

JMETER - TEST PLAN ELEMENTS

<http://www.tutorialspoint.com>

Copyright © tutorialspoint.com

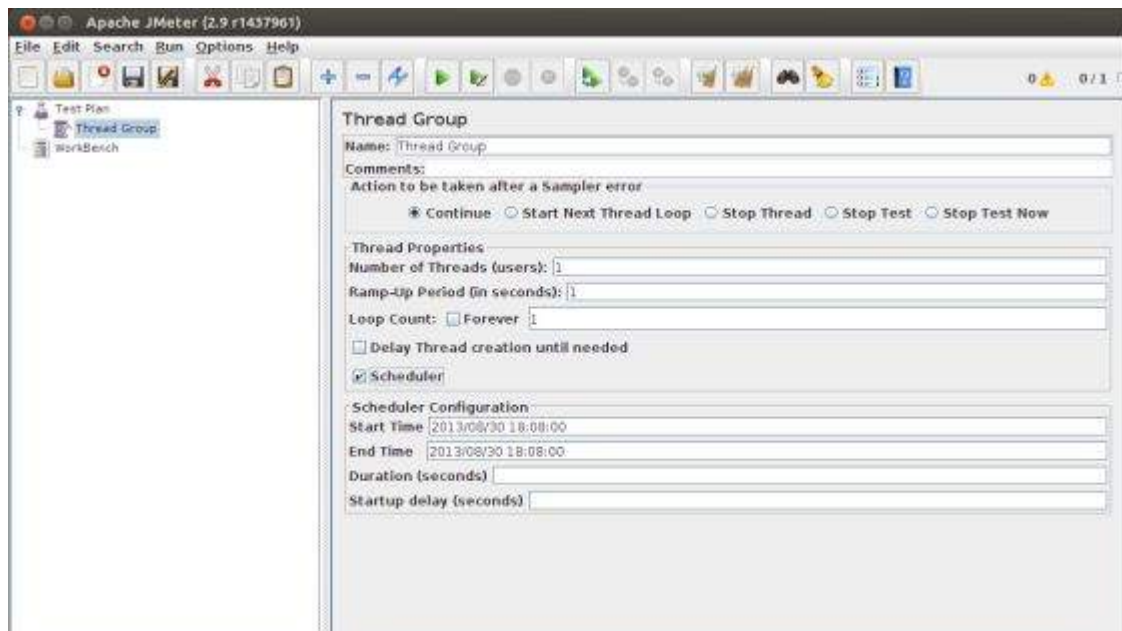
A JMeter Test Plan comprises of test elements discussed below. A Test Plan comprises of at least one Thread Group. Within each Thread Group, we may place a combination of one or more of other elements – Sampler, Logic Controller, Configuration Element, Listener, and Timer. Each Sampler can be preceded by one or more Pre-processor element, followed by Post-processor element, and/or Assertion element. Let us see each of these elements in detail –

Thread Group

Thread Group elements are the beginning points of your test plan. As the name suggests, the thread group elements control the number of threads JMeter will use during the test. We can also control the following via the Thread Group –

- Setting the number of threads
- Setting the ramp-up time
- Setting the number of test iterations

The Thread Group Control Panel looks like this –



The Thread Group Panel holds the following components –

- **Action to be taken after a Sampler error** – In case any error occurs during test execution, you may let the test either –
 - **Continue** to the next element in the test
 - **Stop Thread** to stop the current Thread.
 - **Stop Test** completely, in case you want to inspect the error before it continues running.
- **Number of Threads** – Simulates the number of users or connections to your server application.
- **Ramp-Up Period** Defines how long it will take JMeter to get all threads running.
- **Loop Count** – Defines the number of times to execute the test.
- **Scheduler checkbox** – Once selected, the Scheduler Configuration section appears at the bottom of the control panel.

- **Scheduler Configuration** – You can configure the start and end time of running the test.

Controllers

JMeter has two types of Controllers – *Samplers* and *Logic Controllers*.

