//Processors, Assertions, and Timers

Development Environment:

- Apache JMeter 5.1.1 Version
- JDK Runtime Environment 8.1

This guide has three subsections, namely:

1.3.1 Problem statement for Processors, Assertions, and Timers

1.3.2 Solution for the problem statement

1.3.3 Pushing the code to GitHub repositories

Step 1.3.1: Problem statement for Processors, Assertions, and Timers

- Objective: Create Processors, Assertions, Timers in Jmeter for a particular domain.
- Steps involved:

1) Set up JMeter.

2) Create a test plan.

3) Add thread group.

4) Add HTTP request.

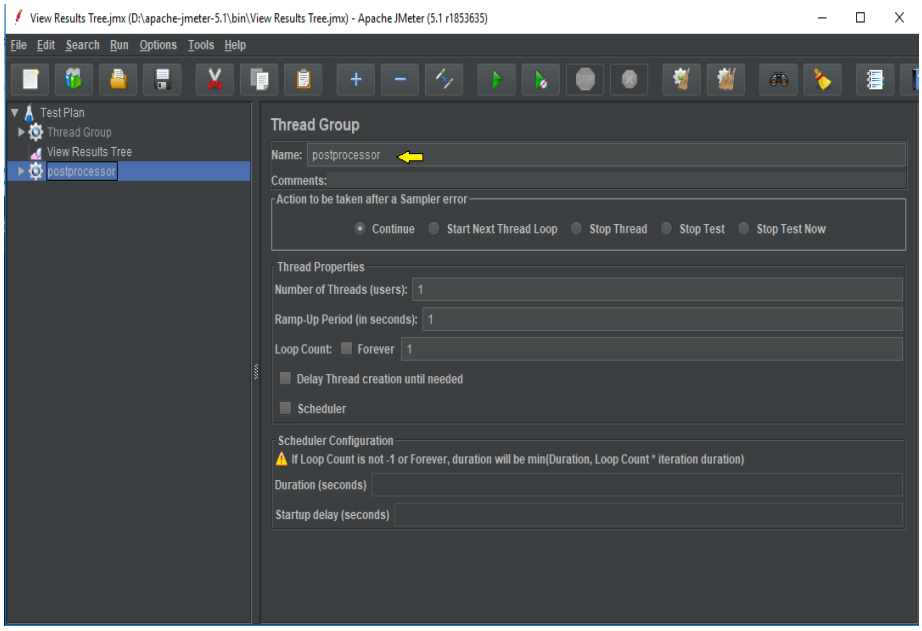
5) Record the scripts.

6) Add Listener and view the results.

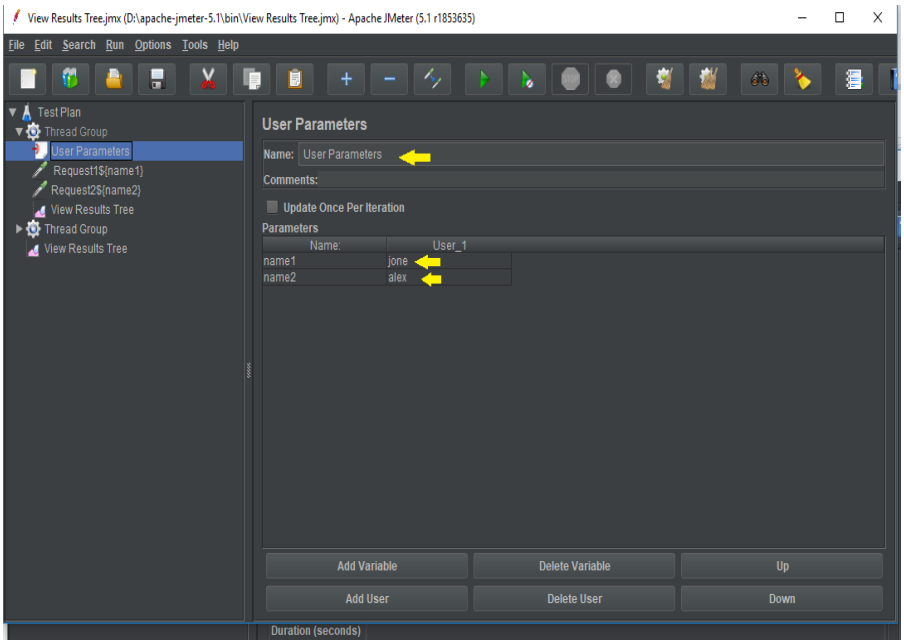
Step 1.3.2: Solution for the problem statement

1) Adding Preprocessor:

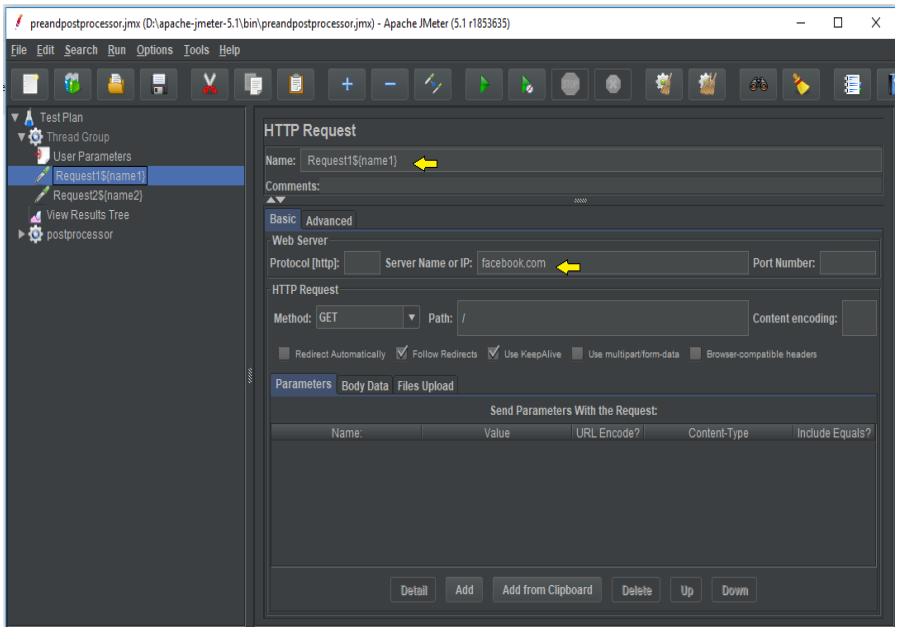
- Right click on test plan-->Add-->Threads-->Thread Group.

