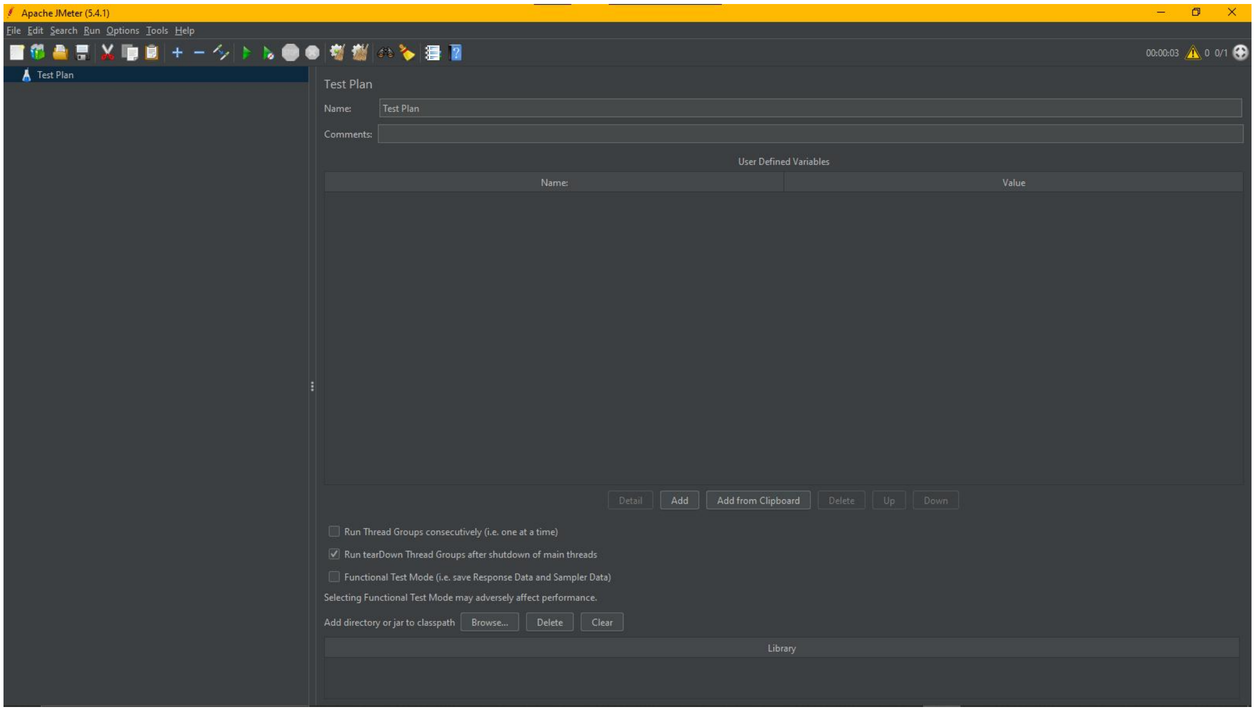
SBA7-JMETER

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.



Test Plan

Name: Test Plan

Comments:

User Defined Variables

Name:	Value

□ Run Thread Groups consecutively (i.e. one at a time)

☑ Run tearDown Thread Groups after shutdown of main threads

□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

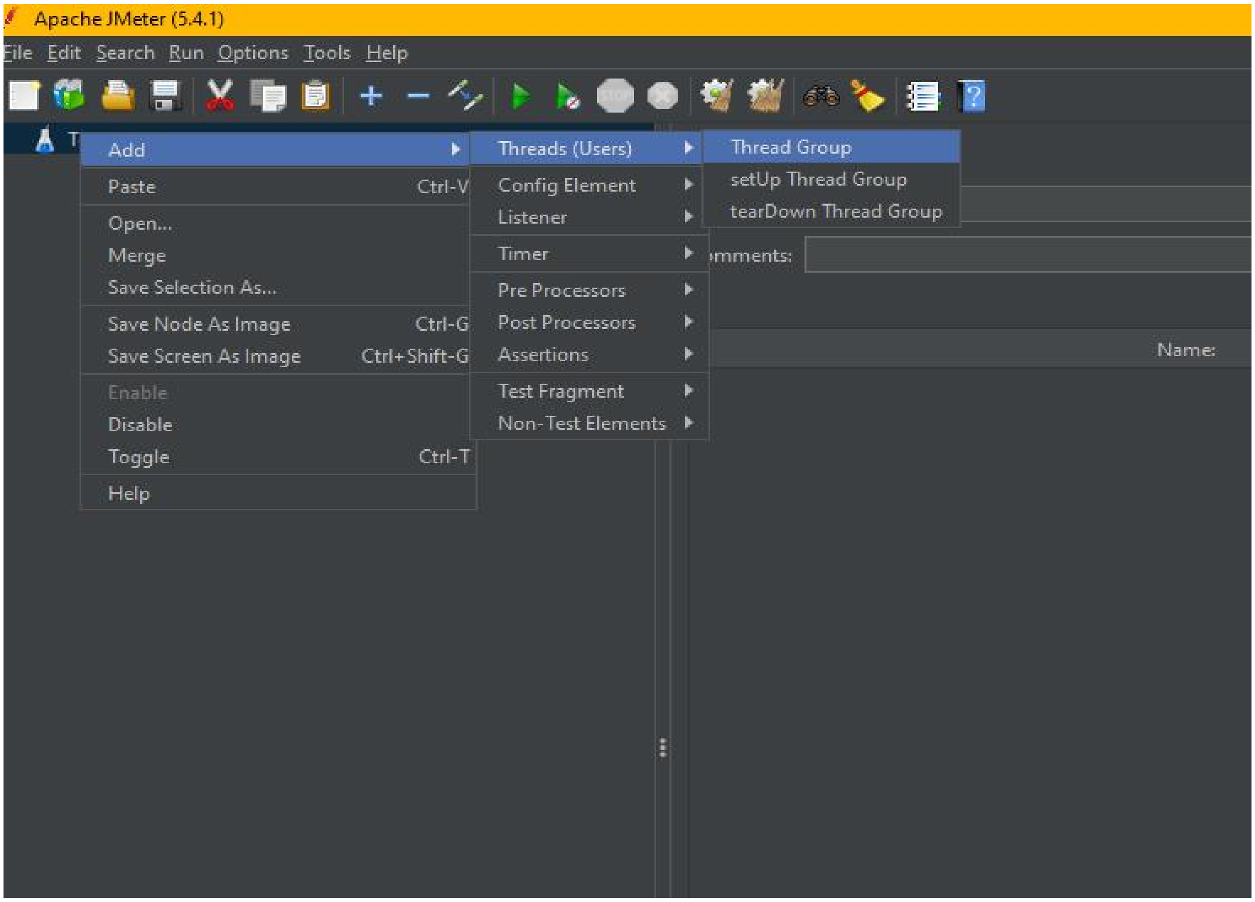
Library

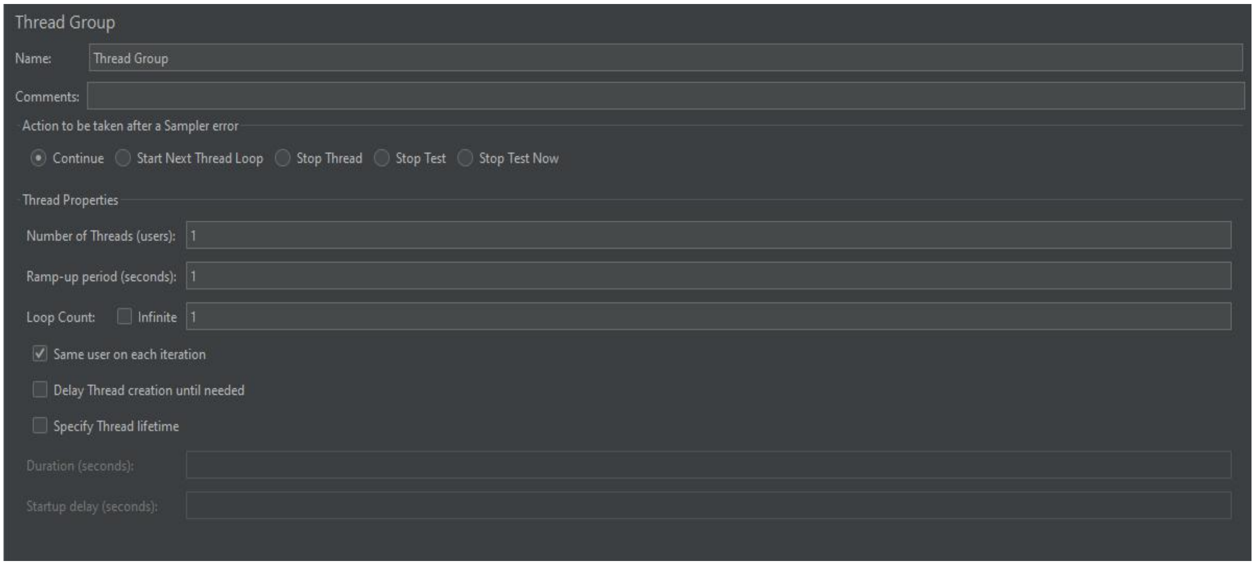
Elements In A Test Plan

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: Thread Group

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): 1

Ramp-up period (seconds): 1

Loop Count: □ Infinite 1

☑ Same user on each iteration

□ Delay Thread creation until needed

□ Specify Thread lifetime

Duration (seconds):

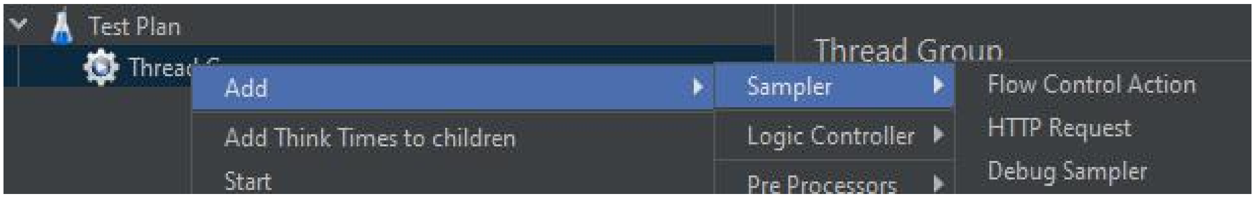
Startup delay (seconds):

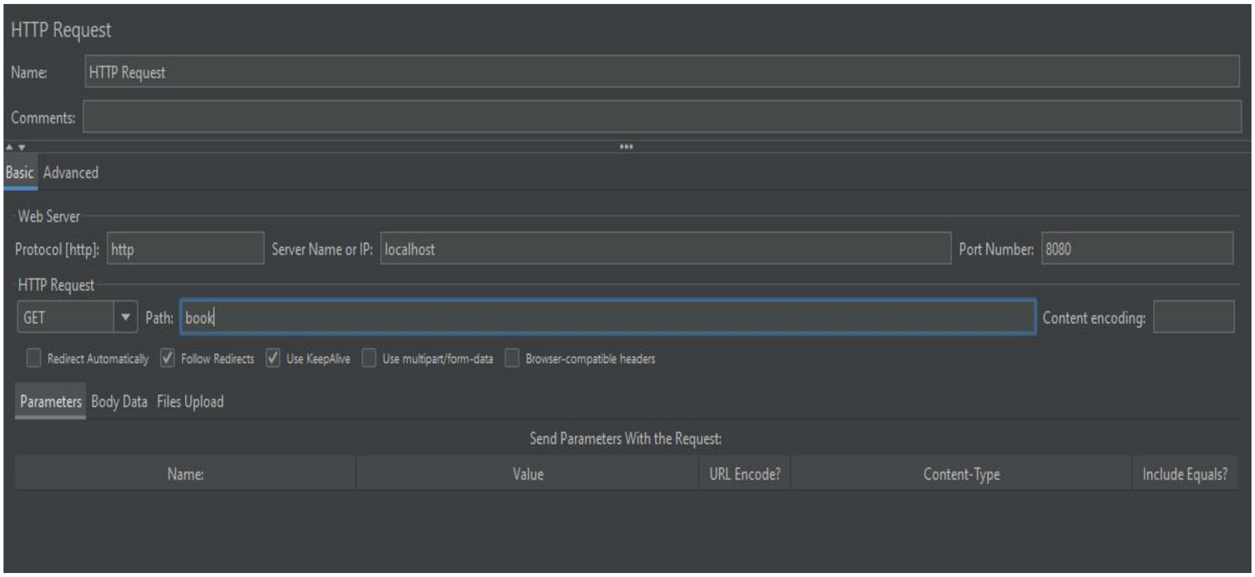
Sampler

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.





HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 8080

HTTP Request

GET Path: book Content encoding:

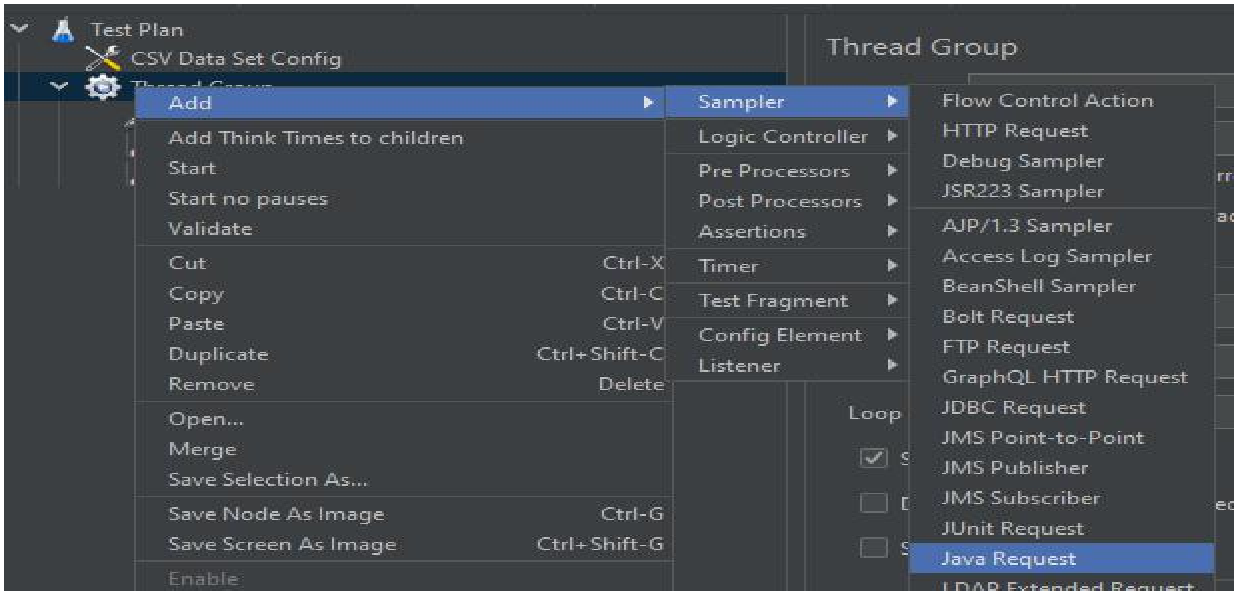
□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

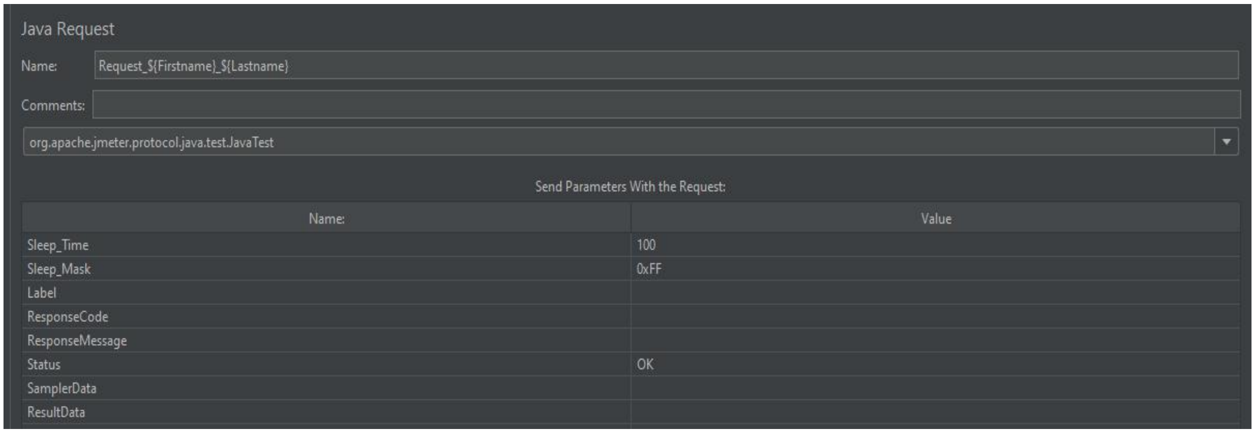
Parameters Body Data Files Upload

Send Parameters With the Request:

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

- **Java Request**





Java Request

Name: Request_${Firstname}_${Lastname}

Comments:

org.apache.jmeter.protocol.java.test.JavaTest

Send Parameters With the Request:

