COMSATS UNIVERSITY ISLAMABAD

SOFTWARE TESTING

PROJECT REPORT

JMETER INSTALLATION AND GUIDE

PROGRAMME: BS-SOFTWARE ENGINEERING

(7th-SEMESTER)-7A

SUBMITTED TO: MS. NAJMUN NISA

SUBMITTED BY

RUBIA GULZAR (FA19-BSE-008)

RIMSHA ASAD (FA19-BSE-006)

JMeter Project Report

Installation

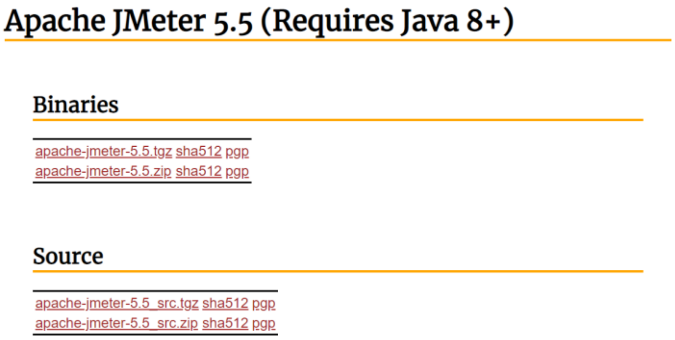
The **Apache JMeter™** application is open-source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions. JMeter is downloaded as an application. All you need is the latest version of Java (Min requires Java 8+) installed on your computer, which you can install from the following link.

https://www.oracle.com/java/technologies/downloads/#jdk19-windows

To download JMeter, use the following link

https://jmeter.apache.org/download_jmeter.cgi

In the Apache JMeter section, you can download zip files either binaries or source



Apache JMeter 5.5 (Requires Java 8+)

Binaries

apache-jmeter-5.5.tgz sha512 pgp
apache-jmeter-5.5.zip sha512 pgp

Source

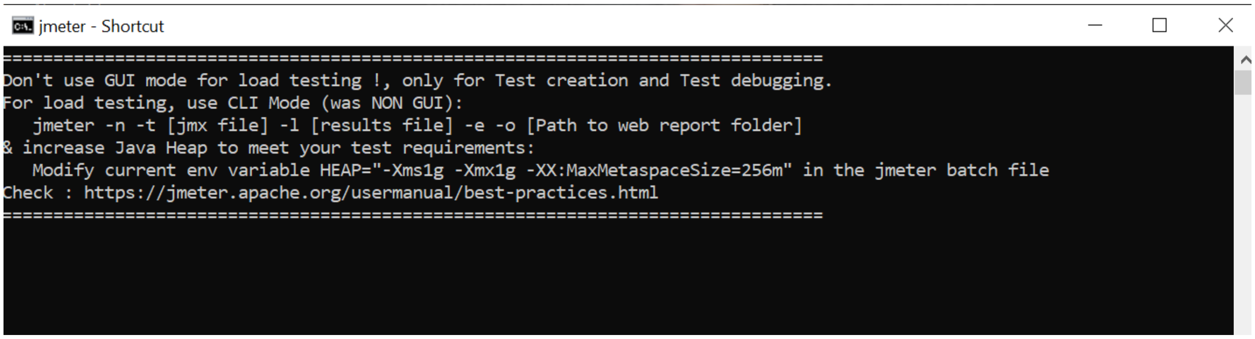
apache-jmeter-5.5_src.tgz sha512 pgp
apache-jmeter-5.5_src.zip sha512 pgp

After a successful download unzip the file and keep the folder at any location. Now start the JMeter

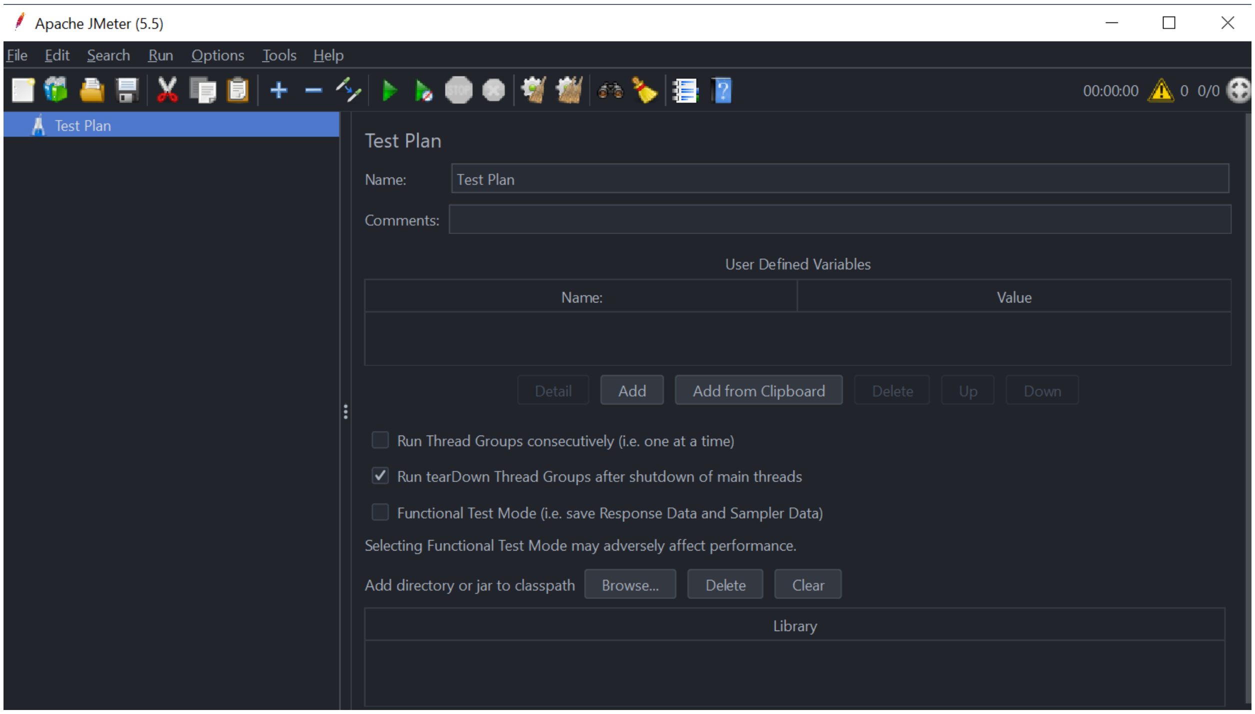
- For Windows: jmeter/bin – jmeter.bat
- For Mac/Unix/Linux: open terminal – jmeter/bin – sh jmeter.sh

Jmeter GUI

After running the ‘**jmeter.bat**’, the Jmeter GUI opens which looks like this:



```
================================================================================
Don't use GUI mode for load testing !, only for Test creation and Test debugging.
For load testing, use CLI Mode (was NON GUI):
   jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
& increase Java Heap to meet your test requirements:
   Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html
================================================================================
```



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

Basic Elements for JMeter

Thread Group

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.

Sampler

Samplers perform the actual work of JMeter. Each sampler (except Flow Control Action) generates one or more sample results. The sample results have various attributes (success/fail, elapsed time, data size, etc.) and can be viewed by the various listeners.

Configuration Element

Configuration elements can be used to set up defaults and variables for later use by samplers. Note that these elements are processed at the start of the scope in which they are found, i.e. before any samplers in the same scope.

Listeners

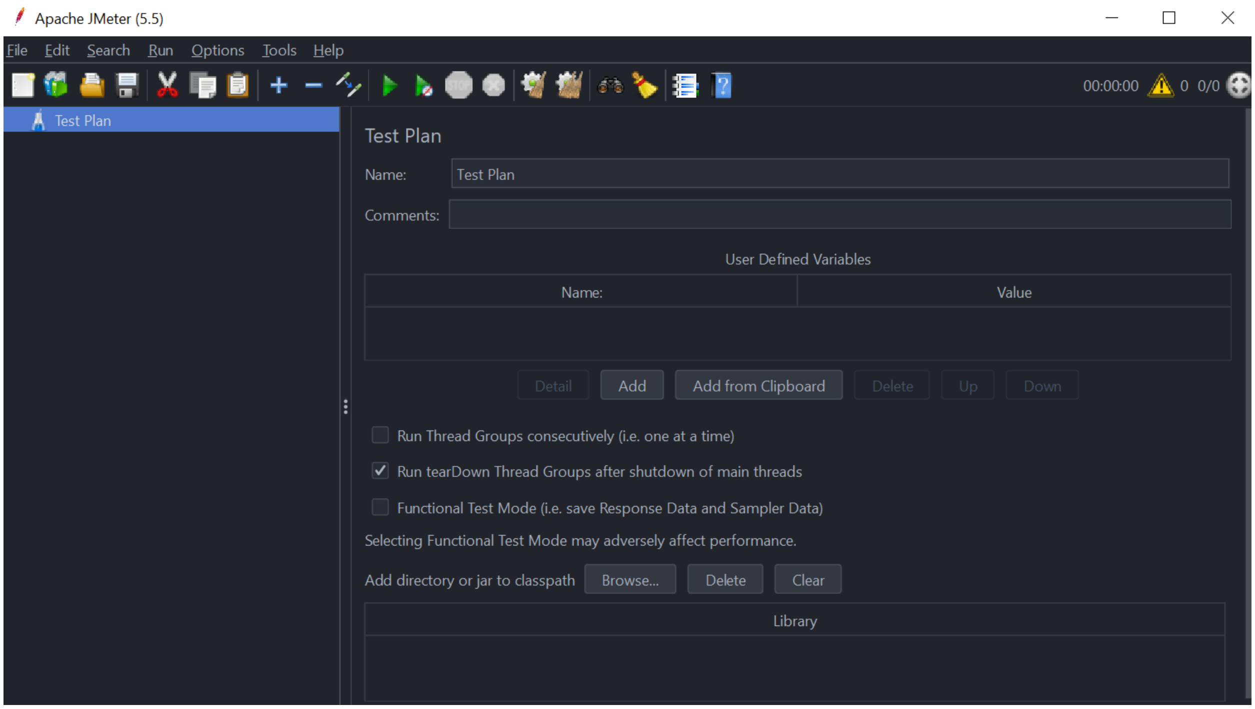
Most of the listeners perform several roles in addition to "listening" to the test results. They also provide means to view, save, and read saved test results.

Assertions

Assertions are used to perform additional checks on samplers and are processed after **every sampler** in the same scope. To ensure that an Assertion is applied only to a particular sampler, add it as a child of the sampler.

Manually add the first test case

1. Start Jmeter and create a Test plan



Test Plan

Name: Test Plan

Comments:

Name:	Value

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

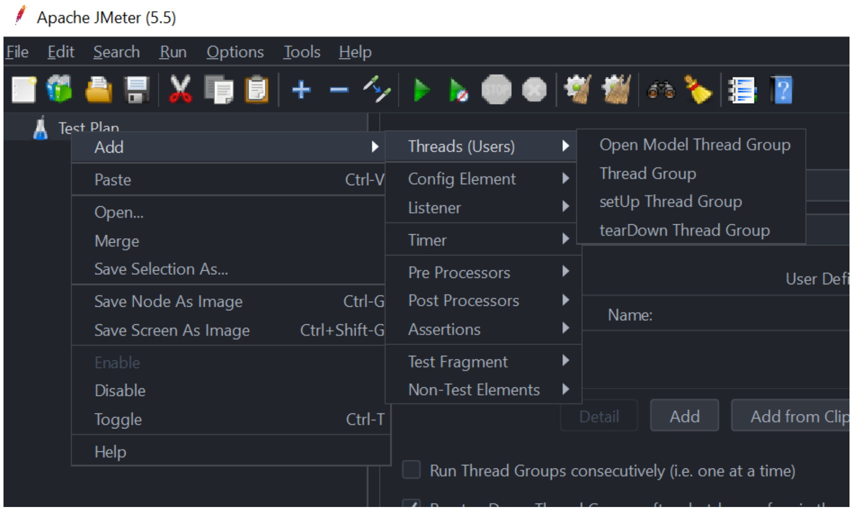
☑ Run tearDown Thread Groups after shutdown of main threads

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

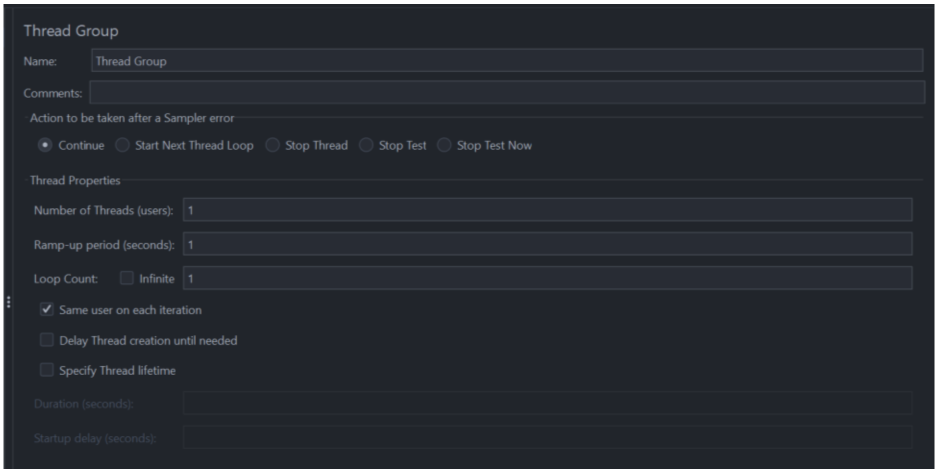
Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

2. Create a Thread Group (Users)



Add number of threads (users) and Ramp-up time (seconds), set loop count (number of iteration to run 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

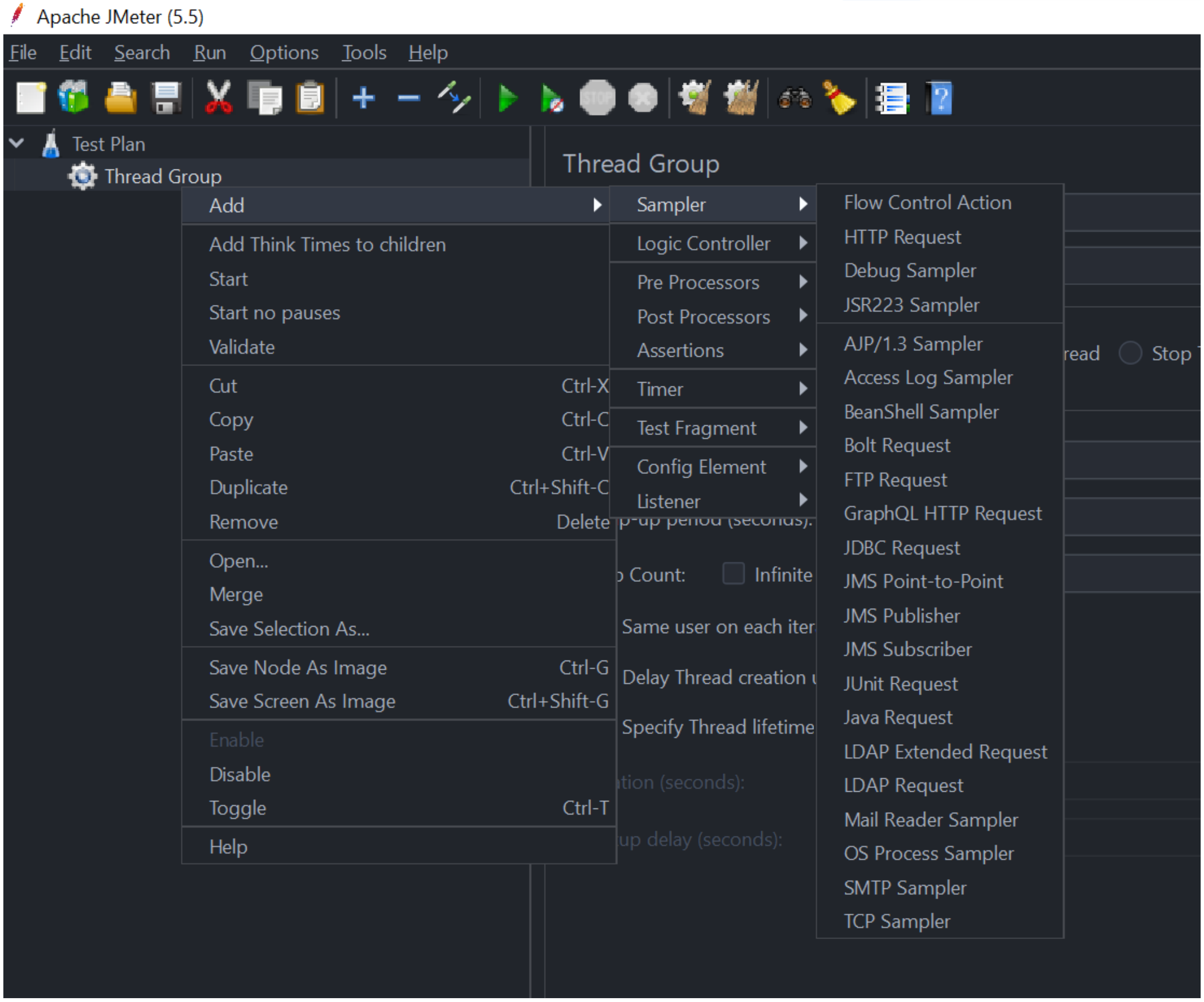
□ Delay Thread creation until needed

□ Specify Thread lifetime

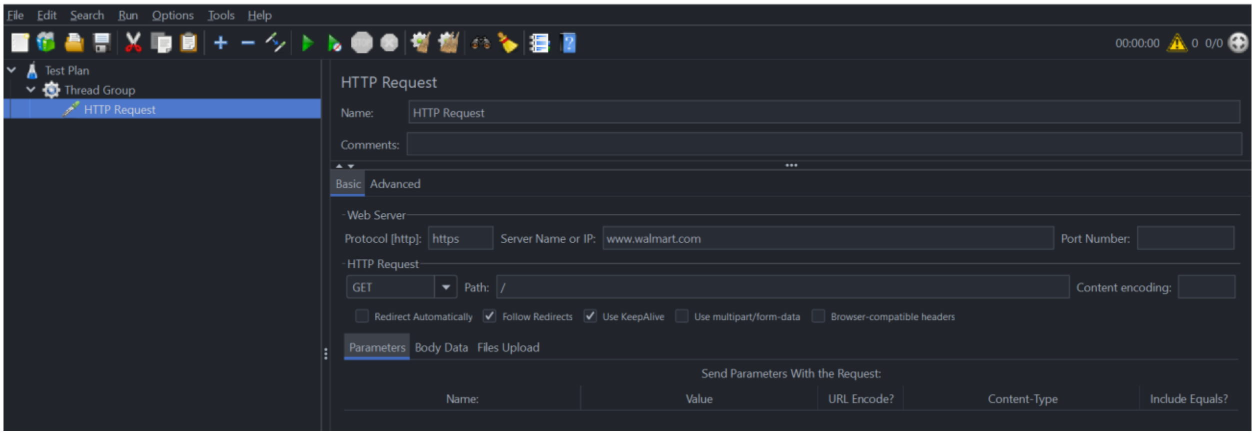
Duration (seconds):

Startup delay (seconds):

3. Add a Sampler (HTTP)



Add Http Request Protocol (if it is http then it's not necessary to write and if its https then write the https) and Server name or IP (e.g. www.walmart.com), Port Number (optional).


File Edit Search Run Options Tools Help
00:00:00 0 0/0
Test Plan
Thread Group
HTTP Request
Basic Advanced
Parameters Body Data Files Upload
Send Parameters With the Request:
Name: Value URL Encode? Content-Type Include Equals?

HTTP Request

Name: HTTP Request

Comments:

Web Server

Protocol [http]: https

Server Name or IP: www.walmart.com

Port Number:

HTTP Request

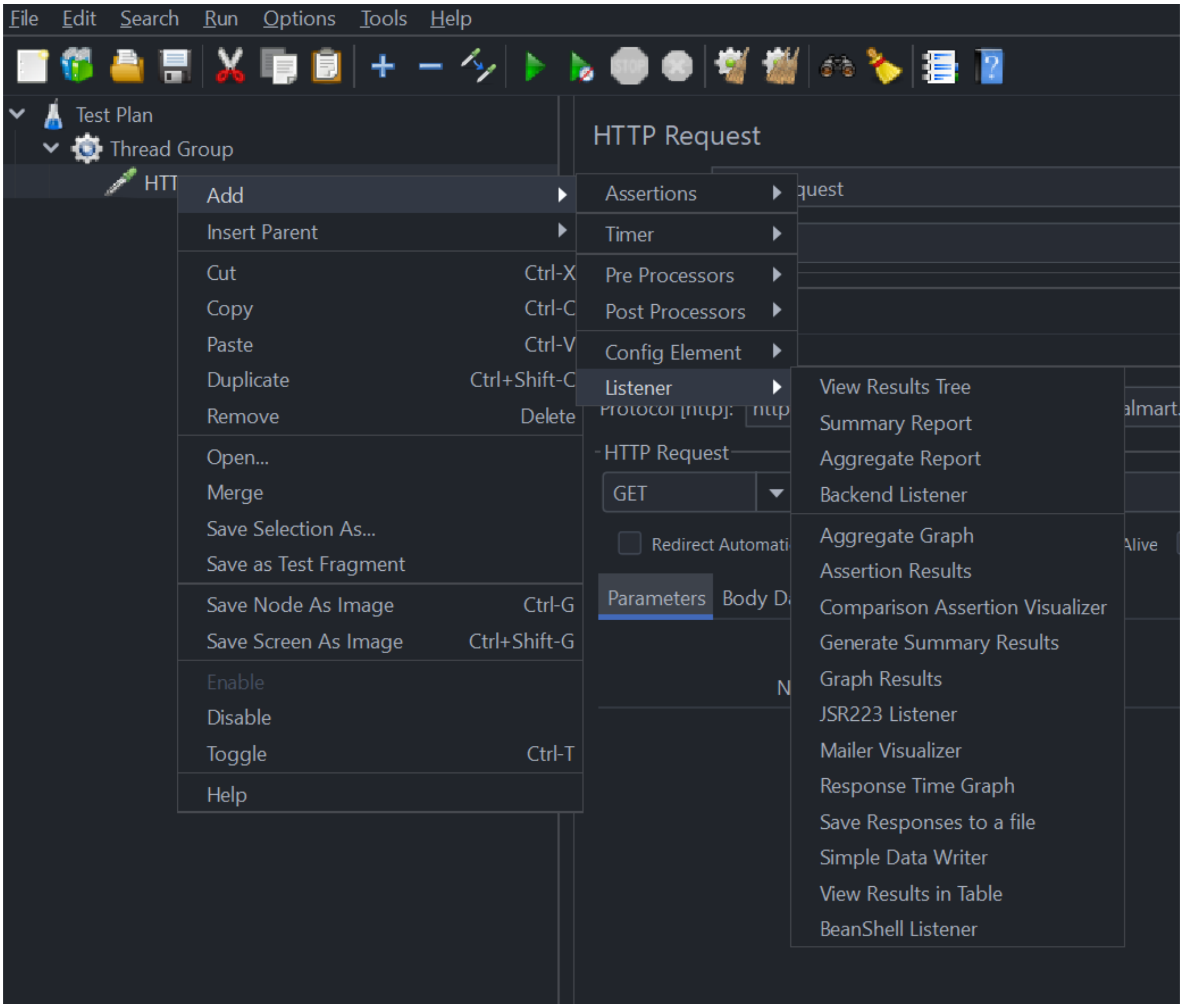
GET

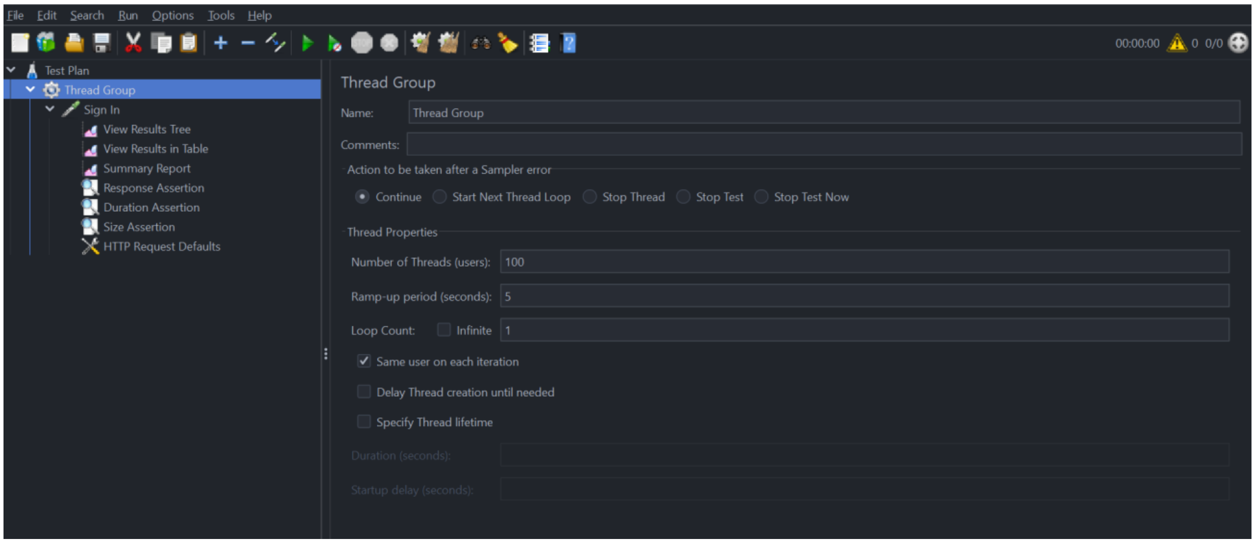
Path: /

Content encoding:

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

4. Add Listeners and Assertions





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): 100

Ramp-up period (seconds): 5

Loop Count: □ Infinite 1

☑ Same user on each iteration

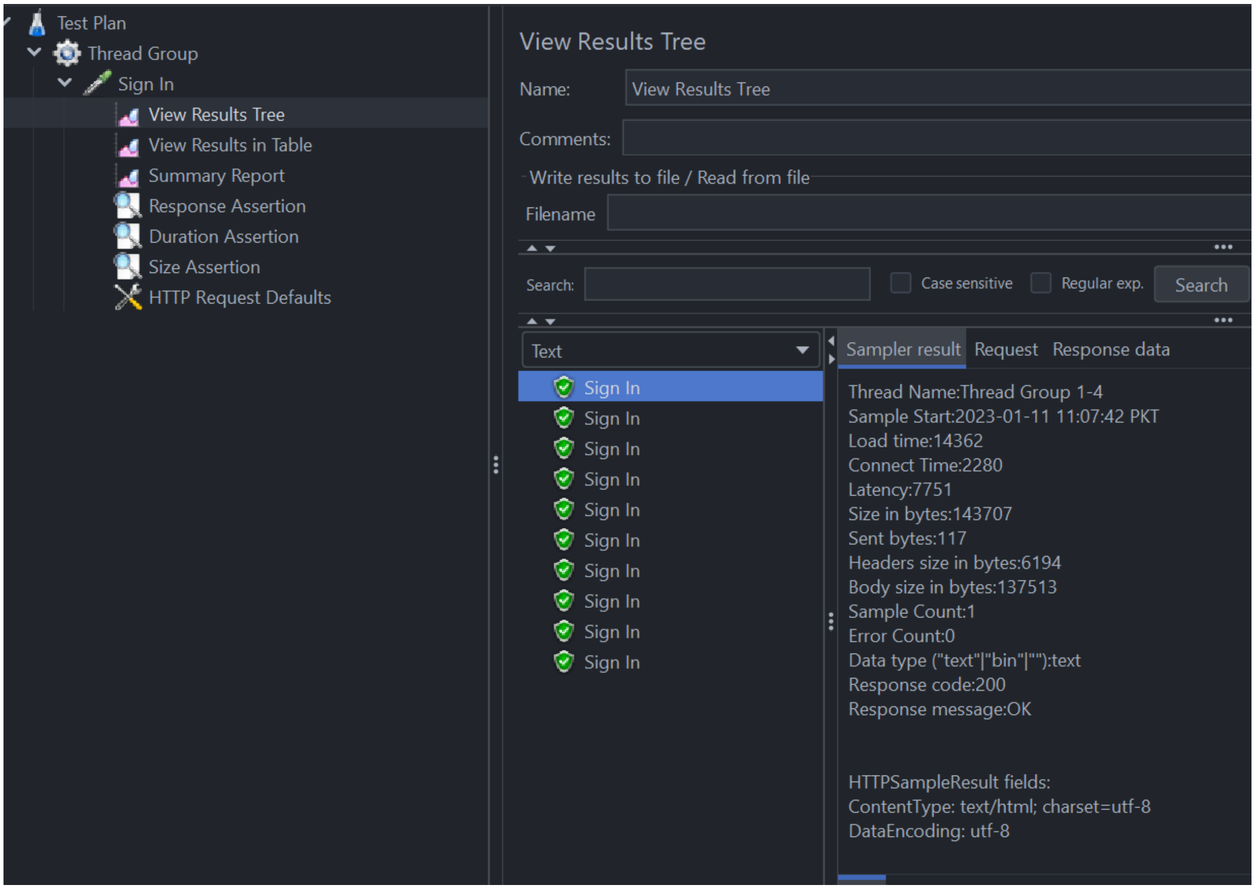
□ Delay Thread creation until needed

□ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

5. Run the test



View Results Tree

Name: View Results Tree

Comments:

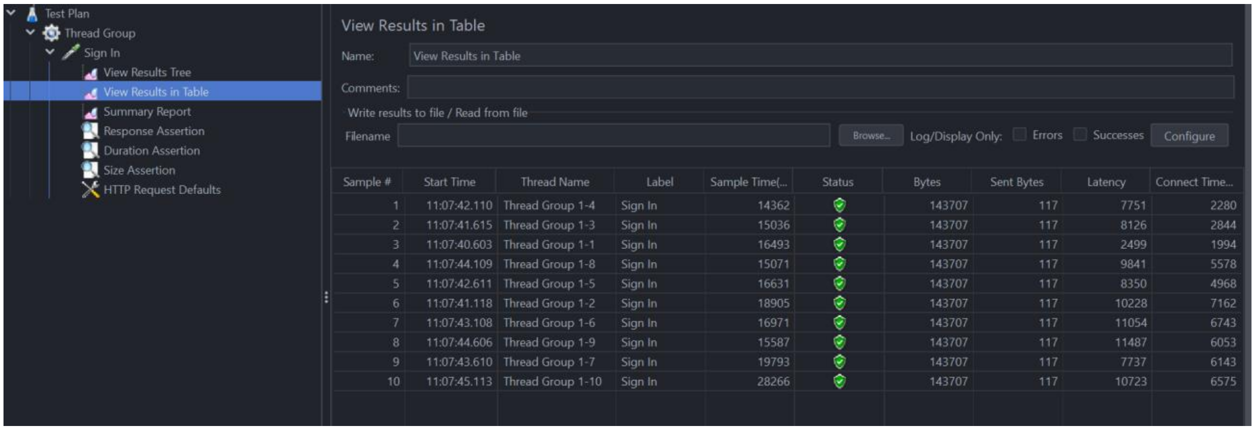
Write results to file / Read from file

Filename

Search: □ Case sensitive □ Regular exp.

```
Thread Name:Thread Group 1-4
Sample Start:2023-01-11 11:07:42 PKT
Load time:14362
Connect Time:2280
Latency:7751
Size in bytes:143707
Sent bytes:117
Headers size in bytes:6194
Body size in bytes:137513
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK


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



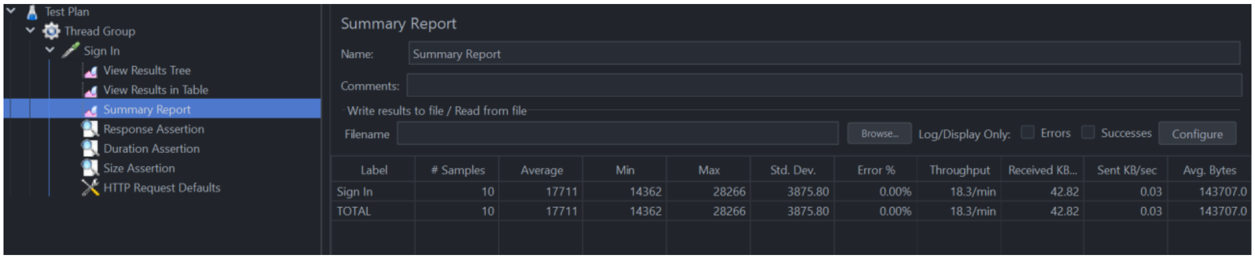
View Results in Table

Name: View Results in Table

Comments:

Filename

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	11:07:42.110	Thread Group 1-4	Sign In	14362	✓	143707	117	7751	2280
2	11:07:41.615	Thread Group 1-3	Sign In	15036	✓	143707	117	8126	2844
3	11:07:40.603	Thread Group 1-1	Sign In	16493	✓	143707	117	2499	1994
4	11:07:44.109	Thread Group 1-8	Sign In	15071	✓	143707	117	9841	5578
5	11:07:42.611	Thread Group 1-5	Sign In	16631	✓	143707	117	8350	4968
6	11:07:41.118	Thread Group 1-2	Sign In	18905	✓	143707	117	10228	7162
7	11:07:43.108	Thread Group 1-6	Sign In	16971	✓	143707	117	11054	6743
8	11:07:44.606	Thread Group 1-9	Sign In	15587	✓	143707	117	11487	6053
9	11:07:43.610	Thread Group 1-7	Sign In	19793	✓	143707	117	7737	6143
10	11:07:45.113	Thread Group 1-10	Sign In	28266	✓	143707	117	10723	6575



Summary Report

Name: Summary Report

Comments:

Filename

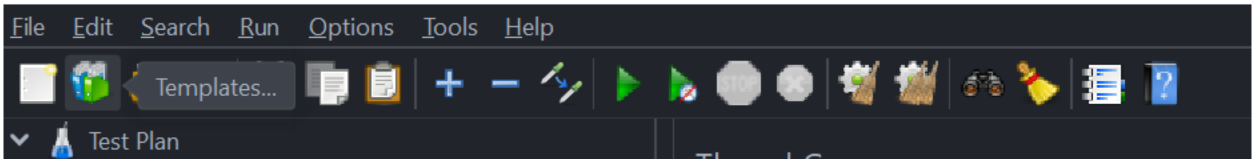
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
Sign In	10	17711	14362	28266	3875.80	0.00%	18.3/min	42.82	0.03	143707.0
TOTAL	10	17711	14362	28266	3875.80	0.00%	18.3/min	42.82	0.03	143707.0

Test Plan Using Test Script Recorder

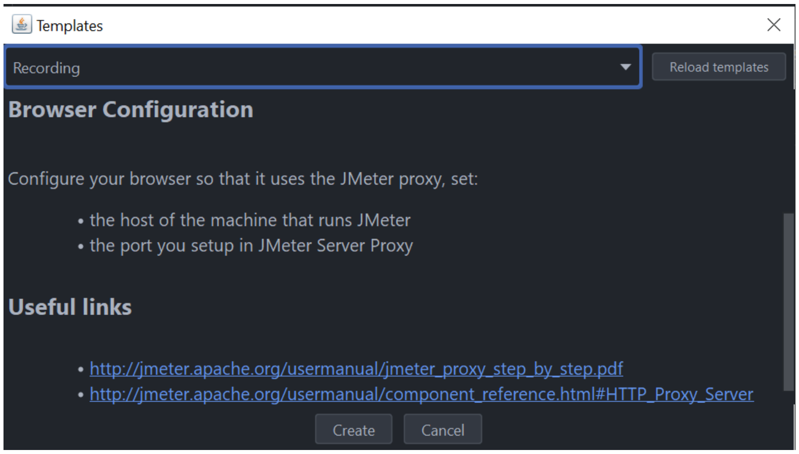
One easy way to create a test plan is to use the Recorder

Basic Steps

1. Go to jmeter/bin and start Jmeter with jmeter.bat in windows and jmeter.sh on Linux/Unix
2. Select Templates on the menu bar



3. Select the Recording template on the list and click on create



Browser Configuration

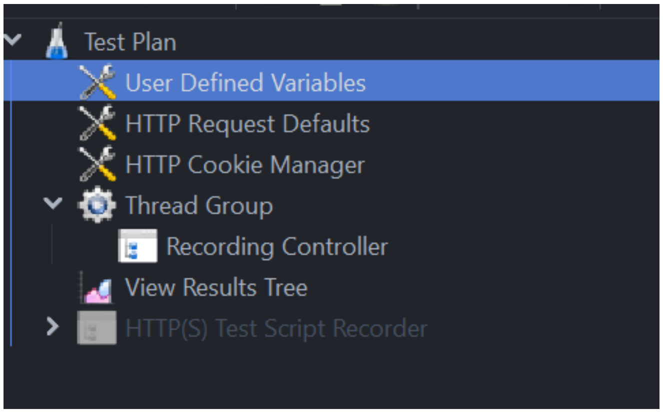
Configure your browser so that it uses the JMeter proxy, set:

- the host of the machine that runs JMeter
- the port you setup in JMeter Server Proxy

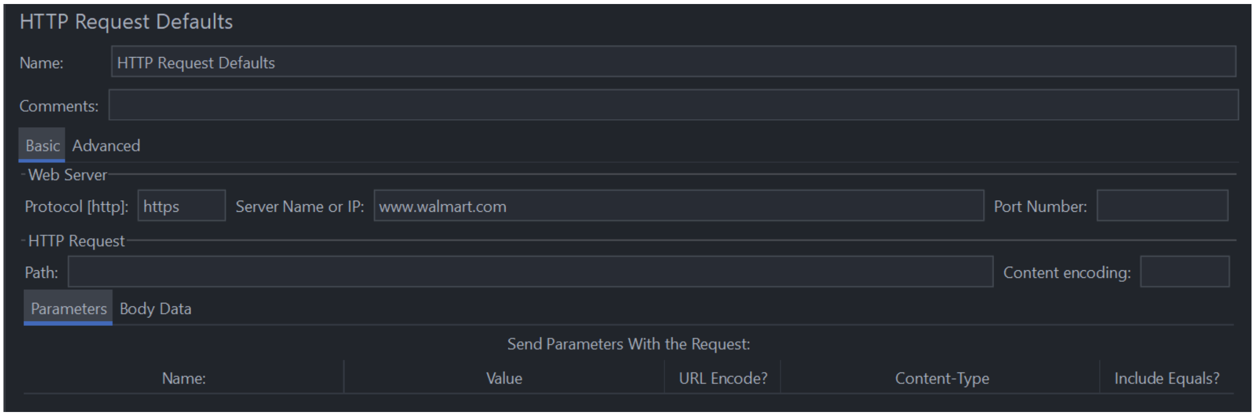
Useful links

- http://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.pdf
- http://jmeter.apache.org/usermanual/component_reference.html#HTTP_Proxy_Server

4. A complete Test Plan is generated



5. In the HTTP Request Defaults elements, enter the server name or IP and leave the path blank.



HTTP Request Defaults

Name: HTTP Request Defaults

Comments:

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: www.walmart.com Port Number:

HTTP Request

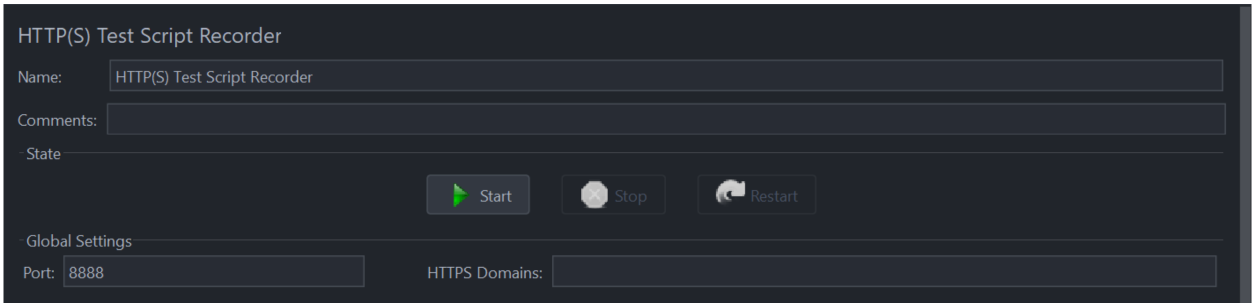
Path: Content encoding:

Parameters Body Data

Send Parameters With the Request:

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

6. Return to HTTP(s) Test Script Recorder and click the Start button at the top.



HTTP(S) Test Script Recorder

Name: HTTP(S) Test Script Recorder

Comments:

State

Global Settings

Port: 8888 HTTPS Domains:

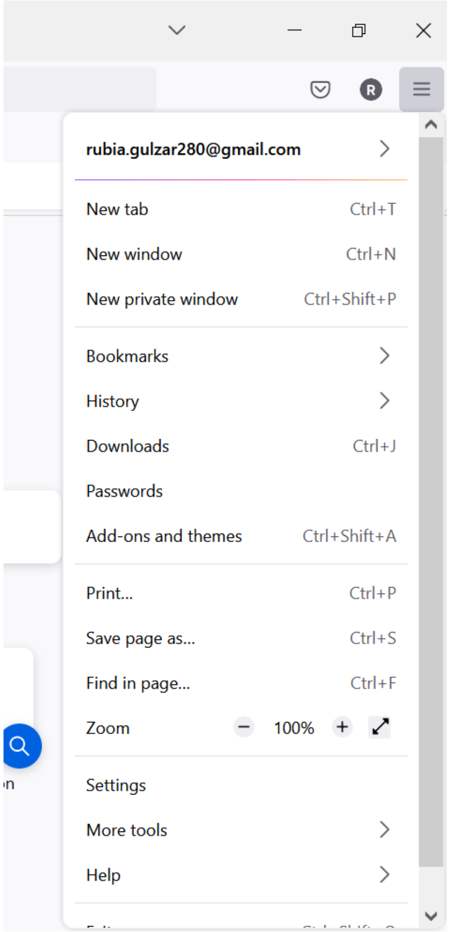
This will start the Jmeter proxy server which is used to intercept the browser requests. A file called ApacheJMeterTemporaryRootCA.crt will be generated in jmeter/bin. Install this certificate in your browser.

Configure your browser to use the Jmeter Proxy

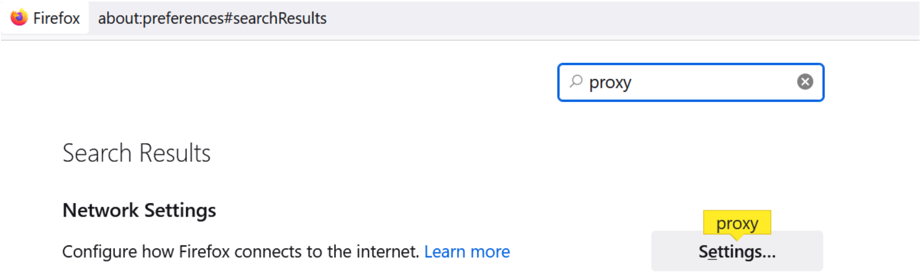
At this point, Jmeter’s proxy is running. For this, we used Firefox and Chrome to view some pages on the Jmeter website

For Firefox

1. Start Firefox, but do not close Jmeter.
2. From the toolbar, open the application menu, click the Settings button



3. On the new pop-up, search for proxy and click Settings button near the top.



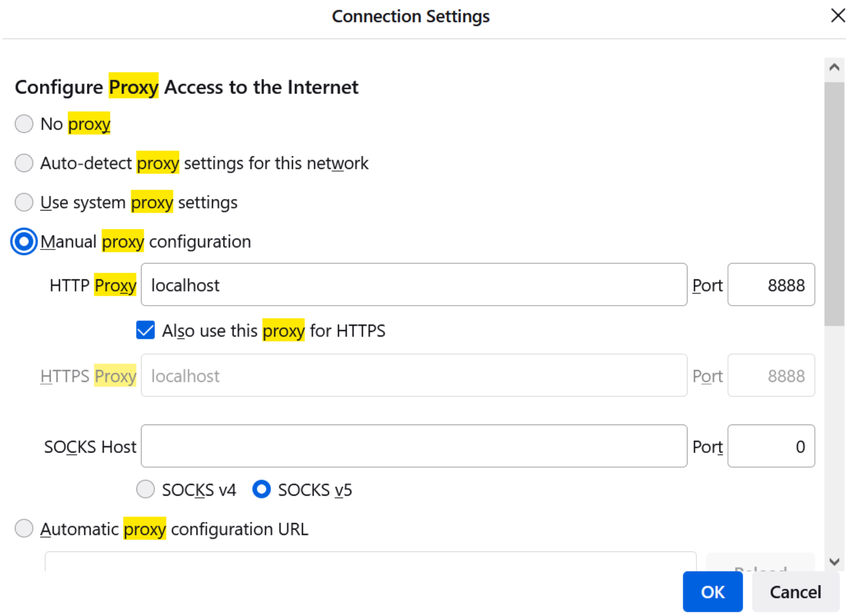
Configure how Firefox connects to the internet. Learn more

4. Check Manual proxy configuration. The address and port fields should be enabled now.

   HTTP proxy: enter localhost or the IP address of your system

   Port: enter 8888

5. Check use this proxy server for HTTPS



Connection Settings

Configure Proxy Access to the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy localhost Port 8888

☑ Also use this proxy for HTTPS

HTTPS Proxy localhost Port 8888

SOCKS Host Port 0

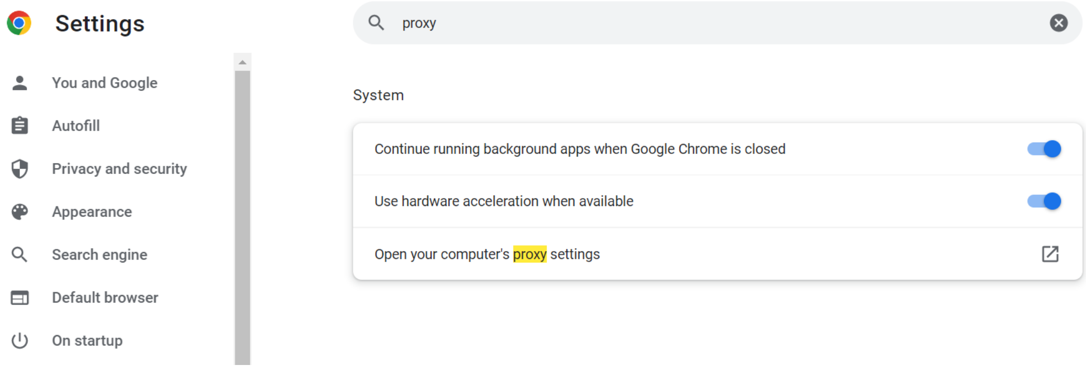
○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

6. Click OK button. This should return you to the browser.

For Windows

1. Start Chrome, but do not close Jmeter
2. From menu, select Settings and search for proxy this will open



Settings

proxy

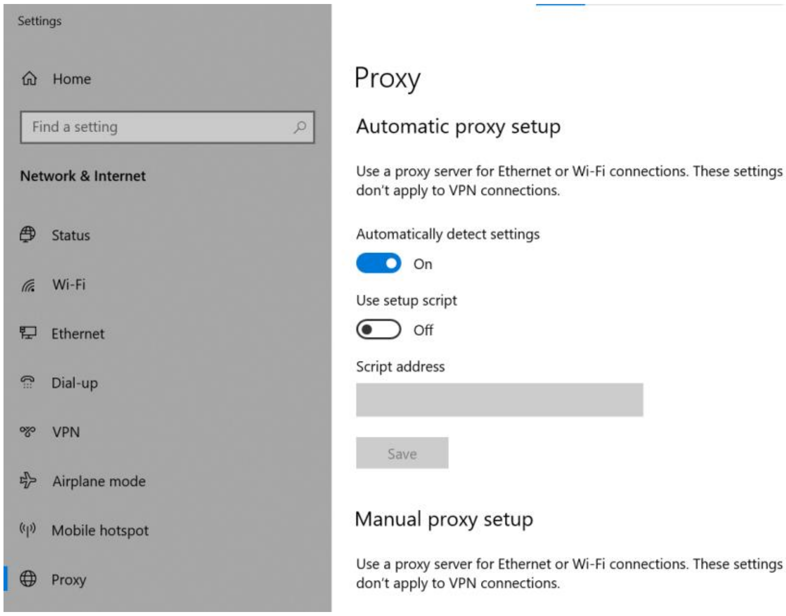
System

Continue running background apps when Google Chrome is closed (on)

Use hardware acceleration when available (on)

Open your computer's proxy settings

3. Click Open your computer’s proxy settings.



Proxy

Automatic proxy setup

Use a proxy server for Ethernet or Wi-Fi connections. These settings don’t apply to VPN connections.

Automatically detect settings

On

Use setup script

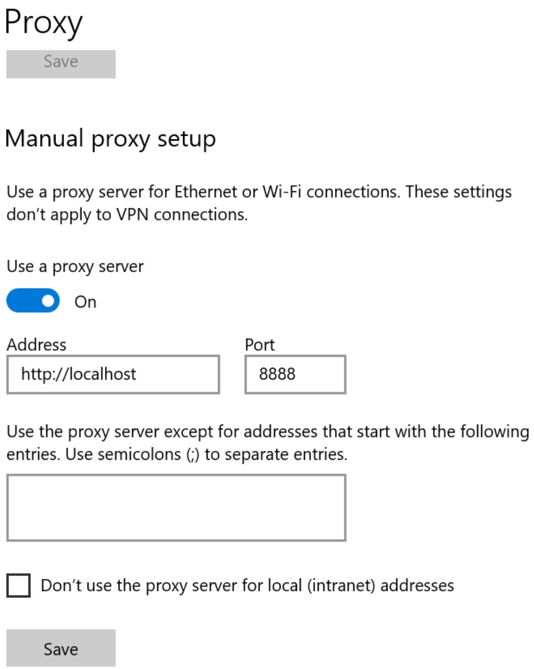
Off

Script address

Manual proxy setup

Use a proxy server for Ethernet or Wi-Fi connections. These settings don’t apply to VPN connections.

4. Go to Manual proxy setup, enable it and add Address localhost and port 8888. Then click on Save button


Proxy

Manual proxy setup

Use a proxy server for Ethernet or Wi-Fi connections. These settings don’t apply to VPN connections.

Use a proxy server

On

Address: http://localhost

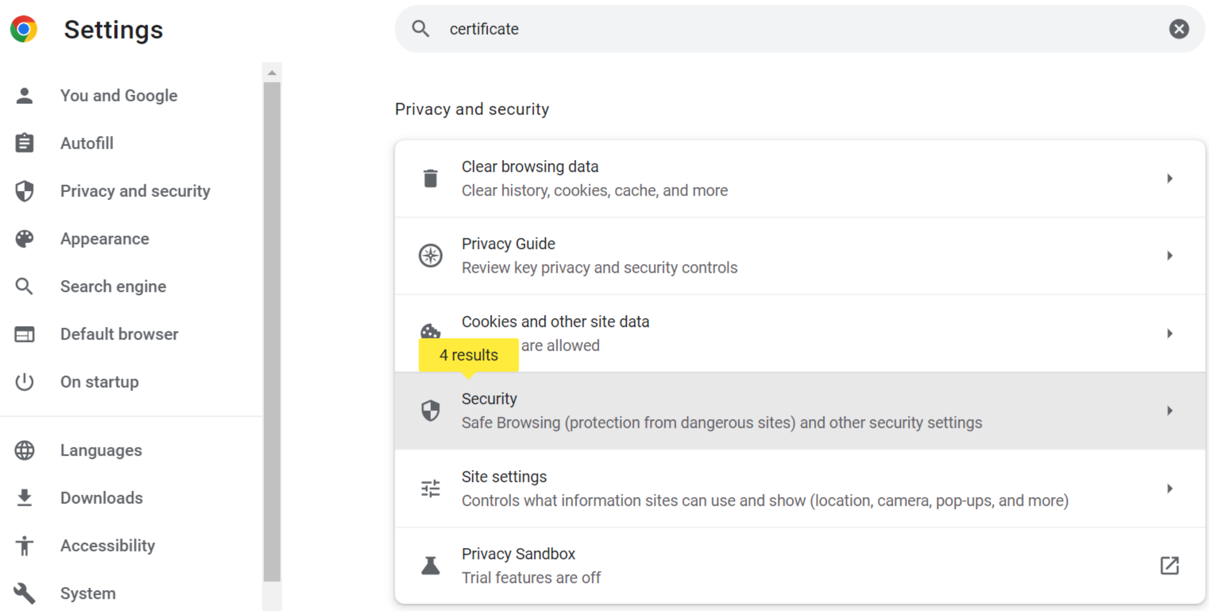
Port: 8888

Use the proxy server except for addresses that start with the following entries. Use semicolons (;) to separate entries.

□ Don’t use the proxy server for local (intranet) addresses

To Add Certificate

1. Go to Settings and search for certificate



certificate

Privacy and security

Clear browsing data
Clear history, cookies, cache, and more

Privacy Guide
Review key privacy and security controls

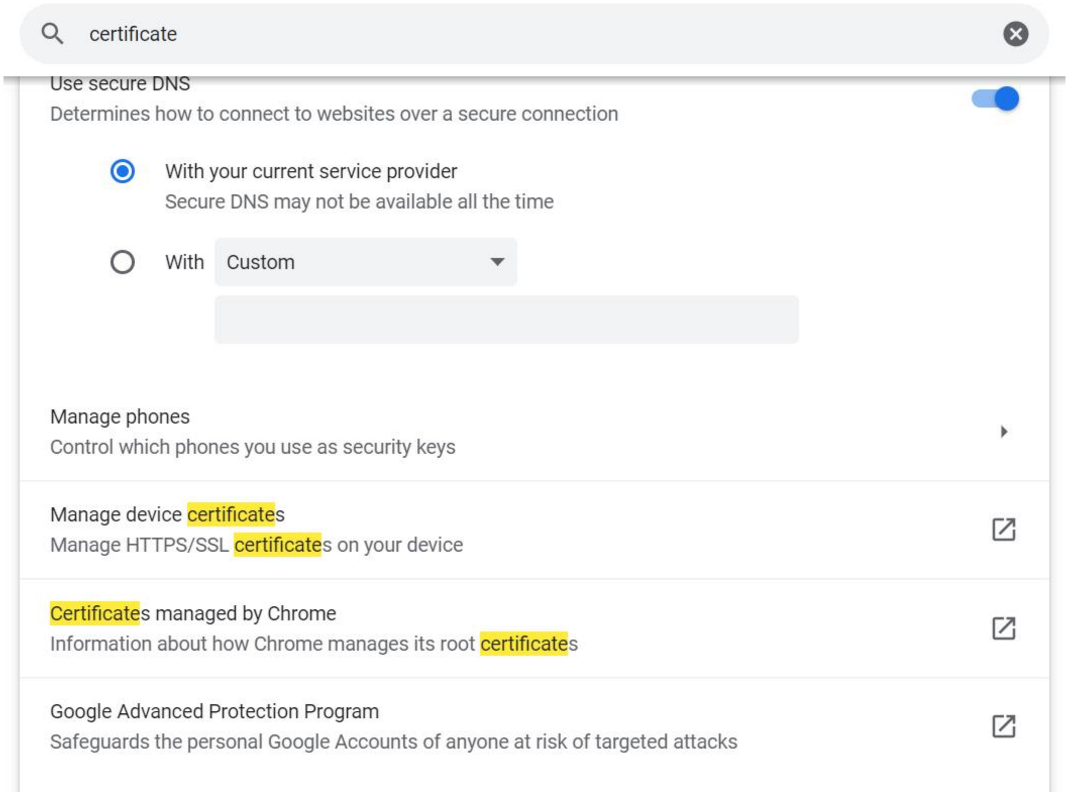
Cookies and other site data
are allowed

Security
Safe Browsing (protection from dangerous sites) and other security settings

Site settings
Controls what information sites can use and show (location, camera, pop-ups, and more)

Privacy Sandbox
Trial features are off

2. Click on Security and go to Manage device certificate



certificate

Use secure DNS
Determines how to connect to websites over a secure connection

● With your current service provider
Secure DNS may not be available all the time

○ With Custom

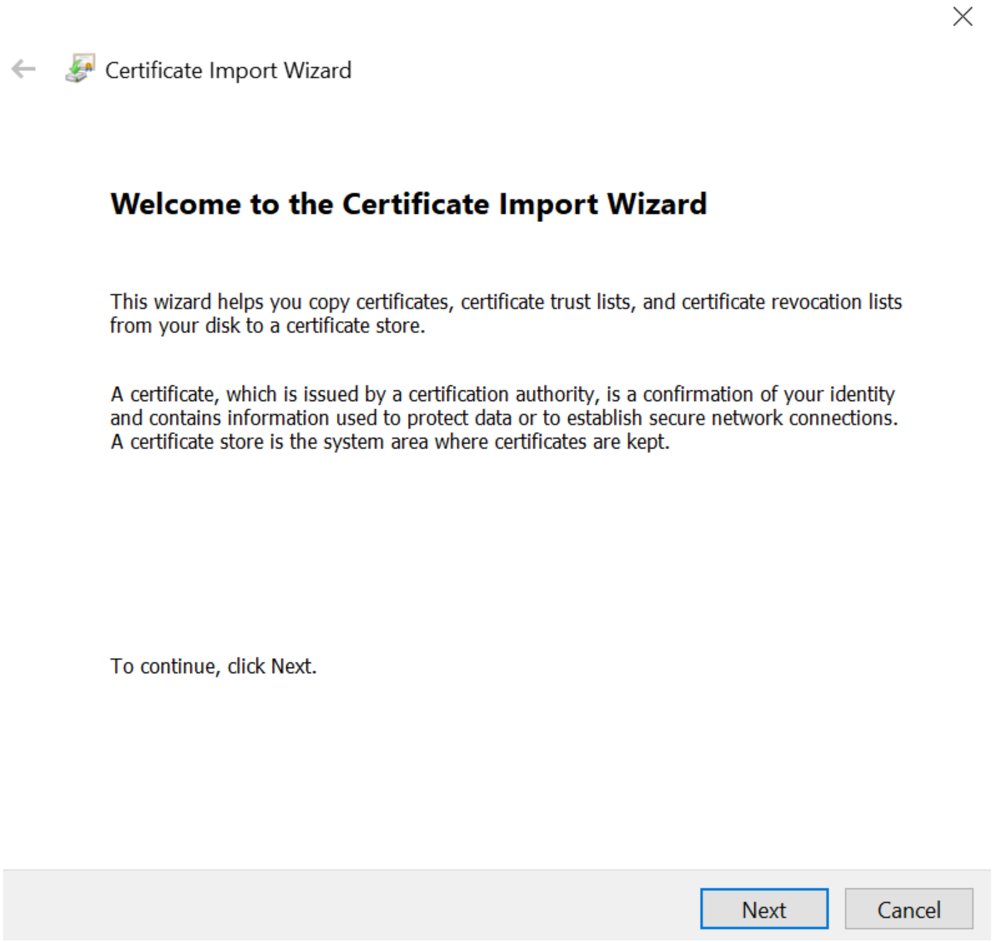
Manage phones
Control which phones you use as security keys

Manage device certificates
Manage HTTPS/SSL certificates on your device

Certificates managed by Chrome
Information about how Chrome manages its root certificates

Google Advanced Protection Program
Safeguards the personal Google Accounts of anyone at risk of targeted attacks

3. Import certificates that will pop up the below screen, click on Next



Certificate Import Wizard

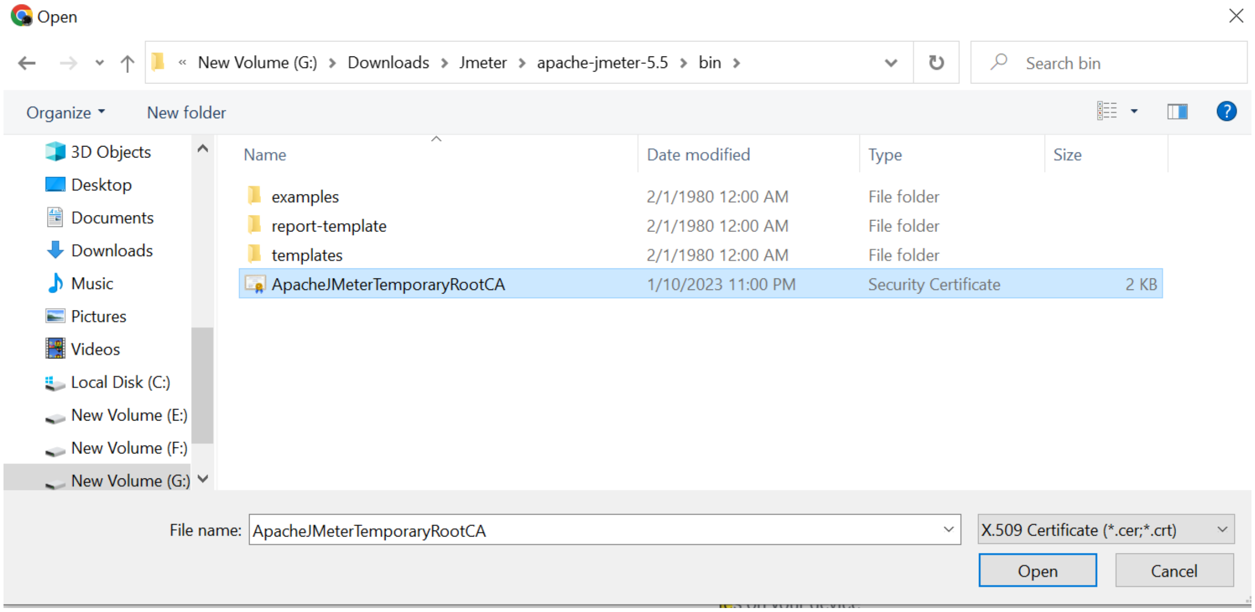
Welcome to the Certificate Import Wizard

This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

4. Open the location where Jmeter installed then go to jmeter/bin and add this file to chrome.



Open

« New Volume (G:) › Downloads › Jmeter › apache-jmeter-5.5 › bin ›

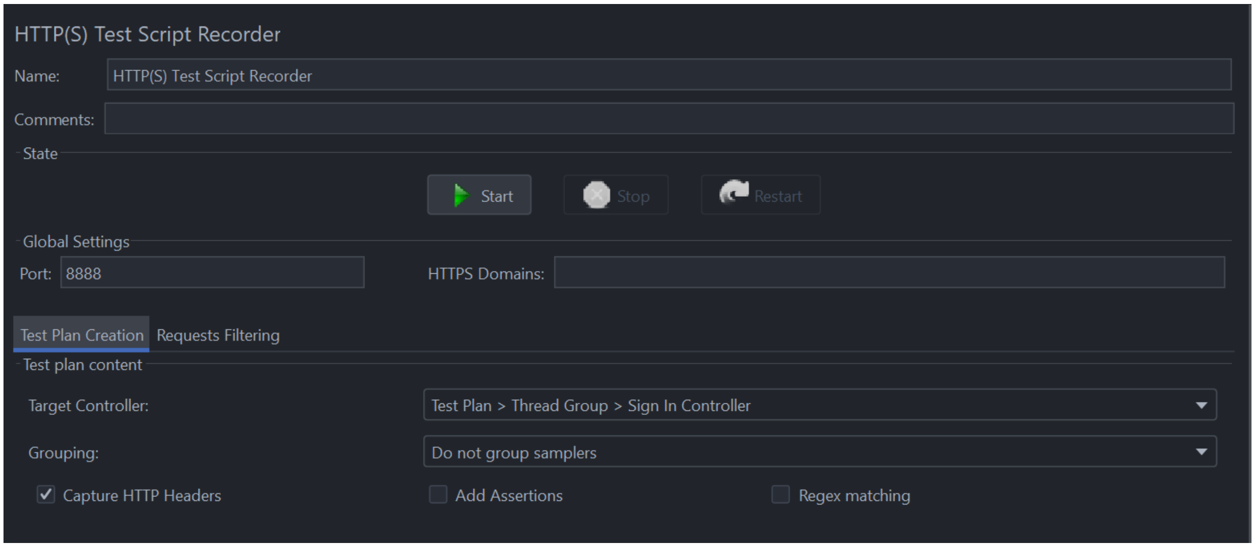
Name	Date modified	Type	Size
examples	2/1/1980 12:00 AM	File folder	
report-template	2/1/1980 12:00 AM	File folder	
templates	2/1/1980 12:00 AM	File folder	
ApacheJMeterTemporaryRootCA	1/10/2023 11:00 PM	Security Certificate	2 KB

File name: ApacheJMeterTemporaryRootCA

X.509 Certificate (*.cer;*.crt)

Record the Navigation

1. In the HTTPS Test Script Recorder, set the target Controller as here Sign In Controller



HTTP(S) Test Script Recorder

Name: HTTP(S) Test Script Recorder

Comments:

State

Global Settings

Port: 8888

HTTPS Domains:

Test plan content

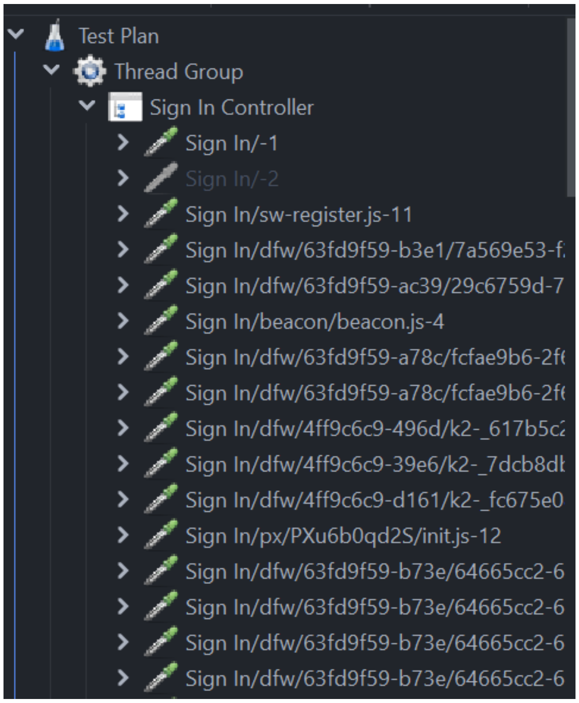
Target Controller: Test Plan > Thread Group > Sign In Controller

Grouping: Do not group samplers

☑ Capture HTTP Headers ☐ Add Assertions ☐ Regex matching

2. Click on Start button, with your browser enter https://www.walmart.com and hit the enter key.
3. Click on a few links on your site pages for sign-in.
4. Stop the recording and close your browser and bring up the Jmeter window.

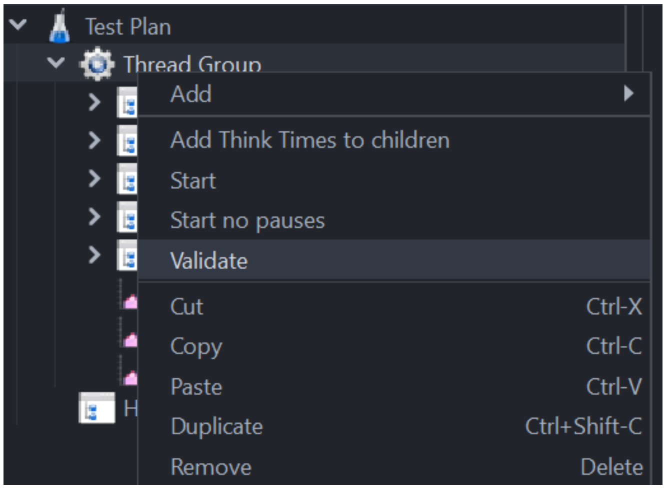
Expand the Thread Group and there should be several samplers. At this point, the test plan can be saved as is.



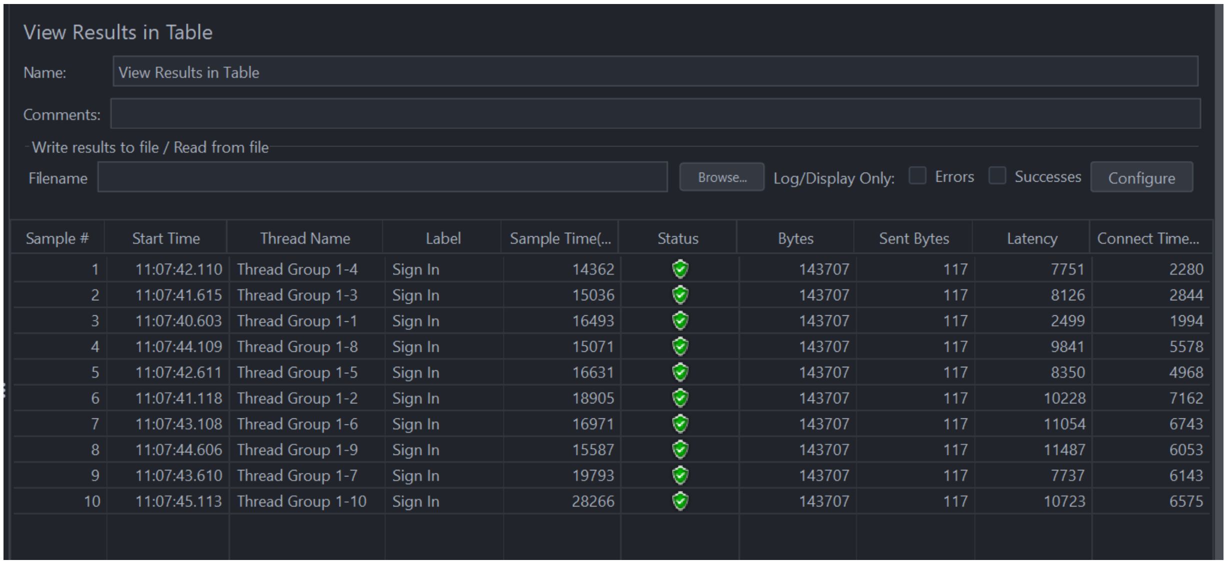
```
Test Plan
  Thread Group
    Sign In Controller
      Sign In/-1
      Sign In/-2
      Sign In/sw-register.js-11
      Sign In/dfw/63fd9f59-b3e1/7a569e53-f.
      Sign In/dfw/63fd9f59-ac39/29c6759d-7
      Sign In/beacon/beacon.js-4
      Sign In/dfw/63fd9f59-a78c/fcfae9b6-2f(
      Sign In/dfw/63fd9f59-a78c/fcfae9b6-2f(
      Sign In/dfw/4ff9c6c9-496d/k2-_617b5c
      Sign In/dfw/4ff9c6c9-39e6/k2-_7dcb8dl
      Sign In/dfw/4ff9c6c9-d161/k2-_fc675e0
      Sign In/px/PXu6b0qd2S/init.js-12
      Sign In/dfw/63fd9f59-b73e/64665cc2-6
      Sign In/dfw/63fd9f59-b73e/64665cc2-6
      Sign In/dfw/63fd9f59-b73e/64665cc2-6
      Sign In/dfw/63fd9f59-b73e/64665cc2-6
```

Validate the script

Now we need to validate the script before to run out test plan. Save the test plan. Right click on the thread Group Validate.



Check the view result tree and view result table.



View Results in Table

Name: View Results in Table

Comments:

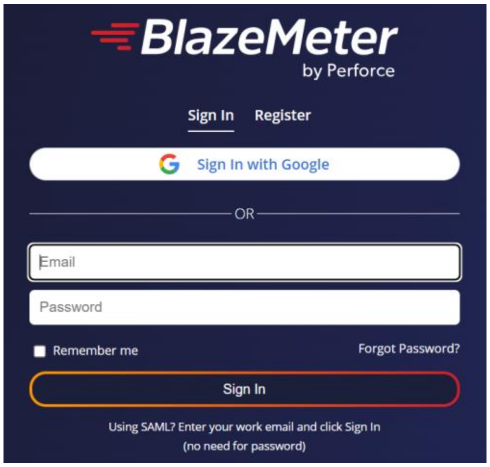
Write results to file / Read from file

Filename Browse... Log/Display Only: □ Errors □ Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	11:07:42.110	Thread Group 1-4	Sign In	14362	✓	143707	117	7751	2280
2	11:07:41.615	Thread Group 1-3	Sign In	15036	✓	143707	117	8126	2844
3	11:07:40.603	Thread Group 1-1	Sign In	16493	✓	143707	117	2499	1994
4	11:07:44.109	Thread Group 1-8	Sign In	15071	✓	143707	117	9841	5578
5	11:07:42.611	Thread Group 1-5	Sign In	16631	✓	143707	117	8350	4968
6	11:07:41.118	Thread Group 1-2	Sign In	18905	✓	143707	117	10228	7162
7	11:07:43.108	Thread Group 1-6	Sign In	16971	✓	143707	117	11054	6743
8	11:07:44.606	Thread Group 1-9	Sign In	15587	✓	143707	117	11487	6053
9	11:07:43.610	Thread Group 1-7	Sign In	19793	✓	143707	117	7737	6143
10	11:07:45.113	Thread Group 1-10	Sign In	28266	✓	143707	117	10723	6575

Blazemeter Recorder

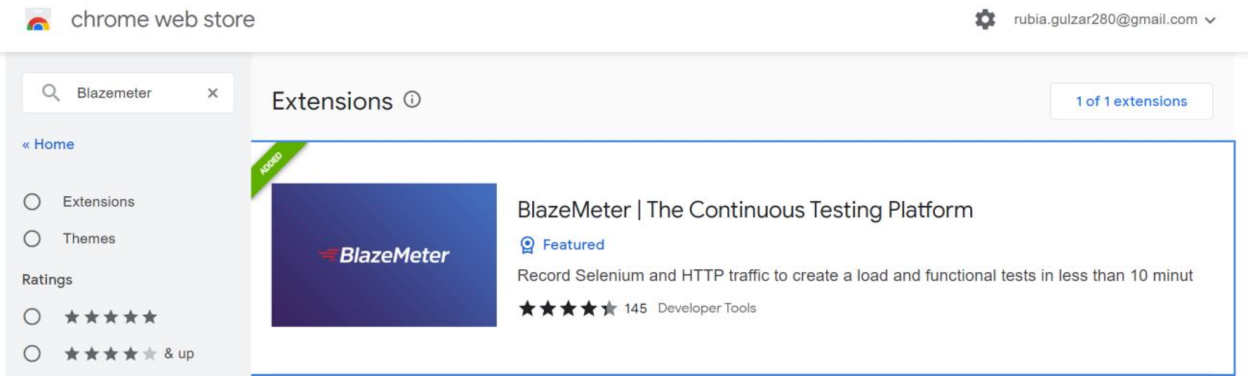
1. Create account on Blazemeter using link https://www.blazemeter.com/



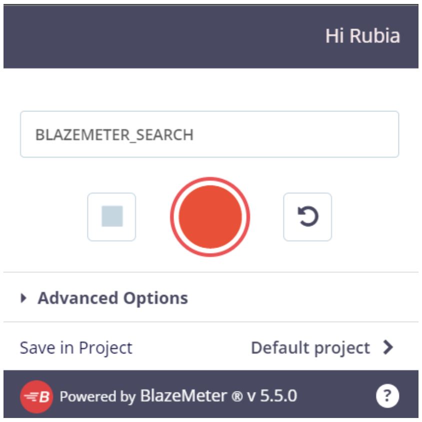
2. Get Blazemeter extension on browser

   Go to Settings – Extensions – Menu – Open Chrome Web Store – Search Blazemeter

   Then add it to Chrome

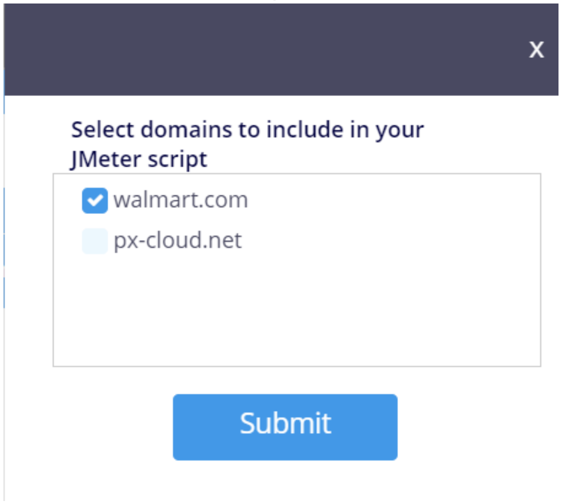

chrome web store
rubia.gulzar280@gmail.com
Blazemeter
« Home
Extensions
Themes
Ratings
★★★★★
★★★★☆ & up
Extensions
1 of 1 extensions
ADDED
BlazeMeter
BlazeMeter | The Continuous Testing Platform
Featured
Record Selenium and HTTP traffic to create a load and functional tests in less than 10 minut
★★★★☆ 145 Developer Tools

3. Login to Blazemeter in your browser extension



Set advanced options as required and click on Start recording.

4. Click on a few links on your site pages, it will record all the actions.
5. Stop the recording. (You can also pause the recording)
6. Go back to the extension and run the recording, select the domain and click on submit button.



x

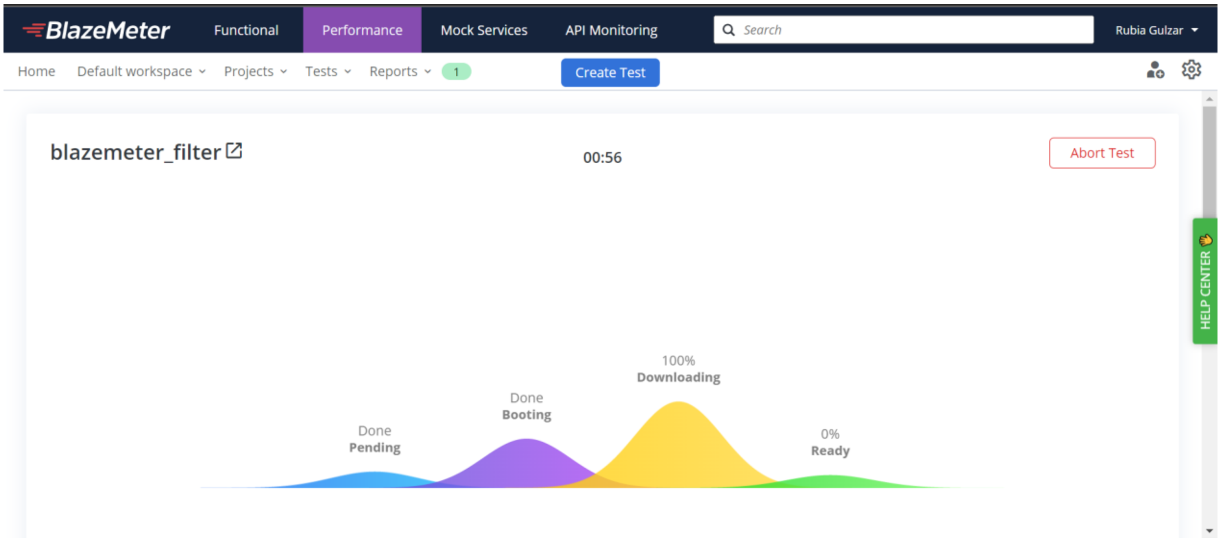
Select domains to include in your JMeter script

☑ walmart.com

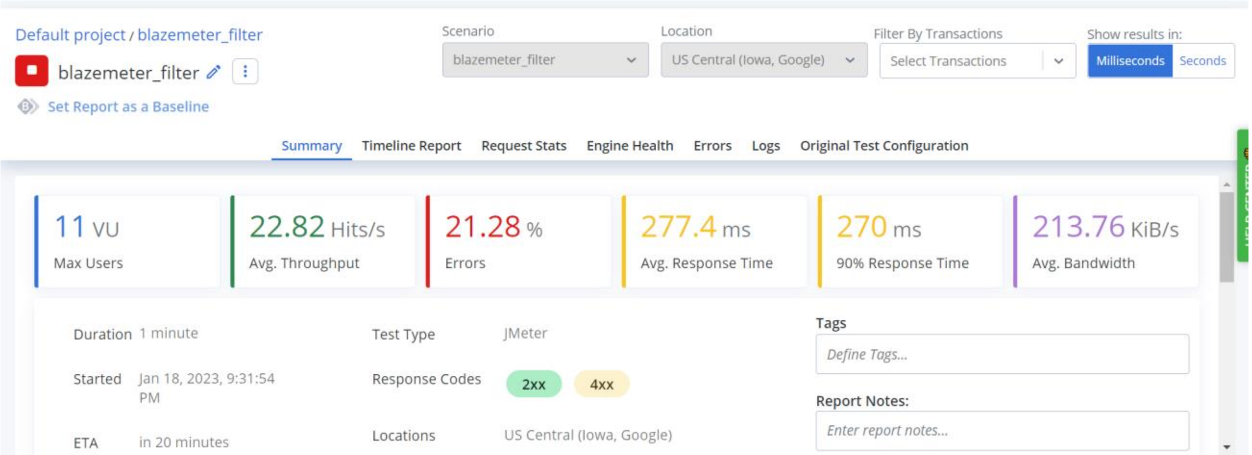
□ px-cloud.net

Submit

7. It will start execution on the Blazemeter dashboard.



Here shows the result



Scenario: blazemeter_filter

Location: US Central (Iowa, Google)

Filter By Transactions: Select Transactions

Show results in: Milliseconds

11 VU	22.82 Hits/s	21.28 %	277.4 ms	270 ms	213.76 KiB/s
Max Users	Avg. Throughput	Errors	Avg. Response Time	90% Response Time	Avg. Bandwidth

Duration 1 minute

Started Jan 18, 2023, 9:31:54 PM

ETA in 20 minutes

Test Type JMeter

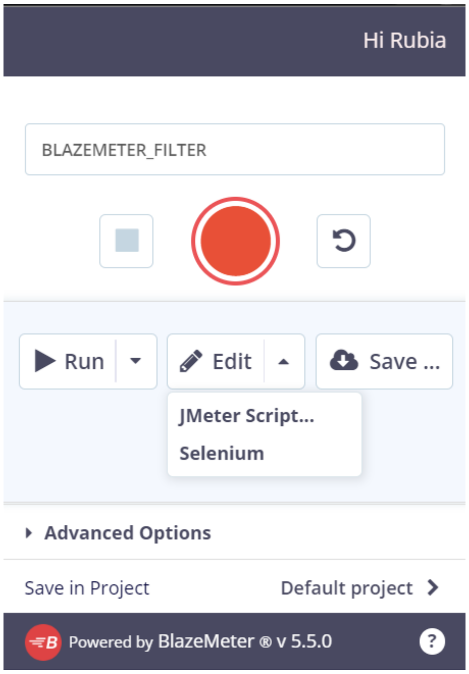
Response Codes 2xx 4xx

Locations US Central (Iowa, Google)

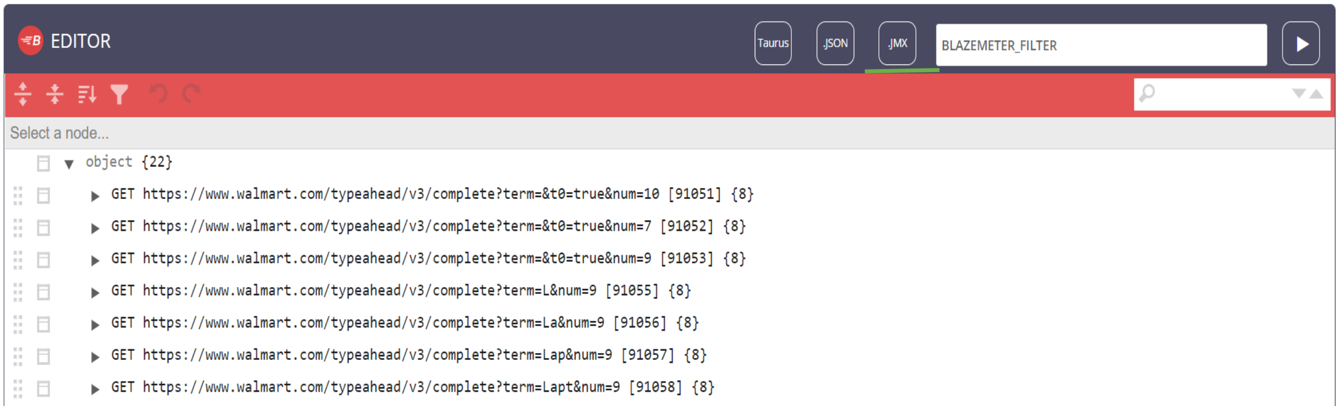
Tags: Define Tags...

Report Notes: Enter report notes...

8. Now go back to the extension and click on Jmeter Script from Edit dropdown

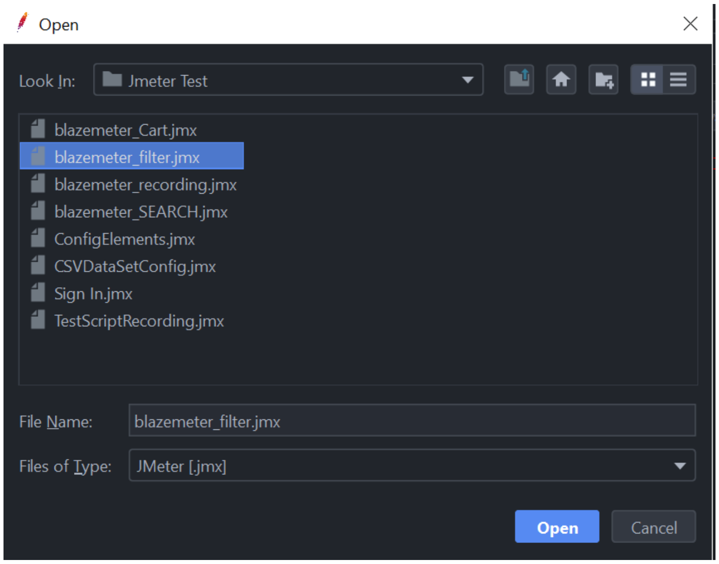


9. Download .jmx file from here



```
object {22}
    GET https://www.walmart.com/typeahead/v3/complete?term=&t0=true&num=10 [91051] {8}
    GET https://www.walmart.com/typeahead/v3/complete?term=&t0=true&num=7 [91052] {8}
    GET https://www.walmart.com/typeahead/v3/complete?term=&t0=true&num=9 [91053] {8}
    GET https://www.walmart.com/typeahead/v3/complete?term=L&num=9 [91055] {8}
    GET https://www.walmart.com/typeahead/v3/complete?term=La&num=9 [91056] {8}
    GET https://www.walmart.com/typeahead/v3/complete?term=Lap&num=9 [91057] {8}
    GET https://www.walmart.com/typeahead/v3/complete?term=Lapt&num=9 [91058] {8}
```

10. Open the jmx file in Jmeter.



Open

Look In: Jmeter Test

blazemeter_Cart.jmx
blazemeter_filter.jmx
blazemeter_recording.jmx
blazemeter_SEARCH.jmx
ConfigElements.jmx
CSVDataSetConfig.jmx
Sign In.jmx
TestScriptRecording.jmx

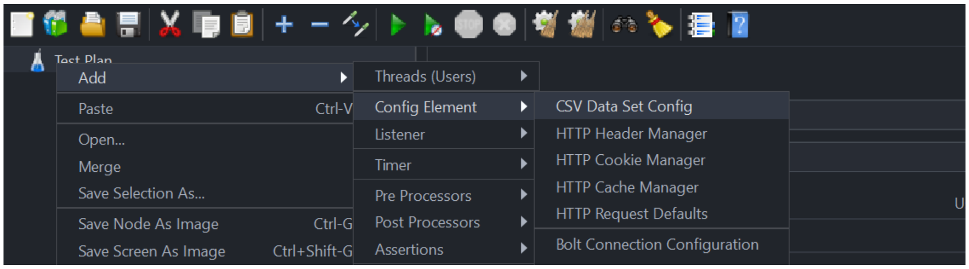
File Name: blazemeter_filter.jmx

Files of Type: JMeter [.jmx]

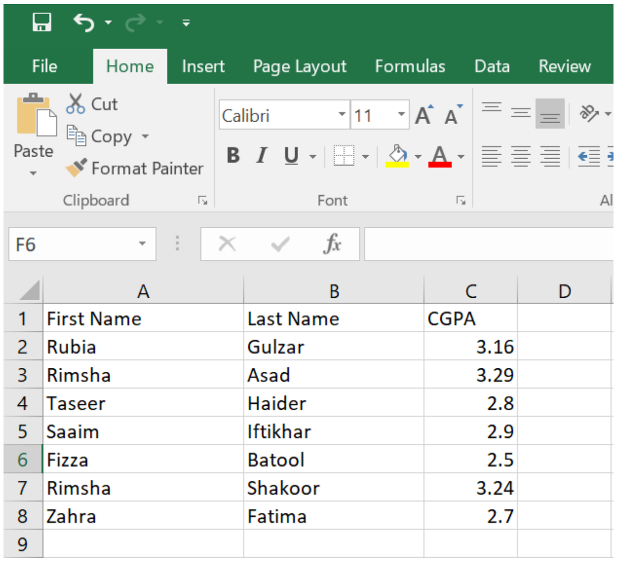
11. Add listeners and assertions then run the test.

Getting Data from CSV File

1. Create new test plan and add CSV Data Set Config

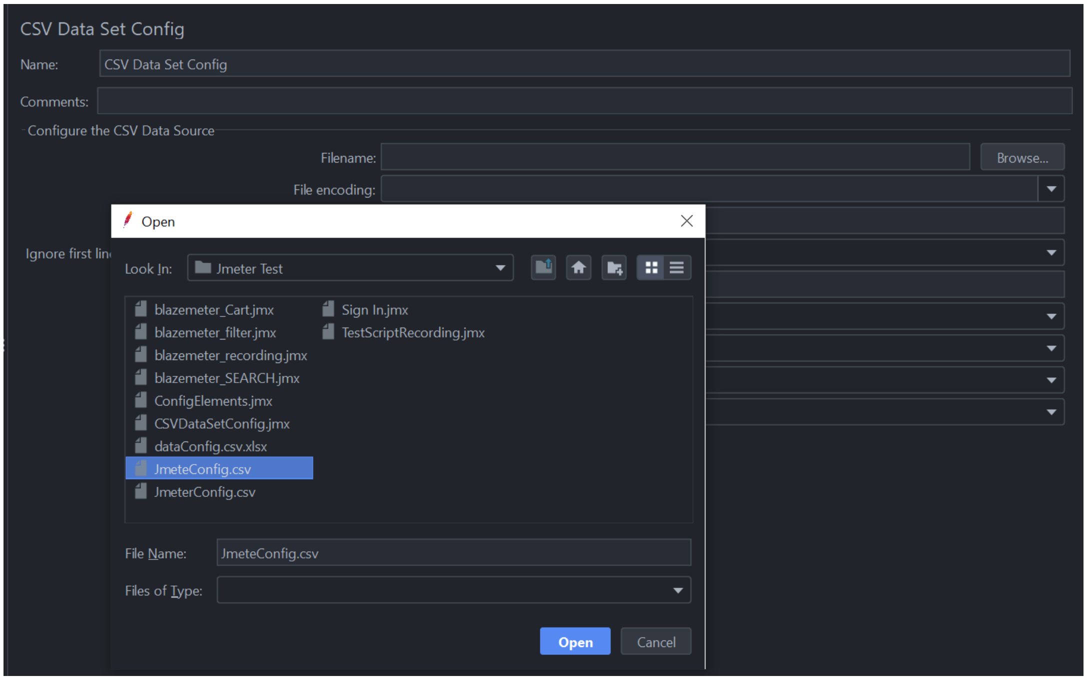


2. Create csv file and add data or used existing file.



	A	B	C	D
1	First Name	Last Name	CGPA	
2	Rubia	Gulzar	3.16	
3	Rimsha	Asad	3.29	
4	Taseer	Haider	2.8	
5	Saaim	Iftikhar	2.9	
6	Fizza	Batool	2.5	
7	Rimsha	Shakoor	3.24	
8	Zahra	Fatima	2.7	
9				

3. Refer the csv file in Jmeter's csv data set config



CSV Data Set Config

Name: CSV Data Set Config

Comments:

Configure the CSV Data Source

Filename:

File encoding:

Ignore first lin

Open

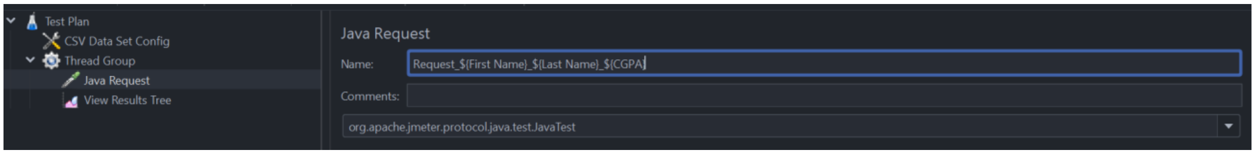
Look In: Jmeter Test

- blazemeter_Cart.jmx
- blazemeter_filter.jmx
- blazemeter_recording.jmx
- blazemeter_SEARCH.jmx
- ConfigElements.jmx
- CSVDataSetConfig.jmx
- dataConfig.csv.xlsx
- JmeteConfig.csv
- JmeterConfig.csv
- Sign In.jmx
- TestScriptRecording.jmx

File Name: JmeteConfig.csv

Files of Type:

4. Refer values from csv file using syntax ${variableName}. For this create a thread group, add Java Request sampler and view result tree.
5. Change the Java Request name to Request_${First Name}_${Last Name}_${CGPA)



- Test Plan
  - CSV Data Set Config
  - Thread Group
    - Java Request
    - View Results Tree

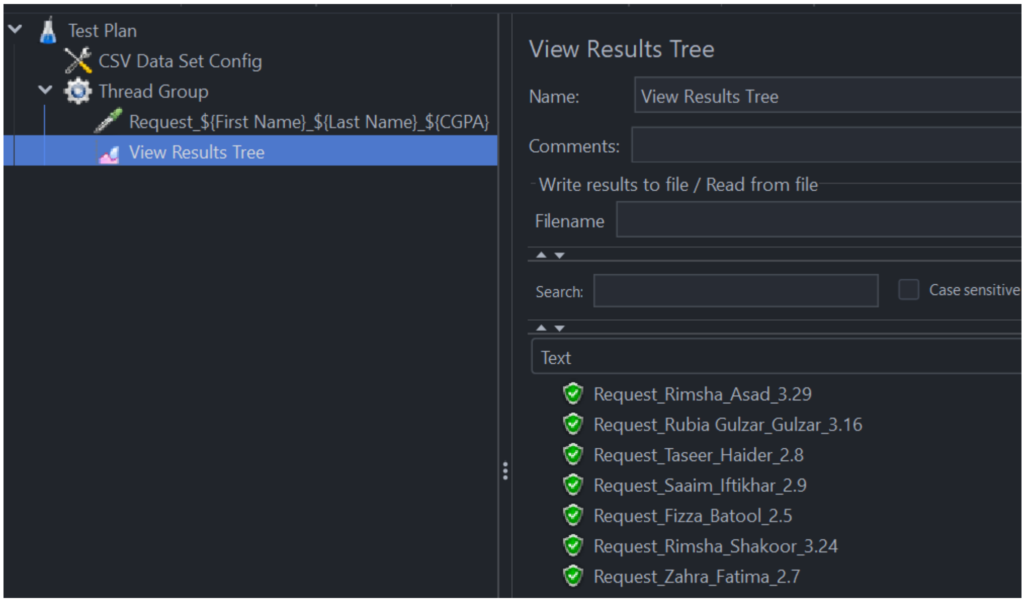
Java Request

Name: Request_${First Name}_${Last Name}_${CGPA}

Comments:

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

6. Run the test by saving it, it shows the below result (getting name and cgpa from file).



- Test Plan
  - CSV Data Set Config
  - Thread Group
    - Request_${First Name}_${Last Name}_${CGPA}
    - View Results Tree

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Search:

□ Case sensitive

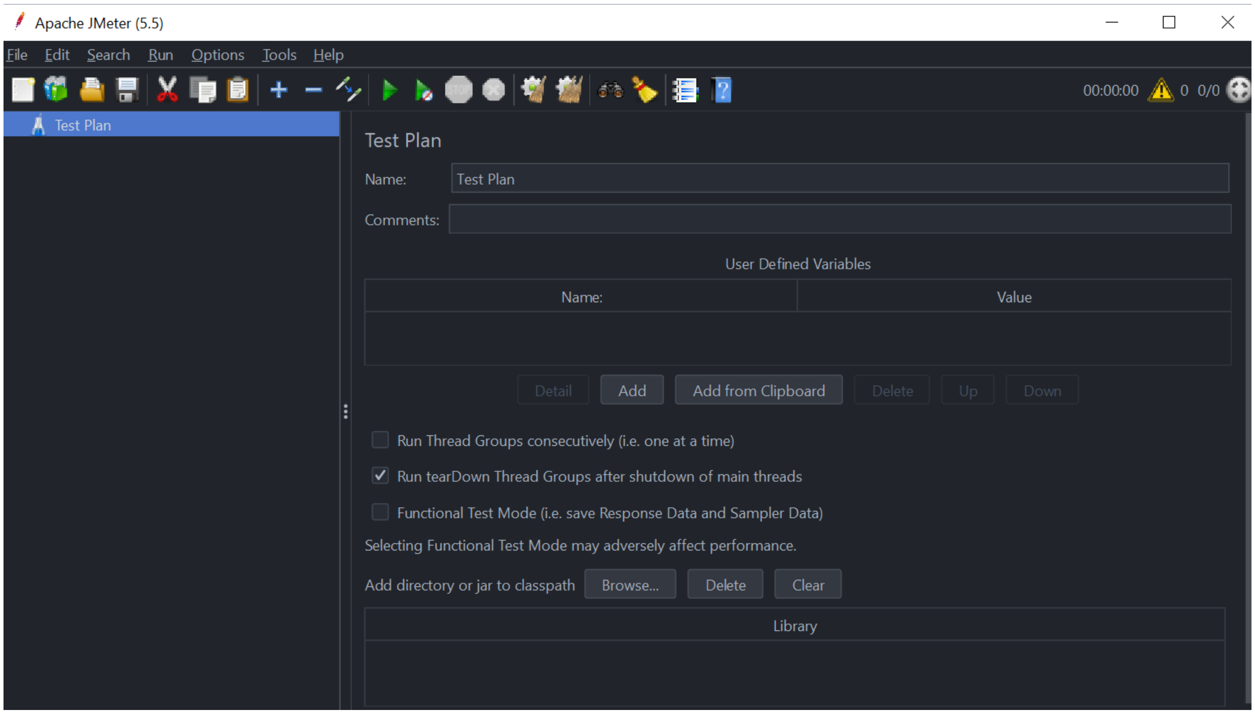
Text

- Request_Rimsha_Asad_3.29
- Request_Rubia Gulzar_Gulzar_3.16
- Request_Taseer_Haider_2.8
- Request_Saaim_Iftikhar_2.9
- Request_Fizza_Batool_2.5
- Request_Rimsha_Shakoor_3.24
- Request_Zahra_Fatima_2.7

API Testing

Steps:

1. Start JMeter and create a Test plan.



Test Plan

Name: Test Plan

Comments:

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

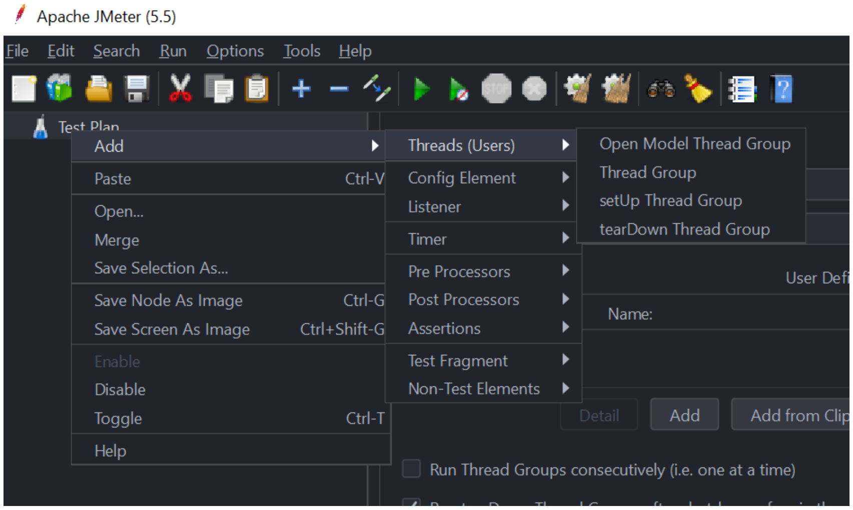
☑ Run tearDown Thread Groups after shutdown of main threads

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

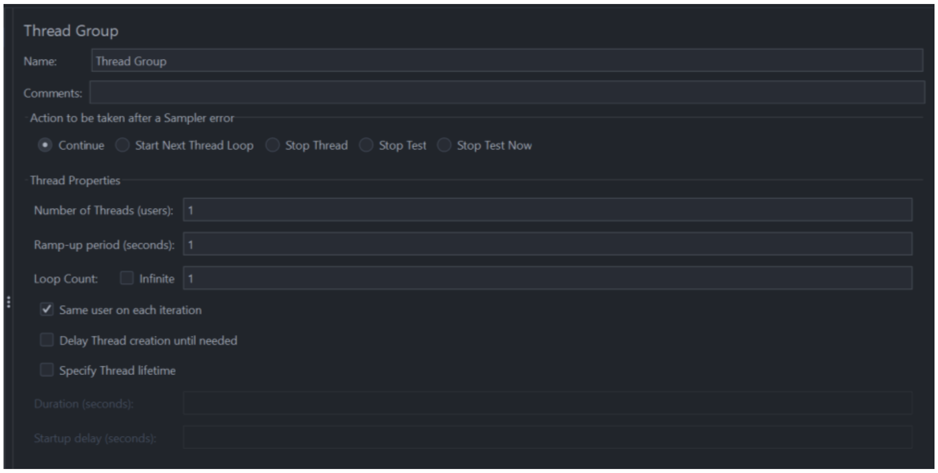
Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

2. Create a Thread Group (Users)



Add number of threads (users) and Ramp-up time (seconds), set loop count (number of iterations to run 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

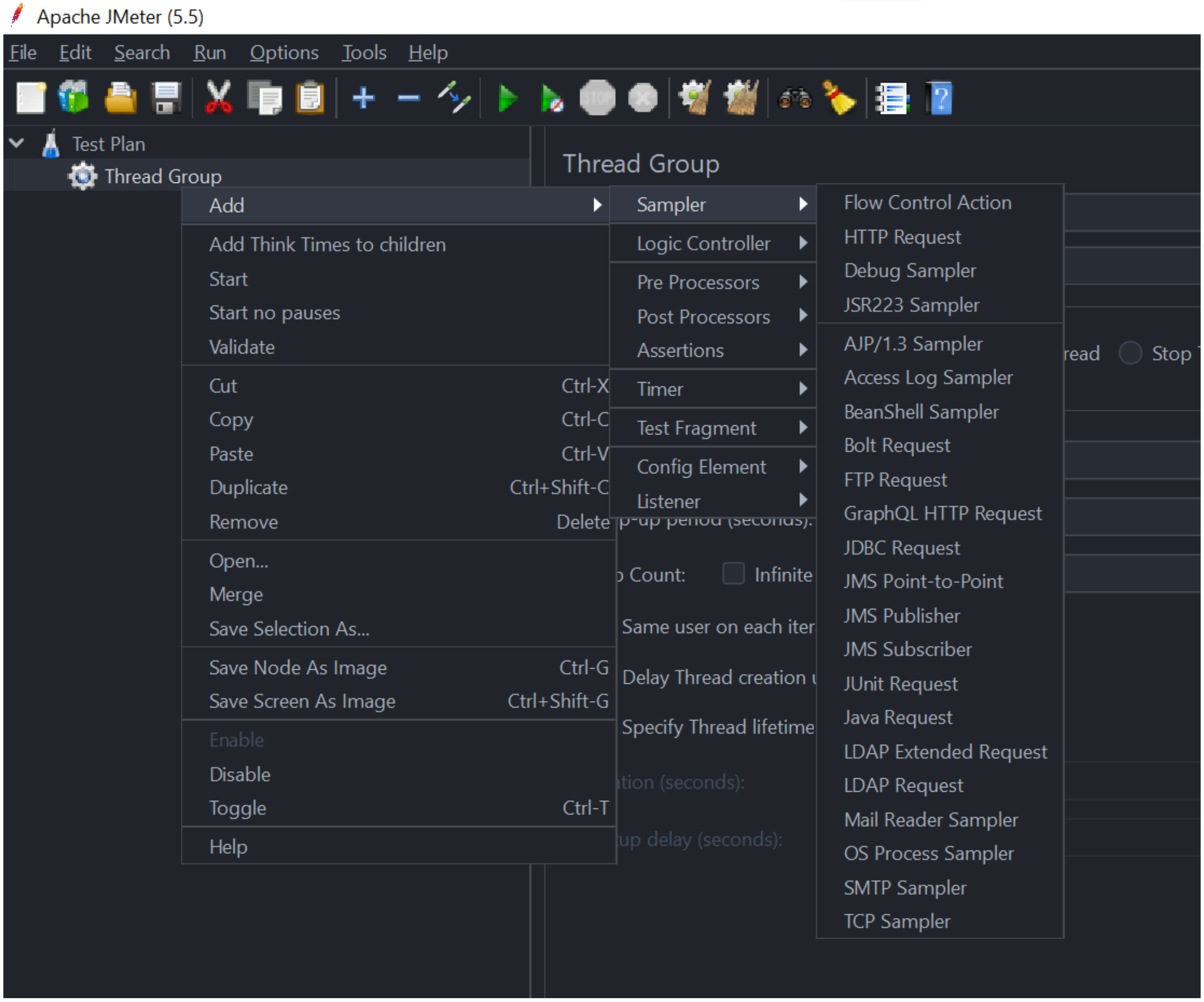
□ Delay Thread creation until needed

□ Specify Thread lifetime

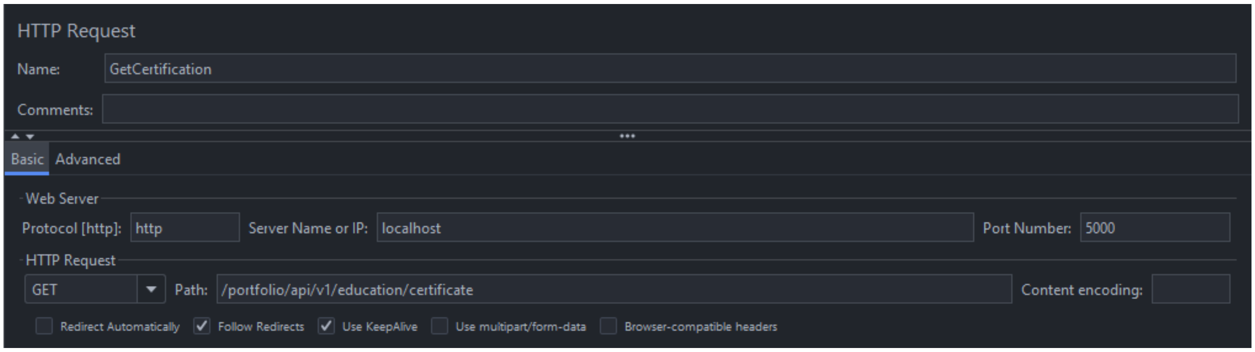
Duration (seconds):

Startup delay (seconds):

3. Add a Sampler (HTTP)



Add Http Request Protocol (if it is http then it's not necessary to write and if its https then write the https) and Server name or IP (localhost), Port Number (optional), http request, path.



HTTP Request

Name: GetCertification

Comments:

Web Server

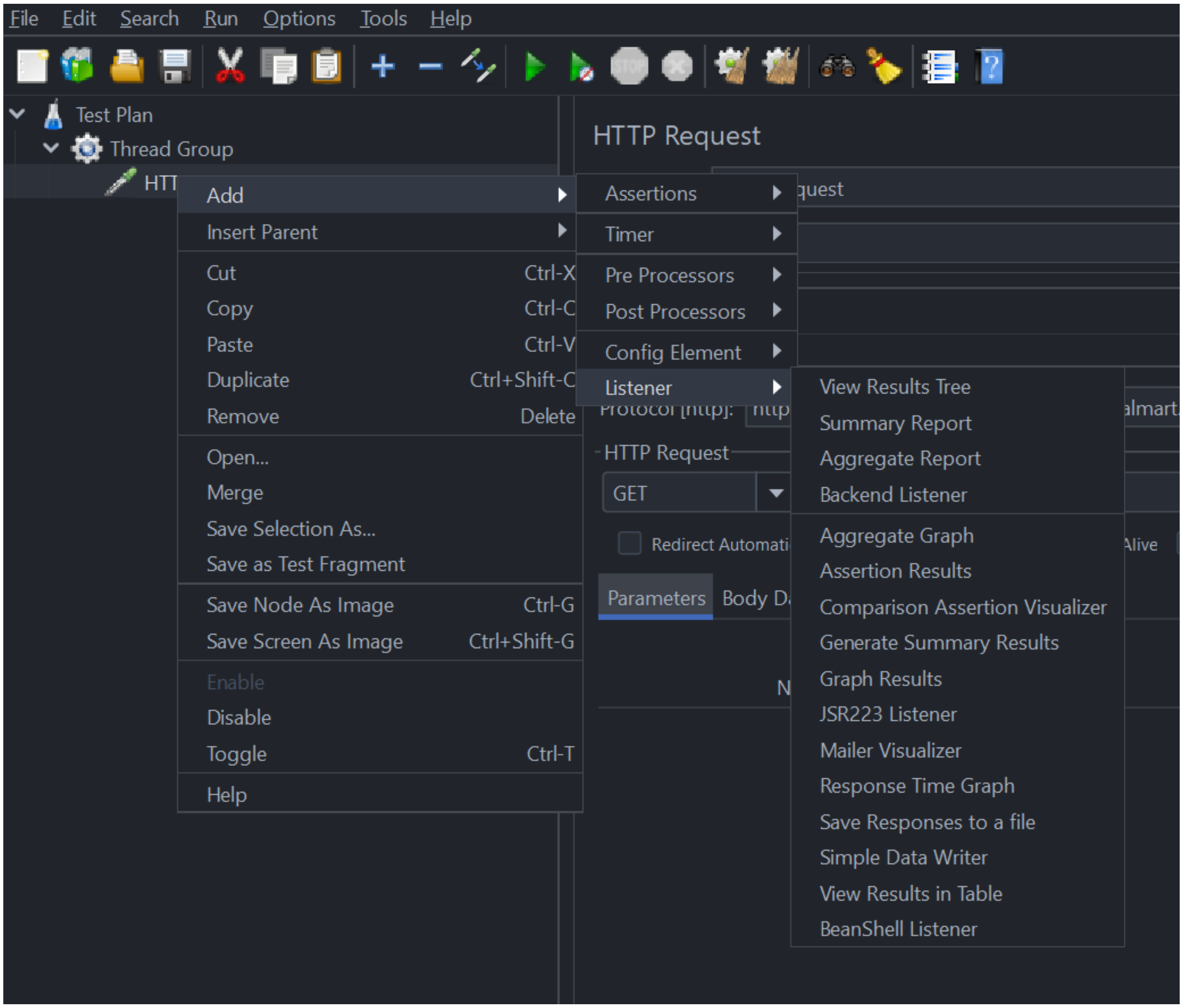
Protocol [http]: http   Server Name or IP: localhost   Port Number: 5000

HTTP Request

GET   Path: /portfolio/api/v1/education/certificate   Content encoding:

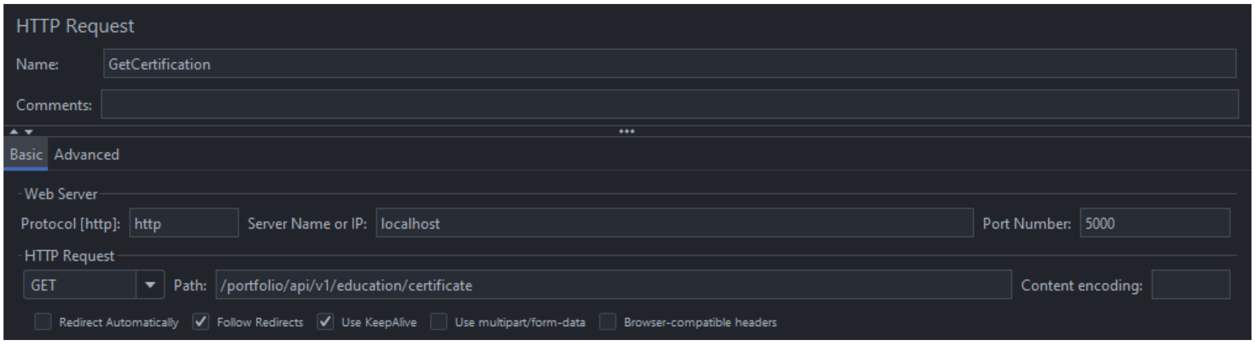
□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

4. Add Listeners



5. Get Certifications:

5.1. HTTP Request



HTTP Request

Name: GetCertification

Comments:

Web Server

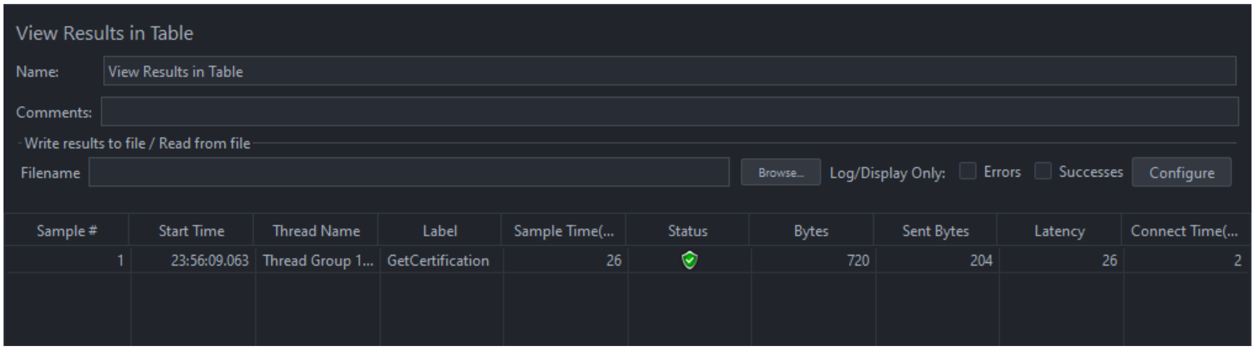
Protocol [http]: http
Server Name or IP: localhost
Port Number: 5000

HTTP Request

GET
Path: /portfolio/api/v1/education/certificate
Content encoding:

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

5.2. Results in table



View Results in Table

Name: View Results in Table

Comments:

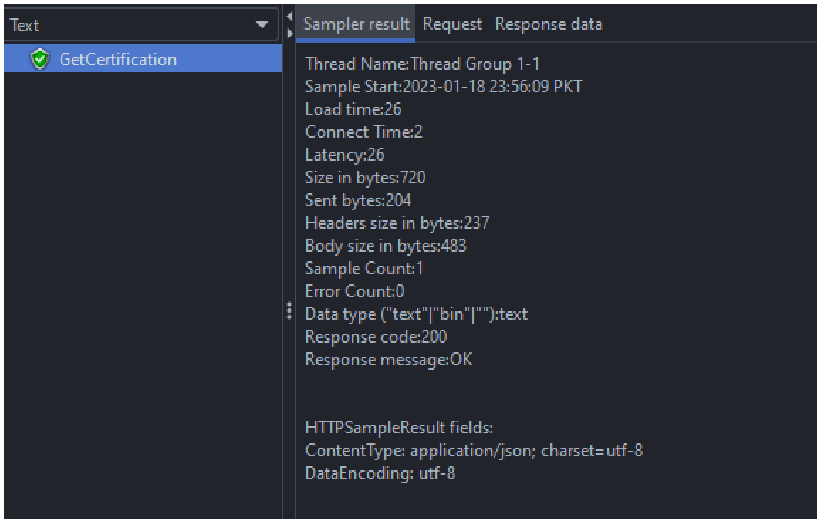
Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	23:56:09.063	Thread Group 1...	GetCertification	26	✓	720	204	26	2

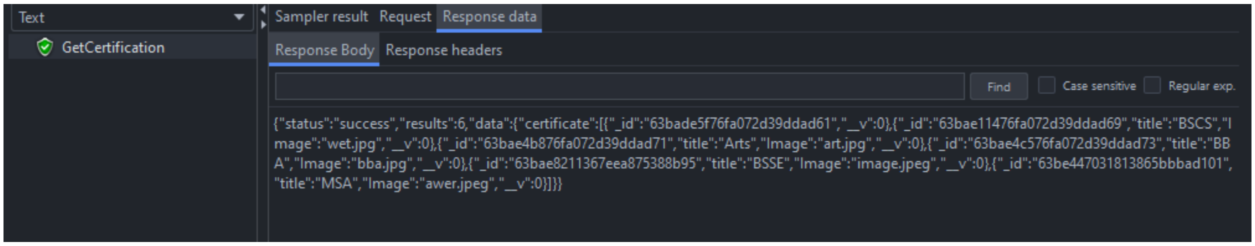
5.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-18 23:56:09 PKT
Load time:26
Connect Time:2
Latency:26
Size in bytes:720
Sent bytes:204
Headers size in bytes:237
Body size in bytes:483
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK


HTTPSampleResult fields:
ContentType: application/json; charset=utf-8
DataEncoding: utf-8
```

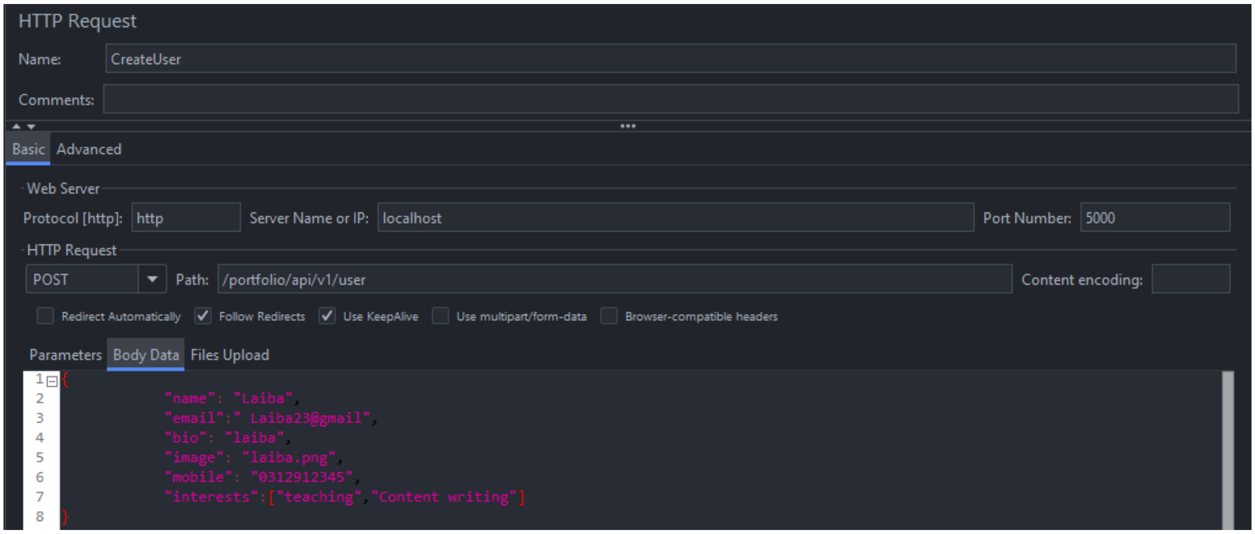
5.4. Results in Tree (Response):



```
{"status":"success","results":6,"data":{"certificate":[{"_id":"63bade5f76fa072d39ddad61","__v":0},{"_id":"63bae11476fa072d39ddad69","title":"BSCS","Image":"wet.jpg","__v":0},{"_id":"63bae4b876fa072d39ddad71","title":"Arts","Image":"art.jpg","__v":0},{"_id":"63bae4c576fa072d39ddad73","title":"BBA","Image":"bba.jpg","__v":0},{"_id":"63bae8211367eea875388b95","title":"BSSE","Image":"image.jpeg","__v":0},{"_id":"63be447031813865bbbad101","title":"MSA","Image":"awer.jpeg","__v":0}]}}
```

6. Create User:

6.1. HTTP Request



HTTP Request

Name: CreateUser

Comments:

Web Server

Protocol [http]: http   Server Name or IP: localhost   Port Number: 5000

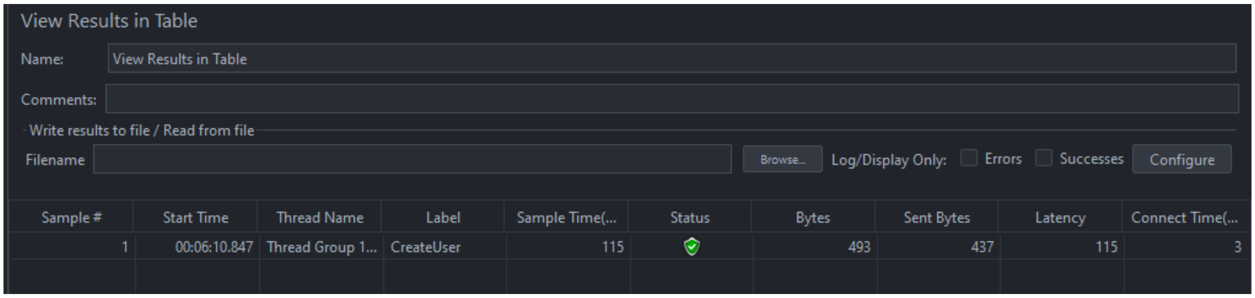
HTTP Request

POST   Path: /portfolio/api/v1/user   Content encoding:

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

```
{
            "name": "Laiba",
            "email":" Laiba23@gmail",
            "bio": "laiba",
            "image": "laiba.png",
            "mobile": "0312912345",
            "interests":["teaching","Content writing"]
}
```

6.2. Results in table



View Results in Table

Name: View Results in Table

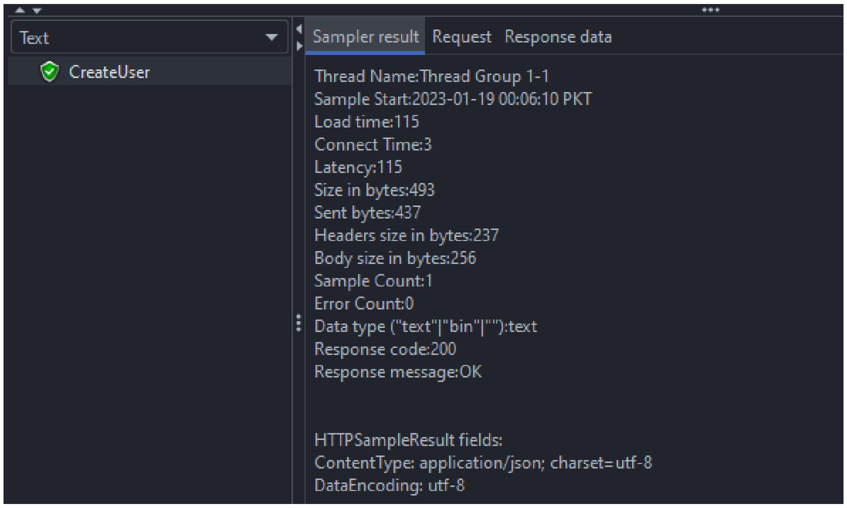
Comments:

Write results to file / Read from file

Filename   Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:06:10.847	Thread Group 1...	CreateUser	115	✓	493	437	115	3

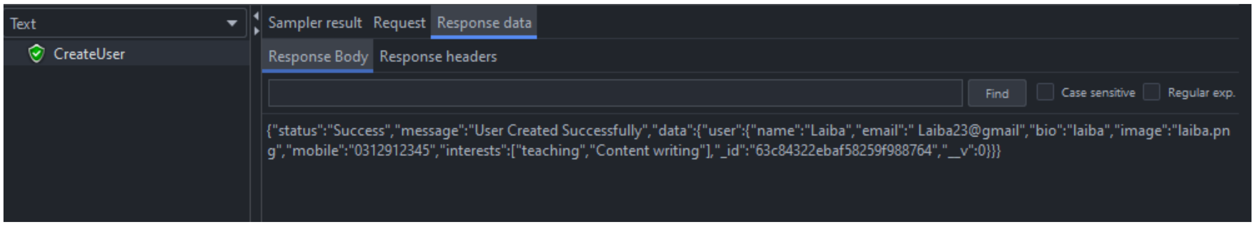
6.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:06:10 PKT
Load time:115
Connect Time:3
Latency:115
Size in bytes:493
Sent bytes:437
Headers size in bytes:237
Body size in bytes:256
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK


HTTPSampleResult fields:
ContentType: application/json; charset=utf-8
DataEncoding: utf-8
```

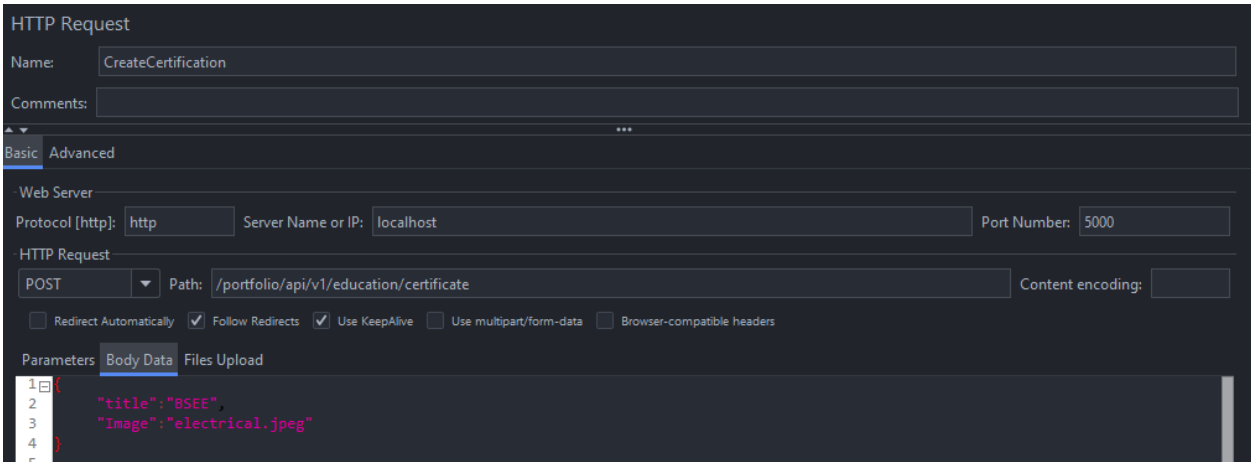
6.4. Results in Tree (Response):



```
{"status":"Success","message":"User Created Successfully","data":{"user":{"name":"Laiba","email":" Laiba23@gmail","bio":"laiba","image":"laiba.pn
g","mobile":"0312912345","interests":["teaching","Content writing"],"_id":"63c84322ebaf58259f988764","__v":0}}}
```

7. Create certification:

7.1. HTTP Request



HTTP Request

Name: CreateCertification

Comments:

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 5000

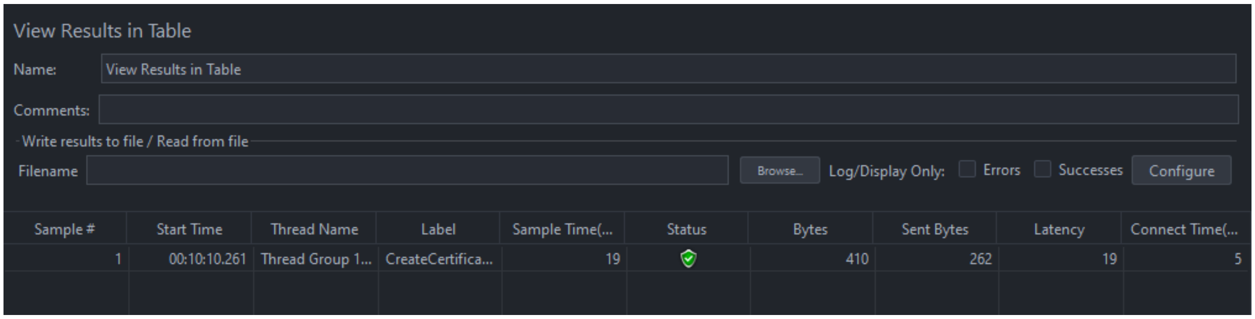
HTTP Request

POST Path: /portfolio/api/v1/education/certificate Content encoding:

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

```
{
    "title":"BSEE",
    "Image":"electrical.jpeg"
}
```

7.2. Results in table



View Results in Table

Name: View Results in Table

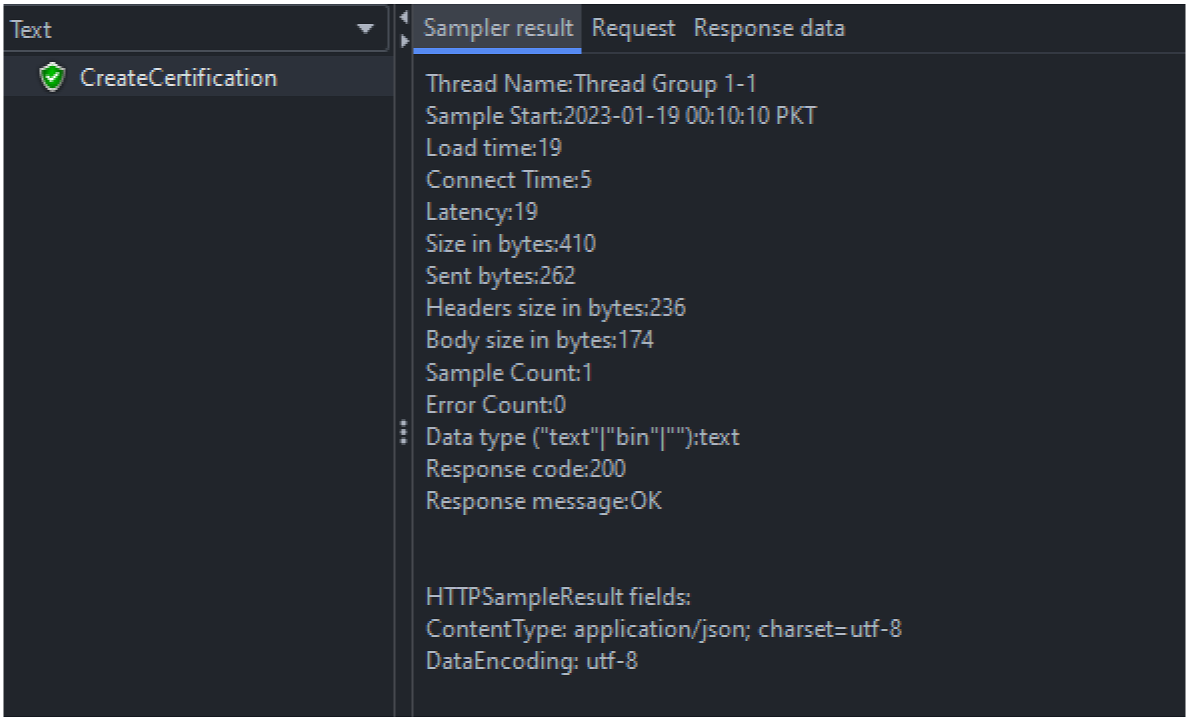
Comments:

Write results to file / Read from file

Filename

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:10:10.261	Thread Group 1...	CreateCertifica...	19	✓	410	262	19	5

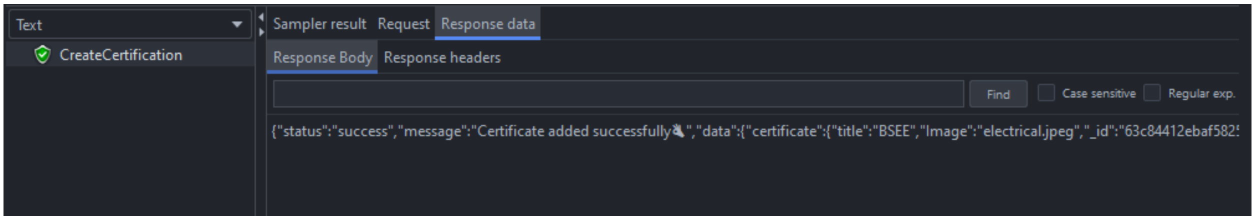
7.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:10:10 PKT
Load time:19
Connect Time:5
Latency:19
Size in bytes:410
Sent bytes:262
Headers size in bytes:236
Body size in bytes:174
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK


HTTPSampleResult fields:
ContentType: application/json; charset=utf-8
DataEncoding: utf-8
```

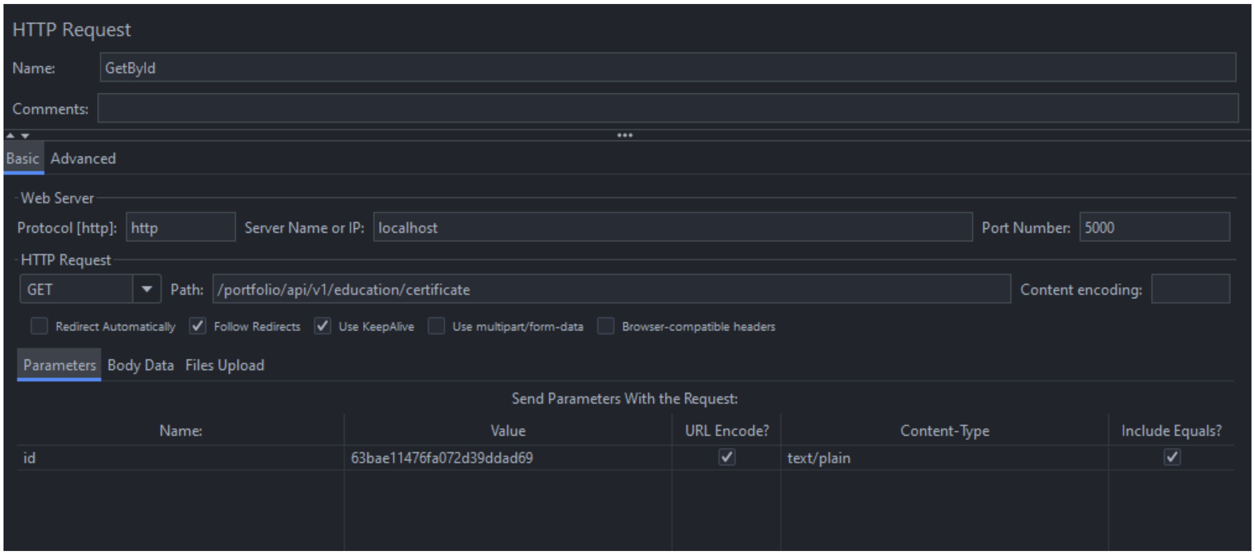
7.4. Results in Tree (Response):



```
{"status":"success","message":"Certificate added successfully🔌","data":{"certificate":{"title":"BSEE","Image":"electrical.jpeg","_id":"63c84412ebaf582
```

8. Get certificate by id:

8.1. HTTP Request



HTTP Request

Name: GetById

Comments:

Web Server

Protocol [http]: http　Server Name or IP: localhost　Port Number: 5000

HTTP Request

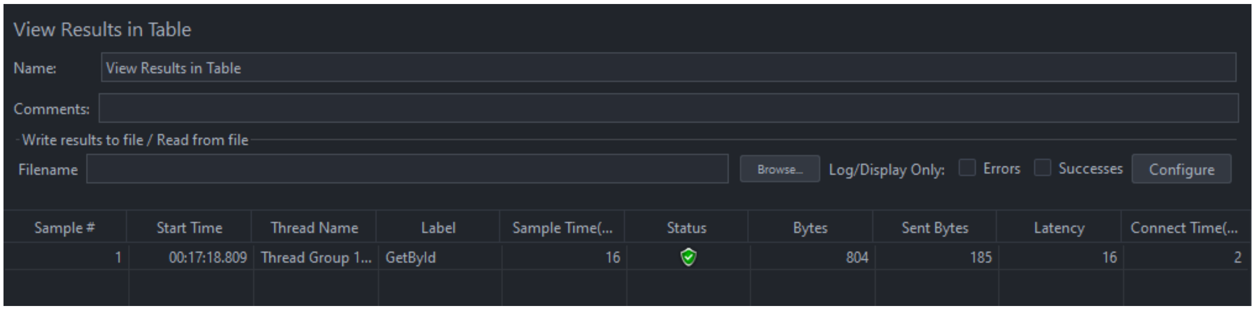
GET　Path: /portfolio/api/v1/education/certificate　Content encoding:

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

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
id	63bae11476fa072d39ddad69	☑	text/plain	☑

8.2. Results in table



View Results in Table

Name: View Results in Table

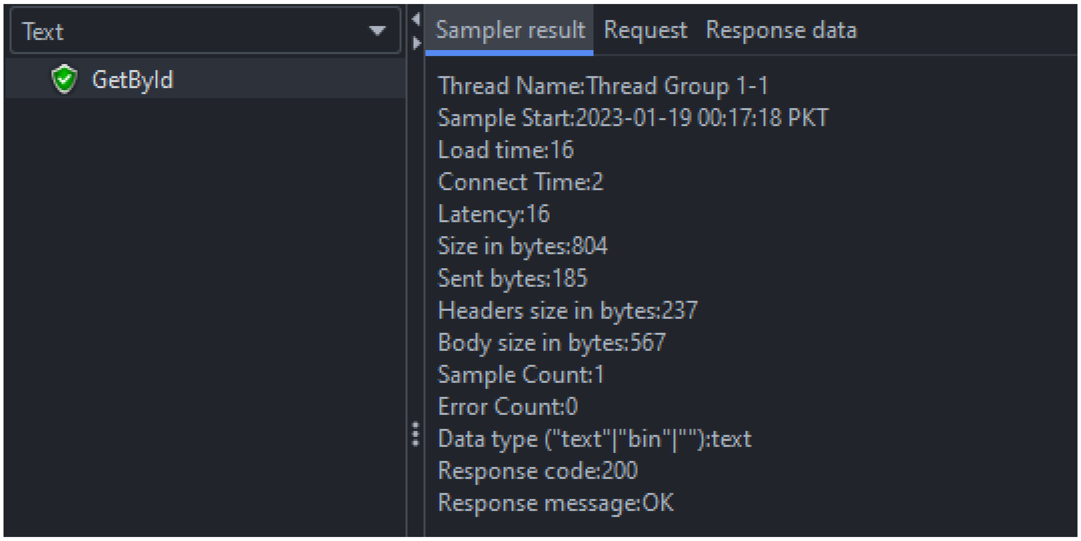
Comments:

Write results to file / Read from file

Filename　Log/Display Only: □ Errors □ Successes

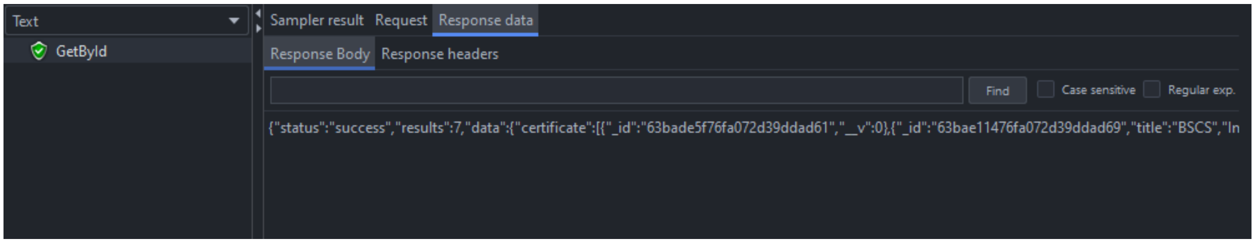
Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:17:18.809	Thread Group 1...	GetById	16	✓	804	185	16	2

8.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:17:18 PKT
Load time:16
Connect Time:2
Latency:16
Size in bytes:804
Sent bytes:185
Headers size in bytes:237
Body size in bytes:567
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK
```

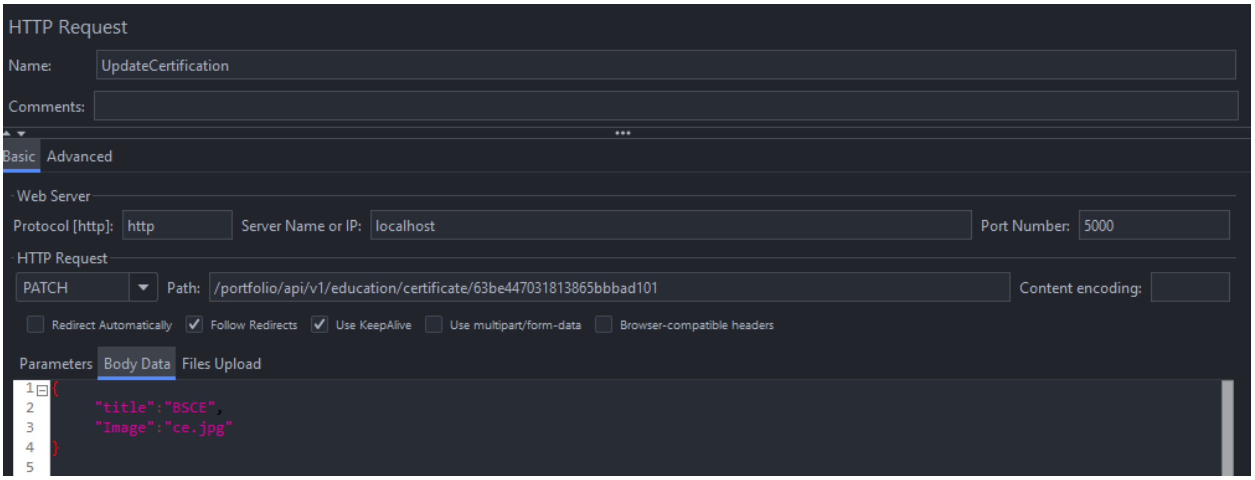
8.4. Results in Tree (Response):



```
{"status":"success","results":7,"data":{"certificate":[{"_id":"63bade5f76fa072d39ddad61","__v":0},{"_id":"63bae11476fa072d39ddad69","title":"BSCS","In
```

9. Update Certification:

9.1. HTTP Request



Name: UpdateCertification

Comments:

Protocol [http]: http Server Name or IP: localhost Port Number: 5000

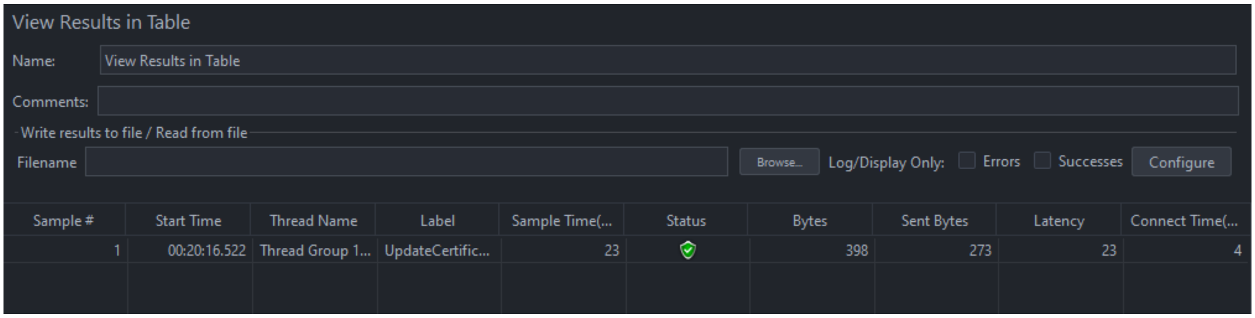
PATCH Path: /portfolio/api/v1/education/certificate/63be447031813865bbbad101 Content encoding:

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

```
{
    "title":"BSCE",
    "Image":"ce.jpg"
}

```

9.2. Results in table



Name: View Results in Table

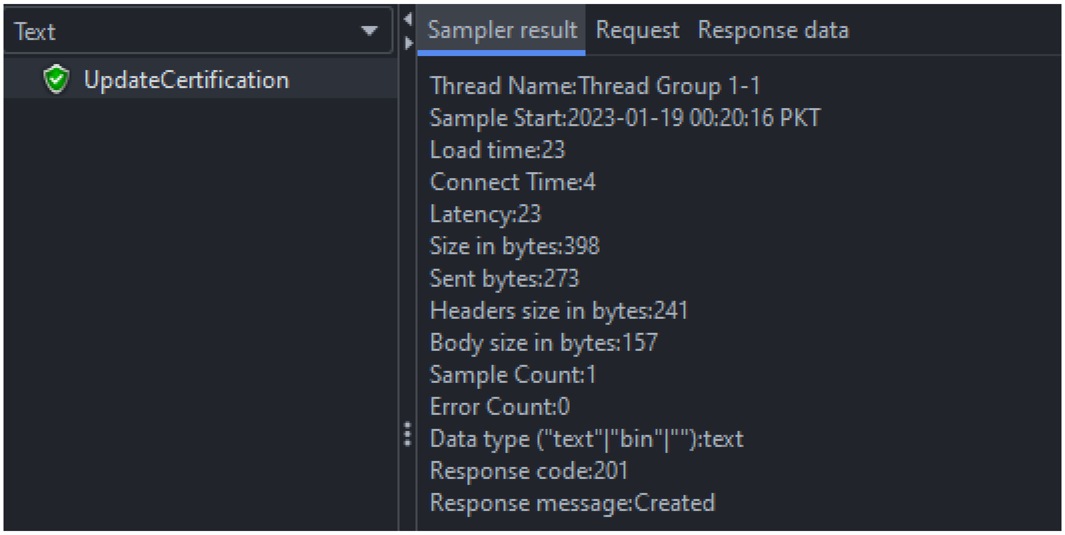
Comments:

Filename

Log/Display Only: □ Errors □ Successes

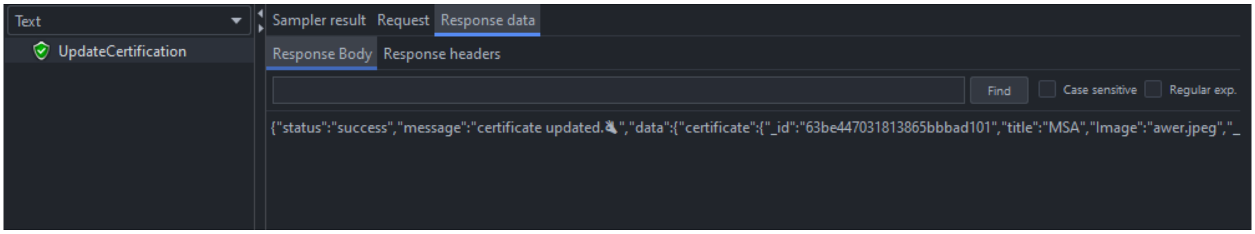
Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:20:16.522	Thread Group 1...	UpdateCertific...	23	✓	398	273	23	4

9.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:20:16 PKT
Load time:23
Connect Time:4
Latency:23
Size in bytes:398
Sent bytes:273
Headers size in bytes:241
Body size in bytes:157
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:201
Response message:Created
```

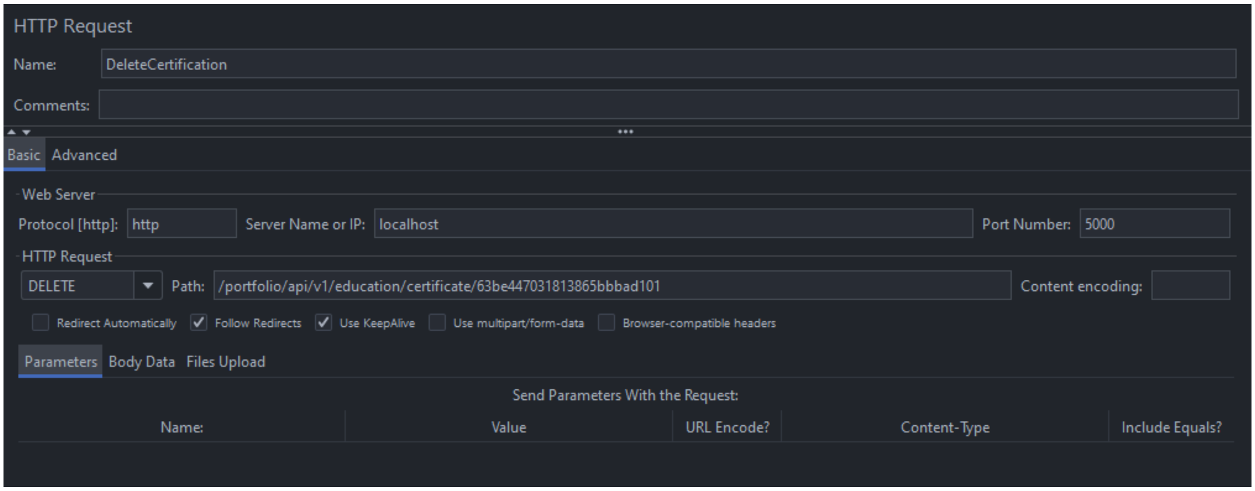
9.4. Results in Tree (Response):



```
{"status":"success","message":"certificate updated.🔨","data":{"certificate":{"_id":"63be447031813865bbbad101","title":"MSA","Image":"awer.jpeg","_
```

10. Delete Certification:

10.1. HTTP Request



Name: DeleteCertification

Comments:

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 5000

HTTP Request

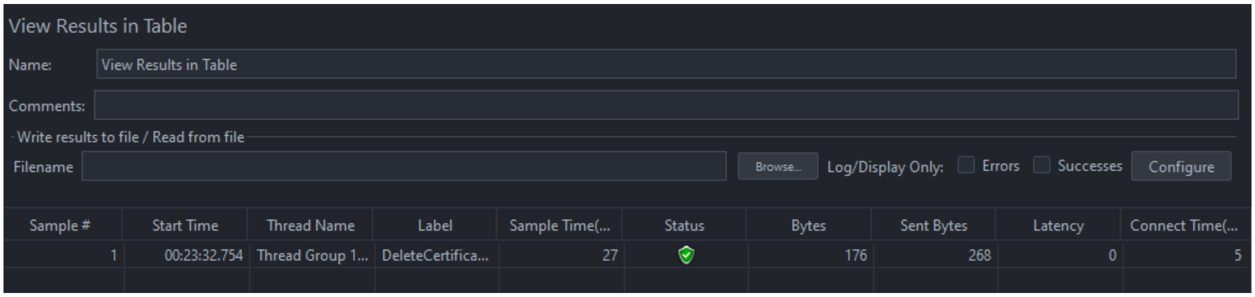
DELETE Path: /portfolio/api/v1/education/certificate/63be447031813865bbbad101 Content encoding:

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

Send Parameters With the Request:

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

10.2. Results in table



View Results in Table

Name: View Results in Table

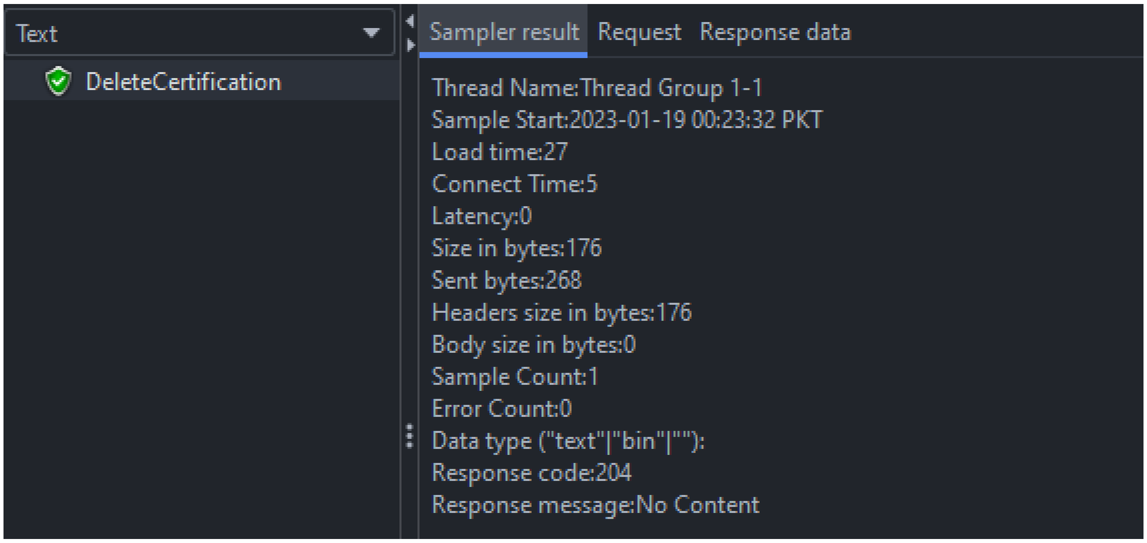
Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:23:32.754	Thread Group 1...	DeleteCertifica...	27	✓	176	268	0	5

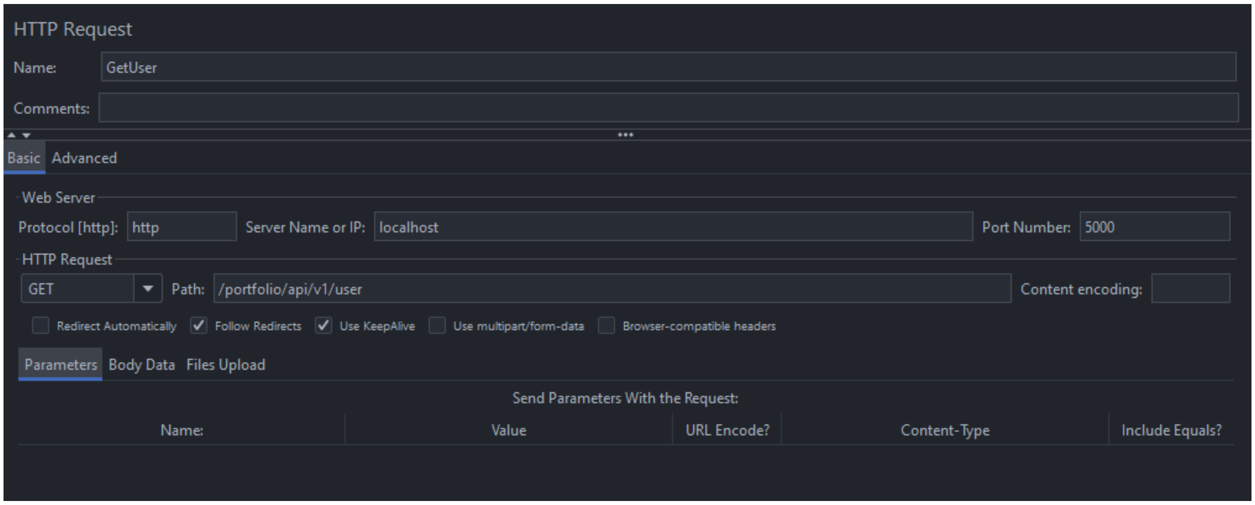
10.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:23:32 PKT
Load time:27
Connect Time:5
Latency:0
Size in bytes:176
Sent bytes:268
Headers size in bytes:176
Body size in bytes:0
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):
Response code:204
Response message:No Content
```

11. Get Users:

11.1. HTTP Request



HTTP Request

Name: GetUser

Comments:

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 5000

HTTP Request

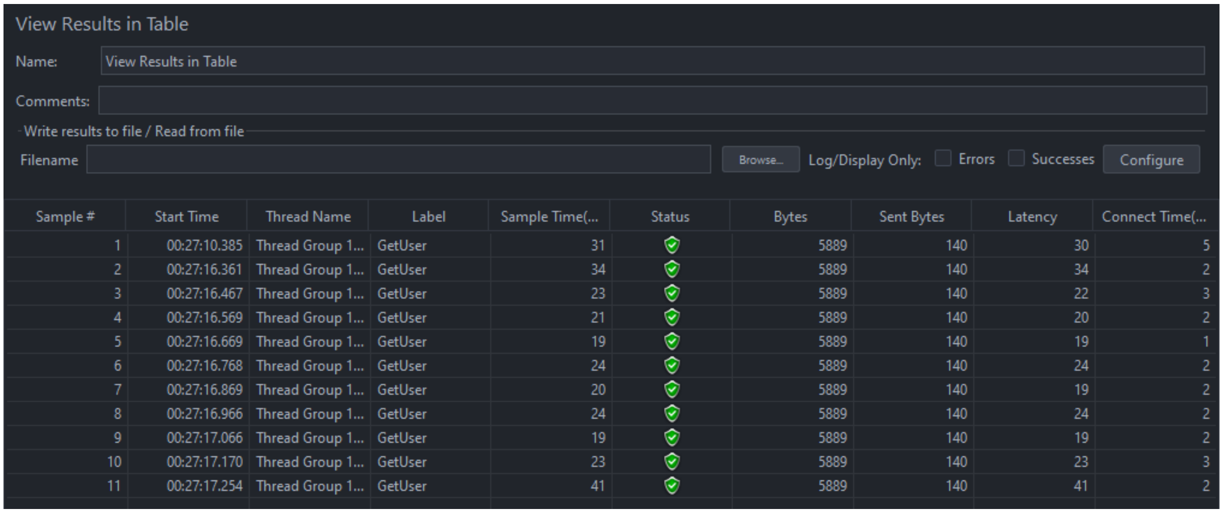
GET Path: /portfolio/api/v1/user Content encoding:

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

Send Parameters With the Request:

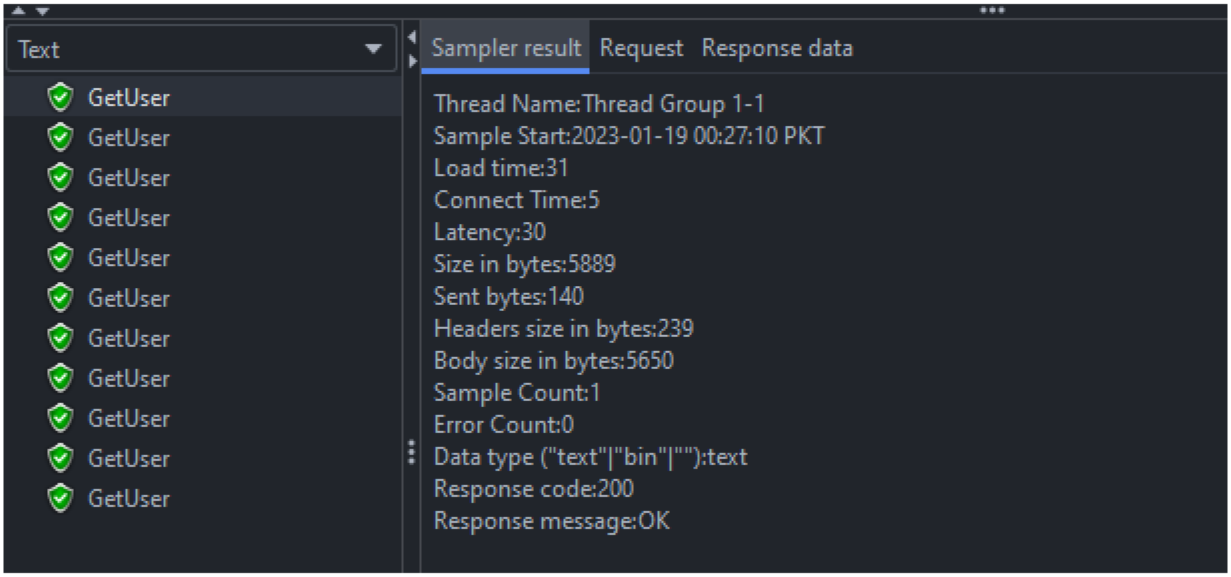
Name:	Value	URL Encode?	Content-Type	Include Equals?

11.2. Results in table



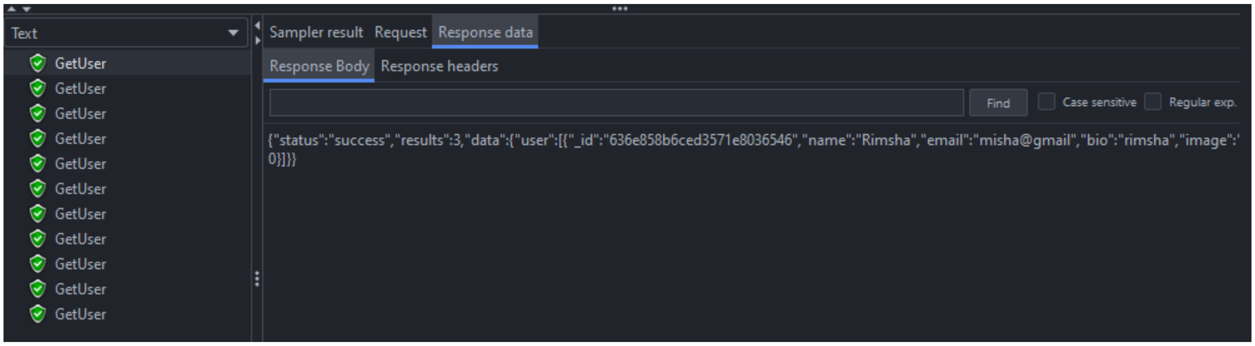
Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:27:10.385	Thread Group 1...	GetUser	31	✓	5889	140	30	5
2	00:27:16.361	Thread Group 1...	GetUser	34	✓	5889	140	34	2
3	00:27:16.467	Thread Group 1...	GetUser	23	✓	5889	140	22	3
4	00:27:16.569	Thread Group 1...	GetUser	21	✓	5889	140	20	2
5	00:27:16.669	Thread Group 1...	GetUser	19	✓	5889	140	19	1
6	00:27:16.768	Thread Group 1...	GetUser	24	✓	5889	140	24	2
7	00:27:16.869	Thread Group 1...	GetUser	20	✓	5889	140	19	2
8	00:27:16.966	Thread Group 1...	GetUser	24	✓	5889	140	24	2
9	00:27:17.066	Thread Group 1...	GetUser	19	✓	5889	140	19	2
10	00:27:17.170	Thread Group 1...	GetUser	23	✓	5889	140	23	3
11	00:27:17.254	Thread Group 1...	GetUser	41	✓	5889	140	41	2

11.3. Results in Tree



```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:27:10 PKT
Load time:31
Connect Time:5
Latency:30
Size in bytes:5889
Sent bytes:140
Headers size in bytes:239
Body size in bytes:5650
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK
```

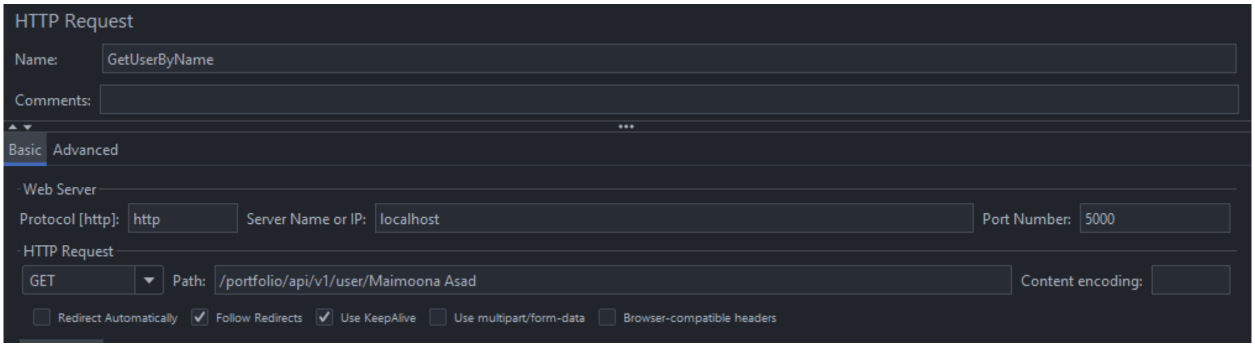
11.4. Results in Tree (Response):



```
{"status":"success","results":3,"data":{"user":[{"_id":"636e858b6ced3571e8036546","name":"Rimsha","email":"misha@gmail","bio":"rimsha","image":'
0}]}}
```

12. Get user by name:

12.1. HTTP Request



HTTP Request

Name: GetUserByName

Comments:

Basic | Advanced

Web Server

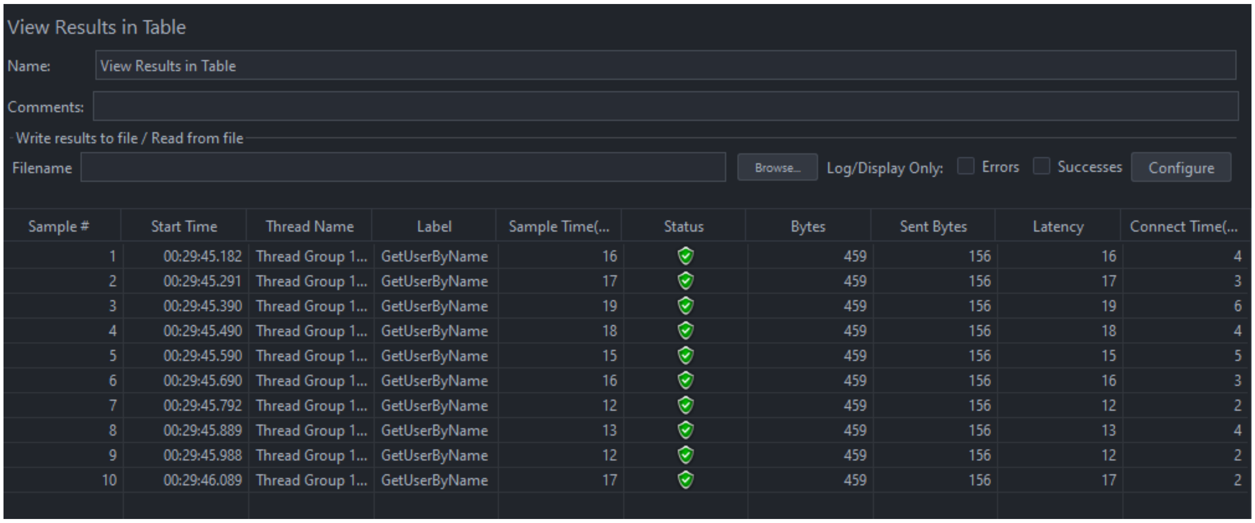
Protocol [http]: http   Server Name or IP: localhost   Port Number: 5000

HTTP Request

GET   Path: /portfolio/api/v1/user/Maimoona Asad   Content encoding:

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

12.2. Results in table



View Results in Table

Name: View Results in Table

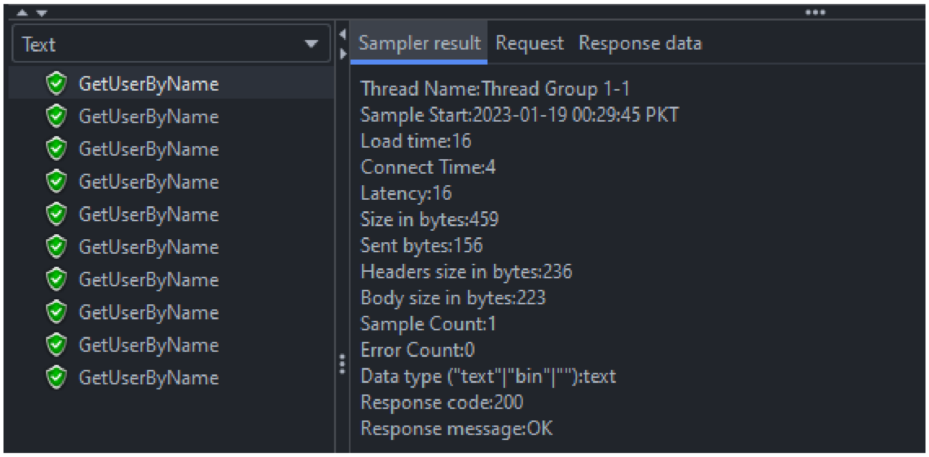
Comments:

Write results to file / Read from file

Filename   Browse...   Log/Display Only: □ Errors □ Successes   Configure

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:29:45.182	Thread Group 1...	GetUserByName	16	✓	459	156	16	4
2	00:29:45.291	Thread Group 1...	GetUserByName	17	✓	459	156	17	3
3	00:29:45.390	Thread Group 1...	GetUserByName	19	✓	459	156	19	6
4	00:29:45.490	Thread Group 1...	GetUserByName	18	✓	459	156	18	4
5	00:29:45.590	Thread Group 1...	GetUserByName	15	✓	459	156	15	5
6	00:29:45.690	Thread Group 1...	GetUserByName	16	✓	459	156	16	3
7	00:29:45.792	Thread Group 1...	GetUserByName	12	✓	459	156	12	2
8	00:29:45.889	Thread Group 1...	GetUserByName	13	✓	459	156	13	4
9	00:29:45.988	Thread Group 1...	GetUserByName	12	✓	459	156	12	2
10	00:29:46.089	Thread Group 1...	GetUserByName	17	✓	459	156	17	2

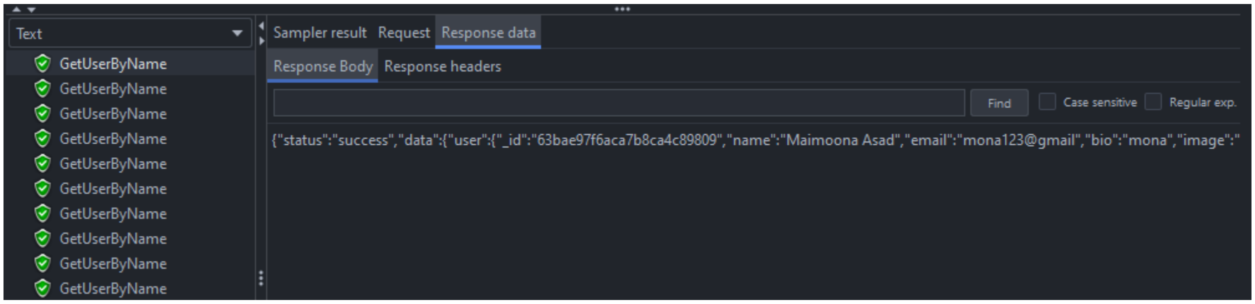
12.3. Results in Tree



✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName
✓ GetUserByName

```
Thread Name:Thread Group 1-1
Sample Start:2023-01-19 00:29:45 PKT
Load time:16
Connect Time:4
Latency:16
Size in bytes:459
Sent bytes:156
Headers size in bytes:236
Body size in bytes:223
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK
```

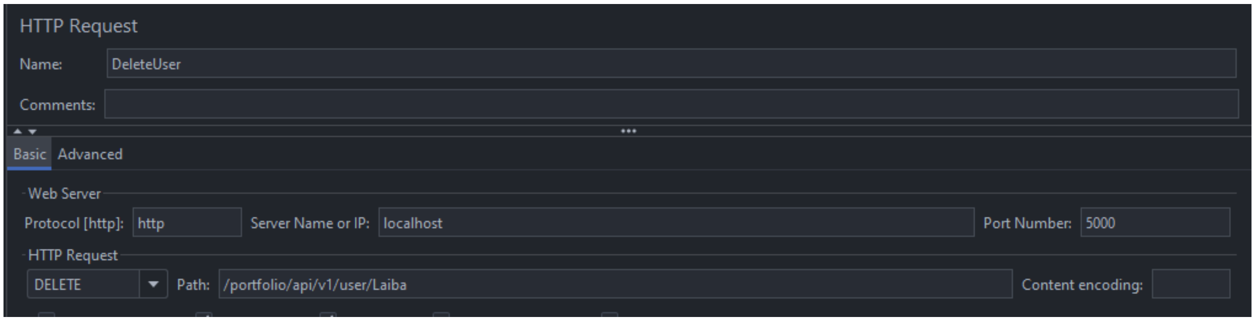
12.4. Results in Tree (Response):



```
{"status":"success","data":{"user":{"_id":"63bae97f6aca7b8ca4c89809","name":"Maimoona Asad","email":"mona123@gmail","bio":"mona","image":"
```

13. Delete User:

13.1. HTTP Request



HTTP Request

Name: DeleteUser

Comments:

Protocol [http]: http

Server Name or IP: localhost

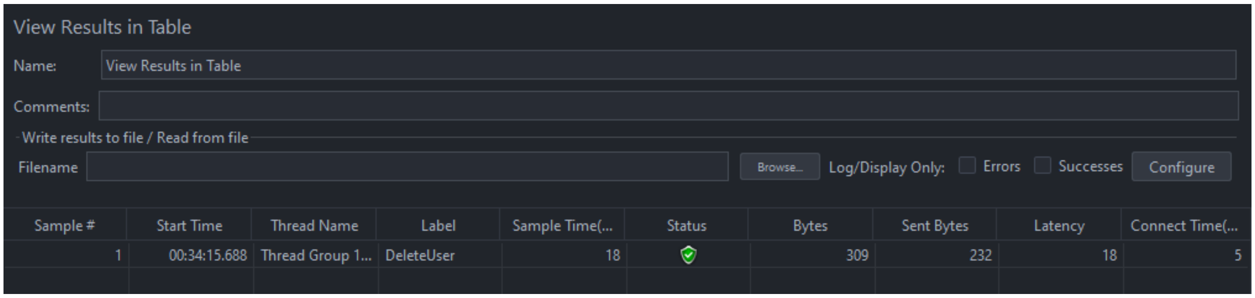
Port Number: 5000

DELETE

Path: /portfolio/api/v1/user/Laiba

Content encoding:

13.2. Results in table



View Results in Table

Name: View Results in Table

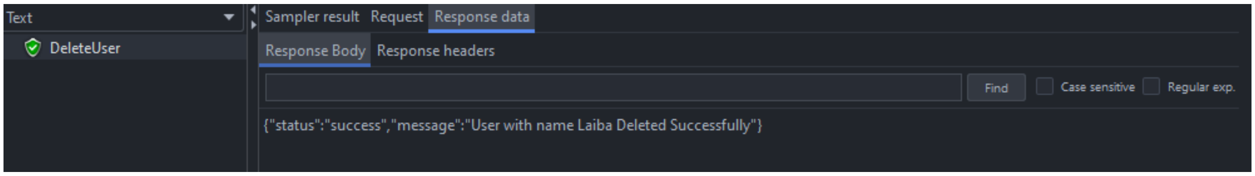
Comments:

Write results to file / Read from file

Filename

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	00:34:15.688	Thread Group 1...	DeleteUser	18	✓	309	232	18	5

13.3. Results in Tree (Response):



```
{"status":"success","message":"User with name Laiba Deleted Successfully"}
```