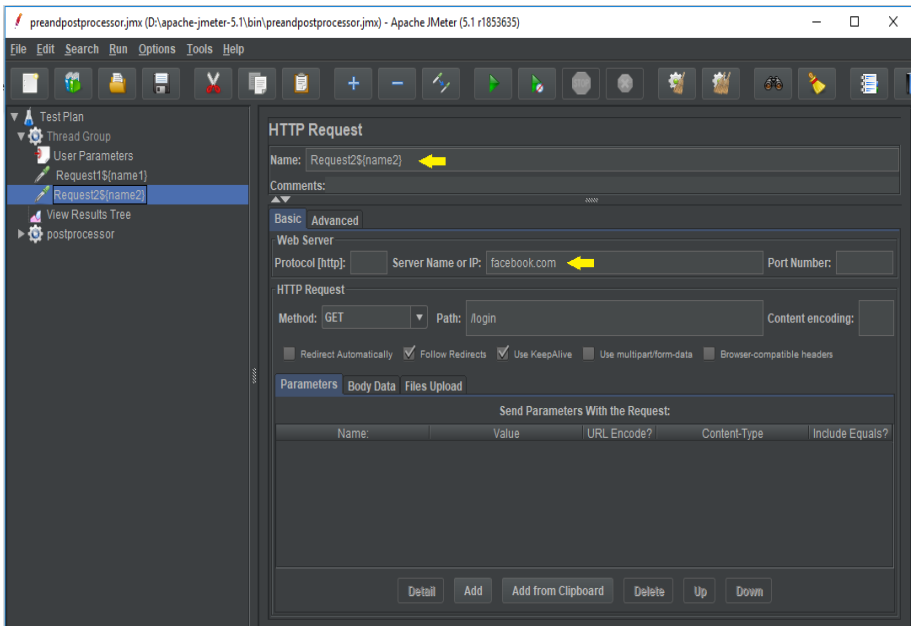


- Right click on Thread-->Add-->preprocessor -->User parameters.



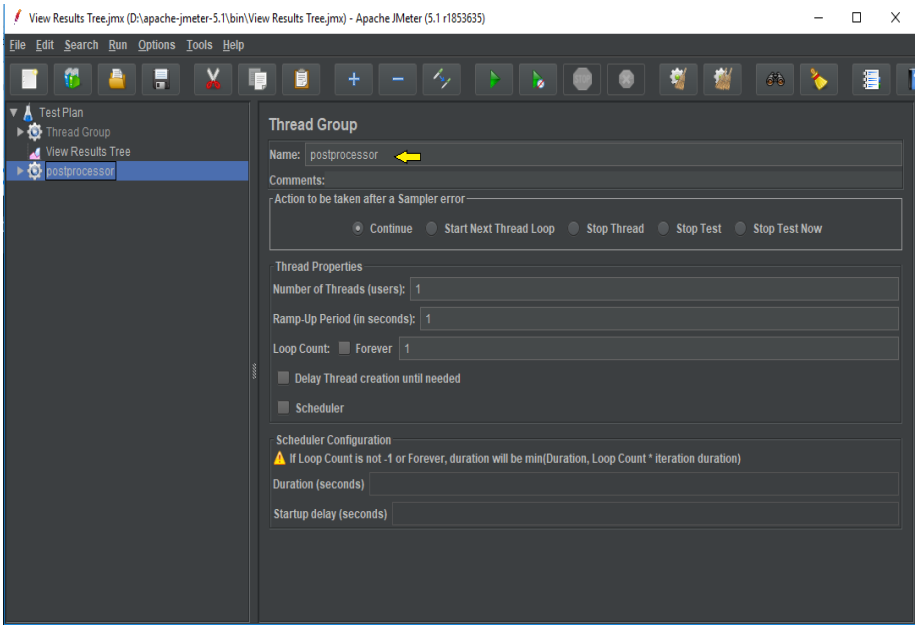
- Right click on Thread Group-->Add Two-->Sampler-->HTTP Request-->consider facebook.com domain.



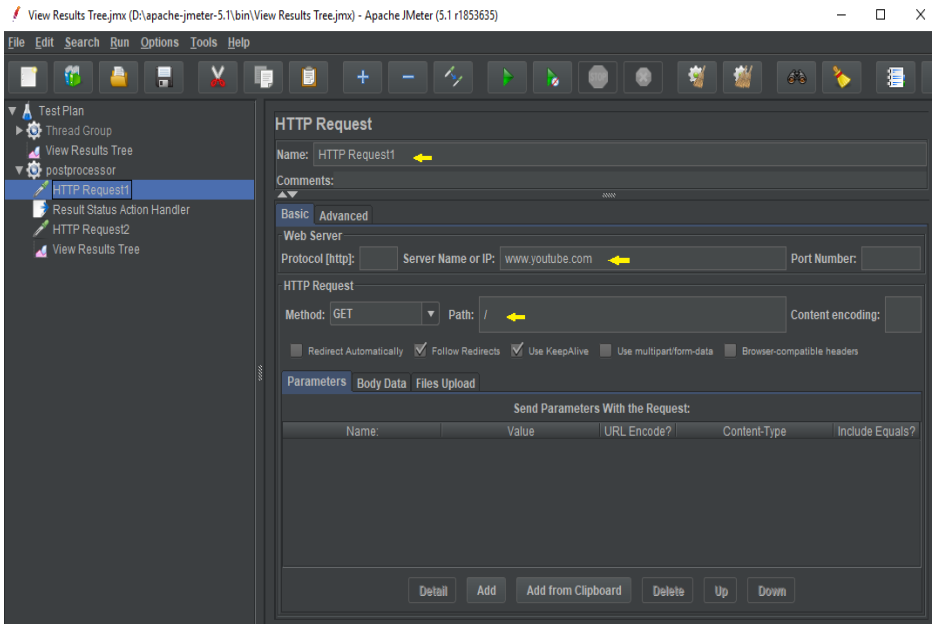


2) Adding Post Processor:

