

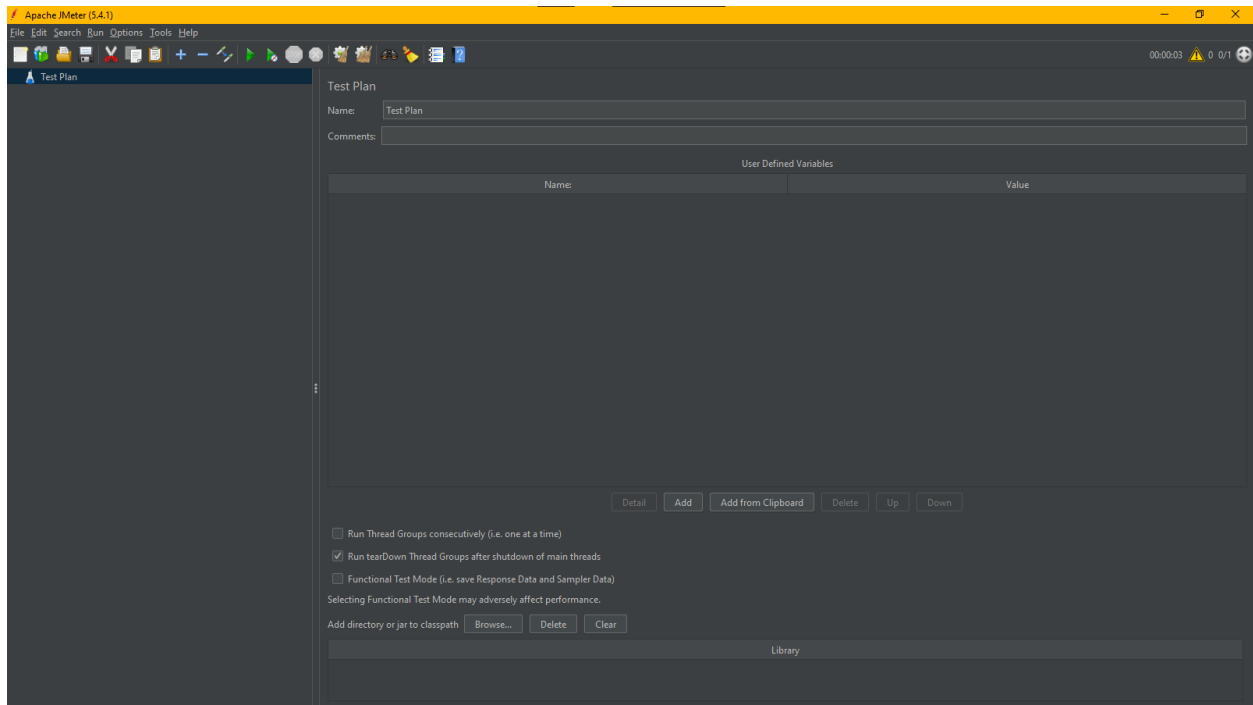
## Jmeter

*Apache Jmeter* Is An Open-Source, Pure Java Platform Software Which Is Designed To Load Test Functional Behavior And Measure Performance.

Initially, Jmeter Was Introduced For Load And Performance Test Web Applications, But Later On Its Scope Has Widened And Can Perform Load And Performance Test On Web Pages, Web Applications And Static Or Dynamic Resources Like *Database, Rest Webservices, Ldap, Java Objects* And More.

### Test Plan

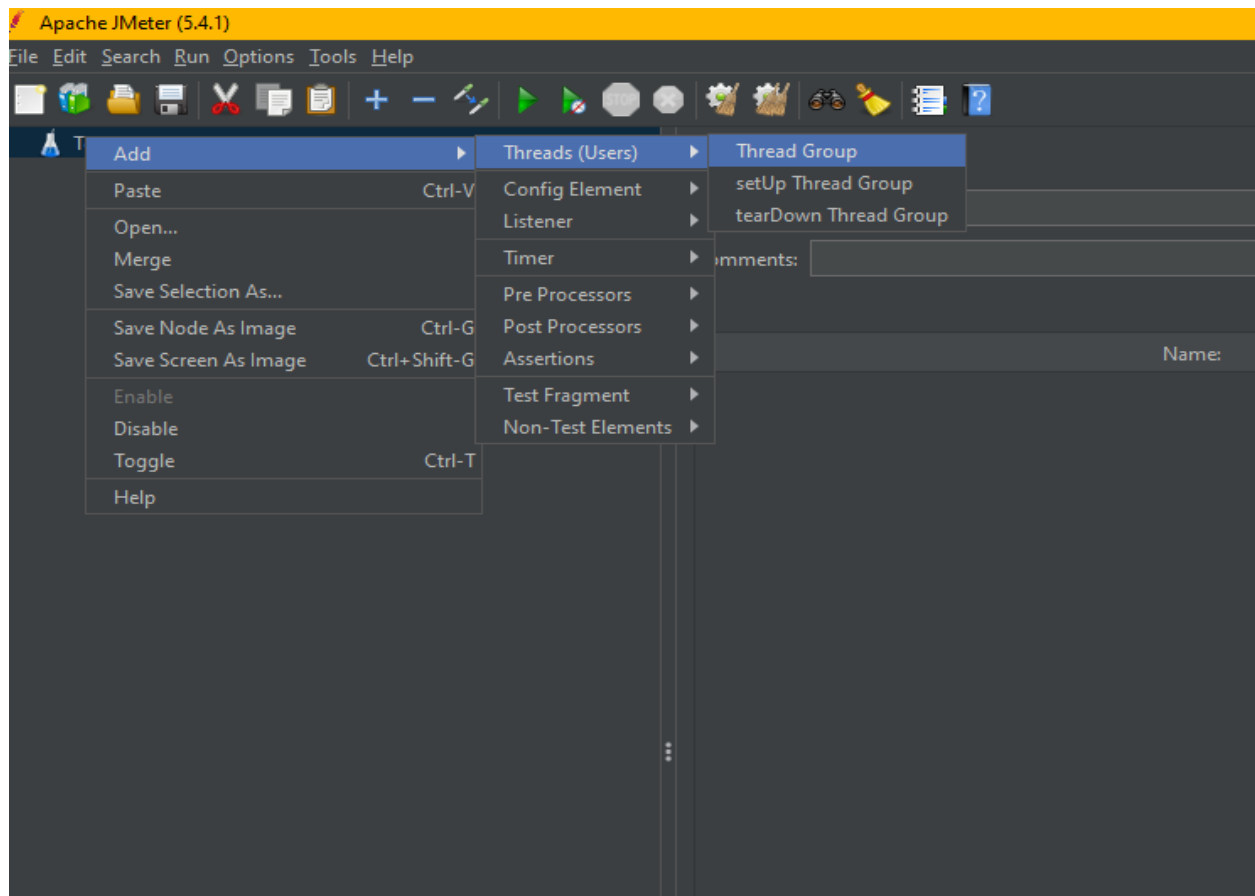
The Test Plan Is Where The Overall Settings For A Test Are Specified.



## Elements In A Test Plan

### Thread

A Thread Group Defines A Pool Of Users That Will Execute A Particular Test Case Against Your Server. In The Thread Group Gui, You Can Control The Number Of Users Simulated (Number Of Threads), The Ramp Up Time (How Long It Takes To Start All The Threads), The Number Of Times To Perform The Test, And Optionally, A Start And Stop Time For The Test.



Thread Group

Name:

Comments:

Action to be taken after a Sampler error

☒ Continue
 ☐ Start Next Thread Loop
 ☐ Stop Thread
 ☐ Stop Test
 ☐ Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: ☐ Infinite

☒ Same user on each iteration  
☐ Delay Thread creation until needed  
☐ Specify Thread lifetime

Duration (seconds):

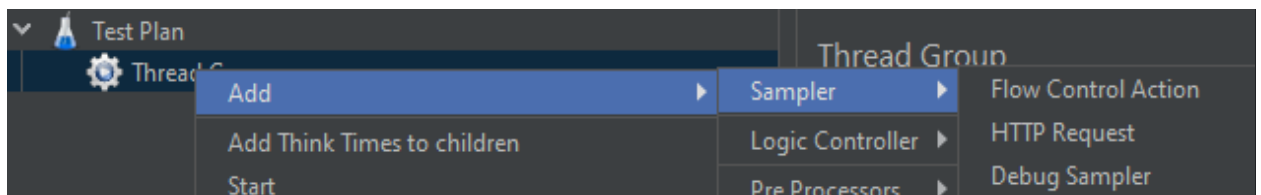
Startup delay (seconds):