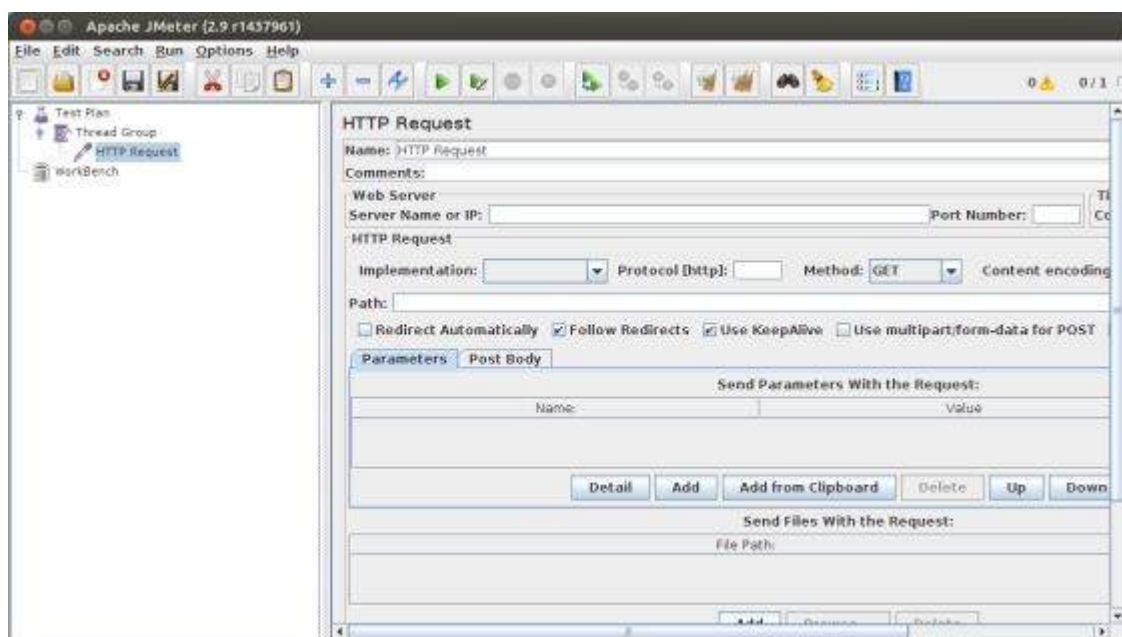
Samplers

Samplers allow JMeter to send specific types of requests to a server. They simulate a user request for a page from the target server. For example, you can add a HTTP Request sampler if you need to perform a POST, GET, or DELETE on a HTTP service.

Some useful samplers are –

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

The following screenshot shows an HTTP Request Sampler Control Panel –



Logic Controllers

Logic Controllers let you control the order of processing of Samplers in a Thread. Logic controllers can change the order of a request coming from any of their child elements. Some examples are – ForEach Controller, While Controller, Loop Controller, IF Controller, Run Time Controller, Interleave Controller, Throughput Controller, and Run Once Controller.

The following screenshot shows a Loop Controller Control Panel –





The following list consists of all the Logic Controllers JMeter provides –

- Simple Controller
- Loop Controller
- Once Only Controller
- Interleave Controller
- Random Controller
- Random Order Controller
- Throughput Controller
- Runtime Controller
- If Controller
- While Controller
- Switch Controller
- ForEach Controller
- Module Controller
- Include Controller
- Transaction Controller
- Recording Controller

Test Fragments

A Test Fragment is a special type of element placed 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 let you view the results of Samplers in the form of tables, graphs, trees, or simple text in some log files. They provide visual access to the data gathered by JMeter about the test cases as a Sampler component of JMeter is executed.

