# API performance with Jmeter

**[ Test Plan ]**

Contents

[API performance with Jmeter [ Test Plan ] 1](#_Toc494309356)

[1. About Jmeter 2](#_Toc494309357)

[2. Test plan 2](#_Toc494309358)

[a. Implementing the REST APIs 2](#_Toc494309359)

[b. Using the transaction controller 2](#_Toc494309360)

[c. Creating a REST Call using the http sampler. 3](#_Toc494309361)

[d. Handling the headers 5](#_Toc494309362)

[e. Validating the response 5](#_Toc494309363)

[f. Parameterizing the request data/payload 5](#_Toc494309364)

[g. Think time 6](#_Toc494309365)

[h. Using the Listeners – View Result Tree 7](#_Toc494309366)

[i. Using the Listeners – Aggregate Report 7](#_Toc494309367)

[3. Running the test and Load Test Execution. 8](#_Toc494309368)

# About Jmeter

Jmeter is an open source performance testing tool from Apache foundation, which supports Web, Web Services, JDBC, Unit testing etc. It is handier and flexible tool compare to other commercial tools such as Load Runner, Neo Load etc.

It provides rich scenario scheduling and reporting mechanism using external plugins which makes it as an extensible tool. One can write own plugin for the jmeter (Core Java knowledge is required).

Jmeter run on Java platform, to run it we need jre/jdk setup in our system. We can download jmeter from here: http://jmeter.apache.org/download\_jmeter.cgi

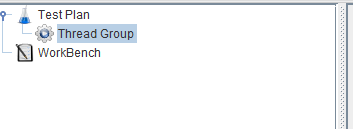
# Building the test plan

To implement any test plan, we need to launch Jmeter using Jmeter.bat under bin folder. Jmeter comes with default container called as Test Plan & Work bench. We can add one or more test scenarios to Test plan as below.

In Jmeter, we refer each virtual User as Thread, group of Virtual Users as a Thread Group.

Now on, we can call each Scenario as a Thread Group.

To add a Thread Group: Right Click on Test Plan ->Add->Threads (User) ->Thread Group



### Implementing the REST APIs

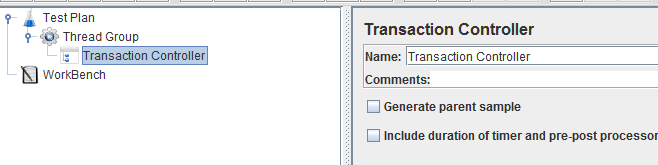
To create test plan for REST APIs using jmeter, we must have at least one Thread group added to the test plan as above snippet.

### Using the transaction controller

Transaction Controller is a component which can be used to measure the response times for any http request. This can be added to the Thread Group as below.

A transactional Controller is a high level component where we can one or more requests to it.

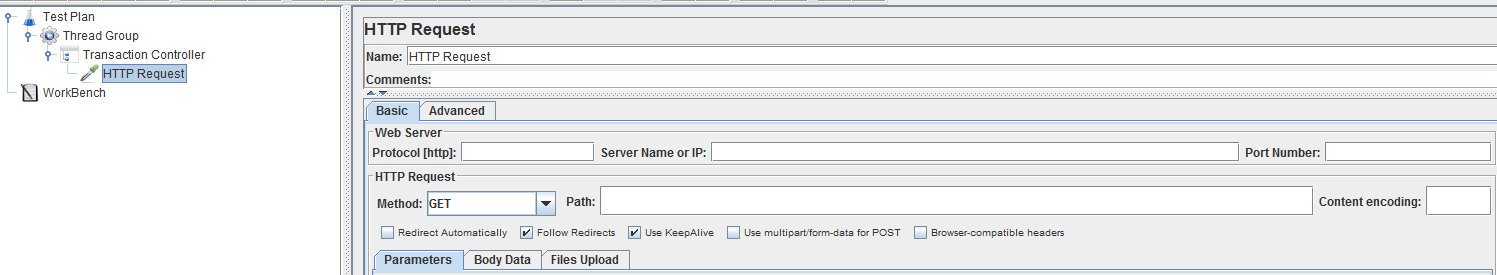
Right Click on Thread Group ->Add -> Logic Controller ->Transaction Controller



### Creating a REST Call using the http sampler.

To implement a REST call we use “Http Request” component to “Transaction Controller” as below.