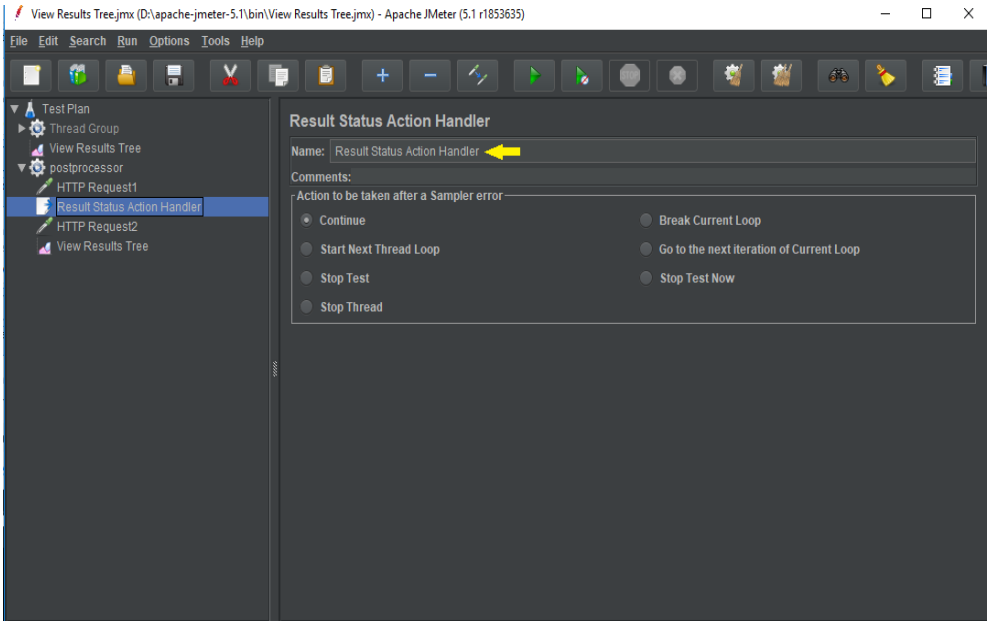
- Right click on test plan-->Add-->Threads-->Thread Group.



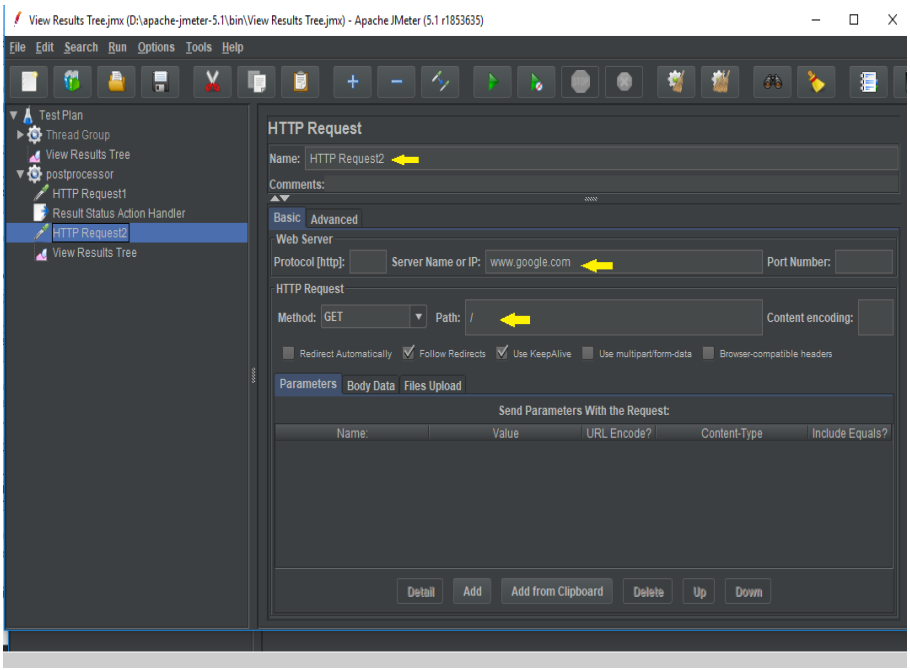
- Right click on Thread Group-->Add -->Sampler-->HTTP Request1.



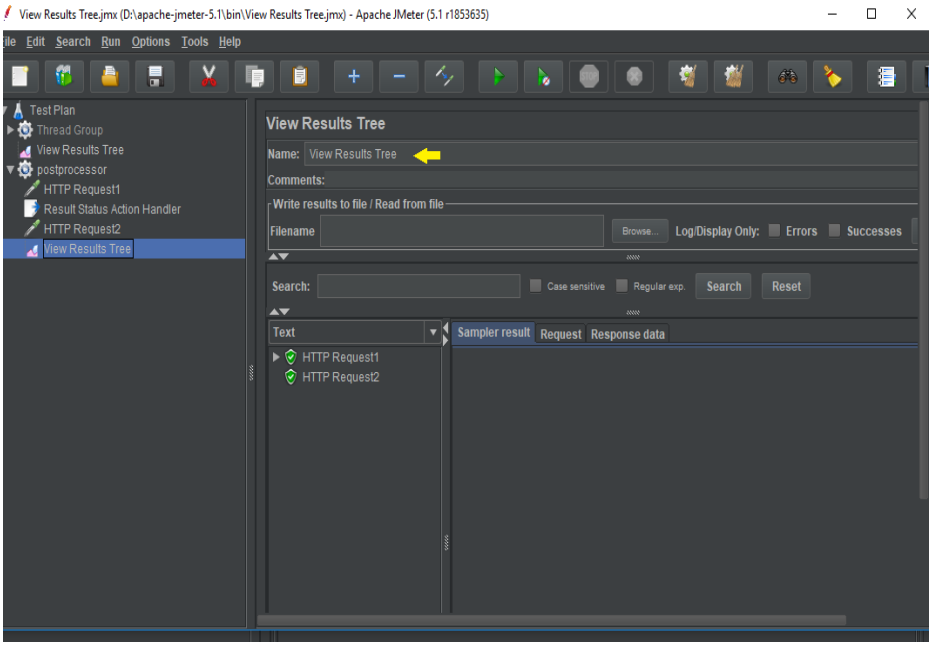
- Right click on post processor Thread Group-->Add-->post processor -->Result Status Action Handler.



