Fresher Academy

Extending Jmeter

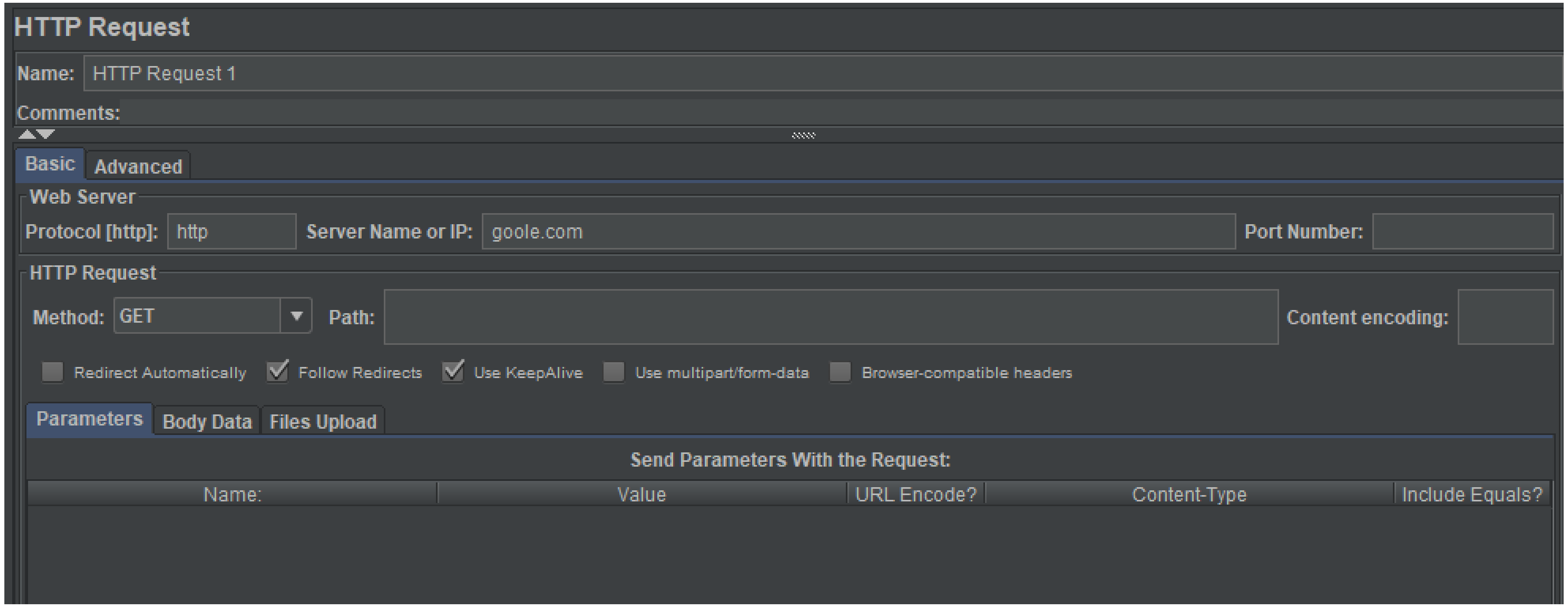
Agenda

- HTTP Request
- HTML Assertion
- Listeners
- Ultimate Thread Group
- Servers Performance Monitoring Kiểm tra thông tin phần cứng khi kết nối với máy, được thể hiện qua report
- Constant Timer Khoảng thời gian giữa 2 request
- User defined variable
- CSV data set config
- Counter
- Random variable

FRESHER ACADEMY

HTTP Request

The HTTP Request sampler is used when you want to use POST, GET, DELETE, PUT, etc., methods over HTTP(S) on the target web application



HTTP Request

Name: HTTP Request 1

Comments:

Basic | Advanced

Web Server

Protocol [http]: http Server Name or IP: goole.com Port Number:

HTTP Request

Method: GET Path: Content encoding:

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

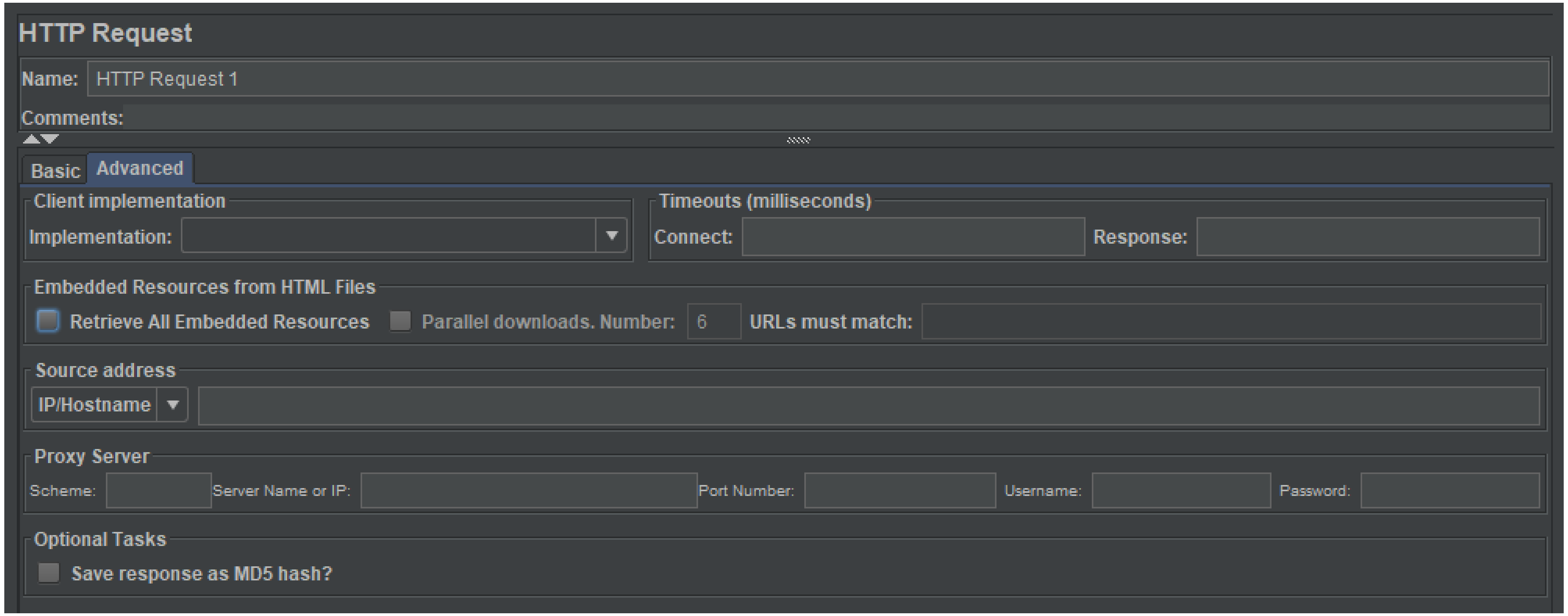
Parameters | Body Data | Files Upload

Send Parameters With the Request:

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

HTTP Request

The HTTP Request sampler is used when you want to use POST, GET, DELETE, PUT, etc., methods over HTTP(S) on the target web application


HTTP Request

Name: HTTP Request 1

Comments:

Basic | Advanced

Client implementation

Implementation:

Timeouts (milliseconds)

Connect: | Response:

Embedded Resources from HTML Files

Retrieve All Embedded Resources | Parallel downloads. Number: 6 | URLs must match:

Source address

IP/Hostname

Proxy Server

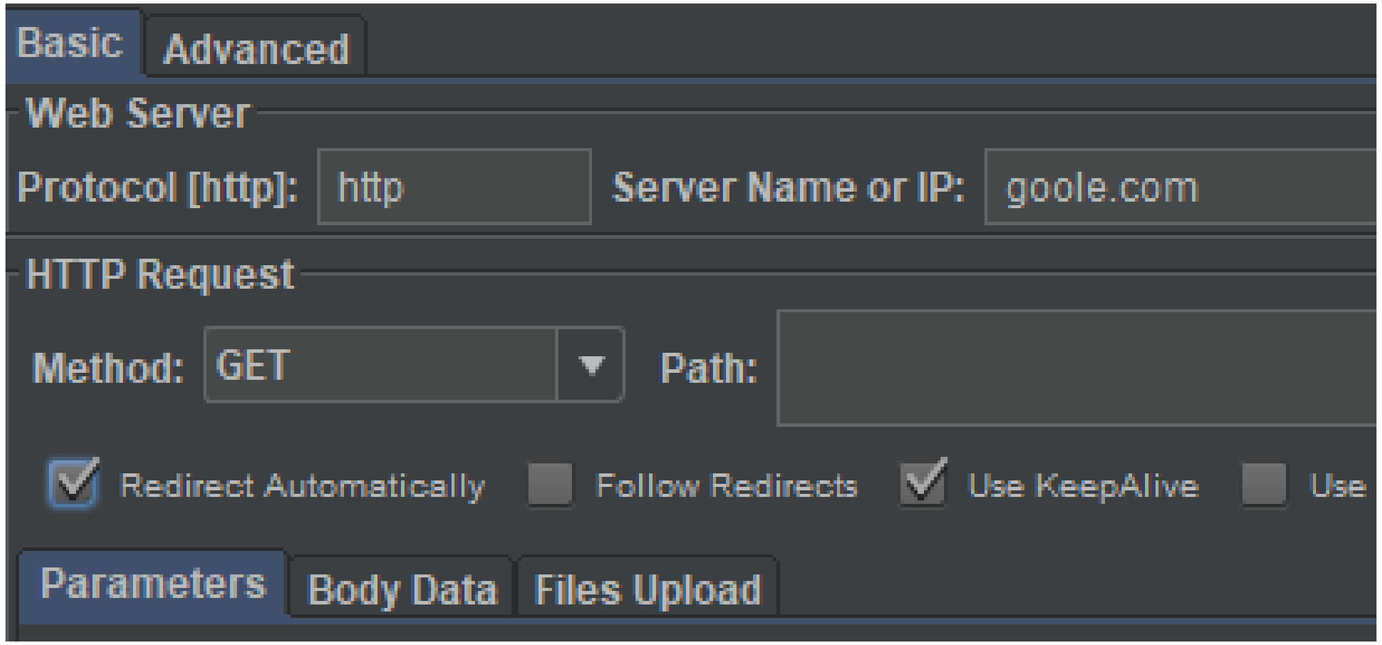
Scheme: | Server Name or IP: | Port Number: | Username: | Password:

Optional Tasks

Save response as MD5 hash?

HTTP Request

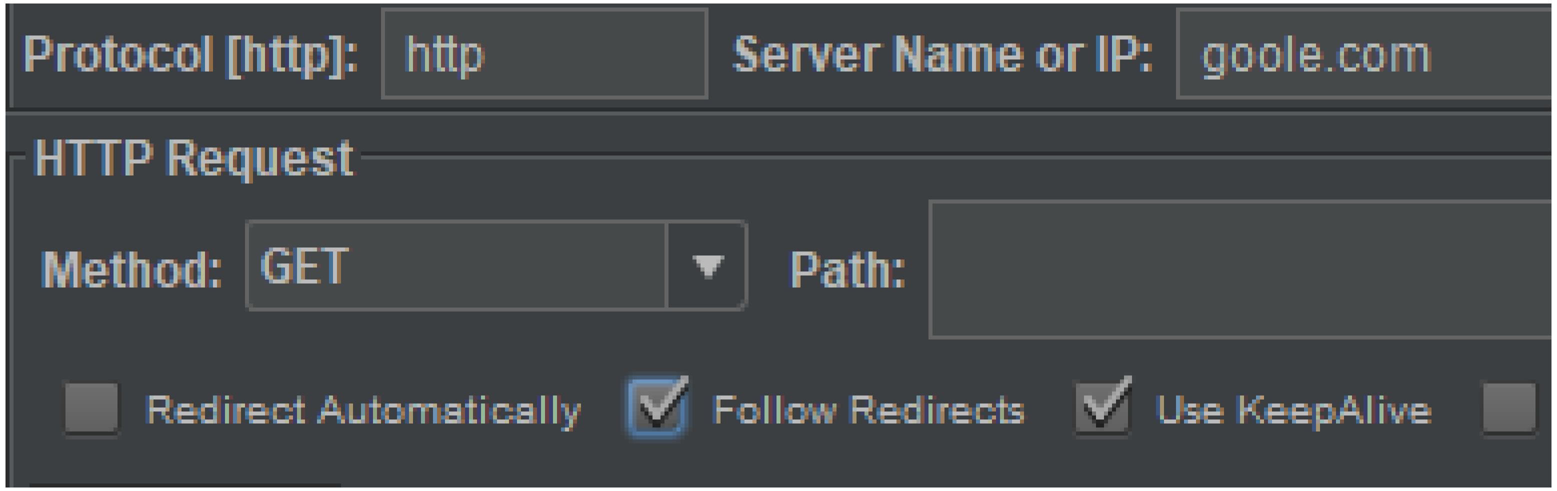
Redirect Automatically: The Redirect Automatically configuration field sets the underlying http protocol handler to automatically follow redirects. This will be helpful when you don’t want to see the requests that are redirected. This should only be used for **GET** and **HEAD** requests, as the HttpClient sampler will reject attempts to use it for **POST** or **PUT**.



Protocol [http]: http
Server Name or IP: goole.com
Method: GET
Path:
[✓] Redirect Automatically
[ ] Follow Redirects
[✓] Use KeepAlive
[ ] Use

HTTP Request

Follow Redirect: On the web, the URLs for the web pages or resources change frequently. The browser is informed of the new URL by the concept of URL redirection. This is also referred to as URL forwarding. HTTP status codes 3XX are used to indicate redirection.



Protocol [http]: http
Server Name or IP: goole.com

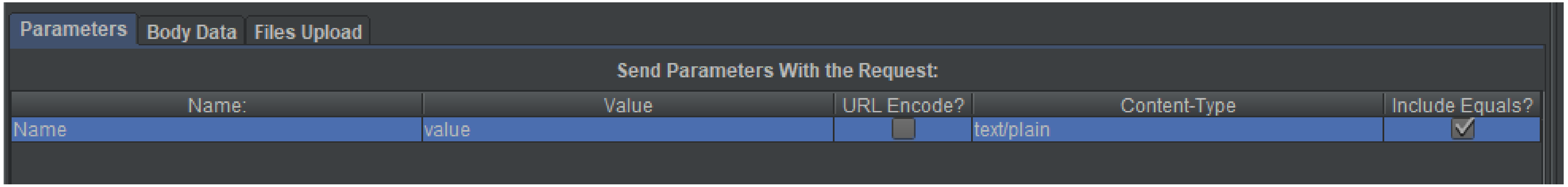
HTTP Request

Method: GET
Path:

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐

HTTP Request

Send Parameter with the Request: With this property, you can specify the request parameters as name/value pairs.
Enabling the Encode: checkbox encodes the special characters in the name/value pair. It is always best to enable this.
Enabling the Include Equals: checkbox will force an '=' character even if the value is empty.

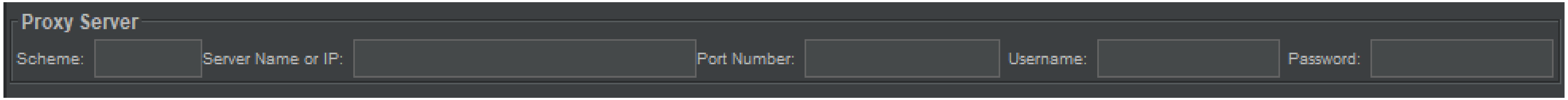


Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
Name	value	☐	text/plain	✓

HTTP Request

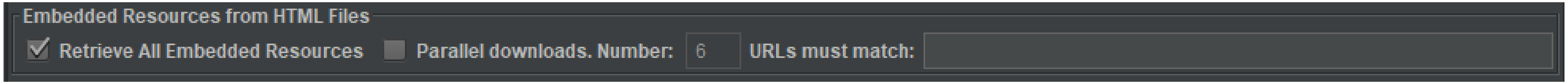
Proxy Server: JMeter provides a simple way to specify the proxy server details for the HTTP Request sampler. This property is only applicable to the current request. You can specify the proxy server name and the port number if needed. In addition, you may also need to specify a username and password.



HTTP Request

Embedded Resources for HTML Files: When you use this property, the JMeter will parse the HTML file and send HTTP/HTTPS requests for all the embedded images, Java applets, JavaScript files, CSSs, etc., referenced in the file

The URLs must match. This must be a regular expression that is used to match against any embedded URLs found. If you only want to download embedded resources from **http://www.yahoo.com/**, you can use the expression: **http://www\.yahoo\.com/.***



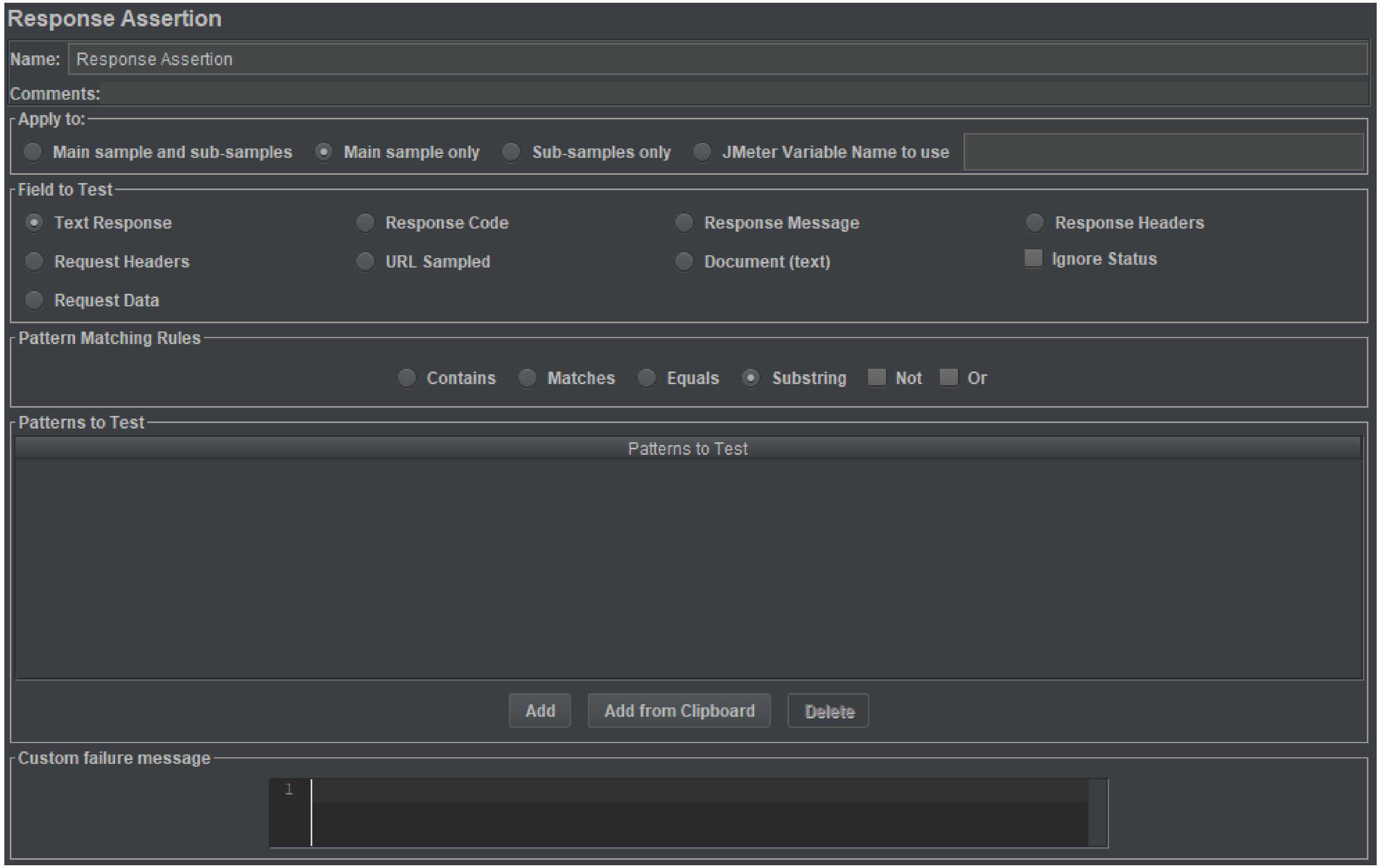
Embedded Resources from HTML Files

[✓] Retrieve All Embedded Resources [ ] Parallel downloads. Number: 6 URLs must match:

FRESHER ACADEMY

HTML Assertion

The primary purpose of assertions is to validate the server response and decide if the test passed or failed.


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

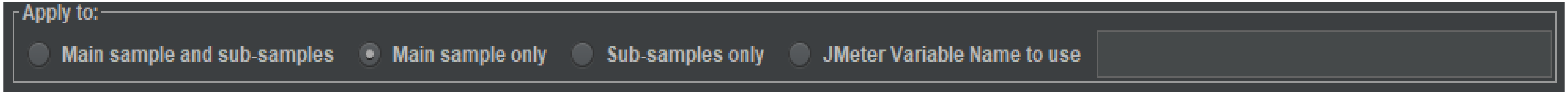
Pattern Matching Rules
Contains Matches Equals [✓ Substring] Not Or

Patterns to Test

Custom failure message

HTML Assertion

Apply to Property : This option specifies where to apply the assertions: Main sample and sub-samples, main sample only, Sub-samples only, and JMeter variable.
Sometimes a sampler may generate a redirected URL, which in turn
will appear as a sub-sample and it is possible to apply assertions to these sub-samples by using Sub-Samples Only option.

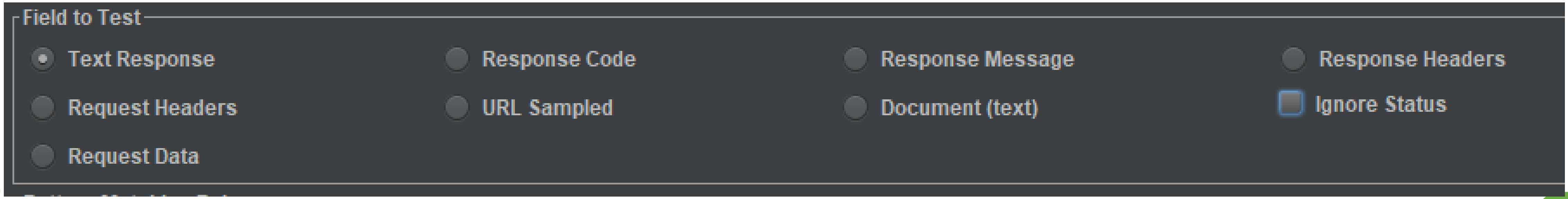

Apply to:
Main sample and sub-samples [✓ Main sample only] Sub-samples only JMeter Variable Name to use

FRESHER ACADEMY

HTML Assertion

Response Field to Test Property: The assertion can apply to any of the following fields in the response:

- Text Response
- Document (Text)
- URL Sampled
- Response Code
- Response Message
- Response Headers
- Ignore Status

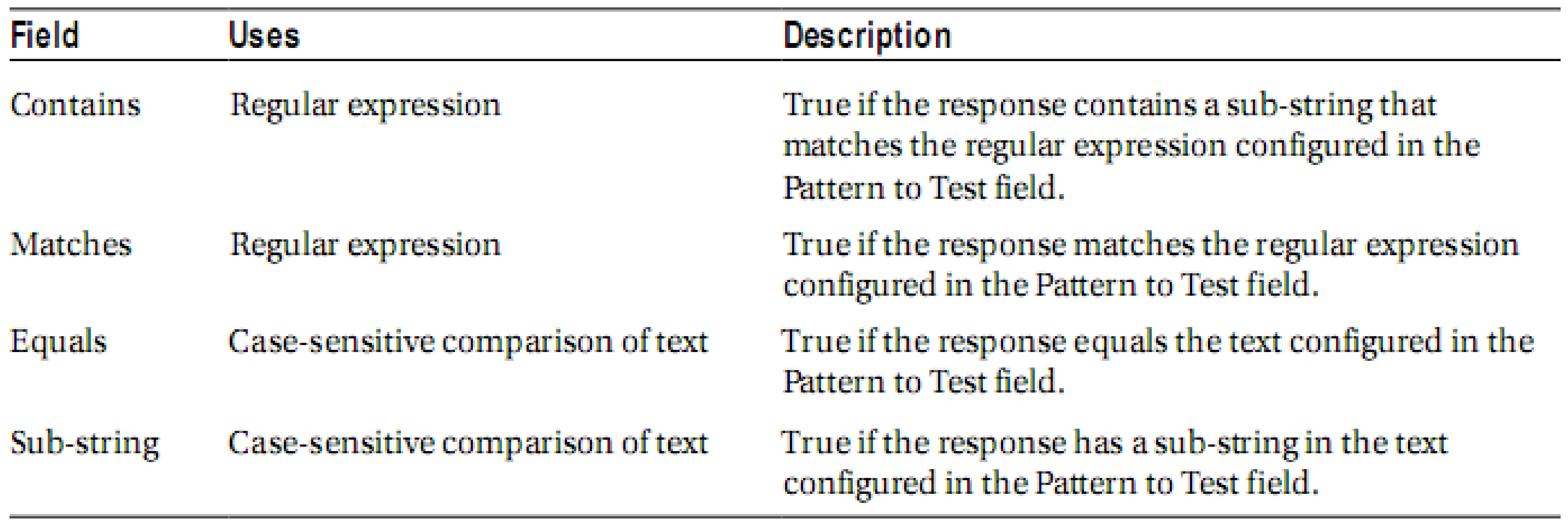


Field to Test

[✓ Text Response]	Response Code	Response Message	Response Headers
Request Headers	URL Sampled	Document (text)	Ignore Status
Request Data			

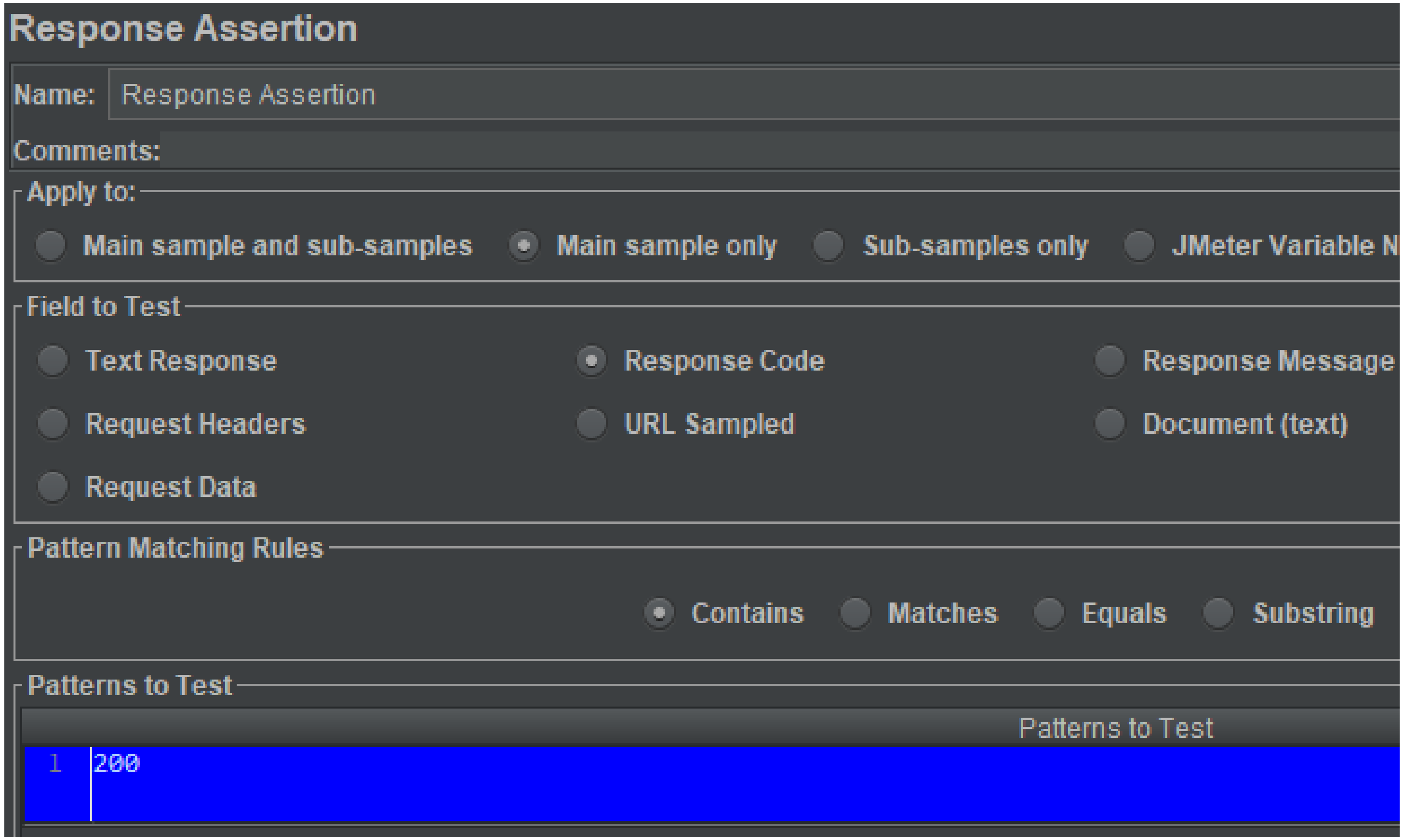
HTML Assertion

Pattern Matching Rules Property:



Field	Uses	Description
Contains	Regular expression	True if the response contains a sub-string that matches the regular expression configured in the Pattern to Test field.
Matches	Regular expression	True if the response matches the regular expression configured in the Pattern to Test field.
Equals	Case-sensitive comparison of text	True if the response equals the text configured in the Pattern to Test field.
Sub-string	Case-sensitive comparison of text	True if the response has a sub-string in the text configured in the Pattern to Test field.

HTML Assertion



Response Assertion

Name: Response Assertion

Comments:

Apply to:

( ) Main sample and sub-samples [✓ Main sample only] ( ) Sub-samples only ( ) JMeter Variable N

Field to Test

( ) Text Response [✓ Response Code] ( ) Response Message
( ) Request Headers ( ) URL Sampled ( ) Document (text)
( ) Request Data

Pattern Matching Rules

[✓ Contains] ( ) Matches ( ) Equals ( ) Substring

Patterns to Test

Patterns to Test
200

Listeners

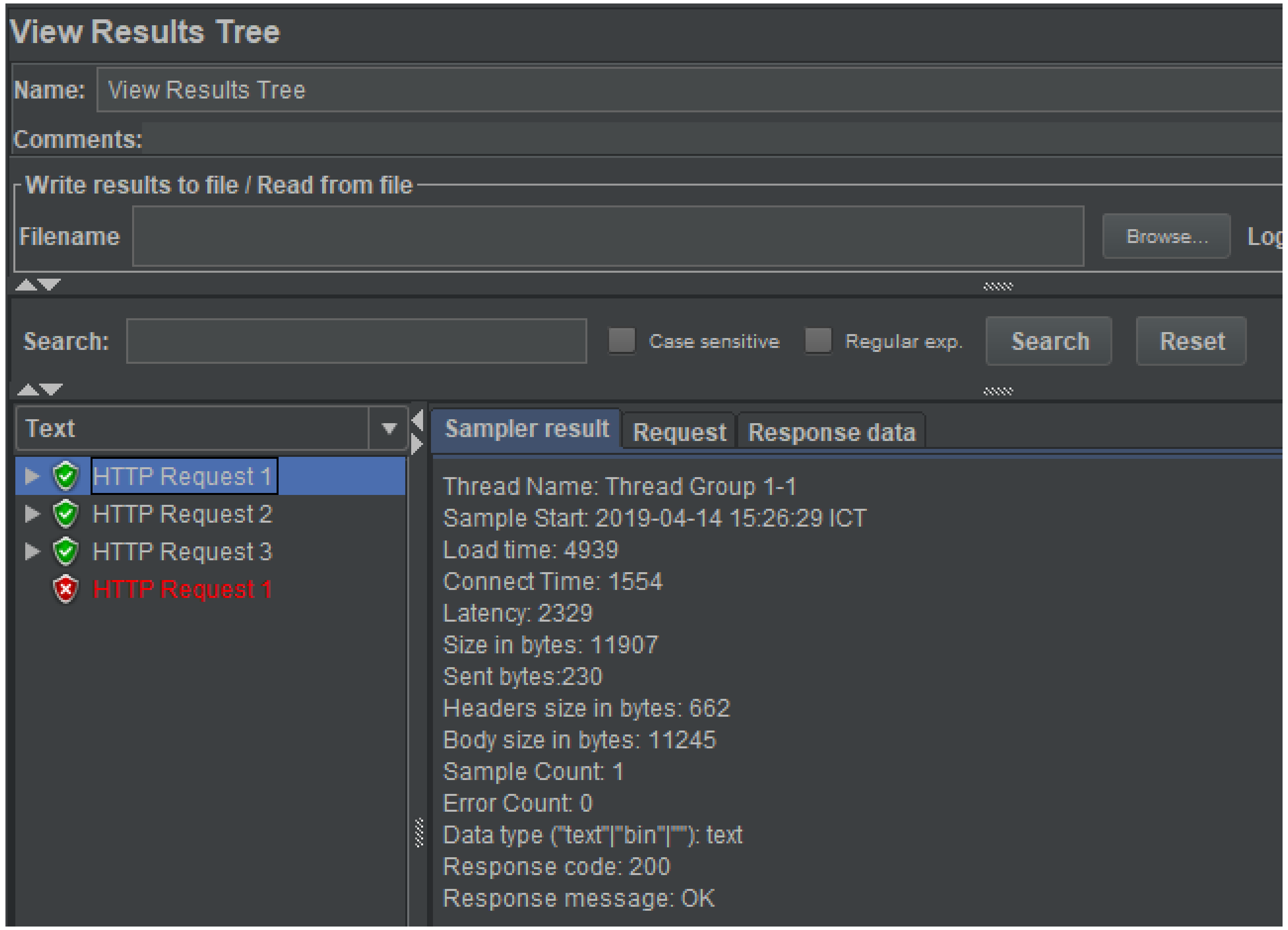
Listeners capture and process the response from the server. Performance testing requires two kinds of listeners. During the test script development, you need to capture and display the entire server response for verifying that the output meets the functional specifications. When the performance test scripts are executed, you need the aggregate results and metrics for the duration of the test execution. You also need listeners to be able to store these results into an external file for later use.

View Results Tree **View Results In Table** **Aggregate Report**

FRESHER ACADEMY

Listeners - View Results Tree

The View Results Tree listener shows responses in a tree-like structure, thereby allowing users to see response data (content), sample results, and requests. It also shows the time taken by the request to get the response.

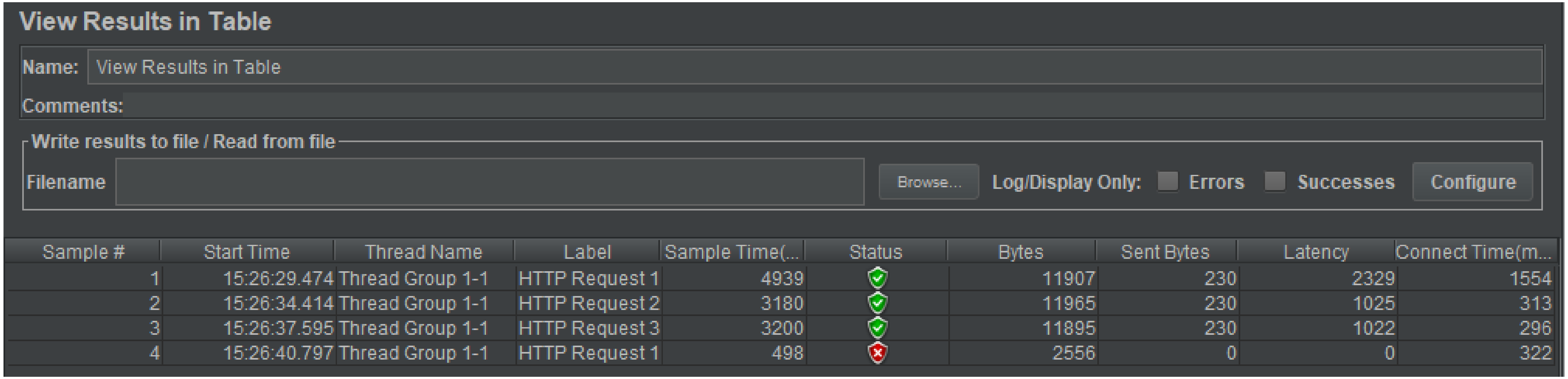


```
Thread Name: Thread Group 1-1
Sample Start: 2019-04-14 15:26:29 ICT
Load time: 4939
Connect Time: 1554
Latency: 2329
Size in bytes: 11907
Sent bytes:230
Headers size in bytes: 662
Body size in bytes: 11245
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK
```

FRESHER ACADEMY

Listeners - View Results In Table

The View Results in Table listener shows responses in a table-like structure. It also shows the time taken by the request/thread to get the response. Unlike **View Results Tree**, it does not have a panel and does not show any headers or response contents.



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(...	Status	Bytes	Sent Bytes	Latency	Connect Time(m...
1	15:26:29.474	Thread Group 1-1	HTTP Request 1	4939	✓	11907	230	2329	1554
2	15:26:34.414	Thread Group 1-1	HTTP Request 2	3180	✓	11965	230	1025	313
3	15:26:37.595	Thread Group 1-1	HTTP Request 3	3200	✓	11895	230	1022	296
4	15:26:40.797	Thread Group 1-1	HTTP Request 1	498	✗	2556	0	0	322

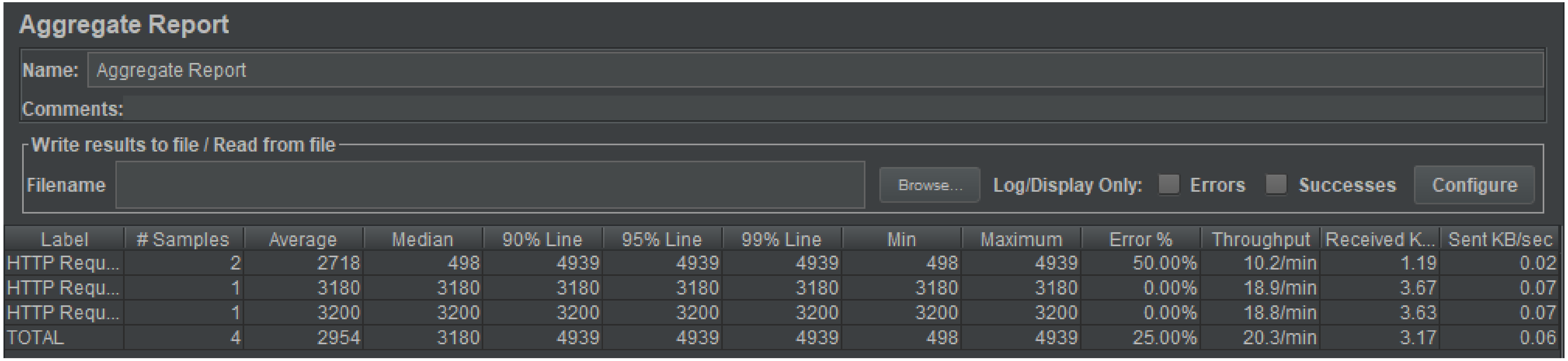
Listeners - View Results In Table

It has the following columns in the view:

- **Start Time**: When this thread started.
- **Thread Name**: Thread group name with the thread number in the format X-X.
- **Label**: Name of HTTP request (sample)
- **Sample Time (ms):** Difference between time when request was sent and time when response has been fully received.
- **Status**: It has two statuses: 1. **Success**, 2. **Warning**. If the thread sample is not valid, it shows Warning; otherwise, it shows Success
- **Bytes**: Total number of bytes received as a part of the response.
- **Latency**: JMeter measures the latency from just before sending the request to just after the first response is received
- **Connect Time(ms):** JMeter measures the time it takes to establish the connection, including the SSL handshake.

Listeners - Aggregate Report

The Aggregate Report listener also shows the responses in a table-like structure for each differently named sampler. Similar to View Results in Table, it does not have a panel and does not show any headers or response contents. Generated results can be saved as .csv files with the use of the Save Table Data button at the bottom of this listener.



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 K...	Sent KB/sec
HTTP Requ...	2	2718	498	4939	4939	4939	498	4939	50.00%	10.2/min	1.19	0.02
HTTP Requ...	1	3180	3180	3180	3180	3180	3180	3180	0.00%	18.9/min	3.67	0.07
HTTP Requ...	1	3200	3200	3200	3200	3200	3200	3200	0.00%	18.8/min	3.63	0.07
TOTAL	4	2954	3180	4939	4939	4939	498	4939	25.00%	20.3/min	3.17	0.06

Trung bình
là 2s

Listeners - Aggregate Report

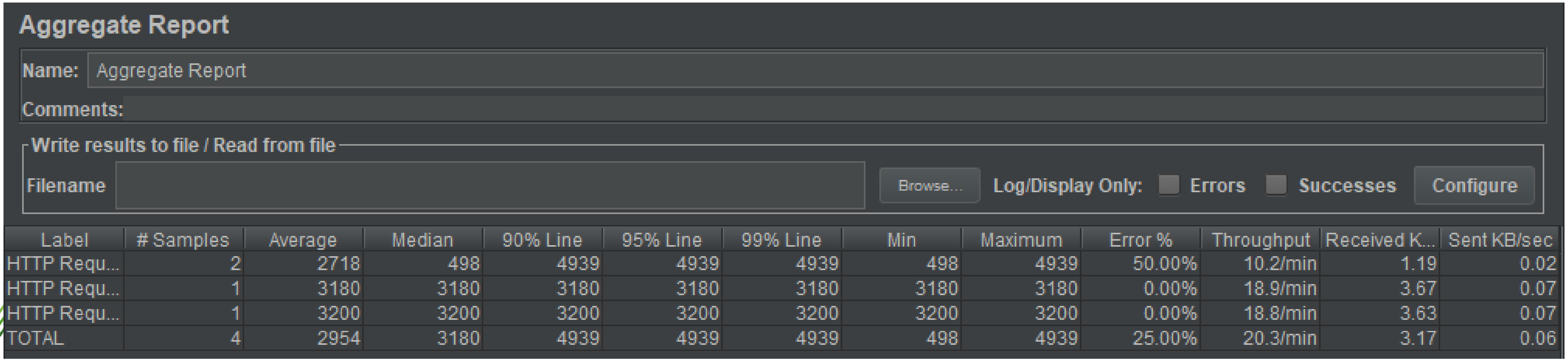
For each sampler, it shows the following columns in the view:

- **Label**: Name given to the sampler. If the Include Group Name in Label? checkbox is checked, then the thread group name is prefixed with the sampler name.
- **Samples**: Number of samples for a sampler. If there is more than one sampler with the same name, it will be combined to a single row and the combined sum is shown.
- **Average**: This is the average time in milliseconds for a set of results.
- **Median**: This is the median time in milliseconds.
- **90% Line**: This is the time in milliseconds below which 90% of the samples fall. The other samples took at least as long as this
- **95% Line**: This is the time in milliseconds below which 95% of the samples fall. The other samples took at least as long as this.
- **99% Line**: This is the time in milliseconds below which 99% of the samples fall. The other samples took at least as long as this.

Listeners - Aggregate Report

For each sampler, it shows the following columns in the view:

- **Min**: This the shortest time in milliseconds for the sampler with the same name.
- **Max**: This the maximum time in milliseconds for the sampler with the same name.
- **Error %**: Percent of requests with errors; it may be in fractions
- **Throughput**: Calculated as requests/unit of time. It is the time difference between the end of the last sample and the start of the first sample, including any gaps between samples. It is expressed as number of requests/total time.
- **KB/sec**: The response received in kilobytes per second



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

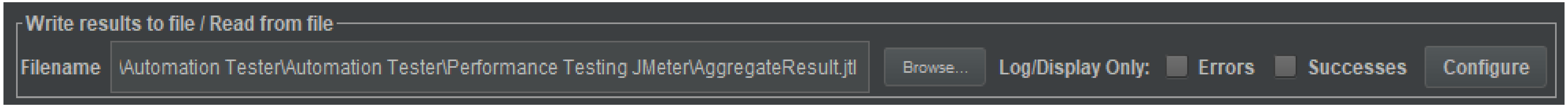
Filename

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
HTTP Requ...	2	2718	498	4939	4939	4939	498	4939	50.00%	10.2/min	1.19	0.02
HTTP Requ...	1	3180	3180	3180	3180	3180	3180	3180	0.00%	18.9/min	3.67	0.07
HTTP Requ...	1	3200	3200	3200	3200	3200	3200	3200	0.00%	18.8/min	3.63	0.07
TOTAL	4	2954	3180	4939	4939	4939	498	4939	25.00%	20.3/min	3.17	0.06

Listeners - Aggregate Report

Open Aggregate Report and find the section Write Results to File/Read from File. Click on the Browser button and locate the file (generated using the command).

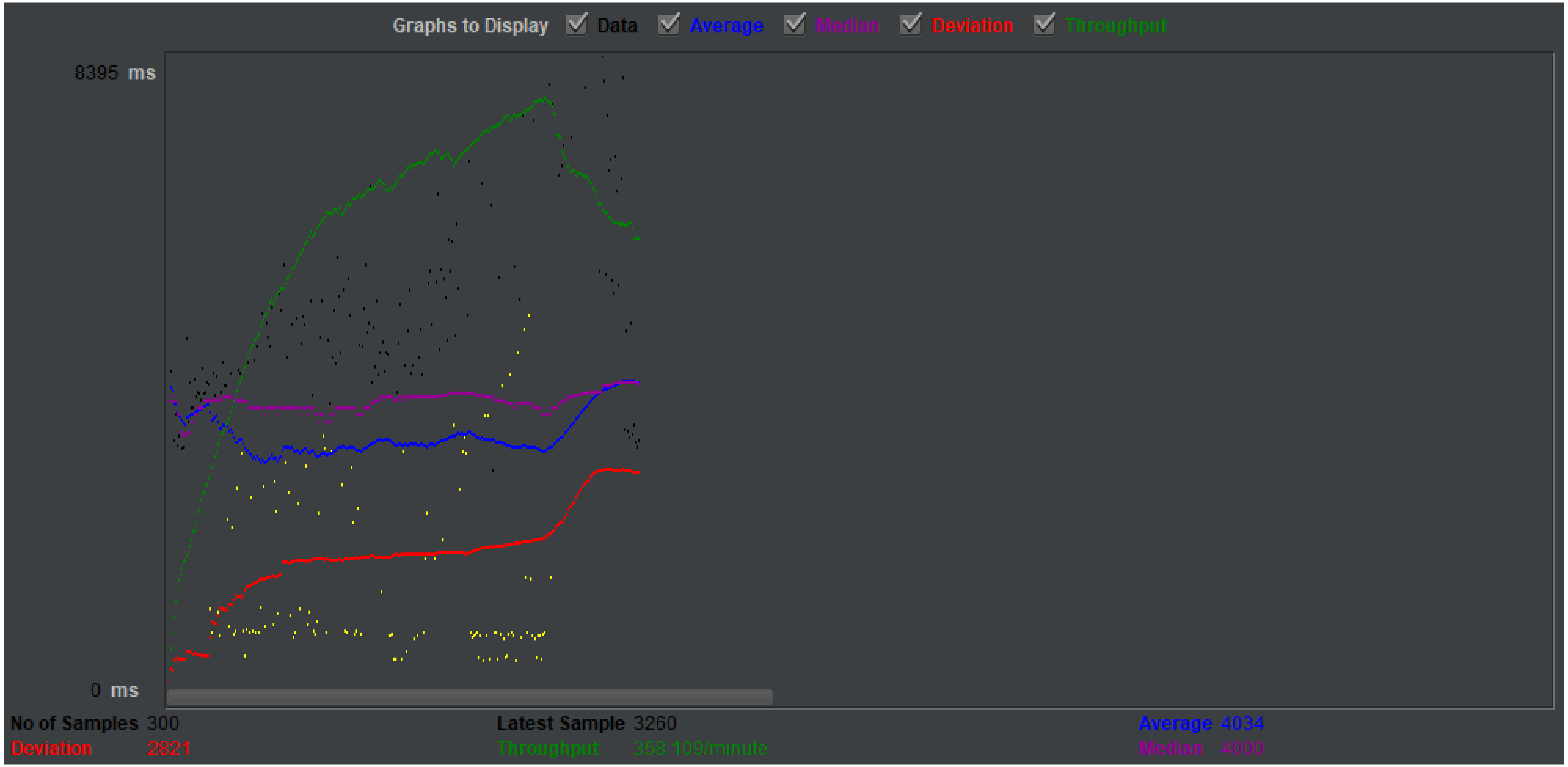
C:\Users\ADMIN\Documents\Automation Tester\Automation Tester\Performance Testing JMeter\AggregateResult.jtl



Write results to file / Read from file

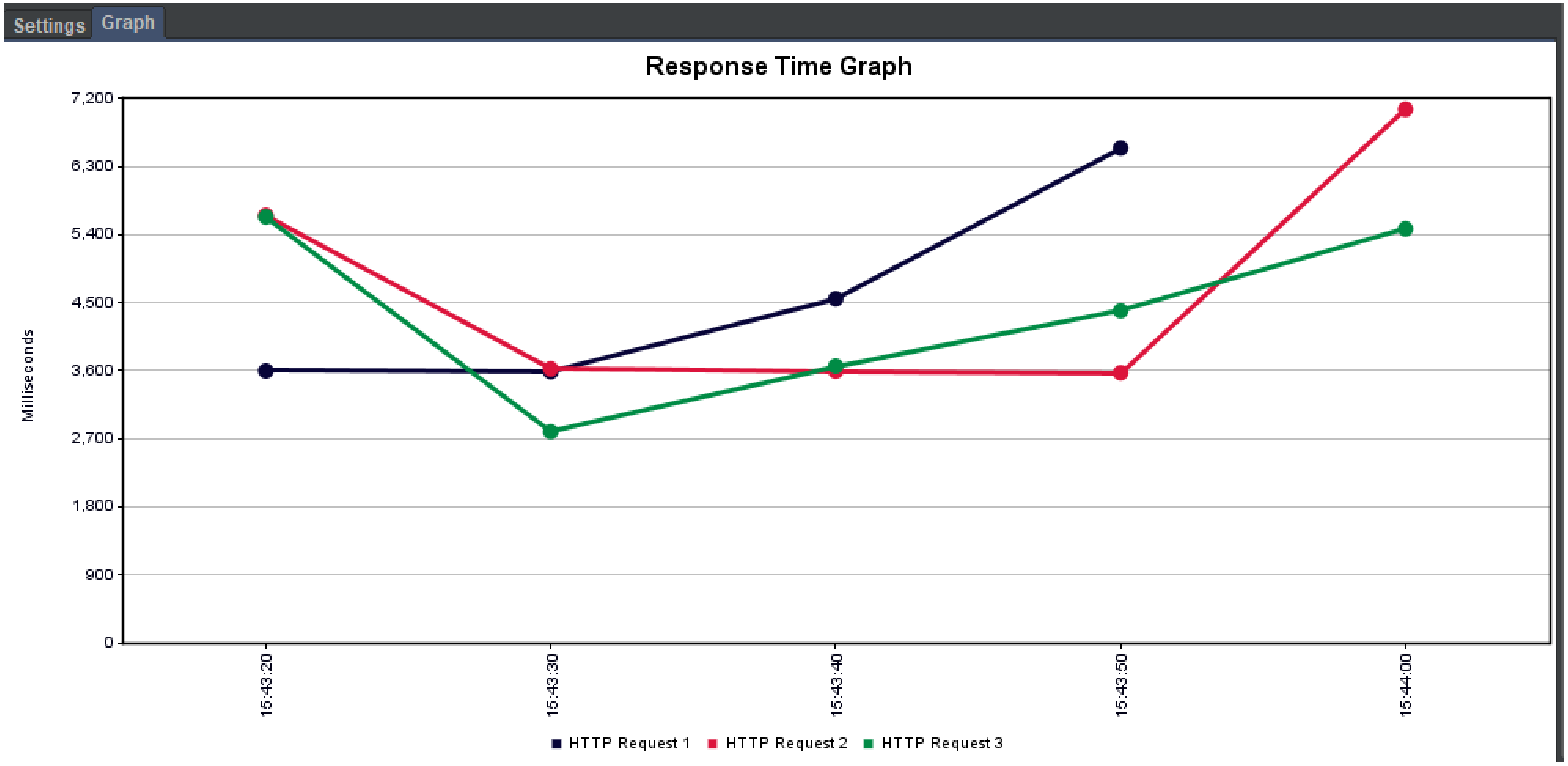
Filename: \Automation Tester\Automation Tester\Performance Testing JMeter\AggregateResult.jtl

Listeners - Graph Results

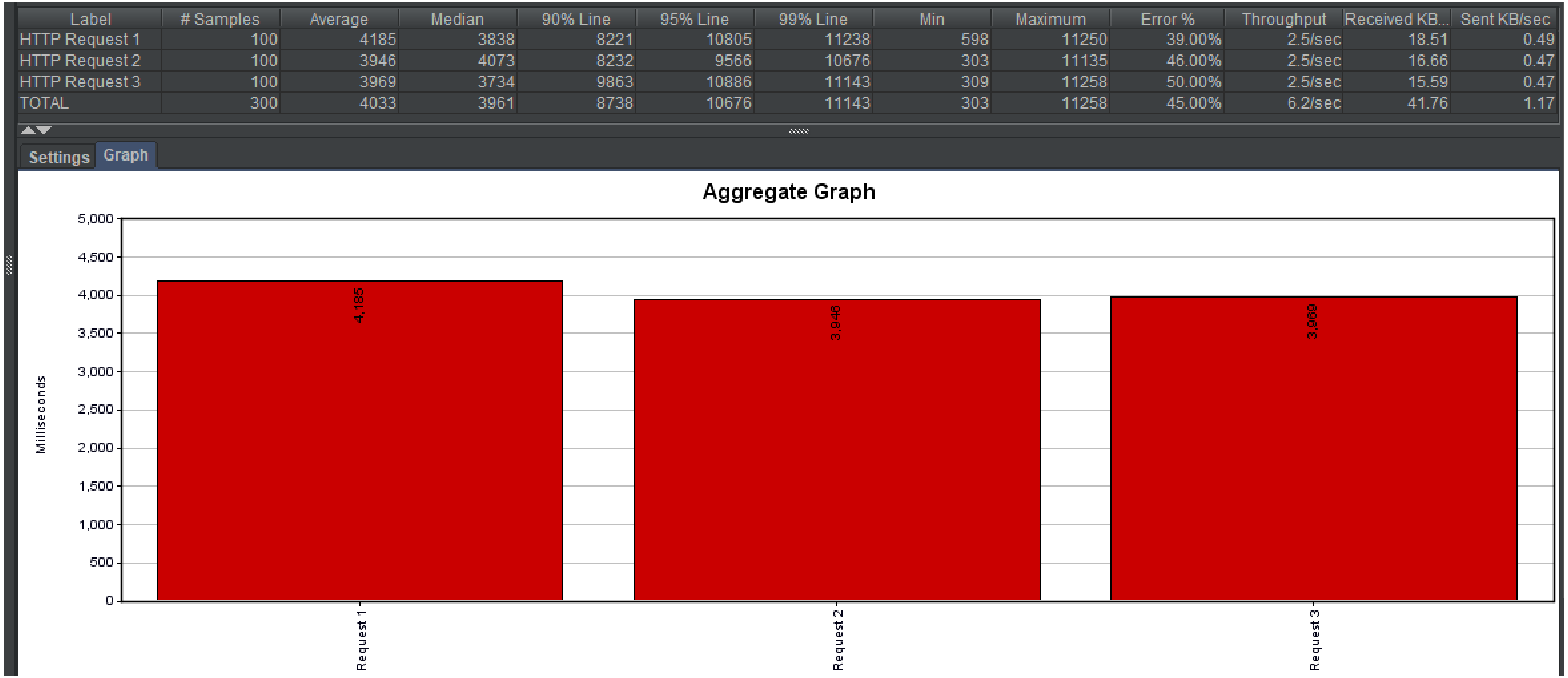


No of Samples 300
Latest Sample 3260
Average 4034
Deviation 2821
Throughput 358.109/minute
Median 4000

Listeners - Response Time Graph



Listeners - Aggregate Graph

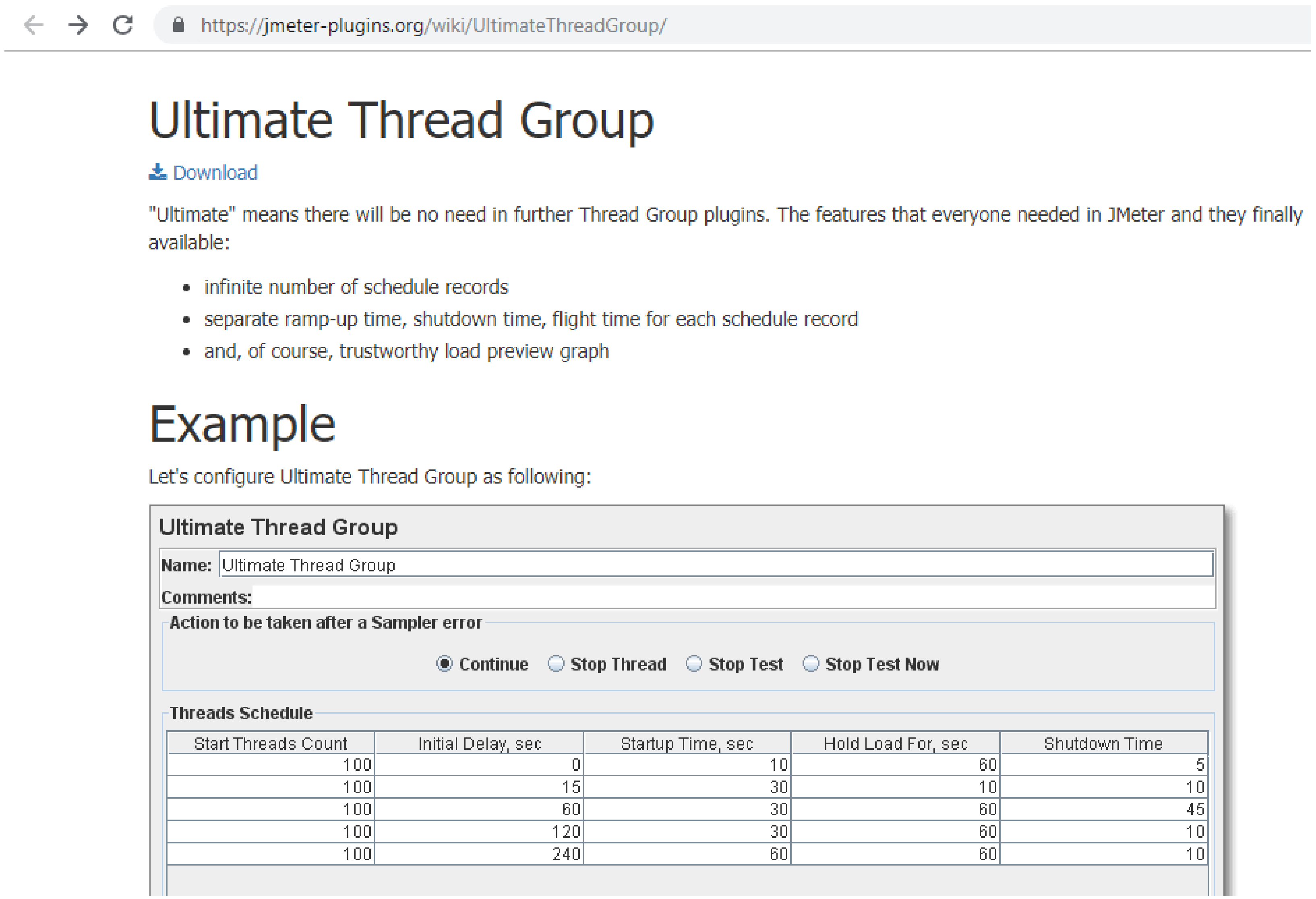


Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB...	Sent KB/sec
HTTP Request 1	100	4185	3838	8221	10805	11238	598	11250	39.00%	2.5/sec	18.51	0.49
HTTP Request 2	100	3946	4073	8232	9566	10676	303	11135	46.00%	2.5/sec	16.66	0.47
HTTP Request 3	100	3969	3734	9863	10886	11143	309	11258	50.00%	2.5/sec	15.59	0.47
TOTAL	300	4033	3961	8738	10676	11143	303	11258	45.00%	6.2/sec	41.76	1.17

Ultimate Thread Group

- We have 3 Thread count as **10, 20 & 30**.
- Initial delay for all 3 Threads is kept same as **10 seconds**.
- So after initiation of execution, Jmeter will wait for **10 seconds to send the 1st thread to the server**.
- Startup Time is also same as **10 seconds**. Hence, within **10 seconds all 60 Users** will be sent to the server.
- Now Hold Load Time is different for each row: **50, 100 & 150 seconds**.

Ultimate Thread Group



https://jmeter-plugins.org/wiki/UltimateThreadGroup/

Ultimate Thread Group

Download

"Ultimate" means there will be no need in further Thread Group plugins. The features that everyone needed in JMeter and they finally available:

- infinite number of schedule records
- separate ramp-up time, shutdown time, flight time for each schedule record
- and, of course, trustworthy load preview graph

Example

Let's configure Ultimate Thread Group as following:

Ultimate Thread Group

Name: Ultimate Thread Group

Comments:

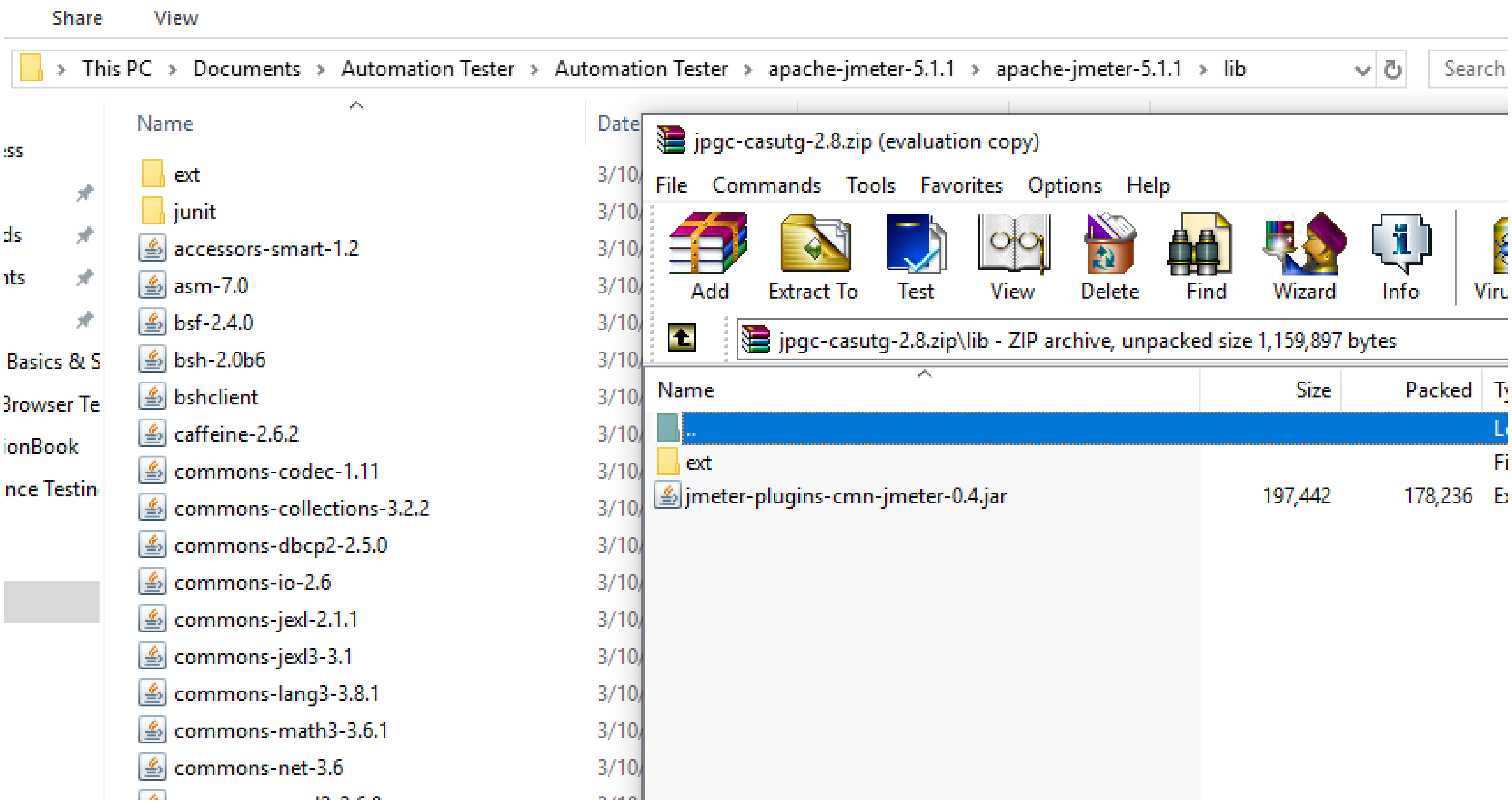
Action to be taken after a Sampler error

[✓ Continue] Stop Thread Stop Test Stop Test Now

Threads Schedule

Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown Time
100	0	10	60	5
100	15	30	10	10
100	60	30	60	45
100	120	30	60	10
100	240	60	60	10

Ultimate Thread Group



Share View

This PC › Documents › Automation Tester › Automation Tester › apache-jmeter-5.1.1 › apache-jmeter-5.1.1 › lib

Name	Date
ext	3/10
junit	3/10
accessors-smart-1.2	3/10
asm-7.0	3/10
bsf-2.4.0	3/10
bsh-2.0b6	3/10
bshclient	3/10
caffeine-2.6.2	3/10
commons-codec-1.11	3/10
commons-collections-3.2.2	3/10
commons-dbcp2-2.5.0	3/10
commons-io-2.6	3/10
commons-jexl-2.1.1	3/10
commons-jexl3-3.1	3/10
commons-lang3-3.8.1	3/10
commons-math3-3.6.1	3/10
commons-net-3.6	3/10

jpgc-casutg-2.8.zip (evaluation copy)

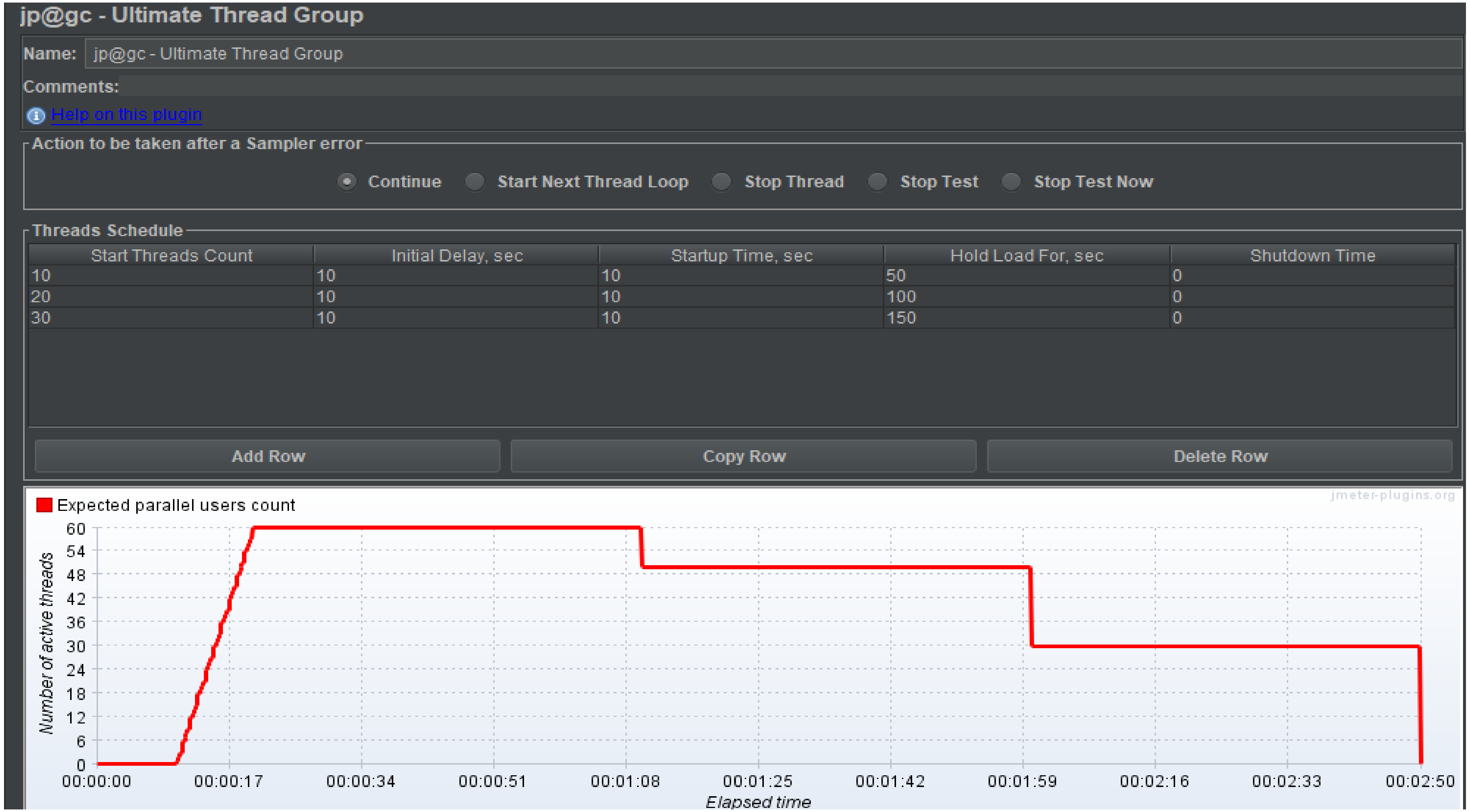
File Commands Tools Favorites Options Help

Add Extract To Test View Delete Find Wizard Info Viru

jpgc-casutg-2.8.zip\lib - ZIP archive, unpacked size 1,159,897 bytes

Name	Size	Packed	Ty
..			L
ext			Fi
jmeter-plugins-cmn-jmeter-0.4.jar	197,442	178,236	E

Ultimate Thread Group



jp@gc - Ultimate Thread Group

Name: jp@gc - Ultimate Thread Group

Comments:

Help on this plugin

Action to be taken after a Sampler error

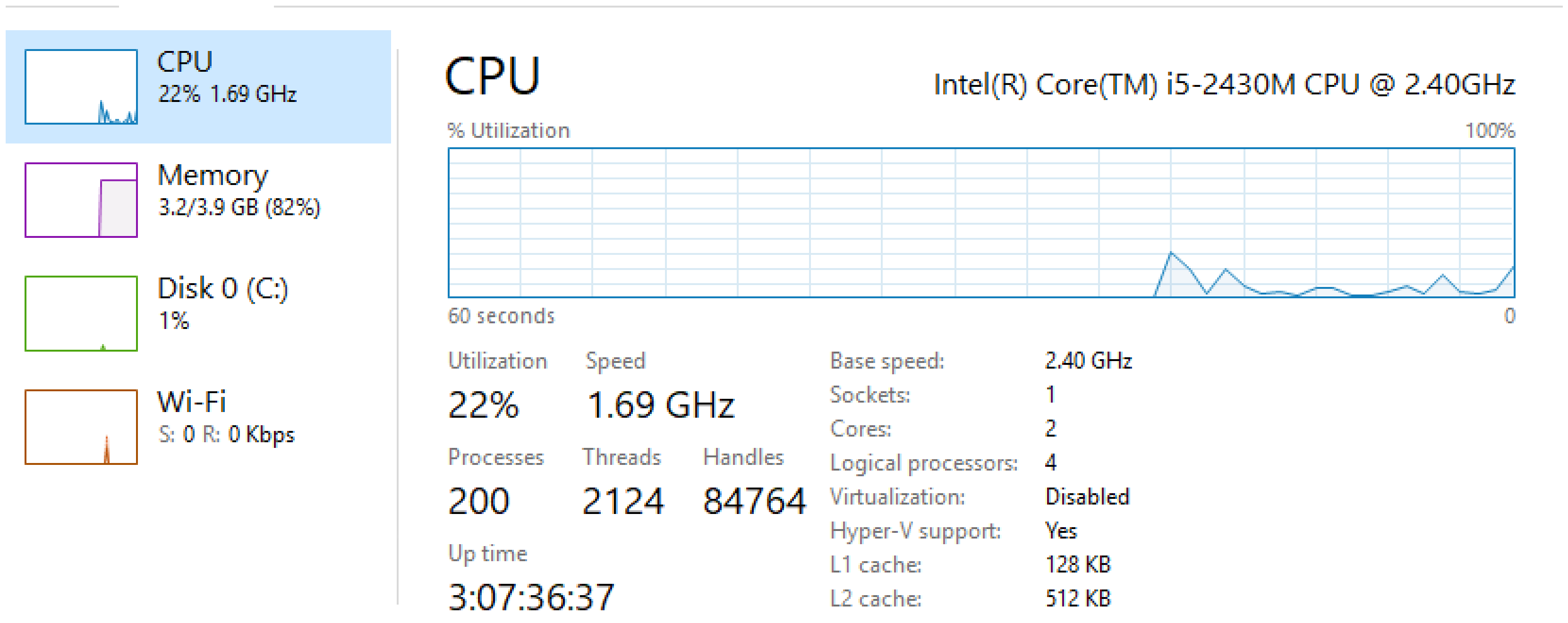
[✓ Continue] Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Threads Schedule

Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown Time
10	10	10	50	0
20	10	10	100	0
30	10	10	150	0

Servers Performance Monitoring

During a load test, it is important to know the health of the servers loaded. It is also nice to see if you are targeting a cluster if the load is correctly dispatched. To address this, the plugin package now supports server monitoring! Using it, you can monitor CPU, Memory, Swap, Disks I/O and Networks I/O on almost all platforms!

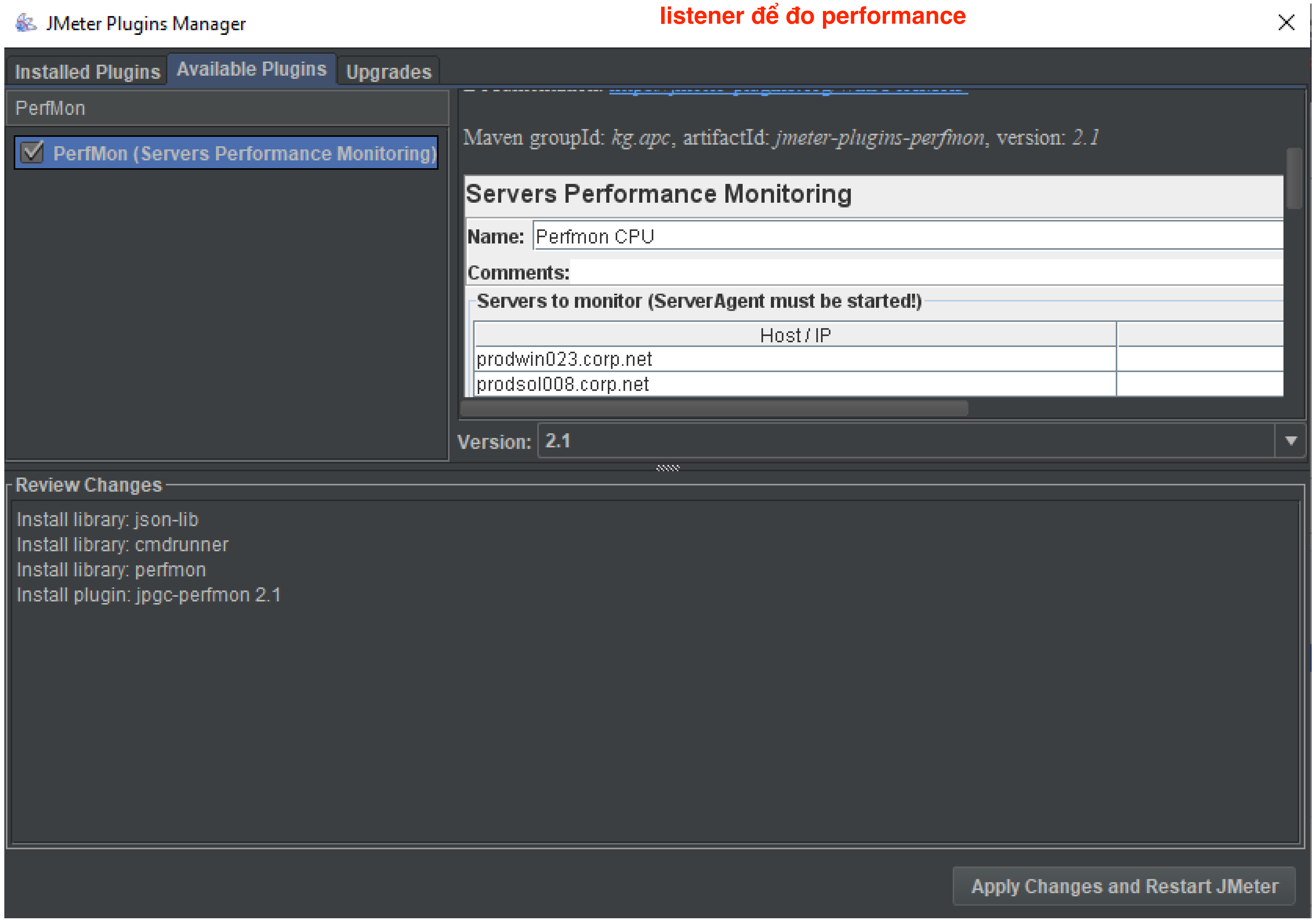


Utilization	Speed		
22%	1.69 GHz		
Processes	Threads	Handles	
200	2124	84764	
Up time			
3:07:36:37			

Base speed:	2.40 GHz
Sockets:	1
Cores:	2
Logical processors:	4
Virtualization:	Disabled
Hyper-V support:	Yes
L1 cache:	128 KB
L2 cache:	512 KB

Servers Performance Monitoring

Server agent là thư viện để mở kết nối, cho phép listener để đo performance



JMeter Plugins Manager

☑ PerfMon (Servers Performance Monitoring)

Maven groupId: *kg.apc*, artifactId: *jmeter-plugins-perfmon*, version: *2.1*

Servers Performance Monitoring

Name: Perfmon CPU

Comments:

Servers to monitor (ServerAgent must be started!)

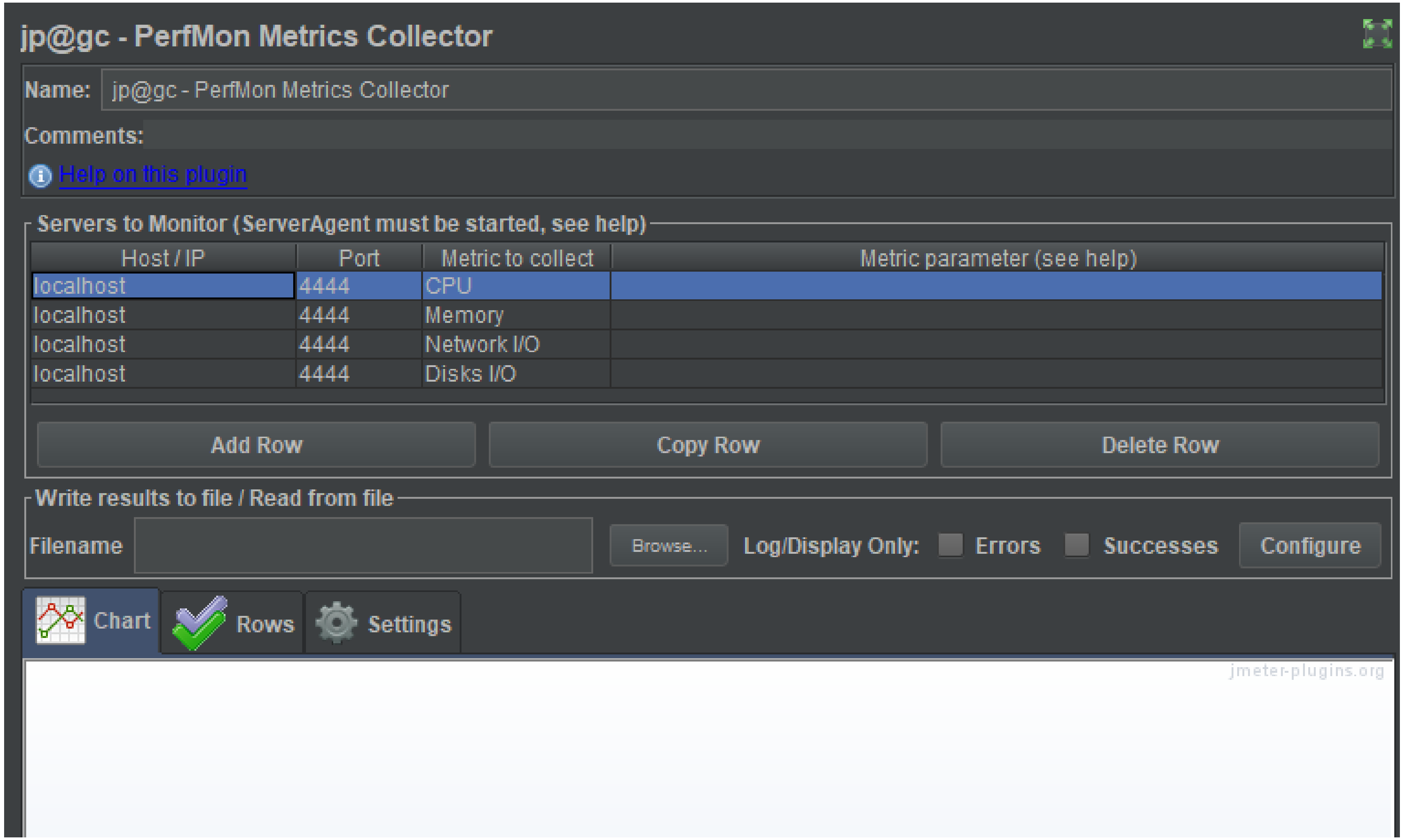
Host / IP	
prodwin023.corp.net	
prodsol008.corp.net	

Version: 2.1

Review Changes

Install library: json-lib
Install library: cmdrunner
Install library: perfmon
Install plugin: jpgc-perfmon 2.1

Servers Performance Monitoring



jp@gc - PerfMon Metrics Collector

Name: jp@gc - PerfMon Metrics Collector

Comments:

Help on this plugin

Servers to Monitor (ServerAgent must be started, see help)

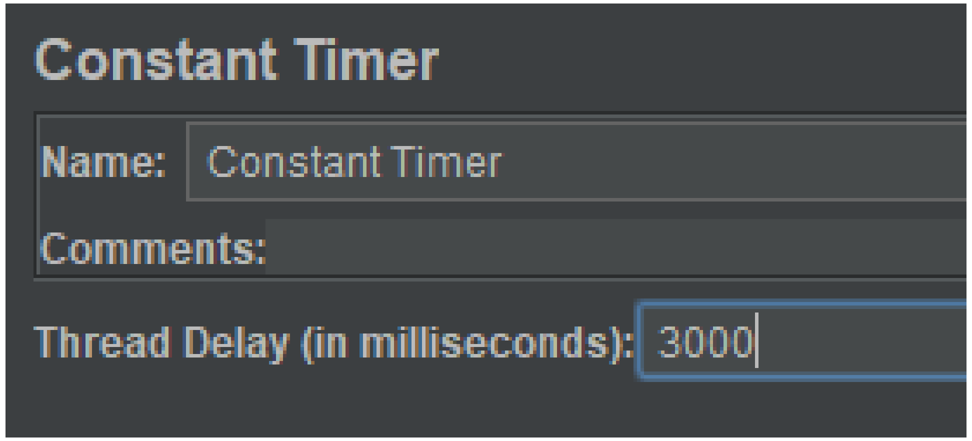
Host / IP	Port	Metric to collect	Metric parameter (see help)
localhost	4444	CPU	
localhost	4444	Memory	
localhost	4444	Network I/O	
localhost	4444	Disks I/O	

Write results to file / Read from file

Filename

Constant Timer

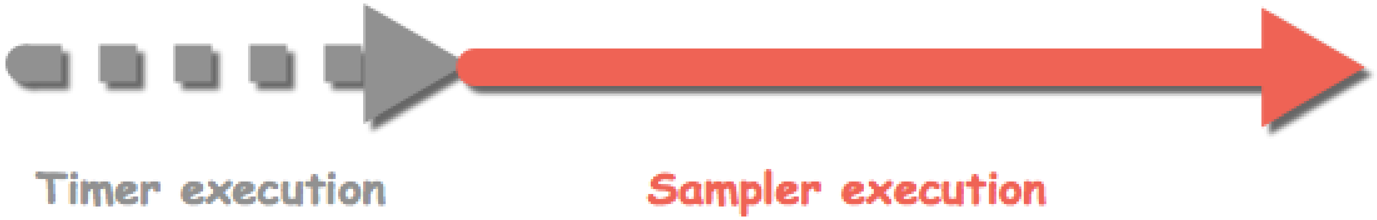
A Timer Component is an option in building a Test Plan. It causes JMeter to pause for a certain amount of time between two successive requests that a Thread Group makes.


Constant Timer

Name: Constant Timer

Comments:

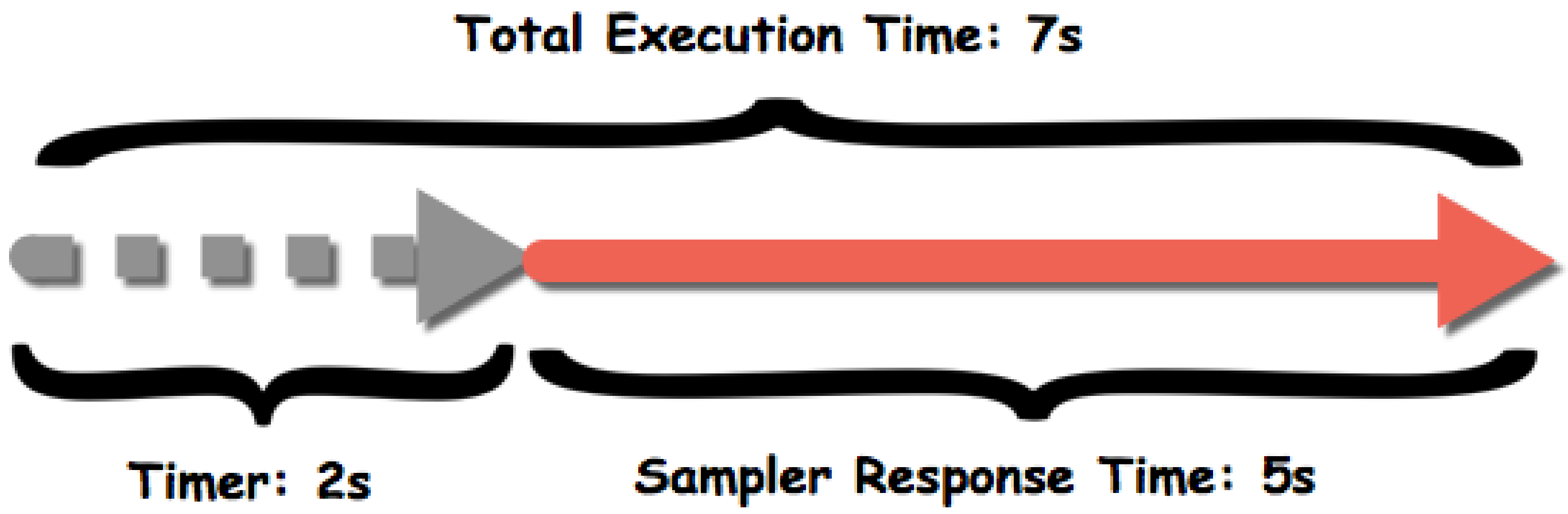
Thread Delay (in milliseconds): 3000



FRESHER ACADEMY

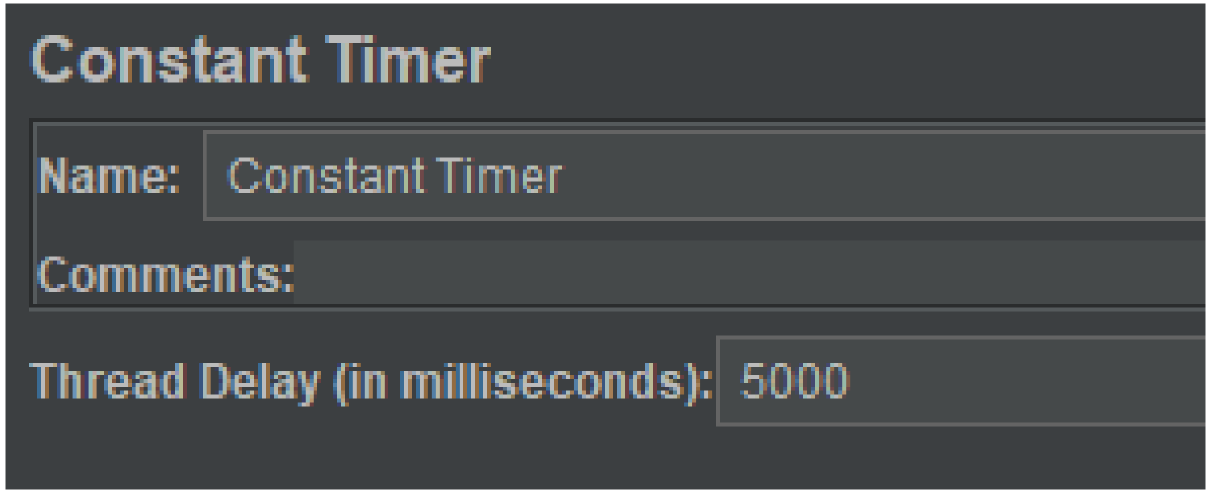
Constant Timer

A Timer Component is an option in building a Test Plan. It causes JMeter to pause for a certain amount of time between two successive requests that a Thread Group makes.



Constant Timer

The Constant Timer introduces a specified delay before the samplers in its scope are executed. The only configuration is the delay that is needed.


Constant Timer

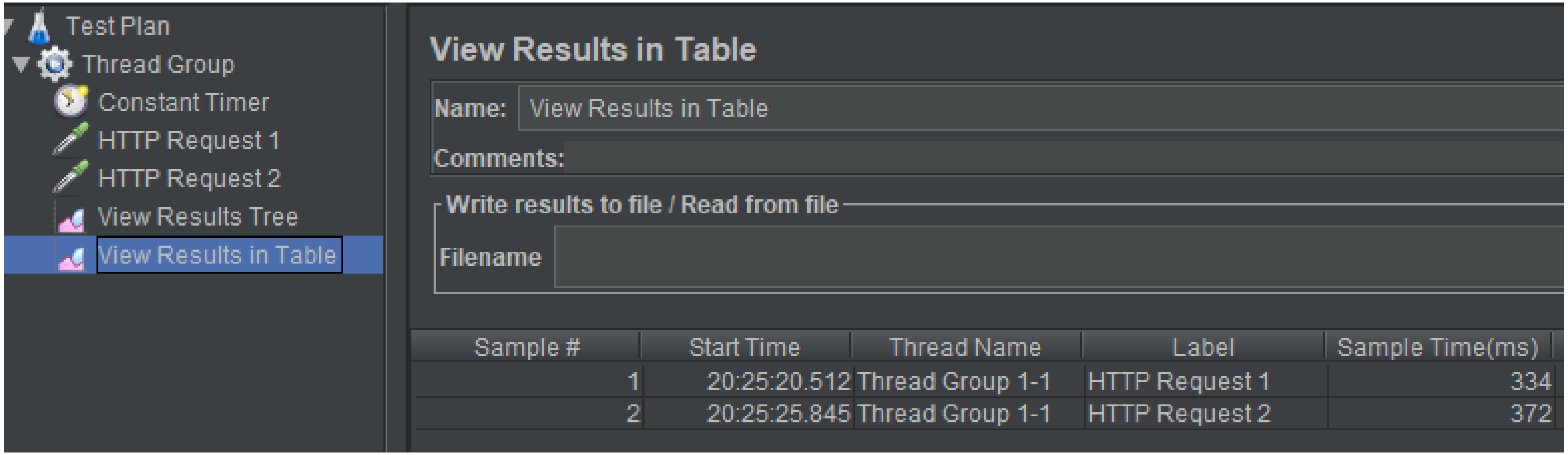
Name: Constant Timer

Comments:

Thread Delay (in milliseconds): 5000

FRESHER ACADEMY

Constant Timer



View Results in Table

Name: View Results in Table

Comments:

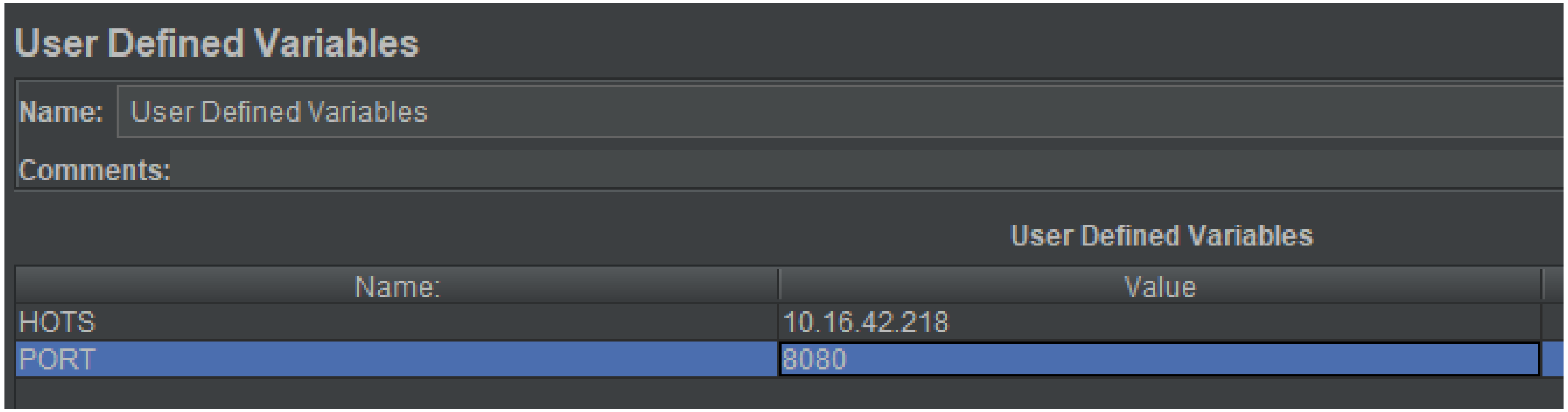
Write results to file / Read from file

Filename

Sample #	Start Time	Thread Name	Label	Sample Time(ms)
1	20:25:20.512	Thread Group 1-1	HTTP Request 1	334
2	20:25:25.845	Thread Group 1-1	HTTP Request 2	372

User Defined Variables

The User Defined Variables element lets you define an **initial set of variables**, just as in the Test Plan.



User Defined Variables

Name: User Defined Variables

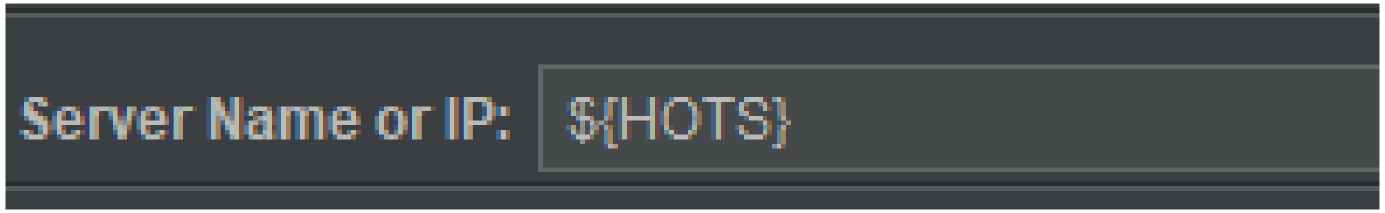
Comments:

User Defined Variables

Name:	Value
HOTS	10.16.42.218
PORT	8080

User Defined Variables

Now let see what inside this element and how to use it:

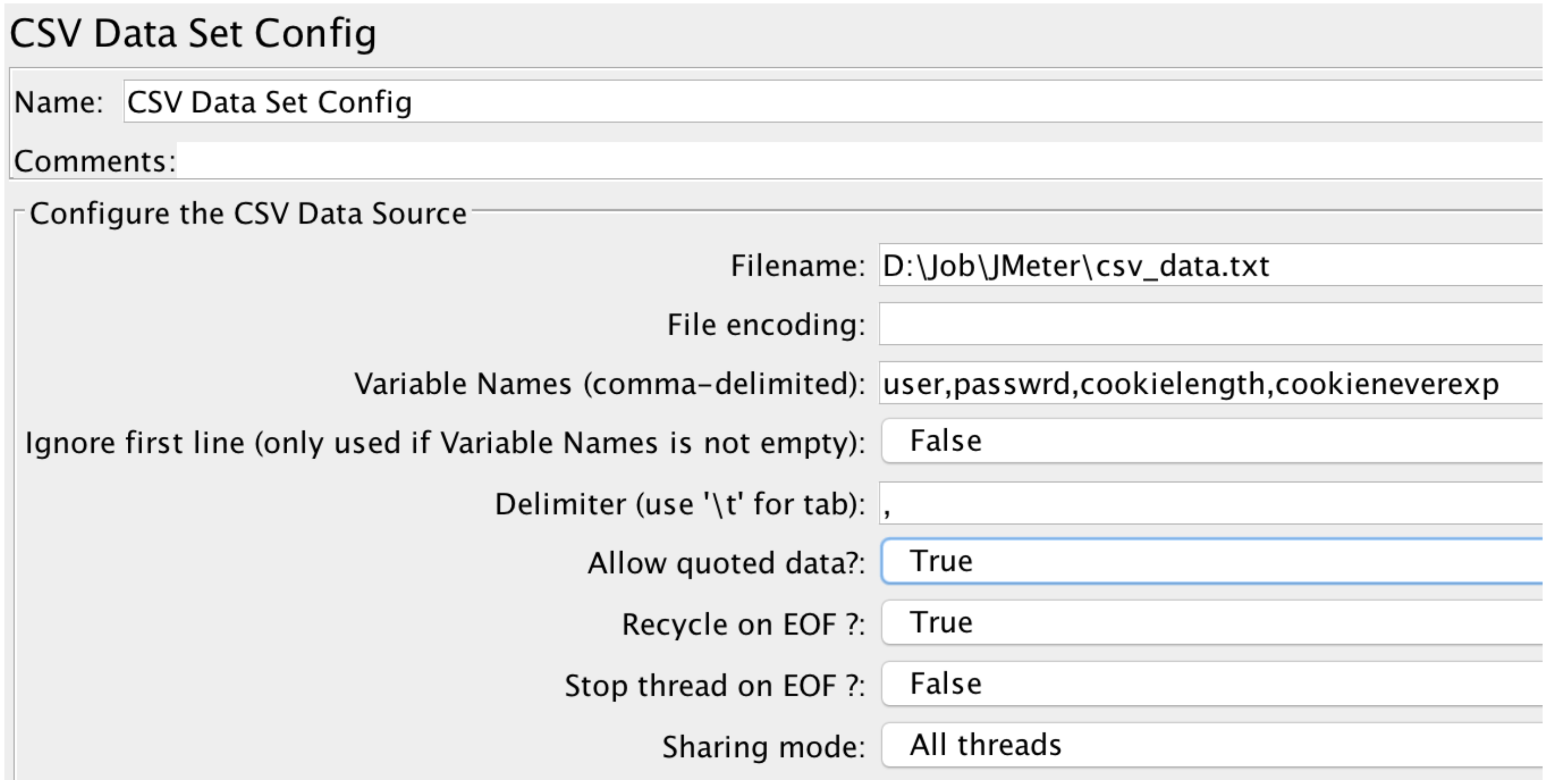


Server Name or IP: ${HOTS}

It's the value corresponding to the variable in the **Name** column. The whole **${VARIABLE_NAME}** will then be replaced by the string in the "**Value**" column.

CSV data set config

Jmeter an open source load testing tool, has an element that allows you to use external data sets in a CSV format. This element is called the “CSV Data Set Config”. The CSV Data Set Config is used to read lines from a file and to split them into variables.



CSV Data Set Config

Name:	CSV Data Set Config
Comments:	

Configure the CSV Data Source

Filename:	D:\Job\JMeter\csv_data.txt
File encoding:	
Variable Names (comma-delimited):	user,passwrd,cookielength,cookieneverexp
Ignore first line (only used if Variable Names is not empty):	False
Delimiter (use '\t' for tab):	,
Allow quoted data?:	True
Recycle on EOF ?:	True
Stop thread on EOF ?:	False
Sharing mode:	All threads

FRESHER ACADEMY

CSV data set config

In the request, you can see the table with the variables. There are four variables, the same quantity as in **'CSV Data Set Config'**. As you can notice, the value field has the same name as variable in the 'CSV Data Set'. The construction **${….}** means that this is a variable and not an absolute value.

${passwd}, ${user}, etc.,

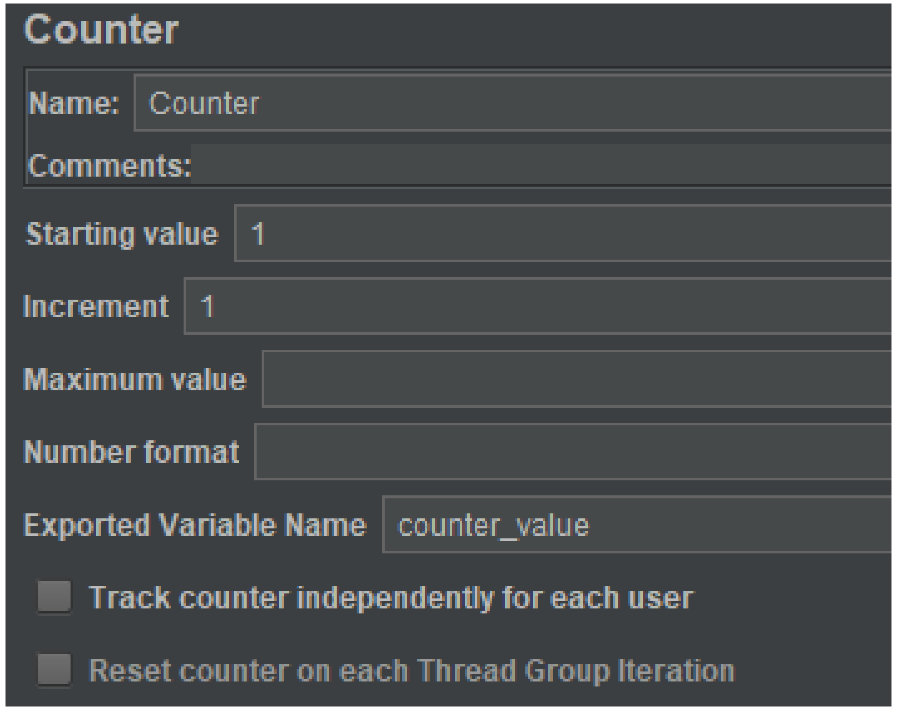
Counter

When it comes to building various types of advanced JMeter test plans, which include not only replaying a recorded test scenario with an increased number of users, but something more complex, you will likely need some form of a **Counter**.

These scenarios can include test plans that:

- ➢ Are dependent on external data sources like a **CSV Data Set, JDBC Result Set** or **previous response**.
- ➢ The target is just to run a test (or a part of a test) several times and each time have a dynamic parameter which represents a current iteration number

Counter



Counter

Name: Counter

Comments:

Starting value 1

Increment 1

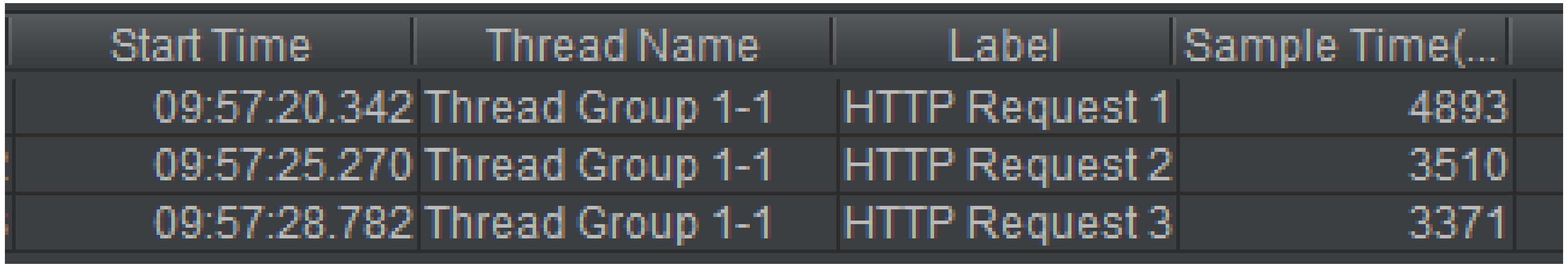
Maximum value

Number format

Exported Variable Name counter_value

☐ Track counter independently for each user

☐ Reset counter on each Thread Group Iteration



Start Time	Thread Name	Label	Sample Time(...
09:57:20.342	Thread Group 1-1	HTTP Request 1	4893
09:57:25.270	Thread Group 1-1	HTTP Request 2	3510
09:57:28.782	Thread Group 1-1	HTTP Request 3	3371

Random variable

JMeter allows you to generate random number values and use it in a variable. You can do so through the Random Variable config element. The Random Variable config element allows you set the following parameters:

Variable name: You can provide the name of the variable that can be used in your test plan elements. The random value will be stored in this variable.
Format String: You can specify the format of the generated number. It can be prefixed or suffixed with string. For example, if you want the generator to produce alphanumeric values you can specify the format like SALES_000 (000 will be replaced with the generated random number).

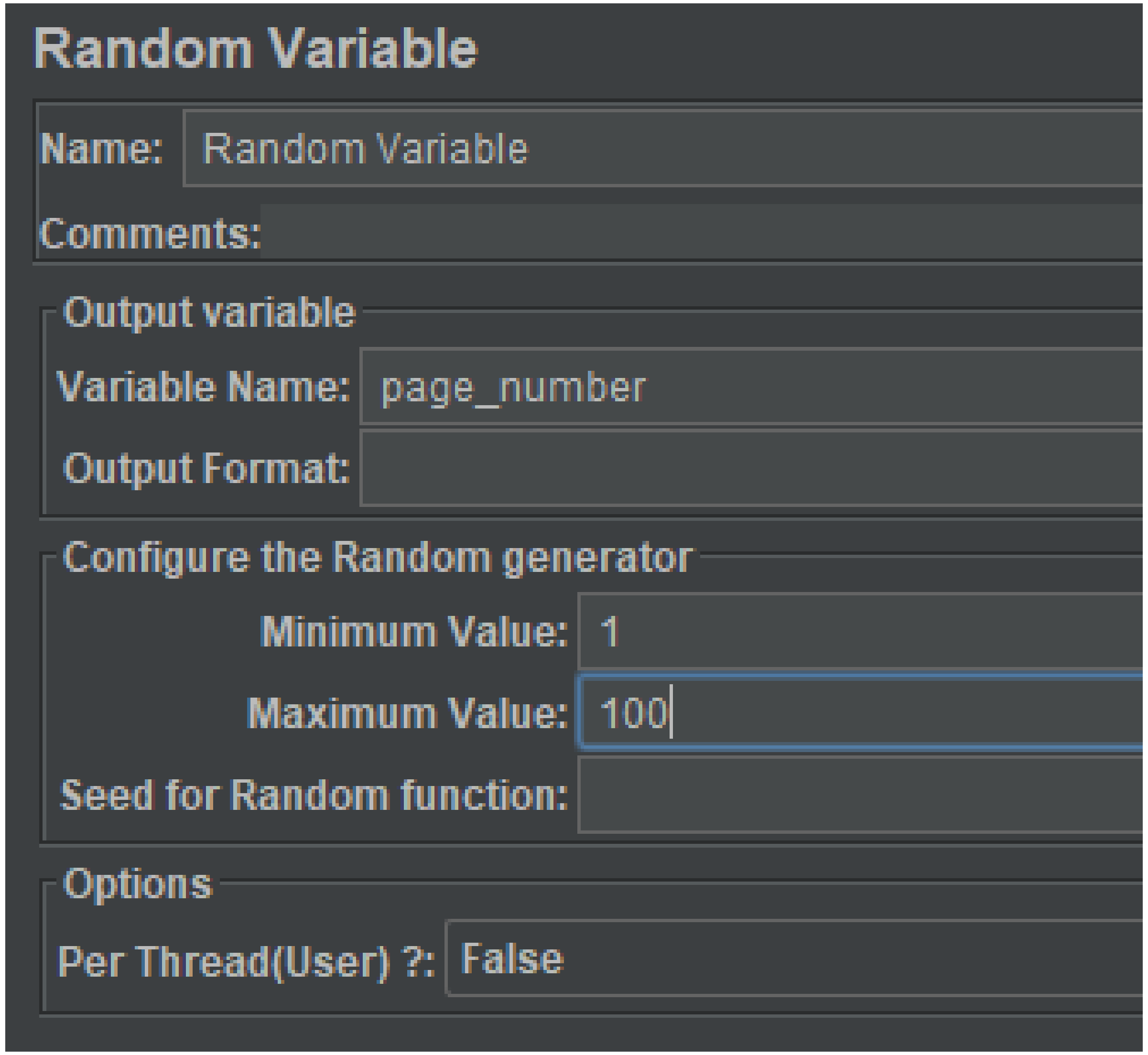
Random variable

Minimum and Maximum value: You can specify range within which the numbers to be generated. For example, the minimum number can be set as 10 and the maximum number can be set as 50. The generator will produce any number within that range.

Per Thread (User): You can specify whether random generator will be shared by all the threads (users) or each thread will have its own instance of random generator. This can indicated by setting false or true respectively.

Random Seed: You can also specify the seed value for your generator. If the same seed is used for every thread (Per Thread is set to true) then it will produce the same number for each thread.

Random variable



Random Variable

Name: Random Variable

Comments:

Output variable

Variable Name: page_number

Output Format:

Configure the Random generator

Minimum Value: 1

Maximum Value: 100

Seed for Random function:

Options

Per Thread(User) ?: False

Fresher Academy

Software

FRESHER ACADEMY

Happy Coding!