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

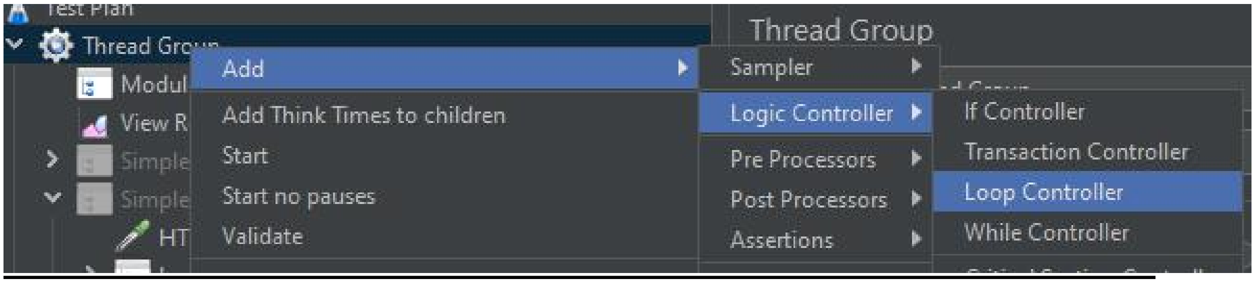
Logic Controller

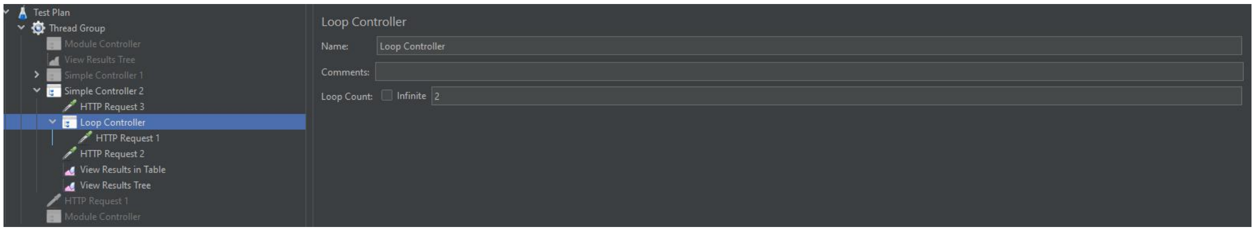
Logic Controllers Are The Test Plan Elements That Are Used To Customize The Order Of Processing Of Samplers And Other Elements Added As Child. Primarily, Logic Controllers Are Used With Sampler Requests To Perform Various Customization Like – Altering Their Order Of Processing, Grouping Them As A Single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



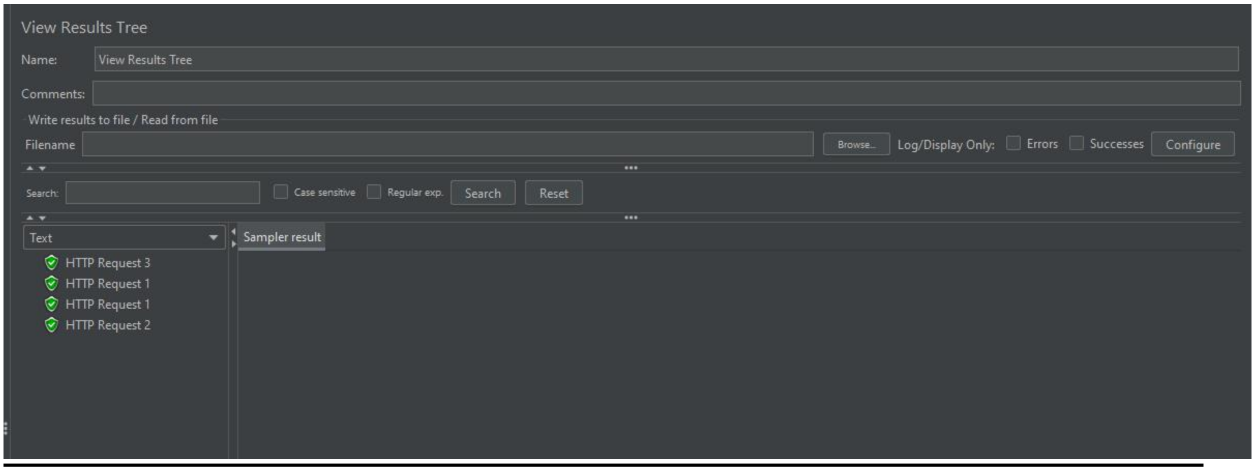


Loop Controller

Name: Loop Controller

Comments:

Loop Count: □ Infinite 2



View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

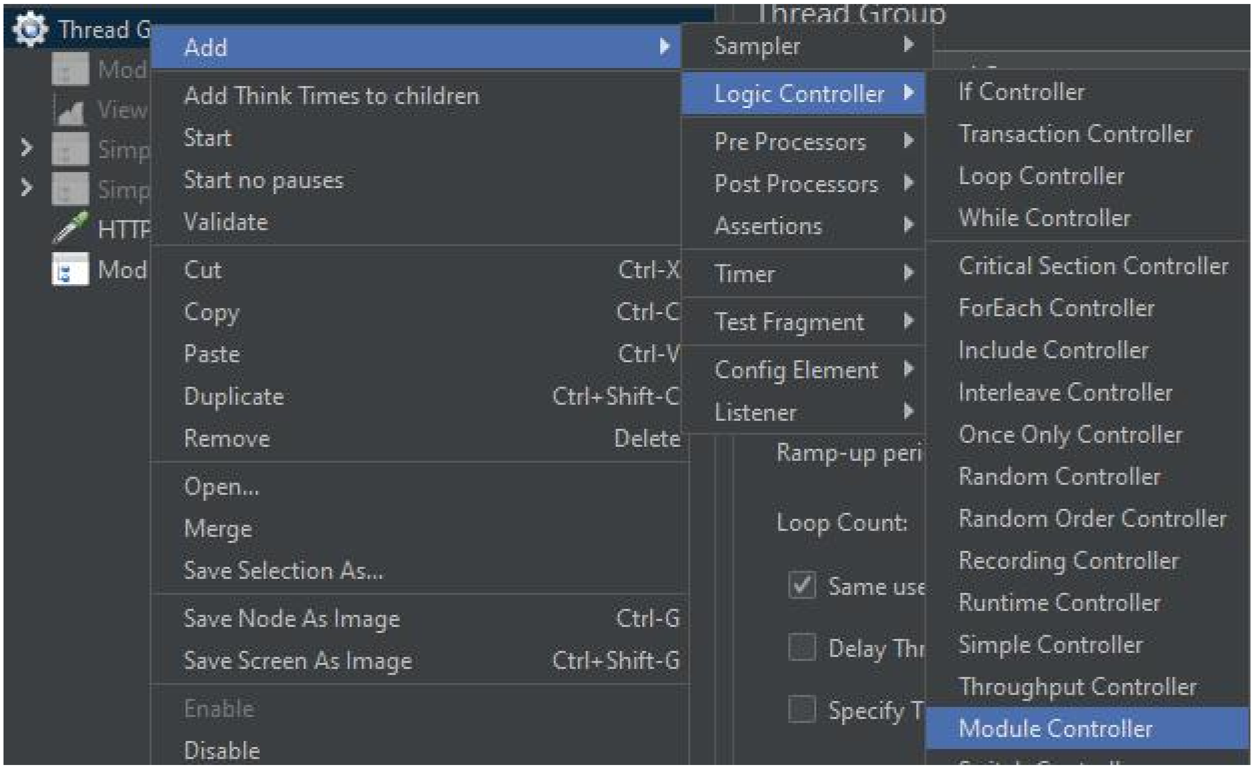
Log/Display Only: □ Errors □ Successes

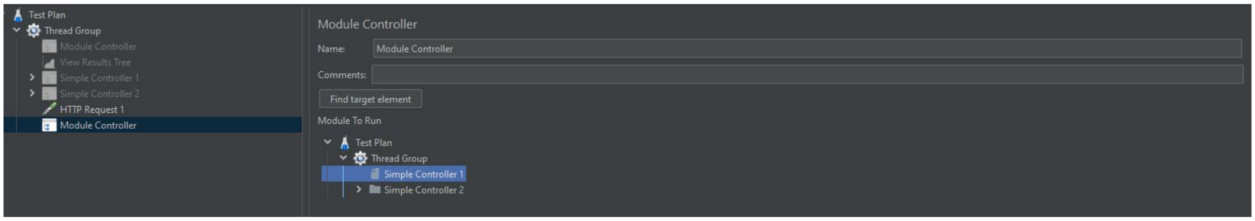
Search: □ Case sensitive □ Regular exp.

HTTP Request 3
HTTP Request 1
HTTP Request 1
HTTP Request 2

- 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.

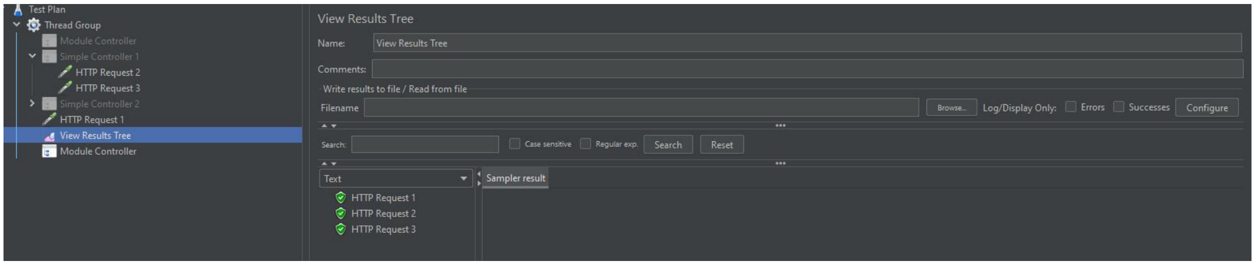




Module Controller

Name: Module Controller

Comments:



View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

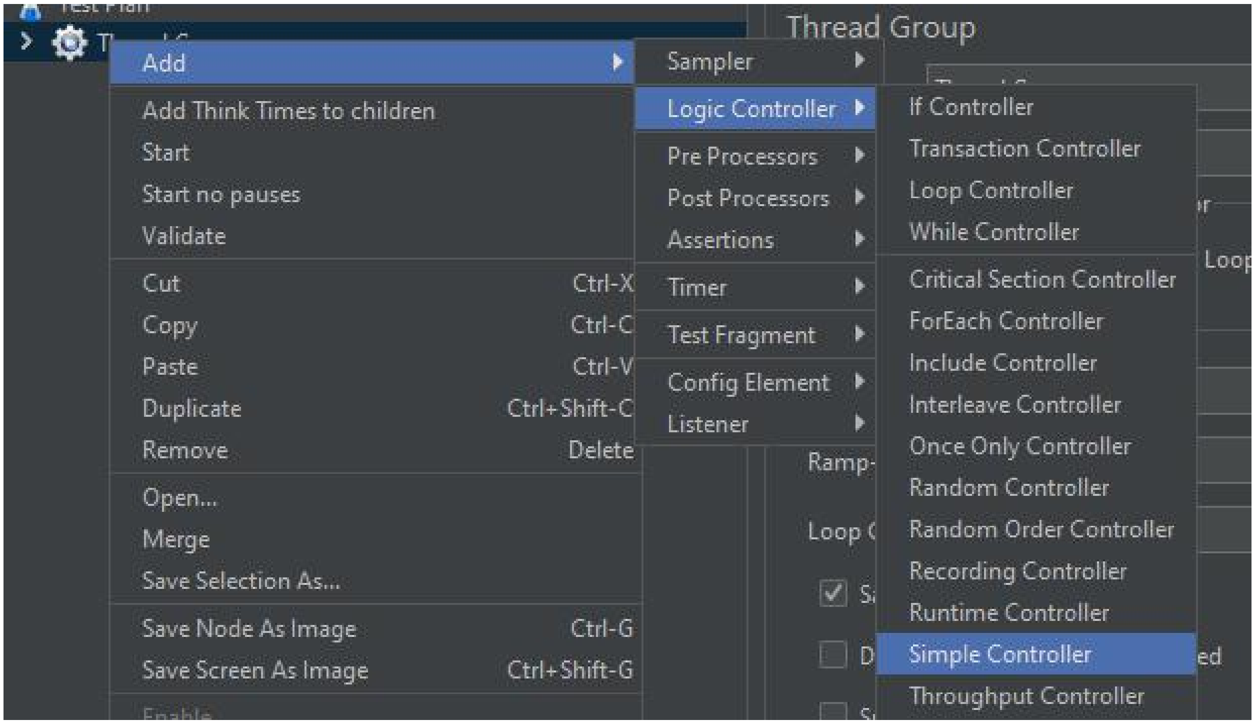
Search: □ Case sensitive □ Regular exp.

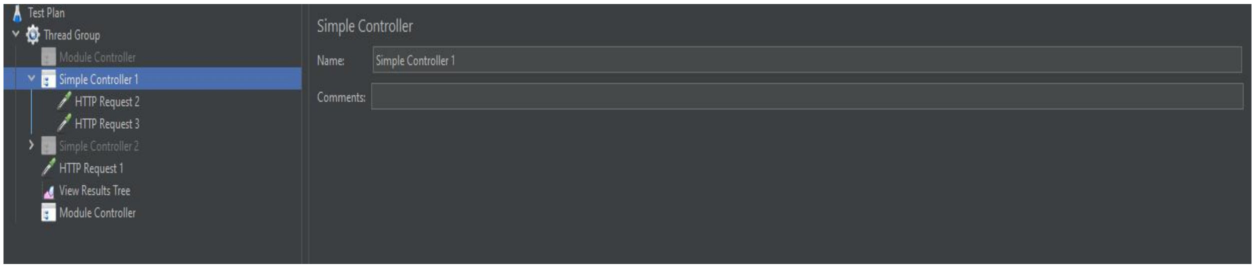
Text

HTTP Request 1
HTTP Request 2
HTTP Request 3

- Simple Controller

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





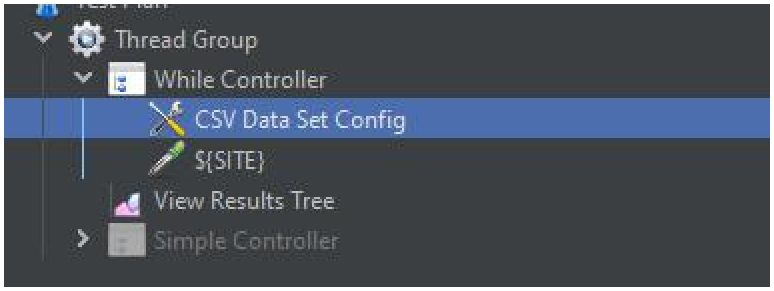
Simple Controller

Name: Simple Controller 1

Comments:

- While Controller

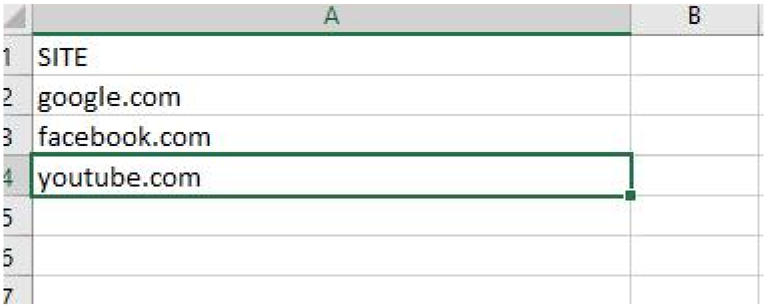
  The While Controller Is Used To Run The Child Elements Inside It Till The Value Specified In Its Control Panel Is Evaluated To False. Steps To Launch A Logic Controller- **Right Click On Thread Group -> Hover Over Add -> Hover Over Logic Controllers -> Click On The While Controller.**



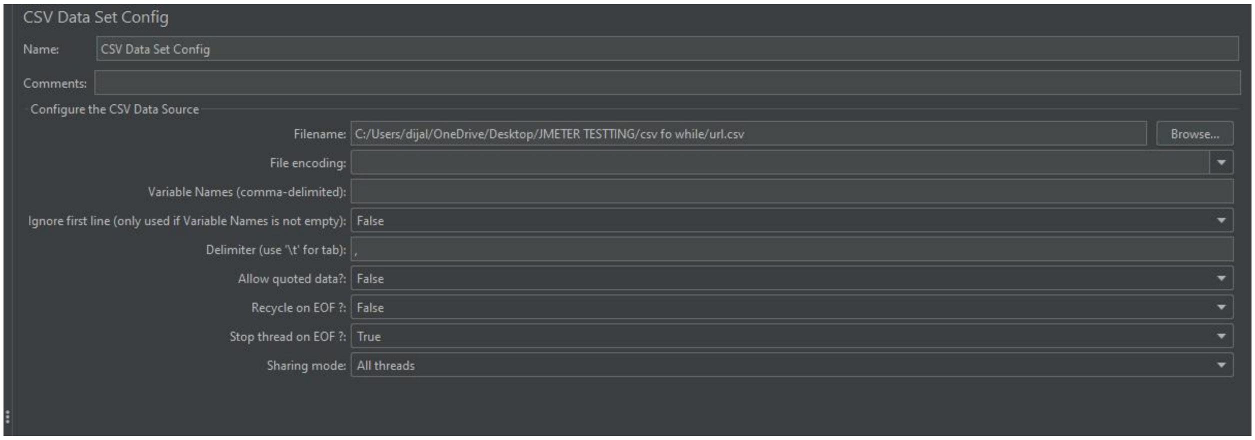
Thread Group
While Controller
CSV Data Set Config
${SITE}
View Results Tree
Simple Controller

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,



	A	B
1	SITE	
2	google.com	
3	facebook.com	
4	youtube.com	
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 TESTTING/csv fo while/url.csv

File encoding:

Variable Names (comma-delimited):

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

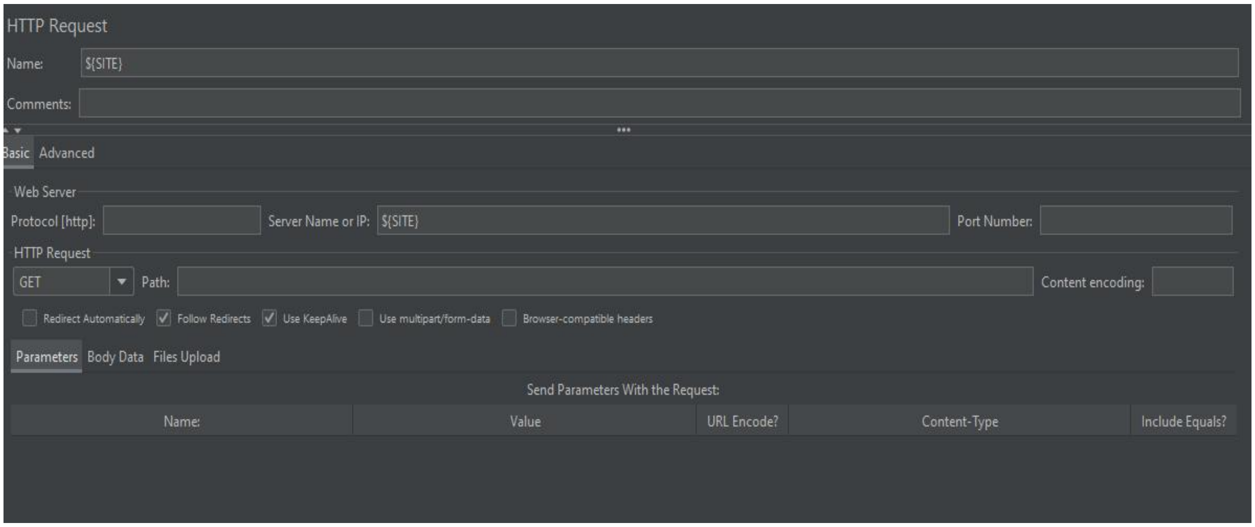
Delimiter (use '\t' for tab): ,

Allow quoted data?: False

Recycle on EOF ?: False

Stop thread on EOF ?: True

Sharing mode: All threads



Name: ${SITE}

Comments:

Protocol [http]:

Server Name or IP: ${SITE}

Port Number:

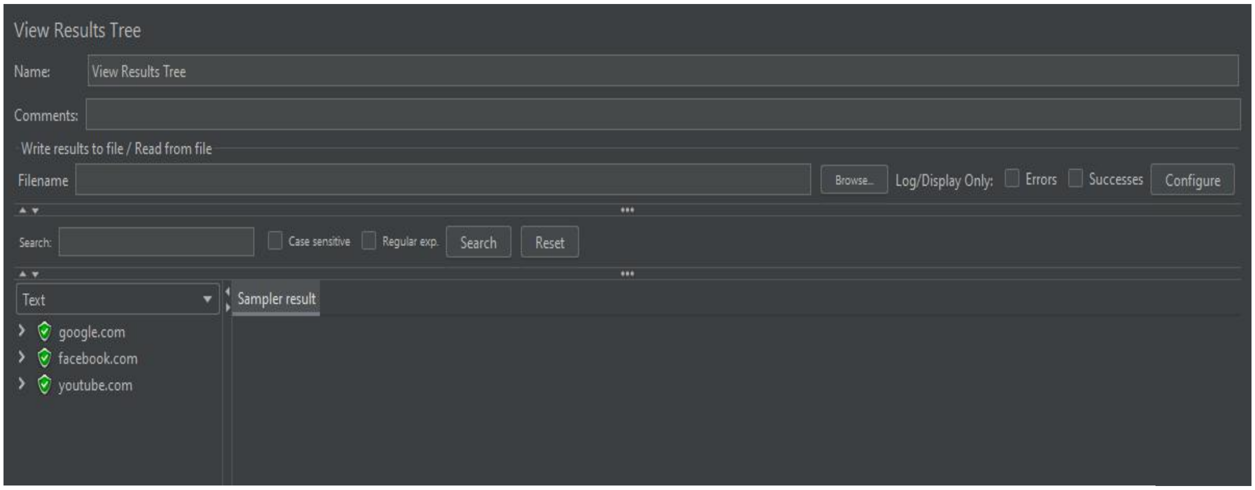
GET

Path:

Content encoding:

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

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



Name: View Results Tree

Comments:

Filename

Log/Display Only: ☐ Errors ☐ Successes

Search:

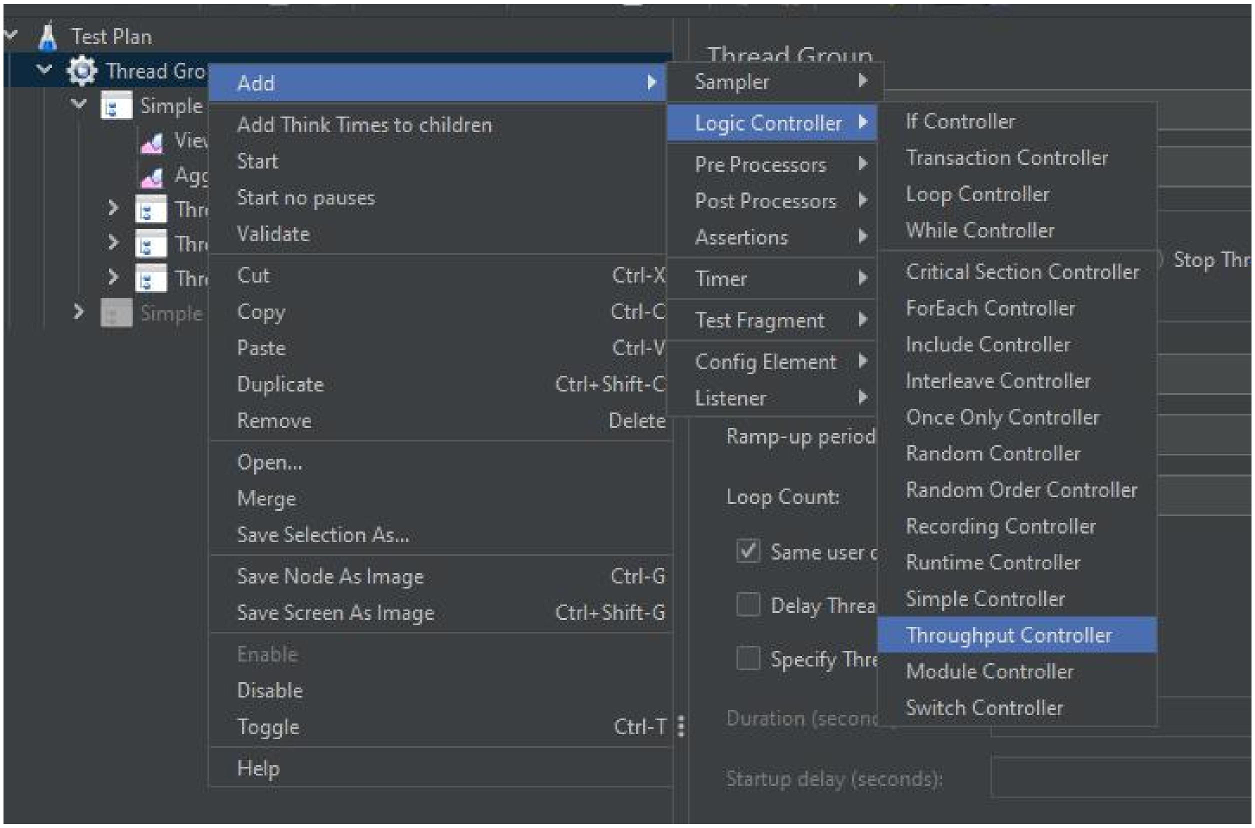
☐ Case sensitive ☐ Regular exp.

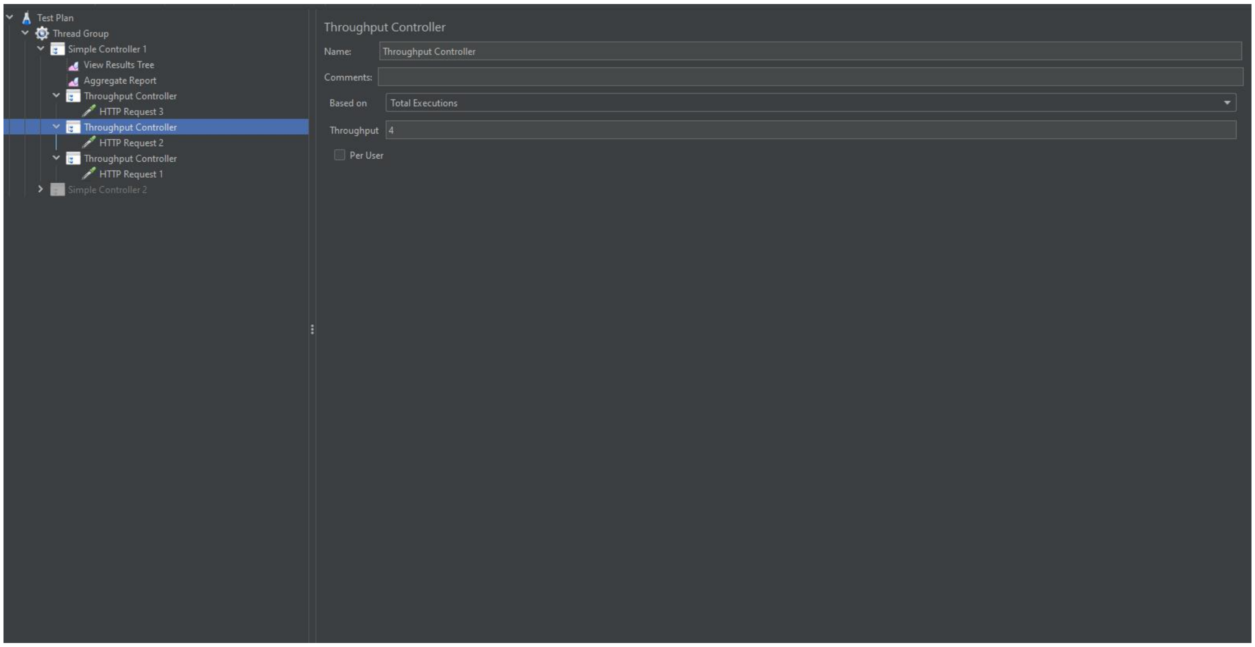
Text

google.com
facebook.com
youtube.com

- Throughput Controller

The Throughput Controller Is Used To Control The Processing Of Its Child Elements In Terms Of The Total Number Of Executions Or The Percentage Of Execution Specified In Its Control Panel.





Throughput Controller

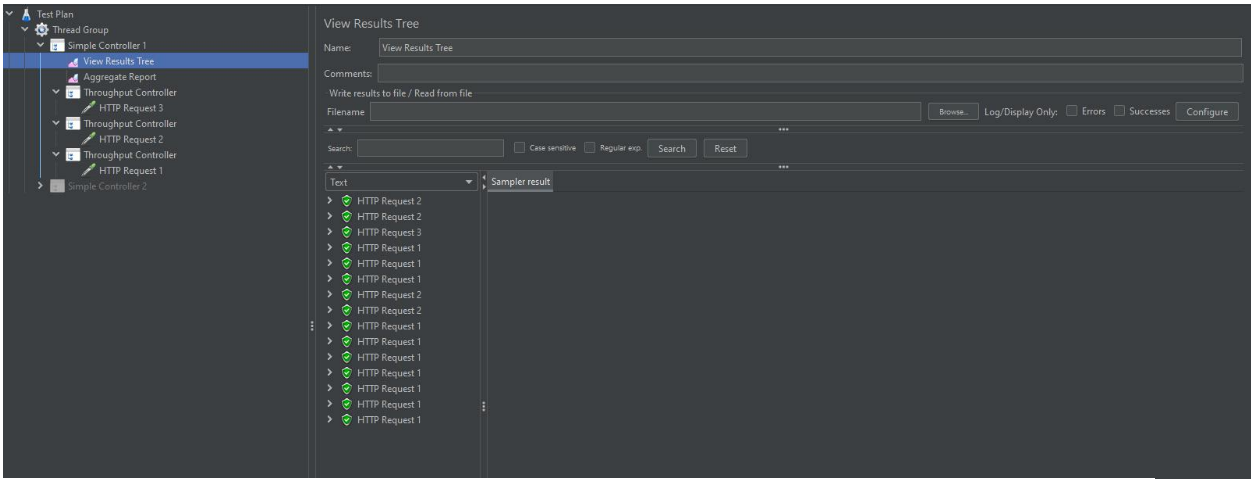
Name: Throughput Controller

Comments:

Based on Total Executions

Throughput 4

□ Per User



```
Test Plan
  Thread Group
    Simple Controller 1
      View Results Tree
      Aggregate Report
      Throughput Controller
        HTTP Request 3
      Throughput Controller
        HTTP Request 2
      Throughput Controller
        HTTP Request 1
    Simple Controller 2
```

View Results Tree

Name: View Results Tree

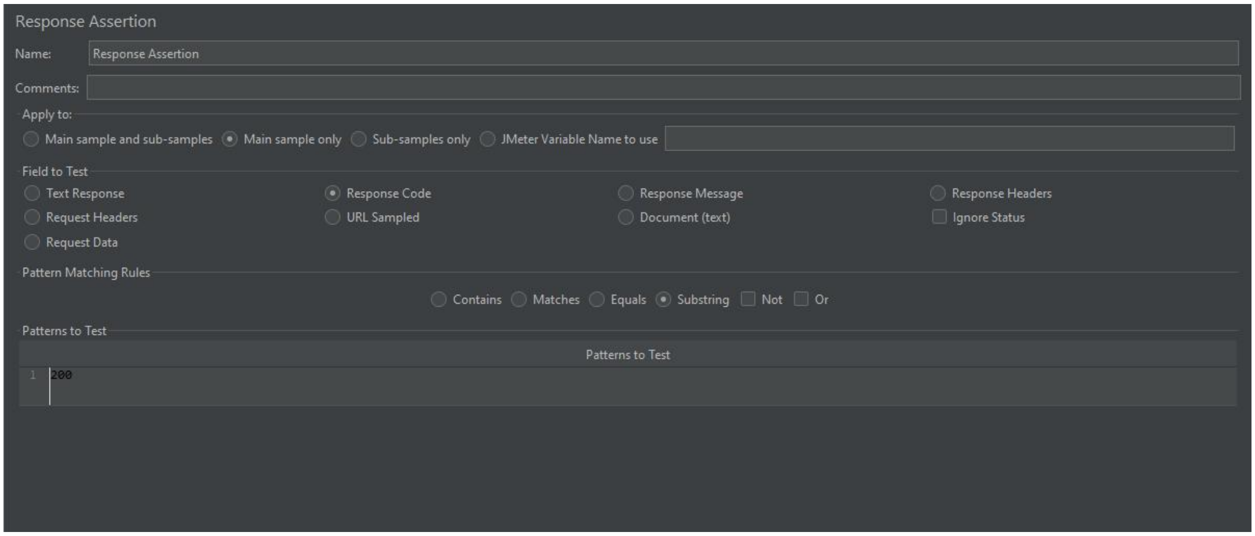
```
HTTP Request 2
HTTP Request 2
HTTP Request 3
HTTP Request 1
HTTP Request 1
HTTP Request 1
HTTP Request 2
HTTP Request 2
HTTP Request 1
HTTP Request 1
HTTP Request 1
HTTP Request 1
HTTP Request 1
HTTP Request 1
HTTP Request 1
```

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



Response Assertion

Name: Response Assertion

Comments:

Apply to:

○ Main sample and sub-samples ● Main sample only ○ Sub-samples only ○ JMeter Variable Name to use

Field to Test

○ Text Response ● Response Code ○ Response Message ○ Response Headers

○ Request Headers ○ URL Sampled ○ Document (text) □ Ignore Status

○ Request Data

Pattern Matching Rules

○ Contains ○ Matches ○ Equals ● Substring □ Not □ Or

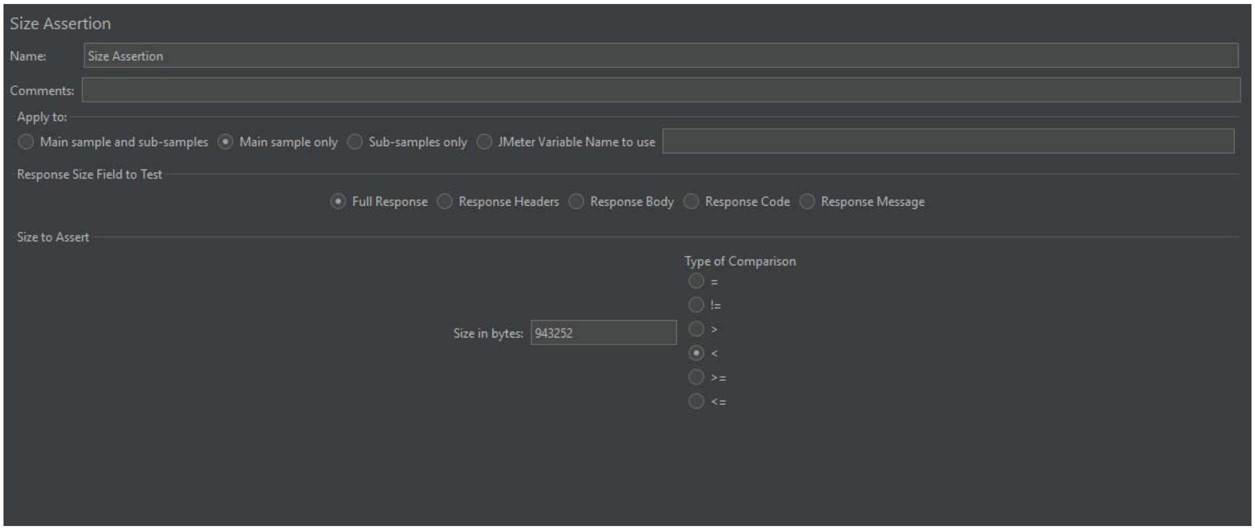
Patterns to Test

Patterns to Test

```
200
```

- **Size Assertion**

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



Size Assertion

Name: Size Assertion

Comments:

Apply to:

○ Main sample and sub-samples ● Main sample only ○ Sub-samples only ○ JMeter Variable Name to use

Response Size Field to Test

● Full Response ○ Response Headers ○ Response Body ○ Response Code ○ Response Message

Size to Assert

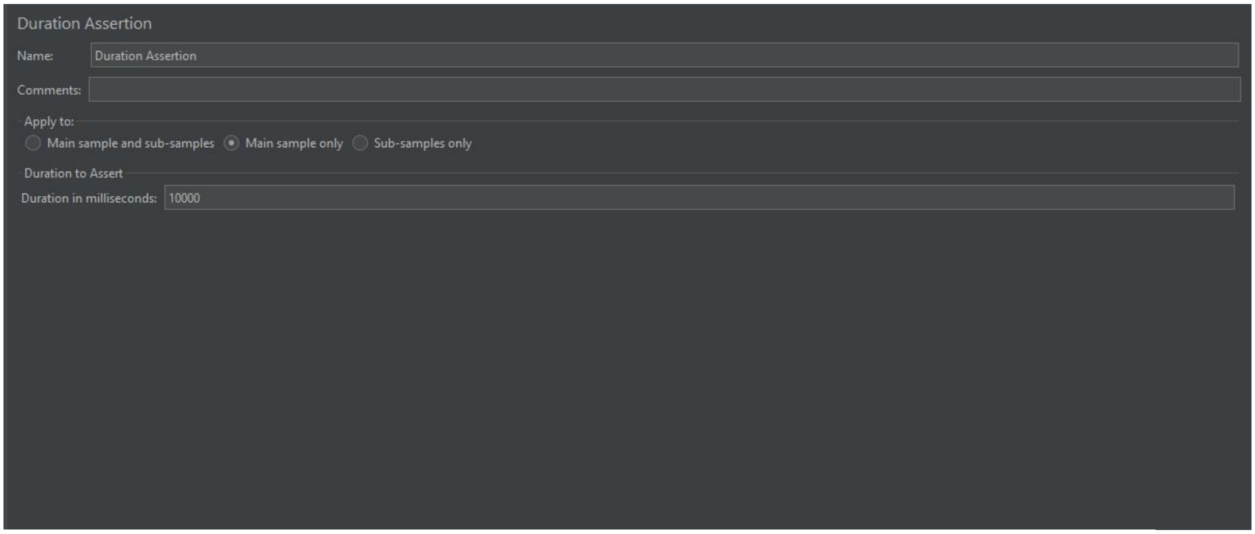
Size in bytes: 943252

Type of Comparison

○ =
○ !=
○ >
● <
○ >=
○ <=

- **Duration Assertion**

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



Duration Assertion

Name: Duration Assertion

Comments:

Apply to:

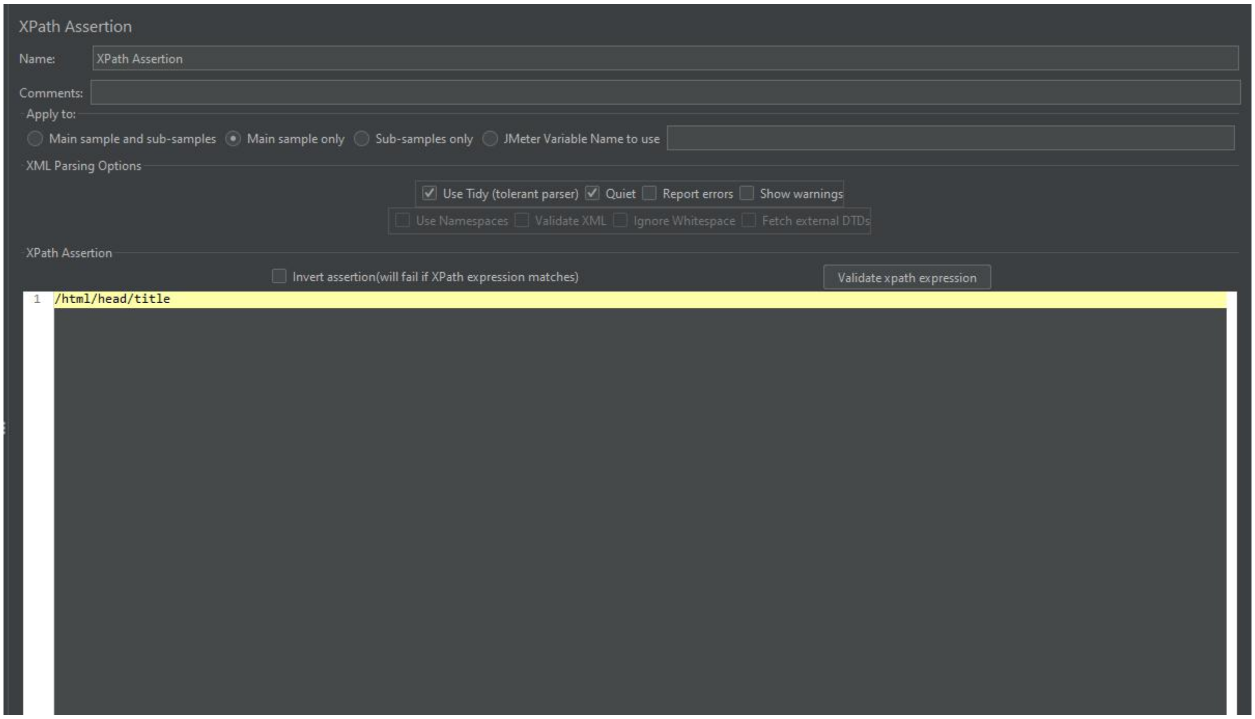
○ Main sample and sub-samples ● Main sample only ○ Sub-samples only

Duration to Assert

Duration in milliseconds: 10000

- **Xpath Assertion**

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



XPath Assertion

Name: XPath Assertion

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

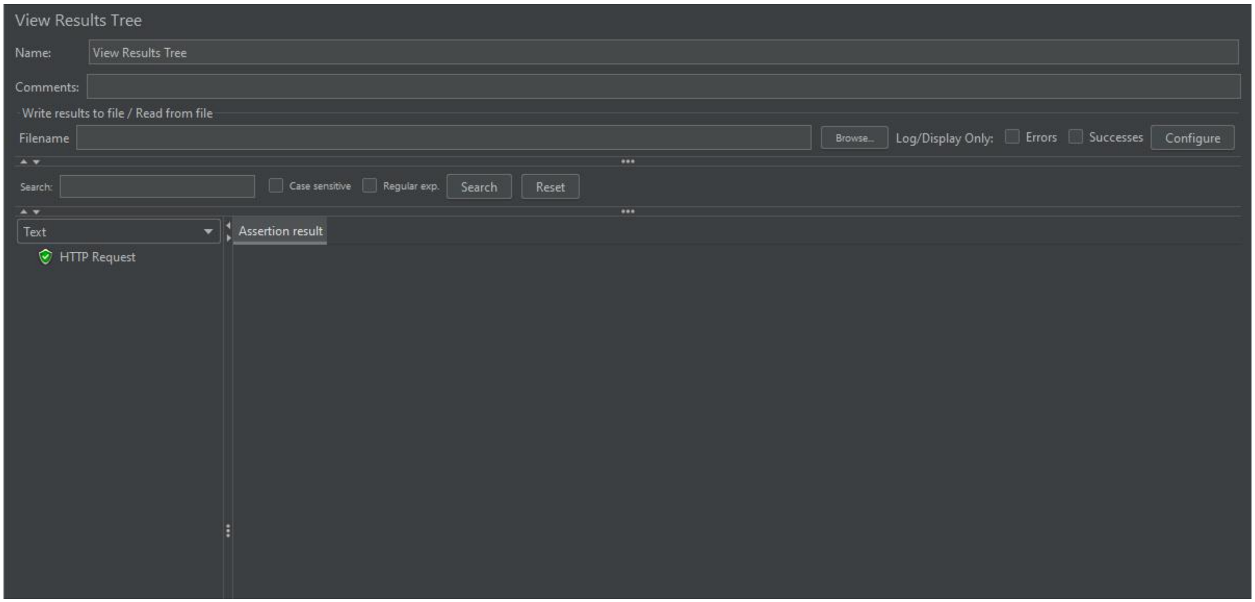
□ Use Namespaces □ Validate XML □ Ignore Whitespace □ Fetch external DTDs

XPath Assertion

□ Invert assertion(will fail if XPath expression matches)

```
/html/head/title
```

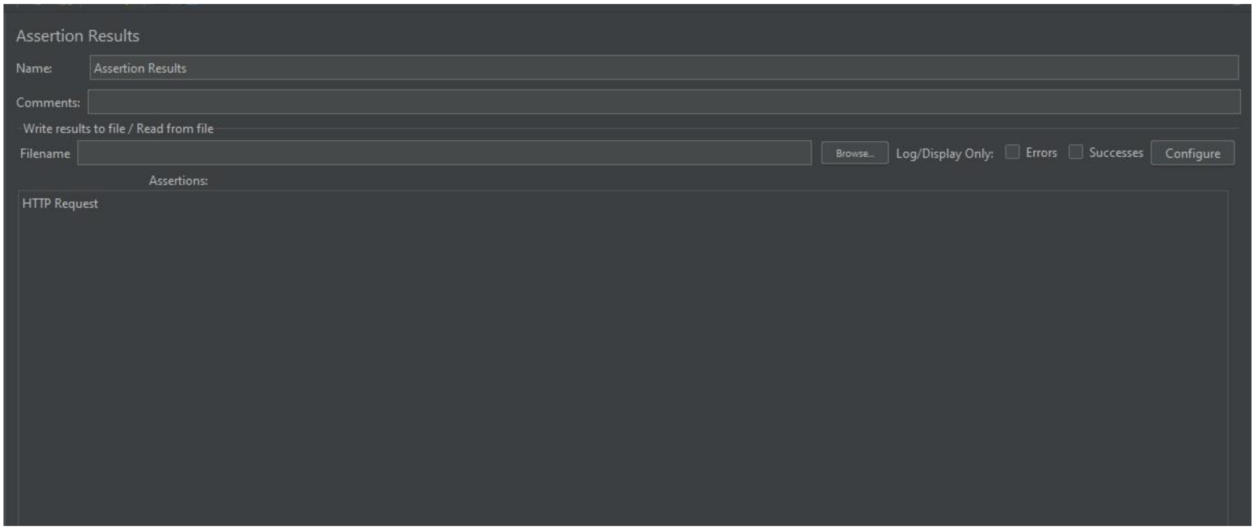
Results



View Results Tree

Name: View Results Tree

HTTP Request



Assertion Results

Name: Assertion Results

Assertions:

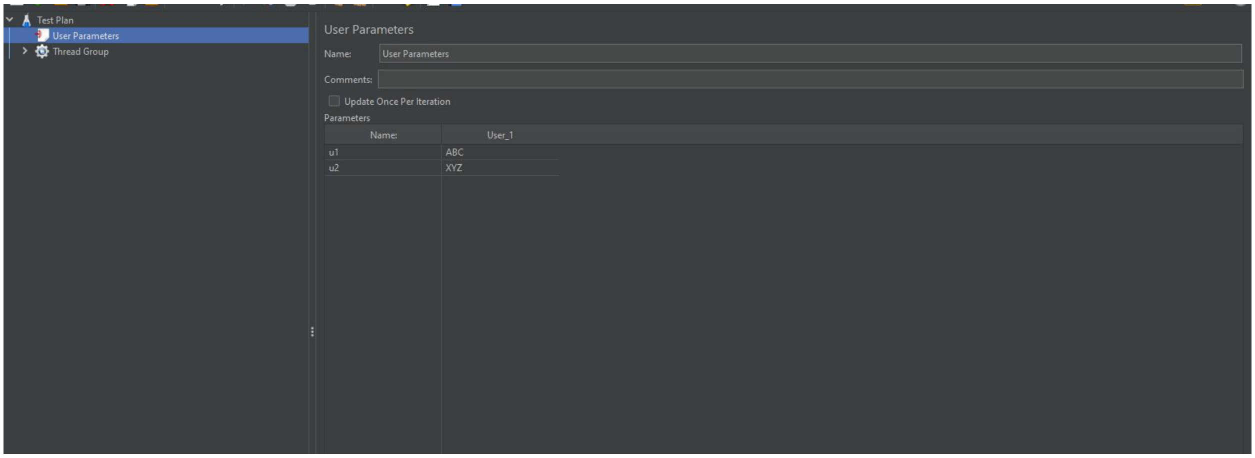
HTTP Request

<u>Pre-Processors</u>

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.



User Parameters

Name: User Parameters

Comments:

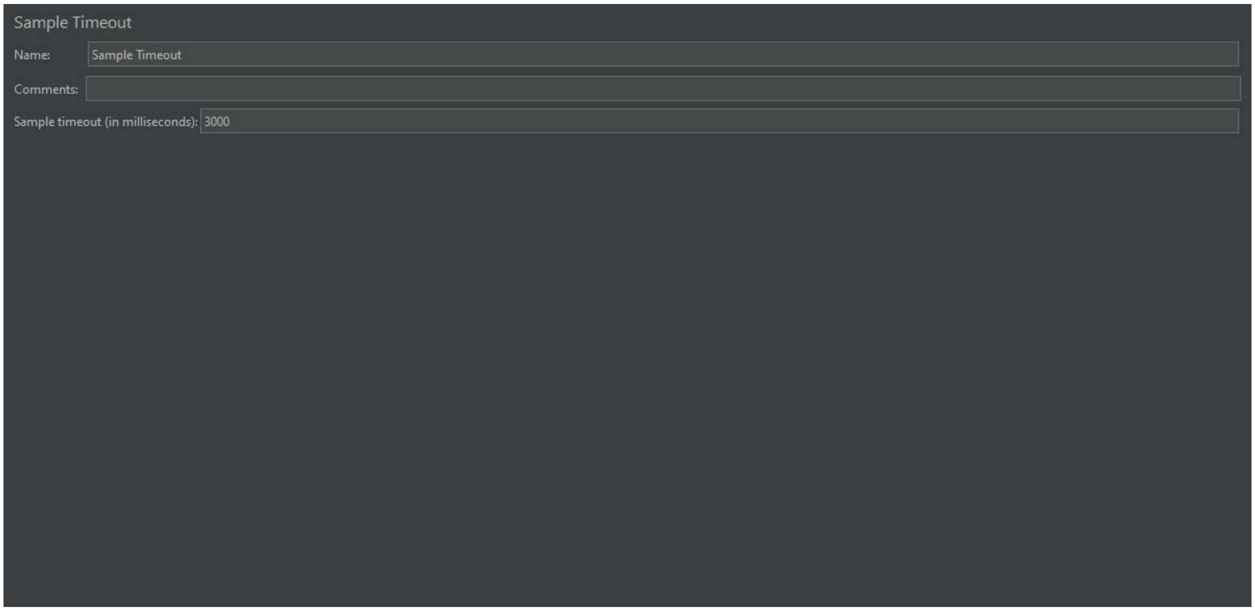
□ Update Once Per Iteration

Parameters

Name:	User_1
u1	ABC
u2	XYZ

- **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.



Sample Timeout

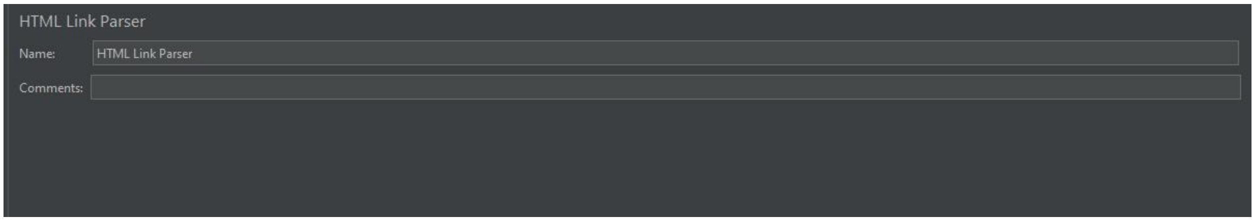
Name: Sample Timeout

Comments:

Sample timeout (in milliseconds): 3000

- **Html Link Parser**

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

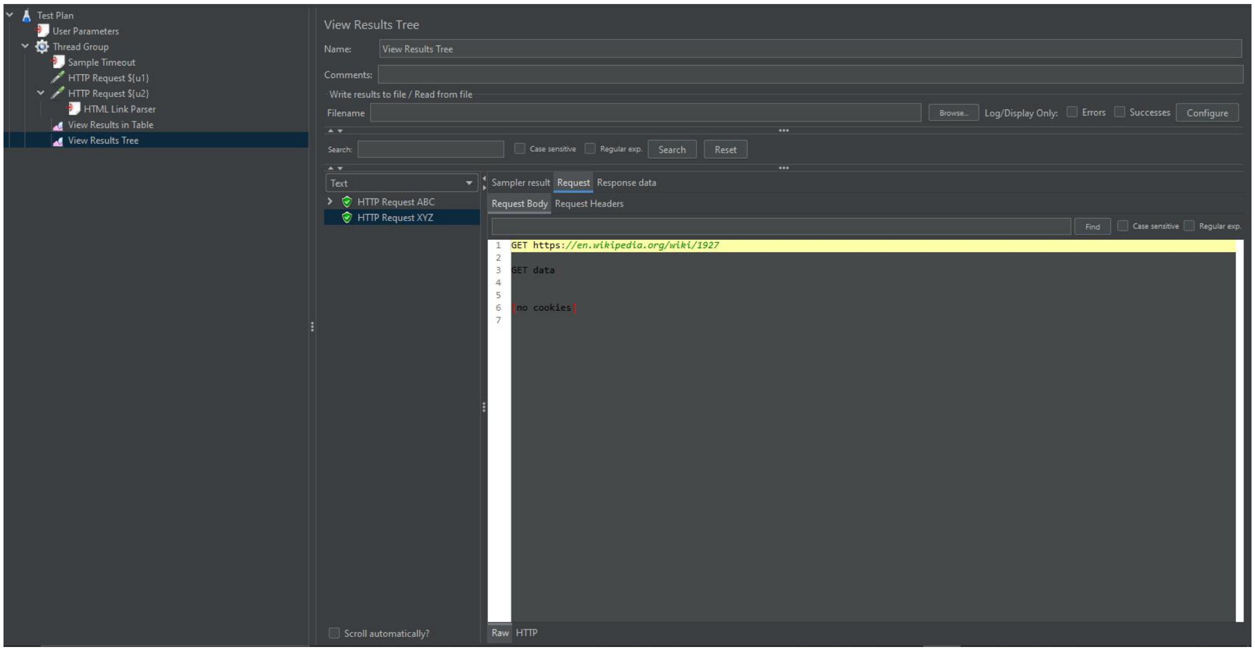


HTML Link Parser

Name: HTML Link Parser

Comments:

Results

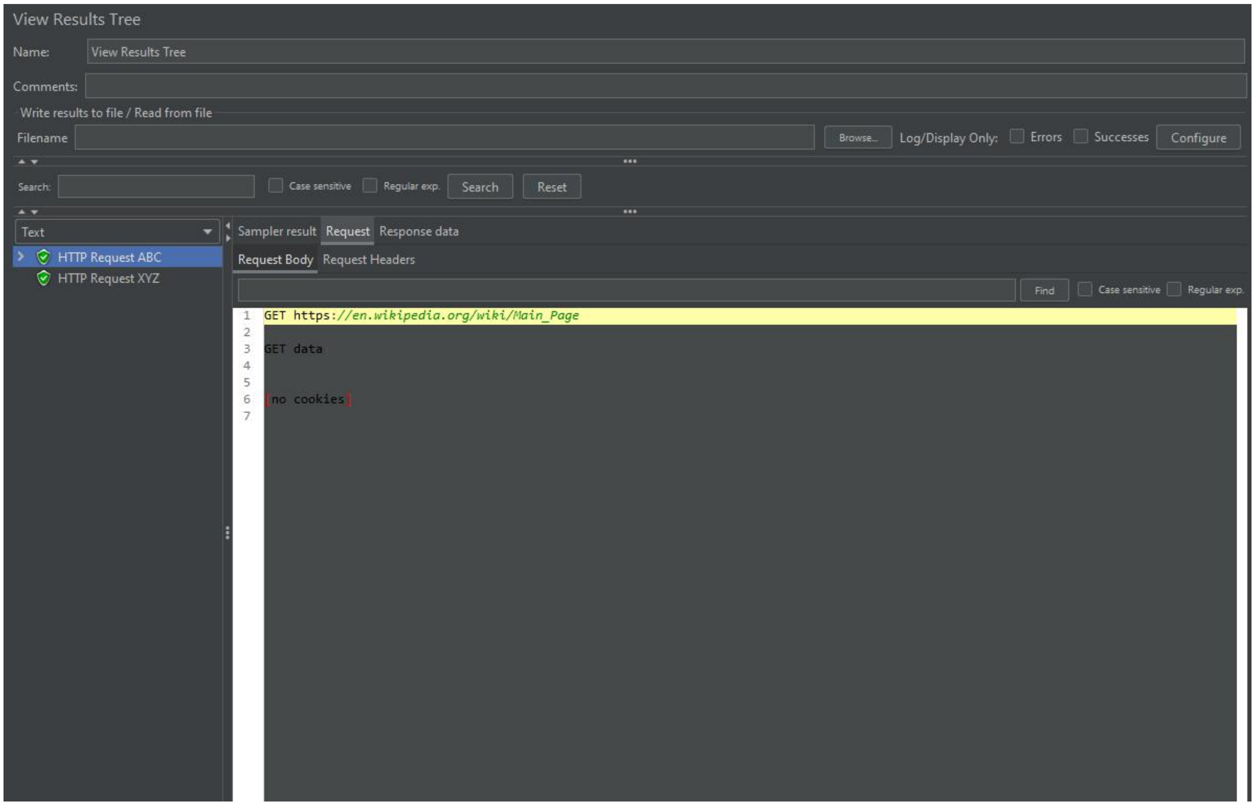


Name: View Results Tree

```
GET https://en.wikipedia.org/wiki/1927

GET data


[no cookies]
```



Name: View Results Tree

```
GET https://en.wikipedia.org/wiki/Main_Page

GET data


[no cookies]
```

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.

