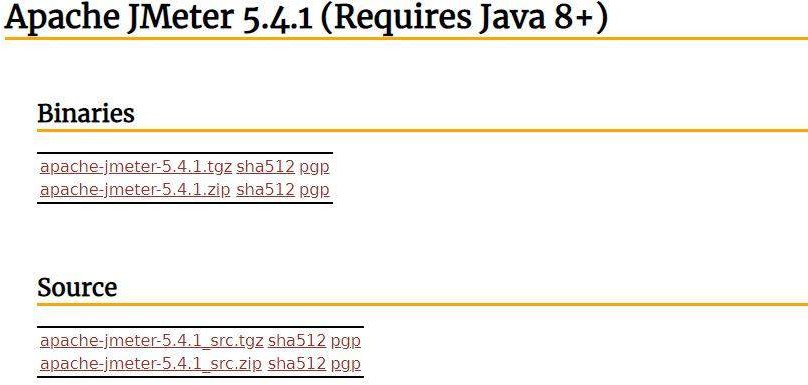
**Download and Install JMeter**

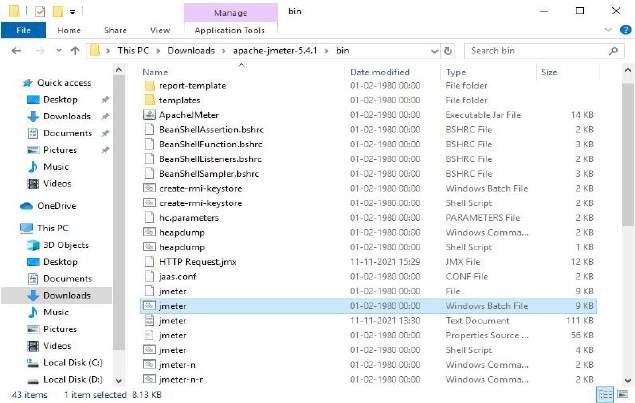
1. Download JMeter

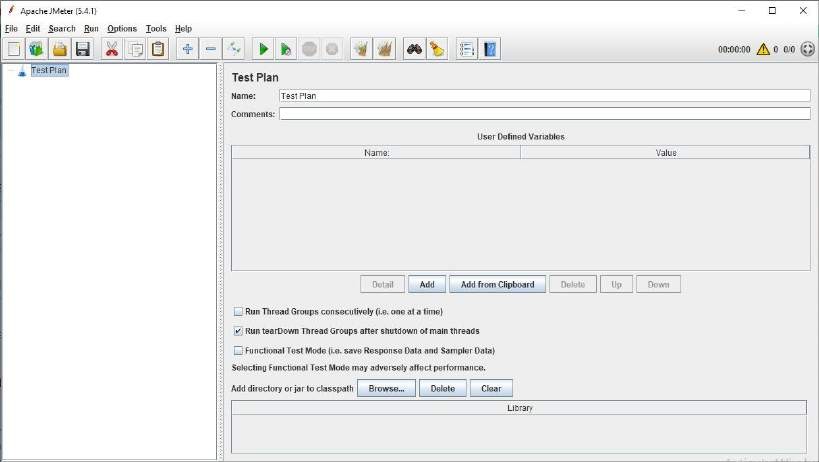
Go to the Apache jmeter download page and select the appropriate release for your machine. Because I'm using a Windows 10 computer, I downloaded the zip file. Extract the downloaded package to a desired location.



1. Start JMeter

To run the jmeter, go to the folder the /jmeter/bin and run jmeter.bat file.





1. Create test plan

A useful test plan is created with minimum 3 components :

Thread Group : contains the simulation of multiple concurrent users. A single thread represent a single user. We can create any number of threads to put the desired load on the application. It also help us in scheduling the delay between two threads, and any repetition of request batches.

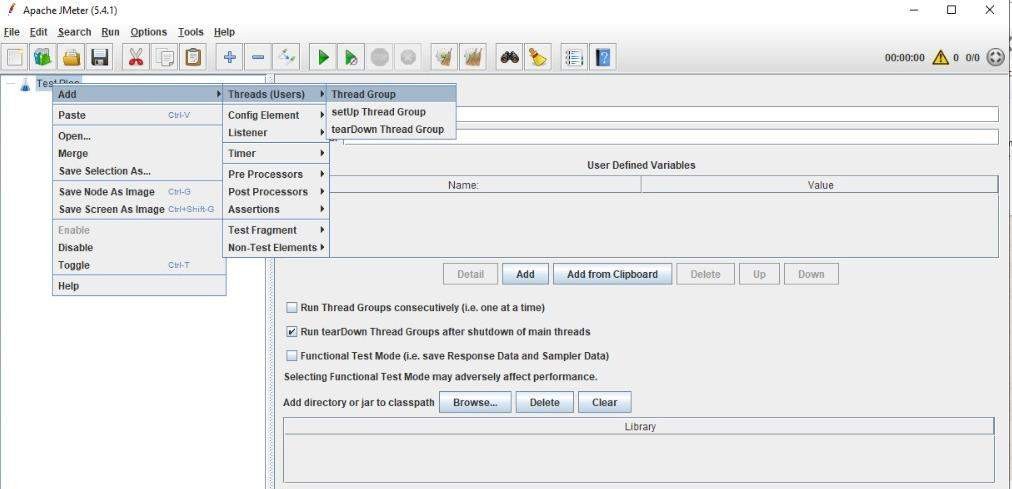
HTTP Request : consist the HTTP request configuration which thread group will be invoking. It is the application URL which you want to load test.

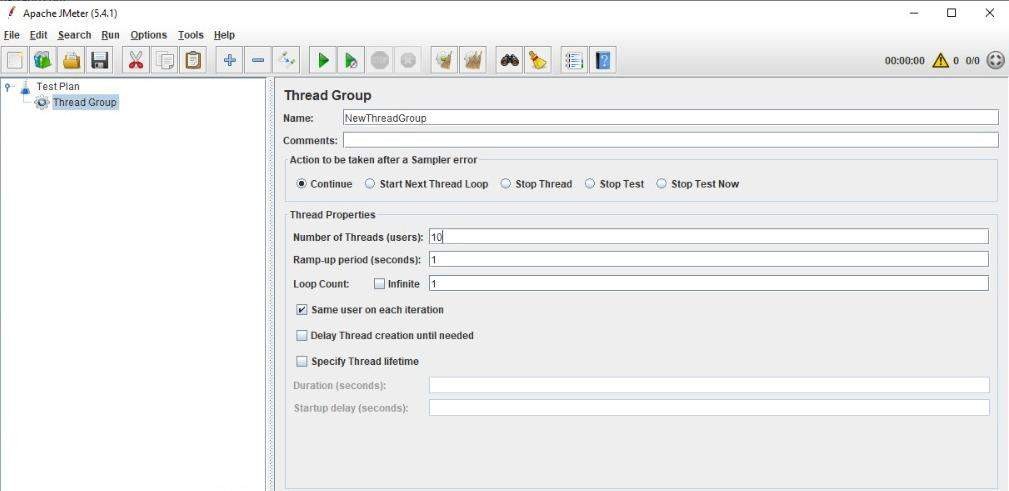
Listener : helps in viewing the result of the whole testing process. There are multiple listener available in jmeter to verify the testing results.

* + Create Thread Group

To create a thread group, navigate to ‘Right click Test Plan -> Add -> Threads -> Thread

Group’.

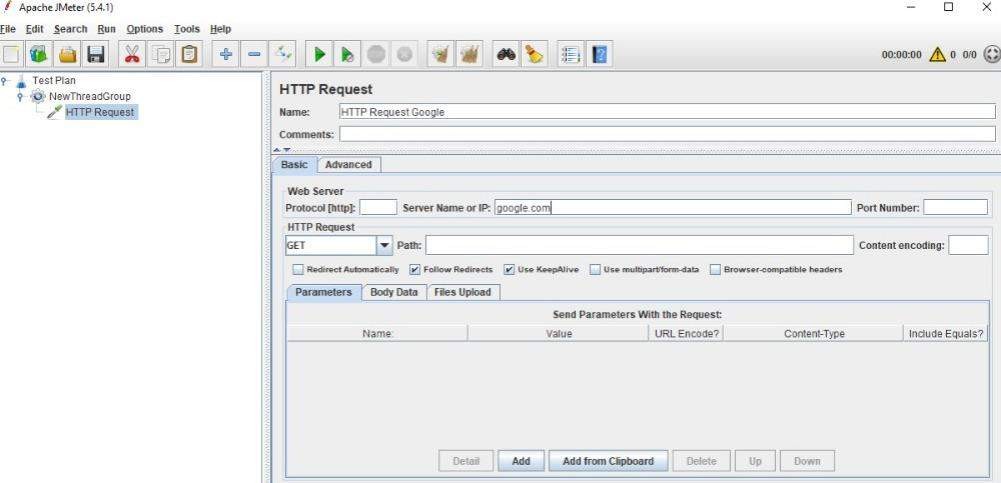
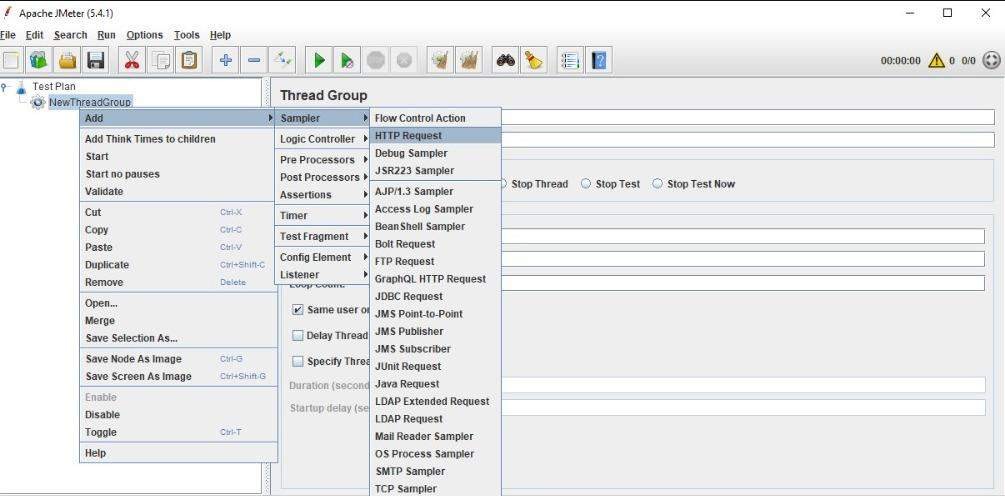




* + Create HTTP Request

To add HTTP request details, navigate to ‘Right click thread group -> Add -> Sampler ->

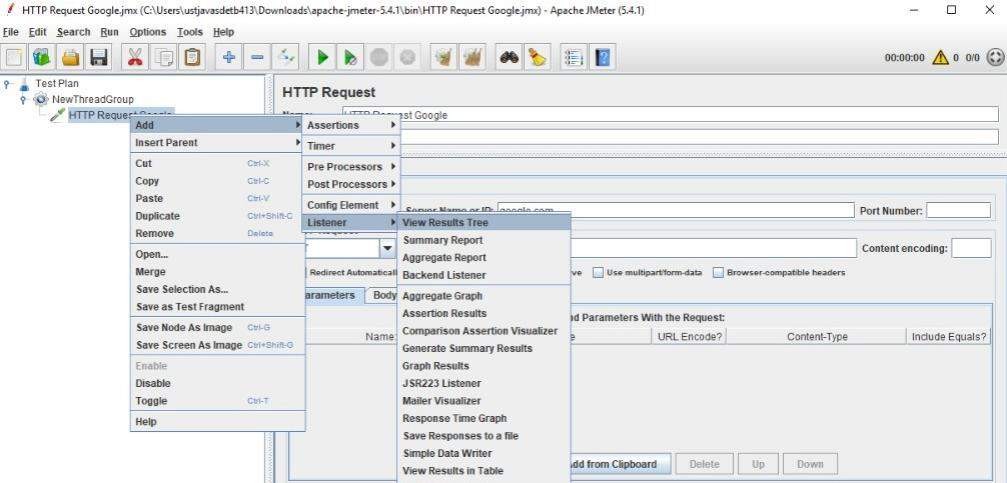
HTTP Request’.

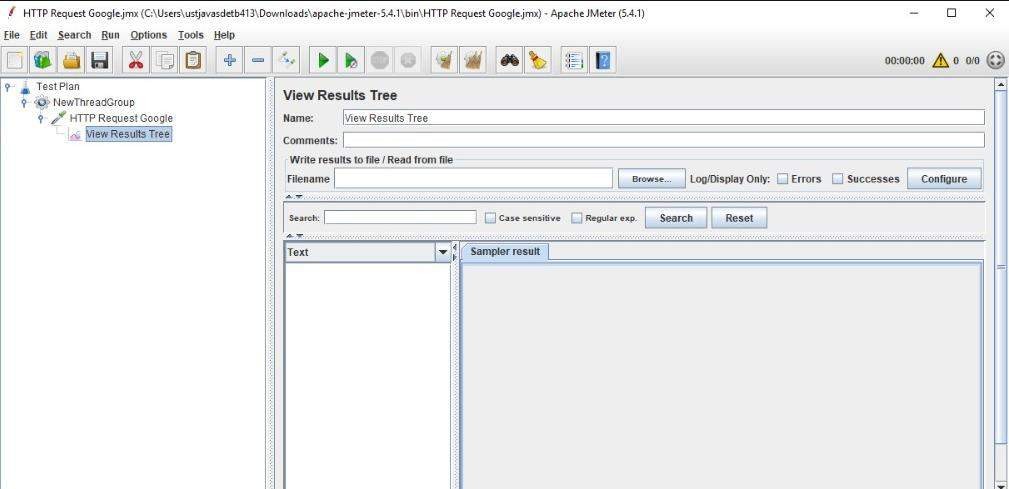


* + Add Listener

To see the results of test plan, add listener named “” by navigating to ‘Right click

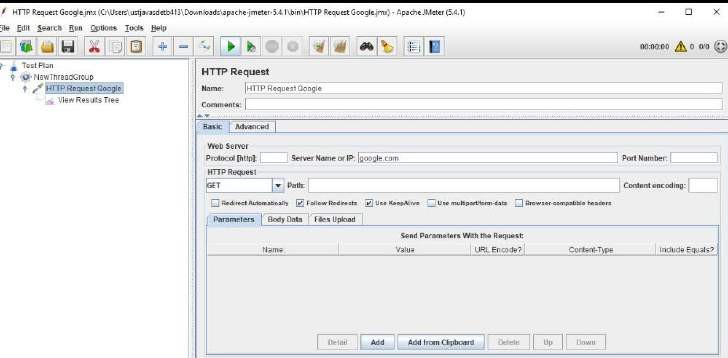
thread group -> Add -> Listener -> View Results Tree’.