Right Click on “Transaction controller->Add->Sampler->Http Request”



We can furnish all the necessary details in http request in order to build REST Service call as below.

**To Build A GET API Call:**

E.g.,: <https://demoserver:8080/getUsername>

Type: **GET**

Header: Content-Type : application/json

Below list of components can be used on Http Request

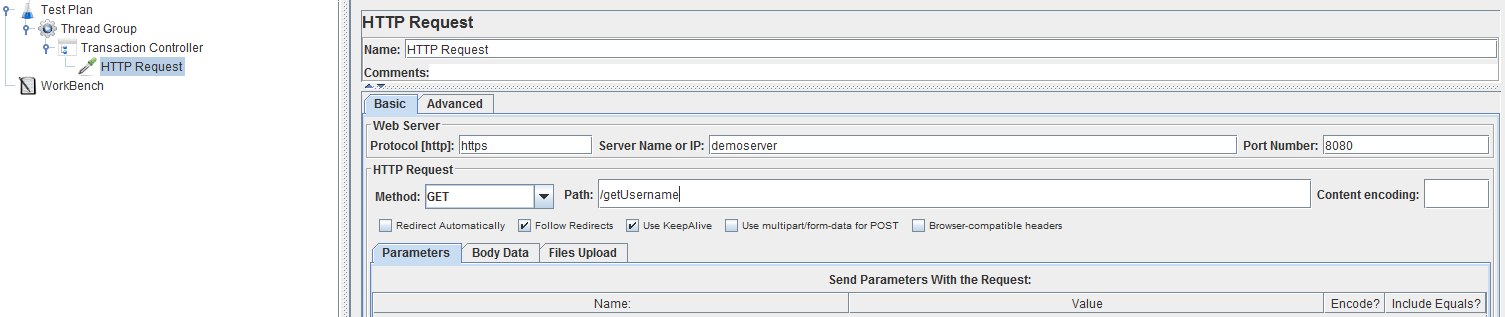
**Protocol**: https

**Server name or IP :** demoserver

**Port Number:** 8080

**Method:** GET

**Path :** /getUsername



**To Build A POST API Call:**

E.g.,: <https://demoserver:8080/createUser>

**Type**: POST

**Payload :** { “Username” : ” testuser”, ”Password” : ”DemoPassword” }

**Header**: Content-Type: application/json

**Protocol**: https

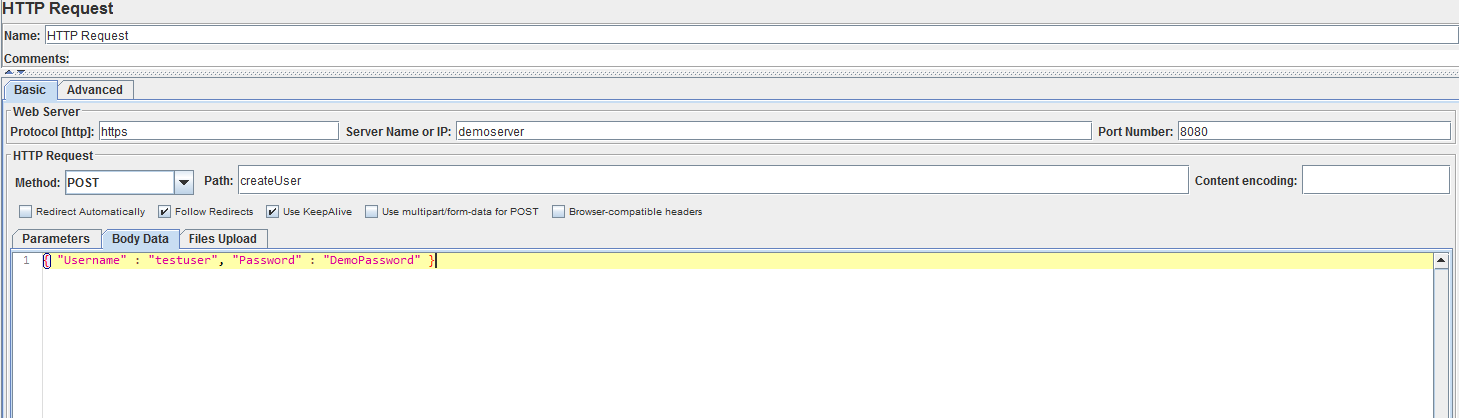