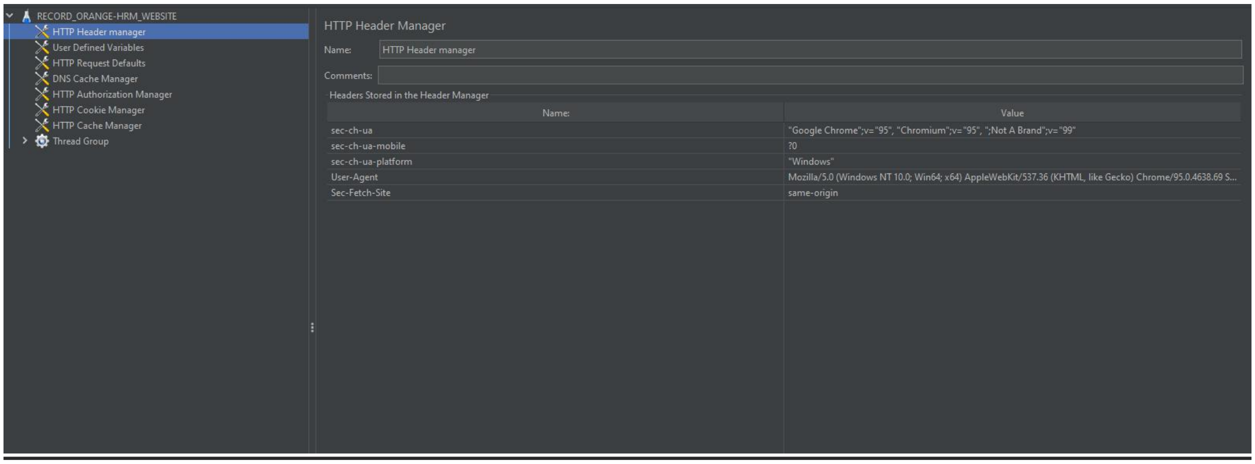
- Http Request Defaults

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



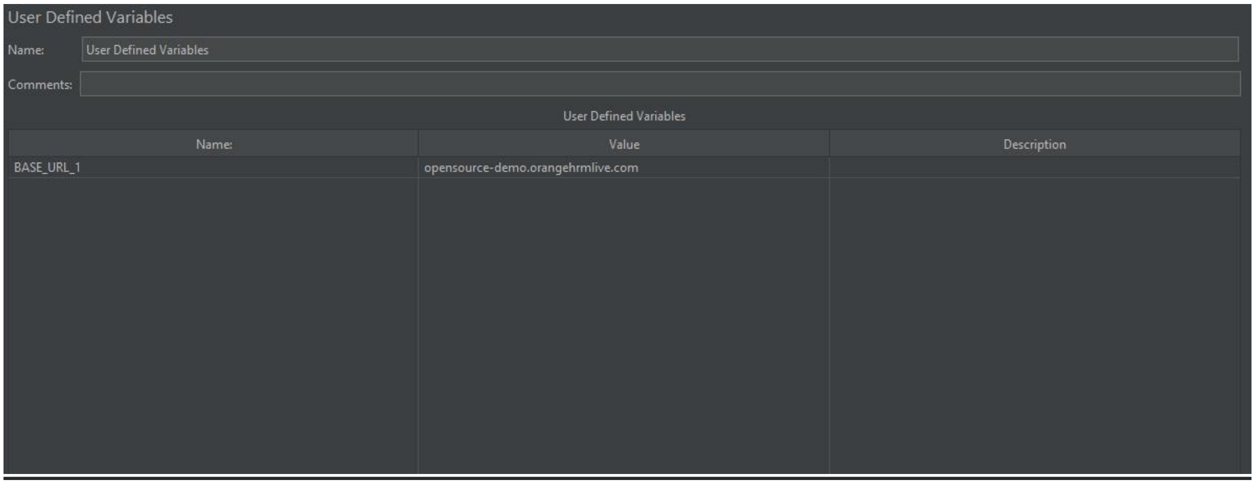
HTTP Header Manager

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



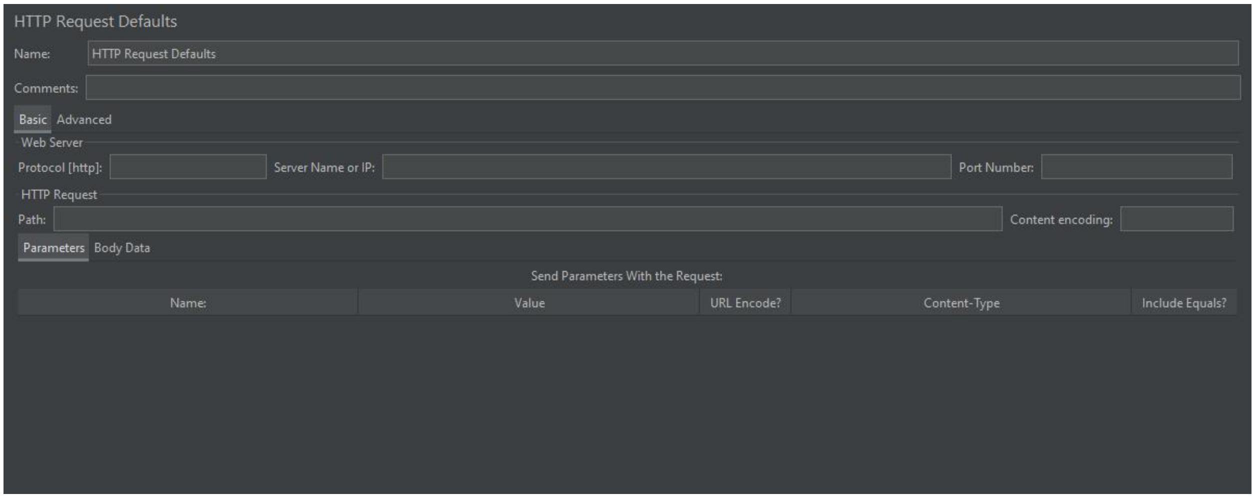
User Defined Variables

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:

Web Server

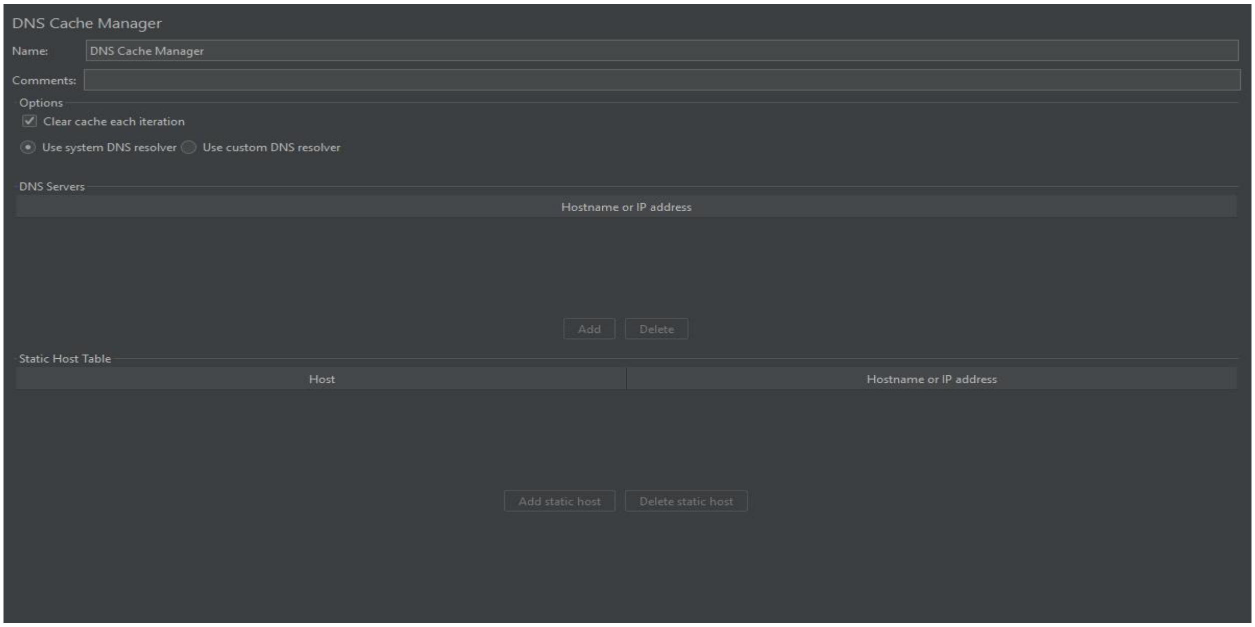
Protocol [http]: Server Name or IP: Port Number:

HTTP Request

Path: Content encoding:

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

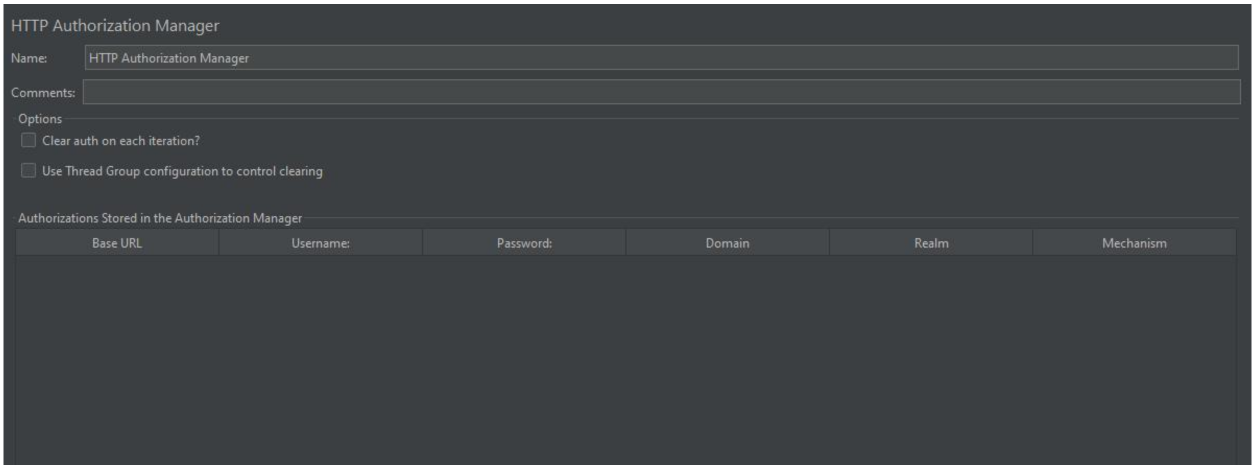
● Use system DNS resolver ○ Use custom DNS resolver

DNS Servers

Hostname or IP address

Static Host Table

Host	Hostname or IP address



HTTP Authorization Manager

Name: HTTP Authorization Manager

Comments:

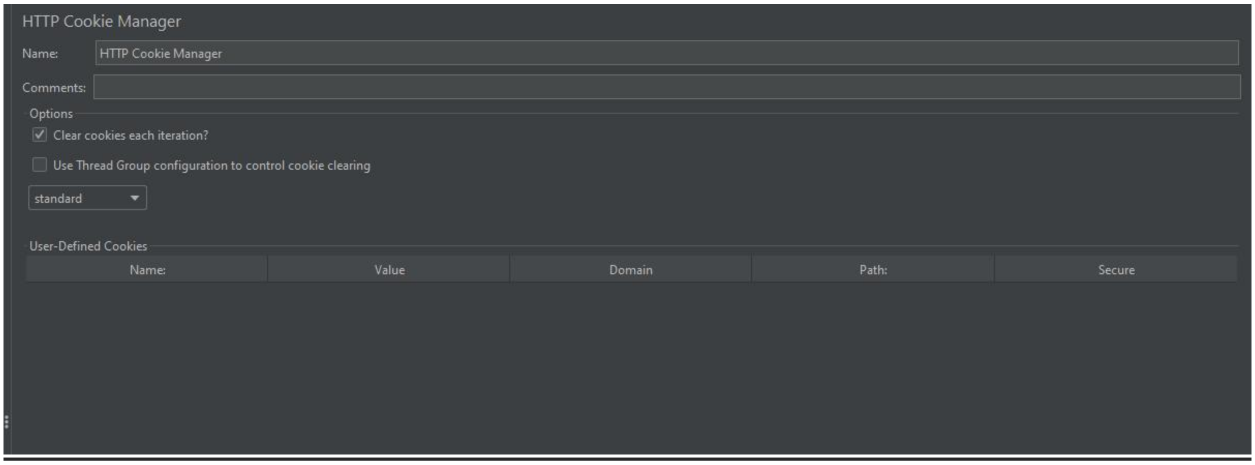
Options

□ 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?

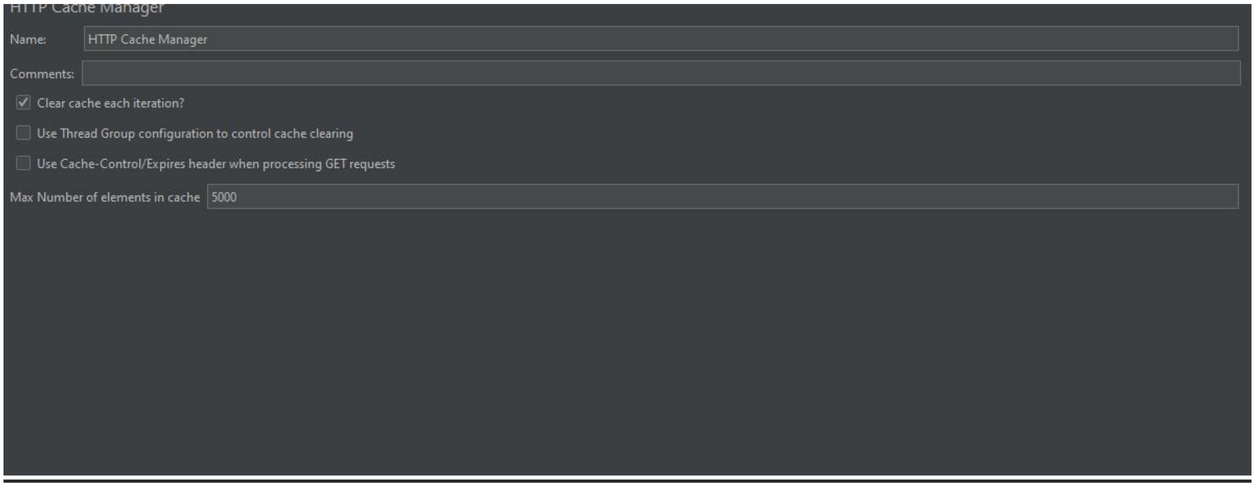
□ Use Thread Group configuration to control cookie clearing

standard

User-Defined Cookies

Name:	Value	Domain	Path:	Secure

Http Cache Manager



HTTP Cache Manager

Name: HTTP Cache Manager

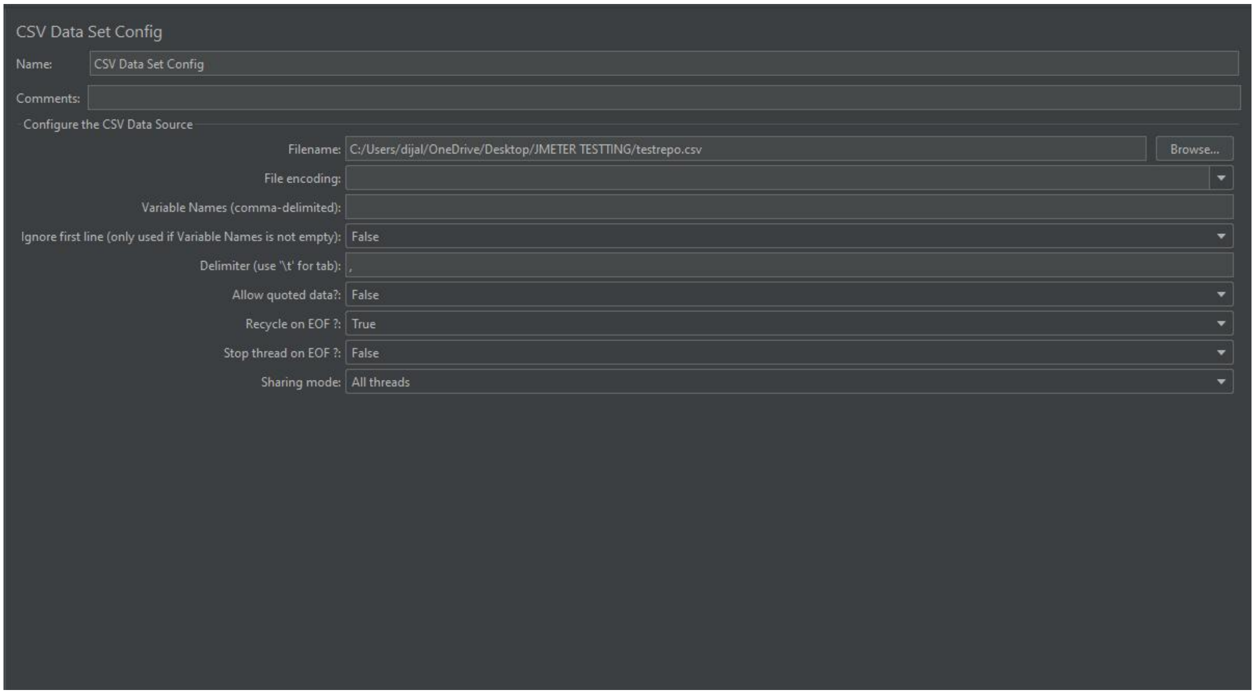
Comments:

☑ 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 5000



CSV Data Set Config

Name: CSV Data Set Config

Comments:

Configure the CSV Data Source

Filename: C:/Users/dijal/OneDrive/Desktop/JMETER TESTTING/testrepo.csv

File encoding:

Variable Names (comma-delimited):

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

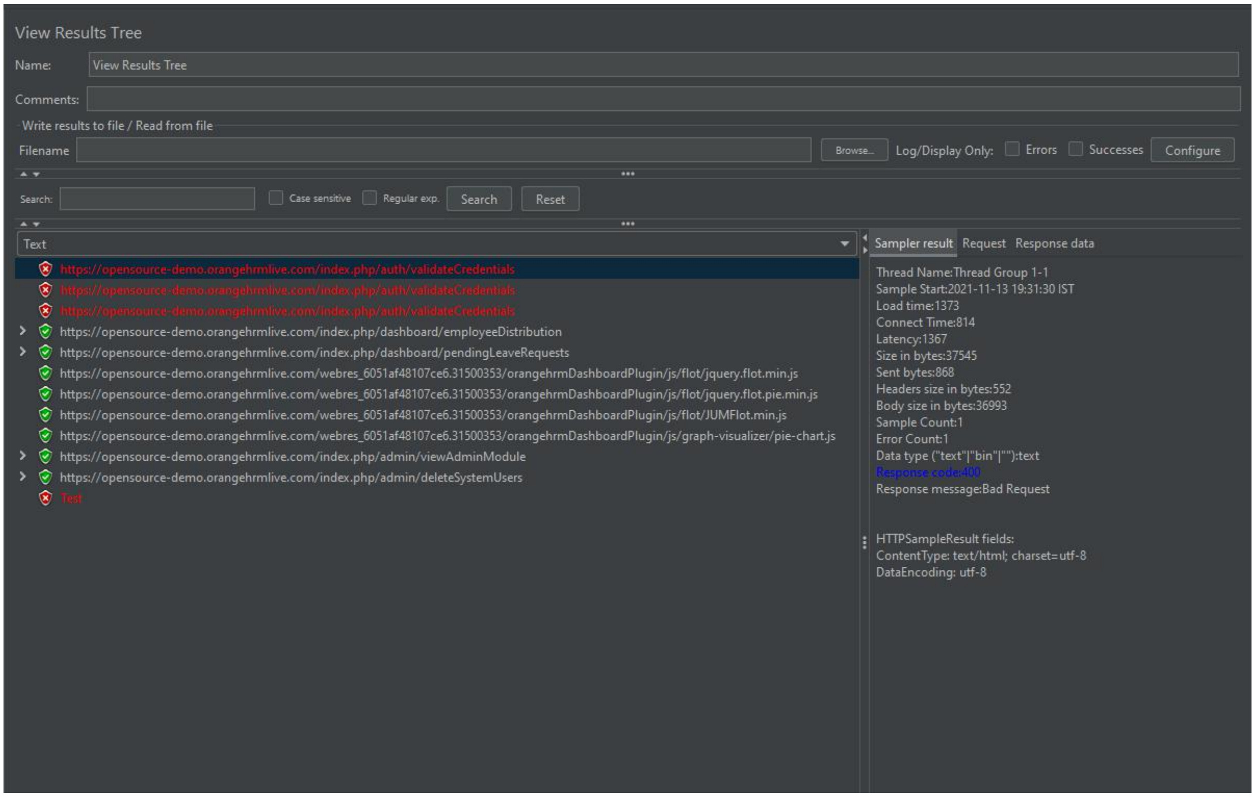
Delimiter (use '\t' for tab): ,

Allow quoted data?: False

Recycle on EOF ?: True

Stop thread on EOF ?: False

Sharing mode: All threads



View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Search:

□ Case sensitive □ Regular exp.

Text

```
https://opensource-demo.orangehrmlive.com/index.php/auth/validateCredentials
https://opensource-demo.orangehrmlive.com/index.php/auth/validateCredentials
https://opensource-demo.orangehrmlive.com/index.php/auth/validateCredentials
https://opensource-demo.orangehrmlive.com/index.php/dashboard/employeeDistribution
https://opensource-demo.orangehrmlive.com/index.php/dashboard/pendingLeaveRequests
https://opensource-demo.orangehrmlive.com/webres_6051af48107ce6.31500353/orangehrmDashboardPlugin/js/flot/jquery.flot.min.js
https://opensource-demo.orangehrmlive.com/webres_6051af48107ce6.31500353/orangehrmDashboardPlugin/js/flot/jquery.flot.pie.min.js
https://opensource-demo.orangehrmlive.com/webres_6051af48107ce6.31500353/orangehrmDashboardPlugin/js/flot/JUMFlot.min.js
https://opensource-demo.orangehrmlive.com/webres_6051af48107ce6.31500353/orangehrmDashboardPlugin/js/graph-visualizer/pie-chart.js
https://opensource-demo.orangehrmlive.com/index.php/admin/viewAdminModule
https://opensource-demo.orangehrmlive.com/index.php/admin/deleteSystemUsers
Test
```

```
Thread Name:Thread Group 1-1
Sample Start:2021-11-13 19:31:30 IST
Load time:1373
Connect Time:814
Latency:1367
Size in bytes:37545
Sent bytes:868
Headers size in bytes:552
Body size in bytes:36993
Sample Count:1
Error Count:1
Data type ("text"|"bin"|""):text
Response code:400
Response message:Bad Request


HTTPSampleResult fields:
ContentType: text/html; charset=utf-8
DataEncoding: utf-8
```

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

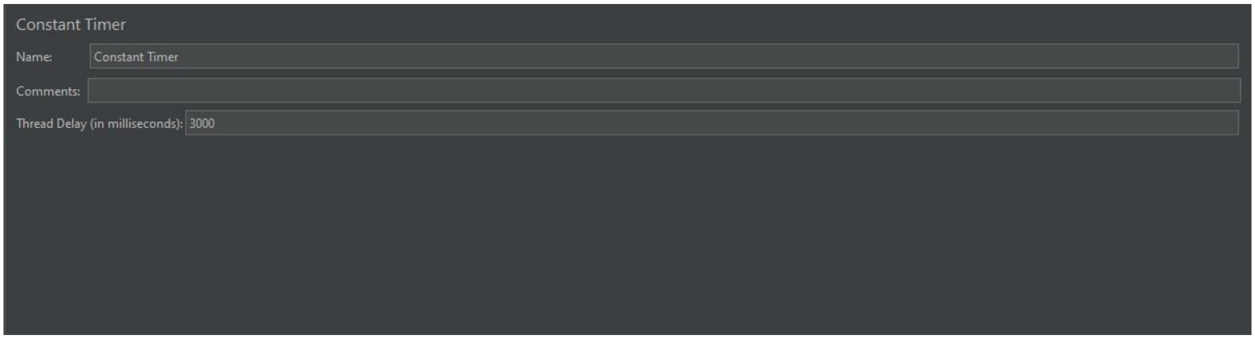
- **Constant Timer**

  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

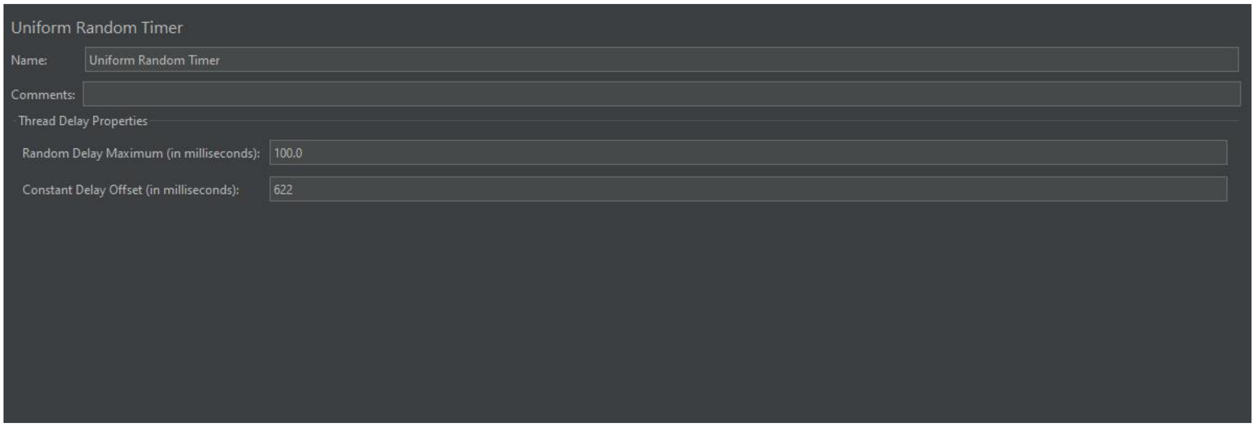


Constant Timer

Name: Constant Timer

Comments:

Thread Delay (in milliseconds): 3000



Uniform Random Timer

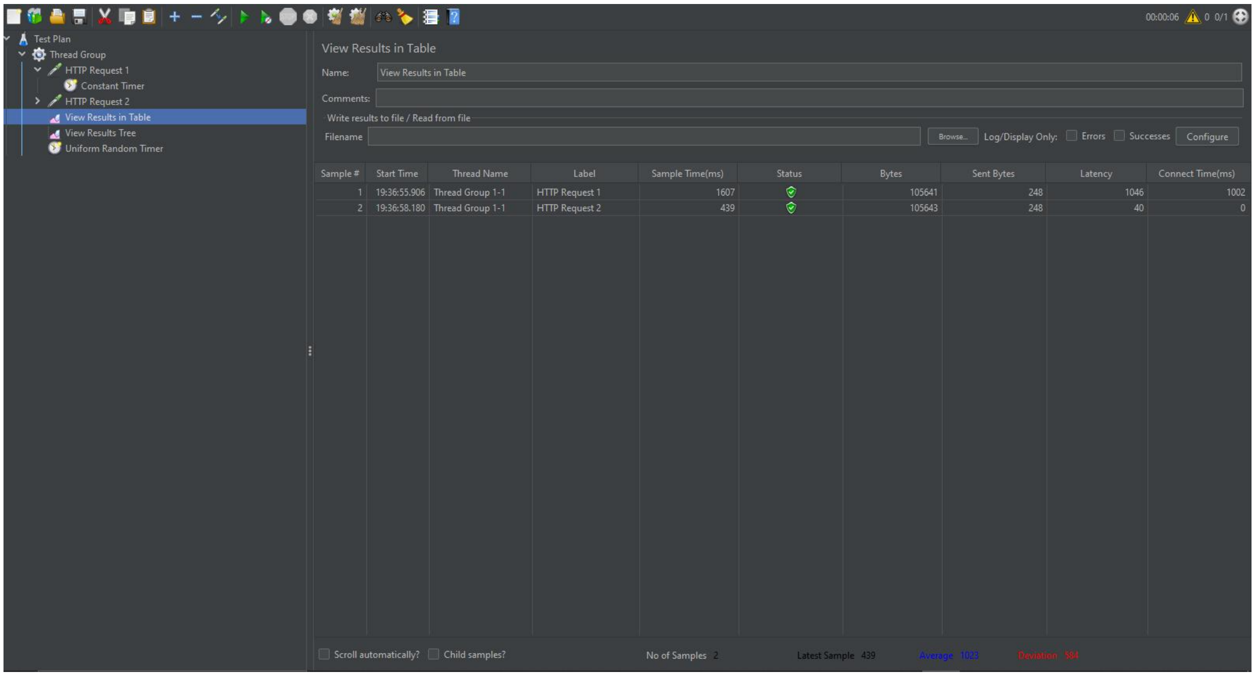
Name: Uniform Random Timer

Comments:

Thread Delay Properties

Random Delay Maximum (in milliseconds): 100.0

Constant Delay Offset (in milliseconds): 622



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	✓	105641	248	1046	1002
2	19:36:58.180	Thread Group 1-1	HTTP Request 2	439	✓	105643	248	40	0

□ Scroll automatically? □ Child samples?

No of Samples 2 Latest Sample 439 Average 1023 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).

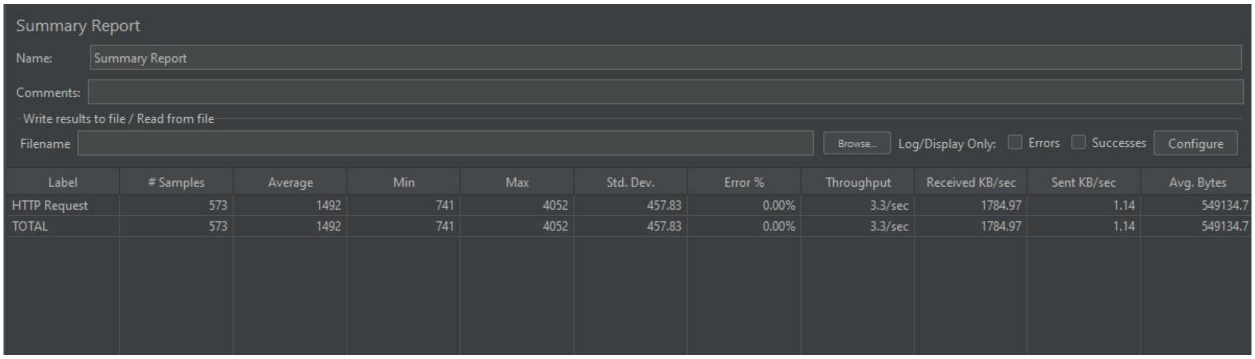
- **View Results Tree**

  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



Summary Report

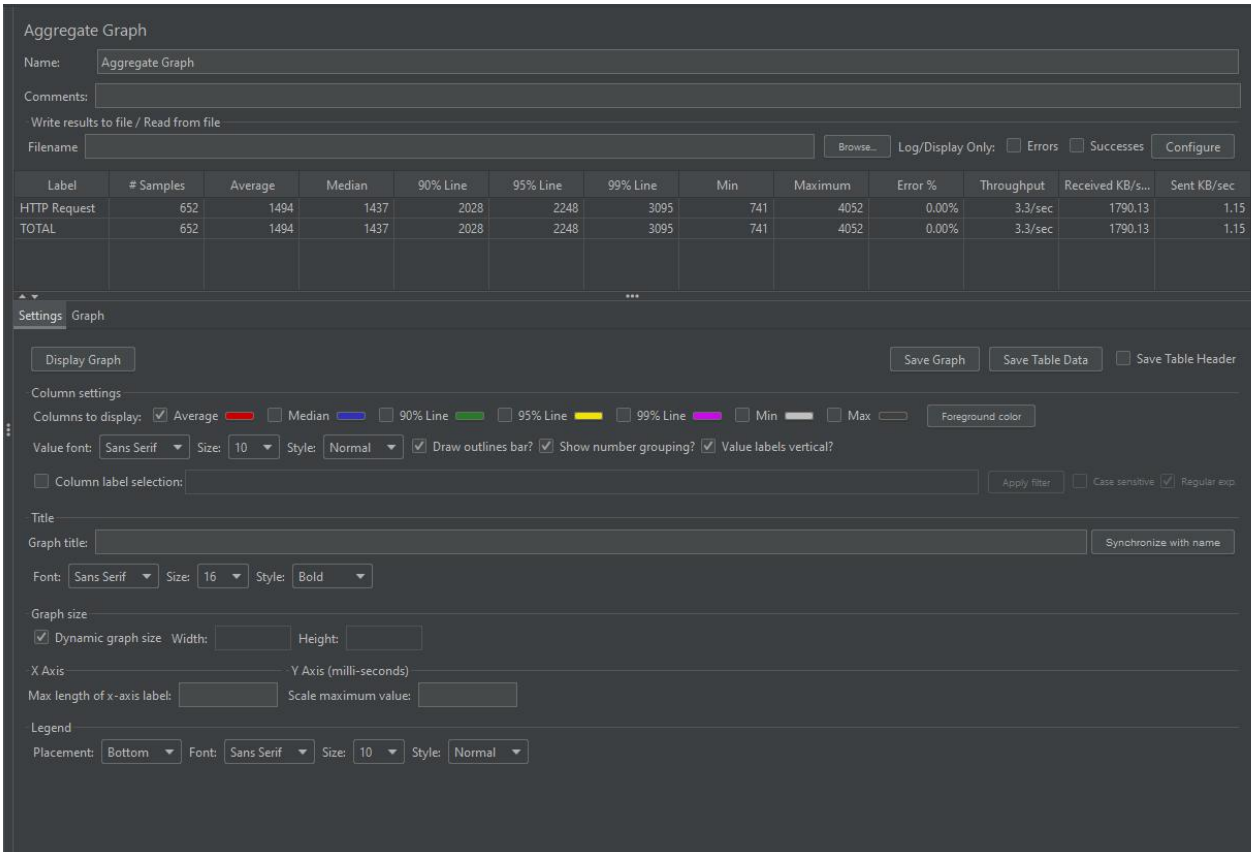
Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

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: Aggregate Graph

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

☐ Save Table Header

Column settings

Columns to display: ☑ Average 🟥 ☐ Median 🟦 ☐ 90% Line 🟩 ☐ 95% Line 🟨 ☐ 99% Line 🟪 ☐ Min ⬜ ☐ Max ▭

Value font: Sans Serif Size: 10 Style: Normal ☑ Draw outlines bar? ☑ Show number grouping? ☑ Value labels vertical?