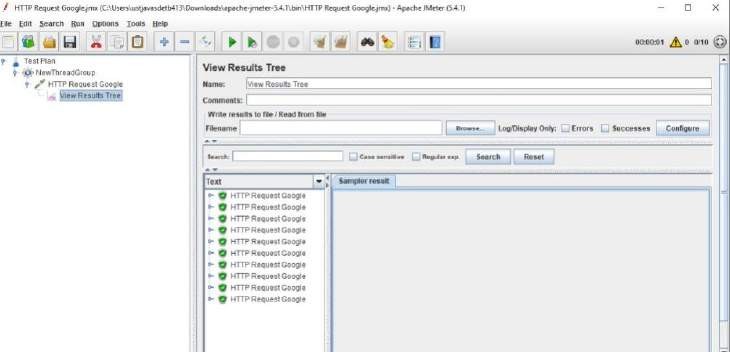




1. Perform load testing

To perform the load testing, start the thread group using the green play icon at the top ribbon in tool.





## JMeter API Testing

Site to be testing:

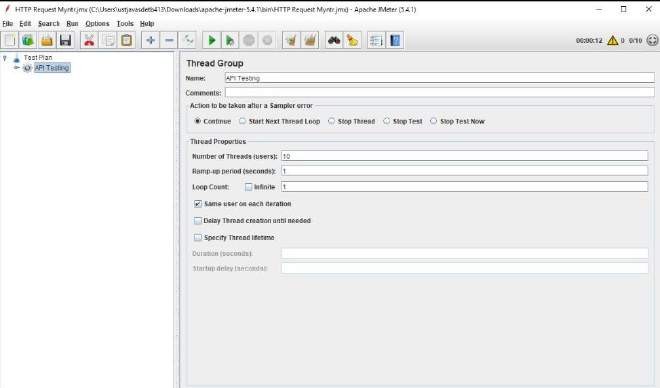


API of Site:

<https://www.snapdeal.com/products/women-apparel-dress-material>?sort=plrty& q=Color\_s%3APink%7C

Steps

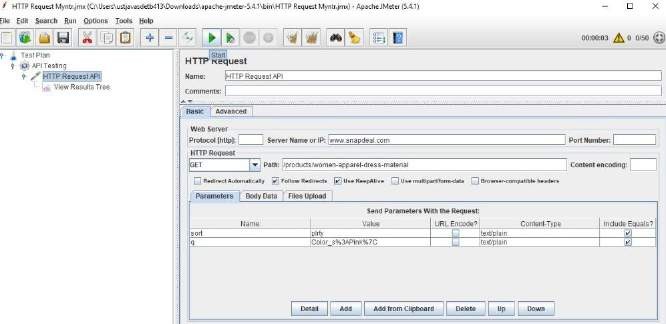
1. Create test plan
2. Create Thread Group Name:API Testing



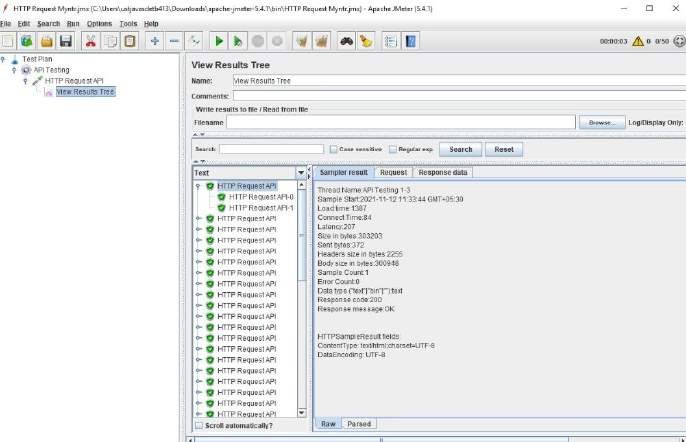
1. Create HTTP Request

Server name : [www.snapdeal.com](http://www.snapdeal.com/)

Path : /products/women-apparel-dress-material Parameters : sort=plrty q=Color\_s%3APink%7C

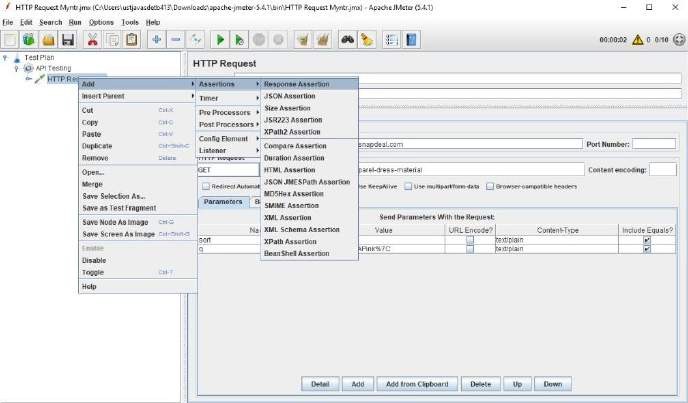


1. Add Listener – View Results Tree
2. Save and Run Http Request API Result :



1. Add Assertion

To check the Respose code

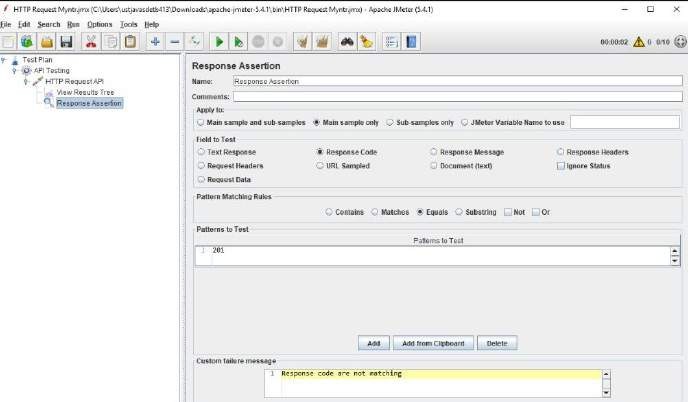


 set Field to Test as Response Code

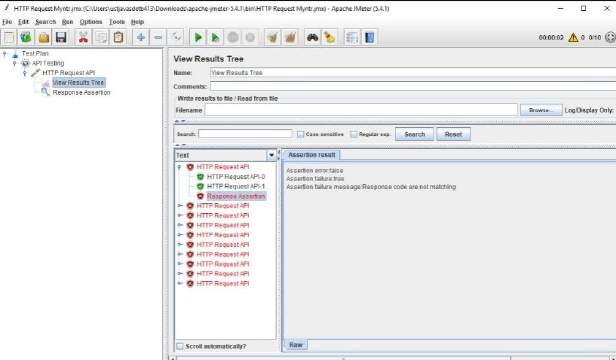
 set Patterns Matching Rules as Equals

 set Patterns to Test as A value of response code

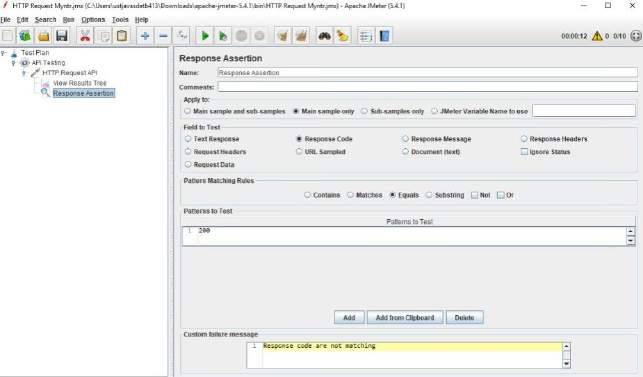
 set Custom Failure Message [message to print if it is error]



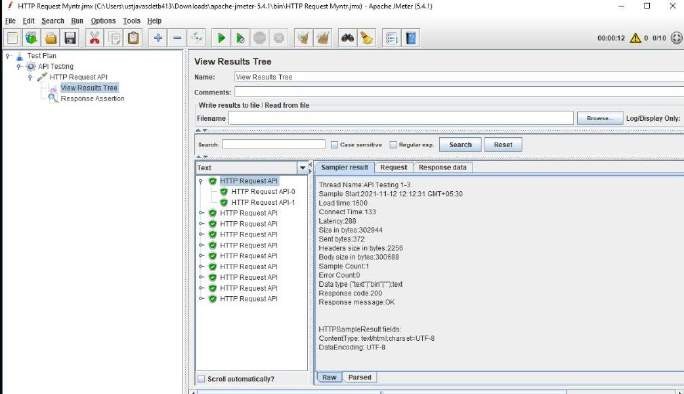
 In this assertion Response code pattern to test is different so it shows some errors



 If Response code is Same



Result:



**JMeter CSV Data Set Config**

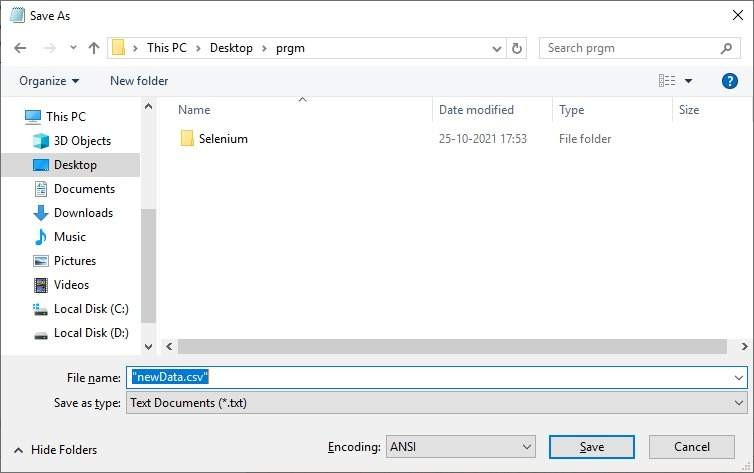
JMeter, an open source load testing tool, has an element that allows you to use external data sets in a CSV format. This element is called the “CSV Data Set Config”. The CSV Data Set Config is used to read lines from a file and to split them into variables.

Steps:

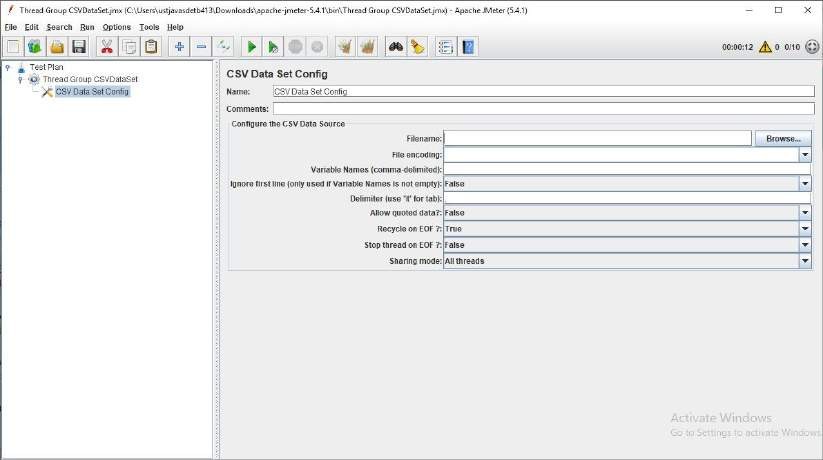
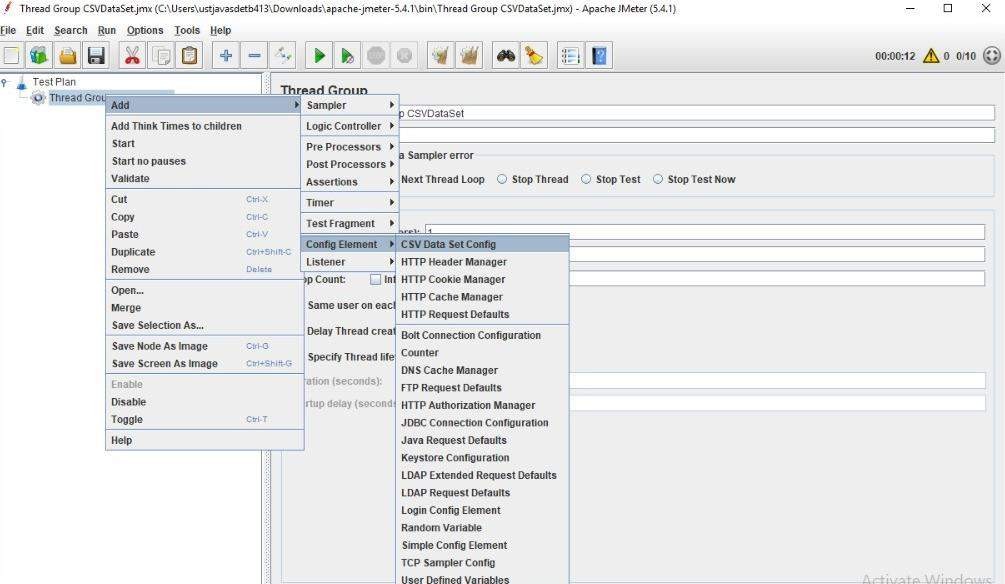
* Create CSV file



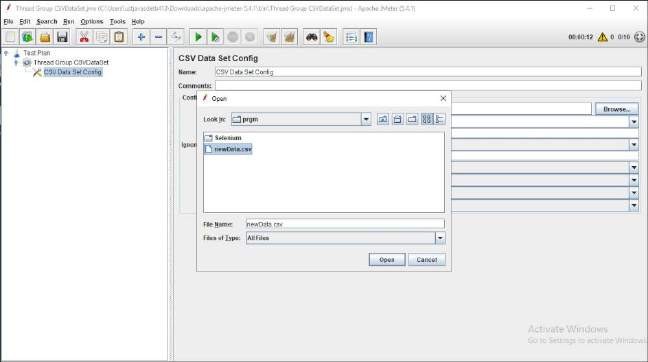
Save



* Create a test plan. The Test Plan contains one Thread Group.
* Add Config Element to the Thread Group



* Browse a CSV file



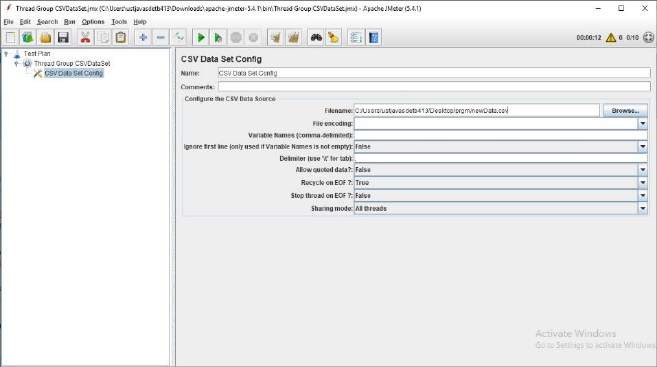
* + Filename: if your file is in the /bin directory, enter just the filename. If it’s

somewhere else, use the full path to the file.