**Server name or IP :** demoserver

**Port Number:** 8080

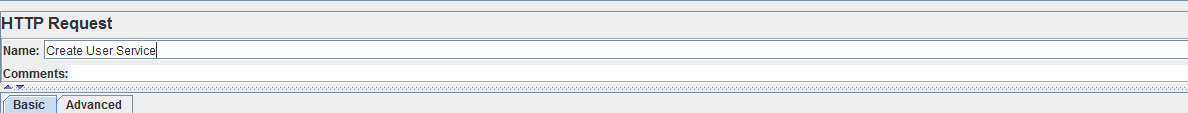
**Method:** POST

**Path :** /createUser

**Body Data Tab:** { “Username” : ” testuser”, ”Password” : ”DemoPassword” }



You can name the service as you need by typing in “Name” text box of request.

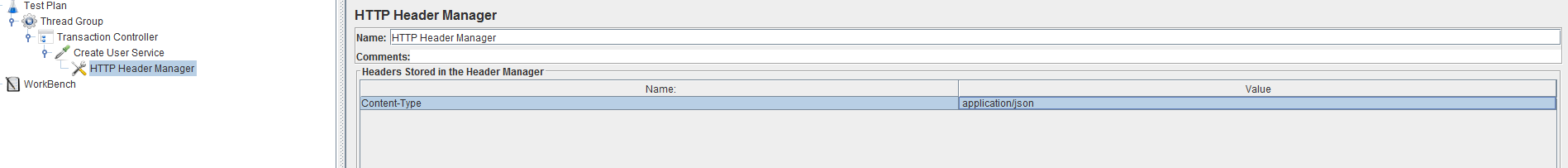


### Handling the headers

We can user specific header by adding “Http Header Manager” component as below

Right Click on Http Request -> Add ->Config Element ->Http Header Manager”

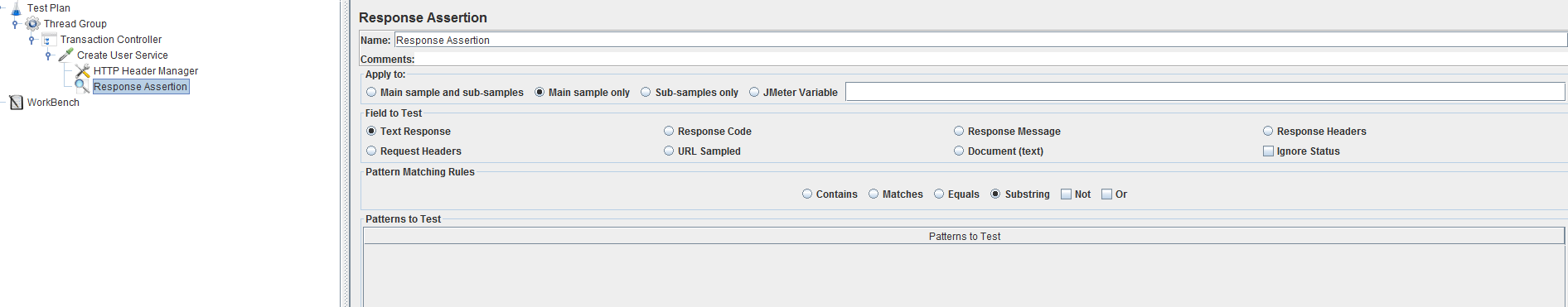
To add user specific header Click on Add button and fill in name value pairs boxes. Likewise we can add as many headers as we need.