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

Title

Graph title:

Font: Sans Serif Size: 16 Style: Bold

Graph size

☑ Dynamic graph size Width: Height:

X Axis

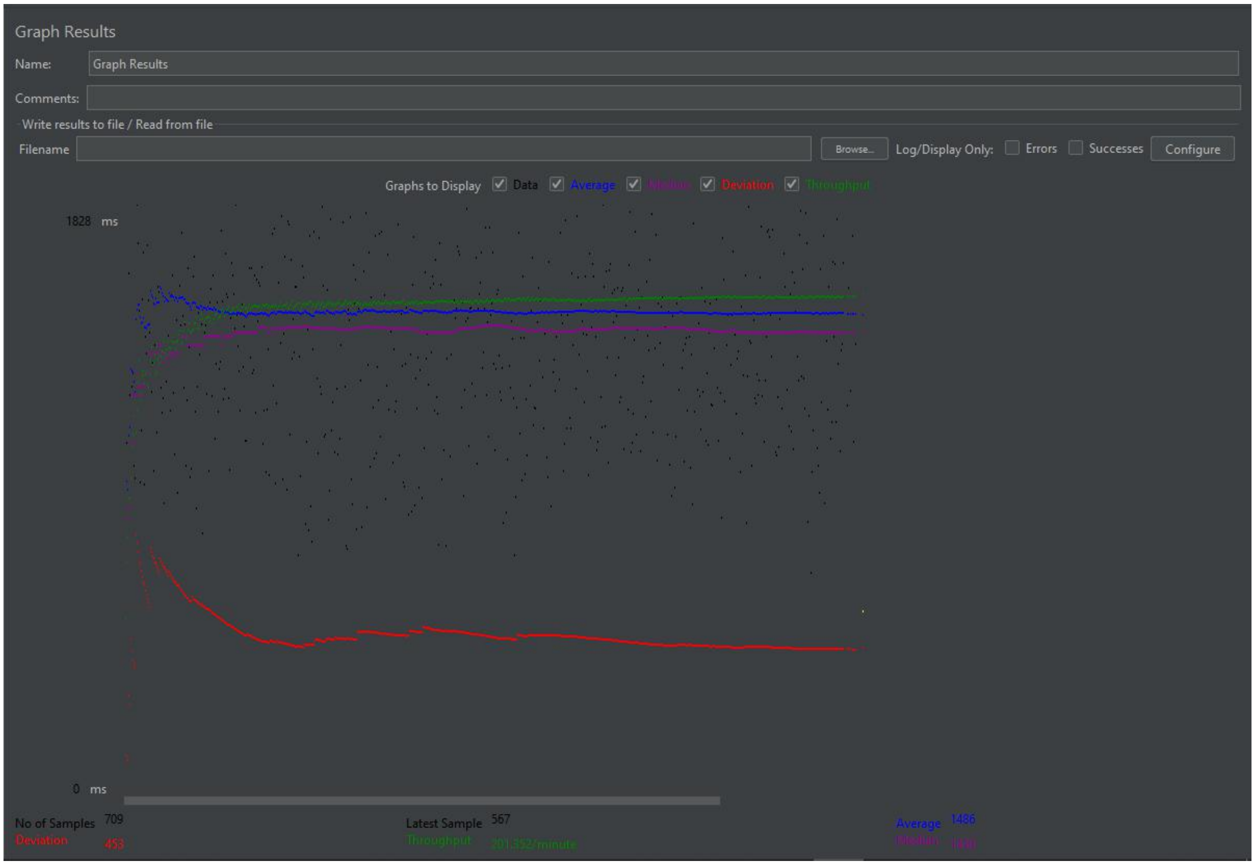
Max length of x-axis label:

Y Axis (milli-seconds)

Scale maximum value:

Legend

Placement: Bottom Font: Sans Serif Size: 10 Style: Normal


Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

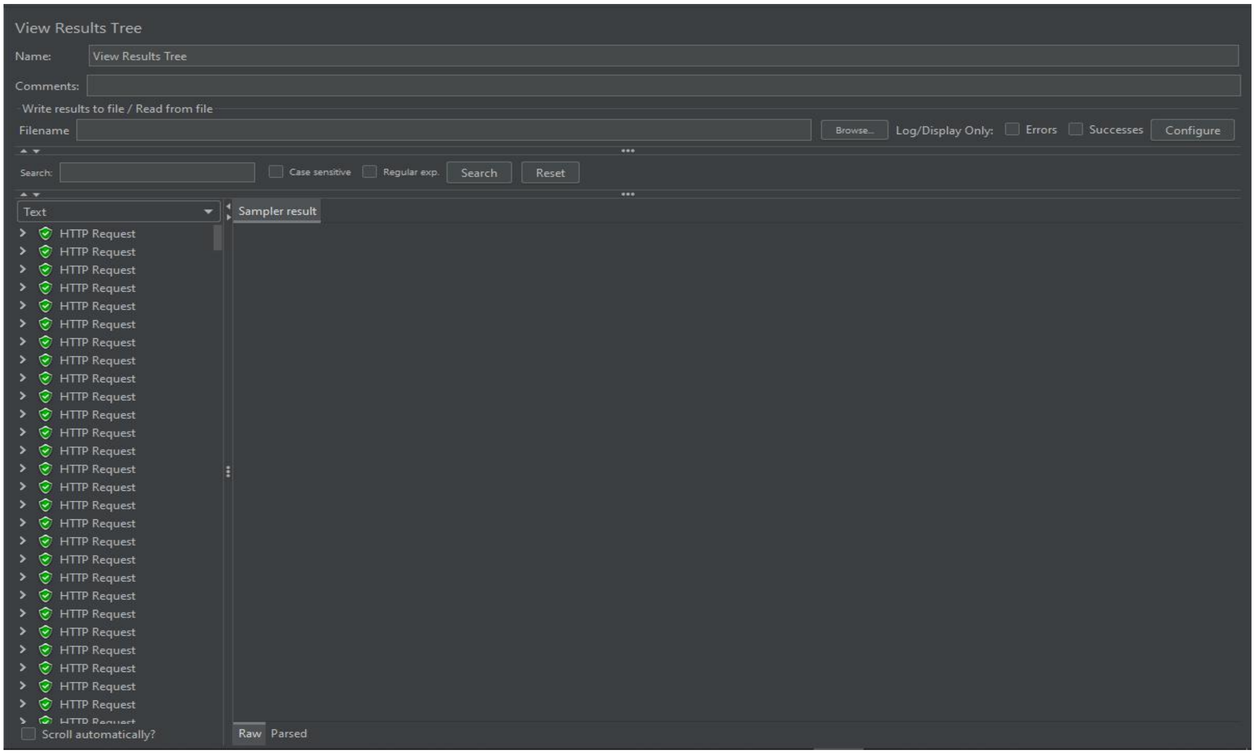
Filename

Browse...
Log/Display Only: □ Errors □ Successes
Configure
Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput
1828 ms
0 ms

No of Samples 709
Deviation 453

Latest Sample 567
Throughput 201.352/minute

Average 1486
Median 1430

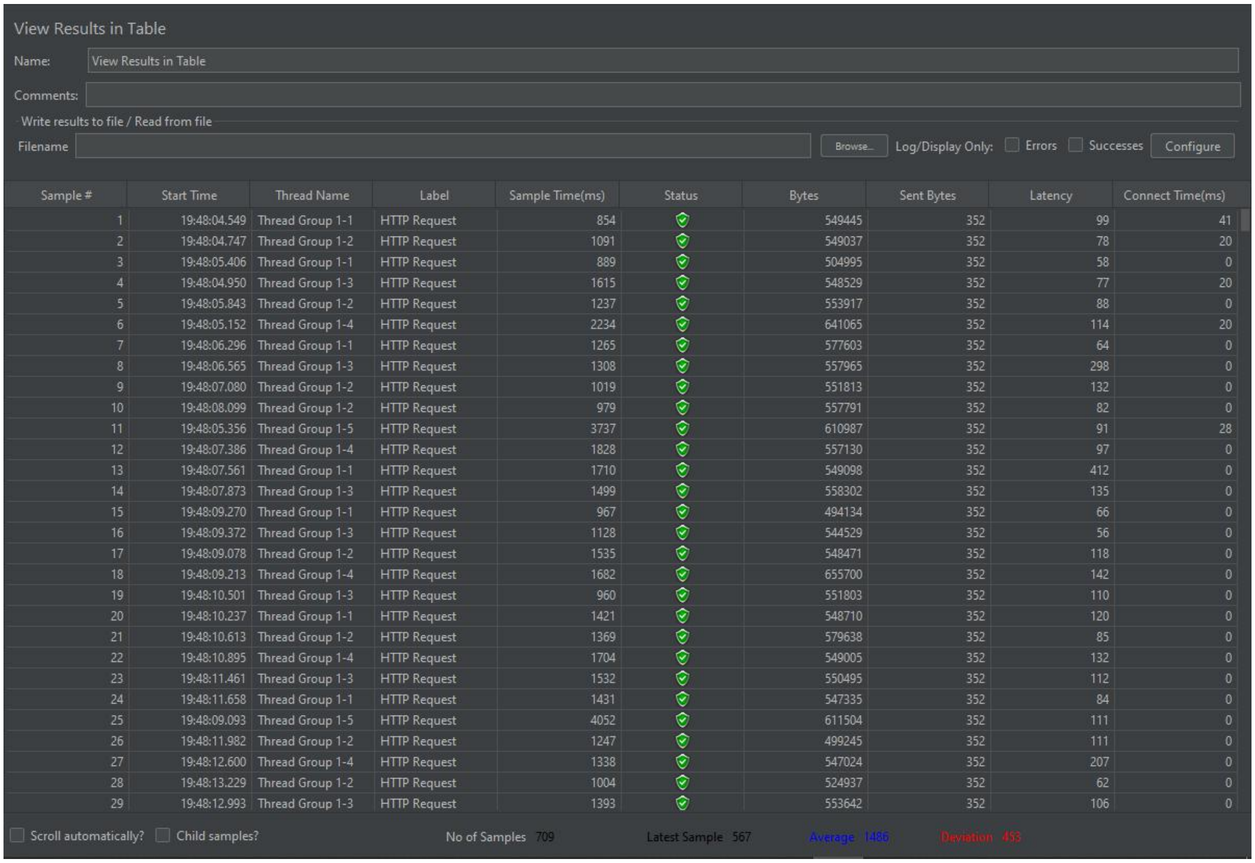

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename



View Results in Table

Name: View Results in Table

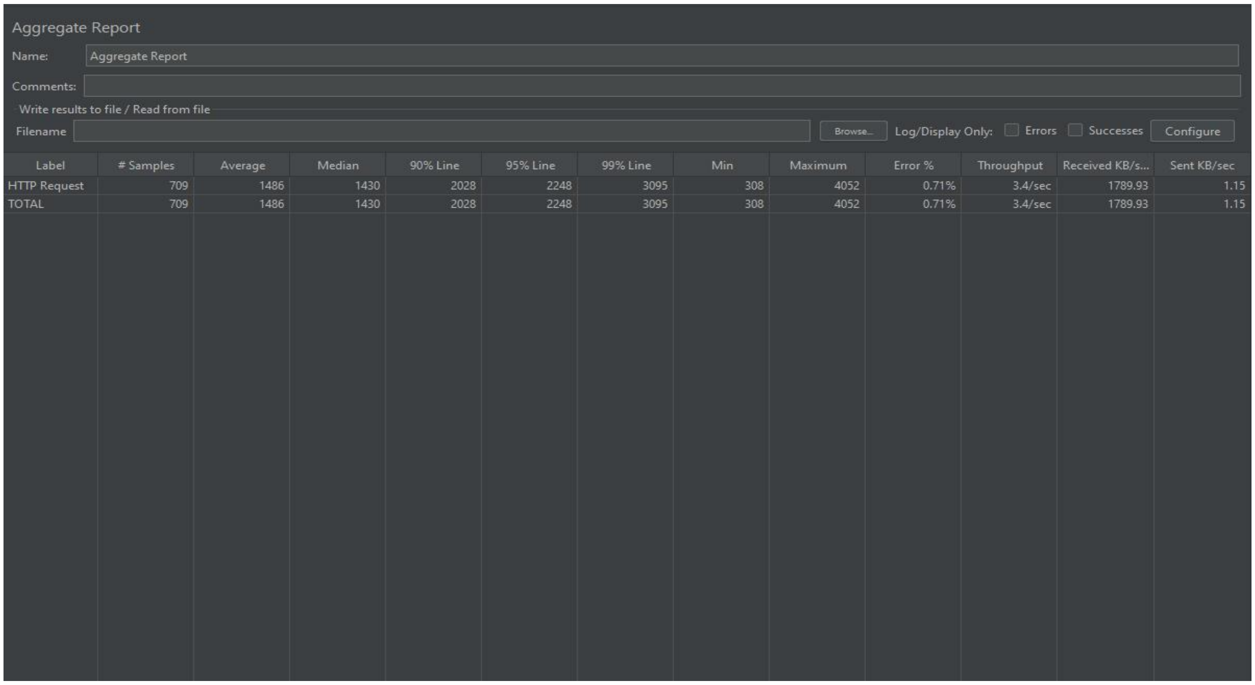
Comments:

Write results to file / Read from file

Filename

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	19:48:04.549	Thread Group 1-1	HTTP Request	854	✓	549445	352	99	41
2	19:48:04.747	Thread Group 1-2	HTTP Request	1091	✓	549037	352	78	20
3	19:48:05.406	Thread Group 1-1	HTTP Request	889	✓	504995	352	58	0
4	19:48:04.950	Thread Group 1-3	HTTP Request	1615	✓	548529	352	77	20
5	19:48:05.843	Thread Group 1-2	HTTP Request	1237	✓	553917	352	88	0
6	19:48:05.152	Thread Group 1-4	HTTP Request	2234	✓	641065	352	114	20
7	19:48:06.296	Thread Group 1-1	HTTP Request	1265	✓	577603	352	64	0
8	19:48:06.565	Thread Group 1-3	HTTP Request	1308	✓	557965	352	298	0
9	19:48:07.080	Thread Group 1-2	HTTP Request	1019	✓	551813	352	132	0
10	19:48:08.099	Thread Group 1-2	HTTP Request	979	✓	557791	352	82	0
11	19:48:05.356	Thread Group 1-5	HTTP Request	3737	✓	610987	352	91	28
12	19:48:07.386	Thread Group 1-4	HTTP Request	1828	✓	557130	352	97	0
13	19:48:07.561	Thread Group 1-1	HTTP Request	1710	✓	549098	352	412	0
14	19:48:07.873	Thread Group 1-3	HTTP Request	1499	✓	558302	352	135	0
15	19:48:09.270	Thread Group 1-1	HTTP Request	967	✓	494134	352	66	0
16	19:48:09.372	Thread Group 1-3	HTTP Request	1128	✓	544529	352	56	0
17	19:48:09.078	Thread Group 1-2	HTTP Request	1535	✓	548471	352	118	0
18	19:48:09.213	Thread Group 1-4	HTTP Request	1682	✓	655700	352	142	0
19	19:48:10.501	Thread Group 1-3	HTTP Request	960	✓	551803	352	110	0
20	19:48:10.237	Thread Group 1-1	HTTP Request	1421	✓	548710	352	120	0
21	19:48:10.613	Thread Group 1-2	HTTP Request	1369	✓	579638	352	85	0
22	19:48:10.895	Thread Group 1-4	HTTP Request	1704	✓	549005	352	132	0
23	19:48:11.461	Thread Group 1-3	HTTP Request	1532	✓	550495	352	112	0
24	19:48:11.658	Thread Group 1-1	HTTP Request	1431	✓	547335	352	84	0
25	19:48:09.093	Thread Group 1-5	HTTP Request	4052	✓	611504	352	111	0
26	19:48:11.982	Thread Group 1-2	HTTP Request	1247	✓	499245	352	111	0
27	19:48:12.600	Thread Group 1-4	HTTP Request	1338	✓	547024	352	207	0
28	19:48:13.229	Thread Group 1-2	HTTP Request	1004	✓	524937	352	62	0
29	19:48:12.993	Thread Group 1-3	HTTP Request	1393	✓	553642	352	106	0

□ Scroll automatically? □ Child samples? No of Samples 709 Latest Sample 567 Average 1486 Deviation 453



