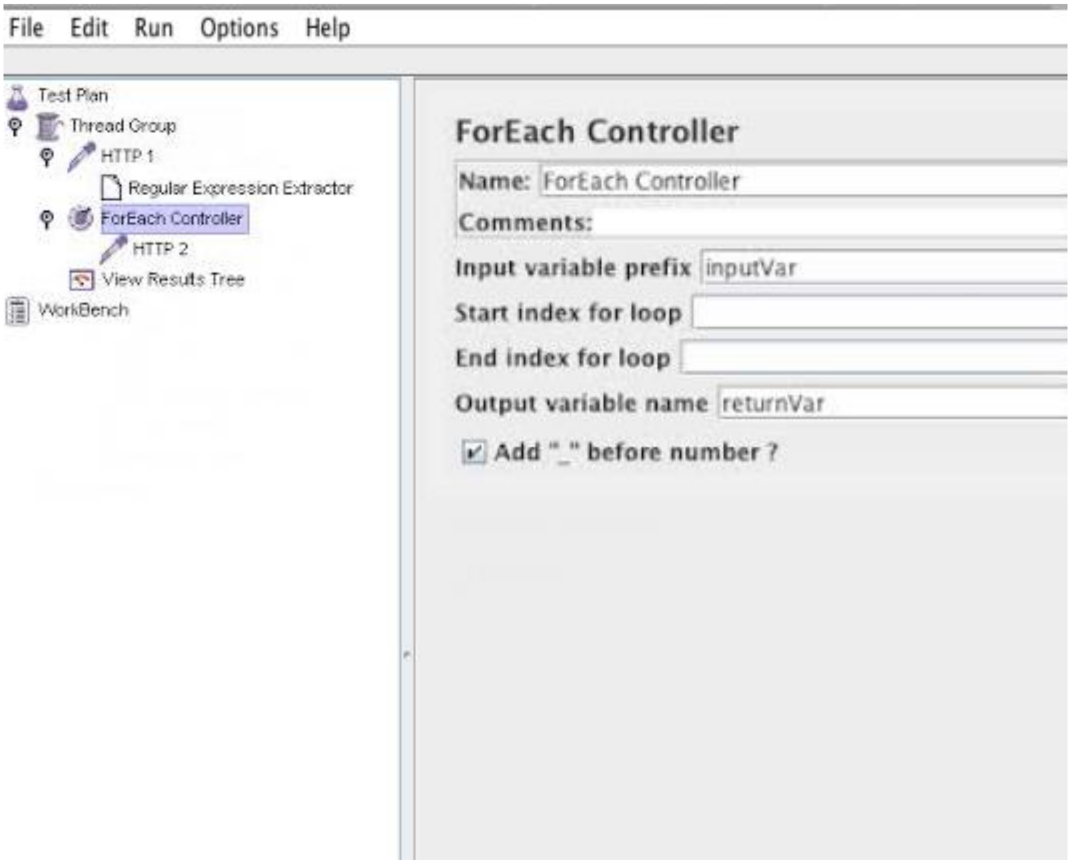
Logic Controllers decide the order of processing of Samplers in a Thread. It offers a mechanism to control the flow of the thread group. Logic Controllers facilitates to customize the logic that JMeter uses to resolve when to send requests. Logic Controllers can alter the order of requests coming from their child elements.

Logic Controllers of JMeter provide,

- Runtime Controller
- IFController
- 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

A ForEach Controller Control Panel looks like the following figure,



ForEach Controller

Name: ForEach Controller

Comments:

Input variable prefix inputVar

Start index for loop

End index for loop

Output variable name returnVar

☑ Add "_" before number ?

TEST FRAGMENTS

Test Fragments is a different type of element located at the same level as Thread Group element. It is eminent from a Thread Group in that it is not performed unless it is referenced by either a Module Controller or an Include_Controller. This element is morally for code re-use within Test Plans.

LISTENERS

Listeners facilitates viewers to view Samplers result in the form of tables, graphs, trees or simple text in some log files and also give pictorial access to the data collected by

JMeter about those test cases as a Sampler component of JMeter is executed. Listeners offer a means to collect, save, and view the results of a test plan and store results in XML format, or a more efficient (but less detailed) CSV format. Their output can also be viewed directly within the JMeter console.

It basically adjusts anywhere in the test, moreover under the test plan, gather data only from elements at or below their level.

Listeners of JMeter provide these many things,

- Graph Results
- Spline Visualizer
- Assertion Results
- Simple Data Writer
- Monitor Results
- Distribution Graph (alpha)
- Aggregate Graph
- Mailer Visualizer
- BeanShell Listener
- Summary Report
- Sample Result Save Configuration
- Graph Full Results
- View Results Tree
- Aggregate Report
- View Results in Table

Timers

To perform load/stress testing on your application, you are using threads, controllers and samplers then JMeter will just blast your application with nonstop requests. This is not the real environment or characteristic of the real traffic. JMeter thread sends request without stopping between each sampler. This not exactly what you want. We can add a timer element which will permit us to define a period to wait between each request.

