Sba7

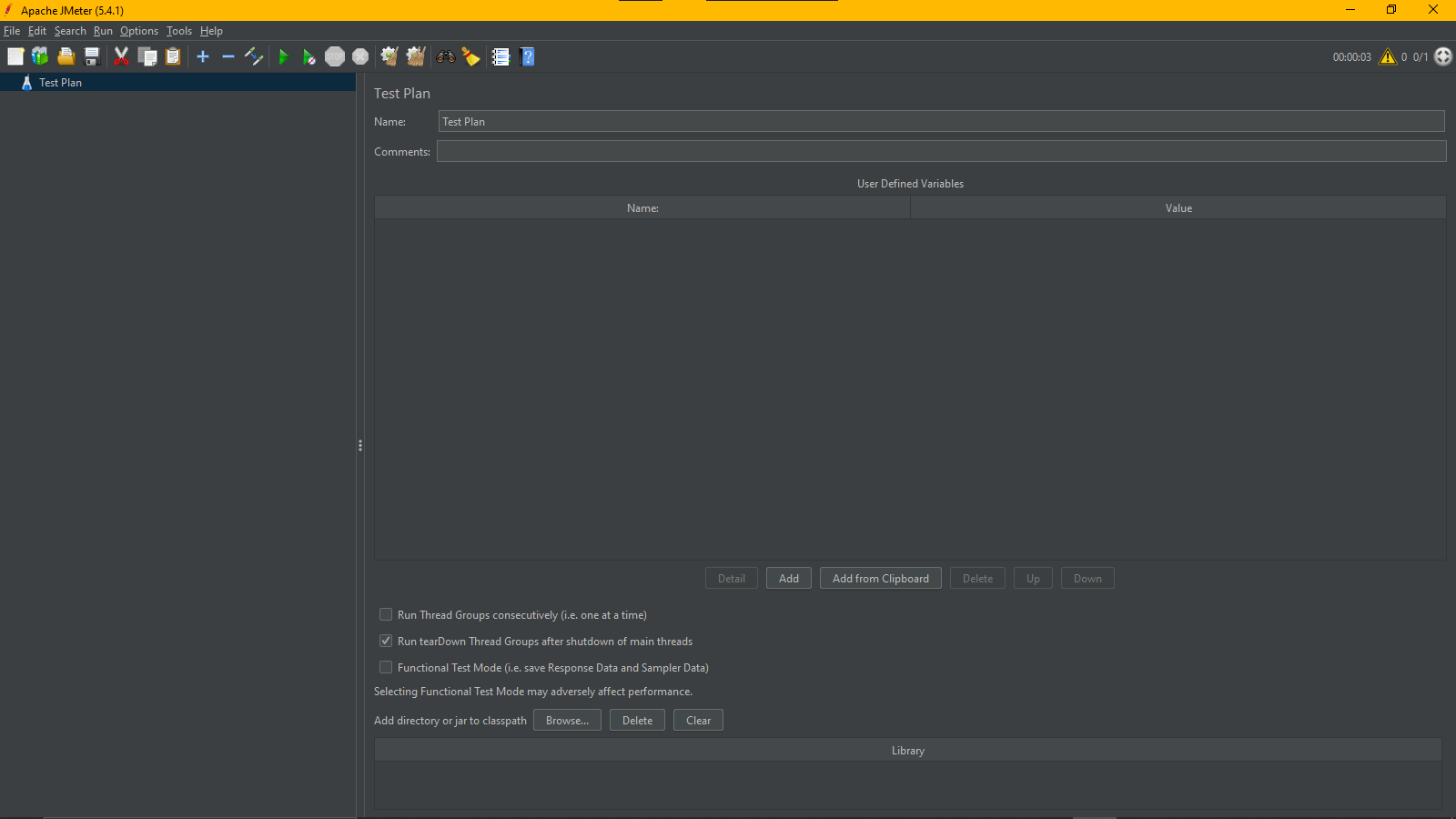
Jmeter

[*Apache Jmeter*](https://jmeter.apache.org/) Is An Open-Source, Pure Java Platform Software Which Is Designed To Load Test Functional Behavior And Measure Performance.

Initially, Jmeter Was Introduced For Load And Performance Test Web Applications, But Later On Its Scope Has Widened And Can Perform Load And Performance Test On Web Pages, Web Applications And Static Or Dynamic Resources Like *Database, Rest Webservices, Ldap, Java Objects* And More.

**Test Plan**

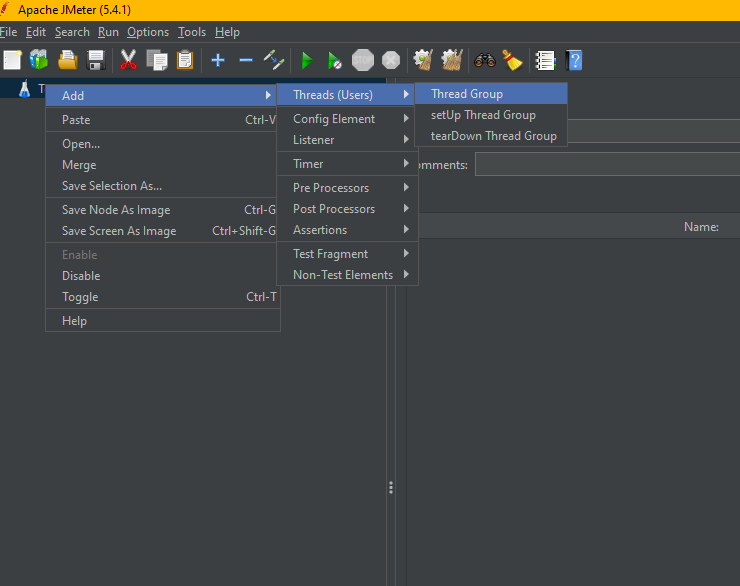
The Test Plan Is Where The Overall Settings For A Test Are Specified.

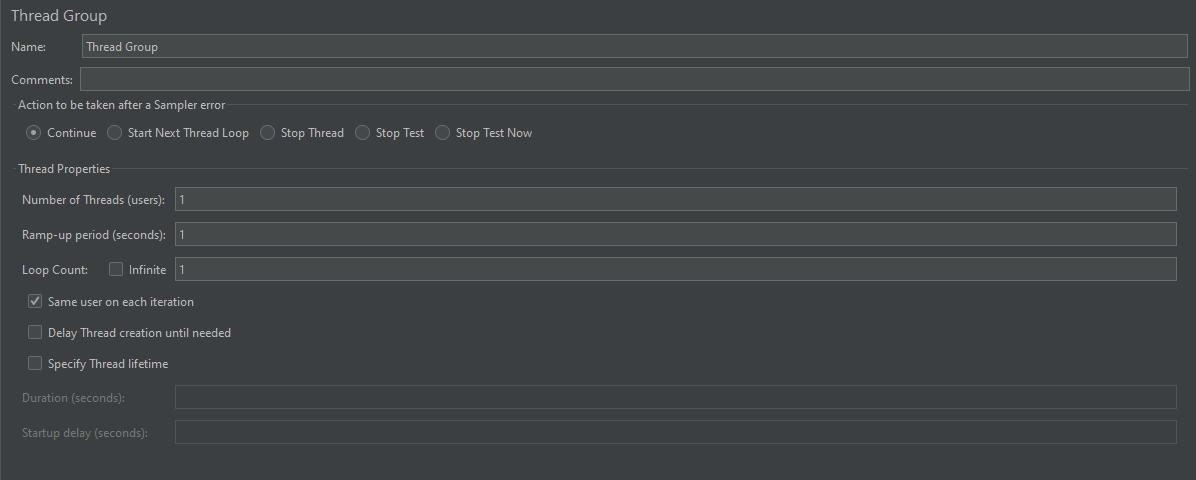


**Elements In A Test Plan**

**Thread**

A Thread Group Defines A Pool Of Users That Will Execute A Particular Test Case Against Your Server. In The Thread Group Gui, You Can Control The Number Of Users Simulated (Number Of Threads), The Ramp Up Time (How Long It Takes To Start All The Threads), The Number Of Times To Perform The Test, And Optionally, A Start And Stop Time For The Test.

****

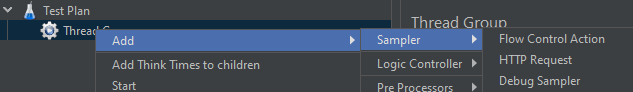
****

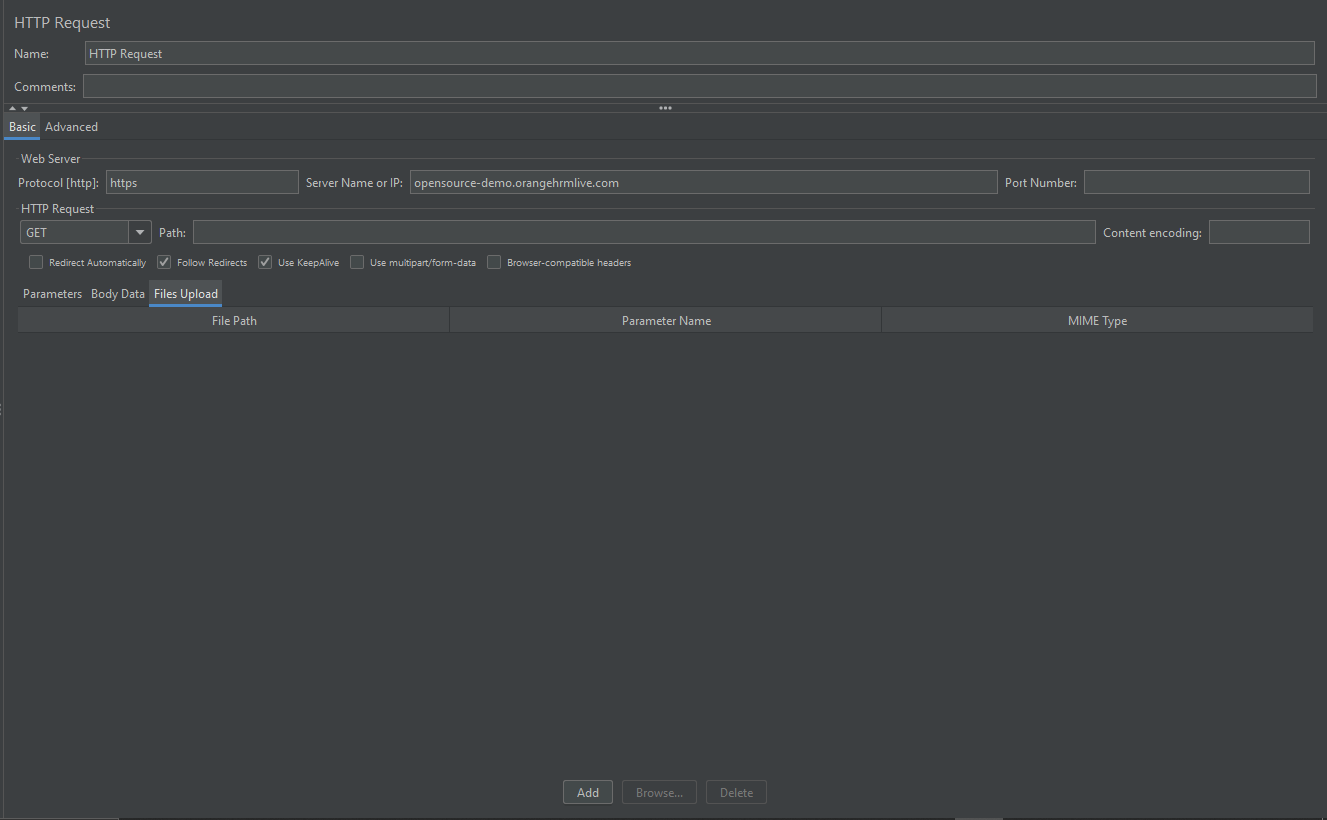
**Sampler**

Samplers In Jmeter Allows Jmeter To Send Different Types Of Requests To A Server

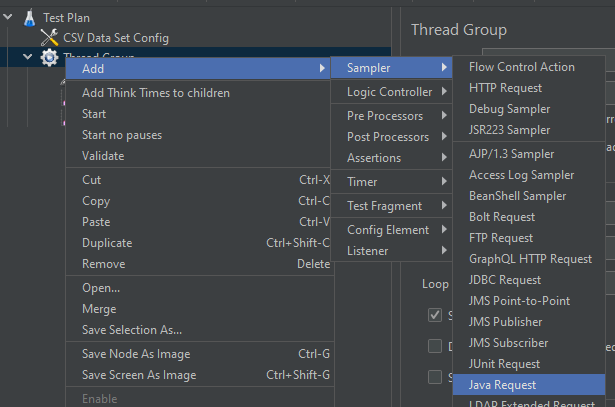
* **Http Request**

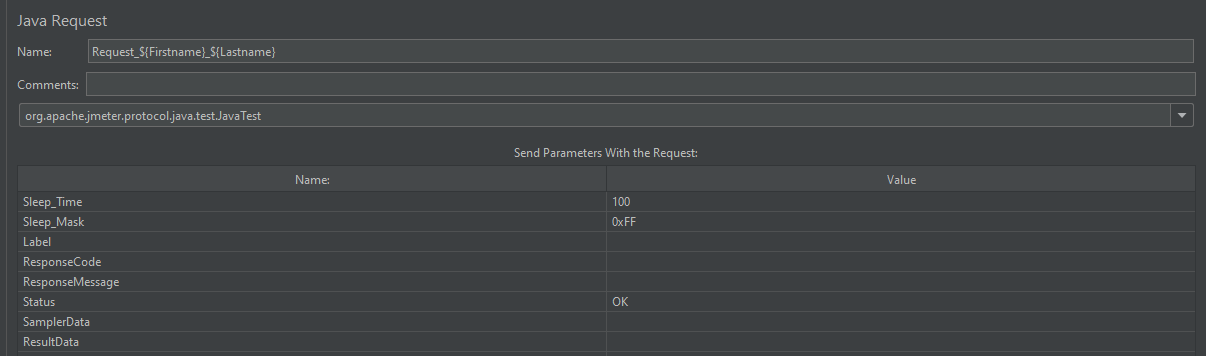
 A Sampler That Lets You Send An Http/Https Request To A Web Server For Load Testing. There Are Different Methods The Sampler Is Able To Use, Like: Get. Post.