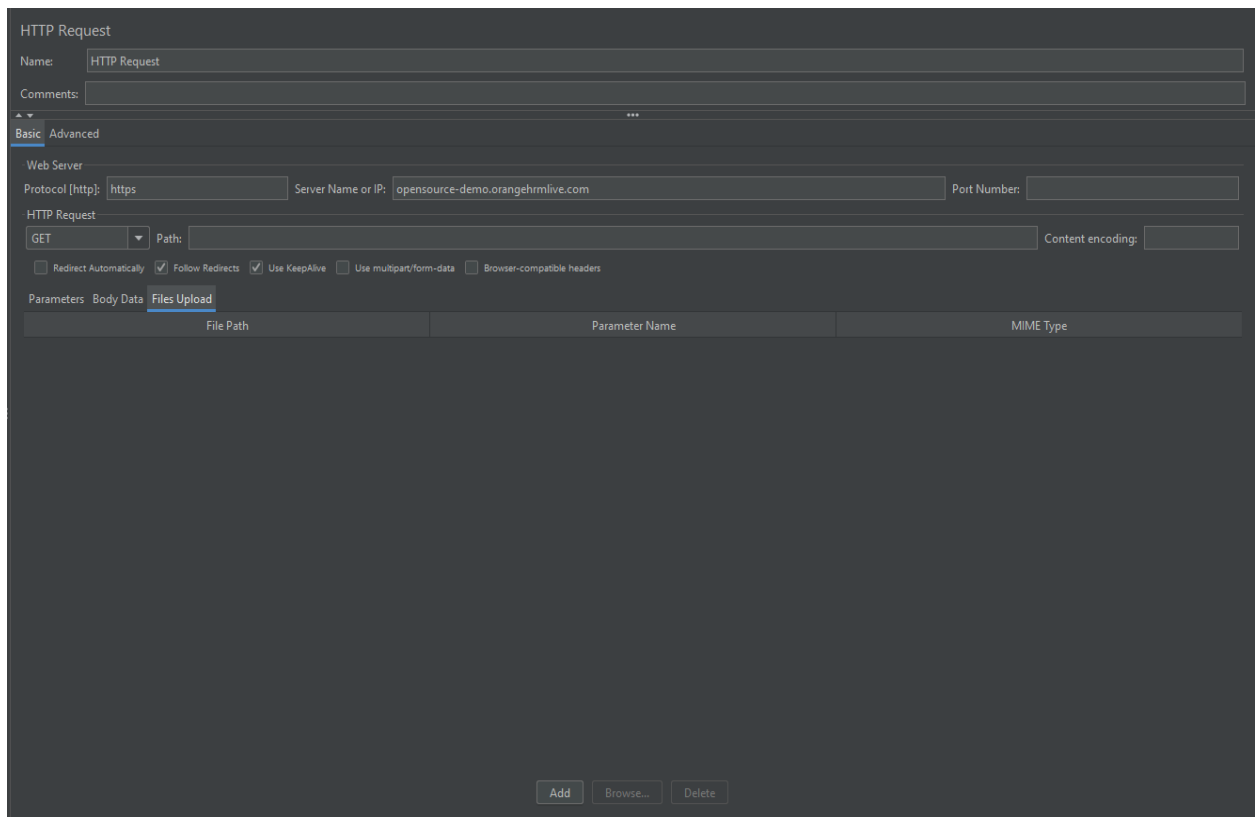
## Sampler

Samplers In Jmeter Allows Jmeter To Send Different Types Of Requests To A Server

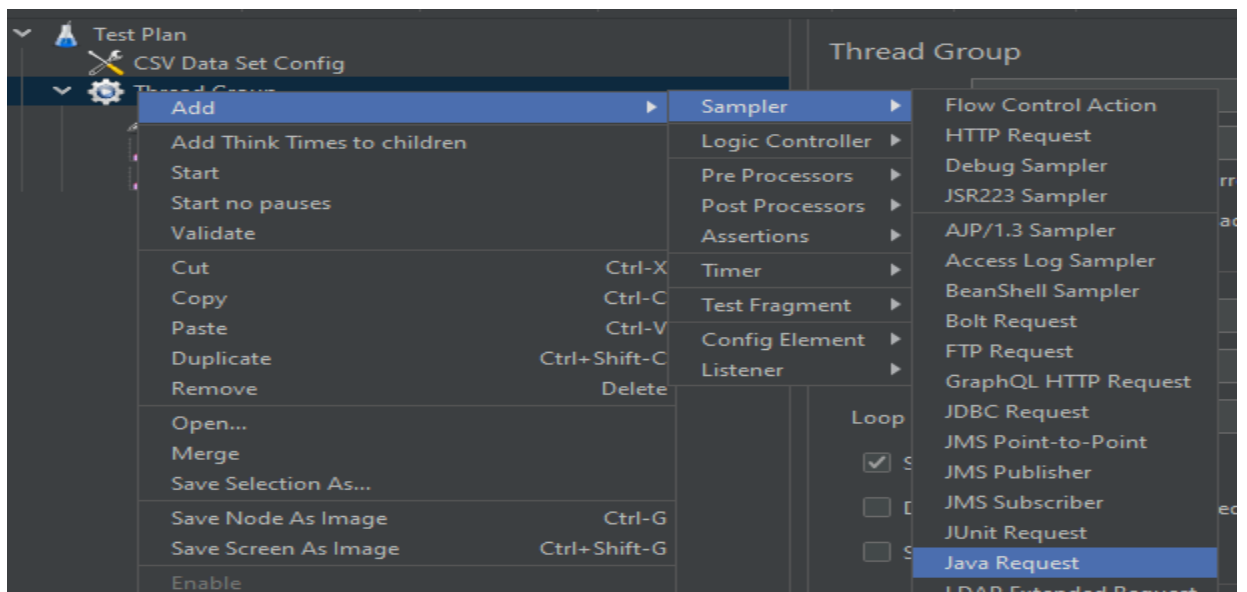
- **Http Request**

A Sampler That Lets You Send An Http/Https Request To A Web Server For Load Testing. There Are Different Methods The Sampler Is Able To Use, Like: Get. Post.





- **Java Request**



Java Request

Name:

Comments:

Send Parameters With the Request:

Name:	Value
Sleep_Time	100
Sleep_Mask	0xFF
Label	
ResponseCode	
ResponseMessage	
Status	OK
SamplerData	
ResultData	

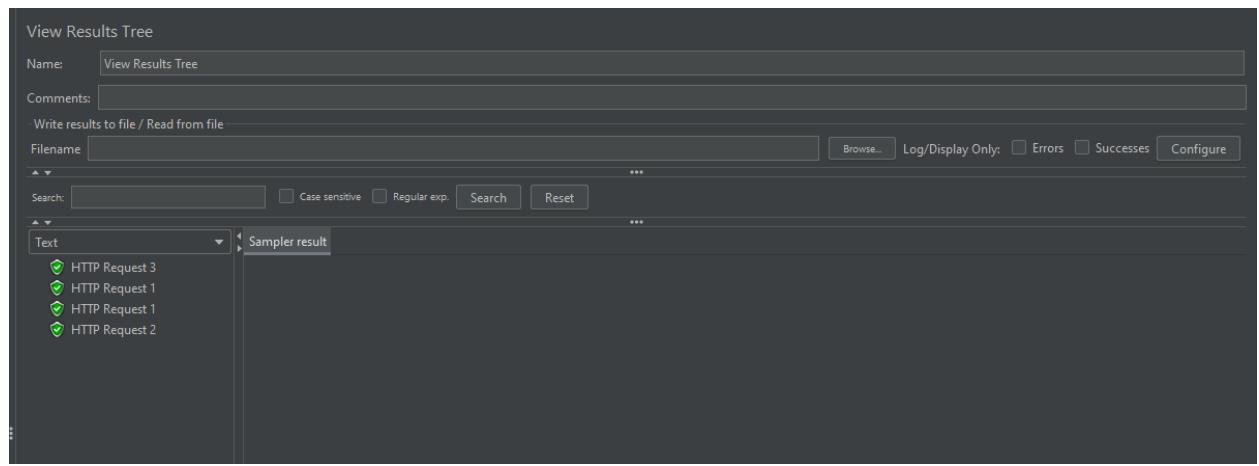
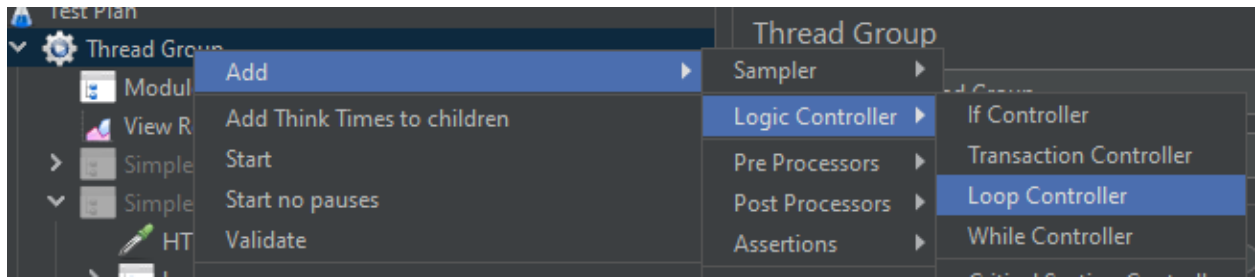
## Logic Controller

Logic Controllers Are The Test Plan Elements That Are Used To Customize The Order Of Processing Of Samplers And Other Elements Added As Child. Primarily, Logic Controllers Are Used With Sampler Requests To Perform Various Customization Like – Altering Their Order Of Processing, Grouping Them As A Single Transaction Or Running The Requests In Loop Etc.