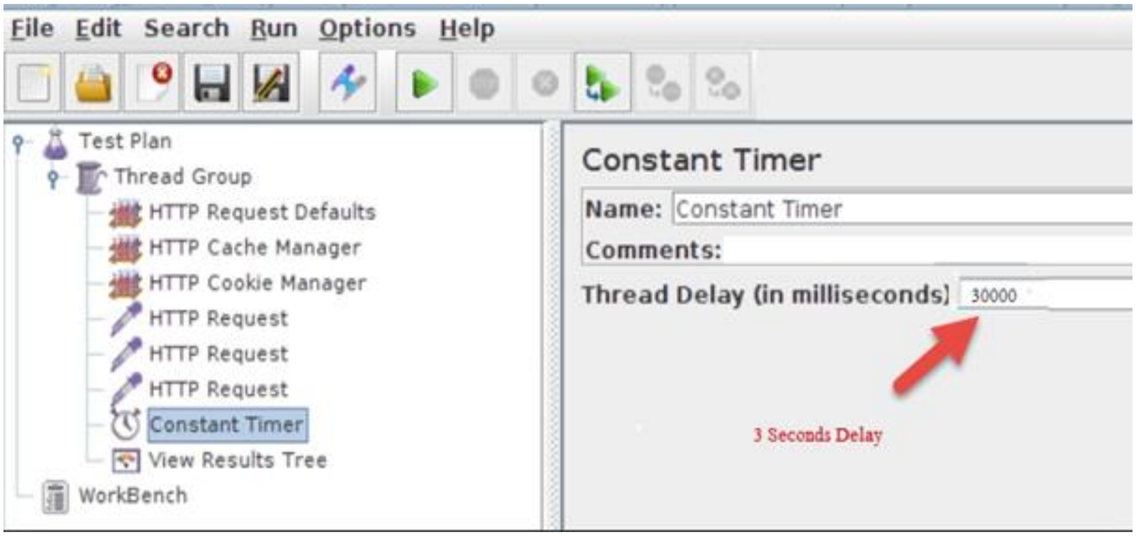
So, to simulate real traffic or create extreme load/stress – orderly, JMeter provides you Timer elements.

Just have a look on the types of Timers available,

- Synchronizing Timer
- JSR223 Time

- BeanShell Time
- Gaussian Random Timer
- Uniform Random Timer
- Constant Throughput Timer
- BSF Time
- Poisson Random Time

As an example, the Constant Timer Control Panel looks like this:



Name: Constant Timer

Comments:

Thread Delay (in milliseconds) 30000

Configuration Elements

Configuration Element is a simple element where you can collects the corporate configuration values of all samplers like webserver's hostname or database url etc.

These are some commonly used configuration elements in JMeter,

- Java Request Defaults
- LDAP Request Defaults
- LDAP Extended Request Defaults
- Keystore Configuration
- JDBC Connection Configuration
- Login Config Element
- CSV Data Set Config
- FTP Request Defaults
- TCP Sampler Config

- User Defined Variables
- HTTP Authorization Manager
- HTTP Cache Manager
- HTTP Cookie Manager
- HTTP Proxy Server
- HTTP Request Defaults
- HTTP Header Manager
- Simple Config Element
- Random Variable

Pre-Processor Elements

A Pre-Processor is an interesting thing which executes before a sampler executes. It is basically used to modify the settings of a Sample Request before it runs, or to update variables that are not extracted from response text.

These are the list made up of all the Pre-Processor Elements JMeter,

- JDBC PreProcessor
- JSR223 PreProcessor
- RegEx User Parameters
- BeanShell PreProcessor
- BSF PreProcessor
- HTML Link Parser
- HTTP URL Re-writing Modifier
- HTTP User Parameter Modifier
- User Parameters

Post-Processor Elements

A Post Processor gets executed after finishing a sampler execution process. This element is basically useful to procedure the response data, for example, to retrieve particular value for later use.

These are the list made up of all the Post-Processor Elements JMeter,

- CSS/JQuery Extractor
- BeanShell Post-Processor

- JSR223 Post-Processor
- JDBC Post-Processor
- Debug Post-Processor
- Regular Expression Extractor
- XPath Extractor
- Result Status Action Handler
- BSF Post-Processor

Test Elements, Execution Series

These are the execution order of the test plan elements,

- Configuration elements
- Pre-Processors
- Timers
- Sampler
- Post-Processors (unless Sample Result is null)
- Assertions (unless Sample Result is null)
- Listeners (unless Sample Result is null)