- Right click on Thread Group-->Add -->Sampler-->Another HTTP Request2 under Result Status Action Handler.



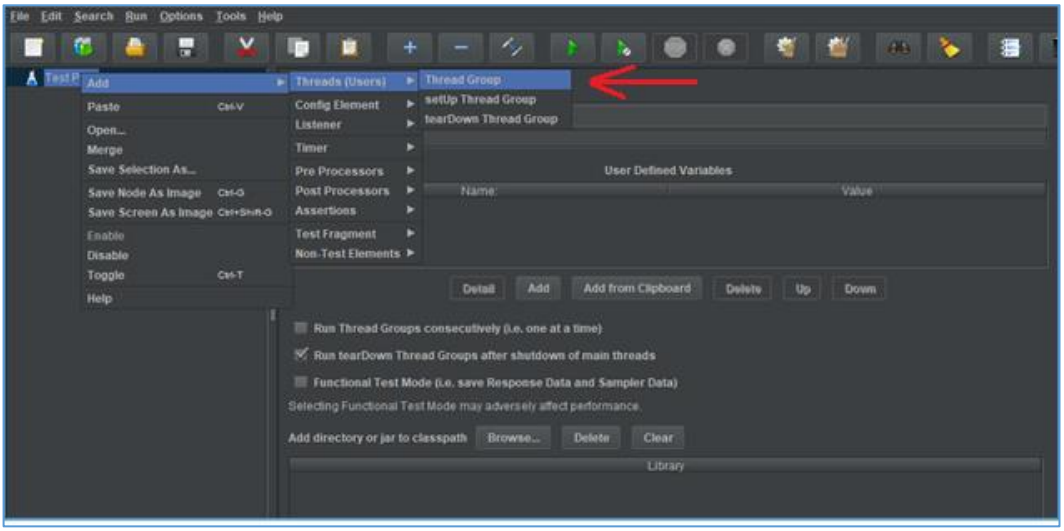
- Right click on postprocessor -->Add --> Listener-->Run result.



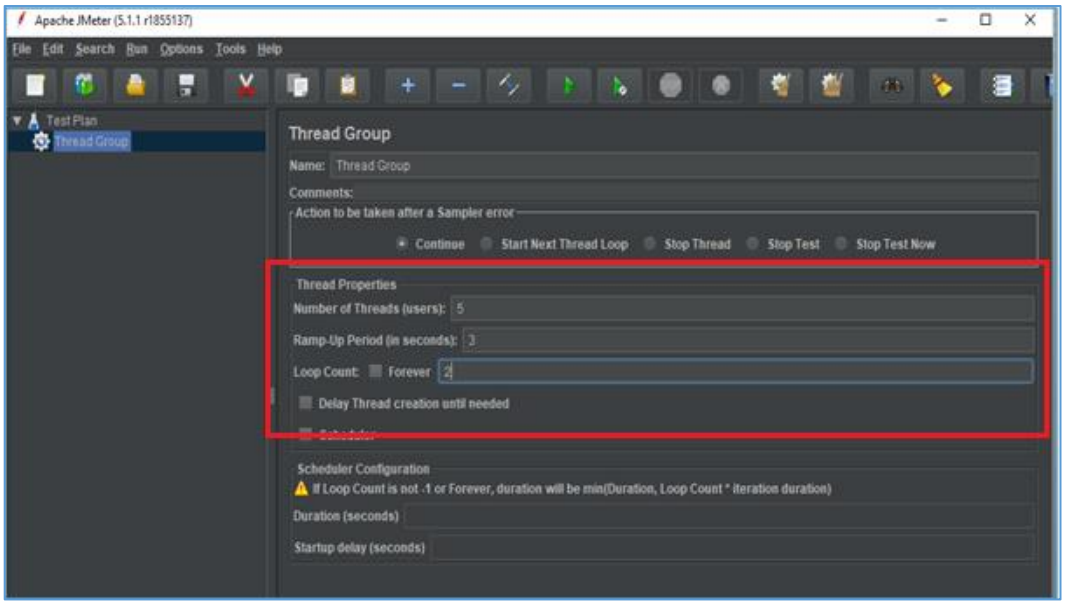
3) Response Assertion:

The Response assertion is used in test scripts to validate a pattern in the response body, header, code, message, etc.

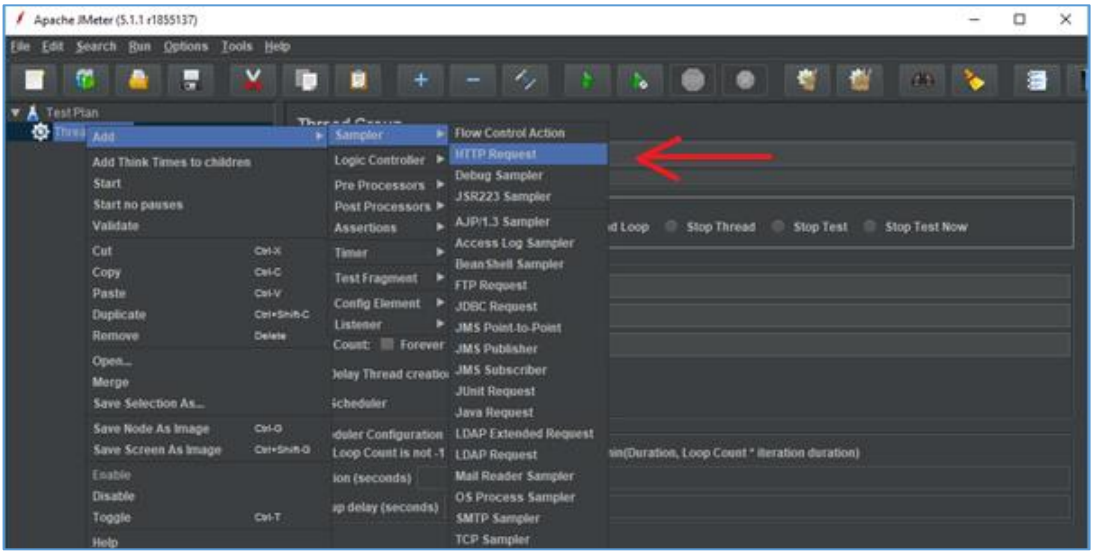
- Open the Apache JMeter.
- Click on the Test plan.
- Rename the Test plan.
- Right click on Test plan --->Add--->Thread(users)--->Thread Group.