Steps To Launch A Logic Controller-**Right Click On Thread Group -> Hover Over Add -> Hover Over Logic Controllers -> Click On The Required Logic Controllers**

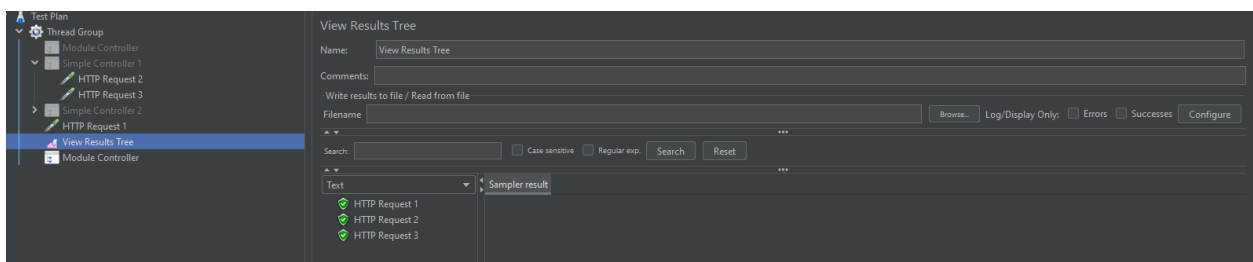
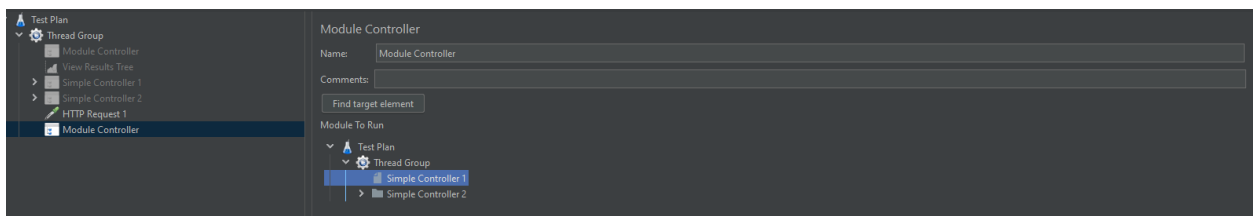
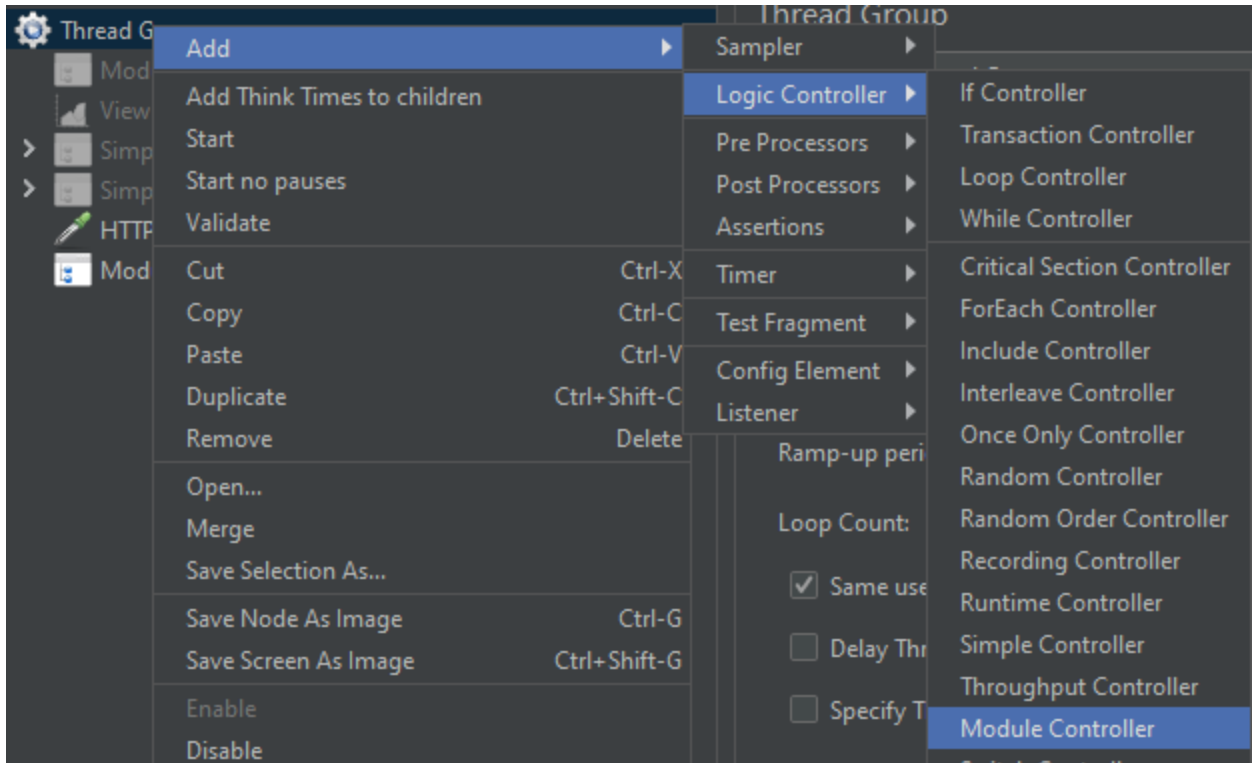
- Loop Controller

It Allows To Execute The Operations Specified As Child Elements In A Loop With Iteration Value Specified In Its Control Panel.



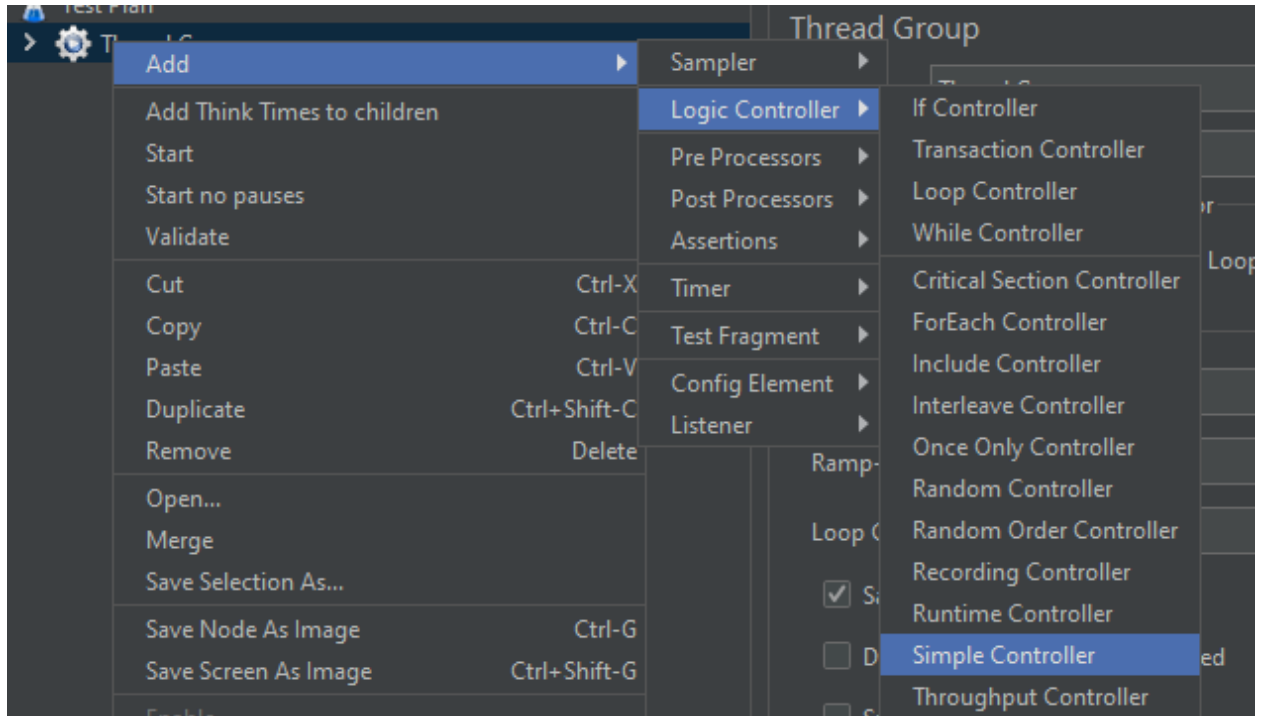
- Module Controller

Using Module Controller, We Can Reuse A Test Fragment (E.G., A Sampler) Into Our Script Again By Selecting The Module From The Module Controller's Control Panel.



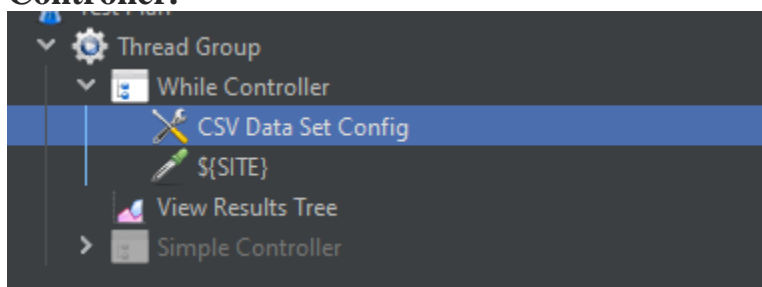
- Simple Controller

This Controller Is Just A Placeholder For Grouping And Ordering The Different Elements Of The Test Plan.