****

* **Java Request**

****

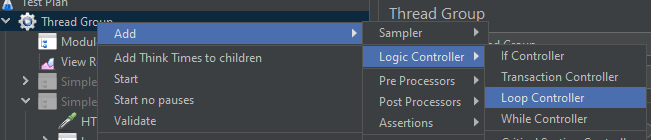
****

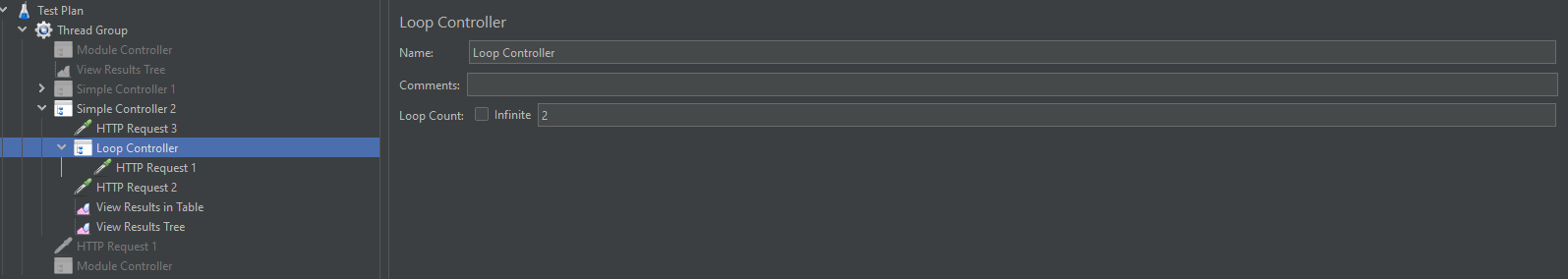
**Logic Controller**

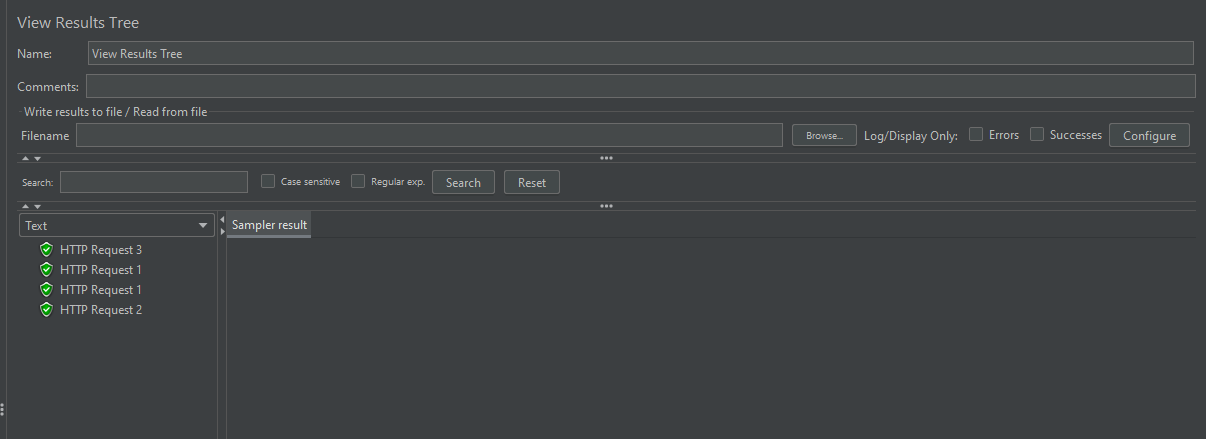
Logic Controllers Are The Test Plan Elements That Are Used To Customize The Order Of Processing Of Samplers And Other Elements Added As Child. Primarily, Logic Controllers Are Used With Sampler Requests To Perform Various Customization Like – Altering Their Order Of Processing, Grouping Them As A Single Transaction Or Running The Requests In Loop Etc.  
  
Steps To Launch A Logic Controller-**Right Click On Thread Group -> Hover Over Add -> Hover Over Logic Controllers -> Click On The Required Logic Controllers**

* Loop Controller

It Allows To Execute The Operations Specified As Child Elements In A Loop With Iteration Value Specified In Its Control Panel.

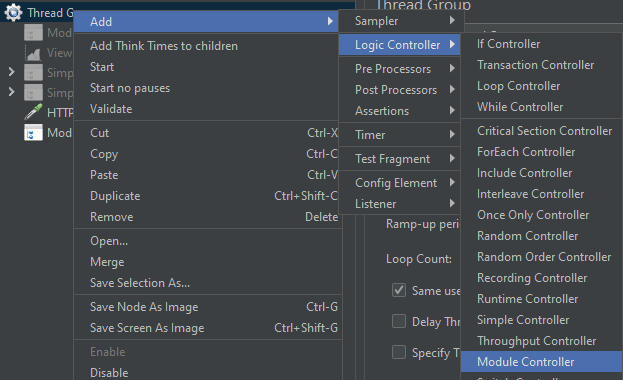
****

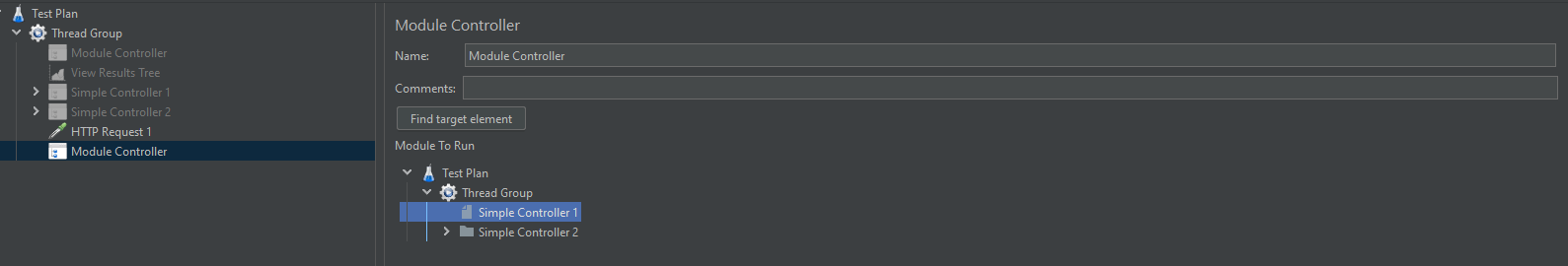


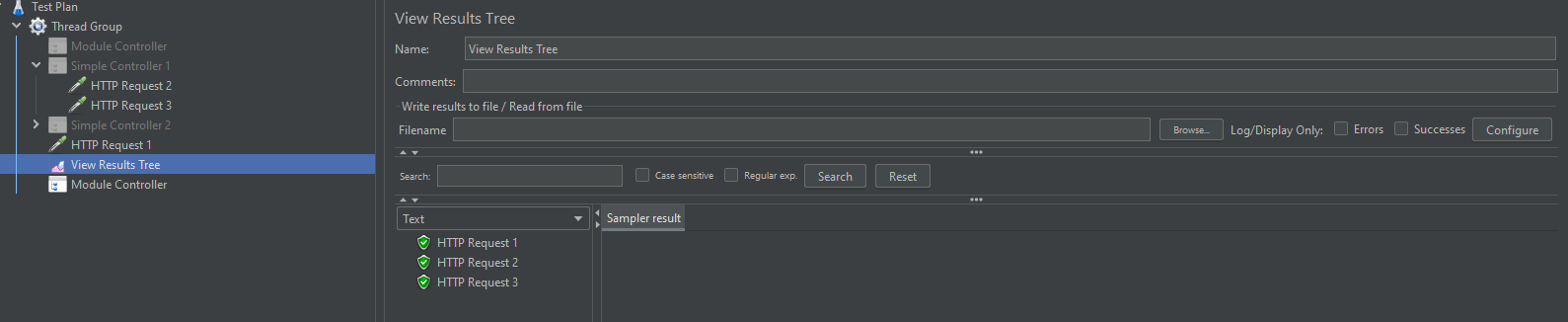
****

* Module Controller

Using Module Controller, We Can Reuse A Test Fragment (E.G., A Sampler) Into Our Script Again By Selecting The Module From The Module Controller’s Control Panel.

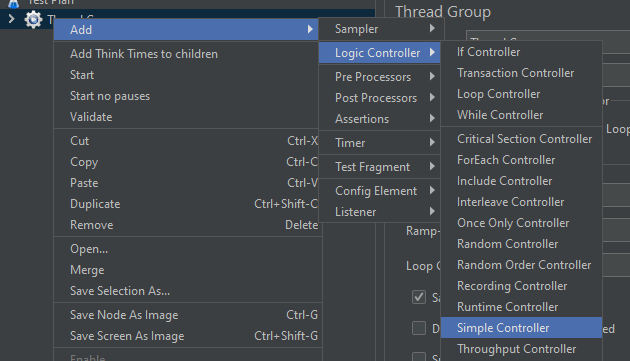


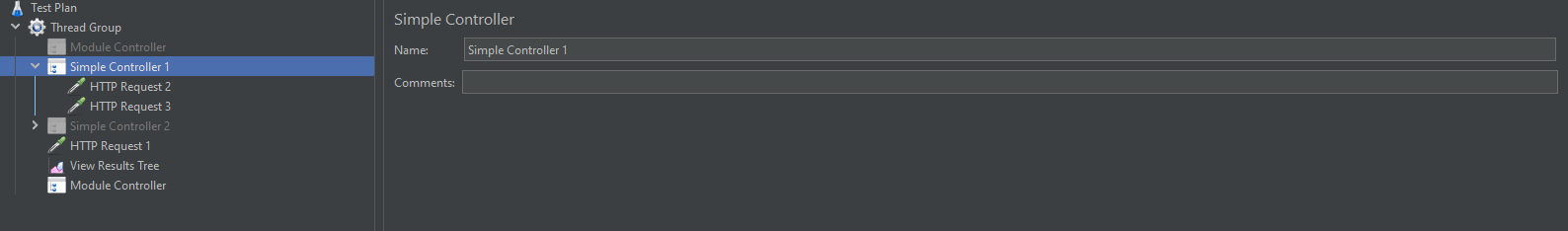




* Simple Controller

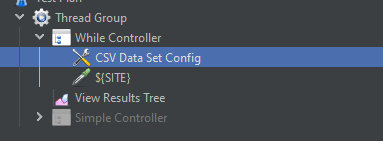
This Controller Is Just A Placeholder For Grouping And Ordering The Different Elements Of The Test Plan.





* While Controller

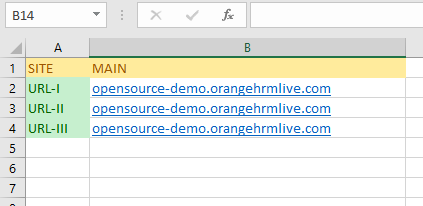
The While Controller Is Used To Run The Child Elements Inside It Till The Value Specified In Its Control Panel Is Evaluated To False. Steps To Launch A Logic Controller- **Right Click On Thread Group -> Hover Over Add -> Hover Over Logic Controllers -> Click On The While Controller.**

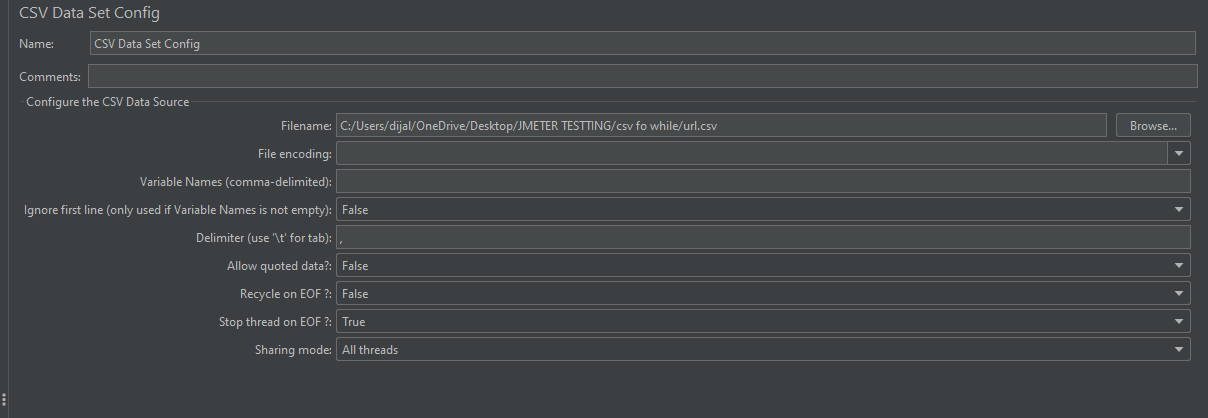


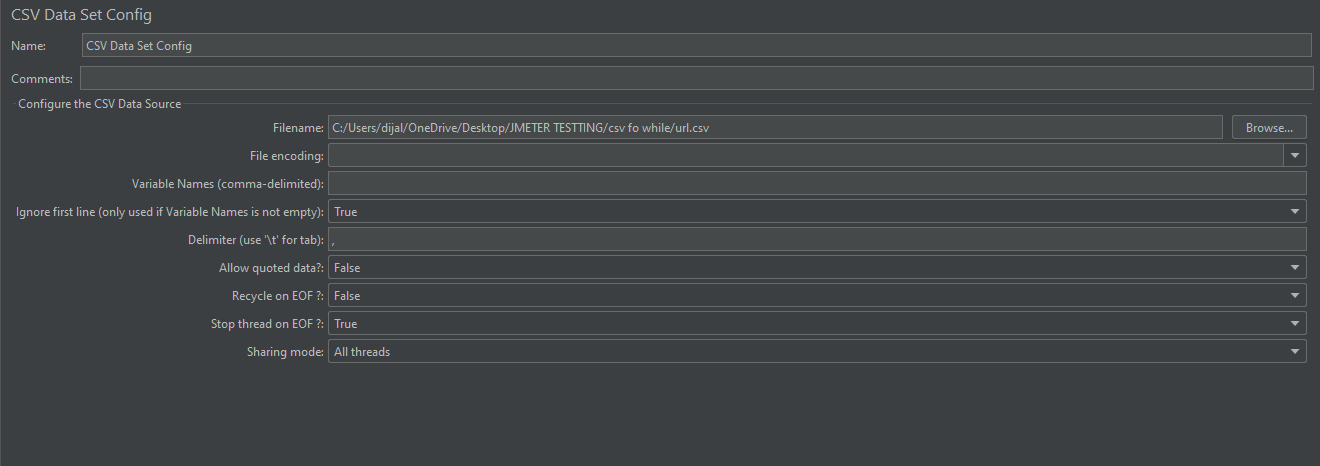
In Here We Take The Urls From A Csv, That Is Why We Use Csv Dataset Config,

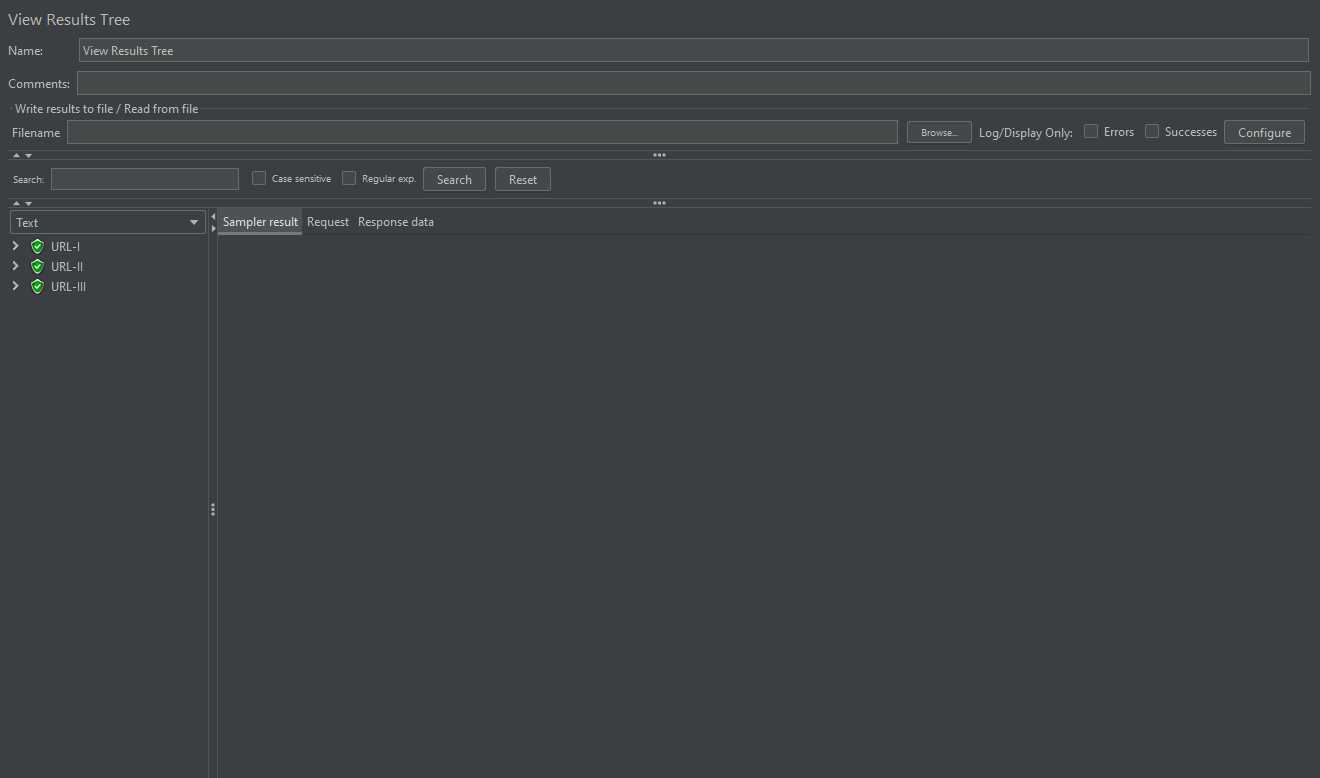
And We Use A Http Request Too.