Aggregate Report

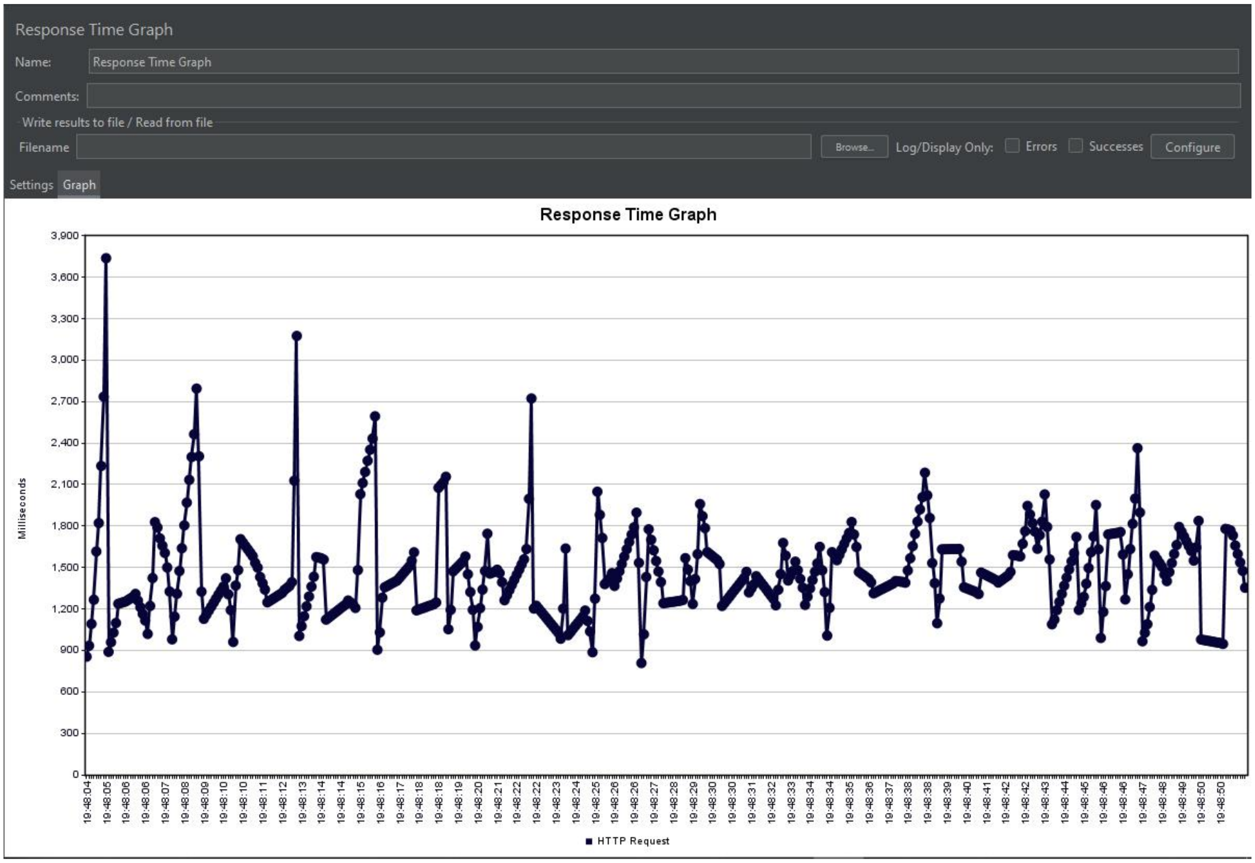
Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

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



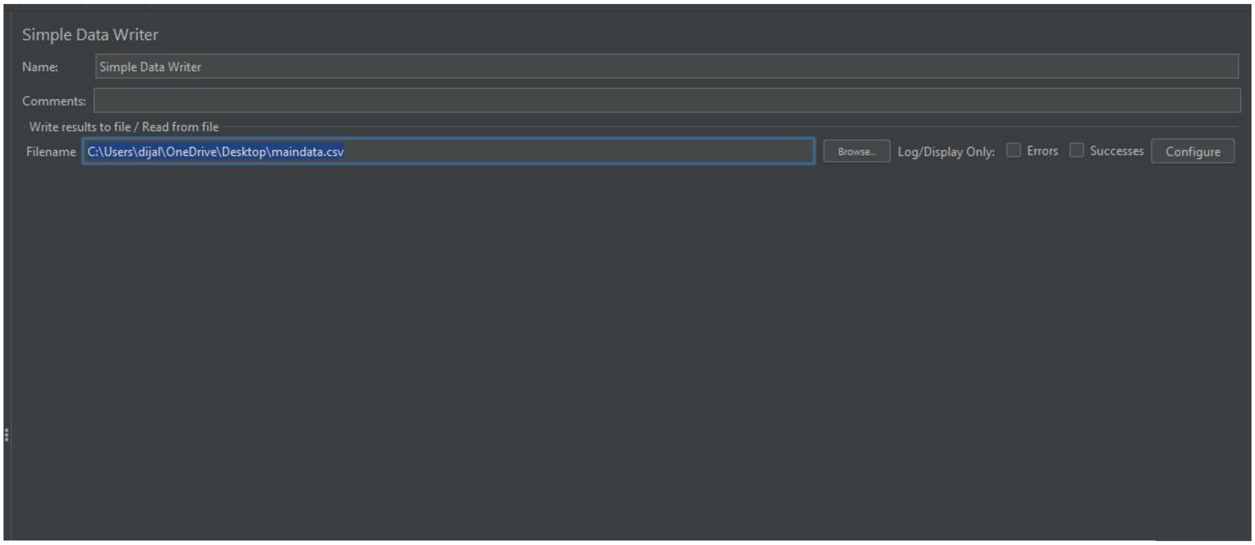
Response Time Graph

Name: Response Time Graph

Comments:

Write results to file / Read from file

Filename



Simple Data Writer

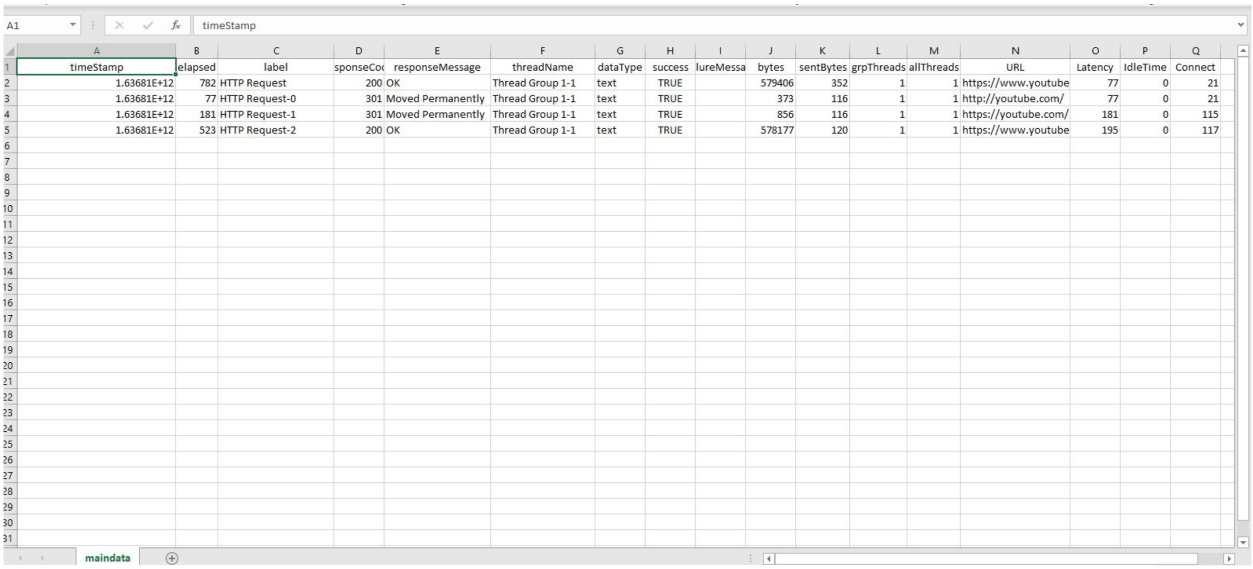
Name: Simple Data Writer

Comments:

Write results to file / Read from file

Filename C:\Users\dijal\OneDrive\Desktop\maindata.csv

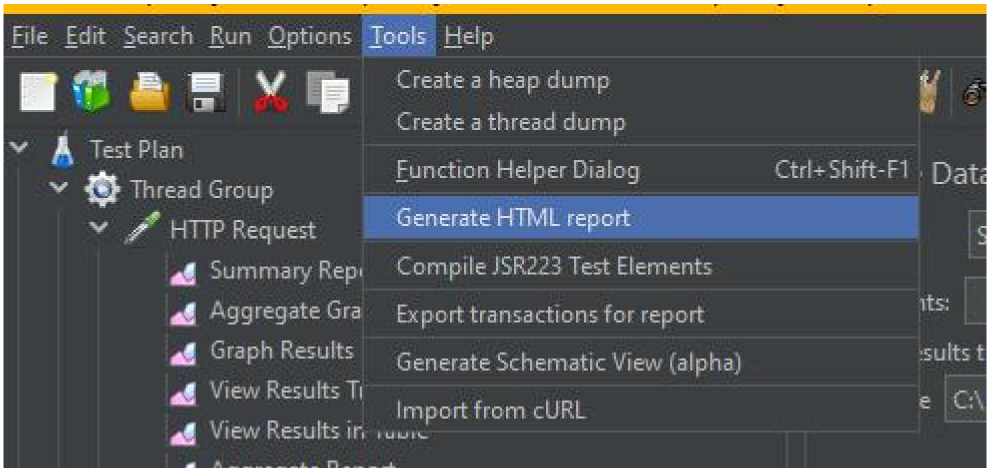
Csv File Create Via Simple Data Writer

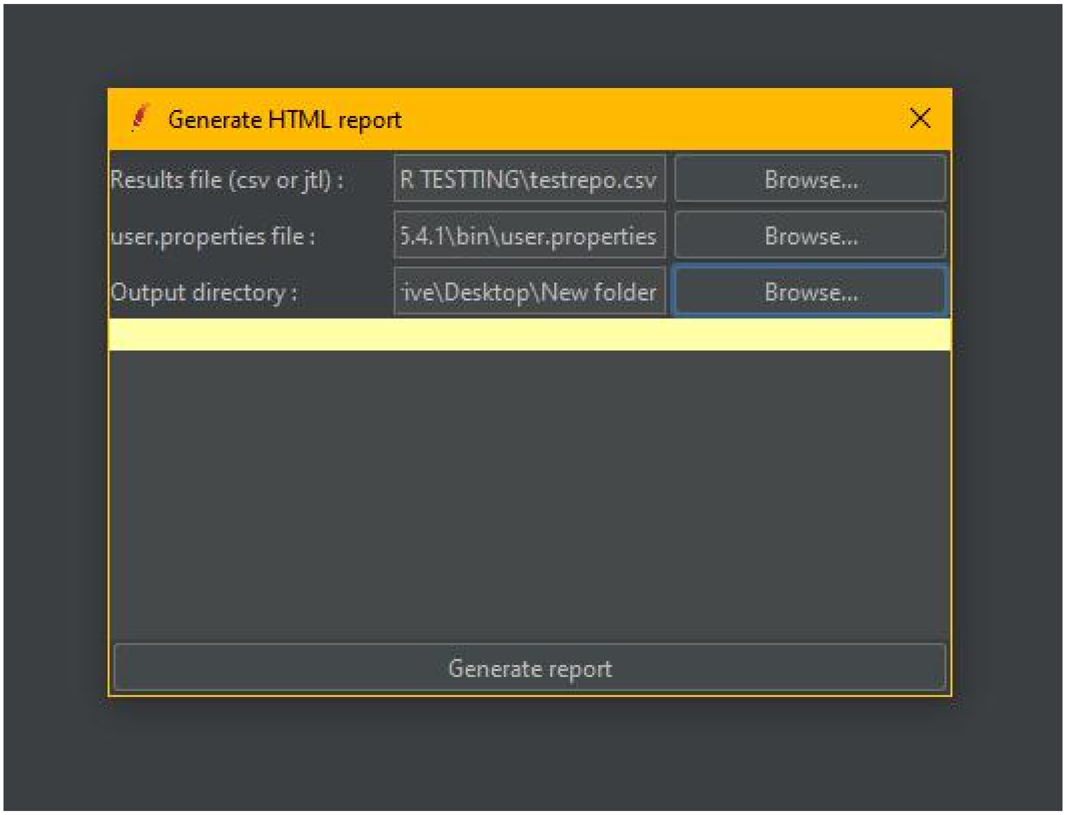


A1 timeStamp

timeStamp	elapsed	label	sponseCo	responseMessage	threadName	dataType	success	lureMessa	bytes	sentBytes	grpThreads	allThreads	URL	Latency	IdleTime	Connect
1.63681E+12	782	HTTP Request	200	OK	Thread Group 1-1	text	TRUE		579406	352	1	1	https://www.youtube	77	0	21
1.63681E+12	77	HTTP Request-0	301	Moved Permanently	Thread Group 1-1	text	TRUE		373	116	1	1	http://youtube.com/	77	0	21
1.63681E+12	181	HTTP Request-1	301	Moved Permanently	Thread Group 1-1	text	TRUE		856	116	1	1	https://youtube.com/	181	0	115
1.63681E+12	523	HTTP Request-2	200	OK	Thread Group 1-1	text	TRUE		578177	120	1	1	https://www.youtube	195	0	117

Creating Html Reports

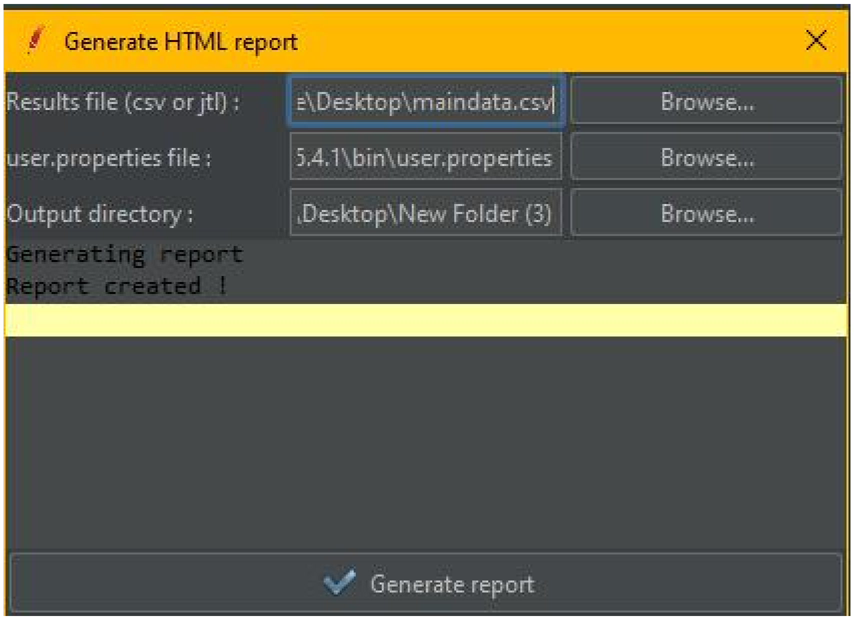




Results file (csv or jtl) : R TESTTING\testrepo.csv

user.properties file : 5.4.1\bin\user.properties

Output directory : ive\Desktop\New folder

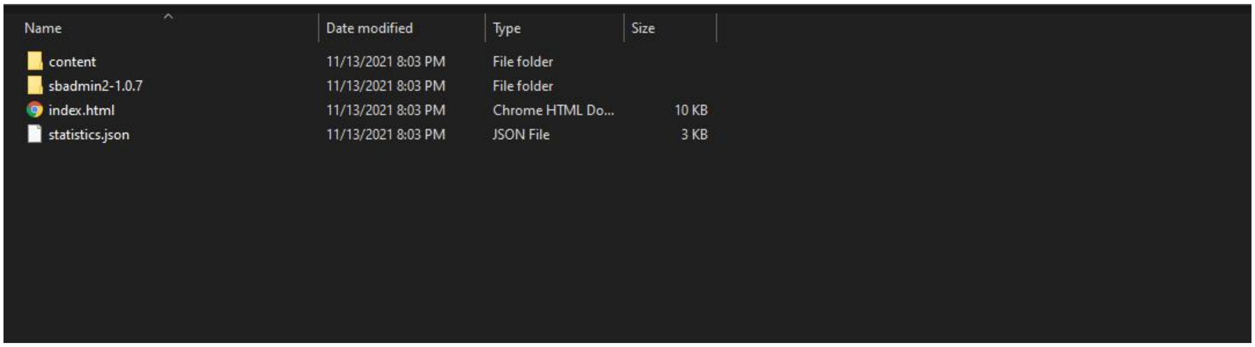


Results file (csv or jtl) : e\Desktop\maindata.csv

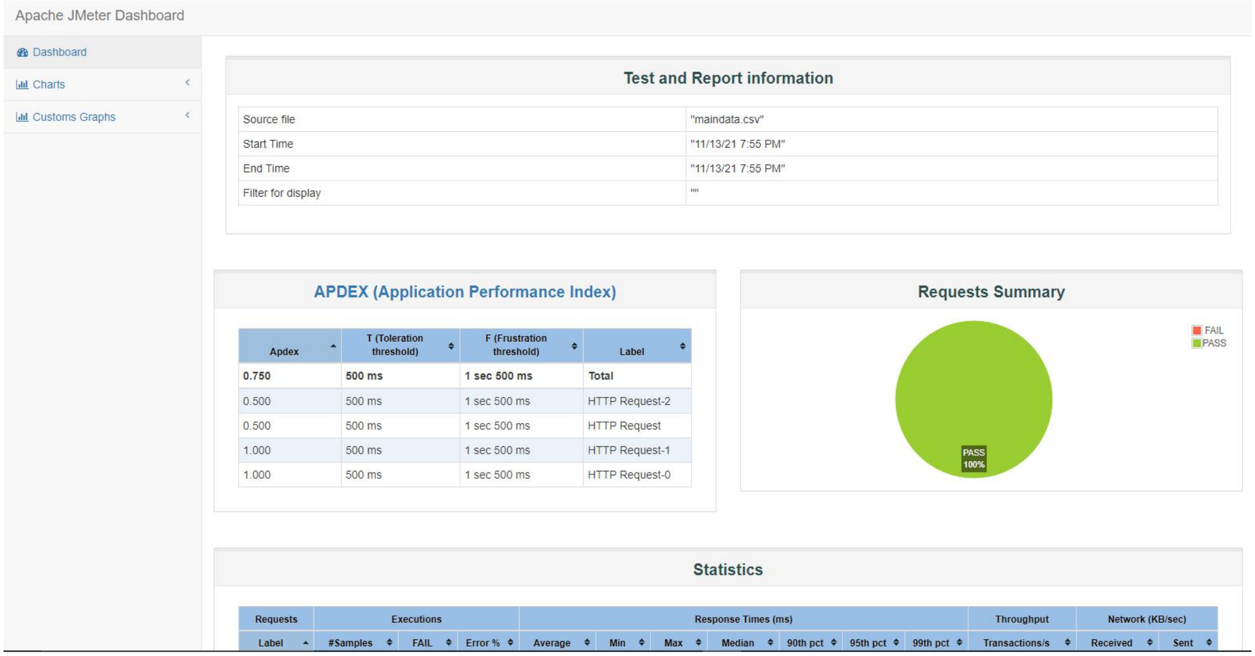
user.properties file : 5.4.1\bin\user.properties

Output directory : ,Desktop\New Folder (3)

```
Generating report
Report created !
```



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



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	""

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