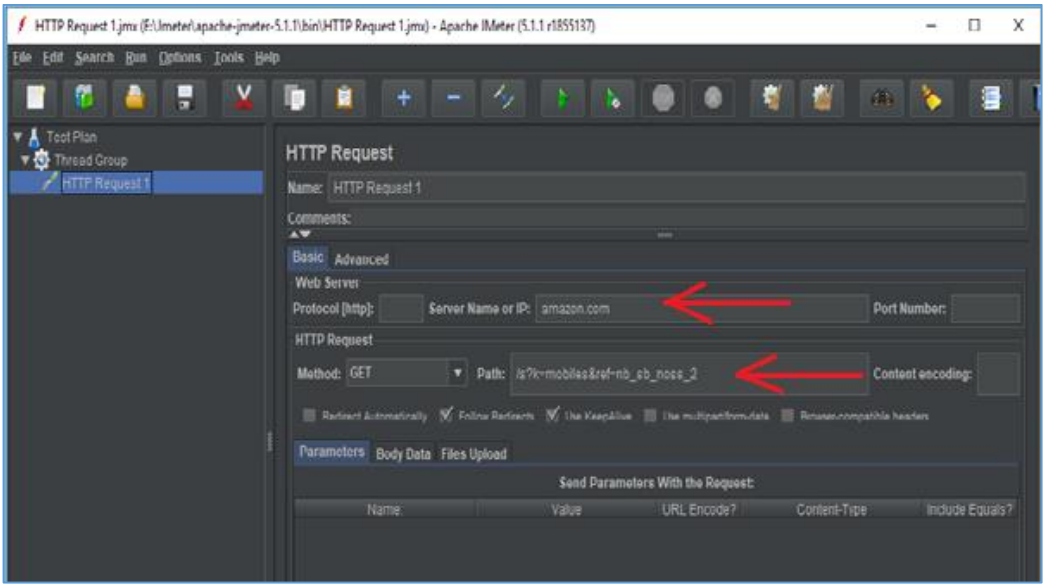
- Navigate through the path: Thread Group--->Name--->Comments---> Continue ---->Thread properties--->Number of Threads(users): ---> Ramp-up period (in seconds): ---> Loop Count --->Save.



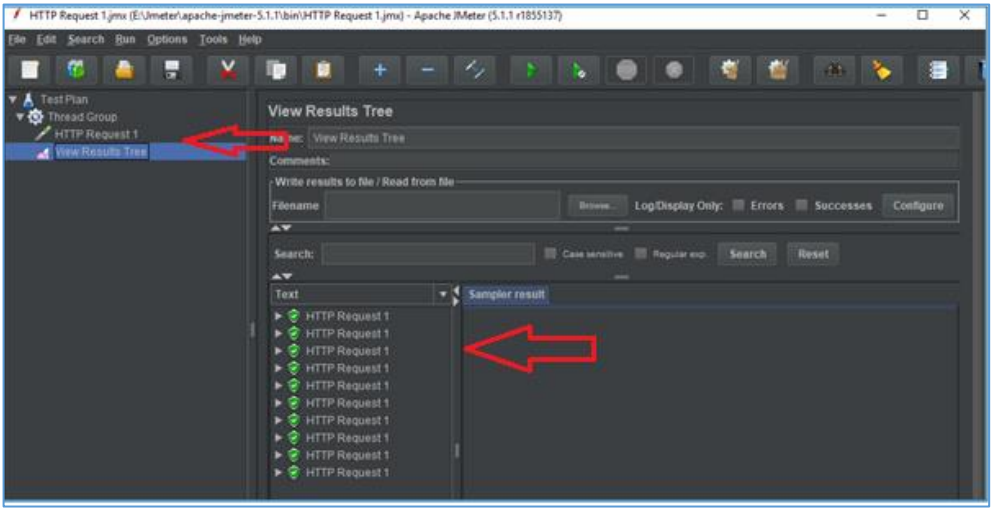
- Right click on Thread group --->Add---> Sampler---> HTTP Request.