In The Csv,



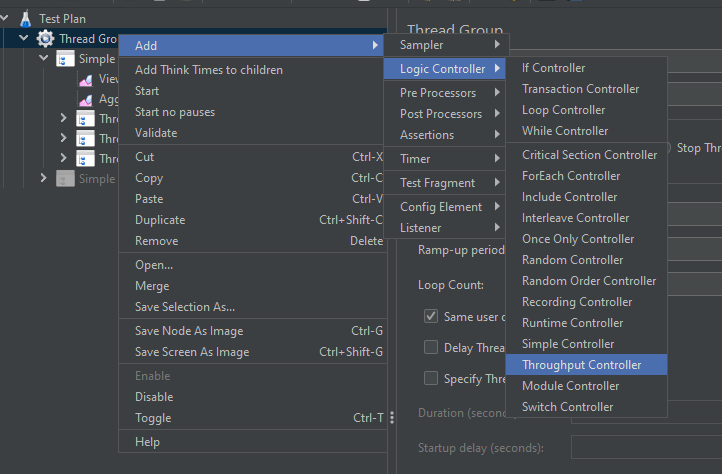


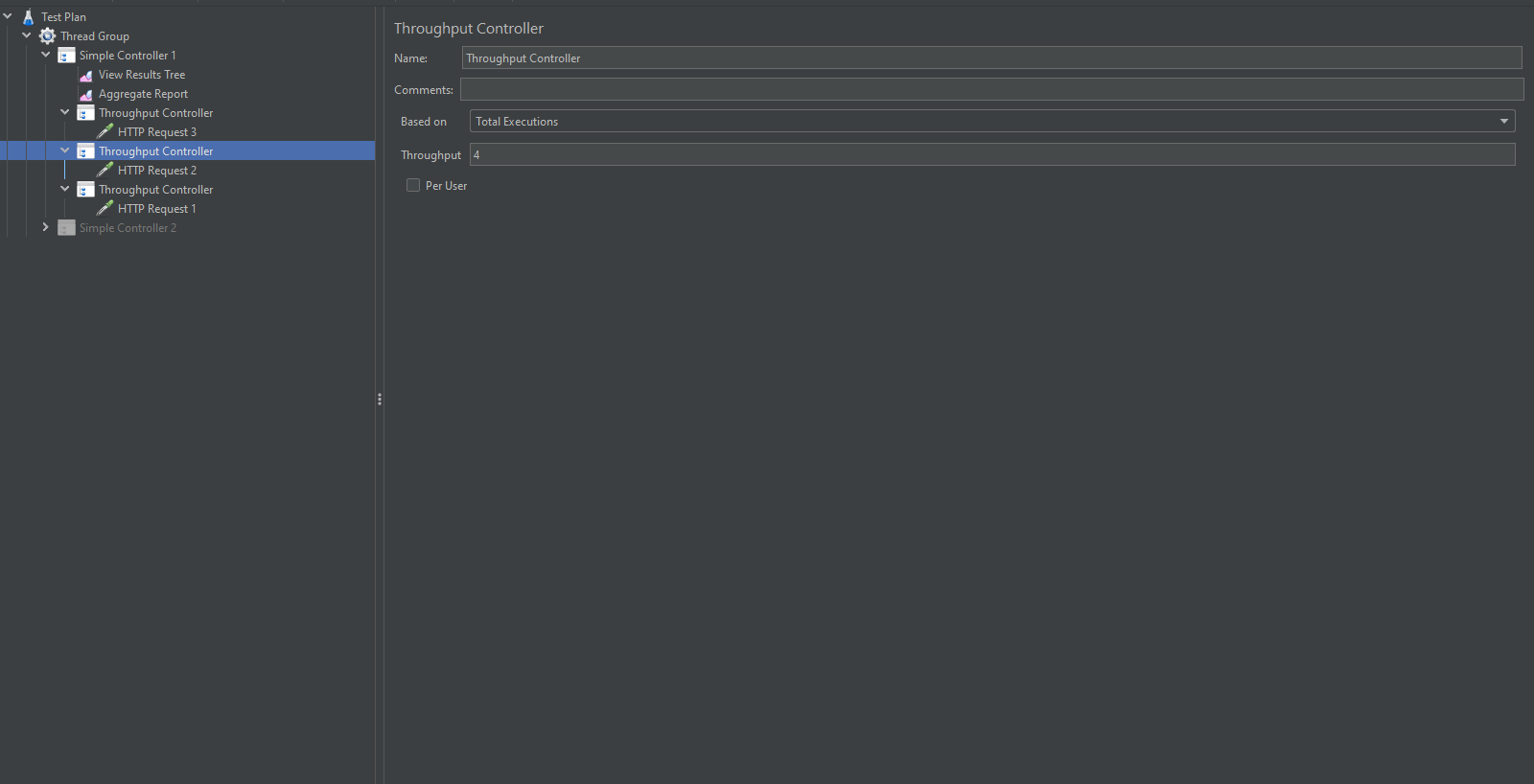


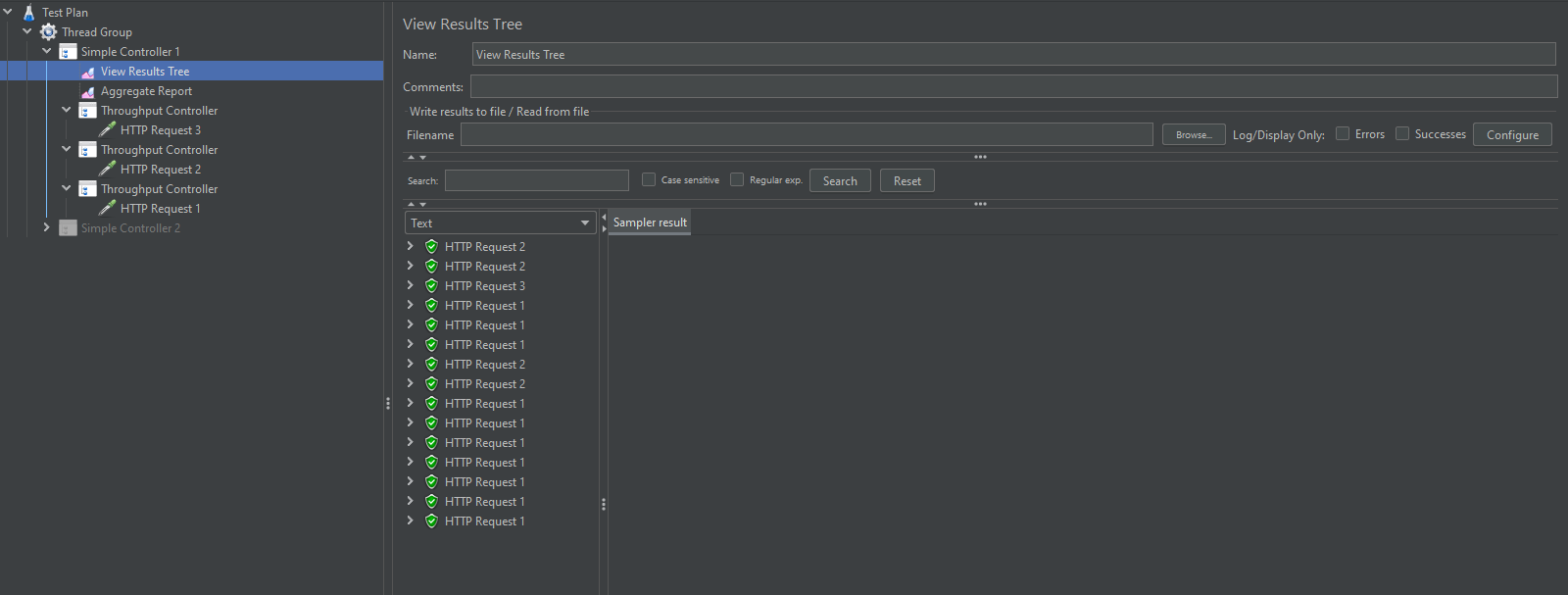


* Throughput Controller

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





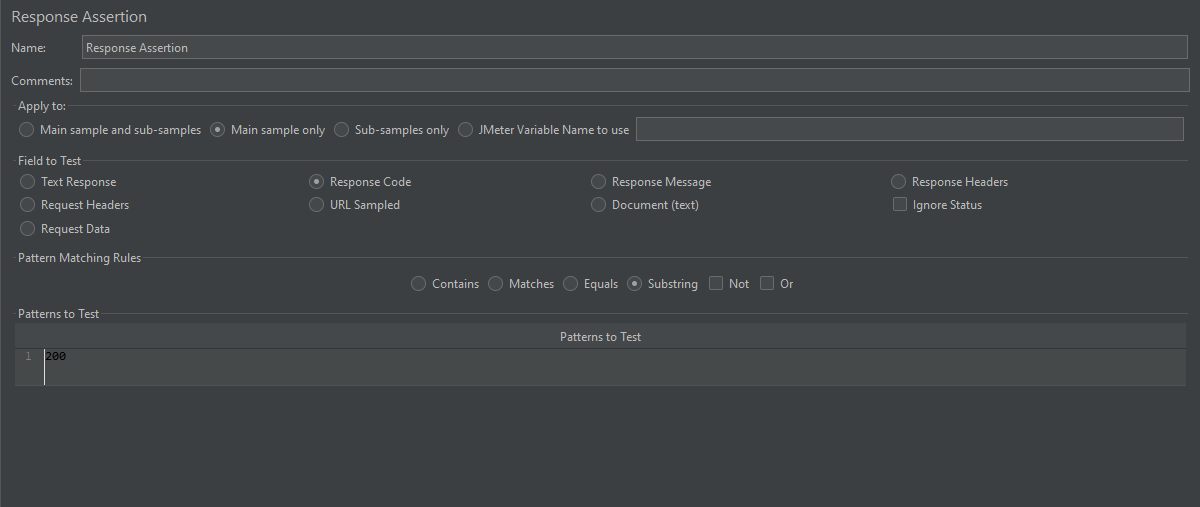


**Assertions**

* **Response Assertion**

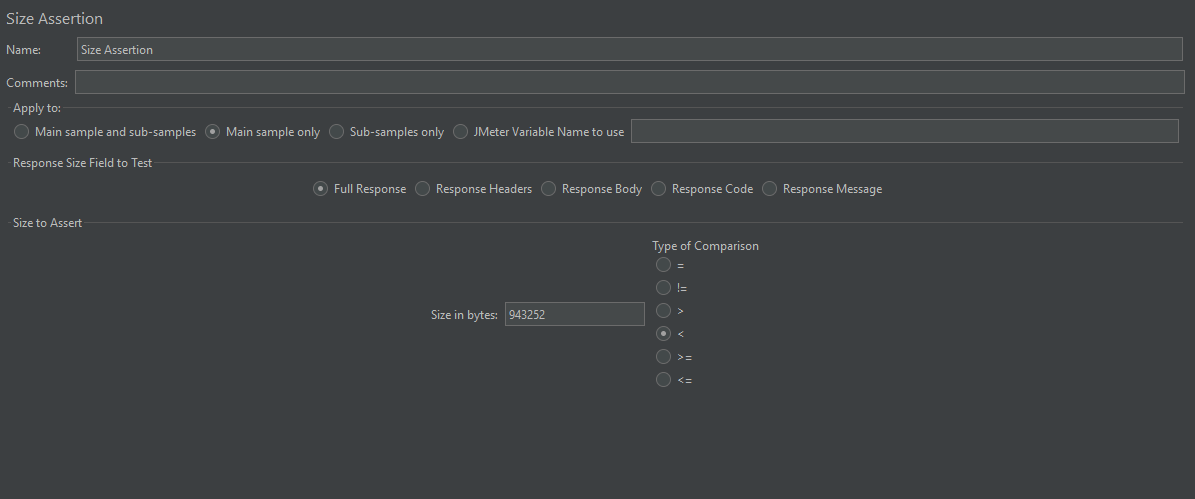
The Response Assertion Used In Test Scripts To Validate A Pattern In The Response Body, Header, Code, Message Etc. There Are Different Pattern Matching Rules To Validate The Response Like-

* Contains – If The Response Text Contains The Regular Expression To Be Matched
* Matches – If The Whole Response Text Matches The Regular Expression
* Equals – If The Whole Response Text Matches The Pattern(Not Regular Expression But The Pattern String)
* Substring – If The Response Text Contains The Pattern(Not Regular Expression)
* Not – To Check That The Pattern Is Not Present In The Response Text