- While Controller

The While Controller Is Used To Run The Child Elements Inside It Till The Value Specified In Its Control Panel Is Evaluated To False. Steps To Launch A Logic Controller-  
**Right Click On Thread Group -> Hover Over Add -> Hover Over Logic Controllers -> Click On The While Controller.**





In Here We Take The Urls From A Csv, That Is Why We Use Csv Dataset Config,  
And We Use A Http Request Too.

In The Csv,

B14		
	A	B
1	SITE	MAIN
2	URL-I	<a href="https://opensource-demo.orangehrmlive.com">opensource-demo.orangehrmlive.com</a>
3	URL-II	<a href="https://opensource-demo.orangehrmlive.com">opensource-demo.orangehrmlive.com</a>
4	URL-III	<a href="https://opensource-demo.orangehrmlive.com">opensource-demo.orangehrmlive.com</a>
5		
6		
7		

CSV Data Set Config

Name: CSV Data Set Config

Comments:

Configure the CSV Data Source

Filename: C:/Users/dijal/OneDrive/Desktop/JMETER TESTING/csv fo while/url.csv Browse...

File encoding:

Variable Names (comma-delimited):

Ignore first line (only used if Variable Names is not empty): False

Delimiter (use '\t' for tab): ,

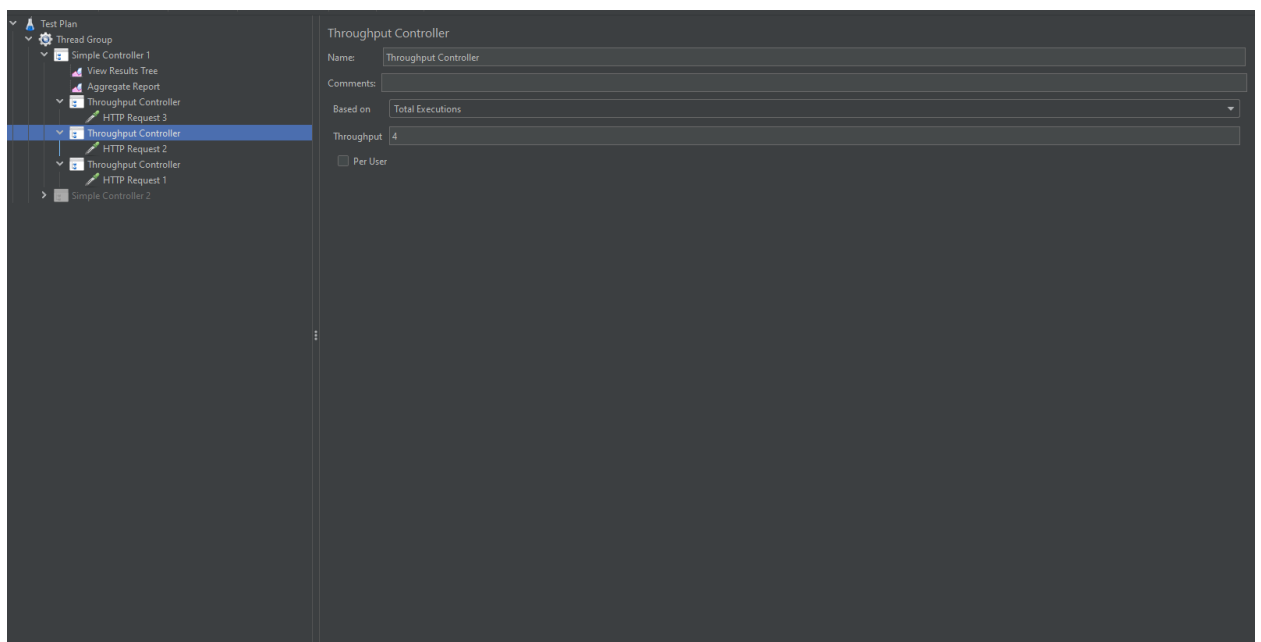
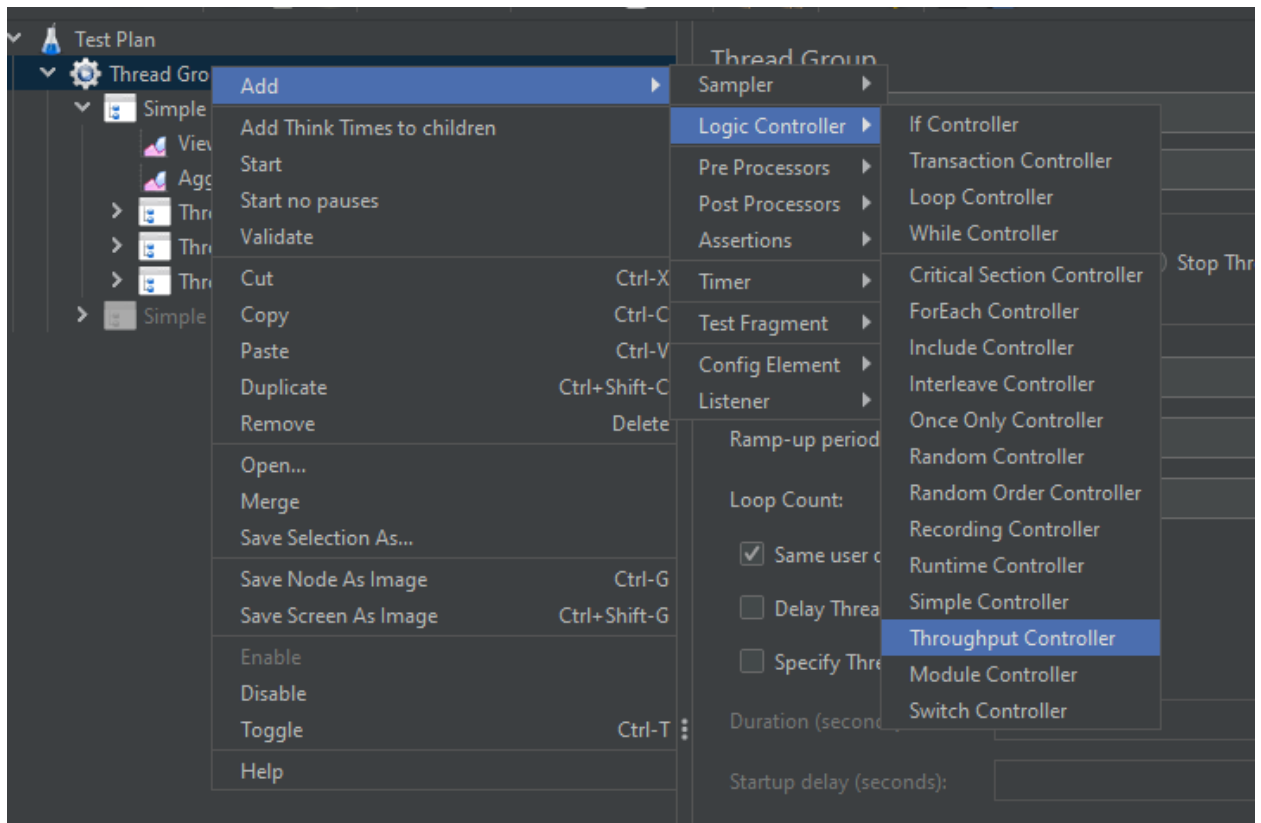
Allow quoted data?: False

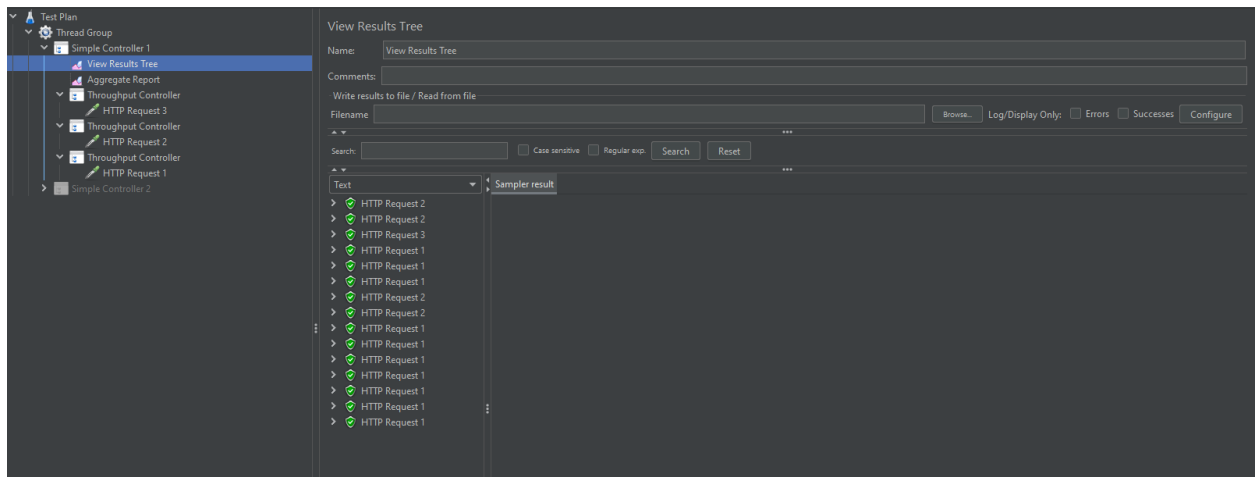
Recycle on EOF?: False

Stop thread on EOF?: True

Sharing mode: All threads







## Assertions

- **Response Assertion**

The Response Assertion Used In Test Scripts To Validate A Pattern In The Response Body, Header, Code, Message Etc. There Are Different Pattern Matching Rules To Validate The Response Like-