* + Variable names: Define your column name mapping as a list of comma-separated strings. BlazeMeter uses these names as parameter names to refer to data columns. To skip a column, add an extra comma with no name.

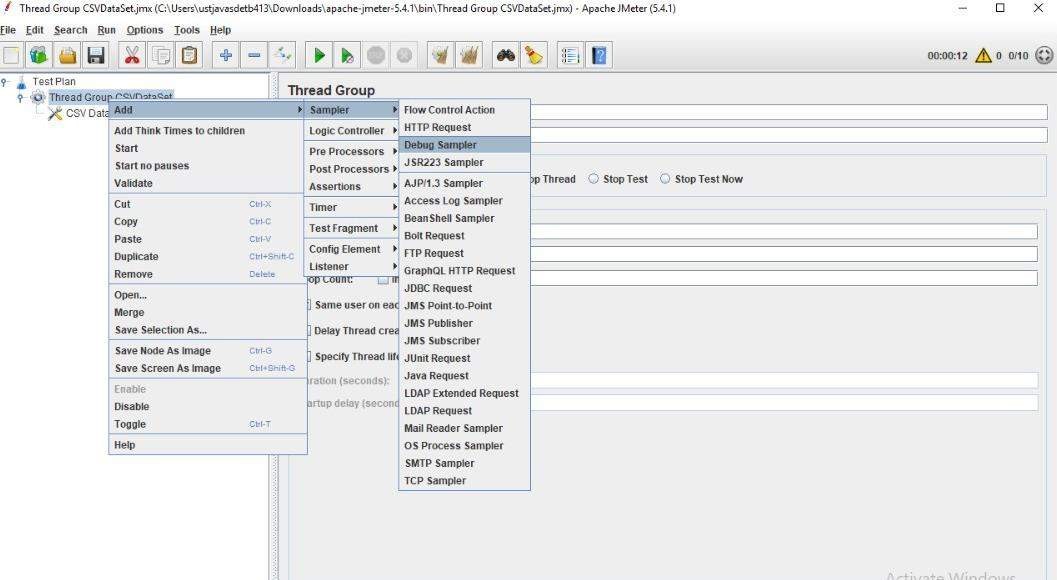
Examples:

* + - user,passwrd,cookielength,cookieneverexp
    - lastname,firstname,,street,number,,
    - For a CSV file with six columns, this mapping lets you reference column 1,2,4,and 5 with the given parameter names, and you ignore column 3 and 6.
  + Delimiter: A comma is the default delimiter, but if your file uses tabs, enter \t here.
  + Ignore first line? If the first row of the csv file contains column names, enable this option. If the first row contains data, disable this option.
  + Allow quoted data? If your column values can contain commas, and you also use commas as delimiters, then allow quoted values.

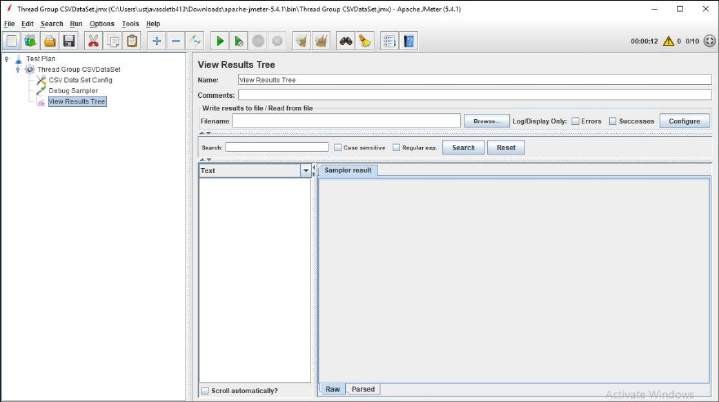
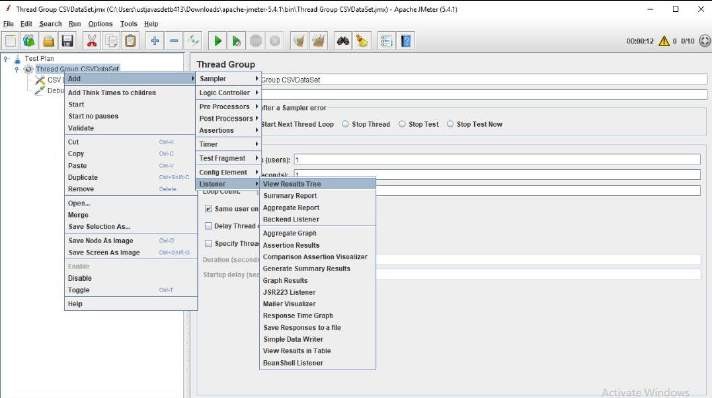


* Add a Sampler called Debug Sampler

This is used for something that allows you to read the values of JMeter variables.

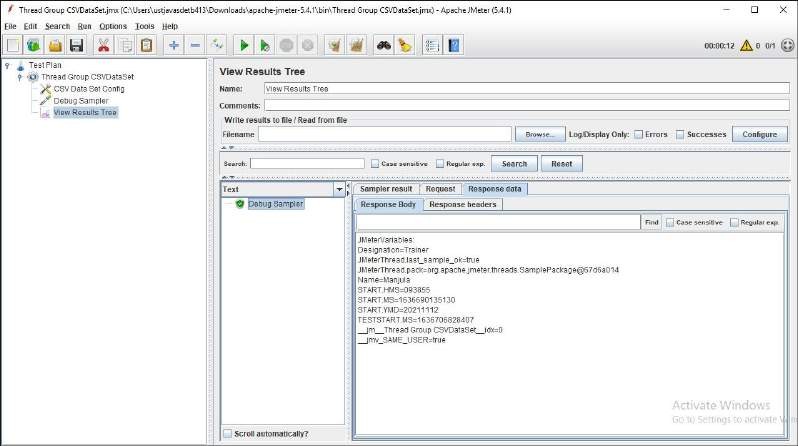


* Add Listeners

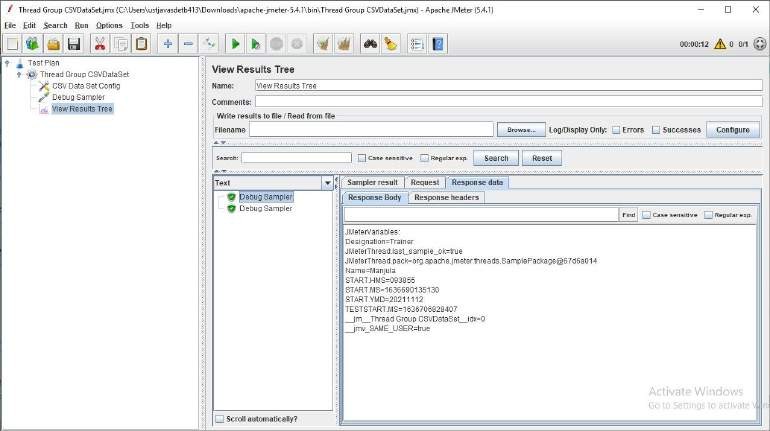


* Run the Test Result:

The response data Shows only one value.

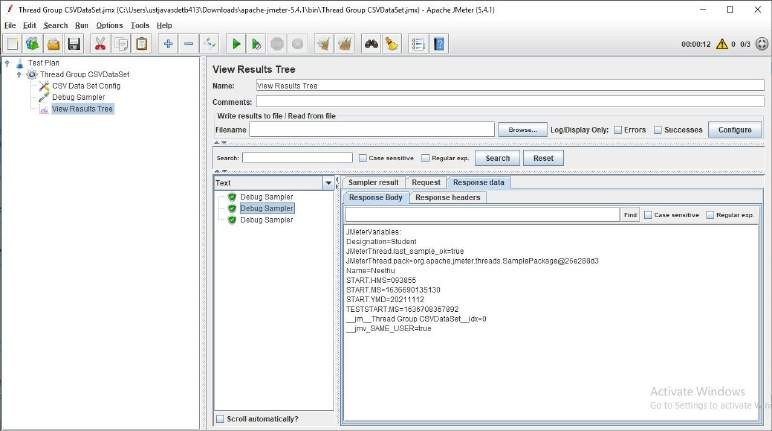
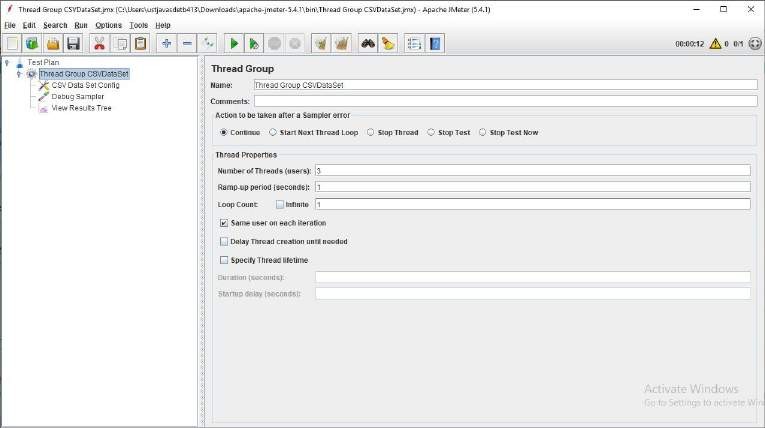


If you run it again, It reading the same data.

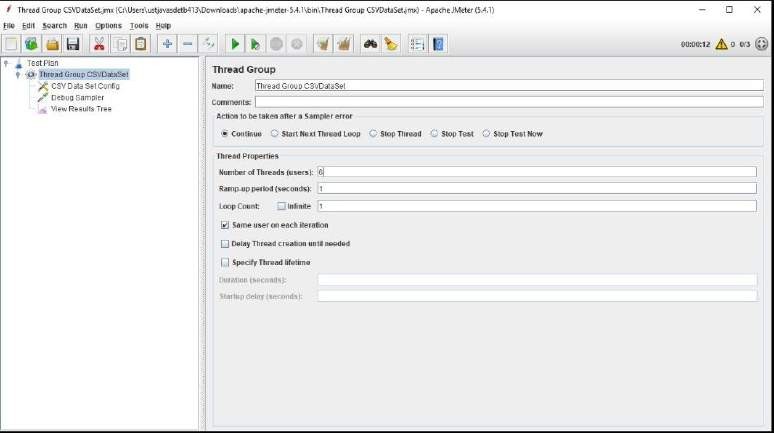


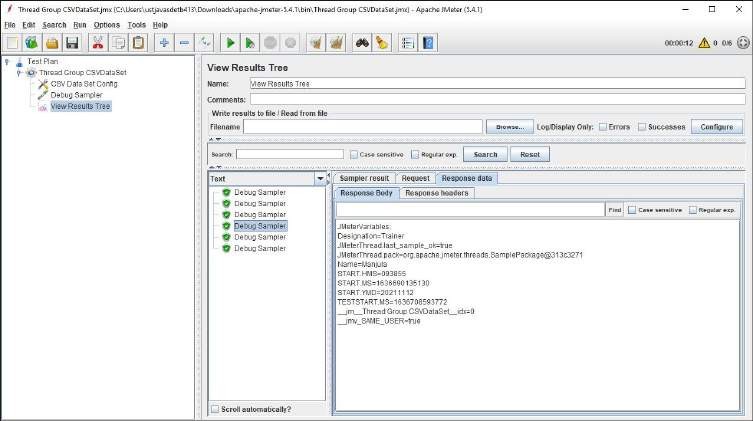
If we want to show all of the data in a csv file, the number of data in the csv file and the number of users in Thread Group must be the same.

Example: If we have 3 data in CSV file then we need to set 3uses in Thread group



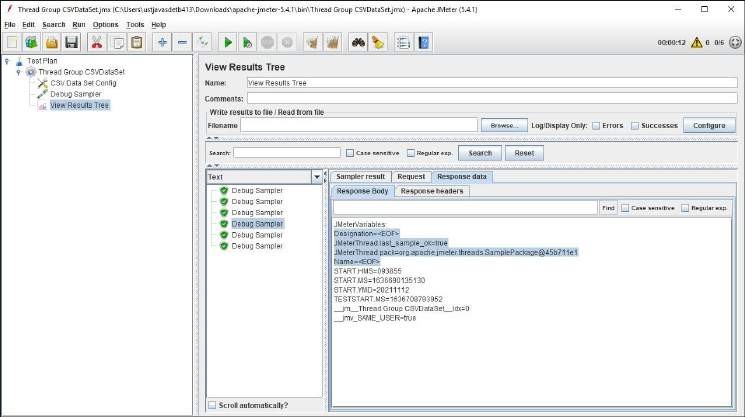
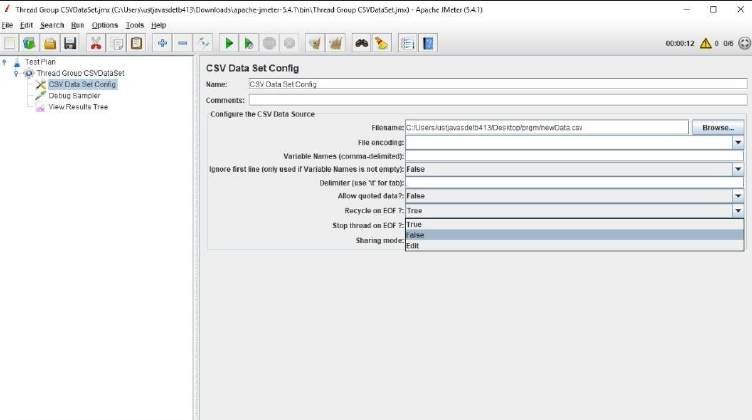
If Specifies more user and run that again, it will execute well and shows the data is started to repeat again.



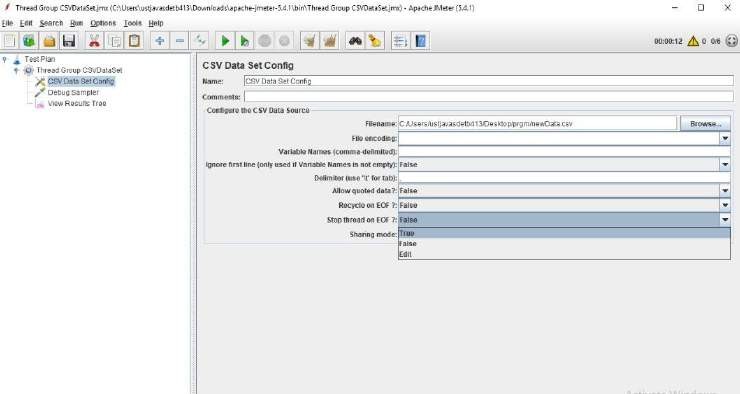


Recycle on EOF: true by default, should JMeter restart from the beginning when the End Of File is reached,

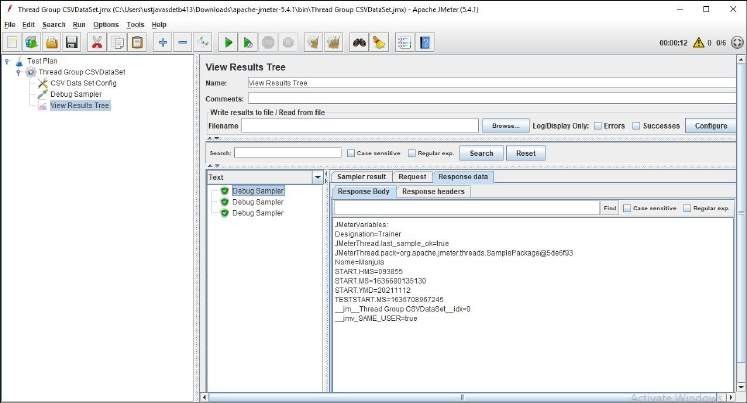
If we choose it is false, It will restart from beginning but shows the value as <EOF>.