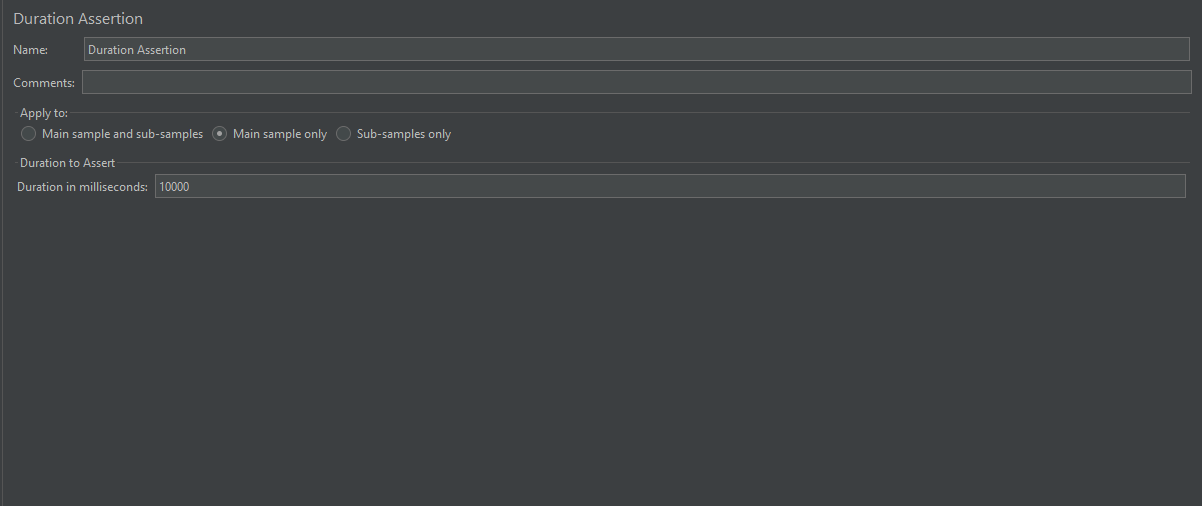
* **Size Assertion**

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



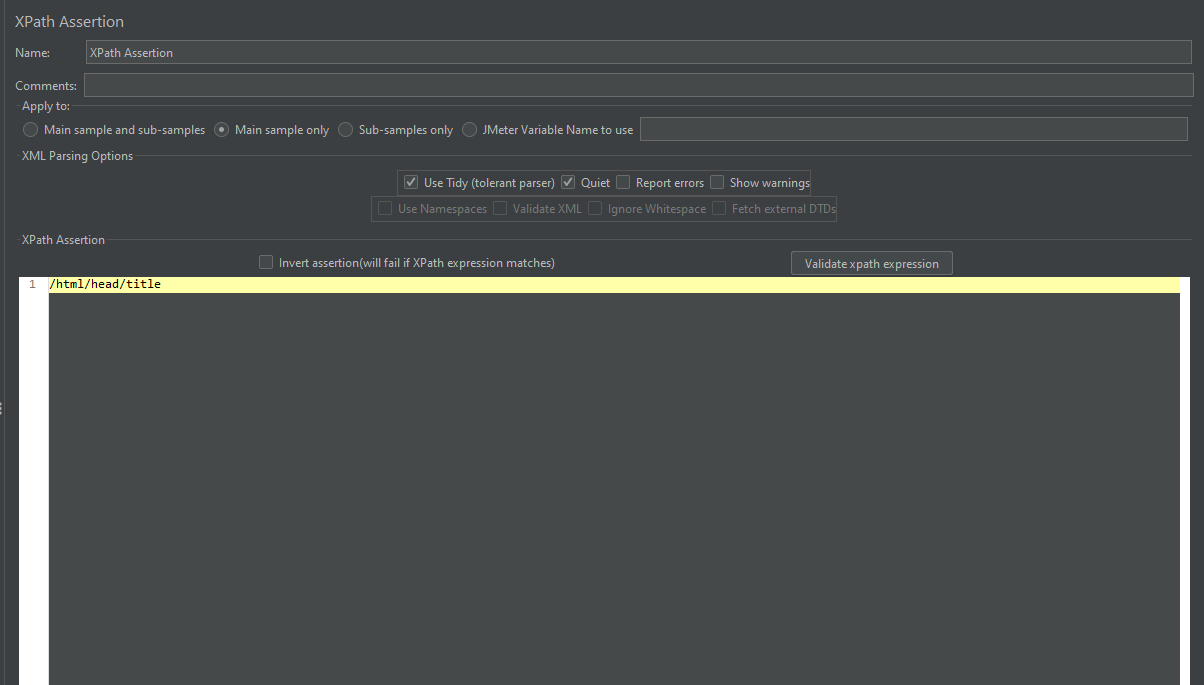
* **Duration Assertion**

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

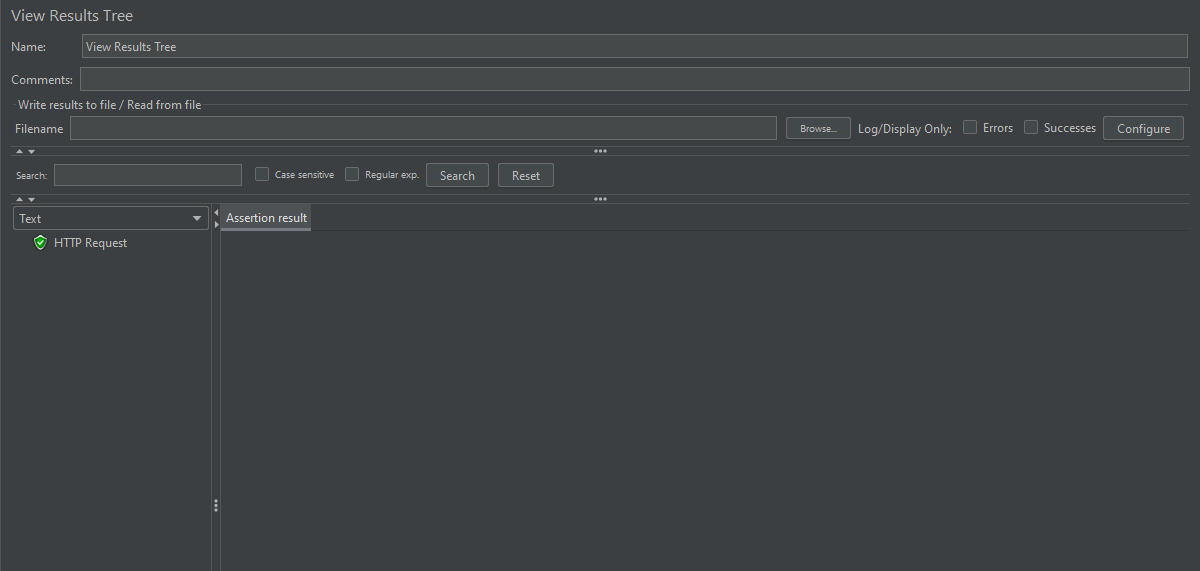


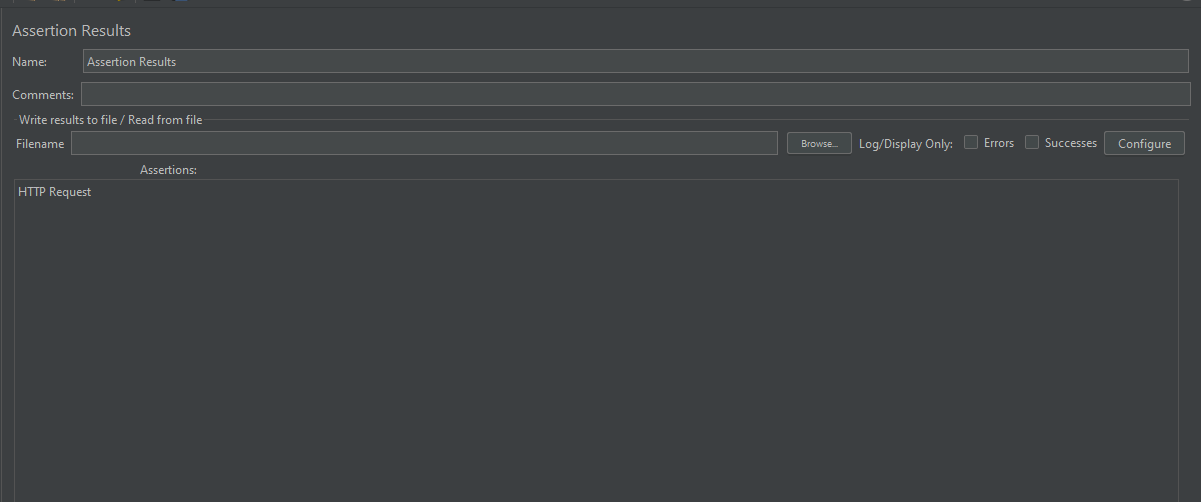
* **Xpath Assertion**

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



**Results**

****

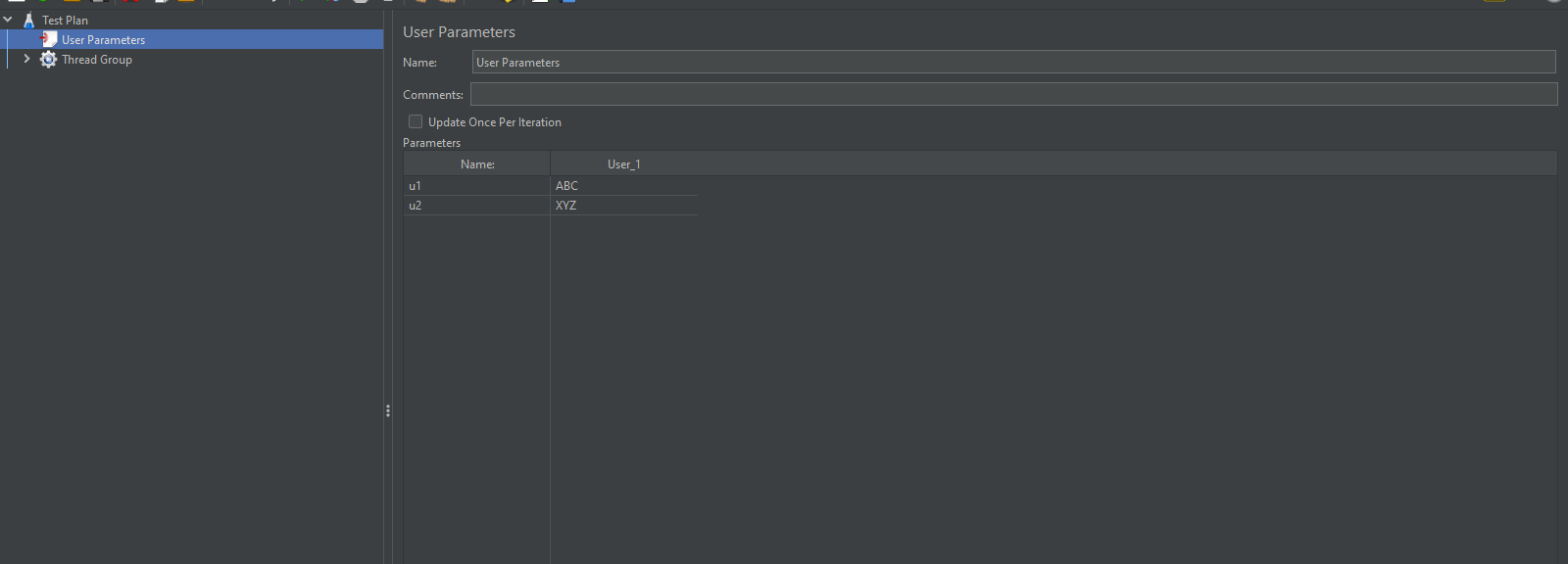
****

**Pre-Processors**

The Pre-Processor Elements Are Used To Modify The Sampler Requests Before Their Processing (Hence The Name Pre-Processor). **How To Add A Pre-Processors In Jmeter-  
Right Click On Either Of Thread Group/Logic Controller -> Hover Over ‘Add’ -> Hover Over ‘Pre-Processors’ -> Click On The Required Pre-Processor Element**

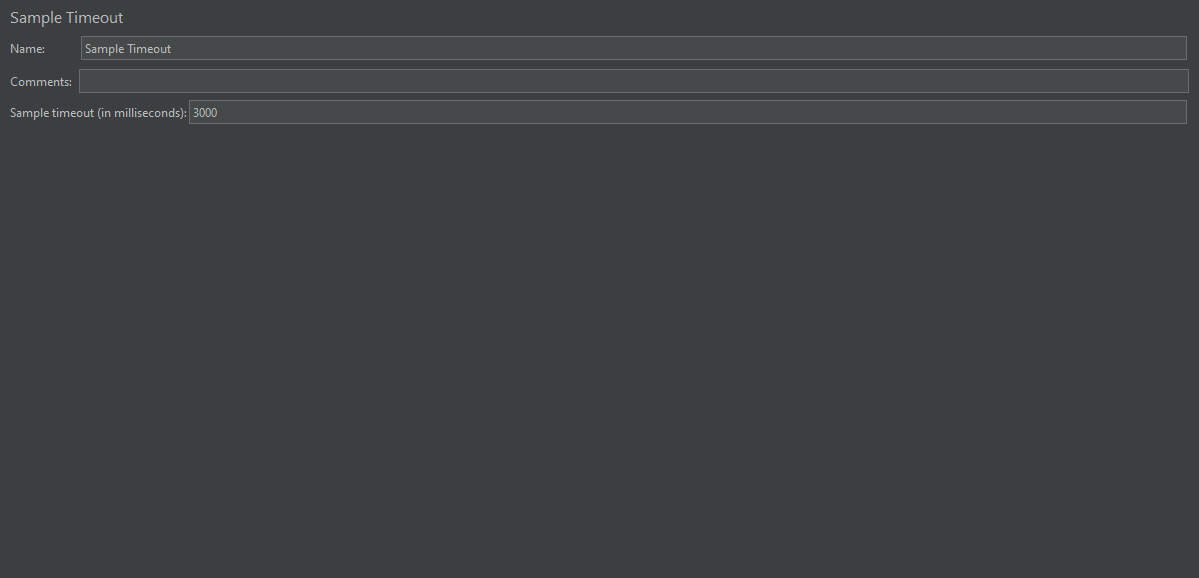
* **User Parameters**

The User Parameters Are Used To Specify Values For User Variables Used Within Thread Groups.



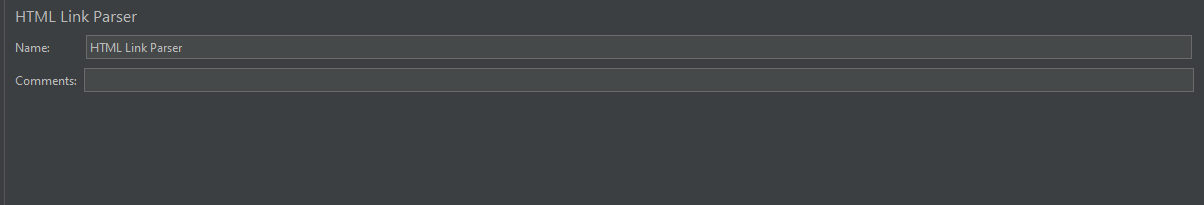
* **Sample Timeout**

Sample Timeout Sets A Maximum Timeout For A Particular Sampler And Executes When The Response Time Exceeds The Given Timeout Value And Instruct Jmeter To Fire The Next Request.

