JMeter

Load testing is kind of performance testing that help determine how application behave when multiple user access it simultaneously.

**Thread**: Thread is one user. Each user simulates one real request

**Sampler**: JMeter support FTP, JDBC, HTTP, BSF, ACEESS LOG, SMTP

**Listener**: It shows the result of Test Execution in form of graph, Table, Tree, Log

**Configuration**: it setup default and variable use by sampler

**Assertion**: used to validate response of the request

**Connection Time**: Time to connect to a server from client

**Response Time**: when we send request it connect to server and get back response

**Throughput**: number of transactions per second an application handle

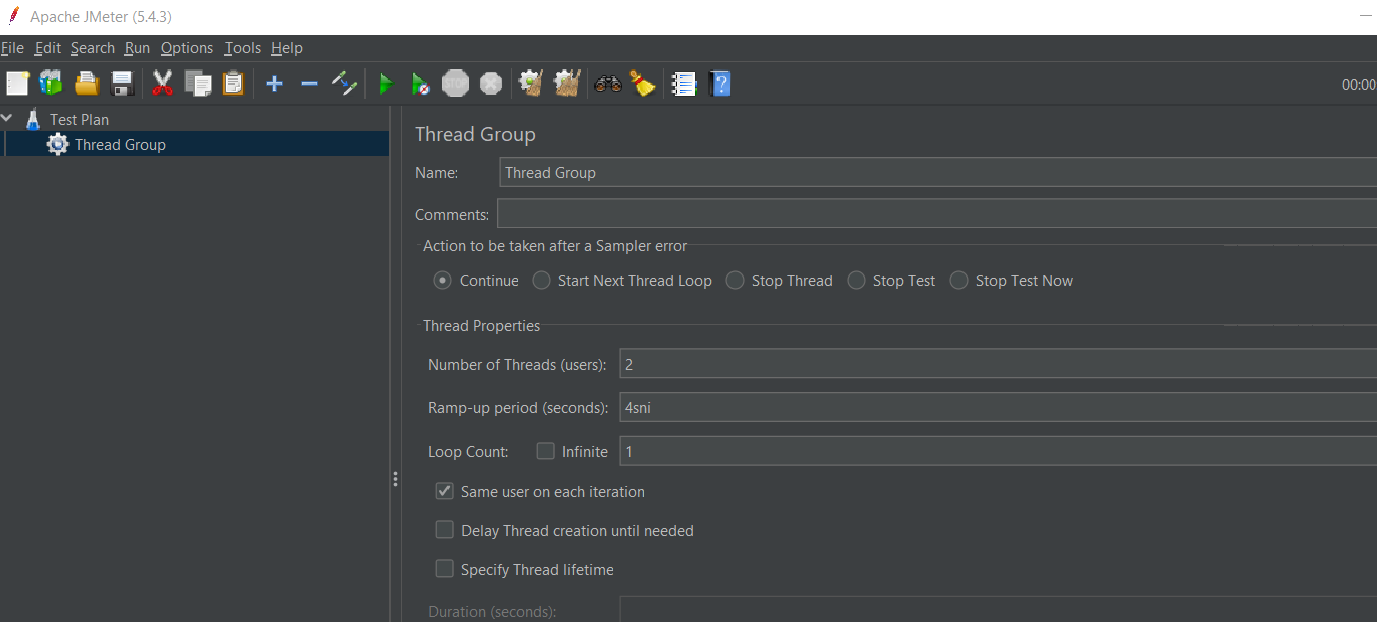
**Bottle neck:** when we hit request and we didn’t get response and it stuck there

**Thread property:**

* No of thread = 10 (user)
* Ramp up period: 100
* Each user (thread) has (100/10=10) 10 sec for up and running

**Configuration**

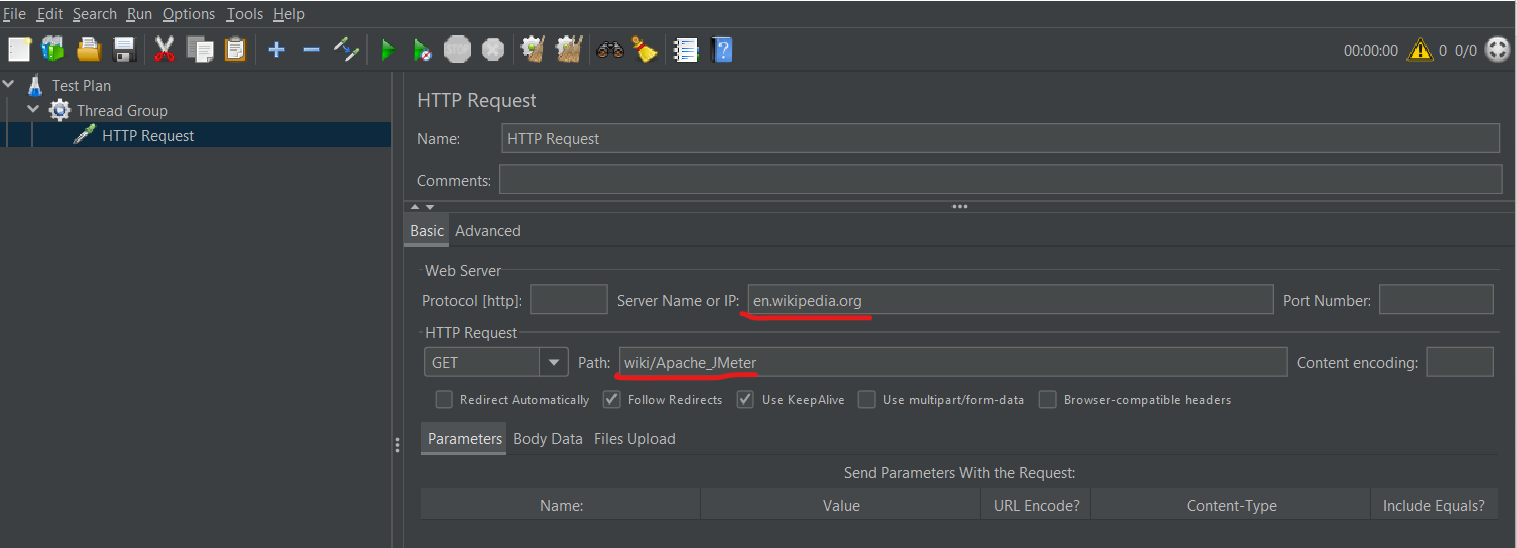
* Click on new create test plan
* Right click on it and add thread
* Give num of thread and ramp up period