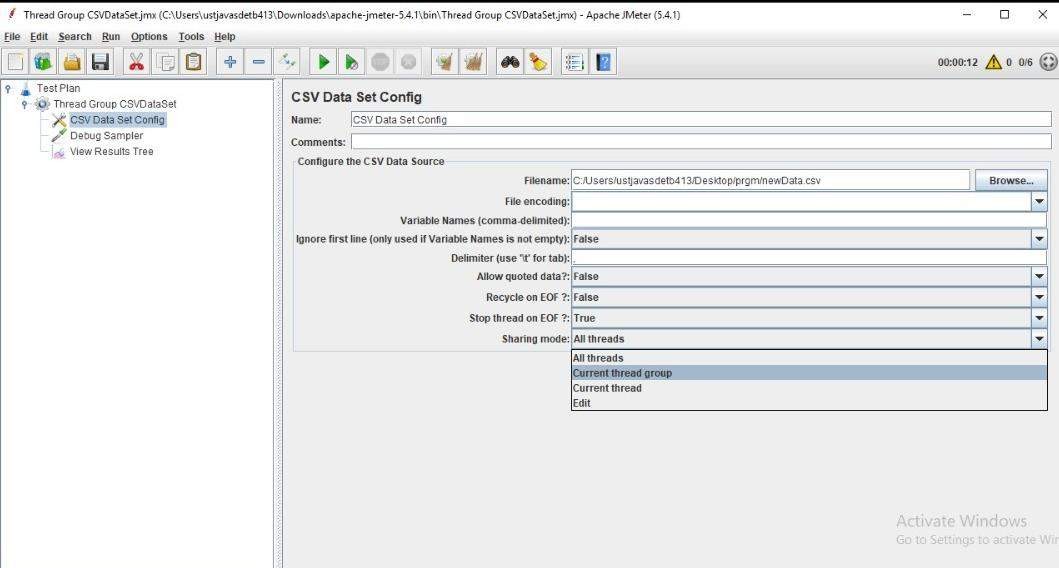
Stop Thread on EOF: stops the current thread gathering the value if EOF is reached, It displays the same number of samplers as the number of data in the CSV file.



It shows 3 sampler even if we specifies 5000 users.



Sharing Mode: defines how values are distributed among concurrent threads.



# JMeter Timers

Timers allow JMeter to delay between each request which a thread makes. A timer can solve the server overload problem.

Also, in real life visitors do not arrive at a website all at the same time, but at different time intervals. So Timer will help mimic the real-time behavior.

Purpose :

 To pause thread (users) for some time

 To add delay between threads

 To avoid over flooding the served and achieve real time behavior by pacing the load.

Type of Timer :

 Constant Timer

 Gaussian Random Timer

 Uniform Random Timer

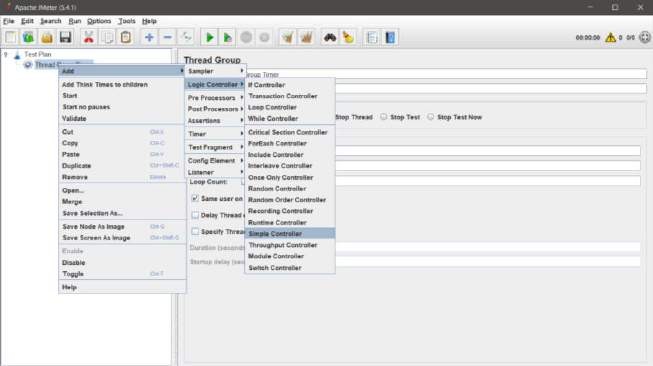
 BeanShell Timer

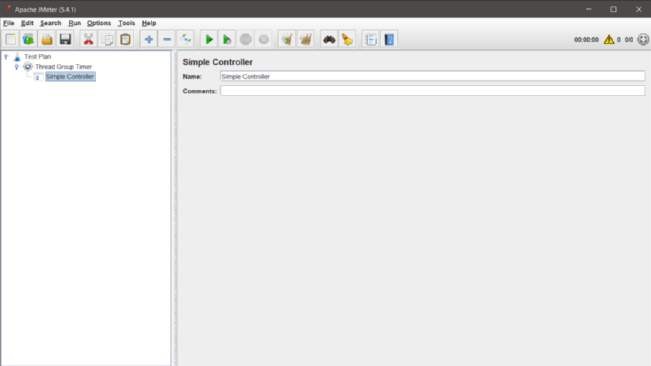
 BSF Timer

 JSR223 Timer

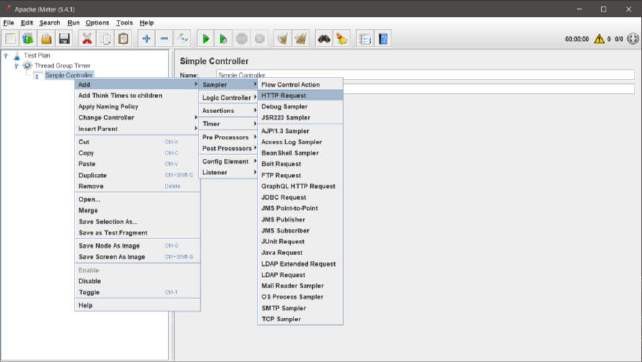
 How to Use Constant Timer Steps :

* Create new Test plan
* Add new Test group
* Add Samle Controller

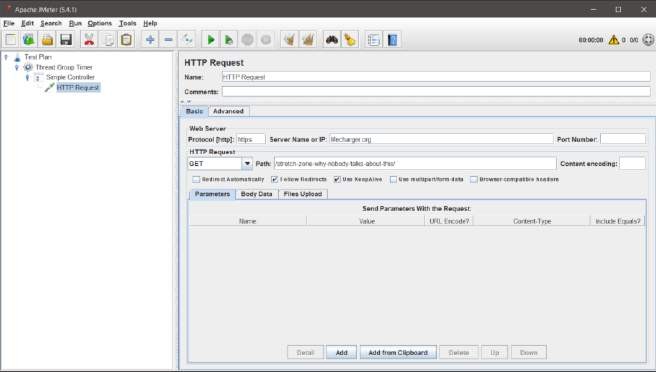




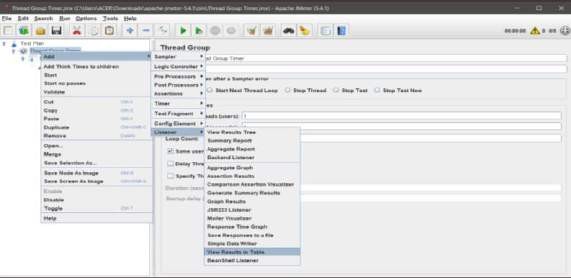
* Add HTTP Request

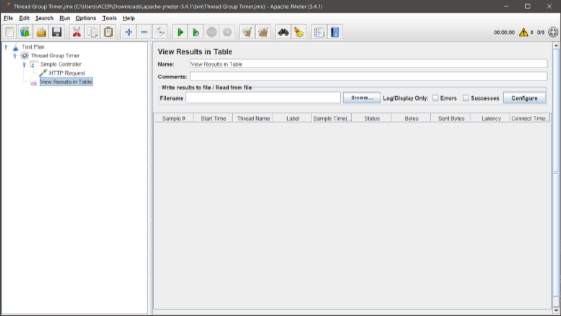


This webpage is used:

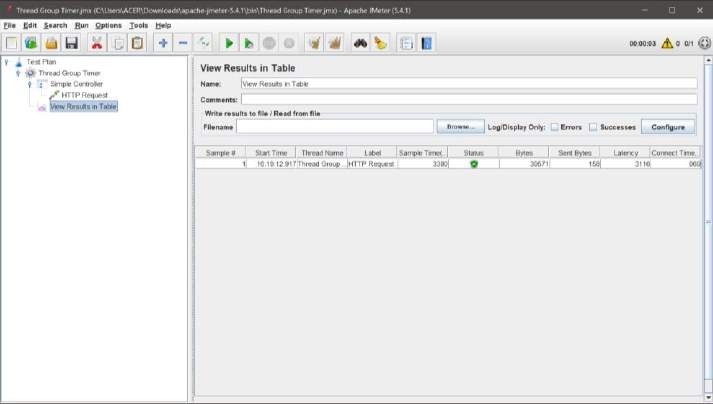


* Add a Listener to display Result



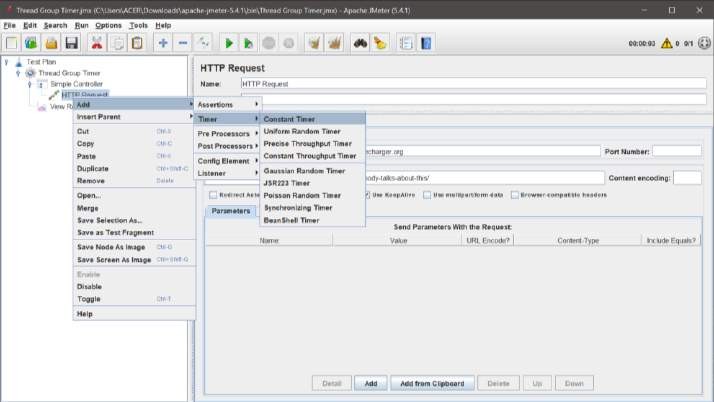


* Run the Test : It will run with time 1 second without any delay

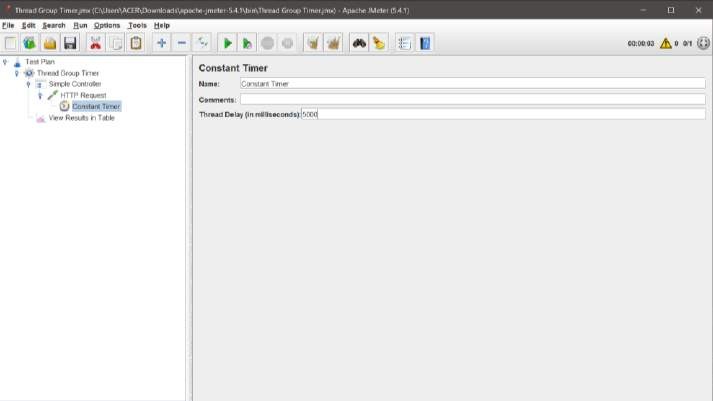


* Add the Timer – Constant Timer .

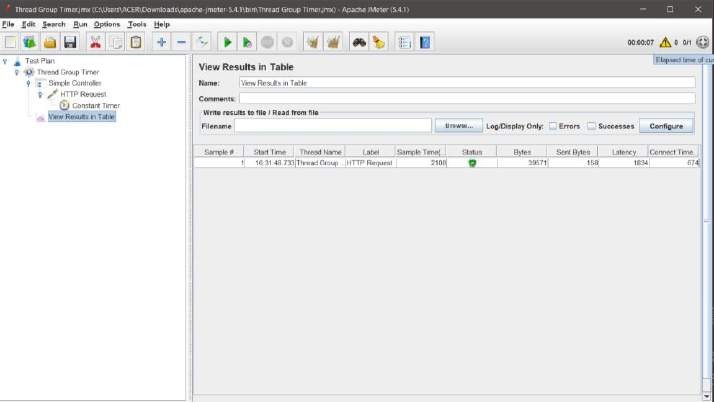
Constant timer delays each user request for the same amount of time.



Delay time : Set a time of delay in millisecond. Here set 5000 milli second as delay

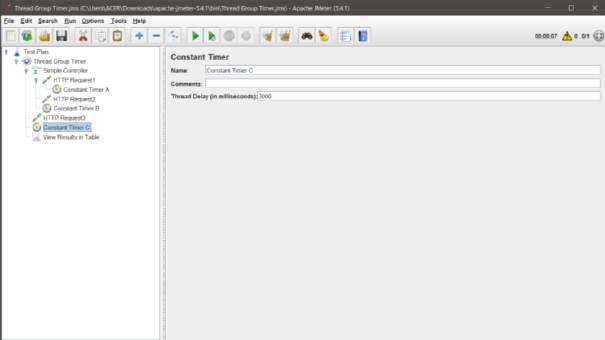


* Run the Test : After 5000millisecond it will execute

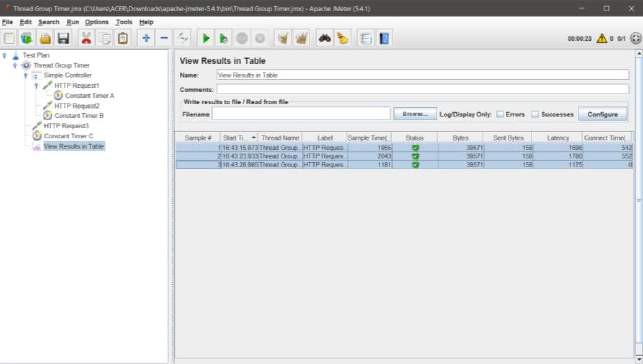


* Add some more HTTP Request and Constant Timer Constant Timer A (delay :3000)is only for HTTP Request 1

Constant Timer B(delay :3000) is inside Test group so it is applicaple for HTTP Request 1&2 Constant Timer C (delay :3000)is inside Test plan so it is applicable for HTTP Request1,2&3

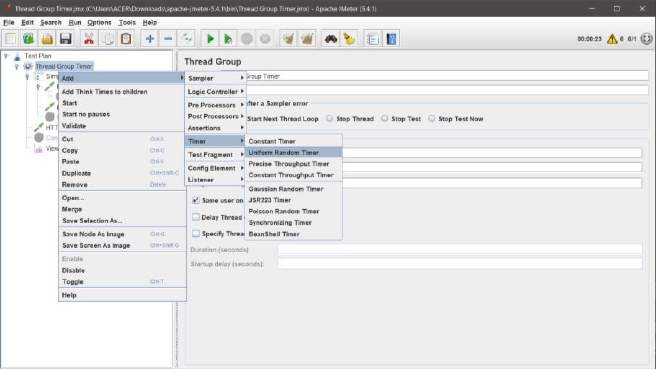


|  |  |
| --- | --- |
| * Run the Test |  |
| Request 1 is executed after delay of | 9000 |
| Request 2 is executed after delay of | 6000 |
| Request 3 is executed after delay of | 3000 |

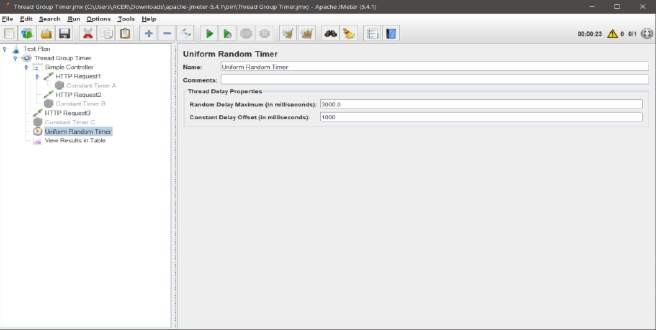


* Disable all timer
* Add a new Timer -Uniform Random Timer

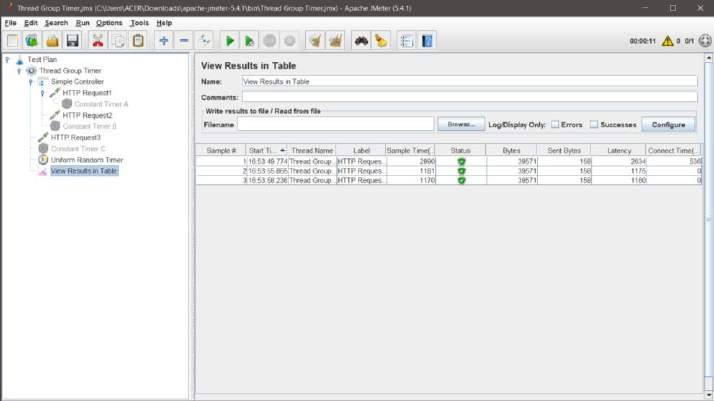
Uniform random timer delays each user request for a random amount of time.