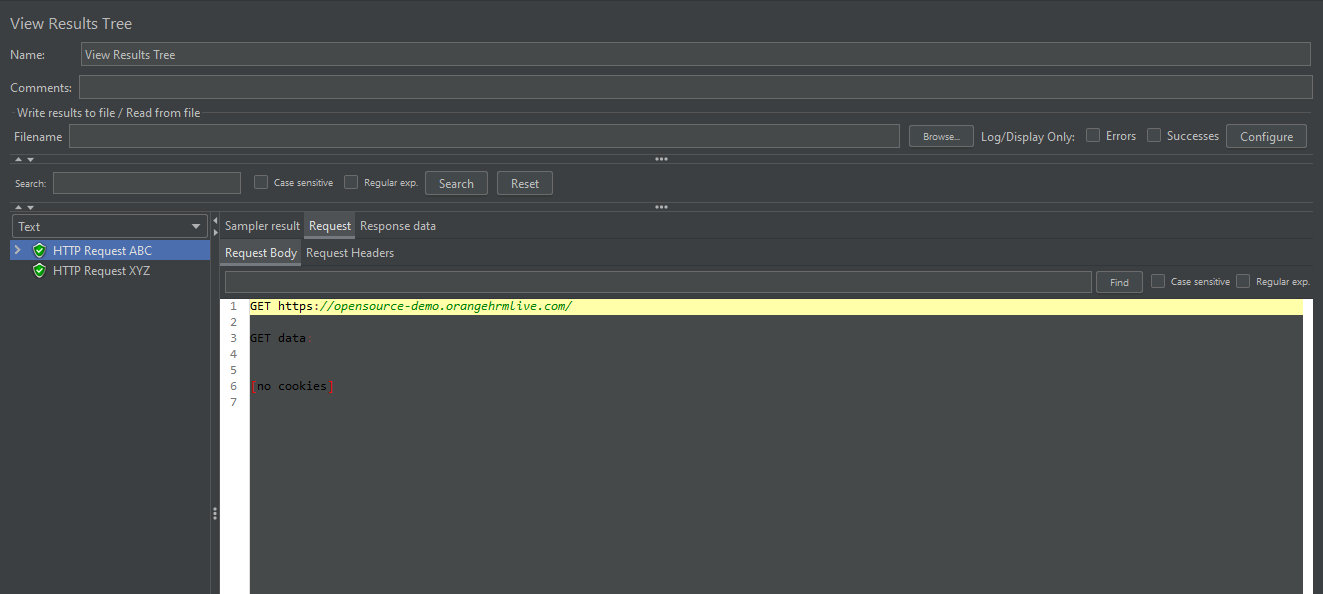


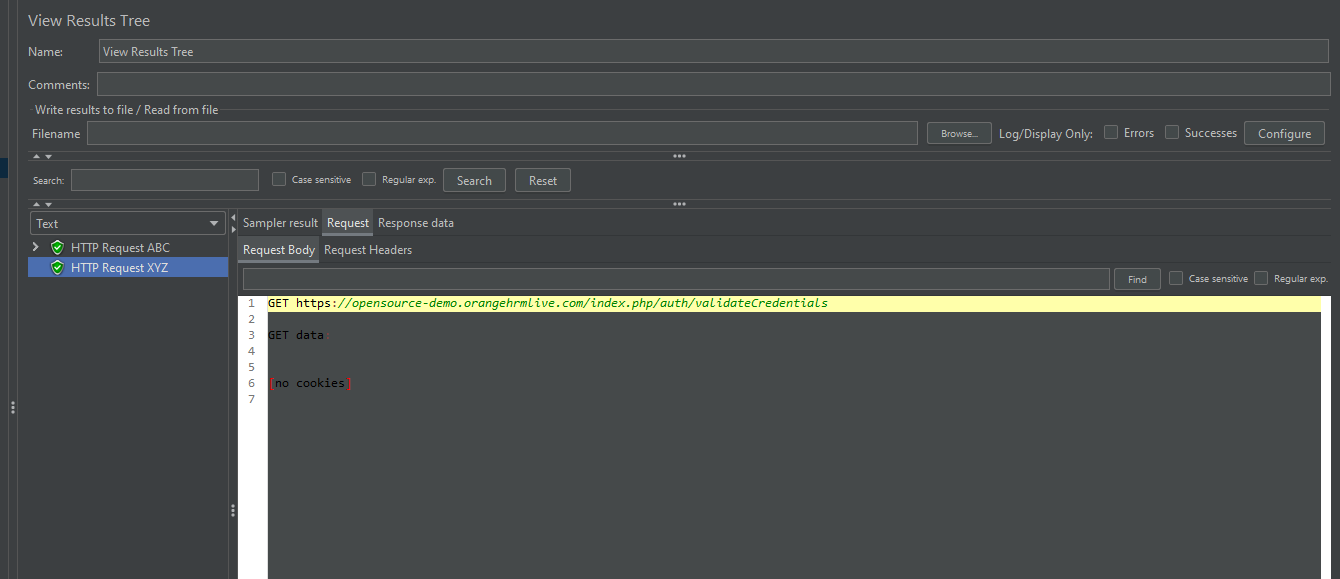
* **Html Link Parser**

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



**Results**

****

****

**Config Elements**

Config Elements In Jmeter Are Used To Configure Or Modify The Sampler Requests Made To The Server. These Elements Are Added At The Same Or Higher Level Of The Samplers That We Want To Configure. **How To Add An Config Elements-  
Right Click On Either Of Test Plan/Thread Group/Logic Controller -> Hover Over Add -> Hover Over Config Element -> Click On The Required Config Element**

* Csv Data Set Config

The Csv Data Set Config Is Used To Read Data From Csv File, Put The Data Into Variable(S) And Then Use The Variable(S) In The Sampler Requests. Http Cache Manager

The Http Cache Manager Is Used In Test Scripts To Add The Caching Functionalities Of Web Applications. This Element Is Just Required To Be Added At The Same Level Or Higher Than The Sampler Request Where Caching Functionality Is Required.

* Http Cookie Manager

The Http Cookie Manager Is Required For Session Handling By Providing The Functionality Of Storing And Sending Of Cookies.

* User Defined Variables

As The Name Suggests, The User Defined Variable Config Element Is Used To Create Variables With A Value (Key-Value Pairs) That Are Used Across The Test Script.

* Http Authorization Manager

The Http Authorization Manager Is Used For Testing Applications Requiring Multiple Logins For Ensuring Authorization.

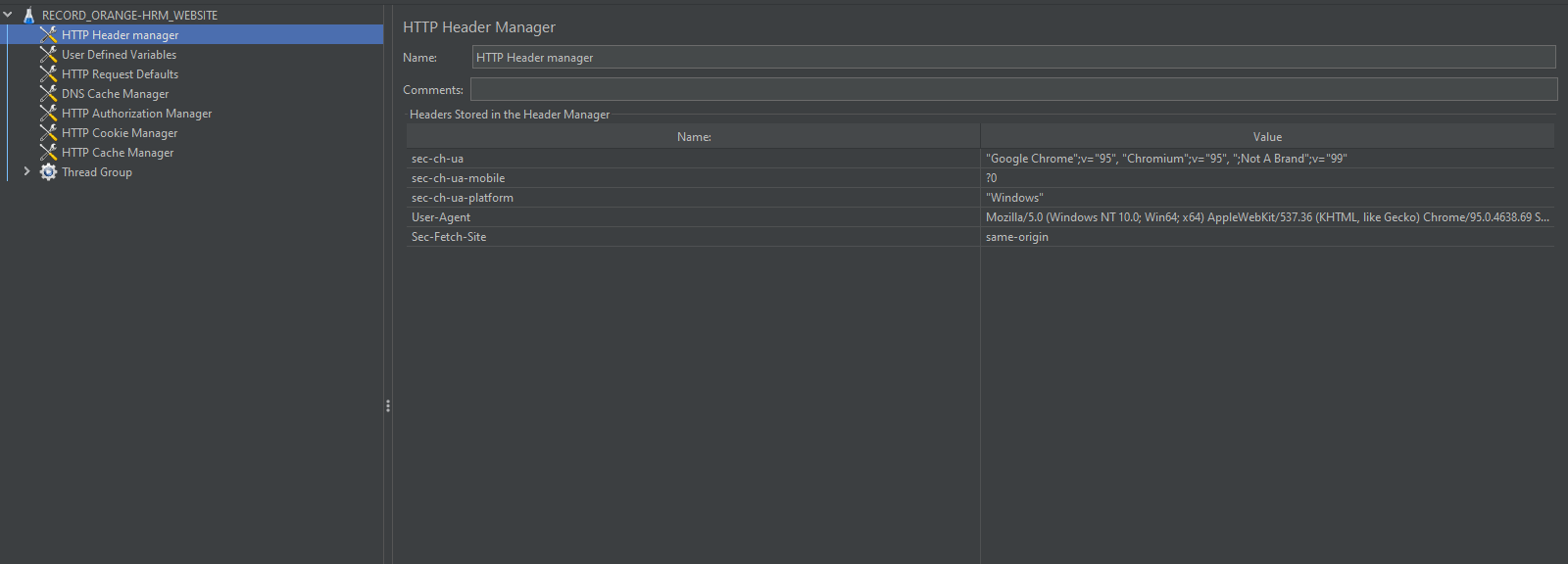
* Http Request Defaults

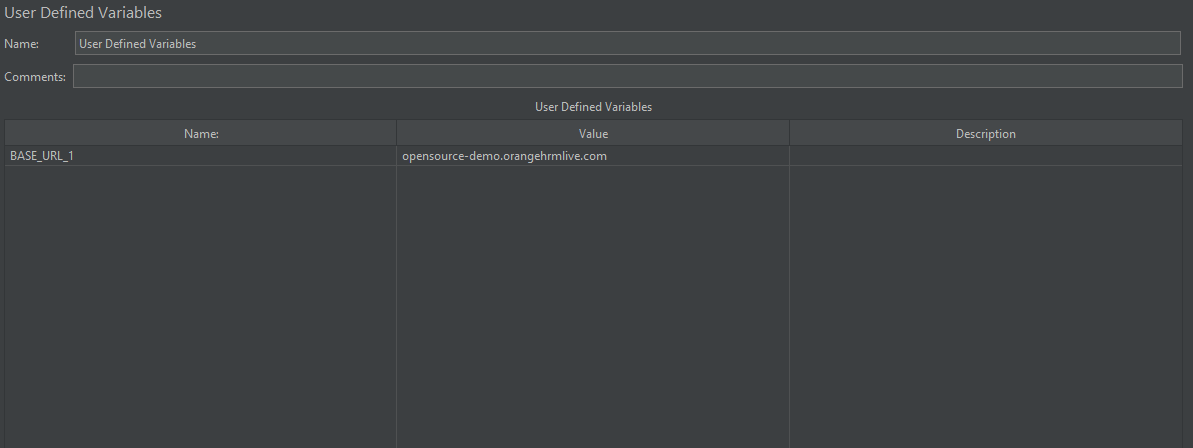
The Http Request Defaults Config Element Is Used For Setting Default Values For Http Requests.

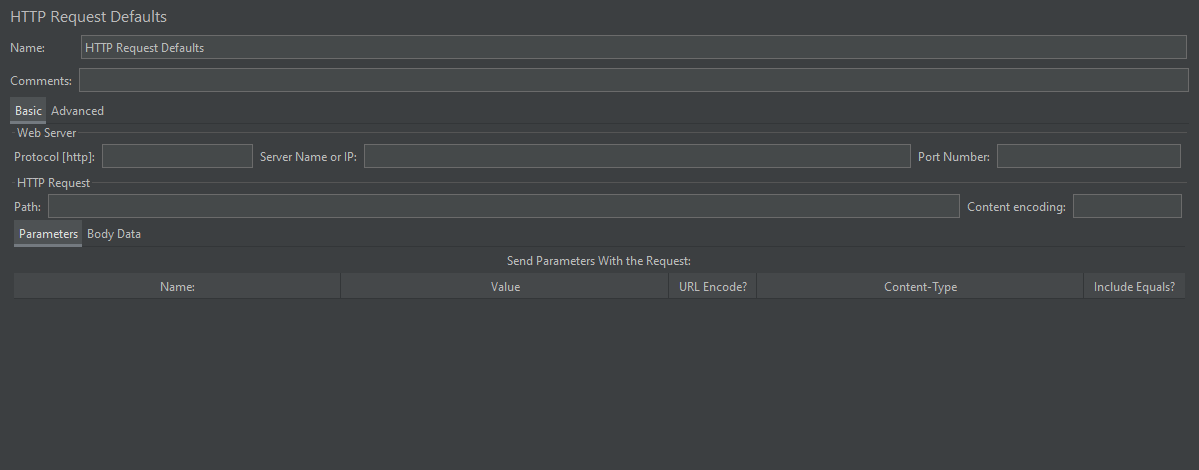
* Http Header Manager

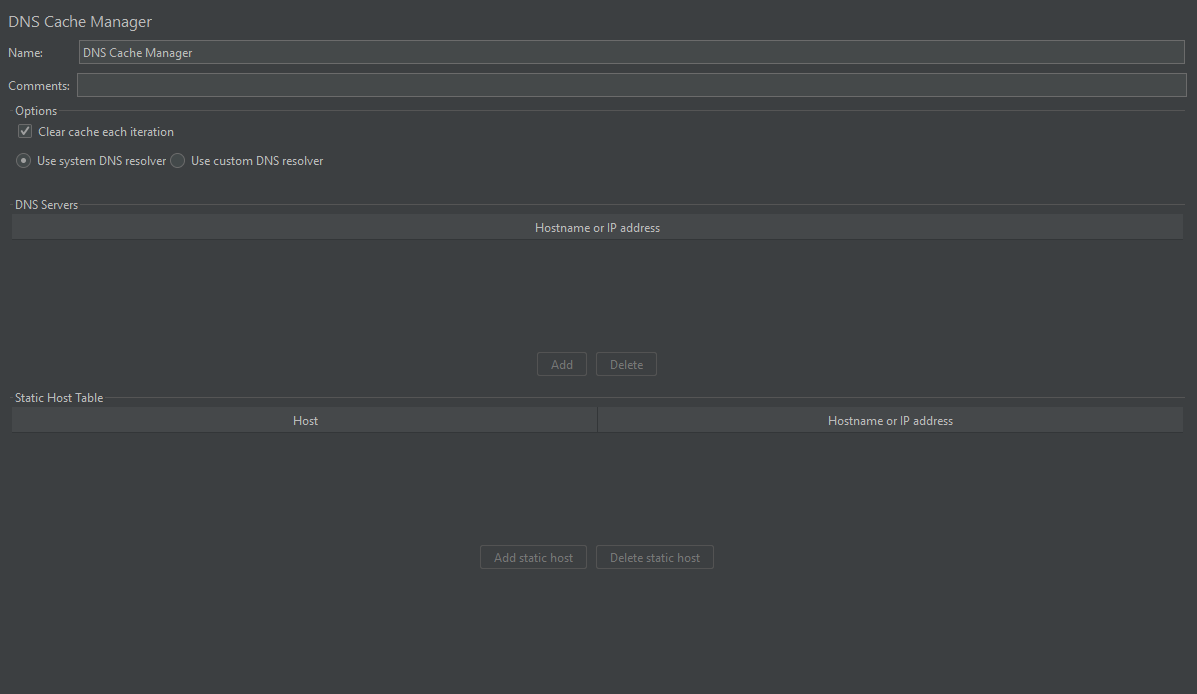
The Http Header Manager Is Used To Override The Http Request Headers.