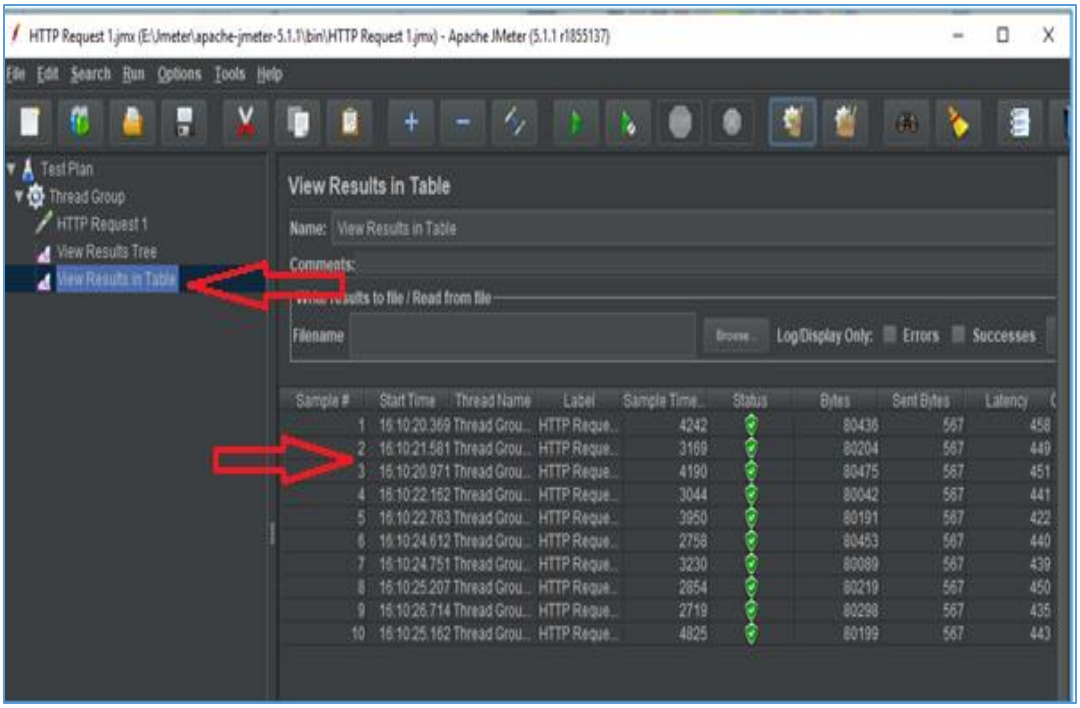
- Navigate through the given path: HTTP Request---> Server Name or IP :---> Path :/---> Save.



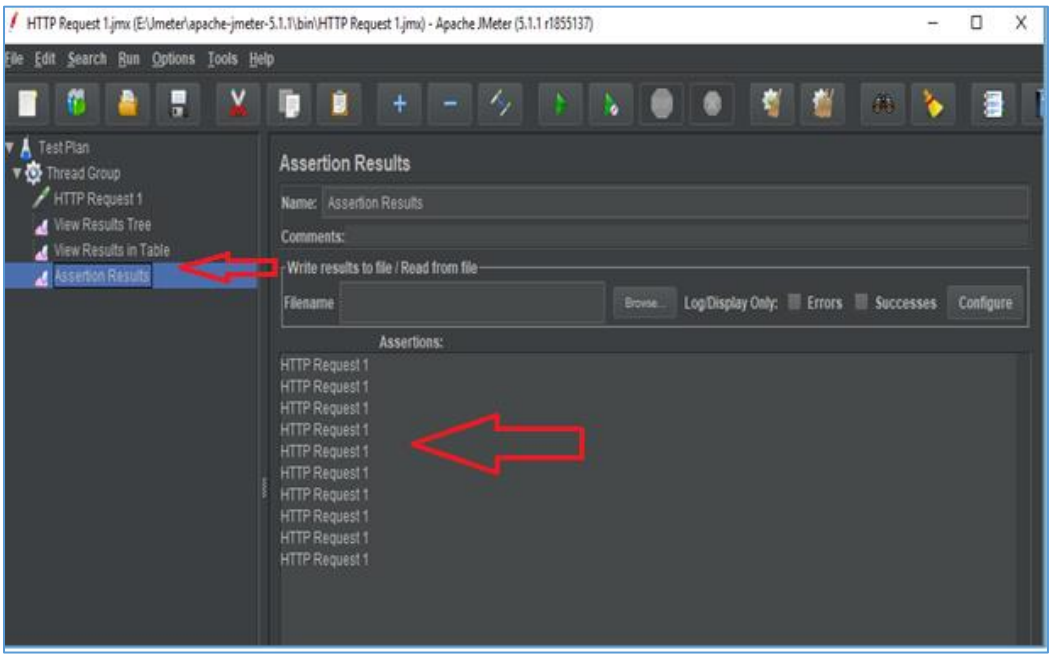
- Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.



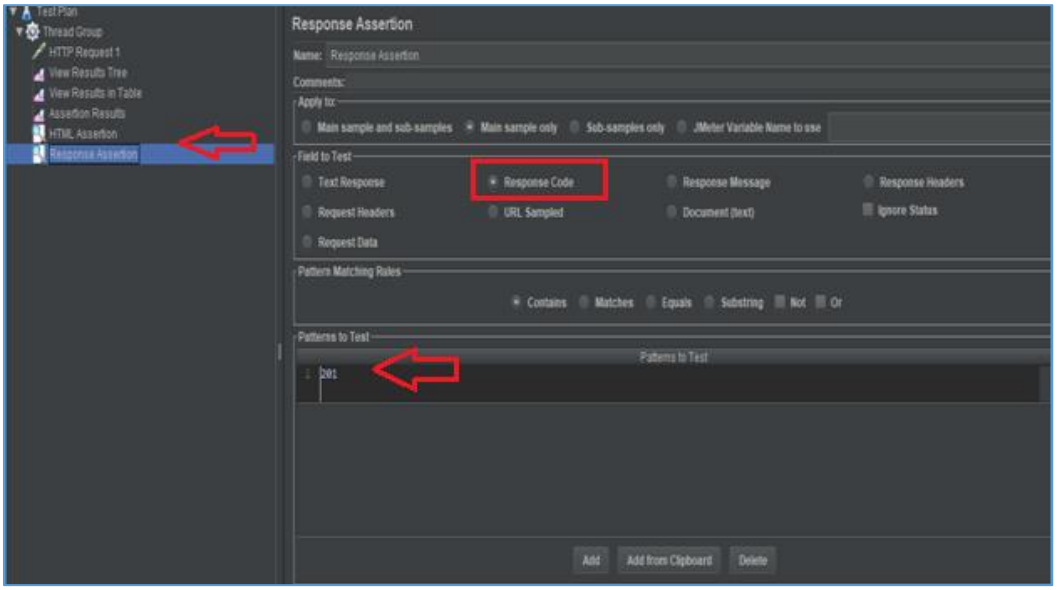
- Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.