Random Delay Maximum: Maximum random number of milliseconds to delay. Constant Delay Offset (milliseconds): Additional value in milliseconds



The total delay is the sum of the random value and the offset value.



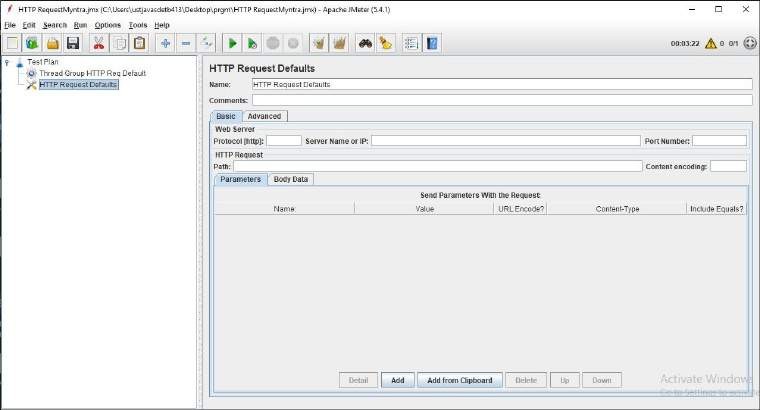
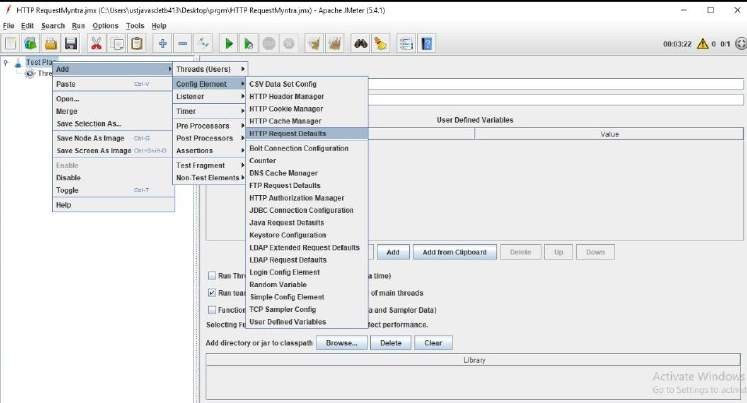
# Configure HTTP Request Defaults

‘HTTP Request Defaults’ is a very basic and key element of JMeter. This config element is used when all requests in the JMeter script are sent to the same server. You can add a single HTTP Request Defaults element under Test Plan with the proper server name or IP address in the field ‘Server Name or IP’. Now, when you add HTTP Request controllers, then leave the ‘Server Name or IP’ field empty. The controllers will inherit this field value from the HTTP Request Defaults element. Make sure you have provided correct method type in each HTTP Request.

It makes JMeter script simple and easily maintainable. If server name or IP changes, then you do not need to change those details in each HTTP Request. Just make changes in HTTP Request Defaults and that will be applicable for all the HTTP Requests.

It reduces the chances of missing the URL/port changes from HTTP Requests Steps:

* Create a Test Plan
* Add a Thread Group
* Add HTTP Request Defaults



* Name: To provide element name
* Comments: To provide arbitrary comments (if any)
* Protocol [http:]: Either http or https (Do not use ://)
* Server Name or IP: Domain name or IP address of the webserver without including prefix http://
* Port Number: Port number of Web-server. In case there is no port number then keep it blank.
* Path: It is a path to the resource. You can keep it blank for home page request or use

/.

Site used for testing is ,

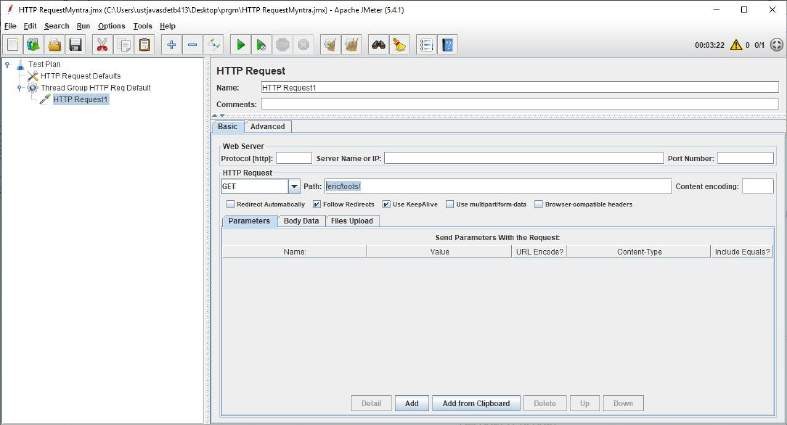


* Add some HTTP Request to the Thread Group

Doesn’t add the Server name/IP only need path

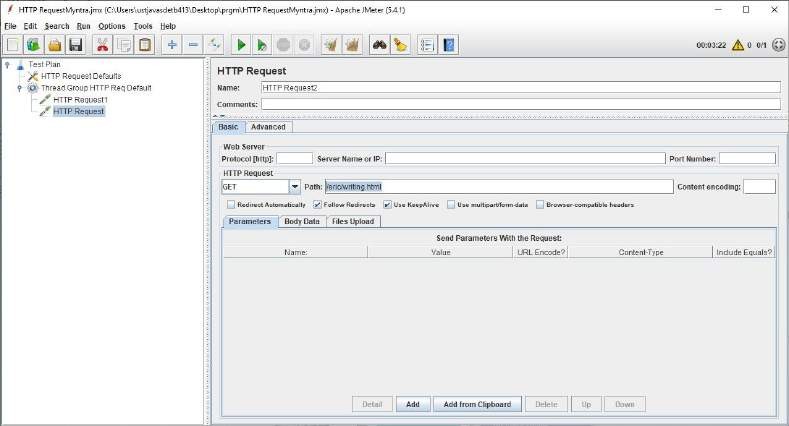
HTTP Request1





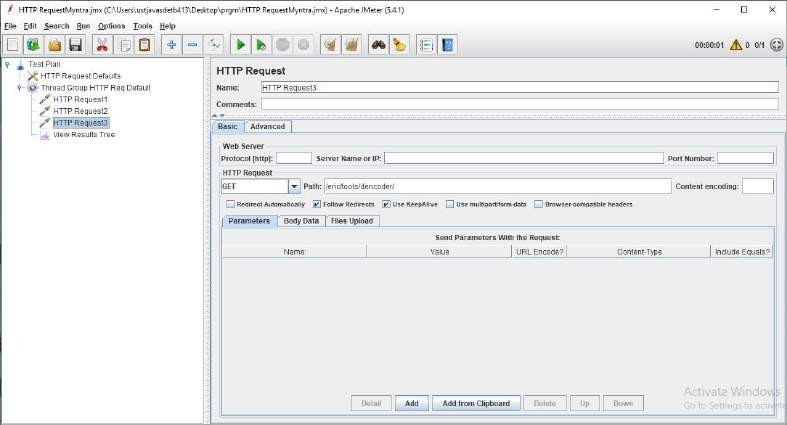
HTTP Request2



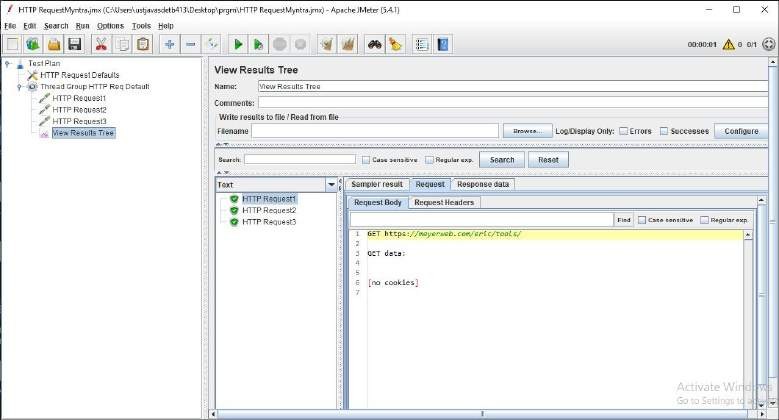


HTTP Request3





* Add a Listener -View Results Tree
* Run the Test plan

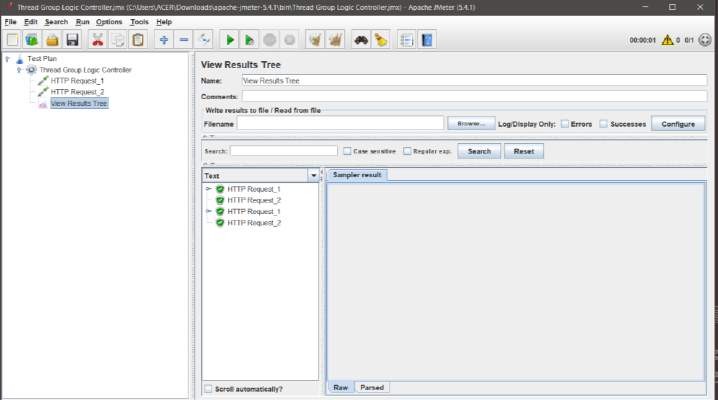


# Logic Controllers in JMeter

Logic controller helps us define the order of request which need to be executed in the JMeter.

Steps:

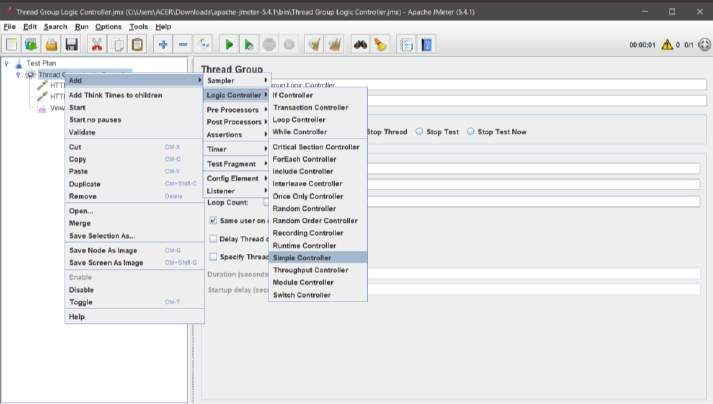
* Create a new Test plan.
* Add new Thread Group.
* Add Two HTTP Request.
* Add a Listener.
* Change the loop number of tread to execute as 2
* Run the test



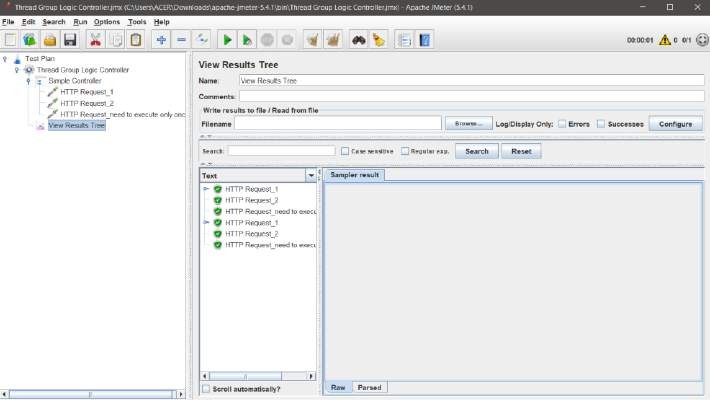
* Add Logic Controller – Simple Controller

Simple Controller doesn’t provide any functionality its just a container that contains user request.

Simple controller group the Thread request

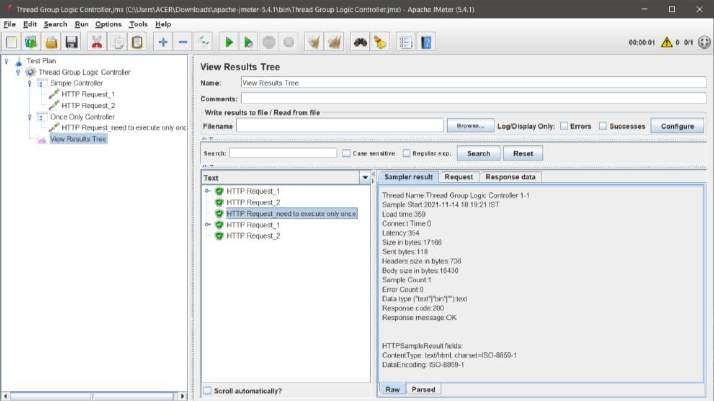
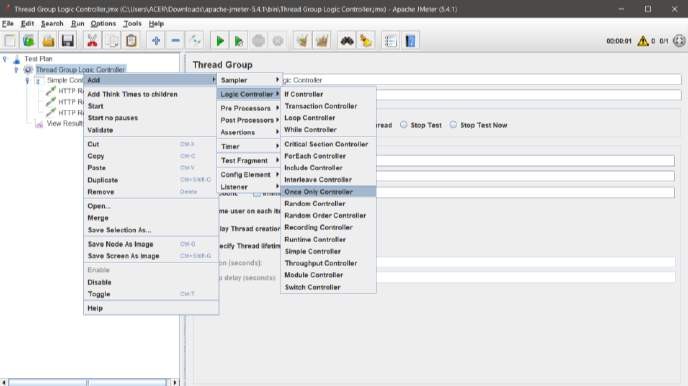


* Add some Thread Request inside the Simple controller and Run



* Add Once Only Controller

Once Only Controller provides functionality to run any user request to run only once per loop. Add once only controller to thread group by using below steps.