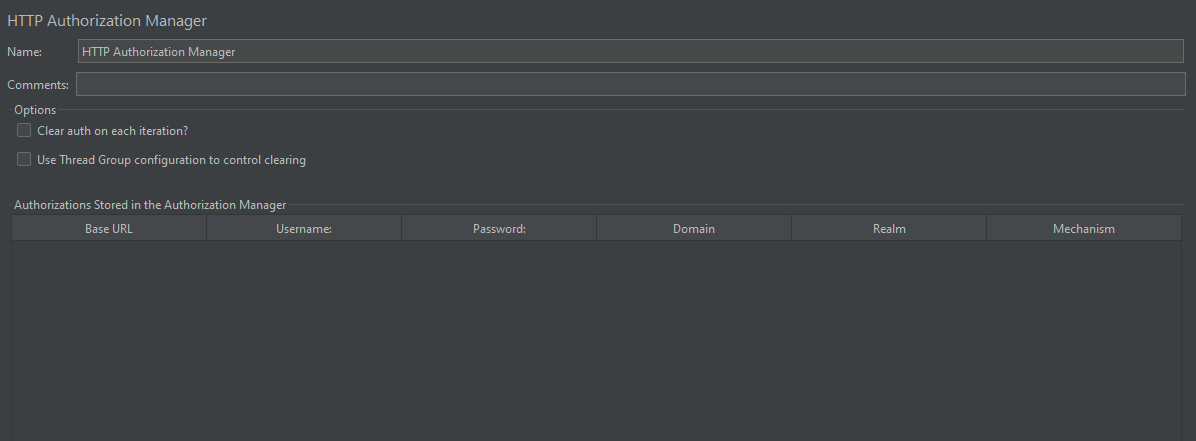
**Screenshots**

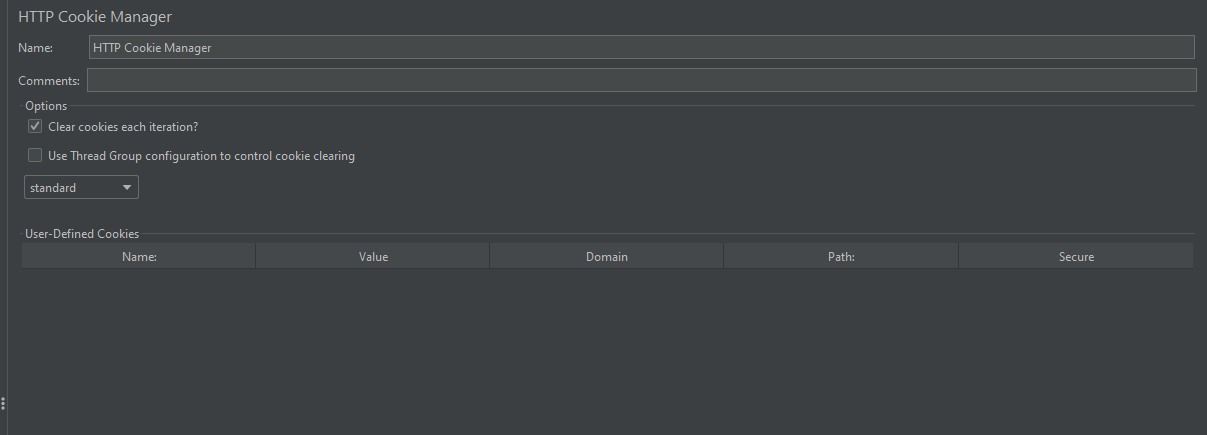
****

****

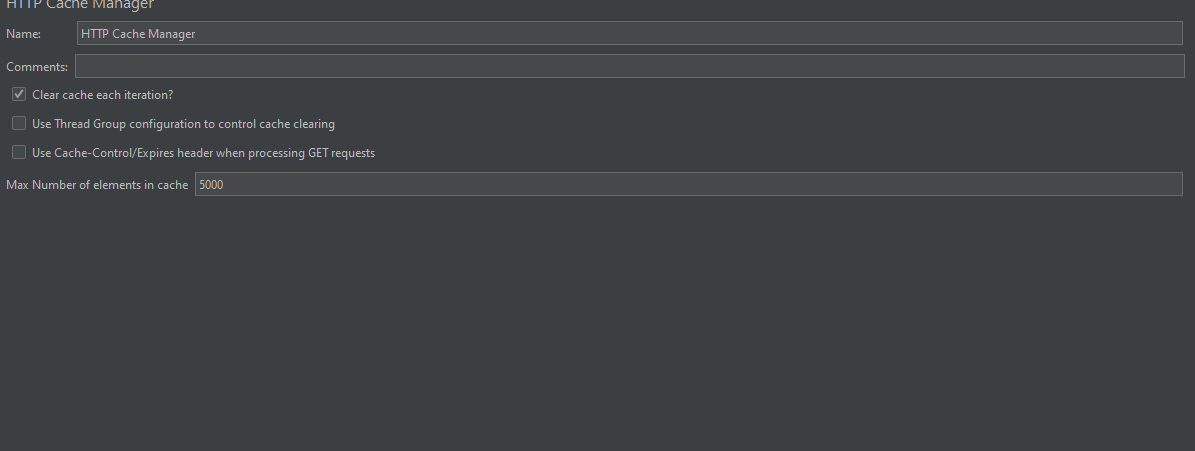
****

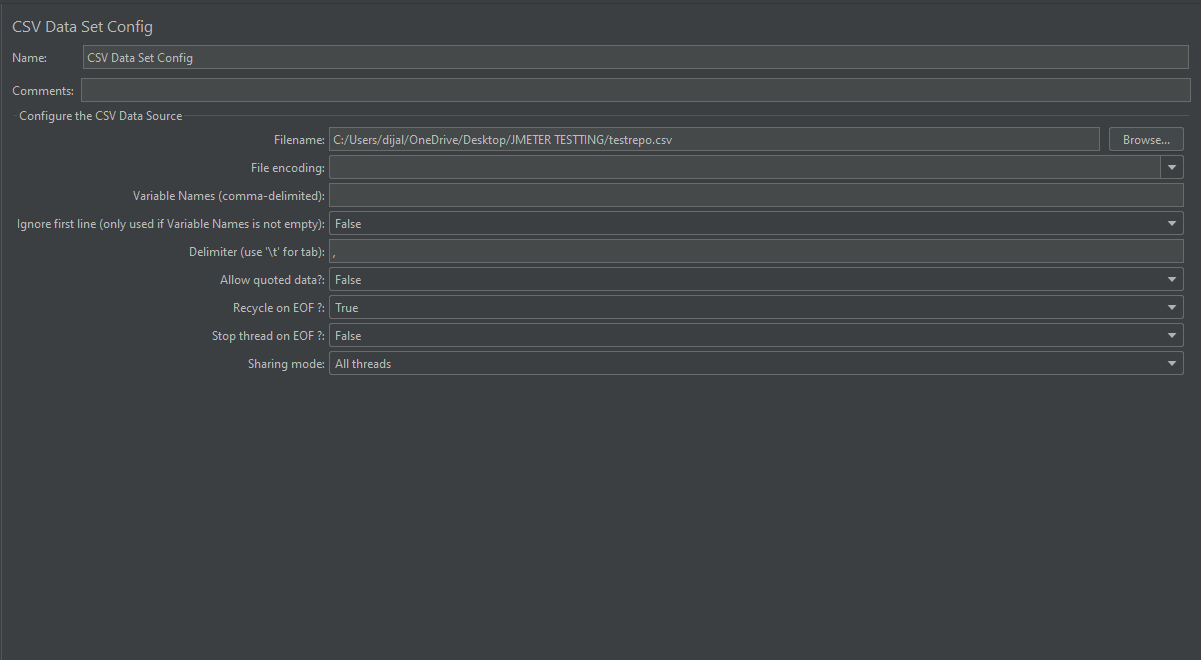
****

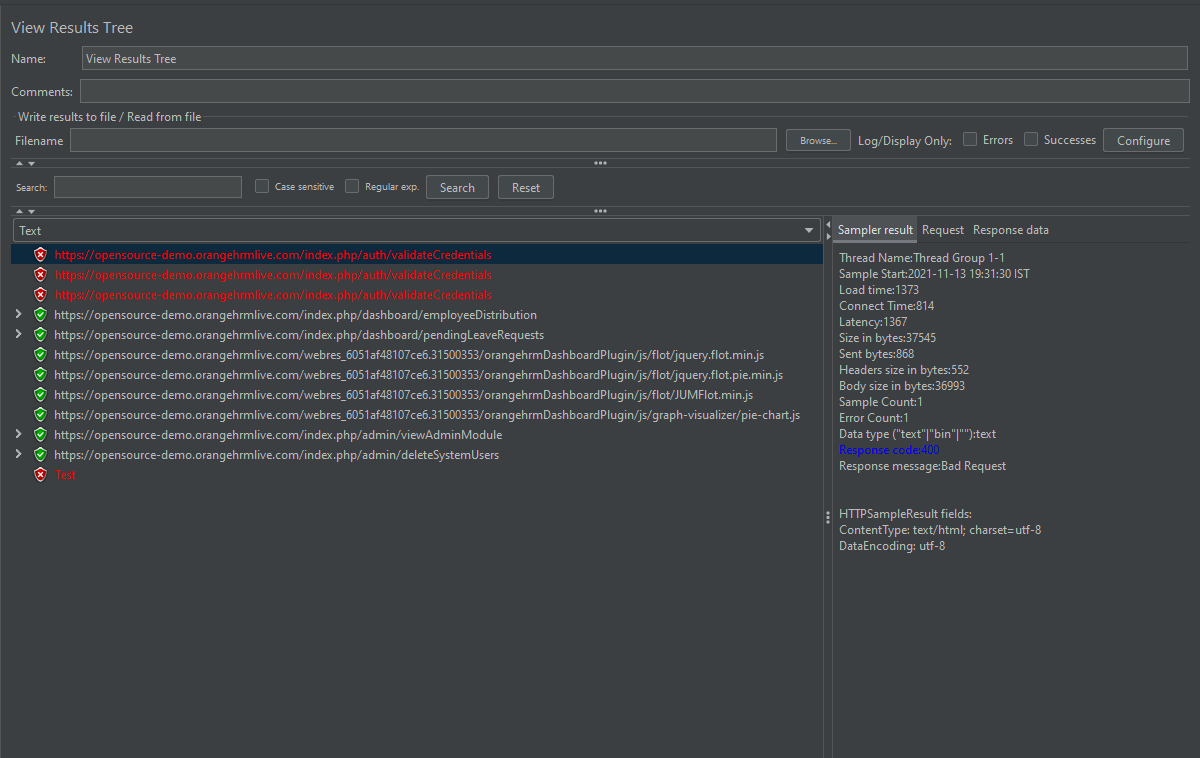
****

****

Http Cache Manager

****





**Timers**

Timers In Jmeter Is The Test Plan Elements Used To Pause The Execution Of Test For A Certain Specified Amount Of Time. This Pause Between Requests Helps In Simulating Real-World Scenarios Like Time Taken By Users To Think, Type Something, See And Process The Information Displayed Etc. **How To Add A Timer-  
Right Click On Thread Group -> Hover Over Add -> Hover Over Timer -> Click On The Required Timer**

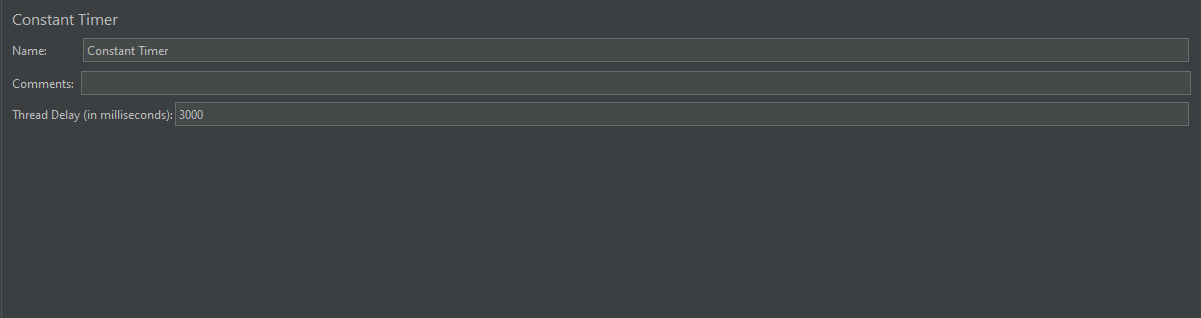
### **Constant Timer**

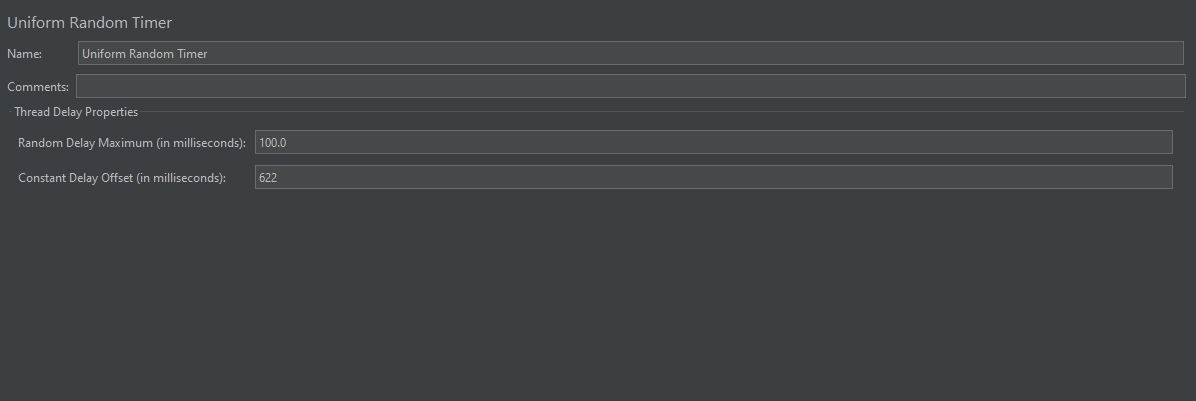
The Constant Timer Is One Of The Most Widely Used Timers In Jmeter. It Pauses The Execution Of Test For A Specified Constant Amount Of Time.

