Load test by creating multiple session by Jmeter

Jumaina Rahman Neen

Associate SQA Engineer

Load testing is the process of putting the load through (HTTP, HTTPS, WebSocket etc) calls on any software system to determine its behavior under normal and high load conditions. Load test helps identify maximum requests a software system can handle.

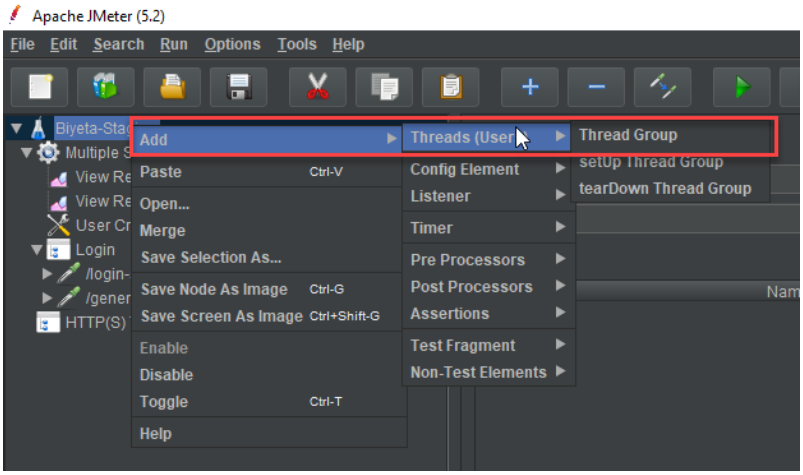
In this document, You will learn about how we can perform load test by creating multiple session by Apache Jmeter. In this learning process following topics will be discussed as well:

1. Basics of creating and a load test
2. Basics of recording scripts
3. Filtering of the recorded scripts
4. Process to read data from CSV file
5. Parameterization

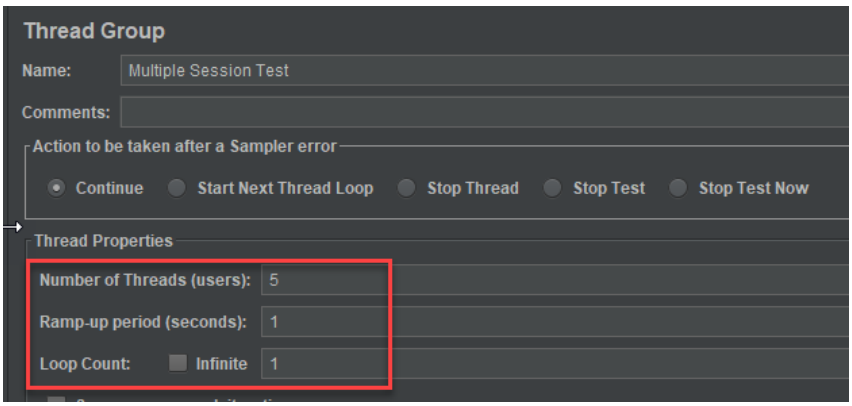
Step 1) Add Thread Group

- Start JMeter
- Select Test Plan on the tree
- Add Thread Group

Right click on the Test Plan and add a new thread group: Add -> Threads (Users) -> Thread Group



In the Thread Group control panel, enter Thread Properties as follows:



- **Number of Threads**: 5 (Number of users connects to the target website: 5)
- **Loop Count**: 1 (Number of time to execute testing)
- **Ramp-Up Period**: 1

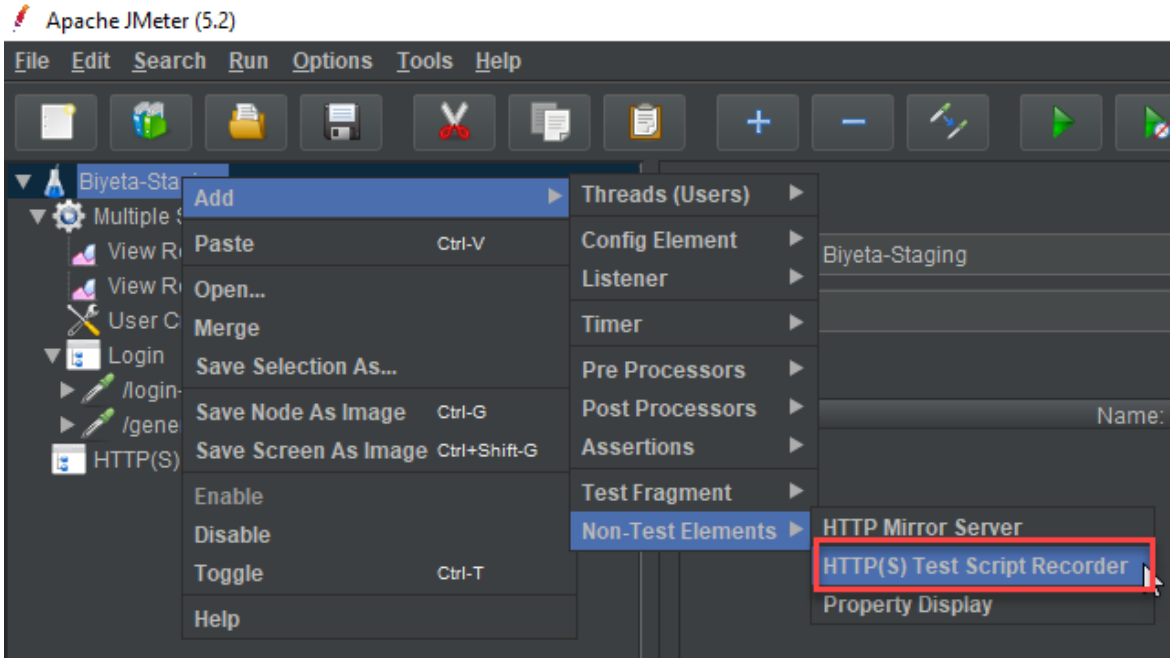
The Thread Count and The Loop Counts are different. Ramp-Up Period tells JMeter how long to delay before starting the next user. For example, if we have 100 users and a 100-second Ramp-Up period, then the delay between starting users would be 1 second (100 seconds /100 users)

Step 2) Adding JMeter elements

Now you need to determine what JMeter elements are required in this test. The elements are:

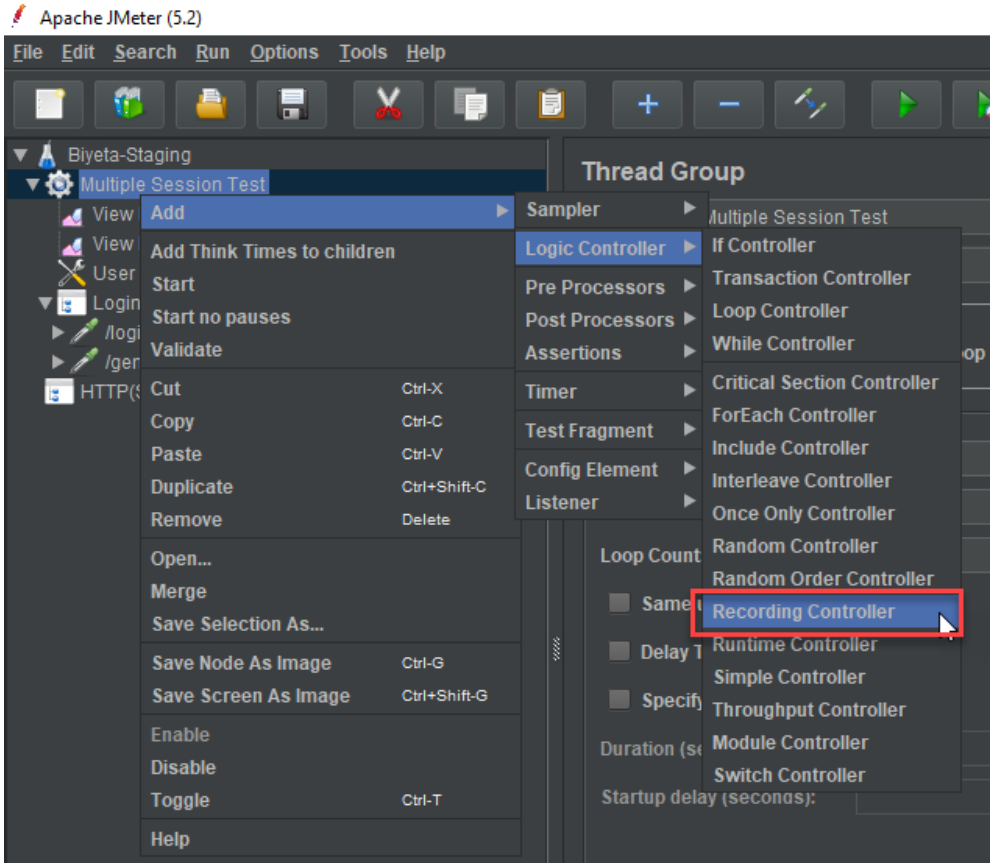
HTTP(S) Test Script Recorder

Right click on the Test Plan and add a new HTTP(S) Test Script Recorder: Add -> Non-Test Elements -> HTTP(S) Test Script Recorder. With the help of this element you will be able to record scripts.



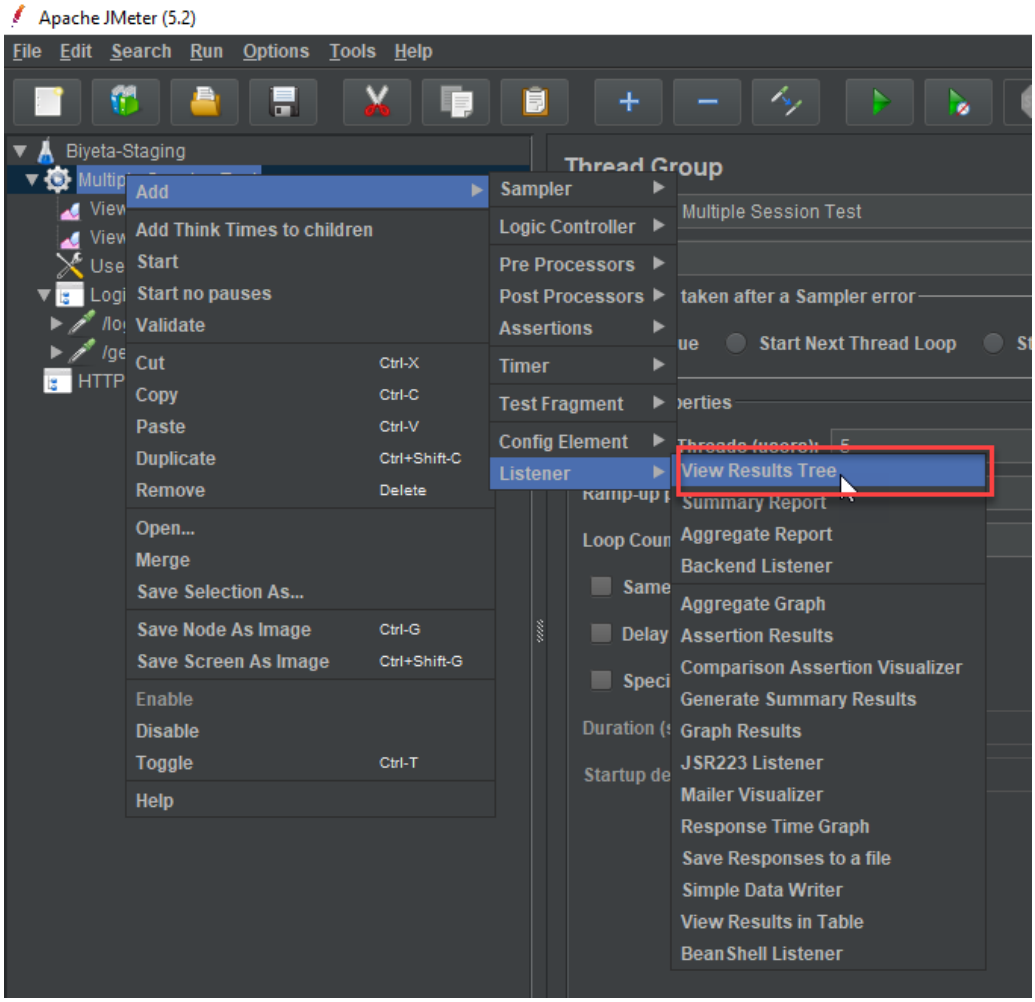
Recording Controller

Right click on the Thread Group and add a new Recording Controller: Add -> Logic Controller -> HTTP(S) Recordisng Controller. With the help of this element you will be able to categorize the recorded scripts based on module. Here we will create a Recording controller called “Login”.



View Results Tree

Right click on the Thread Group and add a new View Results Tree: Add -> Listener -> View Results Tree. With the help of this element you will be able to see the test report.

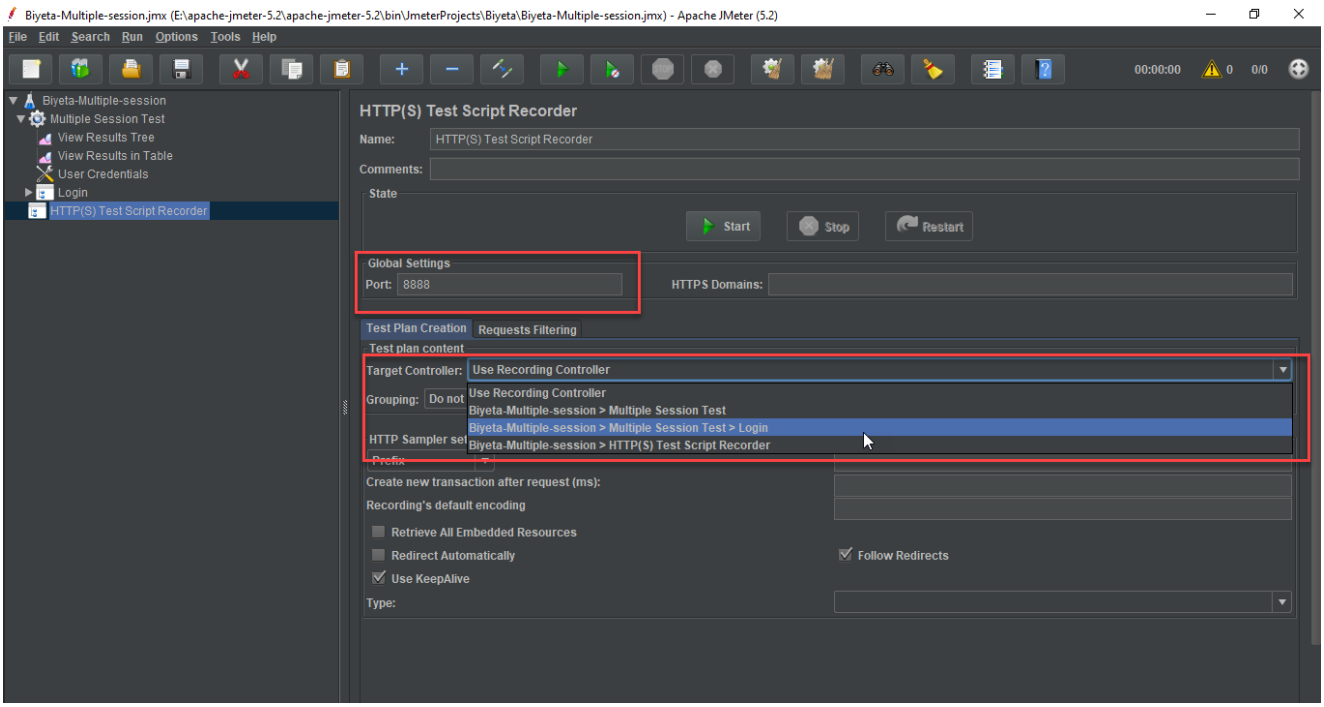


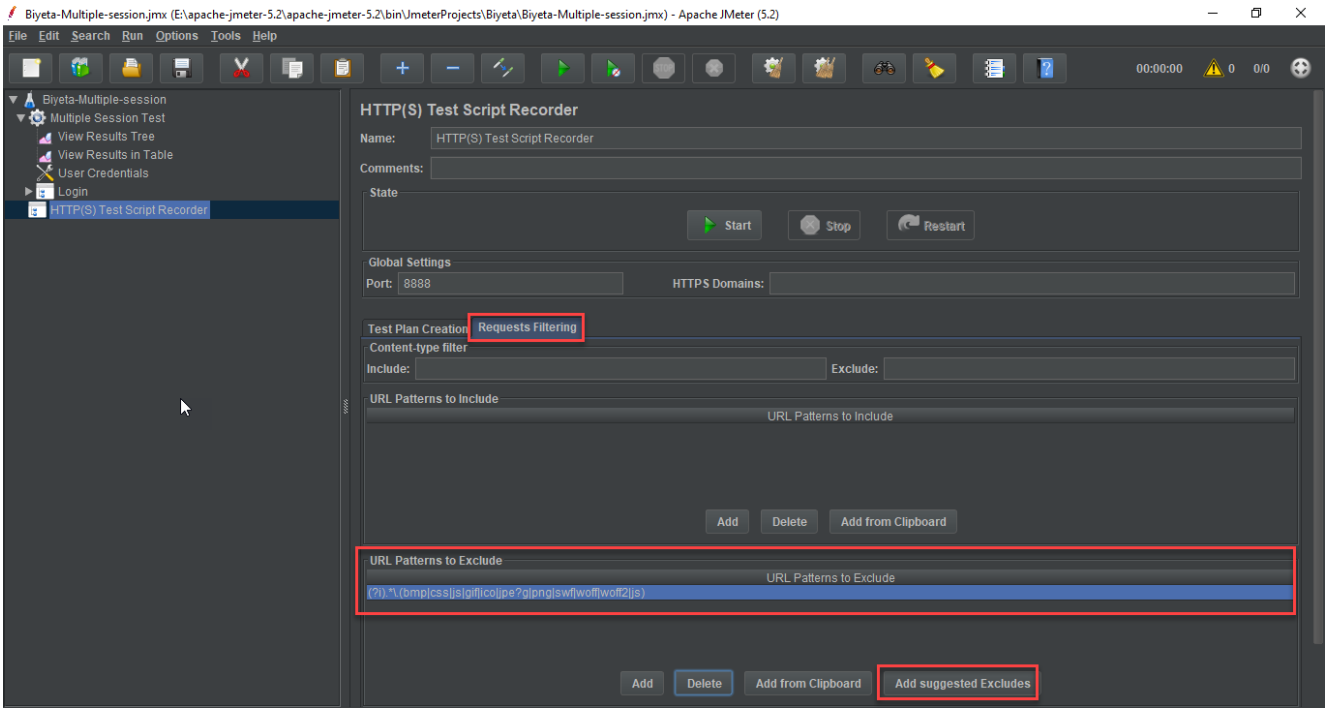
Step 3) Configurations before recording the test

Before starting the recording, You need to setup a few things.

Setup in HTTP(S) Test Script Recorder

- Set the same port number in HTTP(S) Test Script Recorder and the browser's proxy settings. We are using Mozilla Firefox for this test.
- In the HTTP(S) Test Script Recorder, set the port number. We used 8888
- Now in the Test Plan Creation tab, select your desired Target Controller.
- Switch to Request Filtering tab and navigate to URL Patterns to Exclude section. Click on Add suggested Excludes and it will add a few patterns which will help to filter out some unnecessary requests while recording.

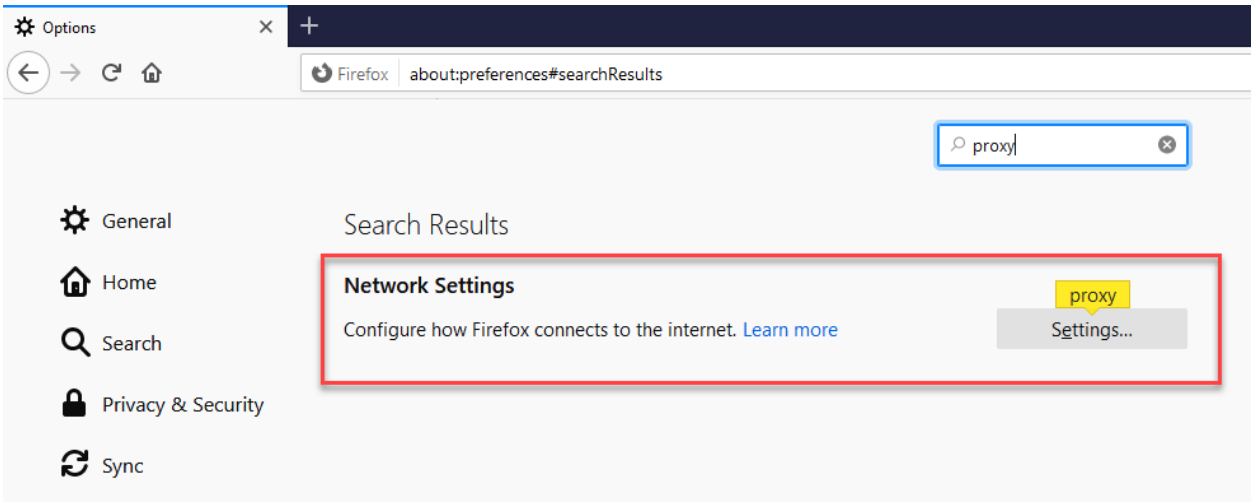


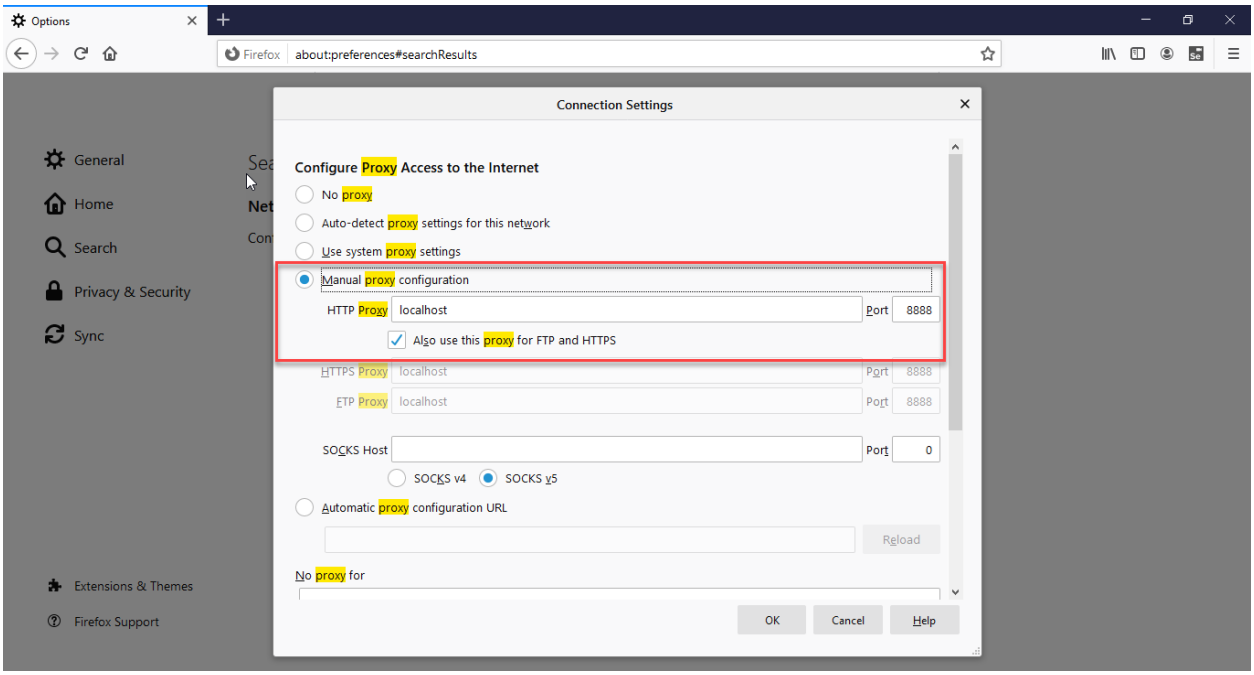


Setup in Browser's proxy settings

- To set port number in browser's proxy settings, go to Options -> Network Settings -> Open Settings pop up
- Select Manual proxy configuration
- Input "localhost" for HTTP Proxy

- Input the same port number that you inputted in for HTTP(S) Test Script Recorder in the Port field. In this test we kept it 8888.
- Mark the checkbox for “Also use this proxy for FTP and HTTPS”
- Click Ok



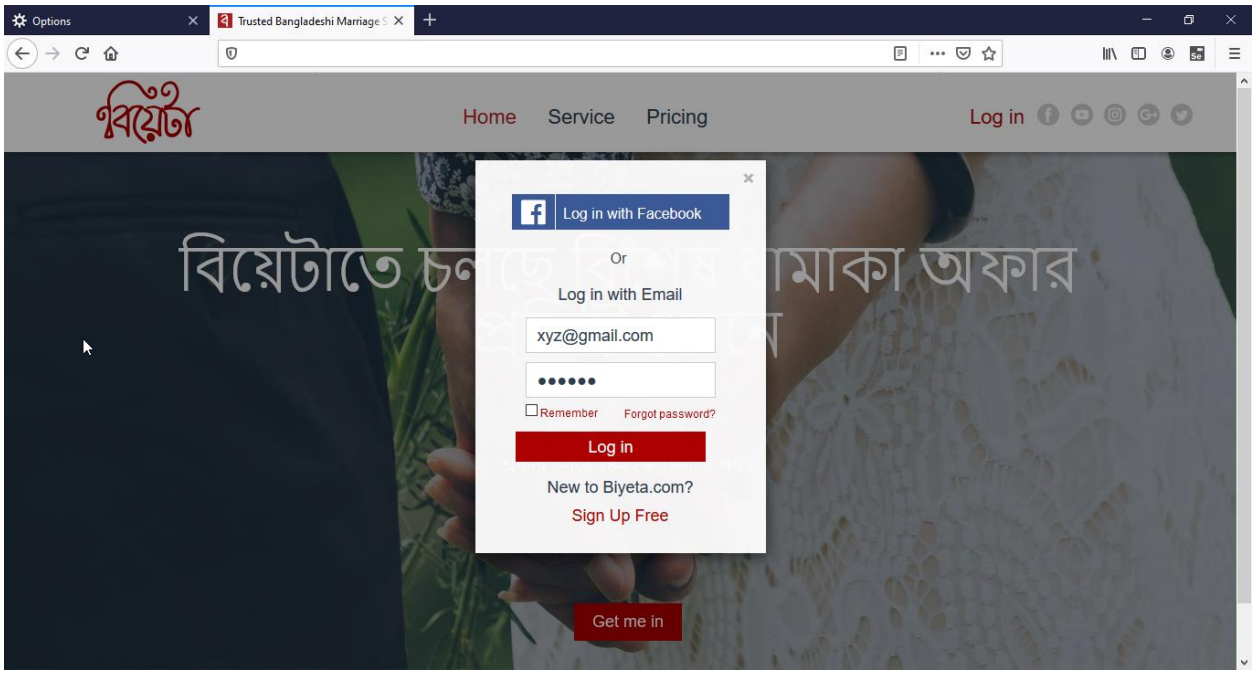


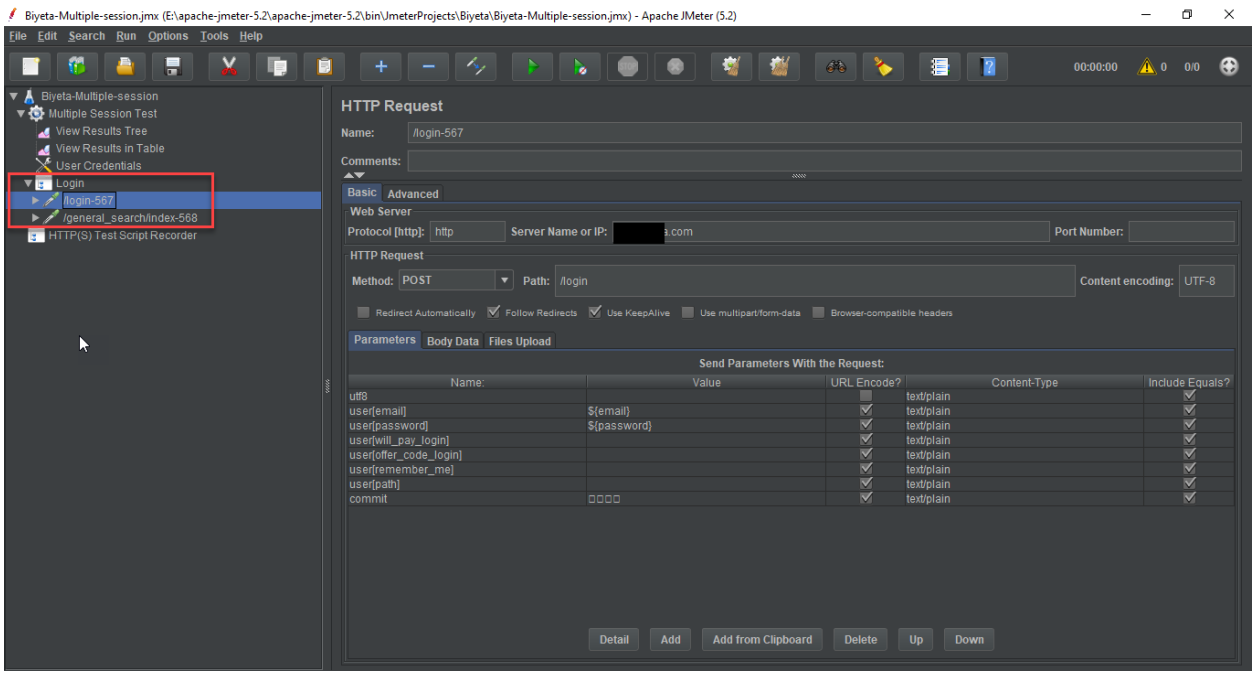
Step 4) Start the Recording

- Click on the Start button in HTTP(S) Test Script Recorder. A pop up for root certificate will appear. Click Ok and your Jmeter will start recording requests on the browser.
- Go to the Firefox browser and hit the desired URL of the site you want to test. We will be recording a login test.

N.B. Make sure before starting the record, your proxy settings are correctly done. Otherwise either you will not be able to access the website or nothing will be recorded on Jmeter.

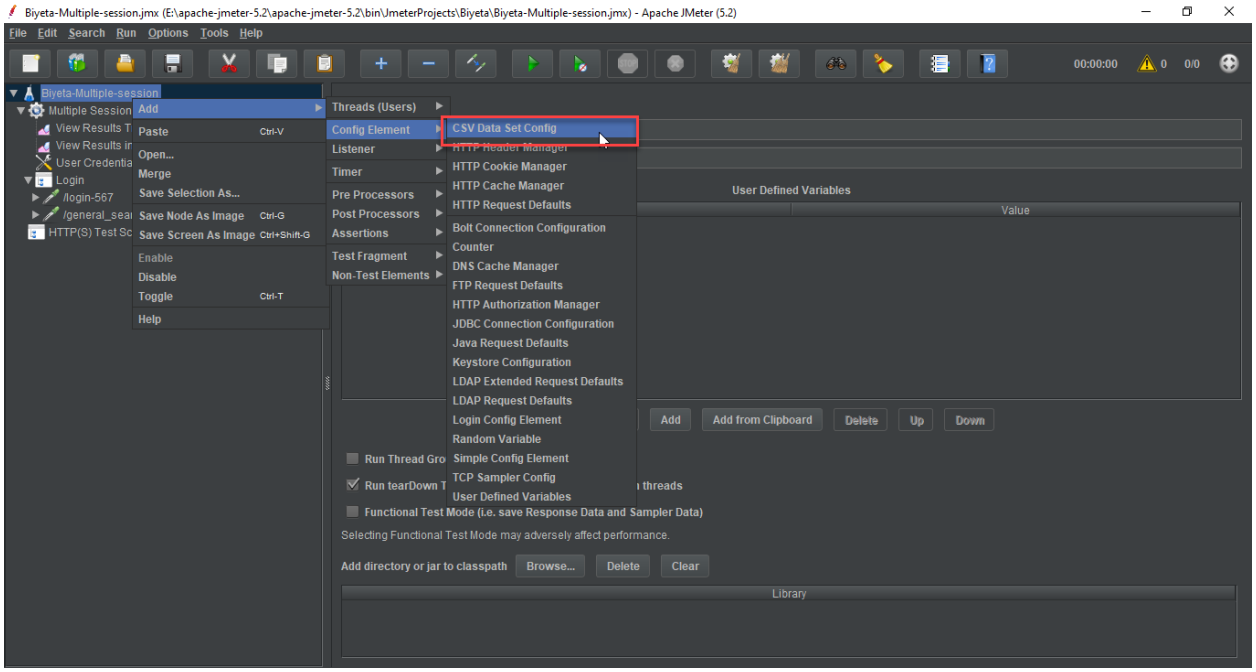
- Now perform the login operation on the website.
- Observe HTTP Requests will be captured on the Recording Controller based on each action performed on the website.
- Stop the recording by clicking on the Stop button on in HTTP(S) Test Script Recorder.





Step 5) Add CSV file to the Test Plan

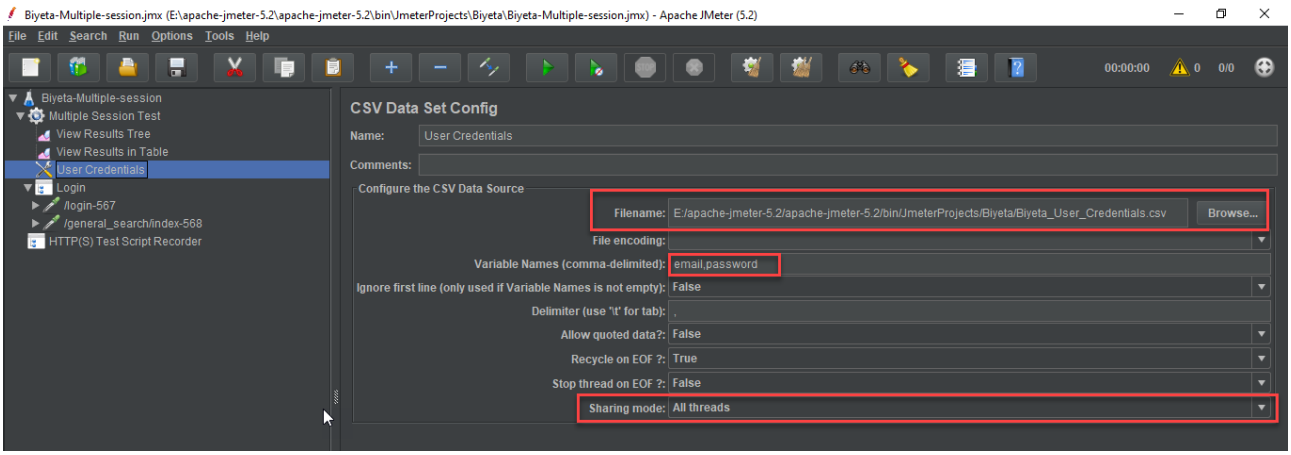
- Right click on the Test Plan and add a new CSV Data Set Config: Add -> Config Elements -> CSV Data Set Config



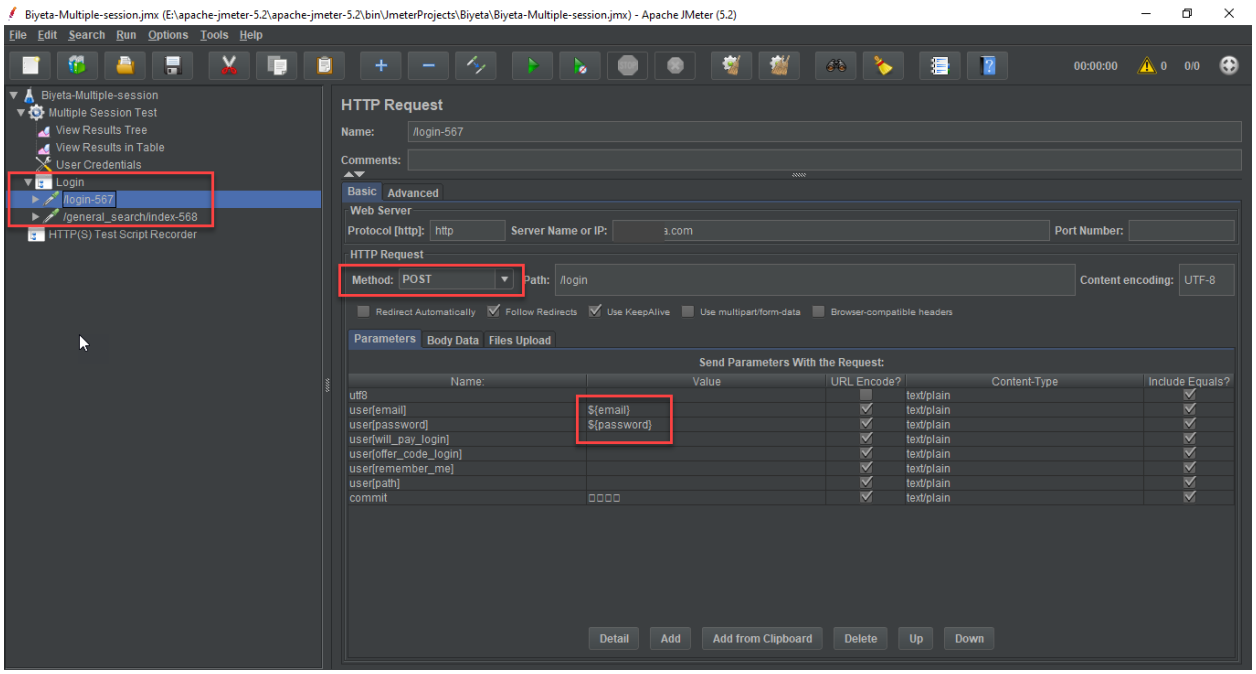
- Browse and upload the CSV file.

N.B. **CSV file creation process:** List down different email address in first column and their corresponding passwords on the second column. No need of column header.

- Input variable names for the columns on the CSV. Make sure to use the variable names on other places as it is provided here.
- Keep Sharing Mode to All Threads.



- Now look for the POST type HTTP request on the Recording Controller.
- Observe in the parameters tab, User and Password table contains the value you entered while recording the test.
- Replace the value with the variables created on CSV Data Set Config.
- Format for declaring variables on Jmeter -> ${variable_name}. Variable name should be same as you declared on the CSV data Set Config.

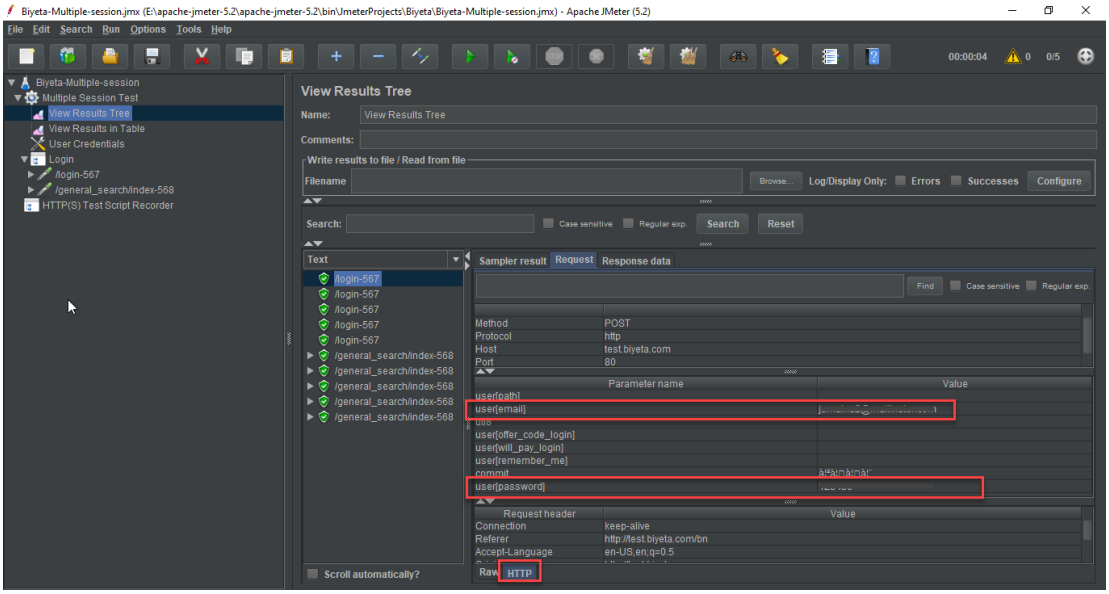


Now you are done with all the setup. Once you run this Test plan, you will be able to see the test result on View Results Tree and View Results Table.

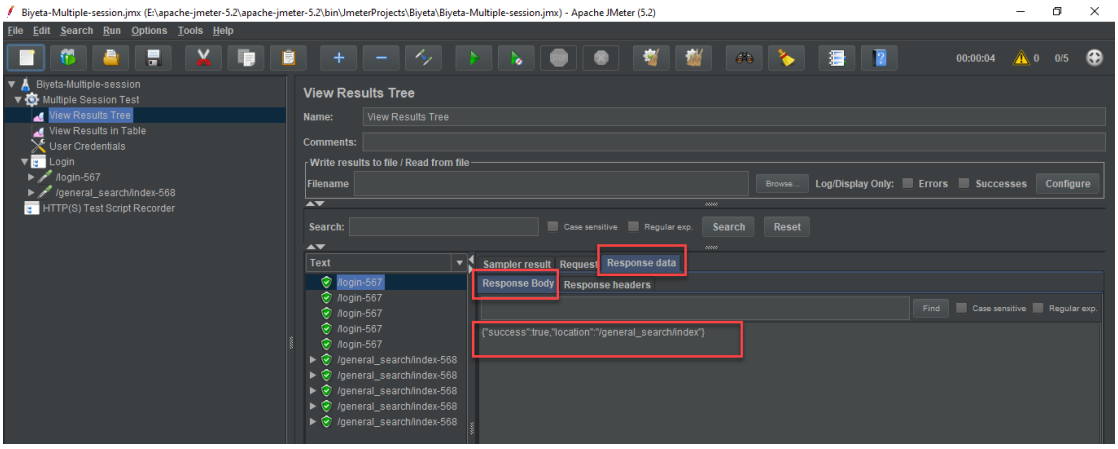
Step 6) Validate the test result

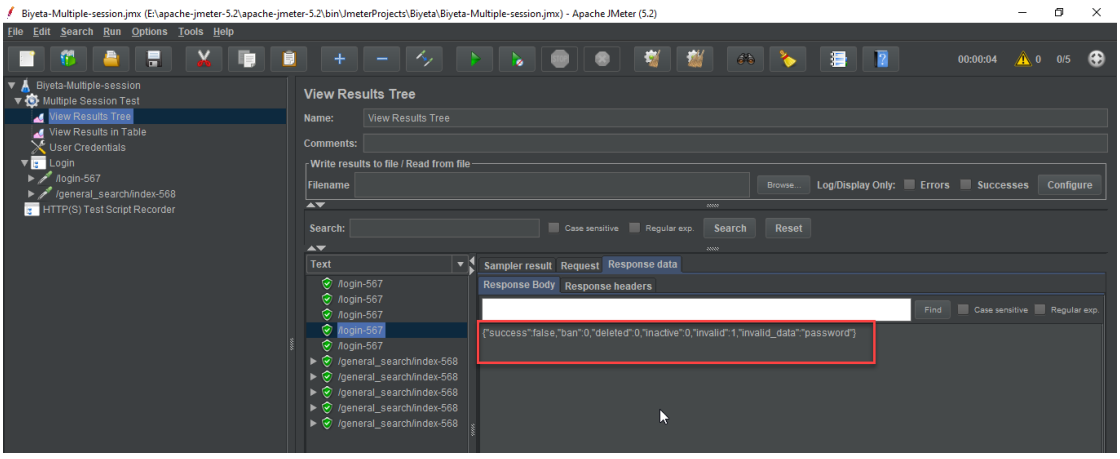
To validate if we are actually able to read data from the CSV file and if actually different user sessions are logging in to the system, we can check the following places –

- Run the test by clicking on the Start button.
- Now go to View Results Tree > find the Login HTTP requests
- Navigate to Request Tab, select format HTTP
- Observe all the Login requests took data from the CSV file sequentially. For example, first login request takes data from first row of the CSV, second login request takes data from second row and it continues this way.

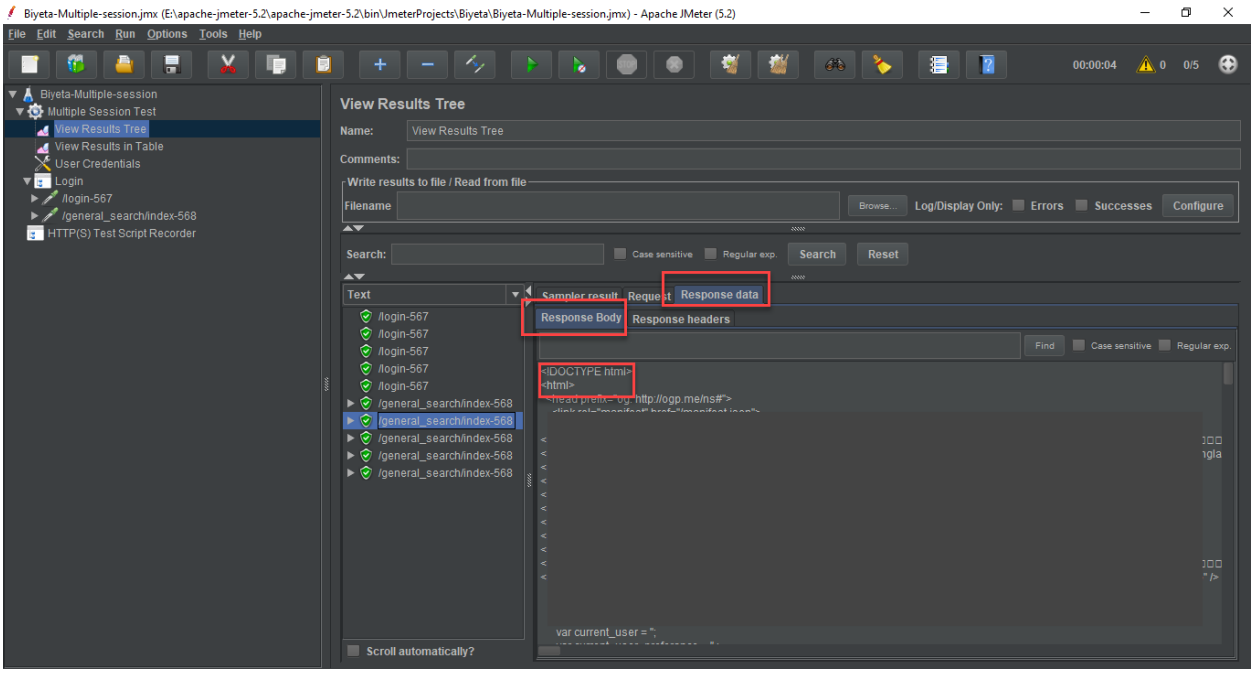


- Now navigate to the response tab, then select response body.
- Observe server authenticates the login and provides success message True as response. If login is not authenticated, it will give Success message as False.





- Now look for the Index page requests and observe all requests have response of the html code of Index page if login was success else html code of Login page.



- Now look at all the Response Header tab of the Index page request, there is a field called Set-Cookie which contains sessions for each user logged in to the system.
- Observe this session is different for each authenticated user.