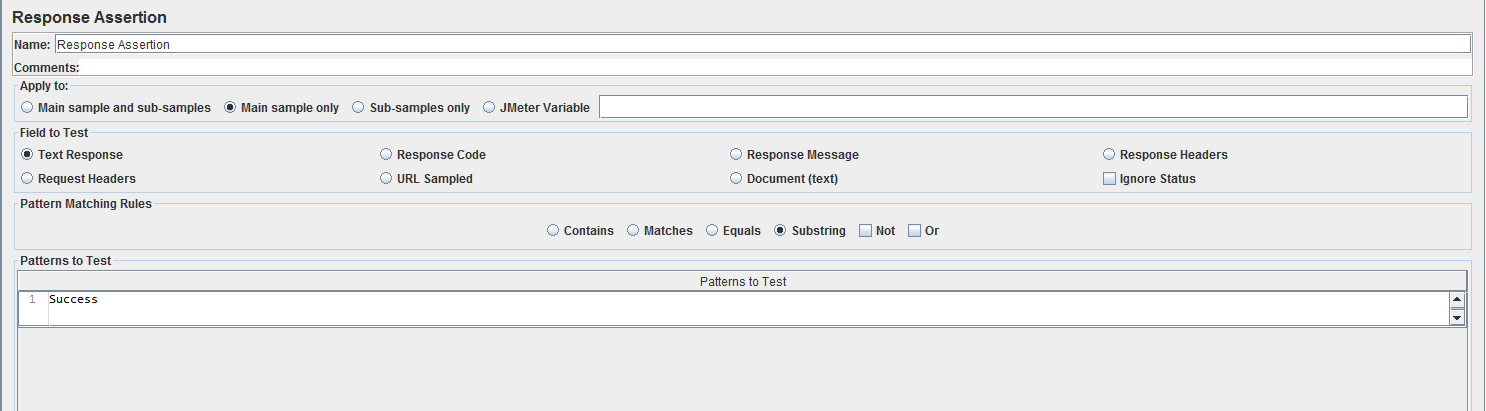
### Validating the response

Validating the response can be done by adding Response Assertion component.

Right Click on Http Request -> Add -> Assertions ->Response Assertion



By clicking on Add button of Assertion you can add whichever the expected text / response code from response to PASS/FAIL the request.

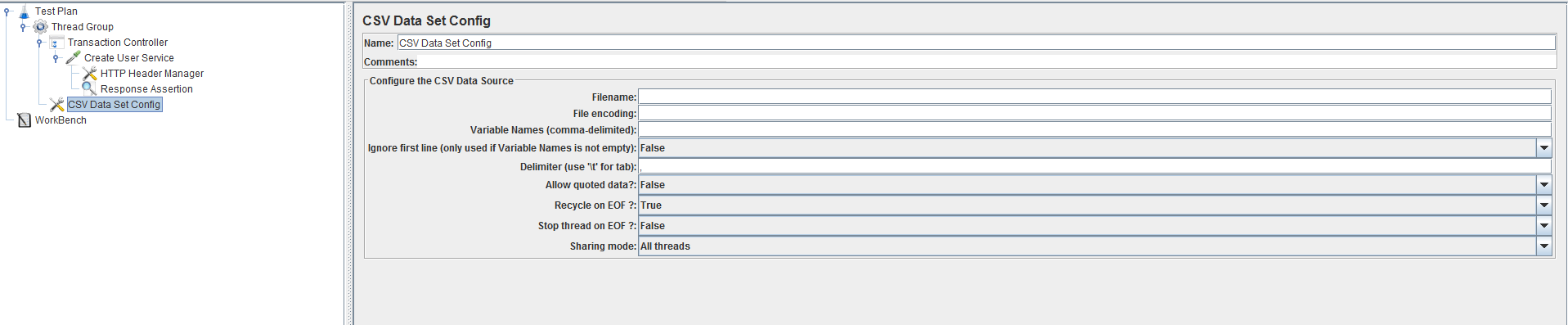


In the above snipped we are using success as text to PASS/FAIL the request

### Parameterizing the request data/payload

To pass the user defined the values to payload/end point we can always use CSV files to supply values from an external data source.

Right Click Thread Group -> Add -> Config Element -> CSV Data Set Config.



We can pass a reference to CSV file as below by filling the necessary details.