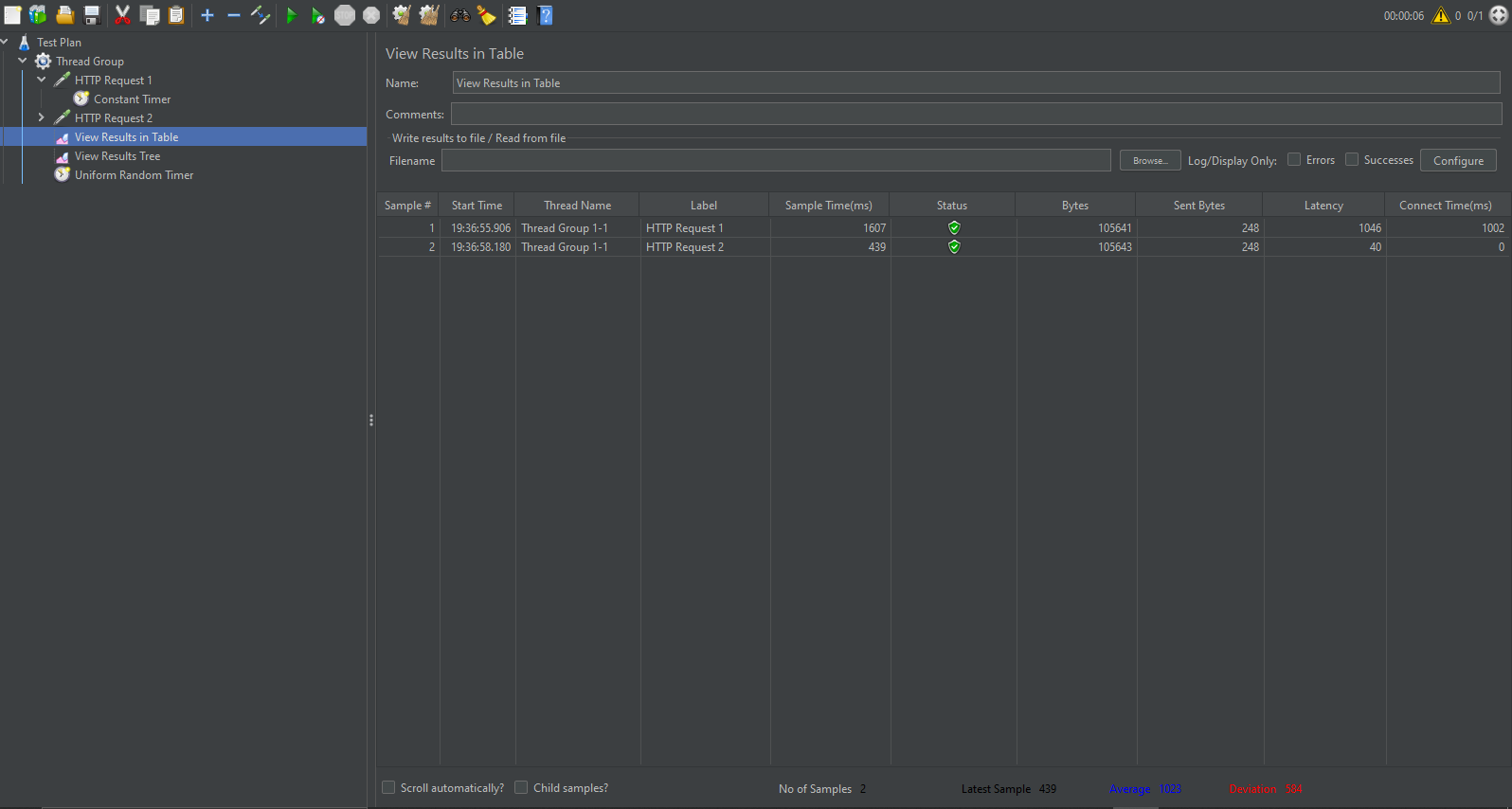
### **Uniform Random Timer**

The Uniform Random Timer Is Used To Pause The Test Execution For A Random Time. The Maximum Value For Random Time Can Be Specified Along With The Additional Constant Time With Each Wait.

**Screenshots**

****

****

****

**Listeners**

Jmeter Listeners Are The Test Plan Elements That Are Used To View And Analyze The Result Of Performance Tests In Tabular Or Graphical Form. They Also Provide The Different Response Time Matrices (Average Time, Minimum Time, Max Time, Etc) Of A Sampler Request. **How To Add A Listener-  
Right Click On Test Plan -> Hover Over Add -> Hover Over Listener -> Click On The Required Listener**

### **Aggregate Graph**

The Aggregate Graph Listener Is Used To Display The Test Results In Both Tabular Form(Reports) And Graphs.

### **Aggregate Report**

The Aggregate Report Listener Is Used To Display And Store Test Results In The Form Of Reports.

### **Assertion Results**

The Assertion Results Listener Is Used To Display The Assertion Result For Each Erroneous Sampler Response. It Is Advised To Not Use This Listener During The Performance Test As It Is Very Resource-Intensive. It Should Be Used While Debugging And Functional Testing Only.

### **Graph Results**

The Graph Results Listener Is Used To Display Each Sampler Request’s Response Time Graph In Terms Of Average, Median, Deviation, And Throughput.

### **Response Time Graph**

The Response Time Graph Is Used To Provide The Graphical Representation Of Response Time With Time Elapsed During The Test Run.

### **Simple Data Writer**

The Simple Data Writer Listener Is Used To Save The Sampler Response To A File After With Different Configurations To Remove Several Unnecessary Overheads.

### **Summary Report**

The Summary Report Is Used To Store And Display The Test Result In Tabular Form Just Like An Aggregate Report Listener But Consumes Less Memory (As Per Apache Jmeter).

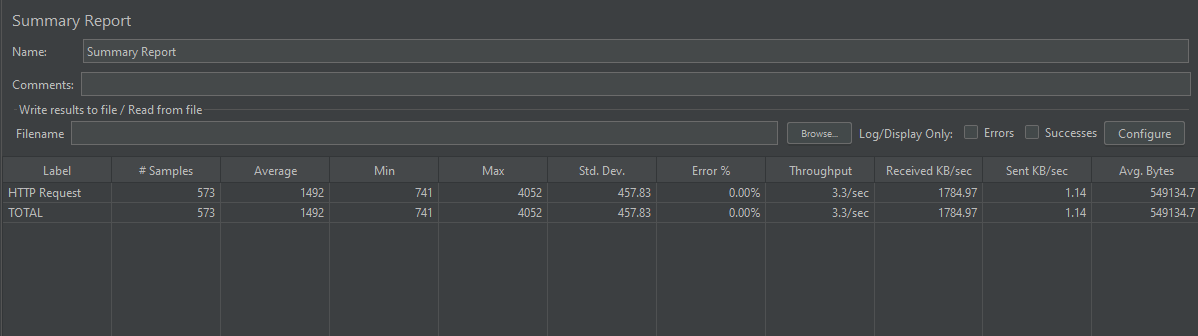
### **View Results Tree**

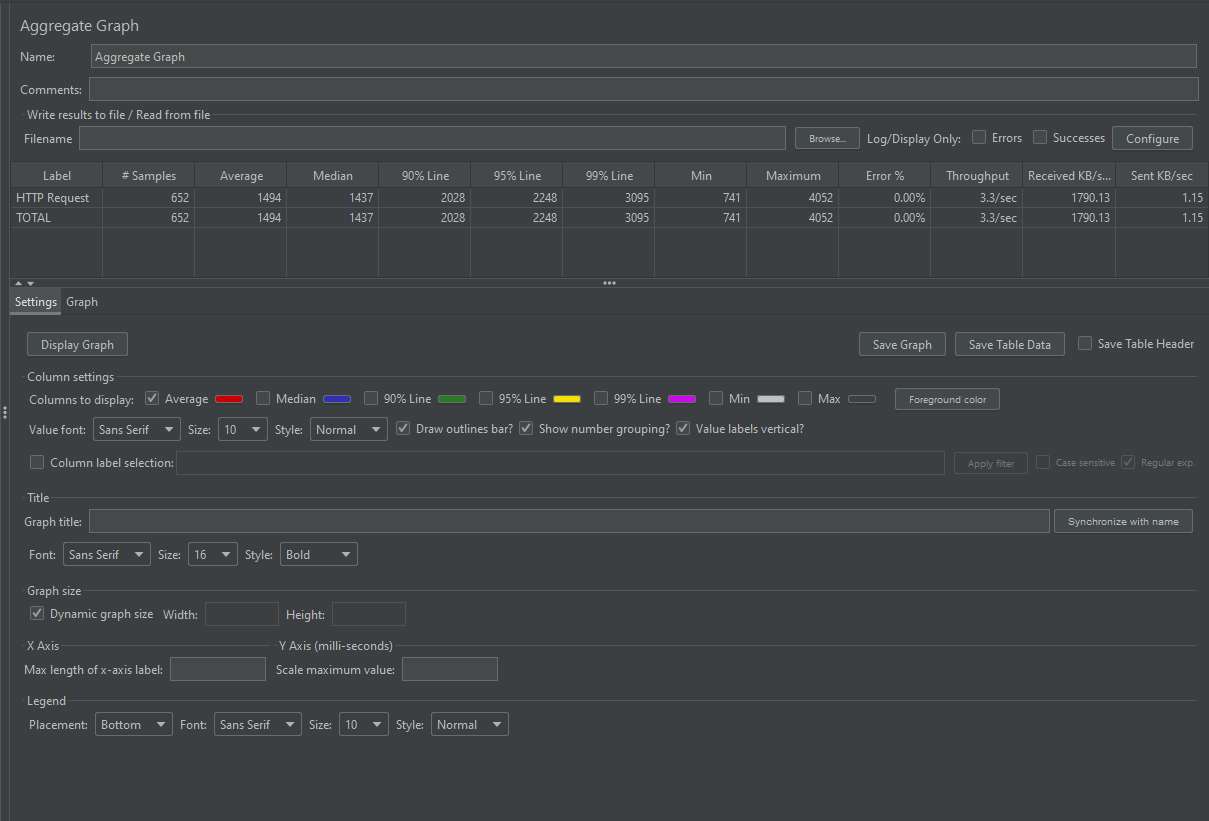
This Listener Is Used To Provide And Store Test Results For Each And Every Individual Sampler.

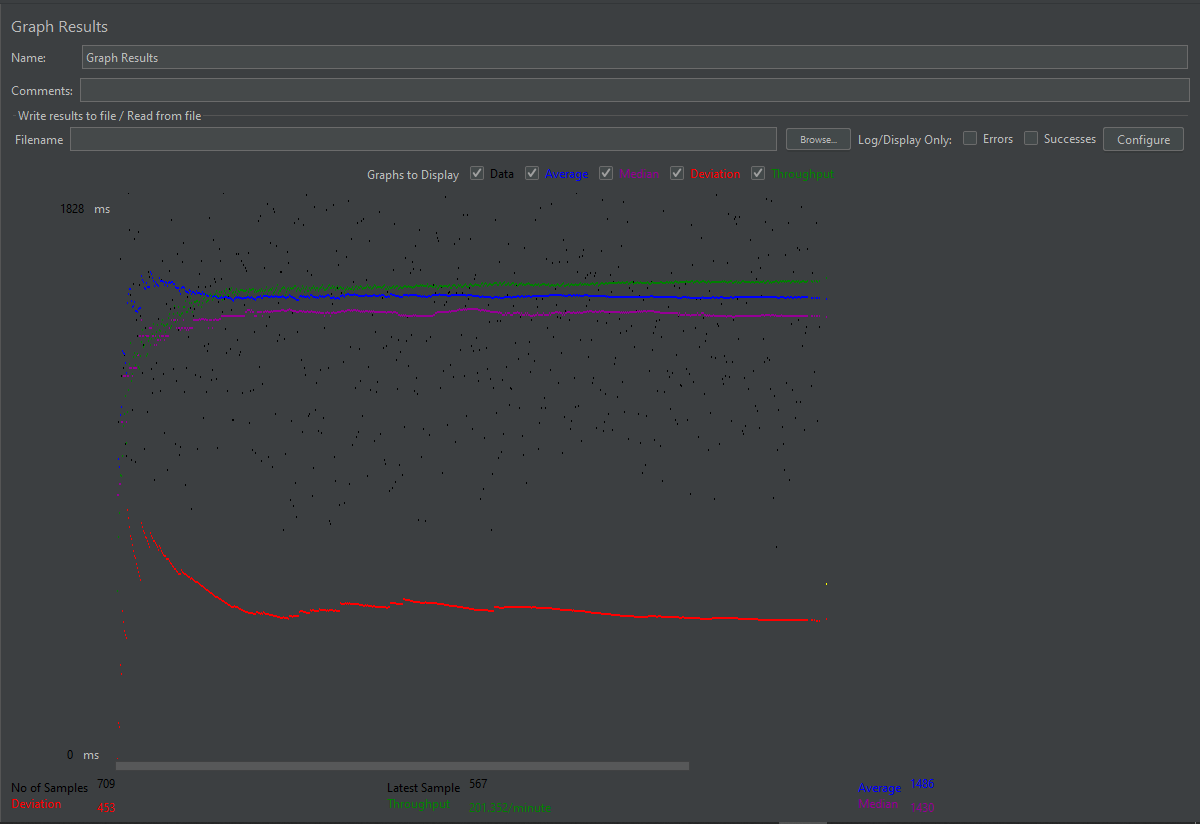
### **View Results In Table**

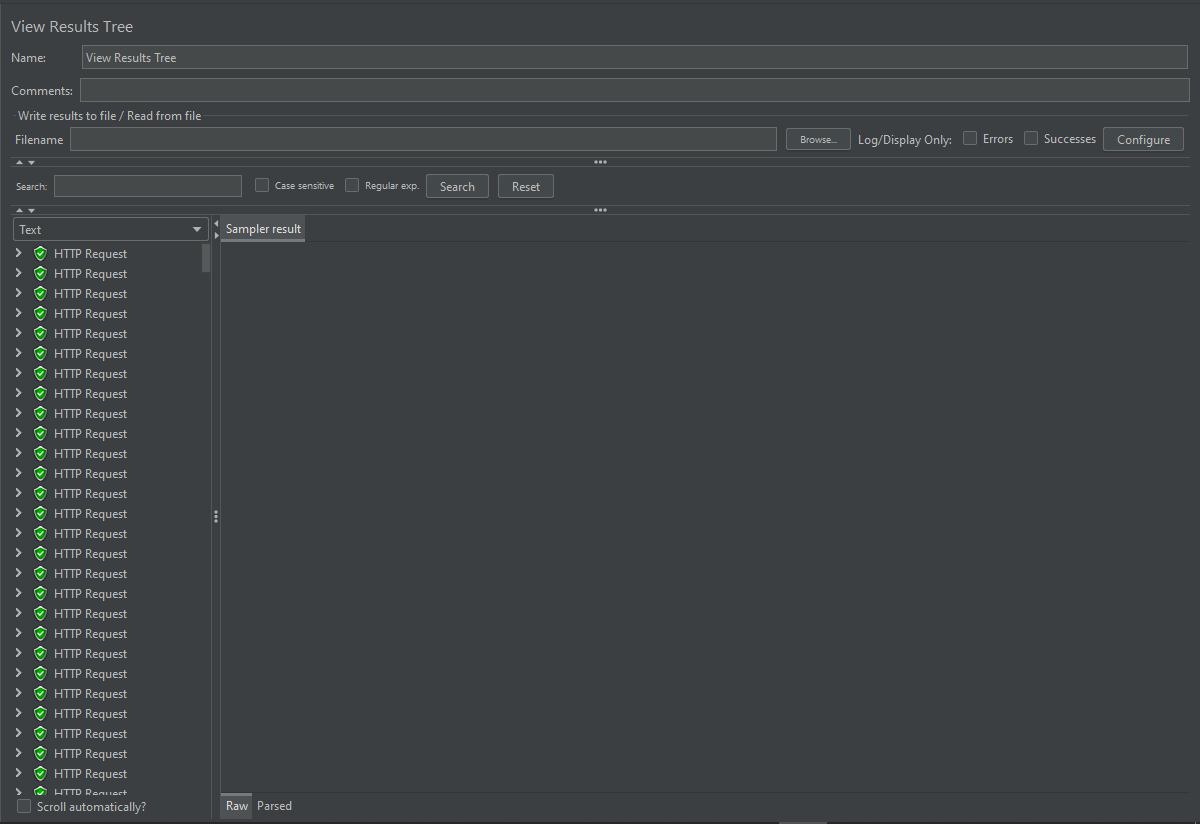
The View Results In A Table Listener Are Used To Display The Sampler Response Header And Response Body.