Loop count = 1 user test for how many times

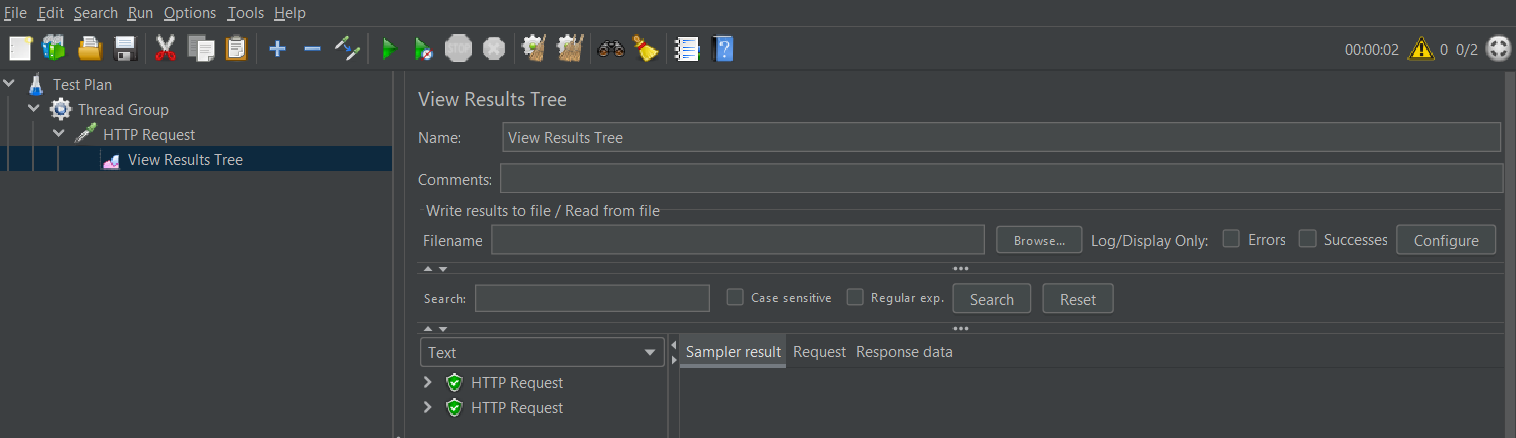
**Now Add Sampler:**

* Right click to thread group
* Add sampler
* Add server name and path

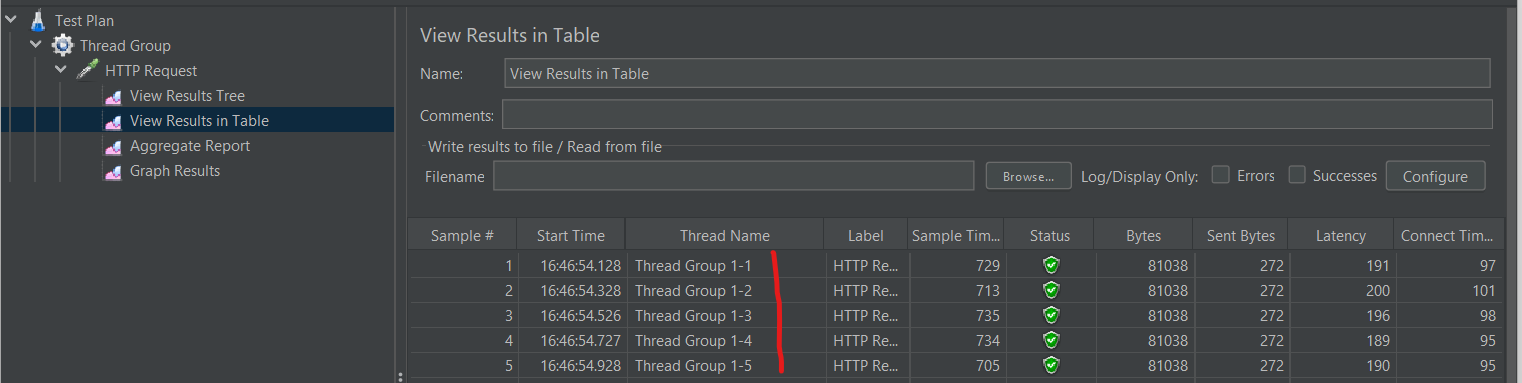
****

**Add Listener:**

* Right click on sampler and add listener

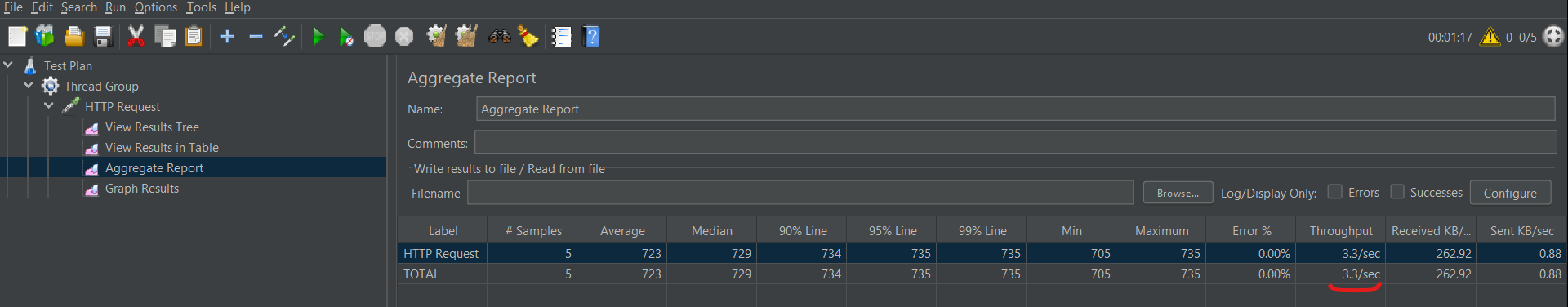


We Add different type of listener



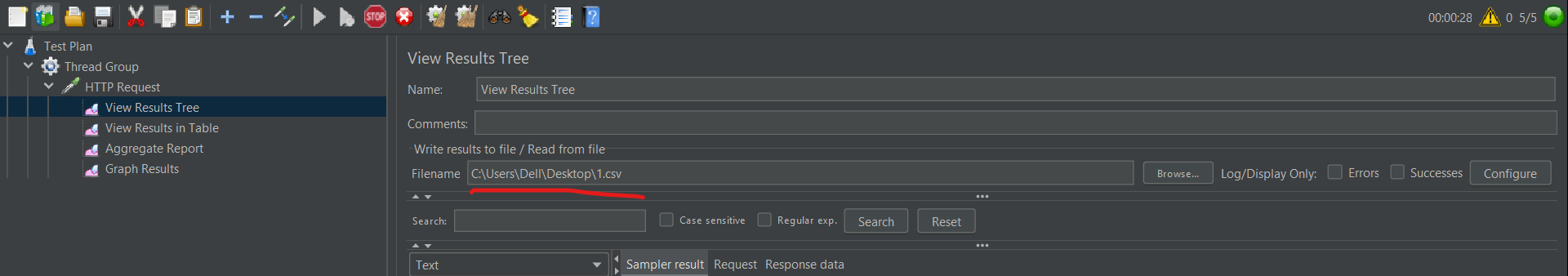
Here 1-1 means 1 iteration 1 user like wise

Throughput:

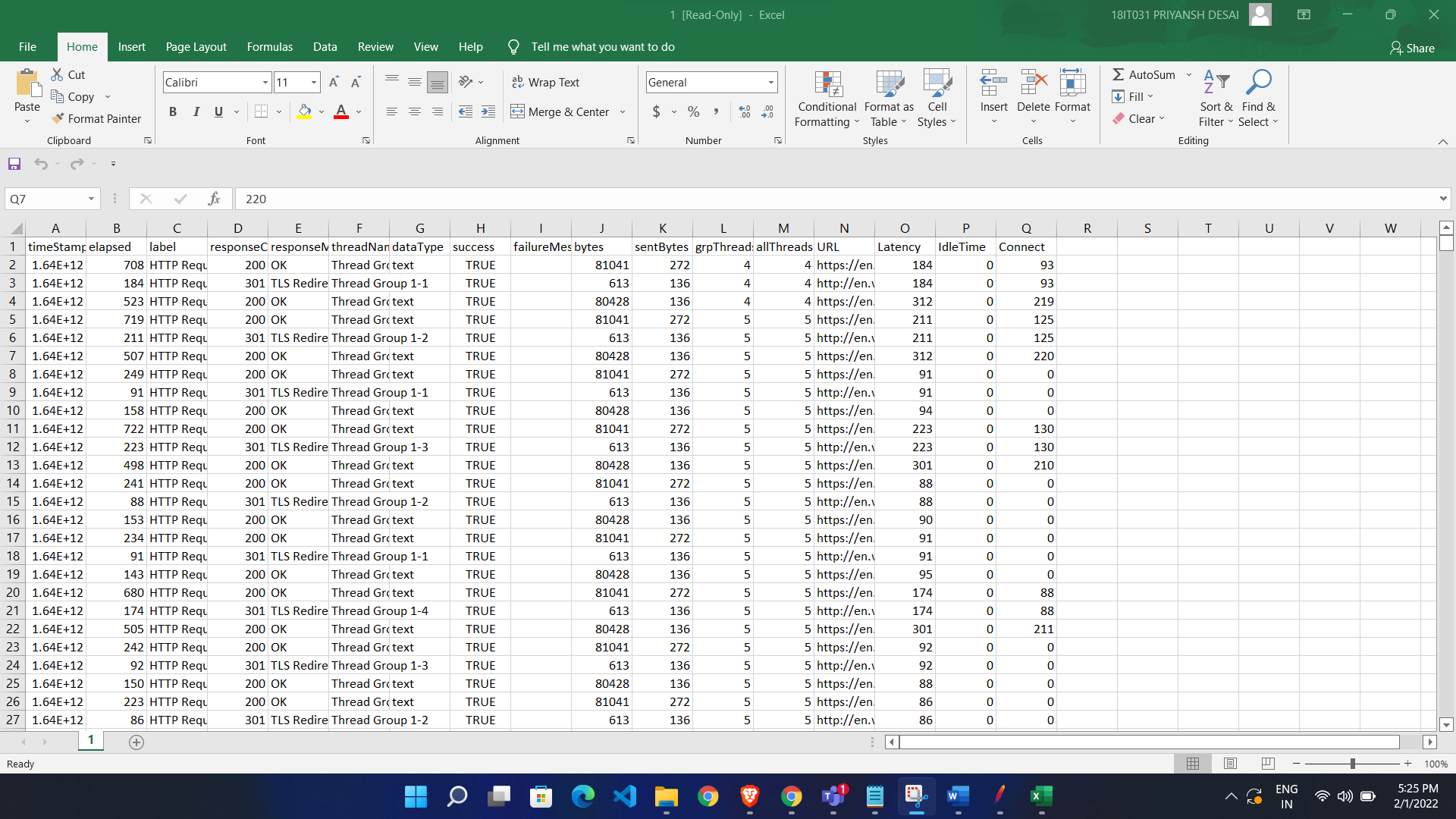


Here we can say that 3.3 req per sec to launch website

**Put Result in some file**



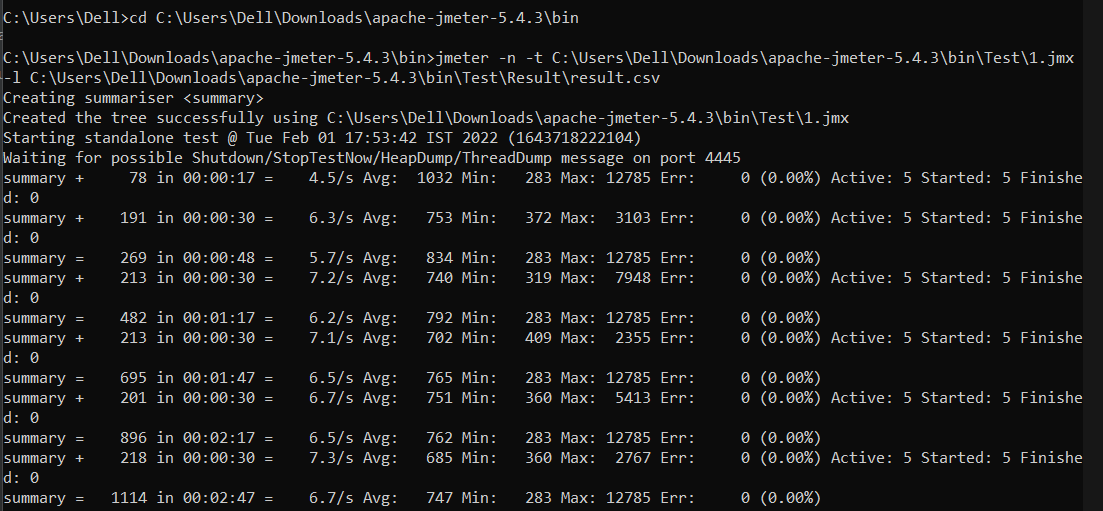
Our result store in that file

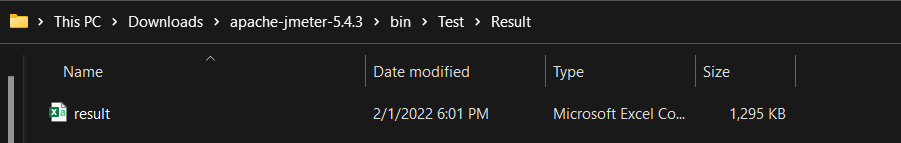


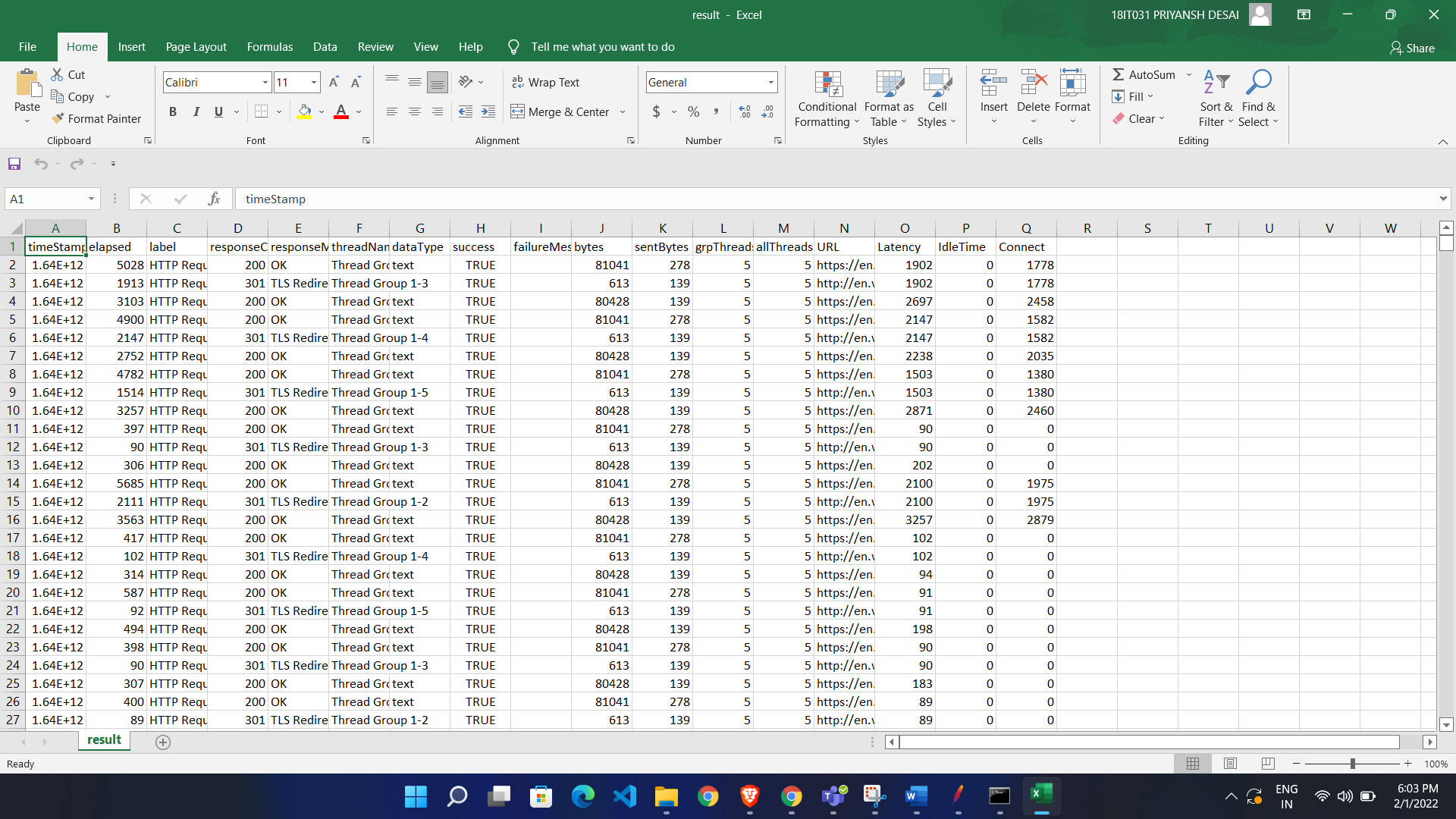
Command Line:

**Jmeter -n -t <path where our file is> -l <where result store>**

jmeter -n -t C:\Users\Dell\Downloads\apache-jmeter-5.4.3\bin\Test\1.jmx -l C:\Users\Dell\Downloads\apache-jmeter-5.4.3\bin\Test\Result\result.csv







Generate HTML Dashboard:

To present some one a report it should be in easy manner so we used it.