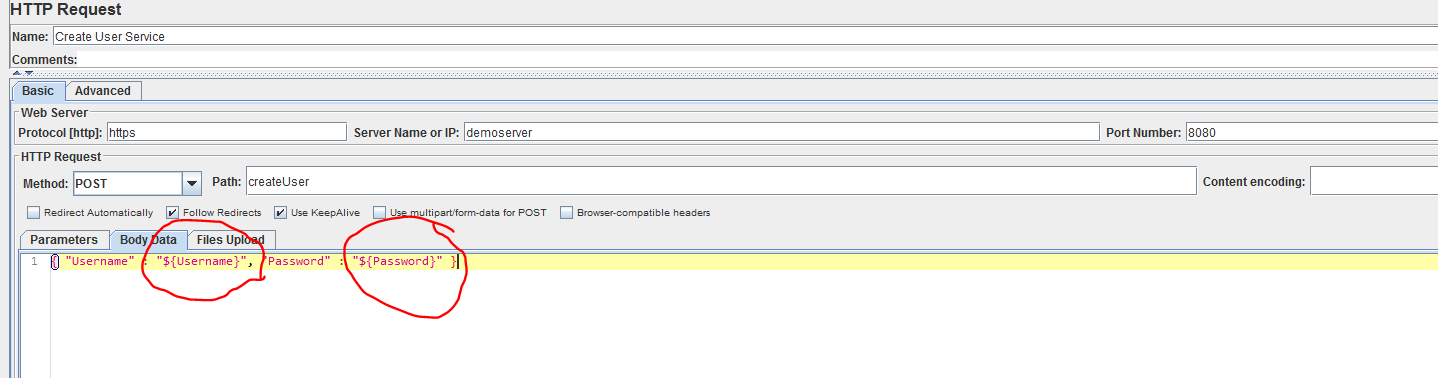
Specify the required fields CSV file path, CSV header names as below.

E.g. Below example uses a csv file contains Username and Password separated by comma.



To substitute CSV variables in script we use below place holders.

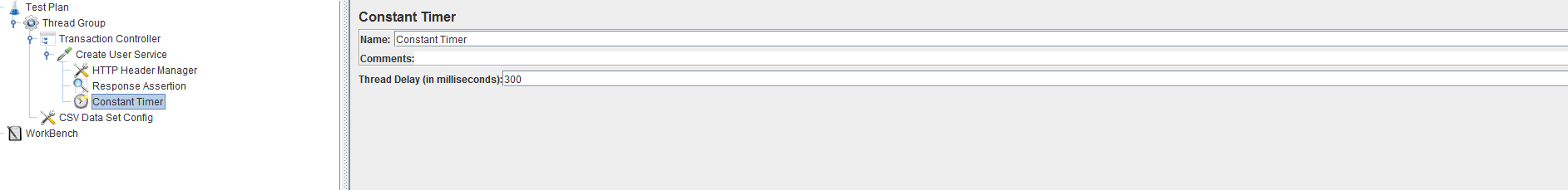
E.g. For Username variable we substitute as ${Username} and for Password variable as ${Password}



### Think time

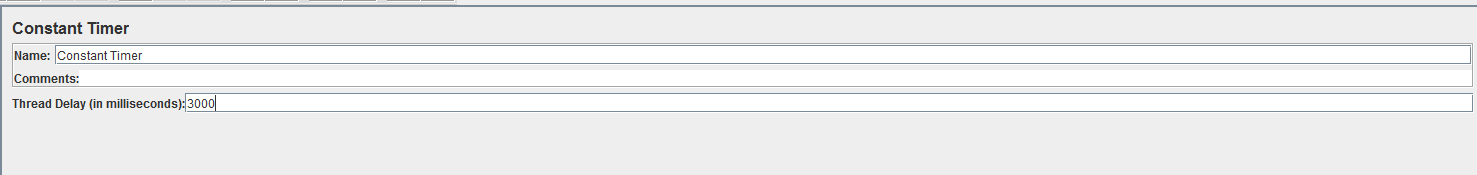
Think time is the one which will add delay between request firing. We can add this as below

Right click “http request ->Add -> Timer->Constant Timer



We need to mentions amount of think time in Mille Seconds under Thread Delay text box.

E.g. I want to add 3 seconds delay before firing the request Create User Service. It can be implemented as below.