- Contains – If The Response Text Contains The Regular Expression To Be Matched
- Matches – If The Whole Response Text Matches The Regular Expression
- Equals – If The Whole Response Text Matches The Pattern(Not Regular Expression But The Pattern String)
- Substring – If The Response Text Contains The Pattern(Not Regular Expression)

- Not – To Check That The Pattern Is Not Present In The Response Text

The screenshot shows the 'Response Assertion' configuration window. The 'Name' field is set to 'Response Assertion'. The 'Apply to' section has 'Main sample only' selected. Under 'Field to Test', 'Response Code' is selected. The 'Pattern Matching Rules' section shows 'Substring' selected with 'Not' and 'Or' checkboxes. The 'Patterns to Test' table contains one entry: '200'.

	Patterns to Test
1	200

- **Size Assertion**

The Size Assertion Is Used To Validate The Size Of The Response With A Specified Value In Bytes.

The screenshot shows the 'Size Assertion' configuration window. The 'Name' field is set to 'Size Assertion'. The 'Apply to' section has 'Main sample only' selected. Under 'Response Size Field to Test', 'Full Response' is selected. The 'Size to Assert' section shows 'Size in bytes' set to '943252'. The 'Type of Comparison' section has '<' selected.

- **Duration Assertion**

The Duration Assertion Is Used To Validate That The Sampler Request Gets Processed Within A Specified Amount Of Time.

Duration Assertion

Name:

Comments:

Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only

Duration to Assert

Duration in milliseconds:

- **Xpath Assertion**

The Xpath Assertion Is Used To Validate The Response Using Xpath Expressions.

**XPath Assertion**

Name:

Comments:

Apply to:

☐ Main sample and sub-samples
 ☒ Main sample only
 ☐ Sub-samples only
 ☐ JMeter Variable Name to use

XML Parsing Options

☒ Use Tidy (tolerant parser)
 ☒ Quiet
 ☐ Report errors
 ☐ Show warnings

☐ Use Namespaces
 ☐ Validate XML
 ☐ Ignore Whitespace
 ☐ Fetch external DTDs

XPath Assertion

☐ Invert assertion(will fail if XPath expression matches)

1 `/html/head/title`

## Results

**View Results Tree**

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only: ☐ Errors ☐ Successes

Search:  ☐ Case sensitive ☐ Regular exp.

Text

☒ HTTP Request
 ☒ Assertion result

Assertion Results

Name:

Comments:

Write results to file / Read from file

Filename   Log/Display Only: ☐ Errors ☐ Successes

Assertions:

HTTP Request

## **Pre-Processors**

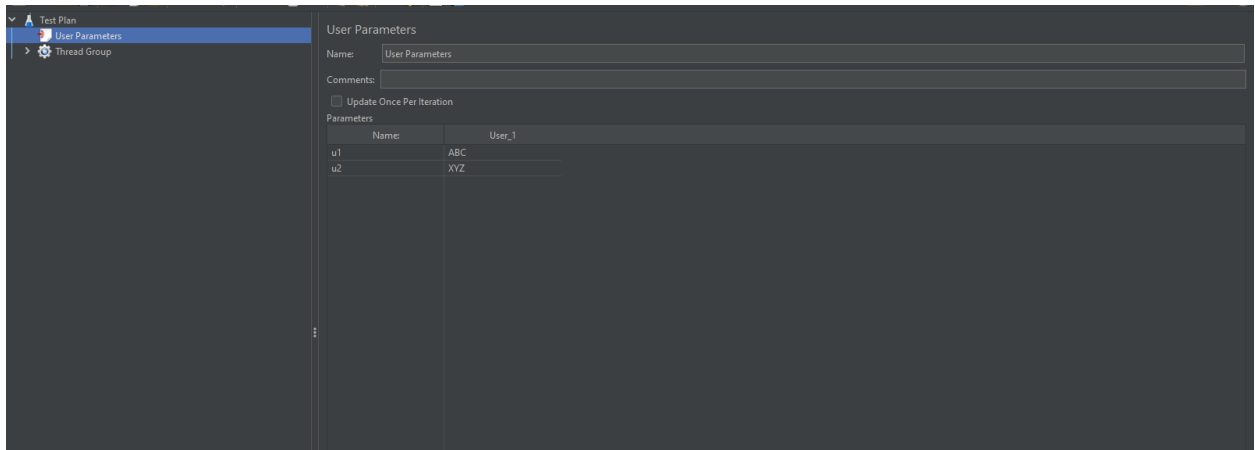
The Pre-Processor Elements Are Used To Modify The Sampler Requests Before Their Processing (Hence The Name Pre-Processor). **How To Add A Pre-Processors In Jmeter-**

**Right Click On Either Of Thread Group/Logic Controller -> Hover Over 'Add' -> Hover Over 'Pre-Processors' -> Click On The Required Pre-Processor Element**

- **User Parameters**

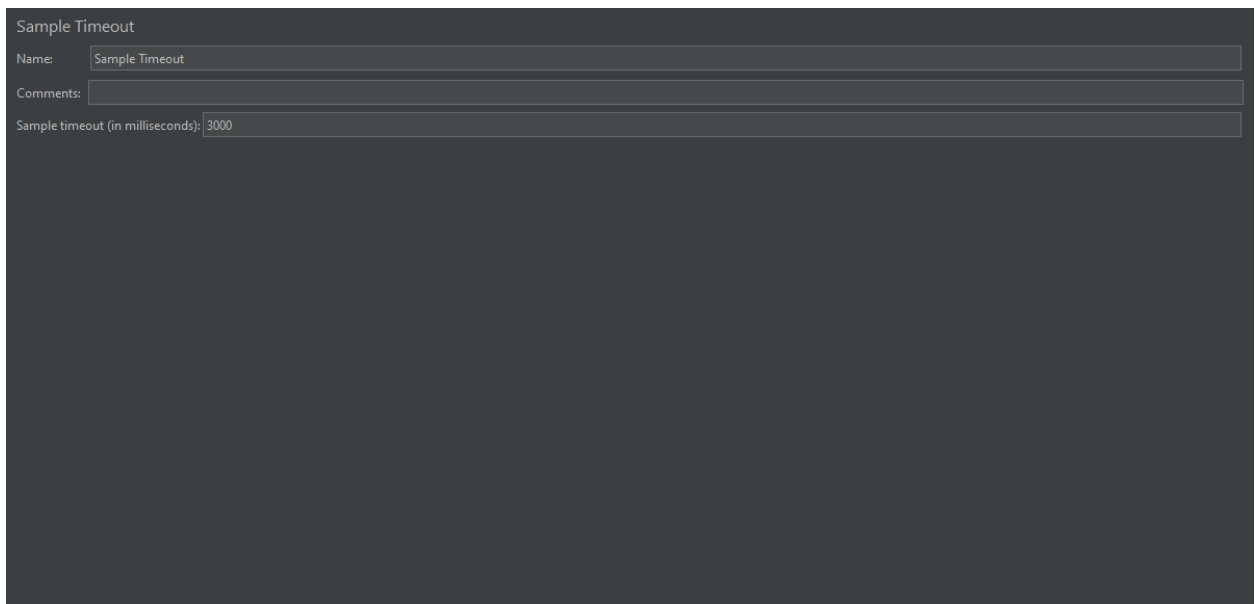
The User Parameters Are Used To Specify Values For User Variables Used Within Thread Groups.





- **Sample Timeout**

Sample Timeout Sets A Maximum Timeout For A Particular Sampler And Executes When The Response Time Exceeds The Given Timeout Value And Instruct Jmeter To Fire The Next Request.



- **Html Link Parser**

The Html Link Parser Is Used To Extract Links From Html Response Fetched From Server.

HTML Link Parser

Name:

HTML Link Parser

Comments:

## **Results**

View Results Tree

Name:

View Results Tree

Comments:

Write results to file / Read from file

Filename:

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Search:

☐ Case sensitive

☐ Regular exp.