**Jmeter -n -t <path where our file is> -l <where result store> -e -o <path where store html report>**

**-n = use command line**

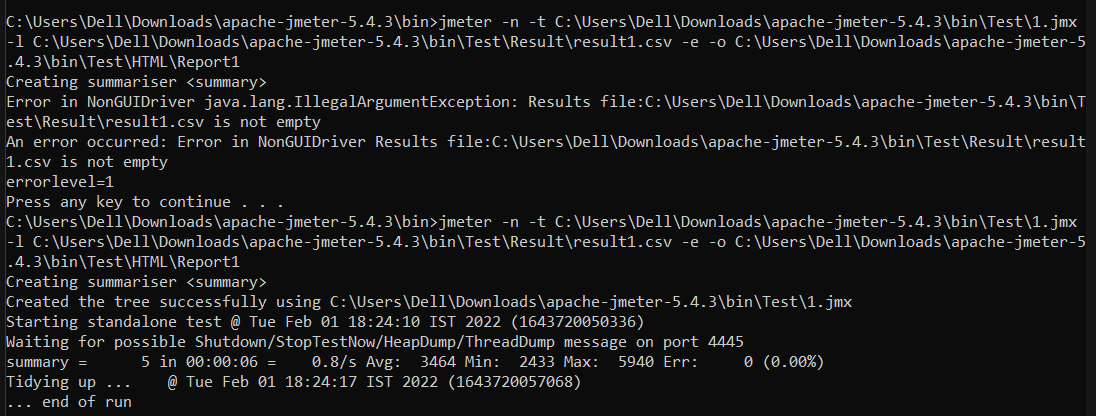
**-t = path where our file**

**-l = where to store**

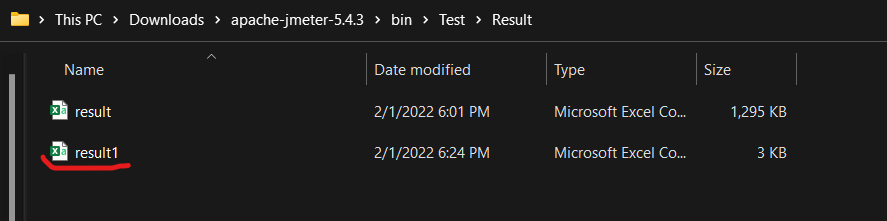
**-e = generate html report**

**-o = path for save html report**

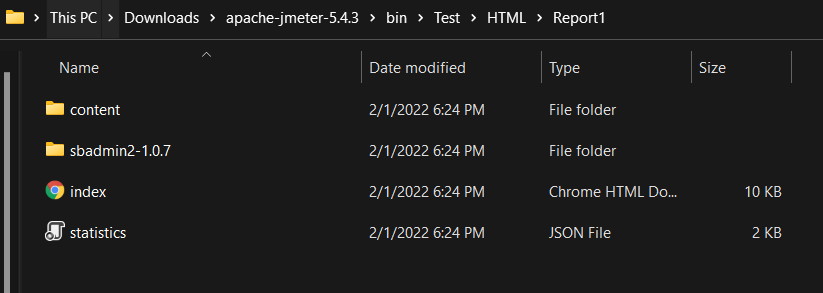
jmeter -n -t C:\Users\Dell\Downloads\apache-jmeter-5.4.3\bin\Test\1.jmx -l C:\Users\Dell\Downloads\apache-jmeter-5.4.3\bin\Test\Result\result1.csv -e -o C:\Users\Dell\Downloads\apache-jmeter-5.4.3\bin\Test\HTML\Report1

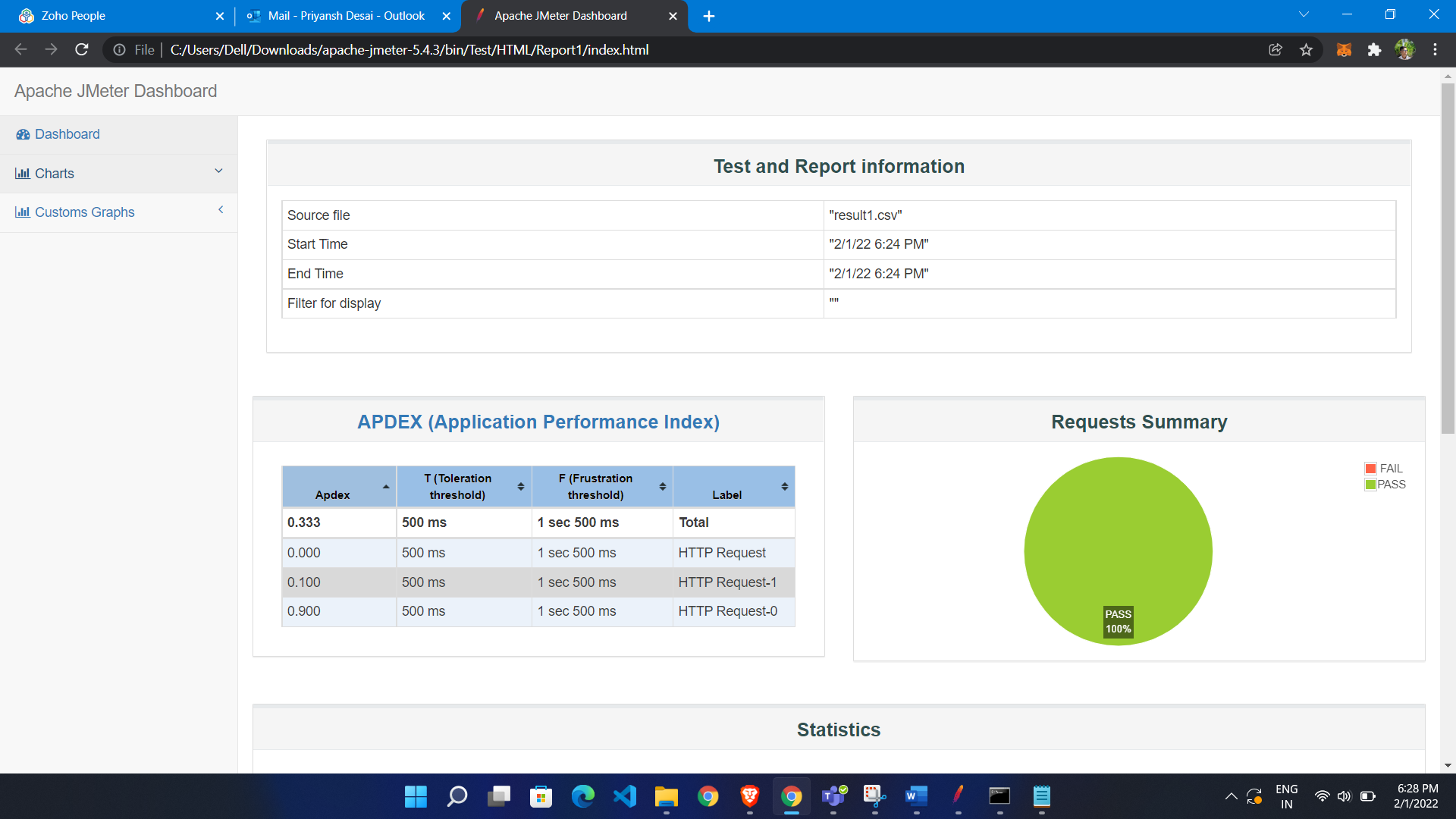


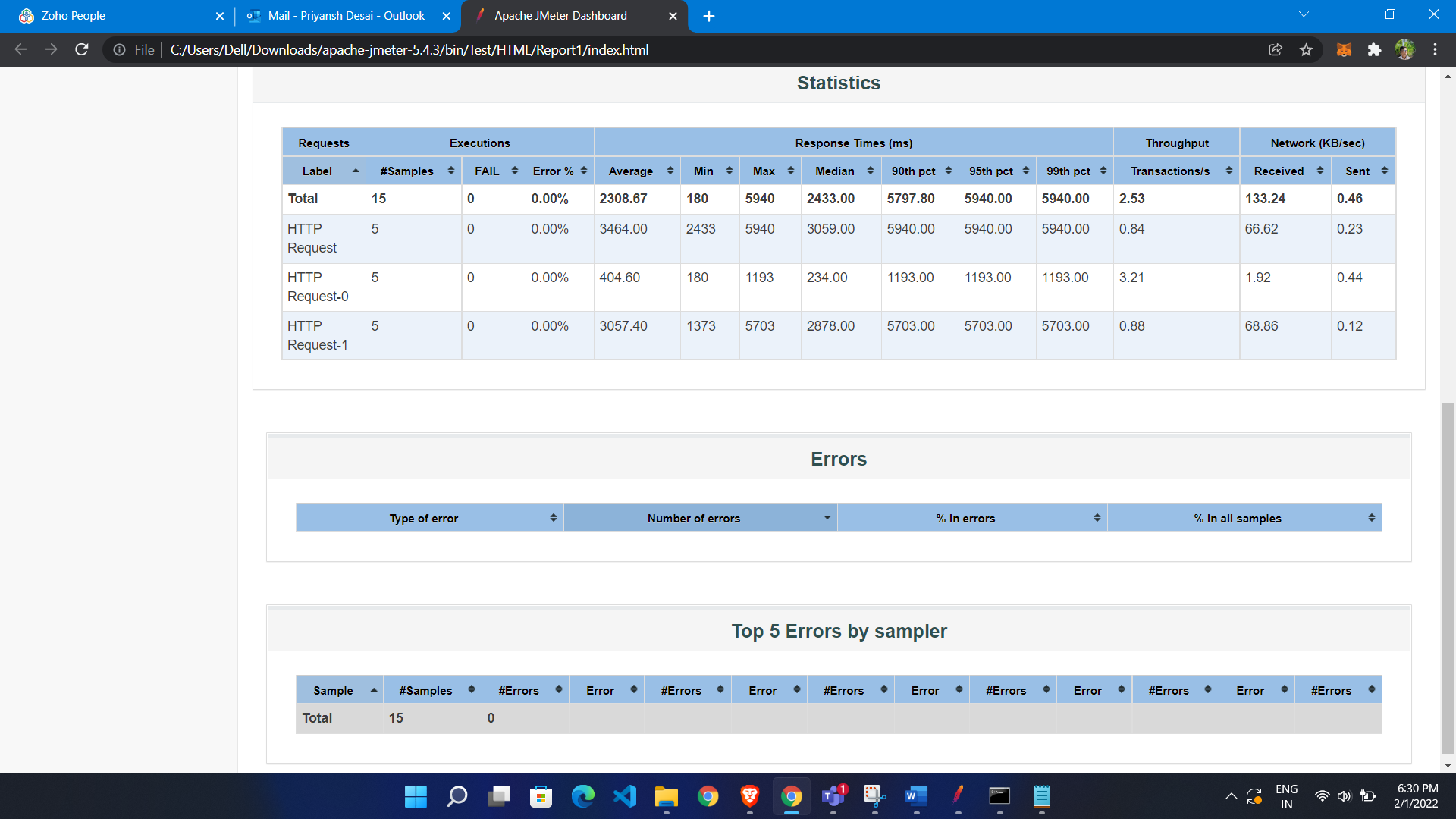
It will generate report 1



And create html report







**Parameterization:**

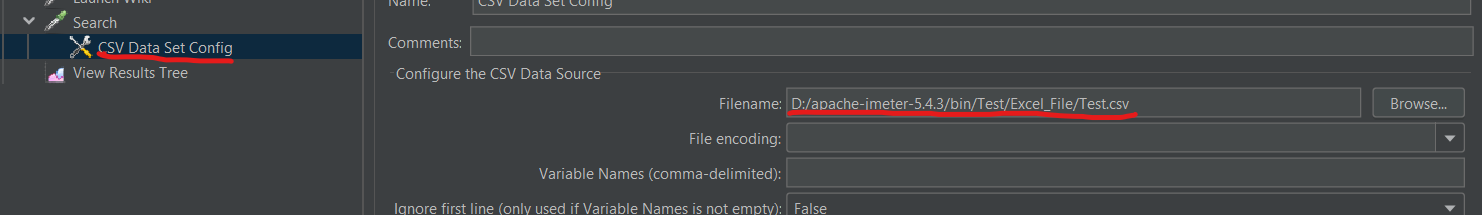
1. Put Test data in csv file
2. Configure JMeter to read that test data from csv file
3. Replace hardcoded test data with variable

**Put data in csv**

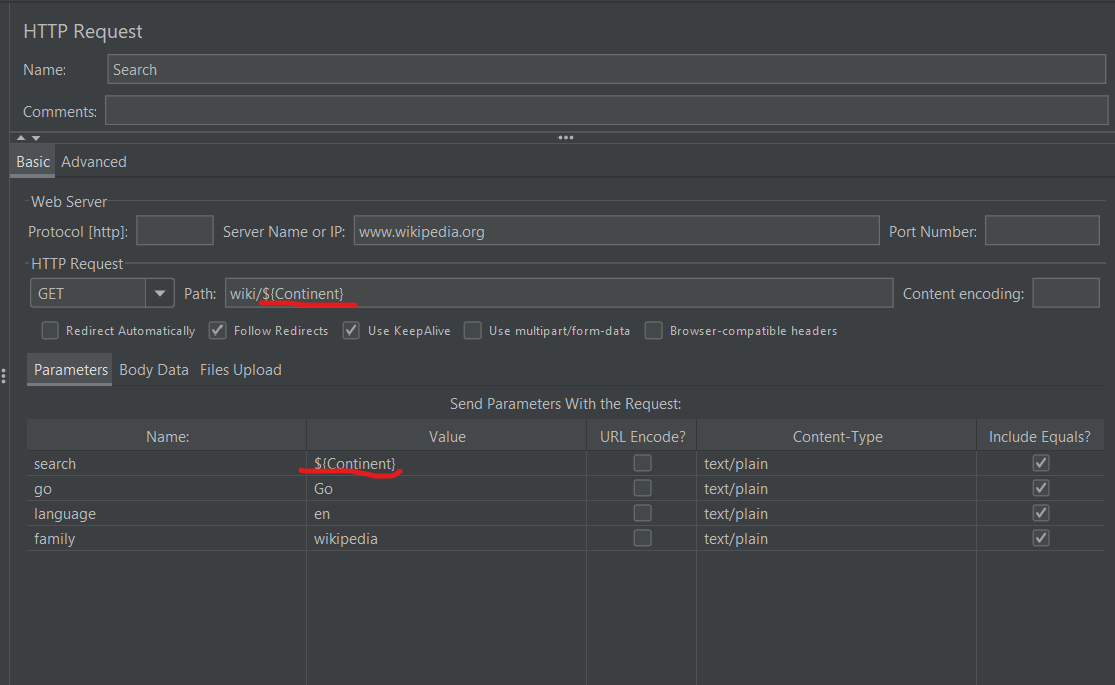


Configure JMeter

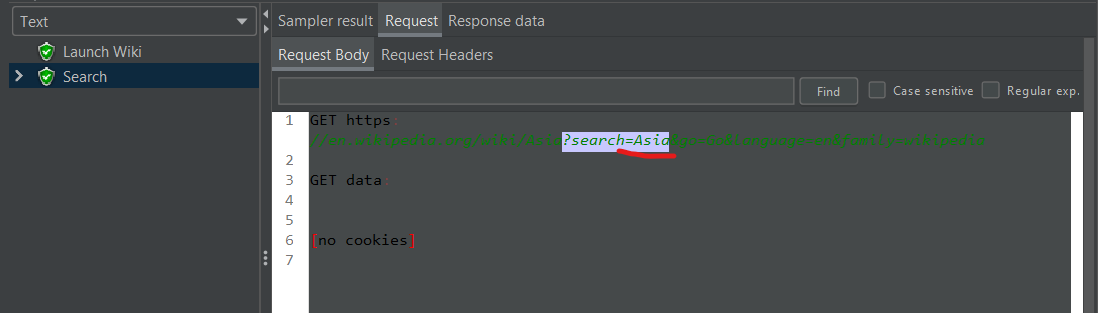
* Right click on search sampler
* Click add
* Click config Element
* Click csv data set config



Add parameter



We get url as Asia



Allowed quote data : if we have data like “IND,india” so this is our one value so if we allow quote data than it will not separate it by , it will take whole one value

Recycle on EOF ( End Of File) : when we have iteration like 2 so once it move to end than it further start it from beginning so we want that we have to make true value for recycle on EOF

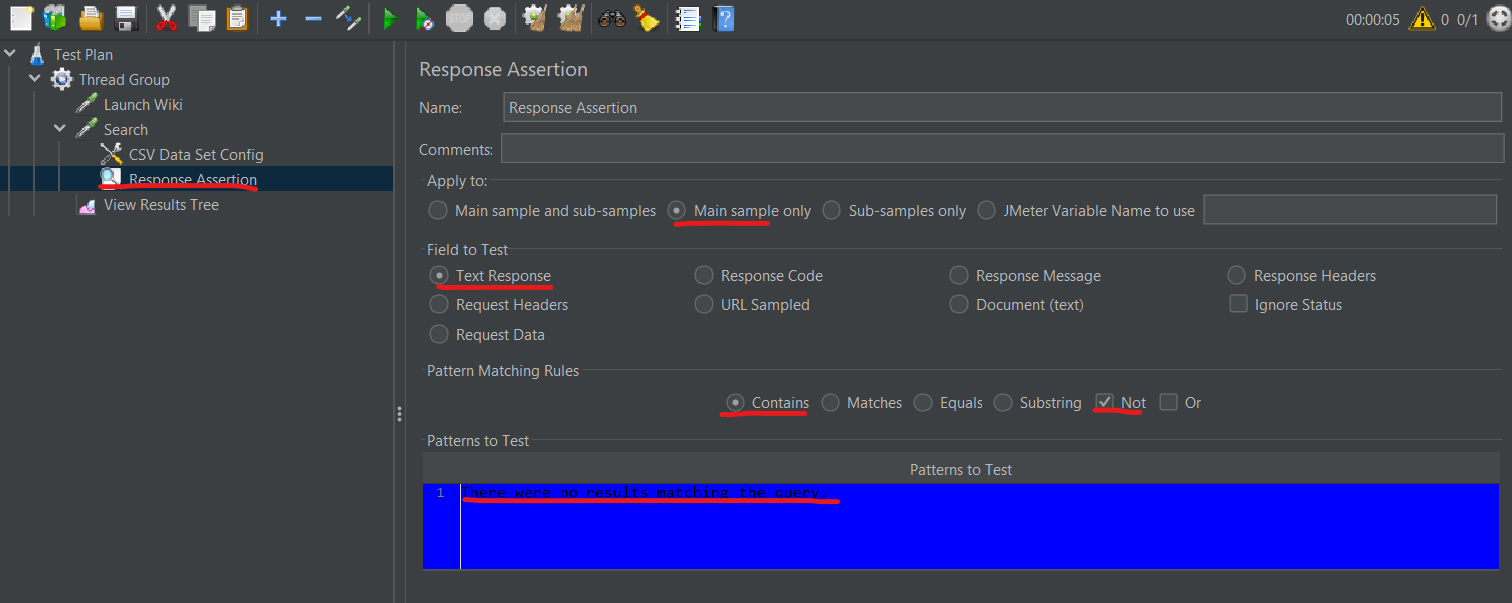
**Assertion:**

It is used for validate our response

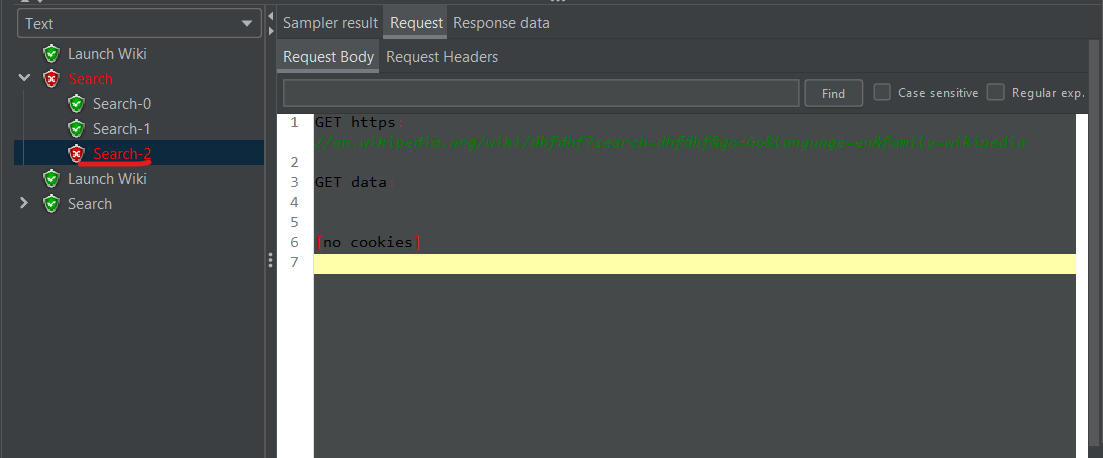
First we add invalid value to test data



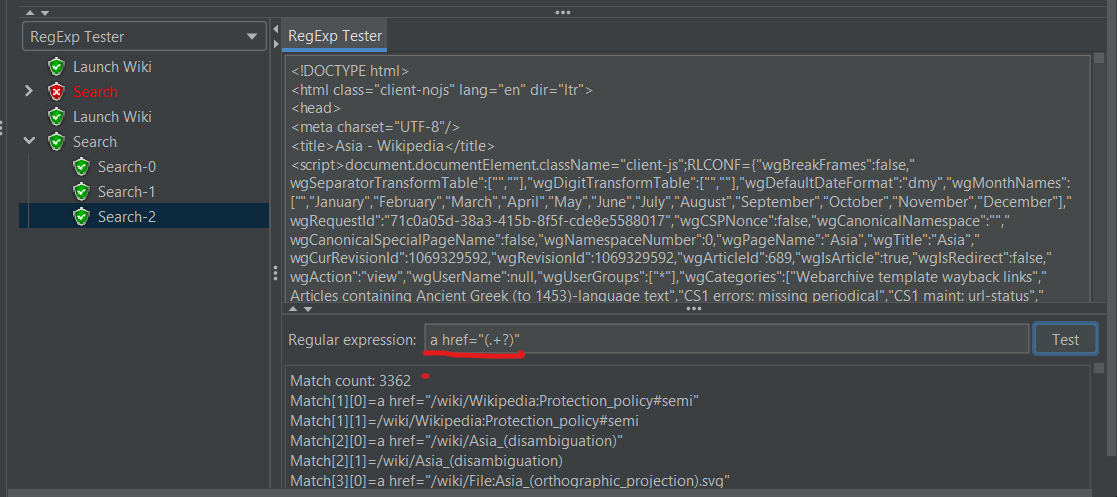
* Now right click on sampler
* Add assertion
* Add response assertion



We got fail response



Debug:



**Dynamic Value:**

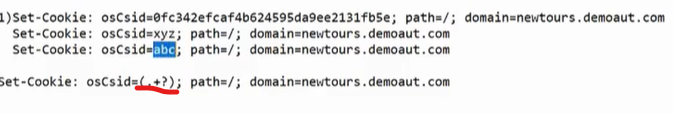
Values get generated in response changing every time for each iterating response

**How to handle dynamic value?**

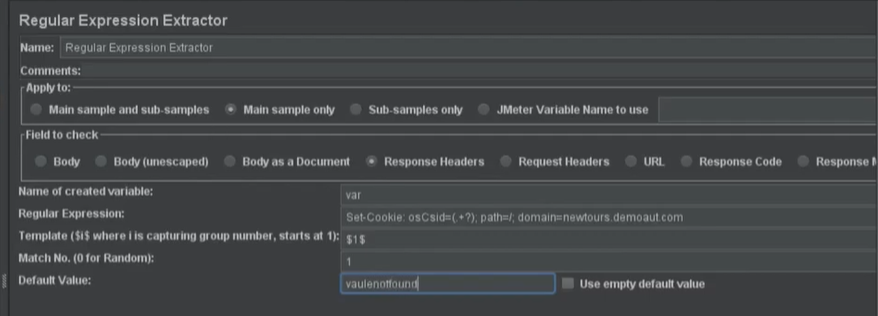
* Capture and store dynamic value from the response
* Use that captured dynamic value in subsequent req

TO get this dynamic value we add regular expression

* Go to where we get response
* Add post Processor
* Add regular Expression Extractor
* Give name of variable

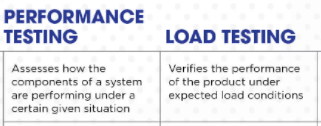


* This will give you dynamic value(Add Regular Expression)



In login in parameter give variable value

**Performance Testing:**

****

Testing stability (ability to with stand) and response time by applying load is called as performance testing.

Stability: when we develop software product manager tell it is able to handle how many users at time

Response time: Time taken to request to server run the program in server and getting response back

It’s Type:

**Load Testing:**

We have developed one application for 50 user.in load testing we test it with only 45 user or max for 50 users

Testing stability and response time by applying load which is less than designed number of user or equal to designed num of user.

**Stress Testing:**

Testing stability and response time by applying load which is more than designed number of users.

Ex. My application able to handle 1000 request in 3 sec so we apply load more than 1000

**Scalability Testing:**

Testing stability and response time by applying load which is more than designed number of users and find out where our application is crashing

**Volume Testing/Flood Testing:**

Testing stability and response time by Transferring huge volume of data.

EX. our database use 1gb of data on our website. if we upload 50video than it is able to handle it.

**Soak Testing:**

Testing stability and response time by applying load continuously for longer period of time.