‘HTTP Request\_need to execute only once’ is executed only once because it is inside the

Once Only Controller

Module Controller

Module Controller provides functionality to run which module needs to run. Add module controller to thread group by using below steps.

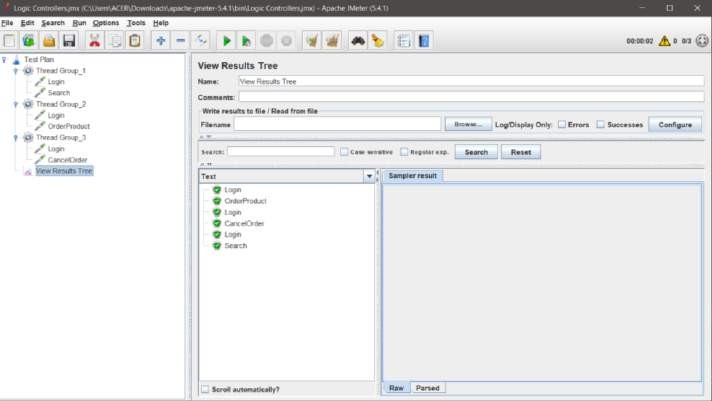
Right click on Thread Group >> Add >> Logic Controller >> Module Controller Examle:-

Creating some Scenario:

1. login and search a product 2.login and order a product 3.login and cancel order

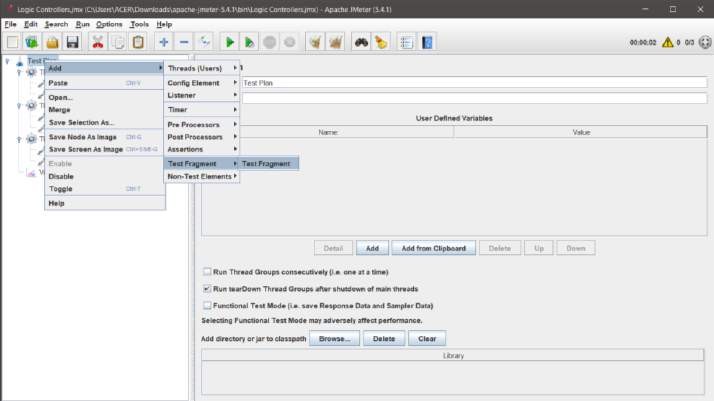
Steps:

* + Create a new Test Plan
  + Create 3 Thread Group for 3 Scenario
  + Create HTTP requestes
  + Add a Listener

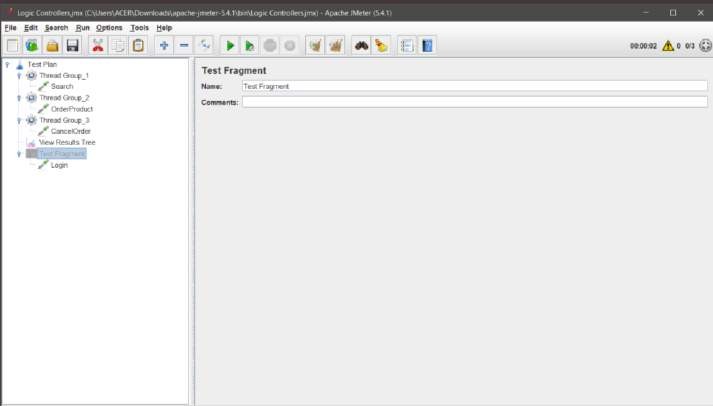


* + Add Test Fragment

Test Fragment element is a special controller which can be added directly under JMeter test plan like Thread Group. But It does nothing except holding other elements inside!! It gets executed only when it is referenced by a Module/Include controller from other Thread Groups.

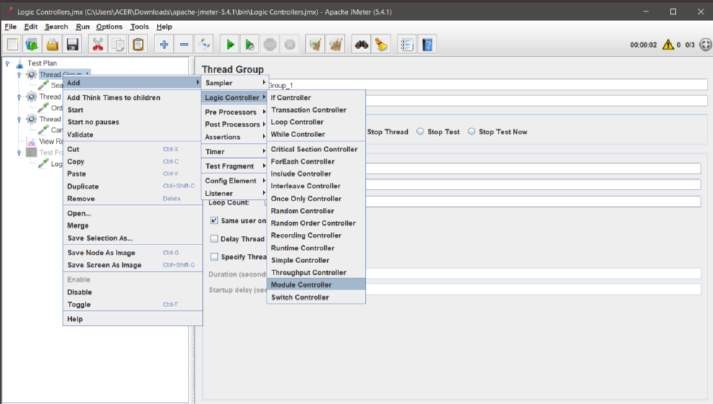


Place the request inside it [repeating Request is login]

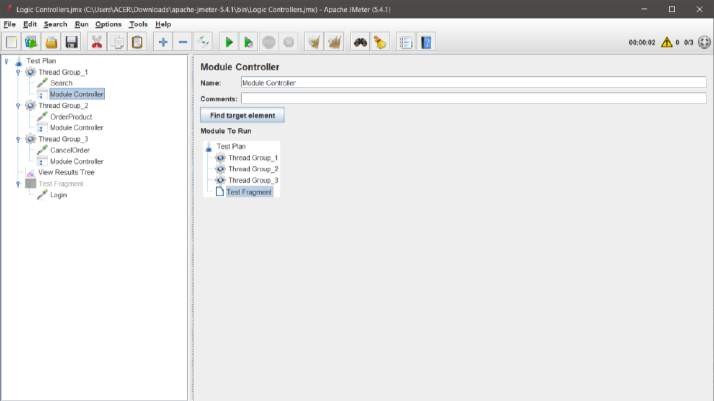


Request gets executed when called from controller like module controller/include controller

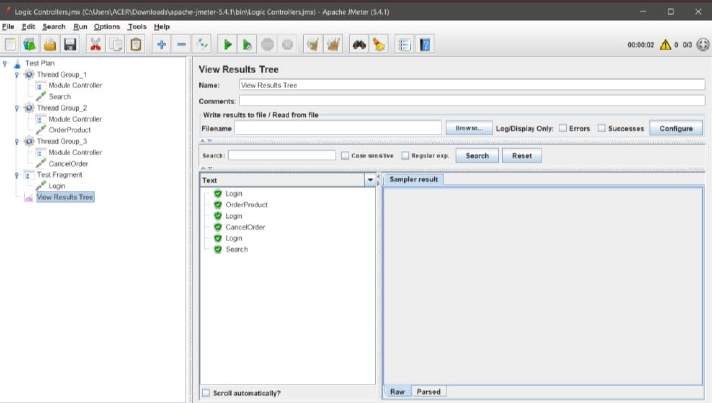
* + Add Module controller



Select Test fragment



Add controller in all Thread group [because every thread group has login] Run the Test plan [Same as previous Tread]



Include Controller

Include controller of JMeter is very useful if you want to break your test plan in small fragments. There are many different jmeter controllers available. Each of them has a different purpose as per its name.

Include controller is useful to include external test fragments in your test. Means you can include external test fragment in your software load test plan using include controller.

Include controller provides you the facility to use external test fragments in your test plan. So you have to save login steps as a test fragment and then you can use that test fragment in your all software load test plans using Include Controller

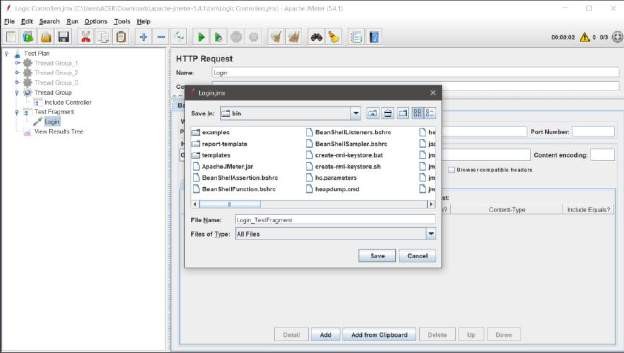
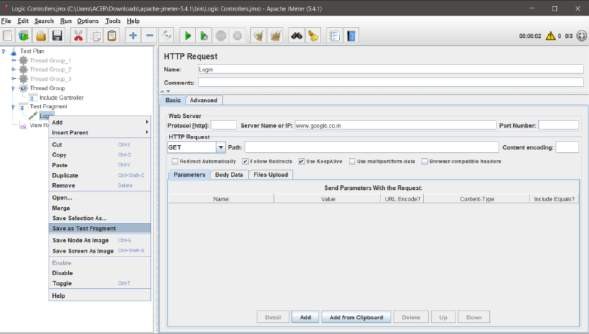
Example:

Scenario 1.Login 2.Search product

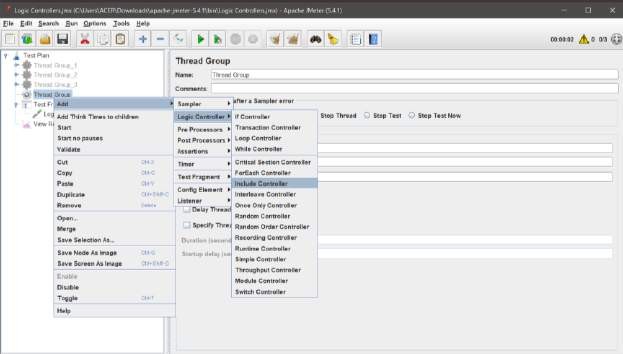
1. Cancel Product

Steps:

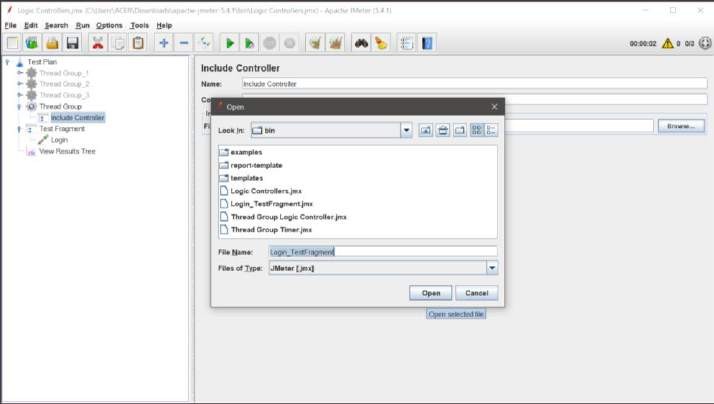
* + Add Thread Group
  + Save login as JMX file for include controller



* + Add include controller inside Thread group



* + Add File



* + Add requests inside the Thread Group and Run



# JMeter Pre & Post Processors

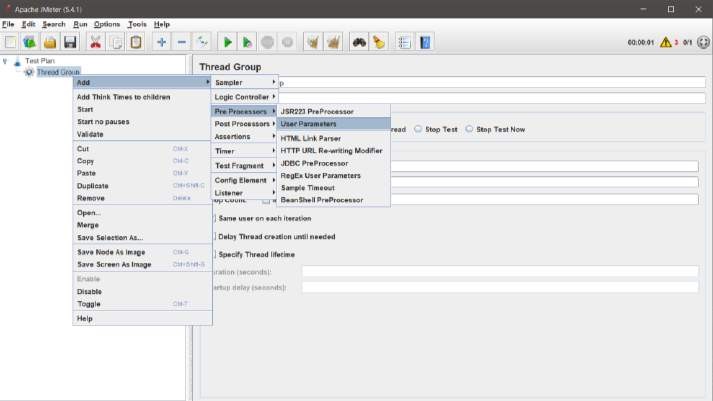
Pre Processor executes before the sampler execution. Post Processors are used to execute the response data from the server and to save the specific extracted values for later use.

User parameter

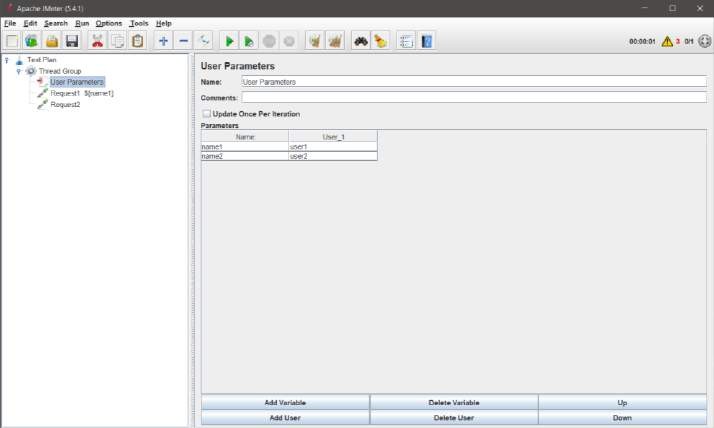
The User Parameters PreProcessor specifies user input parameters specific to individual threads. For each thread, the variable value will be used based on the order of that user thread in sequence.