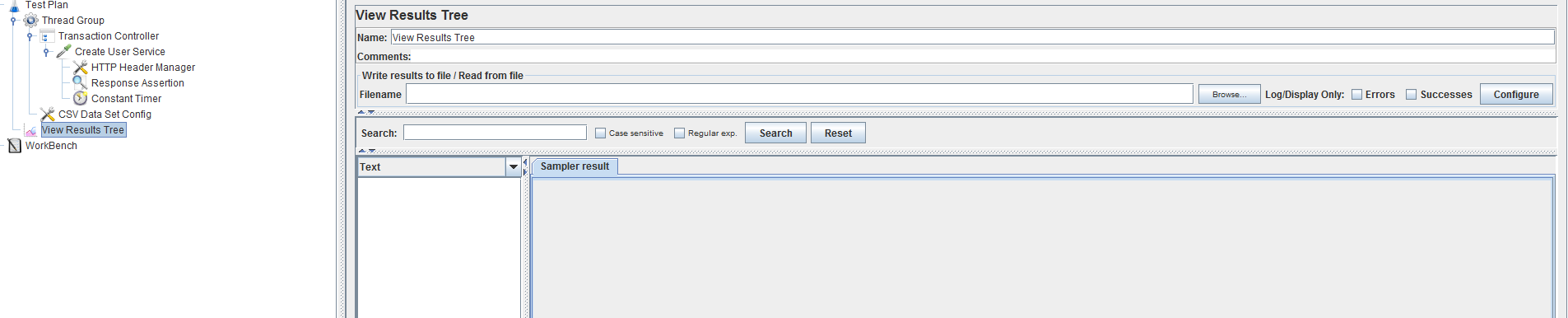


### Using the Listeners – View Result Tree

Listeners are Jmeter components to visualize the output while validating or running the test.

View Result tree component is used to show very low level request information such as Request Information, Headers and Response, Assertion status etc.

To add View Results Tree Right Click on “Test Plan ->Add -> Listener -> View Results Tree

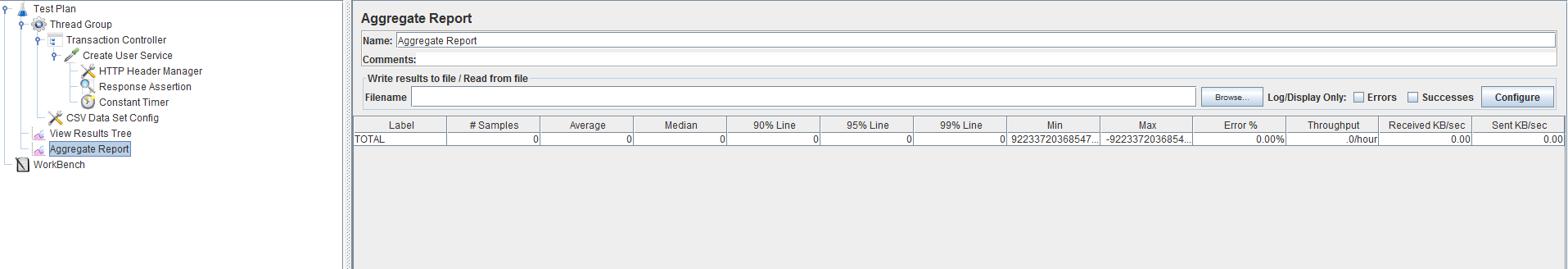


This listener should be used only for debugging the script. We should not use this while running the load test. Since, this will add more load on UI and can cause the Crashing the Jmeter during the load test.

### Using the Listeners – Aggregate Report

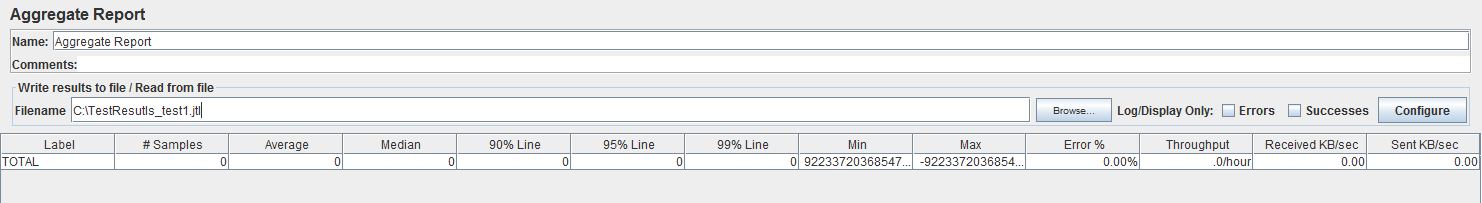
This listener is used to Save the test results and show the response times while running the test. We can add this as below.