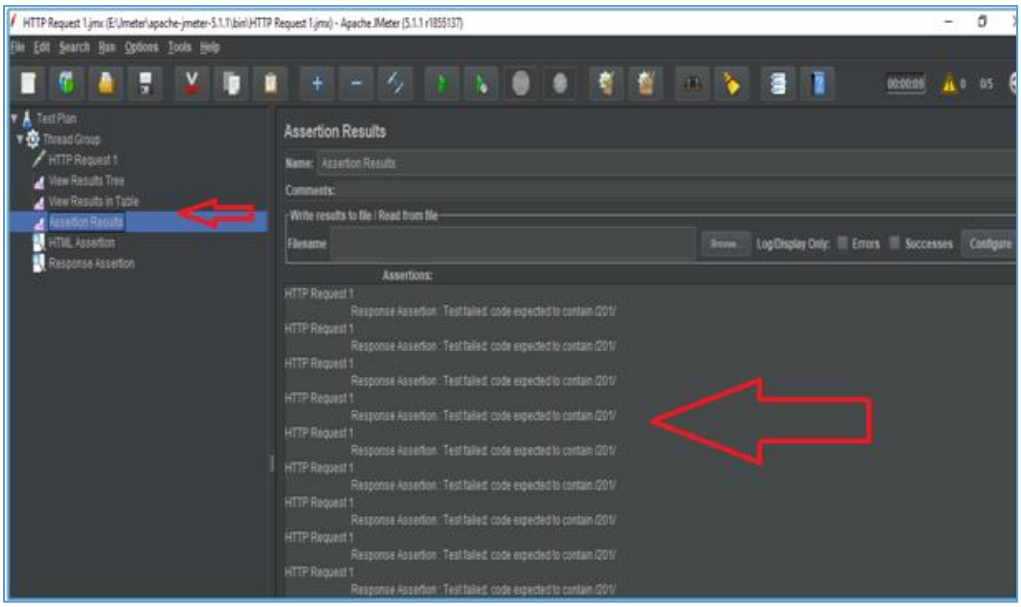
- Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.



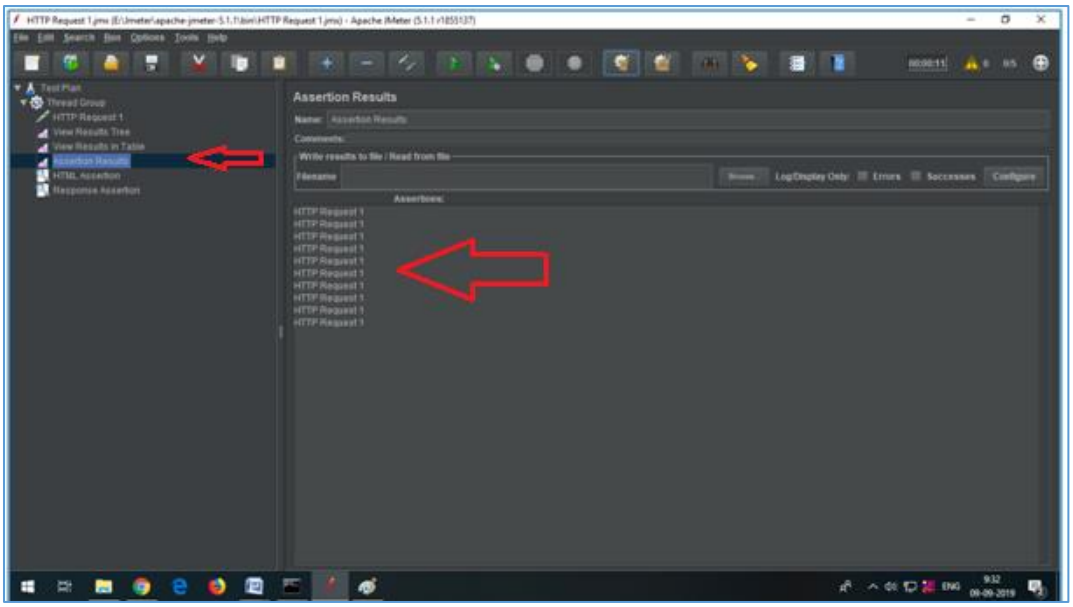
- Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Code---> Contains---> Add---> 201---> Save---> Clear All---> Run.



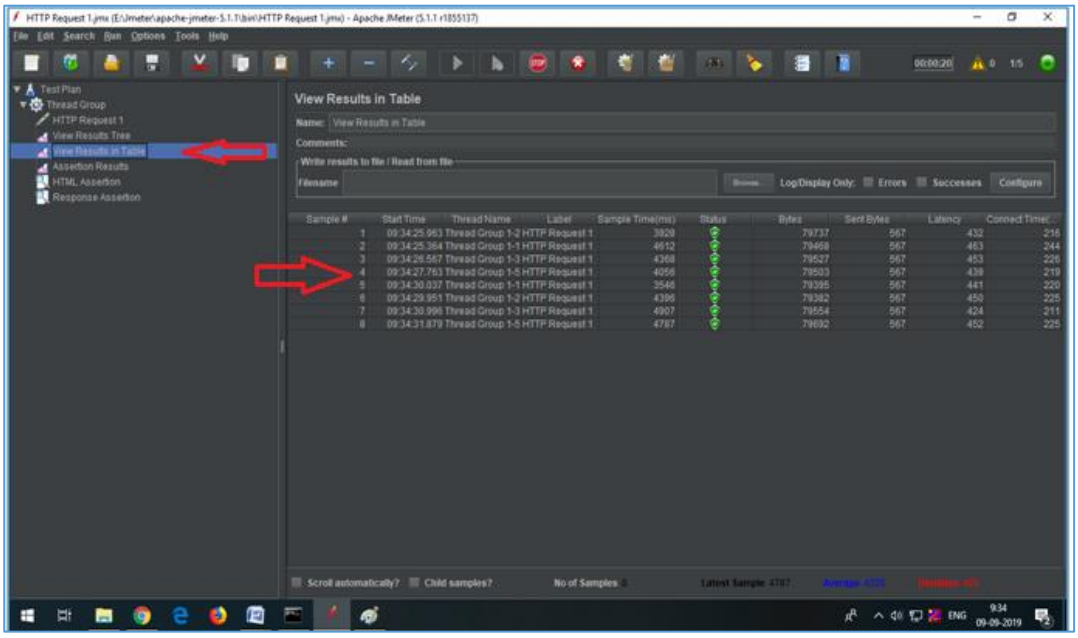
- Navigate through the given path: Assertion Result---> Run. It will show error



- Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Code---> Contains---> Add---> 200---> Save---> Clear All---> Run.