Steps:

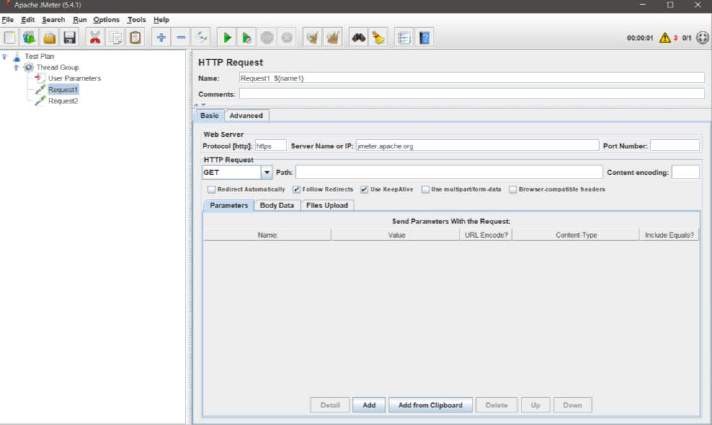
* + Create a Thread group
  + Add pre processor – User Parameters



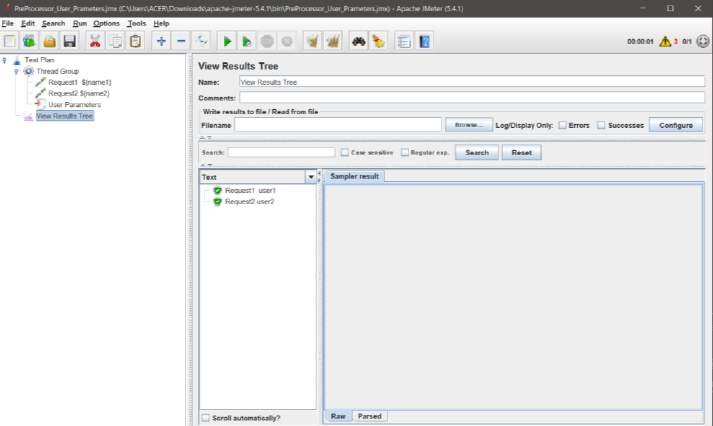
* + Add name and users in User parameter



* + Add Request



* + Run

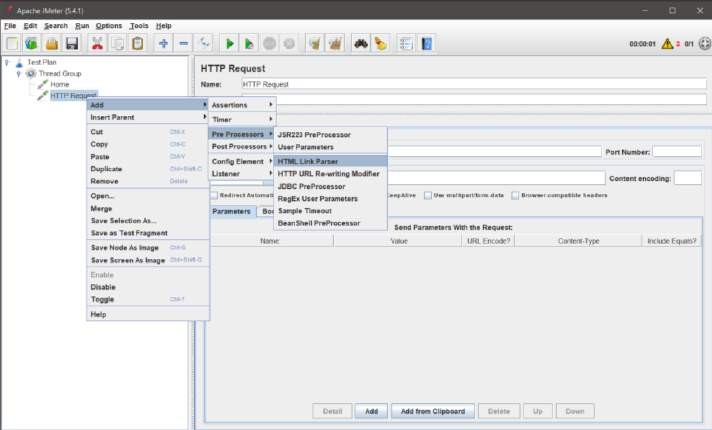


HTML Link Parser

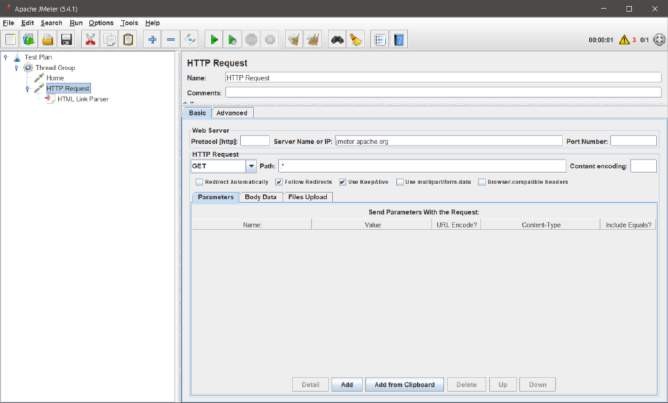
The HTML Link Parser PreProcessor can be used to parse a response, extract all the found links and request them further. This can be useful when the main goal of your script is to simulate web crawling.

Steps:

* + Create a Thread Group
  + Add a HTTP Request
  + Add another HTTP Request and add HTML Link Parser inside the request

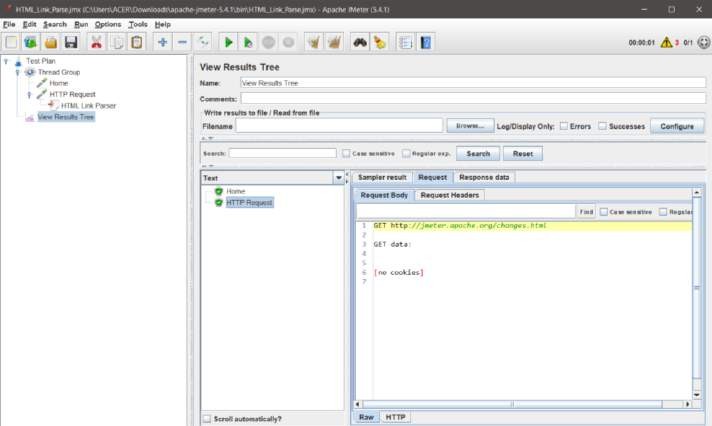


* + .\* take all the links in that html page

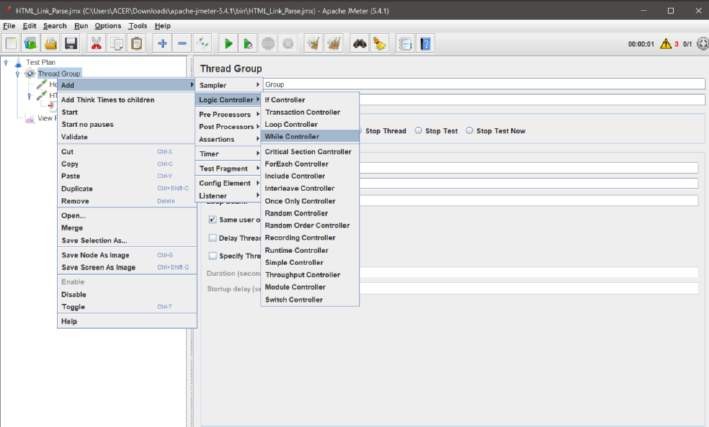


* + Run

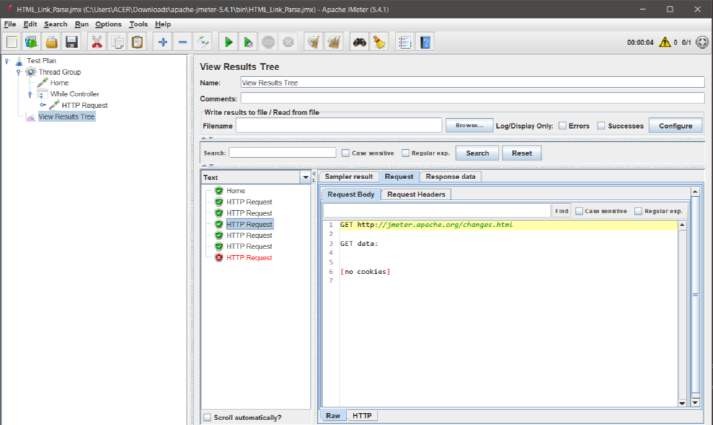
Http Resquest take a random link to test



* + If we wants to test all link the we place the HTTP request inside a While Controller



Run



Post-Processors

Post-Processors are actions that are performed after your sampler has been executed. You can use them to perform some actions on your response or extract some values out from the response and save them in a variable that can be used later.

Result Status Action Handler:

Result Status Action handler let the user select the action to be taken when sampler gets any

error.

Below actions can be taken:

 Continue

 Start next thread loop

 Stop thread

 Stop test

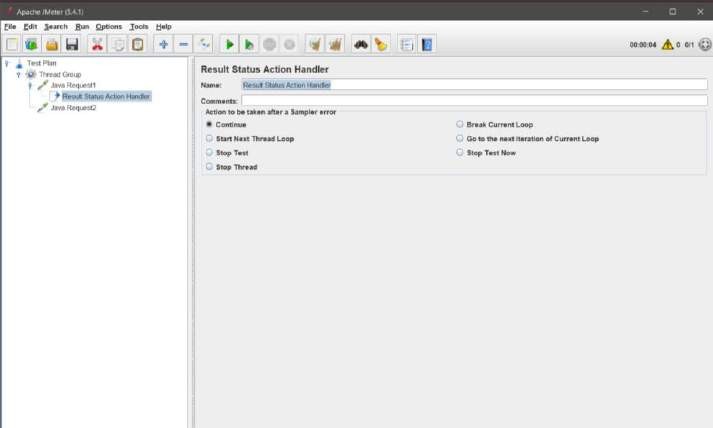
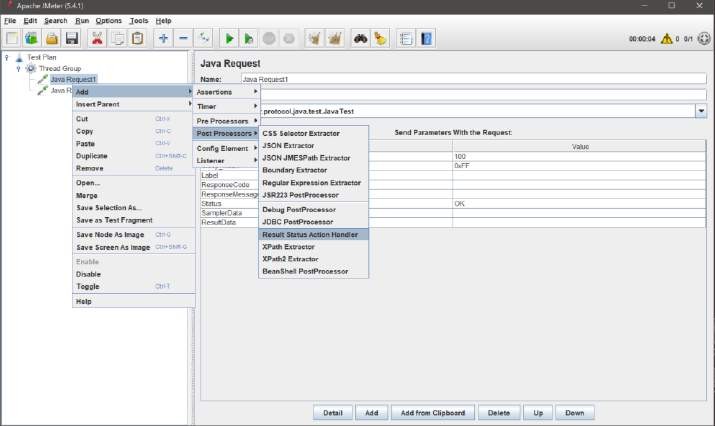
 Stop test now

Steps:

 Break Current loop

 Go to the next iteration of the current loop

* + Create Thread Group
  + Crete two Java Request
  + Add Post Processor - Result Status Action Handler



# JMeter Variables And Functions

Function – methods used to populate fields in any other elements of a test plan Syntax :

${ functionName}

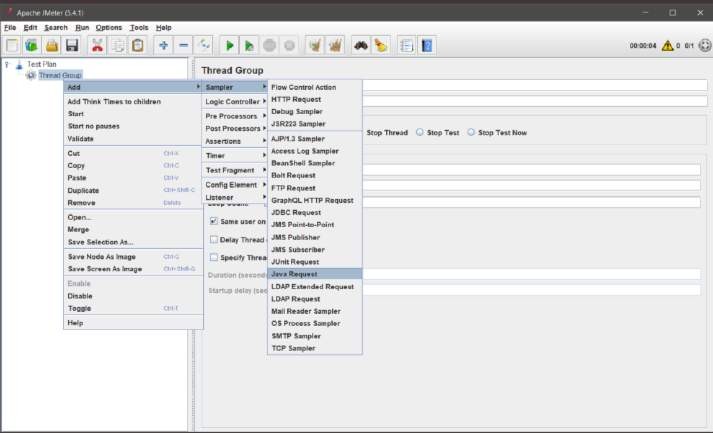
${ functionName(var1,var2,…)}

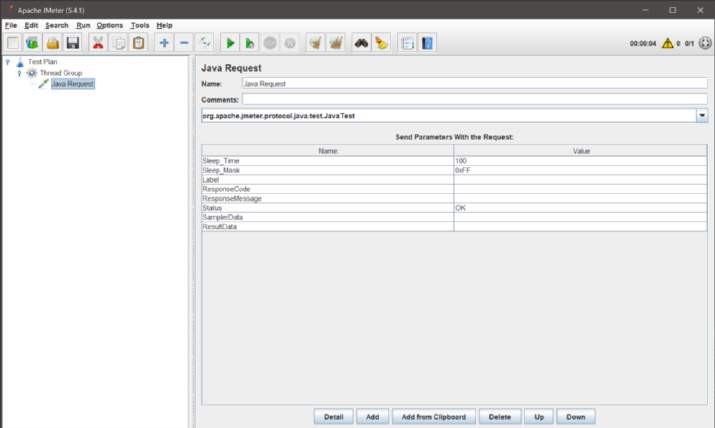
Variables – Containers that can store values, which can be referred in any element within a thread. Syntax :

${varName}

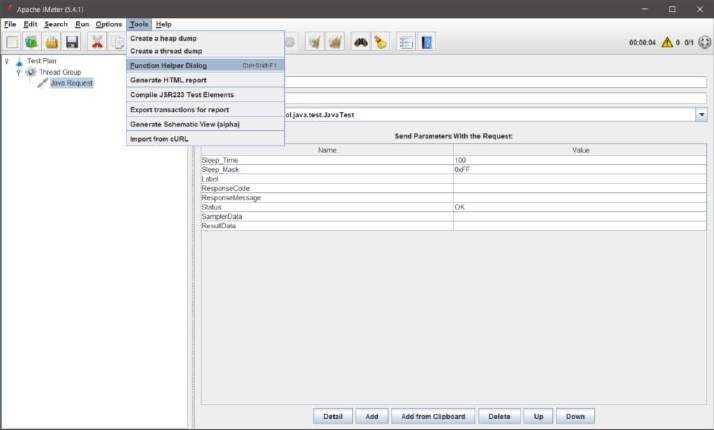
Steps:

* + Create Thread Group
  + Add Java Request

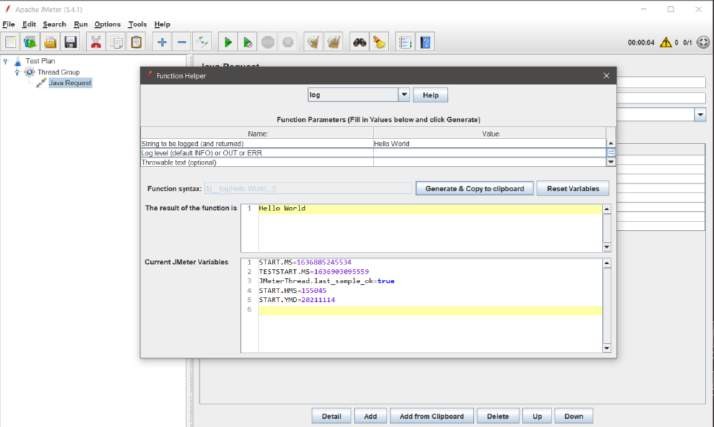




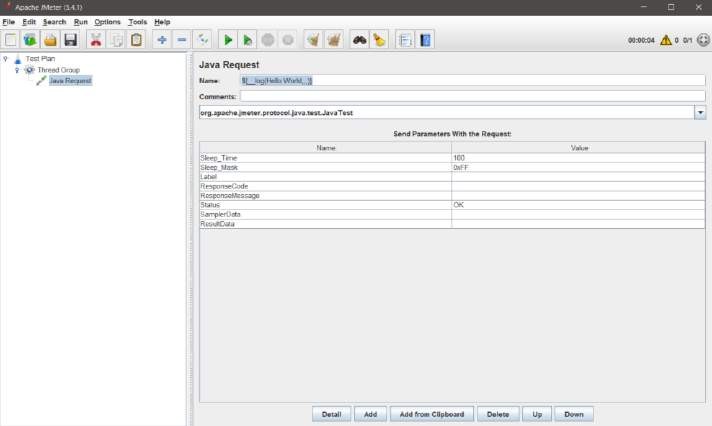
* + help of Functon Helper Dailogue we can copy fuction code



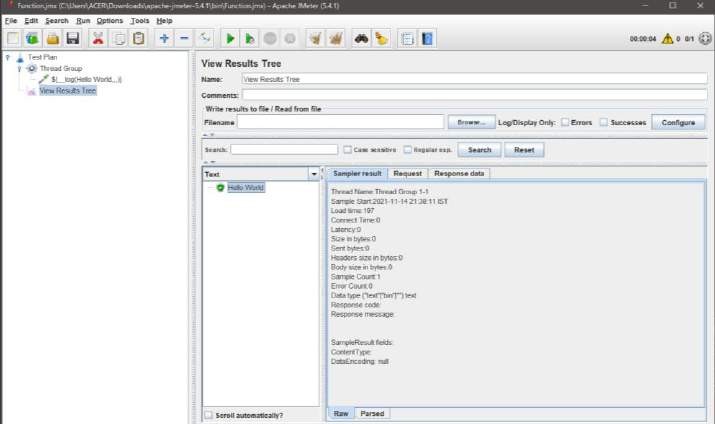
Select the fuction and click Generate & Copy to Clipboard