Search

Reset

Text

> HTTP Request ABC

HTTP Request XYZ

Sampler result

Request

Response data

Request Body

Request Headers

1

2

3

4

5

6

7

GET https://opensource-demo.orangehrmlive.com/

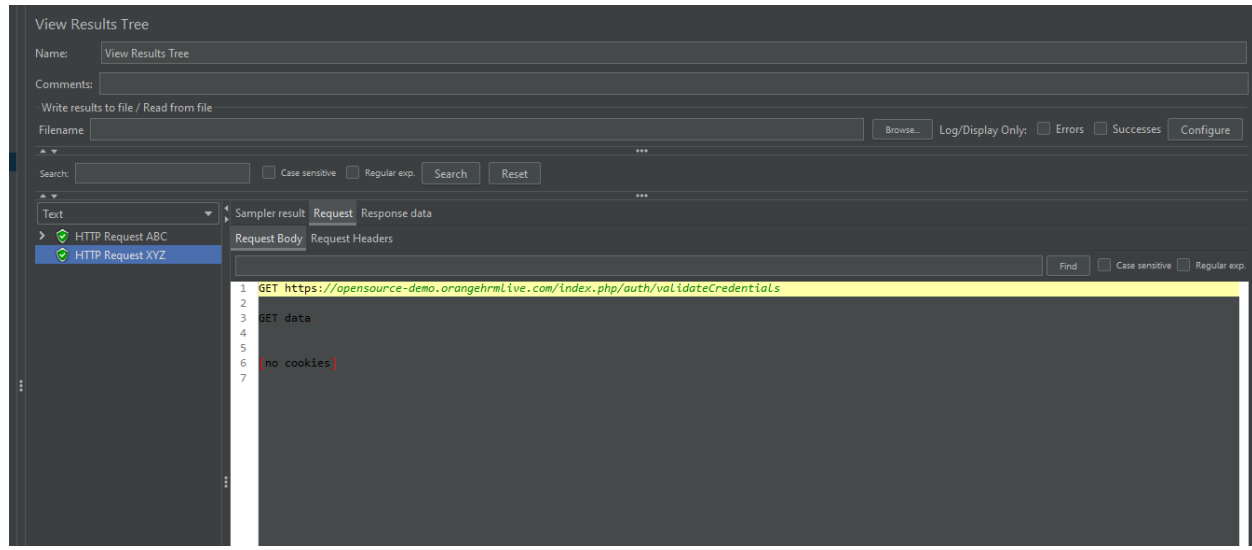
GET data

(no cookies)

Find

☐ Case sensitive

☐ Regular exp.



## Config Elements

Config Elements In Jmeter Are Used To Configure Or Modify The Sampler Requests Made To The Server. These Elements Are Added At The Same Or Higher Level Of The Samplers That We Want To Configure. **How To Add An Config Elements-**

**Right Click On Either Of Test Plan/Thread Group/Logic Controller -> Hover Over Add -> Hover Over Config Element -> Click On The Required Config Element**

- Csv Data Set Config

The Csv Data Set Config Is Used To Read Data From Csv File, Put The Data Into Variable(S) And Then Use The Variable(S) In The Sampler Requests. Http Cache Manager

The Http Cache Manager Is Used In Test Scripts To Add The Caching Functionalities Of Web Applications. This Element Is Just Required To Be Added At The Same

Level Or Higher Than The Sampler Request Where Caching Functionality Is Required.

- Http Cookie Manager

The Http Cookie Manager Is Required For Session Handling By Providing The Functionality Of Storing And Sending Of Cookies.

- User Defined Variables

As The Name Suggests, The User Defined Variable Config Element Is Used To Create Variables With A Value (Key-Value Pairs) That Are Used Across The Test Script.

- Http Authorization Manager

The Http Authorization Manager Is Used For Testing Applications Requiring Multiple Logins For Ensuring Authorization.

- Http Request Defaults

The Http Request Defaults Config Element Is Used For Setting Default Values For Http Requests.

- Http Header Manager

The Http Header Manager Is Used To Override The Http Request Headers.

## Screenshots

RECORD\_ORANGE\_HRM\_WEBSITE

- HTTP Header manager
- User Defined Variables
- HTTP Request Defaults
- DNS Cache Manager
- HTTP Authorization Manager
- HTTP Cookie Manager
- HTTP Cache Manager
- Thread Group

### HTTP Header Manager

Name: HTTP Header manager

Comments:

Headers Stored in the Header Manager

Name	Value
sec-ch-ua	"Google Chrome";v="95", "Chromium";v="95", "Not A Brand";v="99"
sec-ch-ua-mobile	0
sec-ch-ua-platform	"Windows"
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 S...
Sec-Fetch-Site	same-origin

### User Defined Variables

Name: User Defined Variables

Comments:

User Defined Variables

Name	Value	Description
BASE_URL_1	opensource-demo.orangehrmlive.com	

### HTTP Request Defaults

Name: HTTP Request Defaults

Comments:

Basic Advanced

Web Server

Protocol (http): Server Name or IP: Port Number:

HTTP Request

Path: Content encoding:

Parameters Body Data

Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals?
------	-------	-------------	--------------	-----------------

DNS Cache Manager

Name:

DNS Cache Manager

Comments:

Options

☒ Clear cache each iteration

☒ Use system DNS resolver ☐ Use custom DNS resolver

DNS Servers

Hostname or IP address

Add

Delete

Static Host Table

Host

Hostname or IP address

Add static host

Delete static host

### HTTP Authorization Manager

Name:

HTTP Authorization Manager

Comments:

Options

☐ Clear auth on each iteration?

☐ Use Thread Group configuration to control clearing

Authorizations Stored in the Authorization Manager

Base URL	Username:	Password:	Domain	Realm	Mechanism

HTTP Cookie Manager

Name:

HTTP Cookie Manager

Comments:

Options

☒ Clear cookies each iteration?

☐ Use Thread Group configuration to control cookie clearing

standard

User-Defined Cookies

Name	Value	Domain	Path	Secure
------	-------	--------	------	--------

## Http Cache Manager

HTTP Cache Manager

Name:

Comments:

☒ Clear cache each iteration?

☐ Use Thread Group configuration to control cache clearing

☐ Use Cache-Control/Expires header when processing GET requests

Max Number of elements in cache:

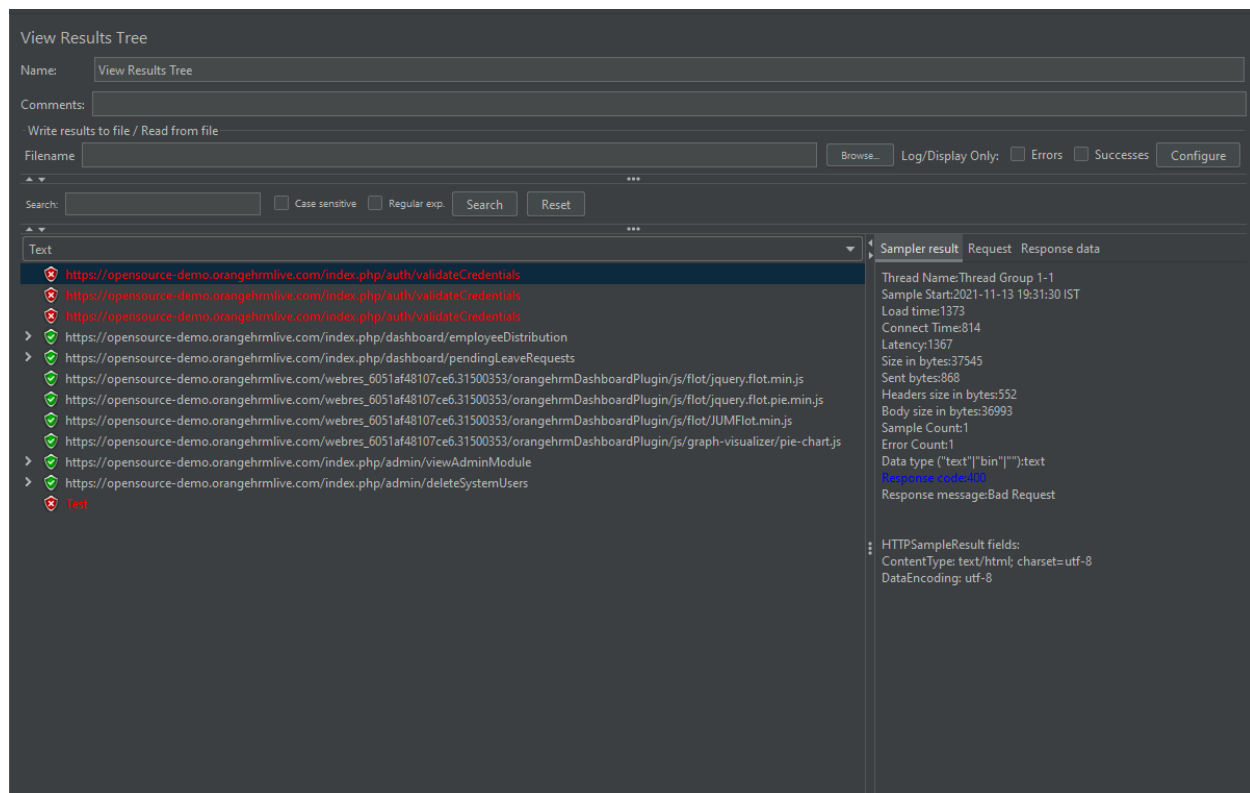
CSV Data Set Config

Name:

Comments:

Configure the CSV Data Source

Filename:	<input type="text" value="C:/Users/djal/OneDrive/Desktop/JMETER TESTING/testrepo.csv"/>	<input type="button" value="Browse..."/>
File encoding:	<input type="text"/>	
Variable Names (comma-delimited):	<input type="text"/>	
Ignore first line (only used if Variable Names is not empty):	<input type="text" value="False"/>	
Delimiter (use '\t' for tab):	<input type="text" value=","/>	
Allow quoted data?:	<input type="text" value="False"/>	
Recycle on EOF?:	<input type="text" value="True"/>	
Stop thread on EOF?:	<input type="text" value="False"/>	
Sharing mode:	<input type="text" value="All threads"/>	



## Timers

Timers In Jmeter Is The Test Plan Elements Used To Pause The Execution Of Test For A Certain Specified Amount Of Time. This Pause Between Requests Helps In Simulating Real-World Scenarios Like Time Taken By Users To Think, Type Something, See And Process The Information Displayed Etc. **How To Add A Timer- Right Click On Thread Group -> Hover Over Add -> Hover Over Timer -> Click On The Required Timer**

- **Constant Timer**

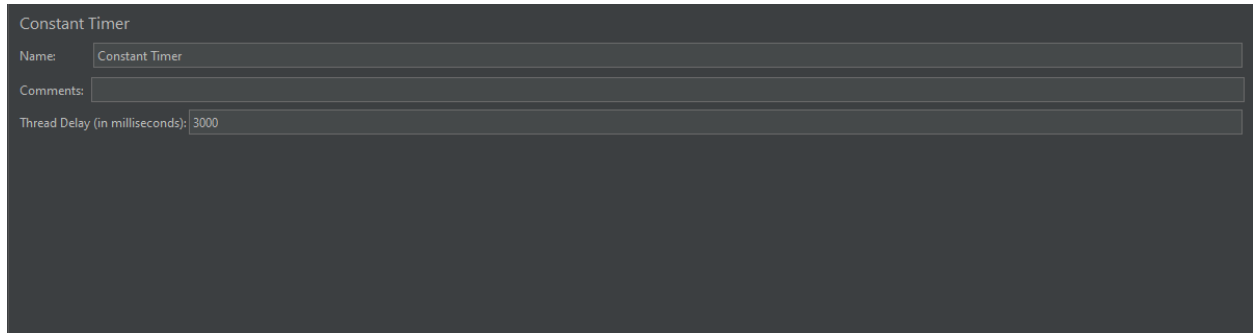
The Constant Timer Is One Of The Most Widely Used Timers In Jmeter. It Pauses The Execution Of Test For A Specified Constant Amount Of Time.

- **Uniform Random Timer**



The Uniform Random Timer Is Used To Pause The Test Execution For A Random Time. The Maximum Value For Random Time Can Be Specified Along With The Additional Constant Time With Each Wait.

## Screenshots



Constant Timer

Name:

Comments:

Thread Delay (in milliseconds):



Uniform Random Timer

Name:

Comments:

Thread Delay Properties

Random Delay Maximum (in milliseconds):

Constant Delay Offset (in milliseconds):

Test Plan

- Thread Group
  - HTTP Request 1
  - Constant Timer
  - HTTP Request 2
  - View Results in Table**
  - View Results Tree
  - Uniform Random Timer

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:   Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	19:36:55.906	Thread Group 1-1	HTTP Request 1	1607	OK	105541	248	1046	1002
2	19:36:56.180	Thread Group 1-1	HTTP Request 2	439	OK	105643	248	40	0

☐ Scroll automatically? ☐ Child samples?

No of Samples: 2 Latest Sample: 439 Average: 1001 Deviation: 584

## Listeners

Jmeter Listeners Are The Test Plan Elements That Are Used To View And Analyze The Result Of Performance Tests In Tabular Or Graphical Form. They Also Provide The Different Response Time Matrices (Average Time, Minimum Time, Max Time, Etc) Of A Sampler Request. **How To Add A Listener- Right Click On Test Plan -> Hover Over Add -> Hover Over Listener -> Click On The Required Listener**

- **Aggregate Graph**

The Aggregate Graph Listener Is Used To Display The Test Results In Both Tabular Form(Reports) And Graphs.

- **Aggregate Report**

The Aggregate Report Listener Is Used To Display And Store Test Results In The Form Of Reports.

- **Assertion Results**

The Assertion Results Listener Is Used To Display The Assertion Result For Each Erroneous Sampler Response. It Is Advised To Not Use This Listener During The Performance Test As It Is Very Resource-Intensive. It Should Be Used While Debugging And Functional Testing Only.

- **Graph Results**

The Graph Results Listener Is Used To Display Each Sampler Request's Response Time Graph In Terms Of Average, Median, Deviation, And Throughput.

- **Response Time Graph**

The Response Time Graph Is Used To Provide The Graphical Representation Of Response Time With Time Elapsed During The Test Run.

- **Simple Data Writer**

The Simple Data Writer Listener Is Used To Save The Sampler Response To A File After With Different Configurations To Remove Several Unnecessary Overheads.

- **Summary Report**

The Summary Report Is Used To Store And Display The Test Result In Tabular Form Just Like An Aggregate Report Listener But Consumes Less Memory (As Per Apache Jmeter).

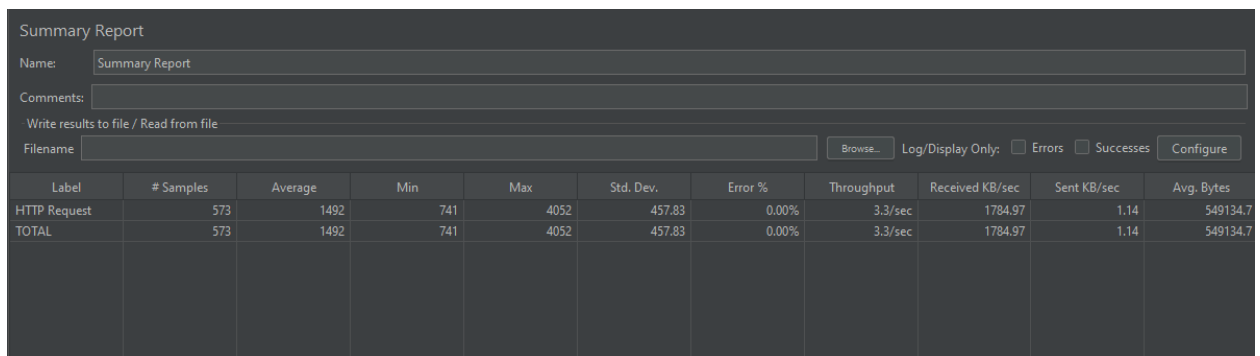
- **View Results Tree**

This Listener Is Used To Provide And Store Test Results For Each And Every Individual Sampler.

- **View Results In Table**

The View Results In A Table Listener Are Used To Display The Sampler Response Header And Response Body.

## Screenshots



The screenshot shows the 'Summary Report' configuration window. It includes fields for 'Name' (Summary Report), 'Comments', and 'Filename'. There are checkboxes for 'Log/Display Only', 'Errors', and 'Successes', and a 'Configure' button. Below the configuration is a table displaying test results.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	573	1492	741	4052	457.83	0.00%	3.3/sec	1784.97	1.14	549134.7
TOTAL	573	1492	741	4052	457.83	0.00%	3.3/sec	1784.97	1.14	549134.7

### Aggregate Graph

Name:

Comments:

Write results to file / Read from file

Filename   Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s...	Sent KB/sec
HTTP Request	652	1494	1437	2028	2248	3095	741	4052	0.00%	3.3/sec	1790.13	1.15
TOTAL	652	1494	1437	2028	2248	3095	741	4052	0.00%	3.3/sec	1790.13	1.15

Settings: **Graph**

☐ Save Table Header

Column settings

Columns to display: ☒ Average ☐ Median ☐ 90% Line ☐ 95% Line ☐ 99% Line ☐ Min ☐ Max

Value font:  Size:  Style:  ☒ Draw outlines bar? ☒ Show number grouping? ☒ Value labels vertical?

☐ Column label selection:   ☐ Case sensitive ☒ Regular exp.