Listeners can be added anywhere in the test, including directly under the test plan. They will collect data only from elements at or below their level. The following list consists of all the Listeners JMeter provides –

- Sample Result Save Configuration
- Graph Full Results
- Graph Results
- Spline Visualizer
- Assertion Results
- View Results Tree

- Aggregate Report
- View Results in Table
- Simple Data Writer
- Monitor Results
- Distribution Graph *alpha*
- Aggregate Graph
- Mailer Visualizer
- BeanShell Listener
- Summary Report

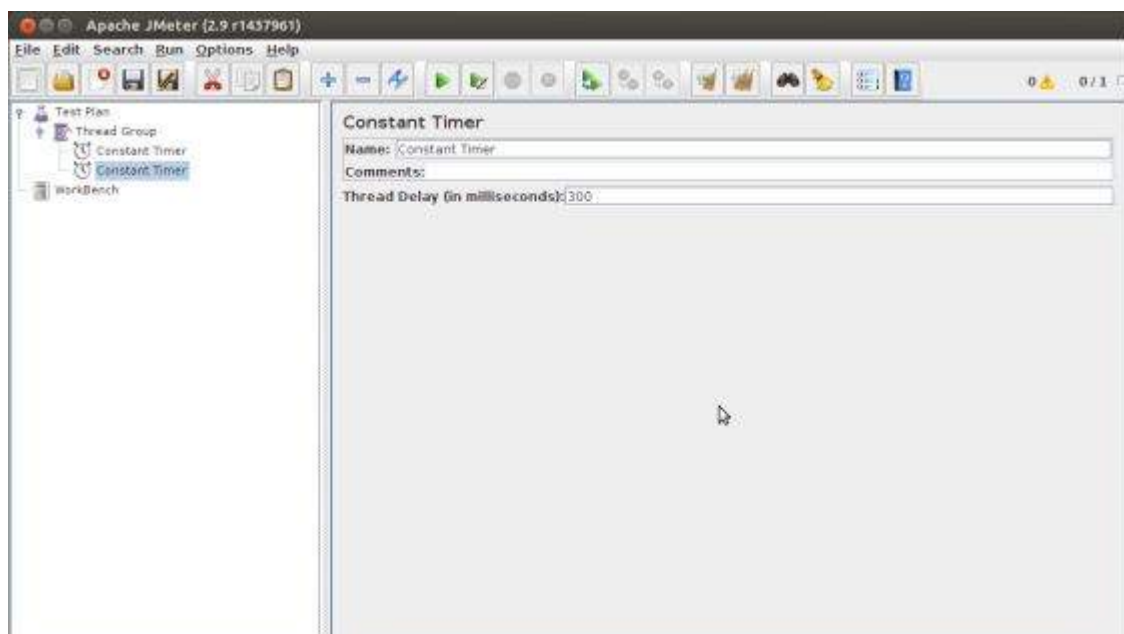
Timers

By default, a JMeter thread sends requests without pausing between each sampler. This may not be what you want. You can add a timer element which allows you to define a period to wait between each request.

The following list shows all the timers that JMeter provides –

- Constant Timer
- Gaussian Random Timer
- Uniform Random Timer
- Constant Throughput Timer
- Synchronizing Timer
- JSR223 Time
- BeanShell Time
- BSF Time
- Poisson Random Time

The following screenshot shows a Constant Timer Control Panel –



Assertions

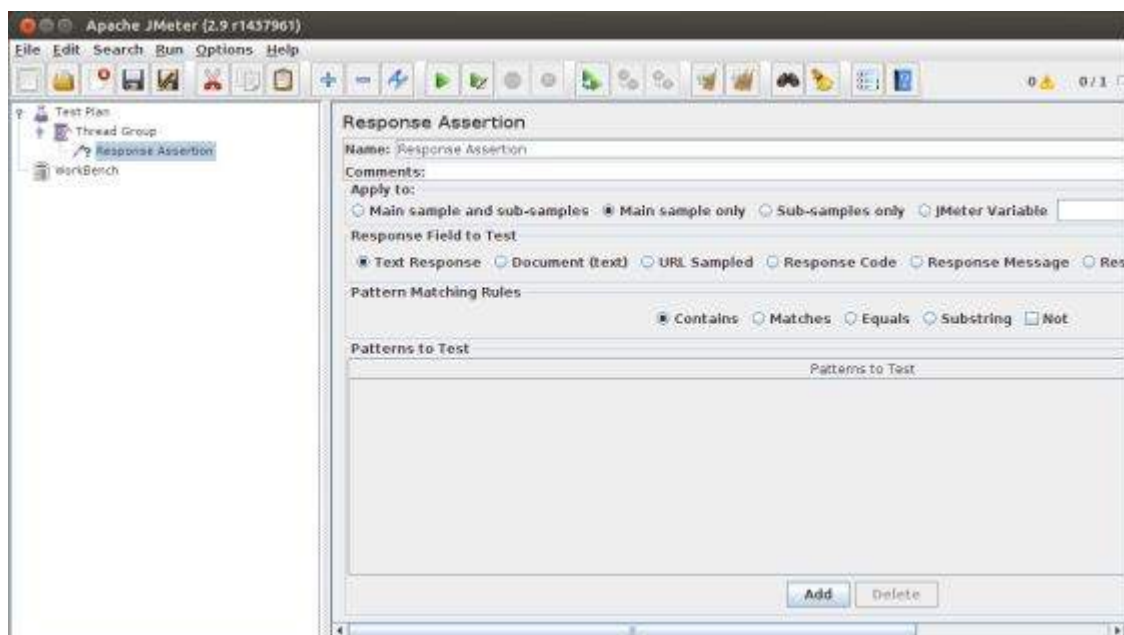
Assertions allow you to include some validation test on the response of your request made using a Sampler. Using assertions you can prove that your application is returning the correct data. JMeter

highlights when an assertion fails.

The following list consists of all the assertions JMeter provides –

- Beanshell Assertion
- BSF Assertion
- Compare Assertion
- JSR223 Assertion
- Response Assertion
- Duration Assertion
- Size Assertion
- XML Assertion
- BeanShell Assertion
- MD5Hex Assertion
- HTML Assertion
- XPath Assertion
- XML Schema Assertion

The following screenshot shows a Response Assertion Control Panel –



Configuration Elements

Configuration Elements allow you to create defaults and variables to be used by Samplers. They are used to add or modify requests made by Samplers.

They are executed at the start of the scope of which they are part, before any Samplers that are located in the same scope. Therefore, a Configuration Element is accessed only from inside the branch where it is placed.

The following list consists of all the Configuration Elements that JMeter provides –

- Counter
- CSV Data Set Config
- FTP Request Defaults
- HTTP Authorization Manager

- HTTP Cache Manager
- HTTP Cookie Manager
- HTTP Proxy Server
- HTTP Request Defaults
- HTTP Header Manager
- Java Request Defaults
- Keystore Configuration
- JDBC Connection Configuration
- Login Config Element
- LDAP Request Defaults
- LDAP Extended Request Defaults
- TCP Sampler Config
- User Defined Variables
- Simple Config Element
- Random Variable

Pre-processor Elements

A pre-processor element is something that runs just before a sampler executes. They are often used to modify the settings of a Sample Request just before it runs, or to update variables that are not extracted from response text.

The following list consists of all the pre-processor elements that JMeter provides –

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

Post-processor Elements

A post-processor executes after a sampler finishes its execution. This element is most often used to process the response data, for example, to retrieve a particular value for later use.

The following list consists of all the Post-Processor Elements JMeter provides –

- Regular Expression Extractor
- XPath Extractor
- Result Status Action Handler
- JSR223 PostProcessor
- JDBC PostProcessor
- BSF PostProcessor
- CSS/JQuery Extractor

- BeanShell PostProcessor
- Debug PostProcessor

Execution Order of Test Elements

Following is the execution order of the test plan elements –

- Configuration elements
- Pre-Processors
- Timers
- Sampler
- Post-Processors *unlessSampleResultisnull*
- Assertions *unlessSampleResultisnull*
- Listeners *unlessSampleResultisnull*

Loading [MathJax]/jax/output/HTML-CSS/jax.js