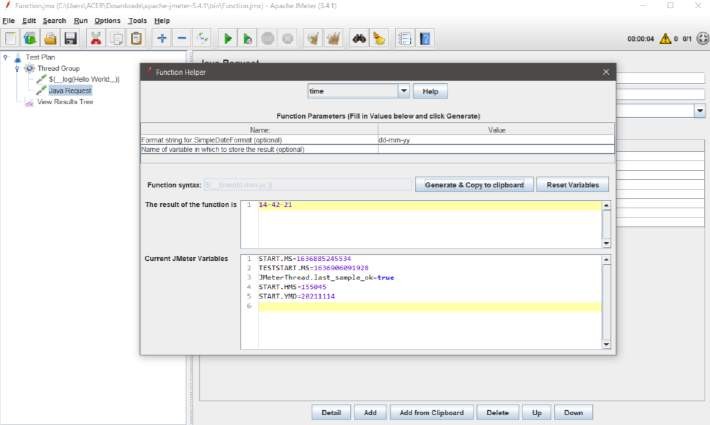
Paste it on Name of Java Request

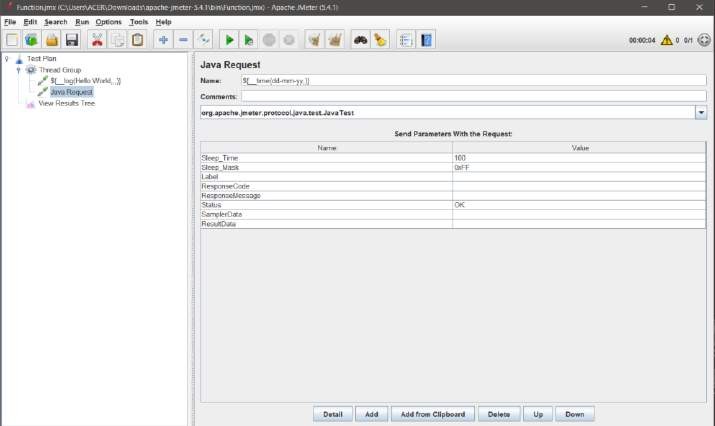


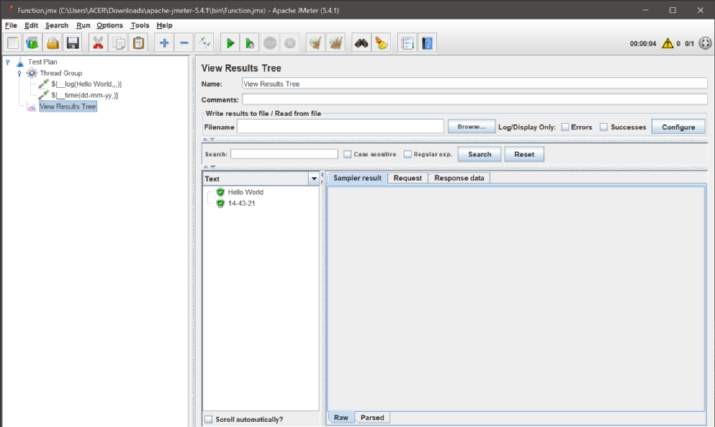
* + Run



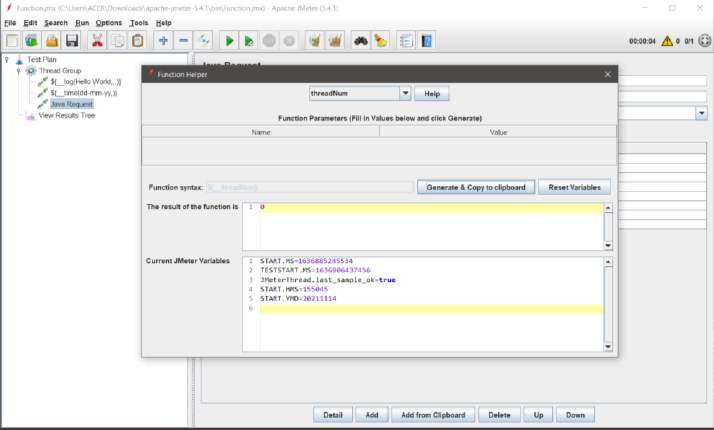
* + Add some other function – time



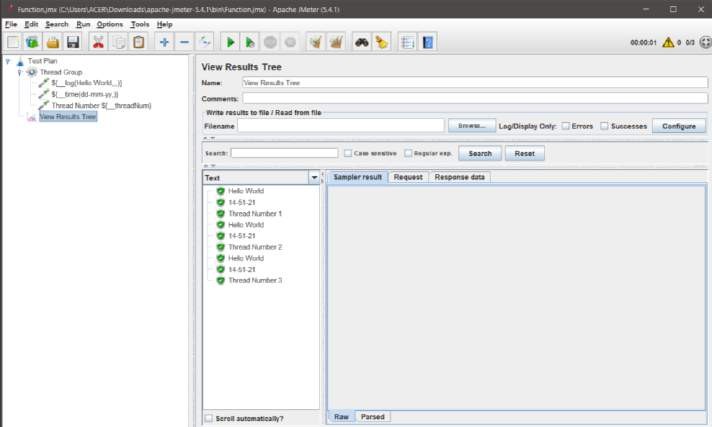




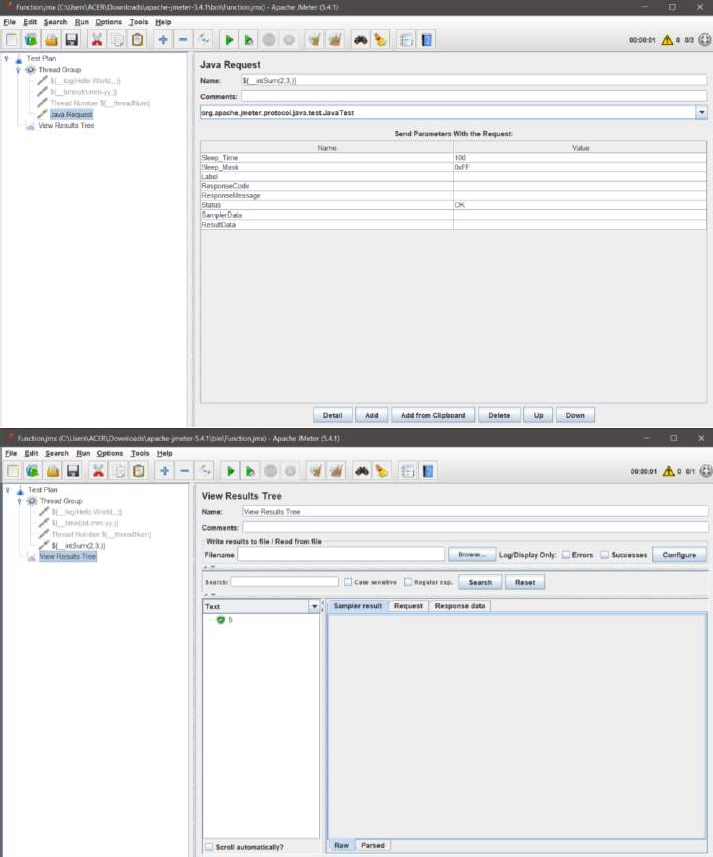
* + ThreadNum – number of current thread







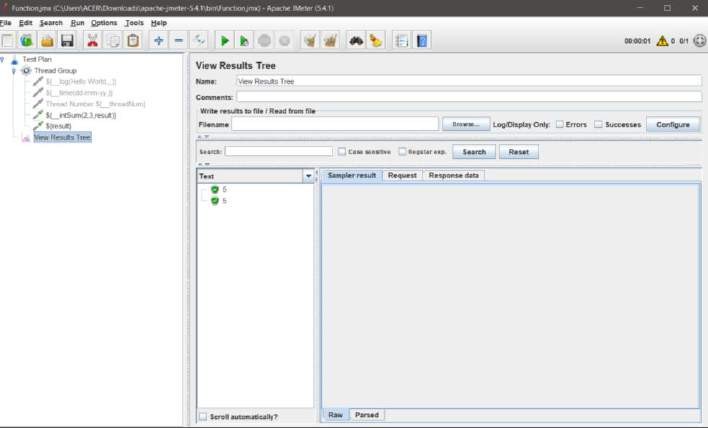
* + IntSum - sum of number

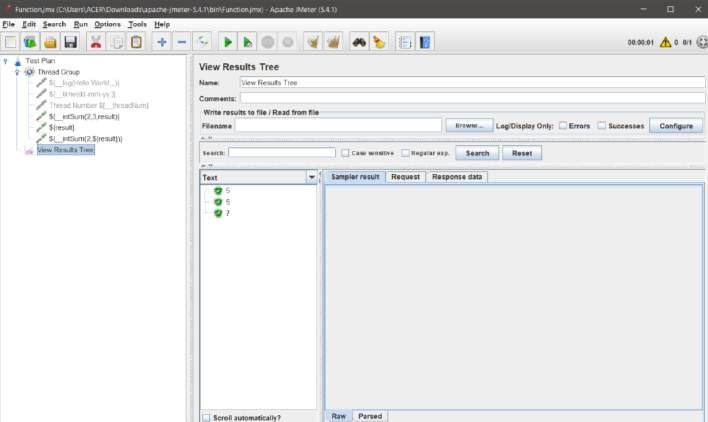


Store the result in variable



Variable can use in any where in this Thread Group



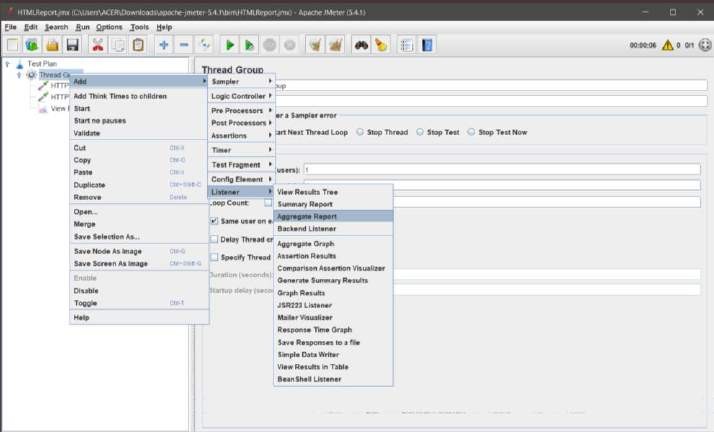


## HTML Report in JMeter

HTML reports are very intuitive. Most of the time development team and upper management need this report to evaluate performance of a particular site or application.

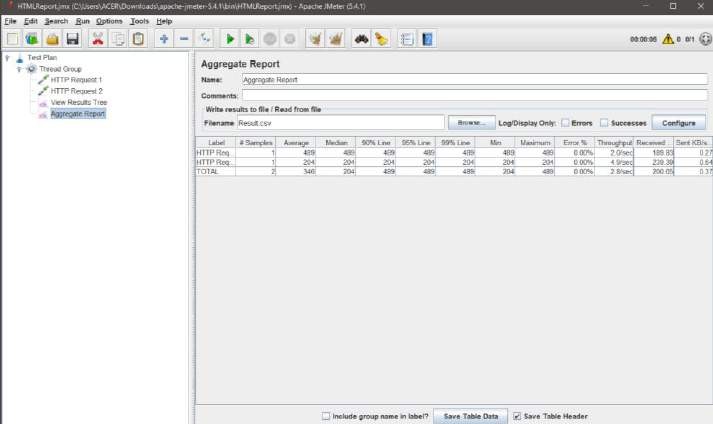
Steps:

* + Create a thread group
  + Add some HTTP requests
  + Add Listener
  + Add Aggregate Reports

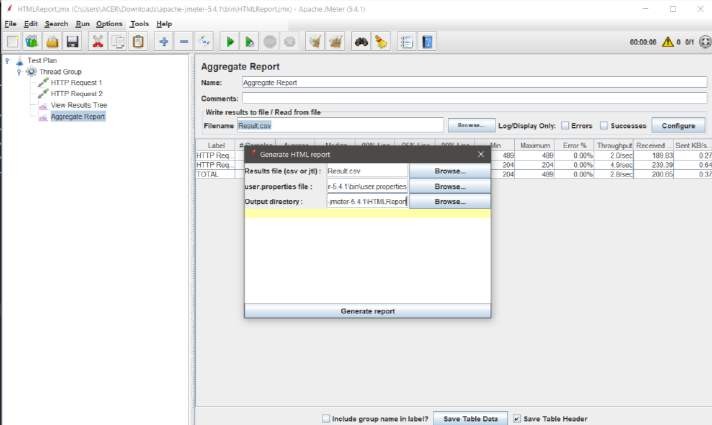
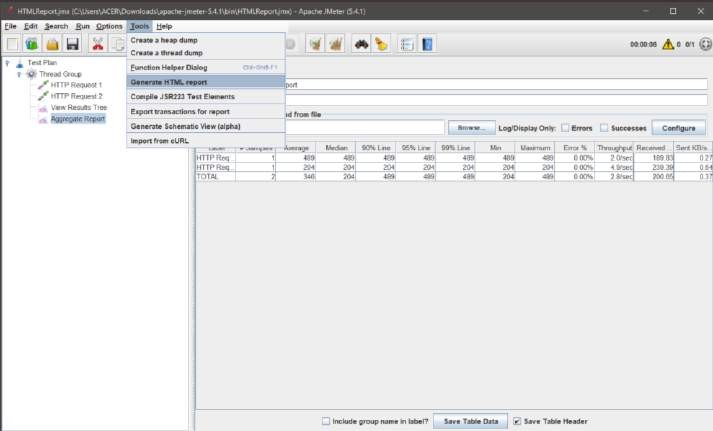


* + Save and Run

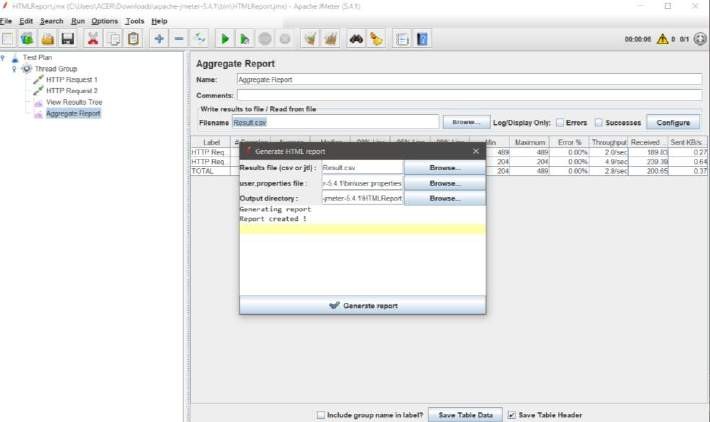
All result are sored in Result.csv file



* + Generate HTML report



Output directory must be empty.



* + Report