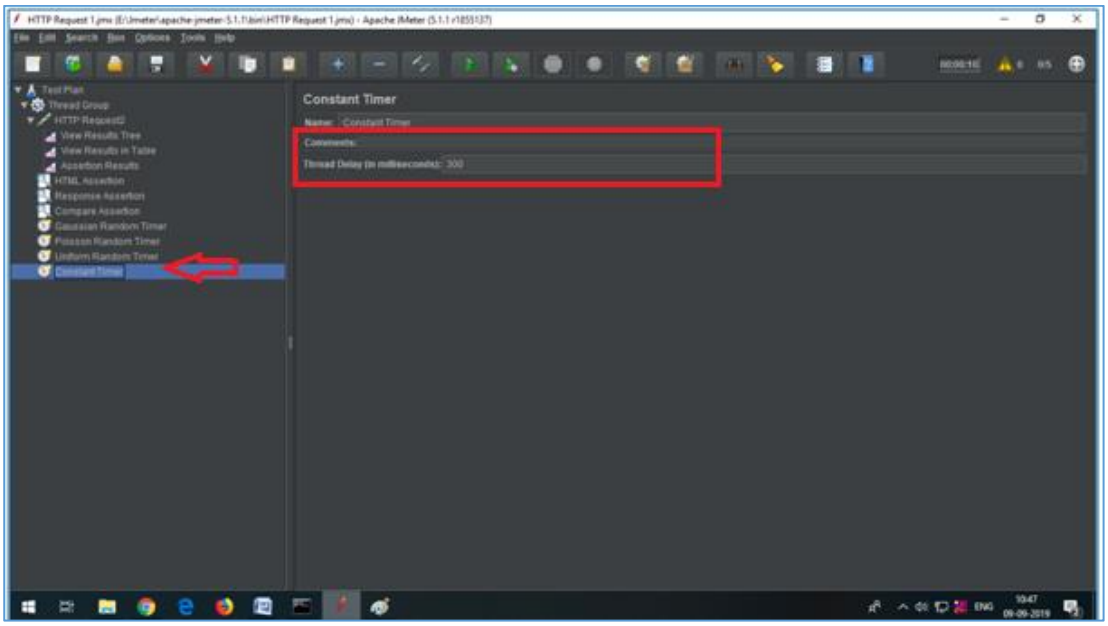


- Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Message---> Contains---> Add---> OK---> Save---> Clear All---> Run ---> View Results in Table.

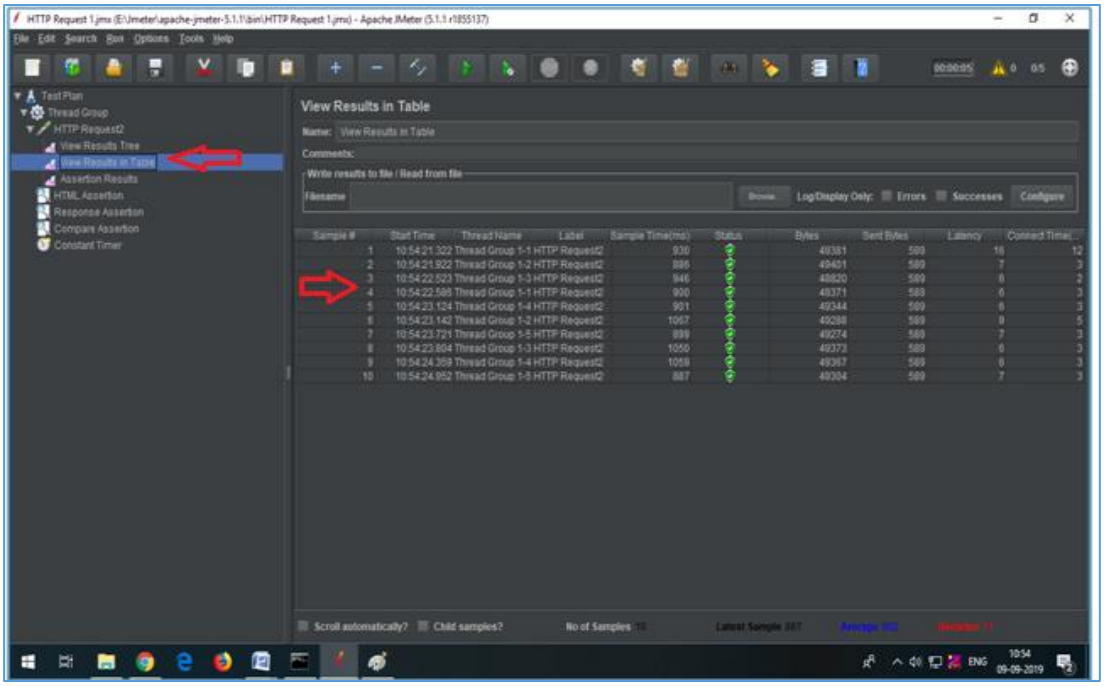


4) Constant Timer:

- Constant timer delays each user request for the same amount of time.
- Right click on Thread Group---> Add---> Timer---> Constant Timer---Thread delay in ms--> Save.



- Navigate through the given path: Clear all---> Run---> View Result in Table.



Step 1. 3.3: Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

```
cd <folder path>
```

Initialize your repository using the following command:

```
git init
```

Add all the files to your git repository using the following command:

```
git add .
```

Commit the changes using the following command:

```
git commit . -m
```

“Changes have been committed.”

Push the files to the folder you initially created using the following command:

```
git push -u origin
```

master