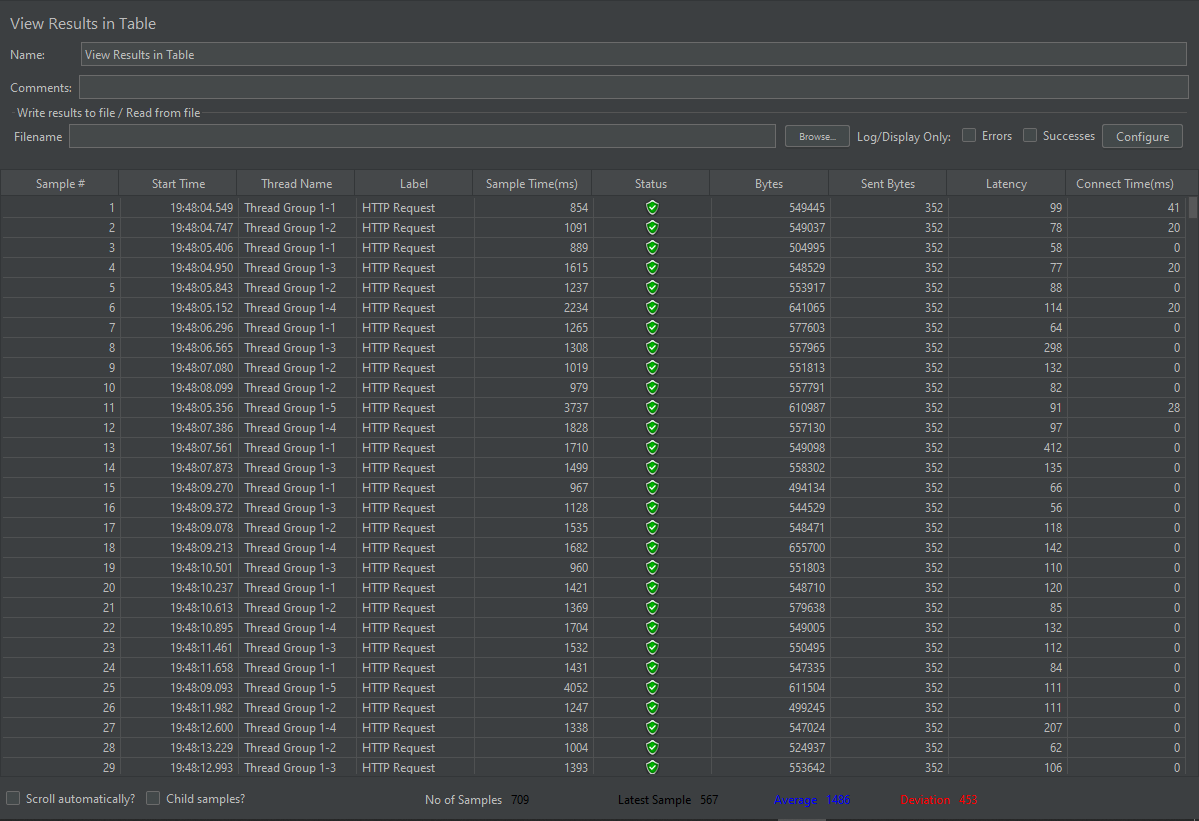
**Screenshots**

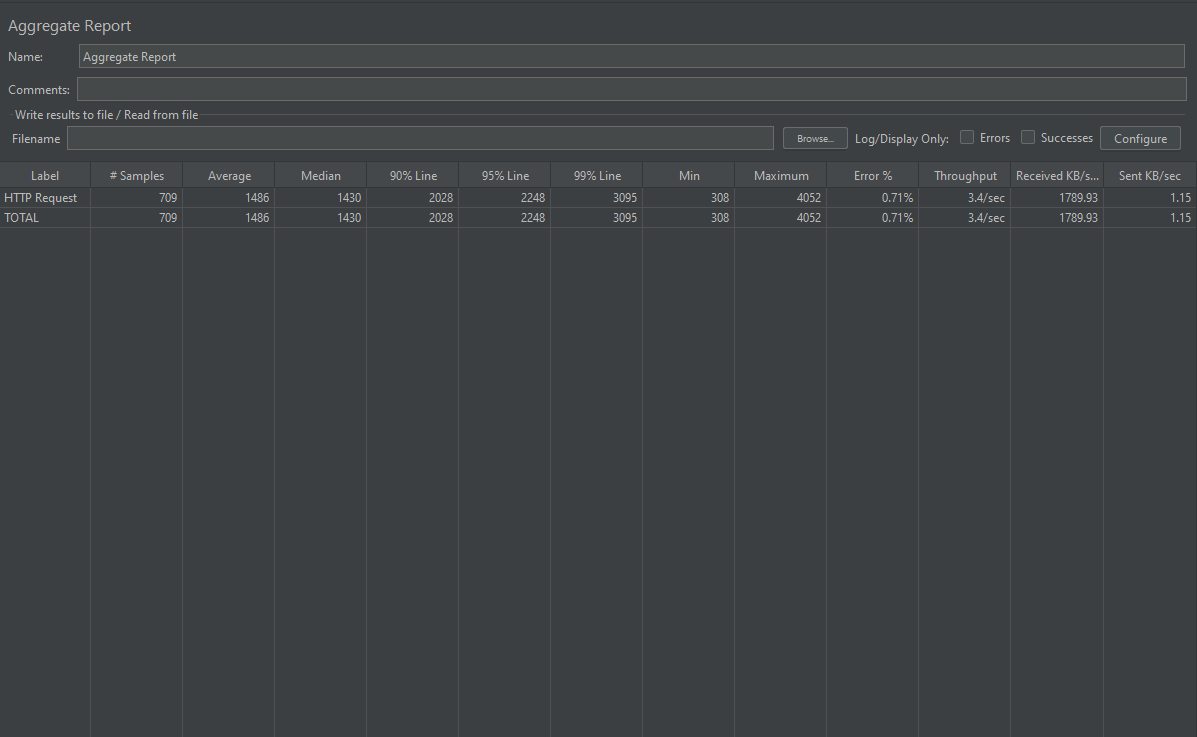
****

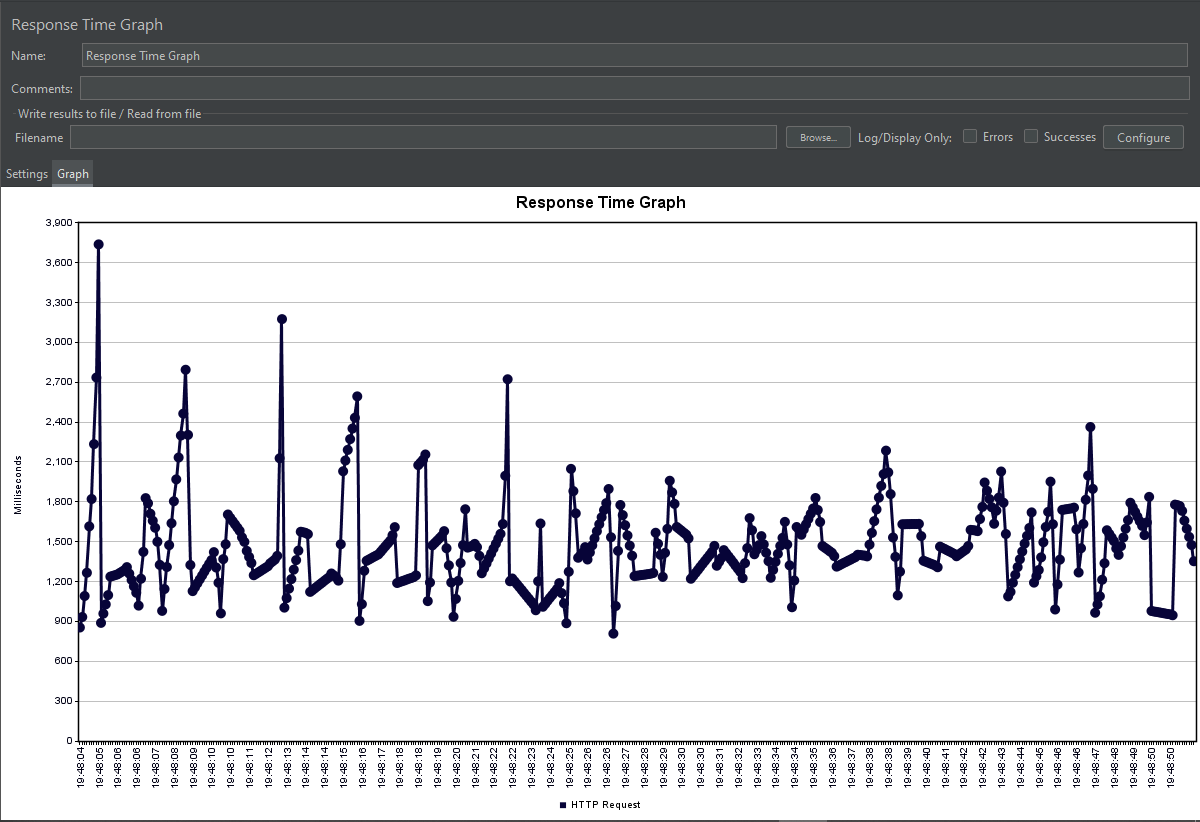
****

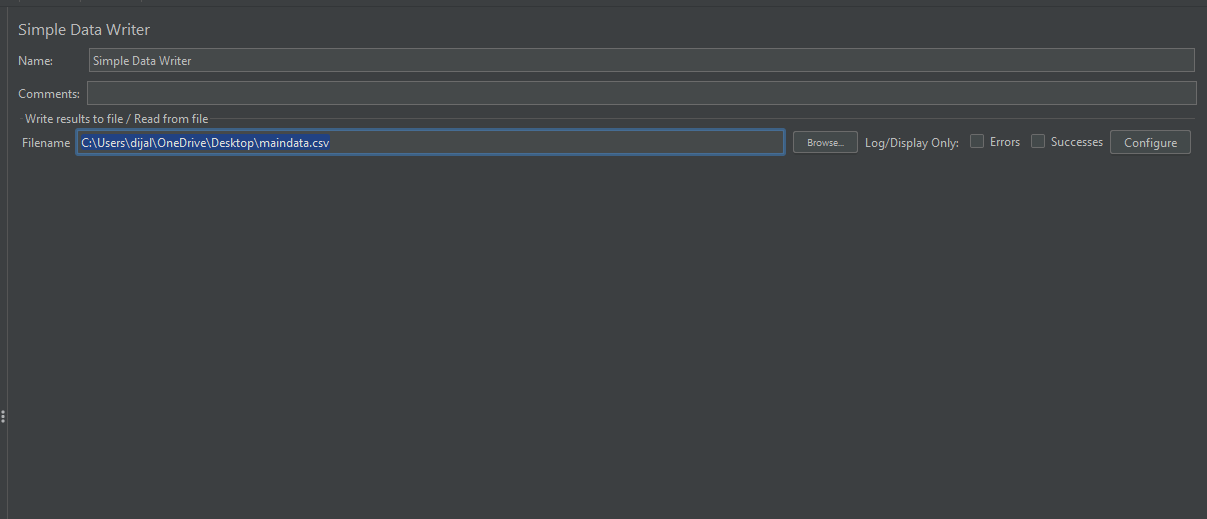
****

****

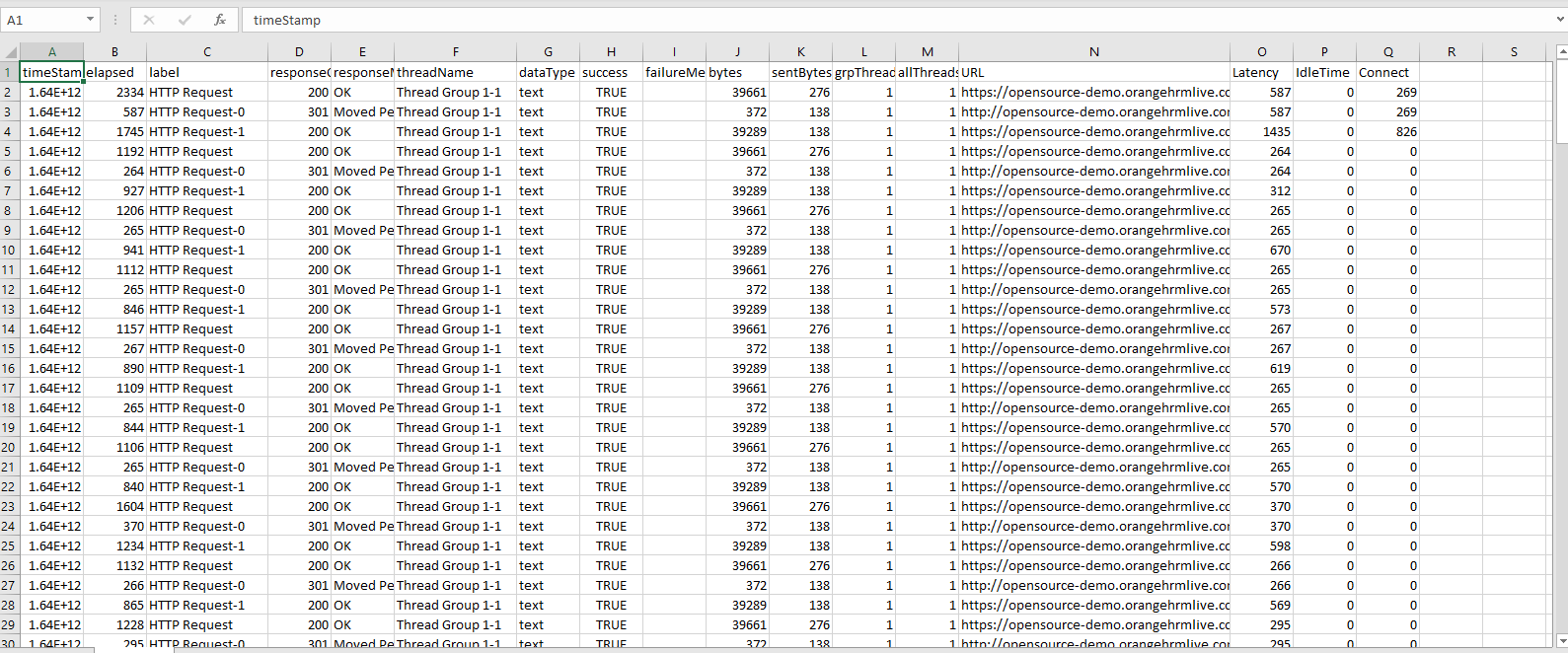
****

****





**Csv File Create Via Simple Data Writer**

****

**Creating Html Reports**