Title

Graph title:

Font:  Size:  Style:

Graph size

☒ Dynamic graph size Width:  Height:

X Axis Y Axis (milli-seconds)

Max length of x-axis label:  Scale maximum value:

Legend

Placement:  Font:  Size:  Style:





### Aggregate Report

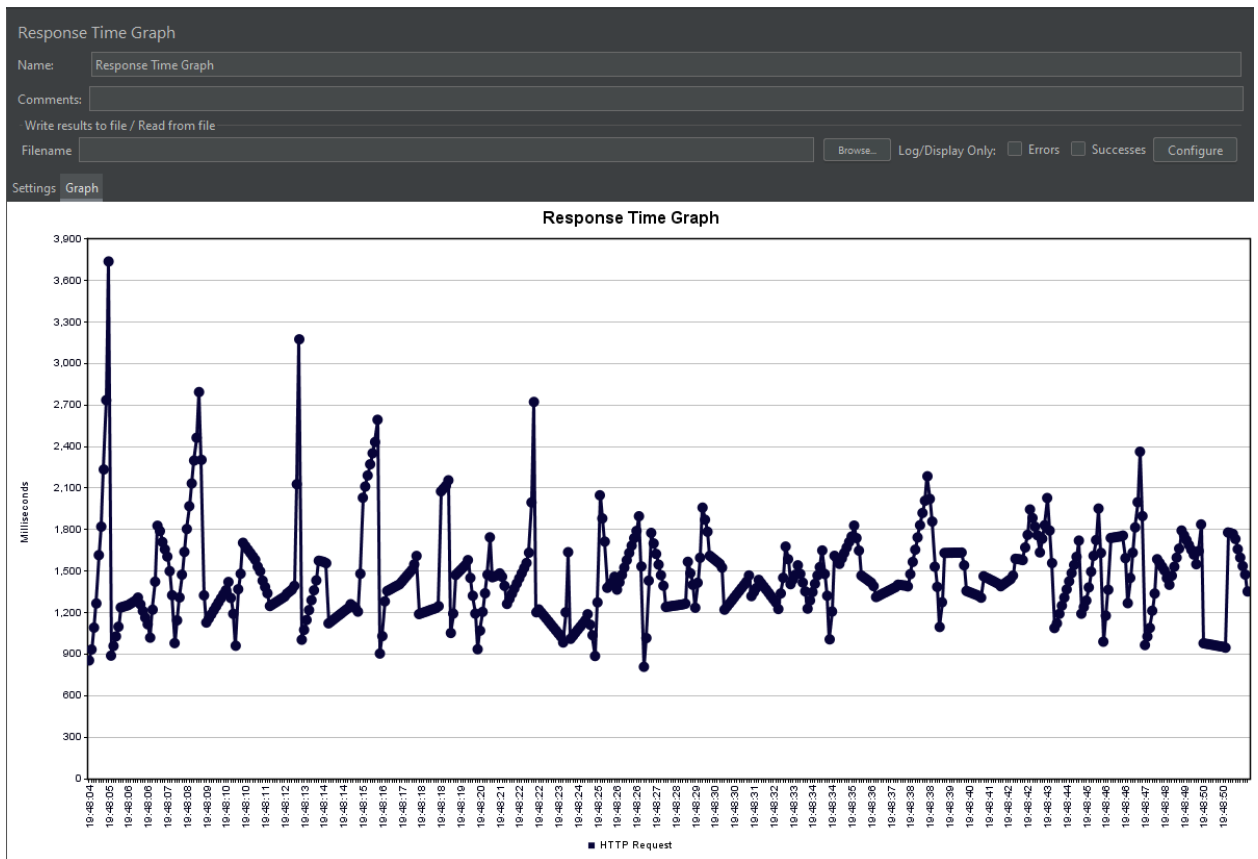
Name:

Comments:

Write results to file / Read from file

Filename   Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s...	Sent KB/sec
HTTP Request	709	1486	1430	2028	2248	3095	308	4052	0.71%	3.4/sec	1789.93	1.15
TOTAL	709	1486	1430	2028	2248	3095	308	4052	0.71%	3.4/sec	1789.93	1.15



### Simple Data Writer

Name:

Simple Data Writer

Comments:

Write results to file / Read from file

Filename

C:\Users\diya\OneDrive\Desktop\maindata.csv

Browse...

Log/Display Only: ☐ Errors ☐ Successes

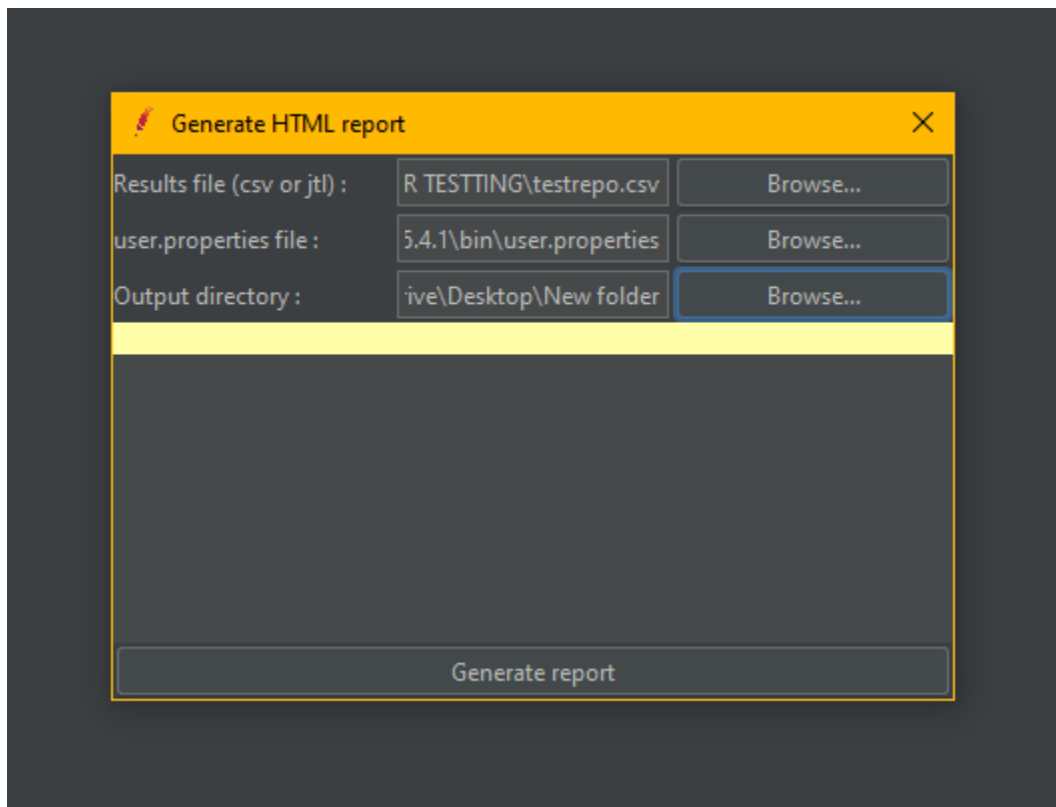
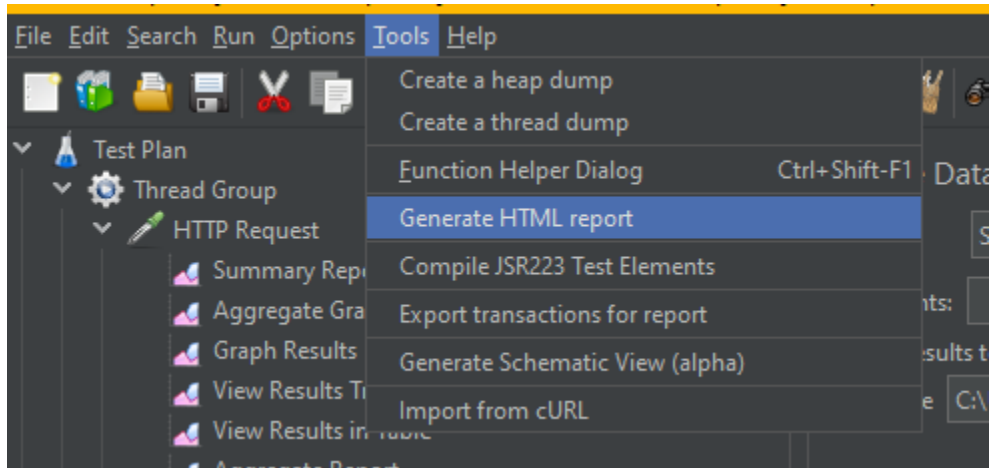
Configure

## Csv File Create Via Simple Data Writer

[illegible]



## Creating Html Reports



Generate HTML report

Results file (csv or jtl) :

e:\Desktop\maindata.csv

Browse...

user.properties file :

5.4.1\bin\user.properties

Browse...

Output directory :

.Desktop\New Folder (3)

Browse...

Generating report

Report created !

Generate report

Name	Date modified	Type	Size
content	11/13/2021 8:03 PM	File folder	
sbadmin2-1.0.7	11/13/2021 8:03 PM	File folder	
index.html	11/13/2021 8:03 PM	Chrome HTML Do...	10 KB
statistics.json	11/13/2021 8:03 PM	JSON File	3 KB

Dashboard

Charts

Customs Graphs

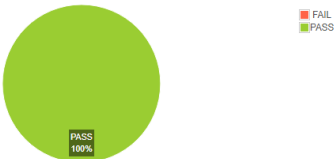
Test and Report information

Source file	"maindata.csv"
Start Time	"11/13/21 7:55 PM"
End Time	"11/13/21 7:55 PM"
Filter for display	all

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.750	500 ms	1 sec 500 ms	Total
0.500	500 ms	1 sec 500 ms	HTTP Request-2
0.500	500 ms	1 sec 500 ms	HTTP Request
1.000	500 ms	1 sec 500 ms	HTTP Request-1
1.000	500 ms	1 sec 500 ms	HTTP Request-0

Requests Summary



Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)		
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent	