

//HTTP Authentication Manager, Cookies, and CSV Set Config

Development Environment:

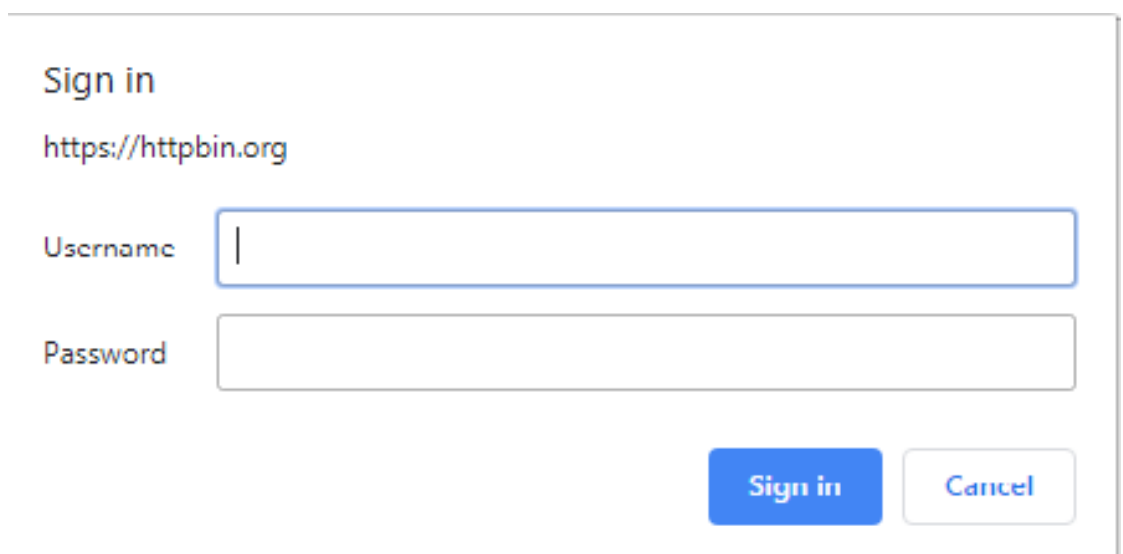
- Apache JMeter 5.1.1 Version

This guide has four subsections, namely:

- 1.29.1 Using HTTP authentication manager
- 1.29.2 Using HTTP cookie manager
- 1.29.3 Reading values from a CSV file using CSV data set config
- 1.29.4 Pushing the code to GitHub repositories

Step 1.29.1: Using HTTP authentication manager

- Let's take an example of "https://httpbin.org/basic-auth/user/passwd".
- When you click on the https://httpbin.org/basic-auth/:user/:passwd link, your browser should prompt you to enter a username and password.

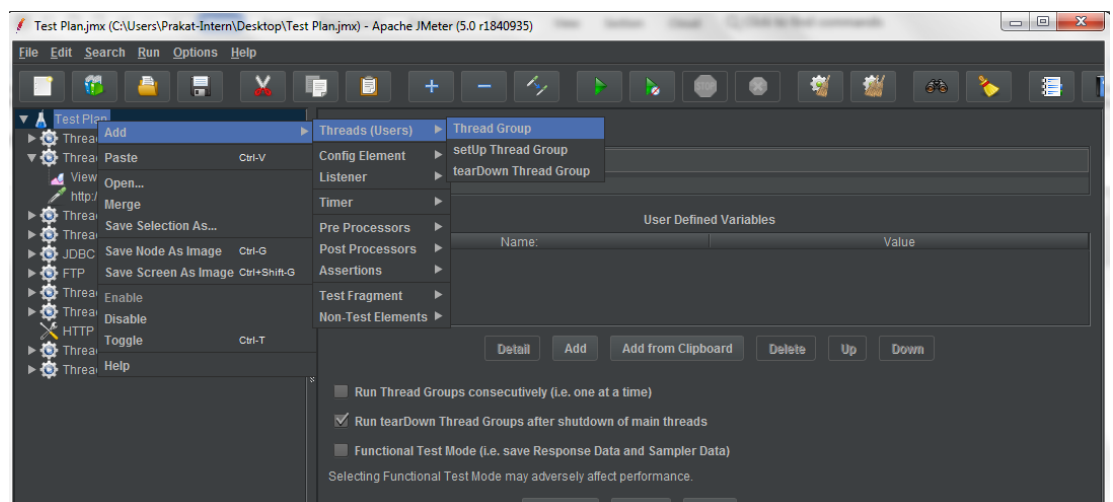


The image shows a standard browser basic authentication dialog box. At the top, it says "Sign in" in blue text, followed by the URL "https://httpbin.org" in red text. Below this, there are two input fields: "Username" and "Password". The "Username" field has a blue border and a cursor, while the "Password" field has a grey border. At the bottom right, there are two buttons: a blue "Sign in" button and a grey "Cancel" button.

- Provide the default username as user and the default password as passwd. The server should respond with the following json:

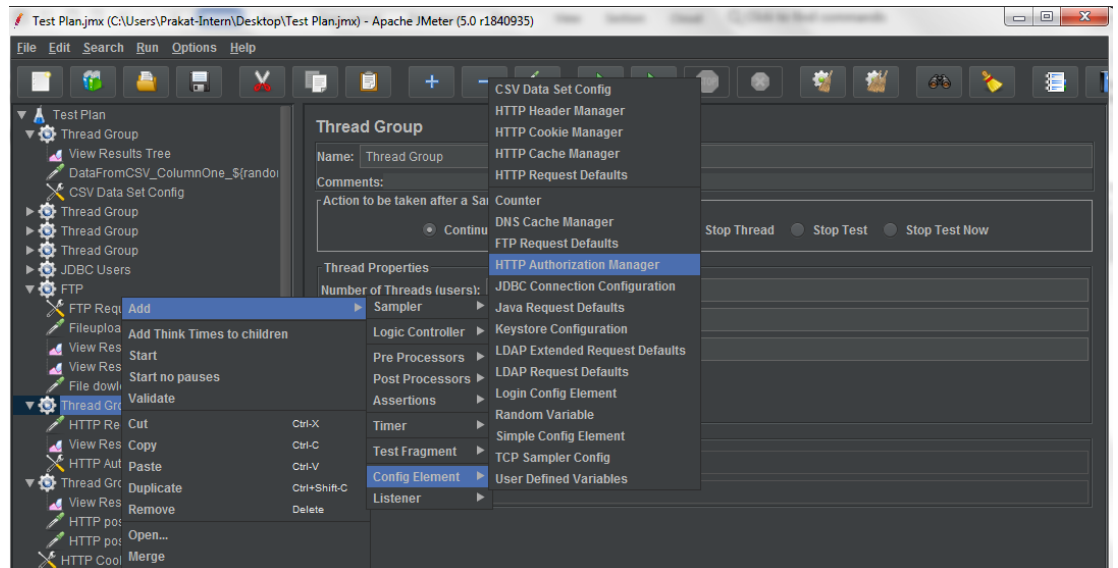


- For wrong username and password, the server rejects access to the protected resource with an **HTTP 401: Unauthorized** error.
- Adding users in thread group:
 1. Right click on Test Plan.
 2. Click on Add -> Threads(users) -> Thread Group.

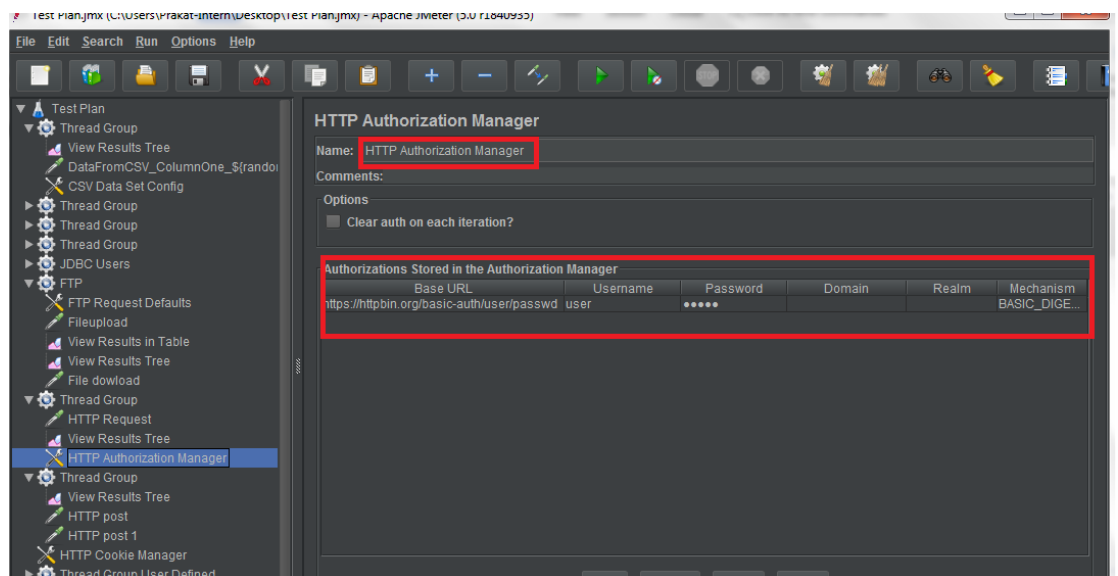


3. Set Number of Threads to 1.
4. Set the Ramp-Up period to 1 Second.
5. Set Loop Count to 1.

- Adding HTTP Authorization Manager:
 1. Right click on Thread Group.
 2. Click on Add -> Config Element -> HTTP Authorization Manager.



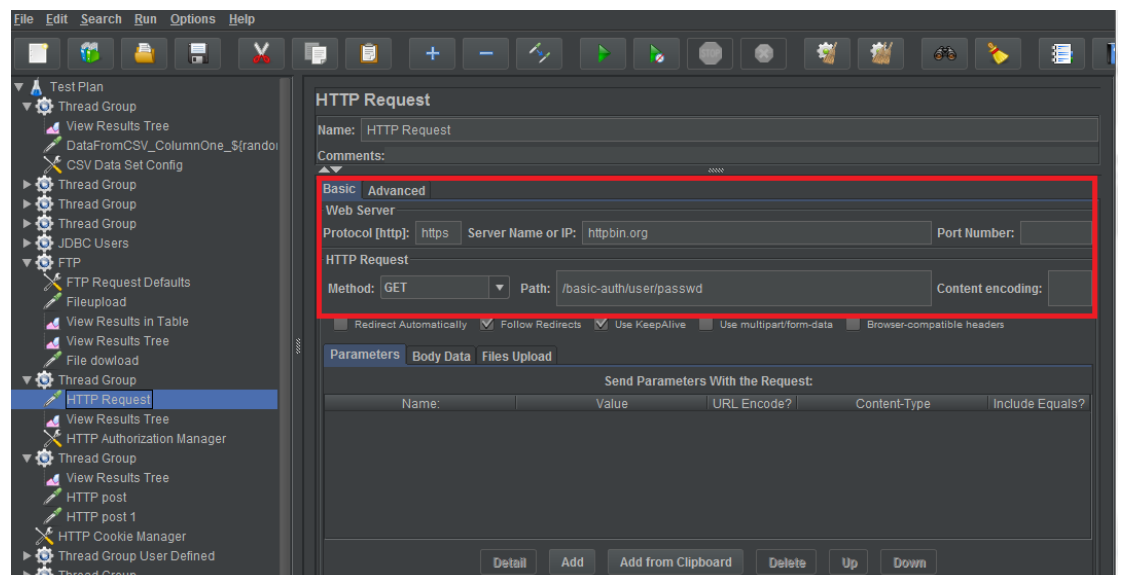
3. Enter the required fields in HTTP Authorization Manager.



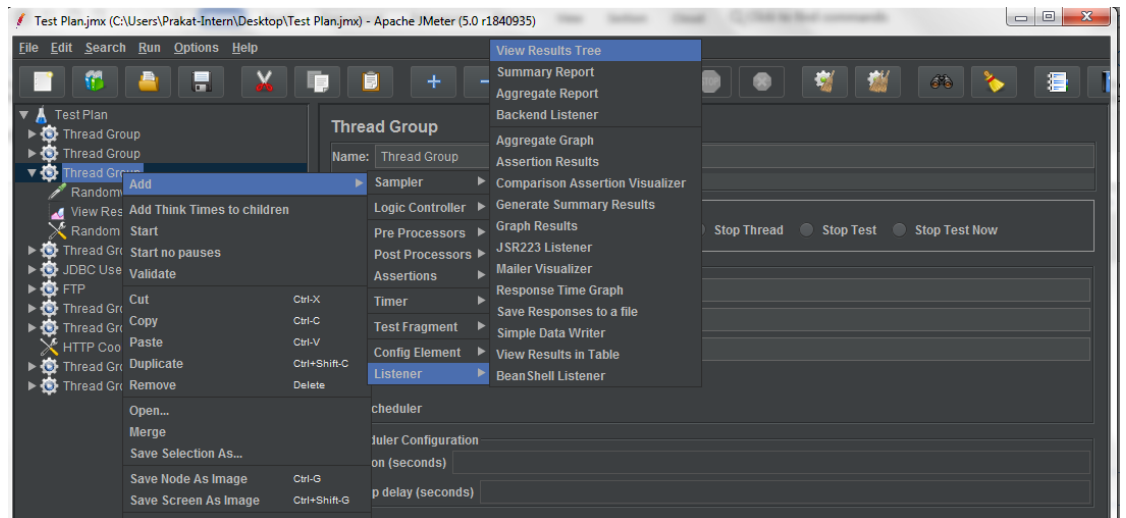
- Base URL:** https://httpbin.org/basic-auth/user/passwd
- Username:** user
- Password:** passwd

IV. Mechanism: BASIC_DIGEST

- Adding HTTP Request:
 1. Right click on Thread Group.
 2. Click on Add -> Sampler -> HTTP Request.
 3. Enter protocol: https.
 4. Enter server name: httpbin.org.
 5. Enter path: /basic-auth/user/passwd.



- Adding View Results Tree:
 1. Right click on Thread Group.
 2. Click on Add -> Listener -> View Results Tree.



3. Run the Thread Group and open the View Results Tree to see the output.

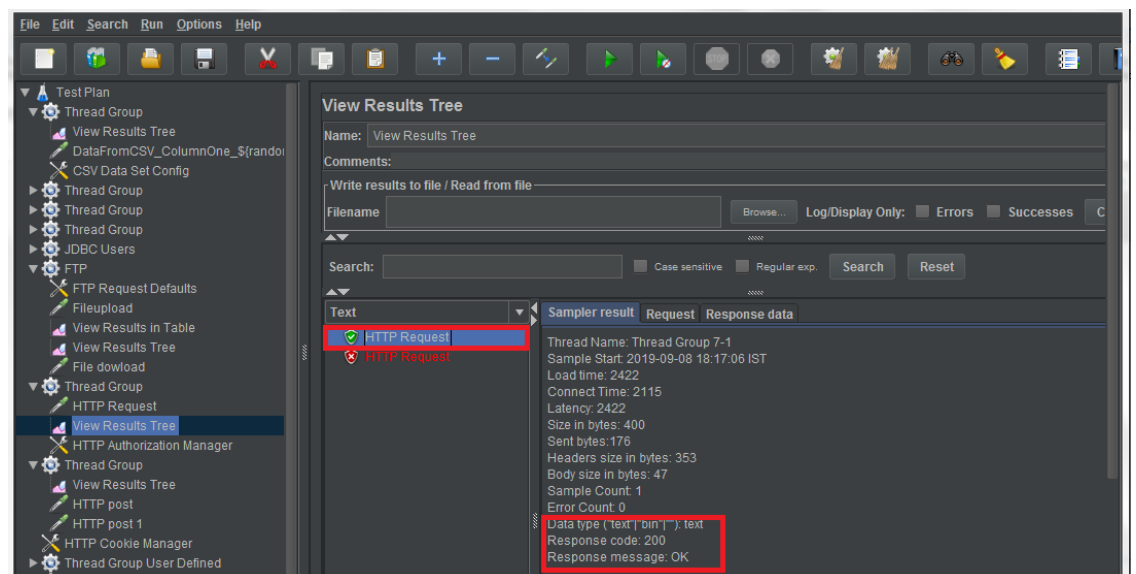
I. Enter valid username: user and password: passwd.

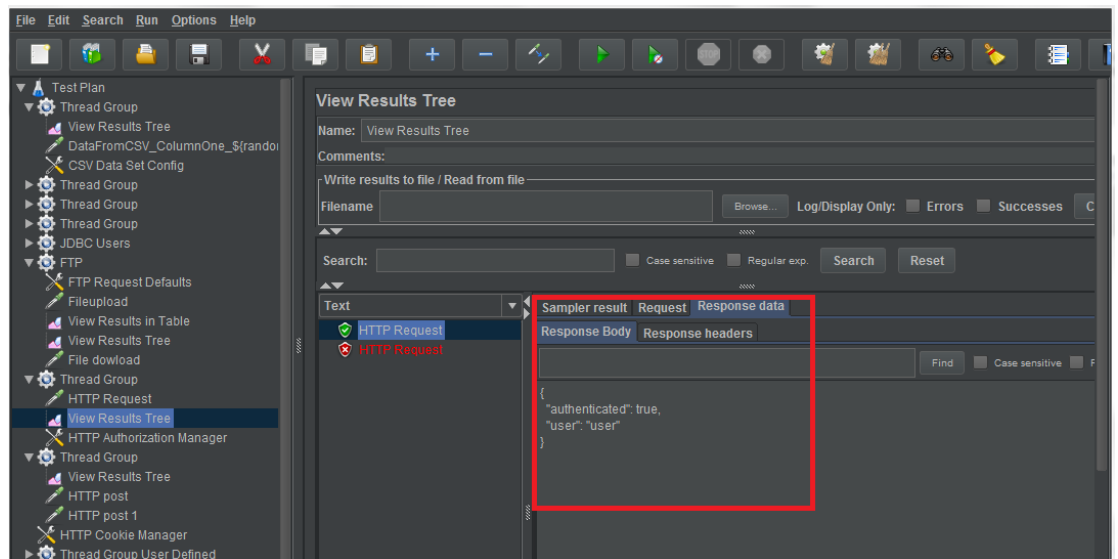
II. The Response code: 200

III. Response Message: OK

IV. Response Data:

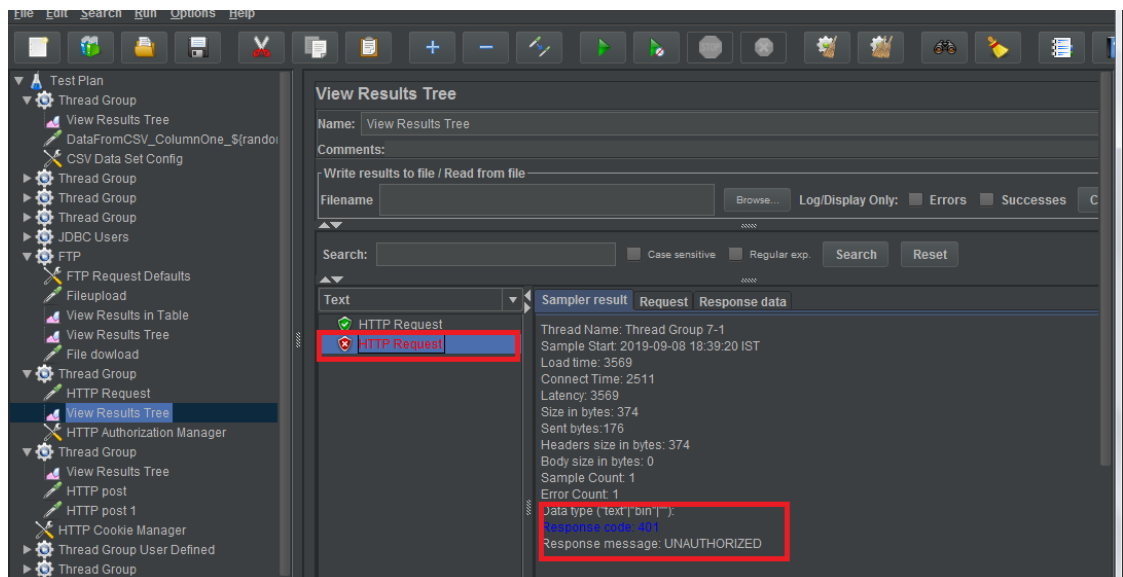
```
{
  "authenticated": true,
  "user": "user"
}
```





For invalid username and password or By Disabling HTTP Authorization Manager the output will be:

- I. Response Code: 401
- II. Response Message: UNAUTHORIZED

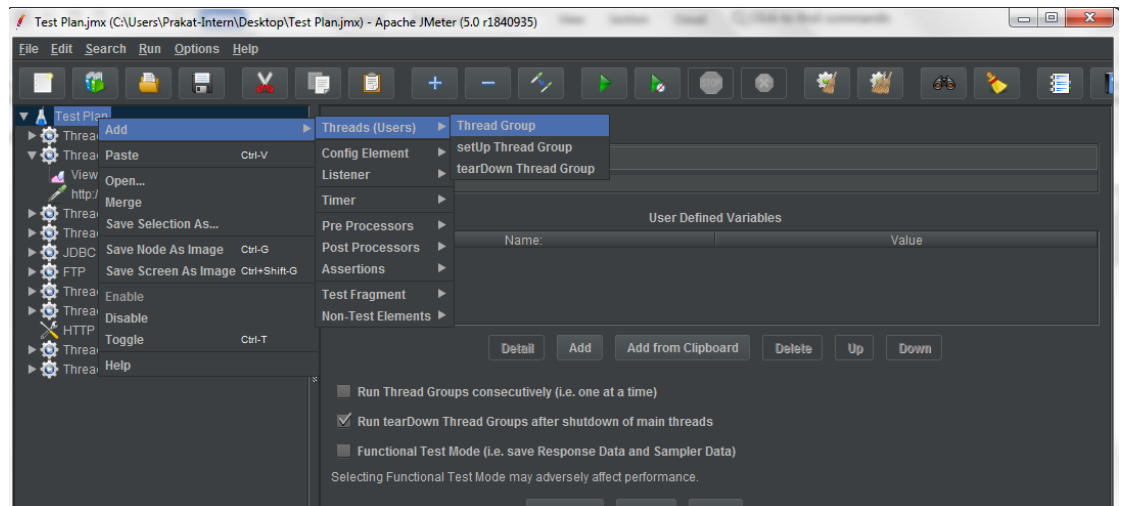


Step 1.29.2: Using HTTP Cookie Manager

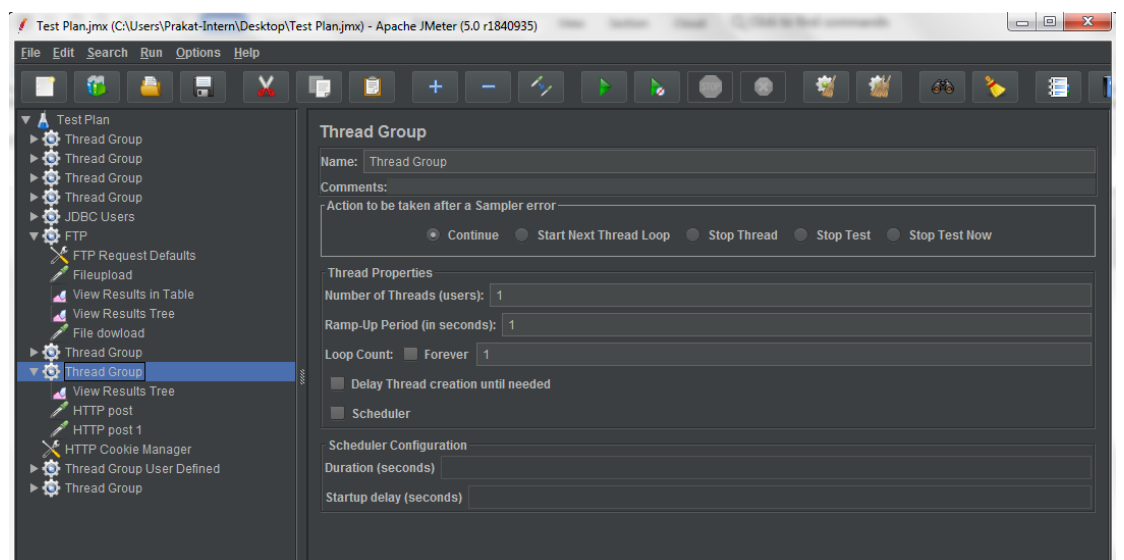
In a stateless internet, many sites and applications use cookies to retain a handle between sessions or to keep some state on the client side. If you are planning to use JMeter to test such web applications, then you will want to use HTTP Cookie Manager.

- Adding users in thread group:

1. Right click on Test Plan.
2. Click on Add -> Threads -> Thread Group.

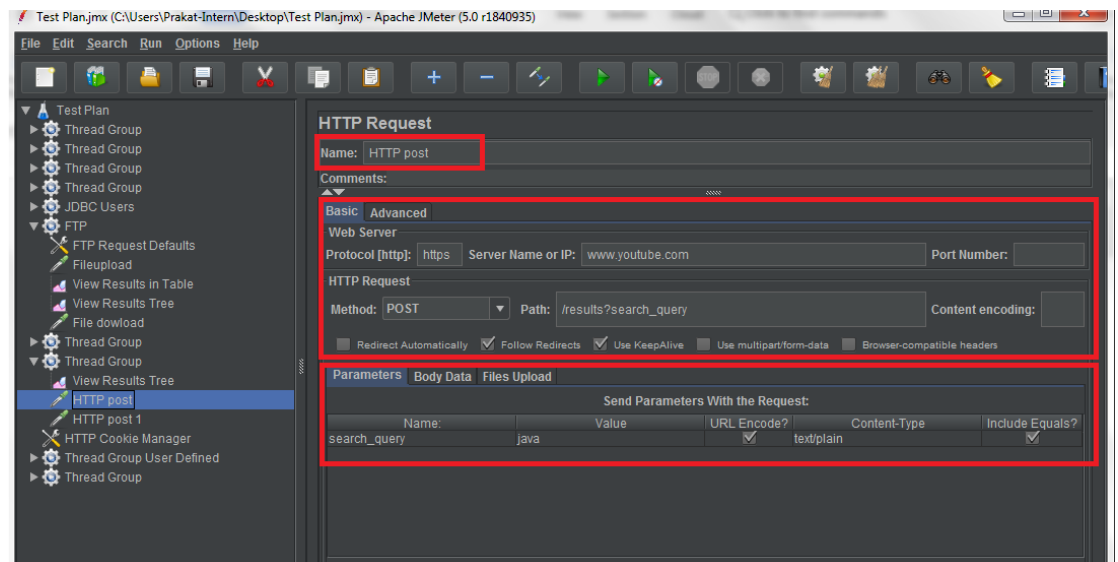


3. Set the number of threads to 1.
4. Set the ramp-up period to 1.
5. Set loop count to 1.



- Adding HTTP request:
 1. Right click on Thread Group.
 2. Click on Add -> Sampler -> HTTP Request.

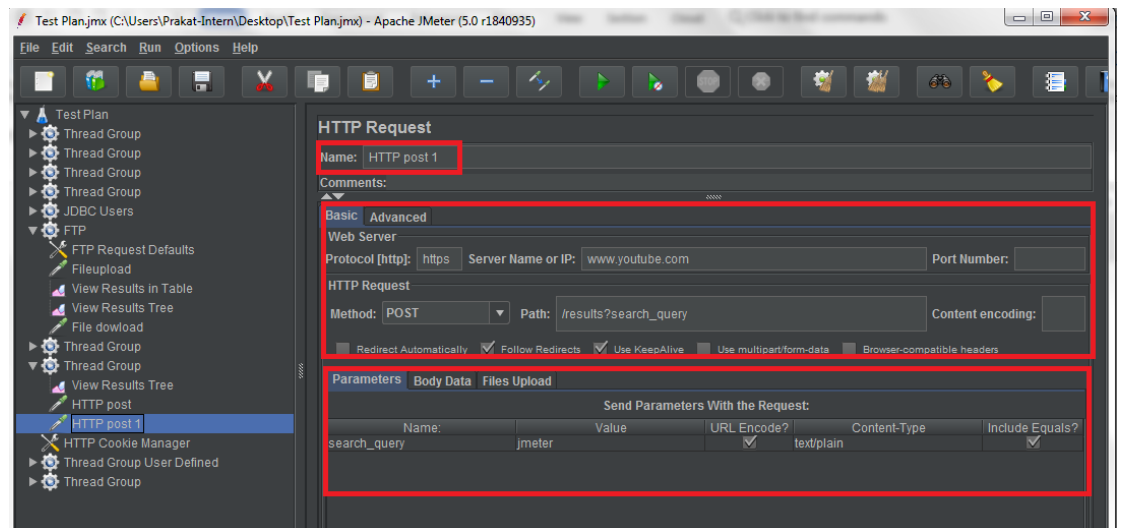
3. Rename HTTP request to "HTTP Post".
4. Set the protocol to "https".
5. Set the server name to "www.youtube.com".
6. Set method to "POST".
7. Set path to "/results?search_query".
8. Add the parameters:
 - I. **Name:** search_query
 - II. **Value:** java
 - III. Check URL ENCODE?
 - IV. Check INCLUDE EQUALS?



- Adding HTTP Request:
 1. Right click on Thread Group.
 2. Click on Add -> Sampler -> HTTP Request.
 3. Rename HTTP request to "HTTP Post 1".
 4. Set the protocol to "https".
 5. Set the server name to "www.youtube.com".
 6. Set method to "POST".
 7. Set path to "/results?search_query".

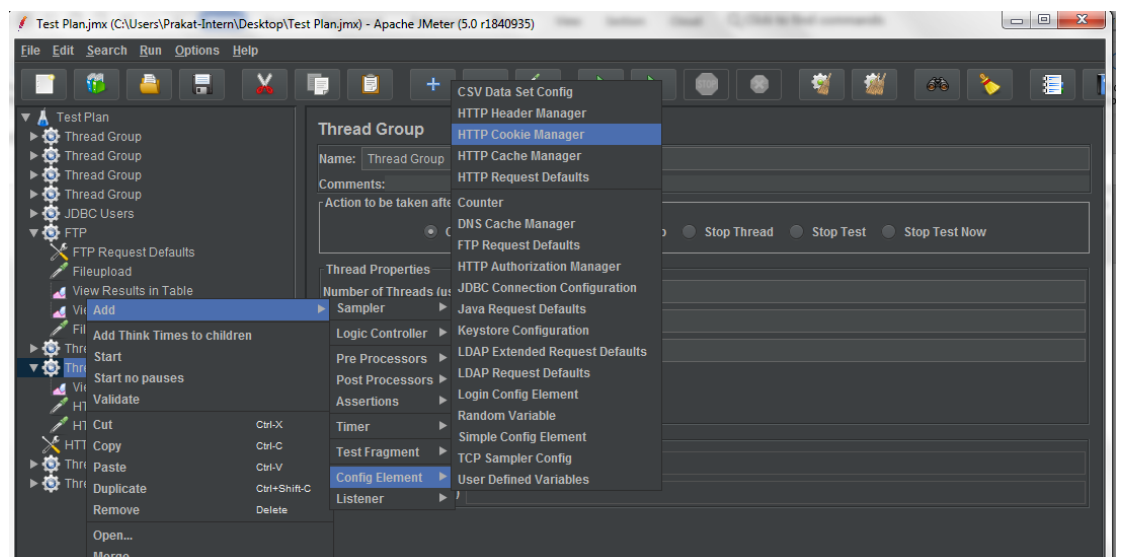
8. Add the parameters:

- I. **Name:** search_query
- II. **Value:** jmeter
- III. Check URL ENCODE?
- IV. Check INCLUDE EQUALS?

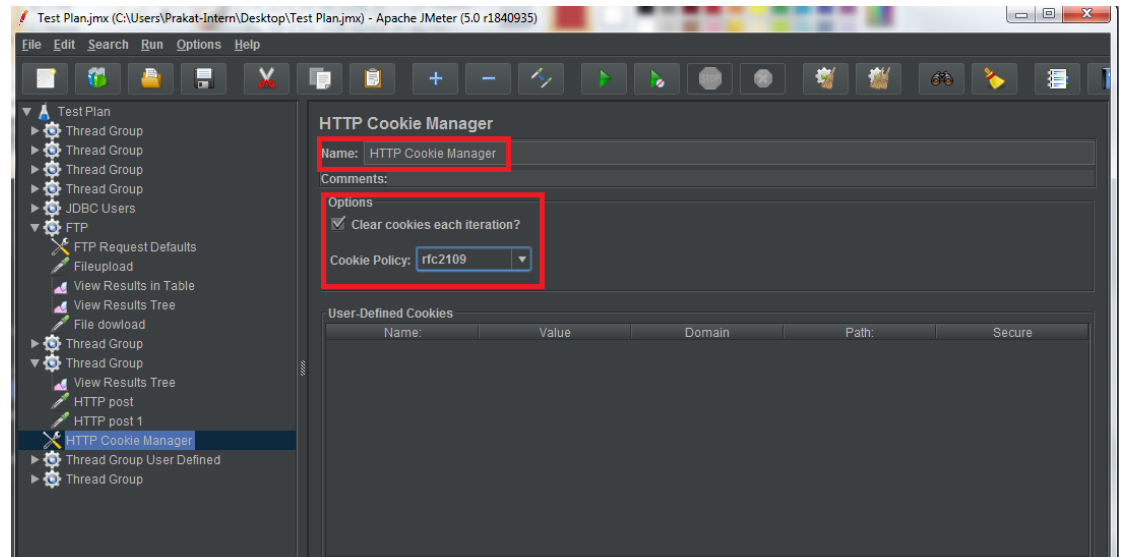


- Adding HTTP Cookie Manager:

1. Right click on Thread Group.
2. Click on Add -> Config Element -> HTTP Cookie Manager.

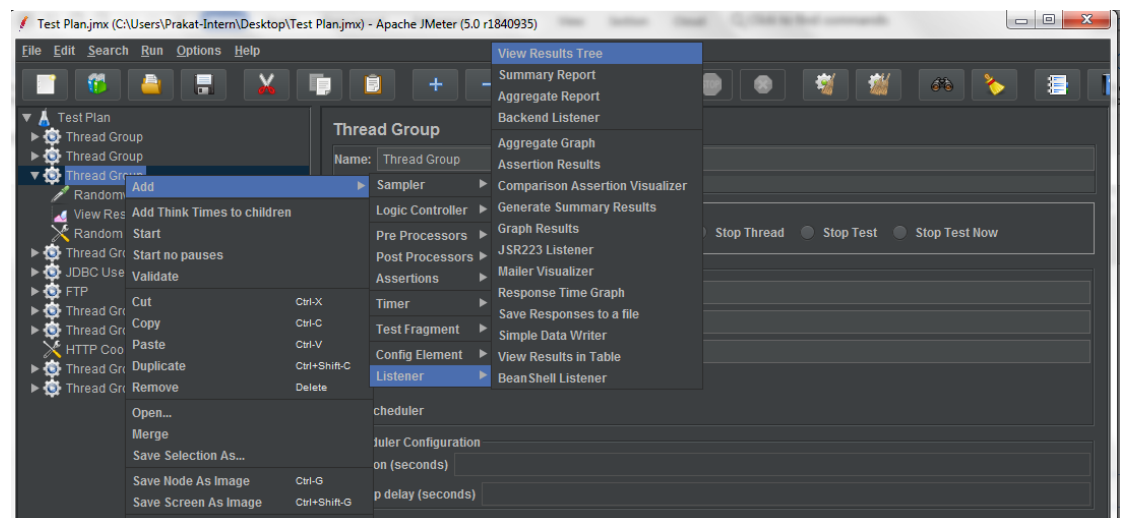


3. Check the “Clear Cookies each iteration?”.
4. Select the cookie policy “rfc2109”.



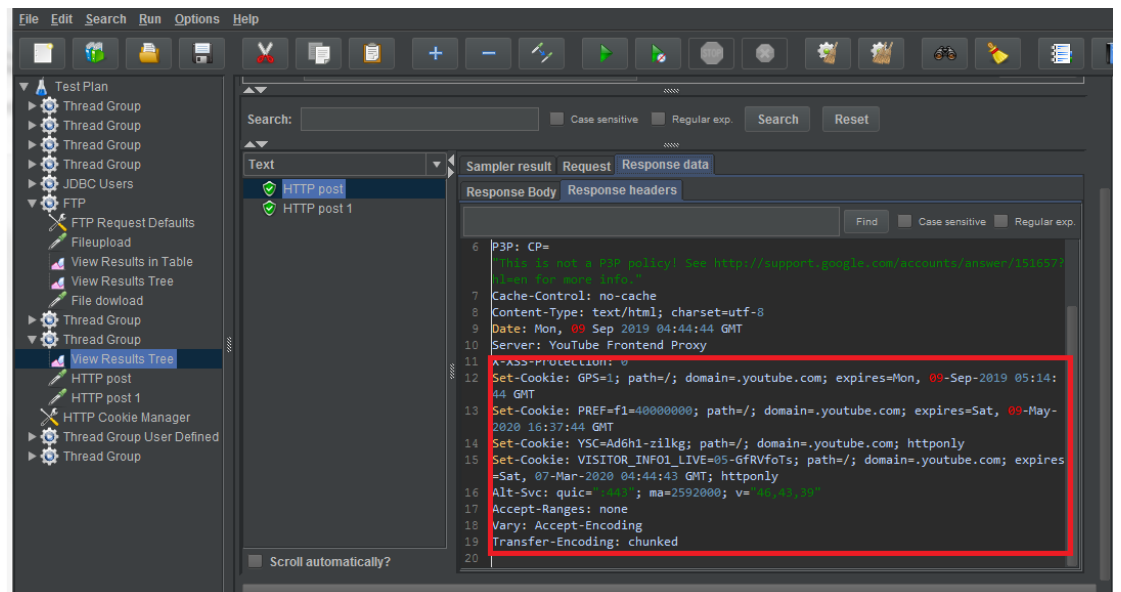
- Adding View Results Tree:

1. Right click on Thread Group.
2. Click on Add -> Listener -> View Results Tree.

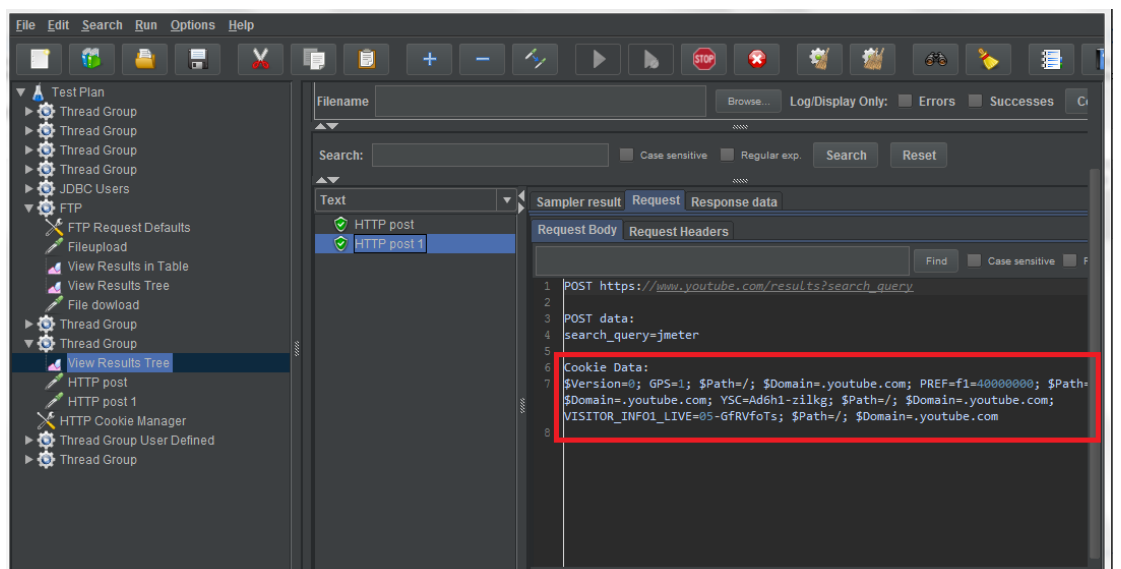


3. Run the Thread Group and open the View Results Tree to see the result.

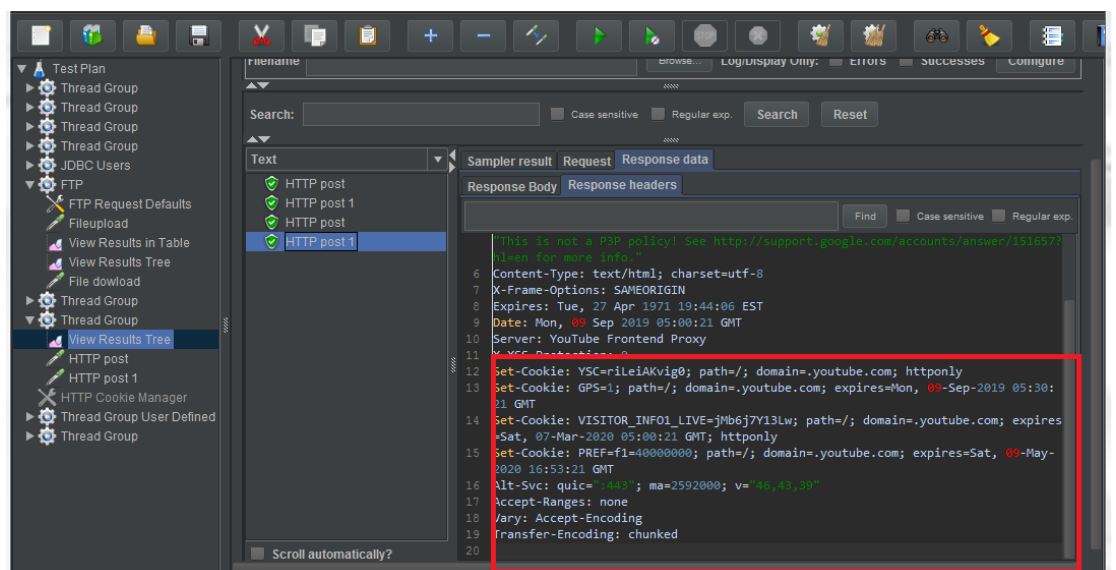
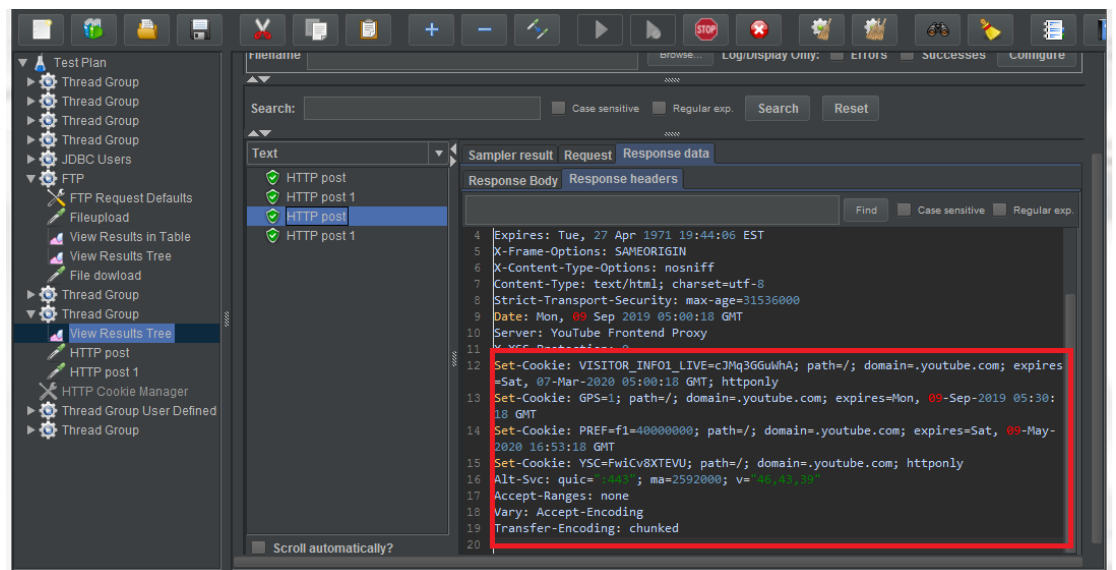
4. As you can see, HTTP Post Request Cookie is set as shown below:



Note that in the additional request HTTP Post 1, the cookie value is the same as when it was received directly from the server.



- Disable HTTP cookie manager:
 1. Right click on HTTP cookie manager.
 2. Click on Disable.
 3. Run the Thread Group and open View Results Tree to see the result.

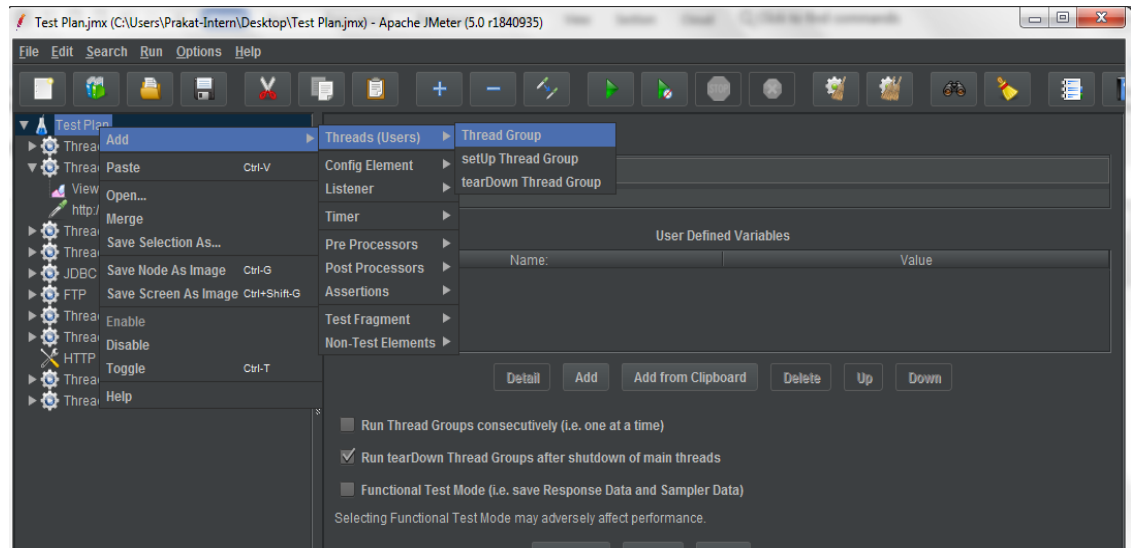


Note that for both the post requests, different cookie values are set.

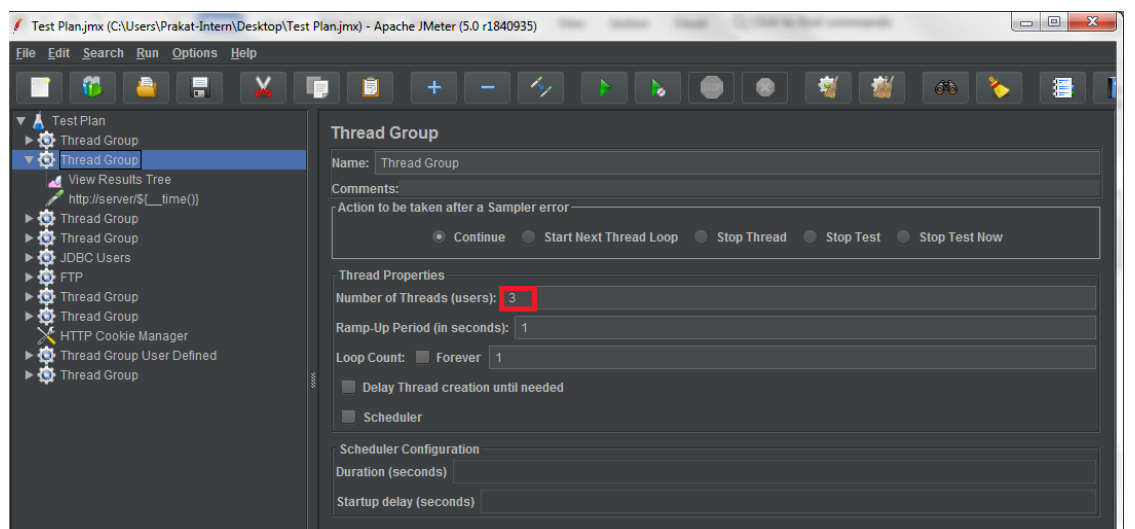
Step 1.29.3: Reading values from CSV file using CSV Data Set Config

- JMeter is an open source load testing tool that 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.

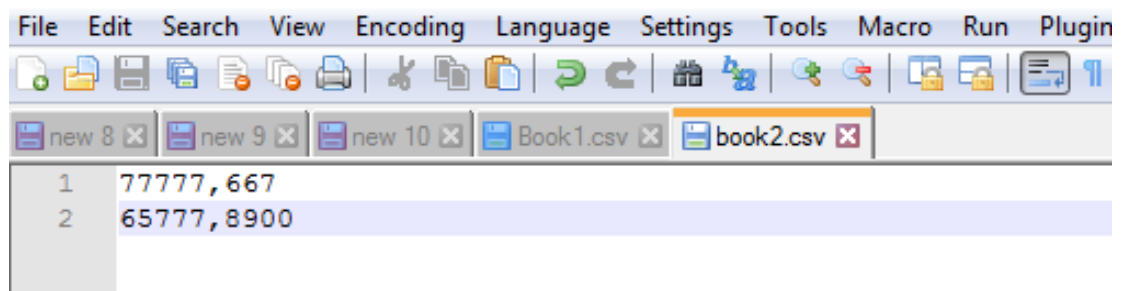
- Adding users in thread group:
 1. Right click on Test Plan.
 2. Click on Add -> Threads -> Thread Group.



3. Set Number of threads to 1.
4. Set Ramp-Up Period to 1.
5. Set Loop Count to 3.

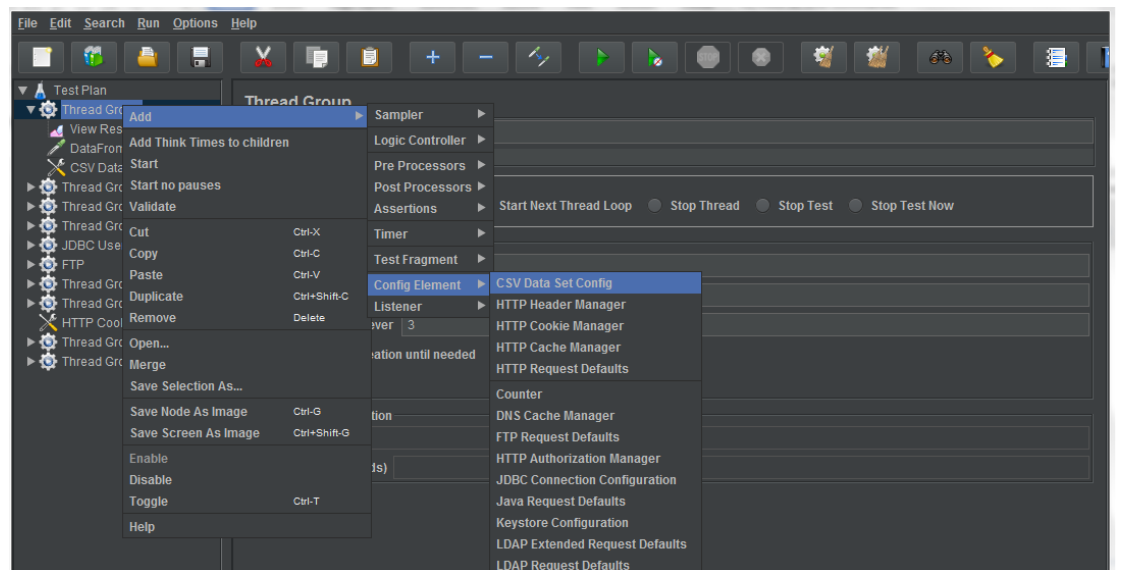


6. Imagine we have a list of random variables that are already generated and saved as "book2.csv".

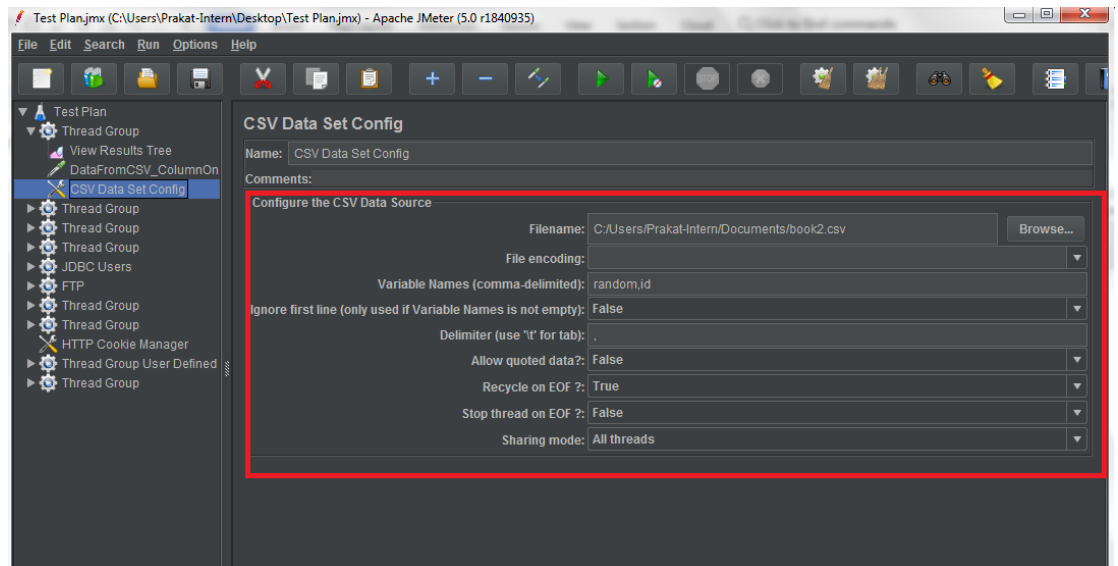


- Now, we want to read this list in our request. So, we have to add a CSV data set Config as we show below:

- Adding CSV Data Set Config:
 1. Right click on Thread Group.
 2. Click on Add -> Config Element -> CSV Data Set Config.

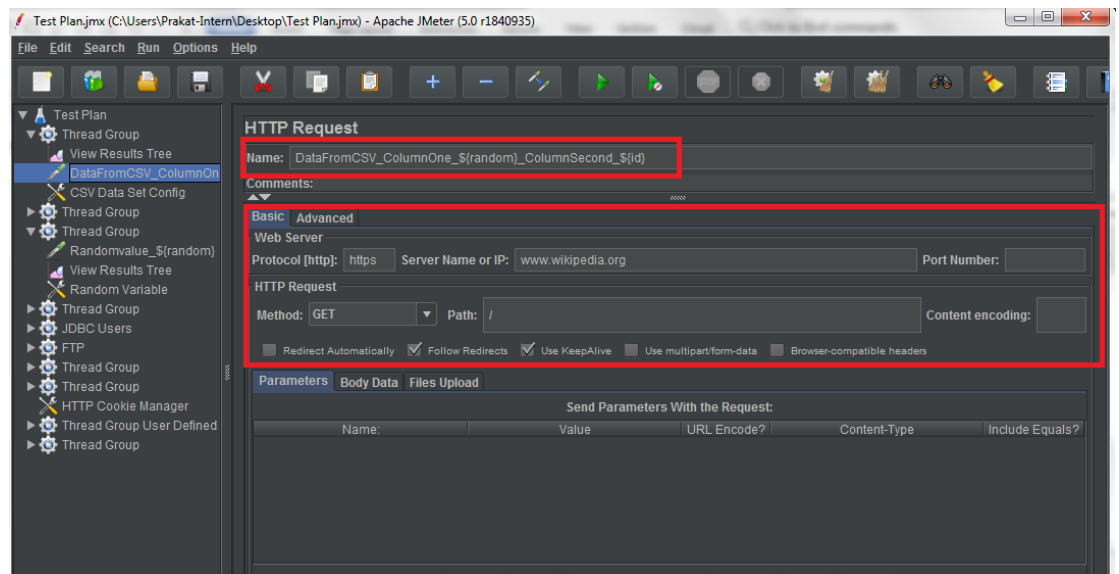


3. Set the CSV Data Set Config as follows:

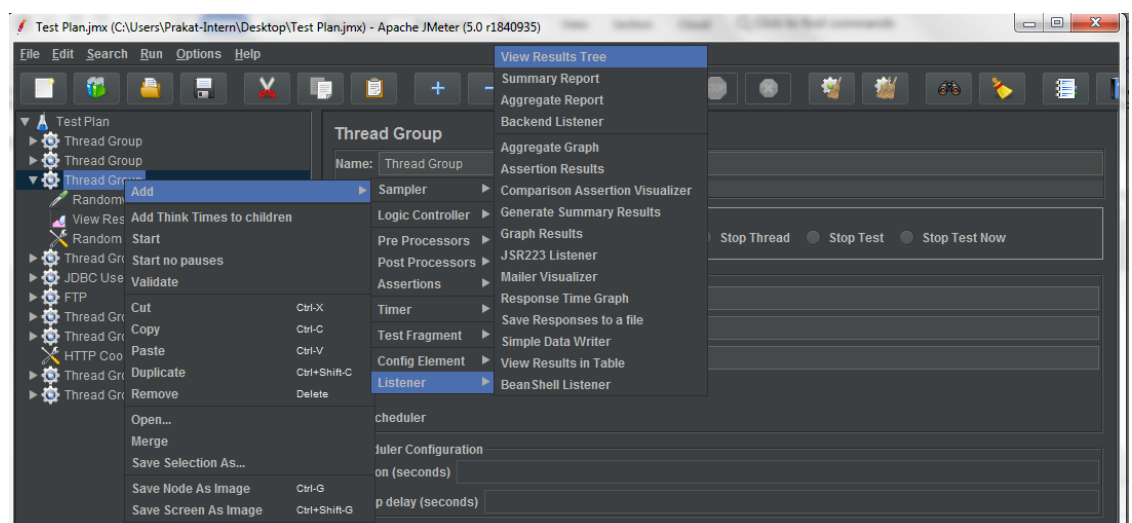


- i. **Filename:** If your file is in the /bin directory, this can just be the filename. If it's somewhere else, use the full path to the file.
 - ii. **Variable Names:** Comma-delimited list of names should be equal number of names than columns in CSV file, example: random, id.
 - iii. **Ignore First Line:** Set as False by default. Used whenever you want to ignore the first line as value (because it may contain the column names).
 - iv. **Delimiter:** A comma is the default delimiter.
 - v. **Allow Quoted Data:** Set to False by default, set to True if you have double quotes in CSV columns.
 - vi. **Recycle on EOF:** True by default, JMeter restarts from the beginning when the End Of File is reached.
 - vii. **Stop Thread on EOF:** Stops the current thread gathering the value if EOF is reached
 - viii. **Sharing Mode:** Defines how values are distributed among concurrent threads
- Adding HTTP Request:
 1. Right click on Thread Group.
 2. Click on Add -> Sampler -> HTTP Request.

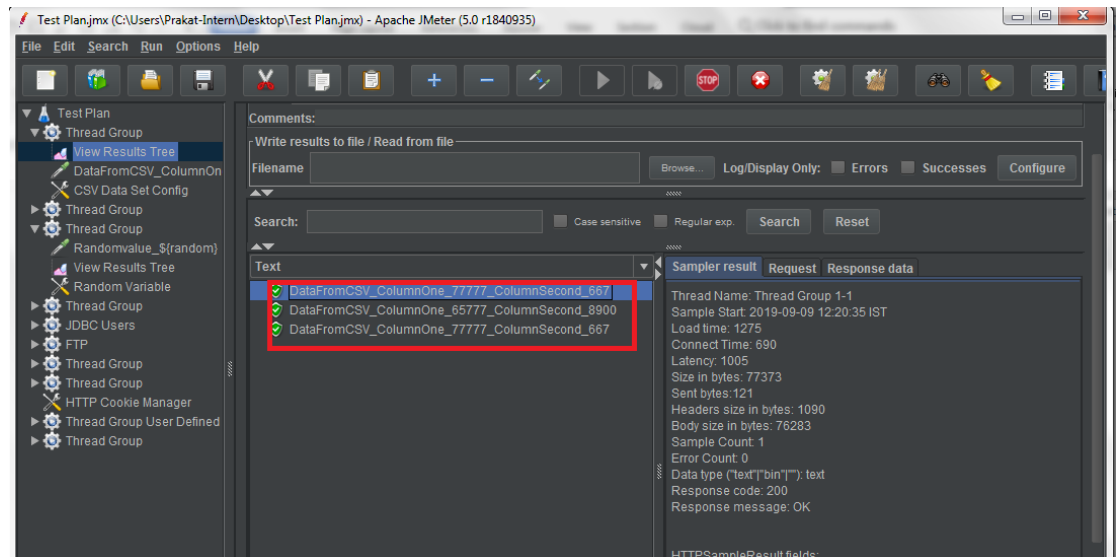
3. Rename HTTP Request to
“DataFromCSV_ColumnOne_\${random}_ColumnSecond_\${id}” , where
random, id are variable names from CSV Data Set Config.
4. Give protocol as “https”.
5. Give server name as “www.wikipedia.org”.
6. Give path “/”.



- Adding View Results Tree:
 1. Right click on Thread Group.
 2. Click on Add -> Listener -> View Results Tree.



3. Run Thread Group and open View Results Tree to see the output.



Note: We can see the values defined in the CSV file.

Step 1.29.4: 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
```

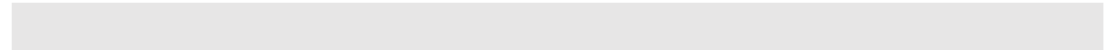
Add all the files to your git repository using the following command:

```
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
```