Right Click on Test Plan -> Add -> Listeners -> Aggregate Report



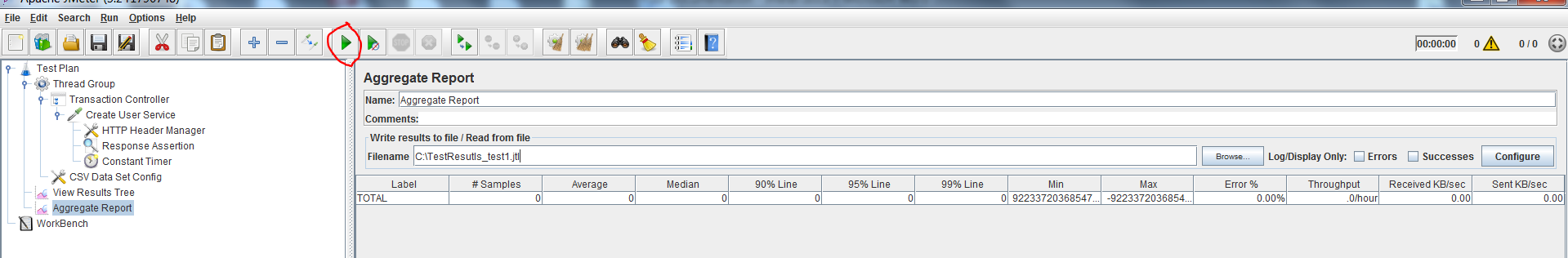
To save the test results to file for a later purpose we can mention the path where we want save. We should name Results path as .jtl extension.

Eg: C:\TestResutls\_test1.jtl



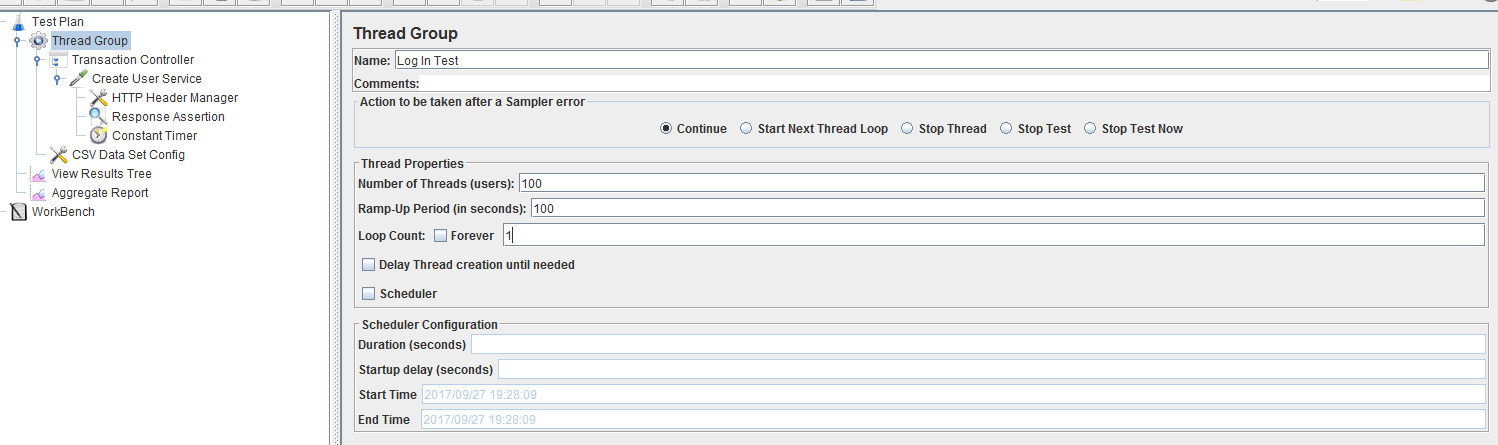
# Running the test and Load Test Execution.

While validating the test we can use “Start” button to execute the request for One or More Users load.



To Stop the test we can always click on “Stop” Button.

To set the required load we can set everything under Thread Group. Click on thread group to get more options for load settings.



The above snippet show the Load as 100 users will be ramped up in 100 Seconds, each user will do one Iteration and exit.

If we want to execute for a particular duration we can user Scheduler.